## Project description – Fridge Chef

## Project Aim

The aim of this project is to provide a variety of recipes depending on the food stock of the user - by an application with an intuitive user interface and a social network element where users can also add and share their own recipes with others on the system. All the user needs to do is input their food stock via a virtual fridge on the system and Fridge Chef handles the rest by checking the ingredients in relation to the database recipes and displays them to the user.

## Target Audience

This project is not being developed by a particular person's request, rather it's intended to be an "off the shelf" application that any customer can utilize rather than a bespoke one. Examples of users include: students, people looking to prevent food waste, families wanting to save time and people who don't know what to cook.

## Essential Application Features

Essential 'must' - Unique Features

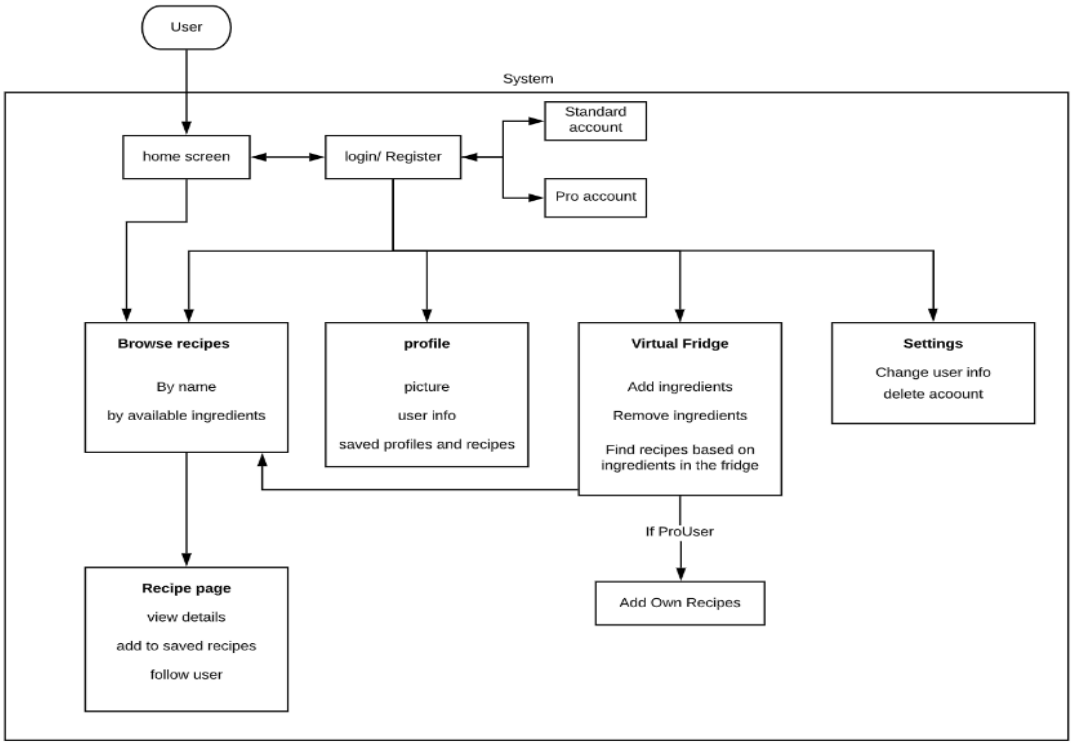
- Keep a virtual fridge that keeps track of items in the fridge.
- Deducts ingredients from 'virtual fridge' based on recipes you cook.
- Recipes based off items you have in the fridge – refined by:
    - recipes based on dietary restrictions.
    - recipes based off time of day.
    - recipes based on number of people needed to cook for.
    - Suggest recipes based on time they have available to cook.
- Users will be able to add their own recipes to the app and these will be viewable by everyone to build out the app's recipe base.
- Allow users to plan out recipes for the next week and build a shopping list of ingredients around this.
- Have a search bar that just lets users look ant recipes regardless of ingredients in fridge.

## Desirable Application Features

Potential 'could' features

- Suggest healthier alternatives to the items already in the fridge.
- Tell user's where to get certain items when they are running out based on location or cost.
- Introduce recipes restricted by budget for the meal.
- Give users warning of food going out of date – based on average lifespan of products and date when it was bought and suggest recipes from this.
- Have a family favourites section allowing users to quickly see their most like recipes.
- Have search results refined by how likely they are to like the recipe based on previous recipes they've eaten and what other people also liked.
- Add social functions to recipes allowing users to review recipes on the platform and allow users to refine by highest rated recipe.

## System Boundary Diagram




## User Views

In our program, the main users will be the 'customers' that download our application. We have split these users into users and pro-users, pro-users only differentiating by being able to add their own recipes.

There may also potentially be an admin/support user which moderates the site. The functional requirements being that they should be able to see all recipes in our database, they should be able to remove users (who spam post for example), they should be able to remove recipes (those that may violate guidelines). The non-functional requirements should be that they should be authenticated each time when logging in providing unique employee details such as their ID. The admins page should be coded in the Java programming language and the password should be between 6 to 20 alphanumeric characters including at least one higher-case letter, number and lowercase letter.

## Transaction Requirements

Data Entry

| Entering details of new user | User/ProUser |
|---|---|
| Adding new item to fridge | User/ProUser |
| Uploading new recipe details | ProUser |
| Adding item to shopping list | User/ProUser |
| Commenting on a recipe | User/ProUser |

Data Update/Deletion

| Update/Deleting a recipe | ProUser |
|---|---|
| Deleting/Editing a comment | User/ProUser |
| Deleting a user | Admin |
| Updating shopping list | User/ProUser |

Data Queries

| List details of recipes |
|---|
| List details of items in virtual fridge |
| List comments on a recipe |
| List items in shopping list |

## User Views – ProUser

Functional requirements:

- User should be able to log into or sign into the system clicking the designated button for such action which would lead to page where user should input the designated data for the authentication.

- Users should access settings through a designated button that leads to seperate page where they can change their username, password, email, profile picture, as well as set up privacy settings of their account.

- User should be able to delete their account in the settings page.

- User should be able to add or remove recipes from its list of favourite recipes on its user-page

- User should be able to update its virtual fridge by adding the ingredients to it.

- User shall use virtual fridge to find suitable recipes for the ingredients included in there.

- User should be able to access other users' user-pages.

- User should be able to view, comment on, and like recipes or professional chefs.

- System's browser should search for recipes based on the food's name, ingredients included in the dish, or the professional chef's name.

- Search through the system's browser should display chefs and recipes connected to key.

- AI admin shoud be able to update the Top recipes list on the homepage as well as to check the recipes for the sensitive context.

Non-functional Requirements:

- System should be administered by the AI which will fulfil certain tasks.

- System's browser should be accessible to the user

- Top recipes should be the list of top 10 recipes with the highest growing amounts of likes. However, each and every recipe should not be on this list for longer than 2 weeks. In such case, the recipe will be taken down from the list and then the next recipe will be placed at the bottom of the list.

- Authentication process while signing up to the system should be comprised of person's name, username, date of birth, password, email, agreement for the terms of service, as well as CAPTCHA authentication.

- Authentication process while logging into the system should be composed of only username/email and corresponding to it password.

- Password should be between 6 to 20 alphanumeric characters including at least one higher case letter, number and lowercase letter
- System should be programmed in the Java language while its database should be programmed in the MySQL.

- System should be developed for the Android Oreo version of operating system using android studio.

- There should be a user-page dependent on the user's type. Those types should be 'homecook' and 'professional chef'. Homecook's user-page should be comprised of recent actions, liked recipes/professional chefs, and virtual fridge, while professional chef's user-page should be composed of recipes created by them.

## Project Plan and Context

### Background Research

- Problem 1: Understanding how to create an android app using Java – In order to understand how to tackle this problem we have identified the solution of using Android Studio. Now we have the task of learning how this software works and will do this by following the tutorials on the android development site.[1]

- Problem 2: How to store data on recipes – for this issue we have found that we will use 'MySQL' to write data on recipes to the database and retrieve them again. Now for the next phase we will need to understand how this functions in relation to our program, so the design of each of our 'program modules' is appropriate. We will be reading a referenced tutorial online to understand the basics of this concept.[2]

### Data Required

Primarily, for the project we have decided that data will be dynamically generated by user suggested recipes. We will be using BBC recipes to provide a baseline for our data and we have chosen this due to the formatting being easy to incorporate into the database.[3]

### Design Stage

For the design method, the team has decided to use object-oriented design principles. For example, this will involve taking the crucial functions of the system taken from our user views and functional requirements and then splitting these into 'modules' which will be coded independently of one another but still keeping in mind each modules interactions with one another (based of the system-boundary diagram)

The design documentation will include the following: Use-case diagrams, object descriptions, interface design, pseudo-code for each module, interaction chart and module evaluation criteria.

### Implementation Stage

The hardware that will be used to implement the project will be a traditional PC and it will be developed on Windows using a Java IDE and also the Android Studio platform.

In terms of the testing, each module will be tested individually according to the criteria specified in the module evaluation section in our design document. For example, this will include testing for correct, boundary and extreme cases of data input.

## Risk Assessment

### Major challenges in carrying out the project

- Communication - it's important that communication is constant, so we end up with a high-quality project being completed on time.

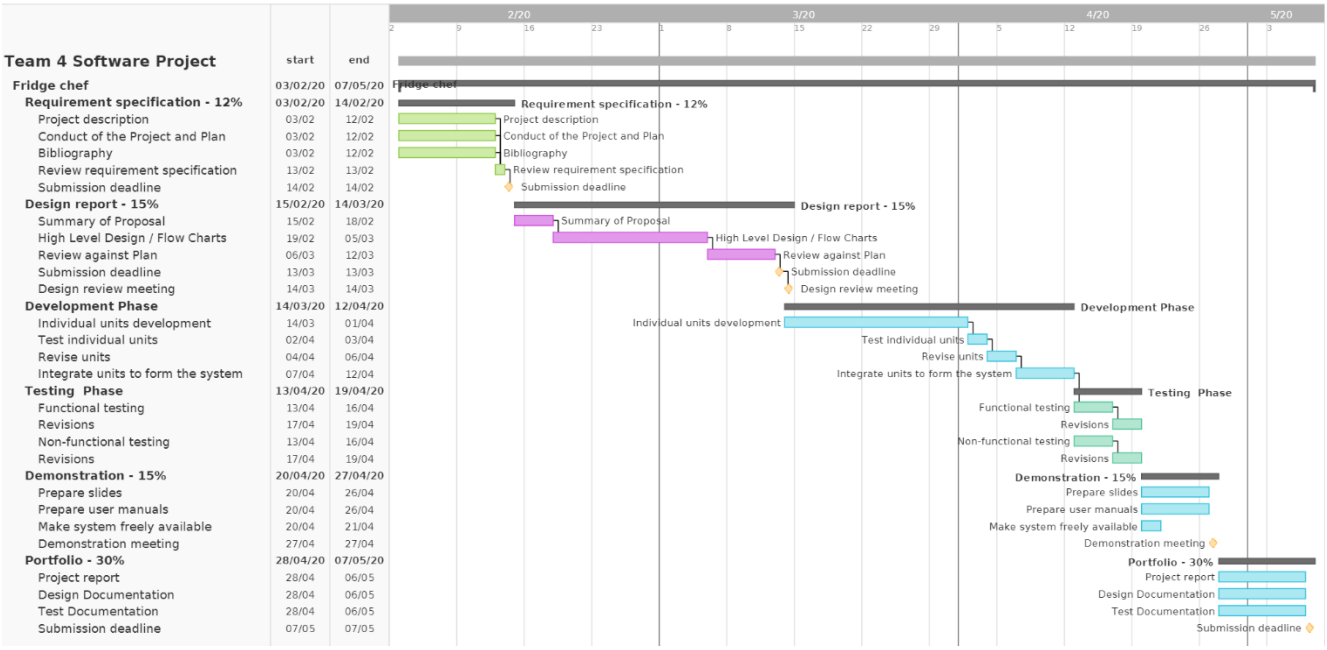
[1] https://developer.android.com/training/basics/firstapp
[2] https://www.vogella.com/tutorials/MySQLJava/article.html
[3] https://www.bbc.co.uk/food/recipes

- Consistency - this is vital because people will be tasked with going off and completing work independently and if their idea of the project differs from everyone else's their work won't fit in or well end up with a mixed match project.
- Version control – this is needed as different teams will be going off developing different parts of the system at different times. So, it's important that this is well documented so we can bring the project together harmoniously.

**New Skills that will be required**

- Android Studio – As a team we are not experienced with app development in android studio and will need to learn how to use it and the features it has.
- GitHub - needed manage the project properly and be able to combine everyone's work from different team within the project.
- Teamworking and communication - this will be paramount to this project for reasons as mentioned above, and although people will all different experience working in teams nobody will have experience on such a scale with such strict deadlines.

## Gantt Chart




## Bibliography


Android Developers. (2020). *Build your first app | Android Developers*. [online] Available at: https://developer.android.com/training/basics/firstapp [Accessed 13 Feb. 2020].

Lars Vogel (c) 2009, 2. (2020). *MySQL and Java JDBC - Tutorial*. [online] Vogella.com. Available at: https://www.vogella.com/tutorials/MySQLJava/article.html [Accessed 13 Feb. 2020].

Bbc.co.uk. (2020). *Recipes - BBC Food*. [online] Available at: https://www.bbc.co.uk/food/recipes [Accessed 13 Feb. 2020].