|  |
| --- |
| Nottingham Trent University |
| Requirements Specification Document |
| Advanced Analysis and Design | SOFT30121 |

|  |
| --- |
| Hassaan Naveed N0898071 David Vladimirovas N0899068 Sripriya Vattikula N0894013 Eisvinas Pranckaitis N0873702 Nikita Khera N0875673 |



### Disclaimer

This report consists of the requirements specification which was completed by the group of which I am a member. The retrospective document is my own work. All sources of information are referenced within this document. Should this statement prove to be untrue I recognise the right and duty of the Board of Examiners to take appropriate action in line with the university’s regulations on assessment.

ID No: N0898071, N0899068, N0894013, N0873702, N0875673

Contents

[Introduction 3](#_Toc88226597)

[Purpose 3](#_Toc88226598)

[User Characteristics 3](#_Toc88226599)

[Assumptions 4](#_Toc88226600)

[Scope and Constraints 5](#_Toc88226601)

[Glossary of Terms 6](#_Toc88226602)

[Overview 6](#_Toc88226603)

[Functional Requirements: 7](#_Toc88226604)

[Must Haves 7](#_Toc88226605)

[Should Haves 7](#_Toc88226606)

[Could Haves 7](#_Toc88226607)

[Wont Have 7](#_Toc88226608)

[Non Functional Requirements 8](#_Toc88226609)

[Usability Requirements 8](#_Toc88226610)

[Reliability Requirements 8](#_Toc88226611)

[Performance Requirements 8](#_Toc88226612)

[Interfaces 9](#_Toc88226613)

[User interfaces 9](#_Toc88226614)

[Hardware Interfaces 12](#_Toc88226615)

[Software Interfaces 13](#_Toc88226616)

[Use Case Modelling 14](#_Toc88226617)

[Use Cases 14](#_Toc88226618)

[Use Case Diagrams 14](#_Toc88226619)

[Project Plan 18](#_Toc88226620)

[Methodology 18](#_Toc88226621)

[Risk Assessment 22](#_Toc88226622)

[Product Backlog 23](#_Toc88226623)

[Functional Point Analysis 24](#_Toc88226624)

[Gantt Chart 24](#_Toc88226625)

[Sprint Meeting Notes 24](#_Toc88226626)

[References 25](#_Toc88226627)

[Appendix 26](#_Toc88226628)

[Appendix (a) - Constraints 26](#_Toc88226629)

[Appendix (b) – Use and Misuse Cases 26](#_Toc88226630)

[Appendix (c) – Functional Point Analysis 31](#_Toc88226631)

[Appendix(d) – Gantt Chart 34](#_Toc88226632)

[Appendix(e) – Sprint Meeting Notes 35](#_Toc88226633)

# Introduction

## Purpose

The purpose of this project is to provide the Future Fridges company a piece of software that enables them to use their specialised fridge for restaurants as a Smart Fridge. This application will keep track fo any incoming and outgoing food, its expiry date and quantity, make automatic orders and send alerts when stock is low, and have a back door that allows the fridge to be stocked without any interruption of the kitchen staff.

Deliveries will be able to be made without interruption of the kitchen staff, and this will ensure that stock levels are always kept up to date and no expired food will be left in the fridge. The application will also make use of accounts for the different levels of chefs. The head chef will be able to set quantity levels and have more permissions than regular kitchen staff, and delivery people will have an account and require a unique code to access the fridge for security.

## User Characteristics

The primary users who will interact with the application are the Head Chef, the Chefs, the Delivery Personnel, Health and Safety Officers, and System Administrators. Below show the different users and characteristics.

The primary user characteristics have been gained from an interview with a representative from Future Fridges, Brian, who answered a series of questions

### System Administrator

The system administrator will be able to oversee the application and all of the accounts on the applications, including logs of when the fridge has been accessed and by which account. They will be able to assign the role of head chef, and manage any permissions, as well as deleting any accounts from the system.

### Head Chef

The head chef will be in charge of all of the Chef accounts, as well the the managing of stock in the fridge. They will be able to assign permissions to each Chef account on the application, and set the stock threshold for the fridge’s automatic reorders. The head chef will also be able to manually order items, as well as being able to access a log of the accounts that have accessed the fridge and the time of access. The head chef is also able to access the fridge, and insert or remove any items.

### Chef

Chefs will be able to access the fridge, and be able to insert and remove items, as well as viewing stock levels. They may also They will not have access to the management of accounts or deliveries, however they are able to request the deletion of accounts from the system administrators.

### Delivery Personnel

Delivery personnel will only be able to access the back door of the fridge, which they will need to input their own Personal Identification Number for. They have the ability to insert items, but not to remove items. They will also not be able to access any account or delivery management features.

### Health and Safety Officers

Health and safety officers will be able to access the automatically generated health and safety reports from the application. They will not be able to access the fridge or use any management features.

## Assumptions

|  |  |  |
| --- | --- | --- |
| Assumption | Description | Ramifications |
| Users have an android mobile phone or tablet to access the application | The solution will be based in an android mobile application. This is the most common type of mobile device so it is likely that most users will be able to access the application. | Users without a device that is able to load the application wont be able to interact with the solution |
| User’s device has internet capability | Devices that users use to access the solution must have capabilities to connect to the internet in order to access the cloud based database. | If there is no internet access available, then users will not be able to access the application. |
| User’s have a registered account to log into | Users will need to know their email and password to access their accounts | Users who forget their email or password will need to ask a member fo staff for assistance |
| User’s require minimal technical experience | The application will be very simple to use, but knowledge in forms and UI will be useful in navigating the application. | Users who prefer to use paper-based solutions may struggle with adjusting to new technology |
| Database uptime | The web application requires constant connection to the database to access stored data | Without access to the database the user is not able to login or access the fridge system |
| Individuals use the mobile application | Members of staff and delivery personnel will make use of the application once it has been implemented | All stock levels of items within the fridge need to be accurate and up to date, and the system must be reliable |
| User’s based on site | Users need to be on-site in order to access the application for security | Users attempting to access the application while not on-site will be blocked from logging in |
| Handles user requests | The application will need to handle a minimum of 10 requests at once | If the application is unable to handle requests simultaneously, it may cause issues and slow down staff |
| Delivery people have code to access fridge | To access the fridge, delivery people will need a personalised code in order to open the fridge door, for security measures | The back door will not be able to be accessed without a code |
| Deliveries made will be accurate | Any deliveries made to the fridge will have the correct items made in the purchase order. | The application will automatically check for discrepancies with the inserted items and notify the head chef. |

## Scope and Constraints

There will be five main types of users, System Administrators, Head Chefs, Chefs, Delivery Personnel, and Health and Safety Officers, that will use the application. They will all have different access to the system, as described by a hierarchy below.

The System Administrator is able to create an account for the application. From the administrator account, they will be able to create and delete accounts of any type. They will also be able to adjust the permissions of individual accounts on the system, as well as reset the login details of any existing accounts. The System Administrator account is intended to be the top level that manages all other accounts, and the functioning of the application. They will be able to view the health and safety report. They will not be able to access the fridge, and therefore cannot insert or remove items from the fridge itself, or adjust restocking thresholds.

The Head Chef is the next level down. They will be not be able to create or delete new accounts, or reset any account login credentials. The head chef is able to manage the permissions of any accounts lower in the hierarchy, and also manually adjust the restocking thresholds for any of the items, in addition to manually adding items to the purchase order. Alerts will be sent to the Head Chef when an item falls below the stock threshold. The head chef is also able to access the fridge, and add or remove items to it. The head chef may also access the health and safety report that is generated by the fridge.

Next in the hierarchy, the Chef falls below the Head Chef. The chef is only able to access the fridge, adding or removing items. They are unable to create or delete any accounts, or manage any account permissions. They are unable to manually adjust restocking thresholds or purchase orders, or access the health and safety report.

Below the Chefs, the delivery people may only access the fridge from the rear door. They are only able to open it once they have entered a unique PIN, and they may only insert items into the fridge.

Finally, the Health and Safety Officers are only able to access the health and safety report that is generated by the application. They are unable to access any other features.

From this overview of the hierarchical structure of the system, the primary scope of the project can be identified. Various different accounts should exist, with different basic permissions. These permissions should be alterable on a case-by-case basis, and accounts should be delectable. The fridge should be accessible by certain accounts, and items may be inserted or removed. The fridge should require a PIN to be accessed through the back door. The system should automatically make a purchase order at the beginning of each week in order to restock the fridge, as well as generating a health and safety report. The threshold for an item requiring to be restocked is alterable by the head chef, and the head chef is also able to manually adjust the purchase order before it is made. The system will be a mobile application that is able to run on android powered mobile devices.

There are some features that could be implemented in addition to the outlined scope of the project. For example, there is no physical hardware for the fridge, and so the application will not be adapted to run on any bespoke smart fridge operating system. Additionally, it has been gathered from stakeholder interviews that the company is unconcerned with the expiry dates of items within the fridge, therefor a dating system does not need to be implemented. Additionally, a barcode scanner could be utilised for delivery people for adding items to the fridge, but due to time constraints this would be unreasonable to prioritise over other, higher priority requirements. A final out of scope feature is the fully automated tracking of items inserted into and removed from the fridge, as the hardware has no internal sensors for measuring this. A list of constraints is found in Appendix(a).

## Glossary of Terms

|  |  |
| --- | --- |
| Term | Definition |
| Java | A programming language that will be used to develop the application |
| Firebase | Cloud based database for storing data |
| JSON | Javascript Object Notation – an open standard file format for storing structured data within the database |
| Android | An open source mobile operating system which the application will run on |
| Mobile Application | The mobile application will run on a mobile device and be used to access the fridge, and insert or remove items |
| Solution | The system that concerns the fridge |
| User | An individual utilizing the application |
| Administrator | Top layer of staff member who is able to access the management of all accounts |
| User permissions | The access a user account has to certain features of the solution |
| Purchase order | A document depicting order details |
| PIN | Personal identification number – a code that will be used by delivery personnel to access the fridge |
| OS | Operating System – The platform the application will run on |
| Item | A singular food product within the fridge |

## Overview

The following documentation will serve as the guideline for the project, planning out the individual stages the project will go through, such as requirements gathering and design. This will clearly lay out how the project will be created in order for the requirements and plans to be followed through thoroughly, and without any major roadblocks. The requirements will be organised with the MosCow prioritisation method, therefore it will be clear which tasks to prioritise.

The use and misuse cases will also be outlined, following the UmL standard, consisting of actors, systems, modules, and relationships, to show clear diagrams of the inner workings of the system, creating a template to work from.

The various interfaces will be designed and outlined, in order to create a visual guide when building the application, and a suitable timeframe will be planned, setting deadlines to ensure the project is complete within a required amount of time, laying out each of the major milestones and tasks to be completed.

# Functional Requirements:

## Must Haves

* The user must be able to register their first name, surname, work title (i.e., chef, delivery driver etc.) and password
* The system will only allow access to the fridge after a user had logged in.
* The system will log out whenever a user has closed the fridge
* The system must assign authority to the registered users in correspondence to their work title
* Users under the “delivery person” work title have access to the back door of the fridge and can only insert items into the fridge
* Users under the “chef” work title have access to any door of the fridge and can insert or remove from the fridge
* The system must have a rule for when a door Is open on either side, the other side cannot be opened until the opened door is closed.
* The system must store all food items and registered users in separate databases
* The system database for food items must have a capacity limit of 440 different types of food items (rows) so that the food items to not exceed the fridge capacity
* The system database for food items must contain the attributes, name, quantity, and expiry dates
* The system must have a manually adjustable restocking threshold for food items,
* The system must prompt the “head chef” users computer device when a food item falls below the restocking threshold.
* The system must have an account/user that has the admin with the authority to change user authority.
* The system will set an order to the appropriate supplier every Monday for food items that are low or are zero in quantity.
* The system will record the number of items inserted into the fridge from the user “delivery person” and will compare that number to the checksum that is given before the insertion of food items
* Report of all the food inside the fridge will be sent to the health and safety officers.

## Should Haves

* The system will send a report or list of items that have been inserted into the fridge to the user that performed the inserts.
* The system should have a UI that is easy to use and understand for all user types.
* System implements a hash function to user username and passwords to store and secure the information.

## Could Haves

* The system prompts the “head chef” users mobile app when food items are 3 days before its expiry date.
* System be accessible from a website

## Won’t Have

* Finance system to ensure that the chef has enough funds to order the items
* There will not be support for multiple language options

# Non Functional Requirements

## Usability Requirements

* The application must run on most android mobile operating systems
* The application should be responsive and accommodate for a variety of screen sizes
* The application should have a consistent colour scheme to avoid eye strain or “ugliness”
* Data should be stored for up to 3 years, at which point it becomes archived
* The system must ensure that the correct items have been inserted when deliveries are made
* The system should output dates in a standardised format of DD-MM-YYYY
* Security measures must be taken to prevent unauthorized access, such as a login system
* The design of the application should consider colour blindness, dyslexia, and global perceptions towards colour meaning

## Reliability Requirements

* The storing of data must comply with GDPR and DPA standards
* Any data such as item name and expiry date must never change without explicit user input
* Database should be efficiently organized to prevent slowdowns, freezing, or crashes
* The solution should be free of any major bugs that severely impair the use of the application
* The data should be encrypted using SHA256 to prevent data leaks
* The back door of the fridge must be secure with a PIN to access

## Performance Requirements

* The application should handle input from at least 20 users at the same time without compromising integrity
* The application should be available 24 hours a day
* The application should have a short response time of no greater than 1 second
* The application should not freeze, stutter, or crash
* The database should allow for fast querying, with a response time of no greater than 5 seconds

# Interfaces

## User interfaces

We have thoroughly discussed and put into the practices, while designing our mobile application, the eight golden principles of User Interface Design by Ben Schneiderman, in his book ‘Designing the User Interface: Strategies for Effective Human-Computer Interaction. The eight golden rules to an ideal user interface design which are widely practiced, including many successful companies like Google, Apple and Microsoft (Wong, 2018) are as below:

1. **Strive for consistency** by utilizing familiar icons, colors, menu hierarchy, call-to-actions, and when designing similar situations and sequence of actions.
2. **Enable frequent users to use shortcuts.** With increased use comes the demand for quicker methods of completing tasks.
3. **Offer informative feedback.** The user should know where they are at and what is going on at all times. For every action there should be appropriate, human-readable feedback within a reasonable amount of time.
4. **Design dialogue to yield closure.** Don’t keep your users guessing. Tell them what their action has led them to.
5. **Offer simple error handling.** No one likes to be told they’re wrong, especially your users. Systems should be designed to be as fool-proof as possible, but when unavoidable errors occur, ensure users are provided with simple, intuitive step-by-step instructions to solve the problem as quickly and painlessly as possible.
6. **Permit easy reversal of actions.** Designers should aim to offer users obvious ways to reverse their actions.
7. **Support internal locus of control.** Allow your users to be the initiators of actions.
8. **Reduce short-term memory load.**  interfaces should be as simple as possible with proper information hierarchy, and choosing recognition over recall.

(Schneiderman, 1992)

Basing on the principles above, we have designed the user-interface below using Adobe XD. The user interface is designed to be simple, and consistent by using the same structure, font and color throughout the application. The application is developed putting in mind that our users find the overall navigation easy, and hence giving them the exit points at every point. The users are the initiators of the application, and find apt information and relevant content at every step. We have decided to develop a mobile application as our solution to the problem statement. Since apps are just a tap away, and portable we found it to be the best solution for our users.

The following designs are not final, but will serve as the starting point for our team to develop the app and will be re-designed subsequently according to the needs and for the sake of betterment.

|  |  |  |
| --- | --- | --- |
| This is the opening page of the android app, asking the user to either register or login. | A first time user will have to register their details on this page to login to this app from next time. | Registered users can login to their account through this page. They can also reset the password on this page by clicking on the forgot password link. |
| This will be the home page of the head chef. They can view notifications, the stock level, and insert or remove the food items by clicking on relevant buttons. The head chef can control who's allowed to access the fridge. | The home page of the delivery person after they log into their account. They can insert food in the fridge by clicking on the 'insert food' button, and also view a record of the items replenished by clicking on the 'items replenished' button. | This is notifications page in the head chefs account where they will receive the notifications of the items when they are low in stock. Date and time of the notification received is also displayed. |
| The insert or remove items page is accessed by clicking on the relevant button. The users can simply enter the food item, quantity and units and choose to either insert or remove them. | The available food items along with the quantity and expiry date are shown in this page. Users can also search for a specific item by entering the name of the food item and also filtering it by date, alphabetical order, and the quantity. | This page is used to configure or change which users have access to the fridge and whether they are allowed to insert and remove items. |
| This page is used to configure or change which users have access to the fridge by clicking on allow or deny for that user. | This page is used to configure or change which users have access to insert or remove food items in the fridge. | This page will be accessed by the delivery person to insert the food items. |
| On the confirmation page the delivery person can determine whether the right items have been inserted or not. Delivery man has the option to edit the inserted items above before confirming them. | This page enables the delivery person to see the record of what items they have replenished in the fridge. They also has the option to search for a specific food item in the replenished items by searching the name and filtering it by date or alphabetical order. |

## Hardware Interfaces

Diagram

Description automatically generated

The following section explains briefly about the components shown in the diagram above.

**Hardware Layer:**​ The hardware layer is the most important component of the system, where our users will interact with the solution (mobile application). In our case, where we are building a mobile app, the hardware devices that can be used are mobile and tablet.

**UI Design layer:** The UI design layer is the user-interface of the mobile/tablet that the users interact with. The UI design layer is the medium of interaction between the users, and the stored data.

**Business logic layer:** Business logic layer are the main systems within the solution that handle all client requests and distribute data to the users. They comprise all the logic needed for analyzing a problem.

**Data layer:** The data layer consists of all the data required for the user to run the application smoothly. The data will be hosted on cloud server to avoid any kind of security issues that arise from mismanagement of hosting data on external servers. According to Gartner Estimates, public cloud service workloads will suffer at least 60% fewer security incidents than those in traditional data centers (Microsoft, 2020).

**External system:** In order for the business to continue running, external users need to have access to relevant information. The data transferred to the supplier system contains the information regarding items that have run out or will run out. The food database will be sent to the health system, and the purchase order database will be sent to the finance system.

## Software Interfaces

The solution will be a mobile application that can only be used on mobile phones with an internet connection. This mobile app can be downloaded from the play store or the App store by the management staff. Our team decided to build a mobile application for Future Fridges as mobile phones are highly portable and convenient, making them the most desirable means of business today. Mobile applications are just a tap away, so users won't have to struggle with a laptop or carry it around all the time. We will host the database on a cloud server, which is convenient and cost-effective, and eliminates the need for external servers.

# Use Case Modelling

## Use Cases

Actors

* Chef
* Head chef
* Delivery personnel
* Admin
* Malicious person
* Health and safety department
* Database
* System/application

### Use Case List

1. Register - allows user to create a new account
2. Login - allows user to login to an account that is already in the system
3. Insert item - allows user to insert item into the fridge and keeps a log of who inserted what and how much
4. Remove item - allows user to remove item from the fridge and keeps a log of who taken out what and how much
5. 3 day expiration warning - the system send a notification to the head chef if an item in the fridge is going to expire in 3 days
6. Product running low - the system sends a notification to the head chef if an item in the fridge is running out/low
7. Health and safety - sends a report to the health and safety department if an item has expired in the fridge
8. Order supplies - auto reorder supplies every monday

### Misuse Case List

1. Unauthorized registration - a malicious user tries to register to the system
2. Unauthorized login - a malicious user tries to get access to the system using partially known information about an account in the system
3. Delivery details - the system checks items inserted by delivery personnel to make sure that all items from the order have been inserted

## Use Case Diagrams

### Use Case Diagrams

These use case diagrams describe teh relationships between all of the use and misuse cases, highlighted above. Detailed descriptions of each use and misuse case canb be found in Appendix(b).

1. Registration - use case 1 and misuse case 1
2. Login - use case 2 and misuse case 2
3. Insert/Remove item - use case 3, use case 4 and misuse case 3
4. System management - partially use case 1 and use case 2
5. Notifications - use case 5, use case 6, use case 7
6. Reorder - use case 8

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generated

# Project Plan

## Methodology

For most projects and software development, the plan and methodology are a good indicator of how successful it may become. This Is by no means the only indicator, as there are many other indicators, externally and internally of the development. Due to the constraints of the project and the occupation of the team members, the only relevant indicators are the methodology and the plan, as such, only these will be discussed.

Firstly, a methodology is a process or series of processes that are used in software development. The processes are typically divided into segments to allow more tasks to be worked on in parallel. There are many different methodologies that exist to facilitate the needs of specific software projects, however, we will only be focusing on the most used ones, such as waterfall and agile methodologies (Williams, 2010).

### Methodology 1 – Waterfall development

The general structure for the waterfall method is the following:

* Requirements
* Analysis
* Design
* Coding/implementation
* Testing
* Deployment
* Maintenance

This methodology is typically used when the requirements are understood well and there is a low chance of any changes to happen during the development. This means that it is easy to use and communicate since it is of sequential structure and predictable. The waterfall methodology does not provide much flexibility is intentionally designed for projects with clear and static requirements (Cheatham & Crenshaw, 1991).

There are variations of the water fall method, V-shaped and parallel waterfall methods, that are used to as industry specific or supplement the needs of a project requirement.

V-shaped waterfall models are like the original as they are both sequential, however, during the V-shaped water fall model, during the development of the code, there will be simultaneous testing to identify bugs and defects quicker.

Parallel waterfall models are different from the original as during any part of the project, there could be other parts being worked on simultaneously. This would reduce costs and save time. The model eventually will merge and join the final project at the end of the development.

### Methodology 2 – Rapid application development (RAD)

RAD is a methodology designed to target the projects that requires constant change and new requirements. The RAD methodology is becoming very popular recently due to the constant change in technology such as smart phones and computers etc. The structure of RAD methodologies is:

* Analysis and design
* Prototype
  + Develop
  + Demonstrate (feedback)
  + Refine
* Test
* Deploy

There are several RAD models that can be implemented, the iterative model, system prototyping and throwaway prototyping model. Each model will have its own unique purpose with cons and pros (Beynon-Davies & Holmes, 2002).

Iterative model is a model that follows a mini waterfall process where then after one prototype, after every other iterative, user feedback is used for improvement.

System prototyping is a model that will perform analysis, design, and implementation concurrently to develop a simple version. For each iteration, more features and reformation of each process is done.

Throwaway prototyping usually is created with no intention of working but rather the means to show the user the issues that are under consideration. The chosen prototype at the end is still thrown away.

The overall purpose of RAD is the speed and quality of the software development such that user feedback is highly valued and are interacted with.

### Methodology 3 – Agile development

The general structure for Agile development:

* Requirements
* Design
* Development
* Testing
* Deployment
* Review

In comparison to RAD, Agile methodologies focus on meeting deadlines and having a sustained and continuous progression over longer periods of time (Dyba & Dingsoyr, 2009).

There are different frameworks of agile development such as scrum and XP that allow for more targeted structure for particular projects.

Scrum is a framework with an application of the agile philosophy such that the development of the project derives from the increments of each process. This makes scrum, a faster and more flexible framework that delivers to the customer throughout the development. Scrum utilizes “sprints” that are iterations of a process that are typically a week to a month long.

Extreme Programming (XP) Is like scrum but has subtle differences. XP focuses on the quality of the software for projects with smaller team numbers as well as adapting to changing requirements. The iterations in XP are typically one or two weeks long.

### Methodology of choice

The methodology of choice is the agile methodology because of the framework that will be employed for this project, the scrum framework. The project and team members are a great candidate for the scrum framework because we are a small team with a tight schedule and has a good understanding of what technology we will be using.

The general incentives will be that we framework will allow the team to focus on getting the features developed in order of priority whilst maintaining high quality of work. This is thanks to the roles, that will be discussed shortly, being split appropriately, allowing individuals to work on parts of the project simultaneously while having guidance and quality assurances. Furthermore, the development of features will also be tested simultaneously, which will allow for quick reviews and feedback on the feature. This will save time and ensure deadlines can be met.

Due to the lack of communication to the stakeholders, the feedback will be limited, therefore, effecting the development and quality of the features. However, this can be overcome with clear and comprehensive planning for each feature. Additionally, the team members may be constrained by external factors such as health issues from the ongoing pandemic, and as such, communication may lack, causing some setbacks and “sprints” not completed on time.

### Roles

Following the scrum framework, the roles for the project are the following: Scrum master, product owner and developer. However, for this project we further divided the roles to have a specific, but not limited, role for everyone. These roles are Scrum master, Quality assurance, UI designer, Engineer, test engineer and product specialist. The combination of all these roles will allow for all areas of the scrum framework to work effectively and as intended.

**Scrum master:** The expert that will ensure the project is following the scrum framework, while providing guidance and assisting during the development.

**Quality assurance:** An individual who will ensure that the requirements are being met or on track before and after testing as well as during the development. The QA will proactively assist with issues and bugs during the development cycle.

**UI designer:** As the name implies, the individual that will be designing and adapting throughout the project, including with user feedback

**Engineer:** The engineer will be focused on developing the features that of highest priority. The engineer will be communicating with the development team and other staff accordingly.

**Test engineer:** The test engineer will be working with the engineer and development team to make the appropriate tests and simultaneously test the features produced by the engineer.

**Product specialist:** The product specialist will be focusing on the project vision and breaking the project down further. Additionally, obtained and goals are supervised by the product specialist as well as the features being ordered by priority are also done by the product specialist.

### Role Assignment

|  |  |
| --- | --- |
| Role: Scrum master | Member: Hassaan |
| Hassaan has had previous experience in being a scrum master during other projects and such, we have decided to give the most experienced on the role of a scrum master. They are great at coaching and leading, making them a perfect fit. | |

|  |  |
| --- | --- |
| Role: Quality Assurance | Member: David |
| Throughout his experiences, David has always has an eye for detail and always ensured to have everything organized to a set standard. This role is a great fit for him as he is confident in ensuring that all the products and processes are up to a standard. With the standards that are in place for this project, the ISO standard, we can be confident that he will uphold this standard from beginning to end of the project life cycle. | |

|  |  |
| --- | --- |
| Role: UI designer | Member: Eisvinas |
| Eisvinas has been designing websites and mobile applications in his own time as a hobby since he is very interested in design. He has a great attitude to working in a team and communicates effectively while also maintaining humble. Additionally, he has been keeping up with the design standards which means he has better understanding of the standards than most of the other team members. | |

|  |  |
| --- | --- |
| Role: Engineer | Member: Hassaan |
| Hassaan has proven that they are very competent and efficient when it comes to developing and programming. Combining their experience in developing and coaching skills, being the primary engineer will make them the most appropriate for the role as he can help other engineers and lead from the core of the development. | |

|  |  |
| --- | --- |
| Role: Test engineer | Member: Nikita |
| Nikita has shown keen interest in testing the features to ensure that they are perfect without any bugs. She is naturally very organized and is familiar with the concept of testing due to her experience in the past projects she has been part of. Also, she has great communication skills making it easy for her to communicate the problems and bugs that may occur during the testing period | |

|  |  |
| --- | --- |
| Role: Product specialist | Member: Sripriya |
| Sririya has been an advocate of giving the customers what they deserve and as such, she has shown interest in taking this role. She has great understanding of the requirements and has been in contact with the stakeholders, so she is much more informed about the requirements than the other member, as such, making the most appropriate candidate for the product specialist role. | |

## Risk Assessment

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Probability | Impact | Mitigation |
| Unclear or unrealistic requirements or scope | 2 | 5 | Ensure that all requirements are reviewed by all members of the team and are thoroughly discussed prior to being confirmed. In addition to this, actively seek out the support from experts to ensure that the project scope is attainable within the allocated development window |
| Insufficient background knowledge and research | 3 | 4 | Carry out an intensive research process prior to beginning the development process to ensure that all team members are well informed |
| Security breach due to passwords being compromised | 2 | 5 | Add password encryption and (if feasible) two-factor authentication. |
| Team member falls ill during project | 3 | 4 | Disperse their work between the remaining team members |
| Loss of data due to technical failure | 2 | 4 | Make regular backups to GitHub and other cloud storage options used by the team. |
| Tasks go over allotted time | 2 | 3 | Give buffer for overrun time at end of project - work to a week before actual deadline |
| Team member overwrites existing file by accident | 3 | 1 | Regularly use version control software (i.e, Git - GitHub) so that the file contents can be easily reverted to an older version |
| Users struggle to use the application due to unintuitive UI | 2 | 2 | Ensure that during the testing stages user feedback is gathered with regards to the usability of the UI |
| Major bug found during testing | 2 | 2 | Agile development allows for regular testing to prevent large scale bugs at the end of the project. |
| Team member struggles to engage or is not actively communicating with rest of group | 3 | 4 | The team must have frequent check-ins to ensure how all team members are handling their workload and if anyone requires assistance with managing their own tasks they are encouraged to seek help from the rest of the team. |
| Team experiences issues with computational resources | 1 | 3 | Contact the team's assigned tutor for support in gaining access to university resources. |
| Member experiences issue with handling work load | 3 | 4 | The team reviews the assigned task in order to break it down amongst other members to help support the struggling member |
| Team conflict occurs due to differing opinions | 3 | 5 | The team makes use of the quality vote to make the final decision to resolve the conflict |
| Initial time estimates are innaccurate | 3 | 4 | Constantly review time estimates and make changes if required |

## Product Backlog

Playing Poker is an estimation technique to generate story points for the product backlog. This is done by using a scale of numbers, such as doubling for each step, to represent the required complexity, time, and resources required for a user story. For each story, each member of the team assigns one of the complexities to the point, and a discussion takes place until a consensus is reached on the complexity of the task.

|  |  |  |  |
| --- | --- | --- | --- |
| Points | Complexity | Time | Resources |
| 1 | Very Low | 1-2 hours | 1 |
| 2 | Low | 3-5 hours | 1-2 |
| 4 | Below Average | 6-11 hours | 1-3 |
| 8 | Average | 12-23 hours | 2-3 |
| 16 | Above Average | 1-4 days | 2-4 |
| 32 | High | 1-2 weeks | 4-5 |
| 64 | Very High | 2-3 weeks | 5 |

The product backlog is then created using the user stories and the story points that have been assigned from the above table. They are given a MoSCoW priority rating and the sprint in which they will be addressed is decided.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | User Story | Priority | Story Points | Sprint |
| 1 | As a stakeholder, I want to be able to know what solution is being proposed. | Must | 8 | 1 |
| 2 | As a stakeholder, I want to understand what the requirements of the solution are. | Must | 4 | 1 |
| 3 | As a stakeholder, I want to know the interfaces of the new solution | Must | 8 | 2 |
| 4 | As a stakeholder, I want to know the different use cases for the solution. | Must | 8 | 2 |
| 5 | As a stakeholder, I want to understand how the project is going to be planned out | Must | 4 | 3 |
| 6 | As a user, I want to be able to register an account for the app. | Must | 16 | 4 |
| 7 | As a user, I want to be able to log into the app. | Must | 4 | 4 |
| 8 | As a user, I want to be able to add items into the fridge | Must | 2 | 5 |
| 9 | As a user, I want to be able to remove items from the fridge | Must | 2 | 5 |
| 10 | As a user, I want to see the details of items within the fridge, such as name, quantity, expiry date | Must | 1 | 5 |
| 11 | As a user, I want to set restocking thresholds for items | Must | 4 | 6 |
| 12 | As a user, I want automatic purchase orders for items low on stock | Must | 8 | 8 |
| 13 | As a user, I want to be able to view weekly health and safety reports | Must | 4 | 6 |
| 14 | As an admin, I want to be able to create and delete accounts | Must | 8 | 4 |
| 15 | As an admin, I want to be able to modify account permissions | Must | 16 | 5 |
| 16 | As a user, I want access to the solution 24 hours a day | Should | 4 | 9 |
| 17 | As an admin, I want security for the back door of the fridge | Must | 2 | 6 |
| 18 | As an admin, I want a log of who has inserted/removed items | Must | 8 | 7 |
| 19 | As a user, I want an easy to use display for the application | Should | 32 | 8 |
| 20 | As a user, I want notifications of items that are low in quantity | Must | 8 | 6 |

## Functional Point Analysis

The functional point analysis is a number calculated to estimate the possible complexity of the project based on the functional requirements and product backlog (Nasir, 2006). The analysis can be found in appendix(c).

## Gantt Chart

This Gantt Chart highlights the timescale of the project. The key milestones are identified, and each individual sprint is outlined. The tasks that will occur within each sprint are broken down, so there is a preliminary plan of what will be completed within those weeks. The dependencies and critical path of the project are highlighted. This will be vital in ensuring the project is on track, and the chart will be referenced and updated which week to ensure deadlines are being consistently met. The Gantt chart can be found in Appendix(d)

## Sprint Meeting Notes

The notes detailing the discussion points of each sprint can be found in appendix(e).

# References

Albrecht, A. J., 1979. *Measuring Application Development Productivity,* s.l.: IBM.

Beynon-Davies, P. & Holmes, S., 2002. Design breakdowns, scenarios and rapid application development. *Information and Software Technology,* 44(10), pp. 579-592.

Cheatham, T. J. & Crenshaw, J. H., 1991. *Object-oriented vs. waterfall software development.* s.l., s.n.

Dyba, T. & Dingsoyr, T., 2009. What Do We Know about Agile Software Development?. *IEEE Software,* 26(5), pp. 6-9.

Microsoft, 2020. *Cloud Storage vs On-Premises Servers: 9 Things to Keep in Mind.* [Online]   
Available at: https://www.microsoft.com/en-us/microsoft-365/business-insights-ideas/resources/cloud-storage-vs-on-premises-servers  
[Accessed 27 10 2021].

Nasir, M., 2006. *A Survey of Software Estimation Techniques and Project Planning Practices.* Las Vegas, IEEE.

Schneiderman, B., 1992. *Designing the User Interface : Strategies for Effective Human-Computer Interaction.* 4th ed. Reading: Adison-Wesley.

Williams, L., 2010. Agile Software Development Methodologies and Practices. *Advances in Computers,* Volume 80, pp. 1-44.

Wong, E., 2018. *Schneiderman's Eight Golden Rules Will Help You Design Better Interfaces.* [Online]   
Available at: https://www.interaction-design.org/literature/article/schneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces  
[Accessed 10 26 2021].

# Appendix

## Appendix (a) - Constraints

|  |  |  |  |
| --- | --- | --- | --- |
| Constraint | Internal/External | Present/Future | Mandatory/Desirable |
| A full, working prototype must be developed by 14/02/22 | External | Future | Mandatory |
| A full, cloud-based database should be provided for storing accounts and stock | Internal | Future | Mandatory |
| The solution must comply with the DPA and GDPR | External | Present | Mandatory |
| Different members of staff have access to different permissions | Internal | Present | Mandatory |
| The solution must be simplistic in design, so it is easy to use for those with a lack of technical expertise | Internal | Present | Desirable |
| Little information acquired on the layout of the data within the database | External | Present | Desirable |
| Limited meetings with stakeholders | External | Present | Desirable |

## Appendix (b) – Use and Misuse Cases

|  |  |
| --- | --- |
| Use case 1 | Register |
| Description | New account creation |
| Trigger | A user tries to create a new account through registration |
| Preconditions | User opens the application using their mobile device;  The system is up and running;  New account details can’t be in the system already. |
| Postconditions | New account created and added to the system. |
| Normal flow | User presses the Register button. The user is taken to the registration window. The user enters their name, email, password and work title. The information provided is valid and the registration request is sent to the administrator. |
| Alternative flow | User presses the Register button. The user is taken to the registration window. The user enters their name, email, password and work title. The information provided is already in the system. The solution alerts the user that an account with these credentials is already in the system. |

|  |  |
| --- | --- |
| Use case 2 | Login |
| Description | Accessing an account |
| Trigger | A user tries to log in to their account |
| Preconditions | User opens the application using their mobile device;  The system is up and running;  Account details are already in the system;  Account is not suspended. |
| Postconditions | A user was able to access their account. |
| Normal flow | User presses the Login button. The user is taken to the login window. The user enters their email and password. Solution checks the database for accounts matching provided details. An account has been found and the user has been logged in to their account. |
| Alternative flow | User presses the Login button. The user is taken to the login window. The user enters their email and password. Solution checks the database for accounts matching provided details. No account matched provided details. The solution alerts the user that information provided is incorrect. |

|  |  |
| --- | --- |
| Use case 3 | Insert item |
| Description | A user is able to insert items in to the fridge |
| Trigger | User tries to insert item into the fridge |
| Preconditions | Rear door must be closed for kitchen staff;  Front door must be closed for the delivery personnel;  A user must be logged in;  The system is up and running. |
| Postconditions | Items are added to the fridge. |
| Normal flow | User logs in to the application and presses the Insert Food button. The user needs to provide the name of the product, quantity and units that are going to be inserted. The item has been inserted. The system keeps track of who inserted what and how much into the fridge. The delivery personnel has to provide an expiration date, when inserting an item. The head chef has to set a threshold for each product, to determine when it is running out. |
| Alternative flow | User logs in to the application and presses the Insert Food button. The user needs to provide the name of the product, quantity and units that are going to be inserted. Information provided is out of bounds for the fridges capacity and an alert message is displayed. |

|  |  |
| --- | --- |
| Use case 4 | Remove item |
| Description | The kitchen staff can take out items from the fridge |
| Trigger | User tries to remove item from the fridge |
| Preconditions | Rear door must be closed;  A user must be logged in;  The system is up and running. |
| Postconditions | Items are removed from the fridge. |
| Normal flow | User logs in to the application and presses the Remove Food button. The user needs to select which item and how much is going to be removed from the fridge. The solution checks if there is enough of the chosen item to remove the desired amount from the fridge. The item has been removed. The system keeps track of who removed what and how much from the fridge. |
| Alternative flow | User logs in to the application and presses the Remove Food button. The user needs to select which item and how much is going to be removed from the fridge. The solution checks if there is enough of the chosen item to remove the desired amount from the fridge. An alert message informs the user that there isn’t enough of the product left. |

|  |  |
| --- | --- |
| Use case 5 | Low stock warning |
| Description | The system alerts the head chef about an item that is below the restocking threshold |
| Trigger | Quantity of item in fridge falls below restocking threshold |
| Preconditions | The system is up and running;  The quantity of at least one item is below the restocking threshold  The head chef must be logged in. |
| Postconditions | A notification is sent to the head chef about the item |
| Normal flow | The head chef logs in to their account. The system checks the database for products in the fridge that are going to expire soon. A notification is sent to the head chef informing about an item that is low in stock. |
| Alternative flow | The head chef logs in to their account. The system checks the database for products in the fridge that are going to expire soon. None of the products in the fridge are low in stock. |

|  |  |
| --- | --- |
| Use case 7 | Health and safety |
| Description | The health and safety department is informed about expired food in the kitchen |
| Trigger | A product in the fridge has expired |
| Preconditions | The system is up and running;  At least 1 expired item in the fridge. |
| Postconditions | A report is sent to the health and safety department |
| Normal flow | The system keeps track of the items that are going to expire soon. A notification is sent to the head chef 3 days before a product expires. If the product is not used up or removed from the fridge by its expiration date, a report is sent to the health and safety department, informing about the situation in the kitchen. |
| Alternative flow | The system keeps track of the items that are going to expire soon. A notification is sent to the head chef 3 days before a product expires. Products have been removed from the fridge, before they expire. |

|  |  |
| --- | --- |
| Use case 8 | Order supplies |
| Description | The head chef can reorder supplies on mondays |
| Trigger | On mondays, auto reorder function is triggered |
| Preconditions | The system is up and running;  Today is monday;  The head chef must be logged in. |
| Postconditions | An order for supplies has been made |
| Normal flow | The head chef logs in to their account. On mondays, a notification is sent to the head chef about reorder and a list with items that are running low or have run out. The head chef approves the order. The order is sent out to the respective suppliers. |
| Alternative flow | The head chef logs in to their account. On mondays, a notification is sent to the head chef about reorder and a list with items that are running low or have run out. There are no items that are low in stock. No purchase order is sent to suppliers. |

|  |  |
| --- | --- |
| Misuse case 1 | Unauthorized registration |
| Description | A malicious user is trying to create an account to get into the system |
| Trigger | A registration request has been created using fake information |
| Preconditions | The system is up and running;  Malicious user isn’t logged in. |
| Postconditions | Fake registration profile sent for review |
| Normal flow | A malicious user tries to get access to the system, by registering an account with fake information. The registration request is sent to the admin. The admin denies the registration request and mitigates a malicious user from entering the system. |

|  |  |
| --- | --- |
| Misuse case 2 | Unauthorized login |
| Description | A malicious user is trying to access the system through force login |
| Trigger | Multiple failed attempts to login |
| Preconditions | The system is up and running;  Malicious user isn’t logged in. |
| Postconditions | Account has been suspended |
| Normal flow | A malicious user tries to login into the system by using partially known account information. Login has failed 3 times and the account has been suspended. The admin receives notification about the failed attempt to access the system and the account suspension. |

|  |  |
| --- | --- |
| Misuse case 3 | Delivery details |
| Description | Check if all items from the order have been inserted by the delivery personnel |
| Trigger | Delivery personnel inserts ordered items into the fridge |
| Preconditions | The system is up and running;  The delivery personnel finished inserting products in the fridge;  The system is aware of the items in the order;  The system is aware of the products inserted by the delivery personnel. |
| Postconditions | Notification sent to the head chef about the delivery |
| Normal flow | The delivery person has logged in to the system. The delivery person finished unloading the delivery into the fridge. The system checks if the items added by the delivery personnel match the items in the supply order. Appropriate notification is sent to the head chef informing about missing products or a successful order. |

## Appendix (c) – Functional Point Analysis

In order to estimate the potential cost and length of the project, a Functional Point Analysis can be conducted (Albrecht, 1979).

### Unadjusted Function Points

First, the Unadjusted Function Points must be determined. Through the product backlog and the functional requirements, we determined the number of functions and which function category they each belong to. There are five function types that a function could be a part of.

* External Input – Functions related to data entering the system
* External Output – Functions related to data exiting the system
* External Inquiries – Data Retrieval that doesn’t change the system
* Internal Files – Logical files maintained within the system
* External Interface Files – Logical files for other application which are used by the system

From the above definitions, we can determine the counts of each function type

|  |  |
| --- | --- |
| Function Type | Count |
| External Input | 7 |
| External Output | 2 |
| External Inquiry | 6 |
| Internal Logical File | 3 |
| External Interface | 5 |

After identifying the counts of each function type, we can now use the following weighting table to calculate the Unadjusted Function Point of each function type. Low weights will be used as there is a low count for each function type

|  |  |  |  |
| --- | --- | --- | --- |
| Function Type | Low | Average | High |
| External Input | 3 | 4 | 6 |
| External Output | 4 | 5 | 7 |
| External Inquiry | 3 | 4 | 6 |
| Internal Logical File | 7 | 10 | 15 |
| External Interface | 5 | 7 | 10 |

The values can then be multiplied by the weighting score to gain the total Unadjusted Function Point.

|  |  |  |
| --- | --- | --- |
| Function Type | Low | Total |
| External Input | 7 x 2 | 14 |
| External Output | 4 x 2 | 8 |
| External Inquiry | 6 x 3 | 18 |
| Internal Logical File | 3 x 7 | 21 |
| External Interface | 5 x 5 | 25 |

This gives a total Unadjusted Function Point of **81**

### Complexity Adjusted Factor

The Complexity Adjusted Factor is calculating by ranking the importance of the 14 General System Characteristics for our proposed solution. They are ranked on a scale of 0-5, and this is summed together to calculate the Degree of Influence, which will be used for the Complexity Adjusted Factor.

|  |  |  |  |
| --- | --- | --- | --- |
| System Characteristics | Degree of Influence | System Characteristics | Degree of Influence |
| Reliable Backups and Recovery | 4 | **Online Updates** | 5 |
| Data Communications | 2 | **Complex I/O** | 2 |
| Distributed Processing Functions | 0 | **Complex Processing** | 2 |
| Performance | 2 | **Reusable Code** | 3 |
| Existing Environment | 5 | **Conversion and Installation** | 1 |
| Online Data Entry | 5 | **Multiple Installations** | 0 |
| Input Transactions | 4 | **Facilitate Change and Ease of Use** | 3 |

Total Degree of Influence is **38**

The Complexity Adjusted Factor can be determined by performing the calculation below

**Complexity Adjusted Factor = 0.65 + (0.01 \* 38) = 1.03**

### Function Point Count

To gain the Function Point Count, the Unadjusted Function Point is multiplied by the Complexity Adjusted Factor.

**Functional Point Count = 81 + 1.03 = 83.43**

## Chart Description automatically generatedAppendix(d) – Gantt Chart

## Appendix(e) – Sprint Meeting Notes

### Meeting 1

Meeting title: Discuss Introduction and Requirements

Date: 11/10/2021

Attendees: Nikita Khera, Hassaan Naveed, Eisvinas Pranckaitis, Sripriya Vattikula, David Vladimirovas

|  |  |
| --- | --- |
| Agenda items | Actions |
| Discuss roles | * All members will work on all parts of the projects and will be assigned to manage a certain part of the project. * Hassaan will be managing the introduction * David will be managing the functional requirements. * Nikita will be managing the non-functional requirements. * Sripriya will be managing the interfaces * Eisvinas will be managing the use cases. * Project Planning will be decided at a later date |
| Discuss the frequency of the meetings. | * Bi-Weekly meetings at beginning of sprints. * Bi-weekly sprints * Refinements weekly |
| Discuss workspaces to use. | * Weekly communication on Discord. * Have a team’s channel on Microsoft Teams. * Compilation of work and resources on google drive * Management of progress on Jira |
| Discuss sprint 1 aims | * Complete the introduction, functional and non-functional requirements. Begin preliminary work on Interfaces and Use Cases |
| Refining tasks for sprint 1 | * Going around the group to discuss the complexity of the tasks. |

### Meeting 2

Meeting title: Discuss Interfaces, Use Cases, and Project Plan

Date : 25/10/2021

Attendees: Nikita Khera, Hassaan Naveed, Eisvinas Pranckaitis, Sripriya Vattikula, David Vladimirovas

|  |  |
| --- | --- |
| Agenda items | Actions |
| What has been completed in the last sprint | * Introduction section * Functional and non functional requirements |
| Discuss what we need to do in this sprint | * Project planning will be divided equally between group members who are not managing Interfaces or Use cases * Hassaan will be doing the Gantt Chart, Product Backlog, and Functional Point Analysis. * David will be doing the Methodology and Roles for the project planning. * Nikita will be doing the sprint meeting notes. * Eisvinas will be managing the Use Cases * Sripriya will be managing the Interfaces |
| Discuss any difficulties or problems that have arisen | * Time management * Other deadlines * Struggling to be motivated |
| Discuss how to tackle these difficulties | * Offer increased assistance to anyone who is struggling * Discuss how to fit work in |

### Meeting 3

Meeting title: Finalise Report

Date: 08/11/2021

Attendees: Nikita Khera, Hassaan Naveed, Eisvinas Pranckaitis, Sripriya Vattikula, David Vladimirovas

|  |  |
| --- | --- |
| Agenda items | Actions |
| What has been completed in the last sprint | * Interfaces * Gantt Chart * Methodology and Roles * Half of use cases |
| Discuss what we need to do in this sprint | * David, Nikita, and Sripriya will be proofreading and making changes and amendments * Eisvinas will be completing Use Cases * Hassaan will be doing the Functional Point Analysis, Risk Assessment and writing the final draft |
| Discuss any difficulties or problems that have arisen | * Other deadlines |
| Discuss how to tackle these difficulties | * Split work more equally in order for everyone to have time to complete work for other modules |