Easy Cooking Application

Pauline Michaud
H00201141

August 2015
Supervisor: Mr Peter King

Computer Science
School of Mathematical and Computer Sciences

Dissertation submitted as part of the requirements for the award of the degree of MSc in Advanced Internet Applications
Statement of non-plagiarism

« I, Pauline Michaud, 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: Pauline Michaud…………………

Date: 18 August 2015 »
Abstract

This document outlines my MSc Dissertation in Advanced Internet Applications. The objective of this Dissertation is to implement an easy cooking application that includes not only recipes but also tutorials, games and quizzes in order to orient more the application into a learning application about cooking. This report starts with an Introduction that introduces the origin of this project and the target users, and sets out the outline of the project in non-specialist language. It will also highlight all the objectives of our application more precisely and will show the related work with a benchmarking analysis. After that, a Literature Review is provided, which explains the reasons behind the choice of implementing this kind of application to learning cooking rather than just a lot of different recipes grouped in one simple application. It will begin with a market analysis and continue with an explanation of the tools and methods that we’ll use in the realization of our application (client-side and server-side technologies). Then, third section about the Requirements, which will set out all the features that we will implement in our application. An Architecture and Design section of our entire mobile application will be provided later, explaining what and where to put the elements matching with a specific feature used in this project. Then, an Implementation and Results section will detail more technically and deeper into the code how to create and establish those elements correctly in our mobile application. An Evaluation section will highlight the results and success of this project. Followed finally by a Project Management section, which provides a discussion about the schedule of our project, a Professional, legal, ethical and social issues section and a Conclusions and Future Work section.
Acknowledgements

I would like to thank:

- Mr Peter King for his generous and continuous support about this project. It’s a pleasure to work with someone who loves his profession and has always relevant advice for his students.

- The « Research Methods » lecturers for all the information and advice about the research report provided in their lectures. It has been a very good help while writing the research report in the second semester.

- Mr Arnaud Bertrand for his professionalism within a team, his motivation, his work and his advice in this project that we are both dedicated to.
# Table of contents

1 INTRODUCTION ......................................................................................................................... 8
   1.1 INTRODUCTION .............................................................................................................. 8
   1.1.1 Origin of the project ................................................................................................ 8
   1.1.2 Project targets ........................................................................................................... 8
   1.1.3 Problematic ............................................................................................................... 8
   1.1.4 Approach .................................................................................................................. 9
   1.2 OBJECTIVES .................................................................................................................. 9
   1.3 PROBLEM STATEMENT ............................................................................................... 10
   1.4 RELATED WORK ........................................................................................................... 11

2 LITERATURE REVIEW ............................................................................................................. 13
   2.1 MARKET ANALYSIS ...................................................................................................... 13
   2.2 GENERAL ARCHITECTURE ......................................................................................... 19
      2.2.1 Web services ........................................................................................................... 19
      2.2.2 REST API ............................................................................................................... 20
   2.3 WEB APPLICATION ....................................................................................................... 21
      2.3.1 Design patterns ........................................................................................................ 21
      2.3.2 JavaScript ............................................................................................................... 21
      2.3.3 Framework AngularJS ............................................................................................ 21
      2.3.4 Angular concept design .......................................................................................... 22
      2.3.5 Web application design .......................................................................................... 22
   2.4 MOBILE APPLICATION ................................................................................................. 23
      2.4.1 Platform Android ...................................................................................................... 23
      2.4.2 Ionic ........................................................................................................................ 23
      2.4.3 Testing ..................................................................................................................... 26
      2.4.4 Summary ................................................................................................................ 26

3 REQUIREMENTS ......................................................................................................................... 27
   3.1 OBJECTIVES .................................................................................................................. 27
   3.2 ACHIEVED OBJECTIVES .............................................................................................. 29

4 ARCHITECTURE AND DESIGN ................................................................................................. 30
   4.1 ARCHITECTURE OVERVIEW ....................................................................................... 30
   4.2 DESIGN .......................................................................................................................... 32
      4.2.1 User interface design ............................................................................................... 32
      4.2.2 Database and server design .................................................................................... 45

5 IMPLEMENTATION .................................................................................................................... 45
   5.1 CLIENT SIDE .................................................................................................................. 45
      5.1.1 Implementation overview ...................................................................................... 45
      5.1.2 General part ............................................................................................................ 47
      5.1.3 Connection part ..................................................................................................... 48
      5.1.4 User part ................................................................................................................. 50
      5.1.5 Recipe part .............................................................................................................. 52
      5.1.6 Game part ............................................................................................................... 55
   5.2 SERVER SIDE .................................................................................................................. 57
   5.3 TESTING ......................................................................................................................... 58
## List of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Overall overview of the project</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Comparison of cooking applications</td>
</tr>
<tr>
<td>Figure 3</td>
<td>BigOven application</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Food Network application</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Cookpad application</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Personas 1 - Nicole</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Personas 2 - Peter</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Personas 3 - Stuart</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Personas 4 - Louiseelle</td>
</tr>
<tr>
<td>Figure 10</td>
<td>MyBalsamiq software</td>
</tr>
<tr>
<td>Figure 11</td>
<td>SurveyMonkey software</td>
</tr>
<tr>
<td>Figure 12</td>
<td>General architecture of the project</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Actions of RESTful API (Source: Bojinov, 2015)</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Angular concept design (Source: Angularjs, 2015)</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Expansion of an application (Source: Foster, 2015)</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Mobile application</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Different nature of applications (Source: Bavari, 2014)</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Architecture of mobile application design using Ionic (Source: Scotter, 2015)</td>
</tr>
<tr>
<td>Figure 19</td>
<td>All used technologies for the mobile application</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Architecture overview</td>
</tr>
<tr>
<td>Figure 21</td>
<td>Link to server by a package design</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Class design</td>
</tr>
<tr>
<td>Figure 23</td>
<td>Git lines of code</td>
</tr>
<tr>
<td>Figure 24</td>
<td>Ionic launch in command line</td>
</tr>
<tr>
<td>Figure 25</td>
<td>Checks errors example</td>
</tr>
<tr>
<td>Figure 26</td>
<td>Menu implementation</td>
</tr>
<tr>
<td>Figure 27</td>
<td>Picture-upload implementation</td>
</tr>
<tr>
<td>Figure 28</td>
<td>Login implementation</td>
</tr>
<tr>
<td>Figure 29</td>
<td>Sign-up implementation</td>
</tr>
<tr>
<td>Figure 30</td>
<td>Profile implementation</td>
</tr>
<tr>
<td>Figure 31</td>
<td>Profile-edit implementation</td>
</tr>
<tr>
<td>Figure 32</td>
<td>Create recipe implementation</td>
</tr>
<tr>
<td>Figure 33</td>
<td>Display recipe implementation</td>
</tr>
<tr>
<td>Figure 34</td>
<td>Search recipe implementation</td>
</tr>
<tr>
<td>Figure 35</td>
<td>Create quiz implementation</td>
</tr>
<tr>
<td>Figure 36</td>
<td>Display quiz implementation</td>
</tr>
<tr>
<td>Figure 37</td>
<td>Search quiz implementation</td>
</tr>
<tr>
<td>Figure 38</td>
<td>Unit tests</td>
</tr>
<tr>
<td>Figure 39</td>
<td>Unit tests example code</td>
</tr>
<tr>
<td>Figure 40</td>
<td>Karma connection</td>
</tr>
<tr>
<td>Figure 41</td>
<td>Unit tests results in command line</td>
</tr>
<tr>
<td>Figure 42</td>
<td>User testing evaluation – results 1</td>
</tr>
<tr>
<td>Figure 43</td>
<td>User testing evaluation – results 2</td>
</tr>
<tr>
<td>Figure 44</td>
<td>User testing evaluation – results 3</td>
</tr>
<tr>
<td>Figure 45</td>
<td>Master Schedule</td>
</tr>
<tr>
<td>Figure 46</td>
<td>Gantt Chart</td>
</tr>
</tbody>
</table>
1 INTRODUCTION

1.1 Introduction

1.1.1 Origin of the project

To begin, Arnaud and I love cooking. But we realized that we made often the same recipes. So, for people who love cooking, when they download an app, it’s in order to give them new ideas and try new things in cooking, either in terms of taste by trying foreign dishes or in terms of presentation by replicating dishes from famous chefs. But when they threw themselves into these new recipes, it sometimes appears unclear what the user who posted the recipe wanted them to do. After some researches, we found a recurrent problem: some steps when they are making a recipe were not clear in their mind. There were some highly domain specific words in recipes from chefs that they did not understand and which they had to go search in order to then be able to make the recipe. And we think that sometimes they also wished they had some extra tips and additional guidance such as those of a grandmother, like how to choose the right vegetable or fruit concerning its maturity. Maybe sometimes they also wondered how many calories it could be in this cake that they were trying to make, etc.

Arnaud and I met on these points, we thought that these points were what it was missing in many cooking apps. A fun side as well as a side to learn to cook a recipe step by step, a bit like a tutorial that explains all the significant details of each step in the realization of the recipe. So we said to ourselves to keep the base of a cooking application, i.e. offer many different recipes of the world, but adding a "learning" side with a lot of tips in it. We also decided on adding quizzes to learn the domain specific terms and the history of cooking in the world.

So that’s where we’ve come to the idea of our project: To create the cooking application that corresponds to us. Therefore, Arnaud Bertrand will be our co-worker on this Dissertation; so we’ll be two during the third semester in Heriot-Watt University to work on this MSc Dissertation. Thus, the subject of this project is to implement an easy cooking application that includes recipes, tutorials, games and quizzes in order to orient the application into a learning application about cooking.

1.1.2 Project targets

The target of our application is any person who wants to learn about cooking and have fun with a cooking application, someone who doesn’t want to just find some online recipes.

Regarding the target user of this application, the panel is varied. These range from teens to adults to the elderly. As long as these people have a passion for cooking and want to learn more than just make a recipe downloaded from the Internet, we think they will be interested in our cooking application that propose a real and funny “learning” side.

The users can come from beginners level, who want to learn how to cook a specific recipe through fun and attractive tutorials, to famous chefs who want to share their tips and recipes with other people passionate by cooking. Different types of target user can be found with the help of the Personas method detailed in the Literature review section later.

1.1.3 Problematic

Therefore, the main problem of our project for this summer is to create a cooking application that can be displayed on the web and on a mobile phone. Moreover, we want to turn our application into a learning application about how to cook the recipes that it can provide to us with tutorials. We also want to add some games in order to have a fun application and some quizzes to learn also the history of cooking.
1.1.4 Approach

To do this, we will have a specific approach. We will firstly detail all the objectives of our project in the Objectives section below. After that, we will research all related work about already existing cooking applications with a benchmarking. Then, the Literature Review section will be provided, explaining the methods and tools that we will use in our project. It will begin with a market analysis and going on more specifically in the mobile application with a client-side and a server-side technology part. Then, we will bring out all the specifications, features and requirements that our target users would like to have in our cooking application in the Requirements section. Followed by the Architecture and Design section, which provides and largely explains each aspects and parts of the design of the mobile application. Once the design has been made, we can go in depth of the technical part of this project with the Implementation and Results section. In this part, we will explain how we create and establish all the different features of our mobile application in the code. Moreover, we will evaluate our cooking application by making a user testing at the end and analyse those results in the Evaluation section. Furthermore, we realised a Gantt chart at the beginning of our project. Now at the end, we can see, analyse and be able to discuss if the project was developed as planned or not. We will also explain the management methods that we used throughout this project. These two discussions will be provided in the Project Management section. There will be also a Professional, legal, ethical and social issues section to discuss about, because we will have some laws to respect and accreditations to have if we want to deploy correctly our cooking application in the world. We can conclude this project with a conclusion, a critical evaluation of the entire project and list out some future work for later.

1.2 Objectives

The objectives are clear and simple, and our supervisor, Mr Peter King, gave us the freedom of choice on any type of technology that we would like to use for the realization of our application. We will see later, in the Literature review section, the explanation of why we used this particular technology for our application instead of any other.

The main objective is to create a cooking application, but we would focus our application more on learning how to cook and not just have online recipes. So after a market analysis we will see what will be the specific functionalities that we want to implement in our application in order to differentiate ourselves from cooking applications that already exist in the market for mobile applications.

Moreover, the main objective can be then detailed into sub-goals, as shown in the following figure:

*Figure 1: Overall overview of the project*
Finally we note that our whole project is divided into 4 parts:
- A mobile part that simply represents the mobile application
- A server part containing the services offered in our application
- A PC part representing the web application
- A tablet part that represents the same application as the mobile application but for touchpad, and we will implement it if the project is progressing well.

To maintain speed and not interfere in either’s work of the project and application, we decided to separate the project as follows: our co-worker will handle the server part, which will be encoded in Node.js, while we will take care of the mobile application, which will be encoded in Ionic (also using Angular and Cordova). It will be more difficult than the web application because we will have to make the connection between the application and the camera of the phone. So, he will handle all the technical and functional part of the web application on his own. Moreover, while designing the mobile application, the Responsive web design of the web application would be thought to be complementary of the mobile application. The explanations of "why use such technologies?" and "what are they?" will be detailed in the Literature review section just below.

Later in this report, the explanation of the approach of our project begins with an analysis of its future users (Personas) and a market analysis (Benchmarking). We will then add our ideas in order to differentiate ourselves from cooking applications that already exist, by orienting our application on the side "learn to cook". We will do some research about the different possible design in order to best integrate these features. So, once the functional specifications clearly established, we will make actually several mock-ups. Thereafter we will submit it to our target users by doing a field survey to collect their point of view about these different designs. Then we will exploit at best and analyse the data that emerge from the surveys in order to choose the best design, both interactive and fun, but also learning. And finally we will really implement our cooking application.

Therefore, in this project, I propose to carry out and explain in depth the reasons: why we chose some technologies instead of others in the Literature review later, and the whole technical and functional parts of the mobile application that we will implement during this third semester. As for our co-worker, who owns the part about the back-end of the application (represented by the server and the external database) and the web application, all the technologies that he is going to use to build it will be explained and detailed in his report.

1.3 Problem statement

Nowadays, we can see that everybody becomes connected whether through mobile devices or Internet. But it’s the mobiles era which develops increasingly recently with the appearance of touchpad and connected watch etc. Obviously the goal of this project is to create a cooking application, that will be accessible from mobile devices first and also from a website.

Our cooking application will be more concentrated on the user needs than on the recipe content. In spite of lot of cooking application already been present in the market, none of them focus on teaching how to cook. The idea here is to focus our application more on learning how to cook and not just have online recipes.

People who love cooking can sometimes be afraid by the complexity and difficulty of making some recipes. It can happen that they don’t know some ingredient used in the recipe and they just skip it because they don’t feel confident to do it. So they just give up because the vocabulary is too complicated for them. Or sometimes they just skip it because the recipe is not well written or not enough clearly explained step by step.

The current cooking applications have a long way to go in terms of all users and not just the experts. We will make the content of our entire cooking application available to everybody, from beginners who want to learn how to cook a specific recipe through fun and attractive tutorials, to famous chefs who want to share their tips and recipes with other people passionate by cooking.
1.4 Related work

In this part, we will do a Benchmarking, in order to see what there is already on the market about existing cooking applications. First of all, we have preselected the three more used, free and downloaded cooking applications on the Android store to discuss about and see what there is in common: Big Oven, Food Network and Cookpad. The first two cooking applications have more than 1,000,000 downloads and Cookpad has more than 10,000,000 downloads on the store. Here are the grades that they have collected from their users:

![Comparison of cooking applications](image)

*Figure 2: Comparison of cooking applications*

Here are the reflexions and notes that we discuss during our research on these three famous cooking applications:

- **BigOven**
  - Design: Main screen is composed of pictures of recipes. You need to slide up and down to have new random recipes. There is a small menu at the bottom of the screen to access recipe list, your grocery list and a planner. When you click on a recipe, the recipe, the ingredients and a picture are all on the same page.
  - Search tool is available to find recipes.
  - Possible to plan your meals for the week.
  - Possible to have a grocery list that you can update with the recipes. Possibility to check/unchecked items from the list.
  - Need to sign up to set your planner and grocery list.
  - Advertisement appears at the bottom of the page sometimes.
  - **Review:** Application that does its job but nothing more. Nothing to stand out of the crowd and appears like a simple cookbook.

*Figure 3: BigOven application*
Food network

- Design: Main screen composed of a horizontal sliding menu at the top and 4/6 pictures by screens that you can slide vertically. Another menu is available to get some useful tools such as a converter and a timer. When you click on a chef and then on a recipe, the recipe, the ingredients and a picture are all on the same page. Possible to switch between a recipe view and an overview to have only relevant information for cooking.
- Some well known cook display their recipe.
- Search tool is available to find recipes.
- Possible to plan your meals for the week.
- Possible to have a grocery list that you can update with the recipes. Possibility to check/uncheck items from the list.
- Need to sign up to set planner and grocery list.

Review: The public targeted are more advanced cooks than what we target. The application is well design and is nice to use.

Figure 4: Food Network application

Cookpad

- Design: Main page is composed of pictures with a very simple menu. A very complete menu is available by clicking on the left side. You can access categories as well as cooking tools easily on the right. When you click on a recipe, the ingredients, the recipe and pictures are on the same page. However the recipes are clearly separated in steps.
- Search tool is available to find recipes.
- Possible to plan your meals for the week.
- Possible to have a grocery list that you can update with the recipes. Possibility to check/uncheck items from the list.
- Need to sign up to set planner and grocery list.
- Recipes steps are available on smart watch.
- Possibility to add your picture to the recipe so other can see what you made.

Review: This is the most outstanding application. It clearly deals with everything a normal cook would like. The app is very complete and the fact that it handles smart watch is really nice. Really pleasant application to use.

Figure 5: Cookpad application
We can see that these three applications have the same goal: they are used to find and display recipes and they do it well considering the grades that they have from their users (4.3 and 4.4 with almost full stars). This implies that a person interested in learning how to cook will just find recipes but will not find an application to help him improve its skills directly. After research on the store we didn’t find any application that has for goal to improve their user’s skills. That is why we wanted to make it our specialty. A lot of people would like to know how to cook well but give up when they see all those recipes and difficult terms. Thus, we have to make them more confident and help them more.

We can also see that, for these three cooking applications, there is every time a main picture related to the recipe. This is due to the fact that people are here to look for food so we have to show them the results the most often and clearly possible. Moreover, we found out that recipes are more seen and used by users when they have pictures to show the final result.

We can also remark that from this benchmarking, most of the cooking applications have common tools like a search engine, a shopping list, a meal planner, a timer, a unit converter and a list by categories. That is why we will consider these tools; see if they can be used in our application and how we can implement them in a different and more attractive way.

There is also another common thing between all these apps. We need to sign up if we want to access to all the resources of the app, like those tools above for example. But it can be a reluctant factor for the user. That is why, instead of doing like them, it might be better to ask in a friendly way the information of the user. Which means by asking the name and email of the user for example, just at the first launch of the app.

So now we have an idea of what are the main downloaded cooking applications, we can add our own functionalities to our application and make a real difference with the other cooking applications.

2 LITERATURE REVIEW

In this Literature review section, we will explain the process that we will use to build our application and show the researches that we’ve done about each part of our entire project. Our main goal is to create a community of people who want to share the love of cooking between them, throughout an easy and funny cooking application. In order to do that, here is the process and choice technologies that we made:

We will begin with a market analysis that will highlight the purpose of this project, followed by the creation of mock-ups in order to do a survey and analyse the data collected from this survey. At the end of this point, we will be able to know how we were going to make our application different from what already exists and how we are going to satisfy users’ needs.

Thereafter, we will explain the general architecture of our application, with the use of web services. Moreover, we will detail in a first time the web application and why we chose to use the Angular framework in this project and other particular tools. And finally, we will provide the part about the process to build the mobile application, with the use of the special language Ionic.

2.1 Market analysis

We will highlight the purpose of this project with the help of the market analysis. Thus, we will be able to decide how we are going to make our cooking application different from what already exists and how we are going to satisfy users’ needs.

We have done a Benchmarking above, in the related work part of the Introduction section, which has
already given us some leads and advices in order to implement the right cooking application, as we wanted. But we can push our market analysis a little deeper by using the Personas method as explain below.

**Personas**

« To make a good digital product it is essential to put users at the heart of its design process. One way to incorporate the views of users to the project is to make personas.» (Chaine, 2013)

A persona is « a representation of a particular audience segment for a website/product/service you are designing » (Mears, 2013) so it’s a specific (hypothetical) person, with a name, an age, a location, a sex, likes/dislikes, motivations, etc. « They are a great way to visualise all the user and audience data you have managed to gather. » (Mears, 2013) and « they represent a major user group for your website and give a clear picture of the user's expectations and how they're likely to use the site » (Usability.gov)

They are really useful for making key decisions while designing our application. According to Jérémie Chaine (2013), here are the 7 reasons to do Personas:

- Personas can:
  1. Look truly the need of users and not to remain only in the speech
  2. Identify actual and specific user needs. This will allow you to make a simple and suitable product and not a tote features
  3. Understand, humanize your target and create empathy for users
  4. Have a simple communication tool, clear and explicit
  5. Having a common reference
  6. Streamline decisions
  7. Prioritize development efforts »

So in our case, we can have 4 types of personas: a woman in a huge family, a single person in his apartment, a teenager who likes cooking for fun and a grandmother in retirement.

<table>
<thead>
<tr>
<th>Nicole</th>
</tr>
</thead>
<tbody>
<tr>
<td>A woman in a huge family, 41 years old</td>
</tr>
</tbody>
</table>

**About Nicole**

- Doesn’t work since she had her first kid
- Lives in London with her husband and three kids
  - Tom (10 years old), Colin (6 years old) and Sarah (4 years old)
- Has a busy life at home
- Managed the whole life of the family in the house
- Is comfortable online and find a lot of recipes online
- Is a “hunter” cooker - wants to get in and out as quickly as possible
- When cooking usually just wants to find something that ticks all the boxes
- Wants to get good value - preferably the best time-spending possible
- Gets frustrated by not being able to see all details about a recipes
- Gets annoyed by slow and clunky websites

“I like cooking during the week, and especially the week-end when the whole family is reunited. “

**Key goals:**

- To get the best recipe possible
- Be inspired
- A quick and painless online step by step to learn how to do the recipe

*Figure 6: Personas 1 - Nicole*
Peter
A single person in his flat, 24 years old

About Peter
- Works in Edinburgh as a salesman in a big company
- Lives in Riccarton, single in his flat
- Has a busy life so is a big fan of time savers
- Go out a lot of time, spending times with his friends after work and during the week-end
- Is an adept of fast-food but not when his friends came to his apartment
- Likes sharing and learning new recipes
- Wants to do simpler recipes and menu for his friends when they came to eat in his flat

“I don’t have a lot of time to cook for myself or anyone else. “

Key goals:
- To get the simpler recipe possible
- A quick and painless online step by step to learn how to do the recipe
- Doesn’t want to loose too much time

Figure 7: Personas 2 - Peter

Stuart
A teenager who likes cooking for fun, 16 years old

About Stuart
- Student in London in 1st year of Mathematics in College
- Lives also in London with his parents
- Is the only child of the family
- Cooking is his passion - especially all possible desserts
- Doesn’t have a lot of time to cook during the week because of his studies
- Has to see with his mum to buy all the ingredients before being able to cook
- Wants to learn as many things as possible in cooking
- Likes share his new recipe with his friends

“I like cooking every time I can. “

Key goals:
- Be inspired
- Tries every time a new recipe
- Likes learning new tips in cooking

Figure 8: Personas 3 - Stuart
**Louise**

A grandmother in retirement, 83 years old

**About Louise**
- Retired, like his husband
- Lives in the centre of Edinburgh
- Has 3 children and 8 grand-sons/grand-daughters
- Likes cooking for every meals of the day
- Wants to share her tips with a community which also likes cooking
- Is not really comfortable with mobile, and Internet
- Likes cooking for her whole family when they come visiting her

“ I cook everything I eat, and especially when my grand-son came visiting me. “

**Key goals:**
- Share tips with a cooking community
- Find some new recipes never tried before
- Likes cooking for her and her family every times

**Figure 9: Personas 4 - Louise**

Whenever we have a question about our design or some features of our application, we will “ask” our personas that can tell us their preferences. So we will refer to them in the design of the interface every time we have a question or a problem in the process of making the design of our application.

Now we have defined our personas and realised a complete benchmarking, we have all the cards in hand to create the cooking application that corresponds to our expectations: with a basic part that all cooking applications possess (search and display a recipe, a lot of pictures, etc.) and with our special part that will make the difference from them (by including pictures, games to learn about cooking, tools in order that users don’t give up on cooking a recipe, etc.)

**Our ideas to add**

As we decided, and with the goal of being different from already existing cooking applications, we want to add some functionalities and our ideas for a learning application about cooking:
- Add the number of calories to the recipe (possible help the recipe creator by calculating the number of calories associated with the ingredients)
- Add the kitchen utensils used for make the recipe so people know whether they can do the recipe or not
- Cross out ingredients from the shopping list by sliding finger onto it
- Make a grandmother icon near the ingredients in order to have a tip on how to select the ingredient (how to know if the ingredient is mature for example)
- Add shopping list to shopping cart of grocery store such as Asda/Tesco (see with them if they have such a technology to connect to their online shop)
- Create cooking games / quizzes to improve culture of food
- Ask user information only at the beginning of the application and never after

We will obviously also keep the basic functionalities of a cooking application, as we saw in the market analysis before:
- Find a recipe in real-time by searching ingredients
- At the end of the recipe ask the person to take a picture of its masterpiece in order to share it to the community.
- A favourite recipe list and a recently visited recipe list
- …
We will provide details more precisely of all the requirements that we have decided to put in our cooking application in the Requirements section below with the different level of priority.

But adding these specific functionalities in our application brought us some problems and questions, especially about the design. We don’t want to overwhelm the user with too many things in the screen, or also we don’t know if it’s better that there is a scroll bar or just any move for the size of the screen. And moreover, it will maybe not possible to implement all these ideas during the time which was accorded to us for this project - three months.

So we have to analyse and do some research about any special design that already exist and that can tell us what the user prefer or want in his applications regarding the design and of such functionalities.

Mock-ups & Survey

Now that we have all our functionalities in mind for our application, we have to think about how to put them with fun, simplicity and interactivity in our cooking application. So at first, we will create some mock-ups about the different design that we can have for our application – Appendices A. With the help of the lecture in Advanced Interaction Design by Mr Lemon and Mrs Foster, we will use for our first set of mock-ups the online software MyBalsamiq:

Every time that we have a problem or a question about the design and the interactivity of our application, we will refer to our Personas, and ask them what is it the best to do.

Once we will agree for the design of our cooking application, we will submit it to an evaluation to our target users. And to do that, we will use the online software SurveyMonkey. We will ask them what they were happy to have and see in our application or if they want other special things and more interactivity for example, and put also some questions about their preferences if we proposed two different designs for a special functionality in our cooking application for example – Appendices B.
Then, we will analyse below the data collected from this survey in order to take the best way of using our application with fun and interactivity and also the best design in order that everyone understand naturally what to do to access to such functionality in our cooking application.

But, according to the conference about organizational surveys (Fulton and Brad, 2014), « surveys provide a critical source of data for scholars, yet declining response rates are threatening the quality of the data being collected. » So, we have to take care about the collected data of the survey. We must take into account the data from this survey but with a reasonable rate and also take into account our own decisions about the design of our application in performing eventually more additional researches.« Despite deteriorating response rates, technological advances have produced several cost-effective strategies researchers can use to increase response rates and improve data quality. Furthermore, scholars have developed innovative methods to assess nonresponse bias within survey data. » (Fulton and Brad, 2014)

**Evaluation of the survey**

According to the Appendices C, representing the results of the survey, the total number of people who answered our survey is 15. We can see that the majority of them are male people (10 against 5 for female) and are aged between 15 and 25 years old. Regarding their activities nowadays, there is approximately the same number of students (7 persons) than employed people (6 persons), and 2 unemployed people.

As we can see for the log in page, the main idea that emerges of this survey is that candidates prefer to have user authentication to fill in before be able to enter into the application. According to them, “not everybody can enter into the application without an account; so, it shows the security of the application.” And obviously, the name of application needs to correctly appear on this page.

For the header page, the majority of candidates want it to be separated from the rest of the page and should contain a menu, a search recipe and log out button principally. Moreover, they prefer to see the picture of recipes than a list of recipe’s title when they search for one. The idea of a random recipes selection even if there is no word enters in the input search is likely appreciate.

Then, for the create recipe page, it seems that the title of the recipe, the list of ingredients and a final picture of the recipe should be asked to the user for priority. The difficulty, total time and number of persons come just after. On the other hand, the type of course is not necessary for them. We can also see that they prefer to have the step’s description explained one by one and not all in the first page. Concerning the list of utensils, the majority is neutral or they don’t really know for now.

Finally, some important information should be put in the beginning of the display recipe page: the recipe’s title, the list of ingredients, the number of like for this recipe and the total time. The other ones are less necessary. It seems that they like the idea of a start recipe button, whose steps appear one by one with each a description, a timer and a picture. They don’t really want to see the entire description of each step in the first page. According to them, “there will be too much information in the first page.” They also agreed on the fact that they can comment the recipe and see the past comments. Concerning the display of the recipe’ pictures from other users once they cooked this recipe, the majority is neutral or they don’t really know for now.

Generally for each page, we can see that the design n°2 has been chosen more often. So we decided to implement this one according to the user’s expectations.

At this point, we can say that the market analysis gives us a first point of view of what already exists. Then we were able to add our ideas in order to be different from them. And finally, the survey was really useful, because with the help of the user’s feedbacks, we saw what they really want and need in our application.
Now, in the next sections just below, we will focus more on the technologies that we will choose to use in order to build our cooking application regarding the user’s needs: the general architecture in a first time, the web application for computer in a second time and the mobile application for mobile phones in a third time.

2.2 General architecture

This section will introduce the topics concerning client-server technology. First of all, here is the general architecture of our application, and therefore the link between the server and the client:

A client-server architecture is an « architecture of a computer network in which many clients (remote processors) request and receive service from a centralized server (host computer). Client computers provide an interface to allow a computer user to request services of the server and to display the results the server returns. Servers wait for requests to arrive from clients and then respond to them. Ideally, a server provides a standardized transparent interface to clients so that clients need not be aware of the specifics of the system (i.e., the hardware and software) that is providing the service. » (Encyclopaedia Britannica, 2015)

But we wanted to create a community inside our cooking application, which can share content and retrieve their information from other devices, that is why we decided to store the data on one server for the both devices (mobile and PC). Therefore, we will use an external database. With this solution, the web application and the mobile application will be able to access the database with same functionalities.

This first point of view about the whole process of the application was useful and gives us the first idea to build our application, which is an external database in order to share the data between the mobile and the web application.

Therefore, in order to understand clearly how this works, we will discuss, firstly about the choices that we made for the web services in our project in the next section.

2.2.1 Web services

A Web Service « refers to the technologies that allow for making connections [over the Internet]. Services are what you connect together using Web Services. » (Barry and Dick, 2013) Two main implementation of web services are nowadays: REST (REpresentational State Transfer) and SOAP (Simple Object Access Protocol) a
message protocol using HTTP (HyperText Transfer Protocol) and XML (Extensible Markup Language) to communicate.

Regarding our application we are going to choose REST rather than SOAP for the implementation of our web services. The reasons of this choice are:

1- Performance: REST implementation is faster than SOAP (3 to 4 times more when it’s about messages communication)

2- Simplicity: REST is using HTTP methods and JSON (JavaScript Object Notation) format which will require less effort to transform data if we are using JavaScript to implement our application, whereas SOAP is using XML format

Barry and Dick (2013) also approve the previous points by mentioning that « SOAP messages are hard-coded or generated without the use of a repository » and that « REST appeals to developers because it has a simpler style that makes it easier to use than SOAP. It is also less verbose so that less volume is sent when communicating. »

2.2.2 REST API

The REST API will be implemented by our co-worker, but we wont skip its explanation in this report because it is part of the good understanding and organization of this report. On the other hand, we can find more information on it in his report.

REpresentational State Transfer Protocol architecture runs over HTTP. One of the authors of HTTP, Roy Fielding, « defined a set of principles built around the HTTP and URI standards. This gave birth to REST as we know it today. » (Bojinov, 2015)

These different principles, which will make our HTTP application a RESTful application, are:

« 1- Everything is a resource: each piece of data available on the Internet has a format that could be described by a content type,

2- Each resource is identifiable by a unique identifier (URI): since the Internet contains so many different resources, they all should be accessible via URIs and should be identified uniquely, example: https://mysterious-eyrie-9135.herokuapp.com/recipes

3- Use the standard http methods: the native HTTP protocol defines 8 actions, also known as verbs (GET, POST, PUT, DELETE, HEAD, OPTIONS, TRACE, CONNECT), but the first 4 of them feel just natural in the context of resources, especially when defining actions for resource data manipulation.

Let’s make a parallel with relative SQL databases where the native language for data manipulation is CRUD (Create, Read, Update and Delete) originating from the different types of SQL statements INSERT, SELECT, UPDATE and DELETE respectively.

In the same manner, if you apply the REST principles correctly, the http verbs should be used as shown here:

<table>
<thead>
<tr>
<th>HTTP verb</th>
<th>Action</th>
<th>Response status code</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>Request an existing resource</td>
<td>&quot;200 OK&quot; if the resource exists, &quot;404 Not Found&quot; if it does not exist, and &quot;500 Internal Server Error&quot; for other errors</td>
</tr>
<tr>
<td>PUT</td>
<td>Create or update a resource</td>
<td>&quot;201 CREATED&quot; if a new resource is created, &quot;200 OK&quot; if updated, and &quot;500 Internal Server Error&quot; for other errors</td>
</tr>
<tr>
<td>POST</td>
<td>Update an existing resource</td>
<td>&quot;200 OK&quot; if the resource has been updated successfully, &quot;404 Not Found&quot; if the resource to be updated does not exist, and &quot;500 Internal Server Error&quot; for other errors</td>
</tr>
<tr>
<td>DELETE</td>
<td>Delete a resource</td>
<td>&quot;200 OK&quot; if the resource has been deleted successfully, &quot;404 Not Found&quot; if the resource to be deleted does not exist, and &quot;500 Internal Server Error&quot; for other errors</td>
</tr>
</tbody>
</table>

Figure 13: Actions of RESTful API (Source: Bojinov, 2015)
Resources can have multiple representations: a key feature of a resource is that they may be represented in a different form than the one it is stored; the format that REST support most is JSON.

Communicate statelessly: once deployed in a production environment, RESTful applications ensure scalability and high availability.

» (Bojinov, 2015)

Now that we have seen the services and server part, we will move in the web application part in a first time and then in the mobile application part.

2.3 Web application

In this part, we will explain in general about the web application, regarding the choice of technology, process and language that we made for building our cooking application.

2.3.1 Design patterns

A design pattern is a standard model for coding in order to design an application. There are three MV* design pattern different known nowadays: Model View Controller (MVC), Model View Presenter (MVP), and Model View ViewModel (MVVM).

Thus, when you start coding an application it is better to follow one of these design patterns rather than write code in the way we like. It is very important for the code maintainability. Moreover, it can help a lot when another developer enters in the middle of a project.

2.3.2 JavaScript

First of all, we decided to code in JavaScript for many reason. The first one is because it’s one of the languages that we share in common with our co-worker on this project. The second one is because JavaScript allow us to view our changes on the browser simultaneously with the changes made in the code. JavaScript also has many different frameworks for purposes, it can work with MV* patterns and JavaScript functions can run on client side, hence less computation on the server.

2.3.3 Framework AngularJS

After comparison and analysis between AngularJS, Backbone and Ember in our co-worker’s report, we decided to take Angular as framework for the web application in our project.

Angular is related to the design patterns that we quoted above but we didn’t have to choose one specifically between MVC, MVP and MVVM because Angular made its own: the MVW design pattern - W means ‘Whatever works for you’ (Minar, 2012).

Moreover, « AngularJS is turning out to be one of the best frameworks for building mobile apps across all major platforms, [which make a good point for the mobile app explained later].

One reason that Angular fits so well into the mobile developer toolbox is its features for interacting with backend web services and external data sources, [which will be useful for our external database]. Since most apps today are data-driven, it makes sense to use a tool built from the ground up to solve this problem.

Beyond that, AngularJS comes with some of the most modern and advanced software development practices ready-made and easy to use. We will find that building mobile apps with Angular is highly efficient and results in solid code that will scale up as us pile on the features. » (Lynch, 2014)

But the main reason of using Angular is the fact that:

« AngularJS helps keep our view (UI), data model, and application logic separate. This means Angular lets us reuse our web application logic on multiple devices on multiple platforms, while still enabling us to customize
the UI for each platform.

Previously, to target users across desktop, tablet, and mobile, our only option was to build a separate website for the Desktop, then build an iOS app with Objective-C, then an Android app with Java. We weren’t able to share any code, making application development expensive, time consuming, and risky.

With AngularJS, we can keep the functionality for our application (in controllers) separates from the way our application looks. This works regardless of whether we take a Responsive Web Design approach, or custom UIs for desktop and mobile. » (Lynch, 2014)

2.3.4 Angular concept design

Now that we have chosen AngularJS as framework for the web application, we will detail in this section how it works with the help of the figure below:

« First of all, when Angular starts, it will use the configuration of the module with the name defined by the ng-app directive in the template file, including the configuration of all modules that this module depends on.

In the example above: the template contains the directive ng-app="display-recipe". This tells Angular to use the display-recipe module as the main module for the application. The code snippet angular.module("display-recipe", ["services"]) specifies that the display-recipe module depends on the services module. By this, Angular uses the DisplayRecipe controller as well as the User service.

Controllers are here to link the data to the view and process the user actions with the help of the services. They know which elements in which views they have to display at some point. To do that, they make call to services which deals with the data represented in JSON. Then, these services make call to the RESTful API.

Now that Angular knows of all the parts of the application, it can create them. » (Angularjs, 2015)

2.3.5 Web application design

We finally decided to start by coding the mobile version of our application and then extend it to a web version on computer, because it’s better to begin with a little screen and continue on a much larger screen instead of the other way.
That’s why we decided to implement in a first time the mobile application and then the web application. With the help of the technologies that we have chosen throughout this report, like Angular mainly, the design will be just adapted with Html andCss coding. Moreover, regarding this part, we will just think about the Responsive web design of the web application in order to tell to our co-worker how to implement correctly the design of the web application that it can be complementary of the mobile application. For more information about the web application, we can refer his report.

Finally, we have covered the server part and the web application part. So, we will move in the mobile application part.

2.4 Mobile application

In this section, we will talk in general about the mobile application, regarding the choice of technology, process and language that we made for building our cooking application.

2.4.1 Platform Android

We can see two main platforms nowadays to develop a mobile application: iOS and Android. But for some reasons, like the costs that involved developing on iOS, we decided to deploy our mobile application on the Android platform in a first time. We could possibly later adapt this mobile application for android tablet, then deploy it on the iOS platform, adapt it again for iPad, and finally do the same for WindowsPhone, etc. But within 3 month, we think that only the Android mobile application will be available.

2.4.2 Ionic

We know that, « it is possible to develop apps in various ways. Going native and developing an app multiple times to achieve the best integration and potentially best user experience, is just one of a few options. Hybrid apps are on the rise and come in various flavours. The most common option is to base the app on web
technologies and use the fact that most devices support web standards, i.e. HTML, CSS and JavaScript. » (Ripkens, 2014)

![Figure 17: Different nature of applications (Source: Bavari, 2014)](image)

The solution we have chosen for the mobile application development is called Ionic. Ionic is a NodeJS framework sitting on top of Apache Cordova and Angular.

« One thing that is often missing when building a mobile app with web technologies is the lack of a native-style UI kit. Luckily, with Ionic, we get that in one free and open source package, complete with amazing AngularJS features. There is a huge collection of mobile-focused components and utilities for building great apps with AngularJS and web technologies. Ionic is focused on using Web Standards, which means an Ionic app will fit right into our frontend stack, from desktop to tablet to mobile, and will take advantage of any existing web development experience on our team. Ionic is part of a quickly growing community of AngularJS powered libraries that make building mobile apps with Angular incredibly fast and easy. » (Lynch, 2014)

That’s why we chose to use Ionic. Because Ionic is actually based on AngularJS and, according to the official website of the Ionic framework, also for all those reasons:

«
- Free and open source, Ionic offers a library of mobile-optimized HTML, CSS and JS components, gestures, and tools for building highly interactive apps. Built with Sass and optimized for AngularJS.
- Speed is important. Ionic is built to perform and behave great on the latest mobile devices. With minimal DOM manipulation, zero jQuery, and hardware accelerated transitions.
- Ionic uses Angular in order to create a powerful SDK most suited to develop rich and robust applications. Ionic not only looks nice, but its core architecture is built for serious app development, and Angular ties in perfectly.
- Ionic is modelled on popular native mobile development SDKs, making it easy to understand for anyone that has built a native app for iOS or Android. Just drop it in our code to get going, and push through Cordova when it's ready.

Develop once, deploy everywhere.
- Clean, simple, and functional. Ionic has been designed to work and display beautifully on all current mobile devices. With tons of popular mobile components, typography, interactive paradigms, and a gorgeous (yet extensible) base theme.
- Use just one command to create, build, test, and deploy our Ionic apps onto any platform.
- All we need to know are HTML, CSS, and JavaScript: the building blocks of the web.
- Built and maintained by developers and designers passionate about web technologies. Ionic focuses on standards compliant code, is forward thinking, and is managed through Github [which is good because we intend to use Github in order to share the code between us (explain later in this paper)].

» (Ionic)
Moreover, to use some special functionalities of the mobile itself like the camera, etc., we will also use two other frameworks as we said before: PhoneGap and Apache Cordova.

**Apache Cordova** is « a set of device APIs that allow a mobile app developer to access native device function such as the camera or accelerometer from JavaScript. Combined with a UI framework, [which is in our case the Ionic framework], this allows a mobile application to be developed with just HTML, CSS, and JavaScript.

When using the Cordova APIs, an application can be built without any native code (Java, Objective-C, etc.). Instead, web technologies (HTML, CSS, and JavaScript) are used. And because these JavaScript APIs are consistent across multiple device platforms and built on web standards, the application should be portable to other device platforms with minimal to no changes. » (Apache Cordova)

![Architecture of mobile application design using Ionic](source:image)

*Figure 18: Architecture of mobile application design using Ionic (Source: Scotter, 2015)*

We can see that Ionic and Cordova work perfectly together, even also with AngularJS. And now we will also add the PhoneGap framework.

**PhoneGap** is « an open source framework for quickly building cross-platform mobile apps using HTML5, Javascript and CSS. » (PhoneGap) It provides additional services and is powered by Cordova, that’s why we will use it.

« Building applications for each device—iPhone, Android, Windows Mobile and more—requires different frameworks and languages. PhoneGap solves this by using standards-based web technologies to bridge web applications and mobile devices. Since PhoneGap apps are standards compliant, they’re future-proofed to work with browsers as they evolve. » (PhoneGap)

Finally, here are all the technologies that we’ve intend to use in the implementation of our cooking application:

![All used technologies for the mobile application](source:image)

*Figure 19: All used technologies for the mobile application*
2.4.3 Testing

After finishing to build our mobile application, we have to test it and see if everything works correctly. So, concerning the testing part, we decided to use three types of test during this project:

❖ **Unit testing**

It concerns code testing to ensure that there are no errors in the code that has been written and that the application’s code is working correctly. Once they are written, they can be processed every time an amendment has been made to the code. So it is very important for the maintainability. Moreover, it can help to see if the code has been broken or not when another developer enters in the middle of a project.

**Karma** will be our test runner, which means it’s a command line tool that runs tests every time we update our code and warn us when our tests fail.

**Jasmine** will be our unit-test framework, which is a well-known framework compatible with Karma.

❖ **End-to-End testing**

They are focused on the “functional” tests and not on the code like unit testing. They ensure that the behaviour of the application is the one that is expected by the end users. **Protractor** will be our end-to-end test framework.

We began with the unit tests of the code, continuing gradually in the functional part of the application with the end-to-end tests, and finally ending with the user tests in reality users.

❖ **User testing**

« User testing can reveal the potential and the brakes use of an interface, once tested by its users. » (Ledeline, 2014) Thus, they allow for the proper observance of user behaviour.

« Indeed, users are numerous, different and don't have the same approaches, nor the same attitudes face to various digital interfaces ... Develop an application on IOS, put in the hands of an Android user, and see ambushes, misunderstandings, dead ends. It's like asking a grandmother to change her jam pot or Orange to use a red graphic charter. » (Ledeline, 2014)

Thus, even if we develop a cooking application just for Android Store, and not for IOS Store because of the time limit devoted to this project, then it will be well to consider just the behaviour and reflexes for Android users in the design of our application, because: « spontaneously the user doesn't think and apply what he used to do ... [Therefore], an interface must keep the same navigation codes on all its pages. » (Ledeline, 2014)

Finally, these tests will be done at the complete end of this project. It’s the users themselves who will test the whole application once implemented and deployed. They will have tasks to do, and we will take their feedbacks and advices to improve our application regarding their expectations.

2.4.4 Summary

After made the whole market analysis and be sure of our design with the help of the feedbacks of our survey, we finally decided which technologies we will use to do our cooking application: AngularJS, Ionic and Cordova.

To conclude, « AngularJS is a great choice not only for desktop apps, but increasingly for mobile apps as well. The combination of tight data binding across all parts of the application make developing data-driven apps with Angular fast and easy. Being able to share application logic across desktop, tablet, and mobile apps is a huge plus which will save an incredible amount of time and money, and make it easier to add features for all our apps. » (Lynch, 2014)

« First of, Ionic is based on AngularJS. Ionic is basically one AngularJS module and a stylesheet that can be used to build native-resembling apps. » (Ripkens, 2014) That’s why they work so well together and why we chose them both.
Moreover, if we have enough time, we have already some ideas of features to add to our cooking application in order to extend it more.

3 REQUIREMENTS

Firstly, we can see from our benchmarking that any cooking application has the goal to find and display recipes. They have also common tools like a search engine, shopping list, a meal planner and a unit converter. But they are only displaying recipes. They don’t help their user to improve their skills in cooking. That’s why we chose to be different by creating a cooking application that takes as a target user any person interested in learning how to cook and who wants to improve his skills for main goal.

Secondly, we want to help them as much as we can in order that they don’t give up when they encounter difficulties to make it. In order to do that, we will focus more on pictures and the idea of steps because we think that a picture can make them more confident than a whole paragraph of words.

Regarding all these first ideas, here is all the requirements of our application:

3.1 Objectives

In this part, we will explain all the requirements that we want to implement for this project. Which mean that, by topic, we have specific requirements for our application as we represented it in the following table:

<table>
<thead>
<tr>
<th>Easy Cooking Application</th>
<th>Priority</th>
<th>Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find a recipe</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>Find a recipe including some ingredients</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Find a recipe by category (breakfast, lunch, dinner, dessert, enter, main meal, meat, fish…)</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Sort recipes (last added, alphabetic order…)</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Bookmark recipes</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Mark recipe</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Comment recipe</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Create your own recipe step by step</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>Add main picture of the recipe</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>Possibility to add picture in a step</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Possibility to add timer in a step</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Add picture of recipe result by the users of the recipe</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Add calories calculator</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Add a converter metric for ingredients</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Symbols/list of special utensil for recipe</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Add difficulty level of recipe</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Help for people on difficult vocabulary</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>Change profile</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>Visit profile</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>Visit recipe profile</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Sign up at the first launch quickly with only few information</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Notifications</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Follow a user</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>Login by Facebook/Google</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>Share content Facebook/Google</td>
<td>5</td>
<td>9</td>
</tr>
</tbody>
</table>
As we can see, we add some priority about some specific requirements. All assembled, these 5 features, represented by the grey boxes, are the base of any cooking application that we can find in the market right now. So, we have in priority:

- Find a recipe among all recipes in the application
  Maybe in a research bar in the top of the screen
- Add a main picture of your recipe at the end of the creation
  In order to see the final result of the recipe
- Change your profile
  By changing your description, picture, name, email… for example
- Visit other users profile
  By clicking on their name or nickname
- Handle multi-languages
  English, French, Spanish, German, Chinese…

But in order to be different from them, as we want, we also add some priority about other requirements. All assembled, these 10 features, represented by the purple boxes, are the adding value of our special cooking application that we decided to build regarding the user’s needs found in during the market analysis. So, we have in priority:

- Create your own recipe step by step
  Clearly define the order on how to do the recipe by creating a step one after another one
- Add a picture in a step
  In order to see more specifically how to do a step
- Add a timer in a step
  By seeing the time flows, you can know if you’re doing well and in time
- Add a picture of the recipe result by other users
  Once you made the entire recipe, you can share a picture of your result with the community
- List of special utensil for recipe
  In order to help them to know if they have the necessary material to make the recipe before they start
- Sign up at the first launch
  At the first launch, sign up very quickly with little information to not bother the user to doing it
- Remove manually ingredients from shopping list
  By sliding with your finger
- Tip for choosing ingredients in supermarket
  Thanks to a button maybe
- Quizzes on different subject about cooking
  About the history of cooking, cultural, maybe about picture of ingredients…
- Badges awarded depending on user’s investment in the application
  Maybe from beginner to expert

We have chosen to highlight and prioritize these 15 features for two reasons:

The first one, represented by the 5 first points cited above, is that these features are those of any basic cooking application as we have seen in the benchmarking early in the Literature review section.

The second, represented by the 10 last points, is that we wanted a cooking application that learns primarily
to cook, learn also the history of the kitchen, has games and evolution in the learning, been fun and interactive. We wanted especially not copy all cooking applications that already exist and are actually simple recipes online books.

Moreover, every user will have his profile page and will be able to add, change, delete and comment the recipes that he posted on our application. But there will be also an administrator, as a “master referent”, who will have the full right and be able to change some of the application’s settings in order for example to verify if there is no copyright.

3.2 Achieved objectives

We described all the features that we would ideally implement in our application. But obviously, even after put a priority to some specific requirements, we encountered some difficulties in implementing some of these requirements, which delayed us in our progression. We also were a little ambitious for the realization of this project at first. Thus, here are the features that we have really implemented (orange boxes) in three month in our application:

<table>
<thead>
<tr>
<th>Easy Cooking Application</th>
<th>Topic</th>
<th>Description</th>
<th>Priority</th>
<th>Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recipes</td>
<td>Find a recipe</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Recipes</td>
<td>Find a recipe including some ingredients</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Recipes</td>
<td>Find a recipe by category (breakfast, lunch, dinner, dessert, enter, main meal, meat, fish…)</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Recipes</td>
<td>Sort recipes (last added, alphabetic order…)</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Recipes</td>
<td>Bookmark recipes</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Recipes</td>
<td>Mark recipe</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Recipes</td>
<td>Comment recipe</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Recipes</td>
<td>Create your own recipe step by step</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Recipes</td>
<td>Add main picture of the recipe</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Recipes</td>
<td>Possibility to add picture in a step</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Recipes</td>
<td>Possibility to add timer in a step</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Recipes</td>
<td>Add picture of recipe result by the users of the recipe</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Recipes</td>
<td>Add calories calculator</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Recipes</td>
<td>Add a converter metric for ingredients</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Recipes</td>
<td>Symbols/list of special utensil for recipe</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Recipes</td>
<td>Add difficulty level of recipe</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Recipes</td>
<td>Help for people on difficult vocabulary</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Users</td>
<td>Change profile</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Users</td>
<td>Visit profile</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Users</td>
<td>Visit recipe profile</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Users</td>
<td>Sign up at the first launch quickly with only few information</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Users</td>
<td>Notifications</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Users</td>
<td>Follow a user</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Users</td>
<td>Login by Facebook/Google</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Users</td>
<td>Share content Facebook/Google</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Shopping list</td>
<td>Add ingredients manually</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Shopping list</td>
<td>Add ingredients through recipe</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Shopping list</td>
<td>Remove manually</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Game</td>
<td>Quiz on different subject (History / Cultural / Photo…)</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Game</td>
<td>Badges awarded depending on user’s investment in the application</td>
<td>5</td>
<td>9</td>
</tr>
</tbody>
</table>
As we can see, only 14 features (over 32) have been really implemented in these three-month. Which is just the half. It hasn’t been easy, especially when we had to connect the mobile application to the camera on the mobile phone for taking pictures.

Regarding on what all has been achieved; we will go now on the explanation of the architecture and design of all these features in our mobile application.

4 ARCHITECTURE AND DESIGN

This chapter will cover the whole system architecture and will explain deeply the user interface design of our mobile cooking application.

4.1 Architecture overview

First of all, our application can be split up into 5 parts:
- General: this part is where the application starts and also contains some views that other parts can access, like the header for example.
- User: this part will contain the management of user profile page, which means all the details of each user in the application displayed in their profile page and also the possibility to edit your personal details.
- Connection: this part contains the login and sign-up pages when a user wants to enter into the application to use it.
- Recipe: this part will contain the creation, search and display about recipes.
- Game: this part will contain the creation, search and display about games.

As we saw in the section where how Angular works above, we can now design our application like this:

![Architecture overview diagram]

*Figure 20: Architecture overview*

So, we have for each functionality for each part into our application, a template file in HTML with its style sheet in CSS and a JavaScript file.
The JavaScript file is the controller of the respective part, which will manage the data to display on its respective page, with the help from another JavaScript file. This other file represents all the services in our application. Each controller uses this file because it will make the connection to the server and the database, and in the same time will deal with all the data in our application.

We can represent these links for the create-recipe part of our application for example like this:

![Class design](image)

*Figure 21: Link to server by a package design*

Now, we can have a look to the class design of the whole application:

![Class design](image)

*Figure 22: Class design*
4.2 Design

This section will look deeper in the user interface and software design. More details about implementations will be provided in chapter 5.

4.2.1 User interface design
When you start to create your own recipe you have to firstly enter some information (3 inputs and 2 selects), then the list of ingredients, and the utensils needed to do it (optional).

When everything is enter, you can click on the “Next” button to enter after the details about each steps. If there are errors or every required information isn’t fulfilled, an error message in red above the button will appear and you won’t be able to go to the next page of the creation of your recipe if you don’t modify it.

When you create a step, you have to also enter some information (1 textarea for the description).

You can set a timer (optional) and add a picture (optional). When you click on these 2 buttons you can see appear (or disappear on the 2nd time) these 2 things just below.

If you have just 1 step, you can click on the “Finish” button. If there are errors or every required information isn’t fulfilled, an error message in red above the button will appear and you won’t be able to go to the next page if you don’t modify it.

With these 2 buttons you can go back (to the page with general info about the recipe) or add another step (step 2) if there are no errors and every required information is fulfilled.

With these 2 buttons you have a direct connection with your phone’s pictures (import one) and mobile phone’s camera (take one). Once you choose one you can upload it with the “Upload” button in order to change it. See example after:
At any point, you can insert a step if you forgot one. Example here if we click on this link: it will add a blank step 2 and shift the rest of created steps “to the right” (so put this 2 into a 3, the 3 into a 4, etc.).

With these 2 buttons you can go back (step 1 which has been already fulfilled completely to see it again) or add another step (step 3) if there are no errors and every required information is fulfilled.

With these 2 buttons you can go back (step 3 which has been already fulfilled completely to see it again) or add another step (step 4) if there are no errors and every required information is fulfilled.

Once you entered all your steps, and click on the “Finish” button.
If there are no errors and that every required information in the last step is fulfilled, you will end on this page:

Again, with these 2 buttons you have a direct connection with your phone’s pictures (import one) and mobile phone’s camera (take one). Once you chose your final picture (the one which will appear in the front page of your recipe), you can upload it with the “Upload” button in order to change it. See example after:

Once you put your final picture, you can save your recipe! Then, you will directly go on the display of your recipe like this:
Perfect rice salad

Difficulty: ★★★★★ 60 mins ① 3 pers

150 G of Tuna
50 G of Anchoises
Mayonnaise
Salt
Pepper

Ingredients:
- 150 G of Tuna
- 2 Eggs
- 2 Tomatoes
- 1 Cucumber

Utensils:
- Pan
- Knife
- Salad bowl

Steps:
1. Cook the rice and put it in a salad bowl.
2. Cut the tomatoes in moon shape.
3. Use half of the cucumber to make slices.
4. Boil 2 eggs for 10 mins.
5. Cut the anchovies in small sticks.
6. When eggs are cooked give them
7. Put all ingredients together. Add mayonaise
8. Ice in the fridge for 2 hours. Then eat.

About how to comment a recipe:

So, you can put a mark thanks to the stars, and put your comment in the textarea and finally click on the "Add comment" button.

You can see all the previous comments regarding this recipe.

If there are errors or every required information isn’t fulfilled, an error message in red above the button will appear and you won’t be able to add it if you don’t modify it.

You can still see all the previous comments regarding this recipe, but now there is your comment too in the list after clicking on the “Add comment” button.

During the process to add a comment, you can’t click again on the “Add comment” button.

This symbol appears only for your comments and it gives you the possibility to delete it.
Recipe display-recipe.html (3/4)

About following a recipe with steps, after clicking on the “Start recipe” button:

- If you want to go to the next step.
- If you want to go to the previous step.
- When you click on the “play” button:
- When it’s the last step, the “End” button appears, and you can click on it to finish the recipe.

You have for each step: its number to see your progression, the description of what to do, a picture to show you exactly, and sometimes a timer to help you with limited time.

Recipe display-recipe.html (4/4)

When you finished the last step, by clicking on the “End” button, you will end on this page:

Thank you a lot for sharing!

The idea here is to share your result with the community by importing/taking a picture of your plate after following the recipe.

You can choose to like it, by clicking on the little black hand on the right. Once done, this will appear on the screen:

Congratulations!

Share a picture of your result with the community.

Import a picture
Take a picture

Do you like it?
Give a mark: ★★★★★
Type your comment here...

You still have the possibility to comment the recipe and also you can see last comments about it if you slide to the bottom.
When you want to search a recipe, you will end on this page:

Input to enter the title of a recipe that you are searching for.

Examples of result if you enter a word:

Even if you don’t enter any word into the input above, or because you don’t know what to search especially, a random selection of existing recipes in our application will appear on the screen.

When you start the application, you will end on this page:

If there are errors or every required information isn’t fulfilled, an error message in red above the button will appear and you won’t be able to go to the next page if you don’t modify it.

You can either:
- LOGIN: enter your email/username and password and click on “Submit” button, or
- SIGN-UP: by clicking on the link below the “Submit” button.

During the process to login, you can’t click again on the “Submit” button:
If there are errors or every required information isn’t fulfilled, an error message in red above the button will appear and you won’t be able to go to the next page if you don’t modify it.

When you want to sign-up, you have to enter some information (3 inputs), then click on “Validate” button.

Once you validate, you will end on this page after, in order to pick an username:

During the process to sign-up by clicking on the “Register” button, you can’t click again on it because it will be blocked:

Once you login (“Submit” button) or sign-up (“Register” button), you will go directly to the search recipe page.

If you sign-up, your user profile will be created.

You can find your user profile by clicking on the « avatar » in the top right corner to access to the menu and go to you profile page.
Easy Cooking Application – Pauline Michaud

**User**

**profile.html**

The first time that you come to your profile page, generally after sign-up, you will end on this page:

So you can edit your profile details by clicking on the “Edit” button.

Everything about you will be blank.

And after some days to use our application, you will appear in the bottom of the page your recent activity.

**User**

**profile-edit.html**

When you clicked on the “Edit” button, you will end on this page:

Example of entered details on the edit profile page:

If there are errors or every required information isn’t fulfilled, an error message in red above the button will appear and you won’t be able to go to the next page if you don’t modify it.

When you click on “Profile Photo” you can import or take one.

You can edit your profile details here (4 inputs), then click on “Save” button.

3 inputs also here if you want to change your password before save everything.

Slide to the bottom.
When you click on the « academic » button in the header, you will reach the games menu:

Click on any of these buttons to go where you want.

When you want to search a quiz, you will end on this page:

Input to enter the title of a quiz that you are searching for.

Examples of result if you enter a word:

Even if you don’t enter any word into the input above, or because you don’t know what to search especially, a random selection of existing quiz in our application will appear on the screen.
If there are errors or every required information isn’t fulfilled, an error message in red above the button will appear and you won’t be able to go to the next page if you don’t modify it.

When you start to create your own quiz you just have to enter its title (1 input), then import/take a picture (optional).

Then you can click on the “Create all questions” to create the questions of your quiz.

If there are errors or every required information isn’t fulfilled, an error message in red above the button will appear and you won’t be able to do what you want if you don’t modify it.

You have entered the title of your question here.

Choose that your answers will be text.

Then you can enter your first answer here.

Then, this answer will appear in the list of answers below with:
- X if it’s wrong
- V if it’s correct
The symbol (·) is here if you want to delete this answer.

Then you can choose if this answer is the right or the wrong one by clicking on the appropriate button.

Then, you can add other answers for a total number between 2 and 4 answers.
If there are errors or every required information isn’t fulfilled, an error message in red above the button will appear and you won’t be able to go to the wanted page if you don’t modify it.

Example of a finish created question:

With this button, you can create or see the next question if she is already created.

With this button, you can delete the current question.

With this button, you finish your quiz. Then, you will go directly to the display quiz page of your quiz.

If there are errors or every required information isn’t fulfilled, an error message in red above the button will appear and you won’t be able to do what you want if you don’t modify it.

You have entered the title of your question here.

Choose that your answers will be image.

Then you can enter an optional text to go to the respective picture here.

Then you can import or take a picture.

Then, this answer will appear in the list of answers below with:
- X if it’s wrong
- V if it’s correct
The symbol (1) is here if you want to delete this answer.

Then you can choose if this answer is the right or the wrong one by clicking on the appropriate button.

Then, you can add other answers for a total number between 2 and 4 answers.

Like this:
If there are errors or every required information isn’t fulfilled, an error message in red above the button will appear and you won’t be able to go to the wanted page if you don’t modify it.

With this button, you finish your quiz. Then, you will go directly to the display quiz page of your quiz.

Then, this page will appear (starting with the first question of the quiz):

So, you can click on the "Start quiz" button.

You can see your current score throughout you’re doing the quiz.

Now you can choose your answer like this:

And finally click on the "Submit" button to go to the next question of the quiz.
Then, this page will appear (starting with the first question of the quiz):

So, you can click on the “Start quiz” button.

You can see your current score throughout you’re doing the quiz.

Now you can choose your answer by clicking on the pictures.

And finally click on the “Submit” button to go to the next question of the quiz.

Once you answered the last question of the quiz, you will end on this page:

This is the summary of your results for this particular quiz.

This is your final average score of each scores that you made for different other quizzes.

You can choose to like it, by clicking on the little black hand on the right. Once done, this will appear on the screen:

You can either retry it or search a new quiz.
4.2.2 Database and server design

There is one database in use for this project: a NoSQL database on the server side. It will be implemented by our co-worker, but we won’t skip its design in this report because it is part of the good understanding and organization of this report. On the other hand, we can find more information on it in his report.

So, we can see in Appendices D how our data is organized, by showing what type is each of the variables that we will use in the implementation of our mobile cooking application.

5 IMPLEMENTATION

With the aim of creating a fun and interactive cooking application, we firstly set up the foundations of our application with the common features that define a basic cooking application (general, recipe, user and connection parts), and then we add our ideas with games that define our cooking application from the others (game part).

Moreover, we decided to work with Git, a distributed revision control system. It allows us to work together in using the same files. In this way it will increase the speed and the quality of our project. Thus, in order to share our code, we have to put in our command line (once we are in the directory MyCookApp representing the mobile cooking application file) these two lines at the end of each significant modification of our code:

- `git commit –am "message"`
- `git push –u origin master`

Then, it will appear this message on the screen of our command line when everything has been correctly saved:

![Git lines of code](image)

Figure 23: Git lines of code

5.1 Client side

5.1.1 Implementation overview

So, for the implementation of our cooking application, we are coding in Html, Css and JavaScript on the software ATOM for Mac. Then, we downloaded the Ionic framework in our directory MyCookApp representing the mobile cooking application file, which allows us now to launch our application firstly in a browser, with the help of this line of code in our command line:

- `ionic serve`
Then, our cooking application will be opened in Chrome.

But we are implementing a mobile application, so in order to see what really appears, we have to launch our cooking application on a real mobile phone. That’s why we downloaded the Ionic platform for Android in our directory MyCookApp representing the mobile cooking application file, in order to build our Android cooking application, like this:

```
ionic platform add android
android sdk
ionic build android
```

Now, it allows us to launch our application on a real mobile phone, with the help of this line of code in our command line (once we’ve connected the mobile phone to the Mac in P2P mode):

```
ionic run android
```

Then, our cooking application can be opened on the mobile phone.

But now we will go concretely in the subject of this section, which means, we will explain deeper into our code how we implement each part of our mobile cooking application.

Furthermore, we won’t enter into details about all of our checked errors, but each time that the user has to enter something in our application (with inputs, textareas, etc.), we will do some checks to know if the entered input is correct and fit with the implementation of the data in our database. We can anyway show a quick example of how we checked the inputs that can help us to know if the input is wrong and display an error message on the screen in that case:

```
//All ingredients in recipe
scope.ingredients = [];
//Current ingredient in edition
scope.currentIngredient = {};

function checkIngredientInfo()
{
    scope.errors.ingredient = {};
    // Name of ingredients
    if(!scope.currentIngredient.name || scope.currentIngredient.name == ''){
        scope.errors.ingredient.name = 'Name of ingredient should not be empty.';
    } else if(!isNaN(scope.currentIngredient.name)){
        scope.errors.ingredient.name = 'Name of ingredient should not be a number.';
    } else if(scope.currentIngredient.name.length < 2){
        scope.errors.ingredient.name = 'Name of ingredient should be at least 2 characters.';
    }
}
```

*Figure 25: Check errors example*
5.1.2 General part

Menu

Once the user reached the menu by clicking on his “Avatar picture” in the header in the right top corner of the application, he will have then to choose what he wants to do now. Thus, when he clicks on the “Log out” button, the html template will launch the “logout()” function that he can find in its respective MenuCtrl controller connected to him. This controller will in a first time find who is currently connected by calling the “currentUser()” function from the AuthenticationService service. After that, inside the “logout()” function, the controller will call the AuthenticationService service again, which has also a “logout()” function.

This service will delete the currently connected user from the sessionStorage and he will be logged out from the application.

Picture-upload

![Diagram of Picture-upload implementation](image-url)
When every time the user clicks on a button to take a picture, the html template will launch the “takePicture()” function that he can find in its respective PictureUploadCtrl controller connected to him. Inside this function, the controller will call the Camera service with the “getPicture()” function. This service will make the connection to the camera of the using mobile phone and after taking, get the picture from it, or to the pictures directory inside the mobile phone and get directly the picture from it.

5.1.3 Connection part

Login

![Login implementation diagram]

When the user clicks on the “Submit” button to login into our application, the html template will launch the LoginCtrl controller connected to him and especially the “login()” function.

Inside this function, the controller will call in a first time the UserService service with the “login()” function. This service is a POST method among our RESTful API connected to our database. In a second time, the controller will call the AuthenticationService with the “setUser()” function.

With the help of this full function we will be able to know who wants to connect to our application in order to give him the right or not to enter in it if he has already an account in the list of our users.
Sign-up

When the user clicks on the “Register” button to sign-up into our application, the html template will launch the SignUpCtrl controller connected to him and especially the “register()” function.

Inside this function, the controller will call in a first time the UserService service with the “signUp()” function. This service is a POST method among our RESTful API connected to our database. In a second time, the controller will call the AuthenticationService with the “setUser()” function. And in a third time, the controller will call the Header service with the “show()” function, in order that the header appears once the user has signed-up correctly.

With the help of this full function we will be able to create a new user account for this user with his username, password and email.
5.1.4 User part

Profile

Figure 30: Profile implementation

When the user enters on his profile page, the html template will launch the ProfileCtrl controller connected to him. Then, the controller will call in a first time the UserService service with the “getUser()” function to get in our database all the information of a user. This service is a GET method among our RESTful API connected to our database. In a second time, the controller will call the AuthenticationService with the “currentUser()” function to find the connected user among all users.

With the help of this full function we will be able to display the right profile page of the connected user.
Profile-edit

When the user clicks on the “Edit” button on his profile page, the html template will launch the ProfileEditCtrl controller connected to him. Then, the controller will call the UserService service with the “getUser()” function to get in our database all the information of a user. As we saw in the profile implementation, this service is a GET method among our RESTful API connected to our database, and the user will be able to edit only his profile’s details, not the ones of another user that he was watching the profile page.

Then, when the user clicks on the “Save” button on his edit profile page, the html template will launch the “saveAll()” function that he can find in its ProfileEditCtrl controller again. The controller will call the UserService with the “updateUser()” function to update the information regarding this connected user. This service is a PUT method among our RESTful API connected to our database.

With the help of this full function we will be able to display, update and save the details of the right profile page of the connected user.
5.1.5 Recipe part

Create recipe

When the user clicks on the “Save all” button at the end of creating his recipe, the html template will launch the “finishRecipeAll()” function that he can find in its respective CreateRecipeCtrl controller connected to him. Inside this function, the controller will call the RecipeService service with the “create()” function. This service is a POST method among our RESTful API, which mean it will create a new entry in our collection of recipes in our server connected to our database.
When the user clicks on a recipe among the list of his search, or click on the “Add comment” button, or click on the symbol representing a “little thumb” to like a recipe, or click on the red cross button to delete his comment, the html template will launch the DisplayRecipeCtrl controller connected to him and the “addMyComment()”, “like()” and “deleteMyComment()” functions.

Inside these functions, the controller will call the RecipeService service with the “get()”, “createComment()”, “likeRecipe()” and “deleteComment()” functions. These services are respectively GET, POST, PUT, DELETE methods among our RESTful API connected to our database.

In the same time, we created the “isUser()” function in our DisplayRecipeCtrl controller. This function will call the AuthenticationService service and especially its “currentUser()” function. With the help of this function we will be able to know who is currently connected on our application in order to give him the ability to delete only his comments and to know the author when he is creating a comment.
Search recipe

![Diagram of search recipe implementation]

So, when the user enters in the search page or when he enters a word in the search input, the html template will launch the “init()” and “searchRecipe()” functions that he can find in its respective SearchRecipeCtrl controller connected to him. Inside these functions, the controller will call the RecipeService service with respectively the “getTrends()” and “getSearchRecipe()” functions.

These two services are GET methods among our RESTful API, which mean it will list all the collection of recipes that we have in our server connected to our database, or only the ones who have the word entered in the search input by the user.
5.1.6 Game part

Create game

Figure 35: Create quiz implementation

When the user clicks on the “Finish quiz” button at the end of creating his quiz, the html template will launch the “finishQuiz()” function that he can find in its respective CreateQuizCtrl controller connected to him. Inside this function, the controller will call the GameService service with the “create()” function. This service is a POST method among our RESTful API, which mean it will create a new entry in our collection of games in our server connected to our database.
Display game

Figure 36: Display quiz implementation

When the user clicks on a quiz among the list of his search, or click on the symbol representing a “little thumb” to like a quiz, the html template will launch the DisplayQuizCtrl controller connected to him and the “like()” function. Inside these functions, the controller will call the GameService service with the “get()” and “likeGame()” functions. These services are respectively GET and PUT methods among our RESTful API connected to our database. With the help of these functions we will be able to display the right quiz that the user wanted to try and also to know if he liked it.
Search game

So, when the user enters in the search page or when he enters a word in the search input, the html template will launch the “init()” and “searchQuiz()” functions that he can find in its respective SearchGameCtrl controller connected to him. Inside these functions, the controller will call the GameService service with respectively the “getTrends()” and “getSearchQuiz()” functions.

These two services are GET methods among our RESTful API, which mean it will list all the collection of games that we have in our server connected to our database, or only the ones who have the word entered in the search input by the user.

5.2 Server side

Concerning the processing of data and the general implementation of the server part of our application, it will be implemented by our co-worker.

Moreover, concerning the database part, our co-worker used a NoSQL database that we put in Appendices D to see how is built all of our data in the application. And as we saw previously, he has also implemented our server REST API in NodeJS.

On the other hand, we can find more information on it in his report.
5.3 Testing

Once we made the major implementation of our mobile application, we started to do the whole testing of our application in the same time that we continued to develop it. This has to be done in order to see if everything works correctly every time that we are moving on in the entire implementation. So, regarding the three types of test that we chose in the Literature review, here are the tests that we really achieved for our cooking application:

Unit testing

With these tests, we will ensure that there are no errors in the code of our application. We firstly downloaded Karma and Jasmine tools, as we decided also in the Literature review before, in order to be able to make those tests.

Here are the unit tests that we actually implemented in the allotted time:

![Figure 38: Unit tests](image)

As we can see, we didn’t have enough time to do the unit test (red boxes) for some special functionality in the recipe part; but also for the whole connection part (login and sign-up), the user part, the games part and the general part.

Otherwise, a unit test is composed for example like this:

```javascript
it('should create errors when wrong parameters for add utensil', function(){
  scope.addUtensil();
  expect(scope.errors.currentUtensil).toEqual('Please enter an utensil name.);
  expect(scope.utensils).toEqual([]);
  expect(scope.currentUtensil).toEqual('');

  scope.currentUtensil = '42';
  scope.addUtensil();
  expect(scope.errors.currentUtensil).toEqual('Utensil should not be a number.');</expect
  expect(scope.utensils).toEqual([]);
  expect(scope.currentUtensil).toEqual('fork');
  scope.addUtensil();
  expect(scope.errors.currentUtensil).not.toBeDefined();
  expect(scope.utensils).toEqual(['fork']);
  expect(scope.currentUtensil).toEqual('fork');
});
```

![Figure 39: Unit tests example code](image)
Then, in order to test our code, we have to run Karma in our command line (once we are in the directory MyCookApp representing the mobile cooking application file) with this line:

```bash
karma start tests/my.conf.js
```

Then, it will open Karma in Google Chrome, to see that it is connected well to our entire application’s code, like this:

![Karma connection](image)

*Figure 40: Karma connection*

And after that, we can see appears in our command line the result of our tests:

![Unit tests results](image)

*Figure 41: Unit tests results in command line*

**End-to-end testing**

Unfortunately, we didn’t have enough time to do the end-to-end tests of our cooking application.

**User testing**

These tests will be done at the complete end of this project, in the Evaluation section below, because with them we will be able to evaluate the potential and breaks of our whole cooking application.

### 5.4 Problem encountered

During this implementation we encountered some difficulty, especially when we wanted to make the connection between the mobile phone and the cooking application, in order to access to the pictures of the phone and to the camera to take pictures.

To resolve this problem, we researched extensively online. Our co-worker also helped us during few days. We made a lot of tries and tests with the solutions that we found online. And we couldn’t skip this problem because this connection is in the foundation of this application. Take and import pictures is most important necessity for this application to reach its goals. So we had to solve it even if it took time.

When we finally made it, we saw that it delayed us a lot regarding the progression of the project. Hence, not leaving us with enough time to do all the unit tests and any of the end-to-end tests after. But we were glad that the connection to the camera of the mobile phone worked fine.
6 EVALUATION

Finally, we decided to evaluate our mobile cooking application. Moreover, in order to have the best evaluation possible, nothing was better than a user testing evaluation among our future users on the field. The objective of performing these tests is to gather enough data to observe what has been achieved, and to make decisions about our design and implementation in order to improve it, based on the level of satisfaction of our users after they’ve done some tasks on the application.

First of all, we will give to our candidates just some general information about the purpose of our application, without any detail on how to do something special on it because we want to observe their behaviour when they are doing the tasks.

Then, we will give them the user testing form – Appendices F. They will have to do each task one by one, put a difficulty number (which is a five point rating system to quantify the level of difficulty to do the task: from 0-very easy to 5-very difficult) and note their comments, advice, ideas, etc. to improve it. The complete list of what task they had to do is presented in Appendices F.

We will be sat next to them in order to observe their behaviour during the user testing evaluation, and if they have any troubles they will be able to ask us. Moreover, we could also ask them special information in addition of the evaluation regarding some parts of our application during the test to have better feedbacks.

With the help of their notes and advices during this user testing evaluation, we will be able to improve our application regarding their expectations.

6.1 Evaluation results

For the evaluation, 15 persons were agreed to do our user testing evaluation on our mobile cooking application (20-30 minutes per persons). The following figures contain the results of the evaluation:

Figure 42: User testing evaluation – results 1
Figure 43: User testing evaluation – results 2

Figure 44: User testing evaluation – results 3
6.2 Analysis of results

According to the results just presented in the section above, the total number of people who answered our survey is 15. We can see that the number of male and female is approximately the same (9 males for 7 females). But there is a real majority in all our candidates who are aged between 15 and 25 years old (10 of them) and are students (11 of them). Regarding their activities nowadays, there is also 4 employed persons, but none unemployed person.

As we can see in the figure 43, the majority of tasks was found easy to do it by our users; especially finding a quiz, searching a recipe and creating an account. But in general for all the tasks as well; which is good as results for our applications because it was one of our purpose. We wanted a fun and mainly easy cooking application, which is apparently the case. The results in the figure 44 are the same as previous in the figure 43 but displayed differently. However, they also confirm this observation.

From their feedbacks, the general comment was, once they’ve tried and known each button in the header they were able to find easily what they were looking for and to do it just after. Here are some recurring comments about our mobile cooking application during the test:

- They had some troubles with the slide; actually when they were moving to another page in the application, the page wasn’t displayed from the top.
- When they were playing a quiz where answers were pictures, they wanted to see in green/red if they were right or wrong before going on the next question.
- They wanted something more obvious and fun to display their scores during they were doing a quiz and also at the end for the summary.
- They don’t want to reload the page once they edited and saved their new profile’s information.

From our analysis, when we were looking at them during they were doing the test, we noticed some behaviours and ideas in order to improve in the future our mobile cooking application like:

- Increase the size of some buttons because they were too small and the user spent time to find it (sign-up link, edit and save buttons for the profile page, …).
- For “liking” a recipe or a game, increase the scope of the “thumb” symbol because they didn’t push on this symbol but on the text “Do you like it?”.
- State firmly when some information is optional and not mandatory required because, for example, they always put a timer when they were creating a recipe’s step, even if the timer was just for 2 minutes.
- They were confused a lot after changing their profile’s information on the app because after edit and save them nothing changed; actually they needed to reload the page, so we deduced that this behaviour isn’t normal.
- The slide was also a recurrent problem.

At this point, we can say that this user testing evaluation really helps us to see the gaps in our mobile cooking application. We are now able to improve our design and implementation in order to satisfy more their expectations facing some parts of our application.

Now, in the next section just below, we will focus more on the management of the entire project.
7 PROJECT MANAGEMENT

Firstly, we will see the project plan that we made in the beginning with the Gantt chart. Then, we will provide and explain what methods we used and how we managed our time during this project. Finally, we will conclude the next section with the real project plan that actually happened.

7.1 Project plan

At the beginning of this project and as agreed with our supervisor, the work planning has been done as a Gantt Diagram. We can find 5 significant parts in this project plan: Pre-research & Analysis, Research & Analysis, Implementation, Tests and Production, for which we will provide details later.

Master Schedule

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
<th>Predecessors</th>
<th>Resource Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Start</td>
<td>0 days</td>
<td>Tue 27/01/15</td>
<td>Tue 27/01/15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 Pre-research &amp; Analysis</td>
<td>57 days</td>
<td>Tue 27/01/15</td>
<td>Fri 03/04/15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 Berenrachting</td>
<td>20 days</td>
<td>Tue 27/01/15</td>
<td>Mon 23/02/15</td>
<td>1</td>
<td>Arnaud</td>
</tr>
<tr>
<td>4 Target evaluation</td>
<td>20 days</td>
<td>Tue 27/01/15</td>
<td>Mon 23/02/15</td>
<td>1</td>
<td>Pauline</td>
</tr>
<tr>
<td>5 Architecture design</td>
<td>50 days</td>
<td>Tue 27/01/15</td>
<td>Thu 26/03/15</td>
<td>1</td>
<td>Arnaud</td>
</tr>
<tr>
<td>6 Work environment</td>
<td>30 days</td>
<td>Tue 27/01/15</td>
<td>Fri 06/03/15</td>
<td>1</td>
<td>Arnaud</td>
</tr>
<tr>
<td>7 Functional researches</td>
<td>37 days</td>
<td>Tue 24/02/15</td>
<td>Thu 02/04/15</td>
<td>3,4</td>
<td>Pauline</td>
</tr>
<tr>
<td>8 Professional, Ethical &amp; Legal issues</td>
<td>37 days</td>
<td>Tue 24/02/15</td>
<td>Thu 02/04/15</td>
<td>3</td>
<td>Pauline</td>
</tr>
<tr>
<td>9 End of pre-research</td>
<td>0 days</td>
<td>Fri 03/04/15</td>
<td>Fri 03/04/15</td>
<td>3,4,5,6,7,8</td>
<td>Arnaud, Pauline</td>
</tr>
<tr>
<td>10 Research &amp; Analysis</td>
<td>90 days</td>
<td>Fri 03/04/15</td>
<td>Wed 05/04/15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 Functional specifications</td>
<td>10 days</td>
<td>Fri 03/04/15</td>
<td>Sat 16/05/15</td>
<td>7</td>
<td>Pauline</td>
</tr>
<tr>
<td>12 Technical specifications</td>
<td>80 days</td>
<td>Sun 17/05/15</td>
<td>Tue 04/08/15</td>
<td>5,6,11</td>
<td>Arnaud, Pauline</td>
</tr>
<tr>
<td>13 Surveys &amp; Analysis</td>
<td>10 days</td>
<td>Fri 03/04/15</td>
<td>Sat 16/05/15</td>
<td>7</td>
<td>Pauline</td>
</tr>
<tr>
<td>14 Documentation specification</td>
<td>0 days</td>
<td>Wed 05/08/15</td>
<td>Wed 05/08/15</td>
<td>11,12,13</td>
<td>Arnaud, Pauline</td>
</tr>
<tr>
<td>15 Implementation</td>
<td>80 days</td>
<td>Sun 17/05/15</td>
<td>Wed 05/08/15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 Back-end architecture</td>
<td>20 days</td>
<td>Sun 17/05/15</td>
<td>Fri 05/06/15</td>
<td>5,6,11</td>
<td>Arnaud</td>
</tr>
<tr>
<td>17 Web services</td>
<td>30 days</td>
<td>Sat 06/08/15</td>
<td>Sun 05/07/15</td>
<td>16</td>
<td>Arnaud</td>
</tr>
<tr>
<td>18 Mobile application</td>
<td>80 days</td>
<td>Sun 17/05/15</td>
<td>Tue 04/08/15</td>
<td>5,6,13,8,11</td>
<td>Pauline</td>
</tr>
<tr>
<td>19 Web application (architecture)</td>
<td>40 days</td>
<td>Sun 17/05/15</td>
<td>Thu 25/06/15</td>
<td>5,6,11</td>
<td>Arnaud</td>
</tr>
<tr>
<td>20 Web application (visual)</td>
<td>40 days</td>
<td>Sun 17/05/15</td>
<td>Thu 25/06/15</td>
<td>5,6,8,11</td>
<td>Pauline</td>
</tr>
<tr>
<td>21 Final Applications</td>
<td>0 days</td>
<td>Wed 05/08/15</td>
<td>Wed 05/08/15</td>
<td>16,17,18,19,20</td>
<td>Pauline, Arnaud</td>
</tr>
<tr>
<td>22 Tests</td>
<td>80 days</td>
<td>Sun 17/05/15</td>
<td>Wed 05/08/15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23 Writing test cases</td>
<td>30 days</td>
<td>Sun 17/05/15</td>
<td>Mon 15/06/15</td>
<td>5,6,11</td>
<td>Arnaud, Pauline</td>
</tr>
<tr>
<td>24 Unit tests</td>
<td>50 days</td>
<td>Tue 16/06/15</td>
<td>Tue 04/08/15</td>
<td>23</td>
<td>Arnaud, Pauline</td>
</tr>
<tr>
<td>25 End tests</td>
<td>50 days</td>
<td>Tue 16/06/15</td>
<td>Tue 04/08/15</td>
<td>23</td>
<td>Arnaud, Pauline</td>
</tr>
<tr>
<td>26 Application tested</td>
<td>0 days</td>
<td>Wed 05/08/15</td>
<td>Wed 05/08/15</td>
<td>23,24,25</td>
<td>Pauline, Arnaud</td>
</tr>
<tr>
<td>27 Production</td>
<td>13 days</td>
<td>Wed 05/08/15</td>
<td>Tue 18/08/15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28 Deployment</td>
<td>3 days</td>
<td>Wed 05/08/15</td>
<td>Fri 07/08/15</td>
<td>16,17,18,19,20</td>
<td>Arnaud</td>
</tr>
<tr>
<td>29 Promotion</td>
<td>5 days</td>
<td>Wed 05/08/15</td>
<td>Sun 09/08/15</td>
<td>16,17,18</td>
<td>Arnaud, Pauline</td>
</tr>
<tr>
<td>30 Feedback analysis</td>
<td>5 days</td>
<td>Sat 08/08/15</td>
<td>Wed 12/08/15</td>
<td>28</td>
<td>Pauline</td>
</tr>
<tr>
<td>31 Correct defects</td>
<td>5 days</td>
<td>Thu 13/08/15</td>
<td>Mon 17/08/15</td>
<td>30</td>
<td>Arnaud</td>
</tr>
<tr>
<td>32 Evolution</td>
<td>5 days</td>
<td>Thu 13/08/15</td>
<td>Mon 17/08/15</td>
<td>30</td>
<td>Arnaud</td>
</tr>
<tr>
<td>33 End of project</td>
<td>0 days</td>
<td>Tue 18/08/15</td>
<td>Tue 18/08/15</td>
<td>28,29,30,31,32</td>
<td>Pauline, Arnaud</td>
</tr>
</tbody>
</table>

Figure 45: Master Schedule

Legend:

- Milestone
- Task
Thereafter, we can see the Gantt Chart to illustrate our planning from the 27 of January to the 18 of August:

**GANTT Diagram**

![Gantt Chart Image]

*Figure 46: Gantt Chart*
Moreover, we will look specifically at each of our Milestone of our Gantt Chart:

### Project Task Analysis

<table>
<thead>
<tr>
<th>Title</th>
<th>Duration</th>
<th>Objectives</th>
<th>Description</th>
<th>Deliverables</th>
</tr>
</thead>
</table>
| Pre-research & Analysis | 57 days  | - Define the subject and objectives  
- Establish the working context  
- Extend knowledge on technologies relatives to project  
- Make a project plan  
- Research about ethical and legal issues | These pre-researches will be analysed so that we can have some solid foundations. It should prepare us to the project and make us ready to go on the specified date. | **MSc Research Report**, Word/PDF: Report retracing all researches and analysis done in the Research Methods and Project Planning class. |
| Research & Analysis  | 90 days  | - Describe functional specifications of the applications  
- Write technical specifications out of functional specs in front of implementation.  
- Make survey and analyse them to improve functional specifications. | This part concerns all researches relative to the implantation of the application. It concerns features and functionality to propose to user, how to implement them in the current application, design a solution... | **Functional specifications**, Word/PDF: Report regrouping our functionalities of the applications, how they will be represented and how the user will interact with the app.  
**Technical specifications**, Word/PDF: Report regrouping how the different functionalities have been implemented technically in the application. |
| Implementation       | 80 days  | - Create a back-end server  
- Create a web application for website front-end  
- Create a mobile application communicating with back-end  
- Website and mobile app should share the same content. | This part concerns the implementation of the application. We will mostly have to read the technical specifications written previously and adapt the code to what has been redacted. Therefore each feature should be done after its technical specifications and should be done ahead of unit tests. In other terms, this part is mainly concentrated on coding. | **Back-end server**, Application: A back-end server providing access to centralised data through services will be implemented.  
**Mobile Application**, Application: A mobile application able to retrieve information from back end server and being available on phone will be implemented.  
**Website/Web App**, Application: An internet website will be available to provide another way of accessing the data than only mobile application. |
| Tests                | 80 days  | - Write and process unit testing  
- Write and process end-tests | This part concerns testing. Unit testing concern code testing to ensure there are no errors in the code that has been written, once they are written they can be processed every time an amendment has been made to the code so it is very important for the maintainability. End-tests are the “functional” tests; they ensure the behaviour of the applications is the one that is expected by the end users. | **Tests scenarios**, Word/PDF: Small report regrouping everything a user can be able to tests on an application and that should be working in compliance with user’s expectations. |
| Production | 13+ days | - Launch the application in production  
- Get some feedback  
- Correct major defects that were not seen during implantation and test  
- Carry out some evolutions that user requires | We’ll try to release the application so everybody can access it.  
We’ll try to get some feedback from the users that can tell us when the application doesn’t work and the eventual things they would prefer to be implemented in the future.  
This duration is set to 13 days but actually if we pursue this project this task will go on for longer and we will probably have to make a new planning.  
Dissertation Report. Word/PDF: Report due for Master Project and Dissertation class. It will retrace all the work done during this project as well as a critical analysis on what was expected and the events that will happen meanwhile. |

7.2 Management overview

First of all, we decided to use the Agile method as management tool in order to be more professional regarding this graduation project. So, we chose to work every day together (with our co-worker on this project) in order to do a checkpoint every morning (just before we started to work) and every evening (just after we finished to work for the day). During these daily morning meetings, we tried to answer 3 questions:

- What did we do yesterday?
- What will we do today?
- Do we see any risk or issue that prevents us to reach our goals?

We had also the possibility during these meetings to talk about any special point or new ideas about the general application development.

Furthermore, regarding the implementation of our application, we decided to use Git, a distributed revision control system. It allows us to work together in using the same files. In this way it increased the speed and the quality of our project. Git was really useful in the management of the development of our cooking application.

Moreover, we take our supervisor as our direct client. In this case, we had a planned meeting with him every Tuesday morning, in order to show him the progress of the development and also in order that he can tell us, at any moment, if he wanted something special to change. He had also the possibility to ask for extra meetings in the week if he felt the need, and we would have been at his disposal.

7.3 Realistic project plan at the end

Finally, as a general point, we can say that, at the beginning of this project, we were too ambitious. Actually, we didn’t have enough time to realise all the tests of our application at the end, neither the whole Production jalon with the deployment of our application on the Android play store and its promotion. The real project plan that actually happened can be synthetized like this:

- Pre-research and Research have been the same.
- Implementation jalon has been also approximately the same. We are just still implementing our application every day, because we want to continue improving it even after the end of this year.
- For the Tests jalon, we actually made only the part about the unit tests of our application, and we didn’t do all the unit tests necessary to say that our application is full tested.
- So, we decided to keep the production for a near future, after choosing the moment when we can say that our application is at a sufficient professional level to be deployed on the Android play store.
This entire project management part was really helpful and useful as a personal and professional point of view. With the help of this retrospective, we can say that we will be careful in the future of what we are really able to do within a specified time before making the same mistake.

8 PROFESSIONAL, LEGAL, ETHICAL AND SOCIAL ISSUES

We will in this section talk about the professional, legal, ethical and social issues for our project. All these issues have to be taken seriously and undergone with professionalism. We will have a professional attitude toward the application of computer technology. Moreover, we will see below the different points to take into account concerning all of these issues.

No plagiarism

We made a statement of non-plagiarism in the beginning of this report, which means that we will reference all sources used for this dissertation as well as for the application code in order to avoid plagiarism issues.

Data protection law

Nowadays, « the protection of Intellectual Property Rights (IPR) from attack by cybercriminals is, for many modern-day business, as important as dealing with crack attacks on computer networks. » (Schell and Bernadette, 2007)

Moreover, our application will access and retain some private and personal user’s data. So we have to secure them and be standards regarding the recent laws about data protection:

As Nick Taylor (2015) says about data protection law:

« 1. The subject of personal data has the right to view and correct that data
2. Personal data should be accurate, adequate, relevant and kept up to date
3. Personal data should not be kept for longer than is necessary
4. Appropriate technical and organisational measures should be taken against unauthorised or unlawful processing of personal data and against accidental loss or destruction of personal data
»

So we will apply a maximum these points in order to be standards about data protection laws on the data of our users.

Copyrights

First of all, according to Nick Taylor (2015), copyright « protects original works, sound recordings, typographical layouts. »

But « access to copyrighted materials and resources for research purposes have been increasingly conducted across borders … [and therefore] leads to different countries adapting different laws pertaining to it. » (Ratnaria and Khadijah, 2014)

So it means that we have to be careful with copyrights because depending on the country where we develop and deploy our application, it’s maybe not the same laws about copyright that applying.

And moreover « could cyberspace be considered a zone of liberty, for the most part unrestrained by government regulation and intrusion into individual rights and freedoms? On the one hand, the Internet has served to expand the boundaries of information sharing to an unprecedented degree. However, as the dividing line between the online and ‘real’ world is becoming ever vaguer, initiatives affecting individuals’ activities in cyber-space are intensified as well. [So] … the protection of intellectual property rights (IPR), … copyright, in cyberspace [has becoming a main goal in professional, legal, ethical and social issues.] » (Ievdokymova, 2013)
Ethical aspect of data storage for the survey

During all our researches about this project and particularly concerning the data storage for the survey, « one major obligation on the part of researchers is to the subjects who are involved in research. Social researchers must strive to protect subjects from undue harm arising as a consequence of their participation in research. This requires that subjects’ participation should be voluntary, and as fully informed as possible. At the same time, no group should be disadvantaged by routinely being excluded from consideration. Subjects should also be aware of their entitlement to refuse to participate at any stage for whatever reason, and to withdraw data just supplied. 

a) For interviews / focus groups: All participants must be fully informed of the nature of the research and give informed consent prior to interview. Participants must be given a plain language statement of the nature and purpose of the research. It is generally preferable not to identify individual participants but, if the identification of participants is necessary, participants must be informed of this, and of safeguards to ensure that this information is restricted to the researcher or a specific research group. No interview should be recorded without the permission of the participant. Written parental consent is required for interviews with participants under age 18 (16 in Scotland), unless such interviews take place in the presence of a parent or guardian or in an institutional setting where the institutional consent has been given.

b) Questionnaires: All written questionnaires must have an opening statement informing the participant of the nature and purpose of the research. If a questionnaire contains any questions likely to cause offence to the respondent, this should be clearly indicated on the front cover, so that the participant may decide not to read on.

And also, all data must be anonymised so that no subject can be identified. » (Heriot Watt University, 2015)

So we will apply a maximum these points in order to be standards and professional about ethical aspect of data storage during our researches regarding all users and participants in our application.

Professional attitude

Despite the fact that we want a fun and interactive application with games, quizzes and enjoyable tutorials, we will ensure the quality of our application in a professional manner. Therefore, we will respect all these issues professionally. Moreover, we will test all our code before going to production because we don’t want to send an application to our users that doesn’t work very well.

9 CONCLUSIONS AND FUTURE WORK

9.1 Conclusion

Our goal was to create a cooking application that will take care of teaching our users how to cook. We wanted to guide them more about learning to cook than just find recipes online. And to make them more confident we’ve chosen to develop our cooking application by including step-by-step recipes in order that they are clearly explained and achievable by anyone. Moreover, with the intention to improve our users’ skills regarding the cooking field, we decided to implement in our application some quizzes and games also, to learn even further about cooking. So we will make the content of our entire cooking application available to everybody, from beginners who want to learn how to cook a specific recipe through fun and attractive tutorials, to famous chefs who want to share their tips and recipes with other people passionate by cooking.

We saw that nowadays the mobiles era is developing increasingly, which is why we decided to implement our cooking application for mobile devices first (but also for website in the same time in order to not put aside computer users).
For the first time, we used an approach exactly turned on the expectations of our target users. Which means, after a market analysis and the implementation of mock-ups and surveys, we went directly to the persons concerned, our target users, in order to meet them and ask their feelings, their expectations, their needs, their remarks, etc. on our mobile application.

Moreover, in order to implement our cooking application, we have chosen special technologies (AngularJS, Ionic, Cordova, etc.) that permit us to develop our mobile application once, but to deploy it everywhere (Android, iOS, etc.), which will be a time saver. But during this implementation and unit testing parts, we encountered some difficulty, which delayed us in the progression of this project.

Nevertheless, we decided to submit our application for user testing evaluation. Even if all the features were not implemented we wanted at least the first evaluation of our future users on the progress we made on our application compared to the early mock-ups. Thus, we have a first application properly functioning to present our master dissertation project in the allotted time (three months), with approximately a little more than half of the requirements that we wanted to implement, and the feedback from our users on this first application. Now, we can imagine an advanced future of our application but during our personal time later this year.

Finally, we also took care of all the professional, legal, ethical and social issues of our entire project; especially be careful about the private data of our users and the copyrights about recipes. We analysed the whole management of our project as well in order to not do the same mistake for future projects.

9.2 Critical evaluation of the project

In a first time, if we have to criticize the test methods used, like mock-ups and surveys, we can say that it is one of the best, especially when the project is to develop a mobile application. Conduct surveys, analyse data, mainly do user testing is the best way to be sure that the design and all the functionality required for this given type of application, a cooking application, correspond to the expectations of our target users.

Moreover, after the analysis about the management of our project, we can say that we will be more careful for future projects. Now, we have a better idea of how much times it takes to implement and develop a whole mobile application with so many features. Thus, we can make better project plan in the future.

Finally, we can say in general that, despite the fact that our researches were mainly found online because of the constant advances in technologies in the IT world, we have chosen the right frameworks and languages to develop our cooking application. Ionic framework has been especially very useful and of great help in implementing our mobile cooking application.

9.3 Future work

One regret after this stage of the project was not having enough time to make it more complex and fun, as we wanted our application to be in a first time, with a lot of features that could help more our future users and improve their cooking’s skills. Nevertheless, the current implementation of our cooking application works well.

Originally there were plans to make users find more tips and help in making recipes, and also to improve their skills by including a progression scale. A better idea that emerged later was to define how to show to the user his progression on the cooking field. The goal of this feature is to make this mobile cooking application funnier by including a “little man” moving on each screens of the mobile application. It could be a different “little man” depending on the level of cooking of the user.

We are planning to start working on this even if it turned out to be a very complex feature; so we decided to
do it in our own time later this year. Besides that it is a good topic for future development.

We can also have different point for future work about our cooking application:

For example, « in 2013, France had 65.3 million French (INSEE) whether as many real users or potential users of our products and services. Over the whole of this population 24.1 million individuals hold a smartphone (Mobile Marketing Association France), whom 44% of mobile users are downloading 7.3 million applications per month. 27.8 million is the number of French households in 2013, of which 21.4 million are now equipped with at least one computer. For a little over two years, touch pads have arrived in the market and already 5.1 million French homes are equipped with one (GfK and Médiamétrie Institute), almost one household in five. All these figures highlight the diversity of technological devices used by the French population, but also … the multiplicity of user profiles. » (Ledeline, 2014)

But it’s not just in France; it’s the same in whole the world. Which means that we can think about deploy our cooking application also on touch pad in order to touch even more population and user profile.

We can as well deploy our cooking application on the iOS Store. Which will shouldn’t be so difficult with the help of the technologies that we have chosen before, because with AngularJS we just have to « keep the functionality for our application (in controllers) separates from the way our application looks. » (Lynch, 2014), and then with Ionic we just have to « drop it in our code to get going, and push through Cordova when it's ready. » (Ionic) The main purpose of using these technologies is to develop once, deploy everywhere, and so save time and money.

Moreover, we can add some functionality to our cooking application, like for example:
- A meal planner in order that the user can plan all his meals for the week.
- Some vocal commands in order to control the app by voice, like “next step” or “set timer” when the user is doing a recipe and he can’t touch the screen with his dirty hands but he still want to go to the next step.
- A link to Asda, Tesco, etc. with the intention of ordering his shopping list.
10 REFERENCES

Angularjs, Conceptual overview, consulted 01/08/2015, [Online], available: https://docs.angularjs.org/guide/concepts

Apache Cordova, About Apache Cordova, consulted 28/03/2015, [Online], available: http://cordova.apache.org


Bavari, J., 10/10/2014, Mobile Applications – with JS & Ionic, Slidedeck.io website, [Online], available: http://slidedeck.io/jbavari/MobileApplicationsWithJsAndIonic

Bertrand, A., 2015, Research Report, Easy Cooking Application


Bojinov, V., March 2015, RESTful Web API Design with Node.js, [eBook]


Foster, M E., 30/01/2015, Design Patterns; Responsive Design, F21AD Advanced Interaction Design


H00116535, 16/08/2012, Masters Project and Dissertation in Advanced Internet Applications, A location-based orienteering game

Heriot Watt University, School of Mathematical and Computer Sciences, Research Ethical Guidelines, consulted 02/08/2015, [Online], available: http://www.macs.hw.ac.uk/~pjbk/cs4/fullEthicsApproval.html


Ionic, Ionic Framework Website, Create amazing apps, consulted 28/03/2015, [Online], available: http://ionicframework.com


Le Page, T., 07/04/2014, SQL for fun! – e-learning tool for SQL tutorials, MSc Project – Research Report


Mears, C., 22/03/2013, Personas - The Beginner’s Guide, theUXreview, [Online], available: theuxreview.co.uk/personas-the-beginners-guide/
Minar, I., 19/07/2012, Angular google+, [Online], available: https://plus.google.com/+AngularJS/posts/aZNVhj355G2

PhoneGap, About the Project, consulted 28/03/2015, [Online], available: http://phonegap.com/about/


Schell & Bernadette, H., 2007, The Internet and Society, Contemporary World Issues [eBook]

Scotter, M., 19/02/2015, Case study: integrating azure with google app engine, [Online], available: http://fr.slideshare.net/MiguelScotter/case-study-integrating-azure-with-google-app-engine

Taylor, N., 28/01/2015, Professional, Legal, Ethical and Social Issues, F21RP Research Methods


PS: Considering the subject of our project, a cooking application, the majority of our references and sources are found online because of the constant evolution of the used technologies nowadays.
11 APPENDICES

Appendices A – MyBalsamiq mockups
Appendices B – SurveyMonkey survey
Appendices C – SurveyMonkey results
Appendices D – Database design
Appendices E – User testing form
Appendices A – MyBalsamiq mockups

**DESIGN 1**

Login 1

Search recipe 1

Create recipe 1

Display recipe 1
## Appendices B – SurveyMonkey survey

### Easy Cooking Application: Cookeasy

**Personal contact information**

This will not be disclosed outside of this study and will stay anonymous.

**1. What is your gender?**
- Male
- Female

**2. In which age group are you in?**
- 10 - 14 years old
- 15 - 25 years old
- 26 - 40 years old
- 41 - 60 years old
- > 61 years old

**3. What is your activity nowadays?**
- Student
- Unemployed
- Employed

**4. In which business sector do you work?**

Now, concerning the 2 designs that we have attached to this survey and presented to you thanks to the software MyBalsamiq, you will ask to answer some questions regarding these 2 designs in the next pages. Thank you in advance to take time to answer this survey.
* 5. For a 1st impression, which design do you prefer for the login page?
- Design n°1
- Design n°2

* 6. Do you think an authentication system (username/password) is necessary for a cooking application?
- Yes
- No

7. If you answered "No" at the last question, tell us why?

* 8. To what extent do you agree or disagree with each of the following statements?

<table>
<thead>
<tr>
<th>Statement</th>
<th>Strongly agree</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
<th>Strongly disagree</th>
<th>Don't know</th>
</tr>
</thead>
<tbody>
<tr>
<td>A sign up link is important when you have accounts in the app</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>Title of the app in the logger is important</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>
Easy Cooking Application: Cookeasy

HEADER PAGE

9. For a 1st impression, which design do you prefer for the header?
- Design n°1
- Design n°2

10. To what extent do you agree or disagree with each of the following statements?

<table>
<thead>
<tr>
<th>Statement</th>
<th>Strongly agree</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
<th>Strongly disagree</th>
<th>Don't know</th>
</tr>
</thead>
<tbody>
<tr>
<td>The header should be separated from the rest of the page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>There has to be a menu button</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>There has to be a profile button</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>There has to be a log out button</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>There has to be a games menu button</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>There has to be a direct create recipe button</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>There has to be a direct search recipe button</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Easy Cooking Application: Cookeasy

SEARCH RECIPE PAGE

11. For a 1st impression, which design do you prefer for the search recipe page?
- Design n°1
- Design n°2

12. To what extent do you agree or disagree with each of the following statements?

<table>
<thead>
<tr>
<th>Statement</th>
<th>Strongly agree</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
<th>Strongly disagree</th>
<th>Don't know</th>
</tr>
</thead>
<tbody>
<tr>
<td>A search input is necessary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A random selection should appear even if there is no word in the input search</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The picture of recipes should appear in the selection list</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The name of recipes should appear in the selection list</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Easy Cooking Application: Cookeasy**

**CREATE RECIPE PAGE**

**13. For a 1st impression, which design do you prefer for the create recipe page?**

- [ ] Design n°1
- [ ] Design n°2

**14. To what extent do you agree or disagree with each of the following statements?**

<table>
<thead>
<tr>
<th>Statement</th>
<th>Strongly agree</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
<th>Strongly disagree</th>
<th>Don't know</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ask for the title of the recipe is important</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ask for the total time of the recipe is important</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ask for the difficulty of the recipe is important</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ask for the type of course of the recipe is important</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ask for the number of persons of the recipe is important</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add each ingredients with quantity and unit is important</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ask to describe each step in the 1st page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>It is better to ask them one by one in the next pages with a next button</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A picture and a timer for each step is important</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ask for the different needed utensils for the recipe in the 1st page is important</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>It is better to ask them one by one in the next pages with a next button</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add a final picture is necessary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
15. For a 1st impression, which design do you prefer for the display recipe page?

- Design n°1
- Design n°2

16. To what extent do you agree or disagree with each of the following statements?

<table>
<thead>
<tr>
<th>Statement</th>
<th>Strongly agree</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
<th>Strongly disagree</th>
<th>Don’t know</th>
</tr>
</thead>
<tbody>
<tr>
<td>The title of the recipe should appear in the 1st page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The difficulty of the recipe should appear in the 1st page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The total time of the recipe should appear in the 1st page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The number of persons of the recipe should appear in the 1st page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The type of course of the recipe should appear in the 1st page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The list of ingredients should appear in the 1st page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The list of utensils should appear in the 1st page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The number of steps for the recipe is needed in the 1st page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The entire description of each steps should appear in the 1st page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Each step’s name is enough in the 1st page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>It is better to have the description with picture and timer of each step one by one once start recipe button</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How it appears on Internet:
Appendices C – SurveyMonkey results

For a 1st impression, which design do you prefer for the login page?

- Design n°1: 5
- Design n°2: 10

Do you think an authentication system is necessary for a cooking application?

- Yes: 5
- No: 10

A sign up link is important when you have accounts in the app.

Title of the app in the login page is important.
**Easy Cooking Application – Pauline Michaud**

**HEADER PAGE**

For a 1st impression, which design do you prefer for the header?

![Bar chart showing preferences for header design]

**SEARCH RECIPE PAGE**

For a 1st impression, which design do you prefer for the search recipe page?

![Bar chart showing preferences for search recipe page design]
Appendices D – Database design

A model is represented by:

<table>
<thead>
<tr>
<th>Name of model</th>
<th>type</th>
<th>definition</th>
</tr>
</thead>
</table>

*Look to other models*

<table>
<thead>
<tr>
<th>Ingredient</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Name of the ingredient</td>
</tr>
<tr>
<td>new</td>
<td>Boolean</td>
<td>True if element hasn’t been treated yet</td>
</tr>
<tr>
<td>picture</td>
<td>Picture*</td>
<td>Picture of the ingredient</td>
</tr>
<tr>
<td>recipes</td>
<td>[ID]</td>
<td>Ids of recipe including that ingredient</td>
</tr>
<tr>
<td>type</td>
<td>Number</td>
<td>Type of ingredient (vegetable, meat…)</td>
</tr>
<tr>
<td>tip</td>
<td>String</td>
<td>Tip for choosing the ingredient</td>
</tr>
</tbody>
</table>

*Look to other models*

<table>
<thead>
<tr>
<th>Picture</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>author</td>
<td>User*</td>
<td>Author of the picture</td>
</tr>
<tr>
<td>createdOn</td>
<td>Date</td>
<td>Date of creation</td>
</tr>
<tr>
<td>format</td>
<td>String</td>
<td>Picture format</td>
</tr>
<tr>
<td>height</td>
<td>Number</td>
<td>Height of the picture</td>
</tr>
<tr>
<td>public_id</td>
<td>String</td>
<td>Cloudinary ID for the picture</td>
</tr>
<tr>
<td>tags</td>
<td>[String]</td>
<td>Tags matching the picture</td>
</tr>
<tr>
<td>width</td>
<td>Number</td>
<td>Width of the picture</td>
</tr>
</tbody>
</table>

*Look to other models*

<table>
<thead>
<tr>
<th>User</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Full name of the user</td>
</tr>
<tr>
<td>username</td>
<td>String</td>
<td>Username</td>
</tr>
<tr>
<td>picture</td>
<td>Picture</td>
<td>Profile picture</td>
</tr>
<tr>
<td>location</td>
<td>String</td>
<td>User’s location</td>
</tr>
<tr>
<td>dob</td>
<td>Date</td>
<td>Date of birth</td>
</tr>
<tr>
<td>email</td>
<td>String</td>
<td>E-mail of the user</td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td>User’s password</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>User’s description</td>
</tr>
<tr>
<td>activities</td>
<td>[Activity]</td>
<td>User’s activities on the app</td>
</tr>
</tbody>
</table>

*Look to other models*

<table>
<thead>
<tr>
<th>Activity</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>content</td>
<td>[Content]</td>
<td>Content</td>
</tr>
<tr>
<td>date</td>
<td>Date</td>
<td>Date of creation</td>
</tr>
<tr>
<td>picture</td>
<td>Picture*</td>
<td>Image related</td>
</tr>
<tr>
<td>title</td>
<td>String</td>
<td>Main message of activity</td>
</tr>
</tbody>
</table>

*Look to other models*

<table>
<thead>
<tr>
<th>Content</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>picture</td>
<td>Picture*</td>
<td>Picture of content</td>
</tr>
<tr>
<td>text</td>
<td>String</td>
<td>Text content</td>
</tr>
<tr>
<td>title</td>
<td>String</td>
<td>Content title</td>
</tr>
</tbody>
</table>
### Recipe

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Name of the recipe</td>
</tr>
<tr>
<td>author</td>
<td>User*</td>
<td>Author of the recipe</td>
</tr>
<tr>
<td>course</td>
<td>Number</td>
<td>Start / Main / Dessert</td>
</tr>
<tr>
<td>createdOn</td>
<td>Date</td>
<td>Creation date</td>
</tr>
<tr>
<td>difficulty</td>
<td>Number</td>
<td>Difficulty of the recipe (1-5)</td>
</tr>
<tr>
<td>comments</td>
<td>[Comment]</td>
<td>User comments on the recipe</td>
</tr>
<tr>
<td>ingredients</td>
<td>[Ingredient]</td>
<td>Ingredients of the recipe</td>
</tr>
<tr>
<td>nbPerson</td>
<td>Number</td>
<td>Number of person for the recipe</td>
</tr>
<tr>
<td>picture</td>
<td>Picture*</td>
<td>Main picture of the recipe</td>
</tr>
<tr>
<td>pictures</td>
<td>[Picture*]</td>
<td>Pictures imported by users</td>
</tr>
<tr>
<td>steps</td>
<td>[Step]</td>
<td>Each step of recipe</td>
</tr>
<tr>
<td>time</td>
<td>Date</td>
<td>Total time of the recipe</td>
</tr>
<tr>
<td>utensils</td>
<td>[String]</td>
<td>List of utensils</td>
</tr>
<tr>
<td>updatedOn</td>
<td>Date</td>
<td>Date of creation</td>
</tr>
</tbody>
</table>

### Comment

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>author</td>
<td>User*</td>
<td>Username of the message creator</td>
</tr>
<tr>
<td>message</td>
<td>String</td>
<td>Content of the comment</td>
</tr>
<tr>
<td>createdOn</td>
<td>Date</td>
<td>Date of the message creation</td>
</tr>
<tr>
<td>mark</td>
<td>Number</td>
<td>Mark that the user gave to the recipe</td>
</tr>
<tr>
<td>updatedOn</td>
<td>Number</td>
<td>Date of update</td>
</tr>
</tbody>
</table>

### Ingredient

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Name of the ingredient</td>
</tr>
<tr>
<td>qte</td>
<td>Number</td>
<td>Quantity of the ingredient</td>
</tr>
<tr>
<td>unit</td>
<td>String</td>
<td>Unit for the ingredient</td>
</tr>
</tbody>
</table>

### Step

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td>Number</td>
<td>Step number</td>
</tr>
<tr>
<td>action</td>
<td>String</td>
<td>Definition of what to do</td>
</tr>
<tr>
<td>picture</td>
<td>Picture*</td>
<td>Picture of the action</td>
</tr>
<tr>
<td>time</td>
<td>Date</td>
<td>Timer for this step</td>
</tr>
</tbody>
</table>

---

*Look to other models

### Game

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>author</td>
<td>User*</td>
<td>Author of the game</td>
</tr>
<tr>
<td>createdOn</td>
<td>Date</td>
<td>Date of creation</td>
</tr>
<tr>
<td>picture</td>
<td>Picture*</td>
<td>Main picture</td>
</tr>
<tr>
<td>title</td>
<td>String</td>
<td>Title of the game</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>Type of game</td>
</tr>
<tr>
<td>updatedOn</td>
<td>Date</td>
<td>Date of update</td>
</tr>
</tbody>
</table>

### Quiz

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>questions</td>
<td>[Question]</td>
<td>List of questions</td>
</tr>
</tbody>
</table>

### Question

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>answers</td>
<td>[Answer]</td>
<td>List of answers</td>
</tr>
<tr>
<td>title</td>
<td>String</td>
<td>Title</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>Type (text or image)</td>
</tr>
</tbody>
</table>

### Answer

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>String</td>
<td>Answer text</td>
</tr>
<tr>
<td>picture</td>
<td>Picture*</td>
<td>Picture as answer</td>
</tr>
<tr>
<td>correct</td>
<td>Boolean</td>
<td>Correct/Uncorrect</td>
</tr>
</tbody>
</table>
Appendices E – User testing form

USER TESTING FORM (1/3)

◆ What is your gender?
  Male   Female

◆ In which age group are you in?
  10 – 14 years old
  15 – 25 years old
  26 – 40 years old
  41 – 60 years old
  > 61 years old

◆ What is your activity nowadays?
  Student
  Unemployed
  Employed

◆ In which business sector do you work?

USER TASKS

◆ Create an account
  Difficulty
  0 1 2 3 4 5

  Notes:   Feedsbacks:

◆ Edit your profile information
  Difficulty
  0 1 2 3 4 5

  Notes:   Feedsbacks:
## USER TESTING FORM (2/3)

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Search the recipe « perfect rice salad »</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feedbacks:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>You want to start this recipe, follow it</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feedbacks:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>You want to share the fact that you liked it</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feedbacks:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add a comment to this recipe</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feedbacks:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Notes:</td>
<td>Feedbacks:</td>
<td>Difficulty</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create your own recipe</td>
<td></td>
<td></td>
<td>0 1 2 3 4 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search the quiz « Find the fish »</td>
<td></td>
<td></td>
<td>0 1 2 3 4 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Try the quiz</td>
<td></td>
<td></td>
<td>0 1 2 3 4 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create your own quiz</td>
<td></td>
<td></td>
<td>0 1 2 3 4 5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>