**Research Project Mini-Dissertation**

**Food Wastage Application: The Sustainable Spoonful**

**By**

**Lea Thumbiran, Lucinda Zachos, Melany Opperman, Santana Bradbury**

A Mini-Dissertation Submitted as a Partial Requirement for the Bachelor Science in Information Technology: Mobile Application and Web Services

In the Faculty of Information Technology, Eduvos

**Supervisor:**

Mr. Ian Masaga

Date: 24 June 2023

1. **Student Details**

|  |  |  |  |
| --- | --- | --- | --- |
| Student Details | Student Number | Telephone Number | Email Address |
| Opperman, Melany | 6P6NPJX46 | 074 709 6778 | 6p6npjx46@vossie.net |
| Thumbiran, Lea | LMTPQFTH6 | 079 898 2004 | lmtpqfth6@vossie.net |
| Zachos, Lucinda | Y34VR5C17 | 071 878 3138 | Y34vr5c17@vossie.net |
| Bradbury, Santana | CNZJB3199 | 071 864 4170 | cnzjb3199@vossie.net |

1. **Supervisor Details**

|  |  |  |
| --- | --- | --- |
| Supervisor Name | Faculty | Email Address |
| Masaga, Ian | IT | ian.masaga@eduvos.com |

# Acknowledgements

We would like to thank Mr Aruleba, who has always gone above and beyond for his students.  We would also like to thank Mr Masaga, though our time was brief, he did his best to be there for us and support us with this research project. In addition, we would like to give special thanks to our families and team members. Without them, completing this research project would not have been possible.

# Abstract

In South Africa food, water, and energy wastage has escalated into a widespread problem in recent years. While this is something that occurs in many countries, South Africa is not alone in this problem. Due to this, we will be specifically focusing on South Africa in this study.

According to research, one of the main factors that contributed to the most resources being wasted is poor storage and disposal of food on farms leading to an even greater wastage of resources. This is evident from the amount of food that shops and restaurants throw out weekly.

To address the problem of food wastage, we have conducted research on this problem, its sub-problems, the benefits of the study, delimitations of the study as well as developing a reasonable timeline and budget for the project. The results of the research have concluded that a food saver mobile application will be developed to hopefully assist in addressing these issues.

The food saver mobile application will be developed to help assist in reducing the number of resources that are being wasted by shops, restaurants, cafes, and wholesalers. It will further provide people in need with a platform to find and purchase products on a discount. For any products that are not sold at the end of the day, they will then be donated to a local charity or organization of the retailer's choice and at their discretion.

**Table of Contents**

[Acknowledgements 3](#_Toc138796209)

[Abstract 4](#_Toc138796210)

[Chapter 1: Problem Settings 1](#_Toc138796211)

[1.1 Background of Research 2](#_Toc138796212)

[1.2 Aim of Research 5](#_Toc138796213)

[1.3 Research Objectives 6](#_Toc138796214)

[1.4 Problem Statement 7](#_Toc138796215)

[1.4.1.1 Proposed solution 8](#_Toc138796216)

[1.4.2 Sub-Problems 9](#_Toc138796217)

[1.4.2.1 Check there is no current legislation in place that could impede our research. 9](#_Toc138796218)

[1.4.2.2 Establish interest amongst retail partners. 9](#_Toc138796219)

[1.4.2.3 Establish interest amongst consumers. 9](#_Toc138796220)

[1.4.2.4 Determine the experience of food retailers when it comes to food wastage. 9](#_Toc138796221)

[1.5 Benefits of Study 10](#_Toc138796222)

[1.5.1.1 Benefits to the academic area. 10](#_Toc138796223)

[1.5.1.2 Benefits to companies or users. 10](#_Toc138796224)

[1.5.1.3 Benefits to researchers. 10](#_Toc138796225)

[1.6 Delimitation of Study 12](#_Toc138796226)

[1.7 Timelines 14](#_Toc138796227)

[1.7.1.1 Reasonable Timelines 14](#_Toc138796228)

[1.7.1.2 Gantt Chart 15](#_Toc138796229)

[1.7.1.3 Work Breakdown Structure 16](#_Toc138796230)

[1.8 Proposed Budget 17](#_Toc138796231)

[1.9 Research Methodology 19](#_Toc138796232)

[1.10 Preliminary Literature Review 21](#_Toc138796233)

[1.11 Outline of Mini Dissertation 23](#_Toc138796234)

[Chapter 2: Literature Review 26](#_Toc138796235)

[2.1 Introduction 26](#_Toc138796236)

[2.2 Mobile Applications 27](#_Toc138796237)

[2.2.1 Mobile Applications 1 - Flashfood 27](#_Toc138796238)

[2.2.1.1 Technologies used in the existing mobile application. 28](#_Toc138796239)

[2.2.2 Mobile Applications 2 – Imperfect Foods 29](#_Toc138796240)

[2.2.2.1 Technologies used in the existing mobile application. 30](#_Toc138796241)

[2.2.3 Mobile Applications 3 – Too Good To Go: End Food Waste 30](#_Toc138796242)

[2.2.3.1 Technologies used in the existing mobile application. 31](#_Toc138796243)

[2.3 Desktop Applications or Websites 32](#_Toc138796244)

[2.4 Mobile Application Tools and Technologies 34](#_Toc138796245)

[2.4.1 Android Studio 34](#_Toc138796246)

[2.4.2 Java 34](#_Toc138796247)

[2.4.3 Android SDK 35](#_Toc138796248)

[2.4.4 SQLite 36](#_Toc138796249)

[2.4.5 Git / GitHub 36](#_Toc138796250)

[2.4.6 YouTrack 36](#_Toc138796251)

[2.4.7 Figma 37](#_Toc138796252)

[2.4.8 Espresso 37](#_Toc138796253)

[2.5 Mobile Applications Research and Design Methodologies 38](#_Toc138796254)

[2.6 Conclusion 42](#_Toc138796255)

[Chapter 3: System Modelling and Architectural Design 43](#_Toc138796256)

[3.1 Introduction 43](#_Toc138796257)

[3.2 Presentation Layer 44](#_Toc138796258)

[3.2.1 User Experience considerations 44](#_Toc138796259)

[3.2.1.1 Fact-Finding Technique(s) 45](#_Toc138796260)

[3.2.1.2 Analysis of User Requirements 45](#_Toc138796261)

[3.2.1.3 Tools and Diagrams used 47](#_Toc138796262)

[3.2.2 User Interface Design 49](#_Toc138796263)

[3.2.2.1 Designs 49](#_Toc138796264)

[3.2.2.2 Design Tools and Techniques 59](#_Toc138796265)

[3.2.2.3 Designs Look and Feel (Aesthetic, Business Colours, etc.) 60](#_Toc138796266)

[3.2.2.3.1 Design Iteration 1 60](#_Toc138796267)

[3.2.2.3.2 Design Iteration 2 70](#_Toc138796268)

[3.3 Business Layer 82](#_Toc138796269)

[3.3.1 Operations/Process Models Design 82](#_Toc138796270)

[3.3.2 Data Handling Operation Design 89](#_Toc138796271)

[3.3.3 Diagrams 91](#_Toc138796272)

[3.3.4 Iterative and Incremental Design 94](#_Toc138796273)

[3.4 Data Layer 96](#_Toc138796274)

[3.4.1 Logical Data Models 96](#_Toc138796275)

[3.4.2 Diagrams, Tools, and Techniques 100](#_Toc138796276)

[3.4.3 Iterative and Incremental Design 106](#_Toc138796277)

[3.5 Conclusion 107](#_Toc138796278)

[Chapter 4: System Prototype Development and Testing 108](#_Toc138796279)

[4.1 Introduction 108](#_Toc138796280)

[4.2 Testing Plan 111](#_Toc138796281)

[4.3 Layouts Development 119](#_Toc138796282)

[4.3.1 User Interface – Main Screen 119](#_Toc138796283)

[4.3.1.1.1 Iteration 1 119](#_Toc138796284)

[4.3.1.1.2 Iteration 2 120](#_Toc138796285)

[4.3.2 User Interface - Registration 122](#_Toc138796286)

[4.3.2.1.1 Iteration 1 122](#_Toc138796287)

[4.3.2.1.2 Iteration 2 123](#_Toc138796288)

[4.3.3 User Interface - Login 127](#_Toc138796289)

[4.3.3.1.1 Iteration 1 127](#_Toc138796290)

[4.3.3.1.2 Iteration 2 128](#_Toc138796291)

[4.3.4 User Interface – Home 132](#_Toc138796292)

[4.3.4.1.1 Iteration 1 132](#_Toc138796293)

[4.3.4.1.2 Iteration 2 133](#_Toc138796294)

[4.3.5 User Interface – Discounts 136](#_Toc138796295)

[4.3.5.1.1 Iteration 1 136](#_Toc138796296)

[4.3.5.1.2 Iteration 2 137](#_Toc138796297)

[4.3.6 User Interface – Products 142](#_Toc138796298)

[4.3.6.1.1 Iteration 1 142](#_Toc138796299)

[4.3.6.1.2 Iteration 2 143](#_Toc138796300)

[4.3.7 User Interface – Account 145](#_Toc138796301)

[4.3.7.1.1 Iteration 1 145](#_Toc138796302)

[4.3.7.1.2 Iteration 2 145](#_Toc138796303)

[4.4 Business Logic Development 150](#_Toc138796304)

[4.4.1 Algorithm 1 – Login 150](#_Toc138796305)

[4.4.2 Algorithm 2 – Discounted Products 152](#_Toc138796306)

[4.5 Data Access Development 155](#_Toc138796307)

[4.5.1 Database Implementation 155](#_Toc138796308)

[4.5.1.1 Database Creation Class – DatabaseHelper.java 155](#_Toc138796309)

[4.5.2 Data Access Adapters 163](#_Toc138796310)

[4.5.2.1 Accessor Methods - Account 163](#_Toc138796311)

[4.5.2.2 Accessor Methods – Discounted Products 164](#_Toc138796312)

[4.6 Conclusion 166](#_Toc138796313)

[Chapter 5: Results, Conclusion, and Recommendations 167](#_Toc138796314)

[5.1 Results 167](#_Toc138796315)

[5.1.1 Research Findings 167](#_Toc138796316)

[5.1.2 Successes and Failures 169](#_Toc138796317)

[5.1.3 Challenges 171](#_Toc138796318)

[5.1.4 Benefits 172](#_Toc138796319)

[5.2 Conclusion 175](#_Toc138796320)

[5.3 Recommendations 177](#_Toc138796321)

[Appendices 179](#_Toc138796322)

[Appendix A 179](#_Toc138796323)

[The Sustainable Spoonful User Manual 179](#_Toc138796324)

[Background 179](#_Toc138796325)

[Tools and Technologies Used 179](#_Toc138796326)

[Accessing The Mobile Application 180](#_Toc138796327)

[Creating an account 180](#_Toc138796328)

[Logging into an account 182](#_Toc138796329)

[The Home Page 184](#_Toc138796330)

[Viewing and Selecting Discounts 186](#_Toc138796331)

[The Account Page 188](#_Toc138796332)

[Appendix B 189](#_Toc138796333)

[User Requirements Specification Template: 189](#_Toc138796334)

[User Requirements Specification Results: 189](#_Toc138796335)

[Appendix C 190](#_Toc138796336)

[Test Template: 190](#_Toc138796337)

[Test Results: 190](#_Toc138796338)

[List of Figures 191](#_Toc138796339)

[List of Tables 195](#_Toc138796340)

[References 196](#_Toc138796341)

# Problem Settings

In this project, we as a group must create a mobile application that can assist in trying to solve the issue of food wastage. When it comes to food wastage, our group has chosen to focus on food retail stores, cafes, and restaurants.

The issue in question as it pertains to this research project is food wastage and how this loss affects communities both morally and financially as the losses increase rapidly. Some of these issues are due to either the retailer not selling their food products before it reaches their expiry date or customer’s not finishing their meals at restaurants, which ultimately results in food being thrown away.

This mobile application will hopefully assist in taking a step towards helping businesses as well as communities that are struggling in South Africa. Currently, communities, as we draft this document, face the issue of not being able to afford food that is needed to keep them healthy.

In this chapter, we aim to discuss the background of the research and identify its aims and objectives; this is discussed in sections 1.1, 1.2 and 1.3. We then need to draft the problem statement and mention any sub-problems relating to the topic; this is discussed in section 1.4. Afterwards, we need to define the benefits of study, which consists of three sections: the benefits to the academic area, the benefits to the companies or users, and the benefits to the researchers. These are discussed in section 1.5. The delimitations of the study are discussed in section 1.6. The benefits of the study are further broken down to include timelines, a Gantt chart of the project and a work breakdown structure with the different chapters; this is discussed in section 1.7. The budget with a summary of the expenses that we will face is covered in section 1.8. The research methodology is covered in section 1.9. The preliminary literature review is covered in section 1.10, and lastly, the outline of the mini dissertation is covered in section 1.11.

The above will form part of chapter one, and it will need to be finalized before we can move on to writing the literature review in chapter two.

## Background of Research

As previously mentioned, food, water, and energy wastage in South Africa has become increasingly more prevalent in recent years.

In an article published by The Franchise Association of South Africa (FASA, 2019), that is titled 'Tackling South Africa's food waste problem’ they stated that about a third (approximately 10.3 million tonnes) of all food produced in South Africa is never eaten, thrown out and ends up in landfills. In addition to this, they state that the water and energy used during the production of food are enough to fill up multiple swimming pools and power an entire city for many weeks (FASA, 2019). As more time passes, more resources are being used and wasted. This leads to a deteriorating economy where people are more likely to struggle to get the resources that they need.

In another article, titled 'South Africa: Impact of Covid-19 on Food security' the impact of the lack of these resources becomes more apparent. In the article, it is stated that about 9.34 million people suffer from food insecurity (IPC, 2021).

In another article, titled 'The Extent of Food Security in South Africa' written by Statistics South Africa, they stated that almost 20% of all households in South Africa have inadequate access to food (Statistics South Africa, 2019). Even though sufficient food is being produced, it is being wasted, thrown out, and never ends up being consumed.

Another article written by Phumzile Mlangeni, titled 'South Africans decry souring food prices', shows that food wastage and inflation are directly connected (Mlangeni, 2021). In the article, they state that due to food wastage and the costs needed to produce food, many businesses increase their prices to compensate for the food that they have had to throw out (Mlangeni, 2021).

A lot of people cannot afford these food products, and, in turn, they suffer from this as a result.

Previous attempts have been made to address these issues, such as those attempted by Shoprite, Woolworths, and Pick n Pay. Spokespeople for the retailers stated that they are making attempts to reduce food wastage in their stores (Bega, 2021).

* One of the ways that they have reduced food wastage is in their bakeries, which have changed from making large batches of baked goods daily to purchasing frozen products that only require to be baked and can then be consumed (Bega, 2021). So, in turn, stores will only bake what will be needed for the day, and the chances that baked goods are thrown out are reduced. In addition, the stores have stated that they have removed products that they viewed were unpopular to customers from their delis (Bega, 2021). This reduces the chance that stores buy stock that is not in demand that ends up sitting on a shelf and being thrown away.
* The stores have also decided to rotate their stock frequently and only purchase what is needed based on the season, such as ordering fruit and vegetables when they are in season (Bega, 2021).
* Store employees have been further trained to refrigerate perishables, frozen produce, fruit, vegetables, and convenience products within 10 minutes of it being delivered (Bega, 2021).

As mentioned previously, steps have been taken to reduce food, water, and energy wastage; however, it is not sufficient.

According to Bega (2021), most food wastage occurs in the agricultural sector itself and not in retail stores. Most fruit and vegetables from farms are left out to rot or are thrown away due to not being suitable for sale. This could be due to the produce being an unusual colour, shape, or size from what is seen in stores; this means that food is discarded for not meeting a certain aesthetical standard.

In addition, it was observed that privately owned or small businesses and restaurants made little effort to reduce food wastage.

The purpose of this research is to address the issues observed in the agricultural sector, as well as in the business sector.

The Sustainable Spoonful was formed in 2023 and was formed in response to the problems of food, water, and energy wastage in South Africa. We aim to address these issues by developing a mobile application that enables users, businesses as well as charity organizations to communicate with each other.

## Aim of Research

The project aims to create a mobile application that can help minimize the issue of food wastage in South Africa.

We have decided to assist food retailers (such as Woolworths, Spar and Checkers), restaurants, cafes, and wholesalers with their problem of food wastage, which is defined as food that is not sold on time and is then thrown away or wasted when it can still be sold or eaten. This issue is especially pertinent in South Africa, where it is often the case that a large majority of the population cannot afford food for their families.

As it is, the problems of food wastage and inflation add further stress to people as the price of food and other necessities are constantly and drastically increasing, with food prices expected to increase by 13.4% annually (Statistics South Africa, 2023).

This mobile application may also assist communities, students and people that struggle with accessing clean water, healthy, affordable food or who are unemployed. It will further provide a means for stores or businesses to upload the items that they would normally throw away and enable people to purchase these products at a lower price.

Another aim of this project is to increase the number of donations that charities receive. If the products have not been sold at the end of the day, then the food will be donated to a charity of the retailer’s choice.

## Research Objectives

The general objective of this project is for us to allow less fortunate citizens of South Africa to afford better food at better prices as well as to help shops minimize their food wastage.

The specific objectives of this research will be (Indeed Editorial Team, 2023):

* Interviewing the public and determining if people would be interested in using the mobile application that we are planning on developing; this should be done before any mobile application development is done.
* Determining the impact that wastage has on the South African community through research. This can be done in the next two weeks.
* Researching which areas would use the mobile application the most, specifically researching which areas are less fortunate.
* Identifying which stores have an increased amount of food wastage. This can be done within the next three weeks to identify which stores would benefit from utilizing the mobile application.
* Determining what types of foods are wasted in stores. Are the products fruits and vegetables, meats, carbohydrates, or sealed items?
* Determining the discount percentage that the goods should be sold at. The discount percentage can be discussed and negotiated with the stores during the interview process.
* Determining what stores would be willing to donate their wasted food to charity. We can ensure that stores agree that if their products are not sold by the end of the day, they will be donated to a charity of their choice. The procedure for determining which charity the store would like to work with will be conducted during the interview process.

## Problem Statement

A third of all food in South Africa is wasted each year (FASA, 2019). This food wastage leads to inflation in prices as businesses try to recover costs for the products that they have lost (Mlangeni, 2021). This leads to South Africans not having the funding or resources needed to obtain healthy, nutritious food at an affordable price (Statistics South Africa, 2019).

While the government has pledged to halve current food wastage by 2030, more work needs to be done, such as addressing the current electricity crisis in South Africa (WWF, 2017). With food and water wastage, as well as having an electricity crisis, many experts expect that food wastage will increase in the coming years, not decrease (WWF, 2017).

Studies have found that food is wasted the most during production, agricultural post-harvest handling (by up to 50%), processing and packaging (by up to 25%) and storage (WWF, 2017). These processes are out of the general public’s control, and as such, as a company, we can try to mitigate this during the retail phase (by up to 20%) and the consumer phase (by up to 5%) (WWF, 2017).

As current statistics stand, the most wasted foods are (WWF, 2017):

* Fruits and Vegetables (by 44%).
* Cereals (by 26%).
* Meat and Dairy (by 15%).
* Roots, Tubers, and Oil Seed (by 13%).

With the above statistics in mind, this could mean (WWF, 2017):

* About 210 kg of food is wasted every year on average per person.
* About 16 weeks’ worth of electricity that could have been used to power the entire city of Johannesburg is wasted.
* About 1 billion Rand is wasted every year on producing food that is never consumed.
* About 1/5th of South Africa’s water is wasted every year on food that will never be consumed.
* Up to 90% of food wasted ends up in landfills.

As a result of this, only about 46% of all South Africans have access to healthy and affordable food; this harms the country and cannot be ignored (WWF, 2017).

Further studies have found that (WWF, 2017):

* Around 43,6% of all children in South Africa are Vitamin A deficient due to malnutrition; this can further lead to a weakened immune system.
* About 10% of all South African children suffer from iron deficiency which can result in them feeling tired and weak. This can impact their ability to learn and leads to lower IQs.
* Adults also suffer from poor nutrition, and this can leave them open to the risk of diabetes, heart disease and certain cancers.This can put adults who already have a compromised immune system under additional stress.

Currently, there are legislations put in place to assist with the issue of food wastage, but some legislations may worsen food wastage instead of mitigating it (*The Foodstuffs, Cosmetics and Disinfectants Act 54 of 1972 and the Health Act 63 of 1977*).

As a company, we need to establish how this food wastage affects retail stores (such as Woolworths, Spar and Checkers). From a business point of view, this daily wastage can and is affecting profit margins. If we can ensure that food is sold, even at a reduced cost, this can prove to be better than food being thrown away long-term.

#### Proposed solution

As mentioned previously, most of the factors that lead to food wastage are out of the general publics’ control; however, as a company, we can try to assist where possible by working with the retail side of the food industry. We propose that a food-saver mobile application should be developed to address the issues of food wastage. The mobile application will provide a means for customers to be notified of specials relating to food items that are nearing their sell-by date. The mobile application will further aid in providing retail stores with the resources to donate their food products to schools, charities, old age homes and pet shelters if they are not sold by the end of the day.

### Sub-Problems

#### Check there is no current legislation in place that could impede our research.

* Research the current laws surrounding expired food products and the donation of food products.

#### Establish interest amongst retail partners.

* Evaluate whether retail stores would be willing to partner with the food-saver mobile application.
* Do they have any requests or questions regarding its implementation?
* Would they benefit from tax exemptions through the donation clause?

#### Establish interest amongst consumers.

* Evaluate whether consumers would be interested in utilizing the food-saver mobile application.

#### Determine the experience of food retailers when it comes to food wastage.

* Determine the approximate loss to income.

## Benefits of Study

#### Benefits to the academic area.

There are several advantages to reducing food waste in academic settings, including:

* Food waste results in a loss of resources as well as greenhouse gas emissions from the decomposition of organic waste. Academic institutions can support sustainability initiatives and lessen their environmental impact by getting rid of food waste (U.S. Department of Agriculture, 2010).
* Academic institutions can save money when they buy food and trash disposal costs by minimizing food waste (Bell, 2012). This can assist institutions in reallocating funds to other crucial initiatives like research and teaching.
* Food waste signifies a lost opportunity to feed those who are hungry. Academic institutions may be able to give extra food to our company or a food bank, in so assisting in the fight against food poverty in their local communities.
* Reducing food waste in educational settings can also give students a chance to learn about the significance of reducing waste and the economic, environmental, and social advantages of doing so. This may encourage future generations to act sustainably.

#### Benefits to companies or users.

Food waste reduction can benefit both businesses and consumers in a few ways, including:

* Food waste results in a loss of resources and money. Companies can save money on food purchases and reduce disposal costs by decreasing food waste. As they will need to buy less food (Harvard T.H. Chan, 2003).
* Users will be able to find more affordable groceries, helping families that are struggling financially.
* Food waste signifies a lost opportunity to feed those who are hungry. To combat food poverty in their communities, businesses and consumers might potentially contribute extra food to our company or other charity groups by minimizing food waste.
* The mobile application can provide users with information about the impact of food waste and ways to fight it (Harvard T.H. Chan, 2003). It can help raise awareness among users on food wastage.

#### Benefits to researchers.

Our food-saver mobile application connects retailers with customers who may buy certain food products at a discount and, as such, reduce food wastage.

While decreasing food waste is the mobile application’s main advantage, there are other advantages for researchers:

* The mobile application will gather information on the volume of food taken from supermarkets, the kinds of food taken, and the locations of the supermarkets. Researchers who are interested in sustainability and food waste may find this information useful (Dunson, 2021).
* To distribute excess food, the mobile application will partner with a variety of organizations, such as non-profits and food banks. This offers researchers the chance to work with these groups to explore the effects of surplus food redistribution on waste reduction and food security.
* Due to the idea of redistributing extra food being recent, there is still a lot that we do not know about how it will affect food wastage, sustainability, and food security (EPA, 2022). By examining the effects of our mobile application and other comparable mobile applications on these difficulties, researchers who study these topics may discover new areas of inquiry.
* Our mobile application can help increase public knowledge of the value of decreasing food waste and the possible advantages of surplus food redistribution by promoting awareness of both food waste and surplus food redistribution. By fostering more interest in and support for this crucial topic, this raised awareness can aid in funding research in this field.

## Delimitation of Study

The boundaries or restrictions placed on a study to focus and limit the scope of the investigation are referred to as its delimitations (AJE, 2022).

Some potential delimitations for a study on the mobile application include the following:

* **Geographical scope:** The study can be limited to just South Africa, where the mobile application is accessible. This would assist in limiting the study's scope and provide a more specialized review of the mobile applications performance in that field.
* **User demographics:** The study may restrict the users in its sample to a certain demographic, such as working professionals, families with children, or college students. This would enable the researcher to examine how a particular group uses and perceives the mobile application and how well it satisfies their unique demands.
* **Food type:** The study might concentrate on a particular category of food or items sold through the mobile application, such as fresh vegetables, meat, or baked goods. This would assist in limiting the study's scope and enable more thorough research of the mobile application’s effects on the reduction of food waste in that specific category.
* **Timeframe:** To give an overview of how the mobile application is utilized over a certain period, the study could restrict its analysis to a week or a month. This would help to narrow the study's focus and give more detailed information about how the mobile application's users use and view it.
* **Research methodology:** The study may restrict its use of a certain technique, such as a survey, focus group, or case study. This would enable a more concentrated investigation of the study's scope and the effectiveness of the mobile application in attaining its goals (Hamk, 2022).

The following could be used to describe how the geographic boundary for the mobile application is defined:

* Johannesburg in Gauteng, South Africa, will be the primary area of interest for the study.

It will focus on the following aspects of the mobile application:

* + This study will look at how residents utilize the mobile application to save money to overstock food items. The types of food items bought, the frequency of purchases, and the reasons people use the mobile application will all be examined.
  + The study will look at how retailers in the area took part in the initiative. This will include a breakdown of the number of participating stores, the different types of stores, and the motivations behind each store's participation.
  + The study will evaluate the effect of the mobile application on reducing food waste in the country. This will involve a study of the quantity and types of surplus food items saved, as well as the environmental advantages of minimizing food waste (Hutchnizeger, 2022).
  + The study will investigate the mobile application's possible social equality advantages. This will involve a review of how the mobile application might improve low-income households' access to food and lower food insecurity in the area.

## Timelines

#### Reasonable Timelines

|  |  |  |
| --- | --- | --- |
|  | **Description of Work** | **Start and End Dates** |
| **Deliverable: 1** | **Project proposal** | **27/02/2023 - 05/04/2023** |
|  | Introduction |  |
|  | Abstract |  |
|  | Background of research |  |
|  | Aim of research |  |
|  | Research objectives |  |
|  | Problem Statement |  |
|  | Benefits of study |  |
|  | Delimitations of study |  |
|  | Benefits of study: diagrams and tables |  |
|  | * Reasonable timelines * Gantt chart * Work breakdown structure |  |
|  | Research methodology |  |
|  | Proposed budget |  |
|  | Outline of mini dissertation |  |
|  | Preliminary literature review |  |
|  | Final review |  |
| **Deliverable: 2** | **Literature review** | **06/04/2023 - 18/04/2023** |
|  | Introduction |  |
|  | Background research |  |
|  | Tools and technologies |  |
|  | Research and design methodologies |  |
|  | Conclusion |  |
|  | Final review |  |
| **Deliverable: 3** | **System modelling an architectural design** | **19/04/2023 - 25/05/2023** |
|  | User experience |  |
|  | User interface design |  |
|  | Business layer |  |
|  | Data layer |  |
| **Deliverable: 4** | **System prototype development and testing** | **26/05/2023 - 18/06/2023** |
|  | Testing plans |  |
|  | Layouts development |  |
|  | Business logic development |  |
|  | Data access development |  |
| **Deliverable: 5** | **Results, conclusion, and recommendations** | **19/06/2023 - 25/06/2023** |
|  | Results |  |
|  | Conclusion |  |
|  | Recommendations |  |
| **Deliverable: 6** | **Mini dissertation** | **26/06/2023 - 02/07/2023** |
|  | Compile research findings |  |
| **Deliverable: 7** | **Oral Presentation and Defense: Mini dissertation** | **03/07/2023 - 07/07/2023** |
|  | User manual |  |
|  | Presentation |  |
|  | Demonstration |  |

Table 1.1: Reasonable Timelines.

#### Gantt Chart

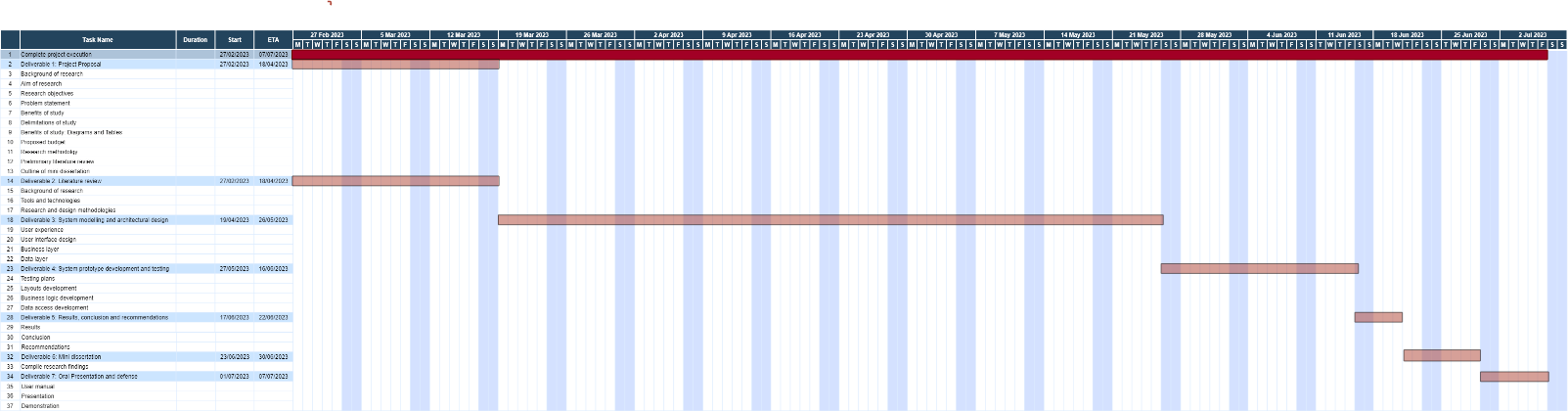


Figure 1‑1: Gantt Chart.

Kindly follow the link [Gantt Chart - Full Version](https://eduvosonline-my.sharepoint.com/:b:/g/personal/y34vr5c17_vossie_net/Ef8QWLYbhC5BiYlbSg785zEBAcQhj2vhdzhLzUSJZ-4ToA?e=WPbO78) to view the image in detail.

Kindly follow the link to our live version: <https://itmdagroup.youtrack.cloud/gantt-charts/166-0>

#### Work Breakdown Structure

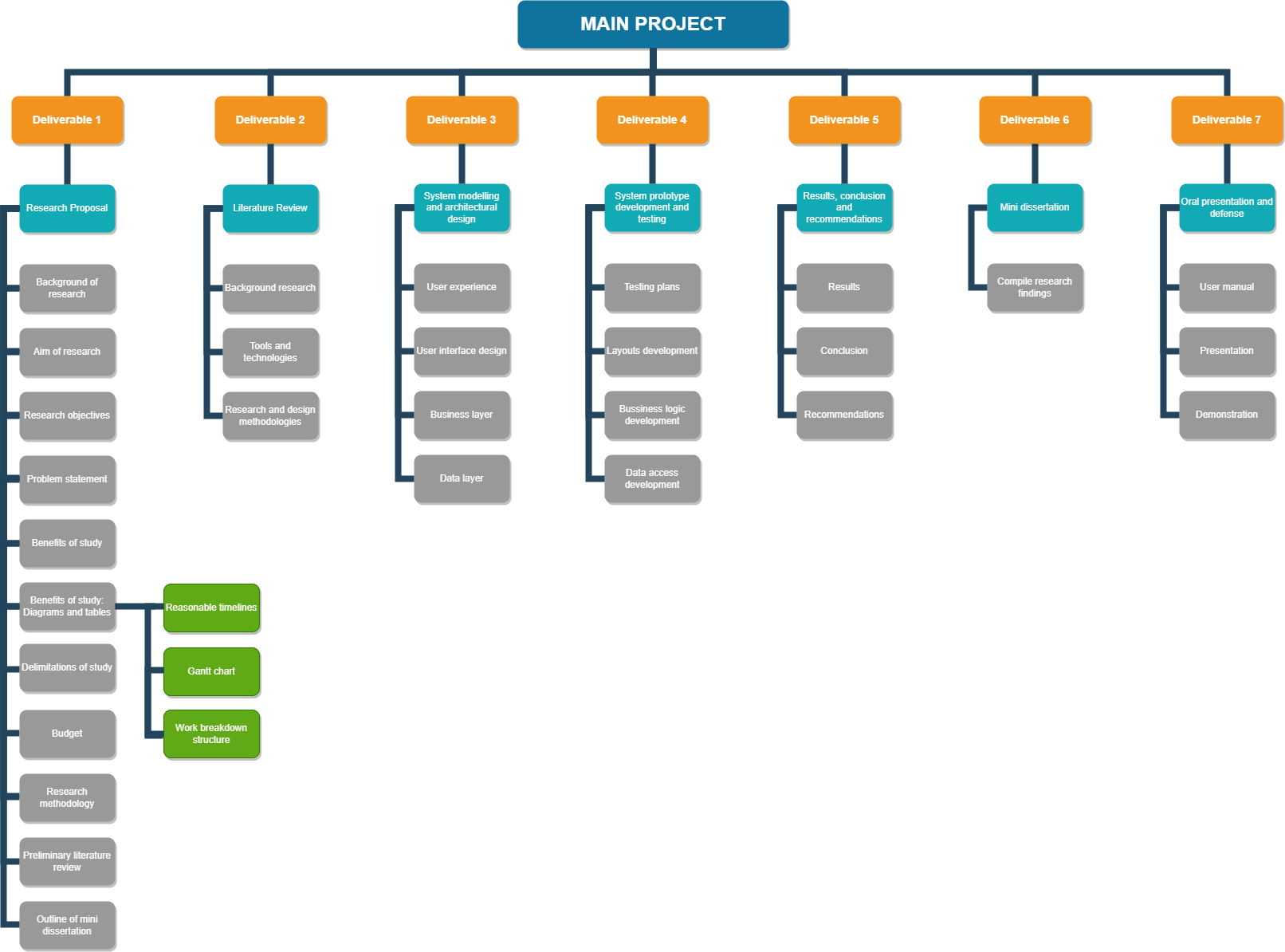


Figure 1‑2: Work Breakdown Structure.

Kindly follow the link below to view the image in detail: [Work break structure](https://eduvosonline-my.sharepoint.com/:i:/g/personal/y34vr5c17_vossie_net/EQ2tT6YpnC9MrvmcgwJNKNMBnbEmPE-MRKLMEfvmrHSMTg?e=fnRbkd)

## Proposed Budget

To manage the budget, it is important to establish a detailed project plan and budget breakdown. This will help ensure that costs are carefully tracked and managed throughout the project’s lifecycle. Regular monitoring and reporting of actual costs versus budgeted costs will also be necessary to identify and address any discrepancies or overruns.

Finally, collaborating closely with each other to ensure effective communication and collaboration will be essential to manage the budget effectively (SOVTECH, 2022).

|  |  |  |
| --- | --- | --- |
|  | **Description of Work** | **Cost (R)** |
| **Deliverable: 1** | **Problem Settings:** *Create a proposal outlining the scope of the project, the mobile applications features, the target audience, the timeline, and the budget.* |  |
|  | Write the proposal. | 0 |
| Compile the proposal into a document. | 1500 |
| Print and distribute the proposal amongst team members and the client. | 500 |
| **Deliverable: 2** | **Literature Review:** *This stage involves researching existing literature and studies related to the problem the mobile application will solve.* |  |
|  | Research existing literature and studies. | 0 |
| Compile the research into a document. | 1500 |
| Hire a proofreader. | 2500 |
| **Deliverable: 3** | **System Modelling and Architectural Design:** *This involves designing the overall system architecture, database schema, and API structure.* | 6000 |
| **Deliverable: 4** | **System Prototype Development and Testing:** *This involves developing a functional prototype of a mobile application and assessing it for bugs and glitches.* |  |
|  | Hire system testers. | 2500 |
| Host the mobile application (R200 x 36 months). | 7200 |
| **Deliverable: 5** | **Results, Conclusion, and Recommendations:** *This stage involves analyzing the results of the testing and making conclusions and recommendations for future improvements.* | 3000 |
| **Deliverable: 6** | **Mini-Dissertation:** *This stage involves documenting the entire mobile application development process in the form of a* *mini-dissertation.* | 3000 |
| **Deliverable: 7** | **Oral Presentation and Defence:** *This stage involves presenting the project and defending the work before a panel of experts.* |  |
|  | Print business cards (double-sided cards: R1,75 x 50 ). | 87,50 |
| Print posters (Size A4 poster: R65 x 2). | 130 |
| Rent a webcam (R500 x 1). | 500 |
| Rent a microphone (R175 x 1). | 175 |
| Rent a projector (R500 x 1). | 500 |
|  | **Total** | **29 092,50** |

Table 1.3: Proposed budget (Thomas, R., 2022).

## Research Methodology

1. **Research Problem and Questions:**

The research problem for this project proposal is to identify the feasibility and effectiveness of developing the mobile application to reduce food waste and promote sustainability in the food industry. To address this, the following research questions will be explored:

* What are the current food waste reduction practices in the food industry?
* What are the factors contributing to food waste in the industry?
* What are the potential benefits and limitations of developing our mobile application to reduce food waste?
* How effective is our mobile application in reducing food waste and promoting sustainability in the food industry?

1. **Research Design:**

To address the research problem and questions, a mixed-methods research design will be used. This design uses both quantitative and qualitative research methods to provide a comprehensive understanding of the research problem and answer the research questions.

1. **Data Collection:**

The following data collection methods will be used:

* **Literature review:** This will involve a comprehensive review of relevant literature, including academic journals, articles and reports related to food waste reduction.
* **Surveys:** Surveys will be used to collect data from potential users and food retailers.
* **Interviews:** Qualitative data will be collected through interviews with experts in the food industry.

1. **Data Analysis:**

The data collected from the literature review, surveys, and interviews will be analysed using both quantitative and qualitative data analysis techniques. Quantitative data will be analysed using statistical software, and qualitative data will be analysed thematically to identify key themes and patterns related to the research questions.

1. **Expected Results:**

The expected results of the study will be a comprehensive understanding of the current food waste reduction practices in the food industry, the factors contributing to food waste, and the potential benefits and limitations of developing our mobile application to reduce food waste and promote sustainability. Additionally, the study will provide insight into the effectiveness of our mobile application in reducing food waste and promoting sustainability in the food industry.

1. **Limitations:**

Some limitations may include limited sample size, potential bias in data collection and potential limitations in the data collection methods used (McCombes, 2023).

## Preliminary Literature Review

In this chapter, we have discussed businesses with a problem of food wastage that is becoming a large problem in South Africa (A third of South Africa's food is wasted every year).

The group will develop a mobile application to try and help reduce that amount. We have been researching the topic of food wastage in South Africa and how it affects other problem areas in the country by exacerbating them.

We have found that the rampant wastage of food has far-reaching consequences in other areas, such as the wasting of electricity, which is currently being used to produce food that is just being wasted, considering that South Africa already has an electricity crisis and we are currently experiencing load-shedding, we do not have electricity to waste.

We have also found that the issue of food wastage can cause water wastage as well, where water is used in growing food that will only be wasted. Therefore, our plan for this mobile application is to help minimize the issues caused by food wastage because water and electricity are two of South Africa’s major concerns.

This mobile application will also help people gain access to food where previously they had little hope; people work extremely hard to afford the bare necessities and can only afford the cheapest/ bare minimum food to survive. This leads to malnutrition in parents and children, affecting learning capabilities. Hunger can also lead to death, where children can be left orphans in the system, and people lose family and friends.

The goal would be to start decreasing hunger in the country by the time this mobile application is created, as well as give the people an opportunity to look after their families and themselves even if they are struggling financially. The group will aim to find food retailers and restaurants that experience a lot of food waste and add them to the mobile application. We will also try to find charities that the food retailers can partner with if the food is not sold at a decreased price.

Other mobile applications that have been created for the same reason of food shortage are ‘Too Good To Go’, which is being used with food retailers and restaurants in major European cities; ‘Imperfect Food’, that is available along the West Coast, and ‘Flashfood’ which is used in Canada and the United States of America.

## Outline of Mini Dissertation

Below is a breakdown of what will be explored in our mini dissertation. We will use this as a guide to ensure that we are as comprehensive in our research as possible.

* Abstract
* Acknowledgements
* Chapter 1: Problem Settings
  + Background of Research
  + Aim of Research
  + Research Objectives
  + Problem Statement
    - Sub-Problems
  + Benefits of Study
  + Delimitation of Study
  + Timelines
  + Outline of Mini Dissertation
* Chapter 2: Literature Review
  + Introduction
  + Mobile Apps
  + Mobile Apps in your area of research
    - Technologies used in the existing mobile apps
  + Mobile Apps 2
    - Technologies used in the existing mobile apps
  + Desktop Apps or Websites
  + Mobile App Tools and Technologies
  + Mobile Apps Research and Design Methodologies
  + Conclusion
* Chapter 3: System Modelling and Architectural Design
  + Introduction
  + Presentation Layer
    - User Experience
      * Fact-Finding Technique(s)
      * Facts Analysis
    - User Interface
      * User Interface Design
        + Iterations
  + Business Layer
    - Operations/Process Models Design
      * Iterations
    - Data Handling Operations
      * Iterations
  + Data Layer
    - User Data
      * Iterations
    - System Data
      * Iterations
  + Conclusion
* Chapter 4: System Prototype Development and Testing
  + Introduction
  + Testing Plan
  + Testing Types
  + Layouts Development
    - User Interface 1
      * Iterations
    - User Interface 2
      * Iterations
    - User Interface 3
      * Iterations
  + Business Logic Development
    - Algorithm one
      * Iterations
    - Algorithm two
      * Iterations
  + Data Access Development
    - Database Implementation
      * Database Creation Class
        + Iterations
    - Data Access Adapters
      * Accessor Methods
        + Iterations
  + Conclusion
* Chapter 5: Results, Conclusion, and Recommendations
  + Results
    - Research Findings
    - Successes
    - Challenges
    - Benefits
  + Conclusion
  + Recommendations
* Appendices
* List of Figures
* List of Tables
* References

# Literature Review

## Introduction

Food wastage is a significant issue that has a negative impact on the economy, environment and on society. It is estimated that about one-third of all food produced is lost or wasted globally, with the biggest contributor to food wastage being consumers. Food-saver mobile applications have become a popular tool in reducing food wastage in recent years. This is due to them providing consumers with a means to reduce food wastage and save money as well.

In this literature review, we will explore the current state of research on food-saver mobile applications as well as their impact on reducing food wastage. We will further examine the features and effectiveness of multiple food-saver mobile applications in different contexts. By incorporating this existing research, we aim to provide a comprehensive understanding of the potential of food-saver mobile applications in addressing the issue of food wastage.

## Mobile Applications

As we embark on this project, we need to acknowledge that there may be competitors in the market. We have managed to identify three similar mobile applications that share our focus but are only available in certain countries, not including South Africa.

**These are:**

* **Flashfood** (Flashfood, 2021).
* **Imperfect Foods** (Imperfect Foods, 2021).
* **Too Good To Go: End Food Waste** (Too Good To Go, 2021).

### Mobile Applications 1 - Flashfood

FlashFood is a mobile application that enables consumers to search for food products that are close to their expiry date and to purchase them at a discounted rate of 50% or more. According to consumers, they have saved at least 50% on their groceries by buying them on this mobile application.

**This mobile application uses the following process (EatSleepBreatheFI, 2021):**

* To utilize this mobile application, users must first create an account. After creating an account, they can then browse through the stores in their area and pick the items that they would like to purchase. They may then pay on the mobile application for the items that they have selected.
* To collect their order, users must go to the stores’ customer service desk and inform them that they are collecting their Flashfood order.
* Customer service will look up the users’ information on the Flashfood system and ask for verification of purchase.
* Once the purchase has been verified, the user will be escorted to the Flashfood fridge, which the retailer will unlock and give the user their order.
* It should be noted that the goods that are listed on the mobile application are still fresh and of decent quality.
* The mobile application is limited to stores across Canada but, as of last year, has started expanding its reach into the United States of America.
* Users cannot pay in cash and must pay through the mobile application and, therefore, must have added their card details to the mobile application.

By comparison, our mobile application will not require the user to link their bank card to pay. Instead, the user will pay at the store that they are collecting from. This way, they can pay through cash or card or even purchase more items should they require anything else from the store (Flashfood, 2022).

Our mobile application will instead generate a discount code for the user to scan at the store to qualify for the discount.

#### Technologies used in the existing mobile application.

In the mobile application, Flashfood, they have used a wide variety of tools and technologies to bring their final product together.

**These tools and technologies include but are not limited to (6sense, 2023):**

* **Microsoft Azure** 
  + A Windows Azure cloud-based platform that enables businesses to run critical applications with a higher production price ratio.
  + This is done by running them on the services’ platforms’ data centres on a pay-as-you-go basis.
* **HTML**
  + HyperText Markup Language (HTML) is the standard markup language for documents that are designed to be displayed in the web browser.
  + HTML can be assisted by CSS (Cascading Style Sheets) and scripting languages.
* **JavaScript** 
  + This is a programming language used alongside HTML and the Web.
* **Amazon DynamoDB** 
  + A flexible database service that does not use SQL (Structured Query Language).
* **Java** 
  + A programming language that has been implemented and used as a computing platform.
* **Typekit**
  + This is an online service that offers a subscription-based library of high-quality fonts, which are either used directly on the website or synced using the Adobe Creative Cloud to applications on the subscriber's computer.

### Mobile Applications 2 – Imperfect Foods

Imperfect Foods is a mobile application where fresh produce that is not accepted by stores due to not meeting a certain aesthetical standard is sold at a cheaper rate. In other words, Imperfect Foods delivers superficially imperfect, excess produce at a lower cost. These products are still fresh and perfect for eating but are being sold at a lower rate as their visual appearance is not too appealing.

**This mobile application uses the following process (Chen, C., 2022):**

* Users must first create a profile on the mobile application.
* The mobile application will then ask the user what food items they would like to receive and what food items they would never want to receive. This is done to build a box based on the users' preferences.
* The user can add fresh produce, dairy, meat, seafood, snacks, and a small selection of household and beauty products to their box.
* The user has four days to edit their box of products before it is shipped to the user's house on a specific day.
* Imperfect Foods assigns a delivery day for the user based on their ZIP code as it is easier and better for the environment.
* This means that there will be a shipping fee added to the users’ bill, which is usually between $4.99 to $8.99 (or about R90 to R160).
* Imperfect Foods is available in the West South-Central region, Midwest, Northeast, and all along the West Coast of the United States.

In our mobile application, the user must go to the selected store to purchase and collect their chosen food items, and it will not be delivered to them. We have decided to exclude this as an option to keep costs low for the user. This would go against our goal of saving users as much as possible when purchasing through our mobile application.

#### Technologies used in the existing mobile application.

In the mobile application Imperfect Foods, they have used a wide variety of tools and technologies to bring their final product together.

**These tools and technologies include but are not limited to (6sense, 2023):**

* **HTML**
  + HyperText Markup Language (HTML) is the standard markup language for documents that are designed to be displayed in the web browser.
  + HTML can be assisted by CSS (Cascading Style Sheets) and scripting languages.
* **JavaScript**
  + This is a programming language used alongside HTML and the Web.
* **Java**
  + A programming language that has been implemented and used as a computing platform.
* **RabbitMQ**
  + Open-source message broker software.
* **Moment.js**
  + Open-source JavaScript library for wrapping objects.
* **PostgreSQL**
  + This is an objected-related Database Management system.
* **Optimizely**
  + Experimentation platform used for testing.
* **GitHub**
  + Collaboration tool to help development teams write better code faster.

### Mobile Applications 3 – Too Good To Go: End Food Waste

Too Good To Go is a mobile application that enables users to get meals at a lower cost from restaurants (Tolcheva, S., 2021).

**Features of this mobile application include:**

* They are enabling restaurants, food kiosks, and different grocery stores to sell their excess products to make extra income to compensate for food that they would have otherwise thrown away.
* They have a ‘Magic Bag’ on the mobile application where mystery items are added into the box where the cost for it is under $5.
  + This magic bag is the most popular order on the mobile application.
    - The food that gets added to the magic bag varies, and the user can either receive sweet or savoury goods. Overall, the magic bag usually consists of three items.

**This mobile application uses the following process:**

* A user creates an account and can view the restaurants around them that are available with the mobile application.
* The user would then make their payment on the mobile application and then go to the store location and pick up their order.

In our mobile application, users will also need to pick up their order at the selected location, but they cannot pay on the mobile application.

#### Technologies used in the existing mobile application.

**The main technology used to create this mobile application is:**

* Artificial intelligence (AI) plays a big role in helping to cut down on food waste. AI may be used to forecast demand, improve supply chain management, and lessen food waste.

## Desktop Applications or Websites

Our mobile application aims to reduce food waste by connecting users with surplus food items at discounted prices. While there are other food waste reduction applications in the market, our application stands out with its unique business model and user-friendly interface.

* One application that is a potential competitor is the application ‘Too Good To Go.’ Like our mobile application, Too Good To Go connects users with surplus food items at discounted prices from local restaurants and grocery stores. However, Too Good To Go operates in multiple countries across Europe and North America, while our mobile application will only operate in South Africa (Too Good To Go, 2023).
* Another similar application is ‘OLIO.’ Unlike Too Good To Go, OLIO allows individuals to share their surplus food items with their local community. The application facilitates peer-to-peer sharing of food, intending to reduce food waste at the household level. OLIO operates in multiple countries across Europe and North America (Olio, 2023).
* When it comes to websites, there are a few resources that provide information related to reducing food waste. ‘Love Food Hate Waste’ is a website that provides tips and recipes for reducing food waste, as well as information on food storage and preservation. The website is run by a non-profit organisation in the UK and has resources available for users in multiple countries (Love Food Hate Waste, 2019).

**In terms of research related to reducing food waste, there is a growing body of literature on the topic:**

* One recent study published in the journal ‘Sustainability’ analysed customers’ attitudes towards food waste and the use of food waste reduction applications (Salins, M. and Aithal, S., 2022). The study found that consumers were generally positive towards the use of such applications but highlighted the need for greater awareness and education around food waste reduction strategies.
* Another study published in the journal ‘Cleaner Production’ explored the environmental impacts of food waste reduction interventions and found that interventions focused on reducing food waste at the household level were the most effective in terms of reducing environmental impacts.

In terms of critiques of existing works related to food waste reduction, one common one is that many interventions focus on reducing food waste at a household level while neglecting the role of larger systemic issues such as supply chain inefficiencies and overproduction.

Additionally, there is a need for greater collaboration between stakeholders in the food system, including producers, retailers, and consumers, to create effective and sustainable solutions for reducing food waste.

Finally, there is a need for more research to be conducted on the long-term impacts and sustainability of food waste reduction interventions to ensure that they are effective and viable solutions for reducing food waste in the long term.

## Mobile Application Tools and Technologies

**The tools and technologies we plan to use include but are not limited to:**

### Android Studio

The mobile application will be primarily designed using Android Studio. This Integrated Development Environment (IDE) is the main IDE for Android application development (Android Developers, 2018).

Android Studio has standout features to enhance our productivity while developing the food-saver mobile application.

**These features include but are not limited to (Android Developers, 2018):**

* A flexible Gradle-based build system.
* A fast and feature-rich emulator.
* A unified environment where development can be done for all Android devices.
* Enabling changes to be made to the live application without the need to restart.
* Easily importing code templates directly into the application.
* GitHub integration for easier task collaboration.
* Supporting a large array of testing tools and frameworks.
* Providing lint tools to catch performance, usability, version compatibility, and other issues.
* C++ support.
* NDK support.
* Providing integrated support for the Google Cloud Platform and Application Engines.

**Why have we chosen to use Android Studio?**

* It is easy to use and provides us with a built-in emulator to test our mobile application during the development process.

### Java

Our language of choice will be Java.

**Java is:**

* A modern Object Orientated programming language (Java.com, 2023).

**Why have we chosen to use Java?**

* Java is a popular programming language that boasts a large online community of developers. This means that should we require support or have an issue during the development phase of our project; there is a wealth of knowledge online to assist us.

**Java versus Kotlin**

* As it stands, Java is a more straightforward language, but, if necessary, we may switch to using Kotlin for developing this mobile application.

### Android SDK

The Android SDK (Software Development Kit) is a collection of Application Programming Interface (API) libraries and software development tools essential to Android application development (GeeksforGeeks, 2021).

It is regularly updated in response to any Android Studio updates and often includes new features. These features include quality-of-life updates to both development and debugging.

**Some of the tools included in Android SDK are (GeeksforGeeks, 2021):**

* Android SDK build tool.
* Android Emulator.
* Android SDK platform tools.
* Android SDK tools.
* Android Debug Bridge.
  + This provides developers with the ability to conduct unit tests and perform automated testing.

In terms of deployment and distribution, the Android SDK also provides the tools necessary to create a signed Android Package Kit (APK), which is required should we wish to list the mobile application on the Google Play Store (Android Developers, 2018).

**Why have we chosen to use the Android SDK?**

* This is an essential tool for mobile development in Android Studio.

### SQLite

Structured Query Language (SQL) Lite is a small, fast, self-contained, high-reliability, full-featured Data Query Language (DQL) database engine (Sqlite.org, 2019).

SQLite is also open-source and free to use.

**Why have we chosen to use SQLite?**

* It is built-in to all mobile phones and is well suited for mobile applications that require local storage.

### Git / GitHub

GitHub is a powerful collaboration tool with unlimited repositories, the best-in-class version control, and a large open-source community (GitHub, 2022).

**Some of the main features of GitHub include:**

* Version control.
  + Makes tracking project changes easier over time.
* Code review.
* Issue Tracking.
  + This includes assigning issues to team members for review.
* Integration.
  + GitHub actions.
    - Assist with automating builds, testing and deployment workflow.

**Why have we chosen to use GitHub?**

* To better help us manage any code changes and provide an easy means of collaboration.
* There is the option of GitHub mobile, so we also have access to our project from any device.

### YouTrack

YouTrack is a project management and collaboration tool developed by IntelliJ under the software company JetBrains.

It can be used to track projects and tasks, use agile boards, plan sprints and releases, create workflows and create and track Gantt Charts (JetBrains, 2022).

We are currently using this tool to track our project and its tasks through a Gantt chart.

### Figma

Figma is a User Experience (UX) web-based design tool used to create interactive wireframes and prototypes. It is a collaboration tool ideal for teams developing mobile applications (Figma, 2023).

**Why have we chosen to use Figma?**

* It is ideal for creating our wireframe concepts.

### Espresso

Espresso was created by Google. It is now the go-to for Android development testing (Android Developers, 2018).

It easily integrates with Android Studio and allows for both Android native view testing as well as hybrid web views.

Espresso can be used to write User Interface (UI) tests, and it is considered extremely robust.

## Mobile Applications Research and Design Methodologies

Mobile application developers typically use a range of research and design methodologies during the development of their mobile applications.

**These methodologies can include but are not limited to (Ortega 2019):**

* user research
* feature-driven development
* scrum framework
* agile development processes

User research is a key component of mobile application development, as it helps developers to understand the needs and preferences of their target users. This can involve a range of techniques, such as surveys, interviews, and observation, as well as the use of personas and user journeys to guide the development process. User research is particularly important for a mobile application like ours, as it is designed to meet a specific need (reducing food waste) and understanding the motivations and behaviours of users is crucial to developing a mobile application that meets their needs effectively (Orbitum Labs, 2023).

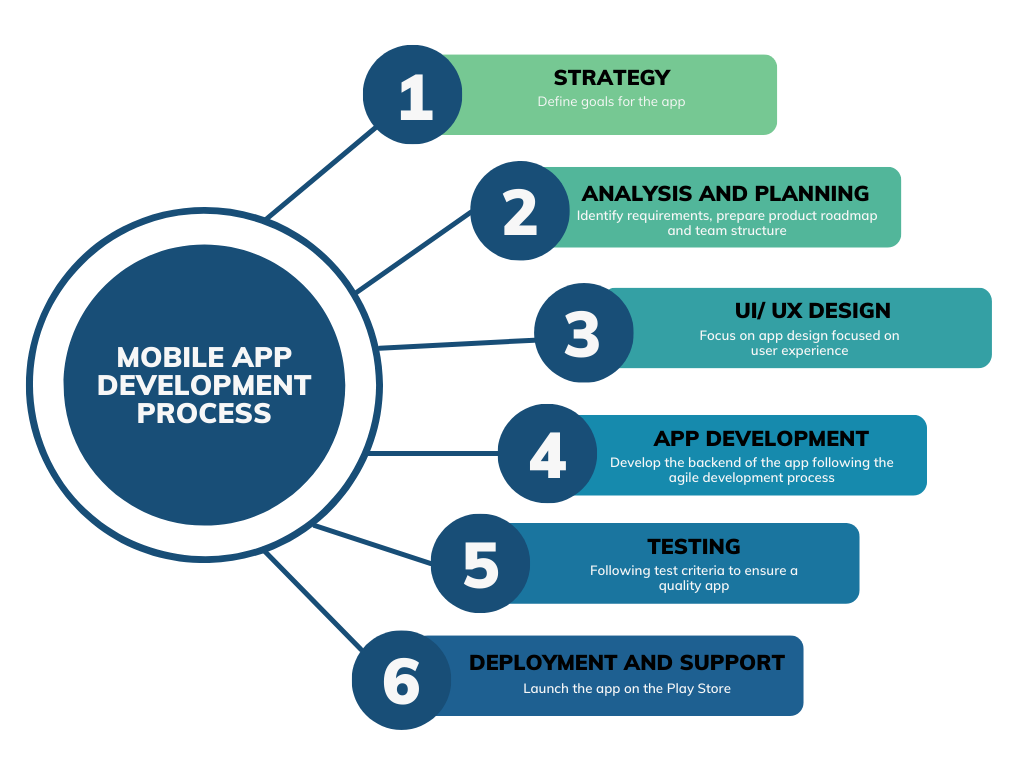


Figure 2‑1: Mobile Application Development Process (Hall, R., 2018)

Feature-driven development is an incremental and iterative method of software development (Ling, 2020). It emphasizes segmenting the development process into more manageable, smaller components known as features, each of which is independently produced, tested, and integrated into the whole system (Ling, 2020). Collaboration is a key component of feature-driven development, and the development team and stakeholders are actively involved throughout the process (Ling, 2020). Feature-driven development is essential for a mobile application like ours, as it allows us to prioritize and develop features based on user needs and preferences, ensuring that our mobile application effectively addresses the problem of food waste (Ling, 2020). This iterative approach allows us to gather feedback from users early and continuously, refining the mobile application based on their input (Ling, 2020). FDD’s focus on collaboration facilitates effective development and ensures a well-rounded mobile application (Mirzoyan, V., 2023).

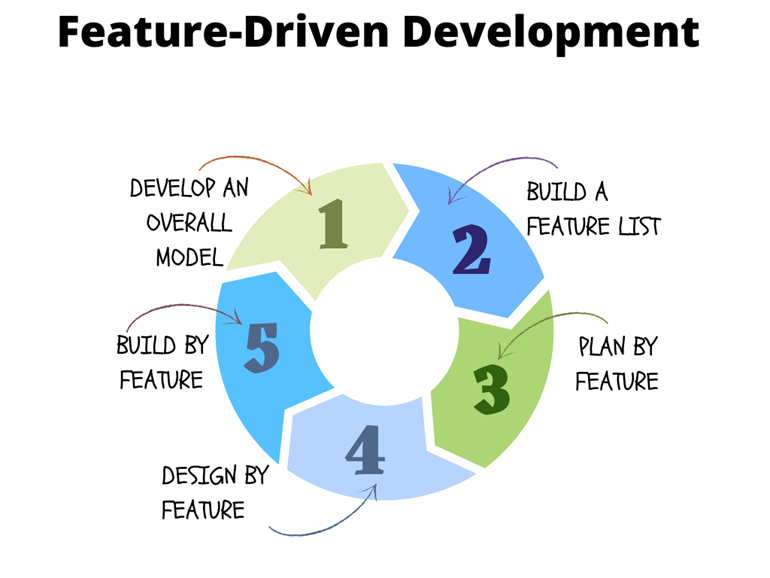


Figure 2‑2: Feature-Driven Development (Mirzoyan, V., 2023).

Scrum is a framework for agile development that emphasizes incremental and iterative product creation (nimble, 2023). It places a strong emphasis on cooperation, adaptability, and ongoing progress (nimble, 2023). Scrum is a set of standardized roles, activities, deliverables, and guidelines that direct the development process (nimble, 2023). Scrum is essential for a mobile application like ours because it enables us to embrace an iterative and incremental development approach (nimble, 2023). By working in short sprints, we can deliver valuable features and functionality faster (nimble, 2023). The regular feedback loops and collaboration within the scrum framework ensure that the mobile application aligns closely with user needs and preferences, leading to a more effective solution (nimble, 2023). Scrum’s flexibility allows us to adapt to changing requirements and incorporate user feedback, ultimately resulting in a high-quality mobile application that solves our problem (reducing food waste) (Peek, 2023).

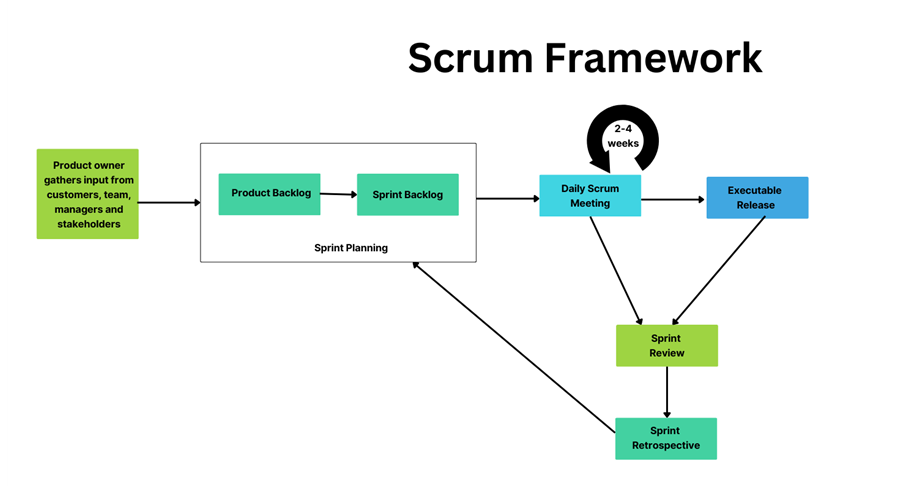


Figure 2‑3: Scrum Framework (Gren, J., 2021).

Agile development processes are also commonly used in mobile application development, which involve iterative cycles of design, development, and testing. This approach allows developers to respond quickly to feedback and make changes to the mobile application’s design and functionality based on user needs and preferences. Agile development processes can be particularly effective for a mobile application such as ours, as it allows developers to respond to the evolving needs of users and ensure that the mobile application is meeting its intended purpose of reducing food waste (Esferasoft Solutions, 2021).

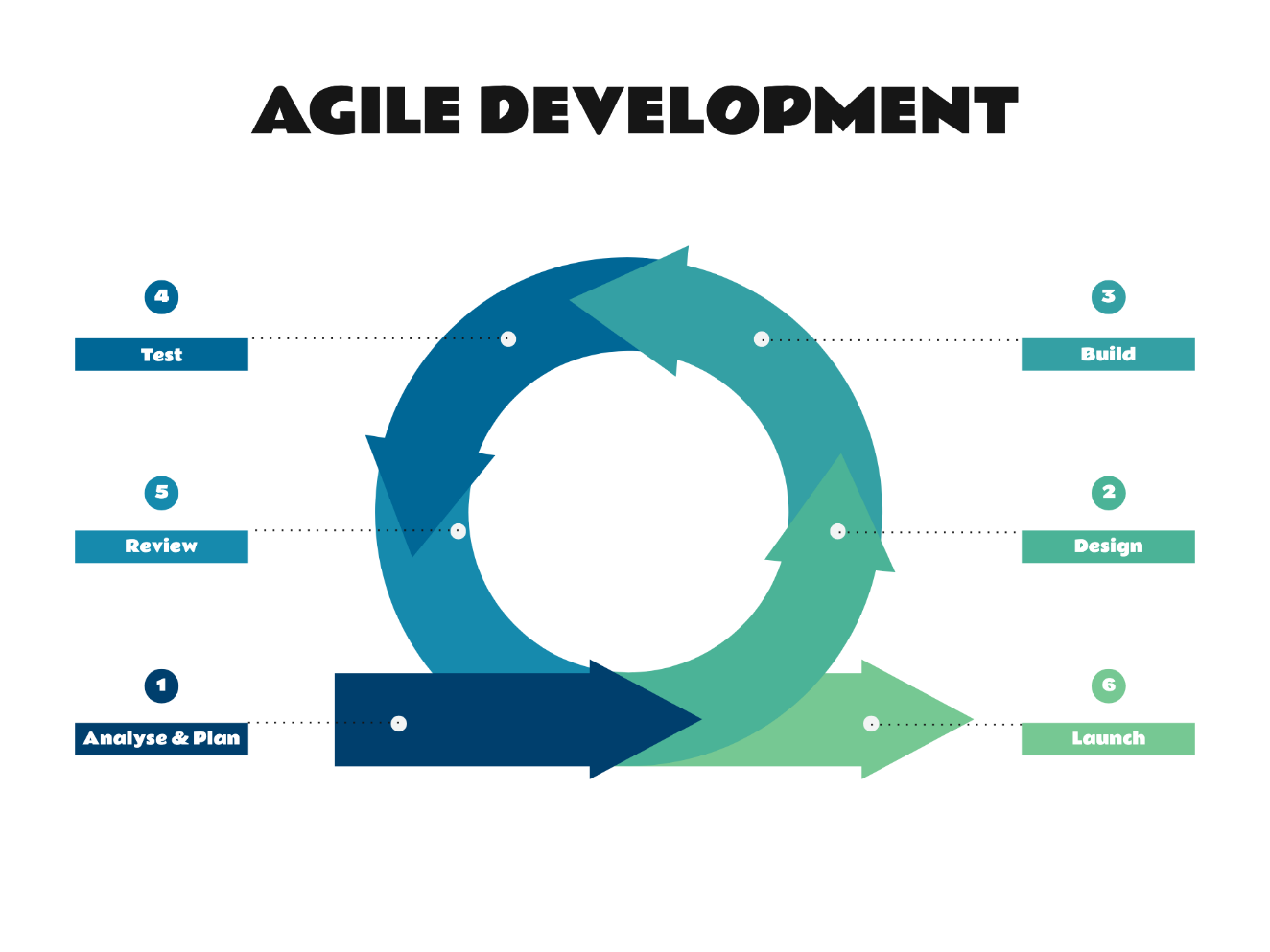


Figure 2‑4: The Agile Development Process (Doherty, A., 2018)

In terms of critiques of these methodologies, one potential issue is that they can be time-consuming and resource intensive, particularly for smaller development teams or those with limited budgets. Additionally, user research and usability testing can be challenging to conduct effectively, particularly if the target user group is diverse or difficult to recruit for research studies. However, these methodologies are generally considered to be essential for effectively developing a mobile application. This is important for a mobile application like ours that is designed to meet a specific need and requires a deep understanding of its users’ motivations and behaviours.

## Conclusion

This literature review highlights the pressing issue of food wastage and the potential of food-saver mobile applications to address it. By providing consumers with a tool to reduce food wastage and save money, this mobile application has the potential to make a significant impact on the economic, environmental, and societal implications of food wastage.

This review combines the existing research on food-saver mobile applications and their impact on reducing food wastage, as well as the features and effectiveness of various applications in different contexts.

The findings suggest that food-saver mobile applications are a promising solution to reducing food waste, however; there is still much to be learned about when it comes to their effectiveness and impact in different contexts.

Overall, this review contributes to the growing body of knowledge on reducing food wastage, and it provides insights for further research and development on food-saver mobile applications to address this issue.

# System Modelling and Architectural Design

## Introduction

When approaching the design aspect of The Sustainable Spoonful mobile application, we needed to consider the user experience and user interface design as well as the architectural design and systems design.

The following sections cover how we plan to approach the UX (User Experience) and UI (User Interface) design, as well as the system modelling and architectural design during the development of this mobile application and outline the structure in which we have tried to ensure user satisfaction and maximize the mobile applications usability.

This will be done using iterative and incremental design processes by way of mapping out our initial plans for the mobile application, gathering requirements, analysis, design, testing, and finally evaluating and then further breaking each section down further into more manageable sections, building and improving on the previous versions (Wrike.com, 2023).

Lastly, with the use of iterative design, it enables flexibility, adaptability, and the ability to address any changes that need to be made to this project.

## Presentation Layer

### User Experience considerations

**The factors that must be considered to enhance the users’ experience are as follows:**

* **Navigation:** How experienced is the user when it comes to using desktop and mobile applications (Whitten & Bentley, 2007)? Can the user be deemed as an expert user or a novice user (Whitten & Bentley, 2007)? This can affect the overall feel of the mobile application as the language used, success messages, warning messages, and error messages differ between an expert user and a novice user (Whitten & Bentley, 2007). In this case, the users of this food-saver mobile application will most likely be novice users. Extra care must be taken to ensure that they are provided with clear and concise instructions to navigate through the mobile application. Users must be able to easily and seamlessly navigate between the different features of the mobile application. Error messages (such as an account or product not being found) and warning messages (such as an incorrect password being entered) must be handled and written in such a way that the user understands what actions to take. These messages can help users to prevent unwanted or irreversible actions (such as the deletion of their account) from occurring (Whitten & Bentley, 2007).
* **System Compatibility:** What operating system will the mobile application support (Whitten & Bentley, 2007)? Will it run on Android or iOS, or perhaps on both? This food-saver mobile application will be provided to users using the Android operating system and will run on devices running at minimum Android 5. Which display resolutions will the mobile application support (Whitten & Bentley, 2007)? At this current point in time, the mobile application will not target specific display resolutions. However, the mobile application will adapt based on the users’ device.
* **Visual appeal:** The mobile application must be visually appealing. Appropriate colours, icons and language must be used to enhance the users’ experience.
* **Feedback and support:** There must be options for the user to provide feedback and request support. For this mobile application, there will be options for the user to provide feedback and request support in the form of a ticket.

#### Fact-Finding Technique(s)

The fact-finding techniques used were observation, research, and questionnaires (Tilley & Rosenblatt, 2017). Observation provided the development team with first-hand experience about how businesses and restaurants operate in South Africa. It enabled the team to understand the current practices, challenges faced and potential areas for improvement when it comes to the issue of food wastage. Conducting research provided the development team with a broader understanding of the impact of food wastage on the economy, its people, and the environment. The research assisted the team in identifying the significance of the problem and the implications it has for South Africa. Researching is a valuable fact-finding technique to gather existing knowledge and data that can be used to inform the design and development process (Tilley & Rosenblatt, 2017). Questionnaires were used to gather specific information from the people of South Africa regarding how they perceived food wastage and whether they experienced it first-hand. The semi-structured questionnaires enabled the team to gather data on how people were currently dealing with the issue of food wastage and what their expectations for addressing it in the future were. Questionnaires are an important fact-finding technique used to gather valuable information from a wider audience (Tilley & Rosenblatt, 2017).

#### Analysis of User Requirements

**The facts gathered were as follows:**

During observation, it was found that restaurants and businesses throw out food products that were close to the end of their shelf-life, were not sold at the end of the day or were bought in excess. This led to a loss in profit. As a result, businesses and restaurants must increase the prices of their food products to cover their losses. Businesses and restaurants need a system where they can sell the food products to customers at a discounted price or donate the products to a local charity if it is not collected at the end of the day.

During research, it was found that about 1/3rd of all food in South Africa is wasted and thrown out. It was also found that farms would throw out crops that they deemed were not fit for sale. It has negatively affected the economy and led to the inflation of food prices. The agricultural sector needs a system where they can sell their produce at a reduced price to consumers instead of throwing it out.

After summarizing the results from the questionnaires, the most common responses found were:

* 65.8% of respondents were between the ages of 18 to 30. This is seen in the diagram below:

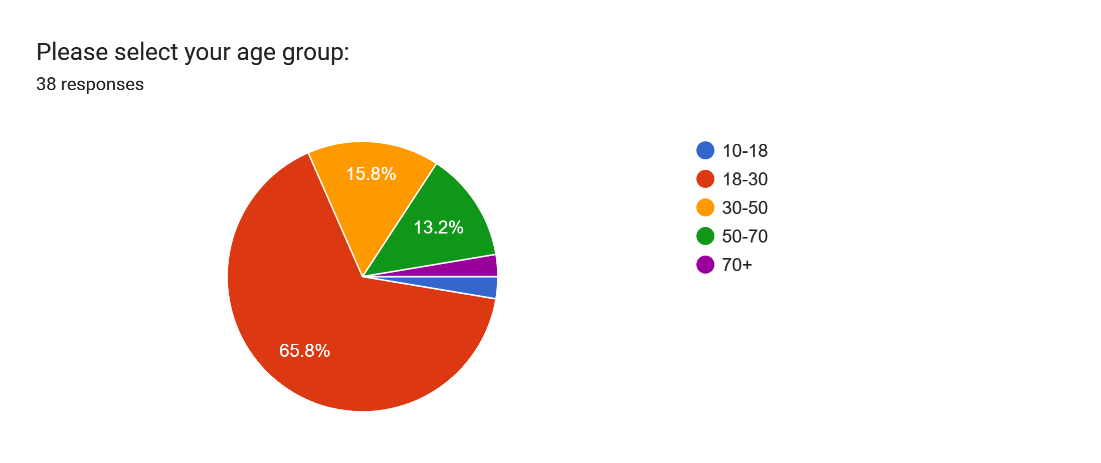


Figure 3‑1: Ages of Respondents.

* 76.3% of respondents found that it was difficult to buy healthy food at an affordable price. This is seen in the diagram below:

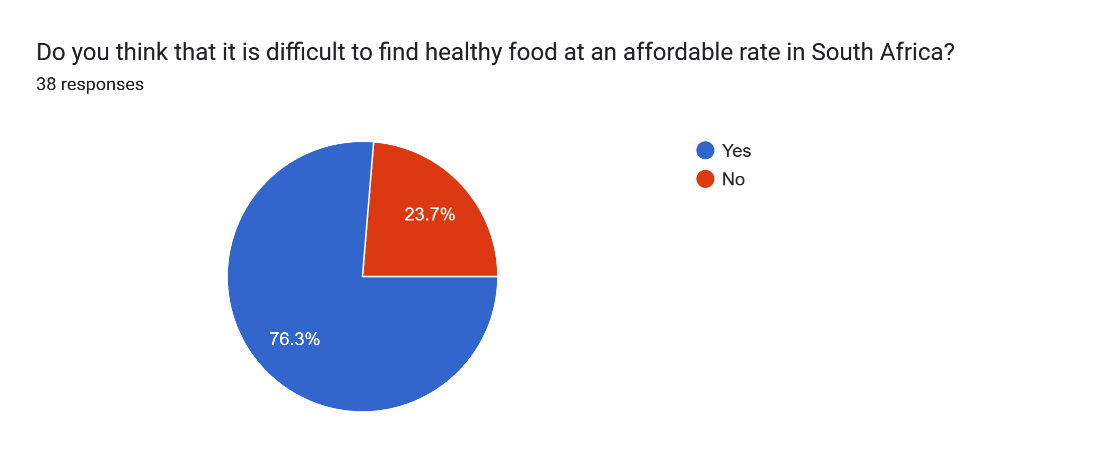


Figure 3‑2: Determining whether finding healthy and affordable food was difficult.

* The reasons why it was difficult to find healthy and affordable food were:
  + Unhealthy food is sold at a lower price than healthy food. This could be due to unhealthy food containing preservatives and extra ingredients to prolong its shelf-life as well as having an overall lower cost compared to its healthier counterparts.
  + The cost of transporting healthy food while ensuring that the food remained fresh and presentable to the consumer further increased its cost when it reached store shelves.
* The most common methods used to cut down on food costs were:
  + Purchasing food products when it was on special.
  + Buying food products in bulk, repackaging them, and freezing them. The food will then only be consumed when it is needed.
  + 97.4% of respondents have not used any food-saver mobile applications. This is seen in the diagram below:



Figure 3‑3: Determining whether food-saver mobile applications have been used before.

* All respondents expressed that they want a food-saver mobile application to be developed.
* Respondents wish that the mobile application:
* Shows restaurants, stores and farms that are offering discounts on their food products.
* Has in-person orders that let customers scan a discount code in a physical store and then collect their items.
* Uses simple and easy-to-understand language.
* Enables customers to pay for their orders in cash and does not request their card information.
* Donates food products that were not sold at the end of the day to the less fortunate.

**The user requirements have been taken into consideration and have been summarised based on their level of priority below:**

* Display food products that are near the end of their shelf life at a discounted price.
* Search for restaurants/stores that are offering products at discounted prices.
* Search for specific products that are at a discounted price.
* Place an order and receive a discount code to scan at the store.
* Pay for the order in cash.
* Donate products that have not been sold at the end of the day to the less fortunate.

#### Tools and Diagrams used

There are many tools used for creating diagrams; these tools include Draw.io, Microsoft Visio and Lucidchart.

For this project, Draw.io has been used to create most of the diagrams.

Draw.io is a free-to-use web-based tool that enables designers to create and edit various types of diagrams, such as use-case diagrams, dataflow diagrams, sequence diagrams and context diagrams, among others (draw.io, 2023).

A user flow diagram can be used to illustrate the login and register process for this mobile application.

The user flow diagram is shown below:

**User Flow Diagram:**

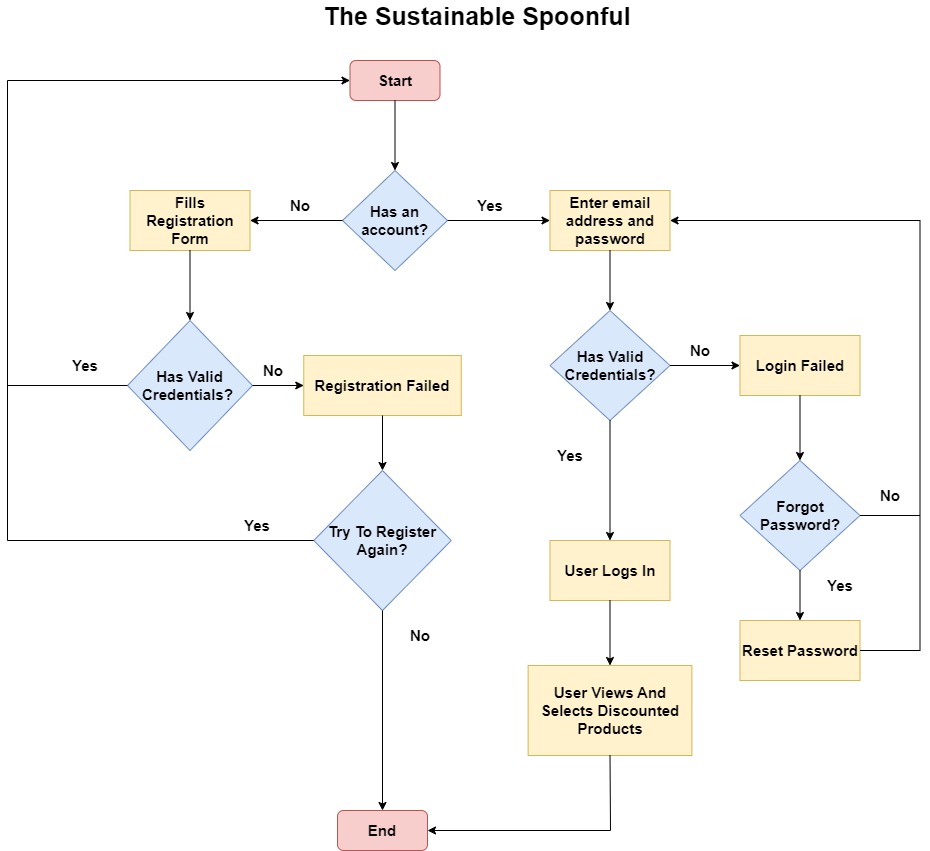


Figure 3‑4: User Flow Diagram for The Sustainable Spoonful System (Cabrera, I., 2022).

### User Interface Design

#### Designs

**Wireframe Designs:**

A link to the wireframe designs can be found here (designshack,2023): [Wireframe Designs](https://www.figma.com/proto/gftl8WufelMBSRWUiJc7YZ/The-Sustainable-Spoonful?type=design&node-id=132-2047&scaling=min-zoom&page-id=132%3A2047)

A screen shot of a phone

Description automatically generated with medium confidence

Figure 3‑5: Landing Page when the user opens the application for the first time.

A screen shot of a phone

Description automatically generated with low confidence

Figure 3‑6: Registration Page so that users can create an account.

A screen shot of a login screen

Description automatically generated with medium confidence

Figure 3‑7: Login Page so that users can login using the account they have just created.

A screen shot of a login screen

Description automatically generated with low confidence

Figure 3‑8: Forgot Password page for when a user wishes to reset their password.

A screen shot of a password reset

Description automatically generated with low confidence

Figure 3‑9: Confirmation screen that shows that a password reset email was sent to the user.

A screen shot of a phone

Description automatically generated with medium confidence

Figure 3‑10: Home Page for the Sustainable Spoonful Mobile Application.

A screenshot of a cell phone

Description automatically generated with medium confidence

Figure 3‑11: Discounted Products Page for users to search for discounted products based on the store and location.

A screenshot of a phone

Description automatically generated with medium confidence

Figure 3‑12: Discounts listed for a specific store.

A screenshot of a phone

Description automatically generated with medium confidence

Figure 3‑13: discount code for the selected discounted product that users can scan at the store to collect their item.

A screenshot of a login screen

Description automatically generated with medium confidence

Figure 3‑14: Account Page for a user to manage their account.

A screen shot of a login screen

Description automatically generated with medium confidence

Figure 3‑15: Prompting the user to log in first before they can edit their account details.

A screenshot of a login screen

Description automatically generated with medium confidence

Figure 3‑16: User logging in.

A screenshot of a login screen

Description automatically generated with low confidence

Figure 3‑17: Edit Account Page.

A screen shot of a phone

Description automatically generated with medium confidence

Figure 3‑18: Confirmation that the user has updated their account details.

A screen shot of a login screen

Description automatically generated with low confidence

Figure 3‑19: Prompting the user to login before they delete their account.

A screenshot of a login screen

Description automatically generated with medium confidence

Figure 3‑20: Logging in.

A screen shot of a phone

Description automatically generated with low confidence

Figure 3‑21: Delete Account Page.

A screen shot of a phone

Description automatically generated with medium confidence

Figure 3‑22: Confirmation that the user has deleted their account.

A screen shot of a phone

Description automatically generated with medium confidence

Figure 3‑23: Prompting the user to log out of their account or go back.

A screen shot of a phone

Description automatically generated with medium confidence

Figure 3‑24: Confirmation that the user has logged out of their account.

#### Design Tools and Techniques

Figma has been used to create the prototype and initial designs for this mobile application.

Figma is a free-to-use cloud-based editor that enables system designers to create wireframes, mock-ups as well as user interfaces for their websites and mobile applications (Figma, 2023). It also can create design teams that enable multiple people to work on a design at the same time (Figma, 2023). Designers in a team can add comments and suggestions to specific areas of the design (Figma, 2023). Animations can be made, such as having a drop-down menu that can change colours (Figma, 2023).

Figma was chosen as the most preferred design tool for the reasons above.

In addition to this, the following design techniques were taken into consideration (Whitten & Bentley, 2007):

* All the fields present in the forms (such as the registration and login forms) in this mobile application have labels.
* Every output has a title and provides additional information underneath to guide the user as to what actions to take next.
* Headings are aligned on each page where possible.
* Simple and easy-to-understand language is used.

#### Designs Look and Feel (Aesthetic, Business Colours, etc.)

The following colour palette was used for this mobile application:

A picture containing screenshot, text, colorfulness, design

Description automatically generated

Figure 3‑25: Colour Palette for The Sustainable Spoonful.

The following image is used as the logo for The Sustainable Spoonful mobile application:

A picture containing circle, logo, symbol, font

Description automatically generated

Figure 3‑26: Logo for The Sustainable Spoonful.

##### Design Iteration 1

The link to the first prototype can be found here: [Design 1](https://www.figma.com/proto/gftl8WufelMBSRWUiJc7YZ/The-Sustainable-Spoonful?type=design&node-id=3-3&scaling=scale-down&page-id=0%3A1&starting-point-node-id=3%3A3)

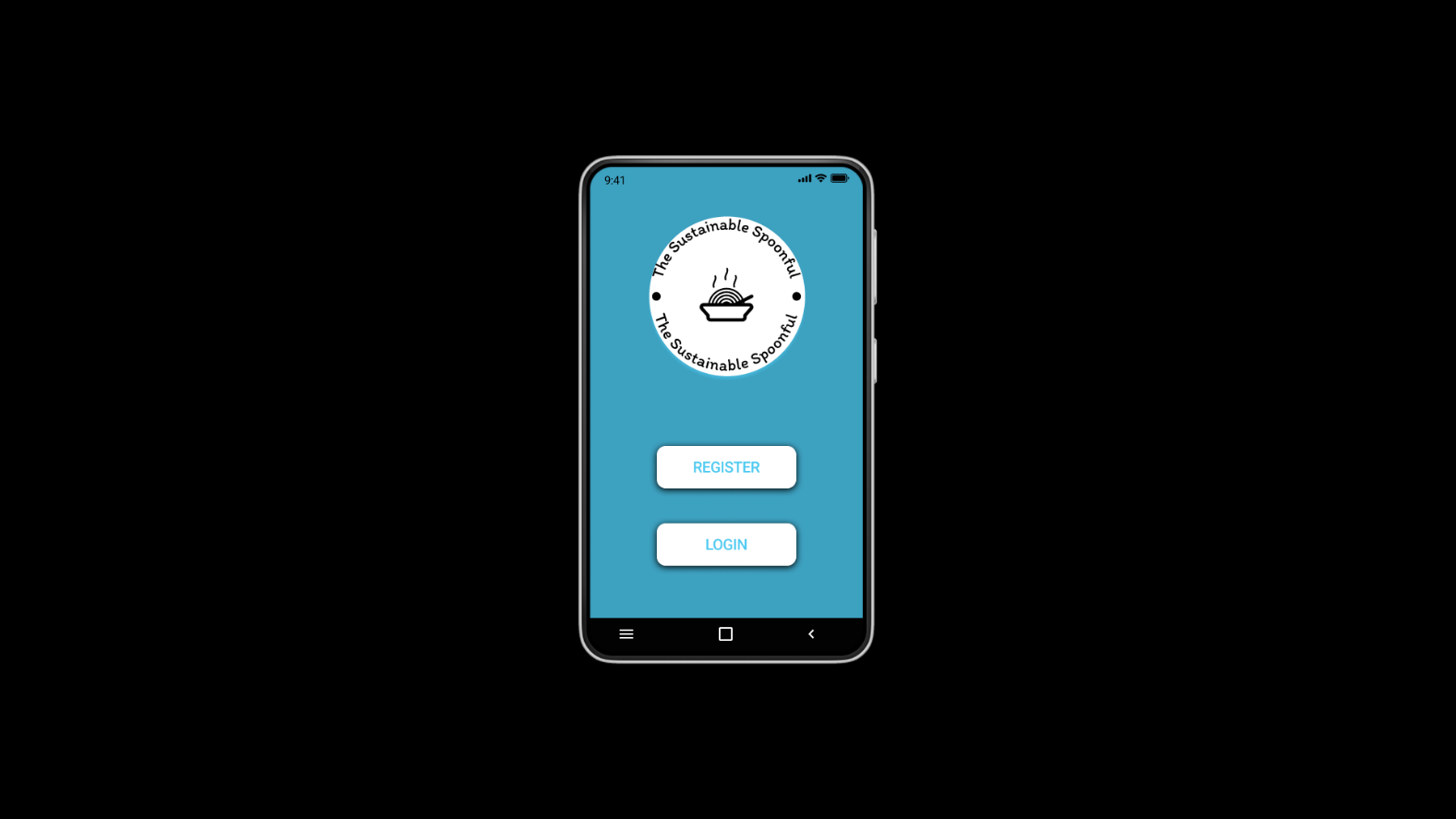


Figure 3‑27: Landing Page when the user opens the application for the first time.

A screen shot of a phone

Description automatically generated with low confidence

Figure 3‑28: Registration Page so that users can create an account.

A screen shot of a cell phone

Description automatically generated with medium confidence

Figure 3‑29: Login Page so that users can login using the account they have just created.

A cell phone with a login screen

Description automatically generated with low confidence

Figure 3‑30: Forgot Password page for when a user wishes to reset their password.

A screen shot of a cell phone

Description automatically generated with medium confidence

Figure 3‑31: Confirmation screen that shows that a password reset email was sent to the user.

A picture containing screenshot, gadget, mobile phone, multimedia

Description automatically generated

Figure 3‑32: Home Page for the Sustainable Spoonful Mobile Application.

A cell phone with text on the screen

Description automatically generated with low confidence

Figure 3‑33: Discounted Products Page for users to search for discounted products based on the store and location.

A picture containing screenshot, mobile phone, gadget, communication device

Description automatically generated

Figure 3‑34: Discounts listed for a specific store.

A cell phone with qr code

Description automatically generated with medium confidence

Figure 3‑35: discount code for the selected discounted product that users can scan at the store to collect their item.

A screen shot of a cell phone

Description automatically generated with medium confidence

Figure 3‑36: Account Page for a user to manage their account.

A screen shot of a cell phone

Description automatically generated with medium confidence

Figure 3‑37: Prompting the user to log in first before they can edit their account details.

A picture containing screenshot, mobile phone, gadget, mobile device

Description automatically generated

Figure 3‑38: User logging in.

A screen shot of a phone

Description automatically generated with medium confidence

Figure 3‑39: Edit Account Page.

A screen shot of a cell phone

Description automatically generated with medium confidence

Figure 3‑40: Confirmation that the user has updated their account details.

A screen shot of a cell phone

Description automatically generated with medium confidence

Figure 3‑41: Prompting the user to login before they delete their account.

A screen shot of a cell phone

Description automatically generated with low confidence

Figure 3‑42: Logging in.

A screen shot of a cell phone

Description automatically generated with medium confidence

Figure 3‑43: Delete Account Page.

A picture containing mobile phone, gadget, screenshot, text

Description automatically generated

Figure 3‑44: Confirmation that the user has deleted their account.

A screen shot of a cell phone

Description automatically generated with low confidence

Figure 3‑45: Prompting the user to log out of their account or go back.

A screen shot of a cell phone

Description automatically generated with low confidence

Figure 3‑46: Confirmation that the user has logged out of their account.

##### Design Iteration 2

After receiving feedback from users, the design has been adjusted accordingly.

The following changes were made:

* We have added clear and concise error messages to our user interface so that the user will be guided as to what actions to take next when they are navigating through the mobile application.
* We have added more screens in our mobile application; these screens are meant to guide the user and provide the user with a user-friendly experience.
* We have adjusted our colour palette so that it provides a more pleasant user experience.

The link to the second prototype can be found here: [Design 2](https://www.figma.com/proto/gftl8WufelMBSRWUiJc7YZ/The-Sustainable-Spoonful?type=design&node-id=143-4585&scaling=min-zoom&page-id=132%3A855&starting-point-node-id=143%3A4585)

A screenshot of a blue screen

Description automatically generated with low confidence

Figure 3‑47: Landing Page when the user opens the application for the first time.

A screen shot of a phone

Description automatically generated with low confidence

Figure 3‑48: Registration Page so that users can create an account.

A screen shot of a phone

Description automatically generated with medium confidence

Figure 3‑49: Login Page so that users can login using the account they have just created.

A screen shot of a phone

Description automatically generated with medium confidence

Figure 3‑50: Forgot Password page for when a user wishes to reset their password.

A screen shot of a phone

Description automatically generated with medium confidence

Figure 3‑51: Confirmation screen that shows that a password reset email was sent to the user.

A screen shot of a phone

Description automatically generated with medium confidence

Figure 3‑52: Home Page for the Sustainable Spoonful Mobile Application.

A screenshot of a phone

Description automatically generated with medium confidence

Figure 3‑53: Discounted Products Page for users to search for discounted products based on the store and location.

A screenshot of a cell phone

Description automatically generated with medium confidence

Figure 3‑54: Discounts listed for a specific store.

A screen shot of a cell phone

Description automatically generated with medium confidence

Figure 3‑55: discount code for the selected discounted product that users can scan at the store to collect their item.

A screen shot of a phone

Description automatically generated with medium confidence

Figure 3‑56: Account Page for a user to manage their account.

A screenshot of a login screen

Description automatically generated with low confidence

Figure 3‑57: Prompting the user to log in first before they can edit their account details.

A screen shot of a login screen

Description automatically generated with low confidence

Figure 3‑58: User logging in.

A screen shot of a phone

Description automatically generated with low confidence

Figure 3‑59: Edit Account Page.

A screen shot of a phone

Description automatically generated with medium confidence

Figure 3‑60: Confirmation that the user has updated their account details.

A screenshot of a phone

Description automatically generated with medium confidence

Figure 3‑61: Prompting the user to login before they delete their account.

A screen shot of a login screen

Description automatically generated with medium confidence

Figure 3‑62: Logging in.

A screen shot of a phone

Description automatically generated with medium confidence

Figure 3‑63: Delete Account Page.

A screen shot of a phone

Description automatically generated with medium confidence

Figure 3‑64: Confirmation that the user has deleted their account.

A screenshot of a phone

Description automatically generated with medium confidence

Figure 3‑65: Prompting the user to log out of their account or go back.

A screen shot of a phone

Description automatically generated with medium confidence

Figure 3‑66: Confirmation that the user has logged out of their account.

## Business Layer

### Operations/Process Models Design

**Use-Case Diagram – Iteration 1:**

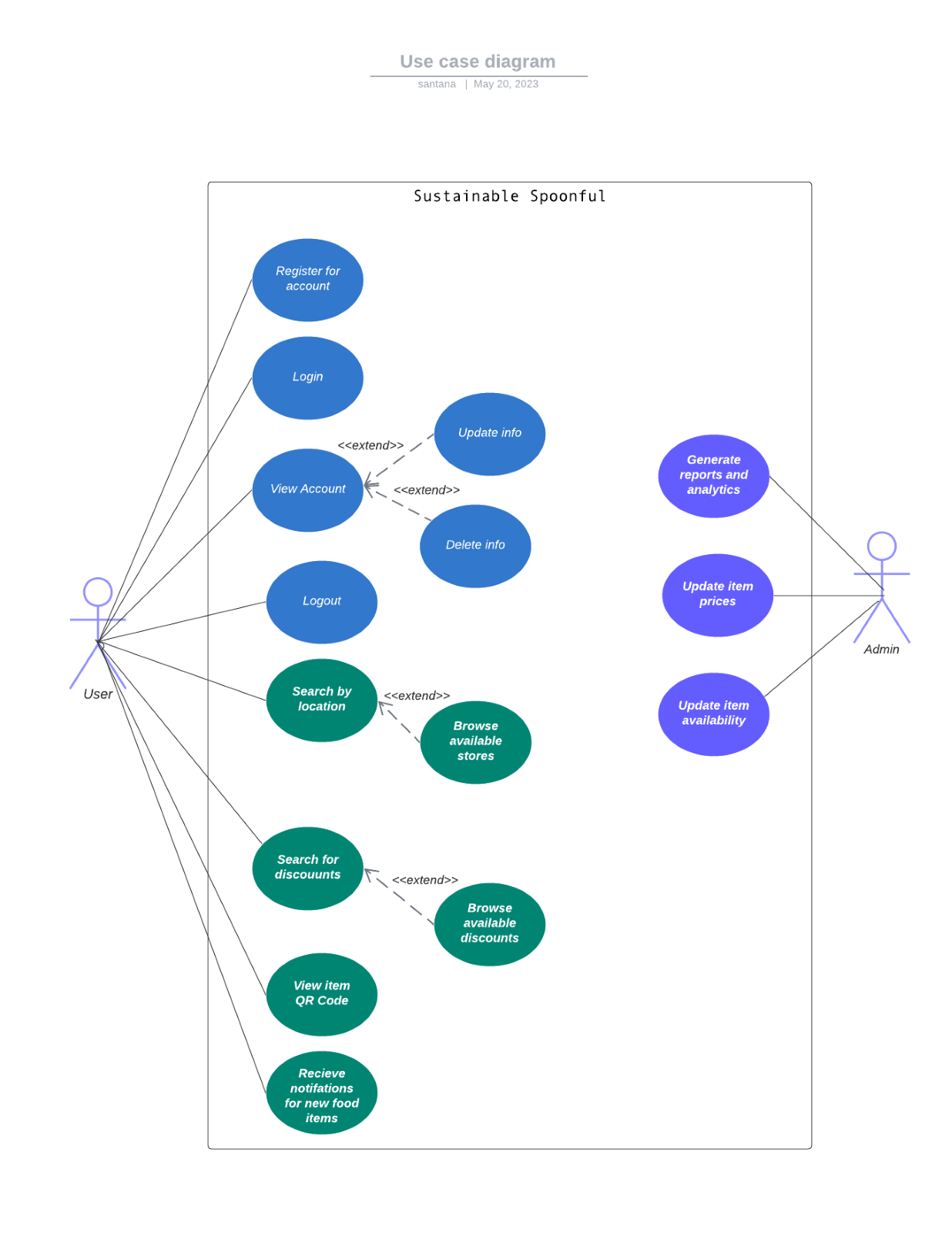


Figure 3‑67: Use Case Diagram – Iteration 1 for The Sustainable Spoonful System (Tilley & Rosenblatt, 2017).

**Use-Case Diagram – Iteration 2:**

**The actors and their use cases have been identified below:**

**Administrator:**

* Login (includes Authentication)
* Manages Stores (includes Adds Store Name, Adds Store Location and Adds Store Products)
* Manages Products (includes Adds Products, Edits Products, Deletes Products and Sets Products as Donated)
* Receives Feedback
* Logout

**Users:**

* Login (includes Authentication)
* Manages Account (includes View Account, Edit Account Details and Delete Account)
* Searches For Products (includes Searches by Store and Searches by Location)
* Places an Order (includes Receiving a Discount Code)
* Sends Feedback
* Logout

**Guest User:**

* Create Account (includes Add Account Details)

The use-case diagram is shown below:

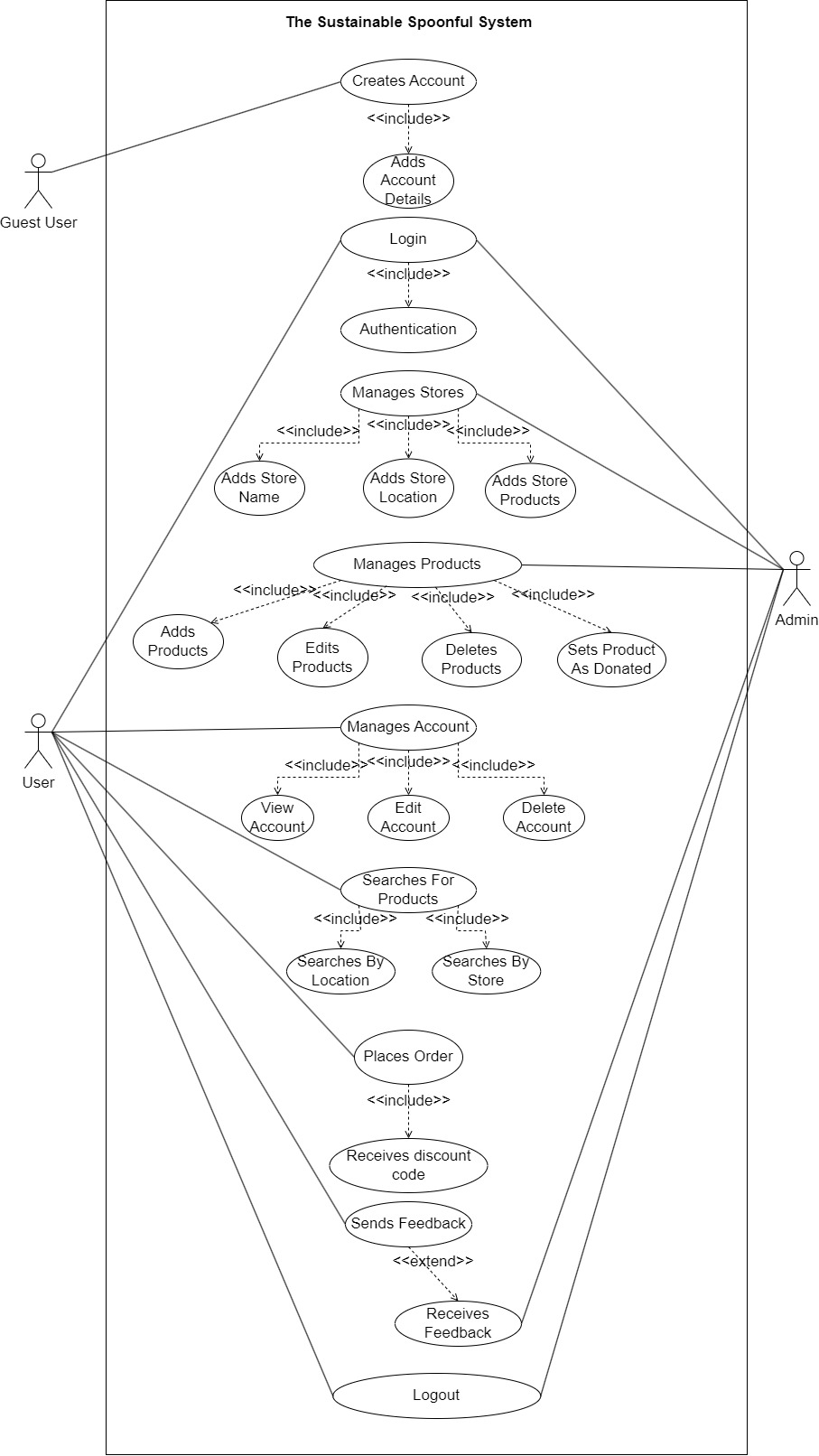
****

Figure 3‑68: Use Case Diagram – Iteration 2 for The Sustainable Spoonful System (Tilley & Rosenblatt, 2017).

**Sequence Diagram – Iteration 1:**

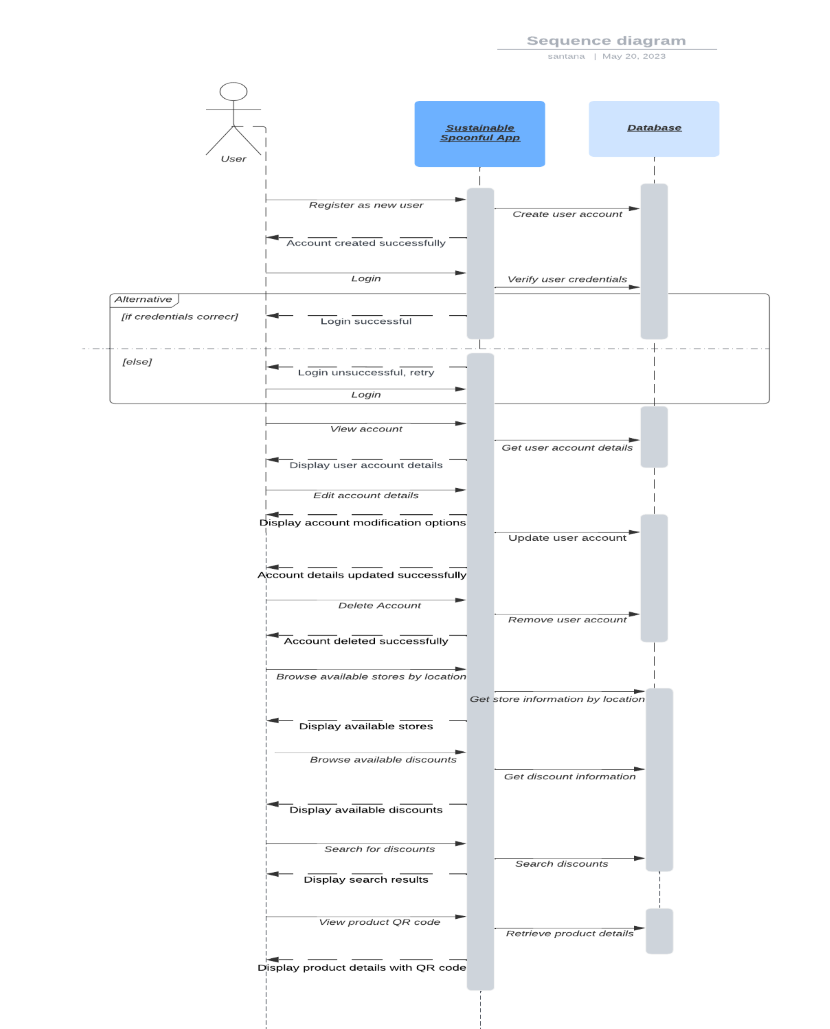


Figure 3‑69: Sequence Diagram – Iteration 1 for The Sustainable Spoonful System (Lucidchart, 2022).

**Sequence Diagram – Iteration 2:**

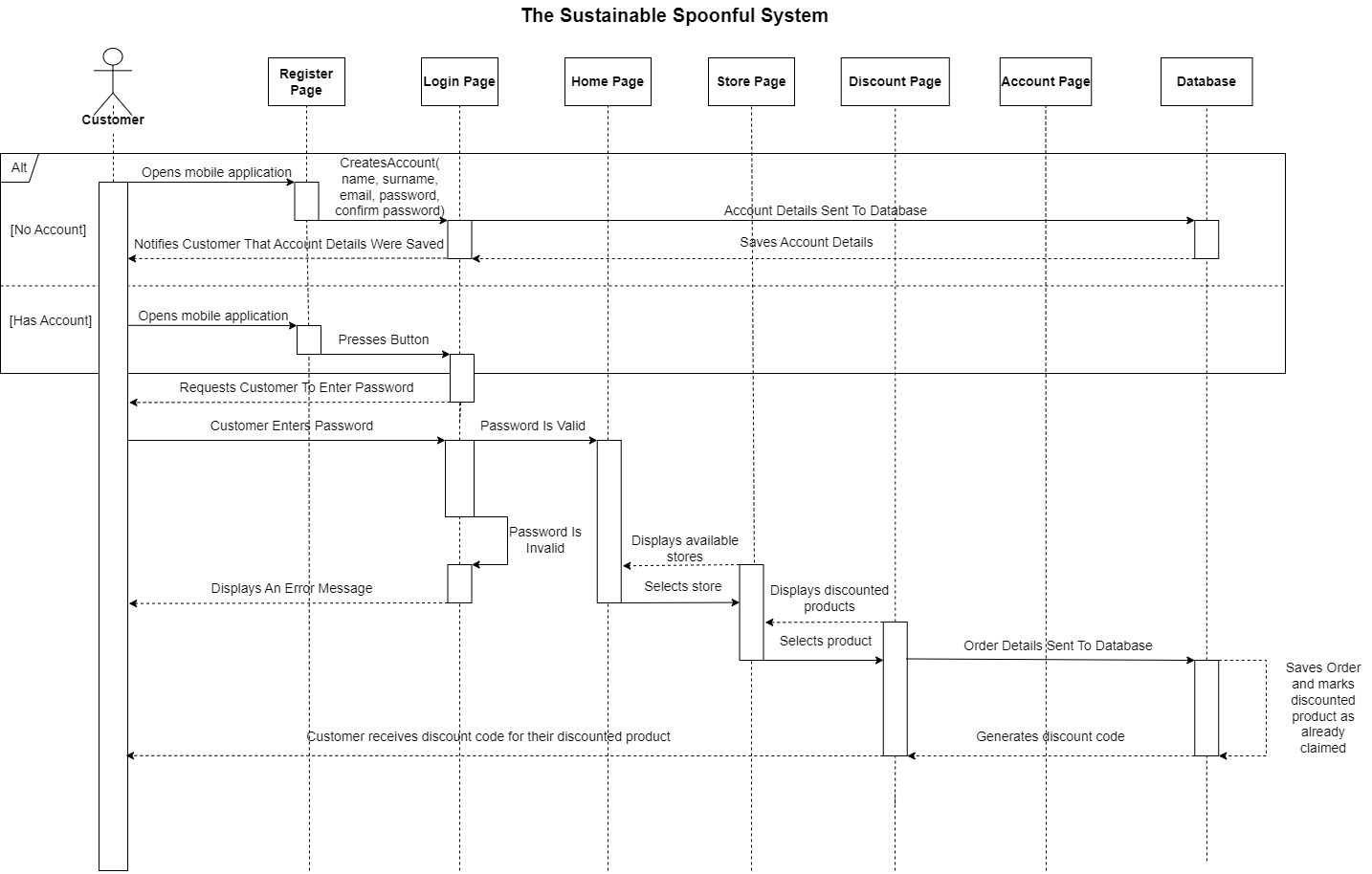
****

Figure 3‑70: Sequence Diagram – Iteration 2 for The Sustainable Spoonful System (Lucidchart, 2022)

The sequence diagram will help illustrate the interactions between the user and The Sustainable Spoonful mobile application, as well as the interactions between the mobile application and the database. It shows the flow of actions and information within the mobile application’s operations and process flow (Athuraliya, 2022).

Explanation of the sequence diagram:

1. **User Registration:**

* If the user does not have an account:
  + The customer opens the mobile application and enters their details on the registration page.
  + These details include:
    - Name
    - Surname
    - Email address
    - Password
    - Confirm Password
  + Their details are sent to the database, where their details are saved, and the customer is notified that their account was created.

1. **Log in to the mobile application:**
   * If the user has an account:
     + The customer opens the mobile application, presses the login button and enters their details.
     + These details include:
       - Email address
       - Password
     + Their details are validated, and if they are correct, then the customer is directed to the home page. If they are incorrect, then an error message will be displayed to the customer.
2. **Browse available stores:**

* From the home page, the customer may browse the available stores or search for a specific store.
  + The customer will also be able to search by their location.

1. **Browse available discounts:**

* Once a store has been selected, the customer will be able to browse discounts on offers or search for a particular product.
* Once the customer finds a discount that appeals to them, they can select it. Their order will be sent to the database, where it will be marked as claimed under their account, and a discount code will be generated.
* The discount code will then be sent to the customer.

### Data Handling Operation Design

To ensure that the data in The Sustainable Spoonful mobile application is validated and secured, we can implement the following measures:

**Data Security:**

1. **Secure Storage**: Implementing secure storage mechanisms can help to protect sensitive data (Smart, 2020). This can be done by utilising Android’s built-in security features (such as SharedPreferences) for smaller amounts of data and encrypted databases (such as SQLCipher) for larger and more sensitive data sets (Smart, 2020).
2. **User Authentication**: Implementing a strong user authentication system ensures that only authorised users have access to the mobile application and its features (DesignRush, 2023). Secure authentication protocols such as OAuth (Open Authorization) and JWT (JSON Web Tokens) can be used to securely authenticate users and protect their credentials (DesignRush, 2023).
3. **Encryption**: Implementing encryption techniques can help to protect sensitive data when it is being sent and stored (Loshin, 2022). This can be done by using Hypertext Transfer Protocol Secure (HTTPS) (Loshin, 2022). HTTPS ensures that the network communication between the mobile application and the server is secured (Loshin, 2022). Encryption algorithms such as the Advanced Encryption Standard (AES) can be used to encrypt the data that is stored in the database (Loshin, 2022).
4. **Access Control**: By enforcing strict access control policies ensures that the data is only accessible based on the users’ specific roles and permissions (Martin, 2019). For our mobile application, we must ensure that sensitive operations and data are only accessible to authorised individuals (such as giving an administrator permission to manage user accounts) (Martin, 2019).
5. **Regular Updates**: By keeping the mobile application and its underlying frameworks, libraries, and dependencies up to date ensures that any known vulnerabilities are detected and with accordingly (Android Developers, 2023). Regularly updating the mobile application helps to protect against potential security risks (such as not properly escaping user input, which can lead to malicious individuals injecting code to retrieve confidential data) (Android Developers, 2023).

**Data Validation:**

1. **Input validation**: Implementing strong input validation techniques can ensure that user input meets the required criteria and specified format (such as having a valid password or email address) (OWASP, 2019). Input validation can be implemented by ensuring that user input (such as usernames, phone numbers, passwords, and email addresses) is properly escaped (OWASP, 2019). Escaping user input can help to prevent malicious individuals from inserting code to retrieve confidential information or potentially crashing the mobile application (OWASP, 2019).
2. **Server-side validation**: Performing server-side validation can help to validate and verify the integrity of the data that is received from the mobile application (IBM, 2022). Implementing server-side validation rules and check ensures that data is consistent and accurate (IBM, 2022).
3. **Data Sanitisation**: Implementing data sanitisation techniques helps to protect against common security threats such as SQL (Structured Query Language) injection attacks and cross-site scripting (XSS) attacks (OWASP, 2021). User input can be sanitised by removing or escaping special characters that could be used for malicious purposes (OWASP, 2021).
4. **Error Handling**: Implementing strong error handling and reporting mechanisms can help to identify and address any data validation errors (such as entering an invalid password in a login form) (OWASP, 2023). Error handling can be implemented by displaying meaningful error messages to users (such as ‘The password that you entered is invalid, please try again’) (OWASP, 2023). This can show the specific validation issue and guide users as to how to correct it (OWASP, 2023).
5. **Regularly Auditing and Monitoring**: Implementing logging and auditing mechanisms can be used to track and monitor data access, changes, and any system activities (IBM, 2023). Regularly reviewing logs and monitoring system behaviour can help to detect any suspicious or unauthorised activities (IBM, 2023).

**Exception Handling Techniques:**

1. **Input Validation for Exceptions** (Greb, 2017)**:**

* Implement checks for user inputs, such as in forms where data needs to be submitted.
* Check for invalid or missing data, data that is in an incorrect format, or data that violates predefined rules (such as a user not meeting a specific password length when they are registering an account).
* If data validation fails, catch the exception at the appropriate level and display an error or exception code as well as a descriptive and meaningful error message to the user.

1. **Database Validation Exceptions** (Android Developers, 2023)**:**

* When interacting with the database, ensure that data integrity checks (such as ensuring that email addresses entered are unique) are performed.
* Validate data before storing or updating it in the database.
* Catch the exception and handle it accordingly, such as displaying an error message to the user or rolling back the transaction in the database.

1. **Network Communication Exceptions** (OWASP, 2021)**:**

* Handle exceptions related to network communication (such as timeouts, connection failures, or invalid server responses).
* Implement mechanisms to retry or fallback strategies to address temporary network issues.

1. **Error Logging** (Kazi, 2021)**:**

* Implement an error logging system to keep a record of exceptions or errors that were encountered during data handling.
* Log relevant information such as the type of exception, its stack trace, its timestamp, and any relevant data.
* Use a logging framework to capture and store error logs that can be used for debugging and troubleshooting purposes.

1. **Graceful Error Handling** (Hope, 2017)**:**

* Provide a user-friendly error handling system to handle exceptions and errors.
* Display meaningful error messages or notifications that inform the user about the issue that they encountered and suggest possible solutions.

### Diagrams

**Functional Decomposition Diagram – Iteration 1:**

A picture containing text, screenshot, font, diagram

Description automatically generated

Figure 3‑71: Functional Decomposition Diagram - Iteration 1 for The Sustainable Spoonful System (Tilley & Rosenblatt, 2017).

**Functional Decomposition Diagram – Iteration 2:**

*A picture containing diagram, technical drawing, plan, sketch

Description automatically generated*

Figure 3‑72: Functional Decomposition Diagram - Iteration 2 for The Sustainable Spoonful System (Tilley & Rosenblatt, 2017).

An explanation for the functional decomposition diagram:

The functional decomposition diagram helps outline the major functional modules or components of The Sustainable Spoonful mobile application. Each module represents a specific set of operations related to user management, store management, discount management, admin operations, and customer support. This diagram helps visualize the different areas of functionality and their relationships within the mobile application (Neha T, 2021).

Here is an explanation of the functional decomposition diagram:

* **User Management:**
* This module includes operations related to user registration, log in, log out, and account management, and search for as well as view discounted products.
* Guest users can register an account, log in to the mobile application, log out of the mobile application, view their account details, and modify their account by editing account details or deleting their account.
* Additionally, there is additional functionality for users to browse available stores or search by name or location.
* Users can browse available stores to explore their offerings.
* **Admin Operations:**
  + This module is specific to admin users and involves operations related to store management and discounted products management.
  + Admin users can update item availability and prices within the mobile application’s database.
  + This includes:
    - Add stores, update stores, or delete stores.
    - Add products, update products, or delete products.

### Iterative and Incremental Design

We have made multiple iterations of our designs; these designs are used to show the functionality of the mobile application. Further improvements will be made during the development of the mobile application.

Through user feedback, we have made the following improvements:

* For our use-case diagram, we have added another actor (the guest user) and expanded upon existing actors and their use cases.
* For our sequence diagram, we have expanded upon the registration and login process as well as renaming certain functionalities (such as registering an account, selecting a product, and receiving a discount code).
* For our functional decomposition diagram, we have expanded upon its processes.

By building on each iteration, the functionality of the mobile application is improved.

## Data Layer

### Logical Data Models

The steps involved in creating our logical data model are as follows:

1. **Gather business requirements for how the mobile application should process data:**

* Users should be able to create an account and log into the mobile application.
* Users should be able to view discounted items.
* Users should be able to search for discounted items by store, location, or by item name.
* Users should be able to claim discounts.
* The mobile application should display a discount code based on the selected discounted item.
* Users should be able to view and edit their accounts.
* Administrators should be able to manage users, retailer information and discounted items from the database.

1. **Define business processes and map out our logical data model:**

* The mobile application should connect to a database and store the users’ registration information.
* The mobile application should then validate the users’ login information against the information stored in the database.
* Users should be able to search for certain stores or discounted items based on their location. Alternatively, they can simply view discounted items based on the store.
* The user should be able to claim the discount code for a discounted item only once. Once the code is claimed, the discount should be assigned to their account in the database.
* The mobile application should display a unique discount code for each discounted item.

1. **Creating the Context Diagram:**

* The context diagram gives us an overview of the system and how different external entities will interact with it.
* External entities:
  + Customer
  + Guest User
  + Database
  + Administrator
* Data Flows:
* A guest user enters the mobile application and attempts to register an account. Their information is sent to the database, and the database stores their information and creates their account. A confirmation message is sent back to the user.
* The user now attempts to log in to their account on the mobile application. They enter their login details; the information is sent to the database to be validated. Once the user's login credentials are validated against the database, they are logged into the mobile application.
* Once logged into the mobile application, the user may view discounts based on the store or search based on their location. The request is sent to the database, and the information is displayed on the mobile application for the user to view.
* Users may also view or edit their account. The users’ account will then be updated in the database.
* Once the user finds a discounted item that they are interested in, they can select it, and a discount code will be displayed for the user. The discount code claim is sent to the database, and the discount is assigned to their account. Each user may only claim a discount once.

1. **Creating the Entity Relationship Diagram:**

* Entities:
  + Customer
  + Retailer
  + Special
  + Discount Code
  + Customer discount Code
* Relationships:
* Customer and Retailer
* N: N (Many-to-Many relationship) - resolved with junction table.
* Each customer can select discounts from multiple retailers/stores, and each retailer/store can be associated with multiple customers.
* However, the Customer\_Retailer table acts as a junction table between the two tables and stores the customerID and retailerID.
* Retailer and Special
* 1: N (One-to-Many relationship)
* Each retailer/store can have multiple discounted items listed for their store, but each special/discount can only be associated with a particular store.
* Special and discount Code
* 1:1 (One-to-One relationship)
* Each special/discount can only be associated with one discount code since they are meant to be unique.
* Customer and Customer Discount Code
* 1: N (One-to-Many relationship)
* One user may have multiple discount codes claimed for different discounted items, but each discount code may only be associated with one user.
* Discount Code and User Discount Code
* 1: N (One-to-Many relationship)
* Each discount code may be associated with multiple user discount code entries for assorted discounted items, but each user discount code entry may only be associated with one discount code. This allows claimed discount codes to be tracked.

1. **Creating the Data Flow Diagram:**

* External Entities:
  + Guest User
  + Customer
  + Admin
* Processes:
  + Register
  + Login
  + Manage Customer
  + Manage Retailer
  + Manage Discounts
  + Claim discount
* Data Stores:
  + Customer Table
  + Retailer Table
  + Specials Table
  + Discount Code Table
  + Customer Discount Code Table
* Data Flows:
  + The guest user fills in the registration form on the mobile application with their details, and it is sent to the customer table. Registration is confirmed, and the customer can now login into the mobile application.
  + The customer fills in their login details, and their information is validated against the Customer Table. If valid, they are directed to the home page.
  + The admin can view/update/delete Customer information from the Customer Table.
  + The admin can view/update/delete retailer information from the Retailer Table.
  + The admin can view/update/delete discount information from the Specials Table.
  + The Customer can claim discounted items. Once claimed, the discount code generated will be assigned to their Customer ID in the Customer Discount Code table, and the discount code will be displayed to the customer on the mobile application.

### Diagrams, Tools, and Techniques

Draw.io has been used to create the Context diagram and the Entity Relationship diagram, whilst Lucidchart was used to create the Dataflow Diagram.

Draw.io is a free-to-use web-based tool that enables designers to create and edit various types of diagrams, such as use-case diagrams, dataflow diagrams, sequence diagrams and context diagrams, among others ​(draw.io, 2023)​.

While Lucidchart is a web-based application that allows users to collaborate on diagrams with certain limitations when using a free account (Lucidchart, 2023).

The context, entity relationship diagram and data flow diagram are shown below:

**Context Diagram – Iteration 1:**

A picture containing text, diagram, screenshot, circle

Description automatically generated

Figure 3‑73: Context Diagram - Iteration 1 for The Sustainable Spoonful System (Pedriquez, D., 2022).

**Context Diagram – Iteration 2:**

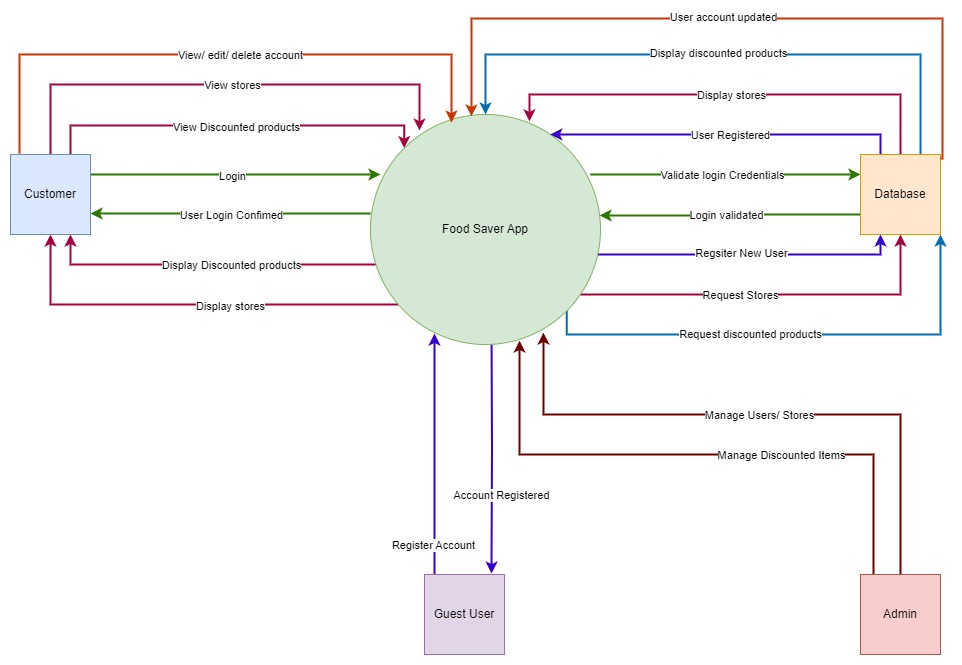
****

Figure 3‑74: Context Diagram - Iteration 2 for The Sustainable Spoonful System (Pedriquez, D., 2022).

The Context diagram (also known as a level 0 Data Flow Diagram) is used to represent a high-level overview of the data flow within a system (Pedriquez, D., 2022). It is commonly used as a reference by engineers, analysts, developers, and stakeholders as it does not require any technical knowledge to understand (Pedriquez, D., 2022).

**Entity Relationship Diagram – Iteration 1:**

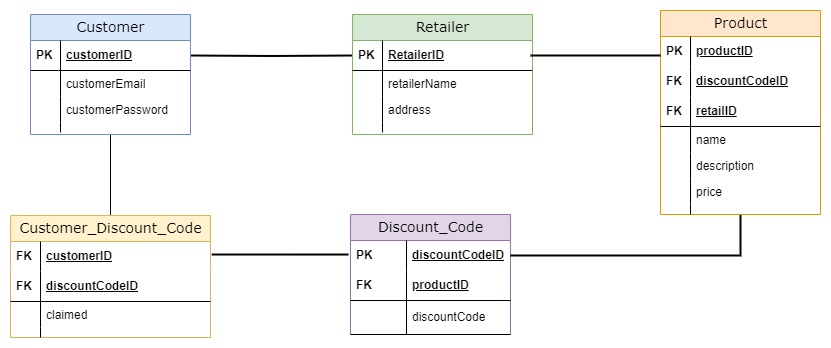


Figure 3‑75: Entity Relationship Diagram - Iteration 1 for The Sustainable Spoonful System (Brumm, B., 2022).

**Entity Relationship Diagram – Iteration 2:**

A picture containing text, diagram, font, line

Description automatically generated

Figure 3‑76: Entity Relationship Diagram - Iteration 2 for The Sustainable Spoonful System (Brumm, B., 2022).

**Entity Relationship Diagram – Iteration 3:**

A picture containing text, font, number, line

Description automatically generated

Figure 3‑77: Entity Relationship Diagram - Iteration 3 for The Sustainable Spoonful System (Brumm, B., 2022).

The Entity Relationship Diagram illustrates how different entities within the system will interact with each other and is often used to design relational database models (Lucidchart, 2023).

**Data Flow Diagram – Iteration 1:**

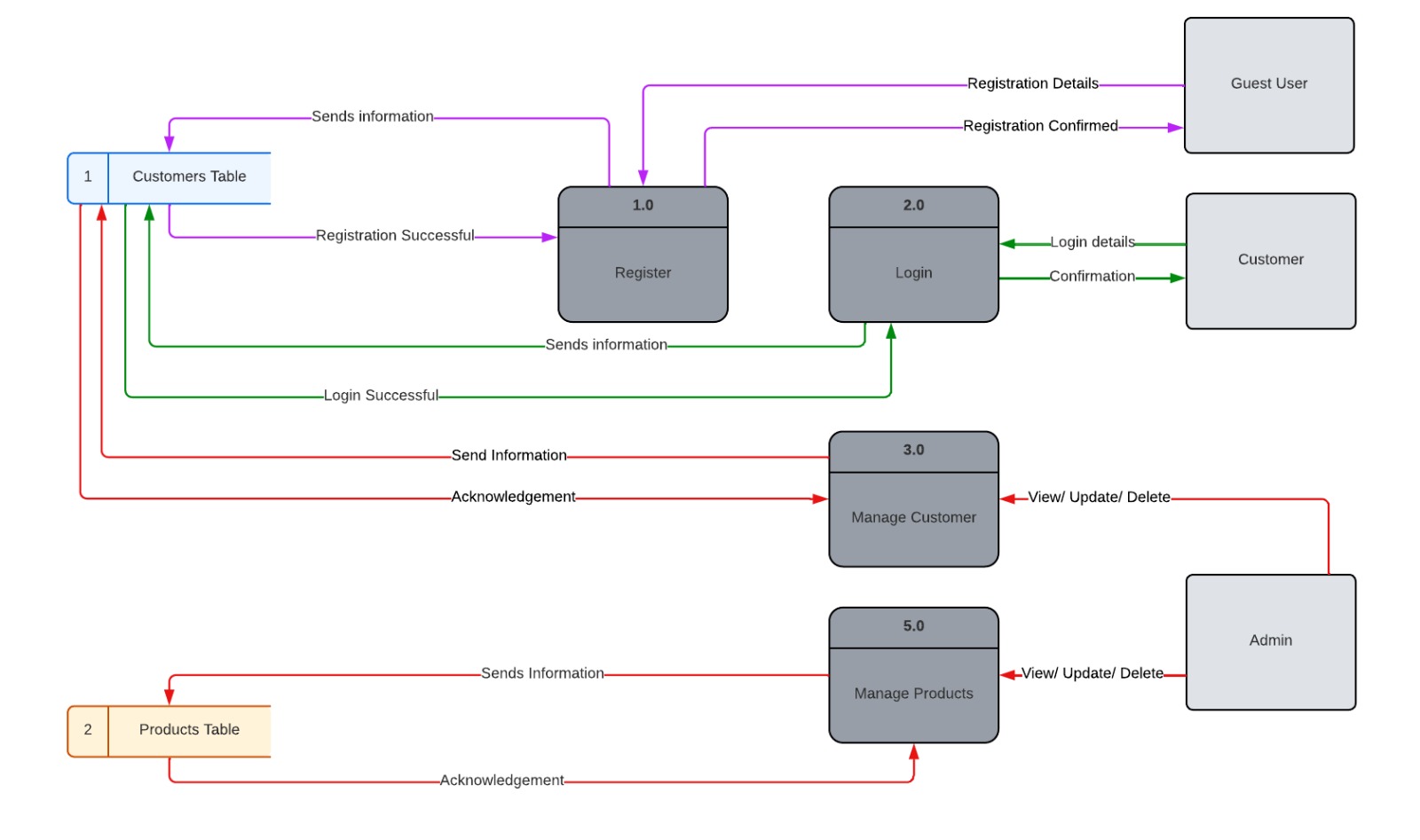


Figure 3‑78: Data Flow Diagram - Iteration 1 for The Sustainable Spoonful System (Lucidchart, 2023).

**Data Flow Diagram – Iteration 2:**

A picture containing text, screenshot, diagram, parallel

Description automatically generated

Figure 3‑79: Data Flow Diagram - Iteration 2 for The Sustainable Spoonful System (Lucidchart, 2023).

The Data Flow Diagram is used to map out the flow of information within a system and can be used to design a new system or analyse an existing one (Lucidchart, 2023).

While the Customer\_Retailer table is not mentioned here, it will simply serve as a junction table between the Customer Table and the Retailer table, storing the customerID and retailerID to prevent duplications.

### Iterative and Incremental Design

While we have multiple iterations of how our mobile application will function, this is an ongoing process that will constantly be improved upon during the development process and with more user feedback and testing.

Our approach was to map out the basic functionality for the first iteration and build on it, as seen in the second iteration.

We have made the following improvements:

* For the Context diagram, we expanded upon the existing processes to give a clearer scope of the data flow within the system.
* For the Entity Relationship Diagram, we determined the nature of the relationships between the tables and resolved any many-to-many relationships.
* For the Data Flow Diagram, we have added an external entity (an Admin) to help better manage the system.

## Conclusion

In summary, we have approached the architectural design phase using multiple methods.

These methods include prototyping our design with wireframes, an interactive prototype, drawing up use-case diagrams to map out which users will interact with the system and what actions they should be able to perform, sequence diagrams to show what messages will be exchanged within the system between all the different objects and processes, context diagrams to observe how data will move within the system, entity relationship diagrams to assist in modelling the database and determining how entities within the system will interact with each other and data flow diagrams to aid in understanding the flow of information within the system.

By basing our design on such a solid foundation, we should be able to move from the design phase into the development phase efficiently.

# System Prototype Development and Testing

## Introduction

The main goal of The Sustainable Spoonful mobile application is to assist in mitigating food wastage amongst retailers and consumers.

During the system prototype and testing phase, the following factors had to be considered; these factors included creating a mobile application that was visually appealing, user-friendly, and met the functional requirements that were initially stated at the start of the development process. After considering these factors, the prototype for how the system will function, as well as the design diagrams for the system, was developed.

The layout development, business logic development, data access development and testing stages will be covered in this section:

The layouts development stage involves recreating the prototype from the previous iteration and building upon it. The prototype from the previous iteration includes the registration, login, store, discounted products, and account screens. During this stage, the users’ experience must be taken into consideration (Soegaard, 2019). The mobile application should be easy to navigate and responsive on mobile devices (Soegaard, 2019). The design of the screens should remain consistent throughout the mobile application (Soegaard, 2019). Usability testing should be conducted to ensure that users can navigate and interact with the mobile application easily (Soegaard, 2019). This can ensure that users will have a positive user experience when using this mobile application (Soegaard, 2019).

The business logic development stage involves translating the business requirements into functional code (Fowler, 2017). During this stage, the business requirements must be clearly defined. Having clearly defined business requirements help ensure that they can be easily translated into functional code that follows the industry’s best practices. In addition, it can ensure that the code is maintainable, readable, and scalable (Fowler, 2017). This can be achieved by adopting an agile approach and continuously improving the code throughout the development process (Fowler, 2017).

The data access development stage involves designing the data access layer that will interact with the mobile applications’ backend system and the SQLite database. Afterwards, testing will be conducted to ensure that the mobile application is reliable and of high quality (Ng, 2021). During this stage, the integrity of the user and the mobile applications’ data must be maintained and secured (Ng, 2021). It is also imperative that an efficient and scalable database is developed; this can assist in minimizing response times by optimizing data retrieval (Ng, 2021).

Lastly, during the testing process, it is important to establish a testing strategy, create test cases, ensure that all features and functionalities of the mobile application are tested and documented and report any issues that have been identified (Hamilton, 2019). Taking these steps can ensure that a stable and reliable mobile application will be developed (Hamilton, 2019).

During these various stages, different roles have been allocated to the development team; this ensures that tasks are divided and distributed evenly throughout the planning and development process.

The roles are shown in Table 1 below:

|  |  |  |
| --- | --- | --- |
| **Roles and Responsibilities** | | |
| **Name** | **Role** | **Responsibilities** |
| Melany | * Frontend Developer * Proofreader | * Designed and developed the various screens of the mobile application. * Proofread all deliverable documents. |
| Lea | * Backend Developer * Proofreader * Documentation | * Designed and developed the functional aspects of the mobile application. * Proofread all deliverable documents. * Compiled the final documentation. |
| Lucinda | * Assistant Backend Developer * Documentation * Testing | * Designed and developed the functions to retrieve and display the discounted product information. * Compiled project documentation. * Testing the mobile application for errors and unexpected behaviour. |
| Santana | * Proofreader * Researcher | * Proofread all deliverable documents. * Researched for all deliverables. |

Table 4.1: Roles and responsibilities of the development team.

This chapter aims to apply our research towards building a mobile application that can make a difference in communities.

## Testing Plan

During this stage, various tests will be conducted to determine whether an average user can navigate through the mobile application with ease. The type of testing that will be conducted is usability testing. Various test cases have been developed for users to work through; once they have completed these test cases, they can complete a survey. The survey will provide the development team with insight into how users view the mobile application and how they wish that it can be improved.

**Objective:**

* Understanding how users will interact with the mobile application and making changes based on those results (Optimizely, 2023).
* Ensuring that the mobile application is effective, efficient, engaging, error tolerant and can be easy to learn how to use. This can include (Quesenbery, 2023):
  + **Effective**: How accurately were tasks completed, and how often did they produce errors?
  + **Efficient**: Evaluate the time taken to complete tasks.
  + **Engaging**: Gauge the users’ feelings towards the mobile application with the use of surveys or interviews.
  + **Error tolerant**: Create test case scenarios that may have the potential to cause errors.
  + **Easy to learn**: Attempt to gather test users from different technical knowledge backgrounds to determine how user-friendly the mobile application is.

**Scope:**

* This test will include all the screens in the mobile application, as well as the content of each screen and navigating between the different screens (Usability.gov, 2019).
* The screens in this mobile application are (Usability.gov, 2019):
  + Landing screen
  + Login screen
  + Registration screen
  + Home screen
  + Store screen
  + Discounted Products screen
  + Account screen

**Components:**

1. Set up a usability test consisting of various tasks that the user would need to complete. These tasks will include:
   * Creating an account.
   * Log in to their account.
   * Browsing the available discounted products.
   * Viewing their account.
   * Logging out of the mobile application.
2. Set up a survey for users to complete after they have tested the mobile application (Optimizely, 2023).

**Equipment:**

* Testing will be conducted through the users’ mobile devices.

**Test Cases** (Zinchenko, 2019):

**Test Case 1**: The user installs the mobile application and opens it for the first time. Upon opening it, the user would need to register for an account; after registering, the user would use their credentials to log in to the mobile application.

* In this scenario, the user would need to open the mobile application, register an account, log in to the mobile application and then locate the discounted products screen.
* For test case 1 to be considered successful, the user needs to register an account and then log in to the mobile application.

**Test Case 2**: The user closes the mobile application after viewing the discounted products screen. If they re-open the mobile application, would the user remain signed in, or would they need to re-enter their login details?

* In this scenario, the user would need to close the mobile application from the recently opened applications list on the device and then re-open The Sustainable Spoonful mobile application.
* For test case 2 to be considered successful, the user needs to open the mobile application without needing to log into the mobile application again.

**Test Case 3**: The user needs to navigate to the accounts screen using the menu. If they log out, will they be redirected to the landing screen?

* In this scenario, the user would need to navigate to the account screen, view their account information and then press the logout button.
* For test case 3 to be considered successful, the user would need to navigate to the account screen, log out and then be redirected to the landing screen of the mobile application.

**Test Case 4**: The user wishes to view a discounted item from Woolworths. Could they easily navigate to it?

* In this scenario, the user would need to navigate to the stores’ page, click on the Woolworths card and then select one of their discounted products.
* For test case 4 to be considered successful, the user would need to navigate to the discounted products section without requesting assistance and complete the action in a reasonable amount of time.

**Test Case 5**: The user wishes to navigate to the account screen and then navigate back to the discounted products page using the navigation menu. Would they be able to easily navigate through it?

* In this scenario, the user would need to navigate to the account screen and then navigate back to the discounted products screen.
* For test case 5 to be considered successful, the user would need to navigate to the account screen and then to the discounted products screen without requesting assistance and completing this action in a reasonable amount of time.

**User Survey Questions:**

A Google form has been created for users to complete once they have worked through the five test cases.

The link to the form is attached below:

<https://docs.google.com/forms/d/e/1FAIpQLSc3EnTE__Z76rWmMWVvez45fT4MDG6OvFQgRelu5jnNebCrZg/viewform>

**Test Report (Zinchenko, 2019):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Usability Testing: Testing Schedule and Report – 18 June 2023** | | | | | | | |
| **User** | **User Description** | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| 1 | Female, age 56, limited technical knowledge. | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |
| 2 | Female, age 32, firm grasp of technology. | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |
| 3 | Male, age 33, firm grasp of technology. | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |
| 4 | Female, age 20, firm grasp of technology. | Close with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |
| 5 | Female, age 20, firm grasp of technology. | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill | Checkmark with solid fill |

Table 4.2: Testing Schedule and Report.

**Analysis and recommendations:**

**Test Case 1:**

Analysis:

* 80% of the users were able to complete test case 1 without any issues.
* 20% of users were not able to complete test case 1 without any issues due to the mobile application lagging after registration.

Recommendations:

* Review the code related to the registration process and make changes accordingly.

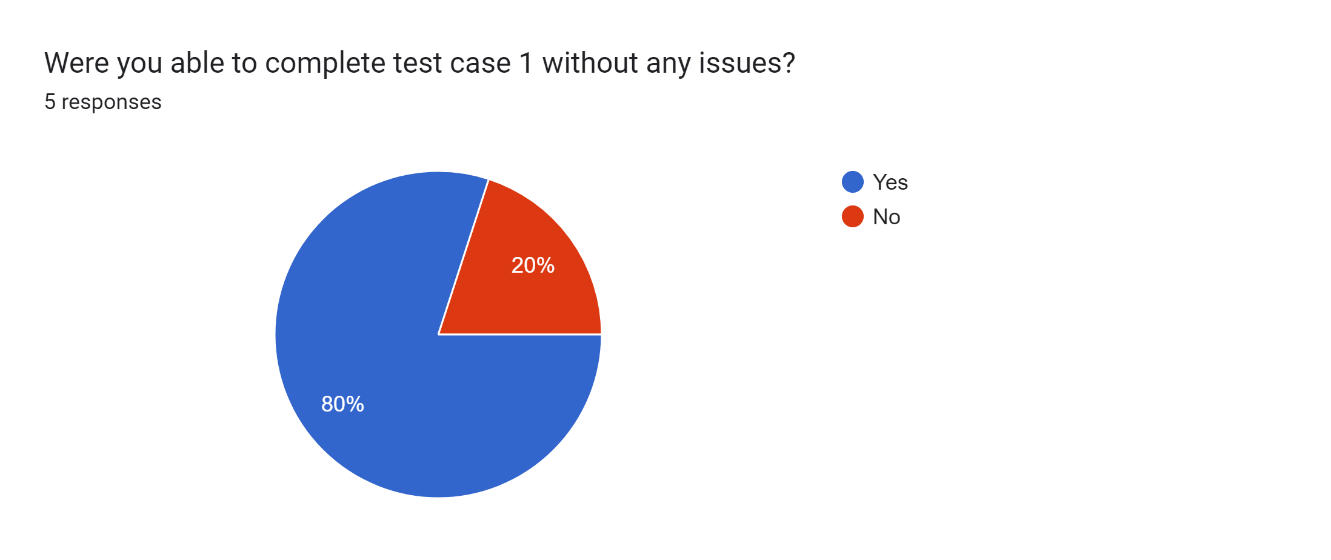


Figure 4‑1: Test Case 1 - Results Graph.

**Test Case 2:**

Analysis:

* 100% of the users were able to complete test case 2 without any issues.

Recommendations:

* Keep building on the current iteration.

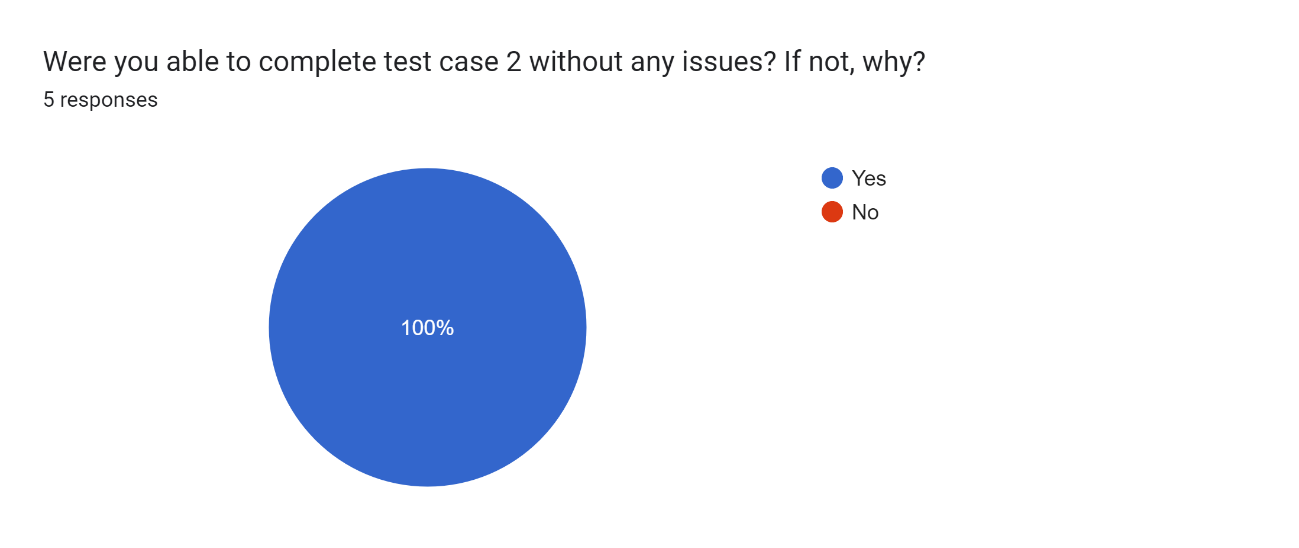
****

Figure 4‑2: Test Case 2 – Results Graph.

**Test Case 3:**

Analysis:

* 100% of the users were able to complete test case 3 without any issues.

Recommendations:

* Keep building on the current iteration.

Forms response chart. Question title: Were you able to complete test case 3 without any issues? If not, why?
. Number of responses: 5 responses.

Figure 4‑3: Test Case 3 – Result Graph.

**Test Case 4:**

Analysis:

* 100% of the users were able to complete test case 4 without any issues.

Recommendations:

* Keep building on the current iteration.

Forms response chart. Question title: Were you able to complete test case 4 without any issues? If not, why?
. Number of responses: 5 responses.

Figure 4‑4: Test Case 4 – Results Graph.

**Test Case 5:**

Analysis:

* 100% of the users were able to complete test case 5 without any issues.

Recommendations:

* Keep building on the current iteration.

Forms response chart. Question title: Were you able to complete test case 5 without any issues? If not, why?
. Number of responses: 5 responses.

Figure 4‑5: Test Case 5 – Results Graph.

**Additional Responses:**

Analysis:

* 80% of the users stated that everything on the mobile application was readable.
* 20% of the users stated that the colours used made it difficult to read the text on the screen.

Recommendations:

* Work on adjusting the colours to make the mobile application more readable.
* Consider implementing a dark mode for users who find the current theme too bright or difficult to read.

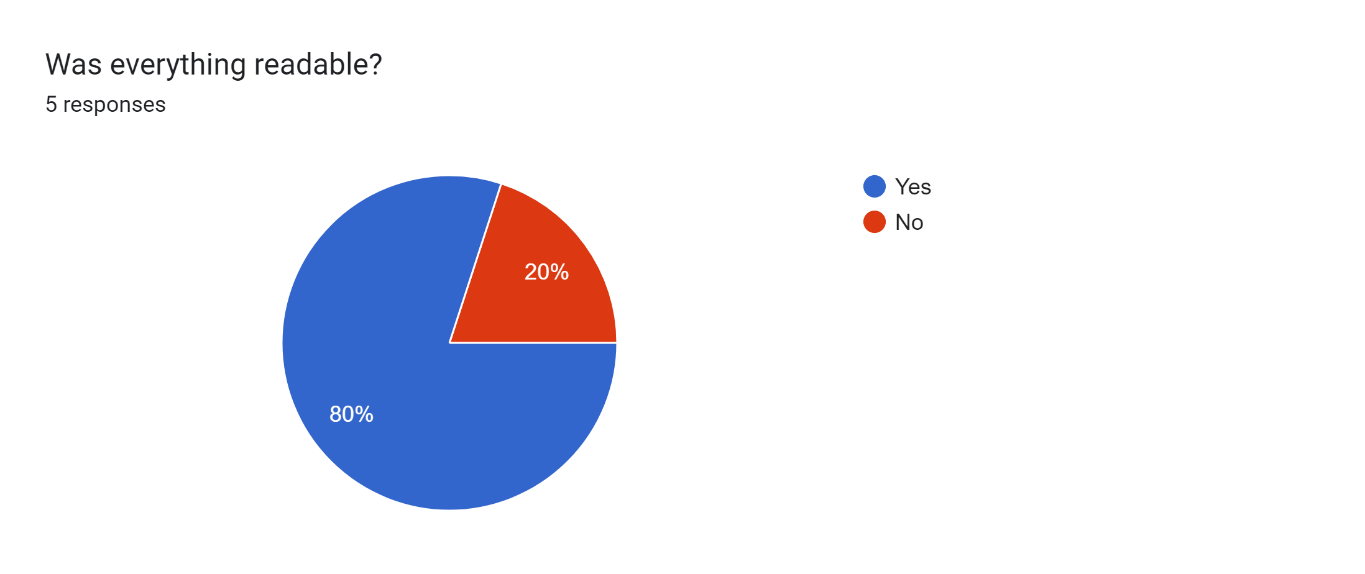


Figure 4‑6: Was everything readable? - Result Graph.

Analysis:

* 100% of the users stated the mobile application was responsive.

Recommendations:

* Keep building on the current iteration.

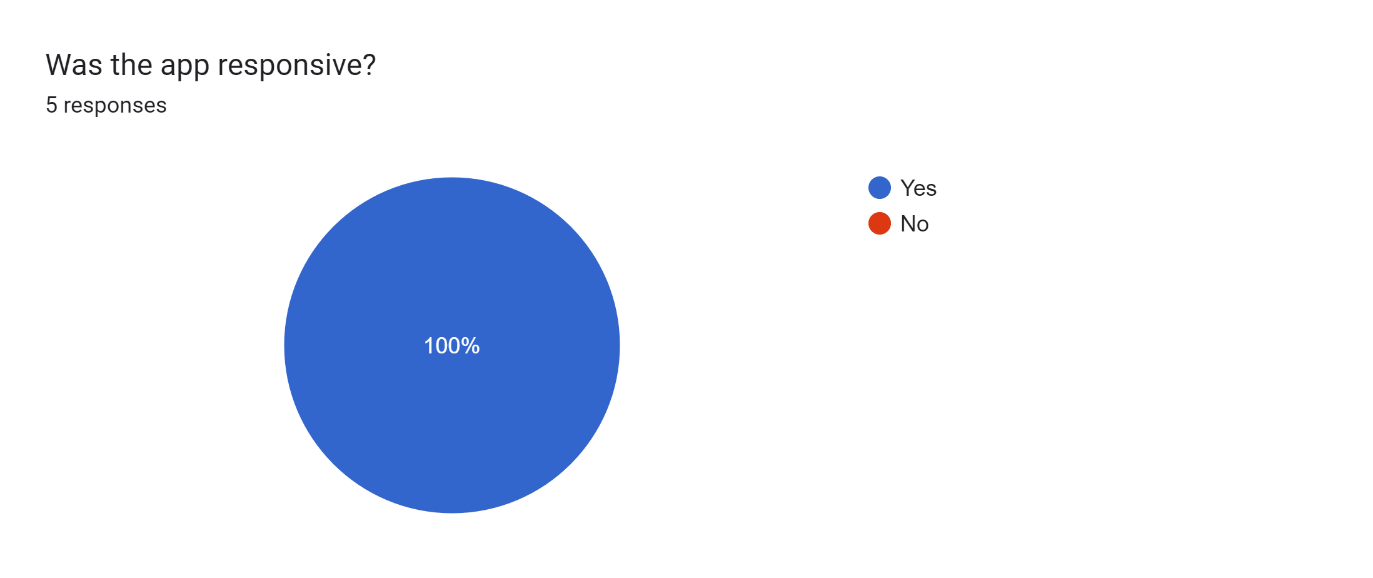


Figure 4‑7: Was the application responsive? - Result Graph.

Analysis:

* 100% of the users found the mobile application easy to navigate.

Recommendations:

* Keep building on the current iteration.

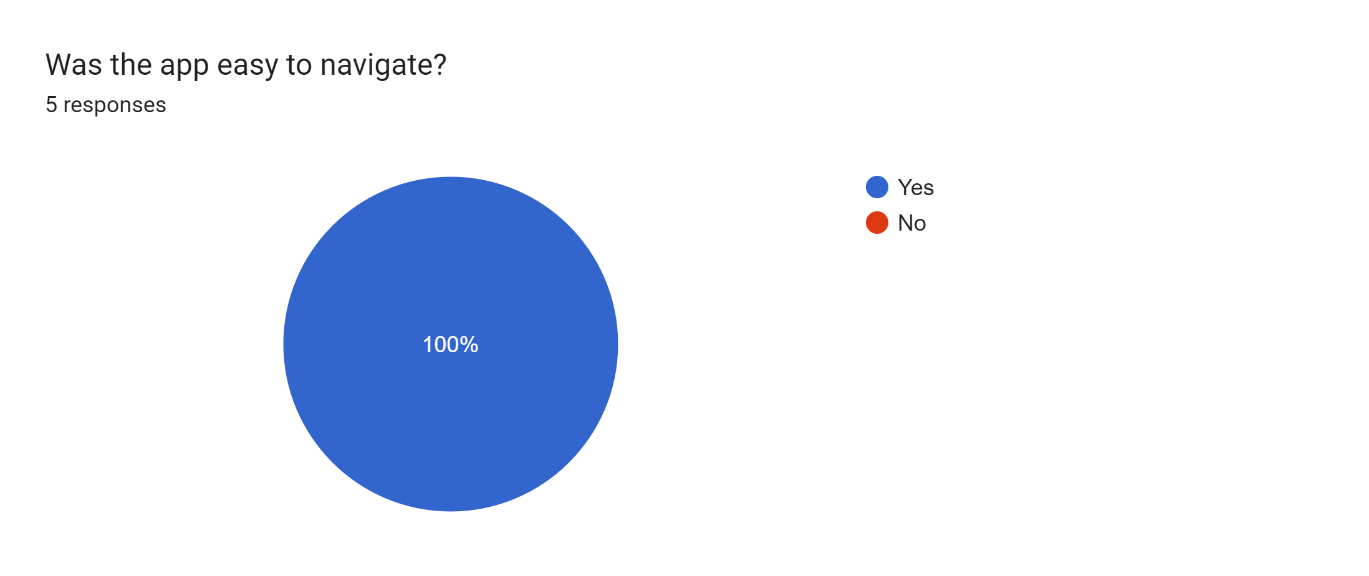


Figure 4‑8: Was the application easy to navigate? – Result Graph.

Overall, most users responded positively to using the mobile application and filling out the survey form. However, some users requested that they wish the mobile application could change themes, and colours and enable them to edit their account details.

## Layouts Development

### User Interface – Main Screen

This is the first screen that users will view when opening the mobile application for the first time. Users are presented with a simple, user-friendly interface that has a logo and two buttons where they can either register for an account or log in with an existing account.

If the user has already registered an account and logged in, upon opening the mobile application again, the user will automatically be redirected to the home page. This removes the need for repetitive login procedures.

By offering both the login and registration options on this initial screen, both types of users (such as new users and existing users) will be catered to. This will ensure that there is a personalized experience for all.

The first iteration of the initial screen as well as the current iteration is shown below:

##### Iteration 1

A screenshot of a blue screen

Description automatically generated with low confidence

Figure 4‑9: Main screen when the user opens the mobile application for the first time.

##### Iteration 2

A screen shot of a cell phone

Description automatically generated with medium confidence

Figure 4‑10: Main screen when the user opens the mobile application for the first time. The mobile application is running on the Emulator Pixel 6 Pro API 30.

**Code Snippet:**

package com.example.sustainablespoonfulapp;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.database.CursorWindow;  
import android.os.Bundle;  
import android.preference.PreferenceManager;  
import android.view.View;  
import android.widget.Button;  
  
import java.lang.reflect.Field;  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*); //Create the main page when opening the application:  
  
 try {  
 Field field = CursorWindow.class.getDeclaredField("sCursorWindowSize");  
 field.setAccessible(true);  
 field.set(null, 100 \* 1024 \* 1024); //increases size to 100mb  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 //Checking if the customer is logged in:  
 SharedPreferences sharedPreferences = PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext());  
 boolean isLoggedIn = sharedPreferences.getBoolean("isLoggedIn",false); //Set is logged in to false:  
  
 //If the customer is logged in, redirect to the home page:  
 if(isLoggedIn) {  
 Intent intent = new Intent(MainActivity.this, LandingActivity.class);  
 startActivity(intent);  
 finish();  
 }  
  
 /\*Navigating to the Registration Page After Clicking the Register Button\*/  
 Button registerButton=findViewById(R.id.*home\_register\_button*);  
 registerButton.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Intent intent = new Intent(MainActivity.this,RegisterActivity.class); //Redirect the customer to the registration page:  
 startActivity(intent);  
 }  
 });  
  
 /\*Navigating to the Login Page After Clicking the Login Button\*/  
 Button loginButton=findViewById(R.id.*home\_login\_button*);  
 loginButton.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 Intent intent = new Intent(MainActivity.this,LoginActivity.class); //Redirect the customer to the login page:  
 startActivity(intent);  
 }  
 });  
 }  
}

The **MainActivity.java** code does the following:

* In the **onCreate()** function, the following code is present:
  + Creates the main screen (this is **activity\_main.xml**) when opening the mobile application. The main screen contains the logo of the Sustainable Spoonful mobile application as well as two buttons; the first button is for registering for a new account, and the second button is for logging into the mobile application using existing credentials.
  + A **try-catch** is used to increase the size of the cursor window to 100mb when the mobile application is opened.
  + If the user is already logged in, it will automatically redirect them to the home screen (which is the **LandingActivity.java**) of the mobile application.
  + There are two **onClick()** functions, one for navigating to the RegisterActivity.java file and the other for the LoginActivity.java file.

### User Interface - Registration

The registration screen (which is the **RegisterActivity.java** file) enables the user to register an account.

The user will need to fill in all fields of the registration form; the form consists of the following:

* First name input.
* Last name input.
* Email address input.
* Password input.
* Confirm password input.
* Confirm button.

If the user is successfully registered, their registration details will be stored in the local database. Afterwards, an alert message will be displayed notifying the user that their registration was successful, and they will then be redirected to the login screen (which is **LoginActivity.java file**).

The first iteration of the registration screen and the current iteration is shown below:

##### Iteration 1

A screen shot of a phone

Description automatically generated with low confidence

Figure 4‑11: Registration screen for users to create an account.

##### Iteration 2

A screen shot of a cell phone

Description automatically generated with low confidence

Figure 4‑12: Registration screen for users to create an account. The mobile application is running on the Emulator Pixel 6 Pro API 30.

**Code Snippet:**

package com.example.sustainablespoonfulapp;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.content.ContentValues;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.database.sqlite.SQLiteDatabase;  
import android.widget.EditText;  
import android.widget.Toast;  
import android.content.Intent;  
import android.database.Cursor;  
  
public class RegisterActivity extends AppCompatActivity {  
  
 //Declaring the database helper variable:  
 private DatabaseHelper databaseHelper;  
 //Declaring variables for all the inputs in the registration form:  
 private EditText nameEditText, surnameEditText, emailEditText, passwordEditText, confirmPasswordEditText;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_register*); //Create the register page:  
  
 //Creating an instance of the DatabaseHelper class:  
 databaseHelper = new DatabaseHelper(this);  
 nameEditText = findViewById(R.id.*register\_name\_text*);  
 surnameEditText = findViewById(R.id.*register\_surname\_text*);  
 emailEditText = findViewById(R.id.*register\_email\_address\_text*);  
 passwordEditText = findViewById(R.id.*register\_password\_text*);  
 confirmPasswordEditText = findViewById(R.id.*register\_confirm\_password\_text*);  
  
 //Inserting the customers' details when pressing the confirm button in the registration form:  
 Button confirmButton = findViewById(R.id.*register\_confirm\_button*);  
 confirmButton.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 insertCustomer();  
 }  
 });  
  
 }  
  
 //Function to insert customer details into the customer table:  
 private void insertCustomer(){  
 //Getting all of the data that was input in the form:  
 String name = nameEditText.getText().toString().trim();  
 String surname = surnameEditText.getText().toString().trim();  
 String email = emailEditText.getText().toString().trim();  
 String password = passwordEditText.getText().toString().trim();  
 String confirmPassword = confirmPasswordEditText.getText().toString().trim();  
  
 //Check if any of the input fields are empty before inserting the customer details:  
 if(name.isEmpty()||surname.isEmpty()||email.isEmpty()||password.isEmpty()||confirmPassword.isEmpty()){  
 //Display a message prompting customers to enter their details on the page:  
 Toast.*makeText*(RegisterActivity.this, "Please fill out all fields in this form.", Toast.*LENGTH\_SHORT*).show();  
 return; //Exiting the method early:  
 }  
  
 //If the password and confirm password match insert the details into the customer table:  
 if(password.equals(confirmPassword)){  
 //Getting a writable database:  
 SQLiteDatabase db = databaseHelper.getWritableDatabase();  
  
 //Checking if the email address already exists in the customer table:  
 boolean emailExists = checkEmailExists(email,db);  
  
 if(emailExists){  
 Toast.*makeText*(RegisterActivity.this, "This email address already exists! Please try again.", Toast.*LENGTH\_SHORT*).show();  
 }else{ //Does not exist so insert details into the customer table:  
 ContentValues values = new ContentValues();  
 values.put(DatabaseHelper.*COLUMN\_CUSTOMER\_NAME*, name);  
 values.put(DatabaseHelper.*COLUMN\_CUSTOMER\_SURNAME*, surname);  
 values.put(DatabaseHelper.*COLUMN\_CUSTOMER\_EMAIL*,email);  
 values.put(DatabaseHelper.*COLUMN\_CUSTOMER\_PASSWORD*, password);  
  
 long rowID = db.insert(DatabaseHelper.*TABLE\_NAME\_CUSTOMER*,null,values);  
  
 //Closing the database after inserting the customer's details:  
 db.close();  
  
 //If the row ID is not equal to minus one, display a success message:  
 if(rowID != -1){  
 //Display a message to the customer notifying them that their registration was successful:  
 Toast.*makeText*(RegisterActivity.this, "Registration was successful!", Toast.*LENGTH\_SHORT*).show();  
 startActivity(new Intent(RegisterActivity.this, LoginActivity.class)); //Redirect the customer to the login page:  
 finish(); //Finishing the current activity so that customers' cannot go back to it when pressing the back button:  
  
 }else{  
 //Displaying a message at the bottom notifying the customer that their registration has failed:  
 Toast.*makeText*(RegisterActivity.this, "Registration failed! Please try again.", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 }else{ //Passwords do not match, display a message at the bottom notifying the customer that their passwords do not match:  
 Toast.*makeText*(RegisterActivity.this, "The passwords entered do not match! Please try again.", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 private boolean checkEmailExists(String email, SQLiteDatabase db){  
 //Define which column to retrieve from the database:  
 String[] projection = {databaseHelper.getColumnEmail()};  
 //Select any email addresses that are in the customer table that match the email address entered:  
 String selection = databaseHelper.getColumnEmail() + " = ?";  
 //Specify the argument for the query, this will be email:  
 String[] selectionArgs = {email};  
 //Query the customer table for any matching records:  
 Cursor cursor = db.query(DatabaseHelper.*TABLE\_NAME\_CUSTOMER*, projection, selection, selectionArgs, null, null, null);  
 //Checking if any records were found/records greater than 0:  
 boolean exists = cursor.getCount()>0;  
 //Close the cursor so that associated resources can be released:  
 cursor.close();  
 //Return the result (if the email exists or not):  
 return exists;  
 }  
}

The **RegisterActivity.java** code does the following:

* In the **onCreate()** function, the following code is present:
  + Creates an instance of the **DatabaseHelper** class.
  + Creates the registration screen (which is **activity\_register.xml**). The register screen contains the title which says ‘Register’ with the following fields: ‘Name’, ‘Surname’, ‘Email Address’, ‘Password’ and ‘Confirm Password.’ These five fields are displayed in input boxes on the screen.
  + **findViewById()** is used to match the variables defined in the **RegisterActivity.java** file with the IDs in the **activity\_register.xml** file. (These are referred to as widgets.)
  + There is one **onClick()** function; this is used to call a function called **insertCustomer().**
  + The **insertCustomer()** function does the following:
    - Checks if the name, surname, email, password, and confirm password input boxes have been filled. If it is left empty, an alert is displayed to the user prompting them to enter their details.
    - Checks if the password and confirm password textboxes match; if they do not match, it displays an alert message warning the user to check the passwords that they have entered.
    - While the code checks if the password and confirm password match, it calls the **checkEmailExists()** function by passing in the email address (**email**) and the writable database (**db**) as parameters.
    - The **checkEmailsExists()** function does the following:
      * Checks if the email address already exists in the database and displays an alert message warning the user to try entering another email address.
      * After all the checks have been done, the users’ details are inserted into the local database.

### User Interface - Login

The login screen (which is the **LoginActivity.java** file) allows the user to enter their login credentials.

The login form consists of the following:

* Email address input.
* Password input.
* Confirm button.

The email address and password need to match the email address and password stored in the local database for the user to gain access to the mobile applications’ home page. Once the user has been verified, they will be redirected to the landing page (which is the **LandingActivity.java** file). An alert message will be displayed on the screen notifying the user that they have been successfully logged in.

The first iteration of the login screen and the current iteration is shown below:

##### Iteration 1

A screen shot of a phone

Description automatically generated with medium confidence

Figure 4‑13: Login Screen so that users can login using their existing credentials.

##### Iteration 2

A screen shot of a cell phone

Description automatically generated with low confidence

Figure 4‑14: Login Screen so that users can login using their existing credentials. The mobile application is running on the Emulator Pixel 6 Pro API 30.

**Code Snippet:**

package com.example.sustainablespoonfulapp;  
  
import android.os.Bundle;  
import android.content.Intent;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import android.content.SharedPreferences;  
import android.preference.PreferenceManager;  
  
public class LoginActivity extends AppCompatActivity {  
  
 //Declaring the database helper variable:  
 private DatabaseHelper databaseHelper;  
  
 //Declaring variables for the email and password inputs:  
 private EditText emailEditText, passwordEditText;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*); //Create the login page:  
  
 //Creating an instance of the DatabaseHelper class:  
 databaseHelper = new DatabaseHelper(this);  
 emailEditText = findViewById(R.id.*login\_email\_address\_text*);  
 passwordEditText = findViewById(R.id.*login\_password\_text*);  
  
 //Setting up the login button click listener:  
 Button loginButton = findViewById(R.id.*login\_confirm\_button*);  
  
 loginButton.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 login();  
 }  
 });  
 }  
  
 private void login(){  
 //Getting the email and password that the customer has entered and trimming it:  
 String email = emailEditText.getText().toString().trim();  
 String password = passwordEditText.getText().toString().trim();  
  
 //Checking if any of the input fields are empty before logging in the customer:  
 if(email.isEmpty()||password.isEmpty()){  
 //Display a message to the customer asking them to fill their details in on the form:  
 Toast.*makeText*(LoginActivity.this, "Please fill out all fields in this form.", Toast.*LENGTH\_SHORT*).show();  
 return; //Exit the method early:  
 }  
  
 //Getting a readable database:  
 SQLiteDatabase db = databaseHelper.getReadableDatabase();  
  
 //Checking if the email and password match a record in the customer table:  
 boolean loginSuccess = checkEmailAndPassword(email, password, db);  
  
 //If a record was found and it matches/login was successful display a message and redirect to the landing page:  
 if(loginSuccess){  
 //Store the customer details in Shared Preferences:  
 SharedPreferences sharedPreferences = PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext());  
 SharedPreferences.Editor editor = sharedPreferences.edit();  
 editor.putString("email",email); //Store the customers email:  
 editor.putBoolean("isLoggedIn",true); //Store the customers' login status:  
 editor.apply(); //Save the changes to Shared Preferences:  
  
 //Displaying a success message to the customer notifying them that they have been logged in:  
 Toast.*makeText*(LoginActivity.this, "Logged in successfully!", Toast.*LENGTH\_SHORT*).show();  
 //Redirecting to the home Page:  
 startActivity(new Intent(LoginActivity.this, LandingActivity.class));  
 finish(); //Finishing the current activity so that users cannot go back to it when pressing the back button:  
 }else{ //Password or email was incorrect:  
 //Display an error message to the customer:  
 Toast.*makeText*(LoginActivity.this, "Invalid email or password! Please try again.", Toast.*LENGTH\_SHORT*).show();  
 }  
 //Closing the database:  
 db.close();  
 }  
  
 private boolean checkEmailAndPassword(String email, String password, SQLiteDatabase db){  
 //Define which columns to retrieve from the customer table in the database:  
 String[] projection = {databaseHelper.getColumnEmail()};  
 //Select any email addresses and passwords that are in the customer table that match the email address and password entered:  
 String selection = databaseHelper.getColumnEmail() + " = ? AND " + databaseHelper.getColumnPassword() + " = ?";  
 //Specify the arguments for the query, this will be email and password:  
 String[] selectionArgs = {email, password};  
 //Query the customer table for any matching records:  
 Cursor cursor = db.query(DatabaseHelper.*TABLE\_NAME\_CUSTOMER*, projection, selection, selectionArgs, null, null, null);  
 //Checking if any records were found/records greater than 0:  
 boolean exists = cursor.getCount()>0;  
 //Close the cursor so that associated resources can be released:  
 cursor.close();  
 //Return the result (if the email exists or not):  
 return exists;  
 }  
}

The **LoginActivity.java** code does the following:

* In the **onCreate()** function, the following code is present:
  + Created an instance of the **DatabaseHelper** class.
  + Creates the login screen (which is the **activity\_login.xml**). The login screen contains the title ‘Login’ with the following fields that consist of ‘Email Address’ and ‘Password’. These two fields are displayed in input boxes on the screen.
  + There is one **onClick()** function, this will call the **login()** function.
  + The **login()** function does the following:
    - If the email and password textboxes are left empty, it displays an alert message for the user to fill in their details.
    - Defines a variable called **loginSuccess** which calls the **checkEmailAndPassword()** function and passes in **email**, **password** and **db** as parameters.
    - The **checkEmailAndPassword()** function does the following:
      * It checks if the email address and password entered match the details in the customer table.
      * It returns true if it matches and false if it does not match.
      * If the result is true, the code does the following:
        + Stores the users’ **email** in the session.
        + Defines a variable called **isLoggedIn** and sets it to true. (This variable will be used on other pages to check if the user is logged in.)
        + Displays an alert message on the screen notifying the user that they have been logged in. Afterwards, the user is redirected to the landing screen (which is the **LandingActivity.java file**).
        + If the result returned was false, display an alert message notifying the user that they have an invalid email or password and that they should try to enter their details again.

### User Interface – Home

This is the home/landing screen for the mobile application. This screen is only displayed once the user has successfully logged in. On this screen, images of the different partnered stores are displayed. Users can click on each store to view a description of the store as well as where they are located. Afterwards, users can navigate to other pages (such as the Discounted Products and Account page) using the bottom navigation menu.

The first iteration of the landing screen and the current iteration is shown below:

##### Iteration 1

A screen shot of a phone

Description automatically generated with medium confidence

Figure 4‑15: Landing/home screen for the Sustainable Spoonful Mobile Application.

##### Iteration 2

A screen shot of a cell phone

Description automatically generated with medium confidence

Figure 4‑16: Landing/home screen for the Sustainable Spoonful Mobile Application. The mobile application is running on the Emulator Pixel 6 Pro API 30.

**Code Snippet:**

package com.example.sustainablespoonfulapp;  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.preference.PreferenceManager;  
import android.view.MenuItem;  
import android.widget.ImageButton;  
import android.widget.Toast;  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
import com.google.android.material.navigation.NavigationBarView;  
import android.view.View;  
  
public class LandingActivity extends AppCompatActivity {  
  
 BottomNavigationView bottom\_nav\_bar;  
  
 ImageButton landing\_picknpay\_button, landing\_foodloversmarket\_button, landing\_checkers\_button, landing\_woolworths\_button;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_landing*); //Create the home page:  
  
 bottom\_nav\_bar = findViewById(R.id.*bottom\_nav\_bar*);  
 bottom\_nav\_bar.setSelectedItemId(R.id.*home\_bottom\_navigation*); //Set the home icon to selected when on this page:  
  
 bottom\_nav\_bar.setOnItemSelectedListener(new NavigationBarView.OnItemSelectedListener() {  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem item) {  
 //Create a switch statement:  
 switch (item.getItemId()){  
 //If the home icon is clicked, stay on the home page:  
 case R.id.*home\_bottom\_navigation*:  
 return true;  
 //If the search icon is clicked, go to the search discounts page:  
 case R.id.*search\_bottom\_navigation*:  
 startActivity(new Intent(LandingActivity.this,DiscountActivity.class)); //Redirect the customer to the search discount page:  
 return true;  
 //If the account icon is clicked, go to the account page:  
 case R.id.*account\_bottom\_navigation*:  
 startActivity(new Intent(LandingActivity.this,AccountActivity.class)); //Redirect the customer to the account page:  
 return true;  
 default:  
 return false;  
 }  
 }  
 });  
  
 //For the image buttons on the landing screen:  
 landing\_picknpay\_button = findViewById(R.id.*landing\_picknpay\_button*);  
 landing\_foodloversmarket\_button = findViewById(R.id.*landing\_foodloversmarket\_button*);  
 landing\_checkers\_button = findViewById(R.id.*landing\_checkers\_button*);  
 landing\_woolworths\_button = findViewById(R.id.*landing\_woolworths\_button*);  
  
 landing\_picknpay\_button.setOnClickListener(new View.OnClickListener(){  
 public void onClick(View v){  
 Intent intent = new Intent(LandingActivity.this,PicknPayStoreActivity.class);  
 startActivity(intent);  
 }  
 });  
  
 landing\_foodloversmarket\_button.setOnClickListener(new View.OnClickListener(){  
 public void onClick(View v){  
 Intent intent = new Intent(LandingActivity.this,FoodLoversStoreActivity.class);  
 startActivity(intent);  
 }  
 });  
  
 landing\_checkers\_button.setOnClickListener(new View.OnClickListener(){  
 public void onClick(View v){  
 Intent intent = new Intent(LandingActivity.this,CheckersStoreActivity.class);  
 startActivity(intent);  
 }  
 });  
  
 landing\_woolworths\_button.setOnClickListener(new View.OnClickListener(){  
 public void onClick(View v){  
 Intent intent = new Intent(LandingActivity.this,WoolworthsStoreActivity.class);  
 startActivity(intent);  
 }  
 });  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
  
 //Checking if the customer is logged in:  
 SharedPreferences sharedPreferences = PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext());  
 boolean isLoggedIn = sharedPreferences.getBoolean("isLoggedIn",false); //Set is logged in to true:  
  
 //If the customer is not logged in, redirect to the home page:  
 if(!isLoggedIn) {  
 Toast.*makeText*(LandingActivity.this, "Please log in to continue!", Toast.*LENGTH\_SHORT*).show(); //Display a message to the customer asking them to log in:  
 startActivity(new Intent(LandingActivity.this,MainActivity.class)); //Redirect to the main page of the application:  
 finish(); //Finishing the current activity so that customers' cannot go back to it when pressing the back button:  
 }  
 }  
}

The **LandingActivity.java** code does the following:

* In the **onCreate()** function, the following code is present:
* Creates the home screen (which is **activity\_landing.xml**). The home screen contains the title ‘The Sustainable Spoonful’ with a sub-heading ‘Proudly in partnership with:’. Below the heading and sub-heading are six clickable images with the stores that are available through the mobile application. These stores include PicknPay, Food Lovers Market, Shoprite, Woolworths, Checkers, and Spar.
* Setups up the bottom navigation bar; the three icons do the following:
  + The home icon will leave the user on the Home screen.
  + The search icon will redirect the user to the Discounts Screen.
  + The user icon will redirect the user to the Accounts screen.
* Changes the home icon to a different colour to show users that they are on this page.
* Loads the images for the stores from the database.
* The are four **onClick()** functions on the store images; these enable users to navigate to each of the store’s pages which shows an image of the store, its location, and a small description.
* The **onResume()** function has the purpose of keeping the user logged in even after they have closed the mobile application.

### User Interface – Discounts

This is the discount screen (which is the **DiscountActivity.java** file) of the mobile application. On this screen, users can view the different stores as well as how many discounts they have available currently. From here, users can select a store by pressing on the store card. They will then be redirected to the discounted products screen.

The first iteration of the discounts screen and the current iteration is shown below:

##### Iteration 1

A cell phone with text on the screen

Description automatically generated with low confidence

Figure 4‑17: - Discounts Screen for users to search for discounted products based on the store and location.

##### Iteration 2

A picture containing text, screenshot, mobile phone, gadget

Description automatically generated

Figure 4‑18: Discounts Screen for users to search for discounted products based on the store and location. The mobile application is running on the Emulator Pixel 6 Pro API 30.

**Code Snippet:**

package com.example.sustainablespoonfulapp;  
  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.preference.PreferenceManager;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.Toast;  
import android.widget.TextView;  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.cardview.widget.CardView;  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
import com.google.android.material.navigation.NavigationBarView;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.widget.ImageView;  
import java.util.ArrayList;  
import java.util.List;  
  
public class DiscountActivity extends AppCompatActivity {  
  
 // Create a list for the store images:  
 private List<Bitmap> storeImages;  
  
 // TextViews used to display the number of discounts that are available on the page:  
 TextView picknpayTextView;  
 TextView checkersTextView;  
 TextView woolworthsTextView;  
 TextView foodloversmarketTextView;  
 BottomNavigationView bottom\_nav\_bar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_discount*);  
  
 // Initializing the store discount images:  
 storeImages = new ArrayList<>();  
  
 // Fetch the store images from the database:  
 DatabaseHelper databaseHelper = new DatabaseHelper(this);  
 List<byte[]> imageBytes = databaseHelper.getAllStoreImages();  
  
 // Convert the byte arrays to bitmaps:  
 for (byte[] bytes : imageBytes) {  
 Bitmap bitmap = BitmapFactory.*decodeByteArray*(bytes, 0, bytes.length);  
 storeImages.add(bitmap);  
 }  
  
 // Displaying the images in the ImageViews:  
 ImageView store\_image\_view1 = findViewById(R.id.*store\_image\_view1*);  
 ImageView store\_image\_view2 = findViewById(R.id.*store\_image\_view2*);  
 ImageView store\_image\_view3 = findViewById(R.id.*store\_image\_view3*);  
 ImageView store\_image\_view4 = findViewById(R.id.*store\_image\_view4*);  
  
 if (storeImages.size() > 0) { // Display PicknPay logo:  
 store\_image\_view1.setImageBitmap(storeImages.get(0));  
 }  
 if (storeImages.size() > 1) { // Display Woolworths logo:  
 store\_image\_view2.setImageBitmap(storeImages.get(1));  
 }  
 if (storeImages.size() > 2) { // Display Checkers logo:  
 store\_image\_view3.setImageBitmap(storeImages.get(2));  
 }  
 if (storeImages.size() > 3) { // Display Foodloversmarket logo:  
 store\_image\_view4.setImageBitmap(storeImages.get(3));  
 }  
  
 // This will be used to find the location of the TextViews on the discounts page:  
 picknpayTextView = findViewById(R.id.*picknpay\_card\_text*);  
 woolworthsTextView = findViewById(R.id.*woolworths\_card\_text*);  
 checkersTextView = findViewById(R.id.*checkers\_card\_text*);  
 foodloversmarketTextView = findViewById(R.id.*foodloversmarket\_card\_text*);  
  
 // Create a new database helper:  
 DatabaseHelper dbhelper = new DatabaseHelper(this);  
  
 // Set the retailer ids for each store that will be passed into the sql count discounted products query:  
 int picknpayId = 1, woolworthsId = 2, checkersId = 3, foodloversID = 4;  
  
 // Set the text for each TextView by calling the getDiscountProductCount function and passing in the retailer id:  
 picknpayTextView.setText(dbhelper.getDiscountedProductCount(picknpayId));  
 woolworthsTextView.setText(dbhelper.getDiscountedProductCount(woolworthsId));  
 checkersTextView.setText(dbhelper.getDiscountedProductCount(checkersId));  
 foodloversmarketTextView.setText(dbhelper.getDiscountedProductCount(foodloversID));  
  
 // Set click listeners for the card views:  
 CardView cardView1 = findViewById(R.id.*cardPicknpay*);  
 CardView cardView2 = findViewById(R.id.*cardCheckers*);  
 CardView cardView3 = findViewById(R.id.*cardWoolworths*);  
 CardView cardView4 = findViewById(R.id.*cardFoodLoversMarket*);  
  
 cardView1.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startProductViewActivity(1);  
 }  
 });  
  
 cardView2.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startProductViewActivity(2);  
 }  
 });  
  
 cardView3.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startProductViewActivity(3);  
 }  
 });  
  
 cardView4.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startProductViewActivity(4);  
 }  
 });  
  
 bottom\_nav\_bar = findViewById(R.id.*bottom\_nav\_bar*);  
 bottom\_nav\_bar.setSelectedItemId(R.id.*search\_bottom\_navigation*); // Set the search icon to selected when on this page:  
  
 bottom\_nav\_bar.setOnItemSelectedListener(new NavigationBarView.OnItemSelectedListener() {  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem item) {  
 // Create a switch statement:  
 switch (item.getItemId()) {  
 // If the home icon is clicked, go to the home page:  
 case R.id.*home\_bottom\_navigation*:  
 startActivity(new Intent(DiscountActivity.this, LandingActivity.class)); // Redirect the customer to the home page:  
 return true;  
 // If the search icon is clicked, stay on the search discounts page:  
 case R.id.*search\_bottom\_navigation*:  
 startActivity(new Intent(DiscountActivity.this, DiscountActivity.class)); // Stay on the search discount page:  
 return true;  
 // If the account icon is clicked, go to the account page:  
 case R.id.*account\_bottom\_navigation*:  
 startActivity(new Intent(DiscountActivity.this, AccountActivity.class)); // Redirect the customer to the account page:  
 return true;  
 default:  
 return false;  
 }  
 }  
 });  
 }  
  
 private void startProductViewActivity(int retailerId) {  
 // Start the ProductViewActivity passing the retailer ID as an extra:  
 Intent intent = new Intent(DiscountActivity.this, ProductViewActivity.class);  
 intent.putExtra("retailerId", retailerId);  
 startActivity(intent);  
 }  
 @Override  
 protected void onResume() {  
 super.onResume();  
  
 // Checking if the customer is logged in:  
 SharedPreferences sharedPreferences = PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext());  
 boolean isLoggedIn = sharedPreferences.getBoolean("isLoggedIn", false); // Set is logged in to true:  
  
 // If the customer is not logged in, redirect to the home page:  
 if (!isLoggedIn) {  
 Toast.*makeText*(DiscountActivity.this, "Please log in to continue!", Toast.*LENGTH\_SHORT*).show(); // Display a message to the customer asking them to log in:  
 startActivity(new Intent(DiscountActivity.this, MainActivity.class)); // Redirect to the main page of the application:  
 finish(); // Finishing the current activity so that customers' cannot go back to it when pressing the back button:  
 }  
 }  
}

The **DiscountActivity.java** code does the following:

* In the **onCreate()** function, the following code is present:
* Creates the discount activity (this is the **activity\_discount.xml**) when opening the mobile application.
* Setups up the bottom navigation bar; the three icons do the following:
  + The home icon will redirect the user to the Home screen.
  + The search icon will leave the user on the Discounts Screen.
  + The user icon will redirect the user to the Accounts screen.
* Changes the search icon to a different colour to show users that they are on this page.
* Images are retrieved from the database and are converted from bitmap format to a drawable file format.
* The variables defined for each store are matched with the IDs in the **activity\_discount.xml** file. (These are referred to as widgets.)
* There are four **onClick()** functions, these are used to call the **startProductViewActivity()** function by passing in the **retailerId** as a variable.
* The **startProductViewActivity()** function does the following:
  + It redirects the user to the **ProductViewActivity.java** file.
  + In addition, it also sends the **retailerId** to the **ProductViewActivity.java** file.
* The **onResume()** function has the purpose of keeping the user logged in even after they have closed the mobile application.

### User Interface – Products

This is the products screen (which is the **ProductViewActivity.java** file). Here users can view the discounted products on offer by a particular store and view the discount codes for the products on offer.

The first iteration of the discounted products screen and the current iteration is shown below:

##### Iteration 1

A picture containing screenshot, mobile phone, gadget, communication device

Description automatically generated

Figure 4‑19: Discounted products screen for each store.

##### Iteration 2

A picture containing text, mobile phone, screenshot, communication device

Description automatically generated

Figure 4‑20: Discounted products screen for each store. The mobile application is running on the Emulator Pixel 6 Pro API 30.

**Code Snippet:**

package com.example.sustainablespoonfulapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.MenuItem;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
import com.google.android.material.navigation.NavigationBarView;  
  
import java.util.List;  
  
public class ProductViewActivity extends AppCompatActivity {  
 BottomNavigationView bottom\_nav\_bar;  
  
 private String getRetailerName(int retailerId) {  
 // Retrieve the retailer name based on the retailer ID from the database:  
 DatabaseHelper databaseHelper = new DatabaseHelper(this);  
 return databaseHelper.getRetailerNameById(retailerId);  
 }  
  
 private List<DiscountedProduct> getProductsByRetailer(int retailerId) {  
 // Retrieve the list of products for the specified retailer from the database:  
 DatabaseHelper databaseHelper = new DatabaseHelper(this);  
 return databaseHelper.getDiscountedProductsByRetailerId(retailerId);  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*recycler\_view*);  
  
 // Retrieve the retailer ID passed from the previous activity:  
 int retailerId = getIntent().getIntExtra("retailerId", -1);  
  
 // Set the title of the activity based on the retailer:  
 String retailerName = getRetailerName(retailerId);  
 setTitle(retailerName + " Discounted Items");  
  
 // Set up the RecyclerView:  
 RecyclerView recyclerView = findViewById(R.id.*discounted*);  
  
 ProductAdapter productAdapter = new ProductAdapter(getProductsByRetailer(retailerId), this);  
 recyclerView.setAdapter(productAdapter);  
 recyclerView.setLayoutManager(new LinearLayoutManager(this));  
  
 }  
}

The **ProductViewActivity.java** code does the following:

* In the **onCreate()** function, the following code is present:
  + Creates the discounted products based on the store screen (this is **recycler\_view.xml**) when opening the mobile application. The discounted products based on the store screen contain cards that will contain an image of the discounted product available, the discounted percentage of the product, and the discount code that will be scanned in a store to receive the discounted product.
  + The **getRetailerName()** function retrieves the **retailerId** from the **DiscountActivity.java** file. It creates a new instance of the **DatabaseHelper** class. It returns the store name after calling the **getRetailerNameById** function in the **DatabaseHelper** class and passing in the **retailerId** as a parameter.
  + Sets up the **recyclerView** widget. The widget takes the styling to display the discounted products and repeats it on the page based on the number of results from the SQL query in the **DatabaseHelper** file.
  + It uses a constructor file (which is the **DiscountedProduct.java** file) and an adapter file (which is the **ProductAdapter.java** file which is responsible for populating and managing the discounted products in the **recyclerView**).

### User Interface – Account

This is the user accounts screen that will display the first name, surname, and email address of the logged-in user. The user can also log out here by clicking the logout button.

The first iteration of the account screen (which is the AccountActivity.java file) and the current iteration are shown below:

##### Iteration 1

A screen shot of a cell phone

Description automatically generated with medium confidence

Figure 4‑21: Account screen.

##### Iteration 2

A screen shot of a cell phone

Description automatically generated with medium confidence

Figure 4‑22: Account screen. The mobile application is running on the Emulator Pixel 6 Pro API 30.

**Code Snippet:**

package com.example.sustainablespoonfulapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.preference.PreferenceManager;  
import android.view.MenuItem;  
import android.widget.Toast;  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
import com.google.android.material.navigation.NavigationBarView;  
import androidx.appcompat.app.AlertDialog;  
import android.content.DialogInterface;  
import android.widget.Button;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.widget.TextView;  
  
  
public class AccountActivity extends AppCompatActivity{  
  
 BottomNavigationView bottom\_nav\_bar;  
 Button logoutButton;  
  
 TextView nameTextView;  
 TextView surnameTextView;  
 TextView emailTextView;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_account*); //Create the account page:  
  
 //Checking if the customer is logged in:  
 SharedPreferences sharedPreferences = PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext());  
 String email = sharedPreferences.getString("email","");  
  
 //If the customer is not logged in, display a message and redirect to the main page when opening the application:  
 if(email.isEmpty()){  
 Toast.*makeText*(AccountActivity.this, "Please log in to continue!", Toast.*LENGTH\_SHORT*).show(); //Display a message to the customer asking them to log in:  
 startActivity(new Intent(AccountActivity.this, MainActivity.class)); //Redirect the customer to the main page when opening the application:  
 finish(); //Finishing the current activity so that customers' cannot go back to it when pressing the back button:  
 return; //Return early so that the rest of the method is not executed:  
 }  
  
 bottom\_nav\_bar = findViewById(R.id.*bottom\_nav\_bar*);  
 bottom\_nav\_bar.setSelectedItemId(R.id.*account\_bottom\_navigation*); //Set the account icon to selected when on this page:  
 logoutButton = findViewById(R.id.*account\_logout\_button*);  
 nameTextView = findViewById(R.id.*account\_name\_text*);  
 surnameTextView = findViewById(R.id.*account\_surname\_text*);  
 emailTextView = findViewById(R.id.*account\_email\_text*);  
  
 bottom\_nav\_bar.setOnItemSelectedListener(new NavigationBarView.OnItemSelectedListener() {  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem item) {  
 //Create a switch statement:  
 switch (item.getItemId()){  
 //If the home icon is clicked, go to the home page:  
 case R.id.*home\_bottom\_navigation*:  
 startActivity(new Intent(AccountActivity.this,LandingActivity.class)); //Redirect the customer to the home page:  
 finish(); //Finishing the current activity so that customers' cannot go back to it when pressing the back button:  
 return true;  
 //If the search icon is clicked,go to the search discounts page:  
 case R.id.*search\_bottom\_navigation*:  
 startActivity(new Intent(AccountActivity.this,DiscountActivity.class)); //Redirect the customer to the search discount page:  
 finish(); //Finishing the current activity so that customers' cannot go back to it when pressing the back button:  
 return true;  
 //If the account icon is clicked, stay on the account page:  
 case R.id.*account\_bottom\_navigation*:  
 return true;  
 default:  
 return false;  
 }  
 }  
 });  
  
 //When the logout button is clicked, call the showLogoutConfirmationBox() function:  
 logoutButton.setOnClickListener(v -> {  
 showLogoutConfirmationBox();  
 });  
  
 loadCustomerDetails();  
 }  
  
 private void loadCustomerDetails(){  
 SharedPreferences sharedPreferences = PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext());  
 String email = sharedPreferences.getString("email", "");  
  
 DatabaseHelper databaseHelper = new DatabaseHelper(this);  
 SQLiteDatabase db = databaseHelper.getReadableDatabase();  
  
 String[] projection = {  
 DatabaseHelper.*COLUMN\_CUSTOMER\_NAME*,  
 DatabaseHelper.*COLUMN\_CUSTOMER\_SURNAME*,  
 DatabaseHelper.*COLUMN\_CUSTOMER\_EMAIL* };  
  
 String selection = DatabaseHelper.*COLUMN\_CUSTOMER\_EMAIL* + " =?";  
 String[] selectionArgs = {email};  
  
 Cursor cursor = db.query(  
 DatabaseHelper.*TABLE\_NAME\_CUSTOMER*,  
 projection,  
 selection,  
 selectionArgs,  
 null,  
 null,  
 null  
 );  
  
 if(cursor != null && cursor.moveToFirst()){  
 int customerNameIndex = cursor.getColumnIndex(DatabaseHelper.*COLUMN\_CUSTOMER\_NAME*);  
 int customerSurnameIndex = cursor.getColumnIndex(DatabaseHelper.*COLUMN\_CUSTOMER\_SURNAME*);  
 int customerEmailIndex = cursor.getColumnIndex(DatabaseHelper.*COLUMN\_CUSTOMER\_EMAIL*);  
  
  
 String customerName = cursor.getString(customerNameIndex);  
 String customerSurname = cursor.getString(customerSurnameIndex);  
 String customerEmail = cursor.getString(customerEmailIndex);  
  
 cursor.close();  
 db.close();  
  
 nameTextView.setText("Name: " + customerName);  
 surnameTextView.setText("Surname: " + customerSurname);  
 emailTextView.setText("Email: " + customerEmail);  
 }  
 }  
  
 //Function to display a confirmation box to the customer asking if they want to logout:  
 private void showLogoutConfirmationBox(){  
 AlertDialog.Builder builder = new AlertDialog.Builder(this);  
 builder.setTitle("Logout") //Title of confirmation box:  
 .setMessage("Are you sure that you want to logout?") //Message in confirmation box:  
 .setPositiveButton("Yes", new DialogInterface.OnClickListener() { //Option 1: Customer clicks yes, call the logoutCustomer function:  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 //Log out the customer and redirect to the Home page:  
 logoutCustomer();  
 }  
 })  
 .setNegativeButton("No",null) //Option 2: Customer clicks no:  
 .show();  
 }  
  
 private void logoutCustomer(){  
 //Clear the customers' session:  
 SharedPreferences sharedPreferences = PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext());  
 SharedPreferences.Editor editor = sharedPreferences.edit();  
 editor.remove("email"); //Remove the email address that was stored in the session:  
 editor.putBoolean("isLoggedIn",false);  
 editor.apply(); //Save the changes to the SharedPreferences:  
  
 //Display a message to the customer saying that they have logged out, then redirect to the main page when opening the application:  
 Toast.*makeText*(AccountActivity.this, "Logged out successfully!", Toast.*LENGTH\_SHORT*).show(); //Display a message to the customer saying that they have logged out:  
  
 //Redirect to the main page if the activity is active:  
 Intent intent= new Intent(AccountActivity.this, MainActivity.class); //Redirect the customer to the main page when opening the application:  
 intent.setFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP*);  
 startActivity(intent);  
 finish(); //Finishing the current activity so that customers' cannot go back to it when pressing the back button:  
 }  
}

This **AccountActivity.java** code does the following:

* In the **onCreate()** function, the following code is present:
  + Creates the account screen (which is the **activity\_account.xml**). The account page contains the title ‘Account’, and below the title, there are fields that will display the users’ ‘Name’, ‘Surname’, and ‘Email Address.’ There is a log-out button after the three fields; this is where the user will click and be logged out of the mobile application.
  + It gets the email stored in the session; if the email is not found, it displays an alert message to the user notifying them that they must log in to continue and then redirects the user to the Main screen.
  + Setups up the bottom navigation bar; the three icons do the following:
    - The home icon will redirect the user to the Home screen.
    - The search icon will redirect the user to the Discounts Screen.
    - The user icon will leave the user on the Accounts screen.
  + Changes the account icon to a different colour to show users that they are on this page.
  + Calls the **loadCustomerDetails()** function, which fetches the users’ details based on their email addresses.
  + In the **loadCustomerDetails()** function, there is one **onClick()** function that calls the **logoutCustomer()** function.
  + The **logoutCustomer()** function does the following:
    - It removes the users’ email from the session.
    - It sets the **isLoggedIn** to false.
    - It displays an alert message to the user notifying them that they have been logged out; afterwards, they are redirected to the Main screen.

## Business Logic Development

### Algorithm 1 – Login

**Code Snippet:**

package com.example.sustainablespoonfulapp;  
  
import android.os.Bundle;  
import android.content.Intent;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import android.content.SharedPreferences;  
import android.preference.PreferenceManager;  
  
public class LoginActivity extends AppCompatActivity {  
  
 //Declaring the database helper variable:  
 private DatabaseHelper databaseHelper;  
  
 //Declaring variables for the email and password inputs:  
 private EditText emailEditText, passwordEditText;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*); //Create the login page:  
  
 //Creating an instance of the DatabaseHelper class:  
 databaseHelper = new DatabaseHelper(this);  
 emailEditText = findViewById(R.id.*login\_email\_address\_text*);  
 passwordEditText = findViewById(R.id.*login\_password\_text*);  
  
 //Setting up the login button click listener:  
 Button loginButton = findViewById(R.id.*login\_confirm\_button*);  
  
 loginButton.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 login();  
 }  
 });  
 }  
  
 private void login(){  
 //Getting the email and password that the customer has entered and trimming it:  
 String email = emailEditText.getText().toString().trim();  
 String password = passwordEditText.getText().toString().trim();  
  
 //Checking if any of the input fields are empty before logging in the customer:  
 if(email.isEmpty()||password.isEmpty()){  
 //Display a message to the customer asking them to fill their details in on the form:  
 Toast.*makeText*(LoginActivity.this, "Please fill out all fields in this form.", Toast.*LENGTH\_SHORT*).show();  
 return; //Exit the method early:  
 }  
  
 //Getting a readable database:  
 SQLiteDatabase db = databaseHelper.getReadableDatabase();  
  
 //Checking if the email and password match a record in the customer table:  
 boolean loginSuccess = checkEmailAndPassword(email, password, db);  
  
 //If a record was found and it matches/login was successful display a message and redirect to the landing page:  
 if(loginSuccess){  
 //Store the customer details in Shared Preferences:  
 SharedPreferences sharedPreferences = PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext());  
 SharedPreferences.Editor editor = sharedPreferences.edit();  
 editor.putString("email",email); //Store the customers email:  
 editor.putBoolean("isLoggedIn",true); //Store the customers' login status:  
 editor.apply(); //Save the changes to Shared Preferences:  
  
 //Displaying a success message to the customer notifying them that they have been logged in:  
 Toast.*makeText*(LoginActivity.this, "Logged in successfully!", Toast.*LENGTH\_SHORT*).show();  
 //Redirecting to the home Page:  
 startActivity(new Intent(LoginActivity.this, LandingActivity.class));  
 finish(); //Finishing the current activity so that users cannot go back to it when pressing the back button:  
 }else{ //Password or email was incorrect:  
 //Display an error message to the customer:  
 Toast.*makeText*(LoginActivity.this, "Invalid email or password! Please try again.", Toast.*LENGTH\_SHORT*).show();  
 }  
 //Closing the database:  
 db.close();  
 }  
  
 private boolean checkEmailAndPassword(String email, String password, SQLiteDatabase db){  
 //Define which columns to retrieve from the customer table in the database:  
 String[] projection = {databaseHelper.getColumnEmail()};  
 //Select any email addresses and passwords that are in the customer table that match the email address and password entered:  
 String selection = databaseHelper.getColumnEmail() + " = ? AND " + databaseHelper.getColumnPassword() + " = ?";  
 //Specify the arguments for the query, this will be email and password:  
 String[] selectionArgs = {email, password};  
 //Query the customer table for any matching records:  
 Cursor cursor = db.query(DatabaseHelper.*TABLE\_NAME\_CUSTOMER*, projection, selection, selectionArgs, null, null, null);  
 //Checking if any records were found/records greater than 0:  
 boolean exists = cursor.getCount()>0;  
 //Close the cursor so that associated resources can be released:  
 cursor.close();  
 //Return the result (if the email exists or not):  
 return exists;  
 }  
}

This snippet of code for login does the following:

* In the **onCreate()** function, the following code is present:
  + Created an instance of the **DatabaseHelper** class.
  + Creates the login screen (which is the **activity\_login.xml**). The login screen contains the title ‘Login’ with the following fields that consist of ‘Email Address’ and ‘Password’. These two fields are displayed in input boxes on the screen.
  + There is one **onClick()** function, this will call the **login()** function.
  + The **login()** function does the following:
    - If the email and password textboxes are left empty, it displays an alert message for the user to fill in their details.
    - Defines a variable called **loginSuccess** which calls the **checkEmailAndPassword()** function and passes in **email**, **password** and **db** as parameters.
    - The **checkEmailAndPassword()** function does the following:
      * It trims the values that the users have entered and checks if the email address and password entered match the details in the customer table.
      * It returns true if it matches and false if it does not match.
      * If the result is true, the code does the following:
        + Stores the users’ **email** in the session.
        + Defines a variable called **isLoggedIn** and sets it to true. (This variable will be used on other pages to check if the user is logged in.)
        + Displays an alert message on the screen notifying the user that they have been logged in. Afterwards, the user is redirected to the landing screen (which is the **LandingActivity.java file**).
        + If the result returned is false, display an alert message notifying the user that they have an invalid email or password and that they should try to enter their details again.

### Algorithm 2 – Discounted Products

**Code Snippet:**

//Function to fetch the discounted products in the discounted products table based on the Retailer ID:  
public List<DiscountedProduct> getDiscountedProductsByRetailerId(int retailerId) {  
 List<DiscountedProduct> discountedProducts = new ArrayList<>();  
 SQLiteDatabase database = getReadableDatabase();  
  
 String[] columns = {  
 *COLUMN\_DISCOUNT\_RETAILER\_ID*,  
 *COLUMN\_DISCOUNT\_CODE*,  
 *COLUMN\_DISCOUNT\_PERCENTAGE*,  
 *COLUMN\_DISCOUNT\_PRODUCT\_NAME*,  
 *COLUMN\_DISCOUNT\_IMAGE* };  
  
 String selection = *COLUMN\_DISCOUNT\_RETAILER\_ID* + " = ?";  
 String[] selectionArgs = { String.*valueOf*(retailerId) };  
  
 Cursor cursor = database.query(  
 *TABLE\_NAME\_DISCOUNTED\_PRODUCTS*,  
 columns,  
 selection,  
 selectionArgs,  
 null,  
 null,  
 null  
 );  
  
 int columnDiscountCode = cursor.getColumnIndex(*COLUMN\_DISCOUNT\_CODE*);  
 int columnDiscountPercentage = cursor.getColumnIndex(*COLUMN\_DISCOUNT\_PERCENTAGE*);  
 int columnProductName = cursor.getColumnIndex(*COLUMN\_DISCOUNT\_PRODUCT\_NAME*);  
 int columnDiscountImage = cursor.getColumnIndex(*COLUMN\_DISCOUNT\_IMAGE*);  
  
 if (cursor != null && cursor.moveToFirst()) {  
 do {  
 String discountCode = cursor.getString(columnDiscountCode);  
 String discountPercentage = cursor.getString(columnDiscountPercentage);  
 String productName = cursor.getString(columnProductName);  
 byte[] discountImage = cursor.getBlob(columnDiscountImage);  
  
 DiscountedProduct discountedProduct = new DiscountedProduct(discountCode, discountPercentage, productName, discountImage);  
 discountedProducts.add(discountedProduct);  
 } while (cursor.moveToNext());  
 }  
  
 cursor.close();  
 database.close();  
  
 return discountedProducts;  
}

This snippet of code for discounted products does the following:

* It creates a new **ArrayList** called **discountedProducts** to store all the discounted products.
* It gets a readable database.
* It runs a SQL query to fetch the values from all the columns in the **discounted products** table based on the **retailer id**.
* It uses **getColumnIndex()** to get the column index for the results and stores it.
* If the cursor is not null and it **movesToFirst()**, it converts the column indexes into strings.
* It creates a new instance of the **discountedProducts ArrayList** and passes in all the columns from the **discounted products** table.
* It adds the columns to the **discountedProducts ArrayList.**
* This is done why the **cursor** moves to the next result.
* Afterwards the **cursor** is closed, and the **database connection** is closed.
* The **discountedProducts ArrayList** is returned.

## Data Access Development

### Database Implementation

For our mobile application, we have chosen to work with SQLite.

SQLite is an embedded relational database management system that is widely used in mobile development (SQLite, 2023).

To implement SQLite, the **gradle.build** was included in the file, and a new class called **DatabaseHelper** was created. The tables were then created, and options were provided for them to be created in the other classes when the **onCreate()** function was called. If it has been upgraded since its last use, the table is dropped and recreated again. This will ensure the code remains reusable.

The reason we have used SQLite for this mobile application is that it is a lightweight, self-contained, reliable database engine that is available locally and does not require any administration to function (SQLite, 2023). SQLite also comes built into Android Studio, making it the ideal choice for this mobile application (SQLite, 2023).

#### Database Creation Class – DatabaseHelper.java

**Code Snippet:**

package com.example.sustainablespoonfulapp;  
  
import android.annotation.SuppressLint;  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import java.io.ByteArrayOutputStream;  
import java.util.ArrayList;  
import java.util.List;  
import android.database.Cursor;  
  
public class DatabaseHelper extends SQLiteOpenHelper{  
  
 //Providing the database name and version:  
 public static final String *DATABASE\_NAME* = "sustainable\_spoonful.db";  
 public static final int *DATABASE\_VERSION* = 1;  
  
 //Customer Table Constants:  
 public static final String *TABLE\_NAME\_CUSTOMER* = "customer";  
 public static final String *COLUMN\_CUSTOMER\_ID* = "customer\_id";  
 public static final String *COLUMN\_CUSTOMER\_NAME* = "customer\_name";  
 public static final String *COLUMN\_CUSTOMER\_SURNAME* = "customer\_surname";  
 public static final String *COLUMN\_CUSTOMER\_EMAIL* = "customer\_email";  
 public static final String *COLUMN\_CUSTOMER\_PASSWORD* = "customer\_password";  
  
 //Retailer Table Constants:  
 public static final String *TABLE\_NAME\_RETAILER* = "retailer";  
 public static final String *COLUMN\_RETAILER\_ID* = "retailer\_id";  
 public static final String *COLUMN\_RETAILER\_NAME* = "retailer\_name";  
 public static final String *COLUMN\_RETAILER\_ADDRESS* = "address";  
 public static final String *COLUMN\_RETAILER\_IMAGE* = "retailer\_image";  
  
 //Discounted Products Table Constants:  
 public static final String *TABLE\_NAME\_DISCOUNTED\_PRODUCTS* = "discounted\_products";  
 public static final String *COLUMN\_DISCOUNT\_ID* = "discount\_id";  
 public static final String *COLUMN\_DISCOUNT\_CODE* = "discount\_code";  
 public static final String *COLUMN\_DISCOUNT\_RETAILER\_ID* = "retailer\_id";  
 public static final String *COLUMN\_DISCOUNT\_PRODUCT\_NAME* = "product\_name";  
 public static final String *COLUMN\_DISCOUNT\_PERCENTAGE* = "discount\_percentage";  
 public static final String *COLUMN\_DISCOUNT\_IMAGE* = "discount\_image";  
  
 private final Context context; // Add a member variable to store the Context object:  
  
 public DatabaseHelper(Context context){  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 this.context = context; //Assign the context to the member variable:  
 }  
  
 @Override  
 public void onCreate(SQLiteDatabase db){ //Creates database tables and inserts initial data:  
 //Create the database tables:  
 //CUSTOMER TABLE:  
 String createCustomerTable = "CREATE TABLE " + *TABLE\_NAME\_CUSTOMER* + " (" +  
 *COLUMN\_CUSTOMER\_ID* + " INTEGER PRIMARY KEY," +  
 *COLUMN\_CUSTOMER\_NAME* + " TEXT," +  
 *COLUMN\_CUSTOMER\_SURNAME* + " TEXT," +  
 *COLUMN\_CUSTOMER\_EMAIL* + " TEXT," +  
 *COLUMN\_CUSTOMER\_PASSWORD* + " TEXT)";  
 db.execSQL(createCustomerTable);  
  
 //RETAILER TABLE:  
 String createRetailerTable = "CREATE TABLE " + *TABLE\_NAME\_RETAILER* + " (" +  
 *COLUMN\_RETAILER\_ID* + " INTEGER PRIMARY KEY," +  
 *COLUMN\_RETAILER\_NAME* + " TEXT," +  
 *COLUMN\_RETAILER\_ADDRESS* + " TEXT," +  
 *COLUMN\_RETAILER\_IMAGE* + " BLOB)"; //Retailer image column  
 db.execSQL(createRetailerTable);  
  
 //DISCOUNTED PRODUCTS TABLE:  
 String createDiscountedProductsTable = "CREATE TABLE " + *TABLE\_NAME\_DISCOUNTED\_PRODUCTS* + " (" +  
 *COLUMN\_DISCOUNT\_ID* + " INTEGER PRIMARY KEY," +  
 *COLUMN\_DISCOUNT\_CODE* + " TEXT," +  
 *COLUMN\_DISCOUNT\_PERCENTAGE* + " TEXT," +  
 *COLUMN\_DISCOUNT\_PRODUCT\_NAME* + " TEXT," +  
 *COLUMN\_DISCOUNT\_RETAILER\_ID* + " INTEGER," +  
 *COLUMN\_DISCOUNT\_IMAGE* + " BLOB," + //Discounted Product Image Column  
 "FOREIGN KEY (" + *COLUMN\_DISCOUNT\_RETAILER\_ID* + ") REFERENCES " +  
 *TABLE\_NAME\_RETAILER* + "(" + *COLUMN\_RETAILER\_ID* + "))";  
 db.execSQL(createDiscountedProductsTable);  
  
 //Function to insert the data into the Retailer Table:  
 insertRetailerData(db);  
  
 //Function to insert the data into the Discounted Products Table:  
 insertDiscountedProductsData(db);  
 }  
  
 //Function to fetch the discounted products in the discounted products table based on the Retailer ID:  
 public List<DiscountedProduct> getDiscountedProductsByRetailerId(int retailerId) {  
 List<DiscountedProduct> discountedProducts = new ArrayList<>();  
 SQLiteDatabase database = getReadableDatabase();  
  
 String[] columns = {  
 *COLUMN\_DISCOUNT\_RETAILER\_ID*,  
 *COLUMN\_DISCOUNT\_CODE*,  
 *COLUMN\_DISCOUNT\_PERCENTAGE*,  
 *COLUMN\_DISCOUNT\_PRODUCT\_NAME*,  
 *COLUMN\_DISCOUNT\_IMAGE* };  
  
 String selection = *COLUMN\_DISCOUNT\_RETAILER\_ID* + " = ?";  
 String[] selectionArgs = { String.*valueOf*(retailerId) };  
  
 Cursor cursor = database.query(  
 *TABLE\_NAME\_DISCOUNTED\_PRODUCTS*,  
 columns,  
 selection,  
 selectionArgs,  
 null,  
 null,  
 null  
 );  
  
 int columnDiscountCode = cursor.getColumnIndex(*COLUMN\_DISCOUNT\_CODE*);  
 int columnDiscountPercentage = cursor.getColumnIndex(*COLUMN\_DISCOUNT\_PERCENTAGE*);  
 int columnProductName = cursor.getColumnIndex(*COLUMN\_DISCOUNT\_PRODUCT\_NAME*);  
 int columnDiscountImage = cursor.getColumnIndex(*COLUMN\_DISCOUNT\_IMAGE*);  
  
 if (cursor != null && cursor.moveToFirst()) {  
 do {  
 String discountCode = cursor.getString(columnDiscountCode);  
 String discountPercentage = cursor.getString(columnDiscountPercentage);  
 String productName = cursor.getString(columnProductName);  
 byte[] discountImage = cursor.getBlob(columnDiscountImage);  
  
 DiscountedProduct discountedProduct = new DiscountedProduct(discountCode, discountPercentage, productName, discountImage);  
 discountedProducts.add(discountedProduct);  
 } while (cursor.moveToNext());  
 }  
  
 cursor.close();  
 database.close();  
  
 return discountedProducts;  
 }  
  
  
 //Function to count the discounted products in the discounted products table based on the Retailer ID:  
 public String getDiscountedProductCount(int retailerId){  
 //Get a readable database:  
 SQLiteDatabase datab = getReadableDatabase();  
  
 //Create the query to count all of the products in the discounted products table based on the retailer id and return the result:  
 String query = "SELECT COUNT(\*) FROM " + *TABLE\_NAME\_DISCOUNTED\_PRODUCTS* +  
 " WHERE " + *COLUMN\_DISCOUNT\_RETAILER\_ID* + " = " + retailerId;  
 Cursor cursor = datab.rawQuery(query, null);  
  
 //Initialise the counter to 0:  
 int count=0;  
  
 if(cursor !=null && cursor.moveToFirst()){  
 count = cursor.getInt(0);  
 }  
 cursor.close();  
  
 if(count == 0){ //No result was found, display "No Discounts"  
 return "No Discounts";  
 }else{ //Result was found, fetch the number of products and concatenate with a string saying 'Discounts'  
 return count + " Discounts";  
 }  
 }  
  
 //Function to convert drawable resources to bitmap:  
 private byte[] convertImageToByteArray(int imageResource){  
 Bitmap bitmap = BitmapFactory.*decodeResource*(context.getResources(), imageResource);  
 ByteArrayOutputStream stream = new ByteArrayOutputStream();  
 bitmap.compress(Bitmap.CompressFormat.*PNG*, 100, stream);  
 return stream.toByteArray();  
 }  
  
 //Method to retrieve all the store images from the retailer table:  
 public List<byte[]> getAllStoreImages() {  
 List<byte[]> storeImages = new ArrayList<>();  
 SQLiteDatabase database = getReadableDatabase();  
 String query = "SELECT " + *COLUMN\_RETAILER\_IMAGE* + " FROM " + *TABLE\_NAME\_RETAILER*; //query to select all retailer images from the retailer table:  
 Cursor cursor = database.rawQuery(query, null);  
 int columnIndex = cursor.getColumnIndex(*COLUMN\_RETAILER\_IMAGE*);  
  
 if(columnIndex != -1){  
 if(cursor.moveToFirst()){  
 do{  
 byte[] image = cursor.getBlob(columnIndex);  
 storeImages.add(image);  
 }while(cursor.moveToNext());  
 }  
 }  
 cursor.close();  
 return storeImages;  
 }  
  
 @SuppressLint("Range")  
 public String getRetailerNameById(int retailerId) {  
 SQLiteDatabase db = this.getReadableDatabase();  
 String retailerName = "";  
  
 // Query the database to fetch the retailer name based on retailerId  
 String[] projection = {*COLUMN\_RETAILER\_NAME*};  
 String selection = *COLUMN\_RETAILER\_ID* + " = ?";  
 String[] selectionArgs = {String.*valueOf*(retailerId)};  
  
 Cursor cursor = db.query(*TABLE\_NAME\_RETAILER*, projection, selection, selectionArgs, null, null, null);  
  
 if (cursor.moveToFirst() && cursor.getColumnIndex(*COLUMN\_RETAILER\_NAME*) != -1) {  
 retailerName = cursor.getString(cursor.getColumnIndex(*COLUMN\_RETAILER\_NAME*));  
 }  
  
 cursor.close();  
 db.close();  
  
 return retailerName;  
 }  
  
 private void insertRetailerData(SQLiteDatabase db){  
 //Create a ContentValues object so that it can hold the column values for each row:  
 ContentValues values = new ContentValues();  
  
 //Inserting the first shop:  
 values.put(*COLUMN\_RETAILER\_NAME*, "PicknPay");  
 values.put(*COLUMN\_RETAILER\_ADDRESS*, "Greenstone");  
 values.put(*COLUMN\_RETAILER\_IMAGE*, convertImageToByteArray(R.drawable.*picknpay\_discount*));  
 db.insert(*TABLE\_NAME\_RETAILER*, null, values);  
  
 //Inserting the second shop:  
 values.clear(); //Clear the ContentValues object so that it can be reused:  
 values.put(*COLUMN\_RETAILER\_NAME*, "Woolworths");  
 values.put(*COLUMN\_RETAILER\_ADDRESS*, "Greenstone");  
 values.put(*COLUMN\_RETAILER\_IMAGE*, convertImageToByteArray(R.drawable.*woolworths\_discount*));  
 db.insert(*TABLE\_NAME\_RETAILER*, null, values);  
  
 //Inserting the third shop:  
 values.clear(); //Clear the ContentValues object so that it can be reused:  
 values.put(*COLUMN\_RETAILER\_NAME*, "Checkers");  
 values.put(*COLUMN\_RETAILER\_ADDRESS*, "Meadowdale");  
 values.put(*COLUMN\_RETAILER\_IMAGE*, convertImageToByteArray(R.drawable.*checkers\_discount*));  
 db.insert(*TABLE\_NAME\_RETAILER*, null, values);  
  
 //Inserting the fourth shop:  
 values.clear(); //Clear the ContentValues object so that it can be reused:  
 values.put(*COLUMN\_RETAILER\_NAME*, "Food Lover's Market");  
 values.put(*COLUMN\_RETAILER\_ADDRESS*, "Greenstone");  
 values.put(*COLUMN\_RETAILER\_IMAGE*, convertImageToByteArray(R.drawable.*foodloversmarket\_discount*));  
 db.insert(*TABLE\_NAME\_RETAILER*, null, values);  
 }  
  
 public void insertDiscountedProductsData(SQLiteDatabase db){  
 //Create a ContentValues object so that it can hold the column values for each row:  
 ContentValues values = new ContentValues();  
  
 //Insert the first discounted product:  
 values.put(*COLUMN\_DISCOUNT\_CODE*, "P1CkNP@yI1");  
 values.put(*COLUMN\_DISCOUNT\_PERCENTAGE*, "10% OFF");  
 values.put(*COLUMN\_DISCOUNT\_PRODUCT\_NAME*, "White Bread Loaf");  
 values.put(*COLUMN\_DISCOUNT\_RETAILER\_ID*, 1); //PicknPay is the first store that is loaded:  
 values.put(*COLUMN\_DISCOUNT\_IMAGE*, convertImageToByteArray(R.drawable.*bread\_compressed*));  
 db.insert(*TABLE\_NAME\_DISCOUNTED\_PRODUCTS*, null, values);  
  
  
 values.clear(); //Clear the ContentValues object so that it can be reused:  
 values.put(*COLUMN\_DISCOUNT\_CODE*, "P1CkNP@yO2");  
 values.put(*COLUMN\_DISCOUNT\_PERCENTAGE*, "20% OFF");  
 values.put(*COLUMN\_DISCOUNT\_PRODUCT\_NAME*, "Oranges 1 KG");  
 values.put(*COLUMN\_DISCOUNT\_RETAILER\_ID*, 1); //PicknPay is the first store that is loaded:  
 values.put(*COLUMN\_DISCOUNT\_IMAGE*, convertImageToByteArray(R.drawable.*oranges\_compressed*));  
 db.insert(*TABLE\_NAME\_DISCOUNTED\_PRODUCTS*, null, values);  
  
 values.clear(); //Clear the ContentValues object so that it can be reused:  
 values.put(*COLUMN\_DISCOUNT\_CODE*, "P1CkNP@yS0");  
 values.put(*COLUMN\_DISCOUNT\_PERCENTAGE*, "30% OFF");  
 values.put(*COLUMN\_DISCOUNT\_PRODUCT\_NAME*, "Sunflower Oil 1 L");  
 values.put(*COLUMN\_DISCOUNT\_RETAILER\_ID*, 1); //PicknPay is the first store that is loaded:  
 values.put(*COLUMN\_DISCOUNT\_IMAGE*, convertImageToByteArray(R.drawable.*oil\_compressed*));  
 db.insert(*TABLE\_NAME\_DISCOUNTED\_PRODUCTS*, null, values);  
  
 //Insert the first discounted product:  
 values.clear(); //Clear the ContentValues object so that it can be reused:  
 values.put(*COLUMN\_DISCOUNT\_CODE*, "W0oLWORTHSM3");  
 values.put(*COLUMN\_DISCOUNT\_PERCENTAGE*, "15% OFF");  
 values.put(*COLUMN\_DISCOUNT\_PRODUCT\_NAME*, "Full Cream Milk 1 L");  
 values.put(*COLUMN\_DISCOUNT\_RETAILER\_ID*, 2); //Woolworths is the second store that is loaded:  
 values.put(*COLUMN\_DISCOUNT\_IMAGE*, convertImageToByteArray(R.drawable.*milk\_compressed*));  
 db.insert(*TABLE\_NAME\_DISCOUNTED\_PRODUCTS*, null, values);  
  
 values.clear(); //Clear the ContentValues object so that it can be reused:  
 values.put(*COLUMN\_DISCOUNT\_CODE*, "W0oLWORTHSBB");  
 values.put(*COLUMN\_DISCOUNT\_PERCENTAGE*, "10% OFF");  
 values.put(*COLUMN\_DISCOUNT\_PRODUCT\_NAME*, "Pack of Bananas");  
 values.put(*COLUMN\_DISCOUNT\_RETAILER\_ID*, 2); //Woolworths is the second store that is loaded:  
 values.put(*COLUMN\_DISCOUNT\_IMAGE*, convertImageToByteArray(R.drawable.*bananas\_compressed*));  
 db.insert(*TABLE\_NAME\_DISCOUNTED\_PRODUCTS*, null, values);  
  
 values.clear(); //Clear the ContentValues object so that it can be reused:  
 values.put(*COLUMN\_DISCOUNT\_CODE*, "W0oLWORTHSA@");  
 values.put(*COLUMN\_DISCOUNT\_PERCENTAGE*, "30% OFF");  
 values.put(*COLUMN\_DISCOUNT\_PRODUCT\_NAME*, "Apples 1.5 KG");  
 values.put(*COLUMN\_DISCOUNT\_RETAILER\_ID*, 2); //Woolworths is the second store that is loaded:  
 values.put(*COLUMN\_DISCOUNT\_IMAGE*, convertImageToByteArray(R.drawable.*apples\_compressed*));  
 db.insert(*TABLE\_NAME\_DISCOUNTED\_PRODUCTS*, null, values);  
  
 values.clear(); //Clear the ContentValues object so that it can be reused:  
 values.put(*COLUMN\_DISCOUNT\_CODE*, "CHECKER$U$");  
 values.put(*COLUMN\_DISCOUNT\_PERCENTAGE*, "60% OFF");  
 values.put(*COLUMN\_DISCOUNT\_PRODUCT\_NAME*, "Ultimate Sandwich");  
 values.put(*COLUMN\_DISCOUNT\_RETAILER\_ID*, 3); //Checkers is the third store that is loaded:  
 values.put(*COLUMN\_DISCOUNT\_IMAGE*, convertImageToByteArray(R.drawable.*sandwich\_compressed*));  
 db.insert(*TABLE\_NAME\_DISCOUNTED\_PRODUCTS*, null, values);  
  
 values.clear(); //Clear the ContentValues object so that it can be reused:  
 values.put(*COLUMN\_DISCOUNT\_CODE*, "CHECKER$OO");  
 values.put(*COLUMN\_DISCOUNT\_PERCENTAGE*, "35% OFF");  
 values.put(*COLUMN\_DISCOUNT\_PRODUCT\_NAME*, "Onions 1 KG");  
 values.put(*COLUMN\_DISCOUNT\_RETAILER\_ID*, 3); //Checkers is the third store that is loaded:  
 values.put(*COLUMN\_DISCOUNT\_IMAGE*, convertImageToByteArray(R.drawable.*onions\_compressed*));  
 db.insert(*TABLE\_NAME\_DISCOUNTED\_PRODUCTS*, null, values);  
  
 values.clear(); //Clear the ContentValues object so that it can be reused:  
 values.put(*COLUMN\_DISCOUNT\_CODE*, "CHECKER$AP");  
 values.put(*COLUMN\_DISCOUNT\_PERCENTAGE*, "60% OFF");  
 values.put(*COLUMN\_DISCOUNT\_PRODUCT\_NAME*, "Ace Pap 1 KG");  
 values.put(*COLUMN\_DISCOUNT\_RETAILER\_ID*, 3); //Checkers is the third store that is loaded:  
 values.put(*COLUMN\_DISCOUNT\_IMAGE*, convertImageToByteArray(R.drawable.*ace\_pap\_compressed*));  
 db.insert(*TABLE\_NAME\_DISCOUNTED\_PRODUCTS*, null, values);  
  
 values.clear(); //Clear the ContentValues object so that it can be reused:  
 values.put(*COLUMN\_DISCOUNT\_CODE*, "F00DLOVER$R1");  
 values.put(*COLUMN\_DISCOUNT\_PERCENTAGE*, "40% OFF");  
 values.put(*COLUMN\_DISCOUNT\_PRODUCT\_NAME*, "Brown Rice 1 KG");  
 values.put(*COLUMN\_DISCOUNT\_RETAILER\_ID*, 4); //Food Lover's Market is the fourth store that is loaded:  
 values.put(*COLUMN\_DISCOUNT\_IMAGE*, convertImageToByteArray(R.drawable.*brownrice\_compressed*));  
 db.insert(*TABLE\_NAME\_DISCOUNTED\_PRODUCTS*, null, values);  
  
 values.clear(); //Clear the ContentValues object so that it can be reused:  
 values.put(*COLUMN\_DISCOUNT\_CODE*, "F00DLOVER$UT");  
 values.put(*COLUMN\_DISCOUNT\_PERCENTAGE*, "60% OFF");  
 values.put(*COLUMN\_DISCOUNT\_PRODUCT\_NAME*, "UTD Potatoes 1 KG");  
 values.put(*COLUMN\_DISCOUNT\_RETAILER\_ID*, 4); //Food Lover's Market is the fourth store that is loaded:  
 values.put(*COLUMN\_DISCOUNT\_IMAGE*, convertImageToByteArray(R.drawable.*potatoes\_compressed*));  
 db.insert(*TABLE\_NAME\_DISCOUNTED\_PRODUCTS*, null, values);  
  
 values.clear(); //Clear the ContentValues object so that it can be reused:  
 values.put(*COLUMN\_DISCOUNT\_CODE*, "F00DLOVER$SC");  
 values.put(*COLUMN\_DISCOUNT\_PERCENTAGE*, "70% OFF");  
 values.put(*COLUMN\_DISCOUNT\_PRODUCT\_NAME*, "Straw-Choc Cake");  
 values.put(*COLUMN\_DISCOUNT\_RETAILER\_ID*, 4); //Food Lover's Market is the fourth store that is loaded:  
 values.put(*COLUMN\_DISCOUNT\_IMAGE*, convertImageToByteArray(R.drawable.*cake\_compressed*));  
 db.insert(*TABLE\_NAME\_DISCOUNTED\_PRODUCTS*, null, values);  
  
 values.clear(); //Clear the ContentValues object so that it can be reused:  
 values.put(*COLUMN\_DISCOUNT\_CODE*, "F00DLOVER$BB");  
 values.put(*COLUMN\_DISCOUNT\_PERCENTAGE*, "80% OFF");  
 values.put(*COLUMN\_DISCOUNT\_PRODUCT\_NAME*, "Bread Rolls x6");  
 values.put(*COLUMN\_DISCOUNT\_RETAILER\_ID*, 4); //Food Lover's Market is the fourth store that is loaded:  
 values.put(*COLUMN\_DISCOUNT\_IMAGE*, convertImageToByteArray(R.drawable.*rolls\_compressed*));  
 db.insert(*TABLE\_NAME\_DISCOUNTED\_PRODUCTS*, null, values);  
 }  
  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion){ //Upgrade the database schema if necessary:  
 db.execSQL("DROP TABLE IF EXISTS " + *TABLE\_NAME\_CUSTOMER*);  
 db.execSQL("DROP TABLE IF EXISTS " + *TABLE\_NAME\_RETAILER*);  
 db.execSQL("DROP TABLE IF EXISTS " + *TABLE\_NAME\_DISCOUNTED\_PRODUCTS*);  
 onCreate(db);  
 }  
  
 //Use a get method for the email column:  
 public String getColumnEmail(){  
 return *COLUMN\_CUSTOMER\_EMAIL*;  
 }  
  
 //Use a get method for the password column:  
 public String getColumnPassword(){  
 return *COLUMN\_CUSTOMER\_PASSWORD*;  
 }  
}

This code snippet does the following:

* It defines the three database tables (**customer**, **retailer**, **discounted\_products**).
* In the **onCreate()** function, it creates the three database tables. Furthermore, it calls the two functions (**insertRetailerData()** and **insertDiscountedProduct()**) that insert initial data when the mobile application is opened for the first time.
* There are additional functions (**getDiscountedProductsByRetailerId()**, **getDiscountedProductCount()**, **convertImageToByteArray()**, **getAllStoreImages()**, **getRetailerNameById()**, **getColumnEmail()**, and **getColumnPassword()**) that are present; these functions are used throughout the mobile application.
* If the database is changed or upgraded, the **onUpgrade()** function is called; this function will drop the database tables and recreate them.

### Data Access Adapters

#### Accessor Methods - Account

**Code Snippet:**

private void loadCustomerDetails(){  
 SharedPreferences sharedPreferences = PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext());  
 String email = sharedPreferences.getString("email", "");  
  
 DatabaseHelper databaseHelper = new DatabaseHelper(this);  
 SQLiteDatabase db = databaseHelper.getReadableDatabase();  
  
 String[] projection = {  
 DatabaseHelper.*COLUMN\_CUSTOMER\_NAME*,  
 DatabaseHelper.*COLUMN\_CUSTOMER\_SURNAME*,  
 DatabaseHelper.*COLUMN\_CUSTOMER\_EMAIL* };  
  
 String selection = DatabaseHelper.*COLUMN\_CUSTOMER\_EMAIL* + " =?";  
 String[] selectionArgs = {email};  
  
 Cursor cursor = db.query(  
 DatabaseHelper.*TABLE\_NAME\_CUSTOMER*,  
 projection,  
 selection,  
 selectionArgs,  
 null,  
 null,  
 null  
 );  
  
 if(cursor != null && cursor.moveToFirst()){  
 int customerNameIndex = cursor.getColumnIndex(DatabaseHelper.*COLUMN\_CUSTOMER\_NAME*);  
 int customerSurnameIndex = cursor.getColumnIndex(DatabaseHelper.*COLUMN\_CUSTOMER\_SURNAME*);  
 int customerEmailIndex = cursor.getColumnIndex(DatabaseHelper.*COLUMN\_CUSTOMER\_EMAIL*);  
  
  
 String customerName = cursor.getString(customerNameIndex);  
 String customerSurname = cursor.getString(customerSurnameIndex);  
 String customerEmail = cursor.getString(customerEmailIndex);  
  
 cursor.close();  
 db.close();  
  
 nameTextView.setText("Name: " + customerName);  
 surnameTextView.setText("Surname: " + customerSurname);  
 emailTextView.setText("Email: " + customerEmail);  
 }  
}

This code snippet does the following:

* It fetches the email from the session.
* It creates a new instance of the **DatabaseHelper** class.
* It gets a readable database.
* It uses an SQL query to retrieve the **name**, **surname** and **email** of the user based on the **email** that is stored in the session.
* If the cursor is not null and it **movesToFirst(),** it uses **getColumnIndex()** to get the column index for the results and stores it.
* It converts the column indexes into strings.
* The **cursor** is closed, and the **database connection** is closed.
* The **name**, **surname** and **email** retrieved from the database are set in the relevant **TextViews** that will be displayed on the account page.

#### Accessor Methods – Discounted Products

//Function to fetch the discounted products in the discounted products table based on the Retailer ID:  
public List<DiscountedProduct> getDiscountedProductsByRetailerId(int retailerId) {  
 List<DiscountedProduct> discountedProducts = new ArrayList<>();  
 SQLiteDatabase database = getReadableDatabase();  
  
 String[] columns = {  
 *COLUMN\_DISCOUNT\_RETAILER\_ID*,  
 *COLUMN\_DISCOUNT\_CODE*,  
 *COLUMN\_DISCOUNT\_PERCENTAGE*,  
 *COLUMN\_DISCOUNT\_PRODUCT\_NAME*,  
 *COLUMN\_DISCOUNT\_IMAGE* };  
  
 String selection = *COLUMN\_DISCOUNT\_RETAILER\_ID* + " = ?";  
 String[] selectionArgs = { String.*valueOf*(retailerId) };  
  
 Cursor cursor = database.query(  
 *TABLE\_NAME\_DISCOUNTED\_PRODUCTS*,  
 columns,  
 selection,  
 selectionArgs,  
 null,  
 null,  
 null  
 );  
  
 int columnDiscountCode = cursor.getColumnIndex(*COLUMN\_DISCOUNT\_CODE*);  
 int columnDiscountPercentage = cursor.getColumnIndex(*COLUMN\_DISCOUNT\_PERCENTAGE*);  
 int columnProductName = cursor.getColumnIndex(*COLUMN\_DISCOUNT\_PRODUCT\_NAME*);  
 int columnDiscountImage = cursor.getColumnIndex(*COLUMN\_DISCOUNT\_IMAGE*);  
  
 if (cursor != null && cursor.moveToFirst()) {  
 do {  
 String discountCode = cursor.getString(columnDiscountCode);  
 String discountPercentage = cursor.getString(columnDiscountPercentage);  
 String productName = cursor.getString(columnProductName);  
 byte[] discountImage = cursor.getBlob(columnDiscountImage);  
  
 DiscountedProduct discountedProduct = new DiscountedProduct(discountCode, discountPercentage, productName, discountImage);  
 discountedProducts.add(discountedProduct);  
 } while (cursor.moveToNext());  
 }  
  
 cursor.close();  
 database.close();  
  
 return discountedProducts;  
}

This code snippet does the following:

* It creates a new **ArrayList** called **discountedProducts** to store all the discounted products.
* It gets a readable database.
* It runs a SQL query to fetch the values from all the columns in the **discounted products** table based on the **retailer id**.
* It uses **getColumnIndex()** to get the column index for the results and stores it.
* If the cursor is not null and it **movesToFirst()**, it converts the column indexes into strings.
* It creates a new instance of the **discountedProducts ArrayList** and passes in all the columns from the **discounted products** table.
* It adds the columns to the **discountedProducts ArrayList.**
* This is done why the **cursor** moves to the next result.
* Afterwards the **cursor** is closed, and the **database connection** is closed.
* The **discountedProducts ArrayList** is returned.

## Conclusion

In summary, multiple methods were used when approaching the system prototyping and testing phase.

These methods include building on the initial designs to develop a visually appealing application, focusing on the core functionality of the application and using unit testing to ensure it is functioning as it should be, finding the most appropriate database implementation technology to better manage the user data, and combing these aspects to create a functioning mobile application.

Using iterative and incremental design, the application has been built on that has been conceptualized in chapter one. While this remains an ongoing process, great strides were made in building on the initial concepts and move further towards a mobile application that meets all of the requirements.

# Results, Conclusion, and Recommendations

## Results

### Research Findings

Through research, the team has made several key discoveries regarding food wastage and its impact on communities. These findings further emphasize the need for a mobile application like The Sustainable Spoonful to address this pressing issue.

1. **The Extent of Food Wastage:**

The research revealed that food wastage is a significant problem in South Africa, particularly in retail food stores, cafes, and restaurants. Large amounts of food are discarded either due to reaching the expiration date or customers not finishing meals (Mandaha, 2021). This wastage contributes to the loss of valuable resources and has both moral and financial implications (cgcsa, 2020).

1. **Impact on Communities:**

Food wastage had adverse effects on communities, especially those facing food insecurity and financial constraint. The research highlighted that many communities in South Africa struggle to afford nutritious food, resulting in health issues and malnutrition (WWF, 2017). By tackling food wastage, The Sustainable Spoonful aims to bridge this gap and provide affordable options for these communities, promoting the community’s overall well-being (WWF, 2017).

1. **Economic Losses:**

The financial implications of food wastage are substantial. The research indicated that businesses, including food retail stores, cafes, and restaurants, experience financial losses due to discarded food products (EPA, 2021). By reducing wastage and optimizing resource utilization, The Sustainable Spoonful can help businesses save money, increase profitability, and operate more sustainably (Averda, 2022).

1. **Environmental Impact:**

Another significant finding of the research is the environmental impact of food wastage (WWF, 2017). Discarded food contributes to greenhouse gas emissions, wasteful use of water, and depletion of natural resources (WWF, 2017). The Sustainable Spoonful can play a crucial role in minimizing this impact by facilitating the efficient distribution and utilization of surplus food, reducing overall waste (Lewis, 2022).

1. **Potential Benefits of The Sustainable Spoonful:**

Based on the research, the team has identified several benefits that The Sustainable Spoonful can provide:

1. ***Reduction of food wastage:***

The mobile application can help businesses minimize food wastage by connecting them with potential customers interested in purchasing surplus food products at discounted prices (FoodHero, 2020). This feature encourages responsible consumption and prevents edible food from ending up in landfills.

1. ***Affordability and Access:***

The Sustainable Spoonful can improve access to affordable and nutritious food for communities facing financial challenges. By offering discounted prices for surplus items, the mobile application enables individuals to access quality food at lower costs, promoting food security and reducing food inequality (Teshome, 2017).

1. ***Economic Savings:***

Businesses utilizing The Sustainable Spoonful can reduce financial losses associated with wasted food. The mobile application provides an avenue to sell surplus inventory, recover costs and increase profitability (Circular, 2022). This benefit incentivizes businesses to actively participate in the reduction of food wastage (Circular, 2022).

1. ***Sustainability and Environmental Consciousness:***

By facilitating the proper utilization of surplus food, The Sustainable Spoonful contributes to environmental sustainability (Jayadevan, 2022). The mobile application promotes responsible consumption practices and minimizes the environmental footprint associated with food production and waste disposal (Jayadevan, 2022).

Overall, the research findings highlight the urgent need for The Sustainable Spoonful and demonstrate its potential to address the issues of food wastage, financial losses for businesses, and the well-being of struggling communities in South Africa.

### Successes and Failures

During the research journey, the team encountered both successes and failures in achieving the aims and objectives of the research, as well as in solving the identified problems and subproblems. Below is an evaluation of the successes and failures:

***Successes:***

1. **Comprehensive Understanding** (Freid, 2020)**:**

One of the major successes was gaining a comprehensive understanding of the problem of food wastage and its impact on communities around South Africa. Through thorough research, the team gathered valuable insight into the causes, consequences, and potential solutions to address this issue.

1. **Solution Alignment:**

The research findings provided a solid foundation for aligning the aims and objectives of the research with the development of The Sustainable Spoonful. The team was able to establish how the mobile application can address the problem of food wastage, provide affordability, and access to nutritious food, and contribute to the sustainability of businesses and communities.

1. **Environmental Considerations** (Bouchrika, 2023)**:**

The team successfully integrated environmental considerations into the research and mobile application development process. By highlighting the environmental impact of food wastage, the team emphasized the importance of The Sustainable Spoonful in promoting responsible consumption and reducing the overall carbon footprint associated with food waste.

***Failures:***

1. **Limitations in Data Availability** (Brock-Utne, 2022)**:**

One of the challenges the team faced during the research process was the limited availability of data specific to South Africa’s food wastage problem. While the team gathered relevant information, there were instances where detailed local data on food wastage and its impact on communities was scarce. This limitation may have affected the depth of the analysis in certain areas.

1. **Resource constraints** (Balie, 2015)**:**

Throughout the research journey, the team faced resource constraints such as time and financial limitations, which impacted the extent of the team’s research and the development of The Sustainable Spoonful. These constraints may have limited the ability to explore certain aspects in greater detail or conduct extensive pilot studies to validate the mobile applications’ effectiveness.

1. **Ethical Considerations:**

The team recognized the ethical considerations associated with surplus food distribution. Ensuring food safety and hygiene standards, verifying expiry dates, and maintaining the quality of surplus food products are critical factors that need to be carefully addressed in the mobile applications’ implementation to avoid any potential harm to consumers.

To address the sub-problems identified, the following actions were taken:

1. **Check for legislation that could impede research** (Kron, 2020)**:**

* Conducted a thorough review of current legislation in South Africa related to food waste, expired food products, and donation of food.

1. **Establish interest amongst retail partners:**

* Reached out to retail stores, including supermarkets, cafes, and restaurants, to evaluate the stores’ willingness to partner with the food-saver mobile application.
* Addressed any concerns or questions raised by retail partners and clarified implementation procedures.
* Explored potential benefits for retailers, such as tax exemptions through the donation clause, to incentivize retailers’ participation and cooperation.

1. **Establish interest amongst consumers** (De Canio, et al., 2022)**:**

* Conducted surveys with target consumers to gauge interest and willingness to utilize the food-saver mobile application.
* Highlighted the mobile applications’ benefits, such as affordable access to better food prices, reduced food waste, and contribution to sustainability.
* Assessed consumer preferences, concerns, and suggestions to refine the mobile applications’ features and ensure alignment with user needs and expectations.

1. **Determine the experience of food retailers when it comes to food wastage** (Kor, et al., 2017)**:**

* Engaged with food retailers through interviews, surveys, or focus groups to gather insights into the retailers’ experiences and challenges related to food wastage.
* Explored factors contributing to food waste, such as inventory management issues, overproduction, or customer behaviour.
* Identified potential areas of improvement and solutions, such as better stock management practices, portion control, or education programs to reduce wastage.

### Challenges

During the research project aimed at addressing the problem of food wastage and developing The Sustainable Spoonful application, several challenges were encountered. These challenges can be categorized into the following areas:

1. **Research Challenges:**

* ***Limited Data Availability*** (Walden University, 2020)***:***

Obtaining comprehensive and up-to-date data on food wastage in South Africa posed a challenge. Accessing detailed information, such as specific figures on wastage levels, types of wasted food, and regional variations, was not always readily available. This limited the depth of analysis and required the team to rely on a combination of existing studies and reports.

* ***Legal Complexity*** (Chartier, 2023)***:***

Researching the current laws and regulations related to expired food products and donation processes proved challenging. Understanding and interpreting the legal framework required thorough examination. Complexities in the legal landscape, including varying regional regulations and liability concerns, added to the challenges faced.

1. **Stakeholder Engagement Challenges** (Lucidchart, 2023)**:**

* ***Retailer Buy-in:***

Convincing retail partners to embrace the food-saver mobile application and participate in the initiative posed a challenge. Some retailers may have been sceptical about the feasibility or profitability of the proposed solution. Addressing concerns, building trust, and demonstrating the potential benefits required persuasive communication and tailored discussions.

* ***Consumer Adoption*** (Maurer, 2022)***:***

Encouraging consumers to adopt the mobile application and change current behaviours presented a challenge. Convincing them to shift current purchasing habits, embrace discounted products, and use the mobile application regularly required effective marketing strategies, building awareness, and establishing trust in the platform.

1. **Group Dynamics Challenges:**

* ***Coordination and Communication:***

Ensuring effective coordination and communication among team members was crucial for the success of the project. Challenges in coordinating schedules, managing tasks, and maintaining consistent progress were encountered due to differences in availability, workload, and individual commitments. Regular meetings and utilizing project management tools helped mitigate these challenges.

* ***Decision-making and Conflict Resolution*** (Chron, 2021)***:***

As with any collaborative project, differing opinions and perspectives among team members led to occasional challenges in decision-making and conflict resolution. Ensuring open and respectful communication, active listening, and consensus-building techniques were employed to address these challenges and reach mutually agreeable solutions.

1. **Technical Challenges:**

* ***Mobile Application Development***(cubix, 2022)*:*

Developing The Sustainable Spoonful mobile application posed technical challenges, including selecting the appropriate technology stack, designing user-friendly interfaces, and ensuring smooth functionality across different devices and operating systems (Mohan, 2023). Overcoming these challenges required expertise in mobile application development and rigorous testing and iteration to deliver a reliable and user-friendly mobile application.

Despite these challenges, the team strived to overcome them through collaboration, adaptability, and perseverance.

### Benefits

1. **Benefits to Researchers:**

* ***Knowledge and Expertise*** (sabp, 2023)***:***

The research project provides an opportunity for researchers to deepen their understanding of the issue of food wastage, its impact on communities, and potential solutions. Researchers gain valuable expertise in conducting research, data analysis, and problem-solving within a real-world context.

* ***Skill Development:***

Researchers acquire and enhance skills in project management, data collection and analysis, stakeholder engagement, and communication. These skills can be transferable and valuable in future research endeavours or professional settings.

* ***Personal Fulfilment*** (Reference\*, 2020)***:***

Contributing to a project that aims to address a significant social and environmental issue can bring a sense of fulfilment and purpose to researchers, knowing that the researchers’ work has the potential to make a positive impact on society.

1. **Benefits to Academic Community Knowledge Base:**

* **Original Research Contribution** (Exel, 2020)**:**

The research project adds to the existing knowledge base by providing new insights into the problem of food wastage, its implications, and potential strategies for reduction.

* **Practical Application:**

The development of the mobile application contributes to the field of technology and innovation, showcasing a practical solution to mitigate food wastage and improve food affordability. It serves as a case study for future research and development efforts in similar areas.

* **Knowledge Sharing** (Artifacts, 2020)**:**

Findings from the research, including data analysis, best practices, and lessons learned, can be shared through academic publications, conferences, and forums. This fosters collaboration and knowledge exchange among researchers, policymakers, and practitioners working in related fields.

1. **Benefits to Customers:**

* **Improved food affordability** (turquoise, 2018)**:**

The mobile application will help provide the less fortunate citizens in South Africa with access to better food at more affordable prices. By partnering with retailers to offer discounted products nearing expiration dates, the mobile application helps customers save money and make healthier food choices.

* **Reduced food insecurity** (Ian, 2023)**:**

The mobile applications’ focus on minimizing food wastage contributes to reducing food insecurity in communities. By diverting surplus food from retail stores to those in need, the mobile application helps ensure that edible food reaches individuals and families who may struggle to afford nutritious meals.

* **Empowerment and Choice** (March, 2022)**:**

The mobile application empowers customers by giving them access to a wide range of food options at lower prices. It allows them to choose from discounted products that would otherwise go to waste, promoting economic inclusivity and reducing dependency on traditional food assistance programs.

## Conclusion

The research project focused on addressing the problem of food wastage and developing a mobile application to help provide less fortunate citizens of South Africa with access to affordable and better-quality food while helping shops minimize food wastage. Through comprehensive research, stakeholder engagement, and technical development, the project aimed to achieve its objectives and make a positive impact on the community.

The research findings shed light on the challenges and implications of food wastage in South Africa, emphasizing the financial and moral impacts on communities. By understanding the types of wasted foods, identifying areas with higher wastage rates, and determining retailer and consumer interest, the project laid the foundation for effective intervention.

The successful engagement with retail partners demonstrated the potential for collaboration in reducing food wastage, and efforts were made to address retailers’ concerns and explore incentives such as tax exemptions through the donation clause. Consumer interest was also evaluated, highlighting the importance of effective marketing strategies, and building trust in the mobile application.

Challenges were encountered throughout the research journey, including limited data availability, legal complexities, stakeholder engagement, group dynamics, and technical aspects. Despite these challenges, the team persevered, leveraging expertise, collaboration, and adaptability to overcome obstacles and move the project forward.

The research project has brought significant benefits to the researchers, enriching the researcher’s knowledge and expertise in the field of food waste and sustainable solutions. It contributes to the academic community’s knowledge base by providing new insights and practical applications for addressing food wastage. Furthermore, the development of The Sustainable Spoonful mobile application offers tangible benefits to customers, empowering them with affordable food options and reducing food insecurity.

In closing, the research project on food wastage and the development of the mobile application have made significant strides towards addressing the problem in South Africa. It is hoped that the mobile application will create a positive impact by minimizing food wastage, improving access to affordable food, and fostering sustainability. The projects’ outcomes and recommendations serve as a basis for future initiatives and interventions aimed at reducing food wastage and supporting communities in need.

## Recommendations

Based on the problem of food wastage and the development of the mobile application, the following recommendations are provided for customers:

1. **Immediate Extra Needs** (Doles, 2016)**:**

* **User Feedback and Improvement:** Encourage customers to provide feedback on their experience with the mobile application. This feedback will help identify areas for improvement and address any usability issues or technical glitches that may arise during the initial usage. Regular updates and bug fixes should be implemented to enhance the overall user experience.
* **Awareness and Promotion:** Customers can play a crucial role in spreading awareness about the mobile application’s benefits and its mission to reduce food wastage. Encourage users to share positive experiences on social media platforms and recommend the mobile application to friends and family. By increasing awareness and usage, more surplus food can be saved from going to waste and made available to those in need.

1. **Future Developments and Expansions** (Doles, 2016)**:**

* **Enhanced Features and Personalization:** Continually improve the mobile applications’ functionality by incorporating additional features that cater to user preferences. This could include personalized recommendations based on dietary preferences, notifications for expiring food items, and integration with loyalty programs or rewards for frequent users.
* **Collaboration with Additional Retailers and Organizations:** Expand partnerships with more retail stores, cafés, and restaurants to increase the availability of discounted food items through the mobile application. Additionally, establish collaborations with food banks, charities, and non-profit organizations to facilitate the donation of surplus food and further support vulnerable communities.
* **Integration of Sustainability Metrics:** Incorporate sustainability metrics within the mobile application to provide users with information about the environmental impact of food choices. This could include displaying the carbon footprint or water footprint associated with different products and encouraging users to make more sustainable choices.
* **Geographic Expansion:** The team would consider expanding the mobile applications’ availability to other regions within South Africa and potentially even to other countries facing similar food wastage challenges. This would enable a larger number of individuals to benefit from affordable food options while reducing food wastage on a broader scale.
* **Partnerships with Government and Policy Advocacy:** Collaborate with government agencies, policymakers, and organizations working on food security and sustainability initiatives. Advocate for policies that support the reduction of food wastage and incentivize participation from retail partners through tax exemptions or other incentives.

# Appendices

## Appendix A

## The Sustainable Spoonful User Manual

### Background

Food wastage is an issue in South Africa, and it worsens each day. The Sustainable Spoonful mobile application was developed to assist in mitigating food wastage amongst stores and consumers.

This user manual will serve as a guide to customers on how to access the mobile application, create an account, log into the mobile application, view stores, view discounted products, view their account, and log out of their account.

### Tools and Technologies Used

* **Java** – Used for the back end of the mobile application, exception-handling as well as communicating with the front-end and the local database.
* **SQL** – Used for storing and retrieving the data collected from the mobile application, such as a users’ account details or details about each store and their products.
* **Figma** – Used for creating the prototype designs for the different iterations of the mobile application.
* **XML** – Used for styling the different screens of the mobile application, such as the styling of the text, images, or buttons.
* **YouTrack** – Used for project management and to manage the Gantt Chart activities.
* **SQLiteDB**– Used for visualising the data that is stored in the database in the form of tables.
* **Grammarly** – Used for checking grammar and proofreading of the documents.
* **Google Forms** – Used for creating the test case scenarios as well as storing the results of each test case. Used to gather information when developing the initial project proposal.

### Accessing The Mobile Application

The steps to download and open the mobile application are as follows:

* Download the attached zip file.
* Extract the zip file.
* Open the extracted file in Android Studio.
* Create an emulator.
* Run the mobile application on the emulator.

### Creating an account

The steps to create an account are as follows:

* Open The Sustainable Spoonful mobile application.
* On the main page there will be a logo and two buttons called ‘Register’ and ‘Login.’ This is seen below:

A screen shot of a cell phone

Description automatically generated with medium confidence

Figure 0‑1: Main Page of The Sustainable Spoonful Mobile Application.

* Click on the ‘REGISTER’ button.
* A register form will be displayed, this is shown below:

A screenshot of a cell phone

Description automatically generated with medium confidence

Figure 0‑2: Registration Page of The Sustainable Spoonful Mobile Application.

* Fill in your details.
* At the bottom of the page click on the ‘CONFIRM’ button.
* Your account has now been created.
* Note: After creating an account, the mobile application will redirect the user to the login page.

### Logging into an account

The steps to log into the mobile application are as follows:

* Open The Sustainable Spoonful mobile application.
* On the main page there will be a logo and two buttons called ‘Register’ and ‘Login.’ This is seen below:

A screen shot of a cell phone

Description automatically generated with medium confidence

Figure 0‑3: Main Page of The Sustainable Spoonful Mobile Application.

* Click on the ‘LOGIN’ button.
* A login form will be displayed, this is shown below:

A screenshot of a cell phone

Description automatically generated with medium confidence

Figure 0‑4: Login Page of The Sustainable Spoonful Mobile Application.

* Fill in your details.
* At the bottom of the page click on the ‘CONFIRM’ button.
* You have now logged into the mobile application.
* Note: After logging in, the mobile application will redirect the user to the home page.

### The Home Page

The Home Page displays each of the stores that are partnered with The Sustainable Spoonful. Users can view more details about the stores such as what the store looks like, why they partnered with The Sustainable Spoonful, their location, and the products that they are offering.

The steps to view the details about each store are as follows:

* Log into the mobile application.
* On the Home Page, each of the stores are displayed, this is shown below:

A screen shot of a cell phone

Description automatically generated with medium confidence

Figure 0‑5: Home Page of The Sustainable Spoonful Mobile Application

* Click on each of the store images.
* The details of each store will be displayed, this is shown below:

A screenshot of a cell phone

Description automatically generated with medium confidence

Figure 0‑6: Store Details Page of The Sustainable Spoonful Mobile Application.

### Viewing and Selecting Discounts

The steps to view the discounts are as follows:

* Log into the mobile application.
* At the bottom of the page, click on the search icon.
* The discounts page will be displayed, this is shown below:

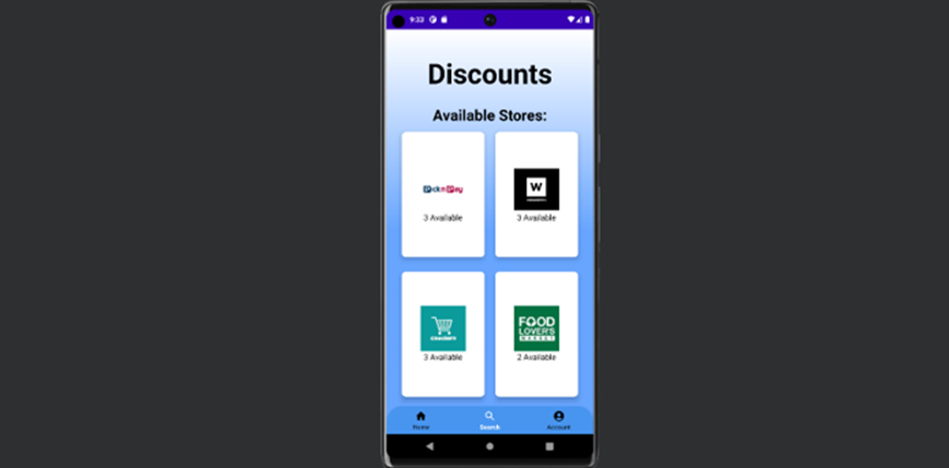


Figure 0‑7: Discounts Page of The Sustainable Spoonful Mobile Application.

* Click on one of the store images.
* The discounted products for that specific store will be displayed, an example is shown below:

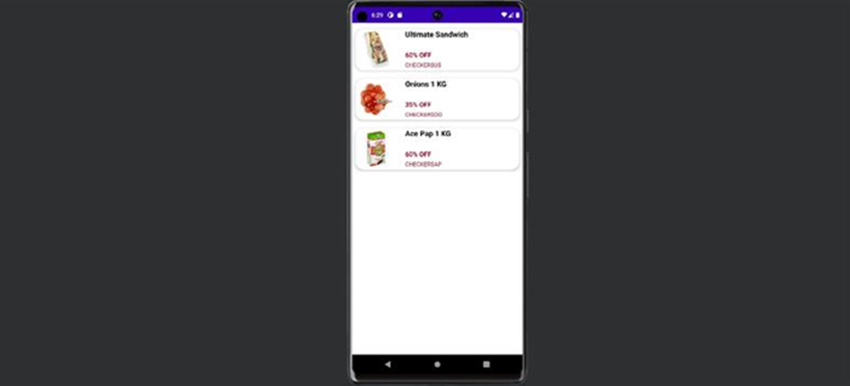
****

Figure 0‑8: Discounted Products Per Store Page of The Sustainable Spoonful Mobile Application.

* Select a product.
* A code will be displayed under the product.
* Take the code to the store and scan it at the till to receive the discount.

### The Account Page

The steps to view the account page are as follows:

* Log into the mobile application.
* At the bottom of the page, click on the account icon.
* The account page will be displayed, this is shown below:

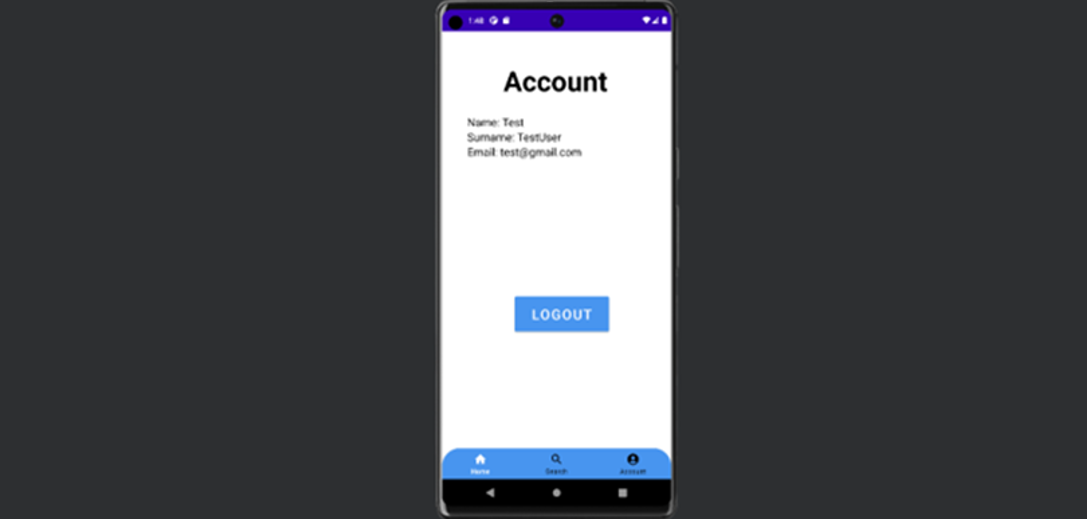


Figure 0‑9: Account Page of The Sustainable Spoonful Mobile Application.

* To log of the account, at the bottom of the page, click on the ‘LOGOUT’ button.
* You have now logged out of the account.

## Appendix B

### User Requirements Specification Template:

The document can be found here:

<https://eduvosonline-my.sharepoint.com/:b:/g/personal/y34vr5c17_vossie_net/EUoIZF2P5lxFj-21vClT6jkBiag1vzMTASB19rpeNloylQ?e=OyqfaX>

### User Requirements Specification Results:

The document can be found here:

<https://eduvosonline-my.sharepoint.com/:b:/g/personal/y34vr5c17_vossie_net/EadKFTOZu6VLki44t0ey1JwBzXjGIQSngny63lf9_Ihajw?e=gZcJwI>

## Appendix C

### Test Template:

The document can be found here:

<https://eduvosonline-my.sharepoint.com/:b:/g/personal/y34vr5c17_vossie_net/ESR5tmKwFMhHuy6qBn_hbwUBKhylbzh4HXV6GCkGb5zFjg?e=9dt2yO>

### Test Results:

The document can be found here:

<https://eduvosonline-my.sharepoint.com/:b:/g/personal/y34vr5c17_vossie_net/EQlKQ_TayvtJuZfUpgCmPSsB8Q9Rl3yG1tFbkI-3oJH5Ow?e=BLQxnV>

# List of Figures

[Figure 1‑1: Gantt Chart. 15](#_Toc138796089)

[Figure 1‑2: Work Breakdown Structure. 16](#_Toc138796090)

[Figure 2‑1: Mobile Application Development Process (Hall, R., 2018) 39](#_Toc138796091)

[Figure 2‑2: Feature-Driven Development (Mirzoyan, V., 2023). 39](#_Toc138796092)

[Figure 2‑3: Scrum Framework (Gren, J., 2021). 40](#_Toc138796093)

[Figure 2‑4: The Agile Development Process (Doherty, A., 2018) 41](#_Toc138796094)

[Figure 3‑1: Ages of Respondents. 46](#_Toc138796095)

[Figure 3‑2: Determining whether finding healthy and affordable food was difficult. 46](#_Toc138796096)

[Figure 3‑3: Determining whether food-saver mobile applications have been used before. 47](#_Toc138796097)

[Figure 3‑4: User Flow Diagram for The Sustainable Spoonful System (Cabrera, I., 2022). 48](#_Toc138796098)

[Figure 3‑5: Landing Page when the user opens the application for the first time. 49](#_Toc138796099)

[Figure 3‑6: Registration Page so that users can create an account. 50](#_Toc138796100)

[Figure 3‑7: Login Page so that users can login using the account they have just created. 50](#_Toc138796101)

[Figure 3‑8: Forgot Password page for when a user wishes to reset their password. 51](#_Toc138796102)

[Figure 3‑9: Confirmation screen that shows that a password reset email was sent to the user. 51](#_Toc138796103)

[Figure 3‑10: Home Page for the Sustainable Spoonful Mobile Application. 52](#_Toc138796104)

[Figure 3‑11: Discounted Products Page for users to search for discounted products based on the store and location. 52](#_Toc138796105)

[Figure 3‑12: Discounts listed for a specific store. 53](#_Toc138796106)

[Figure 3‑13: discount code for the selected discounted product that users can scan at the store to collect their item. 53](#_Toc138796107)

[Figure 3‑14: Account Page for a user to manage their account. 54](#_Toc138796108)

[Figure 3‑15: Prompting the user to log in first before they can edit their account details. 54](#_Toc138796109)

[Figure 3‑16: User logging in. 55](#_Toc138796110)

[Figure 3‑17: Edit Account Page. 55](#_Toc138796111)

[Figure 3‑18: Confirmation that the user has updated their account details. 56](#_Toc138796112)

[Figure 3‑19: Prompting the user to login before they delete their account. 56](#_Toc138796113)

[Figure 3‑20: Logging in. 57](#_Toc138796114)

[Figure 3‑21: Delete Account Page. 57](#_Toc138796115)

[Figure 3‑22: Confirmation that the user has deleted their account. 58](#_Toc138796116)

[Figure 3‑23: Prompting the user to log out of their account or go back. 58](#_Toc138796117)

[Figure 3‑24: Confirmation that the user has logged out of their account. 59](#_Toc138796118)

[Figure 3‑25: Colour Palette for The Sustainable Spoonful. 60](#_Toc138796119)

[Figure 3‑26: Logo for The Sustainable Spoonful. 60](#_Toc138796120)

[Figure 3‑27: Landing Page when the user opens the application for the first time. 61](#_Toc138796121)

[Figure 3‑28: Registration Page so that users can create an account. 61](#_Toc138796122)

[Figure 3‑29: Login Page so that users can login using the account they have just created. 62](#_Toc138796123)

[Figure 3‑30: Forgot Password page for when a user wishes to reset their password. 62](#_Toc138796124)

[Figure 3‑31: Confirmation screen that shows that a password reset email was sent to the user. 63](#_Toc138796125)

[Figure 3‑32: Home Page for the Sustainable Spoonful Mobile Application. 63](#_Toc138796126)

[Figure 3‑33: Discounted Products Page for users to search for discounted products based on the store and location. 64](#_Toc138796127)

[Figure 3‑34: Discounts listed for a specific store. 64](#_Toc138796128)

[Figure 3‑35: discount code for the selected discounted product that users can scan at the store to collect their item. 65](#_Toc138796129)

[Figure 3‑36: Account Page for a user to manage their account. 65](#_Toc138796130)

[Figure 3‑37: Prompting the user to log in first before they can edit their account details. 66](#_Toc138796131)

[Figure 3‑38: User logging in. 66](#_Toc138796132)

[Figure 3‑39: Edit Account Page. 67](#_Toc138796133)

[Figure 3‑40: Confirmation that the user has updated their account details. 67](#_Toc138796134)

[Figure 3‑41: Prompting the user to login before they delete their account. 68](#_Toc138796135)

[Figure 3‑42: Logging in. 68](#_Toc138796136)

[Figure 3‑43: Delete Account Page. 69](#_Toc138796137)

[Figure 3‑44: Confirmation that the user has deleted their account. 69](#_Toc138796138)

[Figure 3‑45: Prompting the user to log out of their account or go back. 70](#_Toc138796139)

[Figure 3‑46: Confirmation that the user has logged out of their account. 70](#_Toc138796140)

[Figure 3‑47: Landing Page when the user opens the application for the first time. 71](#_Toc138796141)

[Figure 3‑48: Registration Page so that users can create an account. 72](#_Toc138796142)

[Figure 3‑49: Login Page so that users can login using the account they have just created. 72](#_Toc138796143)

[Figure 3‑50: Forgot Password page for when a user wishes to reset their password. 73](#_Toc138796144)

[Figure 3‑51: Confirmation screen that shows that a password reset email was sent to the user. 73](#_Toc138796145)

[Figure 3‑52: Home Page for the Sustainable Spoonful Mobile Application. 74](#_Toc138796146)

[Figure 3‑53: Discounted Products Page for users to search for discounted products based on the store and location. 74](#_Toc138796147)

[Figure 3‑54: Discounts listed for a specific store. 75](#_Toc138796148)

[Figure 3‑55: discount code for the selected discounted product that users can scan at the store to collect their item. 75](#_Toc138796149)

[Figure 3‑56: Account Page for a user to manage their account. 76](#_Toc138796150)

[Figure 3‑57: Prompting the user to log in first before they can edit their account details. 76](#_Toc138796151)

[Figure 3‑58: User logging in. 77](#_Toc138796152)

[Figure 3‑59: Edit Account Page. 77](#_Toc138796153)

[Figure 3‑60: Confirmation that the user has updated their account details. 78](#_Toc138796154)

[Figure 3‑61: Prompting the user to login before they delete their account. 78](#_Toc138796155)

[Figure 3‑62: Logging in. 79](#_Toc138796156)

[Figure 3‑63: Delete Account Page. 79](#_Toc138796157)

[Figure 3‑64: Confirmation that the user has deleted their account. 80](#_Toc138796158)

[Figure 3‑65: Prompting the user to log out of their account or go back. 80](#_Toc138796159)

[Figure 3‑66: Confirmation that the user has logged out of their account. 81](#_Toc138796160)

[Figure 3‑67: Use Case Diagram – Iteration 1 for The Sustainable Spoonful System (Tilley & Rosenblatt, 2017). 82](#_Toc138796161)

[Figure 3‑68: Use Case Diagram – Iteration 2 for The Sustainable Spoonful System (Tilley & Rosenblatt, 2017). 84](#_Toc138796162)

[Figure 3‑69: Sequence Diagram – Iteration 1 for The Sustainable Spoonful System (Lucidchart, 2022). 85](#_Toc138796163)

[Figure 3‑70: Sequence Diagram – Iteration 2 for The Sustainable Spoonful System (Lucidchart, 2022) 86](#_Toc138796164)

[Figure 3‑71: Functional Decomposition Diagram - Iteration 1 for The Sustainable Spoonful System (Tilley & Rosenblatt, 2017). 92](#_Toc138796165)

[Figure 3‑72: Functional Decomposition Diagram - Iteration 2 for The Sustainable Spoonful System (Tilley & Rosenblatt, 2017). 93](#_Toc138796166)

[Figure 3‑73: Context Diagram - Iteration 1 for The Sustainable Spoonful System (Pedriquez, D., 2022). 101](#_Toc138796167)

[Figure 3‑74: Context Diagram - Iteration 2 for The Sustainable Spoonful System (Pedriquez, D., 2022). 102](#_Toc138796168)

[Figure 3‑75: Entity Relationship Diagram - Iteration 1 for The Sustainable Spoonful System (Brumm, B., 2022). 103](#_Toc138796169)

[Figure 3‑76: Entity Relationship Diagram - Iteration 2 for The Sustainable Spoonful System (Brumm, B., 2022). 103](#_Toc138796170)

[Figure 3‑77: Entity Relationship Diagram - Iteration 3 for The Sustainable Spoonful System (Brumm, B., 2022). 104](#_Toc138796171)

[Figure 3‑78: Data Flow Diagram - Iteration 1 for The Sustainable Spoonful System (Lucidchart, 2023). 104](#_Toc138796172)

[Figure 3‑79: Data Flow Diagram - Iteration 2 for The Sustainable Spoonful System (Lucidchart, 2023). 105](#_Toc138796173)

[Figure 4‑1: Test Case 1 - Results Graph. 115](#_Toc138796174)

[Figure 4‑2: Test Case 2 – Results Graph. 115](#_Toc138796175)

[Figure 4‑3: Test Case 3 – Result Graph. 116](#_Toc138796176)

[Figure 4‑4: Test Case 4 – Results Graph. 116](#_Toc138796177)

[Figure 4‑5: Test Case 5 – Results Graph. 117](#_Toc138796178)

[Figure 4‑6: Was everything readable? - Result Graph. 117](#_Toc138796179)

[Figure 4‑7: Was the application responsive? - Result Graph. 118](#_Toc138796180)

[Figure 4‑8: Was the application easy to navigate? – Result Graph. 118](#_Toc138796181)

[Figure 4‑9: Main screen when the user opens the mobile application for the first time. 119](#_Toc138796182)

[Figure 4‑10: Main screen when the user opens the mobile application for the first time. The mobile application is running on the Emulator Pixel 6 Pro API 30. 120](#_Toc138796183)

[Figure 4‑11: Registration screen for users to create an account. 122](#_Toc138796184)

[Figure 4‑12: Registration screen for users to create an account. The mobile application is running on the Emulator Pixel 6 Pro API 30. 123](#_Toc138796185)

[Figure 4‑13: Login Screen so that users can login using their existing credentials. 127](#_Toc138796186)

[Figure 4‑14: Login Screen so that users can login using their existing credentials. The mobile application is running on the Emulator Pixel 6 Pro API 30. 128](#_Toc138796187)

[Figure 4‑15: Landing/home screen for the Sustainable Spoonful Mobile Application. 132](#_Toc138796188)

[Figure 4‑16: Landing/home screen for the Sustainable Spoonful Mobile Application. The mobile application is running on the Emulator Pixel 6 Pro API 30. 133](#_Toc138796189)

[Figure 4‑17: - Discounts Screen for users to search for discounted products based on the store and location. 136](#_Toc138796190)

[Figure 4‑18: Discounts Screen for users to search for discounted products based on the store and location. The mobile application is running on the Emulator Pixel 6 Pro API 30. 137](#_Toc138796191)

[Figure 4‑19: Discounted products screen for each store. 142](#_Toc138796192)

[Figure 4‑20: Discounted products screen for each store. The mobile application is running on the Emulator Pixel 6 Pro API 30. 143](#_Toc138796193)

[Figure 4‑21: Account screen. 145](#_Toc138796194)

[Figure 4‑22: Account screen. The mobile application is running on the Emulator Pixel 6 Pro API 30. 146](#_Toc138796195)

[Figure 0‑1: Main Page of The Sustainable Spoonful Mobile Application. 180](#_Toc138796196)

[Figure 0‑2: Registration Page of The Sustainable Spoonful Mobile Application. 181](#_Toc138796197)

[Figure 0‑3: Main Page of The Sustainable Spoonful Mobile Application. 182](#_Toc138796198)

[Figure 0‑4: Login Page of The Sustainable Spoonful Mobile Application. 183](#_Toc138796199)

[Figure 0‑5: Home Page of The Sustainable Spoonful Mobile Application 184](#_Toc138796200)

[Figure 0‑6: Store Details Page of The Sustainable Spoonful Mobile Application. 185](#_Toc138796201)

[Figure 0‑7: Discounts Page of The Sustainable Spoonful Mobile Application. 186](#_Toc138796202)

[Figure 0‑8: Discounted Products Per Store Page of The Sustainable Spoonful Mobile Application. 187](#_Toc138796203)

[Figure 0‑9: Account Page of The Sustainable Spoonful Mobile Application. 188](#_Toc138796204)

# List of Tables

[Table 1.1: Reasonable Timelines. 15](#_Toc138796205)

[Table 1.3: Proposed budget (Thomas, R., 2022). 18](#_Toc138796206)

[Table 4.1: Roles and responsibilities of the development team. 110](#_Toc138796207)

[Table 4.2: Testing Schedule and Report. 114](#_Toc138796208)

# References

6sense, 2023. *Flashfood Company Insights, Tech Stack, and Competitors | 6sense*. [Online] 6sense.com. Available at: [https://6sense.com/company/flashfood/5c3b0112d55ae49f1b775694](https://6sense.com/company/flashfood/5c3b0112d55ae49f1b775694%20) [Accessed 20 March 2023].

AJE, 2022. *Scope and Delimitations in Research | AJE.* [Online] Available at: <https://www.aje.com/arc/scope-and-delimitations-in-research/> [Accessed 13 April 2023].

AllCodingTutorials, 2022. *RecyclerView with SQLite Database | Populate RecyclerView with SQLite Database | Source Code.* [Video] Available at: [https://youtu.be/6enCz2cY18c](https://youtu.be/6enCz2cY18c%20) [Accessed 21 June 2023].

Anderson, J. & Poole, M., 2015. *Assignment and Thesis Writing. South African 4th Edition*. Juta.

Android Coding, 2019. *How to Implement Bottom Navigation With Activities in Android Studio | BottomNav | Android Coding*. [Video] Available at: [https://youtu.be/JjfSjMs0ImQ](https://youtu.be/JjfSjMs0ImQ%20) [Accessed 21 June 2023].

Android Developers, 2018. *Android Developers.* [Online] Available at: [https://developer.android.com/](https://developer.android.com/%20) [Accessed 14 April 2023].

Android Developers, 2023. *App security best practices*. [Online] Available at: <https://developer.android.com/topic/security/best-practices> [Accessed 22 May 2023].

Android Developers, 2023. *Save data using SQLite*. [Online] Available at: [https://developer.android.com/training/data-storage/sqlite#java](https://developer.android.com/training/data-storage/sqlite%23java%20) [Accessed 21 June 2023].

Artifacts, 2020. *Why Is Academic Research Important? Your Complete Guide*. [Online] Available at: <https://artifacts.ai/why-is-academic-research-important/> [Accessed 18 June 2023].

Athuraliya, A., 2022. *Sequence Diagram Tutorial – Complete Guide with Examples | Creately.* [Online] Available at: <https://creately.com/guides/sequence-diagram-tutorial/> [Accessed 23 May 2023].

Available at: [https://zarecruitment.com/how-much-does-it-cost-to-build-an-app-in-south-africa/](https://zarecruitment.com/how-much-does-it-cost-to-build-an-app-in-south-africa/%20) [Accessed 20 March 2023].

Averda, 2022. *South Africa wastes 10-million tonnes of food every year.* [Online] Available at: <https://www.averda.com/rsa/news/south-africa-wastes-10-million-tonnes-of-food-every-year> [Accessed 23 June 2023].

Balie, F., 2015. *The Graduate Research Experience: Successes and Failures*. [Online] Available at: <https://link.springer.com/book/10.1007/978-1-4939-2516-2> [Accessed 22 June 2023].

Bega, S., 2021. *South Africa wastes 10mn tons of food a year*. [Online] The Mail & Guardian. Available at: [https://mg.co.za/environment/2021-08-21-south-africa-wastes-10-million-tons-of-food-a-year/#:~:text=South%20Africa%20wastes%2010mn%20tons%20of%20food%20a](https://mg.co.za/environment/2021-08-21-south-africa-wastes-10-million-tons-of-food-a-year/%23:~:text=South%20Africa%20wastes%2010mn%20tons%20of%20food%20a%20) [Accessed 12 March 2023].

Bell, R., 2012. *Reducing food waste has economic, environmental and social benefits*. [Online] Available at: <https://www.canr.msu.edu/news/reducing_food_waste_has_economic_environmental_and_social_benefits> [Accessed 12 March 2023].

Bouchrika, I., 2023. *Top 10 Qualities of Good Academic Research*. [Online] Available at: [https://research.com/research/top-10-qualities-of-good-academic-research#:~:text=Good%20research%20is%20replicable%2C%20reproducible,to%20test%20the%20study's%20findings.](https://research.com/research/top-10-qualities-of-good-academic-research%23:~:text=Good%20research%20is%20replicable%2C%20reproducible,to%20test%20the%20study's%20findings.) [Accessed 18 June 2023].

Brock-Utne, J. G., 2022. *Case Studies of Successes and Failures*. [Online] Available at: <https://link.springer.com/book/10.1007/978-1-4939-2516-2> [Accessed 22 June 2023].

Brumm, B., 2022. *A Guide to the Entity Relationship Diagram (ERD).* [Online] Available at: <https://www.databasestar.com/entity-relationship-diagram/> [Accessed 22 May 2023].

BTech Days, 2021. *Login and Register Form using SQLite Database in Android Studio | Login Registration Android Studio*. [Video] Available at: <https://youtu.be/o9CVZ1gQgQo> [Accessed 21 June 2023].

Cabrera, I., 2022. *What is a User Flow Diagram and How to Create One?* [Online] Available at: <https://venngage.com/blog/user-flow-diagram/> [Accessed 21 May 2023].

Caulfield, J., 2022. *Writing a Research Paper Conclusion | Step-by-Step Guide*. [Online] Available at: <https://www.scribbr.com/research-paper/research-paper-conclusion/> [Accessed 19 June 2023].

cgcsa, 2020. *Food Loss And Waste*. [Online] Available at: <https://www.cgcsa.co.za/service-offering/food-safety-initiative/food-loss-and-waste/> [Accessed 23 June 2023].

Chartier, M., 2023. *8 Challenges Faced by Researchers (and Tips to Help).* [Online] Available at: <https://fourwaves.com/blog/research-challenges/> [Accessed 22 June 2023].

Chen, C., 2022. *I get produce delivered through Imperfect Foods, which sells the ‘ugly’ fruits and veggies that supermarkets won’t take — and it’s way cheaper than shopping at the store*. [Online] Insider. Available at: [https://www.insider.com/guides/kitchen/imperfect-produce-review#:~:text=Imperfect%20Foods%20delivers%20seasonal%2C%20cosmetically](https://www.insider.com/guides/kitchen/imperfect-produce-review%23:~:text=Imperfect%20Foods%20delivers%20seasonal%2C%20cosmetically%20) [Accessed 21 March 2023].

Chron, 2021. Challenges in Team Development. [Online] Available at: <https://smallbusiness.chron.com/challenges-team-development-39606.html> [Accessed 18 June 2023].

Circular, 2022. The financial and environmental impact of food waste. [Online] Available at: <https://www.circularonline.co.uk/opinions/the-financial-and-environmental-consequences-of-food-waste/> [Accessed 18 June 2023].

Coding Adventure, 2020. *CardView UI Design Android Studio | Using Grid Layout*. [Video] Available at: [https://youtu.be/YKssd\_9x8Eg](https://youtu.be/YKssd_9x8Eg%20) [Accessed 21 June 2023].

Coding With Me, 2021. *LOGIN APP IN ANDROID STUDIO USING SQLite | SQLite DATABASE | LOGIN AND REGISTER APP WITH FULL CODING*. [Video] Available at: [https://youtu.be/yJ02XTKiuAc](https://youtu.be/yJ02XTKiuAc%20) [Accessed 11 March 2022].

CodingWithSaani, 2019. *2- Save and Retrieve Images with SQLite in Android Studio 2019 | Getting Image from Device*. [Video] Available at: [https://youtu.be/9bUkyop6eMg](https://youtu.be/9bUkyop6eMg%20) [Accessed 21 June 2023].

CodingZest, 2021. *Modern Dashboard UI Design Android Studio Tutorial | Cardview Android Studio | Grid layout Android*. [Video] Available at: [https://youtu.be/ixRXEoGAEZM](https://youtu.be/ixRXEoGAEZM%20) [Accessed 21 June 2023].

Computer Hope, 2017. *What is Exception Handling?* [Online] Available at: <https://www.computerhope.com/jargon/e/exception-handling.htm> [Accessed 22 May 2023].

cubix, 2022. 13 *Top Mobile App Development Challenges in 2022*. [Online] Available at: <https://www.cubix.co/blog/mobile-app-development-challenges> [Accessed 19 June 2023].

De Canio, F., Fuentes-Blasco, M. & Martinelli, E., 2022. *Engaging shoppers through mobile apps: the role of gamification*. [Online] Available at: <https://www.emerald.com/insight/content/doi/10.1108/IJRDM-09-2020-0360/full/html#:~:text=Gamification%20influences%20the%20customer%20interaction%20with%20the%20interface,Hamari%20et%20al.%2C%202014%3B%20Yang%20et%20al.%2C%202017%29>. [Accessed 18 June 2023].

Deccon Tech, 2022. *How to insert image and retrieve in sqlite database Android Studio*. [Video] Available at: [https://youtu.be/wGBhi9Vf7yA](https://youtu.be/wGBhi9Vf7yA%20) [Accessed 21 June 2023].

DesignRush, 2023. *What Is User Authentication?* [Online] Available at: <https://www.designrush.com/agency/cybersecurity/trends/user-authentication> [Accessed 22 May 2023].

designshack, 2023. *What Is a Wireframe & How to Design Them: A Beginner’s Guide*. [Online] Available at: <https://designshack.net/articles/graphics/what-is-a-wireframe/> [Accessed 22 May 2023].

DiscoverPhDs, 2020. *Scope and Delimitations - Explained & Example*. [Online] DiscoverPhDs. Available at: [https://www.discoverphds.com/blog/scope-and-delimitations](https://www.discoverphds.com/blog/scope-and-delimitations%20) [Accessed 11 March 2023].

Doherty, A., 2018. *Agile Development at Primetics*. [Online] Primetics. Available at: <https://primetics.co.uk/agile-development-at-primetics/> [Accessed 16 April 2023].

Doles, B., 2016. *5 Ways to Improve Your App’s User Experience*. [Online] Available at: <https://www.tune.com/blog/5-ways-to-improve-app-user-experience/> [Accessed 19 June 2023].

draw.io, 2023. *draw.io*. [Online] Available at: <https://www.drawio.com/> [Accessed 22 May 2023].

Dunson, J., 2021. *Food Waste in Schools and Strategies to Reduce It*. [Online] Available at: <https://extension.sdstate.edu/food-waste-schools-and-strategies-reduce-it> [Accessed 12 March 2022].

EatSleepBreatheFI, 2021. *My Detailed Flashfood Review (Groceries for 50-70% Off) | Eat Sleep Breathe FI.* [Online] Available at: [https://eatsleepbreathefi.com/flash-food-review/#:~:text=You](https://eatsleepbreathefi.com/flash-food-review/%23:~:text=You%20) [Accessed 20 March 2023].

Ellis, M., 2023. *How to Write a Research Paper Conclusion*. [Online] Available at: <https://www.grammarly.com/blog/research-paper-conclusion/> [Accessed 19 June 2023].

EPA, 2021. *International Efforts on Wasted Food Recovery*. [Online] Available at: <https://www.epa.gov/international-cooperation/international-efforts-wasted-food-recovery> [Accessed 22 June 2023].

EPA, 2022. *Food Waste Research | US EPA*. [Online] Available at: <https://www.epa.gov/land-research/food-waste-research> [Accessed 13 April 2023].

Esferasoft Solutions, 2021. *Why Agile Methodology is Essential for Mobile Application Development?* [Online] Available at: <https://www.esferasoft.com/blog/why-agile-methodology-is-essential-for-mobile-application-development/> [Accessed 14 April 2023].

Exel, A., 2020. *Benefits of Engaging in Research*. [Online] Available at: <https://www.sandiego.edu/ugresearch/about/benefits.php> [Accessed 20 June 2023].

FASA., 2019. *Tackling South Africa’s food waste problem*. [Online] FASA Franchise Association of South Africa. Available at: [https://www.fasa.co.za/tackling-south-africas-food-waste-problem/](https://www.fasa.co.za/tackling-south-africas-food-waste-problem/%20) [Accessed 12 March 2023].

Figma, 2023. *Figma*. [Online] Available at: <https://www.figma.com/community/> [Accessed 22 May 2023].

Flashfood, 2021. *Flashfood - Apps on Google Play.* [Online] Available at: [https://play.google.com/store/apps/details?id=com.flashfoodapp.android&hl=en&gl=US&pli=1](https://play.google.com/store/apps/details?id=com.flashfoodapp.android&hl=en&gl=US&pli=1%20) [Accessed 14 April 2023].

Flashfood, 2022. *Download Flashfood | Save money and fight against food waste*. [Online] Available at: <https://www.flashfood.com/en/download> [Accessed 14 April 2023].

FoodHero, 2020. *By the Numbers: The economic impact of food waste*. [Online] Available at: <https://foodhero.com/blogs/economic-impact-of-food-waste> [Accessed 17 June 2023].

Fowler, M., 2017. *Martin Fowler*. [Online] Available at: <https://martinfowler.com/> [Accessed 16 June 2023].

Fredrick,T., 2022. *Why should we care about food waste?* [Online] Available at: [https://www.usda.gov/foodlossandwaste/why](https://www.usda.gov/foodlossandwaste/why%20) [Accessed 12 March 2023].

Freid, A., 2020. 21 *Reasons Research Fails (and how to overcome these challenges).* [Online] Available at: <https://www.curiositytank.com/blog/21-reasons-research-fails-and-how-to-overcome-these-challenges> [Accessed 19 6 2023].

GeeksForGeeks, 2019. *Software Testing | Functional Testing.* [Online] Available at: [https://www.geeksforgeeks.org/software-testing-functional-testing/](https://www.geeksforgeeks.org/software-testing-functional-testing/%20) [Accessed 21 June 2023].

GeeksforGeeks, 2021. *Android SDK and it’s Components.* [Online] Available at: [https://www.geeksforgeeks.org/android-sdk-and-its-components/](https://www.geeksforgeeks.org/android-sdk-and-its-components/%20) [Accessed 14 April 2023].

Gillis, A., 2023. *What is User Acceptance Testing (UAT)?* [Online] Available at: [https://www.techtarget.com/searchsoftwarequality/definition/user-acceptance-testing-UAT](https://www.techtarget.com/searchsoftwarequality/definition/user-acceptance-testing-UAT%20) [Accessed 21 June 2023].

GitHub, 2022. *GitHub: Let’s build from here*. [Online] Available at: [https://github.com/](https://github.com/%20) [Accessed 13 April 2023].

Greb, C., 2017. *Input validation*. [Online] Available at: <https://medium.com/android-testing-daily/input-validation-7616fd010112> [Accessed 19 May 2023].

Gren, J., 2021. *Scrum Methodology: An Agile Framework for Project Management*. [Online] Available at: <https://startinfinity.com/project-management-methodologies/scrum> [Accessed 10 April 2023].

Hajjdiab, H., Anzer, A., Tabaza, H. A. & Ahmed, W., 2021. *A Food Wastage Reduction Mobile Application.* [Online] Available at: <https://ieeexplore.ieee.org/document/8488190> [Accessed 19 June 2023].

Hall, R., 2018. *How To Prototype A Mobile App And Leave A Lasting Impression On Investors - MindSea.* [Online] Available at: [https://mindsea.com/how-to-prototype-a-mobile-app/](https://mindsea.com/how-to-prototype-a-mobile-app/%20) [Accessed 15 April 2023].

Hamilton, T., 2019. *Unit Testing Tutorial: What is, Types, Tools, EXAMPLE*. [Online] Available at: <https://www.guru99.com/unit-testing-guide.html> [Accessed 17 June 2023].

Harvard T.H. Chan, 2017. *Food Waste*. [Online] The Nutrition Source. Available at: [https://www.hsph.harvard.edu/nutritionsource/sustainability/food-waste/](https://www.hsph.harvard.edu/nutritionsource/sustainability/food-waste/%20) [Accessed 13 April 2023].

Hautzinger, D., 2022. *App prevents food waste- And Gives Customers a Deal on Groceries, Too* [Online] Available at: [https://interactive.wttw.com/playlist/2022/10/04/flashfood](https://interactive.wttw.com/playlist/2022/10/04/flashfood%20) [Accessed 13 March 2023].

Ian, E., 2023. *Importance of Academic Research.* [Online] Available at: <https://www.allassignmenthelp.com/blog/importance-of-academic-research/> [Accessed 19 June 2023].

IBM, 2022. *Server-side validation*. [Online] Available at: <https://www.ibm.com/docs/en/mam/7.6.1?topic=validation-server-side> [Accessed 22 May 2023].

IBM, 2023. *Overview of data validation*. [Online] Available at: <https://www.ibm.com/docs/en/order-management?topic=application-data-validation> [Accessed 22 May 2023].

Imperfect Foods, 2021. *Imperfect Foods - Apps on Google Play*. [Online] Available at: [https://play.google.com/store/apps/details?id=com.imperfectfoods&hl=en&gl=US](https://play.google.com/store/apps/details?id=com.imperfectfoods&hl=en&gl=US%20) [Accessed 14 April 2023].

Imperfect Foods, 2023. *Grocery Delivery for Organic Food, Fresh Produce & More*. [Online] Available at: [https://www.imperfectfoods.com/how-it-works](https://www.imperfectfoods.com/how-it-works%20) [Accessed 14 April 2023].

Indeed Editorial Team., 2023. *Research Objectives: Definition and How to Write Them*. [Online] Indeed Career Guide. Available at: [https://www.indeed.com/career-advice/career-development/research-objectives](https://www.indeed.com/career-advice/career-development/research-objectives%20) [Accessed 18 March 2023].

IPC., 2021. South Africa: *Impact of Covid-19 on Food security*. [Online] Integrated Food Security Phase Classification. Available at: <https://www.ipcinfo.org/ipc-country-analysis/details-map/en/c/1153024/?iso3=ZAF> [Accessed 12 March 2023].

Java.com, 2023. *What is Java technology and why do I need it?* [Online] Available at: [https://www.java.com/en/download/help/whatis\_java.html](https://www.java.com/en/download/help/whatis_java.html%20) [Accessed 14 April 2023].

JavaTPoint, 2011. *Functional Testing - javatpoint*. [Online] Available at: <https://www.javatpoint.com/functional-testing> [Accessed 21 June 2023].

JavaTPoint, 2021. *Performance Testing - javatpoint*. [Online] Available at: [https://www.javatpoint.com/performance-testing](https://www.javatpoint.com/performance-testing%20) [Accessed 21 June 2023].

Jayadevan, C., 2022. *Impacts of food wastage on economic growth*. [Online] Available at: <https://onlinelibrary.wiley.com/doi/10.1002/wfp2.12038> [Accessed 18 June 2023].

JetBrains, 2022. *YouTrack: Project management for all your teams*. [Online] Available at: [https://www.jetbrains.com/youtrack/?source=google&medium=cpc&campaign=10593512321&term=youtrack&content=632590181257&gclid=CjwKCAjw0N6hBhAUEiwAXab-Ta7YPxmYkoDBbxoU3l\_rsg\_cdHYJnpTjx-1QrxkeYFWHvY3FCqeiCxoC\_HYQAvD\_BwE](https://www.jetbrains.com/youtrack/?source=google&medium=cpc&campaign=10593512321&term=youtrack&content=632590181257&gclid=CjwKCAjw0N6hBhAUEiwAXab-Ta7YPxmYkoDBbxoU3l_rsg_cdHYJnpTjx-1QrxkeYFWHvY3FCqeiCxoC_HYQAvD_BwE%20) [Accessed 14 April 2023].

JRebel, 2023. *How to Use Java Integration Testing | JRebel & XRebel by Perforce*. [Online] Available at: <https://www.jrebel.com/blog/how-to-use-java-integration-testing> [Accessed 21 June 2023].

Kazi, N., 2021. *Logging and remote error reporting in mobile apps.* [Online] Available at: <https://blog.logrocket.com/logging-and-remote-error-reporting-in-mobile-apps/> [Accessed 19 May 2023].

King, M., 2021. *5 tips on how to write a research aim*. [Online] Available at: [https://timesofindia.indiatimes.com/readersblog/mabast-king/5-tips-on-how-to-write-a-research-aim-36047/](https://timesofindia.indiatimes.com/readersblog/mabast-king/5-tips-on-how-to-write-a-research-aim-36047/%20) [Accessed 9 March 2023].

Kor, Y. Y., Prabhu, J. & Esposito, M., 2017. *How Large Food Retailers Can Help Solve the Food Waste Crisis.* [Online] Available at: <https://hbr.org/2017/12/how-large-food-retailers-can-help-solve-the-food-waste-crisis#:~:text=How%20Large%20Food%20Retailers%20Can%20Help%20Solve%20the,...%204%20%234.%20Team%20up%20with%20consumers.%20> [Accessed 18 June 2023].

Kron, M., 2020. *How do you evaluate and learn from your research failures and successes?* [Online] Available at: <https://www.linkedin.com/advice/3/how-do-you-evaluate-learn-from-your-research-failures> [Accessed 19 June 2023].

Lemon Soft, 2022. *How to Implement Bottom Navigation With Activities in Android Studio | BottomNav | Lemon Soft*. [Video] Available at: <https://youtu.be/y4arA4hsok8> [Accessed 21 June 2023].

Lewis, J., 2022. *How Does Food Waste Affect the Environment?* [Online] Available at: <https://earth.org/how-does-food-waste-affect-the-environment/> [Accessed 23 June 2023].

Ling, K., 2020. *What is Feature Driven Development (FDD)? | Definition*. [Online] Available at: <https://www.productplan.com/glossary/feature-driven-development/> [Accessed 14 April 2023].

Loshin, P., 2022. *What is Encryption and How Does it Work?* [Online] Available at: <https://www.techtarget.com/searchsecurity/definition/encryption> [Accessed 22 May 2023].

Love Food Hate Waste, 2019. *Love Food Hate Waste*. [Online] Available at: [https://www.lovefoodhatewaste.com/](https://www.lovefoodhatewaste.com/%20) [Accessed 14 April 2023].

Lucidchart, 2020. *Context Diagram Software.* [Online] Available at: <https://www.lucidchart.com/pages/examples/context-diagram-software#:~:text=A%20context%20diagram%2C%20also%20known,data%20within%20a%20technical%20system> [Accessed 22 May 2023].

Lucidchart, 2022. *UML Sequence Diagram Tutorial.* [Online] Available at: <https://www.lucidchart.com/pages/uml-sequence-diagram> [Accessed 22 May 2023].

Lucidchart, 2023. *How to manage common stakeholder issues and challenges*. [Online] Available at: <https://www.lucidchart.com/blog/how-to-manage-common-stakeholder-issues-and-challenges> [Accessed 19 June 2023].

Lucidchart, 2023. *What is a Data Flow Diagram*. [Online] Available at: <https://www.lucidchart.com/pages/data-flow-diagram> [Accessed 22 May 2023].

Lucidchart, 2023. *What is an Entity Relationship Diagram (ERD)?* [Online] Available at: <https://www.lucidchart.com/pages/er-diagrams> [Accessed 22 May 2023].

Mandaha, D., 2021. *45% OF AVAILABLE FOOD SUPPLY IN SOUTH AFRICA WASTED, SHOWS NEW CSIR STUDY*. [Online] Available at: <https://www.csir.co.za/food-supply-south-africa-wasted-shows-new-csir-study> [Accessed 23 June 2023].

March, L., 2022. *11 Benefits of Market Research + Free Templates*. [Online] Available at: <https://www.similarweb.com/blog/research/market-research/benefits-of-market-research/> [Accessed 19 June 2023].

Marjot, C., 2021. *10 Steps for Beta App testing of your Mobile App Testing process*. [Online] Available at: <https://blog.mobcoder.com/mobile-app-development-process/> [Accessed 19 June 2023].

Martin, J. A., 2019. *What is access control? A key component of data security*. [Online] Available at: <https://www.csoonline.com/article/3251714/what-is-access-control-a-key-component-of-data-security.html> [Accessed May 2023 2023].

Maurer, M., 2022. *Understanding the Influence and Impact of Stakeholder Engagement in Patient-centered Outcomes Research: a Qualitative Study*. [Online] Available at: <https://link.springer.com/article/10.1007/s11606-021-07104-w> [Accessed 22 June 2023].

McCombes, S. and George, T., 2022. *What Is a Research Methodology? | Steps & Tips*. [Online] Available at: [https://www.scribbr.com/dissertation/methodology/](https://www.scribbr.com/dissertation/methodology/%20) [Accessed 11 March 2023].

McQuillan, R., 2022. *How to Create a Data Model in 9 Steps*. [Online] Available at: [https://budibase.com/blog/data/how-to-create-a-data-model/](https://budibase.com/blog/data/how-to-create-a-data-model/%20) [Accessed 23 May 2023].

Mirzoyan, V., 2023. *Why should you use a feature-driven development?* [Online] Available at: [https://aist.global/en/use-a-feature-driven-development](https://aist.global/en/use-a-feature-driven-development%20) [Accessed 14 April 2023].

Mohan, U., 2023. *Top 10 Challenges Faced in Mobile Application Development*. [Online] Available at: <https://www.sayonetech.com/blog/challenges-faced-mobile-application-development/> [Accessed 19 June 2023].

Nalawade, K., 2023. *How to Write Unit Tests in Java*. [Online] Available at: [https://www.freecodecamp.org/news/java-unit-testing/#:~:text=What%20is%20Unit%20Testing%3F](https://www.freecodecamp.org/news/java-unit-testing/%23:~:text=What%20is%20Unit%20Testing%3F) [Accessed 21 June 2023].

Ng, C., 2021. *What is data integrity and how can you maintain it?* [Online] Available at: <https://www.varonis.com/blog/data-integrity> [Accessed 16 June 2023].

nimble, 2022. *What Is Scrum Methodology? & Scrum Project Management*. [Online] Available at: [https://www.nimblework.com/agile/scrum-methodology/](https://www.nimblework.com/agile/scrum-methodology/%20) [Accessed 14 April 2023].

Olio, 2023. *Olio - Your Local Sharing App*. [Online] Available at: [https://olioapp.com/en/](https://olioapp.com/en/%20) [Accessed 14 April 2023].

Optimizely, 2023. *Usability testing*. [Online] Available at: <https://www.optimizely.com/optimization-glossary/usability-testing/#:~:text=The%20goal%20of%20usability%20testing> [Accessed 21 June 2023].

Orbitum Labs, 2023. *A step-by-step guide on how to conduct user research for different phases of your mobile app development*. [Online] Available at: <https://www.orbitumlabs.com/blog/user-research> [Accessed 14 April 2023].

Ortega, D., 2019. *Which User Research Method Is Best for Mobile Apps?* [Online] Available at: <https://blog.getenjoyhq.com/which-user-research-method-is-best-for-mobile-apps/> [Accessed 14 April 2023].

OWASP, 2019. *Input Validation · OWASP Cheat Sheet Series*. [Online] Available at: <https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html> [Accessed 22 May 2023].

OWASP, 2021. *Cross Site Scripting Prevention · OWASP Cheat Sheet Series*. [Online] Available at: <https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html> [Accessed 22 May 2023].

OWASP, 2023. *Improper Error Handling*. [Online] Available at: <https://owasp.org/www-community/Improper_Error_Handling> [Accessed 22 May 2023].

pandaquests, 2023. *Acceptance testing in JavaScript*. [Online] Available at: [https://levelup.gitconnected.com/acceptance-testing-in-javascript-ca1db5469fe9](https://levelup.gitconnected.com/acceptance-testing-in-javascript-ca1db5469fe9%20) [Accessed 21 June 2023].

Pedriquez, D., 2022. *What is a Context Diagram (and How Can You Create One)?* [Online] Available at: <https://venngage.com/blog/context-diagram/> [Accessed 22 May 2023].

Peek, S., 2023. *What Is Agile Scrum Methodology?* [Online] Available at: [https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html](https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html%20) [Accessed 13 April 2023].

Practical Coding, 2021. *RecyclerView | Everything You Need to Know*. [Video] Available at: [https://youtu.be/Mc0XT58A1Z4](https://youtu.be/Mc0XT58A1Z4%20) [Accessed 28 December 2022].

Quesenbery, 2023. *What Does Usability Mean: Looking Beyond ‘Ease of Use’ - Whitney Interactive Design.* [Online] Available at: <https://www.wqusability.com/articles/more-than-ease-of-use.html#:~:text=An%20understanding%20of%20the%20five> [Accessed 17 June 2023].

Reference\*, 2020. *What Are the Benefits of Conducting Research?* [Online] Available at: <https://www.reference.com/world-view/benefits-conducting-research-97226c9e6ac26d13> [Accessed 19 June 2023].

SABC News., (2021). *South Africans decry souring food prices - SABC News*. [Online] Available at: <https://www.sabcnews.com/sabcnews/south-africans-decry-souring-food-prices/> [Accessed 3 April 2023].

sabp, 2023. *Benefits of Research*. [Online] Available at: <https://www.sabp.nhs.uk/research/about-research/benefits-research#:~:text=The%20benefits%20of%20research%20are%3A%201%20During%20a,readily%20available%20to%20the%20general%20public.%20More%20items> [Accessed 19 June 2023].

Sahani, B., 2018. *SQLiteBlobTooBigException: Row too big to fit into CursorWindow requiredPos=0, totalRows=1.* [Online] Available at: [https://stackoverflow.com/questions/51959944/sqliteblobtoobigexception-row-too-big-to-fit-into-cursorwindow-requiredpos-0-t](https://stackoverflow.com/questions/51959944/sqliteblobtoobigexception-row-too-big-to-fit-into-cursorwindow-requiredpos-0-t%20) [Accessed 21 June 2023].

Salins, M. and Aithal, S., 2022. *An Extensive Analysis of Understanding Consumer Intention towards Reduction of Food Waste*. [Online] ResearchGate. Available at: [https://www.researchgate.net/publication/361314854\_An\_Extensive\_Analysis\_of\_Understanding\_Consumer\_Intention\_towards\_Reduction\_of\_Food\_Waste](https://www.researchgate.net/publication/361314854_An_Extensive_Analysis_of_Understanding_Consumer_Intention_towards_Reduction_of_Food_Waste%20) [Accessed 14 April 2023].

Smart, R., 2020. *Implementing Secure Storage | PSA Certified 10 Security Goals*. [Online] Available at: <https://www.psacertified.org/blog/secure-storage-explained/> [Accessed 22 May 2023].

Smith, N., 2021. *Android Studio Arctic Fox Essentials. Java Edition*. Payload Media Inc.

Soegaard, M., 2019. Usability: A part of the User Experience. [Online] Available at: [Usability: A part of the User Experience | IxDF (interaction-design.org)](https://www.interaction-design.org/literature/article/usability-a-part-of-the-user-experience) [Accessed 18 June 2023].

Software Testing Help, 2014. *How to Write an Effective Test Summary Report [Download Template]*. [Online] Available at: <https://www.softwaretestinghelp.com/test-summary-report-template-download-sample/> [Accessed 21 June 2023].

SOVTECH., 2022. *How much does it cost to build an app?* [Online] Available at: [https://www.sovtech.co.za/how-much-does-it-cost-to-build-an-app#:~:text=One%20can%20develop%20a%20custom,ROI%20and%20exponential%20business%20growth](https://www.sovtech.co.za/how-much-does-it-cost-to-build-an-app%23:~:text=One%20can%20develop%20a%20custom,ROI%20and%20exponential%20business%20growth). [Accessed 20 March 2023].

SQLite, 2023. *SQLite*. [Online] Available at: <https://www.sqlite.org/index.html> [Accessed 19 June 2023].

Sqlite.org, 2019. *SQLite Home Page*. [Online] Available at: [https://sqlite.org/index.html](https://sqlite.org/index.html%20) [Accessed 14 April 2023].

Statistics South Africa. 2019. *The Extent of Food Security in South Africa | Statistics South Africa.* [Online] Available at: [https://www.statssa.gov.za/?p=12135](https://www.statssa.gov.za/?p=12135%20) [Accessed 12 March 2023].

Teshome, M., 2017. *How Food Waste Hurts the Economy – And How You Can Help*. [Online] Available at: <https://www.3blmedia.com/news/how-food-waste-hurts-economy-and-how-you-can-help> [Accessed 23 June 2023].

Thomas, R., 2022. *How much does it cost to build an app in South Africa*. [Online]

Tilley, S. & Rosenblatt, H., 2017. Systems Analysis and Design. 11th ed. Boston: Cengage Learning.

Tolcheva, S., 2021. *How Does the Too Good To Go App Let You Buy Food for Less?* [Online] Available at: [https://www.makeuseof.com/too-good-to-go-app/](https://www.makeuseof.com/too-good-to-go-app/%20) [Accessed 14 April 2023].

Too Good To Go, 2021. *Too Good To Go - fight food waste, save great food - Apps on Google Play.* [Online] Available at: <https://play.google.com/store/apps/details?id=com.app.tgtg&hl=en&gl=US> [Accessed 14 April 2023].

Too Good To Go, 2023. *Save Food - Help The Planet*. [Online] Available at: [https://www.toogoodtogo.com/](https://www.toogoodtogo.com/%20) [Accessed 14 April 2023].

turquoise, 2018. *10 key benefits of market research*. [Online] Available at: <https://thinkturquoise.com/blog/market-research/10-key-benefits-of-market-research/> [Accessed 21 June 2023].

U.S. Department of Agriculture, 2010. *Why should we care about food waste?* [Online] Available at: [https://www.usda.gov/foodlossandwaste/why](https://www.usda.gov/foodlossandwaste/why%20) [Accessed 12 March 2023].

Usability.gov, 2019. Planning a Usability Test | Usability.gov. [Online] Available at: <https://www.usability.gov/how-to-and-tools/methods/planning-usability-testing.html> [Accessed 18 June 2023].

Vaishnavi, M.R., 2019. *Android SDK Tutorial For Beginners | A Complete Guide*. [Online] Available at: <https://www.edureka.co/blog/android-sdk-tutorial/> [Accessed 14 April 2023].

W3Schools, 2023. *XML Comments*. [Online] Available at: [https://www.w3schools.in/xml/comments](https://www.w3schools.in/xml/comments%20) [Accessed 21 June 2023].

Walden University, 2020. *7 Research Challenges (And how to overcome them).* [Online] Available at: <https://www.waldenu.edu/news-and-events/publications/articles/2010/01-research-challenges> [Accessed 22 June 2023].

Whitten, J. L. & Bentley, L. D., 2007. *Systems Analysis and Design Methods*. 7th ed. New York: McGraw-Hill.

Wikipedia Contributors, 2023. *Unit\_testing*. [Online] Available at: [https://en.wikipedia.org/w/index.php?title=Special:CiteThisPage&page=Unit\_testing&id=1156746873&wpFormIdentifier=titleform](https://en.wikipedia.org/w/index.php?title=Special:CiteThisPage&page=Unit_testing&id=1156746873&wpFormIdentifier=titleform%20) [Accessed 21 June 2023].

Wrike., 2023. *What is Iterative and Incremental Development? | Wrike*. [Online] Available at: [https://www.wrike.com/project-management-guide/faq/what-is-iterative-incremental-development/](https://www.wrike.com/project-management-guide/faq/what-is-iterative-incremental-development/%20) [Accessed 23 May 2023].

WWF, 2017. *Food Loss and Waste: Facts and Futures Taking steps towards a more sustainable food future 2017 report za* [online] Available at: [https://wwfafrica.awsassets.panda.org/downloads/WWF\_Food\_Loss\_and\_Waste\_WEB.pdf](https://wwfafrica.awsassets.panda.org/downloads/WWF_Food_Loss_and_Waste_WEB.pdf%20) [Accessed 26 March 2023].

WWF, 2017. *The truth about our food waste problem*. [Online] Available at: [https://www.wwf.org.za/?21962/The-truth-about-our-food-waste-problem](https://www.wwf.org.za/?21962/The-truth-about-our-food-waste-problem%20) [Accessed 12 March 2023].

Zappa, L., 2015. *What is usability testing | checklist | methods | tool and cost | examples and cases*. [Online] Available at: [https://www.userreport.com/blog/usability-testing/](https://www.userreport.com/blog/usability-testing/%20) [Accessed 21 June 2023].

Zinchenko, 2019. *Usability testing: the road to better conversions and user retention – MindK Blog*. [Online] Available at: <https://www.mindk.com/blog/usability-testing/> [Accessed 17 June 2023].