

Recipe Hub

it pROJECT REPORT FOR DEVELOPING A MOBILE APPLICATION FOR THE STORAGE AND SHARING OF RECIPES

2018

Contents

[Chapter 1: Introduction 3](#_Toc511244885)

[1.1. Current Issue and Purpose 3](#_Toc511244886)

[1.2. Scope of the project 3](#_Toc511244887)

[1.3. Objectives and questions 4](#_Toc511244888)

[The Aim 4](#_Toc511244889)

[Objectives: 4](#_Toc511244890)

[Research Question 4](#_Toc511244891)

[Chapter 2: Measurable Organizational value (MOV) 5](#_Toc511244892)

[2.1. Identifying the applications desired area of impact 5](#_Toc511244893)

[2.1.1. Customer 5](#_Toc511244894)

[2.1.2. Operational 5](#_Toc511244895)

[2.2. Identify the application’s desired value of the IT Project 5](#_Toc511244896)

[2.3. Identify and develop an appropriate metric 6](#_Toc511244897)

[2.4. MOV Statement 6](#_Toc511244898)

[Chapter 3: Literature review 7](#_Toc511244899)

[3.1. The limitations of traditional recipe storage and sharing methods: 7](#_Toc511244900)

[3.2. Current Solutions in the market 7](#_Toc511244901)

[A bit of Background 7](#_Toc511244902)

[Paprika 7](#_Toc511244903)

[Big Oven 8](#_Toc511244904)

[Pinterest 9](#_Toc511244905)

[MacGourmet + Gourmet Mobile 10](#_Toc511244906)

[myTaste Recipes 11](#_Toc511244907)

[3.3. What makes a great mobile application? 12](#_Toc511244908)

[Polished Design 12](#_Toc511244909)

[3.4. Comparison Table 12](#_Toc511244910)

[3.5. Added Value and Functionalities 13](#_Toc511244911)

[Chapter 4: Research approach / Methodology 14](#_Toc511244912)

[4.1. Languages and Programs to be used 15](#_Toc511244913)

[Chapter 5: Design and Analysis 16](#_Toc511244914)

[5.1. Potential Risk Management 16](#_Toc511244915)

[5.2. Requirements 17](#_Toc511244916)

[5.3. Design and Analysis – GUI Initial 18](#_Toc511244917)

[5.4. Design and Analysis – GUI Current 20](#_Toc511244918)

[Chapter 6: Implementation and Testing 23](#_Toc511244919)

[6.1. Coding and Errors 23](#_Toc511244920)

[6.1.1. HomePage 23](#_Toc511244921)

[6.1.2. Room Persistence Library - Database 23](#_Toc511244922)

[6.1.3. Add Recipe 24](#_Toc511244923)

[6.1.3. My Recipes and Recipe Item 24](#_Toc511244924)

[Navigation Bar – Bottom of Application 25](#_Toc511244925)

[Other 26](#_Toc511244926)

[Chapter 7: Future WORK AND Conclusion 27](#_Toc511244927)

[References 28](#_Toc511244928)

[Appendix 30](#_Toc511244929)

[Appendix A – Gantt Chart 30](#_Toc511244930)

[Appendix B – Work Breakdown Structure (WBS) 30](#_Toc511244931)

# Chapter 1: Introduction

## Current Issue and Purpose

Cooking and Baking has always been a major part of our society, but with the rapid development of technology we believe that the old-fashioned Cook books are becoming very tedious and time consuming to navigate through. So, we have decided to offer a proposal for developing an application to allow a user to store their own personal and favorite recipes on one central location. This, coupled with an easy to use searching functionality will help solve this issue of time consuming navigation throughout piles of recipe books.

Since mobile technology is becoming the core of today’s technology era, we believe that not only allowing for the storage of recipes for personal use, but also for sharing benefits can add a tremendous addition to our application. Thus, our idea being that we develop a Recipe Book Hub.

We have searched if these applications exist and we found that they do, an article (Steen, 2016) about the ’12 Food and Cooking Apps Absolutely Everyone Should Have’ displays and describes their opinion of the 12 best recipe and cooking apps. After learning this we found a clear feature that all these applications lacked…simplicity and speed.

## Scope of the project

In the project scope there are a few things we need to mention, these include determining and then documenting specific deliverables, features and functions of our application.

In short, we can describe our project scope as: To create a mobile application allowing for the storage and sharing of recipes. But this definition comes out a bit too vague, so below is a more detailed version of the scope.

The Recipe Book Hub must be able to do the following and only the following as anything else would be out of this projects scope:

1. Firstly, RBH needs to be able to store personal recipes, this will be done via inputting the recipe yourself via a form. This can be a tedious process but an additional feature could be importing recipes from a computer (though this is currently not in our scope).
2. Secondly, RBH must allow for the uploading of said personal recipes to a central hub which can be accessed by everyone with the application.
3. Lastly, RBH shall allow for users to download recipes from the central hub into their own personal recipe book.

## Objectives and questions

Before we discuss the specific objectives of this project, we must first get a good understanding of the aim of the project, what do we really want this project to accomplish?

### The Aim

To reduce the time, it takes to search for a specific recipe by developing a collaborative Recipe Book with an ambitious search engine.

Now that the aim of this project has been established we need to provide objectives to accomplish our aim. Below is a list of objectives that we believe we need to follow to accomplish our aim.

### Objectives:

1. Research and analyze any existing applications that can and do relate to our project idea. We investigate said existing applications and find areas for improvement.
2. Create and Design a model for this application.
3. Create a basic prototype and implement our ideas.
4. Analyze our prototype and document our findings.

### Research Question

Now the next step is to encapsulate all the above into research questions that need to be answered by our literature review, development and final documentation stages.

So, we need to ask ourselves:

Can we shorten the time it takes to read and navigate through traditional recipe books by offering a better alternative, a mobile application?

Can we make a more effective way to share recipes over the internet, via a central storage Hub?

# Chapter 2: Measurable Organizational value (MOV)

In IT project management, the overall goal of the project can be referred to as the Measurable Organization Value. For the MOV to be successful, it must line up with and support the IT project’s goals and objectives, therefore any alternations made by the project team must be made in response to the MOV. To have a successful MOV we must follow a few simple steps, these include:

## Identifying the applications desired area of impact

There are five potential areas of impact for a IT project (Marchewka), these include:

* Strategic
* Customer
* Financial
* Operational
* Social

We as the team have decided that we want our applications area(s) of impact to be not only customer based but also operational. Below we will explain the reasoning why we want to focus on these areas of impact, let’s begin with Customer.

### Customer

This application is mobile based, thus our main impact would be the users who use our application, so we need to greatly focus on ensuring that we develop an application that meets the customers’ expectations by developing a better alternative to the current applications out there.

### Operational

Since we are developing an application that has very similar alternatives, we must provide a great impact in the operational area. To make this MOV successful we need to develop an application that is more streamlined than the rest and more simplistic, because a simpler application is faster to navigate through which in the end, reduces time.

## Identify the application’s desired value of the IT Project

Now that we have decided on the area of impact that this application will bring, we now need to focus on determining our desired value that this IT project can bring to our application.

The four main values to consider are as follows:

Better – What do we want this application to do better than others?  
Faster – What do we want this application to do faster than others?  
Cheaper – What do we want to make cheaper in this application?  
Do More – What do we want this application to do more than others?

This section focuses on knowing how we will achieve our goal by deciding on one or more of these values. We have decided that the value most fitting our IT project’s aim and goal is ensure that our application is faster and does more. More exactly we want to reduce the time people spend searching for recipes, which will involve shorter cycle times throughout the application, basically it must take as few clicks and searches to find a desired recipe. Also in terms of do more we want to relate to growth; this value only effects the sharing side of the application in terms of the number of recipes available in the central hub.

## Identify and develop an appropriate metric

That we have identified our desired values which will support our application in succeeding, we now must identify and develop a metric or metrics which will give a sense of direction or target to achieve. Metrics in this case must provide an increase or decrease in the applications future state and must be expressed/described in terms of dollars, numbers or percentages.

We need to find a metric that is suitable in measuring the success of our application in relation to our desired goal, in this case time and speed. So, we will set our applications metric to be:

‘A user must be able to view their desired recipe within 15 seconds of opening the application’.

To achieve such a metric, we need to think of developing our application in terms of simplicity, all ages must be able to utilize this application so making a simple and easy to read application will greatly enhance the navigation speed around the application.

## MOV Statement

Now that we have determined the impact area, value and metric we will be using for this MOV, we will need to summarize it all into one tight and compact statement (Marchewka).

“In order for this IT Project to be deemed successful, a recipe must be able to be viewed within 15 seconds of opening the application.”

# Chapter 3: Literature review

## The limitations of traditional recipe storage and sharing methods:

In the article "Recipes are dead: What the future of cooking might look like” Maura Judkis got the opportunity to consider the future of cooking and recipe sharing from world renowned chef and television host Tyler Florence. Florence is well known for his recipe books such as "Tyler Florence Fresh" and "Dinner at My Place" but late last year he proclaimed that he will not be printing any more recipe books. His reasoning for this is because "I'll [Florence] publish a cookbook and I'll publish 125 recipes. People only use five," this shows how wasteful printing recipe books can be, not only are they limited to a couple of hundred recipes but people will rarely use more than a couple of recipes from a single book. He also stated, "They'll use those as like a guide that they'll kind of interchange different ingredient with," which is a very important thing to bring up – recipe books cannot be altered quite as easily as digital recipes. (Judkis, 2017)

## Current Solutions in the market

Below is a summary of Noelle Carters article titled "Organizing your recipes online? Of course, there's an app for that" where she discusses some of the current digital solutions available to users. (CARTER, 2014)

### A bit of Background

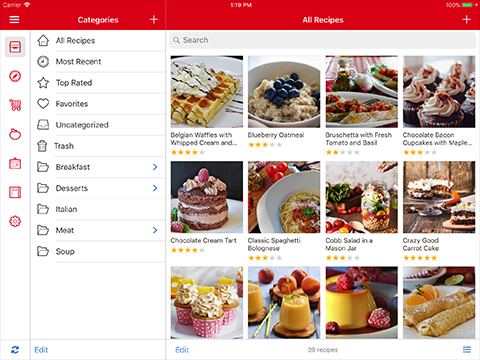
Before writing this article, Carter had been working with recipes professionally for a while, she is a chef, test kitchen director and a writer for L.A Times Food. She claims she was “hesitant to go completely digital” but however once trying some of the most popular recipe/cooking apps she proclaimed that the “results were amazing”. (CARTER, 2014)

Here is what she had to say about some of the most popular recipe apps:

### Paprika

For the price of $4.99 (these prices may no longer be accurate and are using USD currency) Paprika is one of the most popular apps in its field and is great whether you're new to storing recipes digitally or not. It offers meal planning, recipe browsing, collection, creation and management. The design is both easy to follow and intuitive. Paprika offers features that will make for a fun, simple and much more relaxing cooking experience, these features include the ability to track your progress – this is done by crossing off ingredients and highlighting the step you are up to, this feature is especially useful for those who are forgetful and may double up on their ingredients or miss a step. Paprika also allows for the scaling of ingredients, what this means is if you find a recipe that makes 12 cupcakes but you only want to make 7, the app can quickly calculate this and adjust the ingredients accordingly which is especially useful. The list of ingenious features doesn’t end there, Carter claims the app also offers shopping list functions, meal plans and calendar functions. (CARTER, 2014)

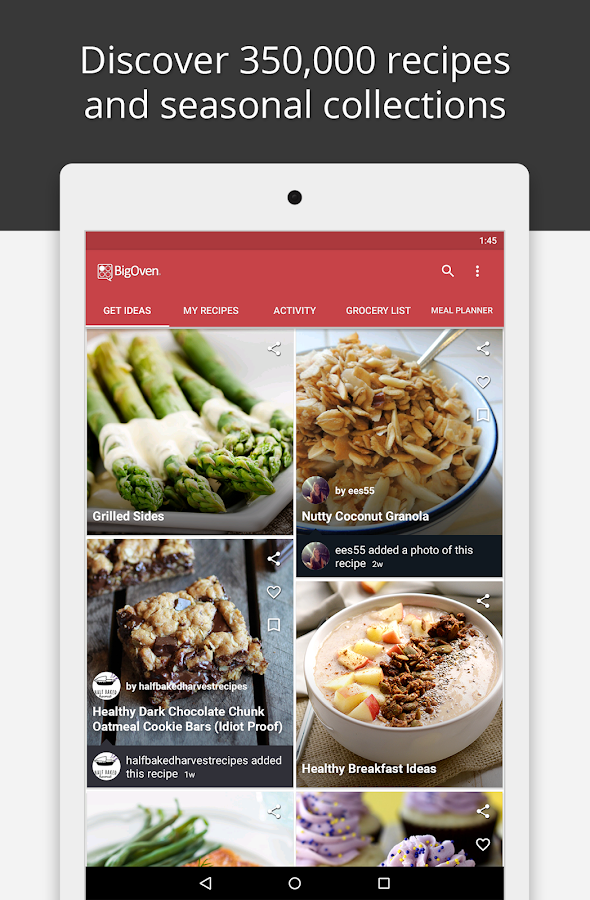
Below is a screenshot of Paprika’s Graphical User Interface (Paprika, 2017) showing the recipe categories menu, I can immediately see that this GUI contains too many features on one page, the menu is visible, the categories and the recipe list. After looking at this application, we can confidently say that our own application will have a more simple and structured layout, this will be our added value over this alternative.

  
Fig. 1. The Paprika App Category page (Paprika, 2017)

### Big Oven

Big oven is another popular app offering many similar features to Paprika, however some more advanced features are available with the pro membership (basic version is free and membership is $2.49 a month or $19.99 a year) it allows for you to take a photo of a recipe and it transcribes it through optical character recognition. The pro version also offers note adding, web clipping and nutrition information. (CARTER, 2014)

Now considering Carter’s opinion on Big Oven, we decided to do some digging ourselves, we have found that this applications GUI is also quite complex (BigOven.com, 2017). Sure, it is nice to have many features in an application, too many can lead to a complex GUI and overwhelming experience. Our idea with our app is to specifically target people who want a very simple application for recipe storage and sharing, taking this into account it means that our functionalities will only be limited to doing these tasks. This puts our application in an advantage in a sense, as we have limited functionalities we can really focus on the simplistic structure and design of our application. Our opinion involving Big Oven is that it contains too many functions that seem nice on paper, but hardly get used.

  
  
Fig. 2. The Big Oven App Recipe page (BigOven.com, 2017)

### Pinterest

Carter also discusses another alternative which is Pinterest. Pinterest is not built solely for storing and sharing recipes but you can use it to "pin" recipes, ingredients or just ideas to different "boards" which can be organized however you like and shared with whoever you want. Pinterest is completely free and is a much more visually appealing and flexible, for this reason alone many users choose to use this alternative even though it isn't designed specifically for recipe storage. (CARTER, 2014)

Although Pinterest is a very popular social sharing application, it does not specialise in recipes, so this app is kind of an outsider to the rest, although we must say Pinterest does an excellent job in implementing many features under one idea. Now looking at the design side of Pinterest in terms of recipe storage, it’s very complex and not recommended for people who want to be just storing recipes.

### MacGourmet + Gourmet Mobile

People often don’t like switching from books to digital, people don’t often like change and that’s why MacGourmet is worth a mention. Much like other recipe apps it can be used to create, organize, edit and share recipes. However, MacGourmet also allows for the creation of your own "cookbook", this is a digital book where you can create your own look including text and images. (CARTER, 2014)

The Problem with MacGourmet is that it’s an apple desktop application that costs $50 (Sofware, N/A), the desktop application is very professionally made and is very simple to navigate through. However, the mobile app that is used alongside MacGourmet is known as Gourmet Mobile. This application contains many functionalities that we want to include in our application, but using the mobile version to its full extent requires MacGourmet. This is due to the syncing functionalities being a core of this applications idea.

To our final view on this application is that it’s well designed in terms of simplicity but is costly to use at its full potential and is only limited to Mac and IOS users. Below are some screenshots of Gourmet Mobile, the design is simple but clustered in our opinion.



Fig. 3. The Gourmet Mobile App Landing Pages (Mariner Software, N/A)

### myTaste Recipes

The application that has come closest to our idea is myTaste Recipes app, this application offers a very straight forwards and simple user interface. The functionalities that this application offers are online searching, personal recipe storage and user management, although this comes close to what we want to achieve, we have found that this application focuses more on the online sharing side than the personal side, whereas our application idea is the opposite.

Below are some examples of the simple and straight forward layout of myTaste:

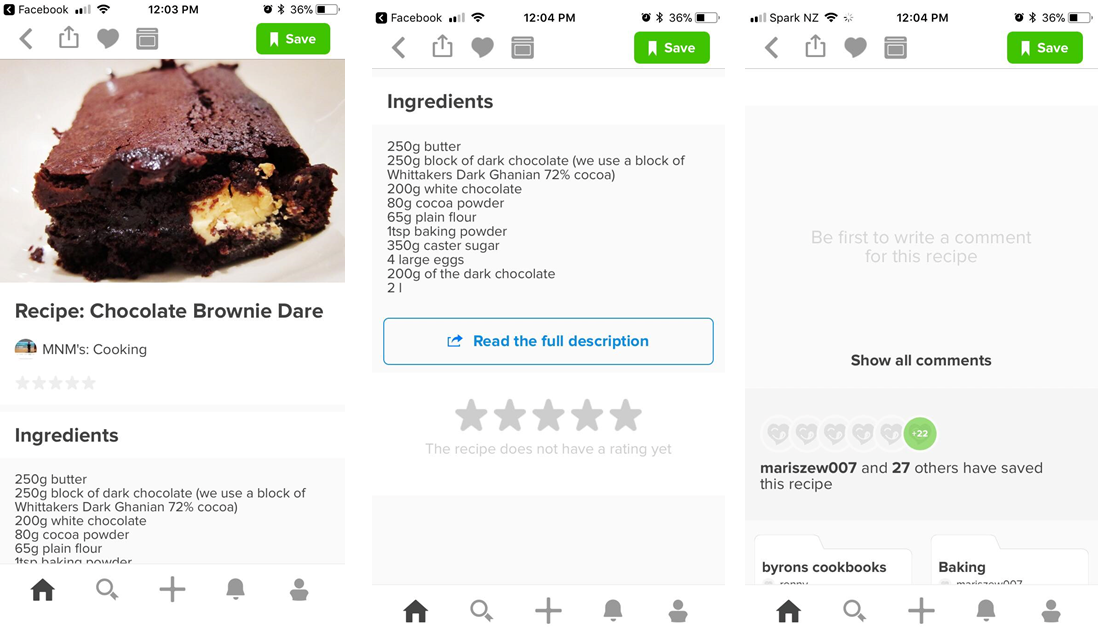


Fig. 4. MyTaste Recipes App Landing pages. (AB, 2017)

In the first image we can see the name of the recipe, a photo of what the result should look like. It is also obvious for the user to work out how to save and share the recipe using the buttons at the top.

In the second you can see the list of ingredients as well as a link to the full description as the recipe is summarised and for in depth instructions you must follow the link, below you can also see where you can rate the recipes.

And the final image shows a comment section and how many people have saved the recipe.

The conclusion for this application is that we believe it does follow the simplistic design approach but as we stated before the application is more focused on recipe sharing rather than your own personal recipe book.

## What makes a great mobile application?

Below is a summarized list of features that make the difference between a successful app and a fail, this list is taken from the article: "How to Create a Good App: 11 Must-Have Features of a Successful Mobile Application." (Mobidev, 2016)

1. The app should be simple to use for the end user
2. The app must cover both iOS and Android
3. The app should be well optimized and not leave the user waiting for tasks to be complete
4. The app should keep data confidential and should be secure and protected
5. The app allows for offline work and isn't limited to only being used with internet
6. Regular updates
7. The users must be able to leave feedback and/or be able to contact the developers
8. You must be able to personalize the app.

In the article: "Eight Tips for A Successful App" Larizadeh explains how you must "Incorporate some viral mechanism" so that "each user that enjoys the app can potentially attract a network of new users." Apps are far more engaging when users can invite their friends to join them on it, take Instagram for example, its only fun when you have friends who use it too. (Larizadeh, 2013)

What this means for our recipe application is that it needs to include some features where users can interact. This could mean the ability to "follow" users, the ability to review/comment on recipes and a share button so you can share your recipes to Facebook or twitter which will attract new users.

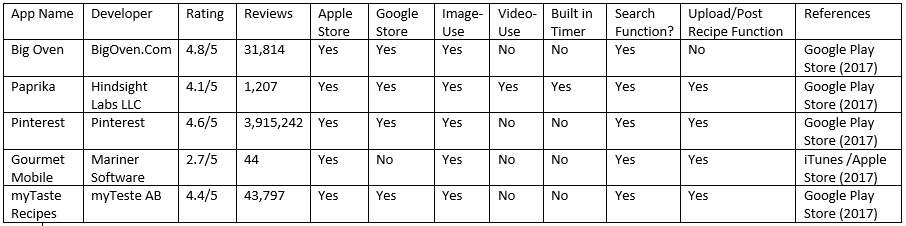
### Polished Design

Max discusses some features which can make for a more polished design in the article "What makes a great mobile app?"

In summary Max states how, a good apps user interface is simple and easy to use. Time must not be wasted on adding elements that don’t add value to the app, unnecessary features will make the app hard to "work out" for first time users.

Max states that "A great app is built on a unique idea, a good understanding of the high-end users and their behaviours, psychology and habits, and has quick response times." (Max, 2017)

## Comparison Table



## Added Value and Functionalities

Talk about pro cons of lit. review, added value/functionalities.

In this section we’ll be talking about our concluded opinions about all the existing applications listed above. We will outline what we think are the good and bad functionalities of each application and how we can make it better.

The first core feature we noticed from all the existing applications is that they all contain more features than our proposed application, for example; Meal Planners, Ratings, User accounts and comments. All these added features can create a positive impact on the application but, a lot of people and especially the older generation have no need for all these extra functionalities. This brings us to our first point of improvement; we believe we can add more value to our application by having less functionalities than the rest, this will free up a lot of space on the GUI which results in a cleaner, faster and easier to navigate through app which is exactly what we are aiming for.

Another feature that all these applications focus on is the sharing side of recipes, this is of course good and well, but our idea is to make a personal digital recipe book with the added functionality that you can share recipes if you want. The idea is to replace the traditional recipe books in your household, not replace your personal computer. So, taking this into account we move on to our second area of improvement; to give the application a more ‘personal’ atmosphere.

Now we must delve into the specific functionalities that these applications can do, firstly the way recipes are stored; most of these existing applications allow you to store recipes into set folders or categories, well we want to add a sense of personalisation to this and allow users to create their own folders/categories and call them whatever they want. This change in functionality will follow our theme of creating a recipe book that is your own.

Another functionality that lies at the core of any recipe application is the ‘create new recipe’ function,  
we want to make this page with functionalities that all the existing applications use, from task timers to photo capture for every step. We believe merging all these functions together will make for a simple yet detailed recipe view.

The existing applications out there contain many functionalities and features but following our current scope, the only functionalities and features we shall be adding to our application currently is; A simple but yet effective home screen that allows for easy navigation throughout the entire application, a side bar that acts as a menu bar for navigation whilst one is doing tasks within the application, a create a recipe function, a share recipe function that uploads the recipe to the online hub, an online Hub for sharing of recipes and lastly an effective recipe viewing screen.

# Chapter 4: Research approach / Methodology

When it comes to mobile development, choosing the correct methodology approach can be key to the success of the project. Taking this into mind we have decided to conduct some research on which SDLC Methodologies would suit our project idea the most (Lotz, 2013). The two most common types of methodologies that come to mind are:

###### Waterfall

This methodology is considered the most traditional SDLC methodology, which follows a very linear approach in terms of software development. The Waterfall methodology consists of distinctive steps that normally must be completed before the next step can begin, these steps consist of:

* Requirements Gathering
* Design
* Code and Development
* Testing
* And Maintenance

Following this methodology, requirements must be correctly set at the start of the project as the scope is known from the beginning. In this methodology there is little room for change as this is highly difficult and costly since it goes against the methodology (Nutshell, N/A).

The Waterfall Methodology is highly recommended for projects where requirements can easily be established, and are typically used for often for larger projects. Because we are relatively new to mobile development we believe that this methodology will not suit our project idea because we predict that we will have rapid change within our development stage.

###### Agile

Now on the other hand the Agile methodology is in a way the opposite to the Waterfall approach, this is because the Agile methodology is all about developing software incrementally using small iterations. All the steps involved in Waterfall are applied in the agile methodology except in agile they all done continuously, this is because Agile methodologies focus on the rapid development of functional components. That all steps are done together, it means that Agile methodologies can handle change very easily because tests are very regularly that faults and errors are found early and can be changed.

Debating between these Methodologies depends on the type of application that is being developed and the time frame you have. In our project we have deemed the Agile Methodology as our preferred approach because we are relatively new to the mobile development scene so mistakes will be made so we need to follow a methodology that allows for change and mistakes (Nutshell, N/A).

## Languages and Programs to be used

Due to our projects time constraints we have decided to only develop an application for the android market. Knowing this, we started researching what software would suit our needs the most and to no surprise Android Studio was the obvious choice.

###### Android Studio

Android studio is chosen for many reasons, the first being obvious… its Android Studio. This software is built for making android applications which is exactly what we need. Our team being new to mobile development will greatly benefit from android studio’s large amount of documentation (Android, N/A).

Android studio also has many features such as Cloud Integration, APK Analyzer, Testing Tools and Frameworks, code templates and a virtual android simulator. All these features and functionalities will aid our project team in succeeding with our application (Studio, N/A).

Our next question after decided the platform to develop on is of course to decide which language(s) to code in. The obvious being Java since it’s the most supported language to develop android applications. Other languages such as C++ and XML shall also be used for finer functionalities like databases for example.

This now brings us to our next point which is concerned with data storage, we have decided to use SQLite as our SQL database engine. Our reasoning behind using this database engine is simple, SQLite is the most widely known and deployed SQL database engine for mobile development in the world (TutorialsPoint, N/A). Another reason that greatly impacted our decision for using this software was the amount of documentation we found, this information is highly valuable to us because of our limited knowledge.

###### Gantt Chart

Refer to Appendix A to view the Gantt Chart.

We have created a Gantt chart that will best follow our idea of developing the application, we have added overlapping activities since we are using the agile methodology which states that we do SDLC steps simultaneously with each other.

###### Work Breakdown Structure (WBS)

Refer to Appendix B to view the Work Breakdown Structure Chart.

# Chapter 5: Design and Analysis

In this section we shall be outlining the various steps that be followed to achieve our prototype. This will include Requirements, Potential Risk Management and Design and Analysis

## 5.1. Potential Risk Management

Before we begin with the Design and Analysis sections, we must first evaluate the potential risks that come with software engineering and how-to management them.

Firstly, let’s begin with Requirement Risks, we have identified two risks that fall under this which are:

* Change of Requirements
  + This refers to changing requirements that are set at the start of the development stage, because readjusting requirements can cost a lot of time which we don’t have a lot of.
  + To manage this risk, we have decided to only change requirements if we have already found and can implement another option, or else we will push the requirement to becoming future work.
* Impossibility of Requirements
  + This refers to requirements being out of our knowledge scope, we did well to research resources before setting the requirements to eliminate this risk as much as possible, because following a requirement which will pose impossible for the time and skill we have becomes a huge risk.

The next set of risks refer to Quality Risks which are vital to management to prevent poor quality applications, these are:

* Human Errors
  + These refer to errors made by us while programming which can be mistypes, incorrect use of code etc.
  + Eliminating and managing this risk can be difficult because its referring to humans which are cannot be measured to easily. However, in our case to manage this risk we will be making sure we check our code over and over to minimize risks.
* Inadequate Knowledge about Programming such as languages (Java, XML) and the tools used (Databases, Android Studio)
  + This refers slightly to the previous risk but is more defined. To minimize lack of knowledge all we can do is accumulate more by researching or avoid task that require too much of it.

## 5.2. Requirements

We first begin this process by dotting down the exact Requirements that we want for this application, this is doing by the Requirements Gathering process.

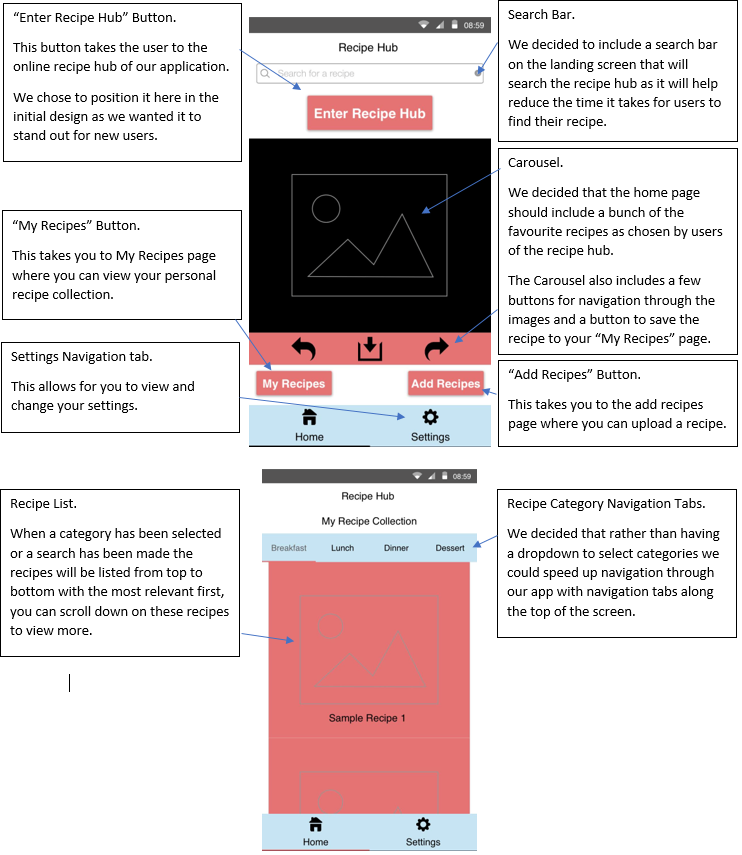
In terms of user interaction, we users to be able to:

* Add a Recipe via a form in a new page.
  + Add Recipe Title
  + Add Serving Size
  + Set Category Type
  + Set Preparation Time
  + Set Cooking Time
  + Add Ingredients
  + Add Instructions
  + Add Picture(s) to recipe
* View Recipes in List form via navigating to a new page.
  + Display only the Recipe Title and Serving Size
* Search for Specific Recipes via a search functionality
  + Quick and fast searching function
* Category filtering
  + Tabs? Folders?
* View Recipes in List form within the online Recipe Hub
  + Display only Recipe Title and Serving Size
* View Individual Recipes
  + Display all Recipe elements/fields
* Upload Recipes to Online Hub
  + Button within Individual Recipe
* Store Recipes Locally on Device
  + SQLite? Room Persistence Library?
* Store Recipes Online
  + Firebase?

These are the current Requirements that we must achieve, which are subject to change due to our Methodology approach being Agile. Functions and GUI’s can be changed during the coding to better suit the situation and since we are learning Android Development, we shall be using components that we can deliver.

## 5.3. Design and Analysis – GUI Initial

The Following Wireframes/ GUI where created with Proto.io



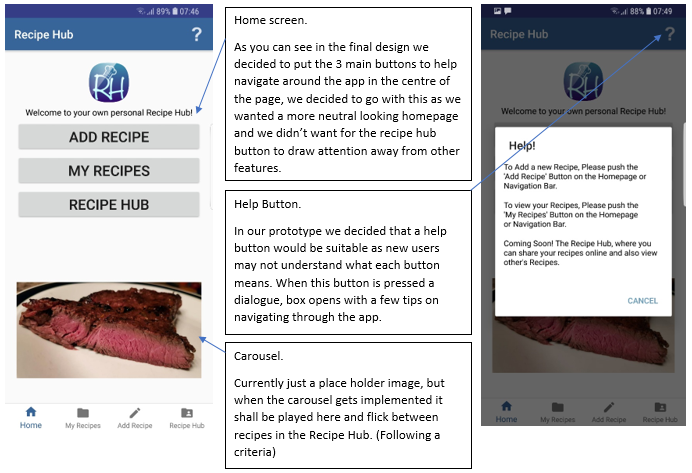


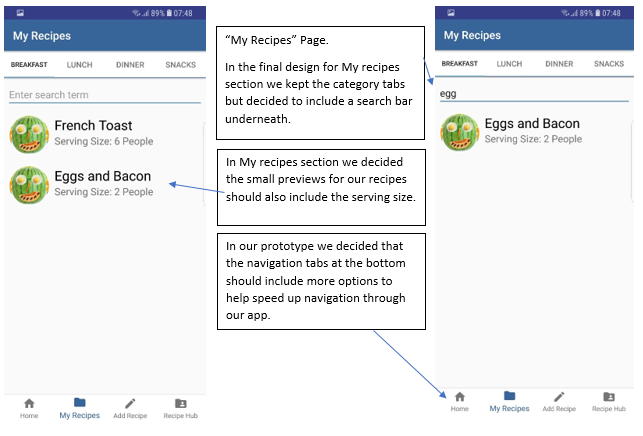
## 5.4. Design and Analysis – GUI Current

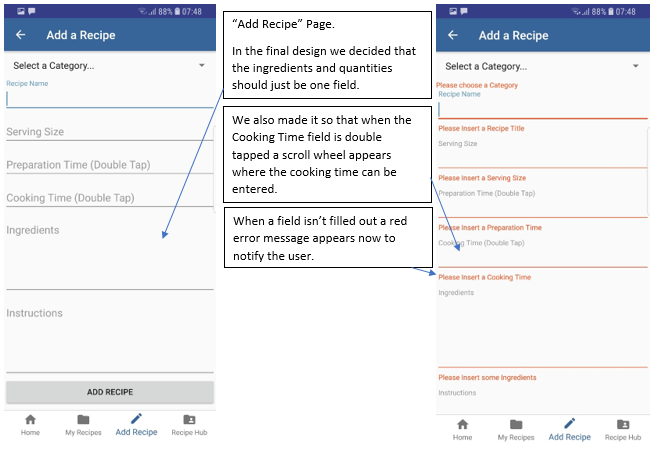
Within the initial Wireframes we started designing an application taking little into mind about how to code the components and features that we wanted. When we started coding, and researching about how to add different components to an android application, we quickly realized that we had designed a GUI that we would not be able to follow completely.

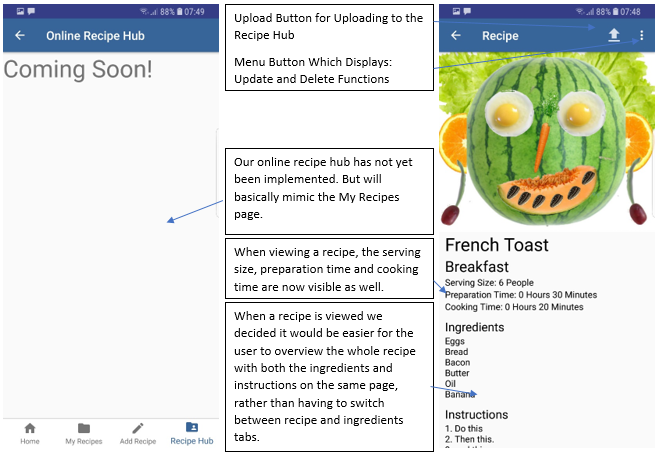
We also believe that the initial wireframe seemed a little overcomplicated for what we were really trying to produce, so during the coding phase we changed our design to fit our theme of speed and simplicity better.

The following GUI was designed and coded taking into mind our initial wireframes/GUI and our own programming skills.









We believe that this change of design has been successful due to delivering a cleaner look to the application which results in easier visibility and navigation throughout the application. We need to constantly keep in mind our Aims and Objectives about Simplicity and Speed throughout the application, did some testing and offered students and family an image of the initial and Current GUI and every single person we showed deemed the Current to be better than the Initial in every possible way.

# Chapter 6: Implementation and Testing

Now beginning the implementation phase, I first went to the Android Developers Website, <https://developer.android.com/index.html>. Reading all the basic Android Guides about the fundamentals I then proceeded to Android Studio to start with my coding.

Throughout the entire implementation phase I was referring to Stack Overflow, https://stackoverflow.com/ and YouTube for advice and tutorials to self-teach myself Java, XML and learn more about Android Components.

## 6.1. Coding and Errors

We will now be speaking about the different implementations within our application, the way we performed them and any issues we ran into. We will not be mentioning errors that are considered small and are fixed with no effort at all.

### 6.1.1. HomePage

Firstly, we became with implementing a generic homepage for the application which contains three buttons that will navigate the user to the three other main activities within the application. These buttons will navigate to the My Recipes, Add Recipe and Recipe Hub activities. The Homepage activity also contains a static image that serves as a placeholder for the carousel that we wish to implement in the future.

For users that are not technology friendly, we have placed a clear ‘?’ button within the action bar located at the top of the app which displays a dialog action when pressed. This dialog is created by creating an override method for onCreateOptionsMenu which inflates the menu\_home.xml file that contains the ‘?’ item. The next override method is the onOptionsItemSelect which recognizes the ‘?’ item and calls the openDialog method, this method then initializes the help\_dialog.xml file to open and display the information within the dialog.

During the implementation of the homepage we faced no errors throughout the application.

### 6.1.2. Room Persistence Library - Database

We decided to use the Room Persistence Library for our offline database since it is powered by SQLite and allows for quick and fluent database access throughout the application. We had to add a dependencies and implementations for this component to our gradle file.

Room Persistence Library consists of three main components; the Database, the Data Access Object (Dao) and the Entity.

Database: This is the data holder component and is to be created as an abstract class which must extend RoomDatabase. During runtime, an instance of the database can be created by calling the Room.databaseBuilder() method.



DAO: This component is the main and most important component of the Room Persistence Library and is made as an interface which handles the defining of methods that access the database, all the database queries are created here.

Entity: This component acts as a row within the database table, this class is basically the Recipe model which contains all the required fields for the recipe object.

I did run into the biggest error during my coding and implantation phases here and was query errors, this error was purely my fault and we will go more in depth in the RecyclerView section.

### 6.1.3. Add Recipe

For the implementation of the Add Recipe activity we had to add a few components to the class, these included; the ‘Spinner’ which acts like a dropdown menu for our Categories and for selecting the Preparation and Cooking Time we decided to implement a Timepicker which is responsible for choosing the times. After all variables are initialized and linked to the layout file we created a addRecipe method which first contains a validation checker, basically it’s just ‘if’ statements that check whether the form fields contain any text input, if they don’t it will display a message below that field telling the user to input data.

After the validation is completed we create a new recipe object and parse all the user input into it, I add:

Log.*d*(*TAG*, "This: " + recipeNameEditText.getText().toString());

This line of code will produce a log that displays the Recipe Name after the add recipe button is pressed, if this log displayed a null value I would know that we had an error, if it was the recipe name then it succeeded.

We then add the recipe object to our database by call the addRecipe SQL method located in our DAO.

MainActivity.*myDatabase*.dao().addRecipe(recipe);

### 6.1.3. My Recipes and Recipe Item

This section of the implementation was the most challenging as I decided to use a tabbed layout to hold four fragment layouts that each contain a RecyclerView of each category.

The My Recipes activity only contains the implementation of the SectionsPageAdapter and the initializing of the ViewPager for setting up and displaying the fragments.

Each fragment activity has basically the same code except for the category references which are Breakfast, Lunch, Dinner and Snacks. The fragment starts by inflating the layout file used for this fragment which contains two fields/components; the search EditText and the RecyclerView.

Initializing the RecyclerView to display all our recipes within our database (sorted by category), we added code which creates the RecyclerView itself and the RecyclerView adapter which is responsible for adding the RecyclerView items to the RecyclerView.

We couldn’t add a generic search filter for our RecyclerView because we had four recyclerViews contain within four fragments in the same activity so we had to get creative. We use the search EditText to handle the user input of the search, the actual searching function is done between the fragment class and the adapter class.

Now we can talk about the error I mentioned earlier, for a user to open a selected recipe by touching an item within the RecyclerView we had to create an OnClickListener within the adapter class that creates an intent to parse the currently selected recipes data to the Recipe Item activity (this activity displays the recipe to the user). The error/problem I faced was that data I was sending to the Recipe Item activity was all null except the Id, Recipe Name and Serving Size. I spent countless hours tackling the error from numerous angles which included re-writing it several different ways. The error/problem in the end was my own human error, the SQL method that was receiving each categories recipe list was only receiving the Id, Recipe Name and Serving Size.

Problem:

@Query("select id, recipeName, servingSize from Recipes where category='Breakfast'")  
List<Recipe> getRecipesBreakfast();

Solution:

@Query("select id, recipeName, servingSize, category, prepTime, cookTime, ingredients, instructions from Recipes where category='Breakfast'")  
List<Recipe> getRecipesBreakfast();

The Recipe Item activity contains the method to retrieve the Intent from the adapter and then sets the values to the TextViews within its Layout file.

### Navigation Bar – Bottom of Application

We decided to implement a navigation bar at the bottom of the application that appears on all the activities except the Recipe Item activity. This is done by creating a Navigation Helper class which works alongside the activity to display the navigation.

This implementation was straightforward and we ran into no errors during this section.

### Other

Most of my debugging was done by setting Log.d functions to methods so that It displays a log every time that method is called, doing this I could pin-point many errors and resolve them very quickly.

The Online Recipe Hub activity is viewable but only displays some text “Coming Soon!”. Implementing this by the looks of it will almost mimic the offline version however, an online database is just required.

The upload to Recipe Hub function will be parsing the current viewed recipe’s object and adding it to the Online database, basically an add button.

We had thought of implementing a function that allows a user to choose an image(s) from their devices gallery or to open the camera and take a picture live. However, implementing this function proved very difficult with our lack of knowledge so we decided to put it on the list for future work.

My explanations may not be on point exactly so please refer to the source code for a better understanding.

# Chapter 7: Future WORK AND Conclusion

All the functionalities that where not achieved are all being pushed to this section for future work. These include:

* Adding the Online Recipe Hub component for sharing and storing recipes online.
* The camera and gallery function to allow users to select images from their devices gallery or take a live photo with the camera function.
* To implement the Update and Delete functions for the recipes to finish off the CRUD component.
* The ability to share to social media (Facebook).
* A login system that allows users to create profiles.
* A rating system so that users can rate other recipes.
* And of course, the ability to be available on IOS devices.

To conclude this paper, we believe that we faced several limitations that restricted us from achieving all our functionalities. These include:

* Time constraint
* Lack of Language Knowledge (Java and XML)

Having more of both would have allowed us to achieve most if not all our functionalities but, for the time and knowledge we had we believe that we successfully achieved our goal of creating a prototype for storing recipes via mobile application. The amount of knowledge we gained from developing this prototype will help us in the future with android mobile development.

# References

AB, m. (2017, December 15). *myTaste Recipes*. Retrieved from Google Play: https://play.google.com/store/apps/details?id=com.mytaste.mytaste&hl=en

Ahmad Ammari, S. D. (2013, September 18). *ITEC N466: Project initialization*. Retrieved from Slideshare: https://www.slideshare.net/AhmadAmmari/itec-n466-project-initialization

Android, D. (N/A, N/A N/A). *Build Your First App*. Retrieved from Developer Android: https://developer.android.com/training/basics/firstapp/index.html

BigOven.com. (2017, November 23). *BigOven: 350,000+ Recipes*. Retrieved from Google Play: https://play.google.com/store/apps/details?id=com.bigoven.android&hl=en

CARTER, N. (2014, March 21). *Organizing your recipes online? Of course, there's an app for that*. Retrieved from Latimes: http://www.latimes.com/food/la-fo-recipe-organization-20140322-story.html

Eduonix. (2017, May 26). *Learn about the different Methodologies for Mobile App Development*. Retrieved from Eduonix: https://blog.eduonix.com/mobile-programming/learn-different-methodologies-mobile-app-development/

Judkis, M. (2017, December 6). *‘Recipes are dead’: What the future of cooking might look like*. Retrieved from The Washington Post: https://www.washingtonpost.com/lifestyle/food/recipes-are-dead-what-the-future-of-cooking-might-look-like/2017/12/06/75a1cdec-d399-11e7-95bf-df7c19270879\_story.html?utm\_term=.bfec62ed6453

Larizadeh, A. (2013, July 19). *Eight Tips For A Successful App*. Retrieved from Forbes: https://www.forbes.com/forbes/welcome/?toURL=https://www.forbes.com/sites/avidlarizadeh/2013/07/19/eight-tips-for-a-successful-app/&refURL=&referrer=#31f0408253e5

Lotz, M. (2013, July 5). *Waterfall vs. Agile: Which is the Right Development Methodology for Your Project?* Retrieved from Segue Technologies: https://www.seguetech.com/waterfall-vs-agile-methodology/

Marchewka, J. T. (n.d.). Retrieved from INFORMATION TECHNOLOGY: http://cmapspublic2.ihmc.us/rid=1P5RLM38Z-JK5PTH-28Q3/Marchewka%20-%202003%20-%20Information%20technology%20project%20management%20Providi.pdf

Mariner Software, I. (N/A, N/A N/A). *Gourmet Mobile*. Retrieved from Itunes: https://itunes.apple.com/us/app/gourmet-for-ios/id711561842?mt=8

Max. (2017, 05 09). *What makes a great mobile app?* Retrieved from Weebpal: https://www.weebpal.com/blog/what-makes-great-mobile-app

Mobidev. (2016, December 13). *What Is The Value Brought By Quality Assurance To Your Software Product?* Retrieved from Mobidev: https://mobidev.biz/blog/what\_is\_the\_value\_brought\_by\_qa\_to\_your\_software\_product

Nutshell, A. i. (N/A, N/A N/A). *Agile vs Waterfall*. Retrieved from Agile in a Nutshell: http://www.agilenutshell.com/agile\_vs\_waterfall

Paprika. (2017, November 14). *Paprika 3.0 has been released for IOS and macOS*. Retrieved from Paprika: https://www.paprikaapp.com/blog/2017/11/14/paprika-3-0-has-been-released-for-ios-and-macos/

Sami, M. (2012, mARCH 15). *Software Development Life Cycle Models and Methodologies*. Retrieved from Mohamed Sami: https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/

Sofware, M. (N/A, N/A N/A). *eStore*. Retrieved from Mariner Software: http://marinersoftware.com/store/

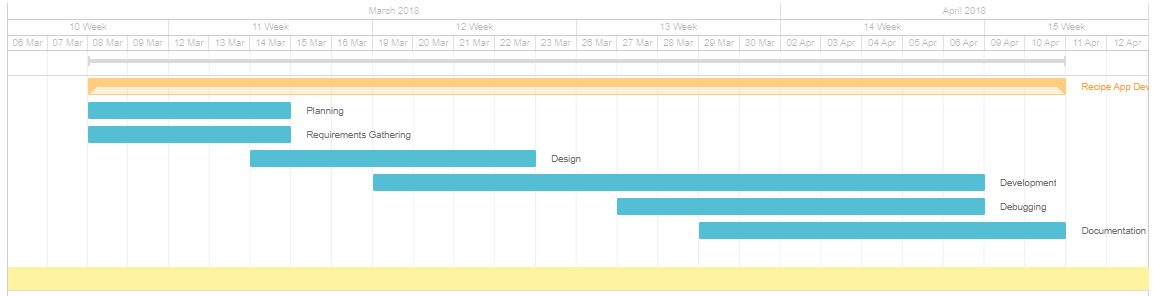
Steen, J. (2016, September 8). *12 Food And Cooking Apps Absolutely Everyone Should Have*. Retrieved from Huffpost: https://www.huffingtonpost.com.au/2016/09/06/12-food-and-cooking-apps-absolutely-everyone-should-have\_a\_21466734/

Studio, A. (N/A, N/A N/A). *Everything you need to build on Android*. Retrieved from Android Studio: https://developer.android.com/studio/features.html

TutorialsPoint. (N/A, N/A N/A). *SQLite Tutorial*. Retrieved from TutorialsPoint: https://www.tutorialspoint.com/sqlite/index.htm

# Appendix

## Appendix A – Gantt Chart

Below is the Gantt chart for the Application Development Stage.

## Appendix B – Work Breakdown Structure (WBS)

