Teams Monitor System
Project Report

Mohammed alwanain

091535421
MSc in software Engineering
Mia22@hw.ac.uk

Project Supervisor: Monica Farrow
19 August 2009
Declaration

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

Name:

.............................................

Date:

.............................................

Signature:

.............................................
Acknowledgements:

First of all, I would thank Allah for all the blessings that He has given me. I would also like to express my gratitude to all those who helped and supported me in the completion of this dissertation. I am particularly grateful to my supervisor Ms. Monica Farrow, whose support, advice and encouragement assisted me throughout the implementation and writing of the dissertation.

I would like to thank my Mother and Father for their great support and for all the effort they have made to encourage me to achieve my ambition. Also I would like to thank my wife for the effort she has made to provide the best atmosphere for me to complete this dissertation.

I am also grateful to my manager Ibrahim Alasqah and my colleagues Fahad Aldhari and Fahad Alwashmi who provided me with valuable suggestions and helped me to get information from the Ministry of Water.
Abstract:
This dissertation represents the preparatory work for an MSc Team monitoring project. Its intention is to develop an application for the Technical Team in the maintenance department of the Ministry of Water and Electricity (MOWE) in Saudi Arabia. The primary objective of this project is to provide an application which manages and controls this Team’s tasks and progress in addition to improving the current process.

The project developed has achieved the mission appropriate to the Ministry’s needs. It has served and controlled the Ministry teams through automotive processing which gives the administrator the power and means to generate and distribute tasks to these Teams and also to display the location details in maps.

This report describes and gives details of the project, starting with background research about the project programming languages, tools and functionalities. The developing methodology was considered when listing requirements and followed at each stage of the development. In the system design and implementation, the author has illustrated the strategies used in these stages and explained how these strategies support the achievement of the author’s goals. Finally, the testing and evaluation of the project is described from the point of view of the real user and reflects their level of satisfaction.
# Table of Contents

**Chapter 1: Introduction** ........................................................................................................ 9

1.1 Aims & Objectives ........................................................................................................ 10

1.2 Dissertation outline .................................................................................................... 12

1.2.1 Chapter one: Literature review ............................................................................ 12

1.2.2 Chapter two: Project requirements .................................................................... 12

1.2.3 Chapter three: Project design ............................................................................. 12

1.2.4 Chapter four: Project implementation ............................................................... 12

1.2.5 Chapter six: Evaluations .................................................................................... 13

1.2.6 Chapter seven: Conclusion and future work ...................................................... 13

1.2.7 Chapter eight: References ................................................................................ 13

1.3 Discussion of the Professional, Ethical and legal Issues .......................................... 13

**Chapter 2: Literature Review** .......................................................................................... 14

2.1 Overview .................................................................................................................... 14

2.2 Organizational systems ............................................................................................. 14

2.3 Programming languages ......................................................................................... 17

2.3.1 PHP .................................................................................................................. 17

2.3.2 ASP .................................................................................................................. 18

2.4 Database system ....................................................................................................... 19

2.4.1 MySQL database .............................................................................................. 19

2.5 AJAX ....................................................................................................................... 19

2.7 Mobile applications .................................................................................................. 20

2.6.1 Java 2 Platform, Micro Edition (J2ME) ............................................................ 22

2.6.2 The PHP Mobile platform ............................................................................... 25

2.7 Mobile database ...................................................................................................... 26

2.8 Tracking system ....................................................................................................... 27
4.2 Human Computer Interaction (HCI) ........................................................................... 48
4.3 Design methodology ................................................................................................. 49
4.4 Design strategy ......................................................................................................... 49
4.5 Selected tools ............................................................................................................ 49
4.6 General System architecture .................................................................................... 51
  4.6.1 Three-Tier Architecture ....................................................................................... 51
4.7 Design Test ............................................................................................................... 64

Chapter 5: Implementation .............................................................................................. 65
5.1 Introduction ................................................................................................................ 65
5.2 Implementation Strategy ........................................................................................... 65
  5.2.1 Home page implementation ................................................................................. 65
  5.2.2 Create user page implementation ....................................................................... 67
  5.2.3 Tasks page implementation ................................................................................ 67
  5.2.4 Assigning tasks algorithm .................................................................................. 70
  5.2.5 Waiting list implementation ................................................................................ 72
  5.2.6 Waiting list algorithm ......................................................................................... 73
  5.2.7 AJAX Report ........................................................................................................ 76
  5.2.8 Problem Location ............................................................................................... 80
  5.2.9 Database implementation .................................................................................... 80
5.3 Technical problems and difficulties ......................................................................... 81

Chapter 6: System Testing .............................................................................................. 83
6.1 Introduction ................................................................................................................ 83
6.2 System Testing Strategy ........................................................................................... 83
6.3 Application Testing .................................................................................................... 83
6.4 Test summary ............................................................................................................. 85

Chapter 7: System Evaluation ......................................................................................... 87
7.1 Evaluation Design Rationale .................................................................................... 87
7.2 Evaluation Methodology and Plan ............................................................................ 87
7.3 Participants ........................................................................................................... 87
7.4 Questionnaire ...................................................................................................... 88
  7.4.1 General questions .......................................................................................... 88
  7.4.2 Usability ......................................................................................................... 89
  7.4.3 Learnability .................................................................................................. 89
  7.4.4 Functionality ................................................................................................. 89
7.5 Analysis of results .............................................................................................. 90
7.6 Recommendations .............................................................................................. 93

Chapter 8: Project Conclusion and Future Work .................................................. 95
  8.1 Achievements ........................................................................................................ 95
  8.2 Project management ............................................................................................. 96
  8.3 Conclusions .......................................................................................................... 96
  8.4 Recommendations for Future Work ................................................................. 97

Chapter 9: References: ............................................................................................. 98

Appendices: .................................................................................................................. 105
List of Figures:

Figure 1: Organisation project plans ................................................................. 15
Figure 2: Web programming performance .......................................................... 18
Figure 3: AJAX mechanism .................................................................................. 20
Figure 4: Mobile interface structure ..................................................................... 21
Figure 5: Java 2 ME program screen ...................................................................... 24
Figure 6: Mobile database structure ..................................................................... 27
Figure 7: tracking system ...................................................................................... 29
Figure 8: GPS ........................................................................................................ 31
Figure 9: RUP phases .......................................................................................... 36
Figure 10: Use-case diagram ............................................................................... 40
Figure 11: General architecture .......................................................................... 51
Figure 12: three-tier architecture ......................................................................... 52
Figure 13: Interface design .................................................................................. 53
Figure 14: Administrator web diagram ................................................................. 54
Figure 15: Team web diagram ............................................................................. 55
Figure 16: Login interface .................................................................................... 57
Figure 17: main page ........................................................................................... 58
Figure 18: System user page ................................................................................ 58
Figure 19: Create task page ................................................................................ 59
Figure 20: Activity diagram ................................................................................ 59
Figure 21: Team management page ...................................................................... 60
Figure 22: Location ............................................................................................... 61
Figure 23: Report page ........................................................................................ 61
Figure 24: Request edit ....................................................................................... 62
Figure 25: ERD diagram ..................................................................................... 63
Figure 26: Team main page ................................................................................ 66
Figure 27: Administrator main page ..................................................................... 66
Figure 28: Create new user page ......................................................................... 67
Figure 29: Create task window ........................................................................... 68
Figure 30: Create task ....................................................................................... 68
Figure 31: Waiting list diagram .......................................................................... 73
Figure 32: Report page diagram .......................................................................... 76
Figure 33: report page result ............................................................................... 79
Figure 34: Location page ..................................................................................... 80
Figure 35:Participant Chart ................................................................................. 88
Figure 36: User background ............................................................................... 90
Figure 37: application difficulty .......................................................................... 91
Figure 38: System design .................................................................................... 91
Figure 39: System concept ................................................................................. 92
Figure 40: System requirements ......................................................................... 93
Figure 41: System errors .................................................................................... 93
Chapter 1: Introduction

Over the last few years, information system software (IS) has been used more and more frequently in government organisations and businesses. In fact, these systems are used today in many aspects of daily life, such as environmental science, the health services, finance and business, criminology, electricity, motoring and in the tourist sector.

Due to this increasing use of IS, such applications are commonly accepted, not only in their long-established role within accounting or finance, but now also in several other domains, such as management and the control of organizations and businesses.

In addition, these systems allow system managers, as mentioned above, to monitor work progress in real time and guarantee a system built within a secure structure, with security rules applied.

These applications also help administrators and top level management to extract reports by effortlessly clicking a button, in contrast to traditional reports that need specific employees to collect particular information and filter it in order to produce the target report.

This research will discuss using IS at the MOWE, to meet urgent needs in terms of monitoring and controlling the technical teams. This will be achieved using the latest programming technologies to reap the benefits of IS in reducing the amount of effort and money that is wasted.

Finally, it will consider the researcher’s point of view in terms of the application methodology and how this methodology aids developers in creating the desired application by following its rules.

1.1 Aims & Objectives:
The aim of this project is to develop software to serve the maintenance department of MOWE in Saudi Arabia, in order to improve and organise the technical group’s progress there and introduce greater flexibility into the organisation. The technical group’s mission is to fix and maintain the water network in Saudi Arabia, e.g. its broken pipes, and maintain water and sewage stations.
The idea for this project emerged when the service and water network in several cities became very complex. Due to this, the Ministry of Water (MOW) received many complaints from citizens about the water services because there was no system in place that could control technical groups. At present, the system of managing these teams operates via phone requests from the operator and written request details. The details are then passed directly to the technical group. This process has led to lots of mistakes in terms of receiving wrong information or losing request tickets before they can reach the team. This means that the teams are out of control most of the time. For these reasons, an urgent need arose to develop a system. This system had to include several important features related to field tasks, such as the authorization and means for the department administrator to generate tasks and monitor team progress (as opposed to the operator fulfilling this role, as is currently the case). The administrator would then organize the teams and ensure their progress was under control. Teams would also be tracked in the main system and the location of each task identified. In addition, tasks could be delegated via a mobile application. These features are significant because they can help to simplify the complexity of the task management of different groups and this would allow the system to balance tasks throughout the team and allocate them fairly. It would also allow the administrators to make the right decisions based on the progress of the task and the results of the system report.

The objectives:
The objective of the project is to:

1. Investigate and compare the current task schedule and future system process.
2. Search for and compare various platforms and programming languages that could be used to implement the monitoring application.
3. Study related websites and focus on the useful features they offer, to gain ideas that can be used in this project.
4. Study system features and sort them to be implemented as priority.
5. Implement the administrator’s features to allow them to generate and schedule tasks for teams and also monitor their progress.
6. Implement the team features to allow them to receive and update their tasks, while also displaying their location.
1.2 Dissertation outline:
This section gives an overview about the structure and layout of the dissertation project.

1.2.1 Chapter one: Literature review:
This chapter describes in detail most aspects of change that the computer systems bring about in business and the control of organisations. This section also reviews the different tools and programming languages designed and implemented in the team monitor and mobile features related to this application. The Literature Review presents background information on the GPS and tracking systems that the project designates on to be used as functionalities. Finally, the section considers the application’s security issues and the secure structure that the developer has taken into account while building the system.

1.2.2 Chapter two: Project requirements:
This section lists all the requirements of the proposed systems. It gives an overview of the project methodology and a brief description of the methodology rules and phases. Finally, it will give an overview of the functional and non-functional requirements, including the user case diagram and a scenario of the system functionalities.

1.2.3 Chapter three: Project design:
This section gives an outline of the system structure and user interface. It also includes a detailed description of each component, such as databases, system interfaces, colours, etc. In addition, there is a web diagram describing the structure of the system and how the pages relate and connect together. Moreover, there is an activity diagram of the main system functionality.

1.2.4 Chapter four: Project implementation:
This section gives an outline of the implementation of the system and database. It includes a description of features implementation and the algorithms that are used to implement these features. Finally, it will include an application code to illustrate the idea of feature implementation.
1.2.4 Chapter five: System Testing:
The system is demonstrated to actual users for the purpose of testing. This section will illustrate all testing functions written to test the critical functionality of the system and the results of this testing.

1.2.5 Chapter six: Evaluations:
This section outlines the user evaluation of the system and the evaluation strategy. It also includes specific methods that were used to evaluate and test the usability, learnability, and flexibility of the system design. It lists the main aspects of the evaluation questions put to users. Finally, it analyses the users’ responses to the questionnaires.

1.2.6 Chapter seven: Conclusion and future work:
This section gives a summary of all the system results obtained, along with the failures and difficulties that occurred while working on the project. In addition, there are suggestions for future work related to the project topic that could not be added to it due to time and other constraints.

1.2.7 Chapter eight: References:
This section includes a list of all references used to gather information for the project.

1.3 Discussion of the Professional, Ethical and legal Issues:
As mentioned in the introduction, the project is based on MOW requirements. The MOW in Saudi Arabia has full permission to use this project and several issues will be addressed. However, it does not cover any ethical issues. The social issue is that the system will make work easier for the whole maintenance department. There is no security data in the project, only virtual data to simulate the real field. Employees at MOW and the IT department will help to evaluate this project. However, no personal data such as names or email addresses will be demanded in the questionnaires.
Chapter 2: Literature Review

2.1 Overview:
This chapter will investigate previous research which considers the researcher’s point of view in terms of using IS in organizations, as well as the advantages and disadvantages of system features, such as the use of AJAX in the system report, integration with mobile GPS and tracking systems and how these features have become necessary in recent times.

2.2 Organizational systems:
During the last twenty years, organisational systems have grown in significance in terms of planning and implementation. Today, many organisational budgets and future planning depend on statistics that come from an organisation’s application systems. Hartono et al. (2003) point out that strategic information system planning is the process of identifying a collection of computer applications that can help organisations to implement their business plans and achieve their business aims.

Hartono et al. go on to explain that, when organisations and businesses become more complex, there will be a need for system applications to organise them. According to Klein and Dologite (2000), business environments are becoming increasingly complex. Therefore it is essential for employees to share knowledge in an organisation. Today, computer software applications are available to help employees complete their tasks. In particular, organisations can currently use the Group Support System software, which is designed to encourage collaboration between members of a group. Using these systems could potentially reduce the number of problems that are traditionally related to group interaction.

For the above reasons, MOW has started to consider the possibility of using systems to help them with planning and monitoring their progress. In fact, strategic information systems planning can help organisations to develop priorities for the development of information systems by sorting the systems in terms of their efficiency. This helps the organisation recognise its collection of planned computer applications, which both coordinate well with corporate strategy and can create an advantage over competitors (Doherty et al., 1999). However, there is a problem as the developers of an information system strategy often have different ideas, plans and
expectations about it. Often, only a few of the systems in the strategy are implemented and some of them take considerably longer than predicted to build (Hackney & McBride, 2002). Actually, to achieve the mission targets of using the organisation system, it is important to improve the knowledge and computer skills of the employees with regard to how they should deal with these applications. This would also make the employees more likely to trust the systems. According to Earl et al. (1995), as organisations begin to understand that IT could improve their employees’ knowledge and skills, computer systems begin to emphasise business process reengineering.

In fact, the main idea for the technical group’s project is to help different departments at MOWE to make decisions and improve their planning. Clarifying this particular idea, Averweg & Erwin (2000) report that information systems at strategic decision-making level are usually classified as being ‘executive’. The effect of these decisions, by definition, will be significant. Consequently, information systems for senior managers need more careful planning, both in terms of their features and their applicability to organisational situations. Executive Information Systems

![Figure 1: Organisation project plans (Rebolj et al, 2008).](image-url)
applications support executive information needs and decision-making activities (Gillan & McPherson, 1993). However, it is significant to note that an effective way to evaluate the success of an Executive Information System is to gather opinions and feedback from the executive users (Monash University, 1996).

However, the organisational applications have not just influenced top level management. According to Finnegan et al. (2003), with the increasing prevalence of Internet applications and inter-organisational systems, decisions are not limited to the organisations’ own adaptation processes; they are also influenced by events within other organisations. This also applies to MOWE because they sometimes share tasks with other companies, such as electrical and communication companies.

Moreover, the application needs to meet certain requirements in order to be successful. Salmeron (2001) notes that information systems comprise the information delivery technology for all business end-users. The normal users or data providers include employees from functional areas such as accounting managers. It is clear that information systems need continuous input from three different groups, i.e. information systems executives, users and the people responsible for developing and maintaining the information systems. This point puts companies or organisations under pressure to sign a contract with development companies if they do not have their own IT professionals or developers, thus costing these company high budgets if they do not focus on this issue from the beginning.

Finally, these systems should encourage employees to improve their work so they can complete their tasks faster and more effectively than before. Modern enterprises seek creative men and women to generate inventive ideas to improve an organisation’s competitive position. This is especially true in the field of information technology, where the focus has transferred from improvements in traditional processes (paper processes) to the promotion of creative changes so that the spirit of competition can be encouraged (Davenport, 1993). In other words, in the field of information technology, it is important to seek both improvements and inventive ways to develop the business. Improvements make an organisation’s business processes faster and better integrated (McLean & Smits, 1993).
This section has illustrated that the importance of organisational systems lies in improving an organisation’s business processes. They also support top level management so they can plan to the most effective degree possible, monitoring the real business or progress of the task. For this reason, MOW have needed to develop a planning application so that the amount of effort required of employees could be reduced and duplications avoided, therefore achieving the best possible planning and ensuring the organisation and management of target teams.

2.3 Programming languages:

Nowadays, several tools are used to develop web applications. Some of these tools are based on general domain programming languages, whilst others are based on special system environments and tools (Malkov, 2010). Regarding this system’s requirements, the developer takes into account the advantages and disadvantages of these languages in order to make comparisons and decide on the best choice for building the desired system, while taking into account the mechanism for transferring data between the user interface and server.

The following points are described as the most common web-based languages:

2.3.1 PHP:

The PHP Hypertext Preprocessor is the extensively used server side programming language, and the implementation of dynamic web applications is one of its significant features. PHP syntax is similar to other language syntax and structure, such as C (Stratton, 2007). One of the significant advantages of using PHP is that it is an open source platform and the developers do not have to have a license to use it. Other features of using PHP are its ability to connect with several different databases like Oracle and MySQL. However, the MySQL database is recommended as they work with the same platform. The PHP has the flexibility to work under different operating systems and via any internet browser without a change of function (Converse et al., 2004). In addition to the features listed above, PHP’s performance outshines that of ASP and JSP. According to Dehinbo, (2005), after carrying out a simple experiment to establish the average time taken for database retrieval per number of records, it is clear that for a minimum of 10,000
records retrieved, PHP has the best performance, followed by ASP. JSP, however, demonstrates the worst performance in this activity, as the experiment result table below illustrates.

<table>
<thead>
<tr>
<th>Number of records / Time</th>
<th>JSP</th>
<th>ASP</th>
<th>PHP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time in seconds (10000 records)</td>
<td>3.0</td>
<td>1.15</td>
<td>0.66</td>
</tr>
<tr>
<td>Time in seconds (20000 records)</td>
<td>6.0</td>
<td>2.31</td>
<td>2.00</td>
</tr>
<tr>
<td>Time in seconds (40000 records)</td>
<td>15.0</td>
<td>4.69</td>
<td>5.0</td>
</tr>
<tr>
<td>Time in seconds (80000 records)</td>
<td></td>
<td></td>
<td>10.5</td>
</tr>
</tbody>
</table>

*Figure 2: Web programming performance (Dehinbo, J., 2005).*

These results agreed with those of Cecchet et al. (2003), who claim that PHP performs better than other web-developing languages due to the fact that ‘it executes as a module in the Web server, sharing the same process (address space), thereby minimizing communication overhead between the Web server and the scripts’. This feature is unlikely with ASP.

On the other hand, PHP also has disadvantages e.g. with regard to support. As an open source language, there are no particular support centres to solve any problems that the developers face. This is not the case with other languages. In addition, in terms of large applications, it is not easy to apply with its complex features so substantial experience is required to deal with it.

### 2.3.2 ASP:

ASP (Active Server Pages) is a server side language produced by Microsoft systems. The ASP language is followed by Microsoft programming language standards and structures. It has several advantages, such as great tools support, a drag and drop feature, support centre and easy programming (startVBDotNet).

On the other hand, ASP has disadvantages that encourage developers to choose PHP, especially students. The first one is the user license. ASP is closed source. Therefore, the platform requires license to develop through. It is also essential to work under the Microsoft Windows Server operation and it does not work under any other operating system.
2.4 Database system:

Normally in applications that have three-tier architecture, the database is highly significant as it gives the system speed and power when the information is already stored in it. A database system can be defined as the basis of the system which can allow the user to store a managed system database (Elmasri & Navathe, 2004). The database structure is divided into data tables. These tables are further divided into rows, and each row contains a numbers of columns according to the data type. These tables have a relationship with each other according to the system business and data flow. In fact, there are many types of databases that can be used by developers in the applications, and each one has advantages and disadvantages but the developer concentrates on the open source database which will be discussed in the next section.

2.4.1 MySQL database:

One of the most popular kinds of database is MySQL. It became an open source database in November 1996 and at the time, still worked only under Linux operating systems (Dennis, G., 2001). Nowadays, MySQL can be managed and operated with several operating systems, such as Microsoft Windows, Linux and UNIX. This has made MySQL flexible and more common. There are many advantages that come with the MySQL database. Firstly, MySQL is an open source and does not require a license. It can also process huge quantities of data in a short time and it is powerful enough to support multiple users with different requirements (Dennis, G., 2001). Today, there are some packages which contain the MYSQL and PHP platform. These packages also have useful tools with which to install the database and manage it simply, for instance: XAMPP and the AppServ application.

2.5 AJAX:

This project is web-based, so the author concentrates on the most useful and latest development technologies that will be practical for users and make the system more user-friendly. One of these technologies is AJAX. (Neill, M., 2007) defines AJAX as Asynchronous JavaScript and XML, which is a technology-enabled website, more like a local application in terms of fast responses and time delay. This technology is aimed at retrieving data from the server without
repetitively refreshing and rebuilding the page (Albert, T., 2008). A good example of Ajax use is found in the Google website. Google search engine users only need to type the first word of a sentence that they are looking for and the search engine will display suggestion information that aids the user to find the required information without reloading the page. To illustrate the AJAX mechanism, it is the application making a server request with an XMLHttpRequest request type object. This object is then validated and sent to the server. ‘The difference from the traditional webpage is that in Ajax the HTTP page can asynchronously make calls to the server from which it was loaded and fetched that may be formatted as xml documents, HTML content, plain text or as JSON’ (Murray, G., 2005). Figure 3 illustrates how AJAX works.

![Figure 3: AJAX mechanism (Albert et al., 2008).](image)

### 2.7 Mobile applications:

When the technical group is assigned its tasks and goes into the field, there is nothing to help them in terms of reporting to the leader of the group about their problems with their progress or ordering parts except for traditional technical communication devices such as walkie-talkies. This is where the idea to use a mobile application in the project stemmed from (for example a mobile phone or GPS). In fact, today, mobile applications are used in many aspects of our lives. Bricon-Souf & Newman (2007) suggest that mobility is a common feature of different kinds of applications, which encompass diverse application domains. An example of ‘standard’ (i.e. initially context-agnostic) applications that can be upgraded to mobile applications is health care systems.
However, it is not very easy to convert these applications into computer applications. According to Jarvenpaa et al. (2003), mobile applications are more complex than other types of information technology. They not only include the hardware (mobile device) and software (mobile application), but also the communication networks that make most of the mobile services possible.

Furthermore, mobile applications are very performance sensitive and communication intensive. For example, browsing the Internet requires a short delay in communication, as multiple http requests may be sent to recover a single page. What is more, they need to have enough bandwidth to download audio files, images and text (La Porta et al., 1998). Thus, many organisations avoid thinking about using mobile applications due to the complexity and high maintenance cost of these applications.

The developers should also focus on the device environment. This means developing applications so they can deal with small devices and small screens. However, in terms of a mobile environment, there are two main aspects that need to be considered. Firstly, the technical environment of mobile phones, such as the screen and keypad design, might pose physical
problems in terms of content delivery and interactivity, when evaluated within computer environments. Secondly, it is relevant to consider how people use their mobiles to interact with applications, as with the delivery of data to mobile phones it is possible to change the way developing companies provide their customers with new products and services (Lu & Cheng, 2004). Anckar et al. (2005) also point out that, in order to make sure that the mobile application is implemented successfully, it is necessary to clearly understand user behaviour in a mobile environment.

It is important for these programs to be appropriately designed, simply because the people using these applications will be from different educational backgrounds. Benbunan-Fich & Benbunan (2007) support this idea and state that if the developers have designed an application well, professional users will exhibit a higher degree of automaticity than normal users, in spite of the fact that both kinds of user will be interested in the mobile application the first time they experience it.

Today, mobile phones are not only communication devices, they have become part of people’s everyday lives. It is essential to be careful when developing applications and dealing with mobile phones because these devices contain many electronic parts for communication, in addition to numerous applications. Furthermore, in terms of designing mobile phone applications, it is important not to make the application too complicated because the application will be implemented by a variety of different users and this will determine whether the application is successful or not. Nowadays, several tools have appeared to help developers formulate mobile applications. Two popular applications currently in use are illustrated below.

2.6.1 Java 2 Platform, Micro Edition (J2ME):
This section will illustrate the Java 2 Platform, Micro Edition which is a mobile programming language used in most mobile devices. Nowadays, the Java 2 Platform, Micro Edition is one of the most common mobile programming languages. It is suitable for use with most mobile operating systems. Economou et al. (2008), states that the Java 2 Platform, Micro Edition, developed by Sun Microsystems, is a Java-based framework for developing applications that are executed on mobile devices. Today, the Java 2 Platform, Micro Edition has achieved a
remarkable penetration and is now supported by different kinds of mobile device. However, it is
difficult to learn how to develop 2 Micro Edition applications, as they are more complex in
comparison with other developer-friendly tools.

Furthermore, Sun Microsystems have recognised the need to develop Java in new areas of
programming servers that support small devices. Later, Java developed a new generation of
programming languages which were suitable for small devices such as mobile phones. This
language has been named Java 2 Micro Edition (J2ME) (Knyziak and Winiecki, 2005).

However, Java 2, Micro Edition is not a unique language for the development of mobile
applications. There are several languages that developers can use to produce their applications,
such as the Symbian platform and the .NET Compact Framework. Janecek & Hlavacs (2005)
point out that there are two famous programming languages which are used to develop pocket
PCs. The first is Java 2, Micro Edition (J2ME) which is from Sun and the second is the .NET
Compact Framework (.NET CF), which is from Microsoft. The main advantage of Java is its
portability, although due to the variability of mobile platforms, the claim ‘write once, run
everywhere’ is not 100% true. .NET CF is suitable for pocket PCs and it integrates better with
Windows, Microsoft’s operating system as well as the new .NET paradigm.
In fact, J2ME has several advantages over other languages. These advantages have encouraged me to research this language further, which has led to my decision to use it in the development of a mobile application. Knyziak & Winiecki (2005) argue that, 'when applied in a distributed measurement system, the J2ME technology combines the power of WAP systems with the primary advantage of SMS. J2ME applications are capable of making HTTP connections to Internet servers and of parsing the response, which, as opposed to WAP, is no longer required to be in the WML format. Graphical data representation can be created on the client side, which [has] eliminated the resource-consuming image transmission over the wireless networks’. As J2ME applications are able to save data in a persistent memory, the application can be used even if no connection is available to display the results.

J2ME supports two main configurations. The first is the Connected Device Configuration (CDC). The second is the Connected Limited Device Configuration (CLDC). The CLDC is suitable for mobile phones and PDAs with approximately 512KB of available memory. However, CDC is suitable for devices which are smaller than a PC, but which have more than
512K of memory. 'CDC targeted devices are also shared, fixed (not mobile) and network connected information devices’ (Choi et al., 2003; White, 2001).

Actually, J2ME is not dissimilar to the main JAVA language. However, some developers are still quite concerned about using J2ME, especially when they start thinking about the complexity of mobile phones and their application. In fact, J2ME has a similar structure to Java and uses some of the same rules. According to Economou et al. (2008), J2ME has inherited the major rules of Java, for example Java’s ability to build up strong applications. Thus, ‘developers are not restricted by the limitations of an authoring tool’s functionality and may implement fullY-fledged innovative applications that either execute standalone or communicate with their peers or service providers, taking advantage of the J2ME’s strong wireless networking support’ (Economou et al., 2008).

From this report, we have discovered that there are several languages that programmers can use to develop their applications. However, according to the previously mentioned advantages and evaluation, J2ME is the most suitable.

2.6.2 The PHP Mobile platform:

The PHP Mobile platform is similar to the PHP web application in terms of syntax and programming structure. There is no essential difference from other platforms such as J2ME, although J2ME has a special platform and syntax which is quite different comparing with JAVA as mentioned in the J2ME section. However, in PHP and web-based applications, the difference is concentrated into two points: page reformatting and page scaling (Blekas et al., 2006). Power Browser is the best example of page reformatting where web images and blank spaces are removed; on the other hand, the WEST browser uses a zooming tool to navigate and display the page content (Björk et al., 1999).

In terms of web scaling, small screens for mobile phones are not able to present a satisfactory view of the original website and as such they are not recommended for use with the zoom techniques. To solve this problem, the developer should configure the web resolution for a
mobile phone between 128 x 128 and 176 x 220 pixels to provide satisfactory viewing on a small screen (Blekas et al., 2006).

Finally, there are advantages to using a mobile web-base application. Firstly, the developer does not have to develop a new application for mobiles, but just changes the web scaling as mentioned and reduces the blank gaps and images in the mobile page. Secondly, databases in mobile applications are the same as in the original application, without a need for tools to be synchronised between these applications.

2.7 Mobile database:
The mobile database is an important part of the application. Actually, it is not possible for the developers to design a system without paying attention to the database because the database is the brain and the source of the data. The application will not work without it. This section will focus on mobile databases.

Recently, mobility has been pervading the communication and computing arena; we envisage application design that will gradually rely increasingly upon it. In fact, standard and traditional databases and information service applications will be required to integrate with mobile units (Holliday et al., 2002). Currently, the improvement of small device database systems, such as mobile phones and PDAs has become an important research area (Shepard, 2000).

In other words, it is challenging to design mobile database systems because of the limited resources on mobile devices. In order to meet this requirement, the mobile database system should (1) be designed to fit in with resource-limited devices, (2) should support simple orders such as ‘insert’, ‘delete’ and ‘update’, without being complex in comparison to their desktop counterparts, (3) it should not need a database administrator to run, and (4) it is important to ensure interoperability between mobile and large-scale databases (Ortiz, Jr., 2000).
However, there are some issues that the developers should take into account when designing mobile databases. Chan & Roddick (2003) argue that there are many issues concerning the effective use of mobile systems, both with regard to the current technology and also for the future. The most important of these issues for mobile databases are storage capacity limitations and privacy and security issues when a computer is in a mobile environment. In fact, after reading papers and Internet websites by mobile application professionals, it seems more suitable to develop web mobile applications due to the advantages mentioned in relation to the PHP mobile, which uses only one database. The developer therefore does not have to create a new database, but simply uses the existing one, thus reducing effort and time.

2.8 Tracking system:
This is one of the most significant parts of the project. The MOW is interested in using this technology for their systems, as it will reduce the amount of effort employees need to make, as well as the number of potential conflicts. The Literature Review will address the various definitions of a tracking system and the points of view of several scholars. Furthermore, the advantages of implementing this technology will also be examined.
Ernest et al. (2004) defines a tracking system as one that incessantly monitors a vehicle or group in a given space. Tracking systems can potentially be advantageous in the automation of existing transportation systems.

In other words, the main reason for tracking systems is to collect information about the current positions of moving vehicles. Actually, this is the main purpose for using this system. Nowadays, in some countries, the Ministry of Transportation uses tracking systems to organise traffic and to avoid traffic jams. In fact, there are many reasons to encourage companies and organisations to use tracking systems, for example tracking systems provide details and updates about the actual positions of all their groups and vehicles, which allows companies to manage and operate more efficiently and effectively (Jacobs et al., 1991).

Tracking systems are also useful in guiding the groups to unloaded roads, in order to reduce the amount of time wasted and so that their tasks can be completed on time. Sterzbach & Halang (1996) point out that ‘It can, for instance, respond to a rush order by redirecting the nearest unloaded vehicle to the customer. Using such systems, operators can re-route their fleets away from storms, traffic jams, and other increasingly common hazards of the road’.

In addition, a tracking system is often fitted for safety reasons. The routes of motor vehicles on emergency missions, such as dealing with water leaking from underground pipes along main roads, can be recorded and managed in real time. Any changes to the planned routes or time schedules become apparent, therefore facilitating corresponding reactions.

In terms of the system requirements, these should incorporate the following components:
1) Field Unit: A mobile or another device on-board a vehicle or technical groups constantly acquiring and sending position data.
2) Control Centre: A station centre that contains computers which receive, process, and store data from the field unit.
3) Communication Network: A mobile communication network which transmits data between the field unit and control centre.
However, the control centre needs to identify the current position of each group work or vehicle in real time. As geographical coordinates are difficult to work with, the positional information of all the groups is illustrated on a graphical display as points on a map.

The operator of a control centre can also control the positioning time between the data transmissions at any time, or even switch off the position reporting completely. Such changes are sent directly to the mobile unit to update it. The control centre system is used to configure the data processing at the mobile units. (Sterzbach & Halang, 1996).

To summarise, a tracking system, as mentioned above, is a system used to control and monitor vehicles or technical groups. It can also be used to avoid and ease traffic jams. These systems must comprise three significant components: a field unit, a control centre, and a communication network.
2.9 GPS system:

This part of the literature review will address the definitions of a GPS system and the researcher’s point of view. Furthermore, the advantages of using this technology will also be examined. Mobile GPS will be used in the project as a navigation system to help the technical group to reach the problem area.

Today, one of the most important services route planning and moving through space to a destination. This service, called navigation, ‘may be considered to be a straightforward and effortless task since it is so common, but it involves multi-level cognitive processing and thus has attracted much theoretical and practical interest from researchers in many fields’ (Montello, 2005). As a result, it is important to develop services to help people move easily between places and to give them the ability to comprehend the layout of an environment which shows large individual differences (Ishikawa & Montello, 2006).

In order to describe how users can use this service, users first need to identify their location in space, or in other words, users need to know where they are and in which direction they are facing. When this has been established, they will want to plan a convenient route by taking into consideration where a destination is located. Next they will execute the planned route to their destination. In these three stages, ‘people access stored knowledge about the surrounding space or refer to navigational aids such as maps, or do both’ (Hightower & Borriello, 2001). In the past, maps have played a major role in guiding people around in space and providing information about the places. Today, there are many kinds of navigational systems that have been developed (Hightower & Borriello, 2001).

However, an important question is how GPS devices work and how they are managed. Zhang et al. (2003) answers this question by saying, ‘(GPS) the Global Positioning System is a satellite-based navigation system made up of a network of 24 satellites placed in orbit. Their ground stations are managed by the U.S. Department of Defense’ (Zhang et al., 2003). Figure 8 illustrates how the Global Positioning System works.
In addition, the GPS system has been used in several areas, such as health, safety, locating an emergency, roads, navigation, vehicle tracking, public transport service, water services, calculating distances and real time for journeys and field environmental decision support. This has led to its rapid spread throughout all segments of society (Kennedy, 2002).

There are many advantages of using GPS for ordinary users in addition to the official authorities. In terms of ordinary users, Herrera et al. (2009) state that GPS devices can find a position and can calculate the speed and time of a vehicle so accurately that they can be used to obtain traffic information. However, in terms of the official authorities using GPS, Sanwal & Walrand (1995) highlight aspects of a traffic monitoring system based on investigated vehicle reports such as on speed or position, and conclude that they constituted a sufficient source of traffic information. Zito et al. (1995) have also investigated the use of GPS devices as one of the main sources of information for traffic monitoring. Several technological solutions to the localisation problem can be found in using mobile phones. Traditionally, the movement of cars was monitored using mobile phones (before the invention of GPS in cellular devices). This accessed signal data from communication towers to identify the location of the mobile. It was usually a difficult and costly method. Today, several researchers have considered using mobile phones for traffic monitoring (Fontaine & Smith, 2007).
However, for the project, as mentioned above, we will use mobile phone GPS. The purpose of this is to leverage the numerous services in the mobile device. According to Herrera et al. (2009), based on GPS-enabled smart phones, the production figures for mobile phones containing several features such as GPS, are growing. This approach can provide precise location data and therefore more exact traffic data such as travel times, speed and distance.

Finally, from the researcher’s point of view of GPS services and devices, as mentioned above. It can clearly be seen that the GPS system has become a significant service for both ordinary users and the official authorities. It is a useful and helpful tool and has many advantages such as facilitating the planning of itineraries and tracking vehicles, as well as accessing traffic information.

2.10 Application security:
This is a significant issue that the developer of the application is concerned about. All application projects should take security into account. Application security is not just for military or police applications, it is for any application that is used by different services to protect them from attack by hackers. According to Power (2006), it is not unusual for developers to experience many situations when they deliver a project too late. This is typically just before a system starts working and goes live, when the plans and design have already been produced, and the code has already been written. Project groups regularly come to the realisation during this time that the contract requires the provider to perform security tests, carry out risk assessments, and comply with international standards.

Unfortunately, it is uncommon for a security expert to be present from the beginning of a project to make sure that the design is produced with security controls built into the code and the project infrastructure. This is actually the difference between a successful and unsuccessful project.
However, some organisations concentrate on networking security and ignore application security because they believe networking security is sufficient to protect the entire network, but this is not true. According to Williams (2006), ‘A stunning number of organisations spend big bucks securing the network and somehow forget about the applications that have all the keys to the kingdom and full access to all of our sensitive information.’

In terms of the application infrastructure and programming, while the developers write the project code, it is important for them to have a good understanding of possible security vulnerabilities in the project and how they can be solved. This also includes being able to understand potential vulnerabilities in particular databases, coding languages, operating systems it works with, system messaging and services and protocols. The aim of this is to allow them to take these vulnerabilities into consideration when developing the project code and to prevent them from being introduced in the first place (Power, 2006).

In fact, programmers usually have background knowledge about the basics of security application roles and they think security is something that can be added later. Yet, this is a real problem when the software engineers and the security professionals come at the end of the project to evaluate and test it in different ways. As Power (2006) mentions above, the security issue should be taken into account at the beginning of the design process in order for a successful project to be built.

2.11 Summary:
As mentioned in the Literature Review, there are many points that have influenced the project’s plan and implementation, the first of which is the security issue. After reading papers about application security, this will now be taken into account during the design and implementation stage, with security controls in the design and coding. In fact, this point is the most important one with regard to the implementation of the project. The second point is how to design mobile applications and determine the most suitable programming language for this project. This actually turned out to be PHP. Yet another issue is the kind of database the project will be dealing with. Thirdly, there is the difference between the tracking system and the GPS. The tracking system will be integrated with the main system and the mobile application will use the
GPS that mobiles already contain. Finally, the main application should fulfil the primary goal of scheduling and planning, in addition to helping the different levels of management to make decisions and track and update emergency tasks.
Chapter 3: Requirements

3.1 Introduction:
In this chapter, we will investigate and analyse the project requirements and specifications which are necessary to develop the project. Project requirements can be compiled using a variety of sources; in this project they came as user demands and comments from experts. The techniques used to collate these requirements focused on interviews as a main resource, as well as prototyping. There are two main types of requirements to be considered in this report: functional and non-functional. The identification of system behaviour is related to functional requirements. This behaviour may take the form of technical details, data manipulation and processing and other specific functionality that define what a system is supposed to achieve, without considering any physical constraints. Input and output behaviour are also specified amongst the functional requirements. Non-functional requirements define how a system is supposed to be and describe the system environment and attributes. In other words, functional requirements identify what the user requires from the system. In contrast, non-functional requirements concentrate purely on the system environment rather than on system user, e.g. on speed, or learnability.

3.2 Methodology:
In successful projects, the developers use common software methodologies to support them while they attempt to achieve their goals. There are many methodologies today that are useful for developers. However, choosing one of these methodologies depends on the project area and environment. In this project, the developer has used the Rational Unified Process (RUP). In fact, the choice of RUP methodology is subject to a number of reasons, the most significant one being that RUP specialises in organisational projects, focusing on the user requirements by establishing agreements between users and developers in terms of what the system should do and its adaptability if clients wish to change their original requirements during the project’s lifecycle.
3.2.1 RUP Methodology:

RUP was created by Rational Software (now a part of IBM) in 1997 and is the most famous implementation of a Unified Process (UP). Over the years, RUP has matured so that it reflects the experience of the many users and companies that have contributed to Rational Software’s rich heritage. Its goal is to ensure that the production of high-quality software meets the needs of its end-users within a predictable schedule and budget. RUP is a comprehensive process framework that provides industry-tested practices for software and systems delivery and implementation, as well as for effective project management.

3.2.1.1 RUP Phases:

RUP divides the development process into four phases:

1. Inception.
2. Elaboration.
3. Construction.
4. Transition.

Each phase has a specific purpose.

3.2.1.1.1 Inception Phase:

The aim of this phase is to establish the case for the system. At this phase the information obtained is used to assess the contribution of the system to the business. Business context, stakeholder agreement, understanding of requirements through use-case diagrams, success factors and financial forecasting are also established.
3.2.1.1.2 Elaboration Phase:

Key risks are identified during analysis. During this stage, the problem domain analysis is formed as well as the basic architecture of the project. Use-case analysis incorporates use-case, use-case model, and scenarios.

3.2.1.1.3 Construction Phase:

The primary objective of this stage is to build the actual system. During the implementation of the design there is:

- A cumulative increase in functionality.
- Implementation stubs are fleshed out at a deeper level.
- A greater stability begins to appear.

3.2.1.1.4 Transition Phase:

The transition phase is responsible for the transition between the old system and the system being developed. Control is subsequently moved to the maintenance team. The maintenance team will then oversee software updates and its integration with existing systems.

3.3 Over view:

The Requirements section corresponds with the inception phase in RUP methodology. This phase concentrates on business information and the system environment. It also takes into account the system users or stakeholders in several ways, for instance in the area of special user requirements (types of reports, private information etc.) and computer skills.

The developer has used two methods to gather data in order to meet the target requirements. The first method consists of qualitative data collection through interviews with responsible individuals in the target field (managers, team leaders and labourers). The second method is quantitative data collection, using questioners with the same sample.
3.4 Interviews:
This step is the most significant in the Requirements section due to the need to identify which kind of system should be developed and what the customers’ needs are. This methodology was used to gather the important information needed to design and build the system. Four kinds of user were interviewed, starting with the vice general manager, maintenance manager, team leaders, and team members. All these people are related directly to this project. They were asked several questions, vital for the illustration of the following points:

- The purpose of the system.
- The problems they are facing at the present time.
- Which kind of features should the system have to avoid these problems?

To an observer and interviewer, the majority of answers to these questions clearly concerned the control and monitoring of technical groups and these groups were sometimes out of the Ministry’s control, as was mentioned in the Introduction. Many features were required in this system. However, in line with the interview data, the developer agrees to implement the list of features from amongst the fictional requirements presented, due to the time limit on this project. More features could then be continually added once the project is applied to the real world.

3.5 Questionnaire:
After analysing the interview data, the developer started to write the user requirements by analysing the interviews. The system was simulated by designing the user case diagram (Figure 10) and the system scenario was written to give the stake-holders an idea about how the system would be designed and implemented. Afterwards, the researcher wrote a questionnaire to reflect the user’s feedback about the system business and the correction of the requirements analysed. In addition to the feedback questions there are sections about the user recommendations and suggestions to improve the system (see appendices for Requirements Questionnaire).
3.6 System use-case diagram:
The use-case diagram is a significant step in the system design phase, to visualise how the system will work and the actors will deal with the system.

3.6.1 General system scenario:

- The administrator logs into the system after he has received a request from a citizen or another source.
- The administrator generates the task to the teams by completing the request information.
- The system allocates the new task automatically to the specialist team.
- Teams log in at the start of the day to receive their tasks by PC or mobile.
- They then use the system to update task status, order parts and get their next task.
- The administrator views the active summary of the latest tasks and updates.
- The administrator and teams can display the location of tasks on the map.

Figure 11 shows the system use-case diagram; this diagram illustrates the team monitor application and functionality for each user as mentioned above.
3.7 Analysis:
After the data gathering stage, the developer starts to analyse users’ requirements that have been collected and concentrates on the features and services that the users need from the system. It is important to define the system requirements before proceeding to the next section. The system requirements contain two parts: functional requirements and non-functional requirements.
3.8 Functional requirements:
A functional requirement defines the function or features of a project that the users need from the system. These functions will define the system environment and behaviour.

There are two types of system functions:

1. Administrator Functions: these functions permit the monitoring and control of the system.

2. Team Functions: these functions have limited levels of permission according to the stakeholder required.

3.8.1 Administrator functional requirements:

3.8.1.1 System authentication: (high priority)
This section is the most significant section in the project. The purpose of this action is to protect the system from unauthorized users. The system must test the username and password at an authentication stage before allowing access to the system.

3.8.1.2 Generate and distribute tasks: (high priority)
The administrator has the authority to generate tasks and the system will automatically allocate the tasks generated by the administrator to the specialist team. The administrator will also be able to edit this distribution. In addition, the system can allocate these tasks to the backup team that specialises in task problems, i.e. if the main team reaches its daily task target of, for example, 5 tasks a day or has an overload of work. Additionally, the system has a waiting list for tasks if the teams have already met their daily task target. The system therefore holds the tasks on a waiting list until one of the specialist teams is available to take it, at which point, the system automatically allocates it.

3.8.1.3 Sending SMS and E-mail: (low priority)
The system should be able to send SMS massages to teams to alert them about urgent tasks. The administrator can also contact teams by sending e-mails.

3.8.1.4 Display location: (high priority)
The system will display the problem location on the map by connecting to the Google map. Google Map has recently offered the developers use of their maps as part of their systems.
### 3.8.1.5 Update tasks: (high priority)
There are two kinds of task update. First, there is the task status which allows access for both the administrator and team members. The second one permits editing and deleting. It is only the administrator who can access these features.

### 3.8.1.6 Tasks report: (high priority)
The administrator can monitor the team’s progress by browsing the report on each team. The report will allow the administrator to search by team name, task problem, statuses and risk level.

### 3.8.1.7 Team management: (high priority)
The system should be able to manage the teams’ groups by viewing existing teams, creating new teams, deleting and managing team members.

### 3.8.1.8 Tracking system and Mobile application: (Low priority)
The system can track each group by identifying their location. The groups can also browse the application on their mobile phones and update their task status.

### 3.8.1.8 Administrator Use- Case Textual Description:

<table>
<thead>
<tr>
<th>1- Administrator website use case:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal:</strong> To add new users and teams.</td>
</tr>
<tr>
<td><strong>Actor:</strong> Administrator.</td>
</tr>
</tbody>
</table>

**Main success scenario:**
- Create new user.
- Choose user type.
- Fill the administrator or team fields.
- System checks username and password.
- System creates a new user.

**Alternatives:**
- Teams or administrator already exist.
- User is informed of error, where a user already exists. Writes a new user name.
- Administrator misses some compulsory information.
  Administrator is informed of error and highlights the missing field to complete them once more.

<table>
<thead>
<tr>
<th>2- Administrator website use-case:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal:</strong> Generate new task and update task.</td>
</tr>
<tr>
<td><strong>Actor:</strong> Administrator.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Main success scenario:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Problem request received.</td>
</tr>
<tr>
<td>- New task generated.</td>
</tr>
<tr>
<td>- Problem type is chosen according to the request description and the required fields are filled.</td>
</tr>
<tr>
<td>- System automatically assigns new task to the team who specialise in the request problem.</td>
</tr>
<tr>
<td>- System displays the new task location on the Google map.</td>
</tr>
<tr>
<td>- System displays the updated task summary to the administrator.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alternatives:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- The Team already has five tasks:</td>
</tr>
<tr>
<td>The system assigns the task to the back-up team.</td>
</tr>
<tr>
<td>- The back-up team already has five tasks:</td>
</tr>
<tr>
<td>The system sends the task to the waiting list.</td>
</tr>
<tr>
<td>- Administrator misses some compulsory information.</td>
</tr>
<tr>
<td>Administrator is informed of error and highlights the missing field to complete once more.</td>
</tr>
<tr>
<td>3. Administrator website use-case:</td>
</tr>
<tr>
<td>----------------------------------</td>
</tr>
<tr>
<td><strong>Goal:</strong> To manage teams and add new team members.</td>
</tr>
<tr>
<td><strong>Actor:</strong> Administrator.</td>
</tr>
</tbody>
</table>

**Main success scenario:**
- Choose team to add new member.
- Create new team member.
- Fill the member fields and choose the position.
- System displays the member in the team monitor.

**Alternatives:**
- The new member already exists:
  - User is informed of error which is that the member already exists. New member name is chosen.
- The team already has a leadership position:
  - User is informed of error which is that the team already has a leader. Leader is changed or removed.
- Member is a leader in the same team.
  - User is informed of error which is that the member is a leader. Member position is changed.

### 3.8.2 Team’s functional requirements:

#### 3.8.2.1 Browsing tasks: (high priority)
The team members can browse their tasks by mobile browser and PC. They can also update the status of each one if the particular task is pending or closed. Additionally, they can write comments about the problem or description. They will also display the location of each task.
3.8.2.2 Ordering parts: (low priority)
The user can also order parts and decide the level of importance these parts.

3.8.2.3 Tasks Report: (high priority)
Team can report to the administrator concerning the problem status.

3.8.2.4 Team GPS: (low priority)
Teams can use the mobile GPS via the team monitor application to reach the problem location. The system will integrate with mobile GPS, so the system will send the address to GPS and thus display the route for the team.

3.8.2.5 Teams’ use-case textual description:

<table>
<thead>
<tr>
<th>1. Teams’ website use-case:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal:</strong> To get and update tasks.</td>
</tr>
<tr>
<td><strong>Actor:</strong> Team leader.</td>
</tr>
<tr>
<td><strong>Main success scenario:</strong></td>
</tr>
<tr>
<td>• Get the team tasks.</td>
</tr>
<tr>
<td>• Leader updates task status and description.</td>
</tr>
<tr>
<td>• Find the request location on Google map.</td>
</tr>
<tr>
<td>• Order parts.</td>
</tr>
</tbody>
</table>

3.9 Non-functional requirements:
There are several points that the developer should take into account when the systems develop. In fact, the users are not involved at this stage. However, the benefits of these points are to make the system better, more efficient and simpler. These requirements will be defined as following:

3.9.1 Speed:
In terms of the speed issue, it will have different kinds of effect. Some of them are related to developing language performance and others relate to the code structure and implementation. In
terms of implementation, the developer concentrates on the system design to produce a high performance system.

3.9.2 Learnability and simplicity:
The system should be easy to learn and use. The developer should also avoid complexity in the system. Furthermore, the operation of the system should be more automated than the manual system to achieve the system’s target and encourage users to use it. According to the results of the questionnaire, there are a number of system users that had never used computer applications before or who did not have basic skills in terms of using computers and systems. This sample forced the developer to take into account the simplicity issues in the system design, as well as implementation of features to help users access the system without facing any difficulties.

3.9.3 Efficiency:
The users’ requirements should be met within an efficient structure which matches the user’s system needs exactly. Furthermore, it should only take a short time to execute any request from a user.

3.9.4 Security:
The security issue is one of the important points that the developer should focus on, due to the fact that security issues are increasing on a daily basis. The developer will therefore follow the MOWE security rules to ensure the system is built in the right way. For instance, the system must have an authentication gate (username and password) to allow users to log into it. It must also have authentication sessions on each page of the system to avoid any attacks.

3.10 Technical requirements:
To be able to establish the project, there are some technical requirements such as hardware components or software resources. The requirements are classified into two kinds: hardware requirements and software requirements.
3.10.1 Hardware requirements:
Hardware is a term for the physical computer resources. These requirements are very important in order to be able to run the application. They include the following:

- **PC**: the PC should include the following specifications:
  - RAM: 1.00 GB or higher.
  - Processor: 2.00 GHz or higher

- **Server (recommended)**: should include the regular specifications to serve and manage the system. The server should include the following specifications:
  - RAM: 4.00 GB or higher to be able to bear an overload of requests.
  - 500 GB hard disk or higher: to be able to store the system and database.

3.11.2 Software requirements:
The following software is also required to run and use the system. Some of these types of software are important to the developers and others, e.g. for the users:

- Operating system: Linex Enterprise edition or Windows server (developers)
- My SQL database to save and organize the system data (developers).
- PHP editor(developers)
- Web browser(users and developers)

3.11 Future modification:
The system is developed using a web application programme. This model makes the system easy to extend and allows for the easy addition of new features. The new modifications should follow the same approach so any future work can easily be developed in the system.

3.12 Requirement test:
RUP methodology is required to test each phase in the development. As previously mentioned, the developer used the questionnaire in this phase to test the requirements gathered from interviews. The developer took the feedback into account and improved the design so that the application could be implemented in the right way.
Chapter 4: System design

4.1 Introduction:
Before designing the system interface, user issues should be considered. This is necessary before deciding on the type of system design to be implemented. Human–computer interaction (HCI) issues must be taken into account to ensure that users are comfortable when they interact with the system. In addition, during the design stage, it is important to remember how the layout of the system interface will be viewed when considering colour, font, images etc.

Moreover, the general architecture of the system and its behaviour should be considered and investigated at the design stage. In fact, it is at this stage that all user requirements will be converted to the corresponding system components.

4.2 Human Computer Interaction (HCI):
Human–computer interaction is the study of the interaction between users and computers. This will affect the design, evaluation and implementation of interactive computing systems for human use (Wright et al., 2000).

As the team monitor application is a web-based system, it was necessary to focus on the HCI issues which related to the application design and present a clear idea about the implementation of the system in addition to the design aspects. Moreover, there are a number of usability rules that need attention while designing the interface. These give the developer an understanding of the methodology behind a correct approach to designing the system. The following rules provide a brief description of the usability principles:

Visibility
This rule takes into account those system functions which are blatantly visible to the user. This is one of the important aspects of usability. The difficulties that the user faces in the design can be decreased through this, and it can be helped with the aid of icons and clear text.

Feedback
Feedback is the main theory of visibility. This reveals the users’ reactions and the main issues of the design to notify the developer about any complexity or weakness in it.
Consistency

Consistency refers to the system design and layout when all features and functionality of the system have been planned. For this system, it is necessary that the design and layout should be consistent in order to render them comfortable and simple to use. The user will therefore not need to make excessive effort to learn about them and he/she will find this process easy.

Affordance

Affordance refers to how the elements or functions can be clearly understood by the user. In other words, does the user understand what the system icons do if he/she clicks on them?

4.3 Design methodology:

As is to be expected with RUP methodology phases, the system design falls within the elaboration phase. The aim of this phase is to enable the project to take shape:

- Software architecture and the description of the system development process.
- A design plan and strategy.
- Prototypes and interfaces that demonstrate and describe the system.

4.4 Design strategy:

The design strategy aids the developer in following the rule so that the phase target can be adequately and correctly met. Here the developer writes the headlines of the system headlines strategy to remind him to follow them step by step:

- Choosing the system architecture.
- Designing the system diagram, ERD and activity diagram for system functionality.
- Designing the standard elements of the system interfaces.
- Applying user feedback when evaluating the system prototypes.
- Illustrating the aim of each interface.
- Testing the design with users to reach agreement.

4.5 Selected tools:

In a three-tier architecture situation, there are several tools which can be used to develop applications. Here the developer has studied and compared some of these tools for this
application, e.g. PHP and ASP. In terms of databases, the developer has also compared some of them such as the SQL server and MYSQL. Following the studies mentioned above, the developer has chosen to work with the tools listed below:

- HTML.
- PHP.
- MYSQL.

There are three main reasons behind the decision to work with the above tools for development:

1. **Cost**: PHP and MYSQL are completely free. No license needed to use PHP tools.
2. **Ease of Use**: PHP and MYSQL are easy to learn compared with many other languages. PHP has syntax that is easy to parse and is actually rather human-friendly. There is also a huge amount of free resources and support.
3. **Embedded**: PHP can be easily embedded directly into HTML.
4. **Compatibility**: PHP runs native on all UNIX and Windows platforms in contrast with ASP that is only compatible with Windows platforms.
4.6 General System architecture:

![Diagram of General System Architecture]

As shown in Figure 11, the system consists of four operative aspects. The first aspect is the system administrator who controls the whole system and monitors the team’s progress. The second part is the administrator web site which contains the control panel for the system and full permission to add, send and delete tasks. The third part is the teams’ web site where teams can get their tasks. It has limited permission to edit task information or display the map to indicate task location. The last aspect involves how the team gets and updates the requests.

4.6.1 Three-Tier Architecture:
Three-tier is a client–server architecture where the system interface, functional process, and data storage (database) are developed and maintained as independent modules, most often developed on separate platforms, for instance, the HTMP, PHP, MySQL database. It was developed by John J. Donovan of the Open Environment Corporation.

The system architecture was designed based on the three-tier architecture mentioned above. In fact, most of web applications are more appropriate to the three-tier architecture. It is very
important to this application because the application is web-based, and therefore a dynamic database supporter. In this design, each element of the three-tier architecture is connected with the other.

Figure 12 shows the three-tier architecture. The next example will illustrate and describe more details about each separate tier. The team, or in other words the end-user, can browse or update the request on the website page, i.e. Tier 1. Tier 2 in this case, is a front-end tier between Tier 3 and Tier 1. The front-end tier may also appear as the middle-tier, which obtains the query or update from the front-end tier to process it and retrieves the results from Tier 3, the back-end tier which is the database behind Middle tier.

- **The advantages of three tiers:**
  - It is easy for the developer to modify or change any tier without affecting the others.
  - Functionality is separated between the application and database, thus better balancing the load. One server can handle a lot of client requests.
  - Security policies can change within the server tiers without hindering the clients.
4.6.1.1 Front-end Design (Tier 1):
This system uses PHP and HTML languages to interact with users. The user (administrator or team leader) can generate or update the request and view the result that appears in the main page on the system after processing. The front-end tier can also respond by validating the request input that the user has inserted. This tier contains static interfaces that do not have logic or dynamic mechanisms such as a login page.

The standard design for the system pages is based on the MOWE standard. The design should be simple and has three main areas as follows: Banner Area, Content Area and Navigation Area.

![Diagram of Interface design]

These three elements of the above homepage will be illustrated as follows:

1. **The Banner Area**: displays the application title which is the Ministry of Water logo.
2. **The Content Area**: displays the results according to the user order. For example, the main page illustrates a summary of the latest teams’ requests.
3. The Navigation Area: contains a list of control panel buttons.

4.6.1.1.1 Application Map Diagram:
The application map diagram is the project scheme that illustrates the architecture of this system, such as the relationship between the project pages. In this project there are two kinds of
application map, the first one being the administrator web map. Figure 14 will show the web pages and their relationships.

**Administrator Web map**

![Administrator Web map diagram]

**Figure 14: Administrator web diagram**

As shows the number of pages in the administrator section and how these pages connect to together. Also shown are some pages inherited from the main page.

The second part is the team page map. Figure 15 shows the web pages and their connections.
There are differences between the teams’ application section and the administrator section in order to differentiate between their tasks and missions and also between the limited permissions of the teams. Teams only require interaction with the above websites (update task, show map, display location, receive task and order parts).

4.6.1.1.2 User Interface Design:
Following this, the front-end was designed. The user interface was also designed before the actual application was published. The user interface was designed according to the user requirements and use-case. The designer has taken into account ease and simplicity in design and avoided complexity and difficulty. The functional features are also equally significant when focusing on the feel and look of the user interface. The user interfaces were then shown to the users to get feedback and they were subsequently improved for suitability before finally developing the application. This phase was very difficult and challenging due to the fact the designer had to spend extra time explaining and interpreting these interfaces and applications with some employees who had not used computer applications before, in order to understand and prompt feedback.
There are in fact two types of user:-

**The administrator:** according to the requirements questionnaire, the majority of system administrators have a good level of computer skill.

**Team leaders:** the level of computer skills amongst these users is around ‘beginner’ according to the questionnaire results.

The designer concentrated on some questions put to the users while they were using the application interface, in order to discover how the interface design could be improved:

- Is the interface easy for the user to use?
- Is the application comprehensible to the user?
- Is the functional feature comprehensible?

The author designed a list of the system interfaces by HTML code as shown below. These figures illustrated the system design before the implementation stage to start helping the users and developers to imagine the real application and evaluate the content of each page.

**4.6.1.1.3 Authentication process:**
The website system switches users to their pages after the system has checked the user authorisation as either Administrator or Team. The login form (see Figure 16) has two fields: the first field is for the username and the second field is for the password. The system also has two buttons: a Login button to send the log in details to the system. A Reset button to reset any data or text inserted inside the text fields. This form was created by HTML code and located in the front-end tier. This page was processed by using the three other languages inside the form: PHP to read from the user, and verify the authentication process and SQL commands to check the information provided in the system database and to open the main page to the user according to his authority. The system will also provide a textual message if the username or password is invalid.
4.6.1.2 Middle-Tier Design (Tier 2):
This layer coordinates the application, processes commands, makes logical decisions and evaluates. This layer is designed to execute the operations behind the scenes using PHP, so the user will only see the result of these executions.

4.6.1.2.1 The Login Page:
The Login page can be defined as a ‘checkpoint’ page that protects the homepage from being directly accessed by external users and ensures that the system provides its services to registered users with different levels of access and permission.

4.6.1.2.2 The Main Page:
The main page works as a control panel interface and it functions across two tiers (front-end tier and middle tier) simultaneously. It contains a summary of the latest group tasks in addition to navigation buttons that allow the system users to carry out their mission according to their level of permission.
4.6.1.2.3 Create user page:
Only the system administrator may allow access to this page. He can manage the system’s users by viewing existing users, adding a new user, and deleting users via this page. He can also determine the level of permission and access.

4.6.1.2.4 New task page:
This page is one of the more significant pages in this application. It is aimed at generating a new task for the Team from the Administrator who fills in the task data. The system sends the new
task to the Team specialising in this task. If the number of tasks for the Team reaches the target limit, then the system sends it to the back-up Team who also specialise in the task but whose main mission is different. The system checks if the back-up Team has reached the task limit. The system will then hold the task on a waiting list until another existing task status closes.

4.6.1.2.4.1 New task activity diagram:
The activity diagram below illustrates the way this part of the system works.
4.6.1.2.5 **Employee management page:**
This page is aimed at stimulating the integration between human resource applications and the team management application. This simulation is to enable the system Administrator to manage the Team members by recognizing the number of technicians in each Team. It also displays important details on each member and adds or removes members from the Teams. The application is designed to be flexible to connect to human resource applications so they can interact with each other or connect to the human resource database tables. These ideas depend on agreement between the human resource application developers, the database administrator, and human resource administrator.

![Employee management page](image)

**Figure 21: Team management page**

4.6.1.2.6 **Location page:**
This page is to display the task location by using Google maps. The developer has created this page to accept two kinds of location descriptions (location coordinates, postcode). It is designed to start from the Team’s central location. This page is aimed at stimulating the GIS system that contains location and infrastructure information to help the Teams identify the problem and the tools that the Technicians will need. The cost and complexity of GIS applications were the main reasons for simulating this technology through Google maps in TMA. However, the TMA application is designed to be flexible so it can connect to the MOWE GIS, as mentioned in the Employee Management page.
4.6.1.2.7 Report Page:
As with most popular applications, there is a Reports feature to display statistics and the progress of the application. However, on this page, the developer displays a system report in a way that differs from that of a standard report. This page utilizes AJAX to display the report. The Team’s name and system will display the details and statistics of the report according to the maintenance requirements.

<table>
<thead>
<tr>
<th>Team name</th>
<th>problem type</th>
<th>Location</th>
<th>Date</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tiger</td>
<td>pipe</td>
<td>DD</td>
<td>19-07-2010</td>
<td>Open</td>
<td>Sending in main pipe</td>
</tr>
<tr>
<td>Tiger</td>
<td>pipe</td>
<td>DD</td>
<td>19-07-2010</td>
<td>Close</td>
<td>DDD</td>
</tr>
<tr>
<td>Tiger</td>
<td>pipe</td>
<td>MN</td>
<td>19-07-2010</td>
<td>Close</td>
<td>N,M</td>
</tr>
<tr>
<td>Tiger</td>
<td>pipe</td>
<td>AF</td>
<td>19-07-2010</td>
<td>Closed</td>
<td>DFF</td>
</tr>
<tr>
<td>Tiger</td>
<td>pipe</td>
<td>fflotk</td>
<td>19-07-2010</td>
<td>Close</td>
<td>k-jkldh</td>
</tr>
<tr>
<td>Tiger</td>
<td>pipe</td>
<td>EH12 3DD</td>
<td>21-07-2010</td>
<td>Open</td>
<td>The main pipe in the central road is broken</td>
</tr>
<tr>
<td>Tiger</td>
<td>pipe</td>
<td>BH23 1Gh</td>
<td>22-07-2010</td>
<td>Pending</td>
<td>Pipe broken</td>
</tr>
<tr>
<td>Tiger</td>
<td>pipe</td>
<td>BH22 1GU</td>
<td>23-07-2010</td>
<td>Close</td>
<td>pipe problem</td>
</tr>
<tr>
<td>Tiger</td>
<td>pipe</td>
<td>EH12 1GU</td>
<td>24-07-2010</td>
<td>Closed</td>
<td>main pipe</td>
</tr>
<tr>
<td>Tiger</td>
<td>pipe</td>
<td>hzdfs</td>
<td>25-07-2010</td>
<td>Open</td>
<td>shelns</td>
</tr>
<tr>
<td>Tiger</td>
<td>pipe</td>
<td>bhh</td>
<td>26-07-2010</td>
<td>Close</td>
<td>kjhk</td>
</tr>
<tr>
<td>Tiger</td>
<td>pipe</td>
<td>ojoe</td>
<td>28-07-2010</td>
<td>Open</td>
<td>ojoe</td>
</tr>
</tbody>
</table>

(Summary statistics)

<table>
<thead>
<tr>
<th>The total task</th>
<th>The number of Open tickets</th>
<th>The number of Pending tickets</th>
<th>The number of closed tickets</th>
</tr>
</thead>
<tbody>
<tr>
<td>14 Tasks</td>
<td>4 Tickets</td>
<td>1 Tickets</td>
<td>9 Tickets</td>
</tr>
</tbody>
</table>

Figure 22: Location

Figure 23: Report page
4.6.1.2.8 Task edit page:
This page is used to edit the task information and it is accessible to both Administrator and Teams. It is designed to update particular tasks according to the level of user permission. For example, the Administrator has permission to update and delete all task information but the teams can only update certain fields (description, status, order parts).

![Request Edit](image)

Figure 24: Request edit

4.6.1.2.9 Colour choices:
In terms of the application’s colour, as mentioned in the project objectives, the system is official and not the same as regular web sites. There are therefore some design rules the developer followed when he designed the interfaces according to MOWE application sanders. One of the most important rules is that the system should have a simple design and that the interfaces have the Ministry logo on each page. The background colour had to be white.

4.6.1.3 Back-end Design (Tier 3):

4.6.1.3.1 Database design:
The user interface design helped the developer to define the required data users needed to view. These were helpful for imagining and designing the number of database tables and deciding how many fields each table needed and the relationship between these tables. The designers made the database clear and easy to understand and avoided duplication. The MTA database consists of
six tables (Users, Tasks, Teams, Department, Employee, and Parts Order). Each table has its own fields and some fields relate to the tables. The next figure will illustrate the database structure.

![Figure 25: ERD diagram](image)

**User Table:**

The table of users contains the fields needed to store the users’ details. This table has a relationship with the ‘Team Table’ and ‘Task Table’. More details see database table’s appendices.

**Task Table:**

This table is the main table in the Team Monitor project. This table is designed to store all the task information. It relates to the ‘User Table’ and the ‘Parts Request Table’. More details see database table’s appendices.
Team Table:
This table contains the full details of each team. It has a relationship with the ‘User Table’ and ‘Employee Table’. More details see database table’s appendices.

Employee Table:
This table contains organization employee information. In fact, this table has been created to simulate the HR application in MOWE. It has a relationship with the ‘Team Table’ and the ‘Department Table’. More details see database table’s appendices.

Parts order table:
This table contains information on parts. It has a relationship with the ‘Task Table’. More details see database table’s appendices.

Department Table:
This table contains information on parts. It has a relationship with the ‘Employee Table’, in identifying the user’s department. More details see database table’s appendices.

4.7 Design Test:
RUP methodology refers to the test at each phase before the developer moves to the next one. This technique aids the developer in getting an agreement from the users about what has been done and corrects any complexity or duplication before the system is implemented.

After the system has been designed using basic HTML pages, the developer uploads these interfaces and allows the users to interact with and evaluate them. It also sends a copy of the database to the IT department at the Ministry of Water so they can evaluate it. The developer has authorized one of the IT experts to observe the user and database administrator feedback and return a feedback notice to the developer.

Finally, the users and experts were agreed on the system design and encouraged the developer to carry on implementing the system so it could be revealed in the real world.
Chapter 5: Implementation

5.1 Introduction:
In this chapter, the design stage of the previous chapter will begin its implementation and interpretation in the real world as an application for meeting user requirements. The system implementation refers to the process of creating the working application using the database as a data source and storage. The application implemented by PHP acts as an interpreter for the server side, and HTML, JavaScript and CSS interpret for the client side.

5.2 Implementation Strategy:
As mentioned before, RUP methodology was used to build this project. This step falls within the Construction Phase in RUP methodology. There are significant points that the developer took into account at the outset of implementation:

- Any difficulty in design or the application code should be redesigned or re-coded to make it easy to understand future improvements.
- The implementation should start from high priority functionalities.
- The application should be built using a code which is easily understood, due to the fact that the application will continue improving at MOW.
- Modifications should be easy to carry out by dividing and labelling the code functions and complexity avoided.
- As mentioned earlier, some of the application functionalities must be integrated with other applications, such as GIS.
- The application interfaces and functionalities should be produced as the user and developer have agreed.

5.2.1 Home page implementation:
The application has two main user groups and each has their own main page according to their level of permission. The system traces users automatically after validating the username and password. These are distinguished by an authentication server through the Login page, which is the main gate into the system. The system uses ‘PHP SESSION’ to store information for user’s entire session. This step helps non-localised information in the server memory, regardless of the user’s access level. The login information can then be stored in the memory and accessed easily.
Most of the system pages require a session so they start with the syntax ‘SESSION\_START();’. Hence, the main page will identify the user by “USER\_SESSION” and then the system will display the user tasks (Figure 26).

**Welcome Tiger**

<table>
<thead>
<tr>
<th>location</th>
<th>date</th>
<th>status</th>
<th>Description</th>
<th>problem_type</th>
<th>customer_main</th>
<th>Risk_level</th>
<th>Team</th>
<th>Parts</th>
<th>Edit</th>
<th>Show</th>
</tr>
</thead>
<tbody>
<tr>
<td>EH11 1QZ</td>
<td>19/07/2010</td>
<td>Open</td>
<td>Seeking in main pipe</td>
<td>pipe</td>
<td>079999883999</td>
<td>High</td>
<td>Tiger</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EH12 3DD</td>
<td>21/07/2010</td>
<td>Open</td>
<td>The main pipe in the central road is broken</td>
<td>pipe</td>
<td>090000009990</td>
<td>High</td>
<td>Tiger</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EH22 1Gb</td>
<td>22/07/2010</td>
<td>Pending</td>
<td>Pipe broken</td>
<td>pipe</td>
<td>09900000</td>
<td>Middle</td>
<td>Tiger</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 26: Team main page**

On the other hand, the Administrator main page is completely different from the Teams page. The Administrator is the system manager. Consequently, in this situation, the system will briefly display the number of tasks for each group and the waiting list. In addition to presenting the task information, here, the Administrator can control and navigate the system (Figure 27).

**Figure 27: Administrator main page**
As shown in Figure 27, the first table displays the group’s information and waiting list to give the system manager a general idea before checking the task details. In terms of the ‘Active Tasks Details Table’, the information only shows active tasks sorted according to the latest task, so if one of these task statuses is changed to ‘close’, then the task will immediately disappear from the active table.

5.2.2 Create user page implementation:
This page is important for the Administrator when choosing system users. The Administrator has two users’ options (Administrator user, Team user). This page deals with two tables in the project database, so when he/she chooses the Administrator user, the application will only be connected to the User Table and the application will only display the User field.

On the other hand, when choosing the user as a Team, the application opens the Team fields and connects to a Team Table, in addition to the User table (see Figure 28).

![Figure 28: Create new user page](image)

5.2.3 Tasks page implementation:
This page is one of the more important pages in the Team monitor application. It allows the Administrator to create new tasks from it. The Administrator will fill in the main fields (Figure 29) that the system requires to identify the problem and deal with it. The application will then automatically allocate the Team who specialise in the task entered by the Administrator.
The developer has used ‘Span’ on this page to display messages to alert the user if he forgets any required field and JAVA script to colour the missing field so that it can be immediately recognised which field is missing (Figure 29).

Once the save button is pressed, the system will send the task to the Team specializing in that type of problem (See figure 30). The developer designs “Team table” in the database as mentioned in the Design chapter to involve ‘Main task and ‘Second task’ for all system teams. The idea of this design is to help the system allocate tasks according to the problem type entered by the Administrator. The system will therefore send the ‘Problem Type’ to the database and
compare it with the main task for each Team. If it is the same, then the system will check the number of tasks for the Team before assigning the task. There is a task quota for each Team which totals (5 tasks). Therefore, if the Team specializing in the assigned task already has 5 tasks, the system will check the back-up Team whose second task matches the ‘Problem Type’. The system will then check the quota again. If a Team has less than (5 tasks), then the system will assign the task to them. However, if the quota of the back-up Team exceeds 5 tasks, then the system will assign it to the waiting list (Figure 31).

This page also contains an e-mail service (Figure 30). It gives the Administrator the option to send the request by e-mail to alert the Team leader of the request. When the check box is clicked, the e-mail will be sent from the system via the organisation’s SMTP, which contains brief information about the new request. The code below illustrates this message sending concept.

```php
if ($_POST['Submit'])
{
//this refers to the message information
$name = $_POST['name'];
$message = $_POST['message'];
if($name&&$message)
{
//this is the SMTP configuration
ini_set("SMTP", "mailhost.hw.ac.uk");
die();
//Here is the message detail
$to = "TeamLeader@gmail.com";
$subject = "Urgent Task";
$headers = "From: mia@hw.ac.uk";
$body = "Urgent Task Please Login to the TMP to See Details";
mail($to,$subject,$body,$headers);
}
```
The previous idea for this step was to send an SMS message. However, this would need to be contracted with a third party, after that the developer would be allowed to configure his system to send SMS messages via their services. Actually, there are two reasons which led to this idea being rejected. The first one is the cost of the service which was difficult for the developer to meet. The second reason is the code developed by the company which is ready to use and so the developer is not allowed to edit it.

5.2.4 Assigning tasks algorithm:
The algorithm of assigning tasks will illustrate how the main function is performing and the technique designed.

```
Select “Team Name” from Team Table where “main task” of the Team = “problem type” is entered;
Select the number of Team tasks that already have ‘without tasks’ status and “close” ;

if the number of tasks < 5 then {
    assign it to this Team;
}

Else
Select Team Name from Team Table where “second task” of the Team = “problem type” is entered;

Select the number of Team tasks that already have ‘without task’ status and “close” ;

if the number of tasks < 5 then {
    assign it to this Team;
}

Else
assign it to the “waiting list”;
```
The following code will illustrate how the Assign Task function works.

```php
// get the Team Name whose main task matches the problem type
$sqlTN="SELECT Team_name FROM team WHERE Main_task='$_POST[prob]'";
$TNq=mysql_query($sqlTN,$conn);
$TN=mysql_fetch_array($TNq);
$main_team= $TN[Team_name];

// Now check if that team is already assigned 5 tasks.
$sqlCheckLimit = "SELECT COUNT(*) CNT FROM gen WHERE Team='$TN[Team_name]' and status <>'close'";
$CheckLimitq=mysql_query($sqlCheckLimit,$conn);
$CheckLimit=mysql_fetch_array($CheckLimitq);

// get the back-up Team Name for the selected problem whose second task matches the problem type
$sqlTN="SELECT Team_name FROM team WHERE second_task='$_POST[prob]'";
$TNq=mysql_query($sqlTN,$conn);
$TN=mysql_fetch_array($TNq);
$second_team= $TN[Team_name];

// Now check if that team is already assigned 5 tasks.
$sqlCheckLimit4 = "SELECT COUNT(*) CNT FROM gen WHERE Team='$TN[Team_name]' and status <>'close'";
```

By Mohammed Alwanain
Team=$TN[Team_name]' and status <>'close'';
    $CheckLimitq=mysql_query($sqlCheckLimit4,$conn);
    $CheckLimit4=mysql_fetch_array($CheckLimitq);

    if($CheckLimit['CNT']<5)
        //assign to main team
        {
            $assign_team= $main_team;
        }
    elseif ($CheckLimit4['CNT']<5)
        {
            //assign to second team
            $assign_team= $second_team;
        }
    else
        {
            //assign to waiting
            $assign_team= 'waiting';
        }

5.2.5 Waiting list implementation:
As discussed in tasks implementation, when each Team (specialist and back-up) reaches its full quota, the system will send the task to the waiting list. The technique of this step is checking the Teams’ quota every 30 seconds. If one of the specialist or back-up Teams has completed an active task, then the waiting task will automatically be assigned to him (see figure 31).
5.2.6 Waiting list algorithm:
The waiting list algorithm will illustrate how the waiting list is performing and the technique designed. As shown in Figure 27, the waiting list function is built into the Administrator page and it acts as a listener to check task progress and updates so that a waiting task can be assigned to a specialist Team.

```
Select the number of waiting tasks in the waiting list;

Select "order id" and "problem yype" from the waiting list;
Select "team name" from the Team Table where "main task" of the Team = "problem type" of waiting task;

Select the number of Team tasks that already have 'without task' status and "close";
if the number of tasks < 5 then {
    re-assign it to this Team;
}
Else
Select Team Name from Team Table where "second task" of the Team = "problem type" of waiting task;

Select the number of Team tasks that already have 'without task' status and "close";
if the numbers of tasks < 5 then{
    re-assign it to this Team;
}
Otherwise do nothing;
```
The following code illustrate how the Assign Task from Waiting List function works.

```php
// Waiting List
// get the number of the Waiting List
$sqlTN0 = "SELECT COUNT(*) FROM gen WHERE team = 'waiting';"
$TNq0 = mysql_query($sqlTN0, $conn);
$TN0 = mysql_fetch_array($TNq0);

// get the order id, problem type from the waiting list.
$sqlTNs = "SELECT order_id, problem_type FROM gen WHERE team = 'waiting';"
$result = mysql_query("SELECT order_id, problem_type FROM gen WHERE team = 'waiting';");
while($row = mysql_fetch_array($result))
{
    $li_order_id = $row['order_id'];
    $li_problem_type = $row['problem_type'];

    // get the team name from the selected problem type
    $sqlTN = "SELECT Team_name FROM team WHERE Main_task='$li_problem_type';"
    $TNq = mysql_query($sqlTN, $conn);
    $TN = mysql_fetch_array($TNq);
    $main_team = $TN[Team_name];

    // How many tasks are assigned to the specialist team?
    $sqlCheckLimit = "SELECT COUNT(*) CNT FROM gen WHERE Team='$main_team' and
```
status =个多小时';
    $CheckLimitq = mysql_query($sqlCheckLimit, $conn);
    $CheckLimit = mysql_fetch_array($CheckLimitq);
    
    // How many tasks are assigned to the specialist team??
    $sqlTN = "SELECT Team_name FROM team WHERE second_task='\$li_problem_type' ";
    $TNq = mysql_query($sqlTN, $conn);
    $TN = mysql_fetch_array($TNq);
    $second_team = $TN[Team_name];
    $sqlCheckLimit4 = "SELECT COUNT(*) CNT FROM gen WHERE Team='$second_team' and status =个多小时' ";
    $CheckLimitq = mysql_query($sqlCheckLimit4, $conn);
    $CheckLimit4 = mysql_fetch_array($CheckLimitq);
    
    // check main team
    if($CheckLimit['CNT']<5)
    {
        //re-assign to the main team
        $updateSQL = "UPDATE gen SET team='$main_team' WHERE order_id= '$li_order_id'";
        $CheckLimitq = mysql_query($updateSQL, $conn);
        //CheckLimit = mysql_fetch_array($updateSQL);
    }
    elseif ($CheckLimit4['CNT']<5)
    {
        //re-assign to the back-up team
$updateSQL = "UPDATE gen SET team='$second_team' WHERE order_id='".$li_order_id."";
$CheckLimitq4=mysql_query($updateSQL,$conn);
}

5.2.7 AJAX Report:
In terms of the Report section, the developer has used Asynchronous JavaScript and XML (AJAX) to display the reports. The idea of this page is to display the system report without using the traditional report concept. This service allows the Administrator to show the report by using the AJAX search engine and display the information inside the dynamic table and statistics summary. The user can search by writing the Team’s name, problem type, status and risk level. In fact, the option to search via these fields was determined by the customer requirements as these fields are the main keys of the report. AJAX deals with all three tiers at the same time to display the user query. Tier 1 contains the HTML page that holds the search field (Figure 30).

![Report page diagram](image)

The HTML page uses the AJAX library from Google. Using this library has advantages and disadvantages. The advantages are that it avoids any complicated codes and aids the developer to directly write the code without any declaration in terms of the browsers type or XML. The disadvantages of using this library are that the system must remain connected to the Internet due to the fact that the library is purely an on-line facility.

The example below shows the Java script of library.
Next, the developer used the Java script function to send the variable to the PHP page. Compared with the common AJAX code, this method is straightforward and easily understood for the developers. It avoids any complexity relating to the code. The code below illustrates two AJAX functions.

The first one of code example below is the common AJAX function required to declare a list of variables; the browsers the user has used to run the application and the declared xmlHttpRequest for these browsers etc.

```javascript
<script type="text/javascript">
function Ajax(){
var xmlHttpRequest;
try{
    xmlHttpRequest=new XMLHttpRequest();// Firefox, Opera 8.0+, Safari
} catch (e){
    try{
        xmlHttpRequest=new ActiveXObject("Msxml2.XMLHTTP"); // Internet Explorer
    } catch (e){
        alert("No AJAX!? ");
        return false;
    }
    catch (e){
        try{
            xmlHttpRequest=new ActiveXObject("Microsoft.XMLHTTP");
        } catch (e){
            alert("No AJAX!? ");
            return false;
        }
    }
}
</script>
```
As shown above, there are many things that need to be declared before writing the main concept of the target page. Moreover, all of the above are only for declaring AJAX.

However, in this project the developer has used the AJAX library mentioned above to reduce the code indicated. It subsequently only needs a small function to send and its value. The code below illustrates the idea.

```javascript
//AJAX Function
<script type="text/javascript">
//this function will send the variable to Ajax.php
function ajax(value){
    $.post("Ajax.php", {partialState:value}, function(data){
        $('#result').html(data);
    }); }
</script>
```
As shown above, only two sentences apply to the AJAX function, so there is no need to declare more or present a complex code as illustrated in the first example. The mechanism of this page sends the variable to the Middle Tier (Tier 2) which is the PHP page. The PHP page then receives the variable sent from the HTML page and checks with the Tasks Table in the database by ‘Select Statement’. This contains each field of the Task Table that might by search.

The example below illustrates how the AJAX function works.

```
//AJAX
//This variable sent from HTML page
$partialState = $_POST['partialState'];
//this statement selection is to identify the type of variable
$states = mysql_query("SELECT * FROM Task WHERE team LIKE '%$partialState%' or status LIKE '%$partialState%' or Risk_level LIKE '%$partialState%' or problem_type LIKE '%$partialState%' ORDER BY gen.order_id");
```

The database (Back-End Tier) will send the result of the request to the Middle Tier (PHP page) (figure 32). In this situation, the Middle Tier will send the result as a dynamic table including short statistics. The table’s information will change automatically without refreshing the page (Figure 33).

![Figure 33: report page result](image-url)
5.2.8 Problem Location:
This part of the system is the aforementioned simulated GIS. The idea is to use Google maps via the application to illustrate the direction of each task. This service is used by the system Administrator as well as the Teams. This page identifies the location from the task’s location and displays the nearest routing (Figure 33). The location accepts the postcode of the area or the coordinates as fixable to use.

![Request Location](image)

Figure 34: Location page

5.2.9 Database implementation:

In order to implement the database, it is important to consider the Entity Relationship and transform it into tables. In fact, the ERD Model only provides basic information to establish the database of the system. However, after studying the requirements and data that need to be stored in the tables and confirming the list of entities and their attributes, it was easy to implement and configure the database in MySQL.

At first, the developer created the database and table using SQL statements. However, adding to and editing the database after that was carried out using MYSQL Wizard which is
PHPMyadmin. It was a very helpful tool to support the developer in editing the database and avoiding any duplication.

The MTA database name is ‘Team Project’. The name was chosen as easy to distinguish from other databases and to avoid clashing with any other external application databases on the same server.

Moreover, the names of the tables and fields were chosen for simplicity so that the developer would find it easy to understand and remember the target created and the information stored in it (see the Database Tables in the appendices). The field settings were implanted according to the data specifications to be stored. For example the primary keys of each table were chosen by creating an ID or unique field that needed to be primary. Some ID fields were automatically incremented; this would further help when implementing the system by uniquely identifying the details through ID.

While the database was being implemented, the developer had to consider the data types and how these would be used within the system. For instance, whether a field anticipates is long or short series of characters to be inserted. These fields were therefore set as ‘LONGTEXT’. Other fields were as ‘VARCHAR’, ‘DATETIME’, and ‘INTEGER’, according to the data type to be inserted. These settings did not check the user data and it is not permitted to enter unexpected data within fields. The developer also decided which fields would expect null values to be entered. The user was not permitted to leave certain fields without values, e.g. ‘Team Name’. On the other hand, there were some fields configured as ‘Null’ which accepted null values.

5.3 Technical problems and difficulties:

Usually, developers face technical problems spatially when they use new programming languages and when they develop complex functions. The author has faced several technical problems, starting with PHP language. This was the first time the author had used it or learned about it but was obliged to meet the system requirements with some features in a short time. It was a kind of challenge but the author decided to follow his idea. The second difficulty was the
database design. This step was very important due to the fact that, once the database is implemented it is very dangerous to change something in the main design, e.g. a dropped table or dropped field that might be linked with other tables. Database design is therefore very time-consuming and requires expert consultation, added to the effort of building the system. The third problem faced was when developing an e-mail service. The original idea for e-mail required an SMTP server to send it or the implementation of a mail server, both of which require time to set up, but it was tested on the MOWE’s mail server.
Chapter 6: System Testing

6.1 Introduction:
In this chapter we will start to test the Team application to verify that it matches the specifications and then validate it. In other words, validation verifies that the Team application is what the user wants. This phase is entered after the developer finishes implementing the system and ensuring it is clear for the users, and meets the requirements. It is also important to check there are no errors in it before the system is evaluated by the users.

6.2 System Testing Strategy:
As per the RUP methodology steps, the testing phase is one of the more significant phases (see Figure 9). In fact, the testing phase not only follows RUP implementation but also each other phase (Requirements testing, Design testing, and Implementation testing). After each phase, therefore, the developer tests the results and demonstrates to the customers in order to get their agreement. These are RUP techniques. The purposes of the Implementation test are:

- To validate the relationship and interaction between the system objects.
- To validate the integration of all components of the application.
- To validate and verify that all the application requirements have been correctly implemented.
- To identify the system faults and ensure they are resolved before the deployment of the software.

6.3 Application Testing:
After the system has been developed, the developer follows the test strategy to ensure and confirm that the system has been successful in all the testing roles. The system has been tested by the developer along with the Team leaders, checking whether the system functionality works as designed or not. Firstly, the developer contacts one of the IT department members to organize a meeting with Team leaders and the system Administrator in order to give them a brief description about how the application works. The users then log into the system via the Internet as the developer hosts the system as a local host and allows them to access the application through the PC IP (http://137.195.111.54/main.html). On the other side, the developer monitors
the users’ progress. The developer schedules the tasks the Administrator and users should accomplish to test the system as follows:

<table>
<thead>
<tr>
<th>Test</th>
<th>description</th>
<th>User</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add new user and Team</td>
<td>Testing the system users and authentications roles</td>
<td>Administrator</td>
<td>The system accepted new users and, Teams and refused the existing users</td>
</tr>
<tr>
<td>Add new task</td>
<td>Test generates new task</td>
<td>Administrator</td>
<td>The system generated and assigned it to the Team</td>
</tr>
<tr>
<td>Edit the existing task</td>
<td>Editing the task information</td>
<td>Administrator, Team</td>
<td>Accepted</td>
</tr>
<tr>
<td>Check the group quotas</td>
<td>Generating more that 5 tasks for each groups</td>
<td>Administrator</td>
<td>Assigned to waiting list</td>
</tr>
<tr>
<td>Check the waiting list</td>
<td>Closing or removing tasks</td>
<td>Administrator, Team</td>
<td>Assigned to the specialist team or a back-up team with less that 5 tasks</td>
</tr>
<tr>
<td>Location</td>
<td>Browsing task locations</td>
<td>Administrator, Team</td>
<td>Showed the task on Google map</td>
</tr>
<tr>
<td>Team management</td>
<td>Adding and removing new team member</td>
<td>Administrator</td>
<td>Accepted</td>
</tr>
<tr>
<td>Progress report</td>
<td>Displaying progress report by Team, problem type, status and risk level</td>
<td>Administrator</td>
<td>Accepted</td>
</tr>
</tbody>
</table>

By Mohammed Alwanain
6.4 Test summary:

- **Add new user and Team:**
  
  This test confirmed that the security of the system worked as expected. When the Administrator adds a new user he has to specify the type of user (Administrator, Team). After the system has accepted the new user, the team or users already registered in the system are refused.

- **Add new task:**
  
  This test was an important test. The system performed this task perfectly by generating the new task after the Administrator had filled in the required data and the system had automatically specified the task date and status and allocated it to the specialist Team.

- **Edit the existing task:**
  
  This test confirmed that the system can retrieve the task data from the database and edit according to the level of user permission.

- **Check the group quotas:**
  
  This test confirmed that the system had assigned the tasks to the specialist Team after checking the Teams quota. The Administrator generated more than 5 tasks and the system automatically assigned to the back-up Team.

- **Check the waiting list:**
  
  The system performed this task perfectly by fulfilling the quota of each team. The system automatically assigned new tasks to the waiting list. Following this, one of the Teams closed a task and the system directly assigned another to them.

- **Location:**
This test was to confirm that the system would allow the users to see the task location on a Google map. The Team did that by browsing the Task location and the system showed it on the map.

- **Team management:**
  This test was to show the Administrator the existing Teams and how the Administrator could add or delete Team members.

- **Progress report:**
  This test involved displaying the progress report. The Administrator tested the report by Team name, problem type, status and risk level. The system performed this task perfectly.

Ultimately, as shown above, the testing demonstrated that all system functionalities worked. However, some users referred to some shortcomings or made recommendations which will be addressed in future work.
Chapter 7: System Evaluation

7.1 Evaluation Design Rationale:
The system evaluation was carried out by MOWE Teams, the IT department developers, and maintenance managers. The developer met and contacted them online through Skype and explained the application functions and aims to them before evaluating the application. The developer selected these samples due to the fact they are the target users of the system, in order to avoid any type of professional, ethical or legal issues.

7.2 Evaluation Methodology and Plan
As the project follows the RUP methodology, the Evaluation step falls within the Transition Phase. The aim of this phase is to ensure that the system is available and can be understood by the end user. This point is to facilitate the user’s evaluation and feedback. The system evaluation was conducted via the IT department. The developer hosted the application as mentioned in the Test chapter and asked users to use the system as they would in the real world to give a better evaluation of it. In fact, there are some significant points that the developer concentrated on, as follows:

- Are the target users satisfied?
- Can the project team learn and use it easily?
- Does the quality of the system satisfy expectations?
- Does the application meet their needs?
- Does the system achieve the established goals?

7.3 Participants:
As mentioned above, the participants of the evaluation were computer experts, the system managers and Team members who tested the application at the Test stage. The aim of this sample was to collect feedback and technical evaluations from the actual users. 50% of the samples were actual users due to the developer’s need to apply and improve the system so that it would be more comfortable and useful for them. 30% of the samples were IT experts or developers because these had experience of evaluating the system in terms of technical issues.
Finally, 20% were the system Administrators and executive who evaluated the system in terms of user permission and the system functionality see Figure35.

![Participants chart](image)

**Figure 35: Participant Chart**

### 7.4 Questionnaire:
Each one of the test samples testing the system were asked to fill in a questionnaire form (see the appendices) in order to obtain their feedback on the system. The developer wrote an evaluation questionnaire using a Google Documents form. This was very useful for the researcher because the questionnaire results were received directly without any forms lost or embarrassment caused to colleagues. The purpose of the questionnaire is to collect the users’ reflections and points of view so that the system can be improved. The questions evaluated three goals: Usability, Learnability, and Functionality. Each one of the three goals has at least two questions.

#### 7.4.1 General questions:
The researcher divided the questionnaire into several sections. Usually, the first section contains identification or general questions. The purpose of these questions is to identify the participant’s information. There are no personal questions, however, but rather questions to indicate the participant’s position and the level of computer skill so that the researcher will be able to more effectively analyse the questionnaires. The general section contains the following questions:

- What is your gender?
- What is your position?
- What is your level of computer skill?
• Have you used any applications before?

7.4.2 Usability:
In the second section, the researcher concentrates on usability questions. The purpose of these questions is to reflect on the user’s feedback for the system design and concept. These questions help the researcher to improve on any shortcomings in the design or system information. The usability section contains the following questions:

• How do you evaluate the application?
• How do you evaluate the system design, graphics and organisation?
• How do you evaluate the amount of information and structure for each page?
• Would you prefer a different style for the website lay out? Why?

7.4.3 Learnability:
Learnability is one of the more significant sections in the evaluation. This section reflecting on how easy or complex it is to use the system. These reasons are very important for achieving the system goals. Consequently, the researcher looked at the answers to these questions very carefully to change and avoid any system complexity. The Learnability section contained the following questions:

• How do you evaluate the level of application difficulty?
• How easy was it to perform all the tasks in the system?
• Could you specify the most difficult parts of the system?

7.4.4 Functionality:
In the second section, the researcher looked at the quality of the functionalities and how the users could interact with them. It was also investigated whether the users faced any errors that the developer had not anticipated. In addition, users were asked to make suggestions for improving the system. The functionality section contains the following questions:

• Does the system meet the requirements?
• Did you spot any errors or problems in the system’s functionality?
• If yes, could you specify where you faced them?
• What change(s) would you suggest, if any, improve the system?
• Could you please write recommendations for improving the application?
7.5 Analysis of results:

After the users had completed the questionnaires, the researcher started to analyse the results. Firstly questionnaires were sorted to identify each group (Managers, Experts and Teams). After that, the results of three goals were analysed: Usability, Learnability, and Functionality.

Learnability:

When the researcher started to analyse the results of this section, it was clear that the Learnability result was based on several questions. These questions measured the level of computer skill and the experience of using applications. Skill level assessment would actually help the developer to improve the system by making it more suitable for the Teams who would deal with it directly. As mentioned in the Design chapter, there are a number of system users who had not used a computer before.

As shown in Figure 36, the questionnaire illustrated that the majority of questionnaire samples had an ‘average’ level of basic computer skill. This was followed by the ‘beginners’ who comprised 33% of the sample. This result proved that system users generally had substantial experience in terms of basic computer skills.

Moreover, in terms of previous experience of using systems, 81% had already used another application, which reduced the training effort and reflected the ability to use the system to help them in their tasks and to support the developer in achieving the system goal.
Figure 37 shows that 52% of the sample found the system easy to use and clear while 38% said the system was moderately easy to use.

It is clear from the above figures and statistics that the system is generally easy to understand, learnable and that people are willing to use it.

Usability:

After the Learnability section had been analysed, the developer started to analyse the questions highlighting the usability of the system. As mentioned above, the Usability section is divided several questions. The answers to these questions measured the system Usability.

Figure 38 show that 57% of participants said that the system design was good and 19% of participants said the design was satisfactory. Some participants commented in the questionnaire that they did not like to deal with links in the Team main page and it would be better to change links to buttons or images so they could be made clearer, especially for beginners.
Furthermore, there were a number of experts who said it would be better if the developer followed the MOWE design and standards in terms of code. However, the language and web application was new for the developer, which would have meant expending a lot of time and effort to learn the basics and then follow the structure established by experts who had already had long experience in this field. But in general the developer was followed the standards interfaces designs as much as he can to be at least same MOWE interfaces standards.

- The information amount and the structure for each page was:
  - Suitable: 15 (71%)
  - Need more: 4 (19%)
  - Confusing: 0 (0%)

Figure 39: System concept

System information and structure comes under Usability evaluation. 19% of the participants found that the system did not provide enough information and needed more. Some participants commented in the questionnaire that directions and problem location needed technical information about the area infrastructure as well as location information and they added that the map was insufficient for finding the location. In fact, this aspect was managed by the GIS system and the GIS database includes all this information which will be covered in future work.

According to the above results, 71% of participants said the system information was suitable for them and 57% also found the design to be good. Therefore, the usability of the design was satisfactory for the participants.

**Functionality:**

In the last section of the Evaluation, the results reflected the points of view of participants. Firstly, the researcher focused on the Requirements question.
The majority of the answers were positive. However, 14% of participants said the system met some of the requirements but needed more functionality, such as contact function between the Administrator and Teams. They also stated they would like to see the most common elements presented as a list so they would not have to repeatedly write by them.

Furthermore, in terms of errors and application of functionalities, 90% of the results were in the negative, which means that almost all functionalities worked perfectly with no errors see Figure 41.

Finally, they all agreed on using and developing the system to improve its work and publishing it as soon as all its functions were complete so that they could implement it to help control the Team and tasks.

7.6 Recommendations:
Some feedback commented on the design and recommended some functions found to be reasonable, including:

- In terms of the administration section, some participants said it would be better if problem locations could be displayed to each Team simultaneously.
The summary should also divide the level of risk (High, middle, Low), each in a special table.

In terms of Team management, the system should display Teams and Team members in one table.

The system should display urgent tasks in red or flashing so they can be easily distinguished.

Ultimately, the recommendations are important. However, due to the limited amount of time allocated for the project, it was not possible to implement all features. The aim of the project was to illustrate and implement the application with some features so it would be ready to use in the real world and easy to improve in future.
Chapter 8: Project Conclusion and Future Work

8.1 Achievements:
Working on this project was a kind of challenge as it required learning and working with new programming languages, i.e. developing web applications by using PHP, MYSQL databases. Applying the system features took a great deal of effort.

However, this challenge helped to produce the system required for achieving the goal. There have been many objectives and steps in this dissertation that led to the achievement of these goals. The following points describe these objectives and steps:

- In Chapter 1, the developer previewed the Literature Review and the previous researcher’s point of view on the systems environment, as well as the benefits of using the system to improve the organisation’s work. In addition, programming languages were previewed to determine which the best to use in implementing this project were. The main system functionalities were also discussed in terms of the researcher’s experience and expertise.

- After the developer had given the background to the system and features mentioned above, he then moved on to the Requirements stage. Here, the developer concentrated on the best methodology for helping to control the system development and which would also be suitable for the system environment.

- In Chapter 3, the system design was taken into consideration. This was an important stage in the dissertation. The author focused on issues such as human-computer interaction in order to design the interface and concentrated on the level of user who would be served by it, each with a different level of knowledge and skills. In addition, the developer chose the system structure that met the system goals to aid the author at the Development stage.

- In terms of the Development stage, the developer wrote the Implementation strategy, taking into account many aspects that had been analysed at the Design stage. Also illustrated were the implementation of system functions in terms of the algorithms and code to aid implementation of the system function.

- At the end of the Implementation stage, the developer tested the system with the target user at the Ministry of Water and received feedback. After that, he analysed the evaluation results that showed satisfaction with the system. The evaluation results were
very satisfactory, as shown in the Evaluation chapter. The comments obtained from the users only recommended a few additional.

8.2 Project management:

The Project Management issue came into each phase of this dissertation. In fact, it was well planned, structured and organized, and allowed the project to be completed within the expected timeframe. The aims and objectives at the beginning of this project aided the author in achieving the goals by the specified date.

RUP methodology was chosen for this project. This methodology was very supportive in identifying the developer and where he should start developing the system, and the strategies to follow in order to achieve the target goal of the Development phases. In addition, its best aspect was the connection and agreement between the users and developer at the end of each phase of the development.

During the System Implementation stage mentioned earlier, development required and consumed more time and effort for learning the new programming languages and applying the system features. There were also a few obstacles to implementation encountered. The extra time spent led to a few delays which affected the timeframe allocated for extra feature implementation but this did not affect the overall target delivery date.

Regular meetings with the supervisor enabled the author to keep track of the project and the development of the system. She advised me on how to implement the system features also pointed out any gaps or problems in the system that only experts would notice. These suggestions were very helpful for building the system in a way which would permit the goals to be achieved.

8.3 Conclusions:

Generally, the developer was satisfied with the improvements to this system, although it was emphasized that further developments could have been applied. As already clarified, the technology and languages used during this project were new but broadened the author’s knowledge while implementing the system. The author now also has a greater understanding of what is necessary for developing a system of this size, and feels that he has have developed greater time management skills.

Ultimately, the author feels that the project aim of producing a system that allows the management and control of Technical Teams has been met. Consequently, this system could be applied and used in the real world for these purposes.
8.4 Recommendations for Future Work:

There were many other potential developments for this application. However, due to the time limitations for this project (just 3 months approximately) these additional features were not developed. The following list describes possible features for addition to this system when it returns to MOWE.

- **Mobile application:**

  This feature allows the Team to receive the tasks by mobile phone, without the need to log into a PC. This application will also connect to mobile GPS to follow the task route without the need for any other tools, e.g. TOMTOM. The idea is to receive the coordinates and task information from the application by synchronizing the mobile with the main system to update the tasks at the beginning of each day. SMS message information could also be analysed, sending task coordinates to mobile GPS.

- **Sending SMS Messages:**

  This feature allows the Administrator to send SMS messages via the system. The Administrator system has an SMS option, where the message body can be filled. The appropriate Team is then selected and the message sent. In addition, the system could be configured to identify a high risk level task and automatically send it to the Team as an emergency.

- **Teams Tracking:**

  Through this feature, the Administrator system can track a Team’s location by receiving a GPS log file every 30 minutes from a mobile GPS display of the location on the Google maps. The Administrator can also send a request to the mobile system to receive live GPS log files the using mobile GPRS service (General Packet Radio Service) to display the location live.

- **System Integration:**

  When the system is applied in the field, it will integrate with two existing systems. The first integration is with the GIS system. This integration needs to replace the location function of the Google map in order for the Ministry of Water’s GIS system to be used. The second integration will be with a HR application (Human resource application). This integration needs to change the Team management by connecting with the real employee’s information.
Chapter 9: References:


Economou et al., 2008. ‘Cultural applications for mobile devices: Issues and requirements for authoring tools and development platforms’, *Mobile Computing and Communications Review*, vol. 12, No. 3.


Google API, Google Maps web services,


Jacobs et al., 1991. „The application of a novel two-way mobile satellite communications and vehicle tracking system to the transportation industry,” IEEE Transactions on Vehicular Technology, 40.

Janecek A & Hlavacs, H., 2005. „Programming Interactive Real-Time Games over WLAN for Pocket PCs with J2ME and .NET CF”, Hawthorne, New York


Appendices:

Appendix A: Database code:

```
CREATE DATABASE team_project;
-- Table structure for table `department`
CREATE TABLE IF NOT EXISTS `department` (  
    `Dept_code` varchar(25) NOT NULL,  
    `dept_name` varchar(25) NOT NULL,  
    PRIMARY KEY (`dept_name`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- Table structure for table `employee`

CREATE TABLE IF NOT EXISTS `employee` (  
    `Emp_id` int(25) NOT NULL,  
    `First_n` varchar(25) NOT NULL,  
    `Last_na` varchar(25) NOT NULL,  
    `Date_of_birth` varchar(25) NOT NULL,  
    `Mobile_nu` varchar(25) NOT NULL,  
    `Email` varchar(25) NOT NULL,  
    `Dept_code` varchar(25) NOT NULL,  
    `Position` varchar(25) NOT NULL,  
    PRIMARY KEY (`First_n`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- Table structure for table `Task`

CREATE TABLE IF NOT EXISTS `gen` (  
    `Emp_id` int(25) NOT NULL,  
    `First_n` varchar(25) NOT NULL,  
    `Last_na` varchar(25) NOT NULL,  
    `Date_of_birth` varchar(25) NOT NULL,  
    `Mobile_nu` varchar(25) NOT NULL,  
    `Email` varchar(25) NOT NULL,  
    `Dept_code` varchar(25) NOT NULL,  
    `Position` varchar(25) NOT NULL,  
    PRIMARY KEY (`First_n`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```sql
-- Team Monitoring application

```
106

-- Msc Software Engineering

-- PRIMARY KEY ('order_id')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=197 ;

CREATE TABLE IF NOT EXISTS `parts_request` (  
  `Part_request_no` int(25) NOT NULL AUTO_INCREMENT,  
  `Task_id` varchar(25) NOT NULL,  
  `Order_type` varchar(25) NOT NULL,  
  `Importance` varchar(25) NOT NULL,  
  `Quantity` varchar(25) NOT NULL,  
  PRIMARY KEY (`Part_request_no`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=12 ;

CREATE TABLE IF NOT EXISTS `team` (  
  `Team_name` varchar(25) NOT NULL,  
  `Team_phone` varchar(25) NOT NULL,  
  `Team_leader` varchar(25) NOT NULL,  
  `Main_task` varchar(25) NOT NULL,  
  `Second_task` varchar(25) NOT NULL,  
  `T_email` varchar(25) NOT NULL,  
  PRIMARY KEY (`Team_name`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `users` (  
  `Emp_id` varchar(25) DEFAULT NULL,  
  `User` varchar(25) NOT NULL,  
  `Password` varchar(25) NOT NULL,  
  `Access` varchar(25) NOT NULL,  
  PRIMARY KEY (`User`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```
Database Tables descriptions:

- Users table:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emp_id</td>
<td>varchar</td>
<td>This field holds the employer number.</td>
</tr>
<tr>
<td>User</td>
<td>varchar</td>
<td>The user names of the system</td>
</tr>
<tr>
<td>date</td>
<td>varchar</td>
<td>The date of the tasks.</td>
</tr>
<tr>
<td>Access</td>
<td>varchar</td>
<td>This field about the access level of the users (admin, Team)</td>
</tr>
</tbody>
</table>

- Task table:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>order_id</td>
<td>int</td>
<td>This field is primary key of this table and its holds the task number.</td>
</tr>
<tr>
<td>Location</td>
<td>varchar</td>
<td>This field stores the tasks location.</td>
</tr>
<tr>
<td>Status</td>
<td>varchar</td>
<td>The tasks status.</td>
</tr>
<tr>
<td>Description</td>
<td>varchar</td>
<td>This field contains the tasks description.</td>
</tr>
<tr>
<td>Problem_type</td>
<td>varchar</td>
<td>The problem type.</td>
</tr>
<tr>
<td>customer_num</td>
<td>varchar</td>
<td>The customer phone number.</td>
</tr>
<tr>
<td>Risk_level</td>
<td>Varchar</td>
<td>The request risk level.</td>
</tr>
<tr>
<td>Team</td>
<td>Varchar</td>
<td>The team who has the request.</td>
</tr>
<tr>
<td>Parts</td>
<td>Varchar</td>
<td>The part order.</td>
</tr>
</tbody>
</table>
### Team table:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team_name</td>
<td>varchar</td>
<td>This field is primary key. It’s to store Teams name.</td>
</tr>
<tr>
<td>Team_phone</td>
<td>varchar</td>
<td>Team contact number.</td>
</tr>
<tr>
<td>Team_leader</td>
<td>varchar</td>
<td>Team leader that comes from employee table.</td>
</tr>
<tr>
<td>Main_task</td>
<td>varchar</td>
<td>The main task of the Team.</td>
</tr>
<tr>
<td>Second_task</td>
<td>varchar</td>
<td>The second task of the Team.</td>
</tr>
<tr>
<td>T_email</td>
<td>varchar</td>
<td>Teams e-mail.</td>
</tr>
</tbody>
</table>

### Employee table:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emp_id</td>
<td>int</td>
<td>This field is primary key. It holds the employer number.</td>
</tr>
<tr>
<td>First_n</td>
<td>varchar</td>
<td>The First name of the employer.</td>
</tr>
<tr>
<td>Last_na</td>
<td>varchar</td>
<td>The last name of the employer.</td>
</tr>
<tr>
<td>Date_of_birth</td>
<td>varchar</td>
<td>The employer date of birth.</td>
</tr>
<tr>
<td>Mobile_nu</td>
<td>varchar</td>
<td>The employer mobile number.</td>
</tr>
<tr>
<td>Dept_code</td>
<td>varchar</td>
<td>The employer department code.</td>
</tr>
<tr>
<td>Position</td>
<td>varchar</td>
<td>The employer Position.</td>
</tr>
</tbody>
</table>
• Parts order table:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part_request_no</td>
<td>Int</td>
<td>This field is primary key. It holds the employer number.</td>
</tr>
<tr>
<td>Task_id</td>
<td>varchar</td>
<td>Task number.</td>
</tr>
<tr>
<td>Order_type</td>
<td>varchar</td>
<td>The order type.</td>
</tr>
<tr>
<td>Importance</td>
<td>varchar</td>
<td>The priority of order.</td>
</tr>
<tr>
<td>Quantity</td>
<td>varchar</td>
<td>Quantity</td>
</tr>
</tbody>
</table>

• Department table:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dept_code</td>
<td>varchar</td>
<td>This field is primary key. It holds the Department number.</td>
</tr>
<tr>
<td>Dept_name</td>
<td>varchar</td>
<td>The department name.</td>
</tr>
</tbody>
</table>

**Application code in the CD**  
**Application installation and development:**

The application is web-base so it doesn’t need more technical requirements just need web browser.

However, for developer, it’s need the following tools:

- Apache server.
- My SQL database.
- PHP Editor.

There are a recommended tools such as XAMPP and appserv. These packages contain Apache server and My SQL database.
Appendix B: System interfaces screen shouts:

Administrator interfaces:

1. System login page.

2. Administrator main page:

From this page, administrator can monitor the team progress and also edit and delete exist tasks.
3. **Add users and Teams:**
   Administrator can add system users and Team from this page

4. **Generate new task:**
   Administrator can create new task from this page and send mail.
5. **Team management:**
Administrator can delete Teams, and show the Teams details.

6. **Employees management:**
Form this page can show the teams members and the administrator can add and delete member.
7. **Add new member:**

![Employee management form]

8. **Request Location:**
   From this page the users can show the task location.
9. **System report:**
Administrator can show the active report. The administrator can search by Team name, Problem type, statues and risk level.

<table>
<thead>
<tr>
<th>Team name</th>
<th>problem type</th>
<th>Location</th>
<th>Date</th>
<th>Status</th>
<th>Description</th>
<th>Rank level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traks</td>
<td>Maintenance</td>
<td>BH11 1NT</td>
<td>19/07/2010</td>
<td>Open</td>
<td>The main hole is broken</td>
<td>Low</td>
</tr>
<tr>
<td>Traks</td>
<td>Maintenance</td>
<td>BH11 1BE</td>
<td>21/07/2010</td>
<td>Open</td>
<td>maintenance station need maintenance</td>
<td>High</td>
</tr>
<tr>
<td>Traks</td>
<td>pipe</td>
<td>BH22 1GU</td>
<td>21/07/2010</td>
<td>Pending</td>
<td>sewage station stopped</td>
<td>High</td>
</tr>
</tbody>
</table>

**Teams interfaces:**

1. **Team main page:**
From this page, Teams can show their tasks and also browse the task location by map.
2. **Team request**:
   Teams can update task from this page and also order parts.
Appendix B: Questionnaires forms

Requirements Questionnaire form:

Teams Project questionnaire

According to the interviews between the developer and the Ministry of water employees and the virtual interfaces that the developer designed, the scenario below illustrated the business of the application in terms of the system administrator and Teams.

1) administrator.
   - Login to the system and show the summary of the latest tasks.
   - The administrator Generate new task and the system assign it to the specialist Team and show the location of the task.
   - If the specialist Team have already 5 tasks then system assign it to the backup Team.
   - If the backup Team have already 5 tasks then system send it to the waiting list.
   - The system will assign the task after one of the specialist Team is available.
   - The administrator can update and delete the tasks.
   - The administrator can add users, add teams and manage them.
   - The administrator can show the live report for each team.
   - The administrator can contact with Teams by e-mail or SMS.
   - The administrator can Track the Team in Google map.

2) Team
   - Login to the system and show the tasks by PC or Mobile.
   - Teams can Show the tasks location by Google map.
   - Teams can update the tasks and also add description.
   - Teams can Order parts from the system.

What is your position?
- Manager
- Team leader
- Technician

Does the application scenario is Clear and understandable?
- Yes
- No

If No, could you please could you specify the Vague parts of the scenario?

How do you evaluated the application interfaces?
- Poor
8/19/2010

Teams Project questionnaire

- Satisfactory
- Good
- Very Good

How easy was it to do all the tasks in the system interfaces?
- Easy
- Average
- Difficult

If its difficult Could you specify the difficult parts of the system?

Could you please write suggestions to improve the System features and interfaces?

Submit

Powered by Google Docs

Report Abuse • Terms of Service • Additional Terms
Evaluation Questionnaire form:

Project Questionnaire

The main objective of The Planning and Monitoring the Technical Groups project is to develop a web application that aids Administrator and Team leaders to organizing and evaluating their group work.

What is your gender?
- Male
- Female

What is your position?
- Manager
- Team Leader
- Expert
- Technician

What is your level in the computer skills?
- Beginner
- Average
- Advance
- Expert

Did you use any application before?
- Yes
- No

How do you evaluate the application?
- Poor
- satisfactory
- Good
- Very Good

How do you evaluate the application difficulty?
- Difficult
- Moderate
- Easy

The system design, graphics and organisation were:
- Very Good
- Good
- Satisfactory

spreadsheets0.google.com/viewform?hl...
8/19/2010

Project Questionnaire

Does the system meet the requirements?
- Yes
- Some of them
- No

The information amount and the structure for each page was
- Suitable
- Need more
- Confusing

How easy was it to do all the tasks in the system?
- Very Easy
- Easy
- Average
- Difficult
- Very difficult

Could you specify the difficult parts of the system?

Did you spot any errors or problems in the system’s functionality?
- Yes
- No

If yes could you specify where did you face?

Do you prefer a different style for the website layout rather than this one? Why?
Project Questionnaire

What change(s) would you suggest, if any, to make the system better?

Could you please write recommendation to improve the application?

Submit

Powered by Google Docs

Report Abuse • Terms of Service • Additional Terms

spreadsheets0.google.com/viewform?hl...