Sharing Surplus

School of Computer Science

Samarth Manjunatha Bedre

230016680

Supervisor: Dr Dharini Balasubramaniam

In Partial Fulfillment of the Requirements for the Degree of
MSc Software Engineering
at the University of St Andrews, School of Computer Science

13 August 2024

Abstract

Food redistribution has gained significant attention in the past few years with various initiatives such as such as FareShare (http://fareshare.org.uk), OLIO (https://olioex.com), and RipeNearMe (https://www.ripenear.me), all attempting to minimise the food waste. However, the wastage persists on a smaller scale with many residents in towns and cities, with their relatively small gardens and orchards, end up producing a glut of fruits and vegetables that their families can't consume, leading surplus food being wasted.

This dissertation introduces 'Sharing Surplus', a digital platform that is designed to address this problem by facilitating sharing of surplus fruits and vegetables grown within the local community. The project reviews various food sharing models that contribute in the reduction of food waste and evaluates the existing solutions to identify their gaps and limitations. To address these shortcomings, 'Sharing Surplus' is proposed as a mobile application that prioritises user anonymity, allowing both producers and consumers to share surplus produce without sharing personal information. The application incorporates neutral pickup locations food sharing model to ensure the privacy and safety of the users, providing a practical and community-driven solution to minimise the food waste.

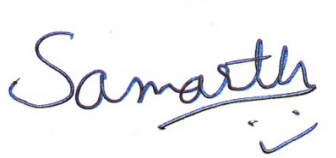
Declaration

I hereby certify that this dissertation, which is approximately *14753* words in length, has been composed by me, that it is the record of work carried out by me and that it has not been submitted in any previous application for a degree. This project was conducted by me at the University of St Andrews from May 2024 to August 2024 towards fulfilment of the requirements of the University of St Andrews for the degree of *MSc Software Engineering* under the supervision of *Dr. Dharini Balasubramaniam*.

In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work."

Date: 13th August 2024

Signature:



Samarth Manjunatha Bedre

Acknowledgments

I would like to thank my supervisor, Dr. Dharini Balasubramaniam, for her unwavering enthusiasm towards this project and the invaluable guidance and support given to me throughout the project. I am very grateful for our weekly meetings where I had the chance to discuss ideas regarding this project. I will surely miss them!

I would like to thank Sonja Potjewijd and Transition University of St Andrews for their valuable feedback and suggestions regarding the app. This surely helped make the application better.

I would like to thank my family for their unconditional support and encouragement throughout my degree. I would also like to thank my friends for their motivation throughout the project, I couldn't have done it without you!

Table of Contents

List of Figures

List of Tables

1. Introduction

Advancements in agriculture, particularly through the integration of digital technology, have led to an increase in the food production, which often exceeds the needs of the growers [1]. This surplus, particularly in communities where individuals grow fruits and vegetables in their backyards or gardens, contributes to the widespread food waste. Whilst existing solutions, such as food banks and community sharing platforms, have tried to address this issue, challenges related to scalability, user engagement, and privacy persist. Recognising the need for an effective solution, 'Sharing Surplus' aims to contribute to the reduction of food waste by offering a new approach.

Facilitating the sharing of surplus produce requires a comprehensive solution. It is essential to develop a common platform that allows the users to connect, communicate and notify each other about the availability of excess produce. This platform would allow the interested users to request surplus produce directly from the producers, which in turn fosters a more connected and sustainable society.

The goal of 'Sharing Surplus' is to create a platform that brings producers and requesters together to share surplus produce within their community. This project has been developed for Android platform, written entirely in Kotlin, and adheres to the standard industrial coding practices. The app is designed to be intuitive and user friendly, ensuring a straightforward and seamless experience for all the users.

The primary motivation behind this project is rooted in the sustainability-led development initiatives, which are increasingly popular worldwide. In an era where global warming and pollution are at the forefront of public concern, it is more important than ever to reduce waste. 'Sharing Surplus' is an initiative aimed at minimising the wastage of surplus produce, thereby contributing to broader environmental sustainability goals.

While similar solutions exist, many are targeted at large-scale operations rather than individual users. These platforms often offer monetary incentives to the users for collecting surplus from big businesses, such as getting discounted price on leftover products. However, such approaches do not necessarily promote community integration or support socially driven initiatives.

'Sharing Surplus' addresses the gap in the existing solutions by providing a sustainable, community-focused platform. The application includes features such as neutral pickup locations to protect the user privacy, 'Good Deed Points' to incentivise participation, and a community forum for the users to engage in discussions.

Extensive planning and requirements gathering were done to ensure the delivery of an effective solution. Significant effort was taken in integrating modern technologies to make the application adaptable and sustainable for the long term.

1.1 Report Structure

The dissertation consists of 9 chapters including the introduction. It is structed as follows.

- **Chapter 1:**
  The *Introduction* chapter outlines the rationale and motivation behind the project.

- **Chapter 2:**
  The *Context Survey* chapter provides context on the problem space, discussing the benefits of food sharing in reducing food waste, the effectiveness of mobile applications, and various food-sharing models. It also examines the existing solutions aimed at sharing surplus food and identifies the gaps in them.

- **Chapter 3:**
  The *Requirements Specifications* chapter presents the functional and non-functional requirements along with the use-cases for the application.

- **Chapter 4:**
  The *Software Engineering Process* chapter discusses the Software Engineering process used to develop the application.

- **Chapter 5:**
  The *Ethics* chapter details the ethical considerations and approval process for the application.

- **Chapter 6:**
  The *Design* chapter focuses on the design aspects, including Human-Computer Interaction (HCI) considerations and key design decisions.

- **Chapter 7:**
  The *Implementation* chapter highlights the implementation of the design decisions.

- **Chapter 8:**
  The *Evaluation* chapter describes the evaluation process of ‘Sharing Surplus’, comparing it with the existing solutions.

- **Chapter 9:**
  The *Conclusion* chapter concludes the study.

- **Chapter 10:**
  The *Limitations and Future Works* chapter highlights limitations of the study and outlining the scope of future work.

2. Context Survey

This chapter provides a brief overview of some of the key aspects of Food Sharing and the role of technology in facilitating it. It then explores different food sharing models, followed by an analysis of existing solutions, identifying their limitations and gaps.

2.1 Background and context

2.1.1 Introduction

It has become a common occurrence for people with gardens to grow fruits and vegetables. Amidst the growing 'cost-of-living crisis' [2], inflation in the UK has reached a 41-year high in recent times. To combat the effects of rising food prices and inflation, almost 30% of the households with gardens have started growing their own produce [3]. Those growing their own products report saving an average of £13 a month, equivalent of £156 a year.

Growing fruits and vegetables offers a wide range of benefits to the people, ranging from health benefits to environmental benefits [4]. One important environment benefit of growing food locally is the reduction of 'food miles' [5]. Food miles refer to the distance the food has travelled from where it was grown to the plate of the consumer [6]. Studies have shown that the higher the food miles are, the greater is the caused emission caused by them [7]. This means that the transportation of food from different parts have to be coordinated by complicated supply chains and have to be transported on vehicles, whose emission leave a higher carbon footprint. This in-turn contributes to the increase in global greenhouse gas emissions, thereby having an impact on the global warming and climate change [8]. Growing food locally ensures that the food does not travel longer distances, thereby reducing the 'food miles' [9]. This also encourages individuals to take initiative to play a role in mitigating climate change.

Whilst this is a welcoming move in the right direction, often it results in the production of excess fruits and vegetables. As most of the growers are not professional farmers, they end up growing more than what their entire family can consume. This can lead to them having a lot of surplus fruits and vegetables that simply go to waste. Many people choose to compost their surplus produce [10]. Composting makes the waste food into a valuable and nutrient rich fertiliser, that can be used for the plants in the garden. Whilst composting is a good way to save the money spent on the fertiliser and reducing the food waste, it still means usable fruits and vegetables are thrown away.

According to a recent House of Commons report [11], around 7.2 million households are 'food insecure'. Food insecurity means household members sometimes disrupt their eating patterns or reduce their food intake because they lack money or other resources for food. This number is increasing due to 'cost-of-living crisis' and ongoing economic challenges. Hence, it is crucial to reduce any sort of food wastage caused and distribute the surplus food to the people in need. This includes reducing the wastage and distribution of the fruits and vegetables grown

in the gardens of the households. Often, the surplus produce from the garden is wasted due to lack of knowledge and opportunities for sharing. Hence, it is important to come up with a solution that both educates the growers on surplus food as well as create a platform for the users to share their surplus produce with the people in need.

2.1.2 Food Sharing as a solution

Food sharing is generally defined as the distribution of surplus food among a group of people through the activity of sharing [12]. Ever since humans started living together as a society, sharing food has been a major human activity [13]. It has fostered the bonding in the community for millennia. As Belk [14] states that "*in sharing two or more people may enjoy the benefits (or costs) that flow from possessing a thing. Rather than distinguishing what is mine or yours, sharing defines something as ours".* This gives sharing modern definition that describes it as a human act or involvement in order to gain benefits, not only for the self, but also for the community by using existing resources [15]. The major aim of food sharing has always been to give the surplus food to the ones in need, instead of throwing it away. As mentioned by Gollnhofer [16], food sharing often helps to reduce food waste to larger degree than simply disposing the surplus food. Hence, sharing the surplus food has better benefits than letting it go waste or composting it.

2.1.3 Mobile applications as a solution to digital food sharing

There are many ways to share the surplus food, from donating it to the needy on the streets to distributing it through a well-known food surplus distributor. However, often, the people who grow fruits and vegetables tend to live in sub-urban and rural areas, and usually have their own gardens and orchards. This makes it difficult for the growers to find opportunities to distribute their surplus produce. Therefore, most of the times, the growers decide to compost the excess fruits and vegetables [4].

One of the prominent ways to bring about food sharing in these sub-urban areas is through the use of technology, in particular, mobile apps. The use of mobile phones has been increasing across all age groups in recent years [17], and the access to the internet has made it a popular choice of technology to have. Around 98% of the adult population in the UK are said to have a mobile phone [18], thus making it the most popular electronic device amongst the population. The Ofcom report [19] also states that around 93% of the UK landmass had 4G coverage from at least one telecom operator. Hence, mobile applications would serve as an effective way to share the surplus food. Food sharing brought about by mobile applications have been proven effective in reducing the food waste and bringing about a positive change in the community.
According to the study done by John Harvey, et al. [20], data from 54913 instances of food sharing between 9054 people collected over 10 months from the app 'OLIO' [21] was analysed. This study revealed that the app helped the users to share excess food within the local community effectively, with 92% of the transactions taking place withing a

geographical radius of 10kms. This suggests that mobile applications are effective for users to share items with the local community.

2.1.4 Case Studies and Existing research on Food Sharing

Food sharing through the internet has been proven to be very effective in reducing the food wastage. For example, in their study of the food sharing platform called 'foodsharing.de' [22], Ganglbauer et al., examined how the platform was used to facilitate food sharing amongst customers, farmers, organisations and retailers in order to reduce food waste in Austria and Germany. Their findings showed that the platform successfully facilitated the sharing food amongst the large number of participants, as evidenced by an active user base of 17,000 users sharing around 1800 food baskets within their online community [23].

Farr-Wharton et al. [24], investigated three apps – Fridge Pal [25], LeftOverSwap [26], and EatChaFood [27] and the role they played in distributing the surplus food and reducing the food wastage. It was found that the apps had a positive impact on raising customer awareness of their food supply, location and literacy, and the benefits of sharing surplus food.

A study by Pisoni et al. [28], on the effect of digital food sharing platforms in Italy and Germany shows that importance of digital platforms in the rising popularity of sharing economic initiatives. This study shows the increase in tendency of the new generation of millennials to use digital platforms to get products through sharing initiatives rather than buying it in an old-fashioned way.

An initiative in the KTH Royal institute of Sweden lead to the development of an app to facilitate employees and students at the university campus to share food. This initiative, led by Katezeff et al. [29], tested the app and the associated food sharing activities for 2 weeks. The results from the study showed the willingness and enthusiasm from the users to use a digital service for sharing food. It was also noted in the study that there was a need of a critical mass of users to make the app effective, and when done so with effective organisation, the users found it effective to use the digital platform for sharing and collecting the food.

A study by Makov et. al [30] showed that digital food sharing not only rose but outperformed all the projections during the Covid-19 Pandemic. The paper further discusses how this number is projected to increase as the years go by, thereby placing an emphasis on the importance of digital sharing platforms for the distribution of food surpluses.

2.1.5 Technology and market insights

The recent survey [31] on the market share of mobile operating systems in the UK showed that Android OS had a market share of 50.26% in April 2024, making it the most popular Mobile Operating system in the UK. Creation of an Android app for sharing the surplus food will ensure wider reach amongst the public and enable sharing of surplus food with

greater audience. Rayan et. al. [32], in their study, show how to design and analyse an android app for sharing the surplus food. This paper goes into detail about the software engineering process for design of such apps.

2.2 Food Sharing Models

In their analysis of 52 food sharing cases, Michellini et. al [33] identified three prominent food sharing models that are currently in use. These models are:

2.2.1 Sharing for Money

The 'Sharing for Money' model operates as a Business-To-Customer (B2C) approach, that aims to reduce food waste while generating revenue. In this model, food is collected from businesses such as like cafes, restaurants and supermarkets and then sold to customers at a discounted price. These discounts can range from 20% to 60% or more, depending on the expiry date of the item. The model benefits the consumers by providing them with products at a discounted price, while also benefitting the businesses by reducing the waste, thereby supporting their environmental commitments. Initiatives like *Too Good to Go* and *Foody Bag* exemplify this model. **Figure 2. 1**, illustrates the structure of this model.

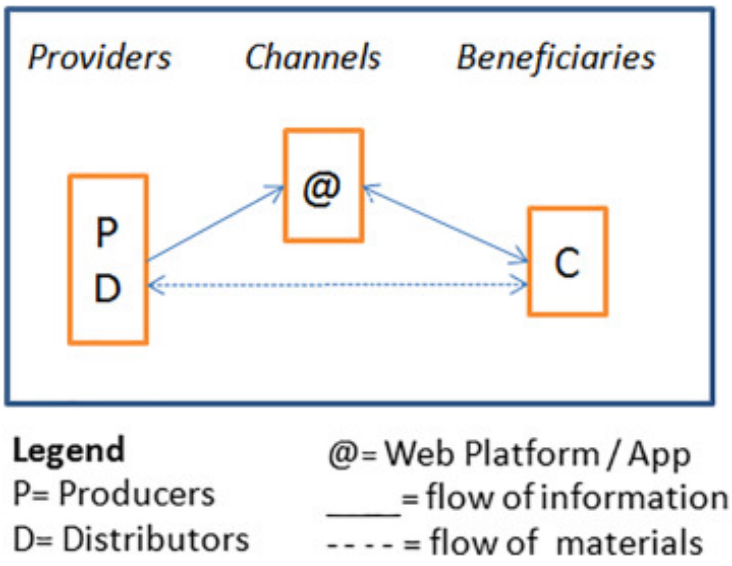


Figure 2. 1: *The 'Sharing for Money' Model [33].*

Although this model has been implemented in various ways by different solutions, they all share the common goal of achieving monetary benefits through reducing food waste.

2.2.2 Sharing for Charity

The 'Sharing for Charity' model primarily involves non-profit organisations and operate on Business-to-Business (B2B) and Customer-to-Business (C2B) delivery methods. In this model, food is collected various providers, both individuals and businesses, and is distributed free of cost to non-profit organisations. These organisations then ensures that the surplus food reaches those in need. This model benefits the providers by helping them reduce their food waste, while also allowing businesses to contribute to their Corporate Social Responsibility (CSR) initiatives. Additionally, it supports the underprivileged by providing them with free food. Initiatives like *FareShare* operate on this model. **Figure 2. 2** depicts the structure of this model.

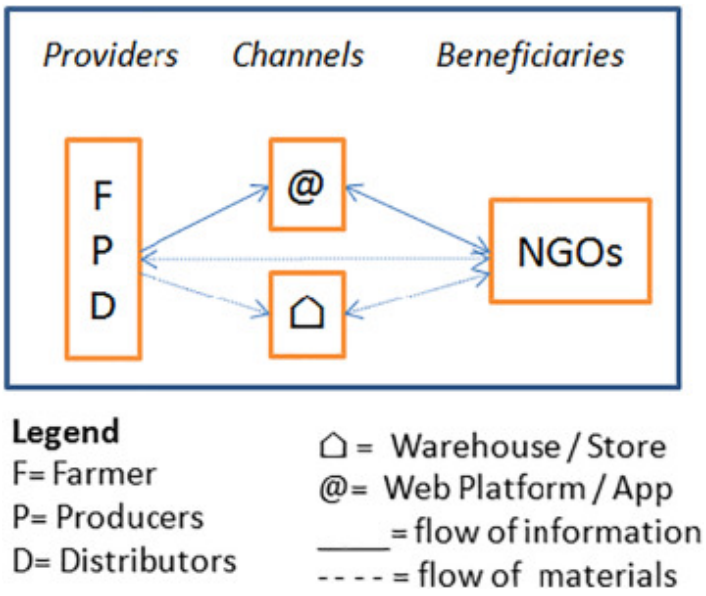


Figure 2. 2 **:** *The 'Sharing for Charity' model [33].*

The key feature of this model is the absence of monetary transactions, focusing instead on a free food sharing initiative.

2.2.3 Sharing for the Community

The 'Sharing for Community' model is a Peer-to-Peer (P2P) approach to food sharing within the community. In this model, food is primarily collected from the producers and is shared free of charge with other consumers at a local level, fostering a community engagement. Initiatives using this model do not manage volunteers or logistics, thereby creating a "for the community, by the community" dynamic. This model also strengthens community connections and fosters networks amongst its members. Initiatives like *Olio* adopt this model. **Figure 2. 3** provides a visual representation of this model.

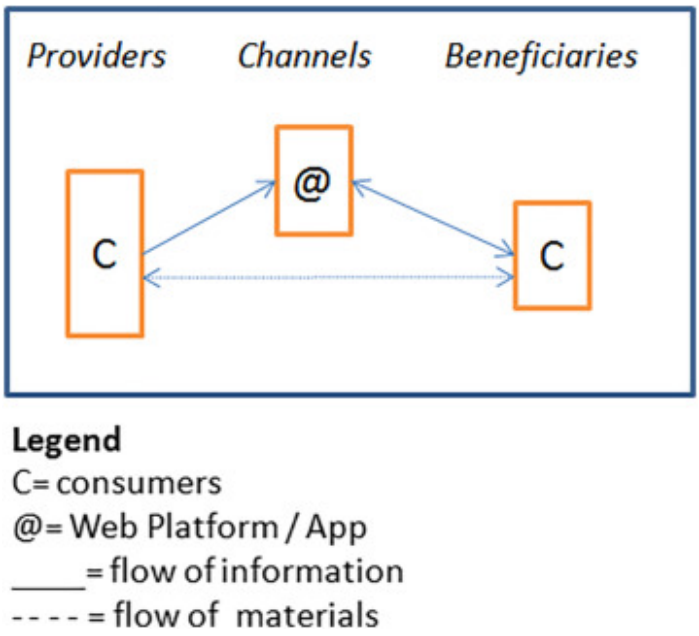


***Figure 2. 3**: The 'Sharing for Community' model [33].*

The core principle of this model is the removal of a central moderator, allowing the users to manage the sharing process themselves.

Since, 'Sharing Surplus' aims to exclude monetary aspects from food sharing and primarily focuses on fostering community growth, it is logical to adopt the 'Sharing for the Community' model. This approach further aligns with the application's goal of strengthening community and encouraging active community participation. **Figure 2. 4** further demonstrates the benefits of each model.

Benefits		Sharing for Money	Sharing for Charity	Sharing for the Community
Consumers	Discounted products	√		√
	Free product		√	√
	Neworking / Sense of community			√
Society	Reduce food waste	√	√	√
	Poverty alleviation		√	
	Increase awarness of food wastage	√	√	√
	Increase awarness of poverty		√	
Providers	Reduce logistics / waste disposal costs	√	√	
	Improve profit for providers	√		
	Enhance reputation and image	√	√	

Value for providers Value for the society Value for consumers

***Figure 2. 4**: Value proposition of different models [33].*

It is evident that the 'Sharing for Community' model not only benefits the consumers using this application but also strengthens the local community ties.

2.3 Related Work

1. FareShare (website)

FareShare is a network of charitable food redistributors. It is made up of 18 independent UK organisations[34]. FareShare works by delivering quality surplus food from across the food industry to nearly 8500 frontline charities and community groups[35]. Their mission is to redistribute food which is nutritious and good to eat. And through their initiative and collaboration with multiple different organisations, FareShare have been able to provide enough food to create around a million meals every week. FareShare collects all kinds of food items including fruits, vegetables, packed meals, etc. from the food and drink industry and redistributes it before it goes waste[36]. However, FareShare collects food from only large retailers, and does not accept individual donations. And because of this reason, FareShare doesn't have a mobile application. The organisations that wish to donate the food need to register to donate on FareShare's website and FareShare arranges for the food to be collected. FareShare also stores operates a Food Safety policy which ensures that the food items collected are handled properly. FareShare also acts as a mediator between the charities and the organisations that donate the food[37].

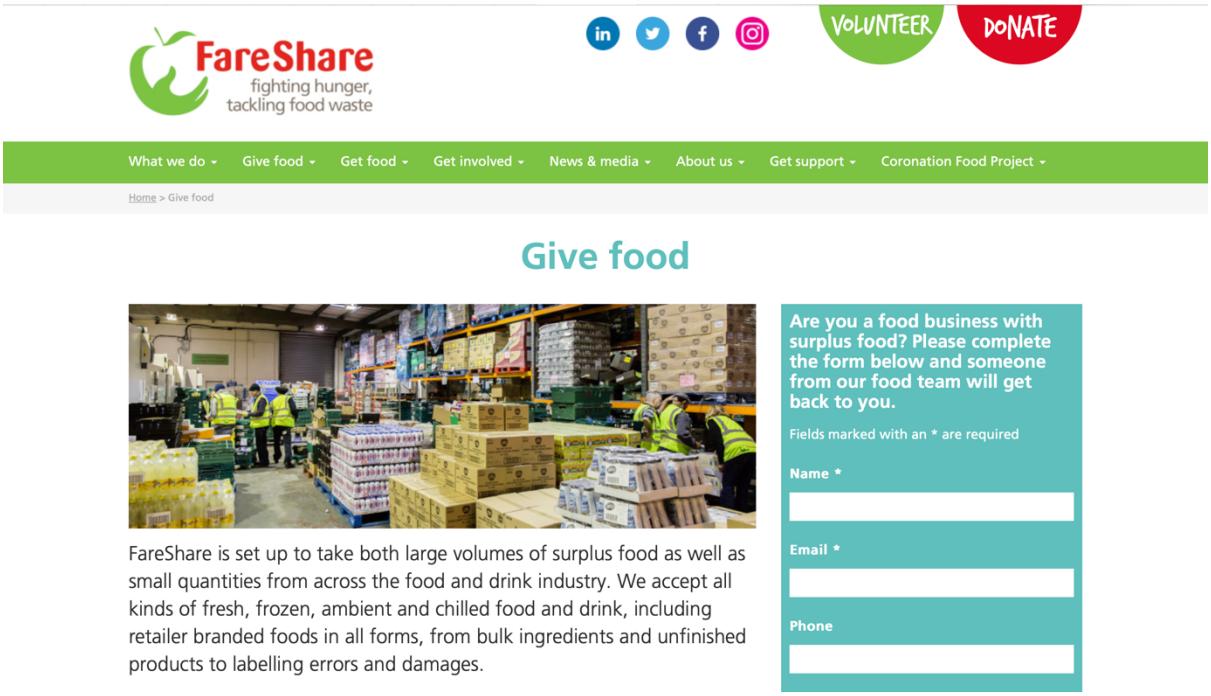


Figure 2. 5: *The food sharing portal of FareShare [36].*

2. Olio (website and app)

Olio is a sharing application that helps people pass on things that they no longer require[21]. This includes anything ranging from food and clothes to household items. The users can share both perishable and non-perishable items. Olio allows the users to give and get items for free. Should the users choose to charge for any item, Olio allows the users to put up an asking price for the item that they are sharing. Users who no longer need an item can choose to advertise it and any other users who are in need of that item can contact the doner and collect it. Olio gives the users the flexibility of selecting their own location as the pick-up location. Olio allows allocates Karma points to the users who share items with others. This gives the users a motivation to share things that they no longer need[38].

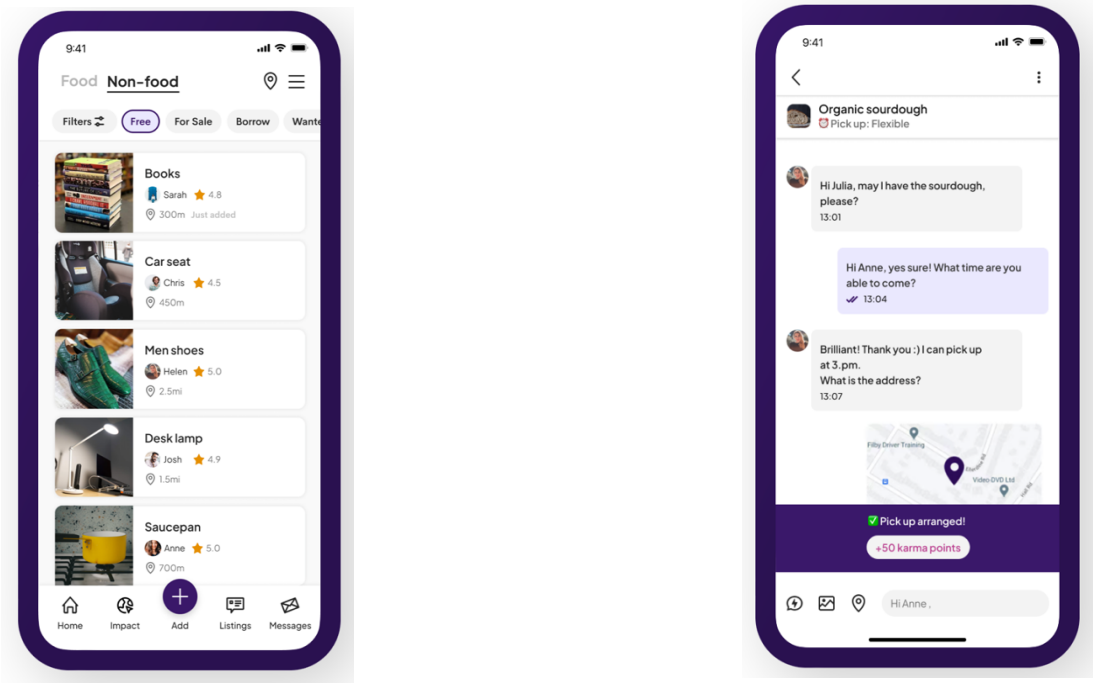


Figure 2. 6: *The User Interface of Olio [21].*

3. Too Good to Go (website and app)

Too Good to Go is an app that is on a mission to inspire everyone and fight food waste together[39]. It acts as a food wastage reducing app by allowing nearby shops, cafes, grocery stores, and restaurants to share their unsold surplus food such as snacks, takeaway meals, and ingredients. The retailers sell these unsold surpluses at reduced prices. This ensures that the food left over from these places are not wasted, instead consumed by the people in need. The users also get access to quality food items at reduced prices. Too Good to Go operates on a model of 'Surprise Bags' [40]. This essential includes items that are left over at the shops at the end of the day. Since the retailers do not know in advance what food will be left over, the items included in the surprise bags change from order to order. This app however works on a first-come-first-serve basis. Hence the users need to register for the surprise bags as there not be many of these left. Whilst this app

does allow the shops to give the surplus items at reduced price, it doesn't allow the retailers to give them out for free.

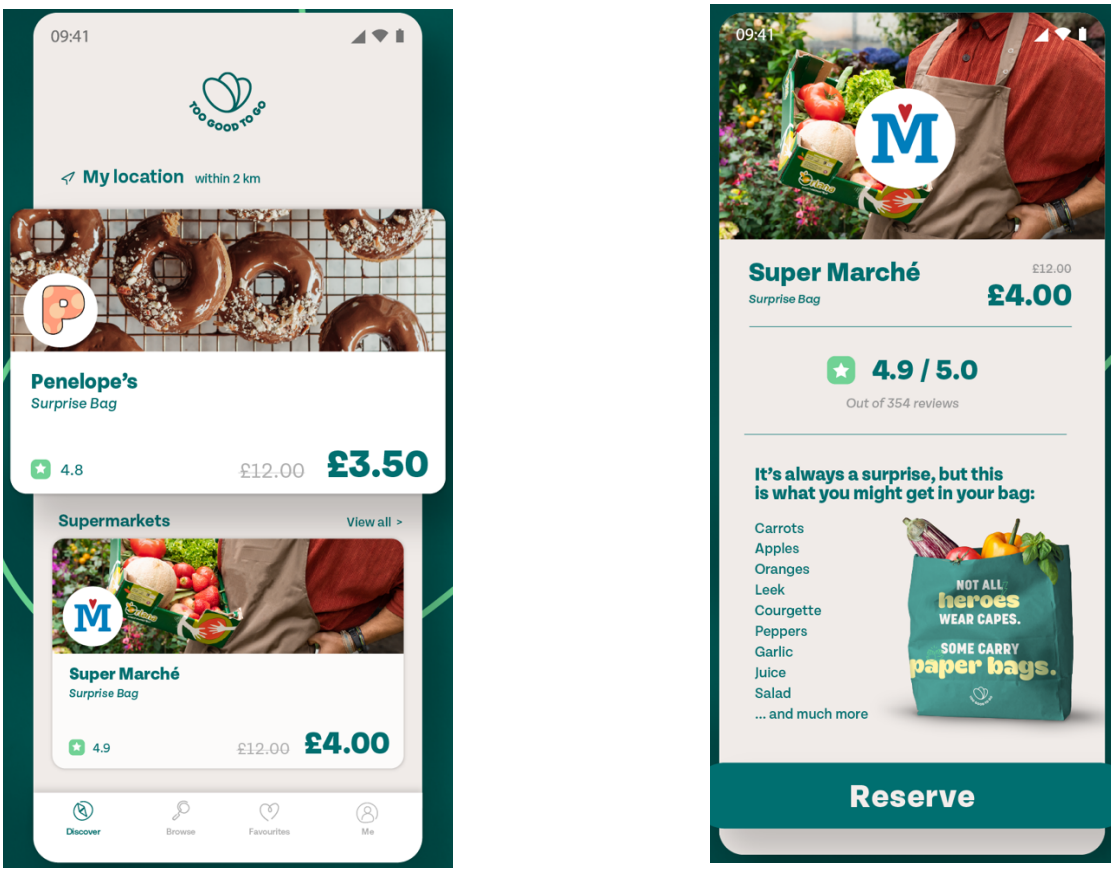


Figure 2. 7: *The User Interface of Too Good to Go [40].*

4. Foody Bag (website and app)

Foody Bag is an app that offers a curated selection of random, delectable food items available at one-third of their retail price [41]. The items offered includes a wide range of options from sandwiches and donuts to rolls, bread, supermarket items, meat, bakery goods and much more. This apps allows the users to filter out what they need, based on their budget, store types, dietary requirements, etc. The users can also opt to get notifications from their favourite stores so that they don't miss out on these deals. However, this app doesn't let the stores give out the surplus food for free. It also doesn't allow individuals to donate their surplus produce [42].

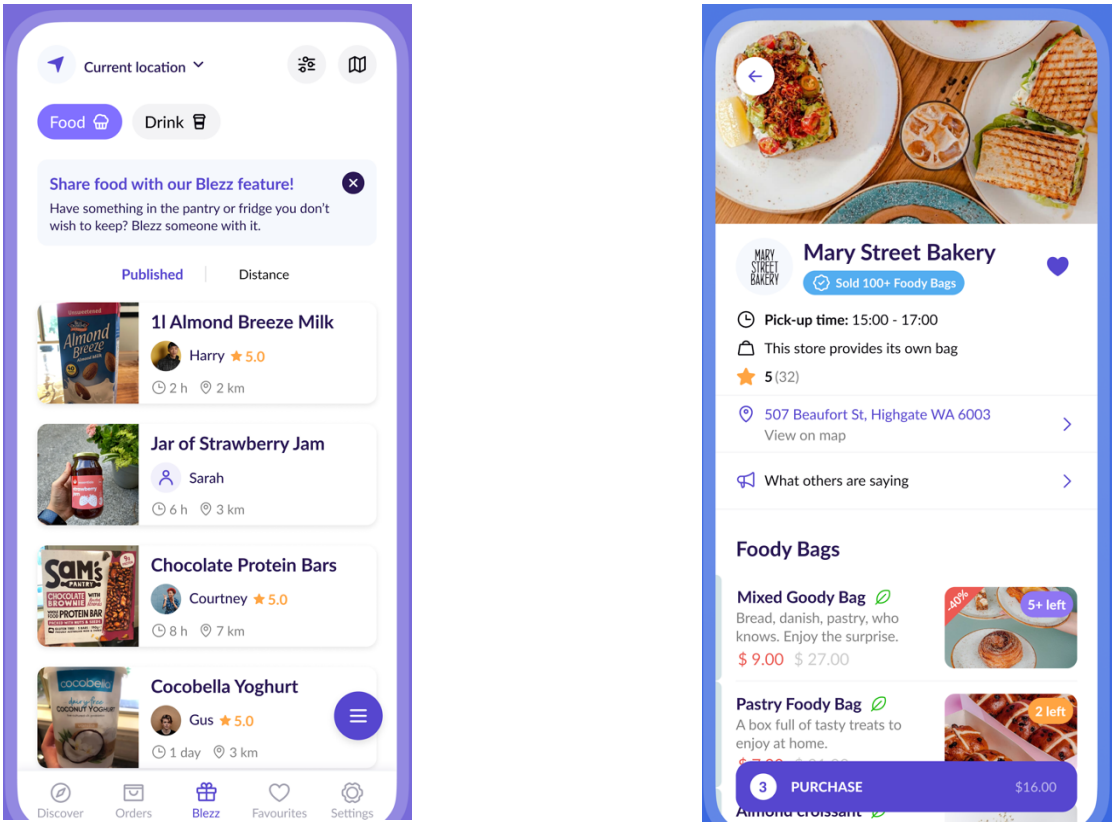


***Figure 2. 8**: The user interface of Foody Bag [42]*

2.3.1 Key Takeaways and drawbacks

The key takeaways from the above related work can be summarised into three following points.

1. The apps use different operational models to share the surplus food. FareShare acts as a mediator for large-scale donations from retailers to charities, OLIO facilitates peer-to-peer sharing, and Too Good to Go and Foody Bag focus on selling surplus food at reduced prices.
2. The apps also target different types of users for food distribution. Whilst FareShare targets large retailers and charities, OLIO, Too Good to Go, and Foody Bag cater to individual users and small businesses. OLIO and Too Good to Go enable direct user involvement, whereas FareShare and Foody Bag primarily deal with businesses and retailers.
3. The apps also have different financial models for sharing the surplus food. Whilst Olio gives users the flexibility to share the items freely or charge money for it, Too Good to Go and Foody bag focus solely on selling the surplus food at a reduced price. FareShare does not engage in monetization and focuses on charitable redistribution instead without any individual user interaction.

2.3.2 Gaps and Limitations

While the existing solutions provide a good framework and a starting point for sharing surplus food and reducing food wastage, they also have several gaps and limitations. These can be summarised as following.

1. **Accessibility and Inclusivity**
   The solutions have varying levels of accessibility and inclusivity. FareShare does not accept individual donations, which limits the involvement of small-scale producers and households. Apps like Too Good to Go and Foody Bag focus on commercial surplus rather than surplus from the households, which excludes a significant number of potential donors.

2. **User Engagement**
   The solutions have different levels of user engagement. FareShare doesn't let the retailers have control over where their excess food goes, thereby limiting a chance to build community engagements. Too Good to Go and Foody Bag focus primarily on transactions rather than fostering a community of sharing.

3. **Financial and Operational Models**
   The solutions also have different ways of functioning, some of which might not be exactly what the community wants. Too Good to Go and Foody Bag operate totally on generating income from selling the surplus food, which might limit the organisations and individuals who are looking to donate the surplus without financial transactions. FareShare's reliance on donations from Large Scale retailers restricts it from adapting to the varied nature of the household surplus.

4. **Geographic Constraints**
   The geographical limitations of the solutions also play a big part in limiting the wider reach of them. FareShare focuses on large-scale redistribution, which is not always efficient for small-scale redistribution of surplus locally. This is due to the reliability of FareShare on an efficient supply chain, which ensures that the surplus food benefits the people in need on a regular basis [43]. Too Good to Go and Foody Bag depend on regions with higher concentration of businesses, which is often not the case in terms of a sub-urban and rural areas.

5. **Anonymity**
   The existing solutions do not put major efforts in ensuring anonymity when it comes to sharing the surplus. Foody Bag and Too Good to Go requires the users to go to the businesses to collect the surplus food. While Olio does facilitate peer-to-peer sharing, it doesn't prioritise anonymity and neutral pick-up locations.

By identifying these gaps and limitations, ‘Sharing Surplus’ aims to create a more engaging and user-friendly platform that addresses these challenges.

3. Requirement Specification

In this chapter, the requirements of the system are detailed. The first section of this chapter uses UML Use Case Diagram to depict the use cases of the application. A UML Use Case diagram helps to describe the high-level functions and scope of the system, by modelling the behaviour of the system based on its requirements [44].

The requirements were derived based on an extensive analysis of the existing solutions, identifying the gaps they provided, and from the research conducted on the similar previous works in the domain of surplus food sharing. The initial DOER document served as the foundation to the fundamental requirements. The second section of this chapter details both the Functional and Non-functional Requirements.

3.1 Use Case

Table 3. 1: *Use Cases for 'Sharing Surplus'*

Use Cases
Register to the app
Log in to the app
Manage the profile information (change or update details)
Create a produce listing (upload a surplus produce with the image and specify a pickup location – either a neutral location/location of choice)
View available produce listings (check the list of available surplus produce)
Browse produce listing
Request a surplus produce (select a suitable time and date)
Manage the surplus produce requests (accept or delete requests)
Select the pickup location (neutral location or specific location)
Participate in forum discussions
Receive status updates

3.1.1 Use Case Diagram

The **Figure 3. 1** depicts a UML Use Case Diagram for the proposed system. This diagram models the interaction of the user with the system. The users are further classified into producers and consumers.

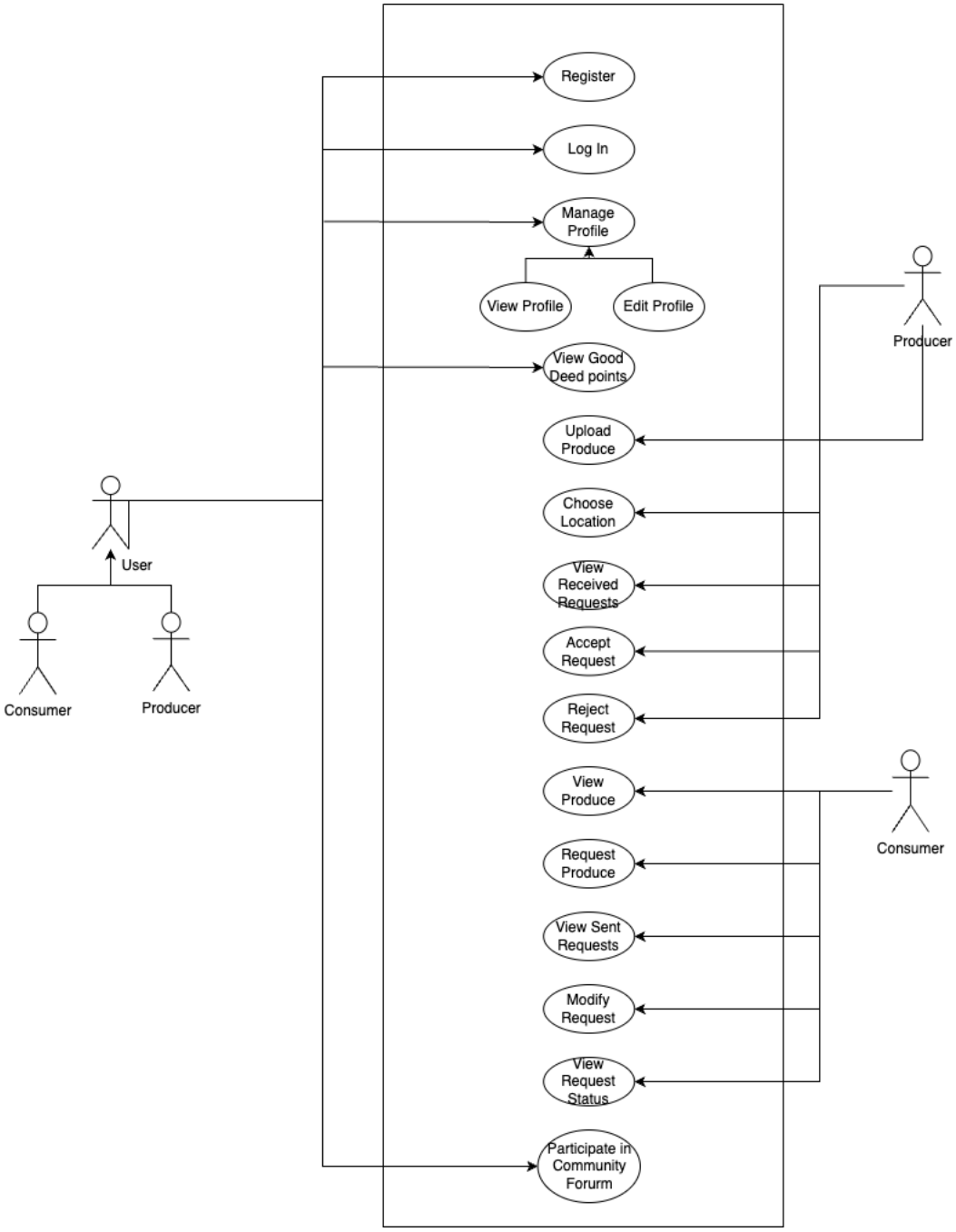


Figure 3. 1: *Use Case Diagram of 'Sharing Surplus'*

3.2 Functional and Non-functional Requirements

This section details the functional and non-functional requirements of the proposed system. As Sommerville describes it [45], functional requirements describe the requested functionality/behaviour of the system, whereas the non-functional requirements describe the constraints on the system. The requirements are given a rank based on their perceived importance for the working of the proposed system. The

priorities High, Medium, and Low detail the importance of a particular requirement for the system.

3.2.1 Functional Requirements

Table 3. 2: *Functional Requirements of 'Sharing Surplus'*

Functional Requirements	Importance	Description
User Registration and Authentication - **Register to the app.** - **Login to the app** - **OAuth**	**H** **H** **H** **M/L**	**This requirement would let the user register and login to the app.**
User Profile Management - **Manage profile information.** - **Check 'Good Deed Points'.**	**M** **M** **M**	**This requirement would let the user to manage the profile and check points.**
Produce Listing - **Upload Produce listing.** - **Manage Produce Listing** - **View Produce Listing** - **Specify pickup Location – neutral or specific**	**H** **H** **H** **H** **H**	**This requirement would let the user to upload, and modify produce, whilst allowing the user to specify the pickup location.**
Produce Browsing and Searching - **Search for product by name.** - **Filter by multiple options** - **Manage user's own postings**	**M** **L** **M** **M**	**This requirement would let the user to view the details regarding the listed produce.**
Produce Exchange Management - **Request for produce.**	**H** **H**	**This requirement would let the user to request the produce and manage the requests.**

- Manage incoming and outgoing requests. - Chat system (if possible)	H L	
Push Notifications - Notification if a supplier agrees your request	M L	This requirement would let notify the user whenever a change in the request status occurs.
Feedback System - Users rate the exchanges	L L	This requirement would let the user give feedback regarding the exchange
User Forum and Community Engagement - Participate in community interactions. - Share Recipes etc. - Comment, Like.	M M L L	This requirement would let the user interact with the community by sharing content.

While most of the fundamental requirements are rated High (H), few requirements are rated Medium (M) or Low (L). Whilst these requirements are a beneficial addition, due to constraints, only the 'High' are prioritised.

3.2.2 Non-Functional Requirements

Table 3. 3: *The non-functional requirements of 'Sharing Surplus'*

Non-Functional Requirements	Importance	Description
Performance - Quick responses and minimum Latency	H H	This requirement would let the app be fast.
Reliability - Data Backup	H H	This requirement would let the data be resilient.
Scalability - Should handle the increase in the number of users and listings	M M	This requirement would let the app handle increase in the number of users.
Usability - Intuitive and user friendly - Easily accessible	H H H	This requirement would let the app be intuitive and user friendly.

Security **- Data to be protected.** **- User Auth needs to be robust**	**H** **H** **H**	**This requirement would let the data in the app be secure.**
Privacy **- User anonymity should be maintained.** **- End-to-end encrypted chats, if done!**	**H** **H** **-**	**This requirement would let the app uphold user privacy.**

The non-functional requirements are mostly rated High, as these are essential for the proper working of the application.

4. Software Engineering Process

This chapter discusses the software engineering process followed in the development of 'Sharing Surplus'. A brief discussion on how the entire development lifecycle was planned is given.

Software Engineering Process encompasses the entire software lifecycle from its conception to competition of the software product [46]. As software grows in size and complexity, it is essential to follow a standardised process to effectively manage the software development. Given the tight schedule, Agile and Scrum methodologies were used in the development.

4.1 Requirements Elicitation

The initial set of requirements was gathered based on the gaps in the existing solutions and the previous work done in the field. These requirements were finalised and rated after a discussion with the supervisor. Since the app demanded frequent modifications of the features, a traditional plan-driven process would have increased the timescale of the project and would have limited the flexibility to modify things without having to go through the software development process. **Figure 4. 1** depicts a typical plan driven development lifecycle.

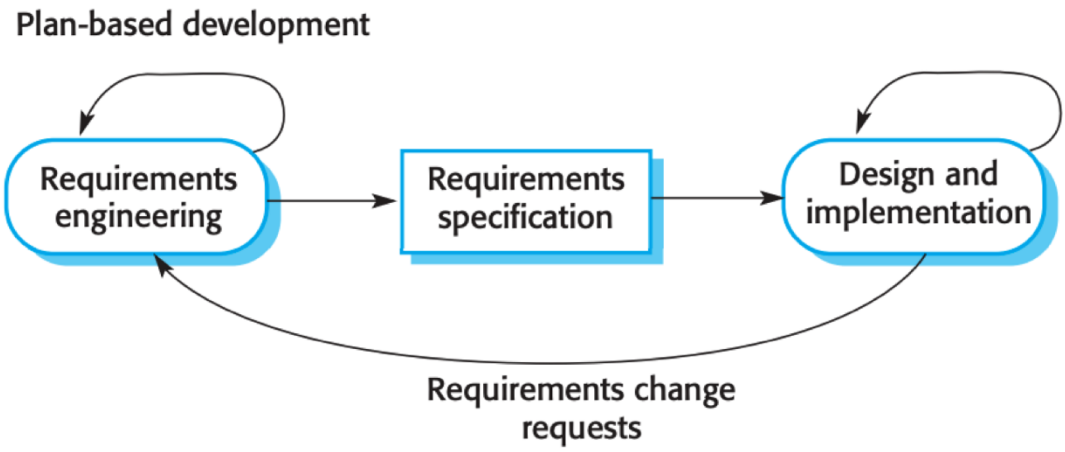


Figure 4. 1: *A typical plan driven development process [45].*

Therefore, Agile Scrum methodology was chosen for the app development.

4.2 Agile Development

For this project, the Scrum Agile methodology was adopted. Scrum allows for software development to be incremental, and adapt and adjust to the changing requirements [47]. **Figure 4. 2** illustrates a sprint lifecycle that was used the development of this project.

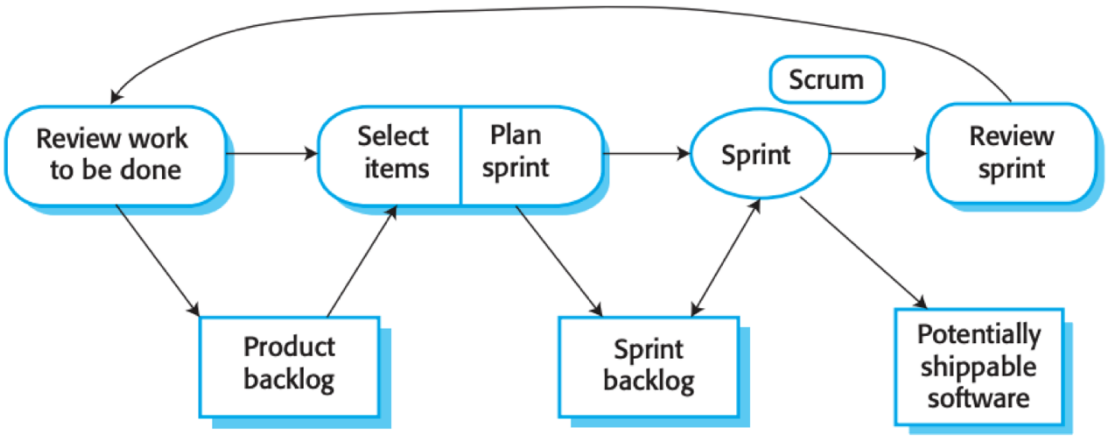


Figure 4. 2: *The Agile Scrum sprint cycle [45].*

The development of this project was planned by creating a feature backlog, which contained all the essential features that were derived from the requirements, mentioned in section *3. Requirement Specification*. The sprints were initially planned to last 2 weeks, with each sprint focusing on a set of goals. The features were selected from the product backlog for development, with new iterations released every 2 weeks. The initial sprint plan can be found in *Appendix C*. While the sprints had a fixed timeline of 2 weeks, the tasks within the sprints were flexible. The order of development of features changed throughout the lifecycle, and few features were refined based on the new insights during the development. The supervisor meetings were held at the beginning of each week, and discussions focused on what has been done, and what the plan was for the reminder of the week. This provided valuable direction and helped me keep up with the schedule. After each iteration, the features of the app were discussed with the supervisor, and suggestions were incorporated into making the features more effective. The app was tested thoroughly through every iteration, to ensure the features worked as expected.

Using agile enabled flexibility in the development process, allowing the development to take full advantage of the modular design of the application. Few features were revisited and modified in different sprints as new ideas emerged during the development. While using a plan-driven waterfall model would have constrained the modification of features later in the lifecycle, Agile allowed for continuous refinement of the features without disrupting the development of other features. The modular design, combined with agile also enabled the separation of concerns and rapid development of the app. Detailed design and implementation discussions are covered in the next chapters.

While most tasks were done in their respective sprints, few non-trivial tasks such as the report writing and weekly meetings with the supervisor were carried out throughout the software development lifecycle. Rapid feedback from the supervisor was crucial to the project's success.

4.2.1 Task Tracking

Managing a large project with numerous features requires keeping track of tasks to manage the milestones effectively. An online tool called Trello [48] was used to keep track of the tasks. Trello provides a Kanban board to organise the tasks into different categories such as 'To Do', 'Doing', and 'Done'. The product backlog with the features was also managed in this Trello board to prioritise appropriate tasks. **Figure 4. 3** shows a snapshot from the Trello board that was used during sprint 1 of the development of this project.

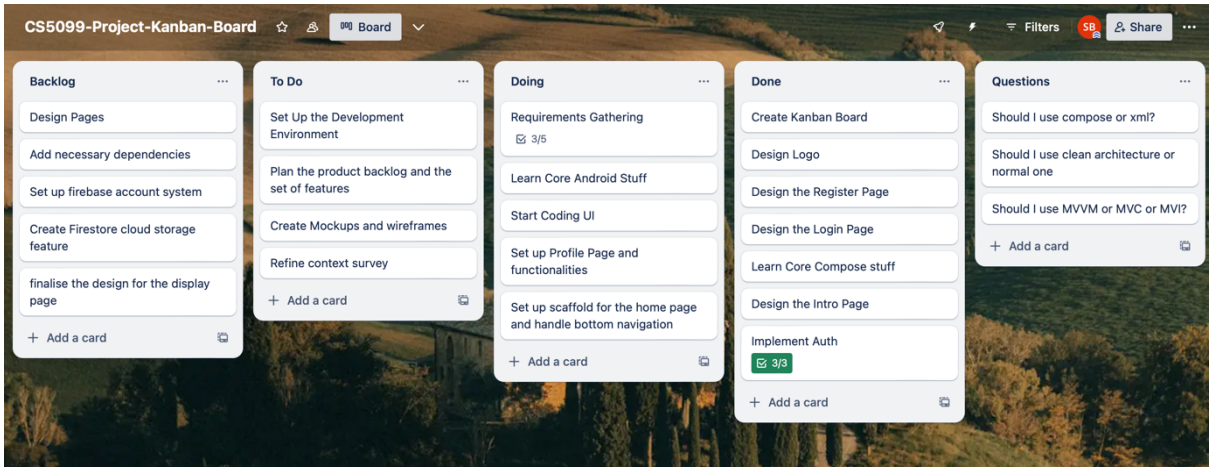


Figure 4. 3: *Kanban board during Sprint 1 which includes Sprint backlog.*

4.2.2 Version Control

While agile offers the flexibility, it is essential to prevent the changes from breaking the application. Hence, version control was used to keep track of all the changes that were being made to the app. Git [49] was chosen as the version control software for this project. The project made extensive use of branching to ensure the development did not interfere with the main code base, maintaining separation of concerns. Each branch was named based on the feature being developed. The changes were committed locally whenever a significant change was done to the codebase and at the end of each day to ensure that working code is saved. Once a feature was completed, it was pushed to a remote repository on GitHub [50]. Since GitHub is an open-source online version control platform, the repository was made private to ensure confidentiality and security. Pushing code to the remote repository to ensured that a working version of software is always stored remotely. This ensures that the development would not stop in case of a breakdown of the local machine.

5. Ethics

Ethical considerations and data privacy are important when it comes to the development of an application, particularly when it involves user interactions. This chapter covers the ethical considerations of this project. It discusses the ethical approval obtained from the university as well as the data privacy measures regarding the app.

5.1 Ethical Approval Process

The process began with the completion of 'Preliminary Self-Assessment Form' to determine what kind of ethics application was required for this project. Given the project's scope of working with Transition St Andrews, and potential end users, it was decided that a full ethics application would be necessary. Hence, a comprehensive full ethics application was prepared and submitted to the ethics committee, along with supervisor's approval. This application detailed the nature of interactions with the potential users and Transition St Andrews. The end-users were asked to test out the application and complete a questionnaire at the end of their testing. The results of the questionnaire were stored anonymously, and ensuring no personal information of the participants was collected.

The full ethics application has been submitted to MMS, and the ethical approval letter is included in the *Appendix B – Ethics Approval Letter*.

5.2 Data Privacy Considerations

To ensure data privacy during the testing phase, test accounts were created and were handed out to the users to test out the functionality of the application. This ensured no personal data of the users were collected during the testing phase. All the data related to the application was stored on the Google Firestore Database. For the prototype of the application, my personal Google account was used to manage the database. This ensured that the security of the database was not compromised.

The users were provided with an anonymous survey link to give feedback about the application. This survey form was designed not to collect any personal information of the user, ensuring that the feedback remained truly anonymous. In contrast, Transition was sent a different questionnaire to gather their feedback. Although this questionnaire was not anonymous, as feedback from Transition was a key objective of this project, no personal data was collected. This aligns with the ethical approval obtained for the app.

6. Design

This chapter discusses the design aspect of ‘Sharing Surplus’. Software design is an important aspect of the software development process. A well thought out software design is crucial to fulfil both functional and non-functional requirements of a software. Software design translates the users’ requirements into a ‘blueprint’ for building a software [51]. Since the project follows the standard coding practices, it is important to have a highly flexible and professional software design, that is backed by the standard practices in the industry. A good design ensures adequate modularity, decoupled modules, minimal duplicate code, and untangled dependencies. This allows the software to evolve without degrading over time.

In this chapter, discussions include the HCI aspects of the design, which guides the User Interface and User Experience (UI/UX) decisions made on the project. The software architecture, including the design principles used in building the project is also discussed.

6.1 Human Computer Interaction (HCI) Principles

Human Computer Interaction refers to the design of interactions between humans and machines [52]. It defines how the application should look like, how the workflow should be designed, and how the users can interact with the application. For this project, design guidelines from Google [53] were applied. Google follows a design framework called Material Design [54], which contains basic guidelines on how to UI/UX of the application should be designed. This includes everything a user sees on the screen, from size and shape of the components to the colour schemes and placement of these components. The current version of material design used by Google is Material 3 [54], which enables personal, adaptive, and expressive experiences for the user, which are customised for their devices. Since different mobile devices have different themes and settings, Material 3 makes use of these implicitly and gives the user a unique experience personalised for their device, from dynamic colours to enhanced accessibility. Since Android is an open-source software owned and maintained by Google [55], using the design guidelines from Google makes it an ideal choice to provide the best user experience. But strict adherence to it without considering core functionality can limit the advantages. Therefore, the guidelines were tweaked and modified as needed to suit specific situations encountered during the development.

A significant consideration in HCI is the adoption of Usability Heuristics by Jacob Nielsen [56]. This refers to a set of 10 general principles used in design that enhance the intuitiveness and natural feel of the application. These heuristics were applied to ensure a consistent and user-friendly experience.

Branding is highly important when it comes to an application. It is crucial in establishing an identity for the application. An effort to incorporate this principle to ‘Sharing Surplus’

was made, and it is reflected heavily in the UI design decisions. A unique logo and a tagline, reflecting the app's core functionality, was incorporated. These elements were designed to attract new users and to convey the application's purpose effectively.

6.2 User Interface (UI) Design

User interface is the point of interaction for a user with a device [57]. In case of mobile devices, it generally means the screen. Hence, it is extremely important to focus on creating sophisticated visual and interactive screens for the application. The domain of User Interface Design focuses on building the interfaces by prioritising looks or styles [58].

6.2.1 Colour Scheme

Since 'Sharing Surplus', is all about sharing the surplus produce, a light greenish colour (**#F1F6DA**) was chosen to be the primary colour of the application. This also was in line with the colour psychology, as the green family of colours symbolises a sense of safety, readiness, and eco-friendliness[59]. Green also puts less strain on the eyes. Google's Material Theme Builder [60]was used to choose the corresponding secondary and accent colours, as well as the corresponding fonts and text colours that adhered to the theme and Material Design Guidelines.

6.2.2 Logo

As a part of the UI design and branding, a logo was designed using Canva [61]. This logo (**Figure 6. 1**) depicts the agricultural background, which adheres to the app's core functionality of sharing surplus produce. It also includes a unique tagline '***From Garden to Table'*** that reinforces the app's purpose. This logo was inspired from the '***Green Vintage Agriculture and Farming Logo'*** by xvector [62]. This inspiration was taken from the 'Free templates' section of Canva, which allows complete freedom for personal use [63].

Figure 6. 1: *Logo of 'Sharing Surplus'.*

6.2.3 Wireframes

Once the logo and the theme of the application were ready, the next step in the design process was to create wireframes. A wireframe is a rough schematic created in the early stages of design to help communicate the flow of the system [64]. Initially, few high-fidelity wireframes were created using Figma to communicate the system flow [65] .A high-fidelity wireframe is the most realistic version of wireframe which resembles the final version of UI design of a product [66]. **Figure 6. 2** show the high-fidelity wireframes of the Authentication feature.

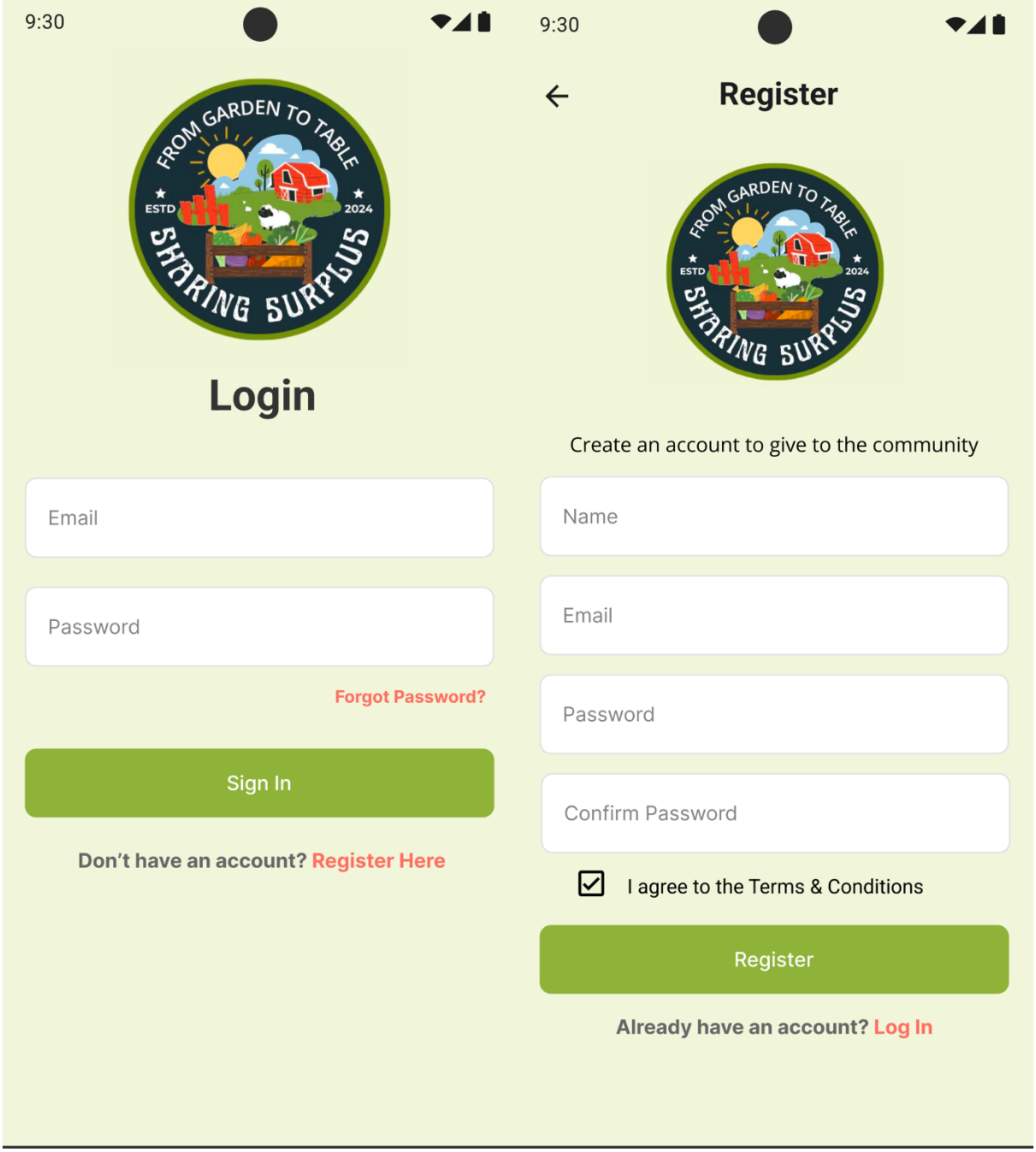


Figure 6. 2: *High Fidelity wireframes of the authentication feature.*

Once these initial wireframes were discussed with the supervisors, due to time constraints, the UI design was integrated with the development process. It meant that instead of creating wireframes in Figma, designs were implemented directly in Compose. These designs were iteratively refined with the supervisor's feedback. This ensured that once all the basic design structure was in place and avoided repetitive work, thus saving time.

The next section includes screenshots of different features of the application.

6.3 Featured Screens

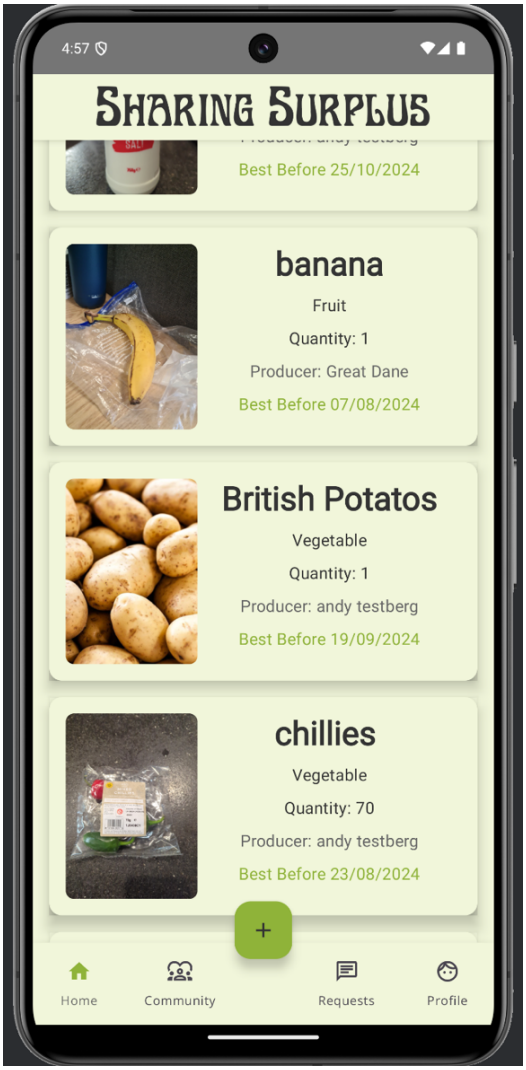


Figure 6. 3: *Home page of 'Sharing Surplus' listing the produce.*

Figure 6. 3 shows the home page of the application where different produces are listed. It is where the users can select a produce and request it from the producer.

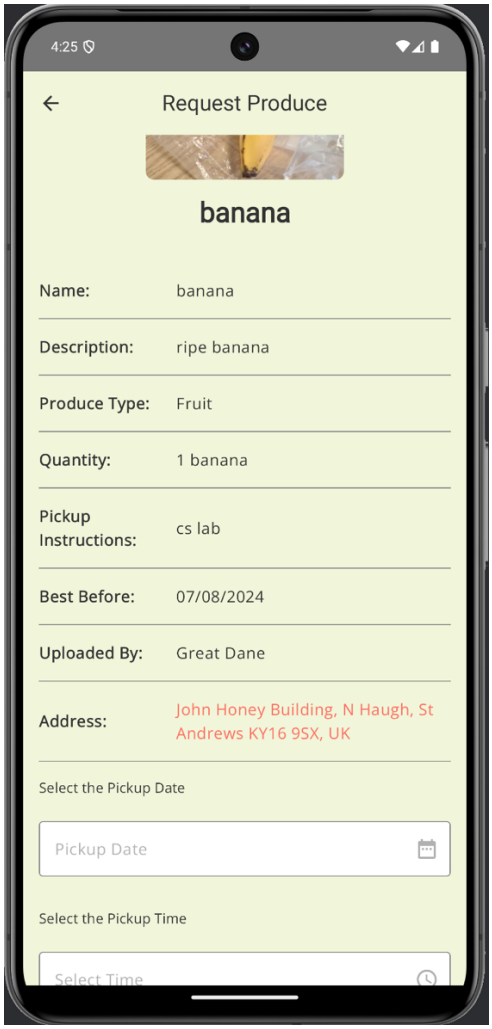


Figure 6. 4: *Produce Request Screen*

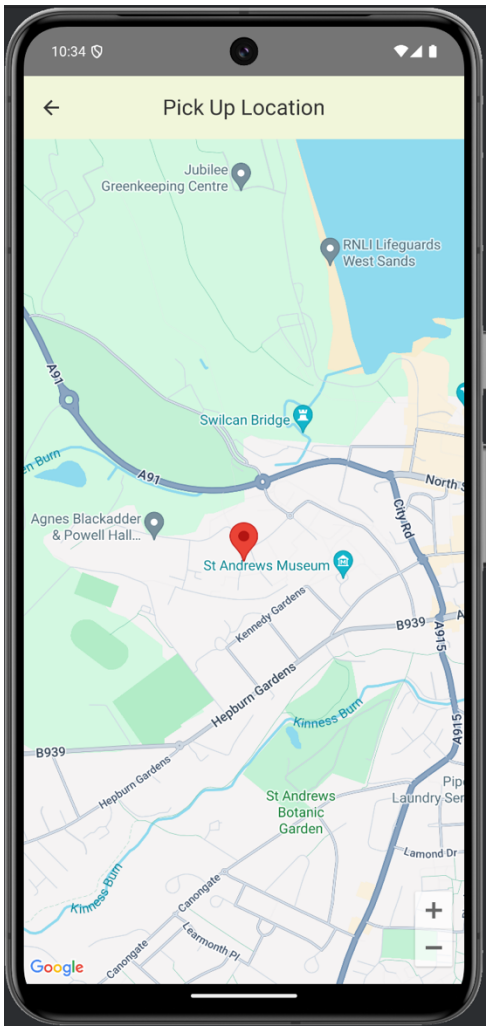


Figure 6. 5: Pickup Location Map

Figure 6.4 shows the produce request screen, where the users can specify the time, date and the quantity they want to request from the producer. They can also optionally specify any pickup instructions. **Figure 6.5** shows the map feature, which tells the users the exact pickup location of the produce. This can be either a neutral location proposed by the user, or the user's home location.

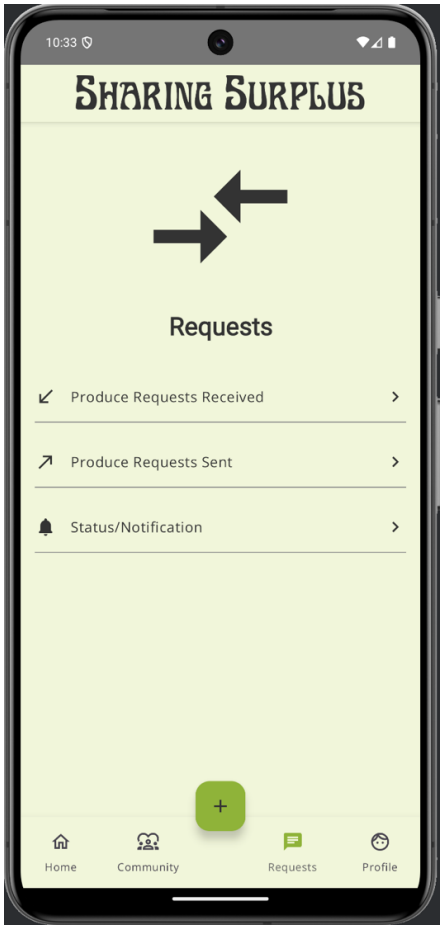


Figure 6. 6: The Requests Screen.

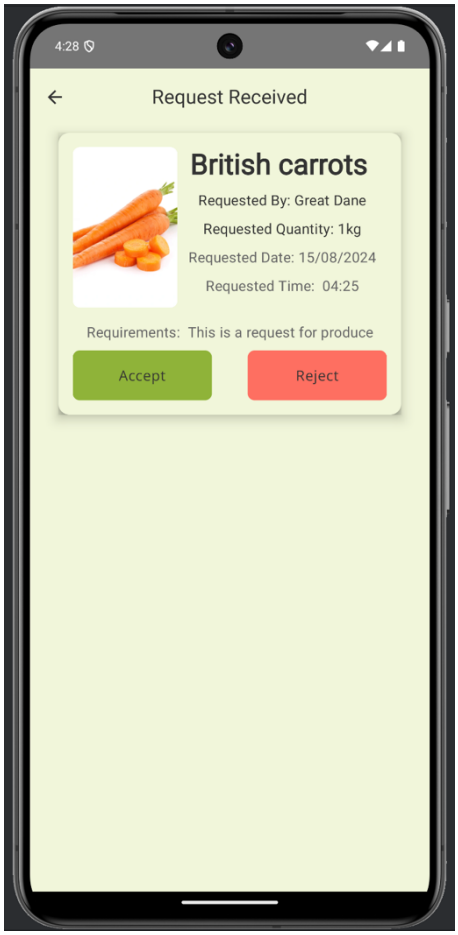


Figure 6. 7: Request Received Screen

Figure 6. 6 shows the requests home page, where a user can manage the requests. The request received screen shows the requests received for a produce that a user had posted. The request sent screen shows the requests made to produces posted by other users. Request status screen shows the status of the request, whether the user's request is accepted, rejected or pending. **Figure 6.7** shows a sample of request received page, where the user can accept or reject a request, citing reasons.

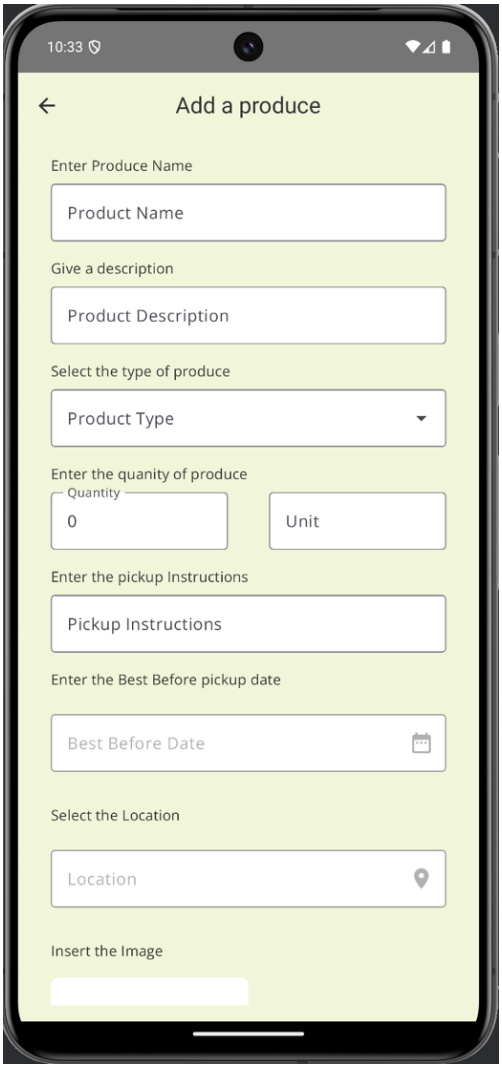


Figure 6. 8: *The Add Produce Screen.*

Figure 6. 8 shows the add produce screen. In this screen, the users can fill up all the details of the produce including the product type and quantity. The users can then select a pickup location of the product. The pickup location can be either a neutral location, or user's current home location. The user can also upload the photo of the produce, which can either be taken from the camera of the mobile phone or selected from the gallery.

Figure 6. 9: *The User Profile Screen.*

Figure 6. 9 shows the profile section of the application. The users can manage their profile details using this section. The users can also manage their produce using this section, including editing or deleting their posted produce.

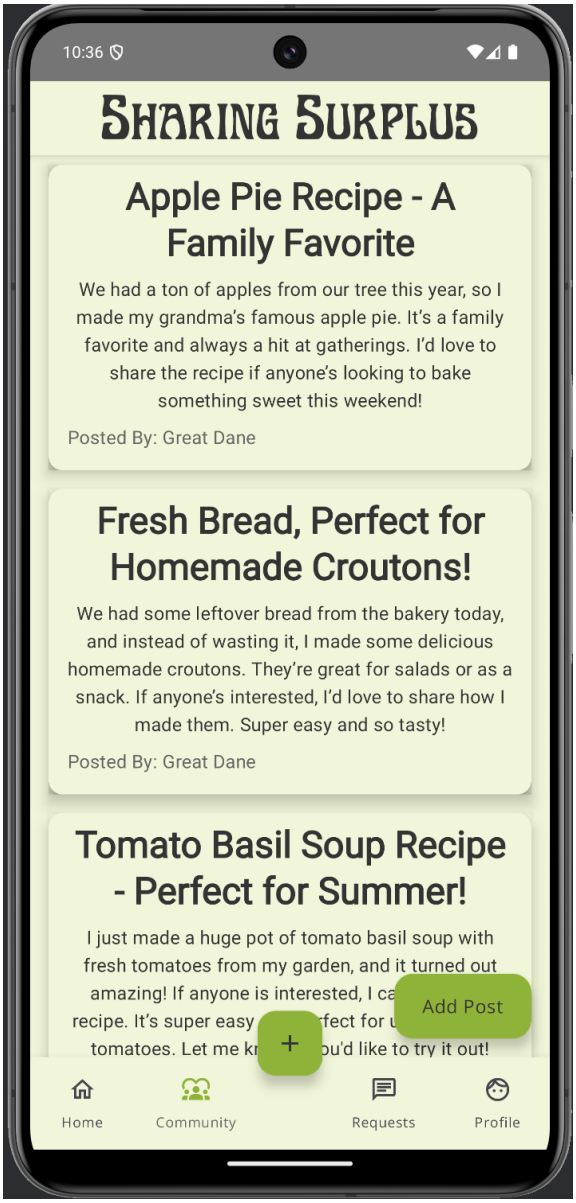


Figure 6. 10: *The Community Forum Screen.*

Figure 6. 10 shows the Community Forum, where the users can share their thoughts and post content visible to the community.

Figure 6. 11: *The About Us Screen.*

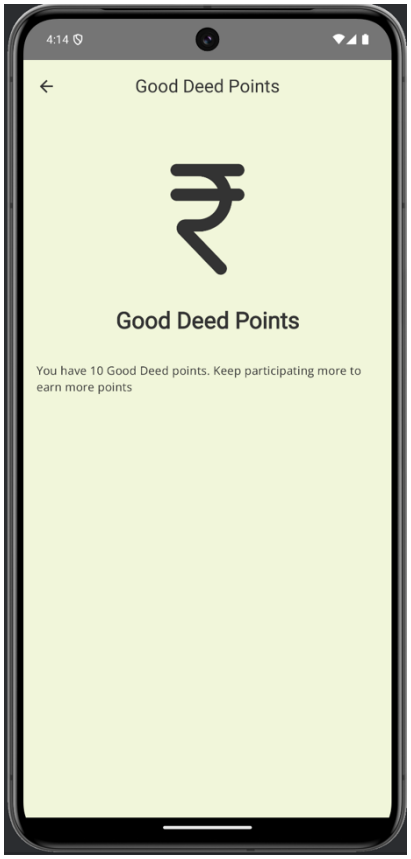


Figure 6. 12: *The 'Good Deeds Points' Page*

Figure 6.11 shows About Us page, which shows the details about the application. **Figure 6.12** shows a 'Good Deeds Points' page, where the user can view their accumulated points.

6.3.1 Design Decisions

Certain design decisions were made to ensure the application fulfilled its desired purpose.

- **Google Maps and Google Places Integration:**
  The decision to integrate Google Maps and Places API was done to ensure that producers could specify the pickup location, and the requesting users could look up the pickup location and decide whether it would be possible for them to go to that location. This visualisation gives greater freedom to the users to see exactly where the pickup location is, instead of just looking at the address, thereby providing greater flexibility and clarity.

- **Neutral Pickup Locations:**
  The decision to allow the users to select a neutral pickup location ensures that anonymity is maintained. This ensures the privacy of the producers, who might not want to have unknown people coming to their house to pick up the produce. Alternatively, few producers might also be more than happy to have people over their place. Hence the app allows the users to select whichever location they prefer.

- **Request Mechanism:**
  The decision to remove the ability to chat between two users is to ensure a streamlined process is followed while requesting the produce. The user can select the desired quantity, suitable date and time, and request the produce. The producer can meanwhile, accept the request or reject it, providing the reasons. The user can then make a fresh request if desired. This allows for a straightforward request mechanism, minimises risks and ensures streamlined interactions. Additionally, the app automatically updates the available quantity if the request is made for less than the posted quantity, ensuring robust produce request mechanism.

- **'Good Deed Points':**
  As the application doesn't offer any monetary benefits for participating in sharing, 'Good Deed Points' were introduced to motivate the users. Both producer and requester earn points accordingly for every exchange. Incentivising the users to share the surplus produce ensures continued user engagement and participation, thereby reducing the food waste. The *section 10.2 Future Work* outline potential expanses for this feature.

- **'First Come First Served':**
  To handle the situation of multiple users requesting the same produce simultaneously, a 'First Come First Served' mechanism was implemented. This ensures fairness by prioritising the first requester and removing the produce from the listing once requested. This system prevents race-around conditions and ensures an unbiased, fair sharing process.

6.4 User Experience (UX) Considerations

User Experience is crucial as it ensures a smooth and engaging flow within the application. A well-designed user experience makes the users get involved in the application and makes them feel valued. To provide an optimal user experience, the app was designed keeping the UX goals [67] in mind.

These UX goals are:

1. Always aim to satisfy the users.
2. Make the product as easy to use as possible.
3. Design it for all users.
4. Make it inviting.
5. Reduce the work for the users.
6. Stop hiding things.

These goals help the app to stay relevant and provide a natural, intuitive experience to the users.

6.5 System Architecture

Software architecture deals with design decisions related to the overall structure and behaviour of a system [68]. It helps set out a blueprint for the software development and helps communicate key design decisions with the stakeholders. Due to the space and processing constraints of a mobile device, it is essential to design the application to ensure optimal performances. For this application, Google's guidelines on application architecture [69] were used.

6.5.1 Clean Architecture

In his blog [70] on Clean Architecture, Robert C Martin emphasises the principle of Separation of Concerns. In this blog, Martin aims to integrate various architectures which emphasise this principle into a unified approach. The core idea of this architecture is that a developer should be able to understand the software's functionality looking at the source code. Everything else, including programming languages, frameworks and libraries are deemed obsolete [71].

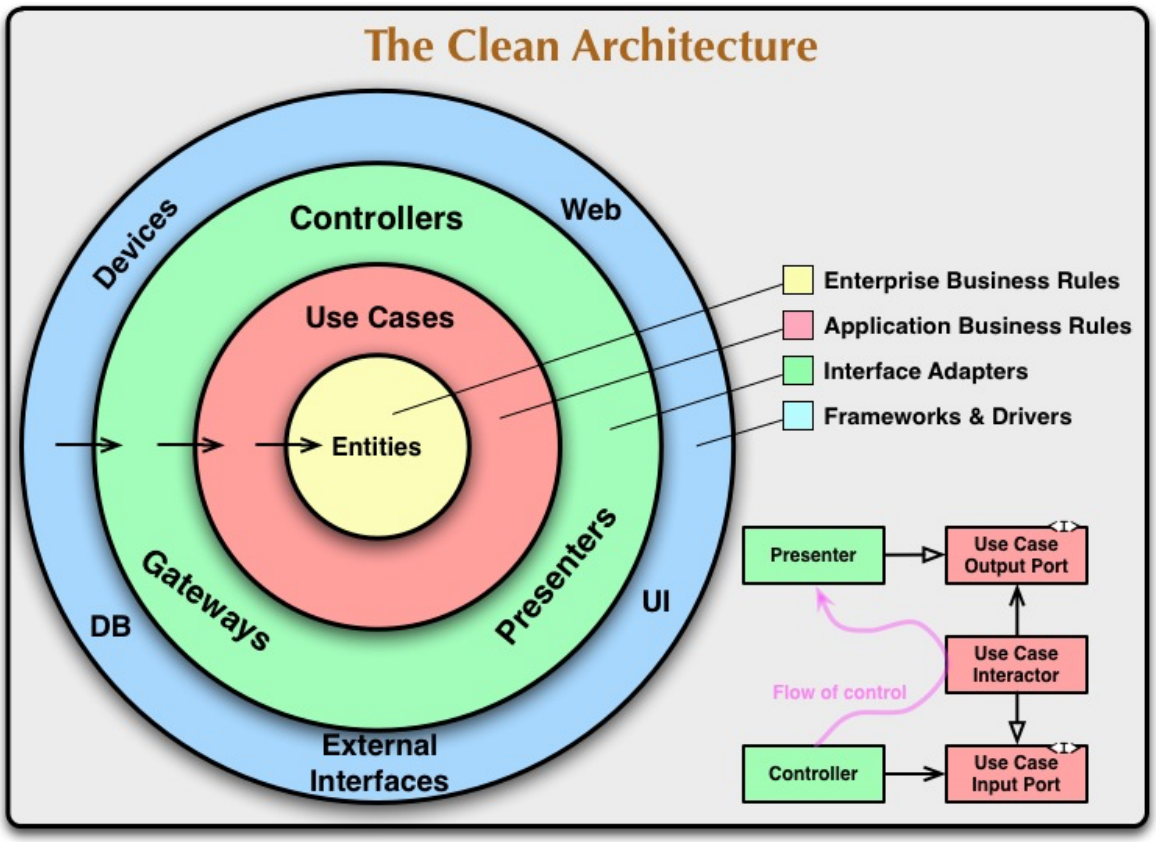


Figure 6. 13: *The Clean Architecture [70].*

Figure 6. 13 shows the diagrammatic representation of the clean architecture. As core idea of separation of concern[70], the architecture is structured in the form of multiple layers. The inner layers focus more on the abstractions, while higher layers focus more on the implementation details [72]. This separates the business logic of the application from the implementing technologies, allowing for a modular design of the application. This can ensure that the architecture is easily evolvable, as the implementation technologies can be replaced with newer ones over time, without affecting the core functionality of the application.

For 'Sharing Surplus', the application has been divided into 3 main layers.

- **Data Layer:**
  This layer deals with the data aspect of the application. It handles all the data level implementations, including Firebase connections as well as all the CRUD (Create, Read, Update, Delete) operations. It also includes the UI state definitions as well as repository implementations for Firebase Authentication, Firebase Cloud Storage and Firebase Firestore DB.

- **Domain Layer**
  This layer deals with business logic part of the application. It manages all the dependency injection modules as well as the state management. It calls the

functions from the data layers, without concerning itself about how they are implemented.

- **Presentation Layer**
  This layer deals with the presentation aspects of the application. It manages the UI screens and their display content. This layer does not store any state, as all the state related aspects are handled by the business logic.

6.5.2 Model View View-Model (MVVM)

To implement the clean architecture in an application, the most common design pattern used is the MVVM. MVVM stands for Model, View, View-Model. It helps in cleanly separating the business and presentation logic from the UI [73]. **Figure 6. 14** shows the relationship between Model, View and View-Model

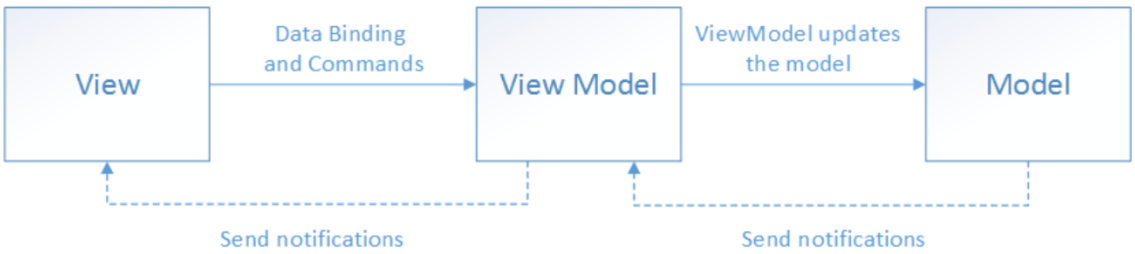


Figure 6. 14: *The Model View View-Model (MVVM) design pattern [73].*

In the context of 'Sharing Surplus', the *Model* represents the data and domain logic of the application. It encapsulates the core business logic and rules of the application and is unaware of the UI. The *View-Model* acts as the bridge between the *Model* and the *View*. It handles the UI-related logic and maintains the states of UI. It communicates with the model to fetch, manipulate, and update the data, while ensuring that the View remains unaware of the business logic. The *View* is responsible for the UI that is visible on the screen. It takes in the data from the *View Model* and displays it on the screen. It doesn't perform any operations on the data.

This application extensively utilises of MVVM pattern to implement the Clean Architecture, associating each screen and views with a View Model to handle the communication with different parts of the application. This is detailed in the section *7. Implementation*

6.5.3 Modular design

The combination of Clean Architecture with MVVM gives rise to a modular design. Modularisation is the practice of organising a codebase into loosely coupled and self-contained parts [74]. This means separating the codebase based on different modules and keeping them loosely coupled from the rest of the code. In this project, the modules

were chosen to be the core features and they were developed independently. The modules contained authentication, request handling, profile management, produce handling, etc. These modules had their own internal MVVM structure and did not directly communicate with the other modules. This ensured that features would be developed independently and did not affect the other modules and their development. This also ensured that issues were contained to a particular module and did not affect others. However, few common elements were to be used throughout the application. Using them normally would inter-twin the modules and therefore increase the coupling. To ensure loose coupling, dependency injection was used. Dependency Injection involves suppling the functionality and services of one module to another module [75]. This was done by Dagger Hilt [76], which is a dependency injecting mechanism prescribed by Google that is optimised for android applications.

6.5.4 Design principles and Patterns

To build an android application, that meets all the functional and non-functional requirements, a strong foundation backed by design patterns is essential. Design patterns are typical and widely accepted solutions to commonly occurring problems [77]. Google, in its '***Guide to App Architecture***' [69], gives guidelines on the design patterns to be used while building an android application. This allows the app to be scalable and responsive. The prescribed design patterns, and how they are implemented are described below.

- **Separation of Concerns:**
  This design principle ensures that the UI logic and the business logic of the application are separated. The app ensures this by adapting Clean architecture along with MVVM.

- **Drive UI from data models:**
  This design principle ensures that UI itself doesn't hold any data, instead it is driven by the Models. The adoption of View Models ensures that the UI elements don't store any data, instead are driven by the Models.

- **Single Source of Truth:**
  This design principle ensures that only a single source of truth exists throughout the application. The dependency injection implemented in this application ensures that the repositories are initialised once and injected wherever needed. This, in-turn, ensures that there is only one source of truth in the application.

- **Unidirectional Data Flow:**
  This design pattern ensures that the data flows in a single direction only. Since this application makes use of MVVM, the state flows in one direction, and the events that modify the data flow in the opposite direction. This ensures that the data always flows from the higher-scoped types of the hierarchy to the lower-scoped ones, while the events flow from lower-scoped to higher-scoped ones.

6.5.5 Tech Stack

A technology stack refers to the set of technologies that used in building an application [78]. This includes a combination of programming languages, frameworks, tools, and technologies that are required to implement the front-end, back-end, and the database aspects of the application. 'Sharing Surplus' makes use of the latest technology stack used in the industry by professional android developers [79]. **Figure 6. 15** illustrates the tech stack used for this application.

Figure 6. 15: *The Android App Development Tech Stack.*

This mostly comprises of Google's or Google supported products, that ease the integration of different and provide a platform for an optimal application development process. This tech stack includes:

- **Kotlin:**
  Kotlin is a programming language developed by Jet Brains and is preferred as the language of choice for Android App Development by Google [80]. It is designed to be concise, expressive, and safe. In this application, Kotlin is used for implementation of both the front-end of the application, as well as the back end of the application.

- **Android Studio:**
  Android Studio is the official Integrated Development Environment (IDE) for Android App Development [81]. This allows the developers to properly develop, test, and package android applications. It comes with library integration as well as an emulator to test out the application.

- **Android SDK:**

Android Software Development Kit (SDK) is a collection of comprehensive set of development tools that enables a developer to build apps in the Android software ecosystem [82]. It contains tools, libraries, documentation needed to build and package an android application.

- **Jetpack Compose:**
  Jetpack Compose is Android's recommended framework for building native UI [83]. It simplifies and accelerates UI development on Android and ensures a single language is used to develop both front end and the back end.

- **Google Firebase Firestore:**
  Firebase Firestore is Google's flexible, and scalable NoSQL cloud database, built on Google Cloud infrastructure [84]. It is used to store and manage the data of the application.

The implementation details of the above are discussed in the Implementation section.

7. Implementation

This chapter describes the implementation details of 'Sharing Surplus'. After the selecting the tech stack and finalising the design decisions, the next step in the development process was the implementation. This phase involves translating the design decisions into a functional application. The implementation phase was conducted iteratively, incorporating the feedback from the supervisor at every step.

7.1 Development Environment

Once the tech stack was chosen, the development environment was set up. This included configuring Android Studio IDE [81] along with all the necessary libraries and dependencies. Gradle [85]was chosen to be the build management system for this app. Gradle helps in automating the development process by managing dependencies, and packing the application into an executable format that can run on a mobile phone. Following Google's guidelines, Kotlin KTX [86] was chosen over the traditional Groovy [87] for Gradle scripts. This was done to maintain consistency in using Kotlin throughout the software development process.

Once the IDE was ready, all the external libraries, including Hilt, Firebase, Coil etc. were set up and were ensured to be running the latest versions. All the necessary permissions like Camera and Location, etc. were specified in the AndroidManifest.xml file.

7.2 Architecture Implementation

As discussed in the System Architectural Design section, *6.5 System Architecture*, Clean Architecture and MVVM pattern were adopted for this project. Adhering to that, the entire codebase was divided into Data, Domain and Presentation sections, with each section further divided into corresponding feature. **Figure 7. 1** shows the file structure of the application. It illustrates a modular structure that ensures separation of concerns, enhancing readability and maintainability.

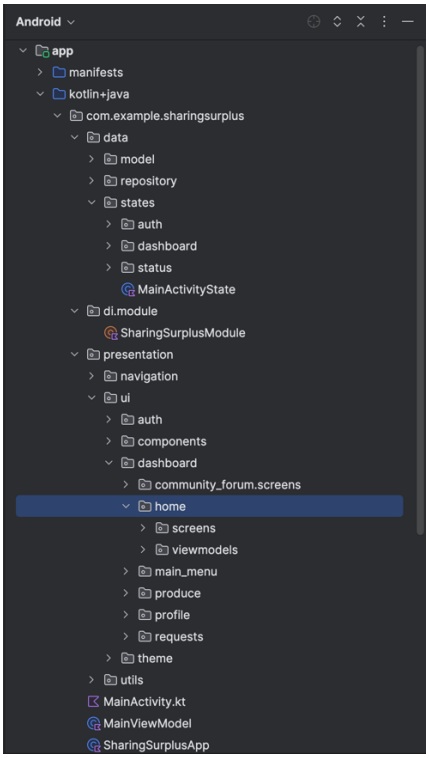


Figure 7. 1: *'Sharing Surplus' file structure organisation.*

This file structure adheres to the Clean Architecture principles, which emphasises on promoting readability, and minimising the impact of changes in one section on other sections.

7.2.1 Data Layer Implementation

The data layer includes all the data related aspects of the application. It includes multiple packages. The Model package contains all the Data Models for Users, Produces and Requests. These models define the necessary fields of information for each entity stored in the database. To maintain concurrency, no local database was used, and all the data relating to the application are accessed only when the device is connected to the internet. This ensures consistency across devices and adheres to the Single Source of Truth principle, avoiding concurrency issues with the local databases.

As discussed in the Design Section, *6. Design*, Google's Firebase was chosen to be the database. Interfaces were created for each repository type and the appropriate CRUD functions were defined, which were implemented in the corresponding classes. The authentication repository handles the user registration and login, while Firestore repository handles all the produce and request actions, such as uploading a produce, requesting a produce, responding to the request etc. The images are stored in the Google cloud storage, with their URLs saved in the Firestore database. The storage repository manages the image uploads. This separation based on Firebase services enhances the flexibility and readability of the application, while aligning to the principles of Clean Architecture. The data layer also contains UI state models, which define all the fields that are to be displayed on different screens.

7.2.2 Domain Layer Implementation

The domain layer encompasses of application's business logic. This includes the definitions of the databases, and the repositories. These definitions establish the Single Source of Truth principle, ensuring the consistency amongst entity instances throughout the application. These entities are then injected to the corresponding classes that require them. Dagger Hilt [76] takes care of injecting the dependencies to different classes. Hilt ensures that there is a single instance definition for the entities in the domain module. Changes made to the instances propagate throughout the application, which maintains consistency. ViewModels communicate with the domain layer to perform actions related to the application. Hilt injects the dependencies to the View Models, which then calls the methods provided by the instances. View Models maintain the state of the screens, which frees up the presentation layer from of any business logic.

7.2.3 Presentation Layer Implementation

The presentation layer includes all the elements a user sees on the screen. Jetpack Compose [83], Google's preferred UI framework for Android, was used to implement this layer. Compose allows building a declarative UI using Kotlin, ensuring consistent language use for both the business logic and the UI of the application. In imperative style of UI declaration, the UI is declared exactly how it is supposed to be and is mutated whenever the data changed. This has been causing a lot of causing of manually updating the UI elements whenever the data changes. It also burdens the developers while transitioning between various UI states [88]. The declarative UI by compose allows the developer to describe how the UI is supposed to look rather than what it is supposed to show. This includes building lightweight, non-mutable 'Composable' [89], which take in a state and display it. Whenever the state changes, the Composable is recomposed with the new data, as opposed to mutating the same object. This allows the app development to be more focused on handling the business logic and states, rather than worrying about the UI. The UI elements are stateless; hence they don't store any data. Each screen is linked to a View Model, which is handles the state management of that particular screen. Whenever the state is modified, the Composables are automatically recomposed thereby displaying new data. This ensures more time is spent on the making the functionalities of the application better rather than tweaking the UI. The stateless Composables and the state-full view-models help in implementing the MVVM pattern.

7.3 Firebase

Firebase is Google's recommended cloud services for android applications [90]. It provides powerful backend services hosted on cloud, eliminating the necessity of having a separate server. This allows the android applications to have a serverless experience, which accelerated the development and deployment.

7.3.1 Firebase Authentication

Firebase Authentication provides the necessary backend services and SDKs to authenticate a user to the application [91], eliminating the need to manage user account details . It provides multiple authentication mechanisms, such as passwords, phone numbers, OAuth, etc. For this application, the authentication mechanism used is the email and password. This traditional way of authentication ensures that users of all age group are familiar with the login and registration process, thereby enhancing the accessibility. The users create an account using their email and password, and a unique profile is created and is stored securely by Google. Google ensures confidentiality and security of the data. Each user is assigned a unique UUID, which can be used to associate a user with the profile. Firebase ensures that only authenticated users can access Firebase services, removing the necessity to have repeated authentication checks.

7.3.2 Firebase Firestore Database

Firestore Database is the NoSQL Cloud database, used to store and sync data between multiple devices using the application [84]. It provides a consistent and synchronous experience for the users, with data updated across all devices in real time. Firestore also allows offline caching of data, allowing the users to see the produces while their device is offline. Firestore works by implementing a NoSQL database on the cloud. The data is stored in form of documents, that contain fields mapping to values. These documents are stored in a collection, which acts as a container for the documents.

The documents allow the flexibility of storage, by supporting different data types from strings and numbers to complex, nested objects.

‘Sharing Surplus’ uses four main collections for storing the app data: Users, Produce, Requests, and Forum. Each collection stores documents with relevant information such as user profiles, produce details, request statuses and forum posts.

Images are stored in Google Cloud storage, with the URLs saved in Firestore. This is a standard practice followed to avoid the slowdowns in database access. **Figure 7. 2** shows the database collections at an early stage of development.

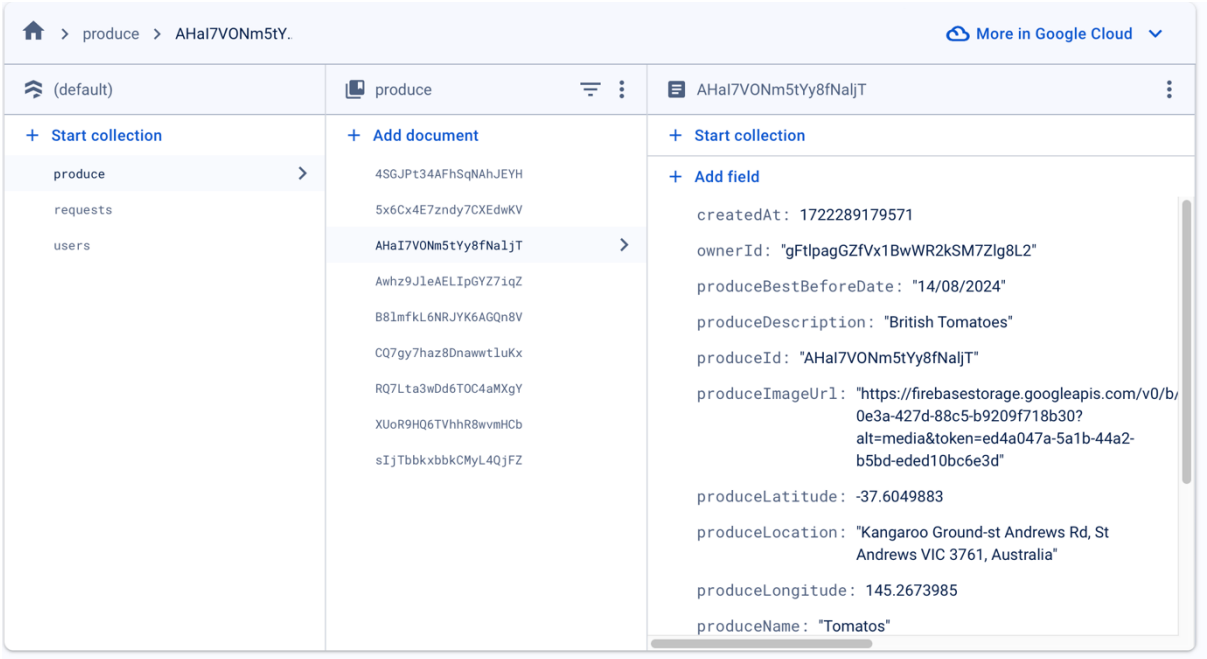


Figure 7. 2: *Firestore Database Collections.*

7.3.3 Firebase Cloud Storage

Cloud Storage allows the users to store and server user-generated content, such as photos and videos, using Google cloud infrastructure [92]. In ‘Sharing Surplus’, the user uploads their produce image to the Cloud storage and get back the corresponding URL for the image, which is stored in the Firestore database along with other information about the produce. Libraries like Coil [93] work directly with the Cloud storage to request and display the images in the application.

8. Testing

Testing is intended to find out if an application does what it is intended to do. It ensures that the application functions correctly, securely, and meets stakeholders' needs, ultimately provides value to the end users [94]. It also helps the developers to identify defects and flaw early in the development process. Throughout the development lifecycle of 'Sharing Surplus', various testing methodologies were employed to ensure the robustness of the application. Sommerville categories testing into two main types [45]:

- **Validation Testing:** This verifies if the system is performing correctly using a set of test cases that reflect the system's expected usage.
- **Defect Testing:** This aims to expose the underlying defects within the system.

'Sharing Surplus' utilises both validation and defect testing at every stage of development. Given the application's adherence to Agile methodology, testing was carried out at every iteration, as detailed in the development schedule in the *Appendix C*.

8.1 Testing Strategy

In alignment with the Agile methodologies, 'Sharing Surplus' adopted the Agile Testing Pyramid [95]. It includes various types of tests carried out throughout the development lifecycle of an application. These testing methodologies are prioritised based on their importance and the cost of fixing bugs at different stages. Detecting a bug at unit testing is more cost effective and manageable than finding it during manual tests. **Figure 8. 1** depicts the Agile Testing Pyramid.

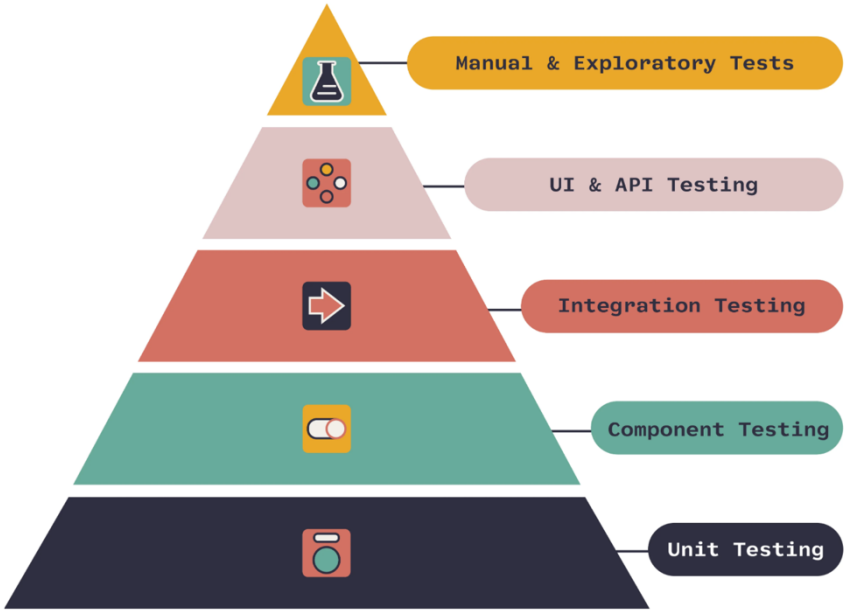


Figure 8. 1: *The Agile Testing Pyramid [95].*

This application also ensures that the comprehensive the testing guidelines from Google for Android Applications [96] is followed. This guideline details the fundamentals of testing android applications to verify its functionality and features.
Figure 8. 2 shows the different tests scopes in a typical android application.

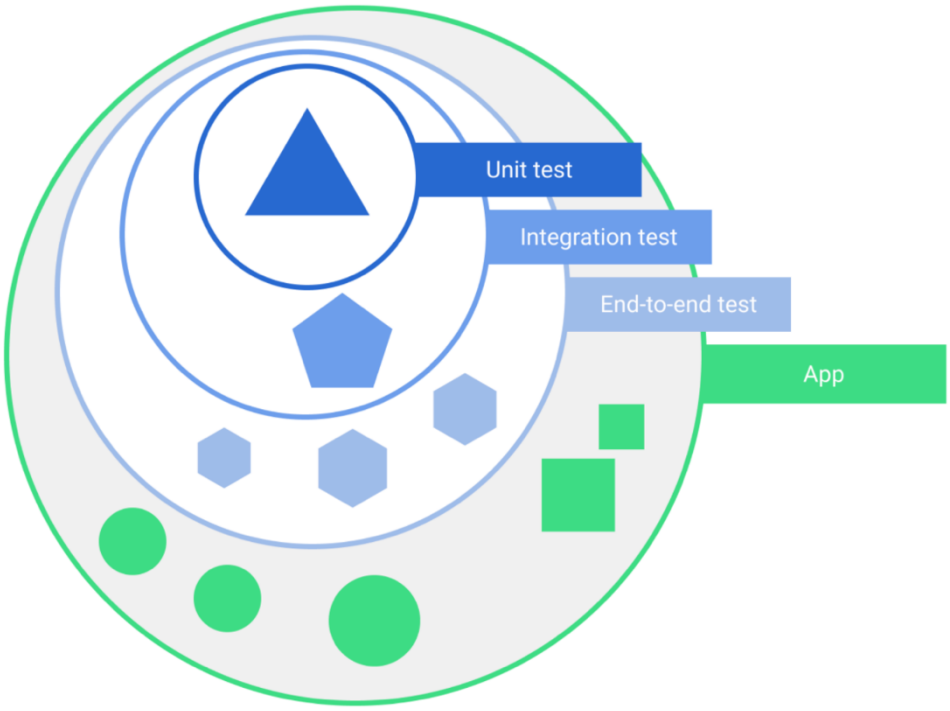


Figure 8. 2: *Test Scopes of a typical Android Application [97].*

In 'Sharing Surplus', Unit, Integration, and Manual testing were carried out extensively.

8.2 Unit Testing

Sommerville [45] defines Unit testing as a process of testing individual program components, such as methods or objects. In 'Sharing Surplus', the decision was taken early in the development process to prioritise testing out methods that implement the business logic of the application to ensure that the core functionality performed as expected. This decision was taken to minimise unnecessary time being spent on testing the less critical parts of the application. According to Google's guidelines on what to test in Android[97], unit testing should focus on ViewModels, since it houses most of the business logic of the application, and the data layer, with particular focus on the repositories.

Junit [98] was used to unit test all the methods of the view models

8.2.1 JUnit

JUnit is a popular automated testing framework for Java and Java Virtual Machine (JVM) languages, including Kotlin [80]. JUnit4, the recommended stable version of the

framework was used [99]. MockK [100], the mocking library for Kotlin, was used alongside JUnit to mock the database calls, which simulated real interaction. This was done to limit the incurring costs of an actual Firebase call. Mocking ensured that unit testing was carried out effectively and efficiently. **Figure 8. 3** shows an instance of unit testing being carried out in the application.

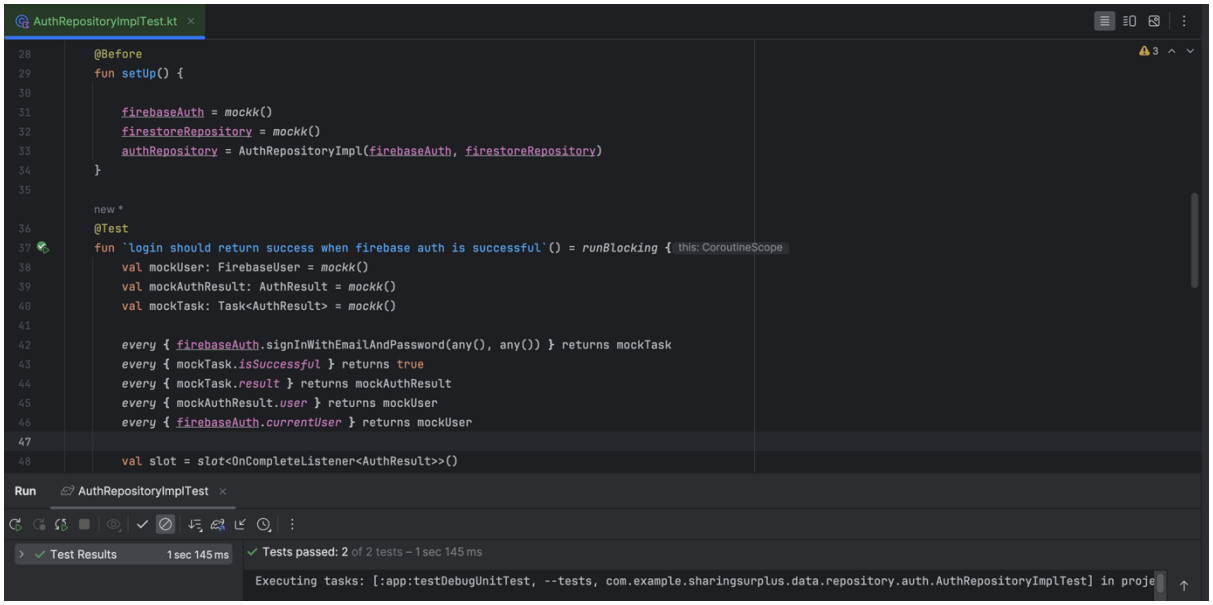


Figure 8. 3: *Unit Test for Auth Repository Implementation.*

8.3 Integration Testing

Integration testing is the next step from unit testing. It involves testing larger units, which are formed combining smaller units. This step is to ensure that the individually working methods functioned correctly when integrated.

In the development process of the application, methods that involved interactions with other methods were tested out either manually or with unit tests using Mockito to mock the interactions.

8.4 Manual Testing

Manual testing, positioned at the top of the agile testing pyramid [95], is the most expensive form of testing, as compared to other methods. However, manual testing is crucial to uncover errors that the automated tests might have missed. This involves stress-testing the application through manual workflows, which verifies the functionality in a real-world scenario.

Throughout the development process, manual testing was carried out by installing the application on actual devices to ensure desired functionality. In the later stages of development, potential users were recruited to test out the application and provide feedback. This feedback processed is discussed in detail in section *9. Evaluation*.

9. Evaluation

This chapter details the evaluation of 'Sharing Surplus'. Software Evaluation is a systematic and comprehensive process carried out to assess an application's suitability for its intended purpose [101]. This chapter discusses the user study that was carried out, the results of the study, and a final comparison with other similar solutions that were discussed earlier in *2.3 Related Work*.

9.1 User Evaluation

The main form of evaluation conducted was the user study. A user study is a type of research and evaluation method where the target audience try out the application and their opinions are collected [102]. This method helps in understanding how well the application meets the user expectations and identifies any user desired changes.

9.1.1 Ethical Considerations

The details about the ethical considerations for the application are detailed in the section *5. Ethics*. A full ethics application was made, which is attached in the *Appendix B – Ethics Approval Letter*.

9.1.2 Participants

Volunteers were recruited from the university by emailing the School of Computer Science's mailing list for PGT, and PGR students. Additionally, given the full ethical approval for the study, few participants external to the university were also recruited through word of mouth. This recruitment procedure ensured that the potential users interested in sustainable initiatives were included, which provided meaningful feedback. A total of 10 participants were recruited. They acted as both producers (uploading a produce), and requesters (requesting a produce). The procedure of the user study is detailed below.

9.1.3 Procedure

The evaluation was carried out using two methods to accommodate user requests. The primary method was sending out an email with the app's APK file to the users. The users would then install the application on their phone and test it out at their convenience. Several dummy test accounts were created for the users to test out the application. This ensured that the users avoided sharing their personal details during the account creation. The participants interacted with the application, tested out features, and put the app under stress. After the testing, they filled out an anonymous questionnaire, as required by the Ethics Application. This ensured the users would be able to give unbiased feedback for the application.

To accommodate the users of non-android phones into the study, an in-person user study was conducted at the School of Computer Science. These users tested out the application on android smartphones provided to them and then completed the same anonymous questionnaire.

In both methods, the participants were given a Participant Information Sheet (PIS), which detailed study procedures. They were then asked to fill out a consent form. Each study session typically lasted 15-30 mins. The in-person study session took around 20-30 mins, while the users testing at their convenience might have spent more time.

Additionally, no pre-requisites or instructions were given to the participants before using the application. This ensured that the feedback was unbiased. This approach also assessed that the intuitive user interface design.

9.1.4 Results

Responses were collected via an anonymous survey sent to the participants. Overall, the participants found the app easy to use and encountered no significant challenges. The design and features of the app were well received, with a few minor shortcomings identified and suggested for improvements. Most suggestions involved minor feature tweaks and were promptly addressed. The other suggestions have been earmarked for future updates. Participants generally appreciated the app's potential impact and regarded it as a valuable community initiative for reducing food waste. Key responses to the survey are outlined below:

1. **Before using the Sharing Surplus platform, how interested were you in participating in food sharing initiatives?**

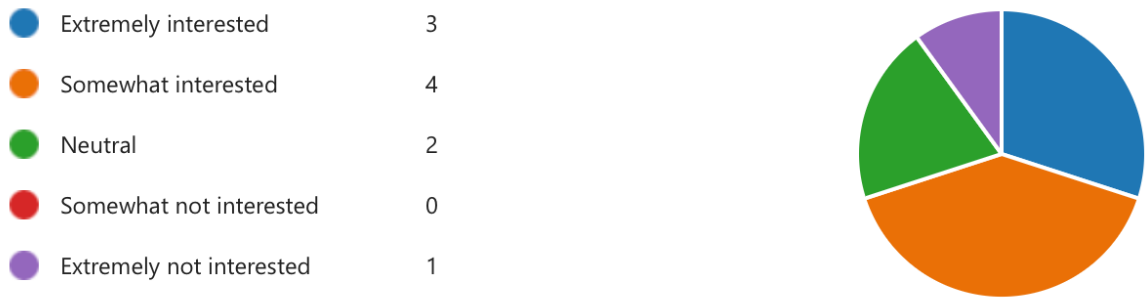


Figure 9. 1: *Participants response towards their interest in food sharing initiatives before using 'Sharing Surplus'*

2. **After using the Sharing Surplus platform, how interested are you in participating in food sharing initiatives?**

Figure 9. 2: *Participants response towards their interest in food sharing initiatives after 'Sharing Surplus'*

Insight: Most users' interest in participating in food sharing initiatives increased positively after using 'Sharing Surplus'.

3. **Did you find the Sharing Surplus platform easy to navigate? Why or why not?**
   Feedback: Nearly all users found the app easy to navigate. Some unique responses included:

   - "Overall, yes. Navigating the app is simple and user-friendly, especially with the four main pages in the navigation bar being very clear. The only issue I personally find a bit troublesome is when I want to view the three options under the requests section; I have to repeatedly enter and exit each option to switch between them."
   - "Intuitive layout with the navigation bar always available, not too many clicks required to perform actions which is very nice, not too many navigation panes to choose from, simple but also has all the functionality that one would expect from this type of application".
   - "Yes, As the UI Simple and easy to navigate and understand with all the functionalities."
   - "The navigation seems to be intuitive. Just did not exactly understand the purpose of the community feature."

4. **Did you find the Sharing Surplus platform aesthetically pleasing? Why or why not?**
   Feedback: While many users appreciated the UI, some felt it could be improved. Notable responses included:

   - "Functionally it's alright. Few things I noticed about the UI: 1) card shadows look weird. 2) The title of the app on the app bar could have had a better font. 3) Text on buttons has very less contrast, should have used bold and white text probably."
   - "Yes, I think it looks aesthetically pleasing, especially with the color combinations being very appealing."

- "Yes, the layout is very intuitive and easy to understand. However, brief descriptions could be included at the top of every section (especially for the Community section as I did not know what this section was really for) to describe what the section is for."
- "The colours are easy on the eye and contrast is also good."
- "Somewhat"
- "No, because the UI could've been much better".

5. **Were there any aspects of the Sharing Surplus platform you found inaccessible? If yes, please specify which aspects.**
   Feedback: Most users found the app accessible, though a few suggestions were offered:

   - "One issue is the three options under "Request" mentioned above."
   - "I granted location permission but still was not allowed to use my current location for the pickup location when posting a new food to be shared. Also, past dates can be selected when sending a pick-up request which does not make sense."

   Note: The second suggestion was corrected immediately, and the bug was sorted.

6. **How did you find the process of sharing surplus produce on the platform? Were there any challenges or barriers you encountered?**
   Feedback: Most participants did not encounter significant barriers in using the application, though some did report a few challenges. Critical issues were addressed promptly, while others have been noted for the future updates. Notable challenges included:

   - "I can book dates before the current date."
   - "Initially encountered an issue with the location permissions. Guess the page was not refreshing at a faster rate."
   - "One is that when I post my own products, I would like to see them directly on the main page with a marker to distinguish mine from others. Also, in the forum, I initially thought the "post" button was the plus sign in the centre at the bottom, but that button is for posting food. The "post" button for the forum is in the top right, which was a bit confusing. Lastly, it would be helpful if the "Request" section had badge notifications for updates, as I often don't realize when there are changes to my requests."
   - "If I did not fill in all fields it would initially tell me that I had to fill everything in, however, after trying to submit a couple of times without filling everything in, the warning message did not appear anymore, but my post also was not successful. Also, it would be nice if users could see their own produce that they have posted".

7. **How did you feel about the way information is presented on the Sharing Surplus platform?**

Feedback: Most participants felt the information was well presented. Notable responses included:

- "I think the way the foods are displayed on the home page is simple and aesthetically pleasing. If a filter feature could be added, it might help me quickly find the foods I want."
- "Descriptions could be included of the different sections to tell users what the purpose of the section is. This is specifically true for the "community" section which I did not initially understand."
- "It gives detailed information of surplus produce".

8. **Did the layout of the Sharing Surplus platform make sense to you? Why or why not?**
   Feedback: There was a general agreement amongst the participants the application's layout was logical. Responses include:

   - "Yes, it's simple and not overly complicated."
   - "Yes - the main page with the feed is a good way to keep users interested and interacting with the platform."
   - "Yes, as things were structured."
   - "Yes, it makes sense. It gives a platform to the people who want a particular produce."

9. **Did you have a particular favourite part of your user experience on the Sharing Surplus platform? Can you tell me about your experience with using the Sharing Surplus platform? Do you think the initiative is helpful in sharing the surplus produce?**
   Feedback: Participants generally had a positive and felt the initiative was helpful. Some of the highlights include:

   - "I like the way how the home page is populated with postings by other users, but it would have been great if there was a filter to choose what kind of things are suggested to you and from how far. I really think the initiative is good, even I have thought of a similar concept, more like a network."
   - "For me, my favourite feature is the location selection. I can choose specific alternative locations, which avoids revealing my personal address. This is very important for users like me concerned about privacy."
   - "The feed and scrolling experience were intuitive and easy to use but could also include one's own posts to ensure that you remember what you have posted."
   - "It was good…especially the each and every script of produce for eg 'filter'…this might be helpful in sharing surplus produce".
   - "Pickup date and time feature is very attractive. It's one of the most useful apps in recent times. Yes, it will definitely help in sharing the surplus produce."

10. **In your opinion, what are the main benefits of using the Sharing Surplus platform for sharing surplus produce within the community?**
    Feedback: Participants felt the app was useful for connecting the community and sharing surplus. Notable responses include:

    - "Connecting with others who are on the same initiative and helping the community."
    - "Main benefits include hopefully a reduction in food waste but also the app can help people feel more engaged in their local community as they might get to socialise with others whom they do not regularly see around town. I also believe that the community pane of the app could be expanded to potentially include events where organizations or individual in the community can organise and post events that deal with eliminating food waste in order to promote sustainable alternatives and foster a more close-knit community."
    - "It can reduce the produce going waste".
    - "Economic savings I would say!"
    - "This app promotes socializing, helping others, and sharing".

11. **How do you think the Sharing Surplus platform could contribute to reducing food waste and promoting sustainability in food consumption practices within your local community?**
    Feedback: Participants generally felt that the application was useful in promoting sustainability and reducing food waste. Responses include:

    - "Provide a nice overview of everything that is available within the community. The app collects and shares everything that is available which creates a good overview as this can otherwise become quite messy to gather. Also, it keeps users engaged through the community which allows people to share their experiences with each other and provide help/tips on how best to utilize the app but also the opportunities in the local community."
    - "Efficient Redistribution and local food networks would increase".
    - "By sharing the surplus, it reduces the wastage since it will be helpful for some other people who need the produce. Therefore, it balances out the resources and helps in sustainable practices."

12. **Would you recommend the Sharing Surplus platform to others as a way to participate in food sharing initiatives?**
    Feedback: All participants responded positively, indicating that they would happily recommend the app to others.

13. **Overall, how would you rate your enjoyment of using the Sharing Surplus platform?**
    Feedback: Participants generally enjoyed the application. Most of them gave a rating of 8/10 on a ten scale, or 4/5 on a five scale. Some responses include:

- "App seems robust."
- "A solid 8/10!"
- "4 stars out of 5"
- "Extremely interested and looking forward for something like this in the market."

9.2 Transition Collaboration Evaluation

Since working with transition was a key objective of 'Sharing Surplus', this evaluation was done separately. Transition University of St Andrews is the university's organisation focused on sustainability initiatives across the town of St Andrews [103]. They run many projects aimed at combating climate change and supporting lower carbon living. This makes transition the ideal collaborator for 'Sharing Surplus'.

9.2.1 Procedure

Transition was contacted via email for a demonstration session. The PIS and Consent Form were provided in advance. A session was then arranged at their office, where a member of staff volunteered for the study. The application's features were then demonstrated in the session, and the volunteer tested the application. This session lasted around 30-40 mins, and the verbal feedback was noted. A follow-up questionnaire was sent to Transition to collect further feedback. The results of the feedback are detailed below.

9.2.2 Results

1. **How intuitive did you find the app's user interface?**

Figure 9. 3: *Transition response towards app's UI.*

2. **How would you rate the overall design and layout of the app?**

Figure 9. 4: *Transition response on the overall design and layout of the app.*

3. **How useful do you find the app's main feature of connecting people with surplus produce to those in need?**

Figure 9. 5: *Transition response on the usefulness of the app.*

4. **Were there any features you found particularly helpful or lacking? Please explain.**
   Feedback: The 'Community' feature is a great idea that connects different people. We discussed adding a feature to collect data about the carbon footprint of the shared food.

5. **How would you rate the performance and speed of the app?**

Figure 9. 6: *Transition response on the performance and speed of the app.*

6. **How likely are you to use this app regularly?**

Figure 9. 7: *Transition response on how regularly will they use the app.*

7. **Would you recommend the Sharing Surplus platform to others as a way to participate in food sharing initiatives?**

Figure 9. 8: *Transition response on whether they would recommend the app to others.*

8. **Do you think this app can effectively help reduce food waste in your community?**

Figure 9. 9: *Transition response on whether the app is helpful in reducing food waste.*

9. **Do you feel that this app can help foster stronger community connections?**

Figure 9. 10: *Transition response on whether the app can foster community connections.*

10. **What improvements or additional features would you suggest for this app?**
    Feedback: I wouldn't put emphasis on 'Those in Need' as this may create a barrier for people to sign up. The app should be for everyone. I loved the idea of a good deed feature and the ability for people to share recipes, pictures, etc.

11. **Do you have any other comments or feedback regarding your experience with the app?**
    Feedback: I really enjoyed looking at the Surplus app you designed for your dissertation. It is such a great idea! I've already spoken to a local cafe owner who is growing her own herbs for her kitchen and is looking to swap herbs for veggies that other cafe owners grow. An app like yours would be so handy in these situations too. I have attached the consent form. Good luck!

9.2.3 Discussion

Overall, the feedback from Transition was positive across most of the areas. The 'Community Forum' and 'Good Deed Points' features were particularly appreciated, reinforcing the idea that the app can be a driving force towards community integration. The constructive suggestions were promptly adopted, and the app was updated accordingly. The closing comments provided a strong motivation, with many of the suggestions integrated into the future work section.

9.3 Comparing Existing Tools with Sharing Surplus

In this section, the existing solutions analysed in section *2.3 Related Work* are compared with the features of 'Sharing Surplus'.

Table 9. 1 shows a side-by-side comparison of major features of 'Sharing Surplus' with other solutions mentioned in *2.3 Related Work*

Table 9. 1: *Comparison of 'Sharing Surplus' with similar solutions.*

Features	Sharing Surplus	Olio	Too Good to Go	FareShare	Foody Bag
Mobile Application	✓	✓	✓	✗	✓
Individual producers and requesters	✓	✓	✗	✗	✗
Request Management	✓	✓	✗	✗	✗
Non-Profit	✓	partial	✗	✓	✗
Non chatting	✓	✗	✓	N/A	✓
User Forum	✓	✓	✗	✗	✗
Neutral Location	✓	✗	✗	✗	✗

10. Conclusion

This chapter concludes the project and reflects on whether the objectives were achieved. It evaluates if the application met its initial goal.

The primary purpose of this project was to develop a mobile application that provides a platform to the users to share and collect surplus fruits and vegetables.

10.1 Review of objectives

The primary and secondary objectives established at the beginning of the project served as checkpoints for the application. The functional and non-functional requirements, detailed in section *3. Requirement Specification*, were inspired by these objectives. The tertiary objectives focused on promoting the societal aspect of the application. This made them non-quantifiable, thus they are not discussed in this section. All the tertiary objectives were attempted, but their evaluation is beyond the scope of this section. The application promotes fair and equitable exchange between producers and requesters by implementing a 'First-Come-First-Serve' mechanism. This ensures unbiased prioritisation, thereby eliminating the race-around conditions.

10.1.1 Primary objectives

'Sharing Surplus achieved all of its primary objectives. **Table 10. 1** details how these objectives were accomplished.

Table 10. 1: *Reviewing 'Sharing Surplus' with primary objectives.*

Objective	Achieved?	Remarks
Conduct a comprehensive survey of existing solutions and initiatives aimed at sharing the surplus produce.	Yes	The existing solutions are extensively discussed in the section *2.3 Related Work*
Investigate various models for sharing the surplus produce.	Yes	Various models of sharing surplus food have been extensively discussed in the section *2.2 Food Sharing Models*
Develop a mobile application that lets users to register and share their surplus with other users of the application while	Yes	The APK file has been produced is ready for download and installation on any Android Phone.

maintaining anonymity.		
Test and evaluate the effectiveness and usability of the mobile application with users for facilitating sharing of surplus produce.	Yes	The app has been extensively tested with potential users and Transition St Andrews. Discussion can be found in section *9. Evaluation*

10.1.2 Secondary objectives

All the secondary objectives were also achieved during the development process. **Table 10. 2** outlines how these objectives were met.

Table 10. 2: *Reviewing 'Sharing Surplus' with secondary objectives.*

Objective	Achieved?	Remarks
Implement a user forum within the mobile application to foster community engagement and facilitate communication among the users.	Yes	A community forum has been implemented, where the users can post with other users
Incorporate neutral pickup locations into the application to facilitate safe food exchanges.	Yes	A feature for selecting neutral location has been implemented.
Seek feedback from Transition University of St Andrews, to refine the application and ensure its effectiveness.	Yes	Feedback was obtained from Transition University of St Andrews. This is detailed in section *9.2 Transition Collaboration Evaluation*

10.2 Final remarks

Overall, the development process of 'Sharing Surplus' covered various areas of computer science, including HCI, Software Engineering, and Software Project Management. The project aimed to connect the community, allowing people to share surplus produce

without monetary transactions, thereby reducing the food waste that. The Agile development process along the Clean Architecture [70] facilitated the integration of various features effortlessly. The application development adhered to standard practices and the industry guidelines. It adopted a modern tech stack to ensure the application's longevity.

Key features that were the driving factors behind the application's conception were attempted and achieved. The app integrated features like neutral pickup location, 'Good Deed Points', and a community forum. The Feedback from Transition University of St Andrews[103] and volunteers confirmed that the application met its primary goals. These insights also identified areas for improvements. Some of the most critical ones were addressed immediately, while others were left for the future work.

10. Limitation and Future work

10.1 Limitations

Due to the project's short timeframe, the app encountered several limitations. The main challenge encountered was the cost of using the Google suite products and APIs, like Firebase [90], Google Places [104] and Maps APIs [105]. Only the Free tier of these products were integrated, limiting the app's performance enhancements. Consequently, the prototype was tested locally with the potential users and was not launched on Google Play Store [106]. Although the open-source APIs were explored, they were dropped due to suboptimal performance. The feedback from transition and potential users gave a basis for the future work to be conducted.

10.2 Future Work

- **Launch on Play Store:**
  One significant feedback was the need to make this application available to the community. The application should ideally be deployed on the Play Store, making it accessible to broader audience.

- **Enhance Community Features:**
  Currently, the community forum feature only allows textual posts. Future enhancements should enable the users to add images and interact more dynamically with the content. This will make the feature more interactive and engaging.

- **Improving Produce Management:**
  Additional options to manage produce includes dynamic posting options and pickup arrangements. This provides flexibility for the producers and requesters, as many producers will be more than happy to welcome people across the community.

- **Notification system:**
  Currently, the user has to check the status of the request manually. Implementing a notification system will notify the users of changes in their request status. This reduces the need for manual checks.

- **Administrator Role:**
  Presently, the application works on the trust of the community. An administrator is essential to oversee the community posts and ensure the benefit of the community, maintain a safe environment.

- **Collaborations:**

Collaborating with similar other initiatives and organisations will enhance the application's effectiveness. This would bring in more users and resources. For examples, partnering with an organisation to exchange 'Good Deed Points' for tangible rewards can incentivise users to participate more in the community driven initiatives.

- **Multi-Platform Availability:**
  To reach broader audience, it is essential to develop the application for other platforms. This ensures inclusivity and broader community engagements.

References

[1] ‘Agriculture’s technology future: How connectivity can yield new growth | McKinsey’. Accessed: Aug. 11, 2024. [Online]. Available: https://www.mckinsey.com/industries/agriculture/our-insights/agricultures-connected-future-how-technology-can-yield-new-growth#

[2] B. Francis-Devine, C. Barton, D. Harari, M. Keep, P. Bolton, and R. Harker, ‘Rising cost of living in the UK’, May 2024, Accessed: May 30, 2024. [Online]. Available: https://commonslibrary.parliament.uk/research-briefings/cbp-9428/

[3] S. Whitelocks, ‘More than two-fifths of households with gardens start growing their own fruit and veg’, inews.co.uk. Accessed: May 30, 2024. [Online]. Available: https://inews.co.uk/inews-lifestyle/money/more-than-two-fifths-of-households-with-gardens-start-growing-their-own-fruit-and-veg-amid-rising-food-costs-1791757

[4] ‘Grow your own vegetables to benefit your health and the environment | ILRiverHort | Illinois Extension | UIUC’. Accessed: May 30, 2024. [Online]. Available: https://extension.illinois.edu/blogs/ilriverhort/2020-08-03-grow-your-own-vegetables-benefit-your-health-and-environment

[5] R. Strauss, ‘Reduce food waste by growing food!’, Zero Waste Week. Accessed: Jun. 07, 2024. [Online]. Available: https://www.zerowasteweek.co.uk/reduce-food-waste-grow-food/

[6] ‘What are food miles and why are they important?’, BBC Bitesize. Accessed: Jun. 07, 2024. [Online]. Available: https://www.bbc.co.uk/bitesize/articles/zjnxwnb

[7] A. Tandon, ‘“Food miles” have larger climate impact than thought, study suggests’, Carbon Brief. Accessed: Jun. 07, 2024. [Online]. Available: https://www.carbonbrief.org/food-miles-have-larger-climate-impact-than-thought-study-suggests/

[8] ‘What are Food Miles? And What Is Their Environmental Impact?’ Accessed: Jun. 07, 2024. [Online]. Available: https://www.thecommons.earth/blog/what-are-food-miles-and-what-are-their-climate-impact

[9] ‘Food Miles: Here is how we can reduce them’. Accessed: Jun. 07, 2024. [Online]. Available: https://www.downtoearth.org.in/blog/agriculture/food-miles-here-is-how-we-can-reduce-them-80209

[10] Joanna, ‘What to Do With Excess Veggies from the Garden’, Gingham Gardens. Accessed: Jun. 06, 2024. [Online]. Available: https://ginghamgardens.com/excess-veggies-from-the-garden/

[11] B. Francis-Devine, ‘Who is experiencing food insecurity in the UK?’, Apr. 2024, Accessed: May 30, 2024. [Online]. Available: https://commonslibrary.parliament.uk/who-is-experiencing-food-insecurity-in-the-uk/

[12] Bente Klüppelholz, ‘Towards sustainable communities: Food sharing as an instrument to reduce food waste’, Master Thesis, Mid Sweden University.

[13] C. L. Apicella, F. W. Marlowe, J. H. Fowler, and N. A. Christakis, ‘Social networks and cooperation in hunter-gatherers’, *Nature*, vol. 481, no. 7382, pp. 497–501, Jan. 2012, doi: 10.1038/nature10736.

[14] R. Belk, ‘You are what you can access: Sharing and collaborative consumption online’, *J. Bus. Res.*, vol. 67, no. 8, pp. 1595–1600, Aug. 2014, doi: 10.1016/j.jbusres.2013.10.001.
[15] S. Wahlen and M. Laamanen, ‘Collaborative consumption and sharing economies’, 2017, pp. 94–105. doi: 10.4324/9781315675015.ch9.
[16] J. F. Gollnhofer, ‘The Legitimation of a Sustainable Practice through Dialectical Adaptation in the Marketplace’, *J. Public Policy Mark.*, vol. 36, no. 1, pp. 156–168, Apr. 2017, doi: 10.1509/jppm.15.090.
[17] M. Faverio, ‘Share of those 65 and older who are tech users has grown in the past decade’, Pew Research Center. Accessed: May 30, 2024. [Online]. Available: https://www.pewresearch.org/short-reads/2022/01/13/share-of-those-65-and-older-who-are-tech-users-has-grown-in-the-past-decade/
[18] ‘Communications Market Report 2023’, 2023.
[19] ‘Connected Nations 2023’, Ofcom. Accessed: May 30, 2024. [Online]. Available: https://www.ofcom.org.uk/research-and-data/multi-sector-research/infrastructure-research/connected-nations-2023
[20] J. Harvey, A. Smith, J. Goulding, and I. Branco Illodo, ‘Food sharing, redistribution, and waste reduction via mobile applications: A social network analysis’, *Ind. Mark. Manag.*, vol. 88, pp. 437–448, Jul. 2020, doi: 10.1016/j.indmarman.2019.02.019.
[21] ‘Olio - Your Local Sharing App’, Olio | At Home. Accessed: May 30, 2024. [Online]. Available: https://olioapp.com/en/
[22] foodsharing, ‘Lebensmittel teilen statt wegwerfen – foodsharing’. Accessed: May 30, 2024. [Online]. Available: https://foodsharing.de
[23] E. Ganglbauer, G. Fitzpatrick, O. Subasi, and F. Güldenpfennig, *Think globally, act locally: A case study of a free food sharing community and social networking*. 2014, p. 921. doi: 10.1145/2531602.2531664.
[24] G. Farr-Wharton, J. Choi, and M. Foth, *Food talks back: Exploring the role of mobile applications in reducing domestic food wastage*. 2014. doi: 10.1145/2686612.2686665.
[25] ‘FridgePal - Apps on Google Play’. Accessed: May 30, 2024. [Online]. Available: https://play.google.com/store/apps/details?id=com.fridgepal&hl=en
[26] ‘LeftOverSwap: new app lets users share leftover food’, *BBC Newsround*, Aug. 28, 2013. Accessed: May 30, 2024. [Online]. Available: https://www.bbc.com/newsround/23863218
[27] G. Farr-Wharton, M. Foth, and J. Choi, *EatChaFood: challenging technology design to slice food waste production*. 2013, p. 562. doi: 10.1145/2494091.2497311.
[28] A. Pisoni, C. Canavesi, and L. Michelini, ‘Food Sharing Platforms: Emerging Evidence from Italian and German Users’, *Transp. Res. Procedia*, vol. 67, pp. 137–146, Jan. 2022, doi: 10.1016/j.trpro.2022.12.044.
[29] C. Katzeff, A. C. Kanyama, and J. Zapico, ‘Share or Waste? Using an ICT-platform to Share Food on a University Campus’, presented at the ICT for Sustainability, 2019. Accessed: May 30, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Share-or-Waste-Using-an-ICT-platform-to-Share-Food-Katzeff-Kanyama/eb21510d7765a2a33960c3a0748067aac5e1941a

[30] T. Makov, T. Meshulam, M. Cansoy, A. Shepon, and J. B. Schor, 'Digital food sharing and food insecurity in the COVID-19 era', *Resour. Conserv. Recycl.*, vol. 189, p. 106735, Feb. 2023, doi: 10.1016/j.resconrec.2022.106735.
[31] 'Mobile Operating System Market Share United Kingdom', StatCounter Global Stats. Accessed: May 30, 2024. [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/united-kingdom
[32] R. P. Pratama, S. Humaira, D. N. Fauzi, E. Gunawan, and E. H. Yossy, 'Analysis and Design of Android-Based Food Sharing Applications', in *2023 IEEE 9th International Conference on Computing, Engineering and Design (ICCED)*, Nov. 2023, pp. 1–6. doi: 10.1109/ICCED60214.2023.10425590.
[33] L. Michelini, L. Principato, and G. Iasevoli, 'Understanding Food Sharing Models to Tackle Sustainability Challenges', *Ecol. Econ.*, vol. 145, pp. 205–217, Mar. 2018, doi: 10.1016/j.ecolecon.2017.09.009.
[34] 'Home', FareShare. Accessed: May 30, 2024. [Online]. Available: https://fareshare.org.uk/
[35] 'What We Do', FareShare. Accessed: May 30, 2024. [Online]. Available: https://fareshare.org.uk/what-we-do/
[36] 'Give food', FareShare. Accessed: May 30, 2024. [Online]. Available: https://fareshare.org.uk/giving-food/
[37] 'FAQs', FareShare. Accessed: May 30, 2024. [Online]. Available: https://fareshare.org.uk/giving-food/faq/
[38] 'Olio — Share More, Waste Less – Apps on Google Play'. Accessed: May 30, 2024. [Online]. Available: https://play.google.com/store/apps/details?id=com.olioex.android&hl=en_GB
[39] 'Too Good To Go | Save Good Food From Going To Waste'. Accessed: May 30, 2024. [Online]. Available: https://www.toogoodtogo.com/
[40] 'Too Good To Go: End Food Waste – Apps on Google Play'. Accessed: May 30, 2024. [Online]. Available: https://play.google.com/store/apps/details?id=com.app.tgtg&hl=en_GB
[41] 'Foody Bag - Save on food, reduce food waste', Foody Bag. Accessed: May 30, 2024. [Online]. Available: https://foodybag.com.au
[42] 'Foody Bag - Save On Food – Apps on Google Play'. Accessed: May 30, 2024. [Online]. Available: https://play.google.com/store/apps/details?id=au.com.appspeople.grocer_customer_app&hl=en_GB
[43] 'Our Work With Best Food Logistics', FareShare. Accessed: Jun. 06, 2024. [Online]. Available: https://fareshare.org.uk/giving-food/who-we-work-with/our-work-with-best-food-logistics/
[44] 'Rational Software Architect 9.7.0'. Accessed: Jul. 21, 2024. [Online]. Available: https://www.ibm.com/docs/en/rational-soft-arch/9.7.0?topic=diagrams-use-case
[45] I. Sommerville, *Software Engineering*, 10th ed. Pearson, 2015.
[46] 'The Software Engineering Process: Definition and Scope', IEEE Computer Society. Accessed: Jul. 23, 2024. [Online]. Available: https://www.computer.org/resources/software-engineering-process/
[47] 'Home | Scrum.org'. Accessed: Jul. 23, 2024. [Online]. Available: https://www.scrum.org/index

[48] ‘Manage Your Team’s Projects From Anywhere | Trello’. Accessed: Jul. 23, 2024. [Online]. Available: https://trello.com/home
[49] ‘Git’. Accessed: Jul. 23, 2024. [Online]. Available: https://git-scm.com/
[50] ‘Build software better, together’, GitHub. Accessed: Jul. 23, 2024. [Online]. Available: https://github.com
[51] ‘Why Software Design Is Important’, IEEE Computer Society. Accessed: Jul. 29, 2024. [Online]. Available: https://www.computer.org/resources/importance-of-software-design-is-important/
[52] ‘What is Human-Computer Interaction (HCI)?’, The Interaction Design Foundation. Accessed: Jul. 29, 2024. [Online]. Available: https://www.interaction-design.org/literature/topics/human-computer-interaction
[53] ‘Design & Plan | Android Developers’. Accessed: Jul. 29, 2024. [Online]. Available: https://developer.android.com/design
[54] ‘Material Design’, Material Design. Accessed: Jul. 29, 2024. [Online]. Available: https://m3.material.io/get-started
[55] ‘Android | Do More With Google on Android Phones and Devices’. Accessed: Jul. 29, 2024. [Online]. Available: https://www.android.com/intl/en_uk/
[56] W. L. in R.-B. U. Experience, ‘10 Usability Heuristics for User Interface Design’, Nielsen Norman Group. Accessed: Jul. 29, 2024. [Online]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/
[57] ‘What Is a User Interface (UI)? | Definition from TechTarget’, App Architecture. Accessed: Jul. 30, 2024. [Online]. Available: https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI
[58] ‘What is User Interface (UI) Design? — updated 2024’, The Interaction Design Foundation. Accessed: Jul. 30, 2024. [Online]. Available: https://www.interaction-design.org/literature/topics/ui-design
[59] InspiringApps, ‘Web & Mobile App Development | Custom Software | Android + iOS | InspiringApps’. Accessed: Jul. 30, 2024. [Online]. Available: https://www.inspiringapps.com
[60] ‘Material Theme Builder’. Accessed: Jul. 30, 2024. [Online]. Available: https://material-foundation.github.io/material-theme-builder/
[61] ‘Home’, Canva. Accessed: Jul. 30, 2024. [Online]. Available: https://www.canva.com/
[62] ‘Templates’, Canva. Accessed: Jul. 29, 2024. [Online]. Available: https://www.canva.com/templates/EAFzZi-J0-E-green-vintage-agriculture-and-farming-logo/
[63] ‘Licensing Explained’, Canva. Accessed: Jul. 29, 2024. [Online]. Available: https://www.canva.com/licensing-explained/
[64] ‘What are wireframes and why are they used? | Wireframing Academy | Balsamiq’. Accessed: Jul. 30, 2024. [Online]. Available: https://balsamiq.com/learn/articles/what-are-wireframes/#
[65] ‘Figma: The Collaborative Interface Design Tool’, Figma. Accessed: Jul. 30, 2024. [Online]. Available: https://www.figma.com/
[66] ‘Mockplus’. Accessed: Jul. 30, 2024. [Online]. Available: https://www.mockplus.com/learn/wireframe/high-fidelity-wireframe
[67] ‘UX Goals in Action — Everything You Need to Know’. Accessed: Jul. 30, 2024. [Online]. Available: https://blog.hubspot.com/website/ux-goals

[68] ‘Software Architecture | Software Engineering Institute’. Accessed: Jul. 30, 2024. [Online]. Available: https://www.sei.cmu.edu/our-work/software-architecture/index.cfm
[69] ‘Guide to app architecture’, Android Developers. Accessed: Jul. 30, 2024. [Online]. Available: https://developer.android.com/topic/architecture
[70] ‘Clean Coder Blog’. Accessed: Jul. 30, 2024. [Online]. Available: https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html?source=post_page-----1a9707e3761d----------
[71] ‘What is Clean Architecture in Android?’, GeeksforGeeks. Accessed: Jul. 30, 2024. [Online]. Available: https://www.geeksforgeeks.org/what-is-clean-architecture-in-android/
[72] N. Visavadiya, ‘Clean Architecture in Android’, Simform Engineering. Accessed: Jul. 30, 2024. [Online]. Available: https://medium.com/simform-engineering/clean-architecture-in-android-12d61c4f5318
[73] michaelstonis, ‘Model-View-ViewModel - .NET’. Accessed: Jul. 30, 2024. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm
[74] ‘Guide to Android app modularization’, Android Developers. Accessed: Jul. 30, 2024. [Online]. Available: https://developer.android.com/topic/modularization
[75] ‘A quick intro to Dependency Injection: what it is, and when to use it’, freeCodeCamp.org. Accessed: Jul. 30, 2024. [Online]. Available: https://www.freecodecamp.org/news/a-quick-intro-to-dependency-injection-what-it-is-and-when-to-use-it-7578c84fa88f/
[76] ‘Hilt’. Accessed: Jul. 30, 2024. [Online]. Available: https://dagger.dev/hilt/
[77] ‘Design Patterns’. Accessed: Jul. 30, 2024. [Online]. Available: https://refactoring.guru/design-patterns
[78] ‘What Is A Technology Stack? Tech Stacks Explained’, MongoDB. Accessed: Jul. 30, 2024. [Online]. Available: https://www.mongodb.com/resources/basics/technology-stack
[79] D. Agency, ‘Best mobile app development tech stacks to use in 2024 | DECODE’. Accessed: Jul. 30, 2024. [Online]. Available: https://decode.agency/article/best-mobile-app-development-tech-stack/
[80] ‘Kotlin Programming Language’, Kotlin. Accessed: Jul. 30, 2024. [Online]. Available: https://kotlinlang.org/
[81] ‘Download Android Studio & App Tools’, Android Developers. Accessed: Jul. 30, 2024. [Online]. Available: https://developer.android.com/studio
[82] ‘SDK Platform Tools release notes | Android Studio’, Android Developers. Accessed: Jul. 30, 2024. [Online]. Available: https://developer.android.com/tools/releases/platform-tools
[83] ‘Jetpack Compose UI App Development Toolkit’, Android Developers. Accessed: Jul. 30, 2024. [Online]. Available: https://developer.android.com/compose
[84] ‘Firestore’, Firebase. Accessed: Jul. 30, 2024. [Online]. Available: https://firebase.google.com/docs/firestore
[85] ‘Gradle Build Tool’, Gradle. Accessed: Jul. 31, 2024. [Online]. Available: https://gradle.org/
[86] ‘Android KTX | Kotlin’, Android Developers. Accessed: Jul. 31, 2024. [Online]. Available: https://developer.android.com/kotlin/ktx

[87] ‘The Apache Groovy programming language’. Accessed: Jul. 31, 2024. [Online]. Available: https://groovy-lang.org/
[88] ‘Introduction to declarative UI’. Accessed: Jul. 31, 2024. [Online]. Available: https://docs.flutter.dev/get-started/flutter-for/declarative
[89] ‘Composable’, Android Developers. Accessed: Jul. 31, 2024. [Online]. Available: https://developer.android.com/reference/kotlin/androidx/compose/runtime/Composable
[90] ‘Firebase | Google’s Mobile and Web App Development Platform’, Firebase. Accessed: Jul. 31, 2024. [Online]. Available: https://firebase.google.com/
[91] ‘Firebase Authentication’. Accessed: Jul. 31, 2024. [Online]. Available: https://firebase.google.com/docs/auth
[92] ‘Cloud Storage for Firebase’, Firebase. Accessed: Jul. 31, 2024. [Online]. Available: https://firebase.google.com/docs/storage
[93] ‘Jetpack Compose - Coil’. Accessed: Aug. 01, 2024. [Online]. Available: https://coil-kt.github.io/coil/compose/
[94] ‘The Importance of Software Testing’, IEEE Computer Society. Accessed: Aug. 02, 2024. [Online]. Available: https://www.computer.org/resources/importance-of-software-testing/
[95] ‘Your QA tester’s hierarchy of needs: what is the agile testing pyramid?’ Accessed: Aug. 02, 2024. [Online]. Available: https://www.onpathtesting.com/blog/qa-testers-what-is-the-agile-testing-pyramid
[96] ‘Fundamentals of testing Android apps | Android Developers’. Accessed: Aug. 02, 2024. [Online]. Available: https://developer.android.com/training/testing/fundamentals
[97] ‘What to test in Android | Android Developers’. Accessed: Aug. 03, 2024. [Online]. Available: https://developer.android.com/training/testing/fundamentals/what-to-test
[98] ‘JUnit 5’. Accessed: Aug. 03, 2024. [Online]. Available: https://junit.org/junit5/
[99] D. Hamilton, ‘JUnit Tutorial With Examples: Setting Up, Writing, and Running Java Unit Tests’, Parasoft. Accessed: Aug. 03, 2024. [Online]. Available: https://www.parasoft.com/blog/junit-tutorial-setting-up-writing-and-running-java-unit-tests/
[100] ‘MockK’, MockK. Accessed: Aug. 11, 2024. [Online]. Available: https://mockk.io/
[101] ‘What is Software Evaluation, & How to do it Effectively?’ Accessed: Aug. 05, 2024. [Online]. Available: https://testsigma.com/blog/software-evaluation/#
[102] ‘The 4 Main Types of User Studies |’. Accessed: Aug. 05, 2024. [Online]. Available: https://kadence.com/the-4-main-types-of-user-studies/#
[103] ‘About Us - Transition St Andrews’. Accessed: Aug. 05, 2024. [Online]. Available: https://transitionsta.org/about-us-and-projects/
[104] ‘Overview | Places API’, Google for Developers. Accessed: Aug. 07, 2024. [Online]. Available: https://developers.google.com/maps/documentation/places/web-service/overview
[105] ‘Google Maps Platform’, Google for Developers. Accessed: Aug. 07, 2024. [Online]. Available: https://developers.google.com/maps
[106] ‘Android Apps on Google Play’. Accessed: Aug. 07, 2024. [Online]. Available: https://play.google.com/store/games?hl=en

Appendix A – DOER Document

DOER – Description, Objectives, Ethics, and Requirements

Matric Number: 230016680
Course Code: CS5099

Description
Food redistribution has been in the news for the past few years with various initiatives such as such as FareShare (http://fareshare.org.uk), OLIO (https://olioex.com), and RipeNearMe (https://www.ripenear.me)attempting to minimise the food waste. However, the wastage persists on a smaller scale with many residents in towns and cities, with their relatively small gardens and orchards, end up producing a glut of fruits and vegetables that their families can't consume.
The aim of this project, Sharing Surplus, is to create a solution that facilitates the sharing of surplus fruits and vegetables grown by the residents of the local community. Different models for sharing the produce will be investigated and implemented as a part of a mobile application that lets potential producers and consumers to register for free and share the produce according to the one of the models permitted by the system. Anonymity for both producers and consumers will be prioritized, allowing them to share produce without sharing personal information unless desired. Neutral pickup locations will be incorporated into the model to ensure the privacy and safety of the users.

Objectives
Primary
- Conduct a comprehensive survey of existing solutions and initiatives aimed at sharing the surplus produce.
- Investigate various models for sharing the surplus produce.
- Develop a mobile application that lets users to register and share their surplus with other users of the application while maintaining anonymity.
- Test and evaluate the effectiveness and usability of the mobile application with users for facilitating sharing of surplus produce.

Secondary
- Implement a user forum within the mobile application to foster community engagement and facilitate communication among the users.
- Incorporate neutral pickup locations into the application to facilitate safe food exchanges.
- Seek feedback from Transition University of St Andrews, to refine the application and ensure its effectiveness.

Tertiary
- Strive to minimize food waste and promote sustainability in food consumption practices by providing a platform in the mobile application for sharing tips, recipes, and experiences related to surplus produce sharing.

- Ensure the application promotes fair and equitable exchanges among producers and consumers.

Ethics
As the project requires human subjects to evaluate the mobile application, I would be requiring a full ethics application.

Resources
I will only require standard resources provided by the school to complete this project.

Appendix B – Ethics Approval Letter

University of St Andrews | FOUNDED 1413

School of Computer Science Ethics Committee

04 June 2024

Dear Samarth,

Thank you for submitting your ethical application which was considered by the School Ethics Committee.

The School of Computer Science Ethics Committee, acting on behalf of the University Teaching and Research Ethics Committee (UTREC), has approved this application:

Approval Code:	CS17905	**Approved on:**	04.06.24	**Approval Expiry:**	04.06.29
Project Title:	Sharing Surplus				
Researcher(s):	Samarth Manjunatha Bedre				
Supervisor(s):	Dr Dharini Balasubramaniam				

The following supporting documents are also acknowledged and approved:

1. Application Form
2. Participant Information Sheet
3. Participant Consent Form
4. Advertisement
5. Questionnaire

Approval is awarded for 5 years, see the approval expiry data above.

If your project has not commenced within 2 years of approval, you must submit a new and updated ethical application to your School Ethics Committee.

If you are unable to complete your research by the approval expiry date you must request an extension to the approval period. You can write to your School Ethics Committee who may grant a discretionary extension of up to 6 months. For longer extensions, or for any other changes, you must submit an ethical amendment application.

You must report any serious adverse events, or significant changes not covered by this approval, related to this study immediately to the School Ethics Committee.

Approval is given on the following conditions:

- that you conduct your research in line with:
  - the details provided in your ethical application
  - the University's Principles of Good Research Conduct
  - the conditions of any funding associated with your work
- that you obtain all applicable additional documents (see the 'additional documents' webpage for guidance) before research commences.

You should retain this approval letter with your study paperwork.

School of Computer Science Ethics Committee
Dr Olexandr Konovalov/Convenor, Jack Cole Building, North Haugh, St Andrews, Fife, KY16 9SX
Telephone: 01334 463273 Email: ethics-cs@st-andrews.ac.uk

Appendix C

Agile Development Plan

Week 1-2: Sprint 1 - Initial Setup, Wireframes and Basic Registration (June 3 – June 14)

Goals:

- Set up the project infrastructure and Git Repo
- Create Wireframes
- Design the Auth pages

Tasks:

1. **Project Setup**
2. **Basic Design Decisions**
3. **Integrate with Firebase**
4. **Wireframes**
5. **Design Auth System**
6. **Test**

Week 3-4: Sprint 2 – User Authentication, Profile, and Home Screen Setup (June 17 – June 28)

Goals:

- Implement user login and auth functionality
- Design and Implement the landing page
- Set up user profile page

Tasks:

1. **Implement Firebase Auth System**
2. **Design and Implement Home Page with Top app bar and Bottom nav bar**
3. **Design and Implement User Profile Page**
4. **Test**

Week 5-6: Sprint 3 - Sharing and Viewing Produce (July 1 – July 12)

Goals:

- Implement the functionality for users to share their surplus produce
- Allow users to view available produce from others

Tasks:

1. **Design and Implement the produce listing**
2. **Design and Implement all the required Produce related Screens**
3. **Allow users to view the Produce Listing**
4. **Test**

Week 7-8: Sprint 4 - Pickup Locations, Chat, Community Forum, and Final Testing (July 15 – July 26)

Goals:

- Implement pickup location feature
- Implement the chatting feature
- Set up a basic community forum
- Conduct final testing and prepare for user review

Tasks:

1. **Design and Implement Pick Up Logic**
2. **Design and Implement Chat feature**
3. **Design and Implement Community Forum**
4. **Test**

Week 9-10: Sprint 5 – Test with potential users and final refinement (July 29 – August 9)

Goals:

- Test the application with potential users and transition
- Make the final refinement
- Prepare for presentation and submission

Tasks:

5. **Invite users and test the application**
6. **Refine the application**
7. **Prepare the final apk and bring it to proper format**
8. **Design and Implement Community Forum**
9. **Test**

Week 1-10: Continuous (June 3 – August 9)

Goals:

- Weekly Meetings with the supervisor
- Dissertation write up

Tasks:

10. **Have weekly meetings with the supervisor**
11. **Continue Writing Dissertation**
12. **Keep refining it**

Gantt Chart