Intelligent Recipe Advisor

Rosemary Josekutty Thomas
H00172653

August 2014

Computer Science
School of Mathematical and Computer Sciences

Dissertation submitted as part of the requirements for the award of the degree of MSc in Information Technology (Business)
Abstract

The purpose is to develop a web application, to be mainly utilised by people diagnosed of common lifestyle diseases that uses character recognition to identify ingredients from the grocery bills. The prime objective of this project will be providing an application that will advise recipes based on the user’s account profile. The application allows a user to upload grocery bill images and it then processes to identify the ingredients using optical character recognition. The users can modify the identified ingredients and add other preferences to their profile if required. The application retrieves recipe suggestions from an application programming interface. It encourages users with common lifestyle diseases to follow healthy diets by making recipe suggestions effortless.
Acknowledgements

First, I would like to thank God Almighty for His constant inspiration and support.

Then I would like to thank my supervisor Professor Talal Shaikh for his continuous assistance, support, suggestions and advices during the phase of this research report.

Additionally I would like to thank Dr. Hind Zantout for her tips and advices provided during the research methods and planning lectures.

I would also like to thank my family especially my Dad and Mom; and my close friends especially Asha Nair dietician, who were always there for me and supported me in pursuing this research report.

This research report is dedicated to all of them.
Declaration

I, Rosemary Josekutty Thomas, confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: ...................

Date: 21st August 2014
# Table of Contents

Abstract .......................................................................................................................... ii  
Acknowledgements ......................................................................................................... iii  
Declaration ....................................................................................................................... iv  
Table of Contents ........................................................................................................... v  
List of Figures .................................................................................................................... vii  
List of Tables .................................................................................................................... x  

## CHAPTER 1. INTRODUCTION .............................................................................. 1  
1.1. Aims and Objectives .............................................................................................. 1  
1.2. Project Overview .................................................................................................... 1  
1.3. Outline .................................................................................................................... 2  

## CHAPTER 2. LITERATURE REVIEW ................................................................ 3  
2.1. Optical Character Recognition ............................................................................ 4  
  2.1.1. GOCR .............................................................................................................. 4  
  2.1.2. OCRopus ......................................................................................................... 4  
  2.1.3. Tesseract .......................................................................................................... 6  
  2.1.4. Summary .......................................................................................................... 7  
2.2. Databases ............................................................................................................... 8  
  2.2.1. MySQL and MongoDB .................................................................................. 8  
  2.2.2. Summary .......................................................................................................... 8  
2.3. Recipes Application Programming Interfaces ................................................... 9  
  2.3.1. Food2Fork ........................................................................................................ 9  
  2.3.2. Yummly .......................................................................................................... 10  
  2.3.3. Summary .......................................................................................................... 10  
2.4. Health Conditions and Related Foods ................................................................. 11  
  2.4.1. Body Mass Index ............................................................................................ 11  
  2.4.2. Cardiovascular Diseases .............................................................................. 11  
  2.4.3. Obesity ............................................................................................................ 13  
  2.4.4. Diabetes .......................................................................................................... 13  
  2.4.5. Summary .......................................................................................................... 17  
2.5. Web Application .................................................................................................... 18  
  2.5.1. Summary .......................................................................................................... 19  
2.6. Critical Analysis .................................................................................................... 19  
  2.6.1. Comparative Analysis .................................................................................... 19  
  2.6.2. Analysis of Issues .......................................................................................... 20  
2.7. Evaluation ............................................................................................................. 21  
  2.7.1. Types of Evaluation ....................................................................................... 21  
  2.7.2. Analysis Method ............................................................................................. 22  
  2.7.3. Motivation Factors ......................................................................................... 23  
  2.7.4. Evaluation Criteria ......................................................................................... 24  

## CHAPTER 3. PROFESSIONAL, LEGAL, ETHICAL AND SOCIAL ISSUES ............. 25  

## CHAPTER 4. REQUIREMENT ANALYSIS ......................................................... 27  
4.1. Mandatory Requirements ..................................................................................... 28  
  4.1.1. Functional Requirements .............................................................................. 28  
  4.1.2. Non-functional Requirements ...................................................................... 28  
4.3. Optional Requirements ........................................................................................ 29
CHAPTER 5. PROJECT PLAN .............................................. 31
  5.1. Methodology ....................................................... 31
  5.2. Task Analysis ..................................................... 32
  5.3. Assessment Plan ................................................... 33
  5.4. Risk Identification and Management ........................... 34

CHAPTER 6. SYSTEM DESIGN ........................................... 35
  6.1. Unified Modelling Language Design ........................... 35
    6.1.1. Use Case Analysis ......................................... 35
    6.1.2. Activity Diagram .......................................... 36
    6.1.3. Class Diagram ............................................. 38
  6.2. Database Design .................................................. 39
  6.3. User Interface Design .......................................... 41

CHAPTER 7. SYSTEM IMPLEMENTATION .............................. 45
  7.1. Coding Languages and Tools ................................... 45
  7.2. Implementation ................................................. 45
    7.2.1. Building of Functions in the Classes ..................... 45
    7.2.2. Challenges ................................................ 64
    7.2.3. Building Modules ......................................... 65
      7.2.3.1. Upload and Recognition Module ....................... 65
      7.2.3.2. Shopping Basket Module ............................. 67
      7.2.3.3. User Profile Module ................................ 68
      7.2.3.4. Recipe Suggestions Module ......................... 69

CHAPTER 8. TESTING ................................................... 72
  8.1. Unit Testing ..................................................... 72
  8.2. Functionality Testing ......................................... 72

CHAPTER 9. USER EVALUATION ....................................... 75

CHAPTER 10. CONCLUSION ............................................. 81
  10.1. Limitations ..................................................... 81
  10.2. Accomplishments .............................................. 81
  10.3. Future Development ........................................... 82
  10.4. Summary ........................................................ 82

REFERENCES .......................................................... 83

BIBLIOGRAPHY .......................................................... 90

APPENDICES .............................................................. I
  Appendix A ............................................................. I
    A.1. Database Creation ............................................. I
  Appendix B ............................................................. VIII
    B.1. Project Plan ................................................... VIII
  Appendix C ............................................................. XI
    C.1. Wireframes ..................................................... XI
  Appendix D ............................................................. XXI
    D.1. Header and Footer Module .................................. XXI
    D.2. Sign Up Module ............................................... XXIII
    D.3. Sign In Module ................................................ XXIII
    D.4. Logout Module .................................................. XXIII
    D.5. Password Reset Module ...................................... XXIV
  Appendix E ............................................................. XXV
    E.1. Source Code .................................................... XXV
List of Figures

Figure 1. Working of Intelligent Recipe Advisor .......................................................... 3
Figure 2. Flowchart of OCRopus system ........................................................................ 6
Figure 3. Flowchart of Tesseract system ........................................................................ 7
Figure 4. Waterfall model ......................................................................................... 31
Figure 5. Use case diagram of Intelligent Recipe Advisor ........................................... 36
Figure 6. Activity diagram of Intelligent Recipe Advisor ............................................ 37
Figure 7. Class diagram of Intelligent Recipe Advisor ................................................ 39
Figure 8. Initial collection diagram of database ....................................................... 40
Figure 9. Final collections diagram of database ..................................................... 41
Figure 10. Logo of Intelligent Recipe Advisor .......................................................... 41
Figure 11. Icon of Intelligent Recipe Advisor ............................................................ 41
Figure 12. Shopping basket page .............................................................................. 42
Figure 13. Profile page ........................................................................................... 43
Figure 14. View recipes page .................................................................................. 44
Figure 15. getConnection() .................................................................................... 46
Figure 16. checkIdle() ........................................................................................... 46
Figure 17. hashCryptPassword() ............................................................................. 47
Figure 18. registerAccount() .................................................................................. 47
Figure 19. registerAccount() .................................................................................. 48
Figure 20. registerAccount() .................................................................................. 48
Figure 21. getUserPassword() ................................................................................ 49
Figure 22. checkEmail() ....................................................................................... 50
Figure 23. updateAccount() .................................................................................. 50
Figure 24. updateAccount() .................................................................................. 51
Figure 25. updateAccount() .................................................................................. 51
Figure 26. updateAccount() .................................................................................. 52
Figure 27. updateIngredients() .............................................................................. 52
Figure 28. resetPassword() ................................................................................... 53
Figure 29. deleteAccount() ................................................................................... 54
Figure 30. signIn() ............................................................................................... 54
Figure 31. account() ............................................................................................ 54
Figure 32. upload() .............................................................................................. 55
Figure 33. basket() .............................................................................................. 55
Figure 34. getToken() ........................................................................................... 56
Figure 35. checkToken() ....................................................................................... 56
Figure 36. writeToken() ....................................................................................... 57
Figure 37. displayHeight() .................................................................................... 57
Figure 38. displayWeight() ................................................................................... 58
Figure 39. displayAllergy() ................................................................. 58
Figure 40. displayCuisine() ................................................................. 58
Figure 41. displayDiet() ................................................................. 59
Figure 42. displayIngredient() .................................................. 59
Figure 43. displayBasket() .................................................. 59
Figure 44. displayCourse() ................................................................. 60
Figure 45. getProfile() ................................................................. 60
Figure 46. getProfile() ................................................................. 61
Figure 47. getCriteria() ................................................................. 62
Figure 48. getCriteria() ................................................................. 62
Figure 49. getCriteria() ................................................................. 63
Figure 50. getCriteria() ................................................................. 63
Figure 51. getCriteria() ................................................................. 64
Figure 52. read.php ................................................................. 66
Figure 53. basket.php ................................................................. 67
Figure 54. basket.php ................................................................. 67
Figure 55. basket.php ................................................................. 67
Figure 56. update.php ................................................................. 67
Figure 57. account.php ................................................................. 68
Figure 58. account.php ................................................................. 68
Figure 59. account.php ................................................................. 68
Figure 60. account.php ................................................................. 68
Figure 61. account.php ................................................................. 69
Figure 62. account.php ................................................................. 69
Figure 63. account.php ................................................................. 69
Figure 64. account.php ................................................................. 69
Figure 65. account.php ................................................................. 69
Figure 66. save.php ................................................................. 69
Figure 67. suggestions.php .................................................. 69
Figure 68. suggestions.php .................................................. 70
Figure 69. suggestions.php .................................................. 70
Figure 70. suggestions.php .................................................. 71
Figure 71. suggestions.php .................................................. 71
Figure 72. Summary of the test cases .................................................. 72
Figure 73. Do you usually use an application for recipe suggestions? .......... 77
Figure 74. Would you like any other health profiles to be included in the application? ................................................................. 77
Figure 75. Totally, how many times did you use the upload bill feature? .......... 78
Figure 76. How you rate the accuracy of recognised ingredients from the bills? 78
Figure 77. How do you rate the design of the application in relation to layout, colour and graphics? ................................................................. 79
Figure 78. Do you find the application easy to use? .................................. 79
Figure 79. How do you rate the recipe suggestions? .................................. 79
Figure 80. Do you find the idea of this application to be innovative and interesting? ................................................................. 80
Figure 81. Would you be keen in using this application for your daily/weekly recipe suggestions? ................................................................. 80
Figure 82. Connection tab ................................................................ 1
Figure 83. Authentication tab ................................................................. II
Figure 84. Connection success alert ................................................................. II
Figure 85. MongoDB selected from connections window ........................................ III
Figure 86. Create Database selected from menu of MongoDB connection ........ III
Figure 87. Create IntelliDelight database .......................................................... IV
Figure 88. Collections menu ........................................................................ IV
Figure 89. Create members collection in IntelliDelight database ............... V
Figure 90. Create ingredients collection in IntelliDelight database .......... V
Figure 91. Menu for ingredients collection ................................................... V
Figure 92. Insert Document window for ingredients collection ................. VI
Figure 93. Validation success alert ................................................................. VI
Figure 94. Documents in ingredients collection ....................................... VII
Figure 95. Disconnect selected from menu of MongoDB connection ........ VII
Figure 96. Gantt chart ................................................................................ VIII
Figure 97. Risk matrix (Probability-Impact) ................................................ IX
Figure 98. MongoDB connection error message ......................................... XI
Figure 99. Home page ................................................................................. XI
Figure 100. Sign up page .............................................................................. XII
Figure 101. Registration details page .......................................................... XII
Figure 102. User exists message ............................................................... XIII
Figure 103. Sign in page .............................................................................. XIII
Figure 104. Incorrect username or password message ............................. XIV
Figure 105. Upload page .............................................................................. XIV
Figure 106. Alert message during upload .................................................. XV
Figure 107. Alert message during update .................................................. XV
Figure 108. About us page .......................................................................... XVI
Figure 109. Disclaimer page ........................................................................ XVI
Figure 110. Profile details page .................................................................... XVII
Figure 111. No recipes found message ........................................................ XVII
Figure 112. Account deleted message ........................................................ XVIII
Figure 113. Contact page ............................................................................. XVIII
Figure 114. Thank you message .................................................................. XIX
Figure 115. Forgot password page ............................................................. XIX
Figure 116. Password reset confirmation ................................................... XX
Figure 117. User not registered message ...................................................... XX
Figure 118. How it works page ..................................................................... XXI
Figure 119. Header.php - Link tags .............................................................. XXI
Figure 120. Header.php - Navigation bar .................................................... XXII
Figure 121. Header.php - Banner ................................................................. XXII
Figure 122. Header.php - Script tags ............................................................ XXII
Figure 123. footer.php ................................................................................ XXII
Figure 124. register.php .............................................................................. XXIII
Figure 125. verify.php ................................................................................ XXIII
Figure 126. logout.php ................................................................................ XXIV
Figure 127. send.php .................................................................................. XXIV
Figure 128. rest.php ..................................................................................... XXV
Figure 129. reset.php .................................................................................. XXV
Figure 130. reset.php .................................................................................. XXVI
Figure 131. RestPass.php ........................................................................... XXV
Figure 132. Sample grocery bill image ....................................................... XL
Figure 133. Upload and analysis of grocery bill ........................................... XLI
Figure 134. Identified ingredients ............................................................... XLI

Intelligent Recipe Advisor

Rosemary Josekutty Thomas

H00172653
Table 1. Fatty acid contents of nuts and oils ......................................................... 12
Table 2. Glycaemic index and composition of various foods ................................... 14
Table 3. Effect of processing on the glycaemic index and total dietary fibre of foods ....................................................................................................................... 14
Table 4. Food group composition of the Therapeutic Lifestyle Change diet ........... 16
Table 5. Low-calorie diet .......................................................................................... 16
Table 6. Nutrient composition of the Therapeutic Lifestyle Change diet ............... 16
Table 7. Comparison of above-mentioned applications/tools ................................... 23
Table 8. getConnection() .......................................................................................... 45
Table 9. checkId() ..................................................................................................... 46
Table 10. hashCryptPassword() .............................................................................. 46
Table 11. registerAccount() ..................................................................................... 47
Table 12. getUserPassword() ................................................................................... 49
Table 13. checkEmail() ............................................................................................ 49
Table 14. updateAccount() ....................................................................................... 50
Table 15. updateIngredients() ............................................................................... 52
Table 16. resetPassword() ....................................................................................... 53
Table 17. deleteAccount() ....................................................................................... 53
Table 18. signIn() ..................................................................................................... 54
Table 19. account() .................................................................................................. 54
Table 20. upload() .................................................................................................... 55
Table 21. basket() ..................................................................................................... 55
Table 22. getToken() ................................................................................................ 55
Table 23. checkToken() ........................................................................................... 56
Table 24. writeToken() ............................................................................................ 57
Table 25. displayHeight() ....................................................................................... 57
Table 26. displayWeight() ....................................................................................... 57
Table 27. displayAllergy() ....................................................................................... 58
Table 28. displayCuisine() ....................................................................................... 58
Table 29. displayDiet() ............................................................................................ 58
Table 30. displayIngredient() .................................................................................. 59
Table 31. displayBasket() ....................................................................................... 59
Table 32. displayCourse() ....................................................................................... 60
Table 33. getProfile() ............................................................................................... 60
Table 34. read.php .................................................................................................... 65
Table 35. basket.php ................................................................................................ 67
Table 36. update.php ............................................................................................... 67

List of Tables
Table 37. save.php ................................................................. 69
Table 38. displayRecipe() ............................................................ 70
Table 39. Test ID 62 .................................................................. 73
Table 40. Test ID 63 ................................................................. 73
Table 41. Test ID 64 ................................................................. 73
Table 42. Test ID 65 ................................................................. 74
Table 43. Test ID 66 ................................................................. 74
Table 44. Summary of user evaluation responses ................................... 75
Table 45. Analysis of tasks ................................................................ IX
Table 46. Risk matrix with action plan ................................................... X
Table 47. register.php .................................................................. XXIII
Table 48. verify.php .................................................................. XXIII
Table 49. logout.php .................................................................. XXIII
Table 50. forgot.php .................................................................. XXIV
Table 51. reset.php .................................................................. XXV
Table 52. Reset.php and ResetPass.php ................................................. XXV
Table 53. Test case ID 1 ............................................................... XXVI
Table 54. Test case ID 2 ............................................................... XXVI
Table 55. Test case ID 3 ............................................................... XXVI
Table 56. Test case ID 4 ............................................................... XXVI
Table 57. Test case ID 5 ............................................................... XXVII
Table 58. Test case ID 6 ............................................................... XXVII
Table 59. Test case ID 7 ............................................................... XXVII
Table 60. Test case ID 8 ............................................................... XXVII
Table 61. Test case ID 9 ............................................................... XXVIII
Table 62. Test case ID 10 ............................................................. XXVIII
Table 63. Test case ID 11 ............................................................. XXVIII
Table 64. Test case ID 12 ............................................................. XXVIII
Table 65. Test case ID 13 ............................................................. XXIX
Table 66. Test case ID 14 ............................................................. XXIX
Table 67. Test case ID 15 ............................................................. XXIX
Table 68. Test case ID 16 ............................................................. XXIX
Table 69. Test case ID 17 ............................................................. XXIX
Table 70. Test case ID 18 ............................................................. XXX
Table 71. Test case ID 19 ............................................................. XXX
Table 72. Test case ID 20 ............................................................. XXX
Table 73. Test case ID 21 ............................................................. XXX
Table 74. Test case ID 22 ............................................................. XXXI
Table 75. Test case ID 23 ............................................................. XXXI
Table 76. Test case ID 24 ............................................................. XXXI
Table 77. Test case ID 25 ............................................................. XXXI
Table 78. Test case ID 26 ............................................................. XXXII
Table 79. Test case ID 27 ............................................................. XXXII
Table 80. Test case ID 28 ............................................................. XXXII
Table 81. Test case ID 29 ............................................................. XXXII
Table 82. Test case ID 30 ............................................................. XXXII
Table 83. Test case ID 31 ............................................................. XXXIII
Table 84. Test case ID 32 ............................................................. XXXIII
Table 85. Test case ID 33 ............................................................. XXXIII
Table 86. Test case ID 34 ............................................................. XXXIV
Table 87. Test case ID 35 ................................................................. XXXIV
Table 88. Test case ID 36 ................................................................. XXXIV
Table 89. Test case ID 37 ................................................................. XXXIV
Table 90. Test case ID 38 ................................................................. XXXIV
Table 91. Test case ID 39 ................................................................. XXXV
Table 92. Test case ID 40 ................................................................. XXXV
Table 93. Test case ID 41 ................................................................. XXXV
Table 94. Test case ID 42 ................................................................. XXXV
Table 95. Test case ID 43 ................................................................. XXXV
Table 96. Test case ID 44 ................................................................. XXXVI
Table 97. Test case ID 45 ................................................................. XXXVI
Table 98. Test case ID 46 ................................................................. XXXVI
Table 99. Test case ID 47 ................................................................. XXXVI
Table 100. Test case ID 48 .............................................................. XXXVII
Table 101. Test case ID 49 .............................................................. XXXVII
Table 102. Test case ID 50 .............................................................. XXXVII
Table 103. Test case ID 51 .............................................................. XXXVII
Table 104. Test case ID 52 .............................................................. XXXVII
Table 105. Test case ID 53 .............................................................. XXXVIII
Table 106. Test case ID 54 .............................................................. XXXVIII
Table 107. Test case ID 55 .............................................................. XXXVIII
Table 108. Test case ID 56 .............................................................. XXXVIII
Table 109. Test case ID 57 .............................................................. XXXVIII
Table 110. Test case ID 58 ............................................................. XXXIX
Table 111. Test case ID 59 ............................................................. XXXIX
Table 112. Test case ID 60 ............................................................. XXXIX
Table 113. Test case ID 61 ............................................................. XXXIX
Table 114. Individual responses of test users ................................... XLV
CHAPTER 1. INTRODUCTION

1.1. Aims and Objectives

The primary objective of the project is to design a technological solution to suggest suitable recipes that can be utilised by people suffering from lifestyle diseases. In addition, anybody can use it to generate recipes. The solution will generate recipes based on the ingredients available with the person. It would encourage healthy eating patterns to help control the lifestyle diseases among the diagnosed users.

The project will be developed as a web application, thus it will be available for access from anywhere. The application will aid in eliminating the harmful ingredients or allow only the restricted amount of an ingredient for particular lifestyle diseases, thereby benefiting users of the application. Healthy eating does not necessarily imply to foods that are non palatable. The aim of the project is to change the mind frame of the people diagnosed with lifestyle diseases. It will recommend matching delicious recipes with the available ingredients and any additional ingredients if required as per the diet requirements of the disease.

1.2. Project Overview

The online grocery shopping provides bills as soft copy, whereas the physical grocery shopping provides bills as hard copy that has to be scanned and saved. This forms the input for the Intelligent Recipe Advisor.

The users will be of type, visitors and registered users. Visitors will be able to register. In addition they will be able to view “Sign In”, “About Us” and “How It Works?” pages. They can also contact the developer. Upon registration their account will be created. Registered users can sign in. Each registered user will have a shopping basket that stores the ingredients that are read from the uploaded bills. All registered users will have to complete mandatory fields on their profile. The recipes are generated based on the profile details of the individual. The user selects the desired recipe from the generated list and the recipe is displayed.
1.3. Outline

The outline of main chapters in this dissertation is as follows:

- **Introduction**: It presents the aims and objectives of this project also state the overview of the project.
- **Literature Review**: It presents background study of the core ideas and technologies used in the project. Familiarises and critically evaluates technologies that will be utilised.
- **Professional, Legal, Ethical and Social Issues**: It elaborates the professional, legal, ethical and social issues involved in the project also the plan of action for these issues
- **Requirement Analysis**: It lists the mandatory and optional requirements for the application. Demonstrates the motivation factors and evaluation criteria.
- **Project Plan**: It describes the project methodology and divides it into stages. Gantt chart is used to presents the project timeline. An assessment plan is created and likelihood of risks evaluated.
- **System Design**: It elaborates the architecture of the application, which includes Unified Modelling Language design, database design and user interface design.
- **System Implementation**: It illustrates the challenges and implementation of mandatory requirements during application development with code samples.
- **Testing**: It summarises results of test cases.
- **Evaluation**: It elaborates the types of evaluation, methods of analysis, plan of evaluation and implementation of questionnaire. Summarises and interprets results.
- **Conclusion**: It presents the summary of the report, limitations of the application and future developments.
CHAPTER 2. LITERATURE REVIEW

The field of cooking is a complex area where one has to remember even the minute details while preparing a meal for oneself, family, friends or others. Intelligent Recipe Advisor is a web application that uses matching patterns and artificial intelligence to aid oneself in the daily meal planning. The initial input is the bill that we receive after a grocery purchase. The bill should be in an image or PDF format. Hard copy bills should be converted to the corresponding format. The ingredients are retrieved from the bills and stored in the database. Majority would be familiar with the basics of cooking. People would be inclined to utilise recipes that match their health profiles using the ingredients purchased. In the present world the linking food items to health status becomes a necessity as the rate of lifestyle diseases are increasing. This web application would assist and simplify the process.

Intelligent Recipe Advisor will use the detailed analysis of the different but prevalent health conditions along with the prescribed or suitable diets and the user profile. This forms the criterion for the algorithms of the recipes application programming interface to match recipes with the desired user profile.

![Figure 1. Working of Intelligent Recipe Advisor](image-url)
The ideas and the technologies to be utilised in this project are outlined in the above figure. The objective of this review is to discuss these components, which also includes their assessment in terms of feasibility, merits and demerits alongside with the supporting literature. Firstly, the Optical Character Recognition section will discuss the GOCR, OCRopus and Tesseract. Secondly, the databases MySQL and MongoDB are examined. Thirdly, the recipes application programming interfaces Food2Fork and Yummly are considered. In addition, the health conditions section will primarily examine cardiovascular diseases, obesity and diabetes. Furthermore, a critical evaluation is done. Finally, evaluation and its concerns are discussed.

2.1. Optical Character Recognition

Optical Character Recognition is the rendering of written or typed manuscripts, written or scanned textual matter into device-alterable textual matter. Optical Character Recognition is a discipline in pattern identification, artificial intelligence and system perception. In developing applications, textual matter on paper or merchandise that are required to be scanned by devices to arrange the items in a particular manner, or proceed with some function based on the recognition of alphabets and numbers. The Optical Character Recognition element of a recognition or detection system is a predominant constituent in elevating the total quality (Menard 2008).

2.1.1. GOCR

According to Kulyukin and Vanka (2013), “GOCR is an open source OCR engine developed under the GNU public license. It preprocesses images via box-detection, zoning, and line detection. OCR is done on boxes, zones, and lines via pixel pattern analysis”. It is a not complex and responsive engine, which would not need any training information. Its identification method incurs two passes. The initial pass requests for the whole document. The next and final pass requests for the unknown characters. GOCR scans images in several formats such as pnm, pgm, pbm, and ppm then provides the output as textual file. At first, the image file usually in .jpg format, transforms into any of the accepted input formats mentioned-above before accepting the specified format file as input. Then the transformation of this file into .txt file occurs with the assistance of GOCR (Dhiman and Singh 2013).

2.1.2. OCRopus

OCRopus is a fairly novel modern open source Optical Character Recognition and document analysis framework. Google sponsors the development of this project
Intelligent Recipe Advisor

Rosemary Josekutty Thomas

H00172653

5

(Vincent 2007). OCRopus system stresses on standardisation and effortless extensibility, focused at both the investigation community and extensive business document conversions. It permits less complex assessment and recyclability of the Optical Character Recognition components by both research workers and organisations. Apache 2 is the open source license utilised by OCRopus that facilitates coaction between business and research workers, because endeavour can be applied commercially without many limitations. This system structures from scratch with multi-lingual and multi-script identification. For instance, OCRopus depends on the HTML and CSS standards for the illustration of typographic occurrences in several languages and scripts but utilises Unicode all over. The system reckons on only a limited amount of moderate deceptions and links, majority being picture based allowing it to effortlessly incorporate both present and new algorithms. “The system is extensible and programmable in a built-in scripting language” (Breuel 2008).

Srinivasan et al. (2010) describes the four main phases:

- The pre-processing phase comprises of binarization i.e. transforming any image into binary image, image smoothening and alignment correction. This phase has a first-hand effect on Optical Character Recognition.
- The layout analysis phase utilises the binarized image generated in the first stage and divides the image into non-textual sections, textual columns, textual blocks and textual-line sections. For each section, characteristics like vicinity or white space are separated and used to classify sections into textual and white space section. OCRopus utilises RAST-based layout analysis, which comprises of column searching, textual line searching and reading sequence selection.
- In the text line recognition phase, the textual lines from the preceding phase are utilised to identify the text in each line. OCRopus integrates Tesseract as the textual line recognition engine. This is explained in the succeeding section of Tesseract Optical Character Recognition engine.
- The statistical language-modelling phase works out the equivocal characters derived in the preceding phase. The instances of statistical language models are dictionaries and stochastic grammar. OCRopus utilises open source OpenFST library as its language-modelling tool. This phase enriches the precision of text identification.
2.1.3. Tesseract

Tesseract is an open-source Optical Character Recognition system, which was created at HP during the period 1984 to 1994. In Bristol, Tesseract started as a Ph.D. research work in HP Labs. It earned strength as a potential software and hardware accessory for HP's product-line of flatbed scanners. HP launched Tesseract for open-source towards the end of year 2005. It can be downloaded from http://code.google.com/p/tesseract-ocr. Tesseract was presumably the first Optical Character Recognition engine capable of managing white on black textual matter very effortlessly. Tesseract did not require its own page layout analysis as HP had already developed it for its scanners (Smith 2007).

The initial step is to furnish input to Tesseract engine, which could be any picture. This picture is then transformed into a 'binary image' with the aid of 'adaptive thresholding'. The Optical Character Recognition consists of sectionalisation and identification. During sectionalisation, a connected component analysis is utilised to preserve the boundary of an element. These boundaries are collected jointly into blobs with the assistance of nesting. Then these recognised blobs are structured into text lines that are further separated into independent words in relation to the space between distinctive characters. This method is progressed with the aid of fuzzy spaces. The identification process occurs in a two-pass procedure. In the initial pass, after identifying each word the acceptable word is transferred to an adaptive classifier as training information. While in the next and final pass, the words not identified in the first pass are identified. The output is the derived text from the image after eliminating the fuzzy spaces in the final phase (Kulyukin and Vanka 2013).
2.1.4. Summary

GOCR is an open source Optical Character Recognition engine that is developed under the GNU’s Not Unix public license. It transforms scanned images of textual form to textual files using two passes identification method. It is an unsophisticated and receptive engine that requires no training input (Dhiman and Singh 2013, Kulyukin and Vanka 2013). While OCRopus is reasonably an updated open source Optical Character Recognition and document analysis framework. It processes scanned images to textual files in four main stages. OCRopus engine stresses on both the investigation community and extensive business document conversions by emphasising on standardisation and effortless extensibility (Vincent 2007, Breuel 2008, Srinivasan et al. 2010). On the other hand, Tesseract is an open-source Optical Character Recognition engine. It was developed at HP. It processes scanned images to textual files in two main stages. It is capable of managing white on black textual matter very facilely. Tesseract was apparently the first Optical Character Recognition engine with this accomplishment. In addition, it can transform colour images (Smith 2007, Dhiman and Singh 2013, Kulyukin and Vanka 2013).
2.2. Databases

2.2.1. MySQL and MongoDB

Structured Query Language or SQL is a benchmark language for managing databases to a great extent. MySQL is a renowned open source “Relational Database Management System” that utilises SQL (Blansit 2006). Over the past years SQL databases have been established and found dependable for various applications ranging from web to desktop. Recently NoSQL databases have achieved enormous growth, which is due to the demand of expanding scalability and accessibility requisites. Enormous quantity of data can be stored in SQL databases. It is capable of querying the same quantity of data stored. The two reasons are that programmers are accustomed to it and the core structures of these databases are stable. It becomes a tedious task to disseminate SQL databases to an enormous degree. In other words, it is strenuous to scale SQL databases horizontally. The dominant concepts of these databases are coherency and accessibility, so these have low resistance toward network partitions. The hardware requirements of the machine have to expand as and when the volume of the database is amplified. (van der Veen et al. 2012).

NoSQL implies not a “Relational Database Management System”. The key feature of a non-relational database is being dynamic. There are document databases, graph stores, Key-value stores and wide-column stores. MongoDB is a document database. It is not limited to follow a single schema but supports dynamic schemas. MongoDB consists of databases, collections, and documents. Each database can have numerous collections and each collection is a manifold of documents. Elements are not defined formerly, so generating collections as per requirement is not a violation. A collection may consist of records with various schema documents. The number of attributes can vary from one record of a document to the next record. “The type of the property can be any basic data types, such as numbers, strings, dates, etc., or an array or hash, and even a sub-document.” The querying process is quickened, as the concept of normalisation is not followed. MongoDB also has the capability of auto-sharding and integrated caching (Zhu et al. 2011, MongoDB 2014).

2.2.2. Summary

The standard language for manipulating databases to any degree is termed as Structured Query Language or SQL. The expanding requirements of the industry towards scalability have lead to the extensive growth of NoSQL databases. The
predominant attributes of SQL databases are coherency and accessibility so it is effortful to disseminate SQL databases horizontally (van der Veen et al. 2012). NoSQL is distinguished being a non-static and non-relational database. MongoDB is an instance of document database, which supports dynamic schemas. The principal attribute of SQL that is normalisation is not adopted by NoSQL databases (Zhu et al. 2011, MongoDB 2014).

### 2.3. Recipes Application Programming Interfaces

Intelligent Recipe Advisor requires a comprehensive database of recipes. There are millions of recipes available on the Internet, which comes from dedicated recipes websites, recipe blogs, e-articles and e-newspapers. It is impossible to manually enter all the possible recipes in the database. There is a wide range of criteria to be analysed into like nutritional information, health conditions and allergies that may not be apparent but represent an important role in meal preparation. Additionally, it would be time consuming to index all the recipes. Therefore recipe application programming interfaces can be utilised for this purpose. Food2Fork application programming interface and Yummly application programming interface are assessed to determine the most feasible and compatible application programming interface for this project.

#### 2.3.1. Food2Fork

Food2Fork provides an application programming interface that uncovers its ingredient search feature throughout its database of publishers (Food2Fork 2014b). Food2Fork.com has a database of 44 publishers which includes 101 Cookbooks, All Recipes, Simply Recipes and Pillsbury Baking (Food2Fork 2014a). Food2Fork application programming interface provides entry to their always growing socially graded recipe index and up-to-the-minute ingredient search activity (Food2Fork 2014b).

This application programming interface provides two types of ordering for queries – “rating and trendingness”. The social media points are used to evaluate the excellent recipes. This is the order by rating. There is a system of tend points for the most fresh recipes from the publishers of Food2Fork, which standardises on the pace at which these recipes earn reputation. Food2Fork’s purchased application programming interface plans are permitted for business use. Also, their free application programming interface plan can be utilised in business services or products stating attribution. Furthermore it requires a backlink to Food2Fork.com. The free application
programming interface plan can make 500 calls per day, while the paid application programming interface plans calls per day range from 2500 to 150,000 costing from $50 to $750 per month (Food2Fork 2014b).

2.3.2. Yummly

The Yummly Recipe Application programming interface allows incorporation of recipes and featured recipe search into web applications. Users can be linked with the excellent recipes accessible according to their special food consideration because Yummly recognises “the recipe – ingredients, diets, allergies, nutrition, taste, techniques and more”. It includes over one million recipes collected from several references like the renowned food blogs and delicious recipe websites around the world. The application programming interface has a rapidly increasing recipe database. Yummly has a database of 59 publishers that includes 101 Cookbooks, All Recipes, Simply Recipes, Gluten Free Goddess, Steamy Kitchen and many more (Yummly 2014a). It has over 100,000 categorisations supported on food genetics and Yummly algorithms. It has over 10,000,000,000 data locations to retrieve appropriate recipes to queries (Yummly 2014b).

The main features of Yummly are listed below (Yummly 2014c):  
- Showcases photos or pictures filled recipes with nutritional information
- Filters the recipes according to health condition or allergy.
- Refines the recipe by the culinary art, flavours, methods, courses, duration or preparation stages.
- The free application programming interface plan can make 500 calls per day, while the paid application programming interface plans calls per day range from 1000 to 50,000 costing from $100 to $2000 per month. All the paid plans include a 14-day free trial. They also provide options for customisation of plans.
- The free plan cannot be used for business purpose and must state attribution. In addition, there should be a backlink to Yummly or the other website mentioned by Yummly and must include the Yummly logo.

2.3.3. Summary

Food2Fork and Yummly provide application programming interfaces that allow integration of recipe search into a web application. Food2Fork.com has a database of 44 publishers; alternatively Yummly has a database of 59 publishers. Food2Fork allows “rating and trendingness” for ordering queries; in contrast Yummly identifies “the recipe
- ingredients, diets, allergies, nutrition, taste, techniques and more”. Food2Fork and Yummly offer plans for personal and business purposes (Food2Fork 2014a, Food2Fork 2014b, Yummly 2014a, Yummly 2014b, Yummly 2014c).

2.4. Health Conditions and Related Foods

2.4.1. Body Mass Index

Body Mass Index is referred to as BMI. It is the presently established standard for categorising risks associated with body mass and is measured using the formula:

\[
\text{Body Mass Index} = \frac{\text{Weight/body mass in kilogram}}{(\text{Height in metre})^2}
\]

Body Mass Index values of less than 25 kg/m² are categorised as normal and Body Mass Index values of greater than or equal to 25 kg/m² are categorised as overweight (Goacher et al. 2012, Janssen et al. 2005).

2.4.2. Cardiovascular Diseases

Incidents of heart diseases and related death are found to be lower among the inhabitants of Mediterranean zones and certain Asian territories. They have conservative diets that mainly consist of plant sources like fruit and vegetables, nuts, organic vegetable oils and whole-grains. The minimised risk for cardiovascular diseases have been linked with nourishing diets comprising of roughage, folic acid, potassium, bioflavonoids and antioxidants present in fruits and vegetables. The higher intake of fruit and vegetables is defensive against cardiovascular diseases. Diets abundant in peanuts, walnuts, or almonds diminishes low-density lipoprotein cholesterol and lowers the ratio of sum to high-density lipoprotein cholesterol as established by several metabolic studies. Arginine is abundant in almost all nuts that are capable of relaxing blood vessels thereby, inhibiting platelet adhesion and aggregation. Also, walnuts are abundant in alpha-linolenic acid. Various researches have linked increased consumption of alpha-linolenic acid to diminish the risk of coronary artery disease presumably due to its antithrombotic and antiarrhythmic properties. Whole-grains food products are abundant in roughage, antioxidant vitamins, magnesium, flavonoids and carotenoids. In addition, it has diminished glycaemic index. High level of whole-grain in diets aid reducing the risk of ischemic heart disease that is separated from the common cardiovascular disease risk factors (Hu 2003).
Nuts have reduced saturated fatty acids and abundant in monounsaturated and polyunsaturated fatty acids. Emerging studies establishes the presence of other biologically active particles in nuts that exhibit cardio protective properties. Nuts constitute of omega-3 fatty acids that aid in protecting against heart diseases. Fundamentally the factors accountable are highly unsaturated omega-3 fatty acids eicosapentaenoic acid and docosahexaenoic acid. Nuts are also abundant in manganese, magnesium, copper, potassium, phosphorus, thiamine, niacin, riboflavin, selenium and zinc. In general, nuts are rich in vitamin E specifically almonds. Peanuts are a substantial source of folic acid. Nuts are a source of roughage including soluble roughage. Soluble roughage has been evident to reduce low-density lipoprotein cholesterol levels and enhance glycaemic index control (Kris-Etherton et al. 1999).

Distributing the roughage intake by incorporating thick roughage into various carbohydrate foods aids in diminishing the glycaemic index of the carbohydrate constituent of the diet. This may be accountable for diminishing the ratio of total to high-density lipoprotein cholesterol and of low-density lipoprotein to high-density lipoprotein cholesterol thereby lowering serum lipids. In cohort analyses, low-glycaemic-index diets were linked with increased high-density lipoprotein cholesterol level and diminished risk of heart diseases. It is required to develop delectable diets by incorporating lower quantities of roughage into daily foods while retaining nutritive value. Foodstuff consisting of adequate psyllium and oat beta-glucan per serving can establish a health claim for heart diseases risk reduction and are as delectable as their low-roughage equivalents. While delectable, the absence of health hazards from these
Intelligent Recipe Advisor

Rosemary Josekutty Thomas

foods demonstrate that intake of adequate servings of roughage-supplemented diets will aid conditions requiring larger diminution in lipid serum for heart diseases (Jenkins et al. 2002).

It is evident that meals having fat concentration increase body weight and insulin resilience. The constant intake of such meals may have a negative impact on the cardiovascular diseases; as epidemiologic, clinical, and in vitro studies demonstrate it. Furthermore, various studies indicate that these sustaining negative impact are previously produced by the chronic disorder of endothelial function and the concentration in oxidative stress determined after the ingestion of fast food meals with minimal content of vitamins. The critical risk factors linked with development and maturation of atherosclerosis are endothelial disorder and oxidative stress. The occurrence of endothelial disorder and oxidative stress has constantly been linked with a higher risk of heart related diseases and deaths in the future (Rudolph et al. 2007).

2.4.3. Obesity

The intake of fried food may accelerate obesity by increased fat ingestion and energy density. Frying alters both the foodstuff and the oil. However olive oil is advised for frying since it is more repellent to oxidation in comparison with other oils. The quantity of oil retained by frying foodstuff varies from the kind of food and the frying states. Moreover, frying causes the pyrolysis of fried foodstuff in outer layers that constitute of heterocyclic amines. Lastly, by frying foodstuff is contaminated by the deteriorated by-products of the frying oil. These products have been linked with several classes of cancer, endothelial disorder and hypertension. The intake of fried food at discretion may lead to accelerate the consumption of foodstuff with high energy density and low satiety index. The correspondingly low satiety index of fats would be linked to their diminished capability to induce insulin and leptin secretion. Furthermore, the consumption of fried foods can be linked with obesity since fats are incorporated and metabolised more effectively unlike other macronutrients (Guallar-Castillón et al. 2007).

2.4.4. Diabetes

Patients with insulin-dependent diabetes mellitus and non-insulin-dependent diabetes mellitus having meal plans consisting extensively of starchy foodstuffs low in glycaemic index have exhibited to be useful for managing blood glucose level i.e. by reducing serum cholesterol and triglycerides. This is also evident when low glycaemic index and prescribed diets contribute nearly uniform quantities of soluble and insoluble roughage
in addition to all the macronutrients. When comparing foodstuffs with low glycaemic index to other common foods in inducing satiety, low glycaemic index foods are found to be more useful. Additionally, foodstuffs with this attribute contribute to prevent and forms a remedy to cure overweight and moderate obesity, thus becomes important in developing their dietary strategies. In vitro studies for starchy foodstuffs and starch preparations indicate vulnerability to alpha amylase action, i.e., speed of starch hydrolysis. Also the glycaemic index and glucose response are comparatively higher than that of roughage. It is evident that the glycaemic index values demonstrate primarily the rate at which glucose enters into the bloodstream after consumption of food (Trout et al. 1993).

Table 2. Glycaemic index and composition of various foods

(Trout et al. 1993)

<table>
<thead>
<tr>
<th>Glycemic index (GI) and composition of various foods*</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Food</strong></td>
<td><strong>GI</strong></td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>Millet</td>
<td>103 (2)</td>
</tr>
<tr>
<td>Oatmeal porridge</td>
<td>89 (2)</td>
</tr>
<tr>
<td>White rice</td>
<td>81 (2)</td>
</tr>
<tr>
<td>Brown rice</td>
<td>81 (2)</td>
</tr>
<tr>
<td>Potatoes (raw)</td>
<td>80 (2)</td>
</tr>
<tr>
<td>Sweet corn</td>
<td>80 (2)</td>
</tr>
<tr>
<td>Yams</td>
<td>74 (2)</td>
</tr>
<tr>
<td>Wheat kernels</td>
<td>63 (2)</td>
</tr>
<tr>
<td>Pinto beans</td>
<td>60 (2)</td>
</tr>
<tr>
<td>White (haricot) beans</td>
<td>54 (2)</td>
</tr>
<tr>
<td>Blackeye (cow) peas</td>
<td>53 (10, 14)</td>
</tr>
<tr>
<td>Green peas (dried)</td>
<td>50 (2)</td>
</tr>
<tr>
<td>Rye kernels</td>
<td>47 (2)</td>
</tr>
<tr>
<td>Chick peas (dried)</td>
<td>47 (2)</td>
</tr>
<tr>
<td>Lima (butter) beans (dried)</td>
<td>46 (2)</td>
</tr>
<tr>
<td>Kidney beans (dried)</td>
<td>43 (2)</td>
</tr>
<tr>
<td>Lentil (dried)</td>
<td>38 (2)</td>
</tr>
<tr>
<td>Pearl barley</td>
<td>36 (2)</td>
</tr>
</tbody>
</table>

* CHO, available carbohydrate. Numbers in parentheses indicate references for data.
† Amount of component per 50-g CHO portion was calculated for raw food.
‡ Uncooked bulgur.

Table 3. Effect of processing on the glycaemic index and total dietary fibre of foods

(Trout et al. 1993)

<table>
<thead>
<tr>
<th>Effect of processing on the glycaemic index (GI) and total dietary fiber (TDF) of foods*</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Food</strong></td>
<td><strong>GI†</strong></td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td>Corn, boiled</td>
<td>80</td>
</tr>
<tr>
<td>Cornflakes</td>
<td>121</td>
</tr>
<tr>
<td>Rice (brown, boiled)</td>
<td>81</td>
</tr>
<tr>
<td>Rice (white, boiled)</td>
<td>81</td>
</tr>
<tr>
<td>Puffed rice</td>
<td>332</td>
</tr>
<tr>
<td>Wheat kernels (pressure cooked)</td>
<td>63</td>
</tr>
<tr>
<td>Bread (whole wheat)</td>
<td>100</td>
</tr>
<tr>
<td>Bread (white)</td>
<td>100</td>
</tr>
<tr>
<td>Puffed wheat</td>
<td>510</td>
</tr>
</tbody>
</table>

* CHO, available carbohydrate. Numbers in parentheses indicate references for data on the CHO and TDF content of the indicated foods, respectively.
† GI values are from Wolever et al (2).
‡ Uncooked bulgur.
Legumes have a low glycaemic index and good sources of roughage. The diets rich in legumes are presumed useful in avoiding type 2 diabetes mellitus. The good examples of legumes are beans, lentils, peanuts, peas, and soybeans. The reduced risk of type 2 diabetes mellitus may be wholly linked to soybeans. Soy protein indicates to control hypoglycaemia, lower body weight, and hyperinsulinemia in diet experiments of patients with type 2 diabetes mellitus and without diabetes. The defensive effect of legumes on type 2 diabetes mellitus may contain numerous biological consequences, such as higher roughage content; lower glycaemic index of combination meals, or both; and the antioxidant effect of polyphenols specifically isoflavones and lignans. Soy protein may reduce adiposity by inhibiting insulin secretion from pancreatic beta cells or by inhibiting lipogenesis and enhancing lipolysis in the liver and adipocytes. Peanuts comprises of roughage, magnesium, antioxidant vitamins, and polyunsaturated and monounsaturated fatty acids. The defensive effect of peanuts on type 2 diabetes mellitus is due to their excessive proportion of polyunsaturated fatty acids, which indicates links with insulin sensitivity (Villegas et al. 2008).

Whole-grains bread, certain ready-to-eat breakfast cereals, pasta, oatmeal, brown rice, lightly pearled barley, and bulgur wheat are all examples of whole-grain foodstuffs. The flour is composed of large mass of the grains when compared to the whole-grains (Jacobs Jr and Steffen 2003). Whole-grains foodstuffs are rich in roughage so would exhibit many coinciding physiologic effects, comprising of useful gastric intestinal function, control of glycaemic index, desirable lipid profile and reduction in oxidative stress. The physical structure; high roughage proportion; and the content of organic acids and enzyme inhibitors in whole-grain foods indicate to act reciprocally to impact digestion and absorption of glycaemic carbohydrates. Lately, in vivo experiments have exhibited that the level of gelatinisation initiated in food processing is likewise highly critical in modifying the eventual glucose and insulin responses (Liu 2003).

The elevated serum concentrations of alpha-tocopherol, Vitamin B6 and folic acid were linked to lower risk of type 2 diabetes. Grain germ has excessive Magnesium content. Magnesium is inversely linked with high insulin levels and risk of type 2 diabetes. Diets rich in saturated fat also contribute to the high morbidity of diabetes. There is a close association between the ingestion of total roughage, particularly cereal roughage, and low morbidity of type 2 diabetes. However roughage in the form of fruit or vegetables did not alter the risk of diabetes. The alteration for cereal roughage noticeably diminished the link between whole-grain ingestion and incidence of type 2 diabetes that
indicates the association of whole-grains may be a result of cereal roughage or factors associated to cereal roughage ingestion. The late absorption and digestion of carbohydrates leads to a decreased requirement for insulin is related to the favourable effect of soluble roughage in meals. Correspondingly there is a distinct link between the ingestion of insoluble roughage, preferably than of soluble fibre, and the incidence of type 2 diabetes. Insoluble roughage diminishes the intestinal cycle that subsequently allocates insufficient time for carbohydrates to be absorbed thereby lowering insulin demand (Montonen et al. 2003, McKeown et al. 2002).

Table 4. Food group composition of the Therapeutic Lifestyle Change diet

(Escott-Stump and Mahan 2004)

<table>
<thead>
<tr>
<th>FOOD GROUP</th>
<th>STEP 1 DIET</th>
<th>STEP II DIET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1200 kcal</td>
<td>1600 kcal</td>
</tr>
<tr>
<td>Fats and oils</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Fish, poultry and meat</td>
<td>6 oz</td>
<td>6 oz</td>
</tr>
<tr>
<td>Egg yolks</td>
<td>3/week</td>
<td>1/week</td>
</tr>
<tr>
<td>Dairy foods</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Bread, beans, grains and starches</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Fruit</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Vegetables</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Sugars, sweets and alcohol</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 5. Low-calorie diet

(Escott-Stump and Mahan 2004)

<table>
<thead>
<tr>
<th>NUTRIENT</th>
<th>RECOMMENDED INTAKE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carbohydrates</td>
<td>50% to 55% of total kilocalories</td>
</tr>
<tr>
<td>Proteins</td>
<td>15% to 25% of total kilocalories</td>
</tr>
<tr>
<td>Fats</td>
<td>Not to exceed 30%</td>
</tr>
<tr>
<td>Fibre</td>
<td>Extra</td>
</tr>
<tr>
<td>Sugar</td>
<td>Limited</td>
</tr>
</tbody>
</table>

Table 6. Nutrient composition of the Therapeutic Lifestyle Change diet

(Escott-Stump and Mahan 2004, Haupt 2013)

<table>
<thead>
<tr>
<th>NUTRIENT</th>
<th>RECOMMENDED INTAKE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saturated fat</td>
<td>Less than 7% of total calories</td>
</tr>
<tr>
<td>Polyunsaturated fat</td>
<td>Up to 10% of total calories</td>
</tr>
<tr>
<td>Monounsaturated fat</td>
<td>Up to 20% of total calories</td>
</tr>
<tr>
<td>Total fat</td>
<td>25% to 30% of total calories</td>
</tr>
<tr>
<td>Carbohydrates</td>
<td>50% to 60% of total calories</td>
</tr>
<tr>
<td>Fibre</td>
<td>22-34g/day</td>
</tr>
<tr>
<td>Protein</td>
<td>Approximately 15% of total calories</td>
</tr>
<tr>
<td>Cholesterol</td>
<td>Less than 200mg/day</td>
</tr>
<tr>
<td>Nutrition</td>
<td>Requirement</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Sodium</td>
<td>Not to exceed 2300mg/day</td>
</tr>
<tr>
<td></td>
<td>Not to exceed 1500mg/day for diabetes</td>
</tr>
<tr>
<td>Potassium</td>
<td>4700mg/day</td>
</tr>
<tr>
<td>Calcium</td>
<td>100mg to 1300mg/day</td>
</tr>
<tr>
<td>Total calories (energy)</td>
<td>Balance energy intake and expenditure to maintain desirable body weight/prevent weight gain.</td>
</tr>
</tbody>
</table>

### 2.4.5. Summary

Body Mass Index is measured using the formula, weight in kilogram/(height in metre)^2.

Wholesome diets containing roughage, folic acid, potassium, bioflavonoids and antioxidants associates to diminished risk for cardiovascular diseases. Low-density lipoprotein cholesterol and the ratio of sum to high-density lipoprotein cholesterol minimises when intake of peanuts, walnuts, or almonds are ample in the diet. Furthermore nuts demonstrate cardio protective attributes. Roughage, antioxidant vitamins, magnesium, flavonoids and carotenoids are ample in whole-grain food products. Administering the consumption of roughage by altering the carbohydrate rich meals to include thick roughage helps in minimising the glycaemic index of the carbohydrate ingredient of the diet. Damaging effects on cardiovascular diseases are apparent when diets are having fat concentration, which results in the increase of body weight and insulin resilience (Goacher et al. 2012, Janssen et al. 2005, Hu 2003, Kris-Etherton et al. 1999, Jenkins et al. 2002, Rudolph et al. 2007).

The intake of fried food at discretion may lead to accelerate the consumption of foodstuff with high energy density and low satiety index. Moreover, metabolism and assimilation of fats are more effective in contrast to other macronutrients. Therefore, increased intakes of fried foods are coupled with obesity (Guallar-Castillón et al. 2007). Controlling the blood glucose level can be accomplished by incorporating a larger proportion of starchy foodstuffs low in glycaemic index to the meal plan (Trout et al. 1993). Legumes and peanuts have a shielding effect on type 2 diabetes mellitus. Soy protein may diminish adiposity (Villegas et al. 2008). Roughage is abundant in whole-grains foodstuffs so would indicate several synchronic physiologic effects, consisting of beneficial gastric intestinal function, control of glycaemic index, desirable lipid profile and reduction in oxidative stress (Liu 2003). The reduced risk of type 2 diabetes was coupled with elevated serum accumulation of alpha-tocopherol, Vitamin B6 and folic acid. Elevated amounts of insoluble roughage reduce the intestinal cycle thus diminishing insulin demand (Montonen et al. 2003, McKeown et al. 2002).
2.5. Web Application

The Internet is a worldwide computer network comprising of collections of coordinated networks of computers utilising established communication protocols. It contributes to the range of information available and also to various communication services (Oxford University Press 2014). There is no entity like the Internet that has transformed the domain of computer and communications. The Internet has a global broadcasting facility; a technique for information transmission; and a system for association and contact between people and their computers placed at different global locations. Additionally, the Internet indicates instances of the most productive utility of sustained investment and responsibility towards the research and development of information system. Furthermore, the Internet contains information of the Internet, including its history, technological application and utilisation (Leiner et al. 2009).

The web applications are application developed utilising established web technologies. Web applications operate in any of the latest web browsers. There are some features that differentiate web applications from websites. The web applications have to be setup by the users, they are usually complete and do not constantly require the support of a web browser. It can be developed to operate without Internet. A website can be transformed into a web application by inserting features to a website thereby modifying it to a robust computing environment. Web browsers and portable devices can operate these web applications. It is less complex to use and has enhanced user experience when compared to websites. Web applications can be extended as it has access to developing functionalities. The only requisite to follow to develop a web application from a website is to insert an application manifest. This is a JSON document that represents the application settings, comprising of the name of the application, the icons of the application, and a plain text description of the application (Mozilla Developer Network 2014).

Bootstrap is also known as ‘Twitter Bootstrap’. It was released in 2011 as open source and maintained on GitHub. It is the most widely accepted design framework for developing active web applications. The Bootstrap framework provides a rapid and less complex scenario for designing the web applications. It provides assistance for the two well-known CSS pre-processors, Less and Sass. Bootstrap has the built-in feature for hybrid web application development from a single code base. In addition, Bootstrap has a comprehensive documentation (Bootstrap 2014, Lerner 2012).
2.5.1. Summary

The Internet has a planetary mass medium facility; an approach for information dispersion; and a mode for linkage between people and their computers placed at international locations (Leiner et al. 2009). The web applications are generally comprehensive and do not persistently require the support of a web browser. An application manifest is a JSON document that portrays the application settings (Mozilla Developer Network 2014). The Bootstrap framework specifies a prompt and less sophisticated outline for designing the web applications. Bootstrap has been integrated with functionalities for composite web application development (Bootstrap 2014, Lerner 2012).

2.6. Critical Analysis

2.6.1. Comparative Analysis

In the above literature review we have analysed and examined the main ideas and the technologies. Now we shall critically evaluate the technologies that pertain to domains of Optical Character Recognition, databases, search engines, application programming interfaces and the front-end framework. This will aid to determine the feasible solutions for developing the web application.

This project will be utilising Tesseract Optical Character Recognition engine to read the bill containing textual matter of ingredients for purpose of adding to the database. Tesseract 3.0 has the paramount precession level of any similar Optical Character Recognition engine when handling officially typewritten and formatted documents (Milligan 2013). GOCR presume to separate less information than Tesseract displaying accelerated processing times. However Tesseract has superior identification rates even though its processing times has delays (Kulyukin and Vanka 2013). Tesseract is compatible with aligned baselines which is a merit when managing with geometric deformation caused by non fronto-parallel examination and removes the requirement to correct image sections (Posner et al. 2010). It can be derived from the above literature that OCRopus utilises the Tesseract Optical Character Recognition engine. This project will therefore use the native Tesseract engine rather than the OCRopus engine, as it would be stable, efficient and is up-to-date.

Intelligent Recipe Advisor requires a very flexible database structure, as the schema cannot be defined earlier. And also it would depend on the profile of each person.
Therefore MongoDB is the appropriate database for this purpose. There is a necessity for fast retrieval of data in response to the queries passed. From the above comparison it can be inferred that MongoDB has accelerated performance than MySQL. In addition, MongoDB supports horizontal scaling so it would not be an issue to add volume without redefining the current hardware structure when the number of users increases in the future. It can be determined that usually programmers restrict to consent database modifications due to the inherent complexity of the database. MongoDB is a dynamic database so alternations can be easily incorporated for further versions.

Moreover, it is evident that Yummly application programming interface has an extensive recipe database than the Food2Fork application programming interface. Food2Fork only supports two types of sorting whereas Yummly can work with a wide range of parameters. The application programming interface plans of Yummly are expensive than Food2Fork but it is apparent that cost is based on the features provided by it. Yummly application programming interface follows an intelligent algorithm from the examination in the literature review. When taking into consideration the future improvements in this application, it is prudent to choose Yummly application programming interface over Food2Fork application programming interface.

The Internet has already transformed to an important medium of information transmission. We will be utilising Bootstrap for front-end development of the web application due to its less complexity and added functionalities. Additionally, it eliminates the requisite to create a mobile application at this stage since it has the capability to scale as per the device. Furthermore, development of technologies utilising the Internet is increasing in the present scenario. Therefore we should utilise this technology in developing web applications to benefit society in addition achieving the objectives of this project i.e. developing Intelligent Recipe Advisor.

2.6.2. Analysis of Issues

We have determined the technologies to be employed. The web application will be built utilising the following components or technologies:

- Tesseract Optical Character Recognition engine
- MongoDB
- Yummly Application Programming Interface
- Information of the nutrients related to the diseases
- Bootstrap front-end framework
Tesseract Optical Character Recognition engine will be utilised to read the grocery bills. The critical issues that could arise are the text is not clear and cases of handwritten bills. This can be avoided to an extent by displaying a message to only use clear scanned bills, which are not handwritten. NoSQL database is utilised to store the information of each user. It can be easily scaled without altering the web application depending on the number of users. As NoSQL is an emerging database locating web hosts that support it would be little difficult. However, discarding NoSQL considering a single negative is not recommended as it has enormous advantages. Yummly has the databases of all the recipes. The queries passed to Yummly to retrieve the suggestions of the recipes for the appropriate health status and the user profile. These queries will be manually written by utilising the information from the literature review of the health conditions and the related foods. This is crucial as it forms the major functionality of the web application and must be completed with utmost focus otherwise would effect negatively to the performance of the application. Furthermore, Bootstrap will enable the less complex but effective design solution with the limited available timeframe. Bootstrap has restrictions in designing however for developers with less experience in user interface designing it is beneficial.

2.7. Evaluation

Evaluation is an important factor for every application. Even though testing would be carried out during and after the development of the application, end-user evaluation is critical in determining if the application is a feasible solution. In this section we will discuss the evaluation types, analysis methods and implementation.

2.7.1. Types of Evaluation

The evaluation in the development of an application is planned according to the duration of the project. There are two types of evaluation:

- **Formative evaluation**: It includes assessing the value of a software application while it undergoes development. Formative evaluation aims on transitional or initial results occurring throughout the developmental phase of the application (Power 2003). This contributes to constant enhancement of the application pertaining to its goals and objectives.

- **Summative evaluation**: It includes assessing the value of a software application on completion of the development and succeeding its implementation. Summative evaluation aims on determining existing and long-term results.
It is utilised to judge the overall performance of the application for the proposed end users.

The evaluation of Intelligent Recipe Advisor utilised formative evaluation since it determined the level at which the application satisfies the goals and objectives in addition to the enhancement that is proposed under future development. We presume that this evaluation is not the deciding factor even though it will be performed on a completely working application. We are not implementing a user-centred design due to the limitation of time. The summative evaluation will be performed preceding the official launch of the application, as the end users will aid in enhancing the application. The aim of the present evaluation will be usability, accuracy of results, user acceptance and to some extent the graphical user interface.

2.7.2. Analysis Method

The data will be collected in different forms so the analysis can be either qualitative or quantitative. The quantitative data analysis includes working with digits while qualitative data analysis includes working with non-numerical written matter (Garbarino and Holland 2009). We will utilise qualitative data analysis as data will be in the form words and requires analysing the positive or negative criticisms and suggestions.

Content analysis is almost comprehensively utilised method in qualitative evaluation to analyse data (Dawson 2007). The process of coding involves reviewing the whole set of data by allocating codes to the various sections of the user responses. The selected codes can be words or numbers. Content analysis adopts three techniques namely conventional, directed, or summative, to clarify meaning i.e. coding from the data collected. The coding categories or themes are formulated immediately from data for conventional content analysis while directed content analysis commences with coding categories or themes that are decided before the analysis starts. However summative content analysis includes adding up and contrasting commonly the important words or phrases within the data proceeded by the clarification of the fundamental background (Hsieh and Shannon 2005).

We will utilise the conventional and directed techniques of content analysis. Our questionnaire will be a semi-structured one, so the user responses will match into some clear categories that indicate the sections on which our questions are aimed. The
responses collected from these sections draws focus as it aims at evaluating the ease of use, functionalities and the graphical interface. The initial categories will be formulated from theory on usability and functionality sections to focus on the application. In addition, conventional content analysis technique will be utilised to formulate new categories from the data collected if it would not match the pre-formulated categories.

### 2.7.3. Motivation Factors

The following indicates a few applications and tools that recommend recipes:

- **RecipeMatcher**: It is a free web application that advices recipe suggestions based on the available ingredients (RecipeMatcher 2012).
- **Recipes by Ingredients**: The free application recommends recipe results only based on the ingredients. However the results are not accurate with all the options. On the contrary, while on subscription it gives the most appropriate results. It offers free membership for 7 days and the paid membership costs about £9 per month. It is supported only on Android, Windows 8 phone, iOS and Google Chrome and Blackberry 10 (AB Mobile Apps 2013).
- **Recipe Puppy**: It is a free web search engine that shows ingredient-based recipe suggestions (Recipe Puppy 2014).
- **Su Chef**: The basic recipe based search is free. The full add on pack costs about £3, which includes major dietary considerations. The cost of individual packs for various health conditions ranges from £1 to £2. It is supported only on iOS (Su Chef 2014).

<table>
<thead>
<tr>
<th>SL. NO</th>
<th>FEATURES</th>
<th>RecipeMatcher</th>
<th>Recipes by Ingredients</th>
<th>Recipe Puppy</th>
<th>Su Chef</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Read ingredients from the grocery bill</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>2.</td>
<td>Search recipe based on the common lifestyle diseases</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>3.</td>
<td>Search recipes based on ingredients</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>4.</td>
<td>Search based on allergens</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>5.</td>
<td>Whole application/tool is</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>
From the above, it can be inferred that these applications and tools do not satisfy all the listed features. The lacking features are added to the needs of the application that is listed below:

- A web solution to recommend suitable recipes for the people diagnosed with common lifestyle diseases.
- Read ingredients from the grocery bills as typing these would be a time consuming process.
- Stimulate wholesome consumption patterns by regulating the amount of ingredients in the recipes to help reduce the intensity of these diseases.
- Alter the mind set of the people with these health conditions by advising appetising recipes with the selected ingredients and if essential add more ingredients, as per the nutritional requirements.
- Suggest recipes for people with normal health conditions by not limiting the users.

### 2.7.4. Evaluation Criteria

The criteria mentioned below will be employed in the evaluation to determine if the developed application satisfies the basic requirements:

- Is the application able to recognise the ingredients from the images of grocery bills?
- Can people with common lifestyle diseases use the application?
- Is the application able to suggest recipes according to the health profile of the person?
- Is the application able to suggest nutritious and delicious recipes?
- Is the application easy to use?
CHAPTER 3. PROFESSIONAL, LEGAL, ETHICAL AND SOCIAL ISSUES

Every application developed requires professional estimation and implementation. Therefore the web application to be developed in the later stage during the dissertation has the same requisite regardless of the size of the application. This will enhance the quality of the final application and thereby have direct impact on the professional experience of the technologist.

The following becomes critical while developing any professional application. The legal, ethical and social issues are also discussed. Furthermore, this is applicable for the application to be developed.

- Quality time is assigned to the development of Unified Modelling Language diagrams and the required documentation to aid a better structure of the application.
- Security of the application is crucial so unauthenticated users will not be able to access or modify user information stored in the application.
- The Intellectual Property Law; the Data Protection Act; the Computer Misuse Act and every cyber related laws and acts are acknowledged. The documentations and programs to be utilised in the dissertation phase are cited to correct authors. In addition these are mentioned in reference section.

The following highlights the main issues and the plan of action:

- User credentials: The application has an individual area for each authenticated user accessible by a username and the matching password. The authentication details are assigned after registering with the application. Therefore the confidential information of each user is secure from leakage. The password is hashed, salted and encrypted before it is stored in the database when the user registers and resets the password. The user only knows the password.

- Storage of user information: The application will comply with the Data Protection Act (Crown copyright 2013). It stores confidential information of all the users secured by respective user credentials. Only relevant information is stored and users are informed. Users will be able to view and modify their
information. User information will be removed when the user deletes their account.

- Forgot password: The auto generation of plain text passwords were not considered due to security issues. Instead the mechanism of password-reset tokens was implemented. The password reset request sends a hyperlink to the email address of the user if registered. The hyperlink contains the reset page appended with an auto generated token number, which is associated with the particular username. Once the hyperlink is clicked, the token is retrieved and verified. In addition, if verification is successful the key-value that stored the token is removed from the database. This prevents the misuse of random tokens, as the key itself does not exist in the database.

- Sessions: The security of sessions is handled in two folds. The session IDs are regenerated at the start of every session. This prevents users to be logged into a session of an attacker. In addition, if the session is idle for more than 30 minutes then the user are logged off automatically and that session is completely destroyed. This minimises the misuse of sessions by unauthenticated users.

- Application employed by end-users: The application is primarily developed for a section of people who are diagnosed with lifestyle health diseases. However the application can be used by anyone to generate recipes without any such conditions. The application does not discriminate any sections of people.

- Results of questionnaires: The confidential information is not collected from the test users. The test users are summarised the purpose of the collected data for the application evaluation. They are requested to provide initial user approval. Their data is not used for a different purpose. If such a case arises, a new approval will be requested for the purpose.

The screenshot of the consent page is located in Appendix H.
CHAPTER 4. REQUIREMENT ANALYSIS

The aim is to motivate people diagnosed with the common lifestyle diseases to consider eating healthy without compromising on the taste factor and thereby improving their health condition. The objective is to develop a web application facilitating to suggest the appropriate recipes for the desired disease. The detailed aims and objectives are discussed in the introduction section. Additionally the motivation factors and evaluation criteria are elaborated in the literature review section.

Fundamentally functional and non-functional attributes establish the benefit of a system. Some of the attributes include usability, expandability, performance, platform and browser independence, and also security. However, there has been an asymmetrical importance laid on the functional attributes of a system despite the functional attributes is not utilisable without the requisite non-functional attributes. In addition, the theory of quality is a prime factor to software engineering like other disciplines. Furthermore, the development of a standard software system has the pre-requisites of the functional and non-functional attributes but earlier primary focus was allocated to methods on structuring the functionalities system. The critical reasons behind these were the discipline of software engineering was not that established, limited time duration for project implementation satisfying the complete requirements and also due to the flexibility of non-functional properties. The standard functionalities are only considered as technical issues, which are linked to the comprehensive design or evaluation of an implemented system. This is a commonly evident incomplete pattern that is due to the asymmetrical importance allocated to the functional section of the system. The phase of requirement analysis is essential before completing the comprehensive design or evaluation as it aids in developing a practical software solution. Additionally, solutions developed require to be altered non-functionally rather than functionally. Appropriate consideration of non-functional attributes were absent with reference to almost all the available requirements models and specification languages even though the requirements engineering community has categorised requirements into functional and non-functional (Chung and Leite 2009).
4.1. Mandatory Requirements

The mandatory requirements are sub-divided into functional and non-functional requirements.

4.1.1. Functional Requirements

The functional requirements of the application are categorised for visitors and registered users:

- Visitors
  - They cannot sign in.
  - They can sign up.
  - They can view “About Us” page.
  - They can view “How It Works” page.
  - They can view “Contact” page.
  - They can send message to developer.

- Registered Users
  - They can sign in.
  - They can upload bill.
  - They can view the recognised ingredients.
  - They can modify their profile.
  - They can reset their password.
  - Their ingredients will be stored.
  - They can view the recipe suggestions based on the profile requirements.
  - They can click and view the desired recipe.
  - They can view “About Us” page.
  - They can view “Contact” page.
  - They can view "How It Works" page.
  - They can send message to developer.
  - They can logout.

4.1.2. Non-functional Requirements

The non-functional requirements of the application are categorised into look and feel, usability, performance, availability, maintainability and security.
Look and Feel
- The interface will be appealing to all age groups.
- The interface will be simple and colourful.

Usability
- To render a user-friendly web application.
- To ensure the user interaction with the web application is simple.
- To choose suitable and consistent font sizes, font styles and colours for textual content, background and hyperlinks.
- To use hierarchical and clear navigation structure between webpages.
- To use suitable webpage titles.
- To provide easy-to-access help feature detailing the purpose and functionalities of the web application.
- To provide validation error messages for invalid or incomplete user credentials.
- To inform the user if the web application is unavailable.

Performance
- To render an acceptable response and processing time.

Availability
- The complete web application has to maintain high availability of 99.9% with a total downtime of only 8.8 hours per year that accounts for 86.40 seconds a day and 43 minutes in a month (Commerce Server 2002).

Maintainability
- To operate on cross platforms.
- To operate on any modern web browser.

Security
- To secure the user account information from unauthenticated users.
- To allow only logged in users to create user profile.

4.3. Optional Requirements

The optional requirements considered for this application are within the non-functional requirements of maintainability and security.
Maintainability

- To provide regular updates for the web application for smooth operation on cross platforms and any web browser.

Security

- To install a standard Secure Sockets Layer certificate on the web server.

4.4. Assessment Criteria

We will utilise the manual software testing methods for assessing the functional requirements of the application. The testing of non-functional requirements will be carried out in different stages.

- The user evaluation will be utilised for judging the look and feel and usability requirements also the user acceptance factor.
- Analysing the application on devices with various hardware specifications, which includes different operating system platforms and their popular modern browsers, will assess the maintainability requirements.
- The security requirements will be investigated by trying to log in with unauthorised credentials.
- The availability requirements cannot be evaluated completely as the total downtime can account to a maximum of 43 minutes in a month (Commerce Server).

We will verify the end results for all the test cases with the expected results.
CHAPTER 5. PROJECT PLAN

All projects will be assigned a planned start date and finish date. There will be several tasks, which forms the milestones of the project. Each task will have preceding or dependent task that has to be completed before the start of that particular task. Therefore a project plan was developed before commencing the project that states clear start dates, deadlines and predecessors for each task. The project plan acted as reference point for all resources until the project is completed. This aided in laying out a systematic approach to the project. The methodology, task analysis, assessment plan, also risk identification and management are discussed below.

5.1. Methodology

The methodology for development of this project utilised Waterfall and Agile models. The figure below indicates the step-by-step processes involved in the Waterfall model.

![Figure 4. Waterfall model](Guntamukkala et al. 2006)
The Waterfall model has the prerequisite of requirement gathering and analysis. The principle behind prior requirement analysis is that it allows additional control over the tasks in the project thereby maximising the success level of the project. This involves developing a comprehensive plan incorporating all the processes or tasks. In addition, the technologies mandatory to fulfil the requirements are to be recognised and evaluated (Guntamukkala et al. 2006). We have already collected and analysed all the requirements. Furthermore, the tools and technologies to be utilised are examined.

It was initially decided to implement the Waterfall model. However, in the Waterfall model, testing is only performed after the coding phase is completed. This is a risk factor if errors or problems are found towards the end of the project that utilises more resources for rectification. Sometimes, the required resources may not be sufficient. The Agile models plan iterations for shorter durations, where coding and testing are performed side by side. This ensures that planned features are working correctly and the necessary changes can be made immediately (Guntamukkala et al. 2006, Paetsch et al. 2003). Therefore, it was determined to use a hybrid of Waterfall and Agile models. The Agile model was used in database design, User Interface design and development phases. The final testing was performed according to the Waterfall model.

5.2. Task Analysis

The background research, literature review, professional, legal, ethical and social issues also the requirement analysis of the project are completed. This laid the foundation for the analysis of tasks. The timeline of the project is from 8th May 2014 to 28th August 2014. It includes all the major milestones:

- Unified Modeling Language design
- Database design
- User Interface design
- Development
- Testing
- User evaluation
- Report writing – I
- Report writing - II
- Prepare compact disk, print and bind report
- Dissertations submission
- Poster preparation
- Poster session
The timeline of the project remained the same. However there were minor changes in the duration of the major milestones. Also, the initial plan was elaborated to display all the critical milestones. The designing of the Unified Modeling Language was as per the schedule. The duration of database design was reduced, as it is easier since MongoDB followed a schema less structure. This provided an additional period for designing the user interface. The development and testing was reduced by a marginal timespan so as to allow time for preparation for the submission. The user evaluation was performed parallel to the report writing, as it was beneficial utilising the idle time since report writing is an extensive task. In addition, the documentation was narrowed down to a quick help to improve usability and allow increased timespan for report writing.

The figures that represent the finalised Gantt chart and task analysis details are located in Appendix B.

5.3. Assessment Plan

We would be assessing the web application with users from different countries so an Internet based questionnaire is the feasible selection. The main users of the application can be anyone with Internet literacy. The application caters to different age groups of people with common lifestyle diseases if any. A minimum number of ten participants will be employed for the evaluation.

The questionnaire will be semi-structured consisting of mainly closed questions and one open ended question. A completely structured questionnaire would limit the users in giving feedback not included as questions. In addition, a completely unstructured questionnaire will make the users impatient as the target users comprise of different age groups. This will result in receiving improperly filled questionnaires or even not attempted at all.

The questionnaire will be designed using the survey service provided by SurveyMonkey under their basic plan (SurveyMonkey 2014b). The user responses will be stored with SurveyMonkey. SurveyMonkey has data analysis tools to interpret data in different forms (SurveyMonkey 2014a). This is a good choice as it is a round-the-clock service and questionnaire can be completed according to the convenience of users. The evaluation will have a time period of seven days.

The closed questions aims to collect data about the user and acquiring accurate data on the different functionalities of the web application, while an open-ended questions will
aim at collecting the user feedback about the application. All the questions in the questionnaire are mandatory apart from the open-ended question that requires user feedback. The questions will emphasis on the usability, functionalities, user acceptance and the graphical user interface of the application. In addition, few questions will focus on user acceptance of the application. The concluding question will be an open-ended question to collect any other feedback from the user. The average time taken to complete the questionnaire will be fixed between 10 to 15 minutes, as the target users are a mixed group and it should not be time consuming. After the questionnaire is structured it will be tested so that the maximum time taken would be 15 minutes.

5.4. Risk Identification and Management

Risk may occur due to any sudden event during the timeframe of the project. All projects are subjected to an element of risk that may be identified or unidentified. These risks may result in increasing the time duration, estimated budget of the project or both depending on the nature of risk involved. Therefore risk should be identified along the plan of action before commencing the project. This project is also inclined to risk so these risks are required to be identified and managed.

The risk matrix and risk management table are located in Appendix B.
CHAPTER 6. SYSTEM DESIGN

In the requirement analysis phase, the functional and non-functional requirements of the application were collected. These requirements are translated into detailed technical specifications using Unified Modeling Language.

6.1. Unified Modelling Language Design

6.1.1. Use Case Analysis

An actor is a role performed by an individual or any external entity that interacts with the system. A use case specifies the type of actors; actions that the actors can perform and response of the system to the actions of actors. In other words, it indicates the interaction of actors with the system (Adolph et al. 2002).

In our system, there is only actor that is the user. The list below indicates the use cases of our system:

- Create Account: It allows a user to register an account with the application by providing name, email, desired password and health profile.
- Sign In: It allows a user to sign in to the created account by providing the username and password.
- Upload Bill: It allows a signed in user to upload a bill.
- Manage Shopping Basket: It allows a signed in user to add, modify or remove items from the shopping basket.
- Manage Account: It allows a signed in user to add, modify or remove details from their profile, in addition account can be removed.
- View Recipes: It allows a signed in user to view recipes matching their profile.
- Send Email to Developer: It allows any user to send an email to the developer.

The figure below shows the use case diagram of the application.
6.1.2. Activity Diagram

The use cases can be modelled into an activity diagram. The activities and the linkages between them are recognised from the use cases to create an activity diagram (Bastos and Ruiz 2002).

The figure below shows the activity diagram of the application.
Intelligent Recipe Advisor

Figure 6. Activity diagram of Intelligent Recipe Advisor

Tool: https://www.lucidchart.com
6.1.3. Class Diagram

The object-oriented programming was mainly used to group certain functionalities into classes using variables and methods. The classes are declared public. The variables in the Database class are declared as private, as other classes do not require individually accessing these variables. All the methods in the remaining classes are declared public. A brief description of each class is given below:

- **Display**: This class consists of all the methods used to display details and retrieve profile details.
- **User**: This class consists of all the methods linked with a user to register, update or delete an account; salt, hash and crypt a password; reset a password and update the ingredients.
- **Database**: This class consists of all the variables and a method used to create a database connection object.
- **Token**: This class consists of all the methods used to create a new token, verify if the token exists and retrieve the token.
- **Verification**: This class consists of all the methods used to check if the session is idle; verify if the username and password is matching and verify if the email address exists.
- **Redirect**: This class consists of all the methods used to redirect to the sign in page, profile page, shopping basket page and upload page.
- **Yummly**: This class consists of a method to retrieve criteria for the Yummly Application Programming Interface.

The next chapter ‘System Implementation’ consists of an elaborate illustration of these methods. The figure below outlines the class diagram of the application.
Figure 7. Class diagram of Intelligent Recipe Advisor

6.2. Database Design

intelliDelight database consist of a collection, members. The '_id' key and its value for every document in a collection is auto generated by MongoDB. It serves a unique identification key. The members collection stores all the user details. The description of the collection is given below:

- members: This collection will have one document each for a registered user. The keys used are explained below:
  - _id: auto generated
  - username: Email address of the user
  - password: hashed, salted and encrypted password of user
  - iv: initialization vector for password encryption
  - name: name of user
  - health profile: health condition of user
  - ingredients: list of ingredients
  - allergens: list of allergens
  - cuisines: list of cuisines
  - courses: list of courses
  - diet: list of diets
- height: height of user
- weight: weight of user
- token: generated number for password reset
- basket: list of ingredients recognised from bill

The figure below displays the initial collection diagrams along with the keys and its data type.

Tool: UML Lab

![Figure 8. Initial collection diagram of database](image)

The ingredients collection was created as one of the solution for an implementation challenge caused by Yummly Application Programming Interface. This is elaborated in the next chapter 'System Implementation'. This collection stores all the ingredients. The description of the collection is given below:

- ingredients: This collection will have a manually created document that stores all the ingredients available with Yummly Application Programming Interface. The keys used are explained below:
  - _id: auto generated
  - ingredients: all the ingredients available in the Yummly Application Programming Interface.

The figures below displays the final collections diagrams along with the keys and its data type.
6.3. User Interface Design

The user interface design plays a crucial role as it forms the medium of communication between the user and the application. The logo was designed using LogoGarden (Logogarden 2014). It is an easy to use logo creation application. The chef hat image was selected from their library. The colours were chosen carefully so that it would complement each other and the logo would be prominent on a white banner. The icon was derived from the logo. The figures below show the finalised logo and icon.
A customised theme of Bootstrap provided by Bootwatch called Readable is utilised for developing the graphical user interface of Intelligent Recipe Advisor (Park 2014). This framework has a collection of simple, responsive and attractive front-end elements, which make it appealing to all age groups. The background colour for all the pages was chosen as light green pastel to complement the logo and keep the overall appearance stimulating.

The navigation bar will be simple displaying all the options available. For a visitor it will be sign in, sign up, contact and about us links; and for a signed in user it will be profile, basket, view recipes and logout links. We will be placing the banner, which contains the logo and taglines of the application below it. In addition, quick help of the application is added as a ‘How It Works’ button. The body of each page is designed as per the requirements. The footer will have ‘© 2014 intelliDelight. All rights reserved.’ to the left and ‘Disclaimer’ link to the right.

The wireframes below are designed for the shopping basket page, profile and recipe matches. The remaining wireframes are located in Appendix C.

Tool: http://app.mockflow.com

Figure 12. Shopping basket page
Figure 13. Profile page
Figure 14. View recipes page
CHAPTER 7. SYSTEM IMPLEMENTATION

Prior to commencing the implementation stage, numerous factors fundamental with regard to design and analysis had been acknowledged. Implementing of the application was supported by these factors. The development of the application along with the various implementation challenges faced during the process is examined.

7.1. Coding Languages and Tools

PHP: Hypertext Processor is a commonly utilised open source that supports server-side and command-line scripting for web development that can be embedded into HyperText Markup Language. PHP is platform independent and can be used with majority of the web servers like Apache, IIS and others. It supports a wide selection of database technologies. In addition, a combination of procedural programming and object oriented programming can be employed. Also it has effective text processing functionalities. Furthermore, it has additional PHP Extension Community Library extensions (The PHP Group 2014a, The PHP Group 2014b). These were the primary reasons for choosing PHP for development of the application.

Eclipse was the integrated development environment used for writing the code. It is a platform framework with a simple user interface design, additional plugins and excellent documentation. W3Schools and Stack Overflow were used as references for clarifications during the design and development phases (W3Schools 2014, Stack Overflow 2014).

7.2. Implementation

7.2.1. Building of Functions in the Classes

Table 8. getConnection()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>getConnection()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Database</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to setup a connection with the database. The new MongoClient() is passed with the MongoDB connection. The Mongo connection class is wrapped around a singleton to retrieve the same connection object establishing the link to intelliDelight database each time getConnection() is called.</td>
</tr>
<tr>
<td>Response</td>
<td>It returns the connection object.</td>
</tr>
</tbody>
</table>
Figure 15. getConnection()  

Table 9. checkIdle()  

<table>
<thead>
<tr>
<th>Function Name</th>
<th>checkIdle()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Verification</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to check the idle time of a session.</td>
</tr>
<tr>
<td>Response</td>
<td>The user is logged out if the session is idle for more than 30 minutes otherwise session variable to store the current time is updated.</td>
</tr>
</tbody>
</table>

Figure 16. checkIdle()  

Table 10. hashCryptPassword()  

<table>
<thead>
<tr>
<th>Function Name</th>
<th>hashCryptPassword()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>User</td>
</tr>
<tr>
<td>Parameters</td>
<td>$password</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to hash and encrypt the password. This method is used to hash password by providing the algorithmic cost of 12 and a random salt is generated by the PHP password_hash(). The initialization vector (iv) is generated using PHP mcrypt_create_iv(). Then the password is encrypted using PHP mcrypt_encrypt().</td>
</tr>
<tr>
<td>Response</td>
<td>It returns true.</td>
</tr>
</tbody>
</table>
```php
function hashCryptPassword($password) {
    $options = [
        'cost' => 12
    ];
    $password_hash = hash('password', $password, HASH_SALT, BCRYPT, $options);
    $cipher = 'rijndael-256';
    $mode = 'ecb';
    $key = hash('sha256', $password_hash);
    $_SESSION['iv'] = mcrypt_create_iv(mcrypt_get_iv_size($cipher, $mode), MCRYPT_RAND);
    $_SESSION['mcrypt_e'] = base64_encode(mcrypt_encrypt($cipher, $key, $password_hash, $mode, $_SESSION['iv']));
    return true;
}
```

**Figure 17. hashCryptPassword()**

**Table 11. registerAccount()**

<table>
<thead>
<tr>
<th>Function Name</th>
<th>registerAccount()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>User</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to create an account for a newly signed up member. The details are retrieved from the form. The findOne() is used to check if the email address exists in the database. If the email address is not found a new user is created in the database with the provided details. The hashCryptPassword() is called to hash and encrypt the password. In addition, the initialization vector (iv) is stored. checkIdle() is called. Then session variables, user and name are created.</td>
</tr>
<tr>
<td>Response</td>
<td>The registration details are displayed on the page along with the 'Upload Bill' button. If the email address is found, an information message is displayed along with the 'Sign In' button.</td>
</tr>
</tbody>
</table>

```php
function registerAccount() {
    require 'includes/Database.php';
    require 'includes/Verification.php';
    $db = Database::getConnection();
    $verify = new Verification();
    $collection = $db->members;
    $username = $_POST['username'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $profile = $_POST['health_profile'];

    $query = array(
        "username" => $email
    );

    $count = $collection->findOne($query);
}
```

**Figure 18. registerAccount()**
```php
if (!count($count)) {
    $user = new User();
    $user->hashCryptPassword($password);

    $user_data = array(
        "username" => $email,
        "password" => $_SESSION['encrypted'],
        "iv" => base64_encode($_SESSION['iv']),
        "name" => $name,
        "health profile" => $profile
    );

    collection->insert($user_data);
    $_SESSION['name'] = $name;
    $_SESSION['user'] = $email;
    $verify->checkId();
    include 'includes/header.php';
}

Figure 19. registerAccount()

```
### Table 12. getUserPassword()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>getUserPassword()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Verification</td>
</tr>
<tr>
<td>Parameters</td>
<td>$email, $password</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to check if the provided username and password matches. The findOne() is used to check if the email address exists in the database. If the email address is found, the corresponding user details are retrieved. The password is decrypted using the PHP mcrypt_decrypt(). The PHP password_verify() checks that the hash after decryption matches the given password.</td>
</tr>
<tr>
<td>Response</td>
<td>If the password verification is successful then session variables, user and name are created.</td>
</tr>
</tbody>
</table>

```php
function getUserPassword($email, $password) {
    require 'includes/Database.php';
    $db = Database::getConnection();
    $collection = $db->members;
    $query = array(
        "username" => $email
    );
    $count = $collection->findOne($query, array(
        "name" => 1,
        "username" => 1,
        "password" => 1,
        "iv" => 1,
        "id" => 0
    ));
    if ($count) {
        $cipher = 'rijndael-256';
        $mode = 'cbc';
        $key = hash('crc32b', $password);
        $iv = base64_decode($count['iv']);
        $data = base64_decode($count['password']);
        $crypt_d = rtrim(mcrypt_decrypt($cipher, $key, $data, $mode, $iv), "\n");
        if (password_verify($password, $crypt_d)) {
            $_SESSION['name'] = $count['name'];
            $_SESSION['user'] = $email;
        }
    }
}
```

**Figure 21. getUserPassword()**

### Table 13. checkEmail()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>checkEmail()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Verification</td>
</tr>
<tr>
<td>Parameters</td>
<td>$email</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to verify if email address exists. The findOne() is used to check if the email address exists or not in the database.</td>
</tr>
<tr>
<td>Response</td>
<td>It returns true if email address is found otherwise returns false.</td>
</tr>
</tbody>
</table>
Figure 22. checkEmail()

```
function checkEmail($email) {
    require 'includes/Database.php';
    $db = Database::getConnection();
    $collection = $db->members;

    $query = array(
        "username" => $email
    );
    $count = $collection->findOne ($query);

    if (count ($count) {  
        $count_name = $collection->findOne ($query, array (  
            "_id" => 0
        ));
        return true;
    } else {
        return false;
    }
}
```

Table 14. updateAccount()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>updateAccount()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>User</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to save the modified profile details of the user. Health profile, height, weight, allergens, courses, cuisines, diets and ingredients fields are retrieved from the form. Then it is updated for the signed in user.</td>
</tr>
<tr>
<td>Response</td>
<td>The profile details are displayed on the page along with the 'View Recipes button.</td>
</tr>
</tbody>
</table>

Figure 23. updateAccount()

```
function updateAccount() {
    require 'includes/Database.php';
    require 'includes/Display.php';
    $db = Database::getConnection();
    $display = new Display();
    $details = null;
    $email = $SESSION ['user'];
    $profile = $_POST ['health_profile'];
    $height = floatval ($_POST ['height']);
    $weight = floatval ($_POST ['weight']);
    $allergy = preg_split ( '/\<\=\>/\n\n/\n\n/\n\n/', $_POST ['allergy']);
    $diet = preg_split ( '/\<\=\>/\n\n/\n\n/\n\n/', $_POST ['diet']);
    $cuisine = preg_split ( '/\<\=\>/\n\n/\n\n/\n\n/', $_POST ['cuisine']);
    $ingredient = preg_split ( '/\<\=\>/\n\n/\n\n/\n\n/', $_POST ['ingredient']);
    $course = preg_split ( '/\<\=\>/\n\n/\n\n/\n\n/', $_POST ['course']);

    $collection = $db->members;
    $query = array(
        "username" => $email
    );
```
function updateAccount() {
    require 'includes/Database.php';
    require 'includes/Display.php';
    $db = Database::getConnection();
    $display = new Display();
    $details = null;
    $email = $_SESSION['username'];
    $profile = $POST['health_profile'];
    $height = floatval($_POST['height']);
    $weight = floatval($_POST['weight']);
    $allergies = preg_split('/\n\n/','$_POST["allergy"]');
    $diet = preg_split('/\n\n/','$_POST["diet"]');
    $cuisine = preg_split('/\n\n/','$_POST["cuisine"]');
    $ingredient = preg_split('/\n\n/','$_POST["ingredient"]');
    $course = preg_split('/\n\n/','$_POST["course"]');
    $collection = $db->members;
    $query = array(
        "username" => $email
    );
    $details = array(
        "set" => array(
            "ingredients" => array_map ( 'trim', $ingredient ),
            "allergies" => array_map ( 'trim', $allergies ),
            "cuisines" => array_map ( 'trim', $cuisine ),
            "courses" => array_map ( 'trim', $course ),
            "diet" => array_map ( 'trim', $diet ),
            "health_profile" => $profile,
            "height" => $height,
            "weight" => $weight
        )
    );
    $collection->update ( $query, $details );
}

<?php
echo "<tr>
    <th><th>
    <th>PROFILE DETAILS<th>
</tr>
</thead>
</tbody>

Figure 24. updateAccount()
Table 15. updateIngredients()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>updateIngredients()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>User</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to save the modified shopping basket details of the user. Basket and ingredients fields are retrieved from the form. Then it is updated for the signed in user.</td>
</tr>
<tr>
<td>Response</td>
<td>The page is redirected to profile after a delay of 5 seconds.</td>
</tr>
</tbody>
</table>

```php
function updateIngredients() {
    require 'includes/Database.php';
    include 'includes/Verification.php';
    $db = Database::getConnection();
    $verify = new Verification();
    $verify->checkIdle();
    $details = null;
    $email = $SESSION['user'];
    $ingredients = preg_split(',',$POST['ingredients']);
    $basket = preg_split(',',$POST['basket']);
    $collection = $db->members;
    $query = array(
        'username' => $email
    );
    $details = array(
        '$set' => array(
            'ingredients' => array_map('trim', $ingredient),
            'basket' => array_map('trim', $basket)
        )
    );
    $collection->update( $query, $details );
    sleep ( 5 );
    header( "Location: account" );
}
```

Figure 26. updateAccount()

Figure 27. updateIngredients()
Table 16. resetPassword()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>resetPassword()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>User</td>
</tr>
<tr>
<td>Parameters</td>
<td>$email, $password</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to reset the password of the user account. hashCryptPassword() is called and passed with the password. The username is updated with the new password.</td>
</tr>
<tr>
<td>Response</td>
<td>An information message is displayed along with the ‘Sign In’ button.</td>
</tr>
</tbody>
</table>

```php
function resetPassword($email, $password) {
    require 'includes/Database.php';
    $db = Database::getConnection();
    $collection = $db->members;
    $query = array (          'username' => $email
    );
    $user = new User();
    $user->hashCryptPassword ($password);
    $details = array (          'set' => array (       "password" => $_SESSION ['encryp_e'],
            "iv" => base64_encode ($_SESSION ['iv'])
        )
    );
    $collection->update ($query, $details );
    $_SESSION = array ();
    session_destroy ();
    include 'includes/header.php';
}?
</br>
</center>
Please sign in with your new password. <br> <a href="SignIn" class="btn btn-primary" role="button">Sign In</a>
</br>
<?php
```

Figure 28. resetPassword()

Table 17. deleteAccount()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>deleteAccount()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>User</td>
</tr>
<tr>
<td>Parameters</td>
<td>$email</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to remove a user from the database.</td>
</tr>
<tr>
<td>Response</td>
<td>An information message is displayed along with the ‘Sign Up’ button.</td>
</tr>
</tbody>
</table>
Table 18. signIn()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>signIn()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Redirect</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to redirect page if the user is not authorised.</td>
</tr>
<tr>
<td>Response</td>
<td>It redirects user to sign in page if not signed in.</td>
</tr>
</tbody>
</table>

```php
function signIn() {
    if (!isset($_SESSION['name'])) {
        header('Location: SignIn');
        exit();
    }
}
```

Figure 30. signIn()

Table 19. account()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>account()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Redirect</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method allows redirect to account page when user is signed in.</td>
</tr>
<tr>
<td>Response</td>
<td>It redirects user to account page after a delay of 5 seconds.</td>
</tr>
</tbody>
</table>

```php
function account() {
    if (isset($_SESSION['name'])) {
        sleep(5);
        header('Location: account');
        exit();
    }
}
```

Figure 31. account()
Table 20. upload()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>upload()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Redirect</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method allows redirect to upload page when user is signed in.</td>
</tr>
<tr>
<td>Response</td>
<td>It redirects user to upload page after a delay of 5 seconds.</td>
</tr>
</tbody>
</table>

```
function upload() {
    if (isset ($SESSION ['name'])) {
        sleep (5);
        header ('Location: upload');
        exit 0;
    }
}
```

Figure 32. upload()

Table 21. basket()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>basket()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Redirect</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method allows redirect to basket page when user is signed in.</td>
</tr>
<tr>
<td>Response</td>
<td>It redirects user to basket page after a delay of 5 seconds.</td>
</tr>
</tbody>
</table>

```
function basket() {
    if (isset ($SESSION ['name'])) {
        sleep (5);
        header ('Location: basket');
        exit 0;
    }
}
```

Figure 33. basket()

Table 22. getToken()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>getToken()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Token</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to retrieve the token from the link.</td>
</tr>
<tr>
<td>Response</td>
<td>It returns $token.</td>
</tr>
</tbody>
</table>
Table 23. checkToken()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>checkToken()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Token</td>
</tr>
<tr>
<td>Parameters</td>
<td>$token</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to verify token. The first occurrence of findOne() is used to check if the token is present in the database. If the token is found, the second occurrence of findOne() is used retrieve the associated username with it. Then the tokens key-value is removed from the database.</td>
</tr>
<tr>
<td>Response</td>
<td>It returns true if token is valid otherwise returns false.</td>
</tr>
</tbody>
</table>

```php
function checkToken($token) {
    require '../includes/Database.php';
    $db = Database::getConnection();
    $collection = $db->members;

    $query = array(
        "tokens" => $token
    );

    $count = $collection->findOne ( $query );

    if (count (< $count )) {
        $count_username = $collection->findOne ( $query, array (            "username" => 1,            "_id" => 0          ) );
        $_SESSION ['username'] = $count_username ['username'];
        $query = array (            "username" => $_SESSION ['username']        );
    }
    $token = 0;
    $tokens = array (        "$set" => array (            "tokens" => $token        )    );
    $collection->update ( $query, $tokens );
    return true;
} else {
    return false;
}
```

Figure 35. checkToken()
Table 24. writeToken()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>writeToken()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Token</td>
</tr>
<tr>
<td>Parameters</td>
<td>$token, $email</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to write the token for the username.</td>
</tr>
<tr>
<td>Response</td>
<td>It updates token.</td>
</tr>
</tbody>
</table>

```php
function writeToken($token, $email) {
    $db = Database::getConnection();
    $collection = $db->members;
    $query = array(
        "username" => $email
    );
    $tokens = array(
        "$set" => array(
            "tokens" => $token
        )
    );
    $collection->update($query, $tokens);
}
```

Figure 36. writeToken()

Table 25. displayHeight()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>displayHeight()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Display</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to display the height of the user, if set.</td>
</tr>
<tr>
<td>Response</td>
<td>It displays height of the user.</td>
</tr>
</tbody>
</table>

```php
function displayHeight() {
    if (isset ($_SESSION["height"])) {
        echo $_SESSION["height"];  
    }
}
```

Figure 37. displayHeight()

Table 26. displayWeight()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>displayWeight()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Display</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to display the weight of the user, if set.</td>
</tr>
<tr>
<td>Response</td>
<td>It displays weight of the user.</td>
</tr>
</tbody>
</table>
Table 27. displayAllergy()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>displayAllergy()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Display</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to display the allergens of the user.</td>
</tr>
<tr>
<td>Response</td>
<td>If session variable is set, it displays each allergen in a new line.</td>
</tr>
</tbody>
</table>

```php
function displayAllergy() {
    if (isset ($_SESSION ['allergy'])) {
        echo implode ( "\n", array_map ( 'strtolower', $_SESSION ['allergy'] ));
    }
}
```

Figure 39. displayAllergy()

Table 28. displayCuisine()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>displayCuisine()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Display</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to display the preferred cuisines of the user.</td>
</tr>
<tr>
<td>Response</td>
<td>If session variable is set, it displays each cuisine in a new line.</td>
</tr>
</tbody>
</table>

```php
function displayCuisine() {
    if (isset ($_SESSION ['cuisine'])) {
        echo implode ( "\n", array_map ( 'strtolower', $_SESSION ['cuisine'] ) );
    }
}
```

Figure 40. displayCuisine()

Table 29. displayDiet()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>displayDiet()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Display</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to display the preferred diets of the user.</td>
</tr>
<tr>
<td>Response</td>
<td>If session variable is set, it displays each diet in a new line.</td>
</tr>
</tbody>
</table>
### Table 30. displayIngredient()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>displayIngredient()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Display</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to display the ingredients of the user.</td>
</tr>
<tr>
<td>Response</td>
<td>If session variable is set, it displays each ingredient in a new line.</td>
</tr>
</tbody>
</table>

```php
function displayIngredient() {
    if (isset ($_SESSION['ingredient'])) {
        $display = implode ('\n', array_map ('strtolower', $_SESSION['ingredient']));
        if (trim ($display) !== '') {
            echo $display;
        }
    }
}
```

**Figure 42. displayIngredient()**

### Table 31. displayBasket()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>displayBasket()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Display</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to display the ingredients in the basket of the user.</td>
</tr>
<tr>
<td>Response</td>
<td>If session variable is set, it displays each ingredient in a new line.</td>
</tr>
</tbody>
</table>

```php
function displayBasket() {
    if (isset ($_SESSION['basket'])) {
        $display = implode ('\n', array_map ('strtolower', $_SESSION['basket']));
        if ($display !== '') {
            echo $display;
        }
    }
}
```

**Figure 43. displayBasket()**
Table 32. displayCourse()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>displayCourse()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Display</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to display the preferred courses of the user.</td>
</tr>
<tr>
<td>Response</td>
<td>If session variable is set, it displays each course in a new line.</td>
</tr>
</tbody>
</table>

```php
function displayCourse() {
    if (isset ($_SESSION ['course'])) {
        echo implode ('\n', array_map ('strtolower', $_SESSION ['course']));
    }
}
```

Figure 44. displayCourse()

Table 33. getProfile()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>getProfile()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Display</td>
</tr>
<tr>
<td>Parameters</td>
<td>$username</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to retrieve the details required for the user profile. findOne() is used to fetch the information associated to the user account.</td>
</tr>
<tr>
<td>Response</td>
<td>Session variables are stored with the retrieved information.</td>
</tr>
</tbody>
</table>

```php
function getProfile($username) {
    require 'includes/Database.php';
    require 'includes/Verification.php';
    $db = Database::getConnection();
    $verify = new Verification();
    $collection = $db->members;
    $query = array (
        'username' => $username
    );
    $count = $collection->findOne ($query, array
        "username" => 1,
        "ingredients" => 1,
        "basket" => 1,
        "health profile" => 1,
        "allergens" => 1,
        "cuisines" => 1,
        "courses" => 1,
        "diet" => 1,
        "height" => 1,
        "weight" => 1,
        "_id" => 0
    );
}
```

Figure 45. getProfile()
Figure 46. getProfile()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>getCriteria()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Yummly</td>
</tr>
<tr>
<td>Parameters</td>
<td>$username</td>
</tr>
<tr>
<td>Description</td>
<td>This method is used to retrieve the values required for the Yummly. findOne() is used to fetch the information associated to the user account for Yummly and to calculate the body mass index of the user. The parameters and its corresponding search values for Yummly are obtained from the documentation (Yummly 2014a). The nutritional requirements for the health conditions are obtained from the literature review and with the help of a dietician.</td>
</tr>
<tr>
<td>Response</td>
<td>If the health profile, height and weight are set; the body mass index is calculated, otherwise it is redirected to the profile page. When the allergens are set each allergen found matching is prefixed with ‘&amp;excludedIngredient[]=‘. In addition, whenever ingredients and diets are set, each ingredient is prefixed with ’%20’ and each diet found matching is prefixed with search values correspondingly. Furthermore in the case the cuisines and courses are set, each cuisine is prefixed with ‘&amp;allowedCuisine[]=cuisine^cuisine-‘ and each course is prefixed with ‘&amp;allowedCourse[]=course^course’. Whenever the variables namely excluded, cuisines and courses do not have any search values; they are replaced with an empty string. Finally, each variable that is created for querying is appended to the search variable. It is stored as a session variable after passing the search variable to json_encode().</td>
</tr>
</tbody>
</table>
```javascript
function getCriteria($username) {
    require 'includes/database.php';
    require 'includes/Verification.php';
    $db = Database::getConnection();
    $verify = new Verification();
    $profile = null;
    $nutrition = null;
    $allergens = null;
    $diet = null;
    $BMI = 0;
    $cuisines = "&allowedCuisine[]=cuisine1&cuisine2-";
    $courses = "&allowedCourse[]=course1&course2-";
    $excluded = "&excludedIngredient[]=";
    $ingredients = "&ingredient=";
    $appid = "http://api.yummly.com/v1/api/recipes?app_id=2562bdsedcebb";
    $appkey = ";app_key=f0bc01e6a5f8f283c4b1844b82875369855";
    $pc = null;
    $maximum = "$maxResult-1000&start=0";

    $collection = $db->members;
    $query = array(
        "username" => $username
    );
    $count = $collection->findOne($query, array(
        "ingredients" => 1,
        "allergens" => 1,
        "diet" => 1,
        "cuisines" => 1,
        "courses" => 1,
        "health profile" => 1,
        "height" => 1,
        "weight" => 1,
        "id" => 0
    ));

    if (!$count['health profile'] && $count['weight'] && $count['height']) {
        $BMI = $count['height'] / pow($count['height'], 2);
    }

    if ($count['health profile'] == "Cardiovascular Diseases") {
        $profile = "$nutrition.K.max=1566&nutrition.CHOLE.max=100 &nutrition.ENERGY.max=660&nutrition.FIBTG.max=15&nutrition.VITC.max=20 &nutrition.CA.max=333&nutrition.FE.max=11&nutrition.FAT.max=0.5 &nutrition.VITA.UI.max=200&nutrition.PROCNT.max=95&nutrition.SUGAR.max=40";
    } else if ($BMI >= 25) {
        $profile = "$nutrition.K.max=1566&nutrition.CHOLE.max=100 &nutrition.ENERGY.max=400&nutrition.FIBTG.max=15&nutrition.VITC.max=20 &nutrition.CA.max=333&nutrition.FE.max=11&nutrition.FAT.max=0.5 &nutrition.VITA.UI.max=200&nutrition.PROCNT.max=60&nutrition.SUGAR.max=20";
    }
    else if ($count['health profile'] == "Diabetes") {
        if ($BMI < 25) {
            $profile = "$nutrition.K.max=1566&nutrition.CHOLE.max=100 &nutrition.ENERGY.max=600&nutrition.FIBTG.max=15&nutrition.VITC.max=20 &nutrition.CA.max=333&nutrition.FE.max=11&nutrition.FAT.max=0.5 &nutrition.VITA.UI.max=200&nutrition.PROCNT.max=90&nutrition.SUGAR.max=25";
        } else if ($BMI >= 25) {
            $profile = "$nutrition.K.max=1566&nutrition.CHOLE.max=100 &nutrition.ENERGY.max=500&nutrition.FIBTG.max=15&nutrition.VITC.max=20 &nutrition.CA.max=333&nutrition.FE.max=11&nutrition.FAT.max=0.5 &nutrition.VITA.UI.max=200&nutrition.PROCNT.max=75&nutrition.SUGAR.max=15";
        }
    }
    else if ($count['health profile'] == "Obesity") {
        if ($BMI < 25) {
            $profile = "$nutrition.K.max=1566&nutrition.CHOLE.max=66 &nutrition.ENERGY.max=500&nutrition.FIBTG.max=15&nutrition.VITC.max=20 &nutrition.CA.max=333&nutrition.FE.max=11&nutrition.FAT.max=0.5 &nutrition.VITA.UI.max=200&nutrition.PROCNT.max=75&nutrition.SUGAR.max=5";
        } else if ($BMI >= 25) {
            $profile = "$nutrition.K.max=1566&nutrition.CHOLE.max=66 &nutrition.ENERGY.max=400&nutrition.FIBTG.max=15&nutrition.VITC.max=20 &nutrition.CA.max=333&nutrition.FE.max=11&nutrition.FAT.max=0.5 &nutrition.VITA.UI.max=200&nutrition.PROCNT.max=60&nutrition.SUGAR.max=3";
        }
    }
    elseif ($count['health profile'] == "Normal") {
        $profile = "";
    }
}
```

Figure 47. getCriteria()

Figure 48. getCriteria()
else {
  header( 'Location:account' );
}

if (isset( $count['allergens'] )) {
  $excluded = implode( '"excludedIngredient[]"', $count['allergens'] );
  foreach( $count['allergens'] as $value ) {
    if ($value === 'wheat') {
      $allergens .= 'BallowedAllergy[]=392\wheat-free';
    }
    if ($value === 'gluten') {
      $allergens .= 'BallowedAllergy[]=393\gluten-free';
    }
    if ($value === 'peanut' || $value === 'peanuts') {
      $allergens .= 'BallowedAllergy[]=394\peanut-free';
    }
    if ($value === 'tree nut' || $value === 'tree nuts') {
      $allergens .= 'BallowedAllergy[]=395\tree-nut-free';
    }
    if ($value === 'dairy' || $value === 'dairy products' || $value === 'dairy foods') {
      $allergens .= 'BallowedAllergy[]=396\dairy-free';
    }
    if ($value === 'egg' || $value === 'eggs') {
      $allergens .= 'BallowedAllergy[]=397\egg-free';
    }
    if ($value === 'seafood' || $value === 'seafoods' || $value === 'sea foods' || $value === 'sea food') {
      $allergens .= 'BallowedAllergy[]=398\seafood-free';
    }
    if ($value === 'sesame') {
      $allergens .= 'BallowedAllergy[]=399\sesame-free';
    }
    if ($value === 'soy' || $value === 'soybeans' || $value === 'soybean') {
      $allergens .= 'BallowedAllergy[]=400\soy-free';
    }
    if ($value === 'sulfite') {
      $allergens .= 'BallowedAllergy[]=401\sulfite-free';
    }
  }
}

if (isset( $count['ingredients'] )) {
  $ingredients .= implode( '"K28", $count['ingredients'] );
}

Figure 49. getCriteria()

if (isset( $count['diet'] )) {
  foreach( $count['diet'] as $value ) {
    if ($value === 'lacto vegetarian') {
      $diet .= 'BallowedDiet[]=388\lacto-vegetarian';
    }
    if ($value === 'pescetarian') {
      $diet .= 'BallowedDiet[]=390\pescetarian';
    }
    if ($value === 'vegan') {
      $diet .= 'BallowedDiet[]=386\vegan';
    }
    if ($value === 'ovo vegetarian') {
      $diet .= 'BallowedDiet[]=389\ovo-vegetarian';
    }
    if ($value === 'vegetarian') {
      $diet .= 'BallowedDiet[]=387\lacto-ovo-vegetarian';
    }
  }
}

if (isset( $count['cuisines'] )) {
  $cuisines .= implode( '"BallowedCuisine[]=cuisine=cuisine-", $count['cuisines'] );
}

if (isset( $count['courses'] )) {
  $courses .= implode( '"allowedCourse[]=course=course-", $count['courses'] );
}

if ($excluded === '"excludedIngredient[]"')
  $excluded = '';
if ($cuisines === '"BallowedCuisine[]=cuisine=cuisine-"')
  $cuisines = '';
7.2.2. Challenges

The following elaborates the challenges faced using implementation phase of the application and the methods applied towards resolving it:

- **Average recognition rate:** Tesseract gave average results after optical character recognition of bills. The feasible way to improve the result is to increase the quality of the image. It can be achieved by image manipulation before passing it to Tesseract. However, images of poor quality cannot be improved. ImageMagick was used for manipulating the bill images.

ImageMagick can pre-process images by using its tools and libraries. It can analyse the most common image formats and transform dynamically into any image format. In addition, ‘shell’ processes can be performed in web applications. The ‘convert’ command is used with various options for image pre-processing (Mineo and Pontillo 2006, Doreva 2011). After several trials the convert command with these options was finalised, convert bill.png -colorspace gray -scale 166% -normalize -colors 32 -threshold 30% bill. ‘Upload and Recognition Module’ in the next section indicates the code.

- **Ingredients mismatch with Yummly:** The recognised ingredients mismatches with the database of ingredients in Yummly. The ingredients in Yummly were duplicated as a collection in the database of the application. This was elaborated in the previous chapter under the section ‘Database Design’. ‘Upload and Recognition Module’ in the next section demonstrates the code.

- **Yummly does not display results:** Yummly would not provide suggestion of recipes if any criterion does not match. This is programmatically resolved by adding an information message. The wireframe labelled ‘No recipes found message’ can be found in Appendix B. In addition, the wireframe labelled ‘Shopping basket page’ was modified earlier to add an ingredients text area with a copy button to transfer those particular ingredients rather than accumulating
all the ingredients of the bills. These facilities generating recipes for those ingredients. The ‘Recipe Suggestions Module’ and the ‘Shopping Basket Module’ in the next section shows the code.

- Abbreviated and local names items: Some of the stores have abbreviation of the items when printed on their bills. These items are associated with codes for few supermarkets but not all. There are also numerous local names of ingredients. The inventory database of the supermarket needs to be collected and added to the database of the application for further processing. This can be done on the basis of preferences of user to the supermarkets. In addition, it helps in incrementing the number of supermarkets with item codes. The local names of ingredients can be translated. As the scope of data collection and its implementation is time consuming, this is added to the future development of the application.

7.2.3. Building Modules

It should be noted that validations are done for all the pages using jQuery. The session_regenerate_id(true) is called wherever a session creation is involved, to replace the present session ID with a recently generated session ID. This serves as an added security measure to prevent session fixation. The core parts of the modules are explained below. The ‘success event’ mentioned in modules occurs after the successful JavaScript validation if applicable. The core modules are elaborated below and the remaining modules are located in Appendix D.

7.2.3.1. Upload and Recognition Module

<table>
<thead>
<tr>
<th>Success Event</th>
<th>Click ‘Upload Bill’ button on the ‘Upload’ page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>This handles the uploading and character recognition of the bills.</td>
</tr>
<tr>
<td>Response</td>
<td>The file types, extension and size are checked. If any errors are found it is redirected to the ‘Upload’ page otherwise the command line ‘convert’ tool preforms the image pre-processing operations to transform image as output. The command line ‘tesseract’ tool takes the output image file and transforms into a text file. Both ‘convert’ and ‘tesseract’ commands are passed to PHP exec() . Thereafter, the contents of the file is read by calling the PHP file_get_contents() and</td>
</tr>
</tbody>
</table>
stored in $filecon variable. The $filecon variable containing all the bill items is split into bill array by calling preg_split(). Each ingredient element of the array bill is compared with each ingredient in the ingredients collection, if a matching ingredient is found then that ingredient is added to the basket of the user. Finally, it is redirected to the 'Basket' page.

```php
if (!isset ($_POST ['submit']));

$allowedxts = array ('gif', 'jpeg', 'jpg', 'png');

$extension = end ($temp);

if (isset ($FILES ['file']) ['type'] == 'image/gif' || isset ($FILES ['file']) ['type'] == 'image/jpeg' || isset ($FILES ['file']) ['type'] == 'image/jpg' || isset ($FILES ['file']) ['type'] == 'image/png') || (isset ($FILES ['file']) ['size'] <= 5242880 || isset ($FILES ['file']) ['size'] <= 0) && in_array ($extension, $allowedxts)

if (!isset ($FILES ['file']) ['error']) 0

$redirect->upload ($);
```

Figure 52. read.php
### 7.2.3.2. Shopping Basket Module

This handles the basket and ingredients of the user.

**Table 35. basket.php**

<table>
<thead>
<tr>
<th>Success Event</th>
<th>Click ‘Copy To Ingredients’ button on the ‘Basket’ page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>This is used to copy the selected ingredients from the basket text area to the ingredients text area. JavaScript selectionStart and selectionEnd properties are used to find starting and position of selected text in the basket text area. The JavaScript slice() is used to retrieve the selected text by passing the start and end as parameters.</td>
</tr>
<tr>
<td><strong>Response</strong></td>
<td>The ingredients text area is updated with the values present and the selected text.</td>
</tr>
</tbody>
</table>

```javascript
$("#Copy").click(function(){
    var baskets = document.getElementById("basket");
    var selected = baskets.value.slice(baskets.selectionStart, baskets.selectionEnd);
    if (selected !=""){
        $('#ingredient').val($('#ingredient').val() + selected + '\n');
    }
});
```

**Figure 53. basket.php**

The `displayBasket()` of Display class is called in the text area of the basket.

```html
<textarea rows=10 style="resize: none" class="form-control" name="basket" id="basket" autofocus
    placeholder="Shopping basket is empty.">
    <?php $display->displayBasket();?></textarea>
```

**Figure 54. basket.php**

The `displayIngredient()` of Display class is called in the text area of the ingredients.

```html
<textarea rows=5 style="resize: none"
    placeholder="One ingredient per line." class="form-control"
    id="ingredient" name="ingredient">
    <?php $display->displayIngredient();?></textarea>
```

**Figure 55. basket.php**

**Table 36. update.php**

<table>
<thead>
<tr>
<th>Success Event</th>
<th>Click ‘Update’ button on the ‘Basket’ page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>This is used to update the shopping basket. It calls the <code>updateIngredients()</code> of User class is called.</td>
</tr>
</tbody>
</table>

```php
if (isset ( $_POST ['submit'] )) {
    $user = new User();
    $user->updateIngredients();
}
```

**Figure 56. update.php**
7.2.3.5. User Profile Module

This handles retrieving the information of the user. The getProfile() of Display class is called.

```php
$username = $_SESSION['user'];
$display->getProfile($username);
```

Figure 57. account.php

The username variable is displayed in the email text box. In the health profile drop down menu, session variable profile is displayed as the selected health profile and depending on the selected health profile, other health profiles are added as option values as shown in the two figures below.

```php
if ($_SESSION['profile'] == "Cardiovascular Diseases") {
    <option value="Diabetes">Diabetes</option>
    <option value="Obesity">Obesity</option>
    <option value="Normal">Normal</option>
}
else if ($_SESSION['profile'] == "Obesity") {
    <option value="Cardiovascular Diseases">Cardiovascular Diseases</option>
    <option value="Diabetes">Diabetes</option>
    <option value="Normal">Normal</option>
}
else if ($_SESSION['profile'] == "Diabetes") {
    <option value="Cardiovascular Diseases">Cardiovascular Diseases</option>
    <option value="Obesity">Obesity</option>
    <option value="Normal">Normal</option>
}
else if ($_SESSION['profile'] == "Normal") {
    <option value="Cardiovascular Diseases">Cardiovascular Diseases</option>
    <option value="Diabetes">Diabetes</option>
    <option value="Obesity">Obesity</option>
}
```

Figure 58. account.php

Figure 59. account.php

The displayHeight() and displayWeight() of Display class is called in the text boxes of height and weight respectively.

```php
<br> Height <input type="text" placeholder="Your height in centimeters" class="form-control" name="height" required value="%display->displayHeight();" />
<br>
Weight <input type="text" placeholder="Your weight in kilograms" class="form-control" name="weight" required value="%display->displayWeight();" />
```

Figure 60. account.php
displayAllergy() of Display class is called in the text area of allergens.

```html
textarea style="resize: none" placeholder="One allergy per line"
class="form-control" name="allergy">
<?php $display->displayAllergy();?></textarea>
```

**Figure 61. account.php**

displayCuisine() of Display class is called in the text area of cuisines.

```html
textarea style="resize: none" placeholder="One cuisine per line"
class="form-control" name="cuisine">
<?php $display->displayCuisine();?></textarea>
```

**Figure 62. account.php**

displayDiet() of Display class is called in the text area of diets.

```html
textarea style="resize: none" placeholder="One diet per line"
class="form-control" name="diet">
<?php $display->displayDiet();?></textarea>
```

**Figure 63. account.php**

displayCourse() of Display class is called in the text area of courses.

```html
textarea style="resize: none" placeholder="One course per line"
class="form-control" name="course">
<?php $display->displayCourse();?></textarea>
```

**Figure 64. account.php**

displayIngredient() of Display class is called in the text area of ingredients.

```html
textarea style="resize: none" placeholder="One ingredient per line"
class="form-control" name="ingredient">
<?php $display->displayIngredient();?></textarea>
```

**Figure 65. account.php**

### Table 37. save.php

<table>
<thead>
<tr>
<th>Success Event</th>
<th>Click ‘Update Profile’ button on the ‘Basket’ page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>This is used to update the user profile. It calls the updateAccount() of User class is called.</td>
</tr>
</tbody>
</table>

```php
if (!isset ($_POST ['submit'])) {
    $user = new User();
    $user->updateAccount();
}
```

**Figure 66. save.php**

### 7.2.3.6. Recipe Suggestions Module

This handles the suggestions of recipes. The getCriteria() of class Yummly is called.

```php
$username = $_SESSION ['user'];
$yummly->getCriteria ($username);
```

**Figure 67. suggestions.php**
This shows the <div> tags that will display the recipes and information message.

![Figure 68. suggestions.php](http://www.yummly.com/recipes)

This shows the ajax() is passed with the search session variable. The success callback is passed the data, which is in turn passed as parameter to the displayRecipes(). The error callback is passed with data and status as parameters; an information message is displayed in case of an error.

```
$(window).load(function() {
  $.ajax({
    type: "GET",
    dataType: 'jsonp',
    timeout: 5000,
    cache: false,
    url: '<?php echo $SESSION['search'] ?>',
    success: function(data) {
      console.log(data);
      displayRecipes(data);
    },
    error: function(data, status) {
      $('#found').append('<h3>'+ 'Sorry, ' + status + '!!! Please try after sometime.' + '</h3>');
    }
  });
});
```

![Figure 69. suggestions.php](http://www.yummly.com/recipes)

Table 38. displayRecipe()

<table>
<thead>
<tr>
<th>Function Name</th>
<th>displayRecipe()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>currentRecipe</td>
</tr>
<tr>
<td>Description</td>
<td>This function is used to display the recipes as a list looping through the available attributes that are retrieved.</td>
</tr>
<tr>
<td>Response</td>
<td>It displays an information message along with the 'Update Profile' button if no matching recipes are found otherwise it displays the recipes.</td>
</tr>
</tbody>
</table>
The link to access the application is [http://li719-186.members.linode.com](http://li719-186.members.linode.com). The complete source code is located in Appendix E.
CHAPTER 8. TESTING

Testing of the application is extremely critical to ensure that the features are developed as per the requirements of the analysis stage. In addition, it is required to verify that it is usable and working as expected.

8.1. Unit Testing

Unit testing is a benchmarking method for testing the functionalities of an application. An application is separated into units and each unit consists a list of its features, which are tested distinctly. In the case of unit testing, test values are provided as input to the unit to assess all of the features (Sen et al. 2005). We have employed unit testing to examine functions of the application on devices with various hardware specifications, including the popular modern browsers in different operating system platforms.

![Test Cases]

Figure 72. Summary of the test cases

It can be inferred from the test cases that all the mandatory functional requirements have been satisfied. In addition, the non-functional requirements pertaining to maintainability and security were met. Furthermore, the optional requirements were not met and are added to the future development. The test cases with the results comments are located in Appendix F.

8.2. Functionality Testing

The following elaborates the testing of the core functionalities of the application that includes uploading a grocery image, identification of ingredients, managing shopping basket, managing profile, and display of recipes suggestions.
Table 39. Test ID 62

<table>
<thead>
<tr>
<th>Test ID</th>
<th>62</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Uploading of grocery bill image</td>
</tr>
<tr>
<td>Prerequisites</td>
<td>User is signed in and on ‘Upload’ page.</td>
</tr>
</tbody>
</table>
| Steps | 1. Click on ‘Choose File…’ button.  
2. Select the image and click ‘Upload Bill’ button. |
| Expected result comments | The image is uploaded and saved with a temporary name on the server. |
| Actual result comments | The image is uploaded with a temporary name to the ’/tmp’ folder server. |
| Status | Passed |

Table 40. Test ID 63

<table>
<thead>
<tr>
<th>Test ID</th>
<th>63</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Identification of ingredients</td>
</tr>
<tr>
<td>Prerequisites</td>
<td>User is signed in and has the image is uploaded.</td>
</tr>
<tr>
<td>Steps</td>
<td>1. User waits for the process to complete.</td>
</tr>
<tr>
<td>Expected result comments</td>
<td>All the ingredients on the bill are to be identified which depends on the quality of the image and the ingredients present in the database of the application</td>
</tr>
<tr>
<td>Actual result comments</td>
<td>The following ingredients were identified that contains accurate and incorrect matches: bran, eggs, hen, coconut, chutney, coco, sugar, su, papaya, pears, bread, cake, rusk, hand, uni, oil, emu, milk and mi. The correct matches are: bran, eggs, sugar, papaya and milk. Coconut chutney and rusk cake ingredients were divided into two ingredients.</td>
</tr>
<tr>
<td>Status</td>
<td>Passed</td>
</tr>
</tbody>
</table>

Table 41. Test ID 64

<table>
<thead>
<tr>
<th>Test ID</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Managing shopping basket</td>
</tr>
<tr>
<td>Prerequisites</td>
<td>User is signed in and on ‘Basket’ page.</td>
</tr>
</tbody>
</table>
| Steps | 1. Modify and/ remove any ingredients.  
2. Selects the required ingredients from the basket.  
3. Click ‘Copy To Ingredients’.  
4. Repeat if necessary.  
5. Click ‘Update’. |
| Expected result comments | The basket and ingredients are updated in the database. |
| Actual result comments | The basket and ingredients are updated in the database and page is redirected to ’Profile’. |
| Status | Passed |
Table 42. Test ID 65

<table>
<thead>
<tr>
<th>Test ID</th>
<th>65</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Managing profile</td>
</tr>
<tr>
<td>Prerequisites</td>
<td>User is signed in and on 'Profile' page.</td>
</tr>
</tbody>
</table>
| Steps | 1. Enter height and weight  
2. Add preferences, if any.  
3. Click 'Update Profile' button. |
| Expected result comments | The profile details are updated in the database. |
| Actual result comments | The profile details are updated in the database and page is redirected to display profile details. |
| Status | Passed |

Table 43. Test ID 66

<table>
<thead>
<tr>
<th>Test ID</th>
<th>66</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Display of recipe suggestions</td>
</tr>
</tbody>
</table>
| Prerequisites | User is signed in  
Steps | 1. Click 'View Recipes' |
| Expected result comments | Matching recipes are displayed. |
| Actual result comments | Six matching recipes are displayed. |
| Status | Passed |

The image of the grocery bill and screenshots of the functionality testing are located in Appendix G.
CHAPTER 9. USER EVALUATION

In the chapter ‘Project Plan’, the ‘Assessment Plan’ was described. The plan was to allow a group of people to test the application for a short period and then complete a short online questionnaire. The user evaluation was conducted by a group of 13 people aged above 21 years. The link for requesting consent of their participation was provided by email or instant messaging, which included the access to the questionnaire. The test users were requested to use the application for a few days so that they can try all its features and then fill in the questionnaire. Furthermore, they were reminded to follow the instructions, which are explained in the ‘How It Works’ page. The table below indicates the questions and responses of the users.

Table 4.4. Summary of user evaluation responses

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>RESPONSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Do you usually use an application for recipe suggestions?</td>
<td></td>
</tr>
<tr>
<td>RESPONSES</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>53.85%</td>
</tr>
<tr>
<td>If no, please specify the reason</td>
<td></td>
</tr>
<tr>
<td>• I’m not much of a cook -- also have Crohn's disease</td>
<td></td>
</tr>
<tr>
<td>• I don’t cook</td>
<td></td>
</tr>
<tr>
<td>• I’m a foodie, just love to eat!!</td>
<td></td>
</tr>
<tr>
<td>• I haven't started experimenting with lives yet.</td>
<td></td>
</tr>
<tr>
<td>• I was not aware of the existence of such applications</td>
<td></td>
</tr>
<tr>
<td>• Have only just become familiar to it</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>46.15%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>RESPONSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Would you like any other health profiles to be included in the application?</td>
<td></td>
</tr>
<tr>
<td>RESPONSES</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>38.46%</td>
</tr>
<tr>
<td>If yes, please specify the reason</td>
<td></td>
</tr>
<tr>
<td>• Digestive Diseases and Disorders - gluten-free - peanut free - vegan - vegetarian</td>
<td></td>
</tr>
<tr>
<td>• Nutritious based</td>
<td></td>
</tr>
<tr>
<td>• Weight gain, sedentary, active</td>
<td></td>
</tr>
<tr>
<td>• Irritable bowel syndrome</td>
<td></td>
</tr>
<tr>
<td>• Digestive diseases</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>61.54%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>RESPONSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Totally, how many times did you use the upload bill feature?</td>
<td></td>
</tr>
<tr>
<td>RESPONSES</td>
<td></td>
</tr>
<tr>
<td>Never</td>
<td>0.00%</td>
</tr>
<tr>
<td>1 - 3 times</td>
<td>92.31%</td>
</tr>
<tr>
<td>4 - 6 times</td>
<td>7.69%</td>
</tr>
<tr>
<td>7 - 10 times</td>
<td>0.00%</td>
</tr>
<tr>
<td>More than 10 times</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>RESPONSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. How you rate the accuracy of recognised ingredients from the bills?</td>
<td></td>
</tr>
<tr>
<td>RESPONSES</td>
<td></td>
</tr>
<tr>
<td>100%</td>
<td>0.00%</td>
</tr>
</tbody>
</table>
### Question 5. How do you rate the design of the application in relation to layout, colour and graphics?

**Responses**
- Excellent: 15.38%
- Good: 30.77%
- Satisfactory: 53.85%
- Unsatisfactory: 0.00%

### Question 6. Do you find the application easy to use?

**Responses**
- Yes: 92.31%
- No: 7.69%

### Question 7. How do you rate the recipe suggestions?

**Responses**
- Excellent: 30.77%
- Good: 38.46%
- Satisfactory: 30.77%
- Unsatisfactory: 0.00%

### Question 8. Do you find the idea of this application to be innovative and interesting?

**Responses**
- Yes: 100.00%
- No: 0.00%

### Question 9. Would you be keen in using this application for your daily/weekly recipe suggestions?

**Responses**
- Yes: 84.62%
- No: 15.38%

### Question 10. Do you have any other comments, questions, or concerns?

**Responses**
- Could be a little more user friendly and attractive.
- Besides the graphics and marketing, I think this is a great idea. I've thought there should be a site like this too! I'm not sure if it is included, but there should also be a way to create and print out a shopping list. Is there a way to save or manage your existing recipes? I also love the line "Nothing prevents you from healthy cooking!" So many people make excuses for unhealthy eating habits and this is a great tool for people not to make excuses anymore!
- Good idea. Gets reliable results. The Interaction flow of the application can be improved. It would be better if the flow followed a 'conversation' approach.
- Very good.
- 1) Would like to get some info after file upload is done, especially if the app fails to pick items 2) Rather than the current set up to 'copy to ingredients', displaying the items in a multi-select list would be more
user friendly (multi-select or group them in divs that can be selected, should be able to find some plugins for that) 3) The navigation links - please highlight them when on the corresponding page 4) In recipe section I would have liked to search for recipes myself as well, not just the ones based on my profile (strictly my opinion, that's all) 5) The success rate is quite low, but I can appreciate the complexity of the idea, so its fine for now (that's why I asked for a recipe search - since most of my items were not scanned, I got no suggestions) 6) Design, colour scheme & font are ok as well 7) Issue with tooltips - would be better if the previous tooltip disappears when I click on a new one (profiles section)

- Nil
- Nothing at this stage.

Figure 73. Do you usually use an application for recipe suggestions?

The overall, user feedbacks of the application have been reasonably positive. There was some variation in the number of test users who were found to use applications for recipe suggestions in comparison with the users who never used such applications. The primary reasons for not using any applications were that they did not cook, only love to eat, did not want to experiment with applications and were not aware of such applications. This indicates that people are open to using technology in their lives but the reason mentioned-above becomes a hindrance. The application aims to remove some of these hindrances.

Figure 74. Would you like any other health profiles to be included in the application?
The users have requested for some additions to the health profile that includes digestive diseases and disorders; nutritious based; weight gain, sedentary, active and irritable bowel syndrome. It is seen that the requests to include digestive diseases is prominent which could be a rising lifestyle disease. A sedentary or active condition is necessary for each health profile and is a valuable suggestion. Presently recipe suggestions for the normal health profile are only retrieved based on default considerations of Yummly. In order to provide better nutritious recipe results for a normal profile, further research will be conducted. These profiles will be studied and incorporated into the updates of the application.

![Bar Chart](image1.png)

**Figure 75.** Totally, how many times did you use the upload bill feature?

![Bar Chart](image2.png)

**Figure 76.** How you rate the accuracy of recognised ingredients from the bills?

The test users did not test the upload feature considerably as would be expected. However their responses demonstrate a wide variation in the recognition rate of ingredients, the application was able to accurately identify from the images of grocery bills. The accuracy rate varied from 90 to even below 50 percentage. Majority of test users claim that the accuracy was between 80 to 89 percentages, while some claimed that the accuracy was 90 to 99 percentage. The wide fluctuation in rates could be mainly due to the quality of the grocery bill images, presence of abbreviated items or local names of ingredients. This will be solved in the future version updates.
Almost all of the test users agreed that the application was easy to use and the overall user experience was ranked as desirable. The recipe suggestions were rated good by an average number of users. The user interface was designed to be simple, attractive and interactive with the user, however the minor details were excluded due to the limitation of time. One user mentioned that the application was not easy to use. It has been requested to improve the user friendliness; additionally one user has suggested a 'conversation' approach. Furthermore there is a request to replace the 'Copy To Ingredients' button with multi-select list. This was planned initially to be incorporated but due to the complexity involved in implementing editing of identified ingredients, it was added as future development.
All the test users experienced that the concept of the application was stimulating and original. Also it motivated majority of the users to employ the application for their daily or weekly recipe suggestions by even those who never used such applications. Moreover, the user feedbacks have been supportive, productive and valuable. The application has satisfied its objectives but the recognition of ingredients is required to worked on to provide optimum results in most cases. The application fulfilled the objective to encourage users to cooking delicious and healthy meals. The users have mentioned the issues, provided suggestions and constructive ideas to improve the application that can be incorporated into the future versions.

The screenshot of the consent page, questionnaire and individual responses of the test users are located in Appendix H.
CHAPTER 10. CONCLUSION

10.1. Limitations

The application that is developed has certain limitations due to restriction of the time duration and complexity. The optional requirements of version updates and installing Secure Sockets Layer certificate were not undertaken. It primarily focuses only on the lifestyles diseases such as cardiovascular diseases, obesity and diabetes. The analysis of bills only works for non-abbreviated and international names of ingredients. Additionally, the ingredients are limited to those in Yummly. The recipe suggestions will be only based on a single health condition at a time. Presently members of the same family will have to create separate profiles. There is no separate application developed for mobile devices. Furthermore, there is no option to view nearby restaurants serving the same dishes.

10.2. Accomplishments

This dissertation has been a valuable source of accomplishments. It is important to define the lessons learnt in numerous phases in the development of this project. The following enumerates the lessons learned by the author:

- The initial planning, background study and risk management of the problem plays an extremely critical role. The successful completion of a project depends on careful planning and risk mitigation.
- Learnt the presence of front-end frameworks that helps developers with limited designing skills to implement solutions with an appealing graphical user interface.
- It aided in refining and building up the knowledge of UML techniques, also had the opportunity to learn various diagramming tools.
- Identified the presence of the online applications that aid in easy creation of logos by saving time and eliminating the requisite of high-end designing software.
- Helped in considerably improving the knowledge towards implementation of PHP programing language and collaborating it with HTML, JavaScript and jQuery.
- Studied and executed different methods to test and evaluate a project before the official release.
10.3. Future Development

The future development of the application aims to remove the previously mentioned limitations. A ‘conversation’ approach to the application will be added. Further research will be conducted on more lifestyle diseases as requested by the test users and it will be incorporated. The basket of ingredients will be an editable multi-check list. The recipe suggestions can be made available to multiple health conditions and ignore option will be made available. Also option to include family members to the profile will be an added functionality. In addition, option to create and print shopping list will be added. A separate application will be developed for mobile devices. Furthermore a choice would be provided to view the nearby restaurants serving the same dishes.

10.4. Summary

This research report documents the literature review, requirement analysis, design, implementation, testing and evaluation of the development of an Intelligent Recipe Advisor that utilises character recognition to identify ingredients from grocery bills. The aims and objectives of this project were to make delicious recipe suggestions thereby encourage healthy eating among individuals diagnosed with lifestyle diseases.

All the mandatory requirements of the project were satisfied. The learning curve was quite steep due to the limited time duration. However, the experience achieved is invaluable in relation to background study, planning, critical thinking and problem solving. The accomplishments from this project will lay strong foundations for the projects the author will undertake in the future. The work of the author has been beneficial and was recognised by the people utilising this application according to the requisites. This motivates the author to enhance or add new functionalities to the application in the future.
References


Food2Fork (2014a) 'Food2Fork | Publishers', [online], available: [http://food2fork.com/publishers](http://food2fork.com/publishers) [Accessed 02/05/2014].

Food2Fork (2014b) 'Recipe API | Food2Fork', [online], available: [http://food2fork.com/about/api](http://food2fork.com/about/api) [Accessed 02/05/2014].


Leiner, B. M., Cerf, V. G., Clark, D. D., Kahn, R. E., Kleinrock, L., Lynch, D. C., Postel, J.,


Liu, S. (2003) 'Whole-grain foods, dietary fiber, and type 2 diabetes: searching for a

LogoGarden (2014) 'Free Logo Design Online | Logo Garden', [online], available:

intake is favorably associated with metabolic risk factors for type 2 diabetes and
cardiovascular disease in the Framingham Offspring Study', The American


Milligan, I. (2013) 'Illusionary Order: Online Databases, Optical Character Recognition,
and Canadian History, 1997–2010', The Canadian Historical Review, 94(4), 540-
569.


MongoDB (2014) 'NoSQL Databases Explained', [online], available:

and fiber intake and the incidence of type 2 diabetes', The American Journal of

Mozilla Developer Network (2014) 'Getting started with app development', [online],
[Accessed 03/18/2014].
Oxford University Press (2014) 'Internet: definition of Internet in Oxford dictionary (British & World English)', [online], available: 
http://www.oxforddictionaries.com/definition/english/Internet
[Accessed 03/18/2014].


Park, T. (2014) 'Bootswatch: Readable', [online], available: 


Power, D. J. (2003) 'What is the difference between formative and summative DSS evaluation?', DSS News, 4(2) available: 

RecipeMatcher (2012) 'Recipes by Ingredients - RecipeMatcher', [online], available: 


Su Chef (2014) 'Su Chef', [online], available: [http://su-chef.com](http://su-chef.com) [Accessed 06/05/2014].


W3Schools (2014) 'W3Schools Online Web Tutorials', [online], available: 
http://www.w3schools.com/ [Accessed 05/17/2014].

Yummly (2014a) 'Yummly | Documentation', [online], available: 
https://developer.yummly.com/documentation [Accessed 02/05/2014].

Yummly (2014b) 'Yummly | Recipe API & Food API', [online], available: 
http://developer.yummly.com/ [Accessed 02/05/2014].

Yummly (2014c) 'Yummly | Terms and Conditions - Yummly API', [online], available: 
http://developer.yummly.com/policies [Accessed 02/05/2014].

Bibliography


Appendices

Appendix A

A.1. Database Creation

Tool: Robomongo

Step 1: Open Robomongo.

Step 2: (a) Click on “create” link. In the “Connection” tab, enter “Name” and “Address” fields. The default port number of MongoDB is 27017.

Figure 82. Connection tab
(b) In the "Authentication" tab, enter “User Name” and “Password” fields.

![Figure 83. Authentication tab]

(c) Click on “Test” button. Click on “Close” button. Then click on "Save" button.

![Figure 84. Connection success alert]
Step 3: Select “MongoDB”. Click on “Connect” button.

![Figure 85. MongoDB selected from connections window](image.png)

Step 4: (a) Right click on “MongoDB”. Then click on “Create Database”.

![Figure 86. Create Database selected from menu of MongoDB connection](image.png)
(b) Enter “intelliDelight” as “Database Name”. Click on “Create” button.

![Create Database](image)

*Figure 87. Create IntelliDelight database*

Step 5: (a) Expand “intelliDelight”. Right click on “Collections”. Click on “Create Collection...”.

![Collections menu](image)

*Figure 88. Collections menu*
(b) Enter “members” as “Collection Name”. Click on “Create” button.

![Create members collection in intelliDelight database](image1)

Figure 89. Create members collection in intelliDelight database

(c) Repeat step 5: (a). Enter “ingredients” as “Collection Name”.

![Create ingredients collection in intelliDelight database](image2)

Figure 90. Create ingredients collection in intelliDelight database

Step 6: (a) Expand “Collections”. Right click on “Insert Document…”

![Menu for ingredients collection](image3)

Figure 91. Menu for ingredients collection
The list of ingredients is obtained from Yummly by using the end point http://api.yummly.com/v1/api/metadata/ingredient?_app_id=YOUR_ID&_app_key=YOUR_APP_KEY, where YOUR_ID is the application ID and YOUR_APP_KEY is the application key provided upon registration with Yummly as a developer (Yummly 2014a). Then all the duplicate values were removed and modified to retain a single array of ingredient key and the values. In addition, "/" were replaced with appropriate values. The file was saved as "ingredients.txt".

(b) Copy and paste the contents of “ingredients.txt” within the parenthesis. “ingredients.txt” is provided in the compact disk.

(c) Click on “Validate” button. Click on “OK” button. Then click on “Save” button.
(d) Double click on “ingredients”.

Figure 94. Documents in ingredients collection

Step 7: Right click on “MongoDB (3)”. Then click on “Disconnect”.

Figure 95. Disconnect selected from menu of MongoDB connection
Appendix B

B.1. Project Plan

Tool: ProjectLibre

Figure 96. Gantt chart
Table 45. Analysis of tasks

<table>
<thead>
<tr>
<th>Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
<th>Predecessors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 UL design</td>
<td>5 days</td>
<td>08/05/14</td>
<td>12/05/14</td>
<td></td>
</tr>
<tr>
<td>2 Use case diagram</td>
<td>1 day</td>
<td>08/05/14</td>
<td>08/05/14</td>
<td></td>
</tr>
<tr>
<td>3 Activity diagram</td>
<td>3 days</td>
<td>09/05/14</td>
<td>11/05/14</td>
<td>2</td>
</tr>
<tr>
<td>4 Class diagram</td>
<td>1 day</td>
<td>12/05/14</td>
<td>12/05/14</td>
<td>3</td>
</tr>
<tr>
<td>5 Database design</td>
<td>2 days</td>
<td>13/05/14</td>
<td>14/05/14</td>
<td>1</td>
</tr>
<tr>
<td>6 Database diagram</td>
<td>1 day</td>
<td>13/05/14</td>
<td>13/05/14</td>
<td>7</td>
</tr>
<tr>
<td>7 Create database</td>
<td>1 day</td>
<td>14/05/14</td>
<td>14/05/14</td>
<td>6</td>
</tr>
<tr>
<td>8 User interface design</td>
<td>7 days</td>
<td>15/05/14</td>
<td>21/05/14</td>
<td>5</td>
</tr>
<tr>
<td>9 Sign in and sign up module</td>
<td>1 day</td>
<td>15/05/14</td>
<td>15/05/14</td>
<td></td>
</tr>
<tr>
<td>10 Home, help and contact module</td>
<td>1 day</td>
<td>16/05/14</td>
<td>16/05/14</td>
<td></td>
</tr>
<tr>
<td>11 Upload module</td>
<td>1 day</td>
<td>17/05/14</td>
<td>17/05/14</td>
<td></td>
</tr>
<tr>
<td>12 Basket module</td>
<td>1 day</td>
<td>18/05/14</td>
<td>18/05/14</td>
<td></td>
</tr>
<tr>
<td>13 Profile module</td>
<td>1 day</td>
<td>19/05/14</td>
<td>19/05/14</td>
<td></td>
</tr>
<tr>
<td>14 View recipes module</td>
<td>1 day</td>
<td>20/05/14</td>
<td>20/05/14</td>
<td></td>
</tr>
<tr>
<td>15 About us and reset password module</td>
<td>1 day</td>
<td>21/05/14</td>
<td>21/05/14</td>
<td></td>
</tr>
<tr>
<td>16 Development</td>
<td>38 days</td>
<td>22/05/14</td>
<td>28/06/14</td>
<td>8</td>
</tr>
<tr>
<td>17 Password module</td>
<td>4 days</td>
<td>22/05/14</td>
<td>25/05/14</td>
<td></td>
</tr>
<tr>
<td>18 Sign up module</td>
<td>1 day</td>
<td>26/05/14</td>
<td>26/05/14</td>
<td>17</td>
</tr>
<tr>
<td>19 Sign in and logout module</td>
<td>1 day</td>
<td>27/05/14</td>
<td>27/05/14</td>
<td>17:18</td>
</tr>
<tr>
<td>20 Upload and recognition module</td>
<td>17 days</td>
<td>28/05/14</td>
<td>13/06/14</td>
<td>18:19</td>
</tr>
<tr>
<td>21 Basket module</td>
<td>4 days</td>
<td>14/06/14</td>
<td>17/06/14</td>
<td>20</td>
</tr>
<tr>
<td>22 Profile module</td>
<td>2 days</td>
<td>18/06/14</td>
<td>19/06/14</td>
<td>21</td>
</tr>
<tr>
<td>23 View recipes module</td>
<td>8 days</td>
<td>20/06/14</td>
<td>27/06/14</td>
<td>22</td>
</tr>
<tr>
<td>24 Contact module</td>
<td>1 day</td>
<td>26/06/14</td>
<td>26/06/14</td>
<td></td>
</tr>
<tr>
<td>25 Help module</td>
<td>1 day</td>
<td>28/06/14</td>
<td>28/06/14</td>
<td>20:21:22:23</td>
</tr>
<tr>
<td>26 Testing</td>
<td>13 days</td>
<td>29/06/14</td>
<td>11/07/14</td>
<td>16</td>
</tr>
<tr>
<td>27 Write test cases</td>
<td>6 days</td>
<td>29/06/14</td>
<td>04/07/14</td>
<td></td>
</tr>
<tr>
<td>28 Create test data</td>
<td>2 days</td>
<td>05/07/14</td>
<td>06/07/14</td>
<td>27</td>
</tr>
<tr>
<td>29 Perform tests</td>
<td>3 days</td>
<td>07/07/14</td>
<td>09/07/14</td>
<td>28</td>
</tr>
<tr>
<td>30 Fix code</td>
<td>2 days</td>
<td>10/07/14</td>
<td>11/07/14</td>
<td>29</td>
</tr>
<tr>
<td>31 User evaluation</td>
<td>7 days</td>
<td>25/07/14</td>
<td>31/07/14</td>
<td>26</td>
</tr>
<tr>
<td>32 Report writing – I</td>
<td>20 days</td>
<td>12/07/14</td>
<td>31/07/14</td>
<td>26</td>
</tr>
<tr>
<td>33 Report writing – II</td>
<td>18 days</td>
<td>01/08/14</td>
<td>18/08/14</td>
<td>31:32</td>
</tr>
<tr>
<td>34 Prepare CD, print and bind report</td>
<td>2 days</td>
<td>19/08/14</td>
<td>20/08/14</td>
<td>33</td>
</tr>
<tr>
<td>35 Dissertation submission</td>
<td>1 day</td>
<td>21/08/14</td>
<td>21/08/14</td>
<td>34</td>
</tr>
<tr>
<td>36 Poster preparation</td>
<td>4 days</td>
<td>22/08/14</td>
<td>25/08/14</td>
<td>35</td>
</tr>
<tr>
<td>37 Poster session</td>
<td>1 day</td>
<td>28/08/14</td>
<td>28/08/14</td>
<td>36</td>
</tr>
</tbody>
</table>

Figure 97. Risk matrix (Probability-Impact)
<table>
<thead>
<tr>
<th>SL. NO.</th>
<th>RISKS</th>
<th>PROBABILITY-IMPACT</th>
<th>ACTION PLAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Not getting access to the required software and/or tools</td>
<td>Medium-Medium</td>
<td>Ensure that the required software and tools will be obtained before the project initiation</td>
</tr>
<tr>
<td>2.</td>
<td>Learning the Tesseract OCR engine, MongoDB and Yummly Application Programming Interface takes longer than expected</td>
<td>Low-High</td>
<td>Allotting adequate time for learning phase before the project initiates</td>
</tr>
<tr>
<td>3.</td>
<td>Inability to continue on the project because of sickness or any other reason</td>
<td>Low-Medium</td>
<td>Incorporating maximum slack time as possible</td>
</tr>
<tr>
<td>4.</td>
<td>Project supervisor inaccessibility because of sickness, busy schedule or any other reason</td>
<td>Low-Medium</td>
<td>Change tasks order wherever possible that requires mediation of the supervisor to the start of the project</td>
</tr>
<tr>
<td>5.</td>
<td>Errors in code</td>
<td>Medium-Medium</td>
<td>Conduct software testing whenever there are major modifications in the code</td>
</tr>
<tr>
<td>6.</td>
<td>Lack for participants for evaluation</td>
<td>Low-Low</td>
<td>Link to the questionnaire will be circulated so it will ensure a considerable level of participation</td>
</tr>
</tbody>
</table>
Appendix C

C.1. Wireframes

Tool: http://app.mockflow.com

Figure 98. MongoDB connection error message

Figure 99. Home page
Figure 100. Sign up page

Figure 101. Registration details page
Intelligent Recipe Advisor

Figure 102. User exists message

Figure 103. Sign in page
Figure 104. Incorrect username or password message

Figure 105. Upload page
# Instructions

1. The scanned copies or digital photos of grocery bills in English language must be uploaded as image files. The supported file types are gif, jpeg, jpg and png. The maximum file size permitted are 10MB. The recommended file size is 5MB or less. The supported file types are in portrait layout. Upload and analysis is in progress.

2. Please place the bill on a white background for optimal recognition. If crumbled, smoothen the bill.

3. The recognition rate depends on the document quality (resolution, contrast, Brightness, exposure etc.) to enhance the image before upload.

4. While the uploading of an image file is in progress, please do not refresh or reload the webpage. Once the image file is analysed, you will be redirected to the "shopping basket" page. The recognition may not be 100% accurate, errors may be present in displayed ingredients. You can review and make changes as necessary or add missing items.

Figure 106. Alert message during upload

Figure 107. Alert message during update
Figure 108. About us page

Figure 109. Disclaimer page
Figure 110. Profile details page

Figure 111. No recipes found message
Figure 112. Account deleted message

Figure 113. Contact page
Figure 114. Thank you message

Figure 115. Forgot password page
Password Reset Confirmation

We will send the password reset information to the e-mail address "<abc@domain>".

Generally, our password reset e-mail reaches you within few minutes but occasionally there may be delays caused by e-mail delivery systems due to unanticipated issues. We will send the e-mail from the "noreply@intellige.com". We suggest that you add "noreply@intellige.com" to your contact list. This will ensure that our e-mail does not go into your junk/spam e-mail folder or gets deleted automatically.

Figure 116. Password reset confirmation

Figure 117. User not registered message
Appendix D

D.1. Header and Footer Module

This “header.php” and “footer.html” files included in all the PHP files that require a display of the page on the browser.

The <link> tags include the Bootstrap cascading style-sheets.

```
<link href="css/bootstrap.css" media="screen">
<link href="css/bootstrap.min.css" rel="stylesheet">
```

The “About Us” and “Contact” are aligned to the left while “Sign In” and “Sign Up” is aligned to the right of the navigation bar. If the user is signed in, “Profile”, “Upload”, “Basket”, “View Recipes”, and “Logout” replace “Sign In” and “Sign Up”. The verification of signed in users is done with the help of $_SESSION['name'] variable and “Welcome,” name of the user is displayed on the navigation bar.
This includes the code for the banner along with the logo and taglines.

```html
<div class="container">
  <!-- Main jumbotron for a primary marketing message or call to action -->
  <div class="jumbotron">
    ...
    <img src="/images/intellidelight.png" class="img-rounded alt="logo">
    <p>Your personal intelligent recipe advisor. Nothing prevents you from healthy cooking!!!</p>
    <a href="help" class="btn btn-primary btn-lg role="button">How It Works?</a>
  </div>
</div>
```

**Figure 121. Header.php - Banner**

The `<script>` tags include the jQuery and Bootstrap JavaScript.

```html
<script src="/ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
<script src="/js/bootstrap.min.js"></script>
```

**Figure 122. Header.php - Script tags**

The footer contains the copyright and disclaimer.

```html
<div id="footer">
  <div class="container">
    <p style="display: inline" class="text-muted">© 2014 Intellidelight. All rights reserved.</p>
    <a style="float: right" href="disclaimer">Disclaimer</a>
  </div>
</div>
```

**Figure 123. footer.php**
### D.2. Sign Up Module

Table 47. register.php

<table>
<thead>
<tr>
<th>Success Event</th>
<th>Click ‘Sign Up’ button on the ‘Sign Up’ page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>This handles the registration of all new users. It calls the <code>registerAccount()</code> of the User class.</td>
</tr>
</tbody>
</table>

```php
if (isset ( $_POST ['submit'] )) {
    require 'includes/User.php';
    $user = new User();
    $user->registerAccount();
}
```

**Figure 124. register.php**

### D.3. Sign In Module

Table 48. verify.php

<table>
<thead>
<tr>
<th>Success Event</th>
<th>Click ‘Sign In’ button on the ‘Sign In’ page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>This handles the authorisation of all registered users. It calls the <code>getUserPassword()</code> of the Verification class.</td>
</tr>
<tr>
<td>Response</td>
<td>If the session variable name is created it is redirected to ‘Upload’ page, otherwise an information message along with ‘Sign In’ button is displayed.</td>
</tr>
</tbody>
</table>

```php
if (isset ( $_POST ['submit'] )) {
    $username = $_POST ['email'];
    $password = $_POST ['password'];
    $verify->getUserPassword ( $username, $password );

    if (isset ( $_SESSION ['name'] )) {
        header ( 'location: upload' );
    } else {
        echo nl2br ( "<center><nSorry, username or password is not valid. Please try again." );
    }
} else {
    <br>
    <br>
    <a href="SignIn" class="btn btn-primary" role="button">Sign In</a>
</center>
</div>
</php>
```

**Figure 125. verify.php**

### D.4. Logout Module

Table 49. logout.php

<table>
<thead>
<tr>
<th>Success Event</th>
<th>Click ‘Logout’ button</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>This handles the logout of users. The session array is removed by calling the PHP session_destroy() and redirected to sign in page.</td>
</tr>
</tbody>
</table>
### D.5. Password Reset Module

#### Table 50. forgot.php

<table>
<thead>
<tr>
<th>Success Event</th>
<th>Click ‘Rest’ button on the ‘Forgot page’</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>This handles the password resets of all registered users. The email address is verified calling the checkEmail() of Verification class. If it exists a token number is generated using the rand(). In addition, the writeToken() of Token class is called to write the generated token for the specified username and an information message is displayed.</td>
</tr>
<tr>
<td><strong>Response</strong></td>
<td>If the email address exists a reset password link is sent to the user otherwise an information message along with the ‘Sign Up’ button is displayed.</td>
</tr>
</tbody>
</table>

```php
if (isset ( $POST ['submit'] )) {
    $email = $_POST ['email'];
    if (!$verify->checkEmail ( trim($email ))) {
        $token = rand ( 10000, 10000000 );
        $tokens->writeToken ( $token, $email );
        $subject = "Reset your password on IntelliDelight";
        $url = 'http://'. $_SERVER ['HTTP_HOST'] . '/forgot.php';
        $message = "Reset your password on IntelliDelight"
        $subject = "Reset your password on IntelliDelight";
        $url = 'http://'. $_SERVER ['HTTP_HOST'] . '/forgot.php';
        $message = "Reset your password on IntelliDelight"
    }
    $email = trim($email);
    if ($verify->checkEmail ( $email )) {
        $token = $tokens->lookupToken ( $email );
        if ($token) {
            $headers = 'From: IntelliDelight <noreply@intelligidelight.com>';
            if (isset ($headers)) {
                $headers = $headers . 'X-Mailer: IntelliDelight';
            }
            if (isset ($body)) {
                $body = $body . 'X-Mailer: IntelliDelight';
            }
            $headers = "X-Mailer: IntelliDelight";
            $body = "X-Mailer: IntelliDelight";
            $subject = "Reset your password on IntelliDelight";
            $url = 'http://'. $_SERVER ['HTTP_HOST'] . '/forgot.php';
            $message = "Reset your password on IntelliDelight"
        } else {
            $subject = 'Forgot password request for "' . $email . '" has been rejected';
            $message = 'Your request to reset your password has been rejected. The email address you provided may not be valid or it may already have been used to reset a password. Please try again with a valid email address.';
            $url = 'http://'. $_SERVER ['HTTP_HOST'] . '/forgot.php';
        }
        if (isset ($body)) {
            $body = $body . 'X-Mailer: IntelliDelight';
        }
        $message = 'A password reset link has been sent to your email address. You can follow the link to reset your password. If you do not receive the email, please check your spam folder.
        
        If you did not request a password reset, or if you do not remember your password, please reset your password. If you still have questions or need further assistance, please contact us at support@intelligidelight.com.';
        $headers = 'From: IntelliDelight <noreply@intelligidelight.com>';
        $headers = $headers . 'X-Mailer: IntelliDelight';
        $headers = "$headers"
        $subject = "Reset your password on IntelliDelight";
        $url = 'http://'. $_SERVER ['HTTP_HOST'] . '/forgot.php';
        $message = "Reset your password on IntelliDelight"
    } else {
        $subject = 'Email address not found';
        $message = 'The email address you provided was not found in our database. Please check your spelling and try again.';
        $url = 'http://'. $_SERVER ['HTTP_HOST'] . '/forgot.php';
    }
    $headers = 'From: IntelliDelight <noreply@intelligidelight.com>';
    $headers = $headers . 'X-Mailer: IntelliDelight';
    $headers = "$headers"
    $subject = "Reset your password on IntelliDelight";
    $url = 'http://'. $_SERVER ['HTTP_HOST'] . '/forgot.php';
    $message = "Reset your password on IntelliDelight"
}
else {
    echo '<br>Sorry, your email address is not registered with us. Please sign up, or <a href="/signUp.php" role="Button">Sign Up</a>.';  
</center>
</div>
</div>
</div>
</div>

Figure 127. send.php
Table 51. reset.php

<table>
<thead>
<tr>
<th>Success Event</th>
<th>Click reset password link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>This retrieves the token by calling the getToken(). The checkToken() verifies the token. Both the functions belong to Token class.</td>
</tr>
<tr>
<td>Response</td>
<td>If the token number is valid the reset password form is displayed, otherwise an information message is displayed.</td>
</tr>
</tbody>
</table>

```php
$token = $tokens->getToken();
if ($tokens->checkToken ($token)) {
    //
}
```

Figure 128. reset.php

```html
<form class="form-signin" role="form" id="reset" method="post">
    <input type="password" id="password" name="password" placeholder="New password" required>
    <input type="password" class="form-control" name="confirm_password" placeholder="Retype new password" required>
    <button class="btn btn-lg btn-primary btn-block" type="submit" value="Reset Password" name="submit" id="submit"></button>
</form>
```

Figure 129. reset.php

```php
} else {
    echo '<br><center>Sorry, this is an expired link.</center><br>';
}
```

Figure 130. reset.php

Table 52. Reset.php and ResetPass.php

<table>
<thead>
<tr>
<th>Success Event</th>
<th>Click ‘Reset Password’ button on ‘ResetPass’ page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The resetPassword() of User class is called to update the password for the specified username.</td>
</tr>
</tbody>
</table>

```php
if (!isset ($POST ['submit'])) {
    $user=new User();
    $email = $SESSION ['username'];
    $password = $POST ['password'];
    $user->resetPassword($email, $password);
}
```

Figure 131. RestPass.php

Appendix E

E.1. Source Code

It is available in the attached compact disk.
F.1. Test Cases

<table>
<thead>
<tr>
<th>Browsers</th>
<th>Devices</th>
<th>Operating Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safari, Opera, Firefox, Chrome, and Internet Explorer</td>
<td>Notebook, laptop, tablet and smartphone</td>
<td>OS X, Windows, Linux, iOS and Android</td>
</tr>
</tbody>
</table>

Pre-condition for test IDs from 1 to 6: User is on "Sign In" page

Table 53. Test case ID 1

<table>
<thead>
<tr>
<th>Test ID</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign in with registered username and correct password</td>
</tr>
</tbody>
</table>
| Steps | 1. Enter registered username and correct password  
2. Click "Sign In" button |
| Expected result comments | Test case is valid |
| Actual result comments | Test case is accepted |
| Status | Passed |

Table 54. Test case ID 2

<table>
<thead>
<tr>
<th>Test ID</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign in with registered username and incorrect password</td>
</tr>
</tbody>
</table>
| Steps | 1. Enter registered username and incorrect password  
2. Click "Sign In" button |
| Expected result comments | Test case is invalid. Application will not allow user to sign in. |
| Actual result comments | Test case is rejected with appropriate action. |
| Status | Passed |

Table 55. Test case ID 3

<table>
<thead>
<tr>
<th>Test ID</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign in with invalid username and password</td>
</tr>
</tbody>
</table>
| Steps | 1. Enter invalid username and password  
2. Click "Sign In" button |
| Expected result comments | Test case is invalid. Application will not allow user to sign in. |
| Actual result comments | Test case is rejected with appropriate action. |
| Status | Passed |

Table 56. Test case ID 4

<table>
<thead>
<tr>
<th>Test ID</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign in with unregistered username and blank password</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Enter unregistered username and leave</td>
</tr>
</tbody>
</table>
Table 57. Test case ID 5

<table>
<thead>
<tr>
<th>Test ID</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign in with registered username and blank password</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Enter registered username and leave password blank 2. Click “Sign In” button</td>
</tr>
<tr>
<td>Expected result comments</td>
<td>Test case is invalid. Application will not allow user to sign in.</td>
</tr>
<tr>
<td>Actual result comments</td>
<td>Test case is rejected with appropriate action.</td>
</tr>
<tr>
<td>Status</td>
<td>Passed</td>
</tr>
</tbody>
</table>

Table 58. Test case ID 6

<table>
<thead>
<tr>
<th>Test ID</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign in with both fields blank.</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Leave both fields blank. 2. Click “Sign In” button</td>
</tr>
<tr>
<td>Expected result comments</td>
<td>Test case is invalid. Application will not allow user to sign in.</td>
</tr>
<tr>
<td>Actual result comments</td>
<td>Test case is rejected with appropriate action.</td>
</tr>
<tr>
<td>Status</td>
<td>Passed</td>
</tr>
</tbody>
</table>

Pre-condition for test ID from 7 to 21: User is on “Sign Up” page

Table 59. Test case ID 7

<table>
<thead>
<tr>
<th>Test ID</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign up with name, valid email address, password, retyped same password and selected health profile</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Enter name, valid email address, password with minimum 8 characters, retype same password and select health profile 2. Click “Sign Up” button</td>
</tr>
<tr>
<td>Expected result comments</td>
<td>Test case is valid</td>
</tr>
<tr>
<td>Actual result comments</td>
<td>Test case is accepted</td>
</tr>
<tr>
<td>Status</td>
<td>Passed</td>
</tr>
</tbody>
</table>

Table 60. Test case ID 8

<table>
<thead>
<tr>
<th>Test ID</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign up with all fields blank</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Leave all fields blank 2. Click “Sign Up” button</td>
</tr>
</tbody>
</table>
| Expected result comments | Test case is invalid. Application will not allow user to
<table>
<thead>
<tr>
<th>Test ID</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign up with only name field blank</td>
</tr>
</tbody>
</table>
| Steps | 1. Leave name field blank and fill the remaining fields with valid input  
2. Click “Sign Up” button |
| Expected result comments | Test case is invalid. Application will not allow user to sign up. |
| Actual result comments | Test case is rejected with appropriate action. |
| Status | Passed |

Table 61. Test case ID 9

<table>
<thead>
<tr>
<th>Test ID</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign up with only email field blank</td>
</tr>
</tbody>
</table>
| Steps | 1. Leave email field blank and fill the remaining fields with valid input  
2. Click “Sign Up” button |
| Expected result comments | Test case is invalid. Application will not allow user to sign up. |
| Actual result comments | Test case is rejected with appropriate action. |
| Status | Passed |

Table 62. Test case ID 10

<table>
<thead>
<tr>
<th>Test ID</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign up with only password field blank</td>
</tr>
</tbody>
</table>
| Steps | 1. Leave password field blank and fill the remaining fields with valid input  
2. Click “Sign Up” button |
| Expected result comments | Test case is invalid. Application will not allow user to sign up. |
| Actual result comments | Test case is rejected with appropriate action. |
| Status | Passed |

Table 63. Test case ID 11

<table>
<thead>
<tr>
<th>Test ID</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign up with only retype password field blank</td>
</tr>
</tbody>
</table>
| Steps | 1. Leave retype password field blank and fill the remaining fields with valid input  
2. Click “Sign Up” button |
| Expected result comments | Test case is invalid. Application will not allow user to sign up. |
| Actual result comments | Test case is rejected with appropriate action. |
| Status | Passed |

Table 64. Test case ID 12
### Table 65. Test case ID 13

<table>
<thead>
<tr>
<th>Test ID</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign up without selecting health profile field</td>
</tr>
</tbody>
</table>
| **Steps** | 1. Don’t select health profile field and fill the remaining fields with valid input  
2. Click “Sign Up” button |
| **Expected result comments** | Test case is invalid. Application will not allow user to sign up. |
| **Actual result comments** | Test case is rejected with appropriate action. |
| **Status** | Passed |

### Table 66. Test case ID 14

<table>
<thead>
<tr>
<th>Test ID</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign up with only name field filled</td>
</tr>
</tbody>
</table>
| **Steps** | 1. Fill name field with valid input and leave the remaining fields blank  
2. Click “Sign Up” button |
| **Expected result comments** | Test case is invalid. Application will not allow user to sign up. |
| **Actual result comments** | Test case is rejected with appropriate action. |
| **Status** | Passed |

### Table 67. Test case ID 15

<table>
<thead>
<tr>
<th>Test ID</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign up with only email field filled</td>
</tr>
</tbody>
</table>
| **Steps** | 1. Fill email field with valid input and leave the remaining fields blank  
2. Click “Sign Up” button |
| **Expected result comments** | Test case is invalid. Application will not allow user to sign up. |
| **Actual result comments** | Test case is rejected with appropriate action. |
| **Status** | Passed |

### Table 68. Test case ID 16

<table>
<thead>
<tr>
<th>Test ID</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign up with only password field filled</td>
</tr>
</tbody>
</table>
| **Steps** | 1. Fill password field with valid input and leave the remaining fields blank  
2. Click “Sign Up” button |
| **Expected result comments** | Test case is invalid. Application will not allow user to sign up. |
| **Actual result comments** | Test case is rejected with appropriate action. |
| **Status** | Passed |

### Table 69. Test case ID 17

<table>
<thead>
<tr>
<th>Test ID</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Sign up with only retype password field filled</td>
</tr>
</tbody>
</table>
| **Steps** | 1. Fill retype password field with valid input and leave the remaining fields blank  
2. Click “Sign Up” button |
| **Expected result comments** | Test case is invalid. Application will not allow user to sign up. |
| **Actual result comments** | Test case is rejected with appropriate action. |
| **Status** | Passed |

Table 70. Test case ID 18

| **Test ID** | 18 |
| **Test Case** | Sign up by only selecting health profile field |
| **Steps** | 1. Select health profile field and leave the remaining fields blank  
2. Click “Sign Up” button |
| **Expected result comments** | Test case is invalid. Application will not allow user to sign up. |
| **Actual result comments** | Test case is rejected with appropriate action. |
| **Status** | Passed |

Table 71. Test case ID 19

| **Test ID** | 19 |
| **Test Case** | Sign up only with invalid email address |
| **Steps** | 1. Fill email address field with invalid format and the remaining fields with valid input  
2. Click “Sign Up” button |
| **Expected result comments** | Test case is invalid. Application will not allow user to sign up. |
| **Actual result comments** | Test case is rejected with appropriate action. |
| **Status** | Passed |

Table 72. Test case ID 20

| **Test ID** | 20 |
| **Test Case** | Sign up only with invalid password |
| **Steps** | 1. Fill password field with few spaces and other characters or characters less than 8 and the remaining fields with valid input  
2. Click “Sign Up” button |
| **Expected result comments** | Test case is invalid. Application will not allow user to sign up. |
| **Actual result comments** | Test case is rejected with appropriate action. |
| **Status** | Passed |

Table 73. Test case ID 21

| **Test ID** | 21 |
| **Test Case** | Sign up only with invalid retype password field |
| **Steps** | 1. Fill all fields with valid input except retype password field. Fill it with characters not matching the password.  
2. Click “Sign Up” button |
### Expected result comments
Test case is invalid. Application will not allow user to sign up.

### Actual result comments
Test case is rejected with appropriate action.

### Status
Passed

Pre-condition for test ID from 22 to 24: User is signed in and on "Upload" page

#### Table 74. Test case ID 22

<table>
<thead>
<tr>
<th>Test ID</th>
<th>22</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Upload by selecting a valid file type and within allowed limit</td>
</tr>
</tbody>
</table>
| Steps | 1. Choose a file type that is gif, jpeg, jpg or png within 5 MB but not 0 MB size  
2. Click "Upload Bill" button |
| Expected result comments | Test case is valid |
| Actual result comments | Test case is accepted |
| Status | Passed |

#### Table 75. Test case ID 23

<table>
<thead>
<tr>
<th>Test ID</th>
<th>23</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Upload by not selecting a file</td>
</tr>
</tbody>
</table>
| Steps | 1. Don’t choose a file  
2. Click "Upload Bill" button |
| Expected result comments | Test case is invalid. Application will not allow user to upload. |
| Actual result comments | Test case is rejected with appropriate action. |
| Status | Passed |

#### Table 76. Test case ID 24

<table>
<thead>
<tr>
<th>Test ID</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Upload by selecting an invalid file type or not within the allowed limit</td>
</tr>
</tbody>
</table>
| Steps | 1. Choose a file type that is not gif, jpeg, jpg or png or over 5 MB or 0 MB size  
2. Click "Upload Bill" button |
| Expected result comments | Test case is invalid. Application will not allow user to upload. |
| Actual result comments | Test case is rejected with appropriate action. |
| Status | Passed |

Pre-condition for test ID from 25 to 27: User is signed in and on "Shopping Basket" page

#### Table 77. Test case ID 25

<table>
<thead>
<tr>
<th>Test ID</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Copy items in basket field to ingredients field</td>
</tr>
</tbody>
</table>
| Steps | 1. Select one item at a time or adjacent items together  
2. Click "Copy To Ingredients" button  
3. Click "Update" button |
### Table 78. Test case ID 26

<table>
<thead>
<tr>
<th>Test ID</th>
<th>26</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Add new items in basket or ingredients fields</td>
</tr>
</tbody>
</table>
| Steps | 1. Enter new items into basket and/ingredients field  
2. Click “Update” button |
| Expected result comments | Test case is valid. |
| Actual result comments | Test case is accepted. |
| Status | Passed |

### Table 79. Test case ID 27

<table>
<thead>
<tr>
<th>Test ID</th>
<th>27</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Delete or edit items in basket or ingredients fields</td>
</tr>
</tbody>
</table>
| Steps | 1. Remove and/modify items from basket and/ingredients field  
2. Click “Update” button |
| Expected result comments | Test case is valid. |
| Actual result comments | Test case is accepted. |
| Status | Passed |

Pre-condition for test ID from 28 to 31: User is signed in and on "Profile" page

### Table 80. Test case ID 28

<table>
<thead>
<tr>
<th>Test ID</th>
<th>28</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Add profile with height, weight, allergens, preferred cuisines, preferred diets, preferred courses, ingredients</td>
</tr>
</tbody>
</table>
| Steps | 1. Enter height greater than or equal to 50, weight greater than or equal to 5, and/fill the remaining fields with input  
2. Click “Update” button |
| Expected result comments | Test case is valid. |
| Actual result comments | Test case is accepted. |
| Status | Passed |

### Table 81. Test case ID 29

<table>
<thead>
<tr>
<th>Test ID</th>
<th>29</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Edit profile with height, weight, allergens, preferred cuisines, preferred diets, preferred courses, ingredients</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Modify height greater than or equal to 50, weight greater than or equal to 5, and/remove and/modify values of the remaining fields</td>
</tr>
</tbody>
</table>
### Table 82. Test case ID 30

<table>
<thead>
<tr>
<th>Test ID</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Update profile with height and weight fields blank</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Leave height and weight fields blank, and/ or fill the remaining fields with input&lt;br&gt;2. Click &quot;Update&quot; button</td>
</tr>
<tr>
<td>Expected result comments</td>
<td>Test case is invalid. Application will not allow user to update.</td>
</tr>
<tr>
<td>Actual result comments</td>
<td>Test case is rejected with appropriate action.</td>
</tr>
<tr>
<td>Status</td>
<td>Passed</td>
</tr>
</tbody>
</table>

Pre-condition for test ID 32: User is signed in and profile is updated with height and weight

### Table 83. Test case ID 31

<table>
<thead>
<tr>
<th>Test ID</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Delete account</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Click &quot;Delete Account&quot; button</td>
</tr>
<tr>
<td>Expected result comments</td>
<td>Test case is valid.</td>
</tr>
<tr>
<td>Actual result comments</td>
<td>Test case is accepted.</td>
</tr>
<tr>
<td>Status</td>
<td>Passed</td>
</tr>
</tbody>
</table>

### Table 84. Test case ID 32

<table>
<thead>
<tr>
<th>Test ID</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>View recipes</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Click &quot;View Recipes&quot;</td>
</tr>
<tr>
<td>Expected result comments</td>
<td>Test case is valid.</td>
</tr>
<tr>
<td>Actual result comments</td>
<td>Test case is accepted.</td>
</tr>
<tr>
<td>Status</td>
<td>Passed</td>
</tr>
</tbody>
</table>

Pre-condition for test ID 33: User is signed in and profile does not have height and weight entries

### Table 85. Test case ID 33

<table>
<thead>
<tr>
<th>Test ID</th>
<th>33</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>View recipes</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Click &quot;View Recipes&quot;</td>
</tr>
<tr>
<td>Expected result comments</td>
<td>Test case is invalid. Application will not allow user to view recipes.</td>
</tr>
<tr>
<td>Actual result comments</td>
<td>Test case is rejected with appropriate action.</td>
</tr>
<tr>
<td>Status</td>
<td>Passed</td>
</tr>
</tbody>
</table>
Pre-condition for test ID from 34 to 36: User is signed in or a visitor and on "How It Works" page

<table>
<thead>
<tr>
<th>Table 86. Test case ID 34</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test ID</td>
</tr>
<tr>
<td>Test Case</td>
</tr>
<tr>
<td>Steps</td>
</tr>
<tr>
<td>Expected result comments</td>
</tr>
<tr>
<td>Actual result comments</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 87. Test case ID 35</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test ID</td>
</tr>
<tr>
<td>Test Case</td>
</tr>
<tr>
<td>Steps</td>
</tr>
<tr>
<td>Expected result comments</td>
</tr>
<tr>
<td>Actual result comments</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 88. Test case ID 36</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test ID</td>
</tr>
<tr>
<td>Test Case</td>
</tr>
<tr>
<td>Steps</td>
</tr>
<tr>
<td>Expected result comments</td>
</tr>
<tr>
<td>Actual result comments</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>

Pre-condition for test ID from 37 and 38: User is signed in or a visitor and on "How It Works" page

<table>
<thead>
<tr>
<th>Table 89. Test case ID 37</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test ID</td>
</tr>
<tr>
<td>Test Case</td>
</tr>
<tr>
<td>Steps</td>
</tr>
<tr>
<td>Expected result comments</td>
</tr>
<tr>
<td>Actual result comments</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 90. Test case ID 38</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test ID</td>
</tr>
<tr>
<td>Test Case</td>
</tr>
<tr>
<td>Steps</td>
</tr>
<tr>
<td>Expected result comments</td>
</tr>
<tr>
<td>Actual result comments</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>
Pre-condition for test ID from 39 and 40: User is a visitor and on “Home” page

<table>
<thead>
<tr>
<th>Table 91. Test case ID 39</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test ID</strong></td>
</tr>
<tr>
<td><strong>Test Case</strong></td>
</tr>
<tr>
<td><strong>Steps</strong></td>
</tr>
<tr>
<td><strong>Expected result comments</strong></td>
</tr>
<tr>
<td><strong>Actual result comments</strong></td>
</tr>
<tr>
<td><strong>Status</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 92. Test case ID 40</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test ID</strong></td>
</tr>
<tr>
<td><strong>Test Case</strong></td>
</tr>
<tr>
<td><strong>Steps</strong></td>
</tr>
<tr>
<td><strong>Expected result comments</strong></td>
</tr>
<tr>
<td><strong>Actual result comments</strong></td>
</tr>
<tr>
<td><strong>Status</strong></td>
</tr>
</tbody>
</table>

Pre-condition for test ID from 41 to 45: User is signed in or a visitor and on “Contact” page

<table>
<thead>
<tr>
<th>Table 93. Test case ID 41</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test ID</strong></td>
</tr>
<tr>
<td><strong>Test Case</strong></td>
</tr>
<tr>
<td><strong>Steps</strong></td>
</tr>
<tr>
<td><strong>Expected result comments</strong></td>
</tr>
<tr>
<td><strong>Actual result comments</strong></td>
</tr>
<tr>
<td><strong>Status</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 94. Test case ID 42</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test ID</strong></td>
</tr>
<tr>
<td><strong>Test Case</strong></td>
</tr>
<tr>
<td><strong>Steps</strong></td>
</tr>
<tr>
<td><strong>Expected result comments</strong></td>
</tr>
<tr>
<td><strong>Actual result comments</strong></td>
</tr>
<tr>
<td><strong>Status</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 95. Test case ID 43</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test ID</strong></td>
</tr>
<tr>
<td><strong>Test Case</strong></td>
</tr>
<tr>
<td><strong>Steps</strong></td>
</tr>
</tbody>
</table>
### Expected result comments
Test case is invalid. Application will not allow user to sign up.

### Actual result comments
Test case is rejected with appropriate action.

### Status
Passed

---

#### Table 96. Test case ID 44

<table>
<thead>
<tr>
<th>Test ID</th>
<th>44</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Send message with only email field blank</td>
</tr>
</tbody>
</table>
| Steps | 1. Leave email field blank and fill the remaining fields with valid input  
2. Click “Send” button |
| Expected result comments | Test case is invalid. Application will not allow user to sign up. |
| Actual result comments | Test case is rejected with appropriate action. |
| Status | Passed |

#### Table 97. Test case ID 45

<table>
<thead>
<tr>
<th>Test ID</th>
<th>45</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Send message with only message field blank</td>
</tr>
</tbody>
</table>
| Steps | 1. Leave password field blank and fill the remaining fields with valid input  
2. Click “Send” button |
| Expected result comments | Test case is invalid. Application will not allow user to sign up. |
| Actual result comments | Test case is rejected with appropriate action. |
| Status | Passed |

Pre-condition for test ID from 46 and 47: User is a visitor and on “Sign In” page

#### Table 98. Test case ID 46

<table>
<thead>
<tr>
<th>Test ID</th>
<th>46</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>View forgot password page</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Click “Forgot password…”</td>
</tr>
<tr>
<td>Expected result comments</td>
<td>Test case is valid.</td>
</tr>
<tr>
<td>Actual result comments</td>
<td>Test case is accepted.</td>
</tr>
<tr>
<td>Status</td>
<td>Passed</td>
</tr>
</tbody>
</table>

#### Table 99. Test case ID 47

<table>
<thead>
<tr>
<th>Test ID</th>
<th>47</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>View sign up page</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Click “It’s free, sign up…” or “Sign Up”</td>
</tr>
<tr>
<td>Expected result comments</td>
<td>Test case is valid.</td>
</tr>
<tr>
<td>Actual result comments</td>
<td>Test case is accepted.</td>
</tr>
<tr>
<td>Status</td>
<td>Passed</td>
</tr>
</tbody>
</table>
Pre-condition for test ID 48: User is a visitor and on “Sign Up” page

<table>
<thead>
<tr>
<th>Table 100. Test case ID 48</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test ID</td>
</tr>
<tr>
<td>Test Case</td>
</tr>
<tr>
<td>Steps</td>
</tr>
<tr>
<td>Expected result comments</td>
</tr>
<tr>
<td>Actual result comments</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>

Pre-condition for test ID from 49 to 51: User is signed in

<table>
<thead>
<tr>
<th>Table 101. Test case ID 49</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test ID</td>
</tr>
<tr>
<td>Test Case</td>
</tr>
<tr>
<td>Steps</td>
</tr>
<tr>
<td>Expected result comments</td>
</tr>
<tr>
<td>Actual result comments</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 102. Test case ID 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test ID</td>
</tr>
<tr>
<td>Test Case</td>
</tr>
<tr>
<td>Steps</td>
</tr>
<tr>
<td>Expected result comments</td>
</tr>
<tr>
<td>Actual result comments</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 103. Test case ID 51</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test ID</td>
</tr>
<tr>
<td>Test Case</td>
</tr>
<tr>
<td>Steps</td>
</tr>
<tr>
<td>Expected result comments</td>
</tr>
<tr>
<td>Actual result comments</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>

Pre-condition for test ID 52: User is signed in and on “View Recipes” page with recipes listed.

<table>
<thead>
<tr>
<th>Table 104. Test case ID 52</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test ID</td>
</tr>
<tr>
<td>Test Case</td>
</tr>
<tr>
<td>Steps</td>
</tr>
<tr>
<td>Expected result comments</td>
</tr>
<tr>
<td>Actual result comments</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>
Pre-condition for test ID from 53 to 55: User is not signed in and on “Forgot Password” page

Table 105. Test case 53

<table>
<thead>
<tr>
<th>Test ID</th>
<th>53</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Get reset link with registered email address</td>
</tr>
</tbody>
</table>
| Steps | 1. Enter registered email address  
2. Click “Reset” button |
| Expected result comments | Test case is valid. |
| Actual result comments | Test case is accepted. |
| Status | Passed |

Table 106. Test case 54

<table>
<thead>
<tr>
<th>Test ID</th>
<th>54</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Get reset link with unregistered email address</td>
</tr>
</tbody>
</table>
| Steps | 1. Enter unregistered email address  
2. Click “Reset” button |
| Expected result comments | Test case is invalid. Application will not send reset link to user. |
| Actual result comments | Test case is rejected with appropriate action. |
| Status | Passed |

Table 107. Test case 55

<table>
<thead>
<tr>
<th>Test ID</th>
<th>55</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Get reset link with invalid or blank email address</td>
</tr>
</tbody>
</table>
| Steps | 3. Enter invalid or blank email address  
4. Click “Reset” button |
| Expected result comments | Test case is invalid. Application will not send reset link to user. |
| Actual result comments | Test case is rejected with appropriate action. |
| Status | Passed |

Pre-condition for test ID from 56 to 61: User is not signed in and on “Reset Password” page

Table 108. Test case 56

<table>
<thead>
<tr>
<th>Test ID</th>
<th>56</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Reset password with filled password fields</td>
</tr>
</tbody>
</table>
| Steps | 1. Enter password with minimum 8 characters and retype same password  
2. Click “Reset Password” button |
| Expected result comments | Test case is valid |
| Actual result comments | Test case is accepted |
| Status | Passed |

Table 109. Test case 57

<table>
<thead>
<tr>
<th>Test ID</th>
<th>57</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Reset password with blank fields</td>
</tr>
</tbody>
</table>
### Table 110. Test case ID 58

<table>
<thead>
<tr>
<th>Test ID</th>
<th>58</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Reset password with password field blank</td>
</tr>
</tbody>
</table>
| Steps | 1. Leave password field blank and fill retype password with valid input  
2. Click “Reset Password” button |
| Expected result comments | Test case is invalid. Application will not send reset link to user. |
| Actual result comments | Test case is rejected with appropriate action. |
| Status | Passed |

### Table 111. Test case ID 59

<table>
<thead>
<tr>
<th>Test ID</th>
<th>59</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Reset password with retype password field blank</td>
</tr>
</tbody>
</table>
| Steps | 1. Leave retype password field blank and fill password with valid input  
2. Click “Reset Password” button |
| Expected result comments | Test case is invalid. Application will not send reset link to user. |
| Actual result comments | Test case is rejected with appropriate action. |
| Status | Passed |

### Table 112. Test case ID 60

<table>
<thead>
<tr>
<th>Test ID</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Reset password with invalid password field</td>
</tr>
</tbody>
</table>
| Steps | 1. Fill password field with few spaces and other characters or characters less than 8 and the retype password field with valid input  
2. Click “Reset Password” button |
| Expected result comments | Test case is invalid. Application will not send reset link to user. |
| Actual result comments | Test case is rejected with appropriate action. |
| Status | Passed |

### Table 113. Test case ID 61

<table>
<thead>
<tr>
<th>Test ID</th>
<th>61</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case</td>
<td>Reset password with invalid retype password field</td>
</tr>
</tbody>
</table>
| Steps | 1. Fill password field with valid input and retype password field with characters not matching the password.  
2. Click “Reset Password” button |
<table>
<thead>
<tr>
<th>Desc. / Item No.</th>
<th>Qty.</th>
<th>Rate</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kelloggs Special X 435 Gm. Box</td>
<td>1 ea</td>
<td>170.00</td>
<td>170.00</td>
</tr>
<tr>
<td>Healthy Heart Rice Bran Refine G 11</td>
<td>1 ea</td>
<td>170.00</td>
<td>-17.00</td>
</tr>
<tr>
<td>Healthy Heart Rice Bran Refine G 11</td>
<td>1 ea</td>
<td>170.00</td>
<td>170.00</td>
</tr>
<tr>
<td>Poultry Chicken Eggs Pack of 6</td>
<td>1 ea</td>
<td>36.00</td>
<td>36.00</td>
</tr>
<tr>
<td>Poultry Chicken Eggs Pack of 6</td>
<td>1 ea</td>
<td>36.00</td>
<td>-3.90</td>
</tr>
<tr>
<td>Lipton Fresh Brew Moutwash 80 ml</td>
<td>1 ea</td>
<td>40.00</td>
<td>40.00</td>
</tr>
<tr>
<td>Mars, Classic Kitchen Promos, 388 Gm. Mix</td>
<td>1 ea</td>
<td>34.00</td>
<td>34.00</td>
</tr>
<tr>
<td>Mars, Classic Kitchen Promos, 388 Gm. Mix</td>
<td>1 ea</td>
<td>34.00</td>
<td>-4.90</td>
</tr>
<tr>
<td>12128 185/100X5463</td>
<td>1 ea</td>
<td>38.00</td>
<td>38.00</td>
</tr>
<tr>
<td>Van Cleef Choco Bar 100 Gm.</td>
<td>1 ea</td>
<td>25.00</td>
<td>25.00</td>
</tr>
<tr>
<td>Van Cleef Choco Bar 100 Gm.</td>
<td>1 ea</td>
<td>25.00</td>
<td>-3.75</td>
</tr>
<tr>
<td>Mars. Choice Superior Sugar, 1 Kg</td>
<td>1 ea</td>
<td>30.00</td>
<td>30.00</td>
</tr>
<tr>
<td>Mars. Choice Superior Sugar, 1 Kg</td>
<td>1 ea</td>
<td>30.00</td>
<td>-5.00</td>
</tr>
<tr>
<td>Surf Excel Natic Detergent Powder Fsc</td>
<td>1 ea</td>
<td>230.00</td>
<td>230.00</td>
</tr>
<tr>
<td>Pears Pure &amp; Gentle 3 x 125 Gm.</td>
<td>1 ea</td>
<td>168.00</td>
<td>168.00</td>
</tr>
<tr>
<td>Pears Pure &amp; Gentle 3 x 125 Gm.</td>
<td>1 ea</td>
<td>168.00</td>
<td>-2.40</td>
</tr>
<tr>
<td>Banana, Cavendish</td>
<td>1 ea</td>
<td>42.00</td>
<td>42.00</td>
</tr>
<tr>
<td>Banana, Cavendish</td>
<td>1 ea</td>
<td>42.00</td>
<td>-3.90</td>
</tr>
<tr>
<td>Modern Bread ? Must Classic 400 Gm.</td>
<td>1 ea</td>
<td>27.00</td>
<td>27.00</td>
</tr>
<tr>
<td>Modern Bread ? Must Classic 400 Gm.</td>
<td>1 ea</td>
<td>27.00</td>
<td>27.00</td>
</tr>
<tr>
<td>Elit. Egg Cake Premix 30 Gm.</td>
<td>2 ea</td>
<td>20.00</td>
<td>40.00</td>
</tr>
<tr>
<td>Elit. Egg Cake Premix 30 Gm.</td>
<td>2 ea</td>
<td>20.00</td>
<td>-8.00</td>
</tr>
<tr>
<td>Cocino</td>
<td>1 ea</td>
<td>38.00</td>
<td>38.00</td>
</tr>
<tr>
<td>Cocino</td>
<td>1 ea</td>
<td>38.00</td>
<td>-2.40</td>
</tr>
<tr>
<td>Tomato, Standard</td>
<td>1 ea</td>
<td>38.00</td>
<td>38.00</td>
</tr>
<tr>
<td>Tomato, Standard</td>
<td>1 ea</td>
<td>38.00</td>
<td>-5.70</td>
</tr>
<tr>
<td>Amul Peanut Roast 175 Gm.</td>
<td>1 ea</td>
<td>38.00</td>
<td>38.00</td>
</tr>
<tr>
<td>Amul Peanut Roast 175 Gm.</td>
<td>1 ea</td>
<td>38.00</td>
<td>38.00</td>
</tr>
<tr>
<td>Delight, Refill, 400 Ml</td>
<td>1 ea</td>
<td>37.00</td>
<td>37.00</td>
</tr>
<tr>
<td>Delight, Refill, 400 Ml</td>
<td>1 ea</td>
<td>37.00</td>
<td>37.00</td>
</tr>
<tr>
<td>Lifebuoy Color Changing Handwash 185 Gm.</td>
<td>1 ea</td>
<td>56.00</td>
<td>56.00</td>
</tr>
<tr>
<td>Lifebuoy Color Changing Handwash 185 Gm.</td>
<td>1 ea</td>
<td>56.00</td>
<td>-5.70</td>
</tr>
<tr>
<td>Orange, Krim Mala Kg</td>
<td>1 ea</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Orange, Krim Mala Kg</td>
<td>1 ea</td>
<td>100.00</td>
<td>-8.00</td>
</tr>
<tr>
<td>Mar. Selecto Natta Vadi Pice 10 Kg</td>
<td>1 ea</td>
<td>588.00</td>
<td>588.00</td>
</tr>
<tr>
<td>Mar. Selecto Natta Vadi Pice 10 Kg</td>
<td>1 ea</td>
<td>588.00</td>
<td>-318.00</td>
</tr>
<tr>
<td>Carry Bag Medium 40 Micron</td>
<td>3 ea</td>
<td>6.00</td>
<td>18.00</td>
</tr>
<tr>
<td>Carry Bag Medium 40 Micron</td>
<td>3 ea</td>
<td>6.00</td>
<td>-6.00</td>
</tr>
</tbody>
</table>

**Slab Offer Expense**

**Total Quantity:**32

<table>
<thead>
<tr>
<th>Total</th>
<th>1,823.67</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net Total</td>
<td>1,798.58</td>
</tr>
<tr>
<td>VAT 4.4%</td>
<td>80.82</td>
</tr>
<tr>
<td>VAT 4.4%</td>
<td>80.82</td>
</tr>
</tbody>
</table>

**PAYMENT SUMMARY**

<table>
<thead>
<tr>
<th>Axis Bank DSC</th>
<th>1,823.67</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axis DSC</td>
<td>2574</td>
</tr>
</tbody>
</table>

*YOU HAVE SAVED*

**CLARION Scheme Detail:**

| 476.58 |
Figure 133. Upload and analysis of grocery bill

Figure 134. Identified ingredients

Figure 135. Managing shopping basket
Figure 136. Updating of shopping basket

Figure 137. Profile page

Figure 138. Profile details displayed
Figure 139. Recipe suggestions

Appendix H

H.1. User Evaluation

The consent page can be viewed at http://li719-186.members.linode.com/consent.

Figure 140. Consent page
**IntelliDelight - User Evaluation**

1. Do you usually use an application for recipe suggestions?
   - [ ] Yes
   - [x] No
   If no, please specify the reason: 

2. Would you like any other health profiles to be included in the application?
   - [ ] Yes
   - [x] No
   If yes, please specify: 

3. Totally, how many times did you use the upload bill feature?
   - [ ] Never
   - [ ] 1-3 times
   - [ ] 4-6 times
   - [ ] More than 10 times

4. How would you rate the accuracy of recognized ingredients from the bills?
   - [ ] 100%
   - [ ] 90-99%
   - [ ] 80-89%
   - [ ] 70-79%
   - [ ] 60-69%
   - [ ] 50-59%
   - [ ] 40-49%
   - [ ] 30-39%
   - [ ] < 30%
   - [ ] Not applicable

5. How would you rate the design of the application in relation to layout, colour and graphics?
   - [ ] Excellent
   - [ ] Good
   - [ ] Satisfactory
   - [ ] Unsatisfactory

6. Do you find the application easy to use?
   - [ ] Yes
   - [x] No

7. How would you rate the recipe suggestions?
   - [ ] Excellent
   - [ ] Good
   - [ ] Satisfactory
   - [ ] Unsatisfactory

8. Do you find the idea of this application to be innovative and interesting?
   - [ ] Yes
   - [x] No

9. Would you be keen in using this application for your daily/weekly recipe suggestions?
   - [ ] Yes
   - [x] No

10. Do you have any other comments, questions, or concerns?

---

**Figure 141. Questionnaire**
### Table 114. Individual responses of test users

<table>
<thead>
<tr>
<th>Test Users</th>
<th>Q1: Do you usually use an application for recipe suggestions?</th>
<th>Q2: Would you like any other health profiles to be included in the application?</th>
<th>Q3: Totally, how many times did you use the upload bill feature?</th>
<th>Q4: How you rate the accuracy of recognised ingredients from the bills?</th>
<th>Q5: How do you rate the design of the application in relation to layout, colour and graphics?</th>
<th>Q6: Do you find the application easy to use?</th>
<th>Q7: How do you rate the recipe suggestions?</th>
<th>Q8: Do you find the idea of this application to be innovative and interesting?</th>
<th>Q9: Would you be keen in using this application for your daily/weekly recipe suggestions?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>No</td>
<td>1-3 times</td>
<td>80 – 89%</td>
<td>Good</td>
<td>Yes</td>
<td>Good</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>No. Have only just become familiar to it.</td>
<td>No</td>
<td>1-3 times</td>
<td>80 – 89%</td>
<td>Good</td>
<td>Yes</td>
<td>Satisfactory</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>No. I was not aware of the existence of such applications</td>
<td>No</td>
<td>1-3 times</td>
<td>50 – 59%</td>
<td>Good</td>
<td>Yes</td>
<td>Good</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>No. I haven't started experimenting with lives yet.</td>
<td>No</td>
<td>1-3 times</td>
<td>80 – 89%</td>
<td>Satisfactory</td>
<td>Yes</td>
<td>Satisfactory</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Yes</td>
<td>No</td>
<td>1-3 times</td>
<td>90 – 99%</td>
<td>Excellent</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>Yes. Irritable bowel syndrome</td>
<td>Yes. Irritable bowel syndrome</td>
<td>1-3 times</td>
<td>90 – 99%</td>
<td>Satisfactory</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No.</td>
<td>Comment</td>
<td>Frequency</td>
<td>Rating</td>
<td>Good</td>
<td>Excellent</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
<td>--------------</td>
<td>------</td>
<td>-----------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>No. I'm a foodie, just love to eat!!</td>
<td>4 - 6 times</td>
<td>&lt; 50%</td>
<td>Yes</td>
<td>Good</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Yes</td>
<td>1 - 3 times</td>
<td>80 – 89%</td>
<td>Yes</td>
<td>Satisfactory</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>No. I don't cook.</td>
<td>Yes. Weight gain, sedentary, active.</td>
<td>1 - 3 times</td>
<td>&lt; 50%</td>
<td>Satisfactory</td>
<td>Yes</td>
<td>Good</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>11</td>
<td>Yes</td>
<td>Yes. Nutritious based</td>
<td>1 - 3 times</td>
<td>70 – 79%</td>
<td>Excellent</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>12</td>
<td>No. I'm not much of a cook --- also have Crohn's disease.</td>
<td>Yes. Digestive Diseases and Disorders - gluten-free - peanut free - vegan - vegetarian.</td>
<td>1 - 3 times</td>
<td>70 – 79%</td>
<td>Satisfactory</td>
<td>Yes</td>
<td>Satisfactory</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>13</td>
<td>Yes</td>
<td>Yes. Digestive diseases.</td>
<td>1 - 3 times</td>
<td>80 – 89%</td>
<td>Satisfactory</td>
<td>No</td>
<td>Good</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Test Users**

Q10: Do you have any other comments, questions, or concerns?

1. -
2. Nothing at this stage.
3. Nil
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
</tr>
</tbody>
</table>
| 8 | 1) Would like to get some info after file upload is done, especially if the app fails to pick items  
2) Rather than the current set up to 'copy to ingredients', displaying the items in a multi-select list would be more user friendly (multi-select or group them in divs that can be selected, should be able to find some plugins for that)  
3) The navigation links - please highlight them when on the corresponding page  
4) In recipe section I would have liked to search for recipes myself as well, not just the ones based on my profile (strictly my opinion, that's all)  
5) The success rate is quite low, but I can appreciate the complexity of the idea, so its fine for now (that's why I asked for a recipe search - since most of my items were not scanned, I got no suggestions)  
6) Design, colour scheme & font are ok as well  
7) Issue with tooltips - would be better if the previous tooltip disappears when I click on a new one (profiles section) |
| 9 | Very good.                                                                             |
| 10| Good idea. Gets reliable results. The Interaction flow of the application can be improved. It would be better if the flow followed a 'conversation' approach. |
| 11| -                                                                                     |
| 12| Besides the graphics and marketing, I think this is a great idea. I've thought there should be a site like this too! I'm not sure if it is included, but there should also be a way to create and print out a shopping list. Is there a way to save or manage your existing recipes? I also love the line "Nothing prevents you from healthy cooking!" So many people make excuses for unhealthy eating habits and this is a great tool for people not to make excuses anymore! |
| 13| Could be a little more user friendly and attractive.                                    |