

Final Report

Name: Meagan Vaughan

Student Number: 1026713

Title: Assessing the Effectiveness of the Mobile Platform
for Market Research

Abstract

Mobile devices are becoming an ever increasing part of our daily lives, combined with social networking this technology could revolutionise the market research and feedback processes for businesses. In this report I will assess the various processes and technologies that can be adopted, along with a working prototype to prove that innovation within feedback mechanisms can bring potential to businesses wishing to understand their customers needs and ultimately improve their services.

Acknowledgement

I would like to thank the following people for their support in my dissertation:

- Wendy Ivins: My supervisor, who gave the best support possible! Thank you!
- James Elsey: For developing the parts of Roosearch Research I could not.
- Andrew Jones: For your useful comments in my interim report.

Contents

Abstract

Acknowledgement

Contents

1.0 Introduction

2.0 The Planned Approach

3.0 Project Planning

3.1 Planning Documents

3.1.1 Time Plan

3.1.2 UML Requirements Document

4.0 Design

4.1 Design Documents

4.1.1 Class Diagram

4.1.2 Sequence Diagram

4.1.3 Medium Fidelity Prototype

4.2 Design Justification (Medium Fidelity Prototypes)

4.2.1 Iteration One

4.2.2 Iteration Two

4.2.3 Iteration Three

5.0 Implementation

5.1 Technologies

5.2 Data Modelling

5.2.1 Data structure (DDL)

5.2.2 Queries (DML)

5.3 Meeting and Communications

5.4 The Prototype

6.0 Testing Evaluation

6.1 Test Cases / Ad Hoc Testing [6]

6.2 Unit Tests

6.3 Test Results And the Effects

7.0 Validation Interviews

7.1 About the Restaurants and Customers

7.2 Interview

7.3 Results

7.3.1 Restaurant

7.3.2 Customer

7.4 What Does This Mean For Roosearch Research

8.0 Evaluations

8.1 Prototype

8.2 Approach

9.0 Future Work

10.0 Reflections on Outcomes and Learning

11.0 Conclusion

12.0 Glossary

13.0 References

1.0 Introduction

The level of technology has developed rapidly over the past number of years and is now part of everyone's daily lives. However, market research has failed to use this to its advantage, as it does not incorporate technology to target the audience required, at the same rate. There is a significant gap between technology development and market research integration. This was the conclusion reached in the interim report and was the foundation of this final report. The requirements, designs and basic prototype will demonstrate how market research could be developed as an application for smartphones, primarily the Android platform. The interim report was able to prove that the hypothesis was theoretically possible; however in this report the aim was to prove or disprove that this is possible in practice.

I met the objectives in the interim report by strictly following the time plan I had in place, talking to my developer and asking myself the following questions:

- Is it possible for mobile applications to be used to collect a range of different types of market research?
- Can this be completed on mobile applications alone, or is a web based management system needed?
- How will I carry out the outsourcing?
- Is there more than one way to integrate market research and mobile technology?
- What are the requirements for such as application?

These questions were developed from my three objectives to ensure that they were met fully and ensure that the interim report was purposeful. My time plan guided me in what I needed to be completing each week and what meetings I should be having with both my developer and supervisor. My developer then aided me in achieving my objective by discussing the development with me to ensure that what I was proposing was suitable for him to complete.

The subsequent chapters of this report will address the following areas:

- The approach that was taken in order to ensure that the development ran smoothly and I was in control and making all the decisions regarding the development and design of Roosearch Research.
- The planning that was undertaken to ensure that the developer knew what I wanted to achieve from the system, the features it had to have and when his deadlines where for each component within the overall development.
- The designs that were further developed from the requirements document developed in the interim report. These include newly developed structural diagrams and designs that the developer had to follow.
- Implementation details are to be addressed below, showing the table development and queries I developed and the technologies that I decided on using for the development. Communication details can also be seen here along with some images of the prototype I designed and was managing.
- The evaluation of the system refers to the testing. It was tested by myself using test cases and by my developer using unit tests. The results can be seen in this section and the impact of these results on the prototype.
- Interviews were then conducted to validate that the target audience would use the application and at the same time we were able to get their opinions. The results of this can be viewed in this section along with details of the questions asked.
- After Roosearch research was completed and tested I evaluated the approach but comparing it to that which was planned and did the same for the prototype. The results of this are in the evaluation section.
- Knowing that the process or prototype did not go as planned, I sat down and reflected on what was let to complete and what else could be completed in the future that was not originally planned. There were numerous features for future work that I have written up in the future work section.
- My report has been concluded with a reflection on the outcomes, achievements and lessons I have learnt whilst researching, planning, designing and managing the prototype.

Based on what I want to achieve, I was able to develop a strong aim for my project in the initial report. Within this report I will be trying to prove or disprove this is practically possible and through planning and developing a prototype application, to validate this.

“The aim of my project is to uncover if mobile phone applications can be developed to help businesses, with consumers, primarily restaurants, in the collection of market research.”

The objects for this report, determined in the initial plan, are:

1. To have the design of the application finished by the end of week 11, in the autumn semester, so I can plan the first iterative.
2. Begin the outsourcing in the week commencing 31st December 2012.
3. To have two meetings per iteration to ensure progress is going well.
4. To have completed the development of the application and write-up regarding the details and justification of choices made, along with a progress report on how the development process progressed. This must be completed by the end of 31st March.
5. My aim is to have the final report finished by start of week 10, to allow time for proofreading and correction.

2.0 The Planned Approach

From the interim report you will be aware that an iterative approach was taken. To expand on the reasons for this decision I have expanded on the original justifications: [1]

1. Faster time to market - the development can be complete faster and brought to the market sooner allowing competitive advantage, compared to native application development cycles.
2. Easier cross platform compatibility - allows the user to right one programme that can then be built and deployed into numerous platforms.
3. Easier to upgrade, maintain and deploy - due to the cross platform compatibility to is easier to maintain upgrade and deploy as the code only has to be adjusted once and it then makes the changes required.
4. Still have access to native features - in application development the developer can use HTML5 properties alongside native features.
5. Can complete planning and basic designs up front, with the view of editing these throughout the iterations if the client requires this.
6. At the end of each iteration there is typically a working piece of software for the client to look at and provide feedback on.

We took a traditional iterative approach. There were three, three week iterations in which time my developer got as much of the development finished as possible. As this is a prototype to demonstrate what can be done, it was more important that the functionality was there and less important was the design aspect as this can be seen in the designs. Iteration three designs were the final designs. Basic planning was undertaken at the beginning and these documents / models were edited throughout the development process where I saw fit, this included the documents from the interim report requirements documentation.

You will have seen on my interim report that I have outsourced my project to an experienced developer. I felt this was the best option for myself as I would be able to project manage the development which enabled me to develop my skills in this area for when I graduate. There was close contact between the developer and myself which helped ensure that the progress was on track. This was primarily in the form of reports, emails and meetings.

The purpose of iteration one was to get the basic structure built. The structure of the web based system was to be completed, including the basics of the mobile application questionnaire feature. The QR code was planned for implementation here but the structure took longer than expected to

complete and so ended up being developed in between iteration one and iteration two. There was a lot to complete in this iteration and we both knew that there was a chance it could run into the next iteration, but with the week in-between the iterations we were able to start iteration two with a clear slate, on the date stated on the time plan, the 28th January 2013. The requirements for this specification were given to the developer on the week commencing 31st December 2012.

Iteration two was supposed to be used for brushing up on the designs and implementing the Facebook and Twitter features, along with a working login system. However, we had also planned it so that could be used to finish off any leftover work from the first iteration. This was not needed in the end but needed to be included in the plan as I had to be flexible with the developer as I was working on my project on top of working full time. Everything went to plan in this iteration apart from the login which was supposed to be completed through Facebook and Twitter, but due to complications (which are described in the prototype evaluation) these features had to be descope in the end and a mock-up log in system created instead, which ate into iteration three, which started on the 25th February 2013.

Iteration three had a number of purposes. The main purpose was to ensure all of the above work had been implemented, before the additional features were created. The additional features that we managed to include in the time left in iteration three included: graphs for each question instead of a reports page and a more suitable QR code feature that generated the QR code within the website, instead of outside. As time was running out and there was little time to create the reports page I decided and instructed the developer to implement the graphs on the question pages of each questionnaire. This saved the time that was then needed to implement a more suitable QR code before the end of development on the 17th March 2013.

A contingency week was planned for week eight. This was to allow my developer extra time which was needed to tidy up what had been completed. It was an important feature to include as the development I planned was large for the time frame that was available; my developer also had a scheduled week off in the second iteration which meant that the time frame was even tighter, but he managed to complete the vital task in the three iteration and contingency week.

Please refer back to the Interim Report for any details related to the planned approach and the iterative decision. Following the development of my approach I started to work on the planning and design documentation; details of which can be seen in the following sections.

3.0 Project Planning

Planning started back when completing the Interim Report in the Winter Semester. During that time, and for that report, I developed a UML Requirements Document which contained a use case model and description, functional and nonfunctional requirements, testing and validation information, any assumptions made and any foreseeable constraints and limitations. I then added a more accurate time plan to this to ensure that the developer and myself knew what we had to do each week. Below I have stated and explained why these are suitable for my project.

3.1 Planning Documents

3.1.1 Time Plan

A time plan was used to indicated what needed to be completed each week, it is not overly detailed, just enough for us to follow and for me to know what I had to do after the development finished. There was very close communication between the developer and myself, which supports my claim of why there is not as much information needed as there would be in the likes of a Gantt chart or general more detailed time plan.

The plan was completed in Microsoft Excel in a week by week format. It details the university week number, the start date for the week, the tasks that need to be completed within that week, what the developer should have been doing and when I should have been having a meeting with my supervisor. Having all this information in one place made it a lot easier to see what needed to be completed and the preparations that I needed to be completing for the following week.

A Gantt chart could have been developed in place of the time plan. Although a Gantt chart is a common and popular technique for representing phases, milestones and dependencies I felt that it was not necessary for this project. A time plan was sufficient in displaying the information needed and is much easier to follow than a Gantt chart. This did not suit my project for two reasons: A) they are not suited for conveying complex dependencies and B) they are not suitable for projects that have potential variability in the completion dates.

The time plan that was followed by my developer and myself; can be seen in Appendix A. Refer to the glossary for further details on Gantt charts.

3.1.2 UML Requirements Document

As stated above, a UML requirements document was developed by myself for the interim report. It contained the main functional requirements that were developed from a Use Case Diagram and Use Case models. There was also details on the desirable nonfunctional requirements and the limitations and constraints present in the project.

A requirements document based on UML is a planning document that details all of the initial requirements of the project, including the other features stated above. I used this document as a planning document and checklist for creating the time plan to ensure that of the development side was covered. It was then passed to my developer up front, before the start of iteration one, with my designs for this iteration. This help to ensure my developer knew what he had to do and the guidelines for design that he had to follow.

To develop this I started by looking at what I wanted to develop, from here I started to fill a use case diagram. This took awhile to complete and a bit of editing to create a use case that reflected what I wanted built. Once this was finished I started on completing use case description based on the use case diagram, so that the developer would have the detail he needed to built what was required of him. The functional requirements were those stated in the use case diagram and so I had to develop the nonfunctional requirements that I felt were necessary for the application and following this identify the limitations and constraints of the system. The limitations and constraints identification was important as it allowed me to see that the constraints will affect the data that we can hold due to the Data Protection Act and in turn this will affect the structure of the system.

A UML requirements document was chosen as my form of planning because it provided all the information that was needed by my developer and myself. It also provided a central point for this important initial data to be stored. The main reasons for using UML in the requirements documentation was that it is universally understood, I had experience in developing such documents and my developer has worked with UML components before and so had the experience needed to understand what they were showing.

The UML Requirements Documentation can be seen in Appendix E of the Interim Report.

With both of the planning documents developed before the start of iteration one I next had to start of the design documents which would be needed for this iteration. The details of these can be seen in the next section “Design”.

4.0 Design

4.1 Design Documents

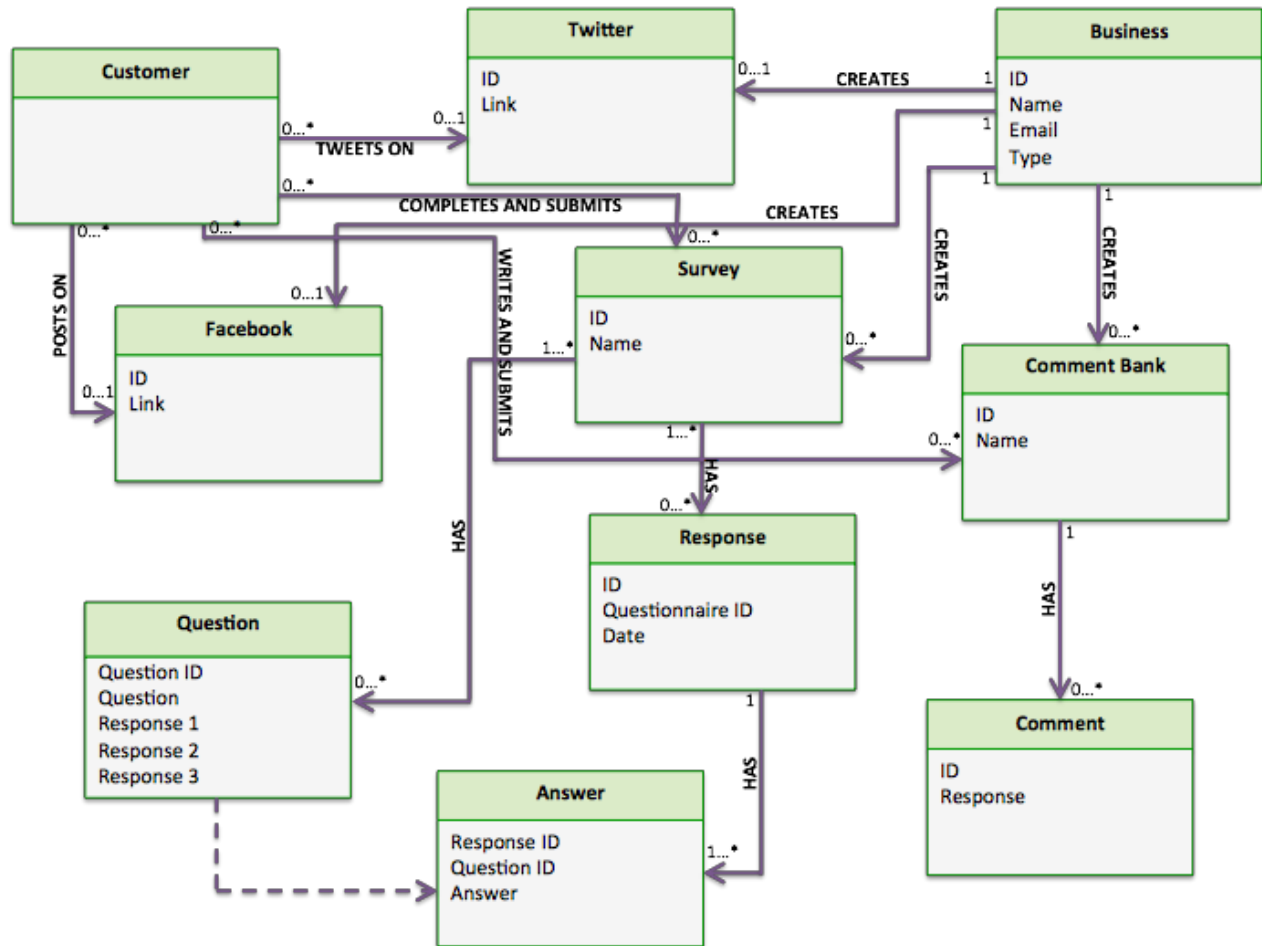
4.1.1 Class Diagram

A UML (unified modelling language) class diagram is a structure model describing the structures of systems through classes, attributes, operations and relationships. It can be used for conceptual modelling of systematics or for detailed modelling allowing the models to be translated into programming code. Class diagrams can be used for data modelling and this is what I have been using it for; to help portray the data structure of the web based management system as the classes show what needs to be programmed and to visually display the interactions in the application. A class diagram was chosen based on the fact that the other parts of UML were designed to aid the developer and as such a class diagram fitted in. However, an entity relationship diagram could have been chosen as these represent almost the same information.

I created this in Microsoft PowerPoint. Unlike the sequence diagram and medium fidelity prototypes the class diagram was created with all aspects of the activities and relationships in it; the design documents were created at the start of each iteration. I started by creating the classes and their attributes. Following this the different types of relationships were created and shown.

The class diagram was used by myself to complete the ad hoc testing and as an aid in developing the table design and data modelling queries. However, my developer made more use of this as it was the basis of the structure that he had to code. He used it through the iterations for guiding him on what he had to create and if possible how they were linked.

An entity relationship diagram is a technique used in the wider business community in addition to the information technology industry; it enables the visual portrayal of how information is stored in business systems. Basically, it is an abstract and conceptual way for describing a database and data. I decided it was not appropriate to use an entity relationship diagram as I have better skills in developing class diagrams and my developer primarily uses class diagrams over other options, such as entity relationship diagrams. A class diagram was more preferred than an entity relationship diagram; as entity relationship diagrams have more of a limited scope, and as so are regularly confined to modelling relational databases rather than whole applications. [2]



A clearer version of the Class Diagram can be referred to in Appendix B. Refer to the glossary for more details in both class diagrams and entity relationship diagrams.

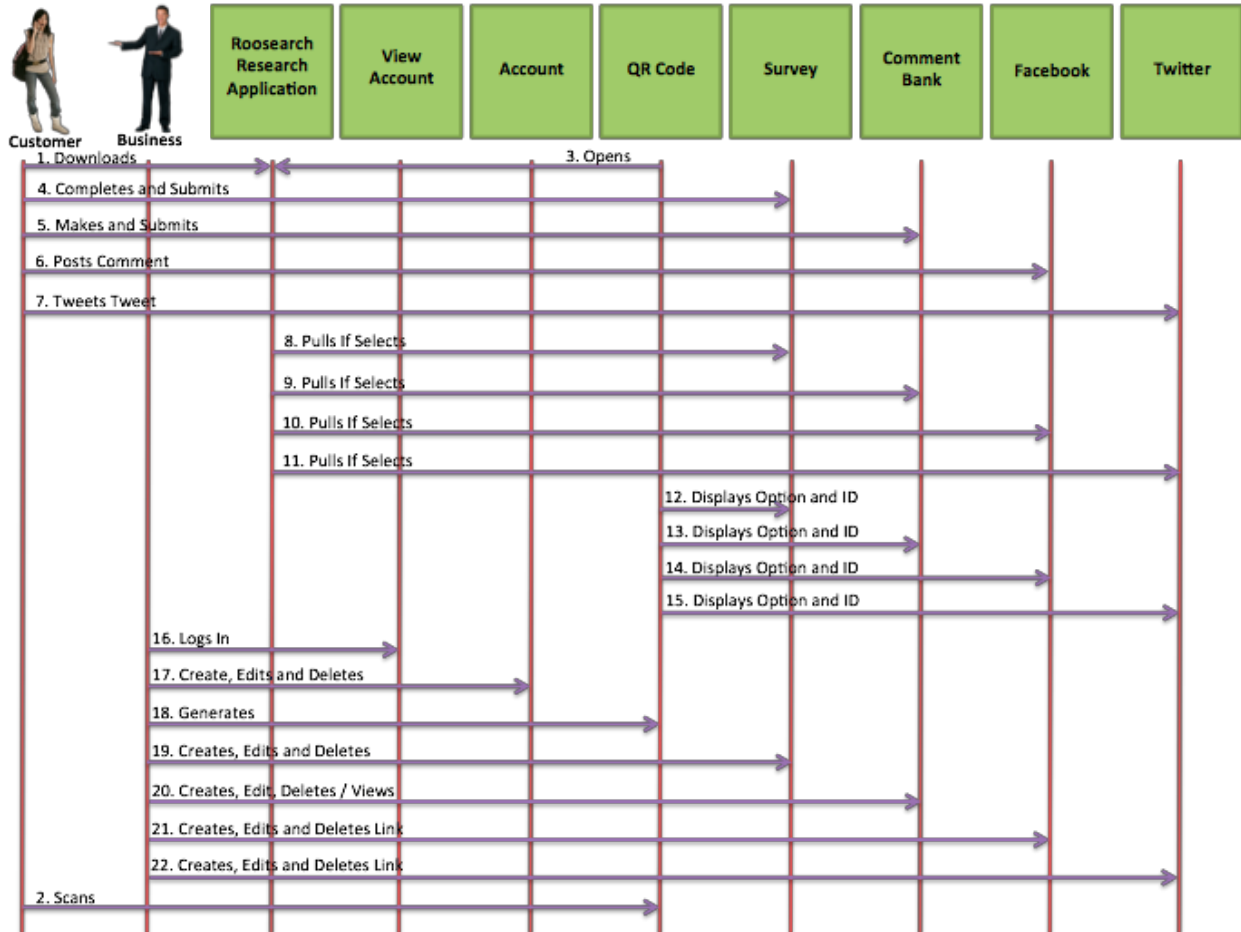
4.1.2 Sequence Diagram

Sequence diagrams are another component of UML, a form of interaction diagrams and used to show how operations are carried out. A sequence diagram depicts objects and classes involved in the scenario and the sequence of messages exchanged between the objects which allows the developer to carry out their functionality, according to the scenario. This was used to show the interactions between the different components in the market research web based management system. It fitted in well with the other UML models to allow the developer to work effectively and without confusion.

This high level interaction model was developed iteration at a time, the first iteration is developed and as the iterations develop and new activities are being planned for introduction, the sequence diagram was updated to allow the developer to understand the interactions and messages to be sent between components.

The sequence diagram was used more by the developer than myself. I found that the medium fidelity prototypes, class diagram and requirements document were enough for me to complete the jobs I needed to do, such as test cases. However, the developer found this model really useful in getting his head around the project and how the different components link together. Although he did not use this throughout the development, it was a good reference for the first iteration.

User stories could have been used in place of a sequence diagram, however they would not provide the detail of visual stimulation that was needed. A sequence diagram allowed my developer to look and visually see the interactions and messages, whereas user stories would have left my developer having to read, decipher and then plot these interactions and messages himself. With the project having a tight time scale the extra time needed for this could not be justified and so a sequence diagram was the best option on this occasion. [3] [4]



A clearer Sequence Diagram can be referred to in Appendix C. Refer to the glossary for more details on both sequence diagrams and user stories.

4.1.3 Medium Fidelity Prototype

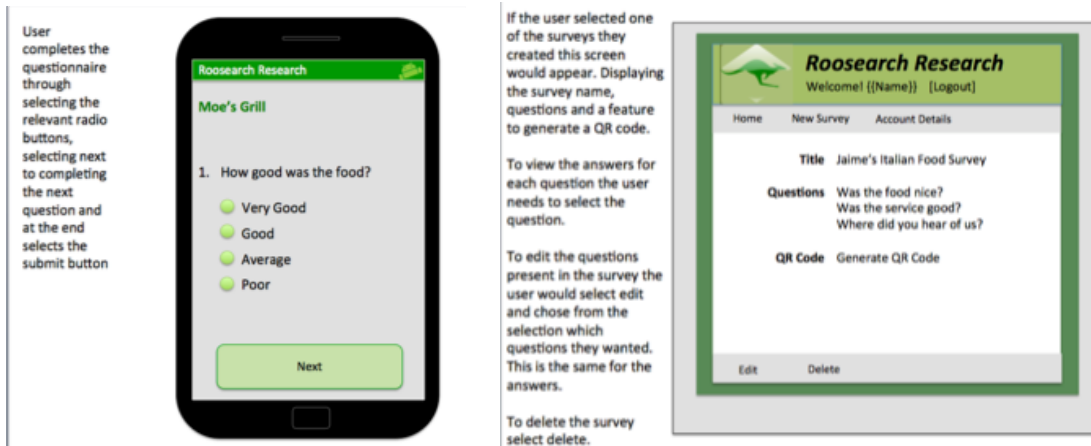
Medium fidelity prototypes are a middle line prototype between low level and high level prototypes. Medium fidelity prototypes provide the best of both worlds as you can implement both high level and detailed views rapidly and still complete iterative changes. Prototype designs such as these depict how individual pages should be structured and if designed correctly can show the steps through each activity.

There was an option to develop low level fidelity prototypes because the benefits of the medium level by far outweighed the advantages of the low level one. Some of these advantages included: that you could catch mistakes early and easily correct them, it allows for effective iterative development and provides more of a guide to the development of later and more detailed stages. There were also risks with producing a lower level fidelity prototype, including: if they are lost or damaged then the developer has no designs to follow unless a backup it made and there is no way for the prototype to be developed in the time frame possible meaning the designs needed to show the full potential of Roosearch Research, but with hand drawn designs this is just not possible. For these reasons you can understand why a medium fidelity prototype design was needed. [5]

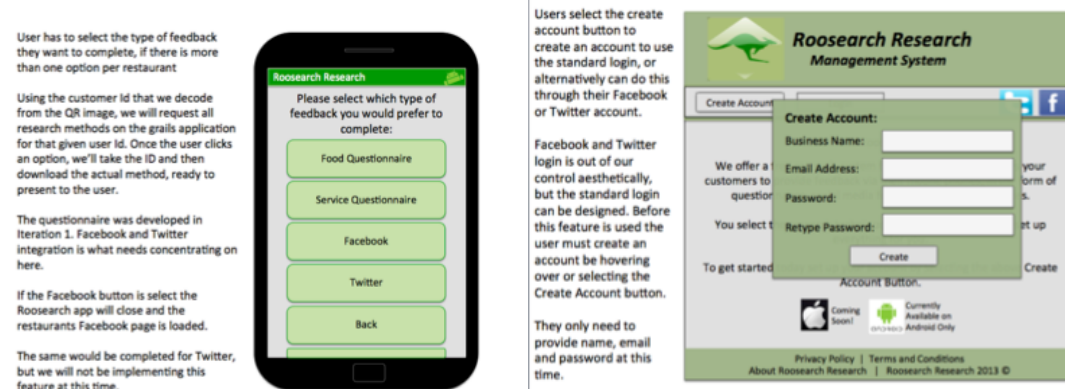
I developed three medium fidelity prototypes, one for each iteration. These were used religiously by the developer as they contained the requirements, how it should be designed and the links between the pages and components. I used these to check the progress of the developer throughout the iterations and to develop more accurate test cases. More details on how these were used can be found in the “Evaluation” section.

The medium fidelity prototype for Roosearch Research project was produced in Microsoft PowerPoint. This was the best facility available for the prototype to be developed in and let my developer see that he had to develop.

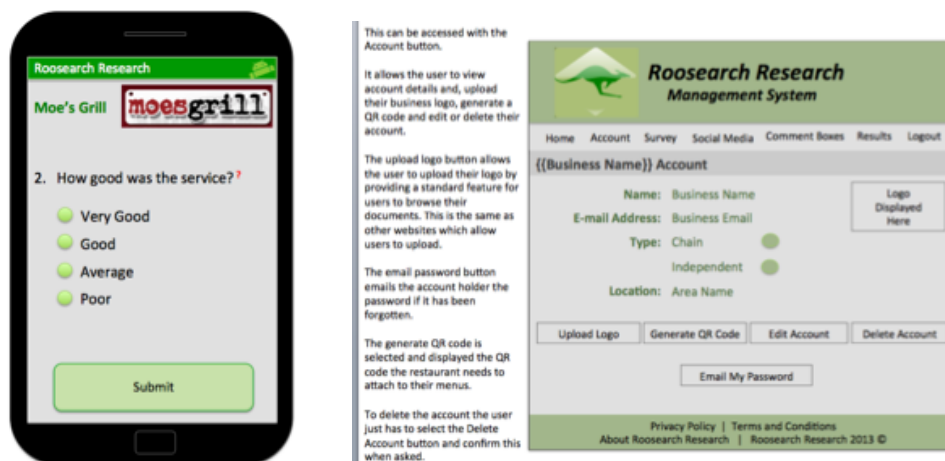
Iteration One



Iteration Two



Iteration Three



These designs can be seen in Appendix D, E and F and further information on prototype details can be seen in the glossary.

4.2 Design Justification (Medium Fidelity Prototypes)

4.2.1 Iteration One

The iteration one designs can be seen in Appendix D.

The first iteration was intended to implement the basic functionality of the prototype, essentially a “proof of concept” to prove that we could develop a prototype that met the design specifications. As such, the first iteration involved developing the questionnaire framework for the web application, and the questionnaire engine for the mobile application.

From my research in the interim report I established that a questionnaire is more than suitable and adaptable enough to be integrated into an Android mobile application for the use of restaurants to gather information from the source. For more details on this refer to the Interim Report, section “Market Research”. The structure of the web application and the questionnaire are core to the project and as such took priority in terms of development in this iteration.

Design took a backseat in this iteration as functionality was more important and as such this has been scheduled for iteration three primarily, with some in iteration two if the development goes well. The main reason for this is stated above, that this is a prototype providing “proof of concept” and so design is beneficial to have, but not vital.

As a result the designs of the interfaces in iteration one fidelity prototype are basic, with the more complex designs coming in the form of how the questionnaire was created by the restaurant manager and completed by the customers. The steps I had designed were with the Roosearch customers in mind, as I wanted to ensure that it was easy to follow and understand.

Web Application

The first iteration involved creating the structures and scaffolding to enable users to create, view and edit questionnaires on the web application. This was a relatively straightforward development task as once our objects were implemented as rails domain classes, it was easy to generate the views and controllers via the rails scaffolding feature (see “Glossary” for information on MVC (Model View Controller)).

Scaffolding is a very useful feature of rails, and is particularly convenient when prototyping web applications. Given a model, you can enable scaffolding in your controller. Then, using convention over configuration, rails is able to determine how to dynamically create the controllers and views for the CRUD (Create, Read, Update and Delete) operations on your domain objects. This meant that we did not have to write controllers or views, but rely on rails to generate these for us at run time.

This application consisted of a two step approach to creating the questionnaire with the answer options having been designed to assign automatically when the questions are selected and the

users select the create questionnaire option. Although this is a one fits all size type of design option I felt it gave this feature more a market research twist and made it more user friendly and quicker to complete.

Mobile Questionnaire Engine

The first iteration also involved a spike of development on the mobile application to create a questionnaire engine. We have designed the mobile application to take a model of questions and then to present them to the user in a wizard style interface, so the user has a single question on a screen at a time, selecting from one of the multiple choice options then progresses to the next question, retaining progress of selected options.

The design intention was to enable the application to retrieve questionnaires from the website via RESTful web services, and then convert these to local domain objects which could be injected into the questionnaire engine.

This side of the questionnaire feature involved a multi-step solution. I decided that the less information on the mobile screens the better and so only one question appears at a time, with a review of all answers given at the end of the questionnaire, before submitting it back to the web application.

4.2.2 Iteration Two

The iteration two designs can be seen in Appendix E.

Iteration two was still focusing on the implementation of functionality, including: Facebook, Twitter, QR Code, and a login system.

From our research in the interim report we established that social media research is a newer and rapidly growing trend. As a result we decided to implement this into our system allowing the restaurants customers to post or tweet with ease through the use of the QR code and mobile application. For more details on this refer to the Interim Report, section “Market Research - Social Media”.

It was felt that a QR Code would provide the optimal and easiest solution to allowing access to the restaurant data via the application. This is part of what is setting us apart from our potential competitors which were identified in the interim report. The QR code was needed to allow the users access to the restaurants application features, without having to search through a list of restaurant names, similar to our competitors application. Now the user only has to scan the QR code and the application with the relevant restaurant details will be displayed on screen, allowing them to easily select the market research feature they wish to complete and submit.

Facebook and Twitter is intended to enable administrator users to register and authenticate against the application by the use of existing authentication mechanisms, such as facebook and twitter connect. This improves user experience and enables seamless integration without the need for the user to fill in a registration form, verify email, and have another set of credentials to remember. Aside from easier authentication, using external services to authenticate the user also enables us to retrieve additional information regarding the user, such as their email address, name and profile picture, which can be used on correspondence from the application. In short, the user would click either “login with facebook” or “login with twitter”. A window would then popup from one of those authentication providers prompting the user to grant permissions on their credentials, once granted, the application will then be able to obtain the unique user id for that provider which we can then use as our identifier for the user. We can also then use this to retrieve additional info from the provider such as the facebook username, or the twitter handle. Unfortunately due to time constraints and difficulty in completing this task it was not able to be implemented and as a result has been descoped for future work. In its place is a mock-up login screen as time was not available to complete one from scratch. With the detailed designs this feature does need to be improved and working, but is useful in the demonstration of what can be done.

The Facebook and Twitter social networking links were implemented within this iteration which now provide the restaurant users with the option to carry out social networking market research. In terms of interface design, this is a one step feature as the users only need to copy and paste their Facebook or Twitter links into the appropriate boxes and select create. This then allow the

users to select these options from the application once the QR code is scanned.

Design took a backseat in this iteration as functionality was more important and time consuming, so this has been scheduled for iteration three. The main reason for this is stated above, that this is a prototype providing “proof of concept” and so design is beneficial to have, but not vital.

4.2.3 Iteration Three

The iteration three designs can be seen in Appendix F.

Iteration three was used to improve on the features present, implement a better QR code feature and add in graphs on the question pages as there was not enough time to create a separate reports page.

Iteration two saw the development of a QR code which needed to user to select a generate QR code button before it would open in an external window to the website. As a result of this time was spent on the QR code so that it was generated automatically, based on my decision as I thought it would be more user friendly. With the automatic generation of the QR code it is now situated in the users accounts page for them to copy onto a menu or print off and place around the restaurant.

Iteration two saw the development of basic reporting facilities, now within this iteration features such as sentiment analysis and more complex graphing features there planned to be enabled. However, due to problems in the previous iteration there was not enough time left to complete this feature. I felt this was an important feature to have implemented and so came to the decision that it would be better to have graphs on the questions page, instead of having no reporting facilities at all. Currently the user a view these by selecting the questions within the questionnaire, on the question page.

Other than the above features, iteration three would have had the aesthetics and usability upgraded, but ran out of time. As a result the following aspects were investigated only and the conclusions and justifications are shown below for reference to the designs, not the prototype.

[10] [11]

The colour scheme chosen for the system had to be professional, yet ensure that it was suitable for all users. This is important as colour blindness affects a significant number of people, of which figures can be seen in the image below from colour-blindness.com. For this reason it is important that the colours are chosen carefully, ensuring I do not have red and green together or blue and yellow. The green and grey together complement each other and provide a professional look to the system and will not have as big an impact as those with the above colour blindness issues.

	Men	Women
Red-green (Overall)	7 to 10%	—
Red-green (Caucasians)	8%	—
Red-green (Asians)	5%	—
Red-green (Africans)	4%	—

Usability also has to be taken account of as the range of users that could use the application will come with a guarantee of users having a ranging level of computer and mobile skills. To help combat any problems users may have the third iteration has allocated time for help features to be implemented in the form of small question marks that will be in different location in both the mobile application and the web application. The user can then select these and a message with guidance will appear. The standard ISO 9241 Ergonomics of Human System Interaction also supports usability issues. It states that a software system should have a low learnability rating, in terms of the time needed to learn how to use a particular aspect.

Shneiderman's eight golden rules also recommend that a system should be consistent in design and layout. I have ensured that every aspect of the system is consistent to allow the user the easiest chance of learning the paths of the system. It also stated that all software systems should have informative error message, so that the user can relay a more detailed report to yourselves so the error can be corrected. If experienced enough the user could then fix the problem themselves, if it is on their side. An error message stating "Error 404" is not helpful to any user, whereas an error message as follows would be useful, "Feature Not Operational Yet". These are also seen in Norman's seven principles.

5.0 Implementation

5.1 Technologies

Given that the project is in a prototype stage and not production, efforts were made to ensure the chosen technology stack could provide a product that had high value, yet low effort to get off the ground. The developer recommended a number of potential technology stacks that could be used such as grails, php, ruby and c#. Upon advice and discussion with the developer, I decided that grails would be the chosen technology stack for this project for the reasons below.

In regards to the mobile element of the project, the developer also recommended a variety of options such as android, iOS, windows, and cross platform development. Android was selected primarily for being the most open and accessible platform, and also it fit in with the skillset of the developer, but also from my research in the interim report.

Web

The chosen technology for the web application is Grails. Grails is a Groovy based web application framework for rapid application development. Grails was an obvious choice for the prototype as the developer resource was already skilled in Java, and grails uses groovy which is a JVM language. The web application was built using gradle, which is included in the grails framework

Mobile

The chosen technology for the mobile application is android. Considerations were given to android, iOS, and cross platform web applications. Given that the developer resource was skilled with android, and android is an open platform which does not have the licensing and device deployment implications that iOS does, it was the obvious choice. Also, android applications are developed in Java, similar to groovy which is used for grails. The application is a native android application, built with maven.

Data Storage

Given the nature of the project and its continual evolution during the iterations, it was considerably easier to avoid having to persist data, as the structures would be changing very often and would incur consistent schema modifications. As such, it was decided that for the prototype it would be sufficient to retain data in memory and not persist it to any form of database or storage system. As part of further development, perhaps if the scope was decreased in some areas or additional iterations were included, it would have been possible to take advantage of the grails GORM framework and map our domain objects into database structures. For all intensive purposes, from the grails side we would still develop against groovy domain classes, but grails would take care of creating the data structures and relationships, mapping the domain objects onto tables, and handling all CRUD operations, almost entirely decoupling the application from the database layer. The benefit of this would be portability, for example if the application was

successful and required scaling up, it would be deployed as a cloud based SaaS, and the appropriate scalable database implementation could be integrated with very little change to the application.

Continuous integration and deployment

Both the mobile and web applications were part of a continuous integration environment, albeit not an automated testing environment. Both applications had their source code residing on a subversion repository, for source control management, and both applications were continuously built on a cloud based jenkins server, so any issues would be spotted immediately and could be corrected by the developer. Ideally, continuous integration projects would also have some level of automated testing, so when multiple developers are committing to the same project, each commit is pulled onto the jenkins environment, built, tested, and feedback is handled immediately, such as notifying developers of test failures, or initiating downstream projects.

In addition to the jenkins CI environment, the web application was also deployed to a application server on Cloudbees, which give free java application hosting. This enabled us to deploy a build to the web container after each commit, so that the mobile application was always interfacing with the latest version of the web application. Ideally, in a production system we would have both the mobile and web application fully automatically tested in clean room environments on the jenkins server, but this was less of a priority for my prototype.

5.2 Data Modelling

As mentioned in the technology section of this document, the applications data was not stored in a database but held in memory, and therefore the below queries are better considered as pseudo queries.

5.2.1 Data structure (DDL)

The following are DDL statements I created, that describe the tables that would be created to support the applications underlying data. It is important to note that these would be generated and executed by the grails GORM framework so may differ slightly syntactically in a production environment, but are present for example of what will be displayed.

```
create table USER(id NUMBER PRIMARY KEY,  
                 firstName VARCHAR(50),  
                 lastName VARCHAR(50),  
                 emailAddress VARCHAR(50) ,  
                 companyName VARCHAR(50) ,  
                 facebookPageLink VARCHAR(50) ,  
                 twitterHandle VARCHAR(50) );
```

```
create table SURVEY (id NUMBER PRIMARY KEY,  
                    title VARCHAR(255));
```

```
create table QUESTION (id NUMBER PRIMARY KEY,  
                      text VARCHAR(255));
```

```
create table RESPONSE (id NUMBER PRIMARY KEY,  
                      text VARCHAR(255),  
                      number_of_people_selected NUMBER);
```

```
create table USER_SURVEY_LINK (link_id NUMBER PRIMARY KEY,  
                              user_id NUMBER NOT NULL,  
                              survey_id NUMBER NOT NULL);
```

```
create table SURVEY_QUESTION_LINK (link_id NUMBER PRIMARY KEY,  
                                   survey_id NUMBER NOT NULL,
```

```
question_id NUMBER NOT NULL);
```

```
create table QUESTION_RESPONSE_LINK (link_id NUMBER PRIMARY KEY,  
question_id NUMBER NOT NULL,  
response_id NUMBER NOT NULL);
```

```
create table SURVEY_TRENDS (survey_id NUMBER PRIMARY KEY,  
submitted_date DATE);
```

5.2.2 Queries (DML)

I created the following queries which are examples of how the application would likely retrieve the data needed for display to the users, for data management and reporting purposes. I have described the below queries in a BDD manner to help emphasize the business requirements.

Given a user has authenticated
And has an Id
When showing the survey summary page
Then show a list of their surveys by title

```
SELECT
    s.title
FROM
    SURVEY s,
    USER_SURVEY_LINK usl
WHERE
    usl.user_id = <USER ID HERE>
    AND usl.survey_id = s.id
```

Given a survey id
When the user requests a survey overview
Then show a list of the questions that relate that relate to that id

```
SELECT
    q.text
FROM
    QUESTION q,
    SURVEY_QUESTION_LINK sql
WHERE
    sql.survey_id = <SURVEY THAT USER CLICKED ON>
    AND sql.question_id = q.id
```

Given a survey id
When a user requests a survey report
Then show the number of votes relevant to each question

```
SELECT
    count(q.number_of_people_selected) as VOTES,
    q.text
```

```

FROM
    QUESTION q,
    SURVEY_QUESTION_LINK sql
WHERE
    sql.survey_id = <SURVEY THAT USER CLICKED ON>
    AND sql.question_id = q.id
GROUP BY 2

```

Given a survey id

When the user requests trend information

Then display the number of responses for each question in the last calendar 7 days

Trends - number of people responding in the past week

```

SELECT
    count(q.number_of_people_selected) as VOTES,
    q.text
FROM
    QUESTION q,
    SURVEY_QUESTION_LINK sql,
    SURVEY_TRENDS st
WHERE
    sql.survey_id = <SURVEY THAT USER CLICKED ON>
    AND sql.question_id = q.id
    AND sql.survey_id = st.survey_id
    AND st.date_submitted AFTER (sysdate-7)
GROUP BY 2

```

Given a survey id

When the user requests trending information

Then display the number of votes against each question

So That the user can compare against the previous year

Trends number of people responding compared to this time last year

```

SELECT
    count(q.number_of_people_selected) as VOTES,
    q.text
FROM
    QUESTION q,

```

```
    SURVEY_QUESTION_LINK sql,  
    SURVEY_TRENDS st  
WHERE  
    sql.survey_id = <SURVEY THAT USER CLICKED ON>  
    AND sql.question_id = q.id  
    AND sql.survey_id = st.survey_id  
    AND to_char(st.submitted_date, 'MON-YYYY') = 'Jan-2011'  
GROUP BY 2
```


5.3 Meeting and Communications

With the project being outsourced to a professional developer I needed to establish and maintain good communication channels to help ensure that the project is running smoothly and the system designs I created were appropriate and achievable. It was decided that we needed meetings on a fortnightly basis (as we were located in different areas) and email checkpoint reports on the weeks where a meeting was absent.

Within iteration one there was communication in the form of meetings every week as I was in the same location as my developer; we also communicated via email at the end of the iteration to confirm what had been completed, how the development was going and if there was any foreseen problems for the next iteration.

Iteration two consisted of one meeting, one at the start of the iteration. The week in the middle was taken as a holiday and the last week of the iteration involved the production of a checkpoint report email consisting of how the iteration development went, what was completed and was not, and if there could be any problems with the next iteration.

For iteration three there were two reports, one for the start of the iteration and the other for the middle week. The last planned week of this iteration involved a meeting between my developer and myself. However, this iteration continued into the contingency week as timings were tight in the previous iterations and due to a week's holiday. The contingency week involved a meeting to complete the project development.

5.4 The Prototype

Although not all features that were planned in the designs and requirements were completed in the time given, we were still able to completed a sizeable amount to create a working prototype that can be used in the interview questions to establish whether market research is viable in the use of mobile phones. It is enough to provide proof of concept. Some images of the prototype can be seen in Appendix G, and in addition to this you will be able to view a run through of the prototype at the project viva.

All design and planning documentation was completed up front by myself to ensure that the developer knew what he had to do and what to create the aesthetics to look like, if he got that far on. During the development of the prototype I was also responsible for organising the meetings and ensuring that I received a checkpoint report email on the week we did not have meetings. Once I had handed over the designs for the iteration I started to work on the test cases for that iteration and complete the ones from the last iteration to ensure that they match what was written in the checkpoint reports, or stated in the meetings.

For details of the development and what was completed please refer to the Evaluation Prototype Section.

6.0 Testing Evaluation

Testing and validation of the web based management system is taking place to ensure that it is not only fully working, but is also being used as a test to ensure that all of the functionality is present and in align with the requirements given in the interim report. The testing for this project will come in the form of test cases, and unit tests. Further details in these can be investigated below.

6.1 Test Cases / Ad Hoc Testing [6]

In accordance with the IEEE standard 610 (1990), test cases are defined as the following:

“A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirements.” [7]

A test case is a set of conditions under which a tester will determine whether a system use test satisfies the requirements and works correctly. They are used as technical explanations and reference guides for systems and include the inputs and outputs of a system and then provides a detailed description of the steps that should be taken, the results that should be achieved and other elements that need identifying, such as pre conditions. Test cases should be atomic and not complicated.

For iteration one there are six test cases which expand to cover the development so far, including the creating, accessing, editing and deleting user accounts, and creating editing, deleting and completing questionnaires.

The test cases for iteration one can be referred to Appendix H, I have also attached a sample one below.

Test ID: 2	Test Name:	Create Questionnaire	
Step Number	Description	Expected Result	Pass / Fail
1	- Load up Roosearch Research web system	The web system home screen will load, displaying a welcome and Facebook login feature.	P
2	- Select Connect with	The users need to	P

	Facebook	completed details when screen is displayed and select login. This will display the user account.	
3	- Select “new survey”	A screen appears displaying the template needed to create a questionnaire.	P
4	- Enter title	You are allowed to enter a title	P
5	- Select desirable questions	You can select multiple questions.	P
6	- Select create	The questionnaire is created and new screen displayed showing questionnaire components, with option to view answers and generate QR code.	P
7	- Check answers are suitable by selecting each question in turn	When you select the questions, a screen is loaded showing the current answers and the question they are relevant to.	P
8	- Select Generate QR Code	From step 6, you can select Generate QR code; this will direct	P

		you to a page displaying the QR code.	
--	--	---------------------------------------	--

On top of test cases, Ad Hoc testing was also completed. This is testing without formal documentation, processes or a plan of action with the process of testing not documented after completion. One of the benefits of this type of testing is the lack of formal requirements allows for flaws to be attended to more quickly, compared to the use of a more systematic approach. Ad hoc approach testing is one of the best ways of discovering early bugs in the code, which is one of the main reasons the it have been chosen to enhance the testing procedure for the Roosearch Research project. [8]

6.2 Unit Tests

Unit tests are a testing methods which takes individual units of source code and sets of one or more computer program modules together with associated control data, usage procedures and operating procedures are tester to determine if they are fit for use. These are created by my developer during the development process to ensure that the code meets its design and behaves as intended.

The intention of a unit test is to isolate each part of the program test the various scenarios that may occur. For example, in the roosearch application we would have a unit test around creating a new survey. The unit test would create a survey object, add it to the collection of existing surveys, then assert that when we request the collection of surveys again, the count has been incremented by 1, and our survey is now in the collection. This is just one example, we would also have test cases around negative scenarios, such as adding an incomplete survey should return an error, or adding a survey that already exists should also give an error. Each of the expected negative, and positive scenarios should be unit tested. It is not wise to test unexpected scenarios otherwise the list of tests would become endless, and even then we may not cover all possible scenarios.

As mentioned, unit tests should specifically target the smallest significant element of code, often public methods (private members are indirectly tested via public APIs). Unit tests should not cover writing data to actual database tables, or reading data from external streams, as these are deemed integration, or functional tests. If a unit test has interaction with a database, that element should be “mocked”, essentially enabling the test to build a mocked wrapper layer around external dependencies, so we have predictable and repeatable behaviour upon each invocation of the unit test. Unit testing is a fundamental part of agile development, TDD, and good software practice.[9]

6.3 Test Results And the Effects

Testing cases evaluation shows that a number of tests failed. However, from a closer look this is due to features not being implemented, not problems with the components that have been developed. This is with the exception of the login features using Facebook and Twitter and had to be dropped from the specification because the time needed to implement these was much greater than expected. However, there is a basic stand alone login to the web application and all tests for this have passed. The features which did not pass include: report page as the graphs were assigned to the questions page to save time, the comment bank, additional account features such as logo uploads and allowing the selection of the type of restaurant and the help features to aid users in utilizing the system to meet their needs. All of these can be completed in the future and further details of this can be seen in the “Future Work” section.

Summary of Results Table:

Test Case	Summary	Pass or Fail Overall
<i>Iteration 1</i>		
Login	The login test did not meet the requirements of the test case as none of the login features were implemented.	Fail
Create Questionnaire	Full requirements of the test case were met here.	Pass
Complete Questionnaire	Full requirements of the test case were met here.	Pass
Edit/ Delete Questionnaire	Full requirements of the test case were met here.	Pass
Delete User Account	Basic requirements of the test case were met here.	Pass
Create User Account	Basic requirements of the test case were met here.	Pass
<i>Iteration 2</i>		
Login 2	The login test did not meet the requirements of the test case as none of the login features were implemented. However, a mock-up login was developed to allow people to see what it would have been like. This could not be implemented due to time	Fail

	constraints and difficulties integrating the Facebook and Twitter logins.	
Create Facebook Feature	Full requirements of the test case were met here.	Pass
Create Twitter Feature	Full requirements of the test case were met here.	Pass
Use Facebook Feature	Full requirements of the test case were met here.	Pass
Use Twitter Feature	Full requirements of the test case were met here.	Pass
Edit / Delete Facebook Feature	Full requirements of the test case were met here.	Pass
Edit / Delete Twitter Feature	Full requirements of the test case were met here.	Pass
Complete Questionnaire	Full requirements of the test case were met here.	Pass
Logo Upload	The requirements were not met as there was not enough development time and non-vital features had to be descoped.	Fail
QR Code Generator	Full requirements of the test case were met here.	Pass
View Accounts Screen	Full requirements of the test case were met here.	Pass
<i>Iteration 3</i>		
Create Comment Bank	The requirements were not met as there was not enough development time and non-vital features had to be descoped.	Fail
Write and Submit Comment	The requirements were not met as there was not enough development time and non-vital features had to be descoped.	Fail
Help Hints	These were the next feature to be implemented as they are important for usability.	Fail

	Unfortunately we did not have the time and these had to be set as a high priority of the descoped features.	
Results Summary and Comparison	Some requirements were met here, as graphs were developed on the question pages, instead of a report page. As a result of this the tests did fail, however with reporting features implemented I felt that the overall aim of the test case was partially met.	Pass
Questionnaire Completion Limit	Full requirements of the test case were met here.	Pass

The unit tests shows my developer the areas in the code bugs were present and so he was able to address these problems early on to avoid the risk of running out of time and not completing testing.

With the development not quite going to plan (which we knew it was not going to due to time constraints), we were lucky that each components was not totally dependent on the previous one allowing for features to be descoped, and not dramatically affecting the rest of the project.

7.0 Validation Interviews

With the testing of the web based management system completed, and knowing that the components which were built are working correctly I can now move onto validating the system through completing interviews with both restaurant owners / managers and their customers. By undertaking this task I should be able to prove or disprove that market research can be corrected using mobile phone applications. Below you can see the information collected and what this means.

7.1 About the Restaurants and Customers

It has been decided that three restaurant owners / managers and three customers will take part in individual interviews to try and establish if a range of people from different backgrounds would use Roosearch Research, if it was completed to a high quality, to conduct market research or give their opinions.

Present below are the details of the restaurants taking place, of which two have asked to remain anonymous as they do not own the restaurant. With this in mind all the participants will be identified by a number.

- Restaurant 1: Restaurant one is an award winning chinese restaurant based in Belfast City centre. The manager was interviewed here as the owner owns a number of different restaurants with a large and loyal clientele.
- Restaurant 2: Restaurant two was a smaller, traditional family owned business with loyal customers in the local area. Here I interviewed the owners of the restaurant which is one of the most successful and popular in its area.
- Restaurant 3: The last restaurant, restaurant three, was a fish restaurant in Suffolk. This restaurant is renowned for the fish it serves and here I was able to talk to the manager about the web based management system.

Details of the customers taking part can be seen below:

- Customer 1: The first customer was a twenty-one to thirty year old female studying at university. She eats out three to four times a month, enjoys a range of different foods and currently uses an Android platform mobile. As a student she has a limited salary of under £500 per month.
- Customer 2: My second interviewee was a thirty-one to forty year old male that works full time, earning between £1500 and £2500 per month. He eats out three to four times a month, enjoying most chinese and indian meals. Currently, he uses an iPhone.
- Customer 3: The third customer was a male of fifty plus years that works full time on a wage of £500 to £1500 per month. He eats out at restaurants on an average of five to six times a month, at a range of different cuisines and currently has an Android platform mobile.

7.2 Interview

For both type of interviews (customer and restaurant) the same process was undertaken to ensure that a fair result could be determined at the of the process. Below is a description of the process that was taken and the questions that were asked.

Once the interviewee had agreed to the interview I started by formally introducing myself and telling them what I was doing and going to ask them about. Following this the web based management system was demonstrated to them and the designs shown and explained in place of the missing features and designs. Now that they knew what the questions were based on I started the information collection. I started with personal questions ensuring that I was not storing exact personal information such as their name, exact age and salaries. After completing this the question about the system were asked with the askers taken down in detail. The interview was concluded after this with me thanking the interviewee for their participation.

The questions that were asked to the restaurant include:

Question	Justification
What type of cuisines do you provide?	I thought it was important to ask this so that I could ensure the interviews conducted were not biased through being based on only one cuisine type.
Are you the owner / manager / manager on duty?	This question needed to be asked so that I knew who I was talking to and where they were ranked in the restaurant.
What do you think of this system?	I wanted to get open answers from this question and so I thought I would start with this and prompt the user to feeling easy about giving their true opinion.
Would you use this system if it was made available to restaurants, free / charged?	This was the main point of the interview, to establish if the system would be used. Therefore the question had to be asked to get a reliable answer and not guess.
Is there anything you would improve / add in?	I thought it would be useful to find out the opinions of the users, in terms of functionality and what they thought. They may say the like it in the third question, but to them it could be missing an important function and so I used this opportunity to find out.
Is there anything you would remove?	I thought it would be useful to find out the opinions of the users, in terms of functionality

	and what they thought. They may say the like it in the third question, but to them it could have redundant features.
What do you think of the colour scheme?	Colour schemes have a big impact on whether people use a system or not. If the colour scheme annoys them or is not clear then they could be deterred from using the system.
Being in the restaurant business, do you think this system is a good idea? Would it be used?	I wanted to ask in general so they did not feel they had to say yes. I thought it would be good to have the opinion on someone in the industry.

The questions that were asked to the customers include:

Question	Justification
Gender?	I thought it was important to ask this so that I could ensure the interviews conducted were not biased through being based on only one gender.
Age Bracket - under 20, 21 - 30, 31 - 40, 41 - 50, 50+	I thought it was important to ask this so that I could ensure the interviews conducted were not biased through being based on only one age bracket.
Employability status - Full Time, Part Time, Unemployed, Student	I thought it was important to ask this so that I could ensure the interviews conducted were not biased through being based on only one employment type. Plus I thought it could be interesting to see if employability status affects what they thought.
Monthly salary range - under £500, £500 - £1500, £1500 - £2500, £2500 - £3500	I thought it was important to ask this so that I could ensure the interviews conducted were not biased through being based on only one salary bracket. Plus I thought it could be interesting to see if salary status affects what they thought.
What they prefer to eat? - chinese, indian fish etc	I thought it was important to ask this so that I could ensure the interviews conducted were not biased through being based only on people who prefer a certain type of cuisine.

How many time they eat out a month?	I thought it was important to ask as I did not think it was appropriate to ask someone who does not eat out at restaurants.
Type of mobile - Android, Windows, iPhone	This was important as I was looking for people with Android mobiles and iPhones, as these are the people who will primarily be using this application.
What do you think of this system?	I wanted to get open answers from this question and so I thought I would start with this and prompt the user to feeling easy about giving their true opinion.
Would you use this system if it was made available in restaurants?	This was the main point of the interview, to establish if the system would be used. Therefore the question had to be asked to get a reliable answer and not guess.
Is there anything you would improve / add in?	I thought it would be useful to find out the opinions of the users, in terms of functionality and what they thought. They may say the like it in the third question, but to them it could be missing an important function and so I used this opportunity to find out.
Is there anything you would remove?	I thought it would be useful to find out the opinions of the users, in terms of functionality and what they thought. They may say the like it in the third question, but to them it could have redundant features.

7.3 Results

Overall the results of the interviews were positive, below the two type of interviews have been split up and analysis independently.

7.3.1 Restaurant

The results of this interview are displayed in the table below:

	Restaurant 1	Restaurant 2	Restaurant 3
Cuisine Type	Chinese	Traditional	Fish
Relationship to Business	Manager	Owner (Family)	Manager
What they thought	Good, interesting idea	They are close with clients, but think it is	Good, would definitely be useful

		a good idea	
Would they use	Yes	Maybe if free	Yes
Improvements	Email Marketing	NA	NA
Removals	NA	Twitter	NA
Colour Scheme	Good	Nice and easy on eye	Very Good (similar to theirs)
Good Idea	Yes	Yes	Yes

From the results above you can see that the overall trend is positive, in favor of the web based management system. All restaurants but the family run one said they would make use of this system. The reason the restaurant two gave for not using it is that they know their customers very well and so did not feel the need for it, but felt it could provide benefits from newer customer they may not know as well as their loyalists. Restaurant one also said they would like email marketing, we addressed this in the interim report but came to the conclusion that this was not suitable for the type of system that is being created. Refer to interim report section "Market Research" for further details. The customer interview results show a similar pattern that can be seen below.

7.3.2 Customer

The results of this interview are displayed in the table below:

	Customer 1	Customer 2	Customer 3
Gender	Female	Male	Male
Age	21 - 30 yrs	31 - 40 yrs	50+ yrs
Employability	Student	Full Time	Full Time
Salary	< £500	£1500 to £2500	£500 to £1500
Preference Eating	Range	Chinese / Indian	Range
Times Out / Month	3 / 4	3 / 4	5 / 6
Mobile	Android	iPhone	Android
What think?	Good idea	Very interesting	Sounds good
Would Use?	Yes	Definitely	Yes, better than other options
Improvements	NA	NA	NA
Removals	NA	NA	Facebook and Twitter

The results above show an overall very positive trend. You can see that there was a diverse range of customers selected to complete the interview; this was to help ensure the results represent a more rounded group of people. All participants thought that this was a really good idea, with Customer two asking when it would be available for both Android and iPhone. They all stated that if this was available they would make use of it as it is much better than the other options being used currently; such as the laptops in Bellinis or the information cards in Las Iguanas. When asked about possible improvements from the designs the customers could not think of any, however, customer three did suggest that the Facebook and Twitter features are removed as he did not understand how they could be useful or relevant based on the fact that he did not use these. Once it was explained that social media research is an up and coming trend he felt it has the potential to be useful.

7.4 What Does This Mean For Roosearch Research

The result above reflects positively for Roosearch Research. From the interviews I was able to prove that a market research collection system which incorporates mobile technology can be implemented and will be used by the general public, and most restaurant owners. The restaurant customers definitely seemed keen on the idea, and most business, restaurants, will do anything within their power to ensure that their customers are happy. Roosearch Research has the potential, if developed fully and to a high standard, to be used by widely within different communities. From the results of the interview about and the market research undertaken in the interim report, it shows that Roosearch Research could have a strong future.

However, before the system can be deployed, it needs the remaining features implemented and tested further to ensure that it is fit for purpose and what the Roosearch customers are wanting. With this in mind, I definitely feel that it is worth further developing Roosearch Research and publishing it on the Android market and following this the App Store for iPhones.

8.0 Evaluations

8.1 Prototype

On the whole, the initial requirements were met, a web application was delivered which enables customers to register and create surveys, a mobile application was also delivered which allowed patrons to feedback information on the services they received at a given establishment.

Some of the requirements needed to be simplified or de-scoped altogether, notably the social networking integration. This needed to be descope because it was a significant time sink on development resources, this was partly due to the support for rails social networking plugins. Of the three available rails plugins, one was outdated and would not work, one did work initially, but was using legacy apis on facebook, and the attempt of using the third was impacted further by not having the required time to understand how to use it. We were able to implement the social sign on, and to display the unique user identifier from facebook, however we encountered issues during the data setup layer whereby we needed to bootstrap the application to initialise the data, which required user information not available at time of bootstrapping. This quickly became problematic so it was decided to remove the social networking integration in favour of investing time in reporting and the mobile application. In place of these I used a mock-up of a login page to demonstrate what it could be like.

Within iteration one I was able to deliver a working web based questionnaire system, along with a questionnaire engine in mobile format. Neither of these systems were interconnected at this stage but it was demonstrated that the basis of the prototype could be delivered.

Iteration two invested a considerable amount of effort into the social networking login, which, for a number of reasons had to be descope from the iteration and a mock login screen was introduced to demonstrate future potential. Twitter and facebook links were introduced to the customer entity, along with a qr code generation link.

Iteration three consisted of integrating the web application with the mobile application. We invested development efforts into producing a variety of web services, or APIs on the web application that the mobile application could consume for retrieving surveys and posting results. Aside from joining the two elements of the prototype, effort was invested into reporting and a variety of charts were introduced to visually display statistics on customer feedback

Appendix G is where the prototype images can be viewed.

8.2 Approach

From the details in the Planned Approach section you can see that we took an iterative approach to the development, splitting it into three iterations. In theory one for the structure, another of features and the last for design. However, the development did not turn out this way, as you can see from the details below.

Iteration one was supposed to see the implementation of:

- The basic structure;
- Questionnaire feature in both the mobile and web applications;
- A login system;
- and a functioning QR code generator.

However, by the end of iteration one we had only managed to complete:

- A basic structure;
- And the questionnaire feature in both the mobile and web application.

This was mainly down to the time it was taking to develop the components. It was taking longer than thought to develop the components and the login in system was proving challenging and as a result descoped to the next iteration. Throughout this iteration my developer and myself both thought the medium fidelity prototypes and class diagram were the best aids as they provided details of the requirements, designs and how the features should operate.

Whilst my developer was programming this iteration I was concentrating on the designs for the next iteration and the test cases for this iteration. I was also looking at the class diagram and sequence diagram and updating these.

Iteration two was supposed to see the development of:

- Social Media Research;
- Login systems;
- Pre Generated QR Code;
- Extensive user account features;
- And a basic reporting page.

Of the features listed above only a few were implemented due to time and skills levels. These features include:

- Social Media Research;
- Pre Generated QR Code;
- Basic user accounts;

Again, this was mainly down to the time it was taking to develop the components. It was taking longer than thought to develop the components and the login in system was not going to work in the time frame available and as such was descoped from the development. Throughout this iteration my developer and myself both thought the medium fidelity prototypes and sequence diagram were the best aids as they provided details of the requirements, designs and how the

features should work together, we thought the sequence diagram was good as it should what the user had to do visually and the relationships between the components.

Whilst my developer was programming this iteration I was concentrating on the designs for the next iteration and the test cases for this iteration. I was also looking at the class diagram and sequence diagram and updating these and completing the test cases I created in iteration one to ensure that the features developed here were working. To date in the development we both felt that the UML requirements document was good for determining the requirements, but not as useful as the other documents when it came to aiding the implementation and design of future iterations.

Iteration three was supposed to see the implementation of:

- High quality designs;
- Comment box;
- Help hint;
- A more advanced reporting page;

However, by the end of iteration one we had only managed to complete:

- A reporting system which displays graphs in the question pages;
- A mock-up login;
- And a tidier system;

The reason for less implementation here was down to the leftover features from iteration two and the need to develop a mock-up login interface. Time was also not on our side here and so I decided that it was more important to have the features already created working well, rather than rushing the implementation of additional features and having a poorer quality system. Throughout this iteration my developer and myself both thought the medium fidelity prototypes and sequence diagram were the best aids as they provided details of the requirements, designs and how the features should work together, we thought the sequence diagram was good as it should what the user had to do visually and the relationships between the components.

While my developer was working on this I was completing the test cases for iteration two to ensure that implemented features were working, completing the test cases for iteration three and reviewing the process which had been taken to date. I feel that the test cases were really useful as they provided a way to test the functionality, but also to see what had successfully be completed. It provided a check list. At the end of this iteration I completed the last set of test cases and started to look at the interviews I had to undertake to validate the prototype.

In hindsight, I feel that time could have been managed better, but with my developer only being able to work a few hours a week I felt we were able to create a good application together. If I had managed the time better then we could have possibly had more features implemented. I general

I felt that the management of the project went well, I thoroughly enjoyed the role and got a lot out of undertaking it. Details of this can be seen in the lessons learnt reflection.

9.0 Future Work

Whilst efforts were given to achieve all set out in the initial plan, some objectives had to be descoped due to limitations on time, knowledge with the development frameworks, and other issues encountered over the duration of the project. The main areas of work available for the system as detailed below:

- As such, there are plenty of opportunities for future development of this project, notably the social network authentication. The social networking elements were dropped from the prototype due to time constraints, and there were difficulties with integrating these frameworks and still being able to bootstrap the application with data.
- Also, if the prototype was progressed into a production quality application, considerations would need to be made to automated testing. Both the grails and android applications could be unit and functionally tested relatively easily with some effort invested on creating stubbed or mocked environments, which would also integrate well in a jenkins CI environment.
- As stated in the interim report, if all goes well with the Android version, there is definitely an opportunity to develop Roosearch Research into the iPhone market and Windows market. This would significantly increase the number of customers who have access to the application to complete market research for the restaurants. According to the figures in the interim report Roosearch would then be accessible for over 84% of the mobile market.
- Accessibility - deaf and sight - Being a web application, there are a number of techniques that can be adopted to make the website more accessible such as adhering to the W3C accessibility guidelines for web sites to ensure that the web content can be accessed from screen speech synthesizers and other hardware. In regards to the mobile app, elements could be enlarged for hard of sight users, and a combination of speech recognition and text to speech could be used to obtain feedback from blind users. [19]
- Social network integration - This was descoped from the project, however as a future enhancement this feature could be revisited to enable roosearch administrators register and authenticate using their facebook / twitter credentials, and also allow patrons to authenticate using these social networks so they can share, like, or comment on an establishment.
- Advanced reporting - Elements of reporting were developed, however these could further be enhanced to give the roosearch administrators a deeper insight into customer feedback, and ultimately enable them to improve their service to their customers. Aside from a deeper analysis of survey results, it would also be possible to gain an insight into the usage and overall usability of the system by embedding tools such as google

analytics into the web and mobile applications. This would give the administrators a view on how the users are using the apps, how long they are using them, which buttons they click, where and when they use the apps and other usage statistics. This would enable the roosearch team to enhance the user experience and ultimately drive more business.

- Comment bank - It was previously envisaged that a comment bank would be implemented, however this was descoped from the prototype. As a future development opportunity this could be integrated with the survey mechanism so that in addition to submitting survey results, users can also post arbitrary text comments which will either be listed publicly on the survey, or remain private to the administrators depending on privacy configuration. Alternatively, with the social networking integration, a comment bank may be better suited to the social networking platform, such as enabling users to post to a facebook page using their facebook account, or to tweet under a specific hashtag. Although the roosearch administrators would have less control over these comments, it may be better suited to a socially driven application and with the appropriate hashtags or facebook links, could increase the visibility of the application and potentially lead to it going “viral”.
- Additional or premium features - Another route for generating revenue from the service would be to implement a premium / paid for service, whereby the administrators would obtain additional services for monetary payment. Extra features may include company promotion via internal advertisement. Other such features may include more personalisation on surveys, additional question styles such as free text answers or polling. Additionally, premium services could include menu printing, so rather than just accessing a QR code, it may be possible to generate entire menus for the restaurant owner.
- Marketing to get money - Roosearch was initially intended to be a free service, however when the popularity increases it is inevitable that resources would also need to be increased such as additional human resources to provide new features and maintenance, and material resources such as increased capacity on the hosting environments. Of course these costs would need to be covered, implementing advertising schemes would be one way to accrue some of the costs. It would be relatively easy to implement Google AdSense into both the mobile and web application to earn money based on advert clicks.

10.0 Reflections on Outcomes and Learning

The developer was satisfied throughout this project as he was able to recommend and advise on technology stack. Aside from adhering to the development designs and project plan, the developer was reasonably comfortable to decide on the implementation detail himself, such as which build frameworks to use, where to host the application and so forth. I firmly believe that a happy medium in regards to instructing a developer to develop an application, but also leaving them space to decide on the actual implementation results in a higher quality piece of software than would otherwise have been developed if they were constrained on every decision.

Leading a project of this scale and nature has given me the opportunity to understand what is involved with the software development lifecycle, and managing a developer. Prior to to commencement of the project I was sceptical of the effort that would be involved to develop such a system, but upon reflection of the project, I am able to understand why and how it was developed, which will benefit me on future projects.

This is my first experience of managing a real project from start to finish, and I was able to apply the knowledge I've learned from previous modules to approach the challenges of this project.

The project was a considerable learning experience for my communication skills, as I needed to liaise with an external developer to develop the roosearch prototype. In addition to communicating with the developer, I also need a level of communication with prospective users of the system, to gain an insight into their requirements and feedback. Communication was primarily achieved via email, with a significant amount of face to face discussions particularly. This was good experience for me as I feel that the skills developed will greatly aid me in future work. I would approach with task in a similar way to what I did with this one as I felt it was effective, however it does depend on the development and the client.

Managing the developer was reasonably easy, as an experienced developer working for a variety of blue chip clients, he was self motivated and able to manage his own work to meet the deadlines I imposed. I do expect that during my career I will encounter a number of developers with ranging skill sets and levels of experience.

Although I was not directly involved with writing the code for the applications (excluding the table creation and queries), I had regular meetings with the developer where we would converse on a technical level in regards to the tasks he was working on, so without a solid understanding of the programming languages used I was reasonably confident that I could have valuable input to the discussions. From a learning perspective, I have an understand of what developers do, the tools they use, and how they develop applications. This was particularly interesting with regards to the mobile element of the prototype, observing the developer deploy and test applications to a virtual

emulator and a real device.

I've learned that a plans and designs are not always entirely equivalent to the delivered end product. I believe that designs are often a starting point for a product, but during implementation issues may arise and the designed application may not be achievable for a number of reasons, such as the difficulty of the social networking integration and may need to evolve and adapt to the capabilities of the resources developing the software. Whilst I believe a plan is required to deliver a quality product, it is certainly not set in stone and the end result may differ slightly from the planned product.

In addition to being the first project that I have managed, this is also the first significant report that I have had to write, which has also been a learning point. This report has evolved considerably over the past few months and has been restructured a number of times in order to make the content flow and be easy to read. In hindsight I would have like to review more reports from other academic students so I could identify the common traits and patterns, and writing styles.

11.0 Conclusion

As a result of my actions above I have been able to prove and validate my aim which was developed in the initial plan.

“The aim of my project is to uncover if mobile phone applications can be developed to help businesses, with consumers, primarily restaurants, in the collection of market research.”

In the interim report I completed the research needed to establish if this was possible in theory and my final report has detailed the planning, designs and implementation I undertook, along with my developers implementation to validate the theory. It is possible for mobile phone applications to be developed to help restaurants in the collection and analysis of market research, although these are most efficient when used with web applications like Roosearch Research has been.

I proved my aim and hypothesis correct by researching, planning, designing, managing the development and completing some myself. Without the outsourcing of the application to an external developer, I would not have been able to validate that this is possible, only prove that it can be completed in designs. To help prove this I also tried to achieve my five objectives, however some of these I failed to achieve due to personal circumstances. Below are details of the objectives

1. To have the design of the application finished by the end of week 11, in the autumn semester, so I can plan the first iterative. The brief up front designs for iteration one were completed, along with the UML requirements documentation and time plan. As a result of this I was able to achieve this target.
2. Begin the outsourcing in the week commencing 31st December 2012. I handed the developer all of the documentation from objective one on this date so he could start the project in the new year.
3. To have two meetings per iteration to ensure progress is going well. We ended up having more than two meetings each iteration as we found it easier to show and see than explain this over email.
4. To have completed the development of the application and write-up regarding the details and justification of choices made, along with a progress report on how the development process progressed. This must be completed by the end of 31st March. Unfortunately I was not able to achieve this target as I had problems with my hand this term which prevented me from completing the report until the end of week nine, after

easter.

5. My aim is to have the final report finished by start of week 10, to allow time for proofreading and correction. Again I was not able to complete my report by this date due to problems with my hand, and so had much less time to proofread than I had previously planned.

If I was to approach a similar task to this in the future I would definitely start the write up earlier to reduce the risk of delays from problems with my hand. This was the main and only issue in this report, other than my developer not having enough time to fully develop the application, which was expected. In the future I would still design the system to be the best it can be, but would try and maximise the time we have available to allow as much to be developed as possible.

The findings in this report are important as they prove there is an opening in the market research market. If we produced the application to a high standard quickly we could be the first to market with this sort of idea. By achieving this we would be putting Roosearch Research in the best position to ensure its success; however with the feedback from potential users I feel that this research application could really support the restaurant industry in a user friendly way.

Throughout the course of my dissertation I have achieved a number of things. I am now more aware of what needs to be undertaken to smoothly take a project through the development lifecycle. From this I was able to produce a good quality set of designs and a reasonably good quality application to demonstrate in my viva.

Overall this has been a really good experience.

12.0 Glossary

Gantt Chart: Is a type of bar chart that illustrates a project schedule.

Class Diagram: A static diagram that describes the structure of a system through classes and relationships.

Entity Relationship Diagram: Abstract and conceptual representation of data.

Sequence Diagram: An interaction diagram that shows how processes interact with one another.

User Stories: A brief story of what the user does or needs to do as part of their job function.

Prototypes: The first or preliminary model of which other forms are developed or copied.

QR Code: A matrix barcode, readable only by a QR code scanner.

JVM: A JVM (Java Virtual Machine) is an environment where java applications can be executed. Generally speaking, java applications can be developed and once compiled should be able to be executed on a JVM on any machine (adhering to platform versions of course). JVMs are designed as an abstraction layer between applications and operating systems, so you can write once, deploy anywhere.

Plugin: Is a set of software components that adds specific capabilities to a larger software application, plugins are often associated with enabling software to integrate with external 3rd party software.

Iteration: The means of repeating a process with the aim of approaching a desired goal.

MVC:

MVC is a design pattern often found in web and mobile applications. It stands for Model View Controller. The idea is to separate responsibility and functionality, this brings a number of benefits such as easier testing, portability, and is often deemed good practice. The layers are summarised as :

- *Model*
 - This layer is designed to incorporate your business model, such as your domain objects and how they interact with each other. In our case this covers the questionnaire and survey domain classes (amongst others) and the business logic involved in mapping relationships between objects, what happens when you add a new survey, a new user and so forth.
- *View*

- The view is often the part that gets displayed back to the user, such as an HTML page. The view should be relatively “dumb” and be a means for displaying information to the user in a friendly manner, it should not do any complex logic on the model.
- *Controller*
 - The controller is the glue that combines the model with the view. When a user clicks a button on the view, the view should invoke the controller. It is the responsibility of the controller to determine if the action is allowed, and service it via interfacing with the model. Once complete, the controller then takes the model and a view (which may be different) and renders it.

One of the benefits of this design pattern is that each layer should be relatively decoupled from the others, which brings the opportunity to replace, or introduce different views, such as a view targeted for mobile browsers. The view would be different (different CSS etc to render nicely on a mobile device) yet the actual model behind it would be the same.

BDD:

Behaviour driven design is a software development approach loosely based on Test driven development. BDD is often seen as a path to enable business rules and requirements to drive software design. Use cases, or scenarios as they are referred to often follow the “given when then” style which enables stakeholders to easily define business requirements that can then be developed or “played” as stories.

TDD:

Test driven development is an agile development approach. TDD involves firstly writing tests for a new feature or defect fix and observing the tests fail, next the code change is implemented, the tests are run again and should pass. Code can then be refactored and the developer would still have confidence that the code meets requirements if it still passes the tests. TDD is a development iteration of “watch it fail, fix it, watch it pass, rinse and repeat”.

GORM:

The grails object relationship mapping framework enables developers to work against domain classes in the groovy language, and to annotate or describe how these can be implemented as database structures, without having to become involved in the database layer directly. It provides a layer of abstraction so that developers can persist, modify, retrieve and delete objects, but are not concerned about how the data is actually handled in the persistence layer. Mapping of relational data is also covered by ORM frameworks which can make development easier in regards to object relationships and structures.

CRUD:

Create, read, update and delete is an acronym often used by developers to group the 4 basic data operations. The roosearch prototype is a CRUD application because it creates surveys, reads surveys either individually or as a collection, updates surveys by adding new questions,

and deletes surveys or questions.

DML

Data Manipulation language, abbreviated to DML is often used to loosely group data manipulation scripts, such as SQL inserts and updates.

DDL

Data Definition Language, abbreviated to DDL is similar to DML except targeted towards the structure of database objects rather than manipulation of data.

13.0 References

- [1] Hybrid Mobile Application development Approaches: White Papers: Resources: TCS. 2013. [Online]. Available at: http://www.tcs.com/resources/white_papers/Pages/Hybrid_mobile_application_development_approaches.aspx [Accessed throughout April 2013].
- [2] Kavindra Kumar Singh: E-R Diagram Theory. 2013. [ONLINE] Available at: <http://kavindrakumarsingh.blogspot.co.uk/2013/04/e-r-diagram-theory.html>. [Accessed Throughout April 2013].
- [3] UML basics: The sequence diagram. 2013. *UML basics: The sequence diagram*. [ONLINE] Available at: <http://www.ibm.com/developerworks/rational/library/3101.html>. [Accessed Throughout April 2013].
- [4] 2013. [ONLINE] Available at: http://www.inf.ed.ac.uk/teaching/courses/seoc/2011_2012/notes/SEOC08_notes.pdf. [Accessed Throughout April 2013].
- [5] Introduction to Medium-Fidelity Prototypes | Inovdesigns | Experts in the Engineering, Art, and Psychology of User Interface Design. 2013. [ONLINE] Available at: <http://www.inovdesigns.com/prototypes.php>. [Accessed Throughout April 2013].
- [6] 2013. [ONLINE] Available at: <http://www.kaner.com/pdfs/GoodTest.pdf>. [Accessed Throughout April 2013].
- [7] IEEE standard 610 (1990) [Accessed Throughout April 2013].
- [8] Has been removed since used. <http://ezinearticles.com/?What-is-AD-Hoc-Testing?&id=2047543> [Accessed in March / April 2013].
- [9] Unit Testing. [http://msdn.microsoft.com/en-us/library/aa292197\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/aa292197(v=vs.71).aspx) [Accessed Throughout April 2013].
- [10] Shneiderman's "Eight Golden Rules of Interface Design". <http://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html> [Accessed Throughout April 2013].
- [11] Interface Design Rules. <http://homepages.inf.ed.ac.uk/da/teach/HCI/rules.pdf> [Accessed Throughout April 2013].
- [12] Grails - The search is over.. 2013. [ONLINE] Available at: <http://www.grails.org/>. [Accessed Throughout Development 2013].
- [13] Groovy - Home. 2013. [ONLINE] Available at: <http://groovy.codehaus.org/>. [Accessed

Throughout Development 2013].

[14] Gradle - Build Automation Evolved. 2013. [ONLINE] Available at: <http://www.gradle.org/>. [Accessed Throughout Development 2013].

[15] Maven - Welcome to Apache Maven. 2013. [ONLINE] Available at: <http://maven.apache.org/>. [Accessed Throughout Development 2013].

[16] Easy to use Maven plugin for Android™! - Google Project Hosting . 2013. [ONLINE] Available at: <https://code.google.com/p/maven-android-plugin/>. [Accessed Throughout Development 2013].

[17] Welcome to Jenkins CI! | Jenkins CI. 2013. [ONLINE] Available at: <http://jenkins-ci.org/>. [Accessed Throughout Development 2013].

[18] CloudBees: The Java PaaS Company. 2013. [ONLINE] Available at: <http://www.cloudbees.com/>. [Accessed Throughout Development 2013].

[19] Web Content Accessibility Guidelines 1.0. 2013. [ONLINE] Available at: <http://www.w3.org/TR/WCAG10/>. [Accessed Throughout April 2013].

References 13 to 18 were used throughout the Technology and Glossary sections.

PDF