**Honours Project Draft**

Kari McMahon

*Abstract - This project details the design and development of Recipes For Life an Android application for collaborative recipe management. For generations collaborative cookbooks and recipes have been a way for people to express themselves and share experiences and traditions with others through the alternative format of cooking. But as we increasingly move more and more into a digital age the concept of the collaborative cookbooks could soon be lost in the onslaught of digital information if it is not brought into the 21st century. Many recipe management applications are on offer but none offer the possibility to collaborate on cookbooks and recipes with friends, family and communities - a tradition which has been ingrained in society for many generations. Therefore this project aims to fit this gap in the market and presents a solution which brings the tradition of recipe books to the 21st century through a collaborative recipe management application.*

1. **Introduction**

A recipe book is defined as a “book of directions explaining how to prepare and cook various kinds of foods” (Merriam-Webster, 2014). Although many successful cookbooks do not just represent directions but they represent people’s experiences and their traditions which we can identify with and get excited about often through tantalising pictures, interesting descriptions and innovative ideas (Ruhlman, 2012). For generations collaborative cookbooks and recipes have been a way for people to express themselves and share experiences and traditions with others through the alternative format of cooking. But as we increasingly move more and more into a digital age the concept of the collaborative cookbooks could soon be lost in the onslaught of digital information if it is not brought into the 21st century. Currently there is no applications on the market that offers collaborative recipe management and therefore this project presents a solution to this an Android application called Recipes For Life. This solution aims to bring the tradition of recipe books to the 21st century through collaborative recipe management. The following report outlines design and development process of the solution as well as reflecting on the successes, challenges and lessons learnt from the various aspects of the project.

1. **Background**

Recipes and cookbooks inextricably link with fond memories. Whether it’s a memory of the recipe of your favourite apple pie you used to bake with your grandma, the delicious recipe for chocolate chip cookies you picked up at the charity bake sale or the recipe which holds the secret to your beloved penne arrabiata. Many of us store these memories away by placing our recipes on pieces of paper in the back of cookbooks, creating our own cookbooks and increasingly placing these recipes on technological devices. By storing these recipes it allows us to keep the recipes to look back on and alter as well as sharing with friends and family to develop new experiences and memories together. Although as we reach a point where many of us are storing and sharing our recipes using technology the possibility of collaborative cookbooks and recipes between groups of people could soon been lost. For generations recipe books have enabled others to add new recipes, alter recipes and view other recipes but as we move into this digital age we appear to be losing this tradition. This can be seen when browsing through the Apple and Android application stores there is a large amount of cooking and recipe management applications but none of these apps offer the possibility to collaborate on cookbooks and recipes with friends, families, clubs or even with strangers who have similar interests. The recognition of this sparked interest into the research into traditional cookbooks and the benefits they provide as well as research into the current recipe applications on offer to help understand the viability of the project being presented.

**2.1 Cookbooks As A Historical Document**

Recipes have been a part of society for thousands of years with the earliest recollection of recipes being the De Re Coquinara which dates back to the 5th century AD. Since recipes have played such a major part in society for so many years, cookbooks have often been seen as an informal historical document. Many papers have explored this topic including Mitchell’s paper Cookbooks As A Social And Historical document – A Scottish Case Study. The paper examined whether Scottish cookbooks published between 1890 and 1990 are historical markers of major events and technological advances in society. The paper found that “although cookbooks might not record events in society as historical facts nevertheless their contents are often a response to historical events.” (Mitchell, 2001). Similarly in the book Eat My Words: Reading Women’s Lives Through The Cookbooks They Wrote, the author examines cookery books from the US and UK from the 17th to mid-twentieth century. The book documents how women from “diverse backgrounds have found the homely cookbook a suitable place to record their stories and thoughts as well as their recipes.” The book aimed to tell untold stories of these women from the cookbooks the author examined as well as getting others to consider cookbooks as worthy objects of serious textual analysis. The book demonstrated how recipe books told the history of that time but also how we could still maintain a connection with these recipe books with the author making a strong point “How deep are the connections between us – a 17th century mother and myself –despite the time that has elapsed.” The book showed we could still connect with the individual even though they lived in a time that was polar opposite to ours by bonding through the medium of cooking and recipes (Theophano, 2003). These papers and books mentioned demonstrated how cookbooks can be seen as a historical document which can tell the story of the time and be beneficial in helping others understand that time period through the analysis of these documents. We can see that cookbooks appear to stand the test of time as we can still connect with the documents that are hundreds of years old. To lose cookbooks would be losing a form of documentation that tell us the stories of society during that time period, that tell us about the varying diets of different cultures around the world and that tells about the food and cooking trends during a specific time period.

**2.2 Traditional Cookbooks And Their Benefits**

As well as cookbooks bringing benefits to historians and researchers by acting as an informal historic document, the success of the cookbooks survival over so many years is the many other benefits they provide in particular the collaborative aspects. For example the paper Intensifying Taste, Intensifying Identity: Collectively Through Cookbooks aimed to look at the negatives of community cookbooks by examining whether community cookbooks implicitly rebuke a social order that devalues women’s work by focussing on Lutheran church women. But in fact the author found the process of creating a collaborative cookbook gave a voice to the community through building the cookbook by organizing it, discussing experiences and producing and selling them. The author found the cookbook was not just a way to raise money but also a way to recognise each of ladies knowledge and experience of cooking and share it with those who were interested. (Ferguson, 2012). Similarly the book Recipes For Reading contains a collection of essays that demonstrate the different stories available in cookbooks and in particular community cookbooks. The author states that cookbooks “tell stories – autobiographical in some case, historical sometimes and perhaps factious or idealized in other instances.” A common theme in the book was the way cookbooks were used to voice stories and experiences as the author mentioned recipes were increasingly becoming “readable with great benefit to our knowledge of women’s experiences and discourses” (Bower, 1997). Although the benefits of community cookbooks can be best summarized by the book Eat My Words - “modifications and modernizations of old recipes and the invention of new dishes in a woman’s cookbook represent the combined effort of many people. Contributions may come from past circles, sometimes from one or more cultures and while we tend to think of cooking as a delight to our senses, the relationships formed through these culinary compositions are social, cultural and economic.” (Theophano, 2003). The benefits that are often re-iterated in the papers and book is that cookbooks enable us to have a voice, tell a story and share our experience and knowledge with others. Collaborative cookbooks have often been a way for communities to come together and bond and so to suddenly lose these benefits to the digital age would be a detrimental loss to cooking and to society. Therefore it seems necessary to try and move collaborative cooking into the digital age.

**2.3 Current Recipe Applications**

AllRecipes.com an extremely popular recipe site states that one-third of online cooks use smartphones to look up recipes. They found that 44% of cooks preferred using cooking websites over 19% who preferred to use traditional cookbooks and in the past 15 years the use of cooking websites have surged 207% (All Recipes, 2012). It is clear that technology is becoming the top cooking resource and although traditional collaborative cookbooks have been seen to have many benefits, they are beginning to be lost in the mass offerings of technological cooking resources. Therefore background research then turned to focus on cooking and recipe management on technological devices. The research was predominantly focussed on tablets as they were the devices that appeared to be increasingly more in use in the kitchen with AllRecipes.com stating in 2013 that social referrals came from tablet devices were up 787% from 2012 to 2013 as well as an increase in page views on the website from tablets (All Recipes, 2013). The recognition of the increasing amount of popularity for using tablets in the kitchen can be seen by the amount of kitchen accessories on offer for tablets such as tablet kitchen stands, covers to protect from spillages and styluses to use when cooking instead of touching the screen with dirty hands, an example of this can be seen in figure 1. With accessories like these making their way onto the market and the statistics seen from allrecipes.com tablets seem the way to go when creating technology applications for the kitchen.



*Figure 1: Belkin Kitchen Tablet Mount*

The research was focussed on the applications available for tablets. These applications often fall into two categories either recipe discovery or recipe management with occasionally some apps merging the two of the categories. Below is an analysis of a few of the most popular cooking apps on the market at the moment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Application Name** | **OS Type** | **Description** | **Features** | **Downfalls** |
| All The Cooks | All OS’s | A social cooking application | Enables users to find new recipes, write reviews as well as add new recipes and a few neat features like shopping lists and conversion features | No collaborative feature |
| Big Oven | All OS’s | A strong recipe management and discovery application. With over 8 million downloads and has won several awards. | Import recipes from websites  Take an image of a recipe and convert to a recipe. | No collaborative feature.  You have to pay for some functionalities. |

MORE ANALYSIS HERE.



*Figure 2: Current applications*

Many of these applications are very popular with a high amount of downloads and star ratings. But all of these applications lacked a collaborative feature that enabled users to set up and maintain shared cookbooks. The research showed there is room on the market for the project and that recipe apps are highly desired and used by a large amount of the population. The research also enabled us to see the successes of the current apps and what users currently like and don’t like with these apps which could work in addition to the collaborative features of this project.

1. **Specification**

This project was the idea of the University of Dundee Applied Computing student Kari McMahon who also undertook the design and development of the project. Therefore this project had no client interaction to gain the specifications for the project. The core aim of the project was to create a minimum viable product that enabled users to collaborate on recipes together in shared cookbooks.

Since the project had no client interaction to meet the core aim of the application the individual student had to create the specifications for the project based on own ideas, data gathering and research.

* 1. **What Is A Minimum Viable Product?**

A description of a minimum viable product can best be described by technopedia:

*“A minimum viable product (MVP) is the the most pared down version of a product that can still be released. An MVP has three key characteristics:*

* *It has enough value that people are willing to use it or buy it initially*
* *It demonstrates enough future benefit to retain early adopters*
* *It provides a feedback loop to guide future development*

*The catch to this development technique is that it assumes that early adopters can see the vision or promise the final product and provide the valuable feedback needed to guide developers forward. “*

* (Technopedia, Unknown)

By creating a minimum viable product it enables an individual to gauge the market need for a product by trying to maximise the amount of learning for the minimum amount of engineering hours. A minimum viable product helps the developers create products that are market led and that are desired by consumers by accelerated learning. This project aims to present a minimum viable product that demonstrates a collaborative recipe management application. The challenge in creating the specifications for this mvp is that there is a large amount of recipe applications on the market with a significant amount of features on offer. This already sets up a high expectation for the standard of recipes apps on the market. So the challenge in creating the specification for the product is setting minimum requirements that are do able in the time frame and are primarily focussed on the core aim of the product but also having enough functionality to put the app on a level playing field with current recipe applications.

* 1. **Initial Specification**

The initial specification for the project was developed based on the students own ideas and research from applications on the market. The specification document type selected for the project was of the format suggested by the IEEE requirements guidelines. The choice of using the formalised IEEE requirements over creating user stories[[1]](#footnote-1) was that the specifications were coming from the individual student and not the client therefore in this case formalised requirements seemed more appropriate than user stories which are very customer oriented.

The initial requirements were an informal list of requirements. The requirements were in two categories functional and non-functional and were set out with shall, should and may. Shall meaning the requirements that should definitely be developed. Should meaning the requirements that should be developed if there is time and may meaning the requirements that could be may be developed if there is time. The initial requirements proved as a starting point for understanding the applications functionalities. With a list of possible application requirement functionalities gathered from research, a survey was then created to learn more about the demographics for the application and to help prioritise the functionalities and find any new functionalities based on the target markets needs and desires. An example of these initial requirements can be seen below:

The application shall enable users to create account.

The application shall work offline

*Figure 3: Requirements specification example for functional and non-functional requirements*

* 1. **Data Gathering**

A survey was placed on the website SogoSurvey.com. SogoSurvey was selected over other popular survey sites like Google Forms or SurveyMonkey as it was free to use, gave a selection of different questions types and provided good analysis tools. The survey asked for anyone over the age of 18 who was interested in cooking particularly those interested in the use of technology when cooking to fill out the survey. The survey was distributed via email, on cooking forums and through communication with those who were interested in the project. The survey aimed to understand people’s interests in using technology to cook with and the devices and applications they use to do this to help better understand the market being targeted. As well as present the 17 possible functionalities set out in the initial specifications to be rated in order of usefulness as well as welcome any other functionality suggestions.

The survey received 19 responses with 13 of the participants being female and 6 being male with the participant’s ages ranging from 18 – 78. Several of the participants in the survey were from a cooking background where their occupations were a pastry chef, dietician and private chef. A summary of some of the question responses can be seen below.

|  |  |
| --- | --- |
| **Question Summary** | **Response Summary** |
| Percentage of participants who frequently use recipe books, apps, recipe websites | 63.69% responded to frequently by selecting 4 or 5 in the scale  1 (Not at all) – 5 (All the time) |
| Percentage of participants who use smartphones or tablets in the kitchen | 78.95% stated they used smartphones or tablets in the kitchen |
| Percentage of participants who were interested in using the project being presented | 47.4% responded yes  26.3% responded possibly  21% responded no |

*Figure ?: Summary of some of the survey responses*

The responses from the survey showed the participants gathered had a clear interest in cooking using technology, the application being presented and from a wide range backgrounds. Therefore they seemed the right match to take into consideration their rating of the applications features. These ratings were then used in connection with the initial requirements document to create the final requirements document. Some interesting additional features were also suggested in the survey such as calculate ingredient pricing and calculate nutrition information which weren’t in the initial document. The data gathering was really beneficial to backing up the interest in the product and helping to create a market led requirements document. The full survey results can be found in the appendix.

* 1. **Finalised Specification**

The finalised specification was then created based on the data gathering from the survey, the student’s ideas and research. This was done by a frequency table generated by SoGoSurvey which showed the amount of people who ranked a certain requirement at that ranking e.g. rank 1 is the most useful requirement. This was then considered with the initial requirements document to create the final document. Some of the requirements that were not ranked so highly by the participants ended up high in the final requirements as they were essential to the creation of the application. The requirements in the document that are marked as shall are the requirements essential to creating a minimum viable product. The full specification document can be found in the appendix and an example requirement is seen below in figure ? .

**R5 Cookbook Privacy**

**Description:** The user shall be able to set cookbook to private or public

**Rationale:** This is essential as it gives users the choice of who is displayed to the public

**Risk:** High

**Priority:** High

*Figure 4: Example of finalised requirement*

* 1. **Creating Specification Flexibility And Managing Requirements**

Creating a formalised requirements document as discussed in section 3.4 is often associated as something rigid and fixed. Since the project was an individual’s idea and not a contractual agreement with anyone this enabled some flexibility in the document. To help create flexibility the technique of using a task board which is popular in the Agile methodology was used. The flexibility was created by splitting the formalised requirements document into smaller requirements in a user story style and then storing them in an online task board called Trello in the same priority as listed in the document. The Trello task board allows for flexibility as you can easily move a requirements priority around based on changes in the project and therefore this makes the project more adaptable to change.

The task board was used to manage the project requirements for the rest of the project after the finalised specification was set. This meant requirements for the project priority could easily be moved around based on new information when developing or redesigning aspects of the application. This flexibility helped create a project that was led by information gained throughout the process instead of requirements simply being fixed and based on assumptions at start of the project. An example of a scenario when this was really useful is the requirements for R1 Recipe Management was above the requirement R2 Account Creation and when it came to the development stage it was clear to see that creating an account should be developed over recipe management as a user’s account was often linked with recipe management. Splitting the requirements into smaller more manageable requirements seen in figure 4 was also really useful as it split tasks into easier to manage chunks and enabled more flexibility for example in the time scale of development it appeared that creating a recipe and then creating a cookbook were more important tasks to achieve than deleting a recipe. So having the requirements split up like this on the task board allowed an easy change of priorities. As time went on the project sometimes it was clear requirements needed to be added as that were not initially thought about in the final specification document and so the task board enabled an easy add and removal of requirements based on new information.

**In the document:**

**R2. Account Creation**

**Description:** The user shall be able to create an account.

**Rationale:** This is necessary as it allows the user access to the application.

**Risk:** High

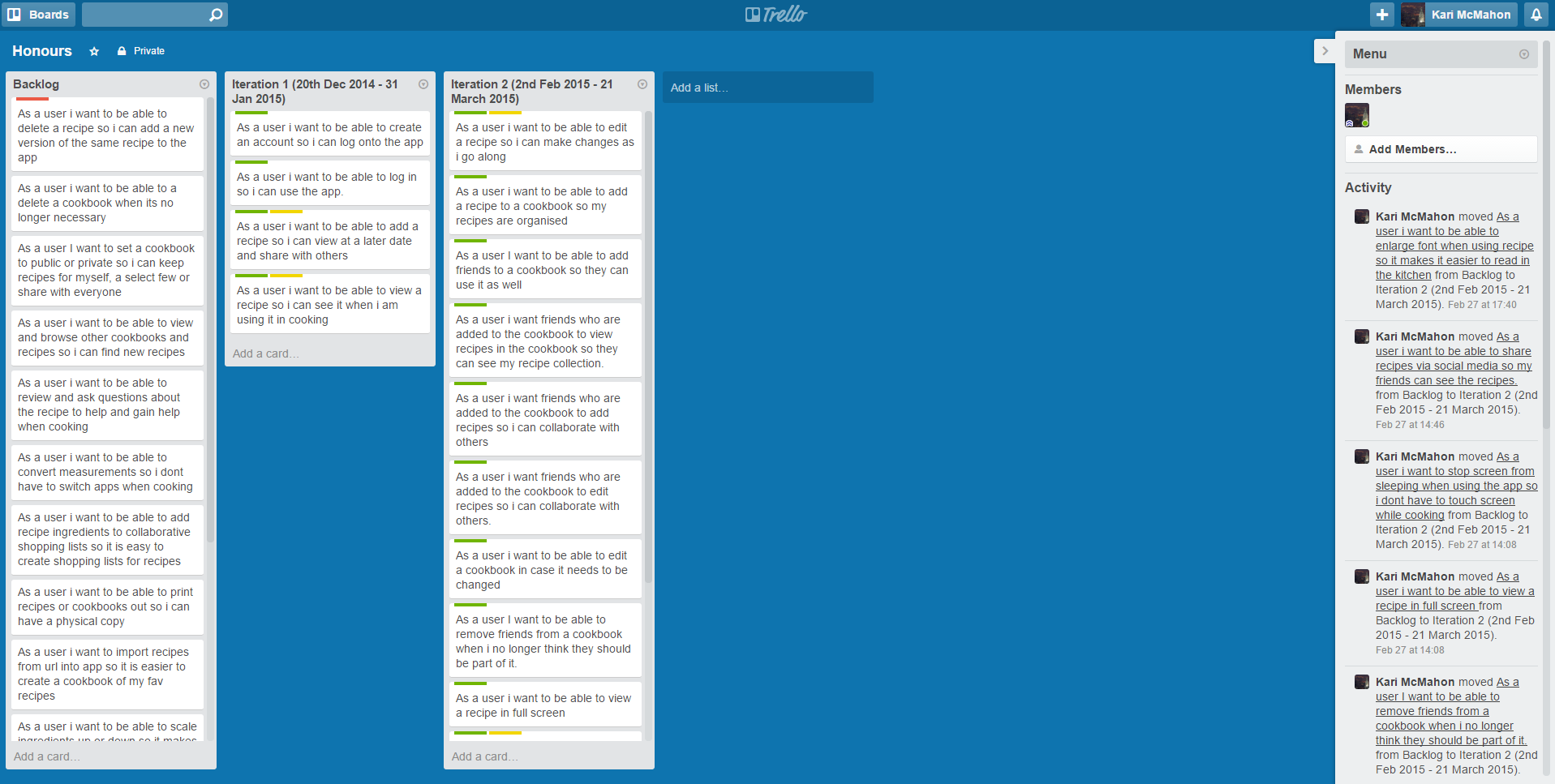
**Priority:** High

**In the task board:**

As a user I want to be able to create an account so I can log onto the app

As a user I want to be able to log in so I can use the app.

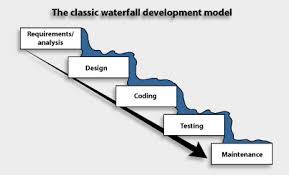
*Figure 4: Contrast of document and task board requirements*



*Figure 5: Trello task board for project*

1. **Project Management**
   1. **Methadology**

A software development methodology is a “framework that is used to structure, plan, and control the process of developing an information system” (IT Knowledge Portal, n.d.). Recipes For Life is a large project and it was necessary to select a software development methodology that would help structure and maintain the project. There are two core software development methodologies – waterfall and agile. Waterfall is the traditional software development process, it is a linear and rigid approach that does not embrace the inevitable changes or revisions that often occur within projects (IT Knowledge Portal, n.d.). Whereas Agile is a set of development processes which are flexible to change, encourages working code over documentation and frequently take opportunities to assess the direction of the project throughout the development lifecycle (Highsmith & Cockburn, 2001). The Agile approach enables a flexible project which can be evaluated honestly and realistically through interaction and collaboration at frequent intervals. An Agile approach was most suitable for this project as requirements, design and evaluation could easily change as the project was explored in more depth and therefore a software development process that enabled flexibility was essential. The most popular Agile methodology is SCRUM which is a very customer centric approach and is based on a lot of customer and team collaboration. This project lacked an official customer and was being developed by an individual so it was necessary to find a flexible Agile development approach that was this less client and team oriented. An Agile approach that was suitable for this was the iterative development process. The iterative development process allows you to develop a system in iterative cycles. The process starts with an initial planning stage where the initial requirements and design are set. Then iterations essentially “mini-projects” cycles occur until the product is ready for the delivery. At the end of each iteration the individual or team has an opportunity to re-evaluate the project and re-organise and change aspects of the project to the fit the needs of the team, individual or client. The iterative approach is very flexible and helps creates a final product that is user centred and information led (Bittner & Spence, 2006).For the needs of the project the iterative approach was built on to create an iterative methodology strategy appropriate for the project, the diagram in figure ? outlines this strategy.

[[2]](#footnote-2)[[3]](#footnote-3)

[[4]](#footnote-4)

*Figure 6: Images representing waterfall, scrum and iterative software development processes*



*Figure 7: Project Iterative Strategy*

**4.2 Project Deliverables – Necessary ?**

The main deliverables for the project was the recipe application for Android devices, the server side SQL database and the C#/ASP.net code which inserts or retrieves a JSON of database details to sync the phone and server. Several other deliverables were to be handed in for the project:

* Requirements document
* Gantt chart
* User manual
* Proof of testing and evaluation.
* Source code for the application and server side code.
* Poster and presentation
* Log book.
* Supervisor minutes.
* Ethics documents.

As well as any other documents the student used throughout the project and felt was relevant to the hand in. These deliverables are all included in the appendix.

* 1. **Project Management Tools**
     1. **Initial Project Plan, Gantt Chart & Trello**

At the beginning of the project an initial plan was drawn out by the student to help outline the overall project view between September 2014 and May 2015 to the project supervisor. This document was then developed further into a Gantt chart at the start of the project. The Gantt chart enables an individual to help visualise tasks and milestones for the project over the year. The Gantt chart was predominantly used as a project overview which was updated and reviewed every so often to help the student understand whether the project was on track and if changes needed to be made to aid progress. The gantt chart changed over many iterations and figures ? and ? show the difference between the first and second gantt chart.

BEGINNING GANTT CHART FIGURE  
FINAL GANTT CHART

For a lower level view of overall project progress trello was used which was also mentioned in section 3.5 . Trello stored all the requirements for the project and whether they had been achieved, in progress or still to be done. Trello enabled a quick analysis of whether the project was on track just by having a glance at the amount of requirements achieved at that point in time.

**4.3.2 Sprint Backlogs**

A sprint backlog is a list of the tasks and requirements to be completed within the sprint (Layton, 2012). For this project sprint backlogs were used for the 2 week development sprints that occurred within the iterations. Requirements would be selected from the product backlog for a 2 week sprint and then these requirements would be split into small tasks on the sprint backlog with an estimated time to spend to achieve these tasks. Each day in the sprint the amount of hours spent on the task would be logged and monitored. The sprint backlog helped show the progress of development and helped the student analyse the amount of time being spent on requirements. This analysis helped inform later requirement selections for other sprints and commitments for project as the student could see on average how quickly they developed requirements. The sprint backlog is also very flexible because if an individual underestimates what they can do in 2 weeks they can easily add new requirements. Sprint backlogs are really simple and effective way to monitor and analyse development progress by giving a snap shot of day to day progress.

IMAGE OF BACKLOG

* + 1. **Risk Assessment**

For a project to be successful it is essential to outline the possible risks and potential problems within the project at an early stage and create contingencies plans for these risks. This is a common practice within the software development industry. A risk assessment was created at the research and design stage of the project. The risk assessment outlined the risks for the project in order of priority with contingencies for the risks. For the top 3 risks two contingencies are listed and for the rest of the risks one contingency is listed. The prioritisation of the risk is based on the likelihood of the risk occurring times by the loss if the risk did occur. By planning contingences early it allows you to put systems in place to limit the risks for example one risk would be losing the application source code files and so to limit this a contingency would be to have an area where the files are backed up and this area should be used frequently. A risk assessment is a really good practice that can help individuals mitigate risks that could have had a serious impact on the success of the project.

RISK ASSESMENT

* + 1. **Supervisor Meetings And Minutes**

Supervisor meetings were scheduled once a week for the majority of the project. A supervisor meeting is a useful way to help manage the project because you have an outside perspective. Another perspective is useful when the student is struggling with challenges that are halting project progress as a different perspective often gives new suggestions and strategies that hadn’t originally crossed their mind. At each meeting you have a chance to present your work and because of this the supervisor can give advice and guidance on the student’s project or their progress. This is a great way to address any progress or project issues early on. Each meeting gives the student an opportunity to learn from the supervisor’s knowledge and experience which is a great tool to have. Minutes were kept to enable reflection on these meetings and can be found in the appendix.

* + 1. **Log Book**

A log book was maintained throughout the project as an informal diary of work. The log book enables the student to write about the tasks achieved on a day to day basis as well as their rough notes, ideas or research throughout the project. The log book was a really useful and informal way to document the daily progress and vision of the project.

**4.3.6 Github**

For the project it was necessary to have an area to manage all the files for the project incase anything went wrong and files needed to be recovered - Github was selected for this. Github is a versioning control tool.It enables access to files wherever there is an internet connection, the ability to revert to old versions and the ability to store a range of files whether it’s code or a word document. The student already had experience with Github from past projects and has a private account on the site. Github is also well supported with over 6 million people using the site (Github, n.d.). Therefore there is a lot of support available if any difficulties were to occur during the project. For these reasons Github seemed the appropriate choice for file management. Although github is not just a file management tool, it is also a great tool for managing projects. The daily commit messages enable an informal log of day to day work and visualisations are produced on Github based on commits outline the students’ progress. This can be very useful for understanding project progress and understanding an individual’s work load and work amount. The figures below show some of the visualisations that are produced.

* + 1. **Ethics maybe ?**
    2. **Realities Of The Methadology ?**

1. **Design**

Before implementation could start on the project several design decisions which are discussed in the following sections.

**5.1 Operating System Selection**

The application for the project is being developed predominantly for tablets as they are increasingly becoming more popular for use in the kitchen. This meant a tablet operating system needed to be selected before development. In terms of tablet operating systems there are three main options you can develop for which are Android, IOS or Hybrid / Web apps. In 2014 IDC.com stated the worldwide smartphone operating system market share for Q2 2012 was 84.7% for Android, 11.7% for IOS and others making up 3.7% of the market

share with the predominant vendors being Samsung with 24.4% market share and Apple with 11.7% market share. IDC.com stated the tablet OS market share was 53.8% Apple and Android 42.7% (IDC, n.d.), (IDC, 2014), (McCracken, 2013). In terms of the two major operating system competitors it is quite a close call between them but Android comes out slightly stronger having a wider overall reach over Apple. This data was taken into consideration with the advantages and disadvantages of each operating system based on previous experience and research. These advanatages outlined in the figures ? and ? below:

**Figure 1: Advantages of Android, IOS and Native Web Apps[[5]](#footnote-5)**

|  |  |  |
| --- | --- | --- |
| **Android** | **IOS** | **Hybrid / Web Apps** |
| Already experienced in Android development | Large amount of resources available for help | Works on both operating experience |
| Not as many recipe applications available on android hence more market share for the application | Popular operating system | Has some experience of html/css with minor javascript experience |
| Large amount of resources available for help | Standardized marketplace | Save time as they port to multiple platforms |
| Standardized market place | Easier to build nicer UX/UI features | Merge web and native features |
| Easier to build nicer UX/UI features | Make use of own hardware and software features | Consistency between apps |
| Make use of own hardware and software features |  |  |

**Figure 2: Disadvantages of Android, IOS and Native Web Apps1**

|  |  |  |
| --- | --- | --- |
| **Android** | **IOS** | **Hybrid / Web Apps** |
| Only covers one OS | Only covers one OS | Little help available, it’s relatively new in comparison to Android or IOS |
|  | No experience with ios or objective C development | Complex to set up and fidgety |
|  | Need to own a mac to develop so would only be able to work from computing building | No centralized market place. |
|  |  | Often work arounds are needed when porting to different apps |
|  |  | Web apps cannot work offline |

Android was the leading operating system in market share in smartphones in last year as well as having the largest amount of advanatages with the least amount of disadvanatages based on research so Android was a natural choice to develop for. Although hybrid/web apps were also a strong choice but the main reason for not selecting this option was the lack of support available at the moment was a concern especially for a large project like this. The device that will be used to test and present the application on is a Samsung Galaxy Tab S4 as Samsung is the largest Android vendor according to IDC.com (IDC, 2014), so it felt best to test on a Samsung device.

**5.2 Server Side Design**

**5.2.1 Database Choice**

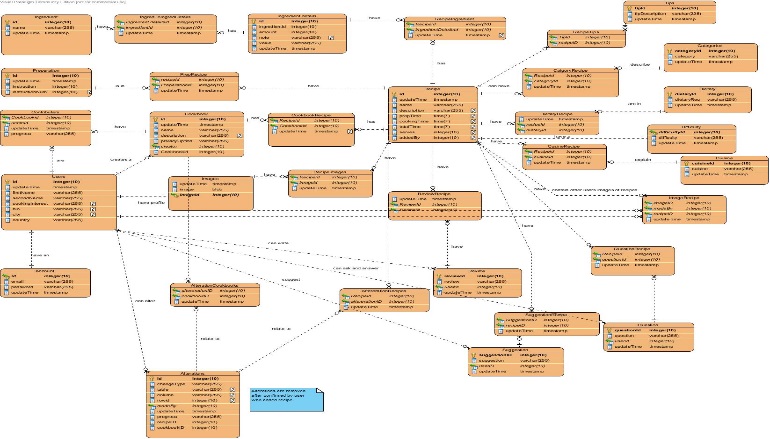
Database consideration was a large section of the design stage of the project as the application will be handling a lot of data from multiple users. Android devices have a SQLite database built into the phone which enables individuals to retrieve data from the database without having to be connected to the internet this means a responsive application as there is no wait times trying to retrieve data which can also work offline. It was essential for the application to work offline as the users should be able to access recipes or grocery lists when offline but when there is internet available they should also be to access other individuals recipes that have been added to the application. This meant having a database on the server and the database on phone and having a sync functionality between them.

Deciding the type of database which would be used on the server was an important consideration. Based on the knowledge that Android has a SQLite database built in, a SQL database server side seemed the natural choice. But research found that CouchDB offered a NoSql database on the Android device. At first glance NoSql seemed like the best option for the data in the application as using SQL meant there was going to be a large amount of tables and quite a few complex joins. But then with closer analysis it seemed harder to visualise queries necessary for the application with CouchDB and with it being a relatively new type of database and lacking support and documentation for Android it just did not seem the right fit for the application. Therefore a SQL database would be used on the server side and a SQLite database on the Android device.

SQL or SQLite

**5.2.2 Database Design**

After the database was selected the tables and columns for the database was created. The database design went through quite a few iterations as the application design developed. The recipe table is connected to many tables like preparation and ingredients which involve linking tables as recipes can often have more than one preparation step or ingredient. This type of design created quite a significant amount of tables. The design of the database involved all possible tables needed for the application but not all tables were used based on what could be implemented in the time frame. All tables needed a timestamp as this would be used to track differences in date between the application and the server which would be essential to syncing changes for the database. This database design is used for both the SQL and SQLite database.



**Figure 3: Database Design**

**5.2.3 Language Selection & Development Tools**

On the server side it was necessary to have some form of server side script that would insert and retrieve data from the database. In the past the student use PHP to retrieve data and insert data into the database and found this simple and easy to use especially when converting the data into the data exchange format JSON. Based on this past experience the student felt it was appropriate to use PHP again for this application. Because not only is the experience beneficial to using it ……….. Discuss PHP advantages. Discuss consideration of ASP.net

Notepad for PHP. MSSQL for managing the database.

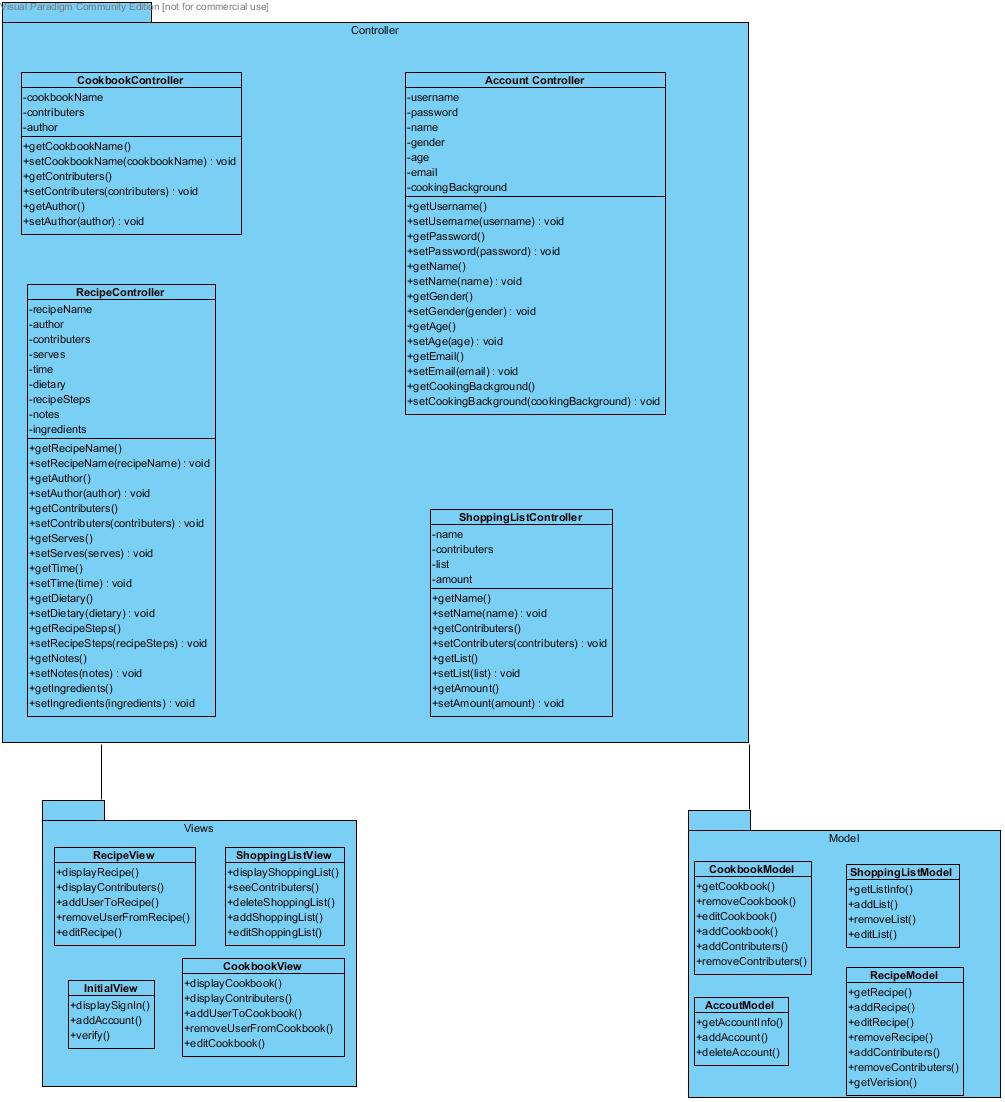
**5.3 Application Design**

**5.3.1 Application Layout & Design Patterns**

The application will be designed with a Model-View-Controller pattern as the application is heavily database focussed and the MVC pattern enables separation of the business logic and view. The benefits this pattern brings is that it limits code duplication and enforces code re-use making code more flexible and easy to test functionality independently (Kotek, 2002). The application will have classes split into the packages model, view and controllers (beans). Figure ? demonstrates the model view controller design pattern.



The application will have a model package which contains classes that purely send and retrieve data from the database. The controller will be information beans which will store information that is being sent too or retrieved from the database. The view will have all the classes (activities) which the user will interact with and retrieve the data from user input which will be stored in the controller. The model view controller pattern creates a strong separation of business logic and view. This pattern is demonstrated in the UML class diagram created for the application which is in figure ? . This class diagram represents the classes that will be thought to be used over during the implementation at the design stage.



**5.3.2 Language Selection & Development Tools**

Java

Junit

Android Studio being an intellij editor which is set to replace Eclipse at some point in the future but is currently in its beta stage. The choice for the project was Eclipse due to the fact Android Studio is still it’s beta stages and may be less stable than Eclipse which is an important factor when taking on a large project, it’s better to work with something stable and has support.

The emulator selection for the project is Genymotion. Genymotion is an android emulator which is trusted by 1500000 developers (Genymotion, n.d.). This is the alternative in comparison to the Android emulators provided and from past experience of using both there is a significant difference in speed and Genymotion is a lot quicker to use.For unit testing Junit will be used as the android test suites are based on Junit (Android, n.d.) And refactoring would be done on the project based on the refactoring rules that can be found in the appendix.

**5.4 Application & Server Communication**

When developing an application that works offline but also needs to retrieve information from other users it is necessary to use the in phone sqlite database which syncs with a on server database which stores all the individuals data. To show how the sync should work a diagram is displayed in figure ? .



This system involves having a database on the server and a sqlite database on the phone. Each table in the database stores a timestamp and these timestamps are used to find rows that need to be sent or retrieved between the database and the device when internet was available. These rows are often sent through a data exchange format between a server side application and the device. To make the syncing possible it was important to select a data exchange format that would be easy to use within the application and server side code. The two core choices were XML or JSON.

Discuss XML or JSON.

**5.5 Refactoring & Testing Strategy**

**5.6 Paper Prototypes**

Paper prototypes was the best way to visualise the application in the initial stages. Paper prototypes are the best way to form a design for an application as they are simple to create, can be rough drawn and easily thrown away ( ref ) so therefore are the best way to try and draw ideas and designs for the application. Several iterations of paper prototypes were created until the design seemed right.

IMAGE OF PAPER PROTOTYPES.

**5.7 Design Sketches**

The final paper prototype was then built on using Axure. An Axure prototype allows the creation of a design sketch that looks more like an application. This helps better visualise how the application would look in real life as it gives the individual the ability to play around with fonts, images and button styles. Seeing aspects of the application in this format helped change areas of the design and helped the student pull out more ideas and possible challenges with the application. It was also useful because as the research into the application and other apps got more involved aspects of the design could be changed on Axure which made it a quick to re-evaluate and change the style of the application. These design sketches did not just help the student form a vision and style for the application but also was used as way to see how individuals would respond to the application via a design focus group.

****

**Figure 4: Design Sketches**

**5.8 Design Focus Group**

**5.9 Ethics (Might be moved or removed)**

The project/application is user centred so to be able to design and implement the application based on user feedback an ethics form has to be created and submitted to the ethics committee. An ethics form outlines the various techniques involving users that will be to create a user centred application. The techniques chosen were an anonymous survey, interviews/focus groups, user testing and evaluation. An anonymous survey was used to gather information about the target market and help gather/prioritise requirements for the application, interviews/focus groups to gain opinions on design sketches and application idea, user testing to understand how usable and easy to understand the application and an evaluation against a popular recipe app to see if it could compete on the market. Ethics enables you to undertaken user studies in the correct manner.

Participant gathering needs to be added …..

**5.10 Intended Audience & Persona’s To Represent The Audience (Might be moved or removed)**

Before developing the application focus was put on understanding the target market that would be using the application. This was done through market research through usage of popular cooking apps, surveys and interviews and online research. A summary of the target market that would use this application is users of any age and gender who are competent with technology and have used technology in the kitchen to some extent whether its recipes from the internet, digital scales or timers or cooking apps.To help demonstrate the type of users that would use the application personas were developed to be used throughout the process. An example persona is shown below and the rest can be found in the appendix. The

**Design changes**

**Implementation And Testing**

**JSON – Example of JSON**

**GUI Challenges – Dialogs, Action Bar, Typeface, Making it unique**

**Background process**

**Errors in syncing**

**Changes database design**

**Handling deletion and edits in syncing**

**Orientation**

**Application**

**Server Side**

**Challenges**

**Achievements**

**Things to be improved**

**Refactoring**

**Unit Testing as you go**

**Action bar support**

**Responsive deisgn**

**Images of app**

**Obsecure Errors**

**Unique id**

**Security**

**Types of testing**

**Debugging challenges**

**Separate database for unit testing**

**Evaluation**

**Design sketch sessions**

**Usability.**

**Evaluation against another app**

**Nielsins Usability Heuristics ?**

**Prototypes**

**Computational accuracy e.g. times etc**

**Final Product**

**Critical Appraisal**

**Summary & Conclusion & Future**

**References**

**Appendix**

**Acknowledgements**

1. ## User Story - A user story is a tool used in Agile development to capture a description of a software feature from an end-user perspective. (TechTarget, n.d.)

   [↑](#footnote-ref-1)
2. https://snailonabike.files.wordpress.com/2010/05/waterfall.jpg [↑](#footnote-ref-2)
3. http://www.mountaingoatsoftware.com/system/asset/file/17/ScrumLargeLabelled.png [↑](#footnote-ref-3)
4. http://agile-development-tools.com/wp-content/uploads/2010/10/iterative-development1.png [↑](#footnote-ref-4)
5. (Budiu, n.d.)**,** (Gorbsky, 2013)**,** (McCracken, 2013) [↑](#footnote-ref-5)