

Final Audit Report

Recipick

Group 8

Date 12/05/2019

Version 1.0

Authors Mohammad Khan
Nithesh Koneswaran
Max Krawiec
Clyde Leal
Sze Lee
Pavlos Lekkas
Oscar Levy

Document Revision History

Date	Revision	Description
12/05/2019	1.0	Initial version

Table of Contents

Table of Contents	2
Introduction	4
Definition	5
Choice of Project	5
Project Constraints	6
Offline Functionality	6
Sensors	6
Server-side Data Gathering Component	6
Benefit to Society	7
Project Extension	8
Planning	9
Team and Role Assignment	9
Choice of Methodology	9
Agile Macro-Management	10
Choice of project management tools	10
Project Planning and Monitoring	11
File Sharing	11
Initial Plan of Deliverables and Milestones	11
Design	12
Requirements	12
Changes to Initial Vision	13
Additional Requirements	13
Applied Constraints	13
Offline Functionality	13
Sensors	13
Server-side data gathering component	14
Analysis of Privacy, Ethical and Social Aspects	14
Data Protection	14
Ethical and Social Aspects	14
Use Case Diagram	15
User Interface	15
Logo	17
Class Diagrams	18
Implementation	19
Choices of Tools and Technologies	19
Android Studio	19
Strategy and methods	21
Code Documentation	22

Evaluation	23
Peer-to-Peer Reviews	23
Test Plan and Unit Testing	24
Requirements Testing	24
Higher Level Requirements	24
Lower Level Requirements (As Agile User Stories)	24
Requirements Test Plan	25
Testing Report	35
Unit Testing	45
Privacy and Ethical Evaluation	46
Respect of User Privacy (GDPR Criteria)	46
Respect of Users and Ethics	46
Evaluation of Team Dynamics	47
Peer-to-Peer Assessment	47
Respect of timeline and milestones	48
Respect of Budget	49
Beta Testing Reports	49
Reflection	50
Aspects for Improvement	50
Beta Testing	50
Continuous Integration	50
Code Testing	50
Code Documentation	50
Positive Aspects of the project	51
Timeliness and Scheduling	51
Team Dynamics	51
Choices of Technology and Coding Tools	51
Choices of Methodology and Non-Coding Tools	51
Testing Procedure	51
Respect of Privacy, Ethical and Social aspects	52
References	53
Appendices	54
Appendix A: Beta Feedback Form	54
Appendix B: Example Minutes	55
Appendix C: Peer-to-Peer Assessment Form	57
Appendix D: Documentation	58
Appendix E: Database	60
Appendix F: Trello	62
Appendix G: Idea Template	63
Appendix H: Change Request	64

1. Introduction

This Final Audit Report (FAR) is the complete narrative - from its inception to completion - of the Android mobile application 'Recipick' developed by Group 8 for the University of Surrey's 2nd year Software Engineering Project (COM2027) module. It documents the whole process through which the application was imagined, refined, designed, and created. This FAR contains exhaustive information regarding all stages of the application's development.

The 'Definition' section delineates the process of choosing and refining the scope of the project. It summarises the different initial project ideas suggested by the team, and justifies how Recipick was picked among these ideas based on its adherence to the imposed constraints. This section also describes how the project is a benefit to society in its final form and initiates the exploration of the privacy, ethical and social aspects for the project.

The 'Planning' section describes the stages and decisions made for project planning in terms of management and methods. This section recounts how members of the team were assigned area-specific roles of focus and how the team's working methodology was researched and consequently adopted. Additionally, this section specifies and justifies the team's choice of tools for communication, planning and file sharing which were used throughout the development of Recipick.

The 'Design' section outlines the steps taken by the team to bridge between the definition and implementation of Recipick. It illustrates how the higher level requirements of the project were turned into Agile user stories and how Recipick's design was purposefully engineered to allow for extensibility. Additionally, this section describes the stages of the application's UI and UX design alongside its privacy, ethical and social components.

The 'Implementation' section rationalises the choices in technologies made by the engineering team to create Recipick. This section compares several alternatives to the technologies the team ultimately used and rationalises why specific products and tools were chosen to build the application. It also examines the project's privacy and ethical requirements along with its development methods and pipelines.

The 'Evaluation' section builds upon the Evaluation Criteria section of the Project Definition Document¹ and evaluates the finished project against this criteria with its privacy, ethical and social components. This section is heavily based on testing as it is the most important criteria in terms of engineering for the team. This section also extends this written criteria by exploring other aspects such as peer-to-peer reviewing and conducting beta tests.

The 'Reflection' section evaluates what the team has achieved throughout the course of this project. It assesses the design choices made and in what ways each choice caused an impact on the project. This section also explores the aspects of the project that were positive, juxtaposed to aspects that could have been done better in light of the future and continued development of Recipick.

¹See PDD Section 3.3

2. Definition

2.1. Choice of Project

During initial group meetings, brainstorming sessions took place where members of the team suggested possible projects. The team were then instructed to prepare presentations of at least one project idea for the next meeting. An idea template, i.e. a form with all the project constraints as fields, was created to be filled up before the presentations. This form was created to ensure that each project idea would meet these constraints.

In the following meeting, each member presented their project ideas. Each presentation had a brief description of the idea and justification as to how it meets the constraints. Any social or ethical aspects that needed to be considered were also made known. When the presentations were complete, a vote took place to choose the project for the proposal. Considerations such as project feasibility, if it comfortably met the constraints and if they were either a benefit to society or a business opportunity were taken into account.

As a result of the vote and further analysis of alternative ideas recorded in the table below, Recipick was chosen as it was a much more extensible project and beneficial to society. Some of the alternative project topics included in the idea template, but not included in any version of the proposal, can be found in Appendix G. From the research we carried out, it was evident that a project in the food and health industry has much more potential for growth as well as could cause a positive impact on the health and eating habits of people.

As our team consists of students that have experienced poor diets caused indirectly by university life, we unanimously agreed that one of the largest problems of a student's lifestyle is their eating habits. The majority prefer ready-made meals due to the convenience. Recipick aims to poor nutrition as a benefit to society with the potential of becoming a business opportunity upon achieving its initial goals as elaborated in sections 2.3 and 2.4.

In addition, Recipick was chosen as it was technically challenging. Members expressed a strong desire to work on an interesting project and to hone their skills in software engineering, consequently agreeing on Recipick as the project required learning new technologies to execute well. Recipick also seemed to be the ideal project to be built while utilising the Agile methodology which emphasises user experience as it is a largely user-centric application that aims to help users and better their lives. Members expressed a specific desire to practise this methodology so that they are better prepared for the industry. The project, being a benefit to society, also aimed to solve an issue that we felt personally passionate about, with its importance detailed in section 2.3.

2.2. Project Constraints

General project constraints have also been analysed in section 2.7 of the PDD.

The following subsections outline how the project would fulfill the constraints when the team initially defined it.

2.2.1. Offline Functionality

Users could save (favourite) recipes for future and offline viewing, creating a subset of functionality that operates with an intermittent Internet connection. This is the case when viewing favourited public recipes that have shared by other users and private recipes created by the user that have not been shared to the public.

2.2.2. Sensors

- Camera

The camera could be used for sharing pictures of completed recipes. Users can upload pictures to the recipes that they upload (private or public). Users may also share pictures of their attempts at cooking an existing recipe. This feature is undoubtedly beneficial for Recipick's future prospects of becoming a social media platform as explored in section 2.4.

- Location gathering component

Recipick could use the location sensor to locate the nearest supermarkets, presenting this information to the user with a marked map pinpointing the locations of these supermarkets. This allows users to identify where ingredients are available to purchase if they are missing ingredients for a recipe.

2.2.3. Server-side Data Gathering Component

- Ingredients that users input to Recipick can be processed server-side in the required data-gathering module.
- Location-based computing could be rendered server-side to maximize efficiency and speed of computation.
- Photos that have been uploaded by users to Recipick could be processed and organized server-side in order to minimise data loss.

2.3. Benefit to Society

The eating habits of students and people who live alone are mostly based on convenience and time-effectiveness at the expense of little to no consideration for their health. People who live alone are also at risk of poor dietary habits due to a lack of motivation and cooking expertise (1). In addition to health problems such as obesity and heart disease (2), poor nutrition can lead to stress, and high levels of stress have been found to cause a negative impact on performance in workers (3).

According to the research article, '*Dietary patterns of university students in the UK: a cross-sectional study*' (4), the 'convenience, red meat and alcohol' dietary pattern which had been 'identified most consistently across universities' is not only associated with a higher weekly spending on food, but also leads to negative 'long-term health effects'. Recipick simplifies the cooking process by recommending recipes based on a user's currently owned ingredients and preferences. As 81.4% of British students have 'unfavourable' diets (5), our goal is to improve the world's health and wellbeing by making it easier to home cook meals.

Among the main benefits of home cooking meals are:

- Health:
 - More energy throughout the day - healthier and more nutritious meals (compared to ready-made or takeaway food) which contain natural sugars that provide users with the energy needed for the day (6).
 - Maintain a healthy weight - having a balanced diet allows a healthy weight to be maintained and prevents type 2 diabetes (7). To further promote the health aspect of the application, each recipe will include a description, informing users of its nutritional values and calories to aid in diet planning.
 - Obtain the nutrients needed for growth and repair: home cooked meals made with fresh ingredients are high in nutrients, leading to an increase in strength and prevention of diet-related illnesses such as obesity and some cancers (8).
- Cost:
 - Cooking at home is more cost-efficient than eating out. Having specific recipes also means knowing the exact amount of ingredients that have to be bought, reducing the potential of buying too much food and wasting money.
 - Most recipes uploaded and curated use cheap ingredients.

2.4. Project Extension

Recipick has been created with great flexibility, extensibility and business potential in mind after initially achieving its initial goals of being a benefit to society. According to a study, 'Millennials still prefer cheaper food, and want it to be convenient. But they are also more willing to pay for fresh and healthy food, and are willing to go to great lengths to find it' (9), lending evidence to the project's endless potential.

Listed below are some of the potential ways to extend the project we foresaw during definition:

1. Social media extension: sharing pictures of food is a trend on social media. As Recipick is already community-based, it could easily grow into a social media platform where users share their passion for food, whether it be cooking, creating recipes, or casual browsing for recipes. Potential features range from being able to comment, rate and review existing recipes, subscribe to favourite users, and the implementation of more traditional features of social media such as 'friending' and messaging systems.
2. Recipe packs: users could purchase exclusive recipe packs from, for instance, famous chefs. Users could cook meals using these recipes for a small fee or monthly subscription, or purchase packs of a specific theme, cuisine or ingredient. This could be further extended by collaborating with wider known recipe providers or websites - these providers may sell their recipes on Recipick to users for some fee.
3. Advertising: the application could host advertisements for more revenue. For example, advertisements related to cooking could be placed in the recipes tab in between the listed recipes. These advertisements could range from kitchen hardware to supermarkets.
4. Promoted supermarkets: supermarkets could pay a monthly fee to be featured first on our map. For instance, when a user is missing an ingredient and proceeds to locate the nearest supermarket, promoted supermarkets would have different marker colors on the map, appear first, and/or be able to promote a specific product or a deal that they have which are relevant to the user.

3. Planning

3.1. Team and Role Assignment

- As a team, it was important to define and distribute precise roles so that members had a clear focus and can move forward in a synchronized and systematic manner.
- Although members were given specific roles to focus on, they were not limited to their roles and can take on and be assigned any task. The purpose of the roles were to give members a specific area to improve on and potentially lead tasks for that specific area. These roles were assigned after discussing the strengths and weaknesses of each member during a group meeting:

Name	Role
Mohammad Khan	Quality assurance, testing and DevOps
Nithesh Koneswaran	Front-end design
Max Krawiec	Back-end design
Clyde Leal	Front-end functionality
Sze Lee	Documentation and planning
Pavlos Lekkas	Back-end functionality
Oscar Levy	Project manager

3.2. Choice of Methodology

- At first, the team subconsciously employed a **waterfall** approach as we were most familiar with it due to previous modules.
- After week one, the group switched to the **agile scrum** methodology after further research. Agile was deemed a more effective approach for the following reasons:
 - Planning the whole project with concrete deadlines from the start seemed overwhelming, having to balance the workload of this module with other modules. We needed a methodology that was more flexible.
 - Agile facilitates independent working which is crucial for a university group project where it is not possible for everyone to meet everyday.
 - Allowed us to develop an application that is geared towards user-centric design. This gave members a personal motivation, rather than it being solely a project for university. It felt more like developing a benefit to society.
 - User stories allow the team to keep track of what needs to be done, and Trello is the perfect platform to utilise this to plan scrum sprints (10).
 - The methodology is flexible and permits continued improvement of the application. We aimed to make Recipick as extensible as possible.
 - Agile is becoming an increasingly popular methodology in the technology industry. The decision to adopt it and learn it allowed personal improvement for all team members to be better prepared for the workforce.
 - Sprints have been unanimously agreed to be the best method of keeping track of the tasks assigned to each member and ensures that at least something is achieved every sprint.

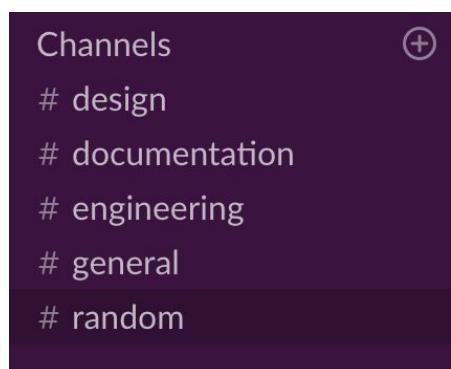
3.3. Agile Macro-Management

- Meetings were held at the end of each sprint in accordance to the Terms of Engagement as defined in the Project Definition Document² of this project.
- Agendas were prepared beforehand to ensure that time is maximised and used effectively during each meeting, crucial due to the busy schedules of team members.
- Reviews of the last sprint took place every meeting to monitor progress.
- The next sprint was then systematically planned with task delegation.
- Minutes³ of every meeting were taken in order to hold stakeholders accountable and to monitor attendance and presence in accordance with Terms of Engagement.
- Minutes are always formatted clearly and stored in a Google Drive folder for everyone to see and recall what has been discussed in previous meetings.

3.4. Choice of project management tools

3.4.1. Communication

- WhatsApp⁴
 - WhatsApp was chosen as the first communication tool as it was essential that we had a communication channel as soon as possible. WhatsApp was used as it requires no prior setup and all team members were already using it regularly.
 - It was used to convey general information and as an quick and informal way to schedule meetings.
- Slack⁵
 - Slack was introduced to the team as WhatsApp was adequate for the general operations but not for more specific tasks such as code review or file sharing.
 - Slack has effective cross-platform integrations which let the team share and work from different devices in a seamless and efficient manner.
 - Slack also offers channels which the team used to divide and focus work such as documentation, design and engineering in their individual channels.



6

² See Project Document Definition Section 3.4

³ An example can be found in Appendix B

⁴ <https://www.whatsapp.com/>

⁵ <https://slack.com/intl/en-gb/>

⁶ Figure 1 : Slack Channels

3.4.2. Project Planning and Monitoring

- Trello⁷
 - Trello was used as a project planning and management tool. It is completely optimized for the agile scrum methodology used by the team.
 - It allowed us to plan sprints during meetings - we populated our board with user stories and monitored their real time completion for each team member.
 - Allows the Project Manager to assign tasks directly on each user story or documentation to a member or several from the team.
 - It had the added benefit of allowing external members such as the project sponsor to monitor the team's progress in real time.⁸

3.4.3. File Sharing

- Google Drive⁹
 - Google Drive was an obvious choice for this project given several facts:
 - Firstly, privacy, security and integrity were guaranteed by Google in accordance to their terms of agreement.
 - Secondly, documents can be distributed on the cross-device platform, making it easy to access by all at any time.
 - Finally, use of the storage was free, guaranteeing budget integrity.

3.5. Initial Plan of Deliverables and Milestones

The following table lists macro-milestones and completion dates, updated from the Project Definition Document¹⁰:

Item	Type	Start Date	Target Date
Project proposal	Deliverable	18/02/2019	22/02/2019
Work plan (Agile)	Milestone	27/02/2019	02/03/2019
Project pitch	Milestone	12/03/2019	15/03/2019
Project Definition Document Draft	Deliverable	22/02/2019	18/03/2019
Back-end infrastructure with testing	Milestone	07/03/2019	12/04/2019
Front-end infrastructure with testing	Milestone	07/03/2019	12/04/2019
Integration with testing	Milestone	12/04/2019	17/04/2019
Recipick beta version with user acceptance testing	Deliverable	17/04/2019	22/04/2019
Project demo	Milestone	05/05/2019	10/05/2019
Recipick release candidate with testing	Deliverable	22/04/2019	13/05/2019
Final audit report	Deliverable	18/03/2019	13/05/2019

⁷ <https://trello.com/en-GB>

⁸ An illustration of the board can be found in Appendix D of the PDD

⁹ <https://www.google.com/drive/>

¹⁰ See Project Definition Document Section 2.6

4. Design

4.1. Requirements

Upon deciding on Recipick as our project choice via a vote, we promptly began discussing possible features for the application. A new project-specific document was created, containing an informal, unrefined outline of the project's ideal functionalities. All members of the team were given permissions to add to this document to maximise the potential of Recipick.

After discussion, a list of higher level requirements was created¹¹. This list of features was influenced by our research on dietary needs in addition to the team's personal preferences for a recipe application. The team, comprised of university students, could most benefit from Recipick, and thus these features were considered to be most sought after based on our experiences in cooking as students. The engineering was also focused on tasks that were likely to be interesting and lead to the professional development of team members.

This list was then refined into a collection of user stories in the form of Trello cards. User stories were chosen as they emphasise user experience due to their point-of-view phrasing. This allows the team to focus on satisfying immediate user needs (11). Using Trello cards also enabled us to plan flexibly in accordance with our usage of the Agile method. User stories were categorised as either a base or extended feature, with base features being essential to achieve the fundamental objectives of the project, necessary for the first release version.

User story	Type
As a user, I want to be able to create an account to log into the app so that I can use the application's full features.	Base
As a user, I want to be able to reset my password if I had forgotten it so that I don't lose access to my account.	Base
As a user, I want to have a clear main app page so that I can easily navigate through the app.	Base
As a user, I want to be able to add my own recipe on to the system so that it is saved and others may view it.	Base
As a user, I want the app to keep track of ingredients I own so that I can be recommended recipes with ingredients already available to me.	Base
As a user, I want to be able to locate the nearest supermarket so that I can buy missing ingredients.	Base
As a user, I want to be able to add ingredients to a shopping list so that I am aware of the ingredients I need to buy to execute a recipe.	Base
As a logged-in-user, I want to be able to favourite recipes so that I can refer to them at a later date.	Base
As a user, I want to be able to find recipes based on ingredients I have at home so that I know what I can make.	Base
As a logged-in user, I want to be able to upload pictures of my cooked meals so that they are shown in the recipes.	Base
As a user, I want to be able to scale ingredients according to servings and/or ingredient weight so that I do not have to calculate these measurements manually.	Extended
As a user, I want to be able to view app-wide popular recipes so that I can find recipes	Extended

¹¹See PDD Section 2.4

and flavours that are currently trending.	
As a user, I want to be able to select multiple meals for a meal plan so that I can generate a shopping list containing all the ingredients to make those meals.	Extended
As a user, I want to be able to rate and review recipes so that I can inform other users of the recipe's quality.	Extended
As a user, I want to be able to filter recipes by rating so that I can find the best recipes first.	Extended

4.2. Changes to Initial Vision

In Recipick's initial vision, additional features such as filtering by cooking time or available equipment were revoked but duly changed using the change management procedures as specified in the Project Definition Document¹² in order to streamline the future. These features were moved from being high-level requirements to extended goals. The decision to do so allowed the team to focus on implementing core features that were necessary in order to deliver the best possible application that also achieved project objectives¹³.

4.3. Additional Requirements

The project should have room for extension thus its design should be based around the idea of an ever evolving application that can grow from multiple angles. The implementation of a login system was made necessary so that the Recipick can be expanded to become a more social platform in the future. The database can also be redesigned to accommodate more specific recipe information in terms of cuisine, diet and calories. Finally, a 'discover' tab should be prototyped so as to accommodate future commercialisation of Recipick¹⁴.

4.4. Applied Constraints

The following section outlines how the team applied the constraints from their theoretical aspects in Section 2 to their practical aspects and actualisation.

4.4.1. Offline Functionality

- Users should be able to access their saved recipes without an internet connection. Users can add their own recipes and access any recipes in their favourites as well as the recipes they added themselves. Upon accessing these recipes all the information provided while offline is the same as while online (Ingredients, method, photos). The design should therefore include implementation of this feature.

4.4.2. Sensors

- Camera
The application requires the use of the camera for taking and uploading photos of completed recipes to their pages. Both the options to take a photo from within Recipick or from the phone's gallery are to be made available in the implementation of the application. Storage and serving should be accounted for in this regard.
- Location
The app uses location data to get the location of a user and then returns the locations of all the nearest supermarkets. This allows users to find where their

¹² See PDD Section 6.1

¹³ See Appendix H

¹⁴ See Section 2.4.

nearest markets are should they need a quick trip to purchase some more ingredients. In terms of design, an adequate API is to be used in order to serve the locations. User location should be kept private and their integrity preserved.

4.4.3. Server-side data gathering component

- In order to use the app, a user creates an account where their credentials are stored on the server. The username, email and password is the only data that is provided by them, the rest are either generated (such as ID) or part of the app (such as uploaded recipes). This processing is done server-side in order to add protection and security.
- Ingredient filtering should be done server-side as to limit the number of calculations done on end devices and to make sure that all recipes which contain the filtered ingredients are served to the user.
- Location should be securely processed server-side in order to have the data fitted for the API that will be chosen for supermarket serving. The processing should include string normalization and truncating as to best accommodate said API.
- Photos will be uploaded by user and stored on the server-side. Sorting and serving will therefore also be processed server-side to avoid errors and to make sure that all photos are available from the database.

4.5. Analysis of Privacy, Ethical and Social Aspects

4.5.1. Data Protection

- Data protection is always the primary concern for the user. The only sensitive data that is needed at the moment is a password which is kept in an encrypted password field meaning that even us as developers are unable to find out what it is. The design of the app should therefore further emphasize data protection and integrity.
- No extension in terms of the collection of data is required for the app's functionality. The app requires a display name for easy readability, a password to log in and an email to reset a password should the user forget or have to change due to an account breach (on this application or another). The use of usernames facilitates the protection of user identity and protects their integrity when using Recipick.
- Recipes are designed with the option to be kept private, therefore guaranteeing the ability to protect anything that is uploaded to the platform by users.

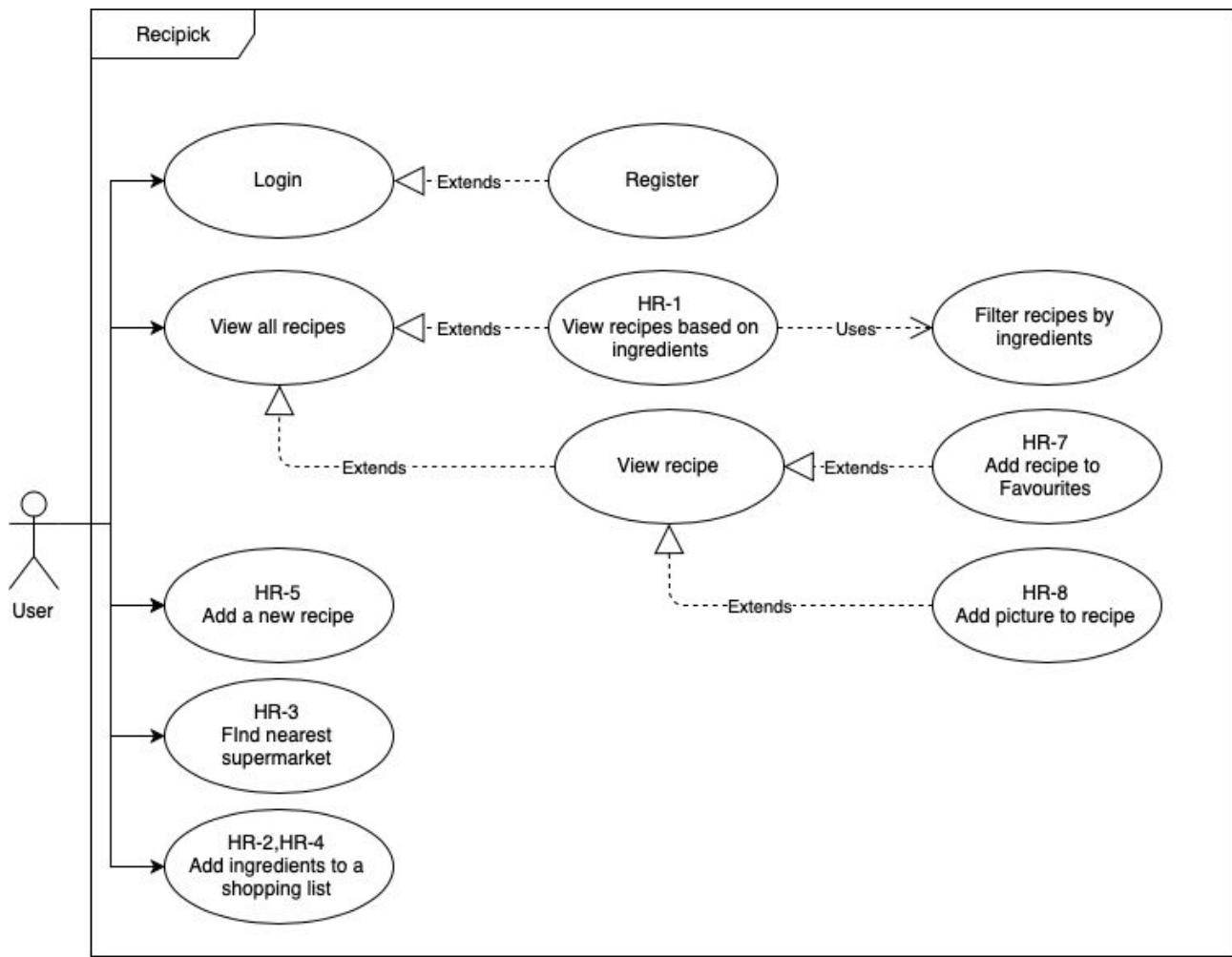
4.5.2. Ethical and Social Aspects

- In order to conform to the '*Code of conduct for data-driven health and care technology*' (13), several ethical concerns must be considered. The two most important ethics

would be the **respect of persons** and of **human rights**. The Code implies the necessity of understanding user needs and context. In this regard, the design should further include metrics and labels such as calories, cuisines and diets.

- The app should be inclusive of all people, regardless of identity or culture. This, in terms of design, translates to an intuitive, ergonomic and easy to use application, with room for extension in terms of filtering of cuisines, diets and ingredients.
- Moderation should be made available in order to avoid copyright issues with uploaded recipes or images on Recipick.

4.6. Use Case Diagram



15

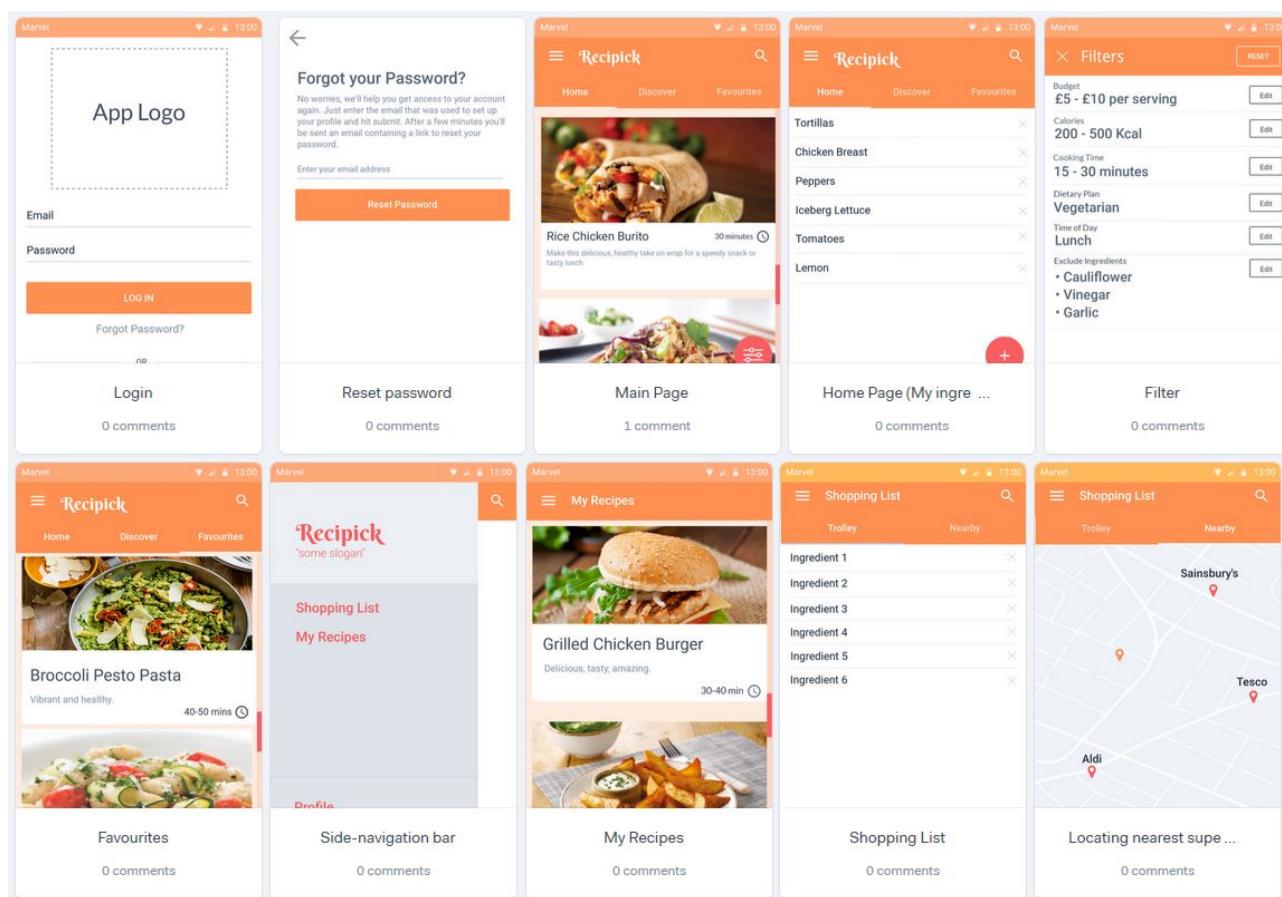
The use case diagram reflects the constraints as defined on the PDD under section 2.4 (High-Level Requirements) and further explained in 6.2.2 of this document.

4.7. User Interface

- Upon project approval, two rough mockups of Recipick were hand-drawn and presented during a group meeting.

¹⁵ Figure 2 : Use case diagram

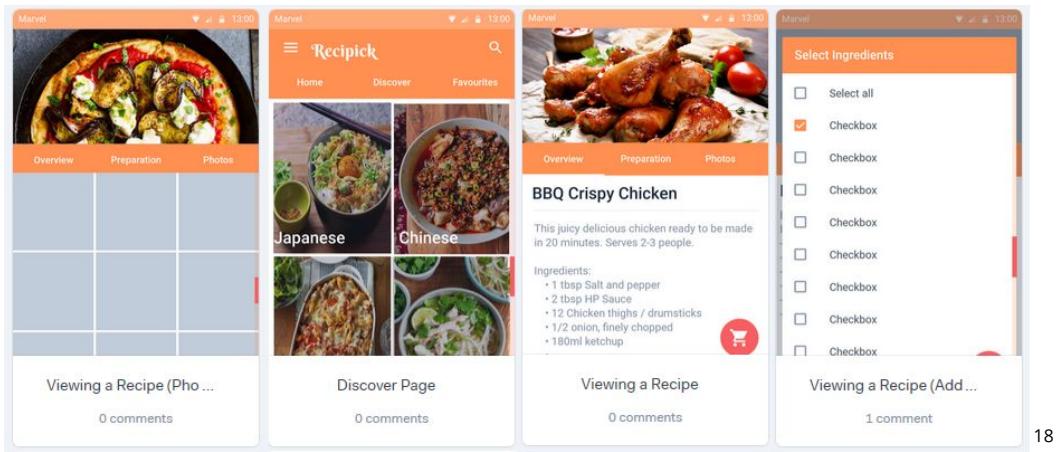
- Team members critiqued and expressed which specific aspects of each mockup they most preferred. These criticisms were considered and taken into account when creating the final mockups of the UI.
- The final mockups were created using Marvel¹⁶ as it was a free-to-use browser-based design platform with a gentle learning curve, especially beneficial as the team had little to no experience in creating UI mockups.
- Creating UI mockups before implementation also facilitated the smooth development of the application as individual team members could easily refer to the mockups during implementation, allowing the UI of Recipick to achieve a level of professionalism with the use of consistent colour schemes and themes.
- The main priority was to create an intuitive experience for users. As Recipick aims to be a benefit to society, it is crucial that the main functionalities of the application are as easily accessible as possible.



17

¹⁶ <http://marvelapp.com/>

¹⁷ Figure 3: UI designs for Recipick



18

4.8. Logo

- Recipick's logo went through multiple initial designs. Ultimately, a logo that gives a strong first impression and fits well with our objectives is necessary to attract potential users and to cultivate a lasting, friendly brand image.



19

- This logo design has a strong culinary theme with the intention of creating the image that the application was the users' personal chef and that food is within reach.
- With Recipick's logo available from the first candidate release, the foundations for Recipick's brand identity can be built from the start. This is beneficial when the application eventually achieves its extended goals of commercialisation²⁰.

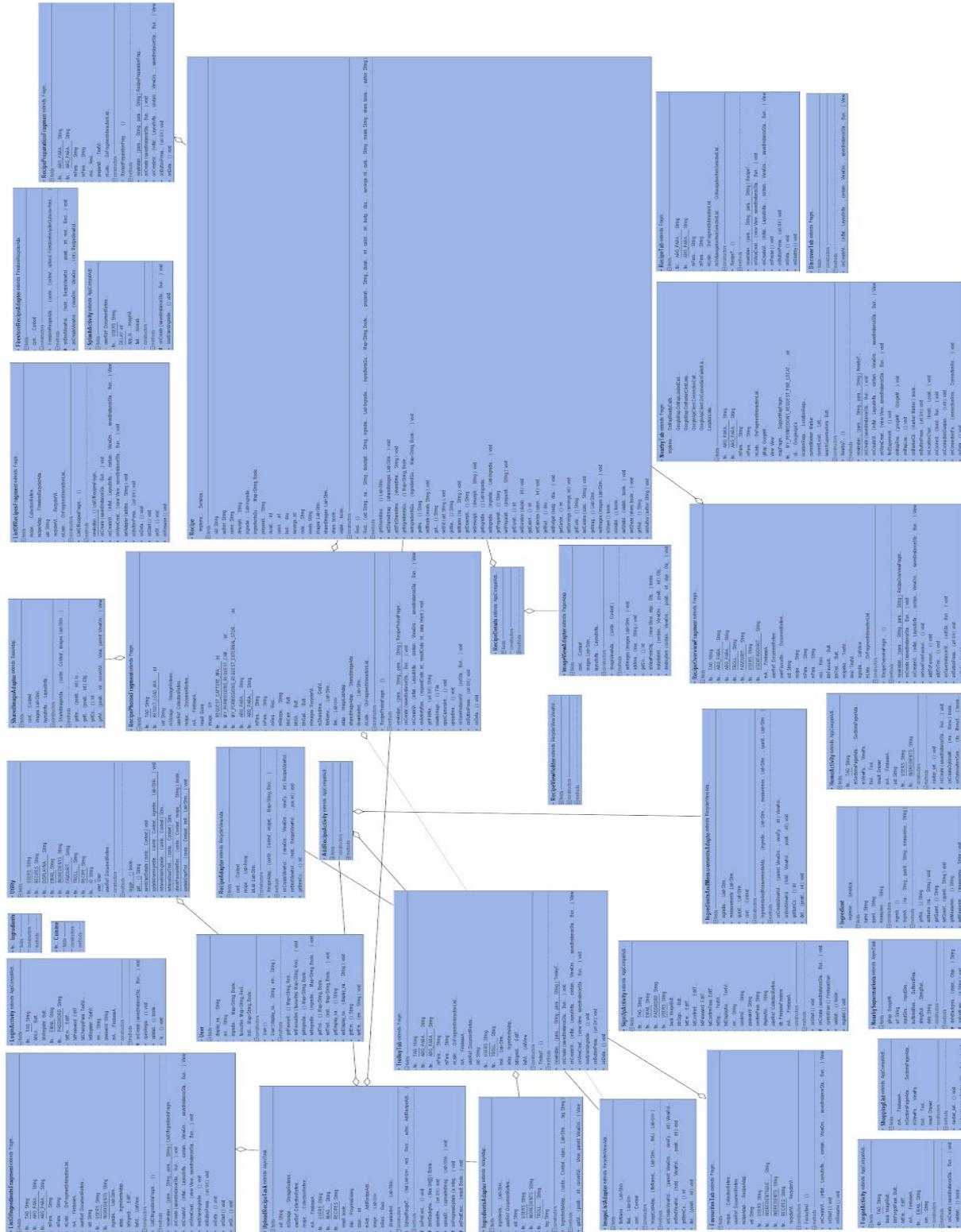
¹⁸ Figure 4: UI designs for Recipick continued

¹⁹ Figure 5 : Logo

²⁰ See Section 2.4

4.9. Class Diagrams

The following UML diagram was established during design and extended with implementation. This shows the full extent of the code base for any team that would take the project in the future.



²¹Figure 6: UML Class Diagram

5. Implementation

5.1. Choices of Tools and Technologies

1. Android Studio

- We selected Android Studio to develop our proposal since we all had experience using the IDE. All but one member of the group have prior experience developing small feature packed applications; such as a To-Do List application or an application that features location services.
- Android Studio is a popular choice for developing Android applications amongst developers, the alternatives to the IDE are:
 - IntelliJ IDEA
 - Microsoft Visual Studio
 - React Native
 - Unreal Engine
 - Apache Cordova
 - Corona
- Most of these IDEs are not appropriate for the proposal and simply do not have the tools and support like Android Studio does. Android Studio's Software Development Kit has a free set of tools that includes documentation, debuggers, emulators, frameworks, libraries and more. It provides a user-friendly experience to developers and allows plugins to be added onto the IDE.
- From research we found that we could make our application cross-platform by developing it on something like React Native. It would be much easier to fix bugs and be much more cost and time effective to manage our project if we were to have our implementation on both iOS and Android devices.
- As a group we concluded that Android Studio heavily outweighs the other alternatives, mainly due to the group's familiarity using Android Studio and the project constraint requiring the implementation to be an 'Android-based mobile application'.

2. Java

- Upon choosing Android Studio we had to make a decision on the programming language when developing our application. The narrowed down options were Java, Kotlin and C++.
- The advantages of Kotlin includes an array of simple syntaxes and is fundamentally an object-oriented, functional programming language allowing the developer to use both paradigms (14). It was also important to consider that our implementation could utilise both Kotlin and Java, though the implications of this is that members could have difficulty understanding code especially since the team has no prior knowledge of Kotlin.
- We decided to reject the idea of using Kotlin and C++ since the team is more experienced with Java. It may be time consuming to learn Kotlin and thus delay progress during sprints as well as affecting the team's work balance.
- The reason for avoiding C++ was that we were confident in our java code-writing abilities and each member of the group has at least 1 year of experience using Java.

3. Firebase Cloud Storage²²

For our implementation we could use a wide range of backend servers such as:

- | | | |
|------------|-----------|---------------|
| - Firebase | - Kinvey | - Backendless |
| - Back4App | - Node.js | - kuzzle |

From research we found that the current flavour in server-side technology seemed to be Node.js. It is an increasingly popular backend tool for various applications such as web development and cross-platform development of applications. Google's Firebase is another strong contender which comes fully equipped with all the support, guides and tools required to develop high-quality applications. Our group favoured Firebase over others because it was simple to use and understand. More importantly, some members of our team had previous experience implementing personal projects using firebase.

Firebase protects data²³ and conforms with the EU General Data Protection Regulation (GDPR). It complies with major privacy and security standards which means that Firebase is a safe environment to store our application's data.

Firebase Authentication²⁴ provides the services which allows users to sign in and register to our application using their email address and password. Its SDK provides methods to manage authentications, allowing users to sign in, register and reset their passwords. It is very safe and secure and allows the engineers in our team to quickly implement the login system feature allowing more focus on the design aspect.

We began implementing our backend using Firebase Realtime Database but ultimately discarded it in favour of its successor Firebase Cloud storage. The key difference between the two is that Firebase Realtime database does not have the features to accommodate the complex filtering required in our application. Firebase Cloud Storage on the other hand is able to chain filters together and sort documents in a collection in a single query, which is quite the contrary to Realtime database. This meant that we had to re-do all the code that was already implemented using Realtime Database. Fortunately, the ideology and syntaxes were quite similar, making the transition to Cloud Storage a relatively simple process. Our team had no prior knowledge of using Firebase Cloud Storage, though the experience our team had from using Realtime Database was translatable to Cloud Storage.

Firebase is also extensively scalable and reliable due to specialist tools such as Real-time databases, crashlytics, cloud firestore and test lab. However, there are some limitations that we must consider²⁵ when using the free Spark Plan from Firebase, these are the following:

²² <https://firebase.google.com/>

²³ <https://firebase.google.com/support/privacy>

²⁴ <https://firebase.google.com/docs/auth>

²⁵ <https://firebase.google.com/docs/database/usage/limits>

- Can only have up to 100,000 simultaneous connections.
- Can only have 100,000 per second responses sent from a single database.
- Can only store up to 5GB of images to Storage.
- Can only store up to 1GB of data on Cloud Firestore.

If our application exceeds these limitations, Firebase recommends scaling our application with multiple databases. The advantages of this is that our database performance would be optimised and the risk of overloading our database will be reduced. This requires our data to be spread across multiple database instances. Furthermore, Firebase provides paid packages so that we could store more data and images.

4. GitLab

We used gitlab as it is provided by the University and each member has a student account already set up. We require Git to track and review changes in our project. More importantly, it allows members of our group to work on different parts of the application simultaneously through the use of branches. GitLab ensures that the repository is private allowing only our team to interact and work on our application. It also provides a set analytical tools to show contributions and provides a user-friendly experience.

5.2. Strategy and methods

- Our strategy involves using an agile method to simultaneously work on different features. The stories which needed implementation were determined at each weekly standing meeting, prototyped, implemented, tested and peer-reviewed in order to guarantee a fast, modular development workflow.
- In terms of commenting standards, the Oracle commenting standard was followed due to most team members being familiar with it. The standard also serves well in the optic of passing the project as the commenting is very thorough and complete, making it easy to understand the code.
- Each completed user story was peer reviewed for maximum security and quality in terms of the code.
- The database design was done with the idea of ethics and privacy in mind. Passwords and sensitive data were protected by default due to the tools we used. Ethics were furthermore respected as a high level of granularity was made available when designing the recipes collection in the database. Fields for dietary restrictions, cuisines and calories have been added as to permit users to filter and feel included in the application with its future development.
- Using remote version control allowed for simultaneous development of features. The use of branching guaranteed that merge clashes would be reduced in the long term.
- All processes and methods mentioned above also permitted team members to develop important and essential professional skills and approaches regarding coding, commenting and deploying such an application.

5.3. Code Documentation

Code documentation was completed using Oracle JavaDoc, therefore guaranteeing a thorough and complete documentation with regards to possible future development of the application. With such a process in place, a double benefit aspect can be viewed. The first benefit stems from the completeness of this documentation, which would allow any new developers to understand the code base easily, once again, allowing the team to focus on important new features rather than losing time trying to decipher old code. The second added benefit is documentation speed. JavaDoc allows for code to be automatically documented almost as soon as it passes tests, therefore guaranteeing an up to date and complete documentation at all stages in the development workflow.

The whole project documentation is available as an HTML file which is attached to this document²⁶. Examples are available in the appendices²⁷.

²⁶ See attached HTML documentation.

²⁷ See Appendix D.

6. Evaluation

6.1. Peer-to-Peer Reviews

The peer-to-peer review log demonstrates thorough vetting of code and documentation along the respect of schedule and timeline constraints. Code requirements will be referenced using the notation found in the Project Definition Document²⁸

Date	Author	Reviewer	Task	Status
09/03/19	Nithesh	Clyde	R1	Pass
09/03/19	Clyde	Nithesh	R2	Pass
21/03/19	Nithesh	Clyde	R3	Pass
24/03/19	Clyde	Nithesh	R4	Pass
06/04/19	Nithesh	Clyde	R5	Pass
06/04/19	Clyde	Nithesh	R6	Pass
20/04/19	Clyde	Nithesh	R7	Pass
09/04/19	Nithesh	Clyde	R8	Pass
21/04/19	Clyde	Nithesh	R9	Pass
21/04/19	Nithesh	Clyde	R10	Pass
20/04/19	Nithesh	Clyde	R11	Pass
21/04/19	Nithesh	Clyde	R12	Pass
21/04/19	Nithesh	Clyde	R13	Pass
15/04/19	Oscar	Sze	PDD V1.0 - Part 1	Pass
15/04/19	Sze	Oscar	PDD V1.0 - Part 2	Pass
12/05/19	Oscar	Pavlos	FAR: Introduction	Pass
12/05/19	Pavlos	Oscar	FAR: Definition	Pass
12/05/19	Oscar	Sze	FAR: Planning	Pass
12/05/19	Sze	Oscar, Pavlos	FAR: Design	Pass
12/05/19	Nithesh	Sze, Oscar	FAR: Implementation	Pass
12/05/19	Max, Mohammad	Nithesh	FAR: Documentation	Pass
12/05/19	Clyde	Max	FAR: Evaluation	Pass
12/05/19	Oscar	Mohammad	FAR: Reflection	Pass

²⁸ See Project Definition Document Section 2.

6.2. Test Plan and Unit Testing

6.2.1. Requirements Testing

This section highlights the requirements testing for the application. It consists of how the testing of the requirements is carried out, providing evidence of their evaluation. In order to fulfill the evaluation criteria, 100% of the tests must pass.

6.2.2. Higher Level Requirements

- **HR1.** To track the ingredients a user has and suggest recipes based on those ingredients.
- **HR2.** To allow users to add missing ingredients for a recipe to a shopping list.
- **HR3.** To inform users of the nearest supermarkets to purchase missing ingredients utilising the user's location (i.e. with the device's GPS sensor).
- **HR4.** Shop planning, where users select meals to generate a shopping list of ingredients.
- **HR5.** To add new recipes which can be stored privately or shared publicly.
- **HR6.** To allow moderators to process and review user recipes before they are published.
- **HR7.** To add other users' recipes to their favourites to view at a later date and while offline.
- **HR8.** Ability to take and share pictures of their freshly cooked meals from following recipes.
- **HR9.** To ensure that a subset of Recipick's functionality continues to operate in spite of an unstable Internet connection, i.e. when a user continuously reconnects and disconnects to the application due to a poor and unstable Internet signal.

6.2.3. Lower Level Requirements (As Agile User Stories)

- **R1.** As a user, I want to be able to create an account.
- **R2.** As a user, I want to be able to log into my account.
- **R3.** As a user, I want to be able to reset my password.
- **R4.** As a user, I want to be able to navigate the app through the main application page.
- **R5.** As a user, I want to be able to add my own recipes.
- **R6.** As a user, I want to be able to keep track of my ingredients.
- **R7.** As a user, I want to be able to add ingredients to my shopping list.
- **R8.** As a user, I want to be able to locate the nearest supermarket to buy ingredients.
- **R9.** As a user, I want to be able to view recipes.
- **R10.** As a user, I want to be able to find recipes based on their ingredients.
- **R11.** As a user, I want to be able to favourite recipes.
- **R12.** As a user, I want to be able to upload pictures of my cooked meal after following a recipe.
- **R13.** As a user, I want to be able to view my uploaded recipes and favourites when there is no Internet signal.

6.2.4. Requirements Test Plan

Test No.	Requirement No.	Preconditions/ Dependencies	Expected Inputs	Expected Results	Test Evaluation
1	R1	<p>Assume that user has Internet connection.</p> <p>Assume that user has pressed the “Register” option in the login page to create an account.</p> <p>Assume that user has an email address that has not been registered before.</p>	<p>Enter display name.</p> <p>Enter valid email address.</p> <p>Confirm email address.</p> <p>Enter password.</p> <p>Confirm password.</p> <p>All fields in this page must be filled, otherwise registration would be invalid.</p>	<p>Valid user details is stored in Firebase Authentication.</p> <p>User is prompt back to the login page of the application with a toast message; “Registered successfully.”</p>	<p>By inspecting Firebase, more specifically under Firebase Authentication - will be able to see all valid users that have been registered.</p> <p>By also inspecting the toast message when registered successfully.</p> <p>Screenshots will be provided to confirm these outcomes.</p>
2	R1	<p>Assume that user has Internet connection.</p> <p>Assume that user has pressed the “Register” option in the login page to create an account.</p> <p>Assume that user has an INVALID email address.</p>	<p>Enter display name.</p> <p>Enter INVALID email address.</p> <p>Confirm INVALID email address.</p> <p>Enter password.</p> <p>Confirm password.</p> <p>All fields in this page must be filled.</p>	<p>Invalid user will not be stored in Firebase Authentication.</p> <p>User will get a toast message that states, “Failed to register user! Please try again later!”</p>	<p>By inspecting Firebase Authentication - will be able to see all VALID users that have been registered, so the invalid email address will not be displayed/stored here.</p> <p>Also inspecting the toast message when registered unsuccessfully</p> <p>Screenshots will be provided to confirm these outcomes.</p>

3	R1	<p>Assume that user has Internet connection.</p> <p>Assume that user has pressed the “Register” option in the login page to create an account.</p> <p>User does not need any valid details but can use them if desired.</p>	Fill every required field but one or do not enter anything at all.	<p>User will get a toast message that states, “Invalid details! Check the entered details and try again”.</p>	<p>By inspecting the toast message when trying to register with at least one of the fields empty.</p> <p>A screenshot will be provided to confirm this outcome.</p>
4	R1	<p>Assume that user has Internet connection.</p> <p>Assume that user has pressed the “Register” option in the login page to create an account.</p> <p>User does not have to use a valid email address but both have to be different.</p>	Fill every required field but for “Enter your email address” and “Confirm your email address” fields, user must enter two different email address.	<p>User will get a toast message that states, “Invalid details! Check the entered details and try again”.</p>	<p>By inspecting the toast message when trying to register with at least one of the fields empty.</p> <p>A screenshot will be provided to confirm this outcome.</p>
5	R1	<p>Assume user does NOT have Internet connection.</p> <p>Assume that user has pressed the “Register” option in the login page to create an account.</p> <p>Assume the user types in something invalid or valid in one of the fields.</p>	Can enter fields with invalid details or valid details. They can also be empty.	<p>User will get a toast message that states, “Invalid details! Check the entered details and try again”. It will print “Failed to register user! Perhaps an account with the username already exists, or check your internet connection!” If the validation passes but there's no internet connection</p>	<p>By inspecting the toast message when trying to register without Internet connection.</p> <p>A screenshot will be provided to confirm this outcome.</p>
6	R1	<p>Assume user does NOT have Internet connection.</p> <p>Assume that user has pressed the “Register” option in the login page to create an account.</p> <p>Assume that user uses a valid email address</p>	Enter details in all fields. (Fields highlighted in Test #1)	<p>User will get a toast message that states, “Failed to register user! Perhaps an account with the username already exists, or check your internet connection!”</p>	<p>By inspecting the toast message when trying to register with an existing email address.</p> <p>Also inspecting Firebase Authentication - does not store two</p>

		that already exists within the Firebase Authentication.			of the same email address. Screenshots will be provided to confirm these outcomes.
7	R2	Assume that user is registered. Assume that user has internet connection.	Enter registered email address. Enter valid password for that registered email address.	User will get a toast message that states, "Successfully Logged in!" and will be directed to the main application page.	By inspecting the toast message when logged in. Also inspecting that the user will be taken to the main application page. Screenshots will be provided to confirm these outcomes.
8	R2	Assume that user is not registered. Assume that user has internet connection.	Enter unregistered email address. Enter password.	User will get a toast message that states, "Invalid details, Check the entered details and try again"	By inspecting Firebase Authentication - user's email address is not stored. Also by inspecting the toast message when user tries log in with unregistered email address. Screenshots will be provided to confirm these outcomes.
9	R2	Assume that user is registered. Assume that user has no internet connection.	Enter registered email address. Enter valid password for that registered email address.	User will get a toast message that states, "Invalid details, Check the entered details and try again"	By inspecting the toast message when logged in. A screenshot will be provided to confirm this outcome.

10	R3	<p>Assume that user is registered.</p> <p>Assume that user has internet connection.</p> <p>Assume that user has pressed the "Forgot Password?" option.</p>	<p>Enter registered email address and press "Reset Password" button.</p>	<p>User will get a toast message that states, "Email successfully sent. Please check your email."</p> <p>User will also get an email sent to the email address they entered with a link that allows them to change their password.</p>	<p>By inspecting the toast message when the "Reset Password" button is pressed.</p> <p>Also by inspecting the email sent to the email address and by clicking the link within that email to allow the user to reset their password.</p> <p>Screenshots will be provided to confirm these outcomes.</p>
11	R3	<p>Assume that user has internet connection.</p> <p>Assume that user has pressed the "Forgot Password?" option.</p>	<p>Enter field with text that does not follow an email address format.</p>	<p>User will get a toast message that states, "Invalid email address."</p>	<p>By inspecting the toast message when the "Reset Password" button is pressed.</p> <p>A screenshot will be provided to confirm this outcome.</p>
12	R3	<p>Assume that user has a valid unregistered email address.</p> <p>Assume that user has internet connection.</p> <p>Assume that user has pressed the "Forgot Password?" option.</p>	<p>Enter field with the valid unregistered email address.</p>	<p>User will get a toast message that states, "Email address entered is not registered or there is no internet connection".</p>	<p>By inspecting the toast message when the "Reset Password" button is pressed.</p> <p>A screenshot will be provided to confirm this outcome.</p>
13	R3	<p>Assume that user is registered.</p> <p>Assume that user has no internet connection.</p> <p>Assume that user has pressed the "Forgot Password?" option.</p>	<p>Enter registered email address and press "Reset Password" button.</p>	<p>User will get a toast message that states, "Email address entered is not registered or there is no internet connection".</p>	<p>By inspecting the toast message when the "Reset Password" button is pressed.</p> <p>A screenshot will be provided to confirm this outcome.</p>

14	R4	<p>Assume that user is logged in.</p> <p>Assume that user has internet connection.</p>	<p>No inputs - only testing on what the user can see once logged in.</p>	<p>Users will be able to see the main application page which consists of the tabs:</p> <ul style="list-style-type: none"> -Recipe -Discover -Favourites <p>which can all be navigated by swiping left and right.</p> <p>They also have access to the side navigation bar which consists of more components of the application, such as:</p> <ul style="list-style-type: none"> -Home -Shopping List -My recipes -Add a recipe 	<p>By inspecting the main application page - navigating through the application to see if every component of the application is accessible.</p> <p>Screenshots will be provided to confirm these outcomes.</p>
15	R5, HR5	<p>Assume that user has internet connection and is logged in.</p> <p>Assume that user has allowed permission for camera and external storage for the application.</p> <p>Assume that user is in "Add a recipe" component of the application.</p>	<ul style="list-style-type: none"> -Enter recipe name -Upload a picture(s) of the recipe -Add a brief description about the recipe. -"Share" on (turns coloured when it is on) if user wants to share the recipe, off if not. -Enter ingredients for the recipe -Enter quantity and measurement -Enter the preparation details -Add duration -Add budget -Add calories -Add number of servings -Add cuisine (N/A is sufficient) -Add meal type (e.g. breakfast, lunch or dinner) <p>User then clicks on "Create Recipe" button.</p>	<p>A toast message that states, "New recipe is added".</p> <p>A loading dialog that says, "Uploading a recipe" with a loading icon.</p> <p>Recipe listed in the "Recipe" tab in the main application page along with the other shared recipes from other users.</p> <p>Recipe stored in Firebase Cloud Storage.</p>	<p>By inspection and a screenshot of the toast message, dialog message, recipe within the "Recipe" tab and in Firebase Cloud Storage.</p>

16	R5	<p>Assume that user has internet connection and is logged in.</p> <p>Assume that user has allowed permission for camera and external storage for the application.</p> <p>Assume that user is in "Add a recipe" component of the application and wants the recipe to be viewed by everyone.</p>	<p>User does not fill all the required fields listed out in Test #15.</p> <p>User then clicks on the "Create Recipe" button.</p>	<p>A toast message that states, "Make sure all fields are filled."</p>	<p>By inspection and a screenshot of the toast message after pressing the "Create Recipe" button.</p>
17	R5	<p>Assume that user has internet connection and is logged in.</p> <p>Assume that user has allowed camera and storage permission for the application.</p> <p>Assume that user is in "Add a recipe" component of the application and wants the recipe to be viewed by everyone.</p>	<p>User uploads more than 7 pictures (from their gallery or using their camera).</p>	<p>A toast message that says, "You can only upload 6 images!"</p>	<p>By inspection and a screenshot of the toast message after trying to upload a 7th image.</p>
18	R5	<p>Assume that user has NO internet connection and is logged in.</p> <p>Assume that user is in "Add a recipe" component of the application and wants the recipe to be viewed by everyone.</p>	<p>May enter all fields required (not needed).</p> <p>User then clicks on "Create Recipe".</p>	<p>A toast message that states, "Please check your internet connection."</p>	<p>By inspection and a screenshot of the toast message after pressing the "Create Recipe" button.</p>
19	R6, HR1	<p>Assume that user is logged in and has internet connection.</p> <p>Assume that user is at the "List of Ingredients" tab within the "Recipe" tab of the application.</p>	<p>User enters ingredient that they currently have then press "Add" to add it to the list of their ingredients (can add more ingredients if desired).</p>	<p>The ingredient(s) will appear in the list below the "Add an Ingredient" field.</p>	<p>By inspection and a screenshot of the process of adding an ingredient and observing the ingredient being added to the list.</p>

20	R6	Assume that user is logged in and has internet connection. Assume that user is at the "List of Ingredients" tab within the "Recipe" tab of the application.	User enters an ingredient that already exists in the list into the input text field and then presses the "Add" button.	A toast message will appear that states, "Ingredient already exists".	By inspection and a screenshot of the process of trying to add an ingredient and observing the toast message appearing.
21	R6	Assume that user is logged in and has internet connection. Assume that user is at the "List of Ingredients" tab within the "Recipe" tab of the application.	User enters a non-alphabetical character (i.e. *&^%)	A toast message will appear that states, "Invalid characters".	By inspection and a screenshot of the toast message.
22	R6	Assume that user is logged in and has internet connection. Assume that user is at the "List of Ingredients" tab within the "Recipe" tab of the application.	User presses "Add" with an empty field.	A toast message will appear that states, "Invalid Ingredient!"	By inspection and a screenshot of the toast message after pressing "Add" with an empty field.
23	R6	Assume that user is logged in and has NO internet connection. Assume that user is at the "List of Ingredients" tab within the "Recipe" tab of the application.	User enters an ingredient in the text field and presses "Add".	A toast message will appear that says, "Please check your internet connection."	By inspection and a screenshot of the toast message after the list of ingredients is pressed.
24	R7, HR2	Assume that user is logged in and has internet connection. Assume that user is at the "Trolley" tab within the "Shopping List" component of the application.	User enters ingredient then press "Add" to add it to their shopping list (can add more ingredients if desired).	The ingredient(s) will appear in the list below the "Add an Ingredient" field.	By inspection and a screenshot of the process of adding an ingredient and observing the ingredient being added to their trolley list.
25	R7	Assume that user is logged in and has internet connection. Assume that user is at the "Trolley" tab within the "Shopping List"	User enters an ingredient that already exists in the list into the input text field and then presses the "Add" button.	A toast message will appear that states, "Ingredient already exists".	By inspection and a screenshot of the process of trying to add an ingredient and observing the toast message appearing.

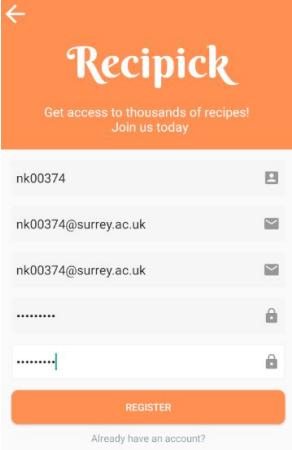
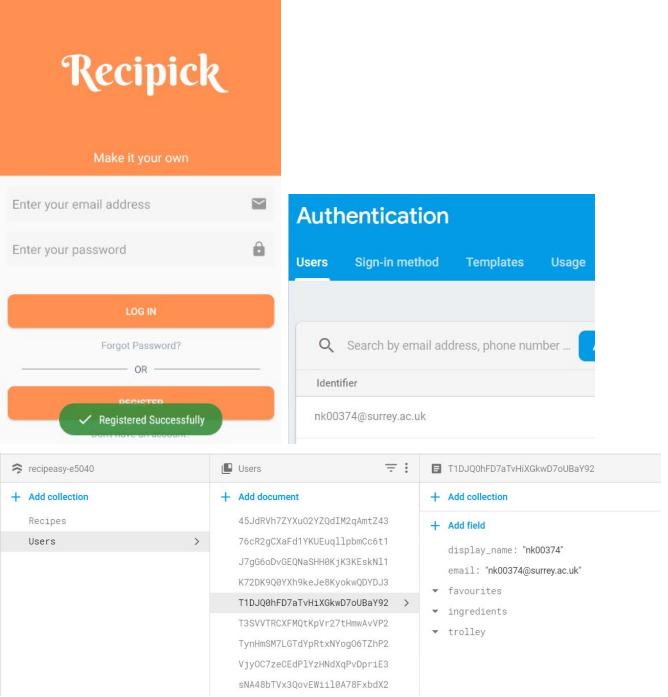
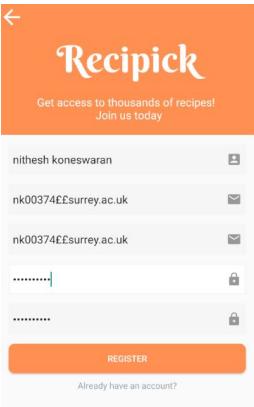
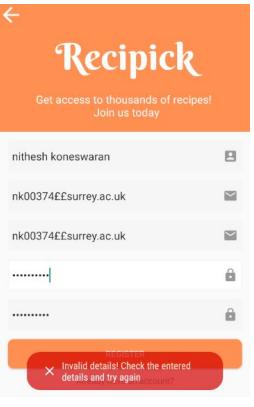
		component of the application.			
26	R7	Assume that user is logged in and has internet connection. Assume that user is at the "Trolley" tab within the "Shopping List" component of the application.	User presses "Add" with an empty field.	A toast message will appear that states, "Invalid Ingredient!"	By inspection and a screenshot of the toast message after pressing "Add" with an empty field.
27	R7	Assume that user is logged in and has NO internet connection. Assume that user is at the "Trolley" tab within the "Shopping List" component of the application.	User enters an ingredient in the text field and presses "Add".	A toast message will appear that says, "Please check your internet connection."	By inspection and a screenshot of the toast message after the list of ingredients is pressed.
28	R8, HR3	Assume that user is logged in and has internet connection. Assume that user has allowed permission for location to be enabled for the application. Assume that user is at the "Nearby" tab within the "Shopping List" component of the application.	User presses "Find Supermarkets".	A toast message will appear that states, "This may take a while. Press it again if it takes too long." Application zooms into the user's current location and displays nearest supermarkets on the map.	By inspection and screenshots of the toast message, user's current location and the nearby supermarkets after the button is clicked.
29	R8	Assume that user is logged in and has internet connection. Assume that user has NOT allowed permission for location to be enabled for the application. Assume that user clicks on "Shopping List" in the side navigation bar.	No inputs.	A dialog appears that tells the user that their GPS is currently enabled and gives them the option to open their location settings and enable it, so that their location the nearby supermarkets can be detected.	By inspection and a screenshot of the alert dialog.

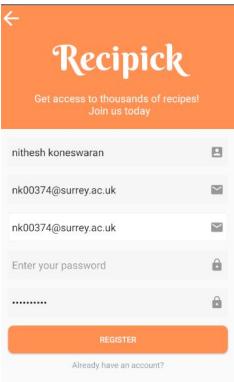
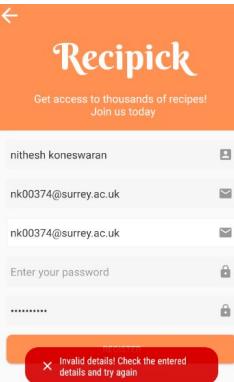
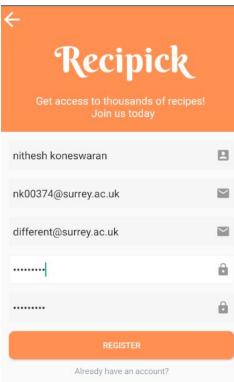
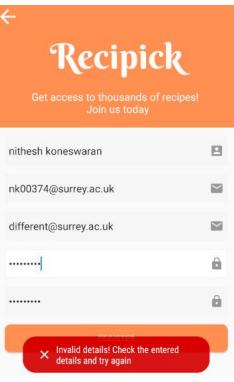
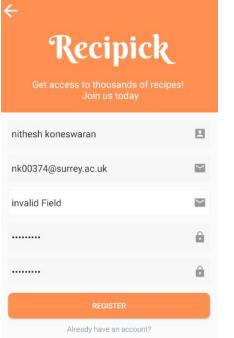
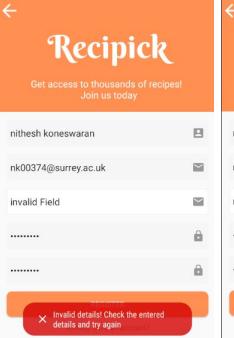
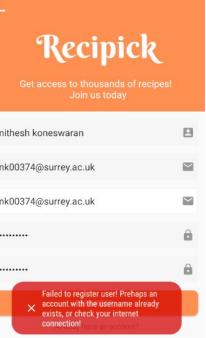
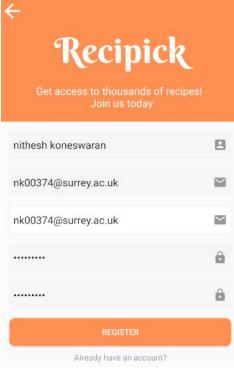
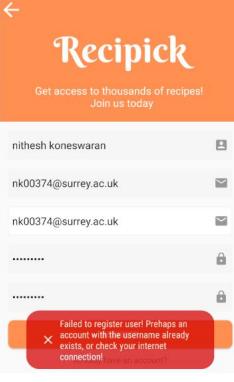
30	R8	<p>Assume that user is logged in and has NO internet connection.</p> <p>Assume that user has allowed permission for location to be enabled for the application.</p> <p>Assume that user clicks on "Shopping List" in the side navigation bar.</p>	User clicks on "Find Supermarkets" button.	A toast message that states, "Please check your internet connection."	By inspection and a screenshot of the toast message after the button "Find Supermarkets" is clicked.
31	R9, HR9	<p>Assume that user is logged in.</p> <p>Assume that user is at the "Recipe" tab.</p>	User clicks on a recipe listed in the fragment.	Recipe "Overview" tab is displayed and user is able to access "Preparation" and "Photos" tab by swiping right from "Overview" tab.	By inspection and screenshots of the "Overview", "Preparation" and "Photos" tab with their contents (based on the clicked recipe).
32	R10, HR1	Assume Test #19 passes and its pre-conditions are still met.	User now clicks "Recipe" within the "Recipe" tab.	Recipes based on user's ingredients are displayed in the "Recipe" tab.	By inspection and screenshots from Test #19 as well as the recipes displayed based on user's ingredients in the "Recipe" tab.
33	R11	<p>Assume user is logged in and has internet connection.</p> <p>Assume that Test #31 passes and its pre-conditions are still met.</p>	User scrolls down "Overview" tab and clicks on "Add To Favourite" button.	<p>A toast message that states, "Recipe added to Favourites".</p> <p>Recipe is added to user's "Favourites" tab.</p>	By inspection and screenshots from Test #31 as well as the recipes in the "Favourites" tab favourited by the user.
34	R11	Assume user is logged in and has NO internet connection.	User scrolls down "Overview" tab and clicks on "Add To Favourite" button.	A toast message that says, "Please check your internet connection."	By inspection and a screenshot of the toast message.

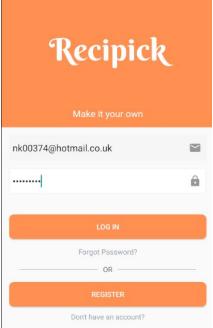
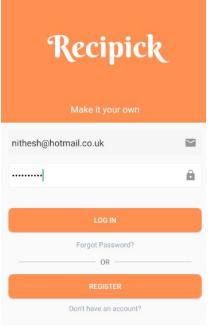
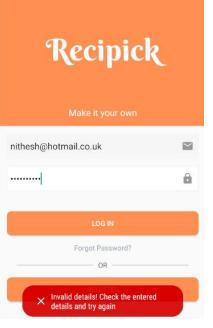
35	R12, HR8	<p>Assume that user is logged in and has internet connection.</p> <p>Assume that user has allowed camera and storage permission for the application.</p> <p>Assume that user is viewing a recipe and is on the "Photos" tab of that recipe.</p>	<p>User uploads image either from gallery or taking a picture.</p> <p>Then clicks on the "Share" button.</p>	<p>A toast message that states, "Uploading...", while the image is uploading.</p> <p>After image has been uploaded, a toast message displays, "Your photo has been shared!"</p>	By inspection and screenshots of the toast messages.
36	R13, HR9	<p>Assume that Test #15 passes.</p> <p>Now, assume that user is logged in and has NO internet connection.</p> <p>Assume that user is in "My recipes" component" of the application which is located in the side-navigation bar.</p>	No inputs.	Users sees their uploaded recipe.	By inspection and screenshot of the recipe uploaded by the user.
37	R13, HR7, HR9	<p>Assume Test #33 is fulfilled.</p> <p>Assume that user is logged in and has NO internet connection.</p> <p>Assume that user is in the "Favourites" tab in the main application page.</p>	No inputs.	Favourited recipe will be listed under "Favourites" tab.	By inspection and screenshots used in Test #33 as well as a screenshot of the favourited recipe under the "Favourites" tab.
38	HR4	<p>Assume user is logged in and has internet connection.</p> <p>Assume that Test #31 passes and its pre-conditions are still met.</p>	<p>User scrolls down "Overview" tab and clicks on "Add To Shopping List" button.</p>	<p>A toast message that states, "Recipe ingredients added to your shopping list".</p> <p>Missing ingredients of the recipe will be added to the shopping list under "Trolley" tab in the "Shopping List" component of the application.</p>	By inspection and screenshot(s) from Test #31 as well as the missing ingredients in the "Trolley" tab.

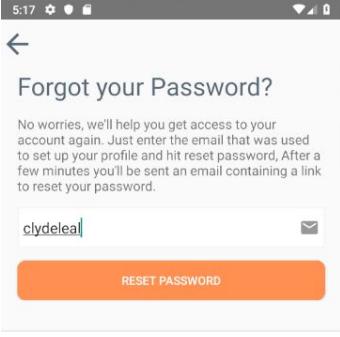
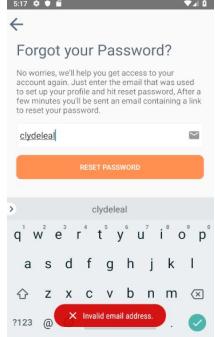
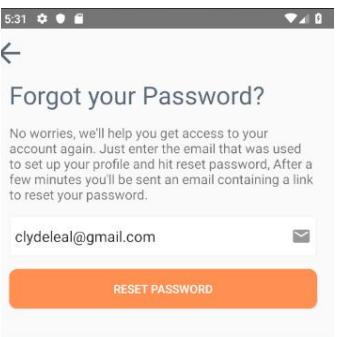
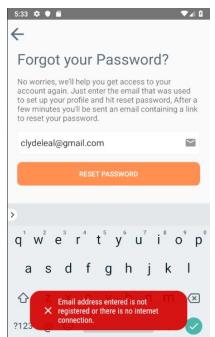
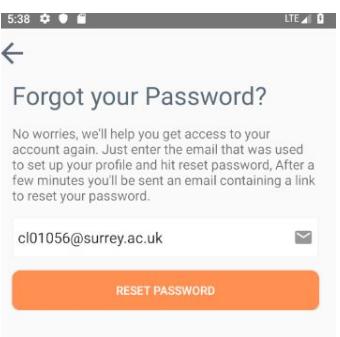
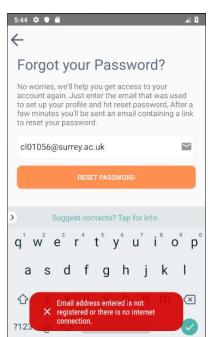
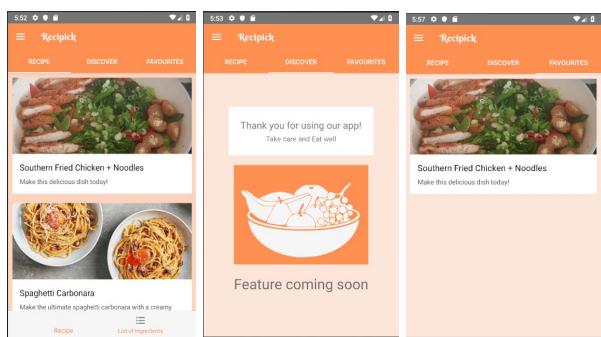
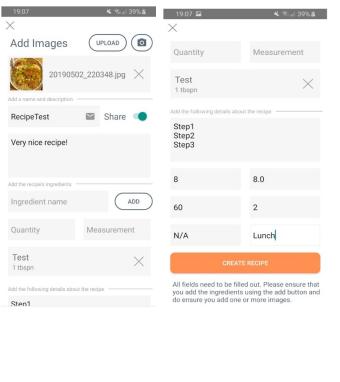
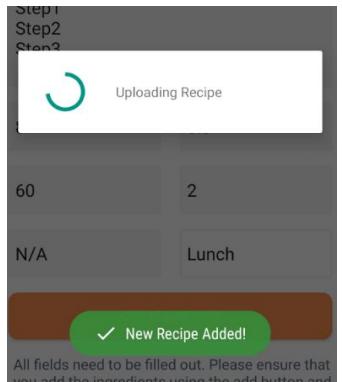
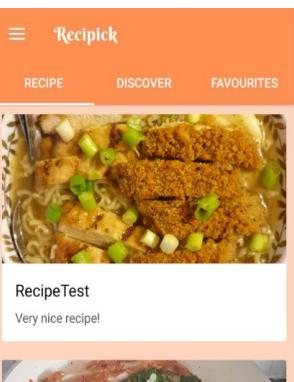
39	HR6	Assume that moderator has access to Firebase server.	A user creates and submits a recipe.	Submitted recipe is stored in Firebase as a document in a collection called "Recipes". The moderators are able to review recipes from firebase cloud storage and if it is not appropriate it can be removed.	By inspection and screenshot of the process of removing a recipe document from the database and checking the main page again.
----	-----	--	--------------------------------------	--	---

6.2.5. Testing Report

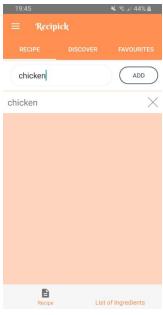
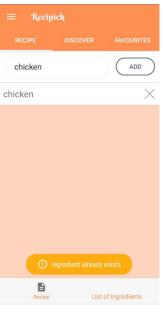
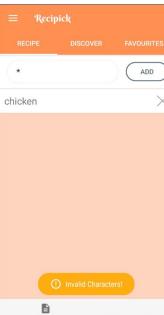
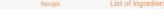
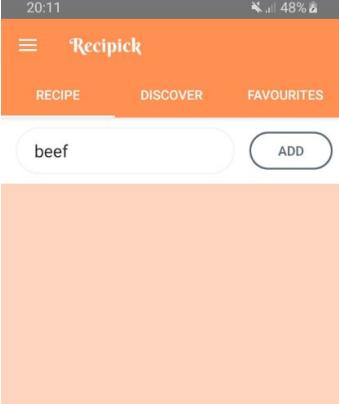
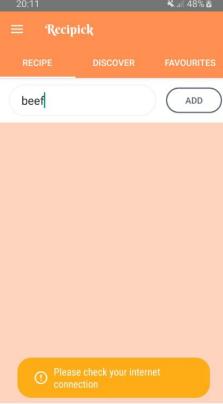
Test no.	Input	Output	Test Status
1			Test passes.
2			Test passes.

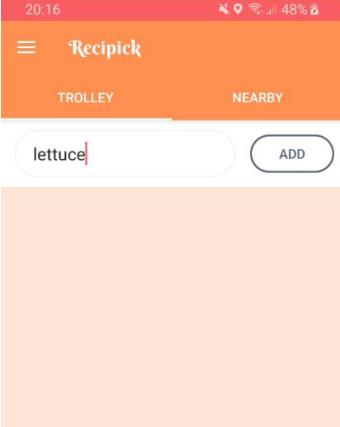
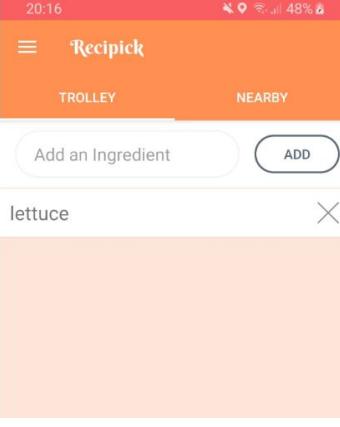
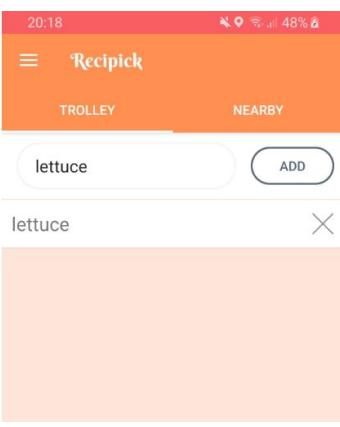
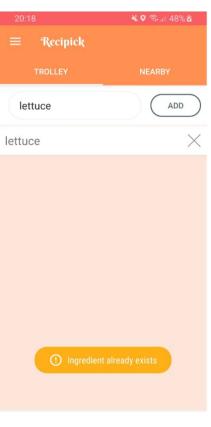
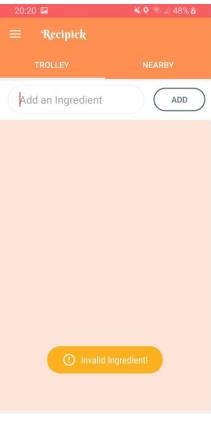
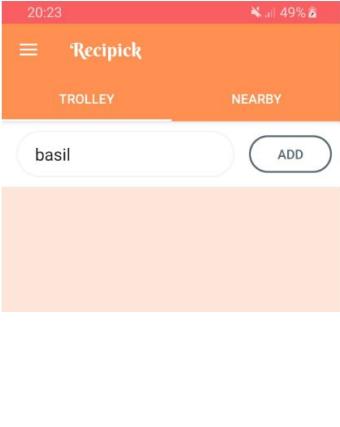
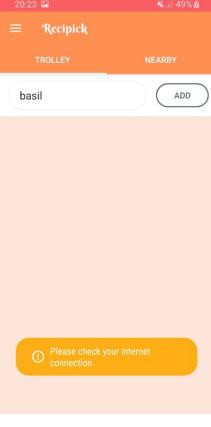
3			Test passes.
4			Test passes.
5		 	Test passes.
6			Test passes.

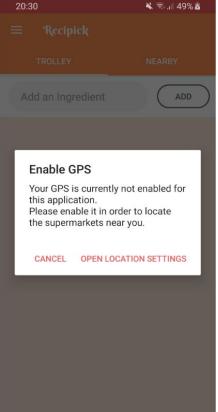
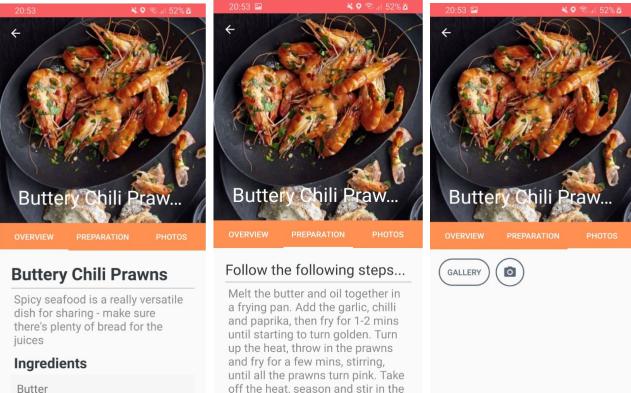
7			Test passes.
8			Test passes.
9			Test passes.
10	<p>Forgot your Password?</p> <p>No worries, we'll help you get access to your account again. Just enter the email that was used to set up your profile and hit reset password. After a few minutes you'll be sent an email containing a link to reset your password.</p> <p>nithesh@hotmail.co.uk</p> <p>RESET PASSWORD</p>	<p>Forgot your Password?</p> <p>No worries, we'll help you get access to your account again. Just enter the email that was used to set up your profile and hit reset password. After a few minutes you'll be sent an email containing a link to reset your password.</p> <p>nithesh@hotmail.co.uk</p> <p>RESET PASSWORD</p> <p>Email successfully sent. Please check your email</p> <p>Reset your password for Recipick</p> <p>StaffAtRecipick <noreply@recipeeasy-e5040.firebaseioapp.com> Sun 12/05/2019 17:37 nithesh@hotmail.co.uk ↗</p> <p>Hello,</p> <p>Follow this link to reset your Recipick password for your nithesh@hotmail.co.uk account.</p> <p>https://recipeeasy-e5040.firebaseioapp.com/_auth/action?mode=resetPassword&cobCode=zwnP_9aMaL6TNR24yqWIC-8wH3g_VBYN3uTizFWzmYAAAFqrOfahQ&apiKey=AlzaSyA3Rlub1UbX47WKVYdPjCE1x5H7t7seAc&lang=en</p> <p>If you didn't ask to reset your password, you can ignore this email.</p> <p>Thanks,</p> <p>Your Recipick team</p>	Test passes.

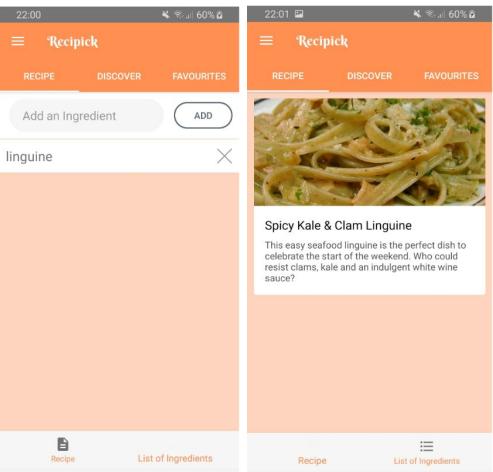
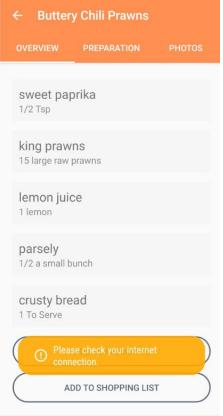
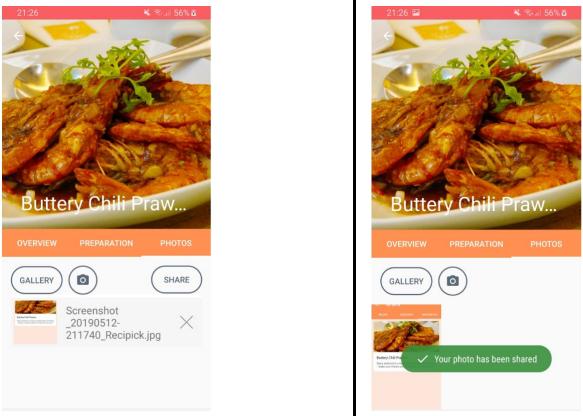
11	 	Test passes.	
12	 	Test passes.	
13	 	Test passes.	
14	No inputs		Test passes.
15	  	Test passes.	

16		<p>na</p> <p>Dinner</p> <p>! Make sure all fields are filled.</p> <p>All fields need to be filled out. Please ensure that you add the ingredients using the add button and do ensure you add one or more images.</p>	Test passes.
17			Test passes.
18			Test passes.
19			Test passes.

20	 	Test passes.	
21	 	Test passes.	
22	Button press - "Add".	 	Test passes.
23	 	Test passes.	

24	 	Test passes.
25	 	Test passes.
26	Button press - "Add".	
27	 	Test passes.

28	Button press - "Find supermarkets".		Test passes.
29	No inputs.		Test passes.
30	Button press - "Find supermarkets".		Test passes.
31	Click on a recipe.	 <p>Follow the following steps...</p> <p>Melt the butter and oil together in a frying pan. Add the garlic, chilli and paprika, then fry for 1-2 mins until starting to turn golden. Turn up the heat, throw in the prawns and fry for a few mins, stirring until all the prawns turn pink. Take off the heat, season and stir in the</p>	Test passes.

32	Click on “Recipe” fragment within “Recipe” tab.		Test passes.
33	Button press - “Add to Favourite”.		Test passes.
34	Button press - “Add to Favourite”.		Test passes.
35			Test passes.

36	No inputs		Test passes.
37	No inputs		Test passes.
38	Button press - "Add To Shopping List"		Test passes.
39			Test passes.

6.2.6. Unit Testing

Test Class	Output	Test Status
RecipeTest.java	<pre> Run: app x RecipeTest x Run: app x RecipeTest x Test Results RecipeTest ✓ Get type of meal. 16 ms ✓ Get duration. 16 ms ✓ Get recipe ingredients 10 ms ✓ Get recipe description 1 ms ✓ Is it visible to everyone. 2 ms ✓ Get budget. 1 ms ✓ Get recipe preparation. 1 ms ✓ Get calories. 1 ms ✓ Get recipe name. 1 ms ✓ Get user ID. 1 ms ✓ Is recipe shared. 1 ms </pre>	Passed
IngredientTest.java	<pre> Run: app x IngredientTest x Run: app x IngredientTest x Test Results IngredientTest ✓ Change quantity. 15 ms ✓ Change measurement. 12 ms ✓ Get the name of the ingredient. 1 ms ✓ Create an Ingredient object. 1 ms ✓ Change ingredient name. 1 ms ✓ Get measurement. 1 ms ✓ Get quantity. 1 ms </pre>	Passed
UserTest.java	<pre> Run: app x UserTest x Run: app x UserTest x Test Results UserTest ✓ Get favourite recipes. 12 ms ✓ Get user email. 8 ms ✓ Setting user ingredients. 2 ms ✓ Set favourite recipes. 1 ms ✓ Get ingredients from Trolley. 1 ms ✓ Get display name. 1 ms ✓ Get ingredients. 1 ms ✓ Setting user ingredients. 1 ms </pre>	Passed

6.3. Privacy and Ethical Evaluation

6.3.1. Respect of User Privacy (GDPR Criteria)

- The use of Firebase, as mentioned in Section 5.1, certifies the respect of privacy, therefore fulfilling the criteria for security and privacy as outlined in the Project Definition Document²⁹.
- The whole implementation pipeline follows privacy requirements to a high standard in terms of data integrity and protection of user data. It is also good to note that the architecture would very easily allow for extension by allowing users the ability to download their own data from Recipick's database.

6.3.2. Respect of Users and Ethics

- The implementation of Recipick was made with ethics and social concerns in mind, causing the developers to include fields for diets, cuisines and calories which can be made readily available to users with further development of the application.
- The platform passes evaluation in term of social inclusiveness and can be further developed for accessibility purposes (such as bigger fonts for the visually impaired or changing colors to accommodate colour-blindness).
- The ability to moderate recipes was also made available for future development as to preserve the integrity of Recipick for all its users.
- If sensitive or offensive content was to be created, it is demonstrated in testing that it can be very quickly flagged and deleted.
- The moderation and control over the database also ensures that no copyright violation can arise. This process can be automated in the future in order to avoid ethical or legal violations.

²⁹See PDD Section 3.3

6.4. Evaluation of Team Dynamics

6.4.1. Peer-to-Peer Assessment

In order to quantify and qualify the teamwork in terms of the actual human relations during the project, a form³⁰ was devised, filled out by each group member and submitted to the project manager. This fills out the team evaluation criteria from the Project Definition Document³¹.

For the sake of integrity, individual reports will be kept anonymous in this report. The following evaluation is therefore the sum of comments and responses from the whole team.

1. What team members found positive:

It seems that the project and its execution pleased the team members. The project and methods used to develop and document it were appropriate and interesting.

2. What team members found negative:

The general consensus was that no significant negative aspects were found up to submission in this group project.

3. Did team members feel like they had too much work compared to others?

It was made clear on a regular basis that team members could come to the manager if they felt work was unevenly distributed. This created a spirit of trust and members did not find that they were working too much or too little.

4. How did team members rate communication within the group?

Members appreciated the different channels of communication which were open to them. Most of them were positively surprised by the speed and availability of other members.

5. How did team members rate spirit and motivation within the group?

The project was deemed interesting and challenging to all, therefore nurturing a dynamic and spirited environment which was deemed as a positive by the team members.

6. Did team members feel valued in the group?

Members overall felt valued in the group, not only by their individual contribution but by the fact that everyone was heard and opinions respected.

7. Did team members encounter issues that needed solving?

No major issues that needed solving arise during the project duration.

8. General remarks

Team members seemed satisfied by the work they and their teammates have done which is a positive for the project.

³⁰ See Appendix C

³¹ See PDD Section 3.3

6.5. Respect of timeline and milestones

The Following chart updates the one available in Section 1 - Planning. It also serves as the proof for the timing and schedule criteria from the Project Definition Document³² and shows we hit the 100% target in terms of schedule.

Labels : Completed on time - Completed (Late) - Not Completed

Item	Type	Start Date	Target Date	Status
Project proposal	Deliverable	18/02/2019	22/02/2019	Completed
Work plan (Agile)	Milestone	27/02/2019	02/03/2019	Completed
Project pitch	Milestone	12/03/2019	15/03/2019	Completed
Project Definition Document Draft	Deliverable	22/02/2019	18/03/2019	Completed
Back-end infrastructure with testing	Milestone	07/03/2019	12/04/2019	Completed
Front-end infrastructure with testing	Milestone	07/03/2019	12/04/2019	Completed
Integration with testing	Milestone	12/04/2019	17/04/2019	Completed
Recipick beta version with user acceptance testing	Deliverable	17/04/2019	22/04/2019	Completed
Project demo	Milestone	05/05/2019	10/05/2019	Completed
Recipick release candidate with testing	Deliverable	22/04/2019	13/05/2019	Completed
Final audit report	Deliverable	18/03/2019	13/05/2019	Completed

A copy of the Trello board used by team members can be inspected at :

<https://trello.com/b/4wBV1BV4/recipick-scrum-public>

³²See PDD Section 3.3

6.6. Respect of Budget

1. The initial budget for the project can be found in the Project Definition Document³³.
2. The working framework involved all Internet and Electricity Expenses to be handled by the University. Furthermore, no unforeseen expenses have appeared at this stage, rendering the **effective spent budget £0**.

6.7. Beta Testing Reports

The beta was distributed to friends of team members for feedback along with the presentation of the project for them to navigate. Feedback was collected with a form that can be found in Appendix A. Their name will be kept anonymous in order to maintain integrity in the feedback.

Tester	Feedback - Positive	Feedback - Negative	Action Taken
Beta Tester #1	Liked the U.I and general flow of app	N/A	N/A
Beta Tester #2	General Concept and Design	Not enough recipes	Added more recipes to database
Beta Tester #3	Shops Tab and Recipe Tab	Not enough recipes	Added more recipes to database
Beta Tester #4	Shops Tab	Scrolling Flow	Fixed the Scrolling
Beta Tester #4	App Flow, U.X	Scrolling Flow	Fixed the Scrolling

The beta feedback was encouraging but from a critical point of view, no actual actionable negative feedback was received. This will be further examined in Section 7.1.1 below.

The user adoption criteria from the Project Definition Document will fully be assessed during the User Acceptance Testing. It is however observable that 100% of Beta Testers could use the application with the required 'minimal explanation'³⁴.

³³ Section 2.8

³⁴ See PDD Section 3.3

7. Reflection

7.1. Aspects for Improvement

7.1.1. Beta Testing

The beta testing process had to be completed with very few testers as our classmates and ourselves were quite busy and hard to contact during the period. Feedback was scarce and not effective in the improvement of Recipick. In order to have a better beta testing experience, 10 or 15 users should have been involved in the testing and the feedback document should be made more exhaustive.

7.1.2. Continuous Integration

Continuous integration could have been included from the start but given time and technical constraints, was foregone in favor of functionality and a simpler deployment pipeline. It would have been an interesting extension of the actual pipeline for the project.

7.1.3. Code Testing

Testing was done in an effective and systematic manner, however, an advanced instrumentation testing process, in line with the aforementioned Continuous Integration pipeline would have been an amazing addition to the project if it were more long term.

7.1.4. Code Documentation

At the start of the project, research was made by several team members regarding a way to effectively and exhaustively document the project codebase. The team stumbled upon Javadoc, an Oracle framework³⁵ which has been useful as to auto generate documentation as code was written and tested. The framework however proved difficult to use in our case figure. Documentation was properly generated but could have been more complete in the optic of transferring the project to another team for further development.

³⁵ <https://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>

7.2. Positive Aspects of the project

7.2.1. Timeliness and Scheduling

As referenced in the **Planning** Section, macro-deadlines were always met in a timely manner. This not only let the group demonstrate good organisational skills but also to review work periodically in a peer-to-peer fashion as seen in the **Evaluation** Section. Furthermore, peace of mind was granted to all as we knew we could rely on each other's meticulous timeliness in a systematic and regular basis. This demonstrates of the teams ability to lead and complete a documented software project over several months and assures that an extended version could be created in the future.

7.2.2. Team Dynamics

It often happens that teams have internal conflicts and struggles. This arises in most cases because of pride, miscommunication and similar issues. These issues were top priority for this project from the start. Members, even if assigned specific roles, were reminded at all times that if they felt that the workload was inadequate for them (excess or the opposite), that they should let the Project Manager straight away. This had the effect of avoiding jealousy or feelings of inequality. Miscommunication was avoided by using clear and systematic communication channels, along with frequent check-ups and peer reviews for code and documentation. This also demonstrates that the team could keep on developing and maintaining Recipick long term which is an extremely positive outcome for all.

7.2.3. Choices of Technology and Coding Tools

The tools used in the development of the app showed to be good choices. The Android Studio framework was familiar to most of the team, therefore reducing techno-fatigue and letting them focus on the core content and code rather than then tools and processes. Furthermore, the use of GitLab for code and version control let parallel coding take place amongst different engineers, speeding up the process overall. The use of Firebase Store also guaranteed that privacy would not be an issue for users with the added benefit of automatic scaling the user base. The development pipelines used (branching, peer-reviews, Javadoc) could also very easily be applied to an increasingly bigger team of developers if the application was to be further developed.

7.2.4. Choices of Methodology and Non-Coding Tools

The chosen communication tools proved to be very practical, especially Slack. The division of labour through the use of channels could be scaled to a much larger team with more divisions in the future if need be. The Trello board was enthusiastically used by all team members and made it easy for everyone to know what to do and for when. It also allowed for effective monitoring from the Project Manager and Project Sponsor. Finally, the Google Drive file sharing system let the team members work on the same documents simultaneously, encouraging the modern practice of remote work which is a good personal improvement for each team member. The Drive and Trello would both be greatly scalable if the number of team members was to grow in the future.

7.2.5. Testing Procedure

It was made very clear in our initial evaluation criteria that testing needed to be thorough and complete. Testing furthermore needed to be done on every aspect of the higher level requirements. The procedure ended up being more in depth than expected as the members

in charge of testing also took the time to test every single user story and lower level requirement from the project which resulted in every evaluation criterion being satisfied.

7.2.6. Respect of Privacy, Ethical and Social aspects

The use of Firebase made sure that privacy concerns were addressed for users due to Google's adherence of data protection and privacy laws. The high granularity and attributes of each recipe created in the database design allow for detail in the selection of recipes in terms of cuisine, diets and a number of other variables. Filtering can therefore easily be made available to users in a future version of the application.

8. References

1. Derla K. Living Alone Linked To Poor Diet: Single People Eat Fewer Fruits, Veg And Opt For Ready-Made Meals [Internet]. Tech Times. 2019 [cited 12 May 2019]. Available from: <https://www.techtimes.com/articles/102847/20151104/living-alone-linked-to-poor-diet-single-people-eat-fewer-fruits-veg-and-opt-for-ready-made-meals.htm>
2. The risks of poor nutrition :: SA Health [Internet]. Sahealth.sa.gov.au. 2019 [cited 12 May 2019]. Available from: <https://www.sahealth.sa.gov.au/wps/wcm/connect/public+content/sa+health+internet/healthy+living/is+your+health+at+risk/the+risks+of+poor+nutrition>
3. Ajayi S. Effect of Stress on Employee Performance and Job Satisfaction: A Case Study of Nigerian Banking Industry. [Internet]. 2018 [cited 12 May 2019];. Available from: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3160620
4. Sprake E, Russell J, Cecil J, Cooper R, Grabowski P, Pourshahidi L et al. Dietary patterns of university students in the UK: a cross-sectional study. Nutrition Journal. 2018;17(1).
5. Jina Tanton, Lorna J. Dodd, Lorayne Woodfield, and Mzwandile Mabhala, "Eating Behaviours of British University Students: A Cluster Analysis on a Neglected Issue," Advances in Preventive Medicine, vol. 2015, Article ID 639239, 8 pages, 2015. <https://doi.org/10.1155/2015/639239>.
6. Wolfson JA, Bleich SN. Is cooking at home associated with better diet quality or weight-loss intention? Public Health Nutrition. Cambridge University Press; 2015;18(8):1397–406.
7. Nutrition in the News Facts behind the headlines [Internet]. Nutrition.org.uk. 2019 [cited 12 May 2019]. Available from: <https://www.nutrition.org.uk/healthyliving/healthydiet/healthybalanceddiet.html>
8. Cancer Prevention Diet [Internet]. HelpGuide.org. 2019 [cited 12 May 2019]. Available from: <https://www.helpguide.org/articles/diets/cancer-prevention-diet.htm/>
9. How 'Millennials' Are Changing Food as We Know It [Internet]. Forbes.com. 2019 [cited 12 May 2019]. Available from: <https://www.forbes.com/sites/bethhoffman/2012/09/04/how-millenials-are-changing-food-as-we-know-it/#5703cf634041>
10. Scrum Methods in Agile [Internet]. International Scrum Institute. 2019 [cited 11 May 2019]. Available from: https://www.scrum-institute.org/Introduction_to_Scrum_A_Real_World_Example.php
11. Ravlani K. 6 Benefits of Good User Stories - Agile For Growth [Internet]. Agile For Growth. 2019 [cited 12 May 2019]. Available from: <https://agileforgrowth.com/blog/userstory-benefits/>
12. Realtime Database vs. Cloud Firestore — Which Database is Suitable for your Mobile App [Internet]. Medium. 2019 [cited 12 May 2019]. Available from: <https://medium.com/datadriveninvestor/realtime-database-vs-cloud-firestore-which-database-is-suitable-for-your-mobile-app-87e11b56f50f>
13. Code of conduct for data-driven health and care technology [Internet]. GOV.UK. 2019 [cited 12 May 2019]. Available from: <https://www.gov.uk/government/publications/code-of-conduct-for-data-driven-health-and-care-technology/initial-code-of-conduct-for-data-driven-health-and-care-technology>
14. Java vs. Kotlin: Which is the Better Option for Android App Development? | Clearbridge Mobile [Internet]. Clearbridge Mobile. 2019 [cited 12 May 2019]. Available from: <https://clearbridgemobile.com/java-vs-kotlin-which-is-the-better-option-for-android-app-development/>

9. Appendices

9.1. Appendix A: Beta Feedback Form

PLEASE FILL IN ALL FIELDS:	
Name	
Date Submitted	
What you found positive	
What you found negative	
How would you rate the U.I ? (tick one)	<input type="checkbox"/> Bad <input type="checkbox"/> Medium <input type="checkbox"/> Good <input type="checkbox"/> Great
Did finding shops around you take time ?	<input type="checkbox"/> Yes <input type="checkbox"/> No
How would you rate the ergonomics ? (tick one)	<input type="checkbox"/> Bad <input type="checkbox"/> Medium <input type="checkbox"/> Good <input type="checkbox"/> Great
On a scale from 1 to 10 how would you rate working with this group ?	
Did you encounter any bugs or issues when running the application ?	
Comments	

9.2. Appendix B: Example Minutes

Meeting held on 07th March 2019 at 16:00 to 17:00 in Group Study Room 3

Present

Nithesh, Clyde, Pavlos, Mohammad, Max, Sze

Apologies

None

Agenda

1	Approval of previous minutes The minutes of the meeting held on 26th February 2019 and 5th March 2019 were confirmed and approved by the team.
2	Agile briefing Motion: To brief on how the team's workflow is to be influenced by the Agile methodology. Sze gave a brief explanation on how the Agile methodology and Scrum framework would be utilised in the implementation of the app. The Trello has now been set up with a backlog that contains tasks to complete, and a list for the first weekly sprint. The cards of user stories and tasks to be addressed during a specific sprint shall be moved to that week's sprint, and once they have been complete, shall be moved to the weekly sprint's completed list. With this new system in place, tasks were then given to all members of the team to address within a week. Action: <ul style="list-style-type: none">• EVERYONE to populate the backlog with tasks and user stories.• EVERYONE to order the backlog by priority and to projected due dates to tasks.
3	Discuss UI mockups Motion: To comment and critique on the UI mockups produced by Nithesh and Clyde. Nithesh and Clyde presented their UI mockups which were produced by pencil on paper to the team. The mockups featured a rough but clear idea of what the app may look like after it has been implemented. The mockups were satisfactory to the team. They conveyed their disagreements on how the search filter system should be structured. Sze suggested that a compromise could be made, with Clyde's category filtering system to be used for the 'Discover' page with cuisine cards (e.g. Japanese, Italian cuisine) and Nithesh's filtering system to be used for the search function.

	<p>Nithesh also suggested that each recipe page should contain 3 tabs, containing the ingredients, preparation instructions and images respectively.</p> <p>Action:</p> <ul style="list-style-type: none"> • CLYDE, NITHESH to produce further refined mockups for the project pitch.
4	<p>Discuss implementation design</p> <p>Motion: To discuss the details of implementing certain features of the app.</p> <p>The fields for registering a new account were agreed to be a user's first name, last name, email address, password and display name. As for the recipe rating system, recipes can be ranked by either their average 5-star score or a thumbs-down and thumbs-up ratio.</p>
5	<p>Miscellaneous</p> <p>As implementing the app has begun, members were reminded to regularly post updates to their code in order to be peer reviewed by others, either by posting snippets on the Slack or to inform others to check the git repository.</p> <p>Members are also reminded to work on their individual contribution report and to keep track of any goals and self-improvements they have achieved.</p>
6	<p>Date of next meeting</p> <p>Tuesday, 12th March 2019, 13:00 to 14:00 at TB20A</p>

9.3. Appendix C: Peer-to-Peer Assessment Form

PLEASE FILL IN ALL FIELDS:	
Name	
How do you find team cohesion ?	
Do you think you are doing too much ? Why ?	
How would you rate communication ?	<input type="checkbox"/> Bad <input type="checkbox"/> Medium <input type="checkbox"/> Good <input type="checkbox"/> Great
How would you rate team spirit and motivation ?	<input type="checkbox"/> Bad <input type="checkbox"/> Medium <input type="checkbox"/> Good <input type="checkbox"/> Great
Do you feel valued as a team member ?	<input type="checkbox"/> Yes <input type="checkbox"/> No
Did you encounter team issues that need to be resolved ?	
Further Comments	

9.4. Appendix D: Documentation

The screenshot shows a JavaDoc interface with three main sections:

- Packages:** A sidebar listing packages: com.example.softeng.recipick, com.example.softeng.recipick.Activities, com.example.softeng.recipick.Adapters, com.example.softeng.recipick.AsyncTasks, com.example.softeng.recipick.Fragments, com.example.softeng.recipick.Models, and com.example.softeng.recipick.ViewHolders.
- Packages Overview:** A table showing package names and descriptions.
- All Classes:** A sidebar listing all classes: AddRecipeActivity, AddRecipeActivityTest, ForgotActivity, HomeActivity, PlaceholderFragment, IngredientsAdapter, IngredientsAndMeasurementsAdapter, IngredientsList, IngredientsListTest, ListOfIngredientsFragment, ListOfIngredientsFragment.OnFragmentInteractionListener, ListOfRecipesFragment, ListOfRecipesFragment.OnFragmentInteractionListener, LoginActivity, MyRecipeActivity, NearbySupermarkets, NearbyTab, NearbyTab.OnFragmentInteractionListener, Recipe, RecipeAdapter, RecipeDetails, RecipeDetails.PlaceholderFragment, RecipeOverviewFragment, RecipePlaceholderFragment.OnFragmentInteractionListener, RecipeHistoryFragment, RecipePhotosFragment.OnFragmentInteractionListener, RecipePreparationFragment.
- All Classes Overview:** A table showing class names and descriptions.

Class Summary	
Class	Description
AddRecipeActivity	Task that allows the user to add a new recipe
AddRecipeActivityTest	
ForgotActivity	Allows users to send an email to reset their password
HomeActivity	Shows available recipes, main page upon login
HomeActivity.PlaceholderFragment	A placeholder fragment containing a simple view.
LoginActivity	Allows users to log into their account to use the app
MyRecipeActivity	Lists the recipes created by a user
RecipeDetails	Shows a more detailed view when a recipe is clicked
RecipeDetails.PlaceholderFragment	A placeholder fragment containing a simple view.
ShoppingList	Activity that displays all items added to the shopping list by the user
ShoppingList.PlaceholderFragment	A placeholder fragment containing a simple view.
SignUpActivity	Activity for user to sign into the application
SplashActivity	Splash screen showing recipick logo

com.example.softeng.recipick Activities

Class AddRecipeActivity

```
java.lang.Object
  android.content.Context
    android.content.ContextWrapper
      android.view.ContextThemeWrapper
        android.app.Activity
          android.support.v4.app.SupportActivity
            android.support.v4.app.FragmentActivity
              android.support.v7.app.AppCompatActivity
                com.example.softeng.recipick Activities.AddRecipeActivity
```

All Implemented Interfaces:

```
android.arch.lifecycle.LifecycleOwner, android.arch.lifecycle.ViewModelStoreOwner, android.content.ComponentCallbacks, android.content.ComponentCallbacks2,
android.support.v4.app.ActivityCompat.OnRequestPermissionsResultCallback, android.support.v4.app.ActivityCompat.RequestPermissionsCodeValidator,
android.support.v4.app.TaskStackBuilder.SupportParentable, android.support.v4.view.KeyEventDispatcher.Component, android.support.v7.app.ActionBarDrawerToggle.DelegateProvider,
android.support.v7.app.AppCompatCallback, android.view.KeyEvent.Callback, android.view.LayoutInflater.Factory, android.view.LayoutInflater.Factory2,
android.view.View.OnCreateContextMenuListener, android.view.Window.Callback
```

```
public class AddRecipeActivity
extends android.support.v7.app.AppCompatActivity
```

Task that allows the user to add a new recipe

Nested Class Summary

Nested classes/interfaces inherited from class android.support.v4.app.SupportActivity

```
android.support.v4.app.SupportActivity.ExtraData
```

Field Summary

Fields

Modifier and Type	Field and Description
java.util.List<java.lang.String>	ingredients the list of ingredient's name
java.util.List<java.lang.String>	measurements the list of ingredient's measurements
java.util.List<java.lang.String>	quantity the list of ingredient's quantity

Constructor Summary

Constructors

Constructor and Description

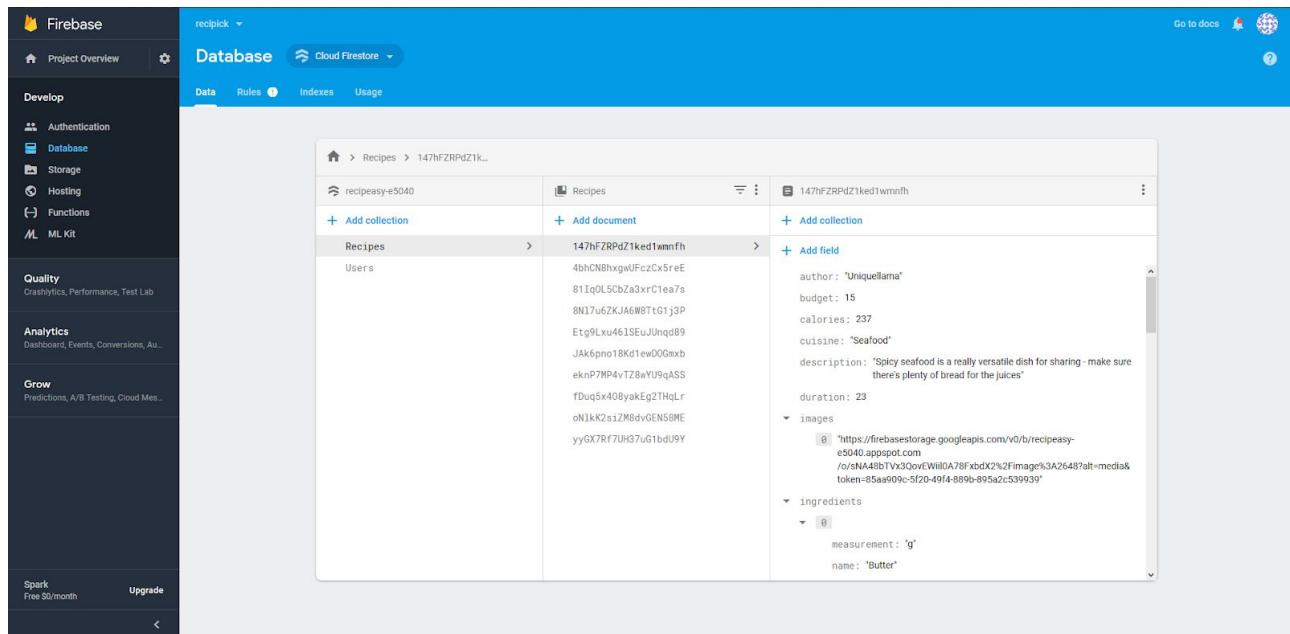
```
AddRecipeActivity()
```

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
java.lang.String	getFileName(android.net.Uri uri) Reference: https://www.youtube.com/watch?v=CXR8-9amqGo 'TVAC Studio'
void	loadUser() Used to get the display name of the currently logged in user
protected void	onActivityResult(int requestCode, int resultCode, android.content.Intent data) When the user selects the images The following happens.
protected void	onCreate(android.os.Bundle savedInstanceState)
protected void	onRestoreInstanceState(android.os.Bundle savedInstanceState) Restores the data and reset's the recycler view retains the recycler view upon rotation of the device
protected void	onSaveInstanceState(android.os.Bundle outState) When the activity enters the onsaveinstance the arraylist that is used to populate the recycler view is saved onsaveinstance retains the recycler view upon rotation of the device

9.5. Appendix E: Database



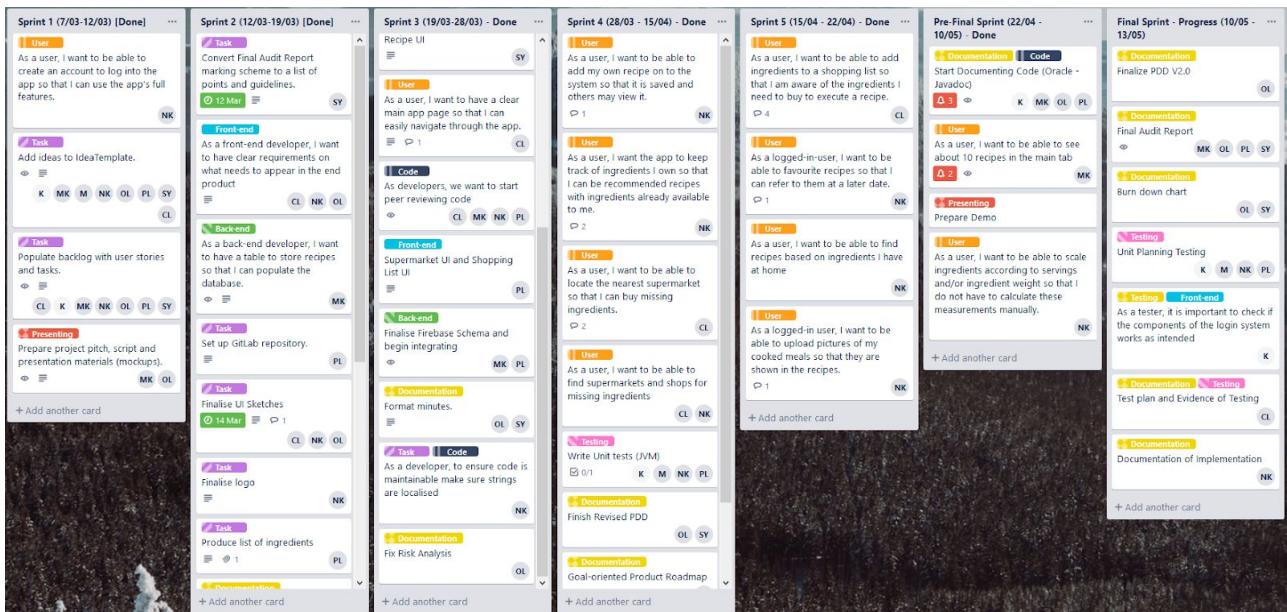
The screenshot shows the Firebase Database interface for a project named "recipick". The left sidebar contains navigation links for Authentication, Database (selected), Storage, Hosting, Functions, and ML Kit. The main area displays a "Database" view with "Cloud Firestore" selected. The "Data" tab is active, showing a hierarchical structure under the "Recipes" collection. A specific document, "147hFZRPdZ1ked1wmnfh", is expanded, revealing its fields: author ("Uniquellama"), budget (15), calories (237), cuisine ("Seafood"), description ("Spicy seafood is a really versatile dish for sharing - make sure there's plenty of bread for the juices"), duration (23), images (a URL to a Google Cloud Storage image), and ingredients (an array of three items). Each ingredient has fields: measurement ("g" for the first, "Tbsp" for the second, and "Cloves" for the third), name ("Butter", "Olive Oil", and "Finely Chopped Garlic" respectively), and quantity ("25", "2", and "2").

```
token=85aa909c-5f20-49f4-889b-895a2c539939"
▼ ingredients
  ▼ 0
    measurement: "g"
    name: "Butter"
    quantity: "25"
  ▼ 1
    measurement: "Tbsp"
    name: "Olive Oil"
    quantity: "2"
  ▼ 2
    measurement: "Cloves"
    name: "Finely Chopped Garlic"
    quantity: "2"
```

```
meals: "Dinner"
name: "Buttery Chili Prawns"
preparation: "Melt the butter and oil together in a frying pan. Add the garlic, chilli and paprika, then fry for 1-2 mins until starting to turn golden. Turn up the heat, throw in the prawns and fry for a few mins, stirring, until all the prawns turn pink. Take off the heat, season and stir in the lemon juice and parsley. Add some lemon slices to a finger bowl of warm water, grab a bowl for the shells, then dig straight in with your fingers and hunks of crusty bread."
servings: 2
share: false
▼ sharedImages
uid: "sNA48bTVx3QovEWii0A78FxbdX2"
visibility: false
```

```
display_name: "Max"
email: "max.krawiec0@gmail.com"
▼ favourites
▼ Etg9Lxu46lSEuJUnqd89
author: "Max"
budget: 5
calories: 120
cuisine: "Cake"
description: "This lovely recipe for your home made rice krispie squares is easy and tasty! Perfect for a children's birthday party or just a pre-dinner snack!"
duration: 15
▼ images
0 "https://firebasestorage.googleapis.com/v0/b/recipeeasy-e5040.appspot.com/o/sNA48bTVx3QovEWii0A78FxbdX2%2Fimage%3A2644?alt=m
```

9.6. Appendix F: Trello



9.7. Appendix G: Idea Template

Idea Description	Type	Data gathering Component	Server Side	Sensor 1	Sensor 2	Brief list of features
"Corner" : Location based voice social media	Business	Analyze trends and tags	Python or Node server and DB to store all posts	Microphone	Location	See below Record yourself and broadcast to people around you, listen to what people say, see the tags and trends of "speakers" around you
"Mood" : Upload video, get playlist	Potential Benefit to society and Business opportunity	Analyse RGB and fourier sound	Node server and Firebase	Camera	Microphone	Record your surroundings for 10 seconds and get a 'mood' playlist
GumTree-like market Application	Business	Analytics	Firebase	Camera	Location	See below Login System, each user will be able to create an account, users once logged in it would be directed to a Map which will show points of interests (point of interest are location of items being sold). 3-4 tabs in main menu, first tab is a map, second tab is a list of items in a close proximity, third tab is user details and products that they have uploaded, potential 4th tab is to add a messaging service to allow clients to message sellers. Perhaps use Geofencing so that only products being sold in a close proximity would show up rather than products across the whole map.
Accessibility route app	Business opportunity + potential benefit to society	User inputs for start and end stations	Firebase	Location	Camera	See below Determines the optimal step-free route for the Tube - takes delays into account and also offers alternative routes.
Media tracker	It is a business opportunity	User inputs for search, adding reviews	Firebase	Camera (barcode scanner)	Location to find the nearest store	See below Keep track of media such as books or TV series, with tags such as "currently watching" or "watched". Add reviews, find other users with similar taste and get recommendations.

Appendix H: Change Request

PLEASE FILL IN ALL FIELDS:	
Change Request #	1
Submitter Name	Sze Ying Lee
Date Submitted	20/03/2019
Date Required	25/03/2019
Change Request Description	<p>Move these requirements of the project charter to extended goals:</p> <ol style="list-style-type: none"> 1. Recipe filtering in terms of budget, cooking duration and diets. Users may exclude specific ingredients and any cooking appliances that the user may not have. 2. The ability to scale ingredients for a specific number of servings and to accommodate a reduced ingredient if the user does not have the required amount of an ingredient. 3. To discover the trending and most popular recipes of Recipick.
Justification	I believe that these requirements are non-essential for the app to achieve its objectives. I suggest that these requirements are moved to extended goals so that the team may focus on other areas of the project and only implement these features if time allows.
Priority (tick one)	<input type="checkbox"/> Low <input checked="" type="checkbox"/> Medium <input type="checkbox"/> High <input type="checkbox"/> Compulsory
FOR PROJECT MANAGER AND/OR PROJECT SPONSOR USE ONLY:	
Impacts	Streamlines the development process. Keeps focus on essential functionality for the engineering team.
Decision (tick one)	<input checked="" type="checkbox"/> Approved <input type="checkbox"/> Approved with amendments <input type="checkbox"/> Rejected
Amendments	//
Decision Date	25/03/2019 -- Accepted by Project Manager