

**Honours Project - MHW225671**

**Final Report**

**2020-2021**

**Department of Computing**

**Submitted for the Degree of:**

**BSc Computing**

**Project Title: Improving the efficiency of Pizza Hut delivery drivers using Optical Character Recognition technology within a mobile application.**

**Name: Philippe Cranskens**

**Programme: BSc Computing**

**Matriculation Number: S1916118**

**Project Supervisor: Robert Law**

**Second Marker: Iain Lambie**

**Word Count: 10,930**(***excluding contents pages, Figures, tables, references, and Appendices***)

**“Except where explicitly stated, all work in this report, is my own original work and has not been submitted elsewhere in fulfilment of the requirement of this or any other award”.**

**Signed by Student: Philippe Cranskens Date: 15/04/2021**

# Abstract

The increasing popularity of the food and drink industry has led to the surge in use of mobile applications for food delivery. Whilst the app market is increasing the number of apps to order food from, there is a distinct lack of applications for delivery drivers.

The purpose of this project is to create a develop and test project that will allow Pizza Hut delivery drivers to use their mobile phone cameras to scan receipts and utilise ML Kit text recognition to identify a postcode and address. The driver will then begin navigation to the identified address using Google Maps, and record completed deliveries. The purpose of the app is to replace the previous way in which Pizza Hut drivers performed deliveries by manually entering address information and recording delivery information. The project will be evaluated by current Pizza Hut drivers in the form of a questionnaire survey. The results of the questionnaire will then be analysed to provide a conclusion regarding the success of the application.

The resulting feedback submitted by the testing participants provided evidence to suggest that the participants experienced increased efficiency whilst using the developed application.

# Acknowledgements

I would like to take this opportunity to give thanks to my supervisor for the help and support provided whilst carrying out this project. The feedback at each stage was extremely helpful.

Additionally, I would also like to thank the participants that agreed to test my app and completed the questionnaires to complete my project.

Finally, I would also like to thank Amy Tucker, for her guidance, support, and for drawing the logo that featured in my application.

Table of Contents

[Abstract 2](#_Toc69382997)

[Acknowledgements 3](#_Toc69382998)

[Introduction 7](#_Toc69382999)

[1.1 Project Background 7](#_Toc69383000)

[1.2 Project Overview 8](#_Toc69383001)

[1.2.1 Project Outline 8](#_Toc69383002)

[1.3 Project Aims and Objectives 9](#_Toc69383003)

[1.3 Hypothesis 10](#_Toc69383004)

[2.0 Literature and Technology Review 11](#_Toc69383005)

[2.1 Investigate Mobile Application Technology 11](#_Toc69383006)

[2.2 An Investigation of GPS Technology 12](#_Toc69383007)

[2.3 An Investigation into mobile document scanning technology 12](#_Toc69383008)

[3.0 Execution 14](#_Toc69383009)

[3.1 Methodology 14](#_Toc69383010)

[3.1.1 Develop an Overall Model 14](#_Toc69383011)

[3.1.2 Build a Features List 15](#_Toc69383012)

[3.1.3 Plan by Feature 15](#_Toc69383013)

[3.1.4 Design by Feature 15](#_Toc69383014)

[3.1.5 Build by Feature 16](#_Toc69383015)

[3.2 Design 16](#_Toc69383016)

[3.3.1 Interface Design 16](#_Toc69383017)

[3.3.2 Class Diagram Design 17](#_Toc69383018)

[3.3 Implementation 18](#_Toc69383019)

[3.4.1 Application Setup 18](#_Toc69383020)

[3.4.3 Bottom Navigation Bar 21](#_Toc69383021)

[3.4.4 Document Scanning 23](#_Toc69383022)

[3.4.5 GPS Navigation 26](#_Toc69383023)

[3.4.6 Session and Delivery Tracking 27](#_Toc69383024)

[3.4.7 Permissions 31](#_Toc69383025)

[4.0 Evaluation and Discussion 33](#_Toc69383026)

[4.1 Testing 33](#_Toc69383027)

[4.1.1 Pre-Testing Questionnaire 33](#_Toc69383028)

[4.1.2 Testing Period 35](#_Toc69383029)

[4.1.3 Post-Testing Questionnaire 35](#_Toc69383030)

[4.2 Evaluation 36](#_Toc69383031)

[4.2.1 Question 1 36](#_Toc69383032)

[4.2.2 Question 2 37](#_Toc69383033)

[4.2.3 Question 3 37](#_Toc69383034)

[4.2.4 Question 4 38](#_Toc69383035)

[4.2.5 Question 5 38](#_Toc69383036)

[4.2.6 Question 6 40](#_Toc69383037)

[4.3 Discussion of Results 41](#_Toc69383038)

[4.4 Ethical and Professional Issues 41](#_Toc69383039)

[5.Conclusions and Further Work 42](#_Toc69383040)

[5.1 Conclusion 42](#_Toc69383041)

[5.2 Further Work 43](#_Toc69383042)

[6. References 44](#_Toc69383043)

[7.Appendices 48](#_Toc69383044)

Table of Figures

[Figure 1 - Example of OCR block and line reading (Alam, 2019). 13](file:///C:\Users\fleep\Documents\Uni\Honours%20Project\Honours%20Project%20Final%20Report.docx#_Toc69382966)

[Figure 2 - Application Main Page 17](file:///C:\Users\fleep\Documents\Uni\Honours%20Project\Honours%20Project%20Final%20Report.docx#_Toc69382967)

[Figure 3 - Class Diagram 17](file:///C:\Users\fleep\Documents\Uni\Honours%20Project\Honours%20Project%20Final%20Report.docx#_Toc69382968)

[Figure 4 - Creating SQLite database. 19](#_Toc69382969)

[Figure 5 - Email Pattern Checker 19](#_Toc69382970)

[Figure 6 - Register User in Database 20](file:///C:\Users\fleep\Documents\Uni\Honours%20Project\Honours%20Project%20Final%20Report.docx#_Toc69382971)

[Figure 8 - Register Screen 20](#_Toc69382972)

[Figure 9 - Login Screen 21](#_Toc69382973)

[Figure 10 - Home Fragment Screen 22](file:///C:\Users\fleep\Documents\Uni\Honours%20Project\Honours%20Project%20Final%20Report.docx#_Toc69382974)

[Figure 11 - Home Page Java Code 23](file:///C:\Users\fleep\Documents\Uni\Honours%20Project\Honours%20Project%20Final%20Report.docx#_Toc69382975)

[Figure 12 - Start Camera Code 24](file:///C:\Users\fleep\Documents\Uni\Honours%20Project\Honours%20Project%20Final%20Report.docx#_Toc69382976)

[Figure 13 - Converting Scanned Image to Bitmap 24](#_Toc69382977)

[Figure 14 - Code for picking postcode line out of an image. 25](file:///C:\Users\fleep\Documents\Uni\Honours%20Project\Honours%20Project%20Final%20Report.docx#_Toc69382978)

[Figure 15- Example of capture image of receipt. Customer name and phone number censored for privacy. 25](file:///C:\Users\fleep\Documents\Uni\Honours%20Project\Honours%20Project%20Final%20Report.docx#_Toc69382979)

[Figure 16 - Example of string replacement for postcodes 26](file:///C:\Users\fleep\Documents\Uni\Honours%20Project\Honours%20Project%20Final%20Report.docx#_Toc69382980)

[Figure 17 – Obtaining latitude and longitude from postcode. 27](#_Toc69382981)

[Figure 18 - Passing latitude and longitude to Google Maps. 27](#_Toc69382982)

[Figure 19 - Checks User Sessions 28](#_Toc69382983)

[Figure 20 - Checking if last delivery is completed. 29](#_Toc69382984)

[Figure 21 - Passing Session Details from fragment to session Info activity. 30](#_Toc69382985)

[Figure 22- Session Fragment. 30](#_Toc69382986)

[Figure 23 - Session Information page displaying deliveries completed for this session. 31](#_Toc69382987)

[Figure 24 - Requesting Fine and Coarse location permission from user. 32](#_Toc69382988)

[Figure 25 - Requesting Camera permission 32](#_Toc69382989)

[Figure 26- Pre-Test Questionnaire first half 34](#_Toc69382990)

[Figure 27 - Pre-Test Questionnaire second half 35](#_Toc69382991)

[Figure 28 - Post-Test Questionnaire Part 1 36](#_Toc69382992)

[Figure 29 - Post-Test Questionnaire Part 2 37](#_Toc69382993)

[Figure 30 - Post-Questionnaire Question 5 Results 39](file:///C:\Users\fleep\Documents\Uni\Honours%20Project\Honours%20Project%20Final%20Report.docx#_Toc69382994)

[Figure 31 - Post Questionnaire Question 6 Results. 40](#_Toc69382995)

[Figure 32 - Participant 7's Error Receipt. 40](#_Toc69382996)

# Introduction

This section will briefly discuss how the fast-food industry has grown in recent years and the technology required to maintain it. It will also discuss technology that can be used in the fast-food industry to provide better support for employees, specifically delivery drivers. It will go on to describe the project outline including the aims and goals the project wishes to satisfy.

## Project Background

The food and drink industry is currently one of the biggest sectors of the UK, employing over 430,000 people nationwide (Food and Drink Federation, 2018). In 2018 around £74 billion worth of food and drink was sold, contributing more than £28 billion to the country’s economy, which was 2.3% more than 2017 (Food and Drink Federation, 2018). Scotland’s own economy generates around £14 billion each year resulting in around 115,000 people employed within the food and drink industry (Scottish Government, 2020). In 2018, the estimated value of food delivery services in the UK was £8.1 billion, however, this Figure is expected to increase to £9.8 billion by 2021 (Malley, 2019). More recently, food delivery and ‘eating in’ is replacing ‘eating out’ in restaurants. One reason is that the younger generation are more inclined towards ordering in than dining out, due to the affordable options online delivery can offer (Gilsenan, 2018).

In response to the growing trend of ‘eating in’, there has been an increase in the number of applications that provide food delivery services. These applications include, but are not limited to, Just Eat, Deliveroo, and Uber Eats (Singh, 2019). These applications cover a wide range of takeaway services, allowing the customer to view a plethora of different restaurants within the one application. Applications such as Just Eat are known as aggregator applications (Business Matters, 2018). The customer places an order within these applications, which is then accepted by the business, at which point the business accepts responsibility for delivering their products to the customer using their own employees. Other applications like Uber Eats and Deliveroo are known as new delivery applications and have their own drivers. New delivery applications will, like aggregator applications, take the customer’s order and place it with the business. However, they will then use the employees working for the new delivery application to collect the order and deliver it to the customer. This allows businesses who do not offer a dedicated delivery service, such as McDonalds, to participate in home deliveries without employing their own drivers. According to statistics, Just Eat, is the highest grossing takeaway application in the UK, with an estimated 105,000 restaurant partners which resulted in a total of 122.8 million orders in 2018 (Luty, 2019).

For the food and drink industry to continue to flourish, the technology to accommodate this growth must be present and continually evolving. Companies such as Just Eat, Uber Eats and Deliveroo would not be as successful without the robust mobile applications and websites currently in use. According to the statistics on SimilarWeb (2020), the traffic for the Just Eat website and app registered 16.6 million visits during the month of December 2019. Handling this volume of online traffic requires the proper investment of time and research into the best techniques and technology available. To cope with the growing need for applications, more courses are being taught to ensure the next generation of developers understand how to create applications effectively (Rahman, 2019). Whilst there is an increase in courses being provided, developers still need to take it upon themselves to improve their coding ability and practises. As more applications are being generated, additional tools are being developed, such as libraries that can extend the functionality of applications. Some examples of these libraries would be, using the phone’s location data to use GPS apps such as Google Maps and unlocking mobile phones through facial recognition. Through continued research into mobile phone application features, more advanced techniques are becoming available to developers, such as using mobile phone cameras for automatic text recognition (Bahi & Zatni, 2019).

Whilst there are numerous applications for customers ordering food online, there is a lack of applications specifically for delivery drivers, especially if working for a smaller business. There are generic applications available for delivery drivers available to download, however, these may not be entirely suitable or practical for every shop as each will have their own methods and requirements. There are applications tailored for Deliveroo and Uber Eats drivers, which work with their respective companies to provide their driver with an easier delivery experience. As more restaurants are opening and securing contracts with companies such as Just Eat, Deliveroo and Uber Eats, this increases the demand for delivery drivers. During the COVID-19 pandemic, it was proven how valuable delivery drivers were as restaurants were some of the few key businesses still allowed to operate during the quarantine lockdowns (GOV.UK, 2020). According to Southey (2020), the revenue for online food delivery increased by 11.5% in comparison to the same period in 2019.

As previously mentioned, mobile development technologies are continually being improved. This creates scope for implementing these technologies in mobile applications, such as the automatic text recognition mentioned above. Automatic text recognition is classified as Optical Character Recognition (OCR) (Shinde & Chougule, 2012). OCR utilises a camera to detect text within a picture, be it a document, a receipt, a street sign, or even handwritten text. The degree of accuracy regarding OCR, however, tends to vary depending on how clear the scanned text appears and the different software performing the task. The OCR systems convert the image text into ASCII characters, which can then be read by computers. The data can then be manipulated in whichever way the developer wishes after it has been processed (Mithe, Indalkar & Divekar, 2013).

## Project Overview

The purpose of this section is to establish the outline of the project, aims and methods used during the project. Objectives that are expected to be accomplished by carrying out the project will also be discussed.

### 1.2.1 Project Outline

The aim of this project is to investigate the possible methods in which mobile application technology can be used to improve Pizza Hut delivery driver’s efficiency.

The method in which Pizza Hut drivers carry out deliveries prior to this project’s application is inefficient. A driver would receive a customer’s receipt along with the items to deliver, they would then manually enter the customers address into a GPS app and either write down the information in a notebook or separate app.

The primary features of the application will feature document scanning technology (OCR) and GPS navigation together to allow a user to deliver more efficiently. Secondary features that will attempted to be implemented are order tracking, mileage per delivery, an anonymous calling feature and a manual address entry. These features will be developed in Android Studio using Java, in a feature driven development life cycle.

To determine whether the application is beneficial to Pizza Hut delivery drivers, several drivers will be selected as test participants. The participants will be asked to test the application for one week. After the testing period, the participants will be asked to complete a questionnaire evaluating the functionality and efficiency of the application. The participants will also be asked if they felt the application was beneficial to them by improving their delivery times.

After successfully completing the outlined objectives, the following question should be able to be answered:

**Can a mobile application implementing Optical Character Recognition technology, successfully streamline the process of performing deliveries for a Pizza Hut delivery driver?**

## 1.3 Project Aims and Objectives

The aim of this project is to create an application that will allow delivery drivers to streamline their delivery services, as well as track their daily orders. By completing a literature and technology review, the information needed to complete the following objectives will be acquired.

Literature and Technology Review Objectives:

* Investigate Mobile Development Technologies.

An investigation into the different mobile development technologies will help determine the best platform for developing the application. This will involve finding out the benefits of each development platform and which one would be most suitable for the features that are intended to be implemented with the least amount of difficulty.

* Investigate Optical Character Recognition (OCR) technologies.

This will include using the mobile phones camera to scan live documents to take certain lines of text and process them within the application. This will be done by investigating which technology other developers are using to provide similar results.

* Investigate GPS Technologies.

This will be implementing a GPS technology that will allow users to navigate successfully to a customer’s house after using the OCR technology to determine the customers postcode and address.

Primary Research Objectives:

* Develop android application.

Using the research into mobile application technologies, an application will be created. The application will be made to ensure all primary requirements regarding the application are implemented. As many secondary features will be attempted to be implemented also, given the strict timeline however, all secondary features may not be possible.

* Create questionnaire.

A questionnaire will be created that allows participants to determine if the app improved participants efficiency whilst performing deliveries. The questionnaire will also determine if the participants found the secondary features useful.

* Obtain participants for testing and completing questionnaire.

Pizza Hut delivery drivers will be asked if they are willing to participate in testing the application for the duration of one week. An ethics form will be created by the researcher and filled out by the participants to ensure that all ethical guidelines are being adhered to.

* Carry out the testing and allow participants to answer the questionnaire.

The testing participants will carry out the test over the defined period of one week. At the end of the defined period, the participants will be given the questionnaire to provide the feedback that will determine the success of the app.

* Gather results and analyse data.

Analysis of the data received from the participants questionnaires will take place, allowing the developer to determine the success factors of the application. The success will be determined on whether the participants drive times were improved from their previous methods of delivery.

## Hypothesis

The created application using Optical Character Recognition should increase the efficiency in which Pizza Hut delivery drivers’ complete deliveries and record relevant details.

# Literature and Technology Review

The literature and technology review is a critical step when carrying out develop and test projects. It allows the researcher to obtain the information that will allow them to create a successful project that encapsulates all the requirements outlined beforehand. During this phase, the researcher should develop knowledge that will allow them to complete the tasks they have outlined in the project aims and objectives. The areas of research are mobile technology, GPS technology, and document scanning technology.

## Investigate Mobile Application Technology

The smartphone industry has grown exponentially since the modern smartphone has become available for widespread use. According to Statista (2020), the number of smartphone users has rapidly increase since 2016. In 2016 there was an estimated 2.5 billion smartphone users, however in 2020 there were 2.8 billion and this is expected to rise by another 300 million by 2021. To accompany the growing number of smartphones, the number of mobile applications (apps) has also increased massively with consumers spending more time on apps than ever before. Whilst there are new mobile phones being released every year, the primary reason that users are spending an increasing amount of time on their phones is the captivating app market. There are a huge number of apps that can cater to almost every demographic (Bapat, 2018).

As well as apps for recreation there are also numerous apps being created to assist and improve the efficiency of businesses. For example, travel agencies or holiday makers can target users with customised holiday destinations based on their data history (Ostdick, 2016). Whilst there are apps that are lucrative for businesses, there are also apps that are created for the sole purpose of improving the quality of life for people with chronic medical conditions. In 2019 a study was undertaken to determine the effectiveness of an application that would help users manage their diabetes and regulate their medication more effectively (Jeffrey et al, 2019). This application proved to be beneficial to many of the participants as it improved their health and quality of life, providing evidence that mobile applications can be beneficial to society.

With regards to delivery driver specific apps, the job of delivery drivers has changed drastically with the development and popularity of apps. Drivers can accept orders, check in with shops, log mileage, plan efficient routes and record all relevant information within a well-designed app (Bringoz Team, 2020). There has also been research carried out in utilising mobile phone sensors to monitor delivery drivers whilst driving. This could give companies greater insight into how drivers perform whilst delivering and allow more transparency between drivers and employers (Castignani, et al, 2015). Whilst there are applications to improve driver efficiency, there is also the risk of using apps while performing deliveries. These risks come in the form of notifications and are incredibly dangerous as they can cause distraction to drivers who are in the process of driving (McNabb, 2017).

There are several development suites to choose from when creating mobile applications. Developers will first have to decide which operating system their app will be created for, such as IOS, Windows or Android. A factor that may influence a developer’s choice is the ease of developing for the specific system. For example, iOS applications can only be used on iPhones as they do not function on devices not produced by Apple (Jakimoski et al, 2017). While iOS apps can only be developed for an iPhone, android apps can work on a range of phones such as Samsung, Huawei, Sony Ericson, Google Pixel, and many more. After choosing a platform, developers must then decide on the programming language to use, such as C#, Java, Python, C++, and many more (Flora et al, 2014). According to Nandhini (2018), Android app development has become easier than ever before and allows developers to implement advanced and powerful code with little difficulty. This allows developers to create robust powerful apps, that have the support of additional libraries that Google have assisted developing. While iOS development is also capable of producing applications of the same calibre if not better than Android applications, the biggest detriment to development is the required hardware. To develop iOS applications effectively, a Mac computer is required (Apple Developer Documentation, 2021).

As well as the platform and language the developer must then choose an environment to develop in. When developing Java apps, the most prevalent environment is Android Studio (Abdullah, 2020). It is a free development environment recommended by Google providing numerous features to assist developers, such as emulators to run the applications, GitHub integration, built in service for Google libraries and a robust IntelliSense.

## An Investigation of GPS Technology

With the advancement of technology and smartphones becoming more accessible as discussed above, the use of Global Positioning Systems (GPS) is becoming more frequent. There are several apps available that can perform GPS functionality that millions of people use daily. These types of apps can perform many different functions, including but not limited to; finding the fastest route to a destination from the user’s location, use voiceover to inform users when to turn or take an exit, warn users about congestion on their routes, provide an array of different displays in which they can see their surroundings, speed limits, speed cameras and road closures. Many GPS apps are free, however, some include payments that enable extra features, such as downloading maps to use offline. The top applications available currently are Google Maps, Waze, MapFactor, HERE WeGo Maps, and MapQuest. Both Google Maps and Waze are owned by Google and as such provide more robust features than some of the other apps mentioned (Hindy, 2020).

Google Maps is one of the first public GPS applications, made available in February 2005 (Reid, 2020). It allows users to enter an address or postcode anywhere in the world and be directed there via different transport types of the user’s choice. Users can mark favourite places, places they work, their home or even places they have parked frequently. Another beneficial feature of Google Maps is that the user can view information about their destination. For example, they could view ratings of a restaurant or other venues nearby (Eadicicco, 2020). Furthermore, when integrating Google Maps in an application, Google provides $200 of credit a month (Google Cloud, 2021). This would allow most applications that do not use the app frequently to use Google services for free.

## 2.3 An Investigation into mobile document scanning technology

Optical Character Recognition (OCR) is the process of transforming optically scanned images of written or printed text characters into character codes such as ASCII (Lais, 2002). The image is taken via camera; therefore, it can also be taken with a mobile phone camera as the quality of mobile phone cameras in recent years has improved greatly (Yilmazturk & Gurbak, 2019). OCR can be used to free up physical space for businesses by scanning important documents, uploading the documents to a computer, and viewing them virtually (itshowcase, 2020).

An example of OCR benefitting a company or business is Quickbooks. Quickbooks is a financial advisor company that allows their customers to scan receipts and upload them to their website to pull the required information from the receipts. With this information they can then work out the best course of action for the customer’s needs (Intuit Quickbooks Website, 2021). Another example of a beneficial application utilising OCR is Camscanner, a free mobile application that allows users to scan library documents and store the information on a phone. The app can alter images to add notes or tags and provides the user with an enhanced experience that would not be possible using a physical copy of a book (Camscanner 2021). These applications demonstrate the use of OCR technology and provide evidence that mobiles can perform this functionality.

There are several different frameworks that are available that can provide OCR capabilities. Some of these frameworks specialize in certain aspects, such as scanning receipts, or redacted documents. Some are open source whereas others require a fee. The most popular currently are Tesseract, Google Vision, ML Kit for Firebase, and Aabby Cloud (Han & Hickman, 2019). According to Iakovlev (2020), Google Vision slightly outperforms Tesseract when it comes to documents such as receipts.

A picture containing text

Description automatically generated

Figure 1 - Example of OCR block and line reading (Alam, 2019).

OCR technologies function by dividing images into smaller sections and analysing the smaller pieces of data. As seen above in Figure 1, the yellow line indicates blocks, whereas the green indicates lines of text and the pink specifies elements or words. This allows the developer to divide the image into segments and identify areas of text that are relevant to the required purpose.

# Execution

The aim of this project is to build a mobile application that will allow Pizza Hut delivery drivers to perform their duties and record their deliveries more efficiently. This will be achieved by developing an application that will scan order receipts, navigate to the correct address, and keep track of all deliveries during a shift. The methodology used during this project is Feature Driven Development (FDD).

Through development of this application, information and knowledge gained during the literature and technology review will be applied to ensure the required and necessary functions are developed and implemented. This research included the best mobile development technologies, OCR functionality and GPS technology.

The testing stage of this project will be carried out using test participant drivers, allowing the developer to determine through participant feedback, if the application improved driving times and worked accurately.

## Methodology

Project lifecycles refers to the Software Development Lifecycle (SDLC). This is the process in which a product is created through design, development, and testing. There are several different styles of SDLC, however the process that will be used to create this project is the iterative model. The iterative model is cyclical, meaning there will be repetition of stages. The developer will carry out and repeat the design, development and testing stages until the product is completed (Shylesh, 2019). If the software were to go into production, the production stage would also be revisited, always improving the model. This process of visiting stages more than once is part of the agile methodology, which allows a more flexible style of working. In the past the waterfall method was widely used, which only visited each stage of production once throughout a project without revisiting. It could be argued this is a rigid method of creating software and has been, in recent years, surpassed by agile which is more effective at reacting to and resolving problems (Baseer et al, 2015).

The application proposed for this project will contain several features, however not all features will be required for the application to function. There will be a priority list of features that will be implemented first, based on their importance to the application. Therefore, the process in which the application will be developed is using Feature Driven Development (FDD). FDD is a development cycle modelled around the agile ideology and focuses on an iterative cycle of designing then developing software features individually. The FDD development process has five distinct phases. These phases are: Develop an Overall Model, Build a Features List, Plan By Feature, Design By Feature, and Build by Feature (Nawaz et al 2017), which will be discussed below.

### 3.1.1 Develop an Overall Model

The optimum model for a proposed system would normally be chosen from a list of several options within a team-based setting. However, this project was carried out by a single developer and multiple layouts were not necessary. Therefore, after considering the flow of the application, a basic outline was created.

### 3.1.2 Build a Features List

A features list is an important part of the development cycle and outlines the specific tasks that the project will need to complete.

* Navigation function, allowing the application to pass the extracted postcode over to the GPS navigation system and navigate the user to the desired postcode address.
* Implement a secure login feature, making sure that only valid entries are accepted within the application.
* Implement an activity in which a user can request a new password when they forget their password.
* Store the relevant information in a database, allowing the user to record and review all deliveries they have made for a session.
* Implement a register function, ensuring only valid entries are accepted for a new user signing up.
* Ensure the application is visually acceptable and can be navigated and interacted with easily.
* Scanning function, allowing the application to access the mobile phones camera and take a picture, allowing a postcode to be extracted from the image of a receipt.
* Allow deliveries to be marked as complete so that the user can ensure deliveries have been completed as well as allowing the user to look at delivery times and distance.

### 3.1.3 Plan by Feature

During the plan by Feature phase, the list of features to be implemented are given a priority status depending on the importance of the task. The higher the priority, the sooner the feature will be developed. These priorities can be identified by their importance to the functionality of the project.

1. Implement a secure login feature, making sure that only valid entries are accepted within the application.
2. Implement a register function, ensuring only valid entries are accepted for a new user signing up.
3. Scanning function, allowing the application to access the mobile phones camera and take a picture, allowing a postcode to be extracted from the image of a receipt.
4. Navigation function, allowing the application to pass the extracted postcode over to the GPS navigation system and navigate the user to the desired postcode address.
5. Store the relevant information in a database, allowing the user to look back and see all deliveries they have made for a session.
6. Allow deliveries to be marked as complete so that the user can ensure deliveries have been completed as well as allowing the user to look at delivery times and distance.
7. Ensure the application is visually acceptable and can be navigated and interacted with easily.
8. Implement an activity in which a user can request a new password when they forget their password.

### 3.1.4 Design by Feature

The Design by Feature stage requires the developer or team to plan the way in which they will implement the feature. This determines how the feature will integrate with the program, the naming convention of methods and functions, what values they return, how they will interact with the program and how the team or developer will ensure the feature integrates smoothly.

### 3.1.5 Build by Feature

After the task has been designed in the previous stage, the last remaining step is to develop the task or feature, following the guidelines laid out previously. When this stage is completed, a working prototype displaying the functionality of this task should be able to be seen and used.

## Design

The design stage is where the initial layout concepts of the application will be created. This is where decisions such as the layout for application pages, colour, feel and any other aesthetic properties will be decided. The design stage is an important part of developing any application where a user will be interacting with it. The biggest influence of the design for this application was from the developer’s personal experience as a Pizza Hut delivery employee. With the knowledge gained from performing the role of delivery driver, the author can perceive accurately what the end user would be looking for in the application. As a result, the application would be held to more practical standards as opposed to a developer with potentially no practical experience of performing these tasks. As well as their own experience, the author also gained suggestions from other drivers about features they would like to see in an application such as this.

### 3.3.1 Interface Design

As the deadline for the project had time constraints with little room for error or extra features, it was decided that the layout of the application would be minimal to avoid a lengthy process of creating a full user interface. If more time were available for this project, a more detailed design phase would take place with wireframes and user input to create an aesthetically pleasing interface.

As the application was to solely assist Pizza Hut drivers, the colour scheme was based on Pizza Huts colours of red, black, and white. An example of the colours can be seen in Figure 2 below, as stated previously, the design was kept as minimalistic as possible to allow users to interact easier. Figure 2 displays the general layout the application will take throughout development. According to Shokurova (2020), it is important to make use of the small space available and avoid clutter or information overload. For this reason, the application will use a general and common control used in popular social media apps, such as “Instagram” and “TikTok”. This would provide users with a familiar layout, aiming to enable easier use of the application.

Graphical user interface, text, application

Description automatically generated

Figure 2 - Application Main Page

### 3.3.2 Class Diagram Design

Diagram

Description automatically generatedAs the application being created would be an object orientated program utilising a database, a class diagram is helpful for creating classes and entity relationships. The diagram in Figure 3 displays the three entities used in the application, Users, Sessions, and Deliveries. The User class stores information about each registered user and stores a username, email address and password. Each User class is assigned a unique ID that helps identify each user. As Figure 3 displays, via the relationship line, a user can have many sessions, but a session can only belong to one user. The User and Session class are linked via the userId key, it is the primary key in the User class and the foreign key in the Session class. Having the foreign key and primary key matchup between entities is important for correct linking in the database. If they do not share the same ids between the primary and foreign key, the data will not match. A Session is the period of time in which the user will be performing deliveries, they will be required to start a new session before recording any deliveries. A Session class will contain a unique session ID; it will contain a variable called active which determines if the session is active or not; the date in which the session took place; the session start time; the session end time; the number of deliveries in the session and the userId to link to the User class as mentioned above. Like a User having many Sessions, a Session can have many Deliveries, however a Delivery can only belong to one Session. A Delivery is linked to an individual Session by the session ID key which is a foreign key for a Delivery instance, and the primary key for the Session instance. It records the start time, delivered time, duration, address, and distance of a delivery. The attributes seen in the class diagram in Figure 3 display how the objects are stored in the database.

Figure 3 - Class Diagram

## Implementation

This section describes in detail the process in which the application was constructed. There will be a detailed explanation on the coding procedures, as well as the structure of the application layout including class files. The process in which the application utilises the camera to acquire the specified information from the scanned document will be explained in detail, followed by what is done with the captured information. To ensure coding practises are upheld to an acceptable standard, the Google Java Style Guide(Google.github.io, 2021), will be adhered to where applicable.

### 3.4.1 Application Setup

The correct application must be selected to ensure there are no problems at a later stage of development. There are several types of starter templates available to choose from when starting an Android project. For this project, a basic empty activity was chosen, using the Java language so that no additional code was implemented. When creating a new project, the main activity screen is where a user will be directed to upon starting the application. In this application, the main activity is where the user chooses to register or login. At this stage of the application, the developer created the initial Java class activity files required for basic functionality of the application for registering, logging in, and the home page screen. After creating the Java class activities, the XML files to display the activities had to be created. After the empty files were created, the next step was to create a registration system and initialise the database.

#### 3.4.1.1 Login and Registration

As the project is a FDD project, it requires creating new features in durations of one or two weeks. As the login and registration features are less complicated than other features, such as the camera scanning, they can be combined and implemented as one feature.

#### 3.4.1.2 Registration

With the initial files created, the next stage would be to allow users to register with the application. To allow the application to track user registrations, a database would have to be created. After researching multiple options for performing this function, SQLite proved to be the most popular framework when storing data on Android devices. As the information would only be viewed by the driver to keep track of deliveries, it was decided that a local database would be sufficient for the purposes of this application. The method in which the database was created can be seen in Figure 4.

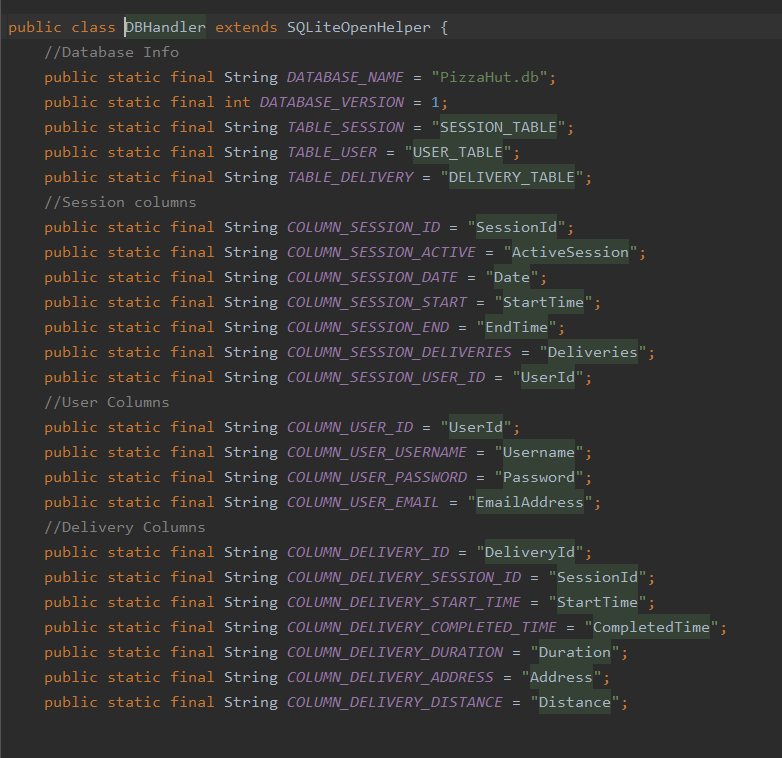


Figure 4 - Creating SQLite database.

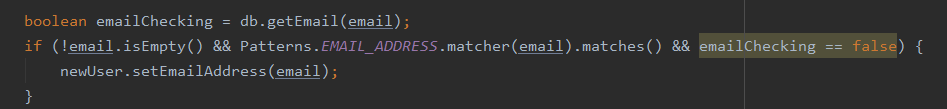
Figure 4 displays the initial database being created according to the class attributes seen in Figure 3. This allows methods to be created that can interact with database data via reading or writing. The register user method will allow new users to be entered into the database, allowing the data to be queried when logging into the application. The application also checks that the email address supplied by the user is a valid email type using a pattern checker supplied by Android studio. This can be seen in Figure 5.

Figure 5 - Email Pattern Checker

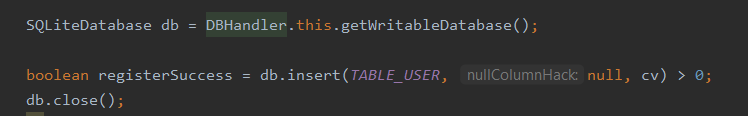


Figure 6 - Register User in Database

The method seen in Figure 6 is inside the register user method and creates a Boolean variable depending on whether the user is inserted into the database or not. This allows the developer to check if the method was successful and display the information to the user.

Application

Description automatically generated with medium confidence

Figure 8 - Register Screen

The image seen in Figure 7 is the register screen a user will fill out. This will call the method, displayed in Figure 6, when the user selects the register button.

#### 3.4.2.3 Login

The login feature relies on the registration feature being implemented and thus could not be completed before registration. The login functionality checks the SQLite database for the information entered in the login activity. The login activity will then inform the user if there is a problem logging in or navigate to the home page if successful.

Graphical user interface, application

Description automatically generated

Figure 9 - Login Screen

### 3.4.3 Bottom Navigation Bar

After a successful login, the user will be taken to the home page. The home page for this project does not display an activity but acts as a control panel for displaying fragments. Fragments are partial views that cannot exist on their own but must be hosted by an activity. In the case of this project, there are three fragments being hosted by the home page, they are the home fragment, the camera fragment, and the sessions fragment.

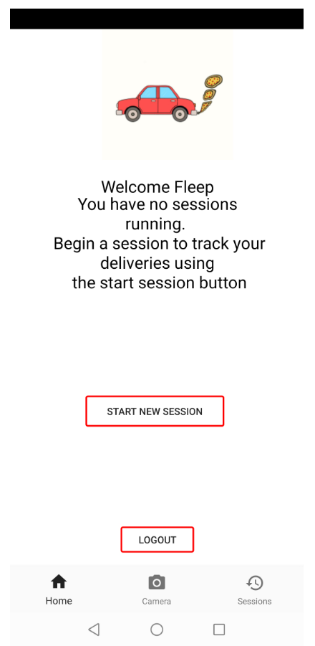


Figure 10 - Home Fragment Screen

The three fragments can be seen at the bottom of Figure 9 in the navigation bar, with the titles ‘Home’, ‘Camera’ and ‘Sessions’. These icons act as buttons and allow the user to navigate between the three fragments with a single button click. This is like the popular social media applications already widely used, as previously mentioned. The selected fragment defaults to the home screen upon logging in, however as the code in Figure 10 displays, when the user clicks any of the other icons, the selected fragment variable becomes the chosen fragment and loads the corresponding fragment XML file. The home page XML file contains an empty fragment that loads the corresponding fragment selected by the user, as well as the bottom navigation bar resource file that is rendered directly below the empty fragment.



Figure 11 - Home Page Java Code

### 3.4.4 Document Scanning

The document scanning phase, being the driving force behind the project, required the largest time and effort investment. There were several setbacks when implementing this feature as it was new technology to the developer. Initially, as a result of the literature and technology review, the choice was made to implement this feature using Google Vision. This proved difficult to implement however, and after further research into the issue, a new possibility was identified as a way in which the document scanning feature could be implemented. This was the ML Kit by Firebase. After switching to ML Kit, the developer found more detailed instructions on how to implement the technology, with settings and options for ML Kit already installed on Android Studio. After configuring the application to work with ML Kit, the application was then able to utilise the mobile phones camera to take a picture and convert that image to a BitMap variable.

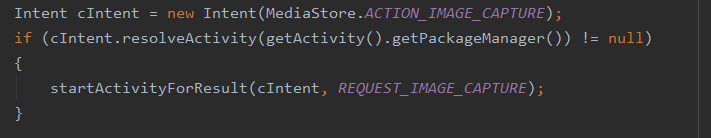


Figure 12 - Start Camera Code

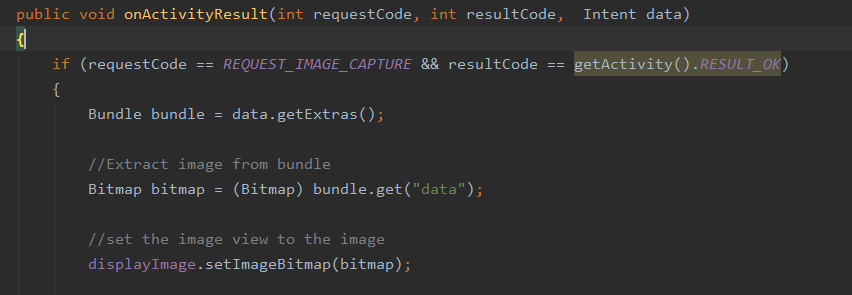
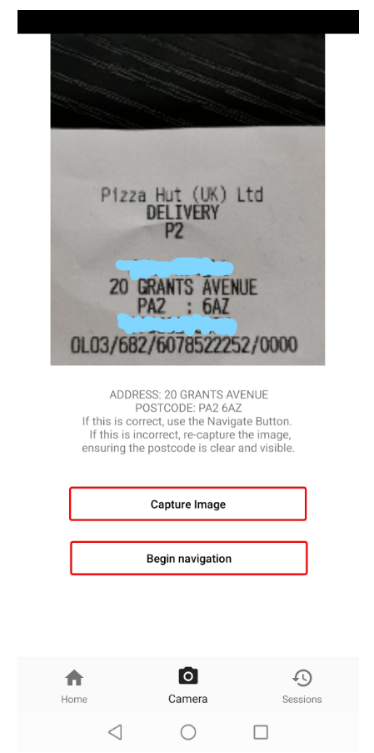
In Figure 11, the code to start the camera activity and return data can be seen. This starts the camera function and informs the program that once the camera function is finished it will return data.

Figure 13 - Converting Scanned Image to Bitmap

In Figure 12, the function that runs here is when the user returns from the camera screen after taking a picture. The “Intent data” seen in the parameters for this method is the image taken from the camera, the code then converts the image to a Bitmap. With the image now converted to a Bitmap, the ML Kit can then convert the Bitmap to a Firebase Vision Text object. This then allows the developer to filter through the text captured and search for a postcode.

Figure 13 displays what a typical image capture of a receipt would look like when using the camera function. The main block of text is what will be filtered to determine a correct postcode. Figure 14 seen below, displays the code used to determine the postcode from the block of text processed.



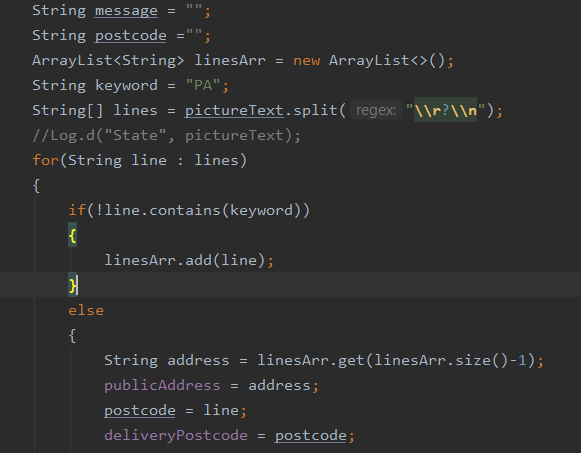


Figure 14 - Code for picking postcode line out of an image.

Figure 15- Example of capture image of receipt. Customer name and phone number censored for privacy.

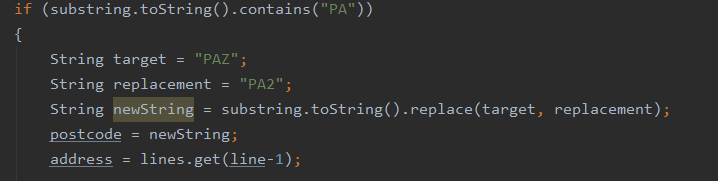
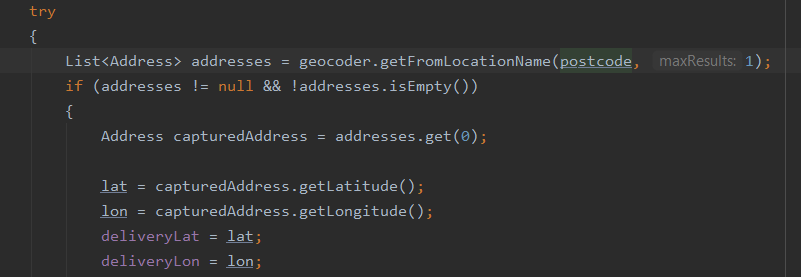
As the area that deliveries take place for this project is in the Paisley area, all postcodes begin with the two letters “PA”, as such, this is the perfect identifier for the line that indicates the postcode. Splitting the text block into an array of strings allows the developer to loop through each line in the array and then search for the line that contains the postcode, identified by the two letters mentioned. After acquiring the postcode from the text block, the address can also be determined as the line directly above the postcode typically contains the customers street address and number. This allowed for a greater degree of accuracy when passing the navigation data over to Google Maps. One issue that began to occur on certain instances of image captures was when the postcode started with the code “PA2”. As seen in Figure 15, the number 2 looks similar to the letter Z and as such the text recognition software mistook the 2 for a Z. This was problematic as the postcode had to be accurate for deliveries to occur as normal if not faster than before using the application. The way in which this was solved was by ensuring any time the program identified a postcode containing the string “PAZ” was to replace that part of the postcode with “PA2”. Postcodes for Paisley will only allow the format of letters and numbers as follows “AA11AA”. This will eliminate the possibility of “PAZ” appearing in a legitimate postcode. Similarly, the software at times would confuse the number 1 with the letter “l”. As with the “PAZ” there is no postcode that would exist with this format, and was similarly fixed, substituting part of the string for “PAl” with “PA1”. An example of this procedure can be seen in Figure 15 below.

Figure 16 - Example of string replacement for postcodes

### 3.4.5 GPS Navigation

The next feature to be worked on after the application can identify a postcode and address from an image would be to pass that information to a GPS application. Through the research carried out in the literature and technology review, the GPS chosen to perform the task for this project was Google Maps. First the latitude and longitude must be extracted from a given postcode, this can be done using the code shown in Figure 16 using a Geocoder instance.

Figure 17 – Obtaining latitude and longitude from postcode.

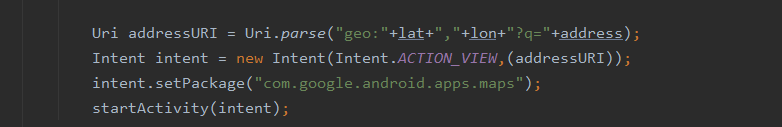
After acquiring the latitude and longitude from the postcode, the next stage was to create a URI to pass over to Google Maps. For this stage to be successful, Google Maps should already be installed on the mobile device. The app will then create a new intent with the destination being Google Maps and the data being sent is the URI, as seen in Figure 17.

Figure 18 - Passing latitude and longitude to Google Maps.

As can be seen in Figure 17, the address captured from the image is also sent over to further increase the accuracy of the location being sent to Google Maps.

### 3.4.6 Session and Delivery Tracking

The final feature of the implementation stage for the project was a way to track deliveries and sessions for a user. A session had to be started before allowing the user to begin taking deliveries, and then any deliveries taken during this session had to be recorded for the appropriate session. Creating a new session is what the user should do after they login, thus the “begin session” and “end session” buttons were placed on the first screen the user sees after logging in, which is shown in Figure 9. If the user attempted to take a delivery before starting a session, their deliveries would not be tracked, therefore it was decided that a user would not be able to take a picture of a receipt or begin navigation to an address without first beginning a session. The begin session button creates a new session in the SQL database, similarly to how a user is created as described in the previous registration stage. When checking if a session is already running, the logged in user object must be passed to the method that checks for active sessions. The way in which the code checks for running sessions can be seen in Figure 18 below.

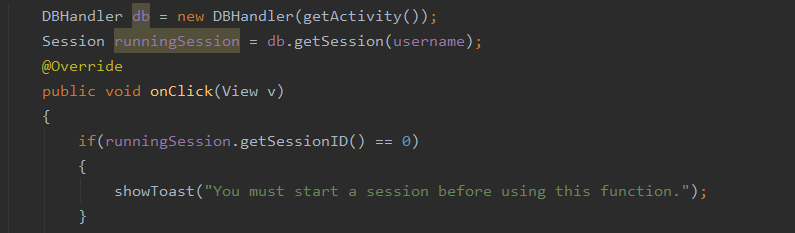


Figure 19 - Checks User Sessions

After the application was able to track sessions, the next stage was to track each delivery a user completed, as well as recording the distance and time taken to complete the delivery, then display it for the user. The delivery object is created when a user presses the begin navigation button, which marks that they have begun the delivery and thus the delivery object can be safely created. When the delivery object is created, the session id attribute is marked as the currently active session id. The delivery start time value takes the current local time when the navigate button is clicked, the end time is also set to the current time so it can be updated after the delivery is completed, and the duration is set to “Not Completed”. The distance attribute for the delivery is the distance between the address passed over from the image, and the GPS location of the phone when beginning navigation. This creates the delivery object to be stored into the database when beginning navigation to the address. After completing the delivery, the user will return to the application from Google Maps, where the “Mark Delivery as Complete” will now be visible as the user has a delivery in progress. The app checks for the user having a delivery in progress by querying the database and looking for a delivery with “Not Completed” as the duration. As well as displaying the button to mark the delivery complete, the user will not be able to logout, end the session, or begin a new delivery until they mark the current delivery as complete. An example of checking the last delivery is completed can be seen below in Figure 19, where if the delivery is unfinished, it displays a message to the user informing them that they must complete the previous delivery.

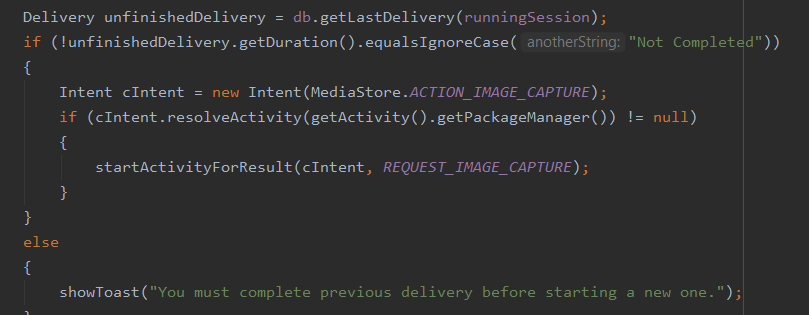
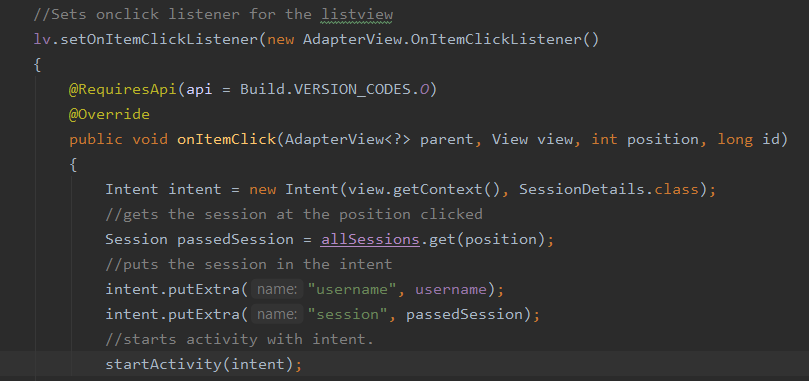


Figure 20 - Checking if last delivery is completed.

After creating a new delivery object and detecting when it is not complete, the next step was to then update the delivery object as completed when the user is finished. When the user clicks the button to complete a delivery, the delivery object is then updated in the database. The completed time changes from “Not Completed” to the exact time the user clicks the complete button, and the duration is calculated between the start time and end time.

With the delivery object updated, it is ready to be displayed to the user as a completed delivery in the session fragment. The sessions are displayed on the sessions fragment and can be accessed via the session icon on the navigation bar. This fragment displays a list view component that lists all the sessions the user has created in the past as well as the current session they are running. When clicking on any session, the user will be taken to another activity, displaying all the deliveries they completed during that session in another list view. The session list view is populated by querying the database and returning all sessions that correspond to the user logged in. The returned data is then inserted into the list view with an action listener implemented for each item on the list. When the user clicks a session item in the list, the on click listener will take the session id matching the one clicked and pass the session information over to the session info page as shown below in Figure 20.

Figure 21 - Passing Session Details from fragment to session Info activity.

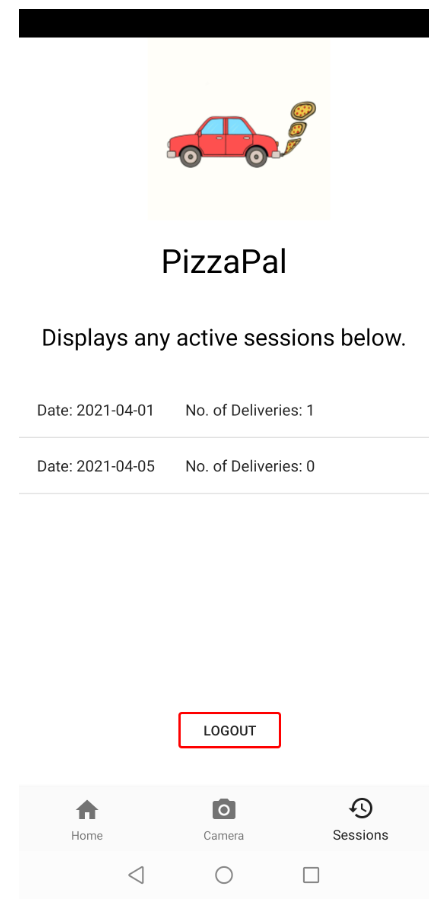


Figure 22- Session Fragment.

After passing the selected session to the session information activity, the activity will perform a similar function to the sessions fragment by querying the database for all deliveries completed in that session. The list will then be populated with the returned deliveries.

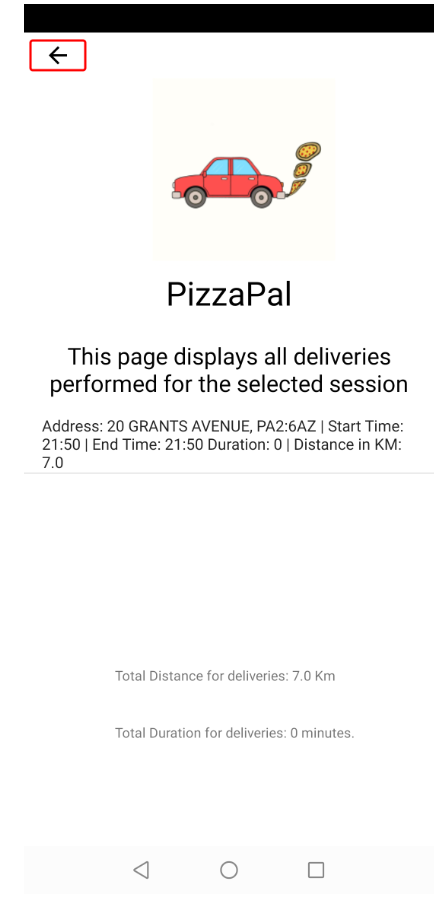
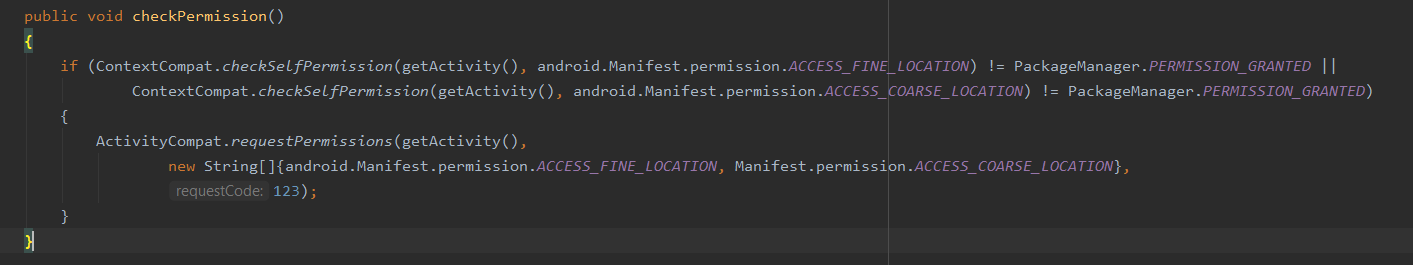


Figure 23 - Session Information page displaying deliveries completed for this session.

As seen in Figure 22, the session information page will display all deliveries performed for the session selected in the session fragment screen. Below the list of deliveries is the total number of minutes spent on deliveries for this session, as well as the total distance for all deliveries.

### 3.4.7 Permissions

As this application utilises the camera to scan receipts and accesses the phones’ location when calculating the distance from addresses, it requires permissions to run. These permissions require interaction from the user to permit the application to use these features, without access the application will not run. As such, when logging in for the first time, the application will request these permissions from the user so the program can function as normal. The location permissions can be seen below in Figure 23, which allows fine and course location permissions.



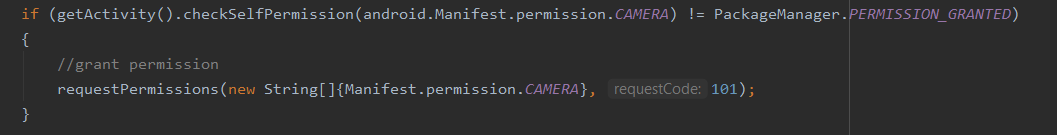
Figure 24 - Requesting Fine and Coarse location permission from user.

Figure 25 - Requesting Camera permission

Whilst Figure 23 shows the application requesting location service permissions, Figure 24 displays the code for requesting camera permissions from the user so that the app can make use of the camera for document scanning.

# 4.0 Evaluation and Discussion

## 4.1 Testing

This section details how the app was tested to provide results, determining if the app was able to increase delivery driver efficiency as detailed in previous sections. Several Pizza Hut delivery drivers were selected as test participants to trial the app. Participants were required to complete a pre-testing questionnaire to gain information about their current methods for tracking deliveries. After using the application for one week, the participants would then fill out a post-testing questionnaire. The post-testing questionnaire would determine if the participants had improved their delivery times whilst using the application. Questionnaires were chosen as the method to gather results as findings suggest that questionnaires can be an effective and efficient tool for collecting user input (Fox et al, 2000). The questionnaires were constructed in a way that let the developer receive feedback that would assist in developing a conclusion regarding the projects question. The questionnaires will be created using Office 365’s Microsoft Forms, which the participants would then be able to access via email.

### 4.1.1 Pre-Testing Questionnaire

The pre-testing questionnaire would determine how participants delivered to addresses and how they recorded information about their deliveries before using the app. This would determine how many participants already recorded deliveries through their own means, if they chose to record the information, and what GPS app they would use. Prior to using the application, drivers received delivery information through paper receipts, and would manually enter the address into their choice of GPS app. If the results of the questionnaire were to display that the participants preferred to use another app in place of Google Maps, then the developer would have to take that into consideration when implementing Google Maps. The questionnaire also aimed to determine if drivers felt that recording delivery information was important to them and if they would be interested in an application that automatically tracks their deliveries.

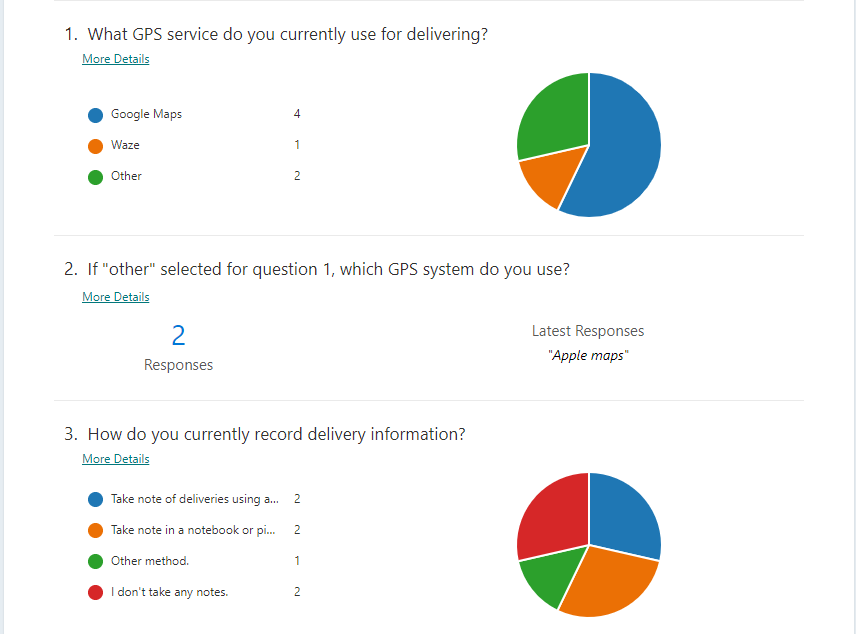


Figure 26- Pre-Test Questionnaire first half

Figure 25 displays the first half of the pre-test questionnaire given to the participants and displays 4 out of 7 drivers (57%) were already using Google Maps, surmising that these drivers would feel more familiar using the app. Figure 25 also shows that most drivers record their deliveries in some form whereas two participants do not take any note. As the created application can combine the functions of navigation and recording delivery information, the application was expected to improve delivery times.

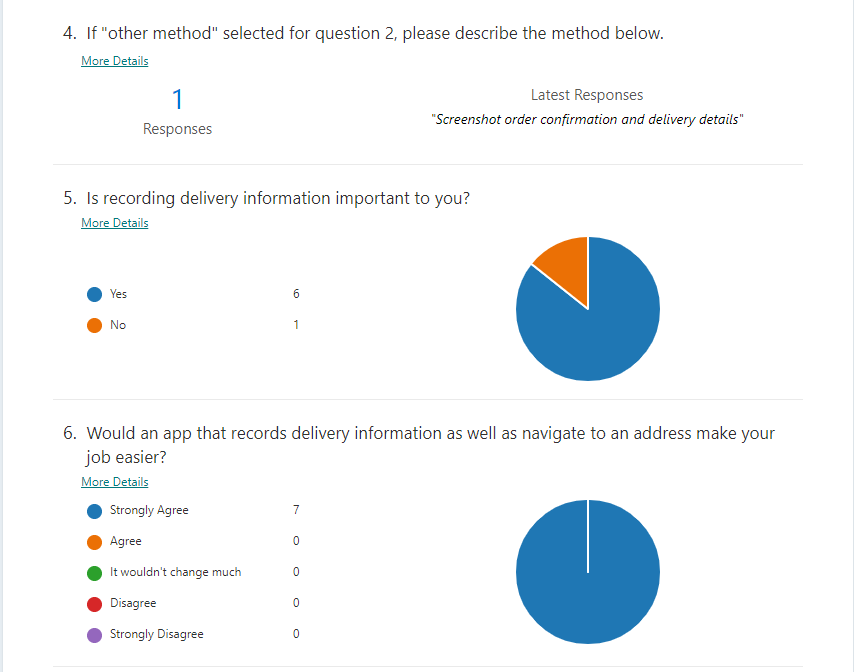


Figure 27 - Pre-Test Questionnaire second half

Figure 26 is the second half of the pre-testing questionnaire and shows that six out of the seven participants expressed that recording delivery information is important and that an app that automatically records the information and navigates to addresses will be beneficial to them.

### 4.1.2 Testing Period

The testing period is the scheduled time for the participants to use the application and develop an opinion on how it affected their duties as a delivery driver. Whilst most participants had an android device capable of running the application, some drivers had iPhones which were not. For the iPhone users, there was an option to use an android device which would allow use of the application. After a demonstration on how to use the application, participants then able to use the application during their scheduled working hours for one week.

### 4.1.3 Post-Testing Questionnaire

Following the testing period, a second questionnaire was issued to the participants. The questionnaire gathered feedback from the participants to allow analysis of the results. The analysis of these results determined if the project succeeded in increasing driver efficiency when performing deliveries compared to the previous method, as mentioned in the introduction. As mentioned, drivers before using the app would have to manually enter the details of the receipt into a GPS app of their choice, and then record delivery information in a notebook or separate app, causing delay in the delivery service. The questions posed to the participants determine if the app was easy to use, if use of the app was quicker than the previous method of manually entering and recording, and the accuracy of the app. Participants also provided feedback on any features they thought worked well within the app, as well as any suggestions they may have for future development.

## 4.2 Evaluation

Chart, logo, company name, pie chart

Description automatically generated

Figure 28 - Post-Test Questionnaire Part 1

### 4.2.1 Question 1

As seen in Figure 27, the first question shows that six of the seven participants found the app extremely easy to use with one participant finding it somewhat easy to use. As stated in the “Plan by Features” list, one of the features of the project was to make the application easy to use and navigate. The results gathered from question one of the Post-Test Questionnaire suggest that this was successful due to all answers indicating a positive response.

### 4.2.2 Question 2

Question 2, seen in Figure 27, asked participants if use of the application to perform deliveries was quicker than the method they used before of manually entering the address into a GPS app and recording deliveries. The feedback shows that six of the seven participants found using the app was “Definitely quicker” than previous methods, with one of the seven finding it “Somewhat quicker”. The purpose of this application is to improve the efficiency of delivery drivers performing their delivery duties. The results from this question can undoubtedly be comprehended as positive, with the majority selecting that it was “Definitely” quicker. These results determine unanimously that the application increased the participants efficiency when performing deliveries.

### 4.2.3 Question 3

Question 3, seen in Figure 27 determines how accurate the application was when using the camera to scan receipts to acquire the postcode and address. Of the seven participants, only two found the accuracy “Extremely accurate”, four found it “Mostly accurate” and one person found it “Mostly Inaccurate”. Question 3 determines the accuracy of the chosen OCR technology that was defined through the literature review and developed in the execution stage. The question determines how difficult or easy it was for the participants to make use of the camera to capture the correct postcode and address. Despite one occurrence of a participant receiving a bad experience using the scanning function, the overall results can be described as positive, as six of the seven participants provided positive results.

A picture containing graphical user interface

Description automatically generated

Figure 29 - Post-Test Questionnaire Part 2

### 4.2.4 Question 4

Question 4 seen in Figure 28, determines how often the correct data was passed to Google Maps. Two of the seven participants found that it passed the right information “Every time”, four found that it was “Most of the time”, and one person found that it only happened “Some of the time”. Question 4 determines the accuracy in which the application was able to pass the information captured after processing, to the Google Maps app. As six of the seven responses returned positive, with one neutral and no negative results, this can be deemed as a successful part of the application.

### 4.2.5 Question 5

Question 5, seen in Figure 28 allows the participants to write feedback regarding what worked well for them whilst using the app. Question 6, also seen in Figure 28, allows the participants to leave feedback about what features they think would improve the app further.

Table

Description automatically generated

Figure 30 - Post-Questionnaire Question 5 Results

Figure 29 displays all results of question 5 in the Post-Testing questionnaire. With these results, it can be determined that the features that worked well, according to the participants, was the camera scanning function, the use of Google Maps, and the recording and viewing of deliveries. The feedback from participants one and four specifically mention the accuracy of the camera function. This provides evidence that the camera successfully managed to capture the correct information and helped improve the driver’s efficiency. Whilst the other participants do not mention the camera feature specifically, participant two and seven mention the ease of use of the app. By cross referencing the feedback of these two participants with the results in question 3, seen in Figure 27, it can be determined that the users also were able to use the camera successfully and accurately as they were able to view their deliveries at the end of their shift.

As well as displaying results of the camera functioning successfully, Figure 29 also displays that participants liked being able to view their completed deliveries. As seen by responses three, five and six in Figure 29, the participants found it helpful to be able to view their deliveries at the end of their shifts. Whilst the main feature of the application was the document scanning camera, the overall goal of the project was to increase the efficiency of the participants while performing deliveries. When combining the feedback from Figure 29, a conclusion can be drawn that the application was successful in targeting the problem described in the project proposal. The app is mostly accurate when scanning receipts, it integrated well with Google Maps, and it was easy to use. The positive results gathered from questionnaire allow the conclusion to be made that use of the application improves the speed at which drivers perform deliveries.

### Table Description automatically generated4.2.6 Question 6

Figure 31 - Post Questionnaire Question 6 Results.



Figure 32 - Participant 7's Error Receipt.

As well the survey gathering information about which features worked well within the app, question 6 seen in Figure 30 also gathered the participants thoughts on how the application could be improved. As seen in Figure 30, participant one indicated that an alternative GPS app such as Waze may help improve functionality. As previously mentioned in the literature review, Waze is another popular navigation app developed by Google. If the app were to be improved later, Waze could be implemented to improve efficiency of the application. The other issue raised by participants was the camera failing to pick out the correct address at times. In Figure 30, participant seven mentions a particular problem with the address “Hawkhead Road”, the receipt in question can be seen in Figure 31. Whilst an ideal situation would require the receipt to be clear and uncreased, it may not be possible every time in a practical setting. A possible solution may be to use a different OCR technology or even alter the way in which the image is processed. As stated by the participants however, this occurrence was not frequent after one full week of use. Whilst this problem may result in loss of data or even slow the delivery process, it does not appear frequently enough to cause severe disruption. Should development proceed further, the issue would have to be resolved.

## 4.3 Discussion of Results

The evaluation method carried out for this project included finding participants who would be willing to use the application for the duration of one week, and then carrying out a survey in the form of a questionnaire. This was carried out successfully with seven participants. The details gained from the survey allowed the developer to reach a conclusion regarding the topic of this research paper. The topic being “Can a mobile application implementing Optical Character Recognition technology, successfully streamline the process of performing deliveries for a Pizza Hut delivery driver?”.

To summarise, the feedback from the seven participants was positive. Most participants felt that the camera function was successful with statistics showing that 86% of participants found it accurate. The findings also display that the implementation of Google Maps was efficient, and integrated seamlessly with the app. Lastly, the participants found the app quick and easy to use, spending less time manually writing information regarding deliveries, which resulted in improved delivery efficiency.

## 4.4 Ethical and Professional Issues

Adhering to Glasgow Caledonian Universities ethical guidance regarding this report, an ethics approval was required to be completed if necessary. As this project required individuals to test the application, the submission of an ethical form was required. This form ensured that no participants data would be compromised as a result of performing testing and completing the questionnaire. The form can be found in the appendix.

As the application created does not require any information to be uploaded to the internet, all files are stored locally on the user’s mobile phone. Prior to creation of this app, Pizza Hut drivers would already have access to customer information with no additional information being provided through use of the app. Thus, it can be determined that no ethical or professional issues would be violated through use of the app. Data will be kept a maximum of two months to allow drivers to view the number of deliveries they have made for the previous month to compare figures in relation to their wage. Data would be automatically deleted after this two-month period. This would comply with the Data Protection Act 2018 (GOV.UK 2018), specifically regarding the retention of data for as long as necessary.

# 5.Conclusions and Further Work

## 5.1 Conclusion

This project was carried out to determine if the use of a mobile application would be capable of improving Pizza Hut driver’s efficiency whilst performing their delivery duties. Before using this application, drivers would enter information from a receipt into a GPS app of their choosing such as Google Maps or Waze manually. Some drivers would then record delivery information in a notebook or separate app. This process was time consuming and would cause delays when delivering. The purpose of the application was to improve the delivery model by eliminating the manual entry of postcodes and automatically record delivery information, thus improving the efficiency of deliveries.

A detailed literature and technology review was then carried out to determine if the application was possible or had been created before. The literature review was carried out into current mobile development methods and mobile technology, Optical Character Recognition software and methods, and lastly, GPS technologies and how they could be integrated when developing an app. The results from this review allowed the developer to learn and improve their understanding of the discussed technologies and influenced the methods that would be used to develop the app.

Before development began, several Pizza Hut drivers were asked if they would participate in testing the app. The application was then developed over the course of several weeks using Java and Android Studio. Before the participants used the application, they were asked to fill in a pre-test questionnaire that would give the developer insight into how they performed their delivery duties. After the application was developed, it was then tested by the participants over the course of one week where they would use the app during their shifts. After the week of testing, each participant was given a post-test questionnaire to complete, providing feedback on use of the app.

The post-test questionnaire results were analysed to determine if the application was successful in improving the efficiency of drivers and if it was beneficial. The results from the post-test questionnaire were largely positive and successful, with only a small margin of error when using the applications camera scanning feature. As a result of using the app, it was identified that the participants found the feature of automatically recording deliveries and being able to view them afterwards extremely useful.

In summary, the problem identified in the problem statement and hypothesis was addressed successfully through the creation of a mobile application. The application utilises the mobile phones camera to capture an image containing the customers information, filters the information to a postcode and street address, passes this information to the Google Maps app, then record the information for the user to view. Whilst not all secondary features detailed in the project outline were implemented, given more time they could be added to provide further functionality. After a testing period of one week, the test participants found that their overall delivery speed and efficiency was improved, satisfying the project proposal.

## 5.2 Further Work

While the application can be considered successful given the feedback of test participants, there is still room for improvement. Whilst using the camera does increase efficiency compared to manually entering the address and recording information, there is the possibility of further increasing performance of the app. One example may be an auto-complete function in place of the camera function. Instead of using the phones camera to scan the receipt, the user would begin to manually enter the information and the app would suggest the most likely address based on location and the text entered. The app would still record delivery information, however, entering the address into an auto complete may save time by reducing the risk of incorrect information being identified due to the inaccuracy of a camera. This could also allow the application to be used by delivery drivers outside Paisley or even non-Pizza Hut drivers.

Another feature which could be added as stated by one of the test participants, would be to add an alternative GPS navigation method. As stated by participant number one in Figure 30, there are some drivers who prefer using Waze rather than Google Maps. Adding another option for GPS may improve efficiency of the drivers who are more comfortable using a different GPS app.

As well as adding additional features to the app to increase efficiency, there is also the possibility of including more users by developing the application on iOS for iPhone users. As mentioned in the testing stage, there was a test participant who had to use a spare Android device to carry out the testing. This would mean using a phone the user is unfamiliar with, possibly causing disruption when using the service. Developing an iOS version of the application would allow a more inclusive user base and provide the service for iPhone users as well as Android. This would only be possible if the developer had access to the required hardware and software for developing in iOS.

# 6. References

FOOD AND DRINK FEDERATION, 2020*. Our World at a Glance* [online]. *Food And Drink Federation.* [viewed October 15, 2020]. Available from*:* <https://www.fdf.org.uk/statsataglance.aspx>

GOV.SCOT, 2021. *Food And drink* [online] *Policies* [viewed October 18, 2020]. Available from: https://www.gov.scot/policies/food-and-drink/#:~:text=Scotland%20has%2018%2C850%20food%20and,sector%20in%20our%20Economic%20Strategy.

MALLEY, K., 2019. *Evolution of the booking food delivery market* [online]. AHDB. [Viewed on: Mar 20, 2020]. Available from:<https://ahdb.org.uk/news/consumer-insight-evolution-of-the-booming-food-delivery-market>

GILSENAN, K., 2018. *5 Key Insights into Takeaway Food Delivery Apps and Services* [online]. Globalwebindex. [viewed October 15, 2020]. Available from: <https://blog.globalwebindex.com/trends/takeaway-food-delivery/>.

SINGH, A., 2019. *Top 10 Successful Online Food Delivery Apps in the World* [online]. Netsolutions. [viewed Apr 6, 2020]. Available from: <https://www.netsolutions.com/insights/top-10-successful-food-delivery-apps-in-the-world/>.

BUSINESS MATTERS, 2018. *Top 5 best food delivery apps in UK* [online]. bmmagazaine [viewed October 16, 2020]. Available from: https://bmmagazine.co.uk/in-business/top-5-best-food-delivery-apps-in-uk/

LUTY, J., 2019. *Just Eat: number of orders in the UK 2013-2018* [online]. Statista. [viewed Mar 27, 2020]. Available from: <https://www.statista.com/statistics/579289/just-eat-number-of-orders-united-kingdom-uk/>.

SIMILARWEB. 2020.*just-eat.co.uk Traffic Statistics* [online]. SimilarWeb. [viewed Apr 8, 2020]. Available from: <http://similarweb.com/website/just-eat.co.uk>.

RAHMAN, F., TAWAB, E, S., 2019. *App Development for the Social Good: Teaching a Socially Conscious Mobile App Development in an Upper-Level Computer Science Course* [Viewed June 20, 2020]. Available from: <https://www.researchgate.net/publication/334729332_App_Development_for_the_Social_Good_Teaching_a_Socially_Conscious_Mobile_App_Development_in_an_Upper-Level_Computer_Science_Course>.

EL BAHI, H. & ZATNI, A., 2019. *Text recognition in document images obtained by a smartphone based on deep convolutional and recurrent neural network. Multimedia Tools and Applications.* [Viewed November 14, 2020]. Available from: <https://www.readcube.com/articles/10.1007/s11042-019-07855-z>.

GOV.UK., 2020.*Guidance for schools, childcare providers, colleges, and local authorities in England on maintaining educational provision* [online]. Gov.uk*.*[viewed Apr 11, 2020]. Available from: <https://www.gov.uk/government/publications/coronavirus-covid-19-maintaining-educational-provision/guidance-for-schools-colleges-and-local-authorities-on-maintaining-educational-provision>.

SOUTHEY, F., March 19, 2020. *Online food delivery one of the only winners in coronavirus outbreak* [online]. FoodNavigator. [Viewed Aug 27, 2020]. Available from: <https://www.foodnavigator.com/Article/2020/03/19/Online-food-delivery-one-of-the-only-winners-in-coronavirus-outbreak>.

SHINDE, A, A., CHOUGULE, D, G., January 2012. *Text Pre-processing and Text Segmentation for OCR* [online]. International Journal of Computer Science Engineering and Technology, pp. 810- 812. [Viewed Aug 29, 2020]. Available from: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.414.3478&rep=rep1&type=pdf

MITHE, R. INDALKAR, S. DIVEKAR, N., March 2013. *Optical Character Recognition*. [viewed November 20, 2020]. Available from: <https://www.ijrte.org/wp-content/uploads/papers/v2i1/A0504032113.pdf>.

STATISTA., 2020. *Number of smartphone users from 2016 to 2021* [online]. Statista. [viewed Oct 31, 2020]. Available from: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>.

BAPAT, S., 2018*. Understanding the factors contributing to the awareness, usage and popularity of Mobile Apps - ProQuest.* [viewed Nov 6, 2020]. Available from: <https://www.proquest.com/docview/2090302793/fulltext/5DD15DE570F64117PQ/2?accountid=15977>.

OSTDICK, N., 2016. *Pros and Cons of Mobile Travel Apps* [online]. Travel Tech Insights. [viewed Nov 6, 2020]. Available from: <http://www.dcsplus.net/blog/3-pros-and-cons-of-mobile-travel-apps>.

JEFFREY, B., BAGALA, M., CREIGHTON, A., LEAVEY, T., NICHOLLS, S., WOOD, C., LONGMAN, J., BARKER, J., PIT, S., 2019.*Mobile phone applications and their use in the self-management of Type 2 Diabetes Mellitus: a qualitative study among app users and non-app users - ProQuest*. [viewed Nov 6, 2020]. Available from: <https://www.proquest.com/docview/2306781453/176895DDE43A4847PQ/2?accountid=15977>.

BRINGOZ TEAM., 2020. *How Do Driver Apps Increase Driver Efficiency* [online]. Bringoz.com [viewed Nov 9, 2020]. Available from: <https://www.bringoz.com/how-do-driver-apps-increase-driver-efficiency/>.

CASTIGNANI, G., DERRMANN, T., FRANK, R. & ENGEL, T., 2015. Driver Behavior Profiling Using Smartphones: A Low-Cost Platform for Driver Monitoring*. IEEE Intelligent Transportation Systems Magazine.***7**(1), pp.91-102. [viewed Apr 8, 2021]. Available from: 10.1109/MITS.2014.2328673.

MCNABB, J.C. 2017. *Warning a Distracted Driver: Smart Phone Applications, Informative Warnings and Automated Driving Take-Over Requests.*Available from: <https://www.proquest.com/docview/1899468463/abstract/24E35E9C18E54926PQ/1>.

JAKIMOSKI, K. & LAZARESKA, L., 2017. Analysis of the Advantages and Disadvantages of Android and iOS Systems and Converting Applications from Android to iOS Platform and Vice Versa*. American Journal of Software Engineering and Applications.***6**, pp.116-120. [viewed Apr 8, 2021]. Available from: 10.11648/j.ajsea.20170605.11.

FLORA, K., HARLEEN, WANG, X. CHANDE, V., SWATI., 2014. *An Investigation into Mobile Application Development Processes: Challenges and Best Practices - ProQuest*. [viewed Nov 21, 2020]. Available from: <https://www.proquest.com/docview/1627736042/D68452F411CB4BBBPQ/7?accountid=15977>.

NANDHINI, R., 2018. *Android and its background developments. International Journal of Advanced Research in Computer Science, vol.9*. [Viewed November 20, 2020]. Available from: <https://www.proquest.com/docview/2101243438/9132BE6F558B4A97PQ/1?accountid=15977>.

APPLE, 2021. *Documentation Archive* [online]. Getting Started. [viewed April 2, 2021]. Available from: <https://developer.apple.com/library/archive/referencelibrary/GettingStarted/DevelopiOSAppsSwift/>

ABDULLAH, M., 2020*. 10 best IDE For Android App Development in 2020* [online]. Inlelvue [viewed March 28 2021]. Available from: <https://www.intelvue.com/10-best-ide-for-android-app-development/>

HINDY, J., 2020. *10 best GPS apps and navigation apps for Android* [online]. Android Authority. [viewed Nov 19, 2020]. Available from: <https://www.androidauthority.com/best-gps-app-and-navigation-app-for-android-357870/>.

REID, E., 2020*. A look back at 15 years of mapping the world* [online]. Google.com. [Viewed November 19, 2020]. Available from: <https://blog.google/products/maps/look-back-15-years-mapping-world/#:~:text=1.,over%20the%20world%20every%20month.&text=2>.

EADICICCO, L., 2020. *19 incredibly useful Google Maps features everyone should know about* [online]. Business Insider. [Viewed on November 17, 2020]. Available from: <https://www.businessinsider.com/google-maps-best-features-tips-tricks-2019-5?r=US&IR=T>.

GOOGLE., 2021*Pricing & Plans | Google Maps Platform.*[viewed Apr 7, 2021]. Available from: <https://cloud.google.com/maps-platform/pricing>.

LAIS, S., 2002. *Optical character recognition - ProQuest.* [viewed Nov 13, 2020]. Available from: <https://www.proquest.com/docview/216090155/fulltextPDF/F01CF5155BAE4B0BPQ/1?accountid=15977>.

YILMAZTURK, D., GURBAK, A, E., 2019. *Geometric Evaluation of Mobile-Phone Camera Images for 3D Information* [online]. International Journal of Optics. [viewed March 12, 2021]. Available from: https://www.hindawi.com/journals/ijo/2019/8561380/

ITSHOWCASE., 2016. *Why OCR (optical character recognition) is revolutionising businesses all over | Business Software Showcase* [online]. Itshowcase.co.uk. [viewed Nov 13, 2020]. Available from: <https://www.itshowcase.co.uk/why-ocr-optical-character-recognition-is-revolutionising-businesses-all-over/>.

INTUIT QUICKBOOKS., 2021. *Maximise your tax savings*. [online]. Quickbooks. [viewed February 19, 2021]. Available from: https://quickbooks.intuit.com/uk/

CAMSCANNER., 20201*CamScanner Features | Turn your phone and tablet into scanner for intelligent document management.*[viewed Mar 16, 2021]. Available from: [https://v3.camscanner.com](https://v3.camscanner.com/).

HAN, T., HICKMAN, A., 2019. *Our Search for the Best OCR Tool, and What We Found* [online]. OpenNews. [viewed Nov 13, 2020]. Available from: <https://source.opennews.org/articles/so-many-ocr-options/>.

IAKOVLEV,M., 2020*The Battle of the OCR Engines - Tesseract vs Google Vision | Blog | Fuzzy Labs.*[viewed Apr 7, 2021]. Available from: <https://fuzzylabs.ai/blog/the-battle-of-the-ocr-engines/>.

ALAM,A.A., 2019*Firebase ML Lesson 01: Recognize Text and Label in Image using ML-Kit in Android🔥📱.*[viewed Apr 9, 2021]. Available from: <https://proandroiddev.com/firebase-ml-lesson-01-recognize-text-and-label-in-image-using-ml-kit-in-android-7aa9aef45cde>.

S, SHYLESH., 2017. *A Study of Software Development Life Cycle Process Models.* [viewed January 27 2021]. Available from: https://papers.ssrn.com/sol3/papers.cfm?abstract\_id=2988291

BASEER, K, K., REDDY, M, R, A., BINDU, S, C., 2015. *A Systematic Survey on Waterfall Vs. Agile Vs. Lean Process Paradigms - ProQuest*. [viewed Nov 18, 2020]. Available from: <https://www.proquest.com/docview/1728276011/A0106785E13C43BCPQ/1?accountid=15977>.

NAWAZ, Z., AFTAB, S., ANWER, F., 2017. *Simplified FDD Process Model. International Journal of Modern Education and Computer Science. 9, pp.53-59*. [viewed Nov 20, 2020]. Available from: 10.5815/ijmecs.2017.09.06.

SHOKUROVA, K., 2020. *13 basic mobile UI patterns to know about.* [viewed March 6, 2020]. Available from: <https://uxdesign.cc/mobile-ui-13-basic-patterns-of-app-ui-design-to-know-about-d3f7c6176f13>

GOOGLE., 2021. *Google Java Style* Guide [viewed March 10, 2021]. Available from: <https://google.github.io/styleguide/javaguide.html>

FOX, N., MATHERS, N.& HUNN, A., 2000. *Surveys and Questionnaires*. [viewed April 3, 2021]. Available from: https://www.researchgate.net/publication/270684903\_Surveys\_and\_Questionnaires

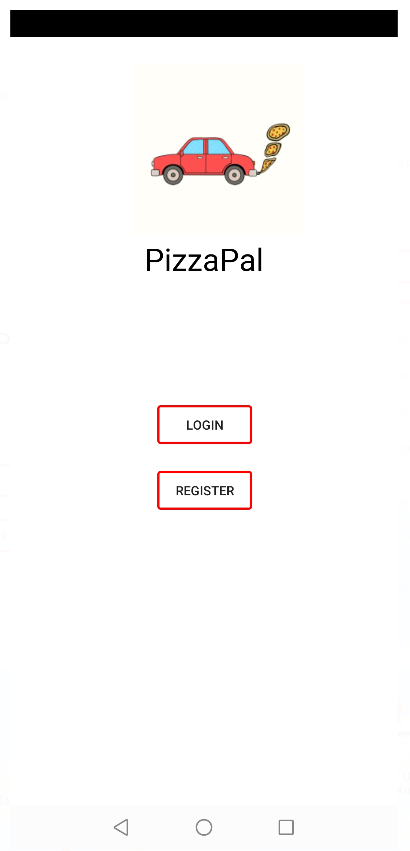
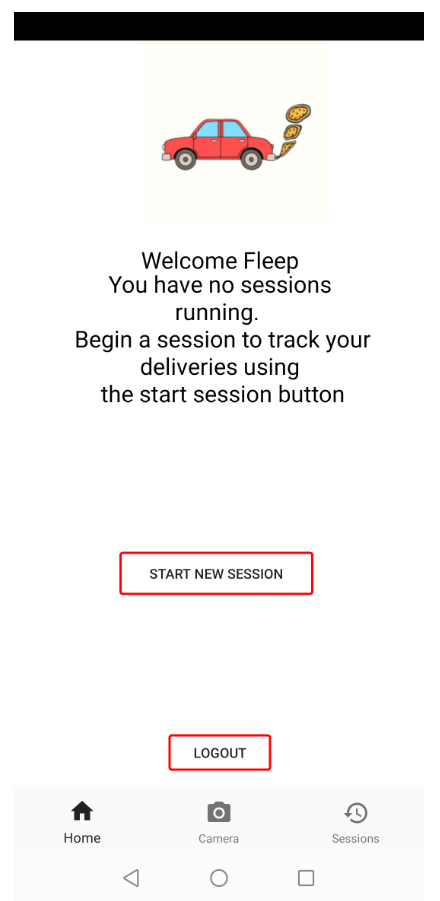
GOV.UK., 2018*Data Protection Act 2018.*[viewed Apr 11, 2021]. Available from: <https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted>.

# 7.Appendices

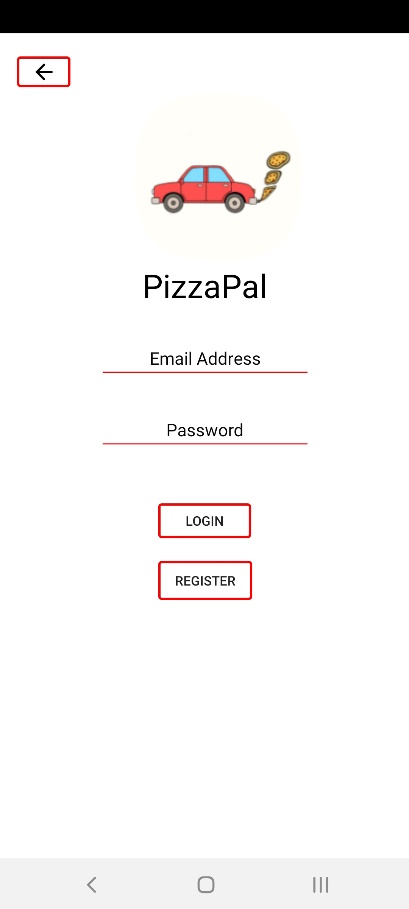
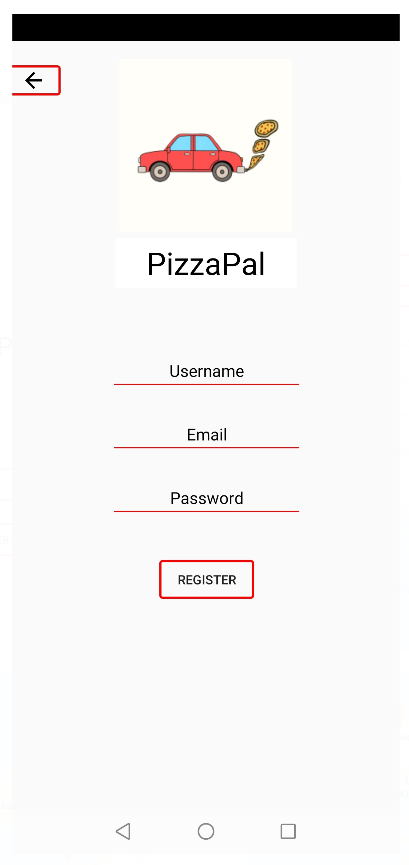
Appendix A – OCR Example



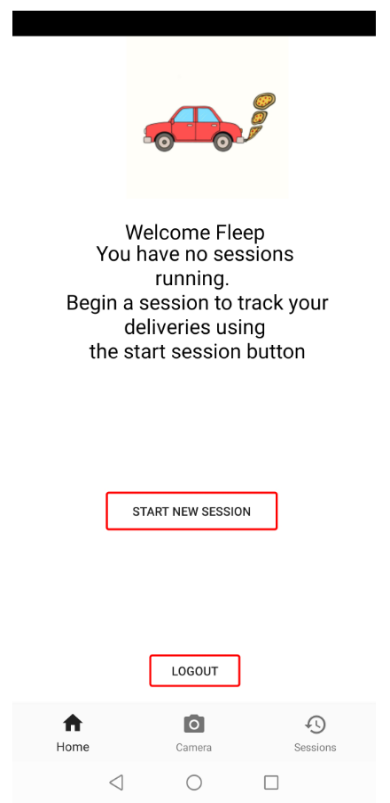
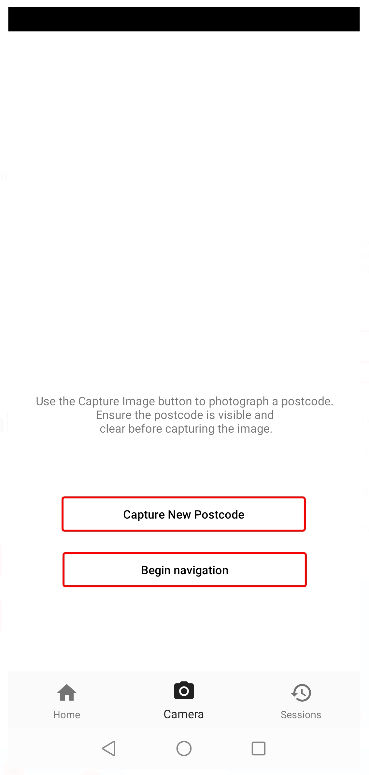
Appendix B – Application Screenshots

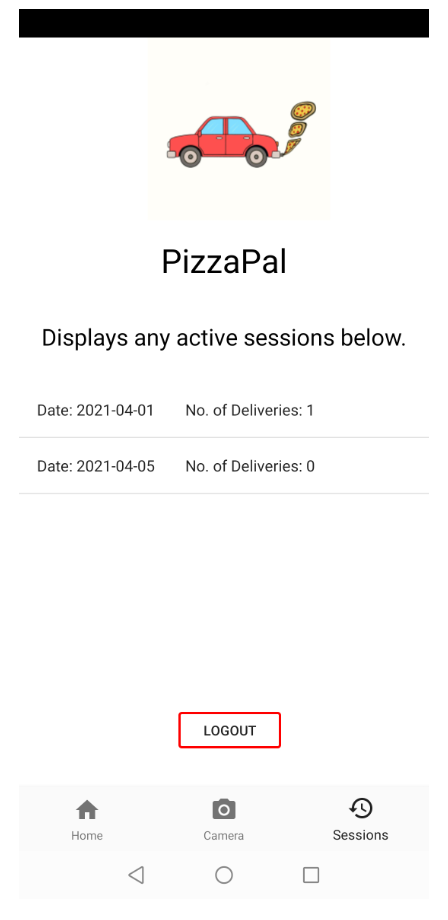
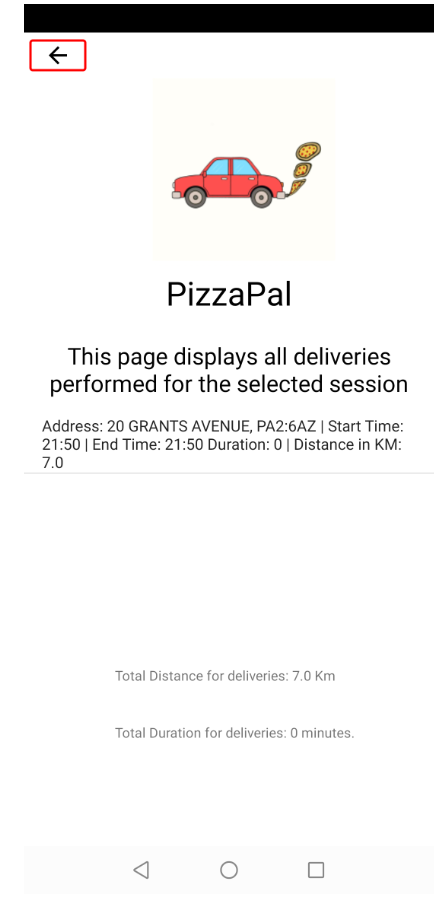
Landing Screen Home Fragment

Login Screen Registration Screen

Home Fragment Camera Fragment

Session Fragment Session Info activity

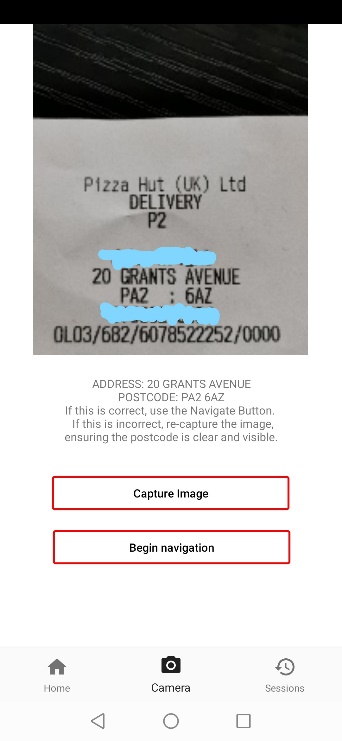
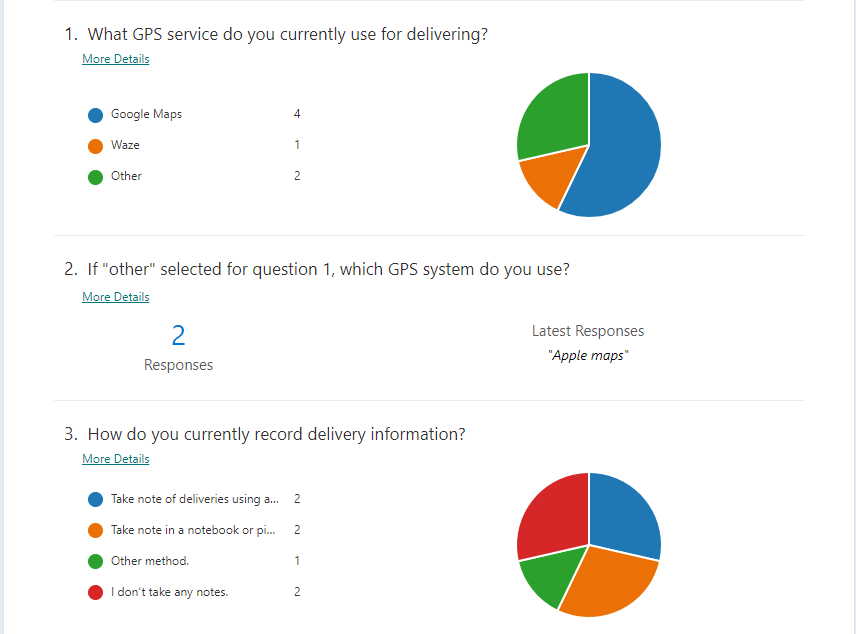
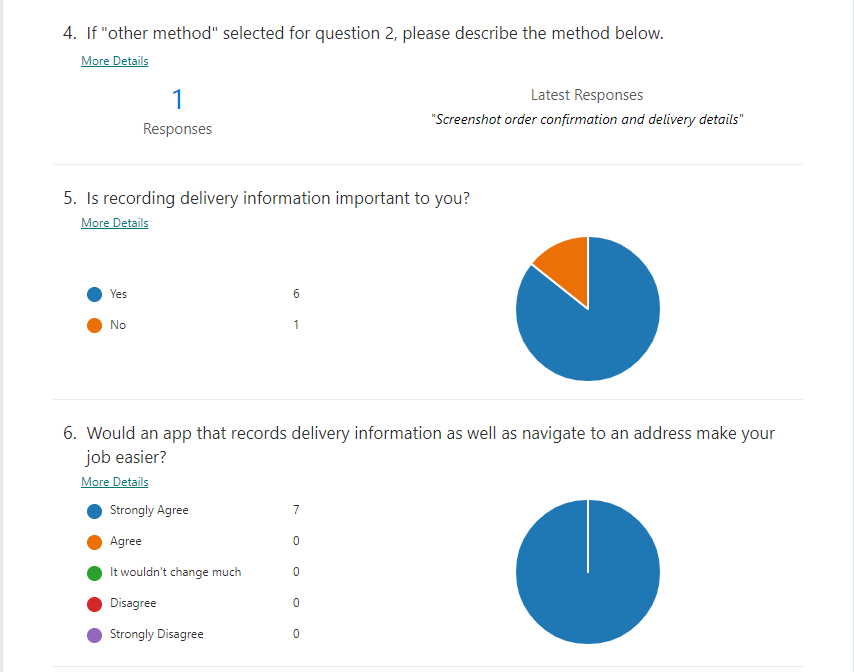


Image capture example

Appendix C – Pre-Test Questionnaire File and Images

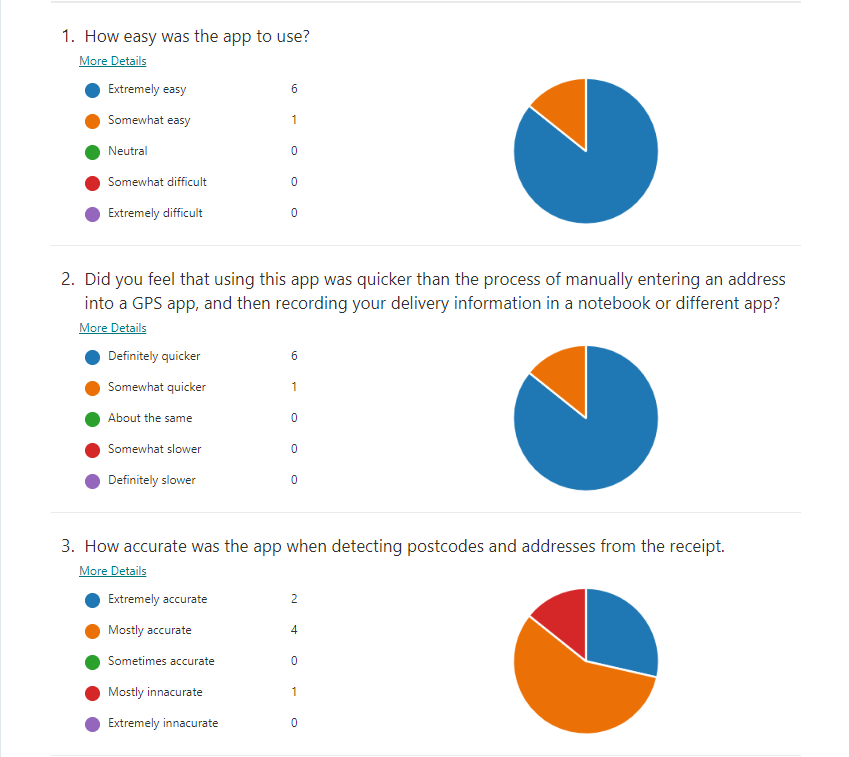
GitHub Questionnaire Link: <https://github.com/S1916188/PhilippeCranskens_S1916118_Honours_Project_Delivery_App/blob/main/Ethics%20Form%20-%202021-s1916118-%20PhilippeCranskens.docx>

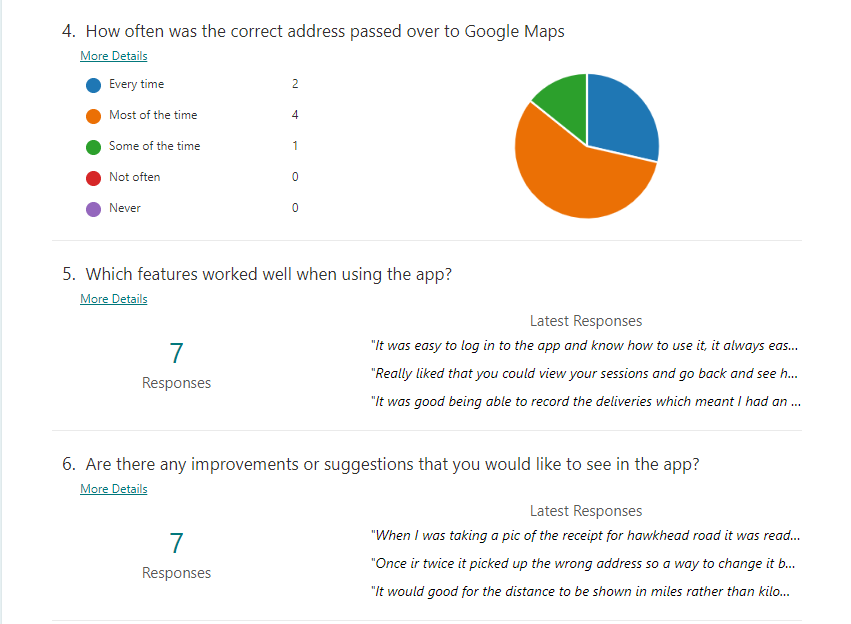
First half of pre-test questionnaire

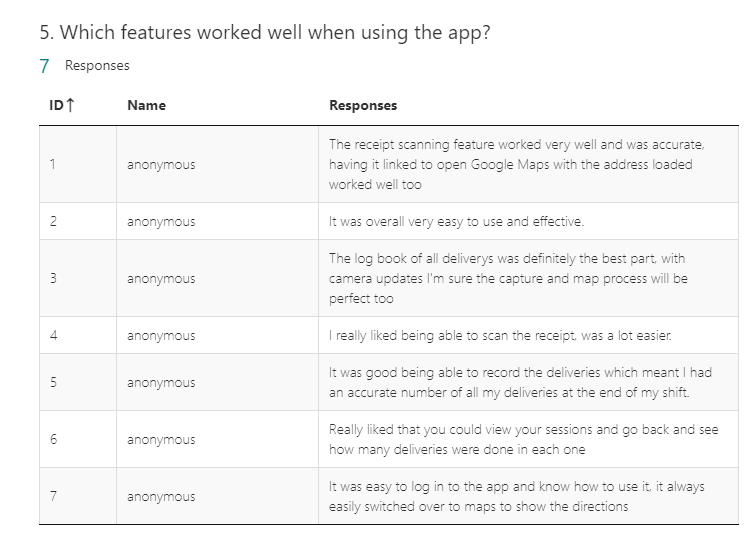
Second half of pre-test questionnaire

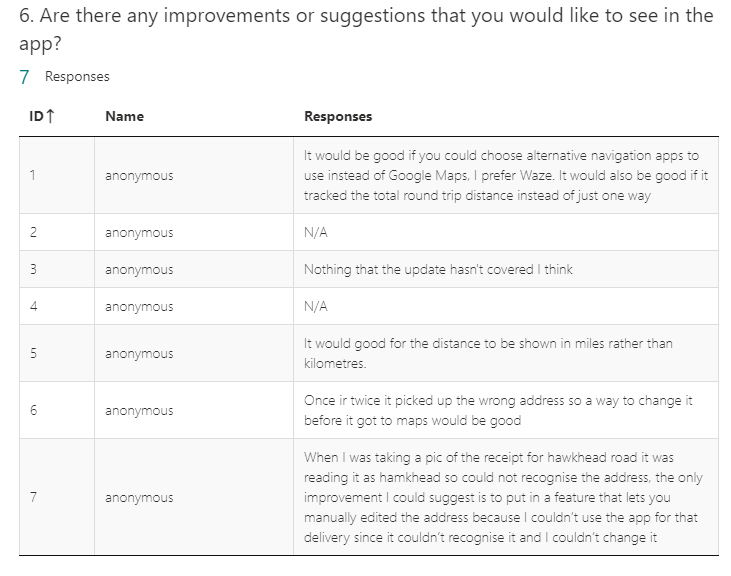
Appendix D – Post-Test Questionnaire File and Images

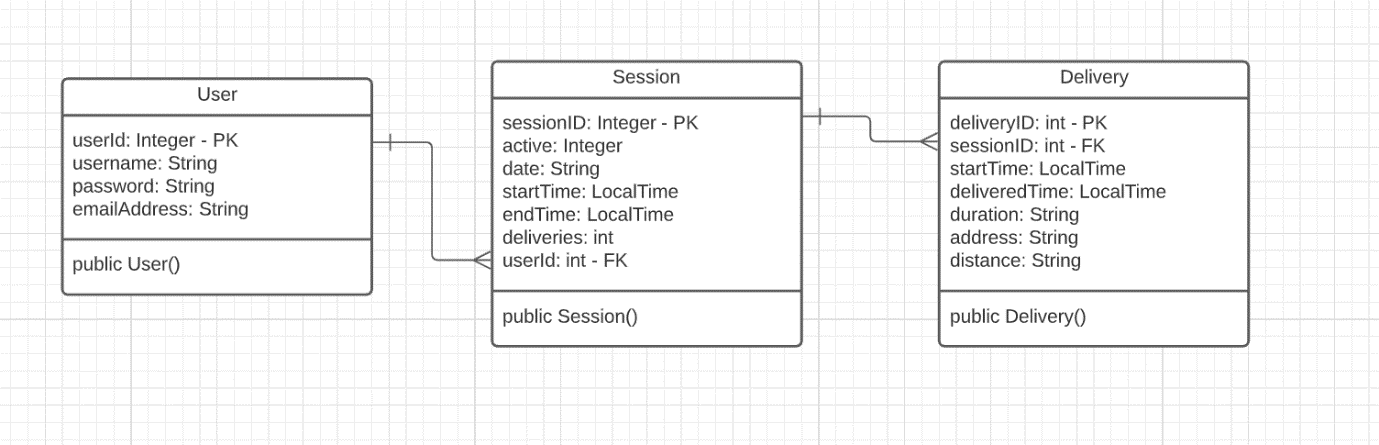
GitHub Questionnaire Link: <https://github.com/S1916188/PhilippeCranskens_S1916118_Honours_Project_Delivery_App/blob/main/Post-Testing%20Questionnaire(1-6).xlsx>

First half of post-test questionnaire

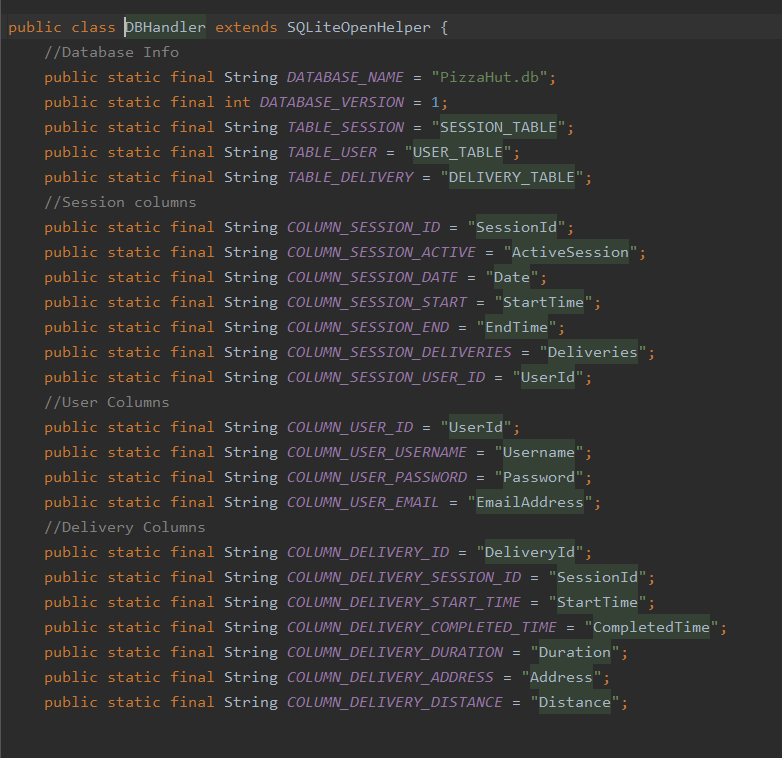
Second half of post-test questionnaire

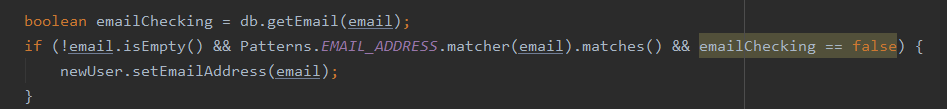
Post-test questionnaire question 5 results

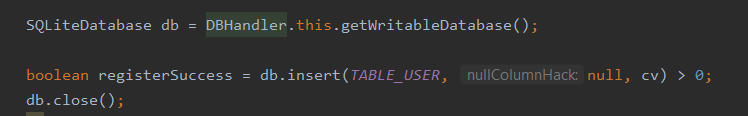
 Post-test questionnaire question 6 results

Appendix E – Class Diagram

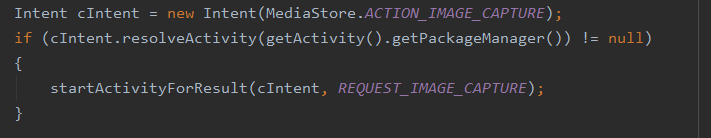
Appendix F – Code Examples

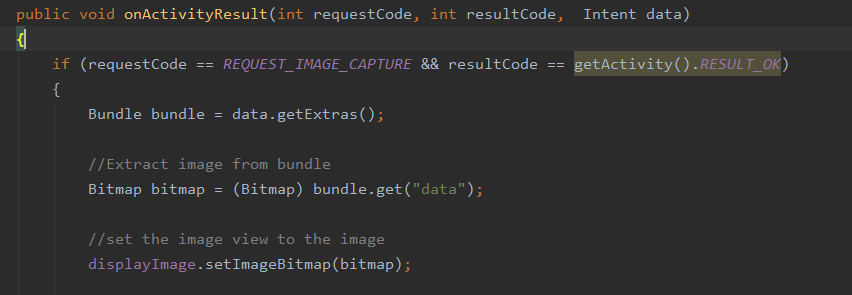
Creating database

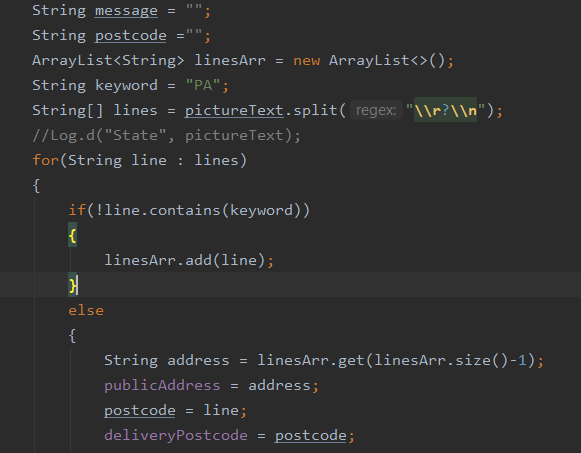
Checking email pattern

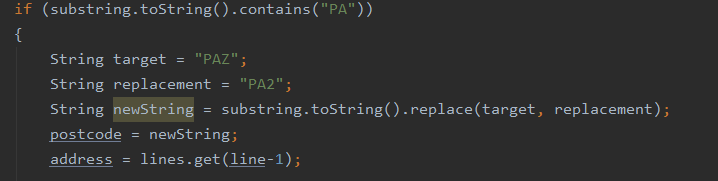
Register user in database.

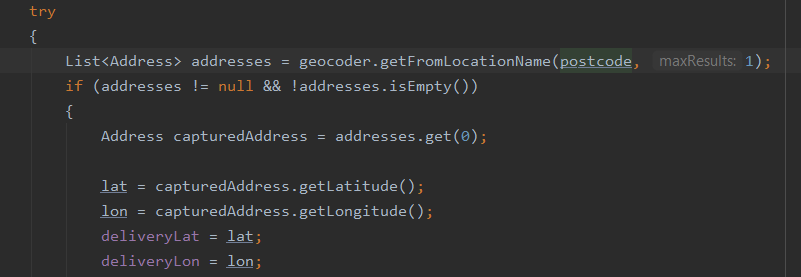
Home page code to select fragment

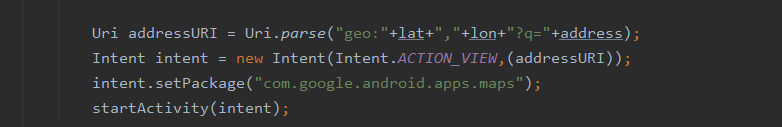
Start Camera Code

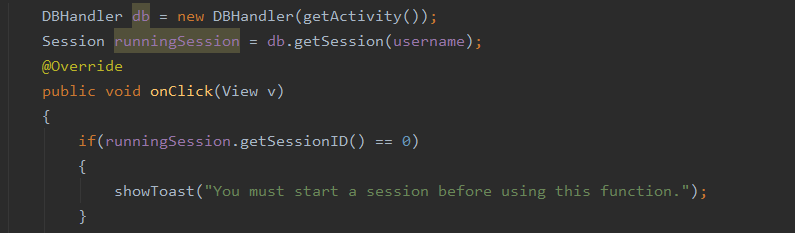
Convert image to BitMap code

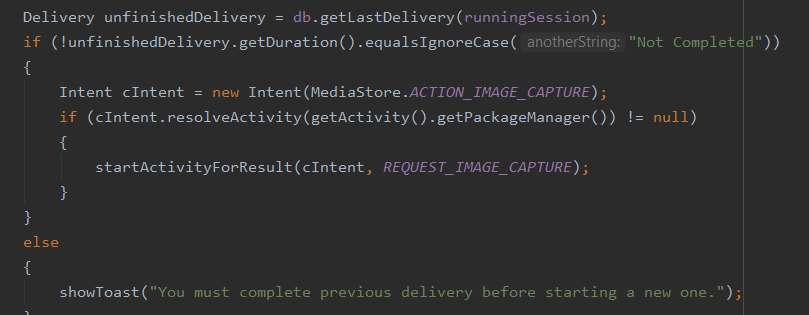
Identifying postcode code

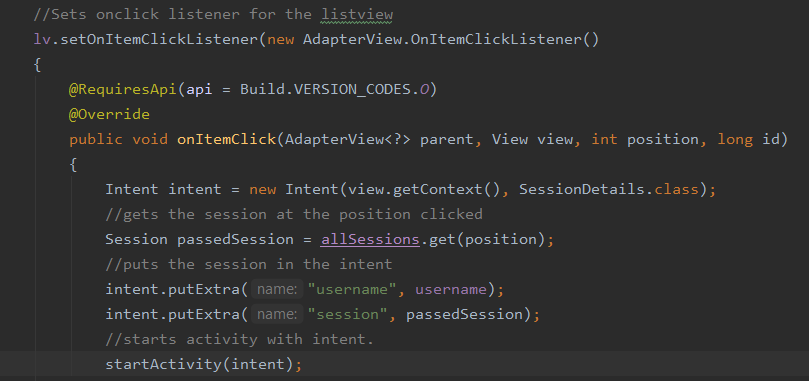
String replacement code

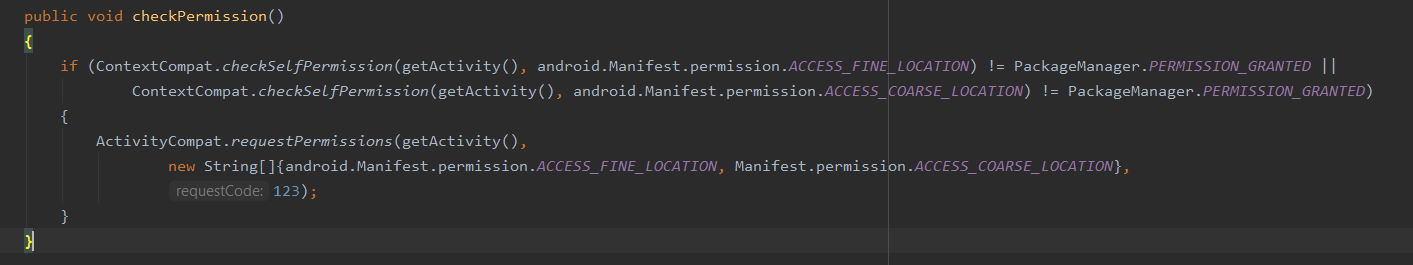
Finding latitude and longitude code

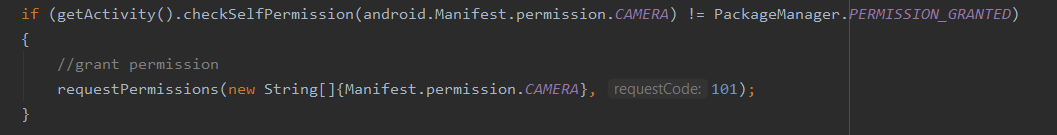
Passing latitude and longitude to Maps

Check user session code

Checking last delivery code

Passing Session details from fragment to session info activity

Requesting fine and course permissions code

Requesting camera permissions code

Appendix G – Receipt Examples

Test participant problematic receipt

Typical customer order receipt with customer information redacted.

Appendix H – Code Listing

GitHub repository: <https://github.com/S1916188/PhilippeCranskens_S1916118_Honours_Project_Delivery_App>

App download link on GitHub:

<https://github.com/S1916188/PhilippeCranskens_S1916118_Honours_Project_Delivery_App/blob/main/app-debug.apk>

Full Project on Google Drive: https://drive.google.com/drive/u/1/folders/1zGIiZWprq5B4kEK-lsZbMEhraoessZmI

Appendix I – Ethics Form

<https://github.com/S1916188/PhilippeCranskens_S1916118_Honours_Project_Delivery_App/blob/main/Ethics%20Form%20-%202021-s1916118-%20PhilippeCranskens.docx>