

Table of Contents

1. Introduction	2
1.1 Overview	2
1.2 Business Context.....	2
1.3 Glossary	3
2. General Description	3
2.1 Product / System Functions	3
2.2 User Characteristics and Objectives	4
2.3 Operational Scenarios	4
2.4 Constraints	5
3. Functional Requirements	6
3.1 Registering a new Account/Logging in with an Account	6
3.2 Receipt Scan using OCR.....	6
3.3 Receipt Scan using QR Code	7
3.4 Viewing Receipts	7
3.5 Expense Tracker	8
4. System Architecture	9
5. High Level Design	10
5.1 High-Level Design Diagram	10
5.2 Use Case Diagrams:	11
5.3 Data Flow Diagram	17
6. Preliminary Schedule	18
7. Appendices	18

1. Introduction

1.1 Overview

ReceiptLess is an application built for iOS/Android mobile devices which will help users in organizing their expenses and storing their receipts in a more convenient way than tightly packed in their purses/wallets. It will also help shops and consumers reduce the use of paper by going fully digital in the way they receive receipts.

With this information the consumer can also track their own expenses from these receipts and get a more in depth view and where, when and what they are spending their money on. Each user will have their own account which will be registered with their email and password.

Users will have access to the 2 scanning options. Scan the physical receipt, or if the option is available within the retail store to scan a QR code provided in the shop, which holds the necessary data of the receipt. The necessary information is then saved within the users account. This data will then be used to format our digital receipt. Users can categorize their receipts into groupings such as Business, Personal (Clothing, Food, Travel).

An expense tracker is also available which will allow the user to track their expenses. This gathers information from the receipts on the account as well as allows for the users own input. Finally a subscription reminder will be available to the user, where they may set up a reminder for when any subscription payment will be taken out. This information will be used in conjunction with the expense tracker to add into their weekly/monthly expenses.

1.2 Business Context

A big selling point of ReceiptLess is of course the aimed reduction in paper. The environment and going green are a very hot topic now and receipts are something everyone has used and something which is creating a large amount of waste in the world. According to Green America, "over 12.4 million trees and 13 billion gallons of water are consumed each year in the creation of paper receipts, generating 1.5 billion pounds of waste and 4 billion pounds of CO2." (<https://hackernoon.com/digital-receipts-in-retail-b415fbdfde3f>)

As well as this, the app will add a more convenient approach to storing receipts safely and securely. For many people a big problem with receipts is that they lose them frequently or forget where they put them and as a consequence can't return any items purchased or track their spending habits. Also seeing as it's the consumer scanning a barcode they are not giving out their email to the shop, thus avoiding any spam marketing emails and reducing the turnaround for each transaction making check outs more efficient. The use of the expiry date reminder is also perfect as many consumers forget when they need to return an item and miss the expiry date, therefore most of the time, resulting in their right to a refund being denied. Our user demographic is all consumers that own a mobile device, which is a very wide bracket, but we feel receipts are a part of everyday life at this stage regardless of it is the weekly grocery shop or the purchase of a new pair of shoes. They are used on a daily basis with any purchases resulting in an abundance of paper wasted with purchases that could be electronically recorded through the use of Receipt-Less. The USP of ReceiptLess is our implementation of the QR code. This gives both the user and the store the choice to

avoid a paper receipt altogether and instead scan a QR code in participating shops allowing for a quick and easy process for acquiring a digital receipt. This also benefits the store in the fact it will reduce the use of till receipt paper for the shop i.e. reduce costs, as well as ensuring convenience for the consumer.

1.3 Glossary

- OCR - Optical Character Recognition.
- QR code - Quick Response Code.
- USP - Unique Selling Point.
- ePOS - Point of Sale
- API - Application Programming Interface

2. General Description

2.1 Product / System Functions

User Sign-Up

User will first download the application from the app/android store. Once the app is downloaded correctly, the user will be prompted to sign up/sign in. Within signup the user will be asked for a valid email and also for a password. The user's information will be stored in an external database. This will allow user to access their account from any mobile.

User Login

Once a user has successfully signed up with a valid email address, the user will log in their unique email address and password. If the user's details are not correct they will be given access.

Scanning a QR Code

If the stores ePOS system uses a QR code system for receipts, the app user will open the scan screen and scan the QR code. The data from the QR code will then be formatted into a receipt on the app and necessary data will be stored in the database.

Scanning a Paper Receipt

If the user has a paper receipt, they will go to the scan screen and capture image of the receipt. OCR will then be performed on it and store the data in the database.

Viewing Receipts

After adding a receipt, the user will be able to see all receipts from their account. They have the option to edit the information manually as-well as removing that receipt.

Deleting Receipts

If a user wishes to delete receipt they can press the delete button to the right of the the stored receipt. This will remove all data of that receipt from the external database.

Expense Tracking Reports

Users can go onto the "Expense Tracking" page where they will see automated reports based on their expenses.

2.2 User Characteristics and Objectives

Overall, our aim for this application is to make the storing and collection of receipts for the consumer, more convenient. This why the expected user demographic for this application is all consumers that own a smartphone. The application is going to be very simplistic as the main aim is to help the user and make life a bit less stressful! We aim to make the app very accessible to all users, using techniques such as high colour contrast and large fonts. As this application is aimed for such a big demographic we need it to be minimalistic to allow users of all technical ability to be able to use it without any inconvenience The learning curve to using this application will be very minimal.

2.3 Operational Scenarios

Download/Installation

User will download the application from the App/Play Store. The application will be installed on the device.

Sign In/Sign-Up

User will create an account with an email and password. Will then be prompted to login. Any invalid credentials will result in the user to try again. A forgot password button will also be available if user forgets given password.

Home Page

Once signed in correctly, the user will be on the Home Page. This will contain a scan button that sends you to the scan page, and also a view receipts button and tracking button.

Scan Page

If the "Scan" button is clicked on the Home Page, this will send the user to the Scan Page, where they will choose to scan a QR code or physical receipt. Once Receipt/QR code is scanned the data will be processed and formatted into a digital receipt.

Receipts Page

Once the digital receipt is formatted the user will be able to view their receipts on the Receipts Page. This will contain all receipts on the account. There will be different options for the user such as editing the receipt data and also to remove the receipt from the account.

Expense Tracking Page

The user can also go to the Expense Tracker to see reports on their receipts. Within this page charts and totals will be displayed. A Subscription Tracker will also be put in place on this page for the user to input their subscription amounts and date.

Return Button

On each page there will be a Home Button to allow the user to get back to the Home page with ease. The home page contains the scan button.

2.4 Constraints

Hardware

As this application is aimed to be for both iOS and Android primarily the code will be run in Android Studio and Xcode. This will be developed on a Windows and Mac machine.

Time

The submission for this project is 06/03/20, so we will ensure that the project work is ready for production on this date. If we have additional time we will attempt to add additional features to the app.

User

As our user demographic is so big, we will have to ensure that we meet all kinds of user requirements, for example large fonts for users with poor vision.

Security

As we intend to store account information we must ensure that the data is secure. Using AWS Cognito, the users passwords will be encrypted to ensure safe and secure data storage.

3. Functional Requirements

3.1 Registering a new Account/Logging in with an Account

- **Description:**
For every user they must create an account before using the application. This will require basic information such as an email and password and is then saved to the database. After they register they can then proceed to login to the application and use the full functionality of it.
- **Criticality:**
This is a very critical aspect of the application as without this the user can not securely save their receipt to a specific account and if it fails cannot proceed to login and use the full functionality of the application.
- **Technical Issues**
It is very important that when the user attempts to create a new account that they can enter all details easily and all the data is securely saved into our database and no problems such as duplicate emails and non valid emails are saved to it.
- **Dependencies**
Without a fully functioning login system the whole application is inaccessible to the user.

3.2 Receipt Scan using OCR

- **Description**
Upon the user successfully logging in, they will be presented with 3 buttons, one of which will be a Scan button. After pressing it, they will then choose 1 of the 2 options, OCR or QR Code. If they choose the OCR option it will open the camera and allow the user to take a picture of a receipt. This will then run OCR on the image and attempt to read characters from the image. It will extract the necessary data such as merchant name, total etc. The user will then be prompted to see whether the data is correct. If data is missing the user can input this data manually. A save button is then finally pressed and the receipt is added to the account.
- **Criticality**
This is one of the main aspects of what this application will do. This is also a huge option for the user when the store does not have the ability to produce a QR code with the receipt data for the customer. Because of this, it is crucial for this to be at full working capabilities.
- **Technical Issues**
It is important that the OCR produces accurate enough image recognition and reads the receipt format as good as possible. The OCR should attempt to also brighten up the image as well as sharpen it and crop out the borders of the receipt itself for it to be able to read the image better. The output should then be parsed into the correct format for the user to review what it thinks are the receipts contents/data. The user should then have the option to make any necessary changes.

- **Dependencies**

The OCR will rely on many aspects such as the quality of the users camera, the lighting it is taken in and the Tesseract library and how well it translates the image to text through the scanning and reading of it's characters. Luckily receipts tend to always use a white background and black block text making part of the recognition a bit easier.

3.3 Receipt Scan using QR Code

- **Description**

Upon the user successfully logging in, they have the other scan option of QR Code. If they choose the QR Code option it will open the camera just like the OCR and allow the user to scan a QR code provided by the shop of purchase. This QR code will contain all the necessary data of the receipt, such as the expiry date, items, prices etc. This will then transfer the data to be formatted by the Application and then saved to the users account.

- **Criticality**

This is again one of the main aspects of what this application will do and it's USP. This is what we hope to promote as what our application does differently compared to competitors and helps do exactly what one of our purposes of this application is. Reduce paper. This is therefore hugely vital part to this application.

- **Technical Issues**

We must make sure that when data is stored on the QR code that it is easy for the program to be able to parse and format it correctly. We must also make sure the QR code is of high quality for the camera to be able to capture it correctly.

- **Dependencies**

The QR code functionality will rely on whether or not the store has implemented the option of generating the receipt data into a QR code. This also relies on the users camera being of a standard decent quality.

3.4 Viewing Receipts

- **Description**

The user will have the option on the main screen to view their receipts which they have scanned. When they click into it it will give a list of all receipts scanned and give them options to view or delete each one. These can also be sorted or filtered by things such as scan date or expiry date etc. or even view by category. Upon the main view receipts page each receipt will have a quick description such as a name and also the expiry date and value.

- **Criticality**

This is a major part of the application as without it the application would be pointless as users can not view their receipts.

- **Technical Issues**

We need to ensure all scanned receipts whether it be through QR code or OCR end

up in this part of the application correctly and formatted in the right way for the user to be able to easily view and read their receipts.

- **Dependencies**

This heavily relies on the OCR and QR code scanner working effectively as this is the way in which this part of the application will receive any of its data. Without those 2 parts functioning there would be no receipts to display or at the very least display correctly for the user to view.

3.5 Expense Tracker

- **Description**

The expense tracker is used to allow the user to gain an insight into how their spending habits look for the week/month and allow them to see what they are spending their money on and how much. It will also allow the user to break up the spending into categories so they can easily see how much they are spending on a certain thing e.g clothing, food etc.

- **Criticality**

This is an add on part of the main application. Although it can be very useful and some users may make great use of it, it is not the main function of the app. It should be critical enough to allow the user to use but is not as of great importance as previous parts.

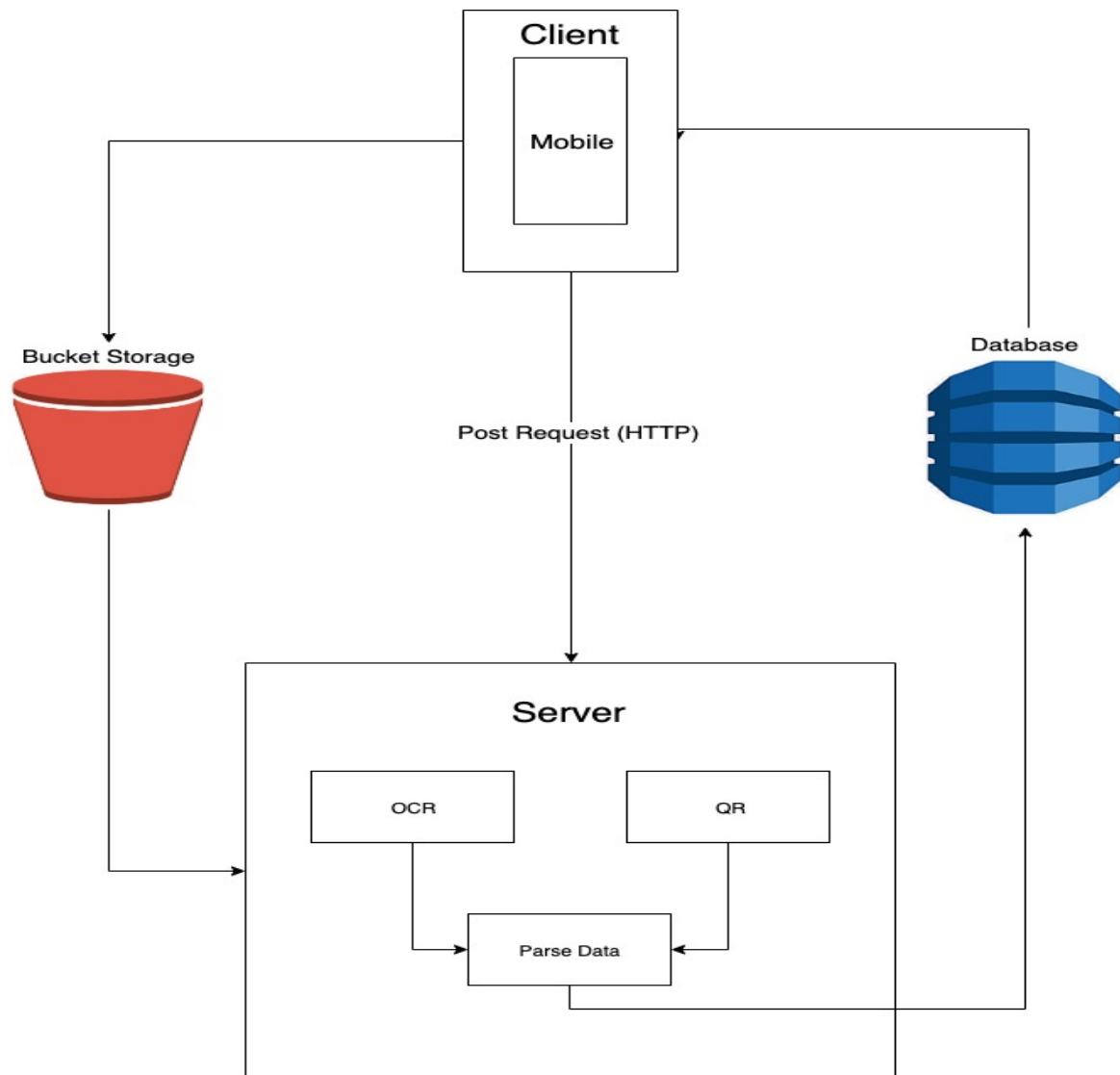
- **Technical Issues**

We need to ensure all scanned receipts amounts get transferred over to the expense tracker correctly in terms of category and date of purchase as if this is wrong or corrupted may lead to false readings and incorrect values for the users spending habits.

- **Dependencies**

This heavily relies on the receipt database and the accuracy of the SOCR recognition and the formatting of the receipts to achieve accurate expenses. This can also rely on the users correct input of which category the receipt may belong to.

4. System Architecture

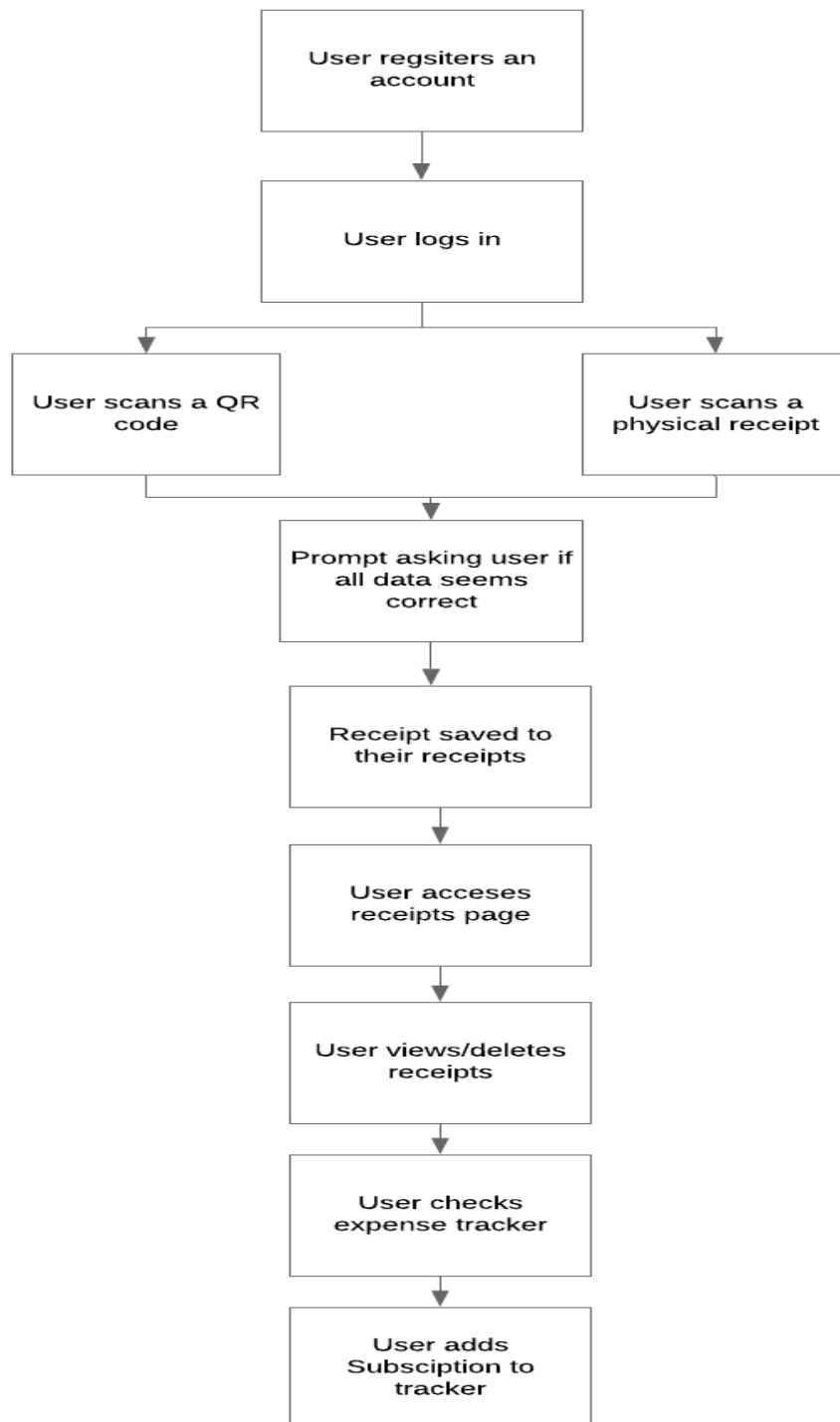


Above is our provisional system architecture. We have not finalised our architecture and are doing research into the exact architecture needed. The purpose of this diagram is to be able to envision how the application will be run and be structured. The advantages of a diagram like this is that it gives us guidance in the development and keeps structure to the application, ensuring best functionality is achieved.

The user will scan a QR code or physical receipt. The client will then send a post request to the server, containing metadata. The image will also be stored in object storage in an AWS S3 bucket, containing metadata. The server will be able to match the metadata in the s3 bucket to the post request and will run either the OCR process or QR process. Once the necessary data from each process is obtained it will then be stored in the database (DynamoDB, AWS RDS). The client then uses the data to format a digital receipt.

5. High Level Design

5.1 High-Level Design Diagram



5.2 Use Case Diagrams:

USE CASE 1	User Signs up	
Goal in Context	The user registers a new account with their email and password	
Scope & Level	System, Database	
Preconditions	The user has never used the App before and has no previous registration information with the app	
Success End Condition	User has successfully created an account with their unique email and password and can log in using it.	
Failed End Condition	User cannot successfully create the account with the application and therefore cannot log in to use it.	
Trigger	User presses sign up button after downloading app.	
DESCRIPTION	Step	Action
	1	User installs the application
	2	User opens the newly installed application
	3	User is prompted with login and sign up options
	4	User selects sign up option
	5	User brought to sign up page
	6	User enters unique personal email and a password
	7	Verification email sent to user's email associated with newly created account

EXTENSIONS	Step	Branching Action
	4a	User selects login option
	6a	Email address is already associated with a created account
	6b	Users asked to enter a valid email
	6c	User asked to enter a stronger password

USE CASE 2	User Logs in
Goal in Context	The user logs in and is brought to the main page
Scope & Level	System, Database
Preconditions	The user has installed the application and created an account with Receiptless
Success End Condition	User has successfully logged in and is on the main page of the application with their account info.
Failed End Condition	User cannot successfully log in.
Trigger	User presses Login button after opening the app.

DESCRIPTION	Step	Action
	1	User opens the application
	2	User presses the login button
	3	User is prompted to enter their email and password
	4	User enters their login details and presses the login button
	5	User is logged in and is directed to the main page of the application.
EXTENSIONS	Step	Branching Action
	2a	User selects sign up option
	4a	Database is checked for valid email and password combination
	4b	User enters an incorrect email/password and a prompt is shown saying login error.
	4c	User selects forgot password button

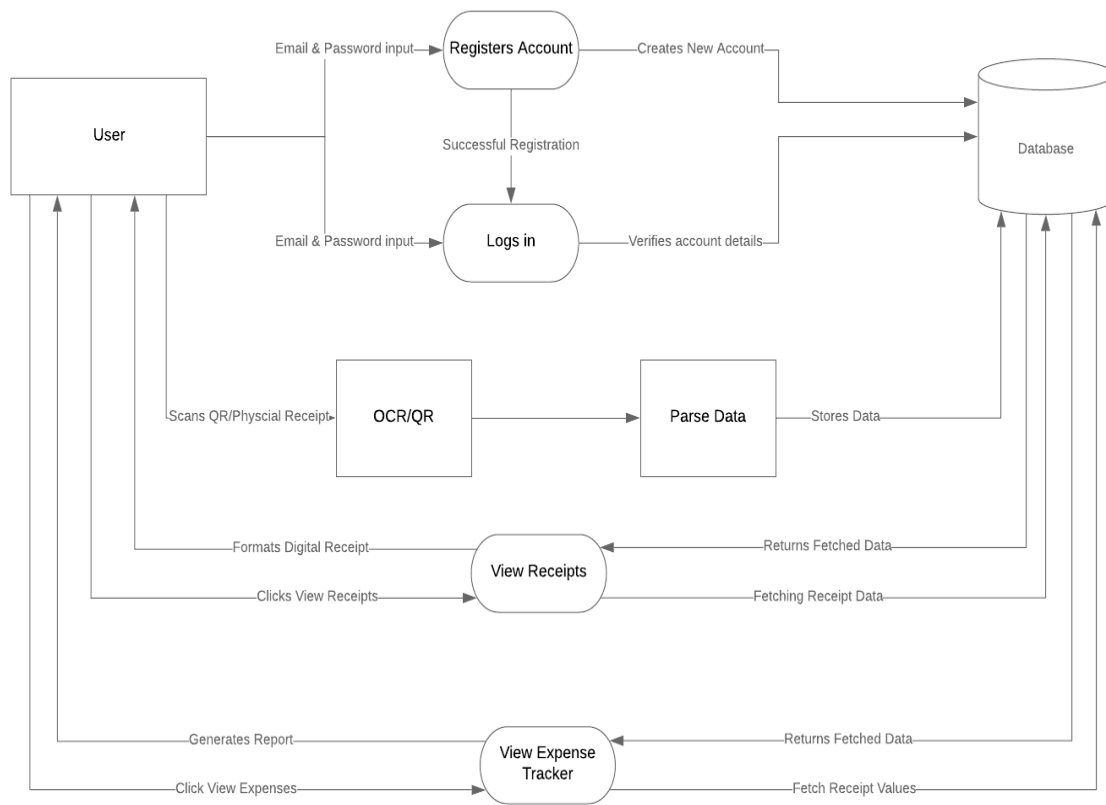
USE CASE 3	User Scans a QR code	
Goal in Context	The user successfully scans a QR code to retrieve data for a receipt to be saved to their account	
Scope & Level	System, Database	
Preconditions	The user has successfully logged in and is on the main page and has a QR code to be scanned	
Success End Condition	User has successfully scanned the QR code and the receipt data from it is transferred and formatted correctly and added to their account.	
Failed End Condition	User cannot scan the QR code	
Trigger	User presses Scan QR button after logging in and navigating to the scan option.	
DESCRIPTION	Step	Action
	1	User opens the application and logs in
	2	User presses the scan button
	3	Users then chooses the scan a QR code option
	4	User then presented with what seems like their normal phone camera
	5	User aims camera at the QR code presented to them from the shop
	6	User is then brought to a review page to see if information from the QR code is correct

	7	The receipt data is saved to the users account.
EXTENSIONS	Step	Branching Action
	3a	User selects scan a receipt option
	6a	User changes some information if they wish
	7a	Receipt data also added to expense tracker such as value

USE CASE 4	User Deletes a receipt
Goal in Context	The user deletes a receipt which was scanned and saved on their account successfully.
Scope & Level	System, Database
Preconditions	The user has installed the application and created an account with Receiptless and has also already scanned a receipt (QR or OCR way) and it was saved on their account.
Success End Condition	User has successfully been able to delete and remove the selected receipt from their account..
Failed End Condition	User unable to delete the receipt.
Trigger	User presses delete button beside the specific receipt.

DESCRIPTION	Step	Action
	1	User opens the application
	2	User presses the My Receipts button
	3	Users has clear view of all their saved/scanned receipts
	4	User presses the delete button beside the receipt they wish to delete
	5	User is given a prompted asking if they are sure they wish to delete
	6	User selects yes.
	7	The receipt and all its data is erased.
EXTENSIONS	Step	Branching Action
	6a	User selects No and the delete operation is cancelled.
	7b	All data for that specific receipt in the Database is erased.

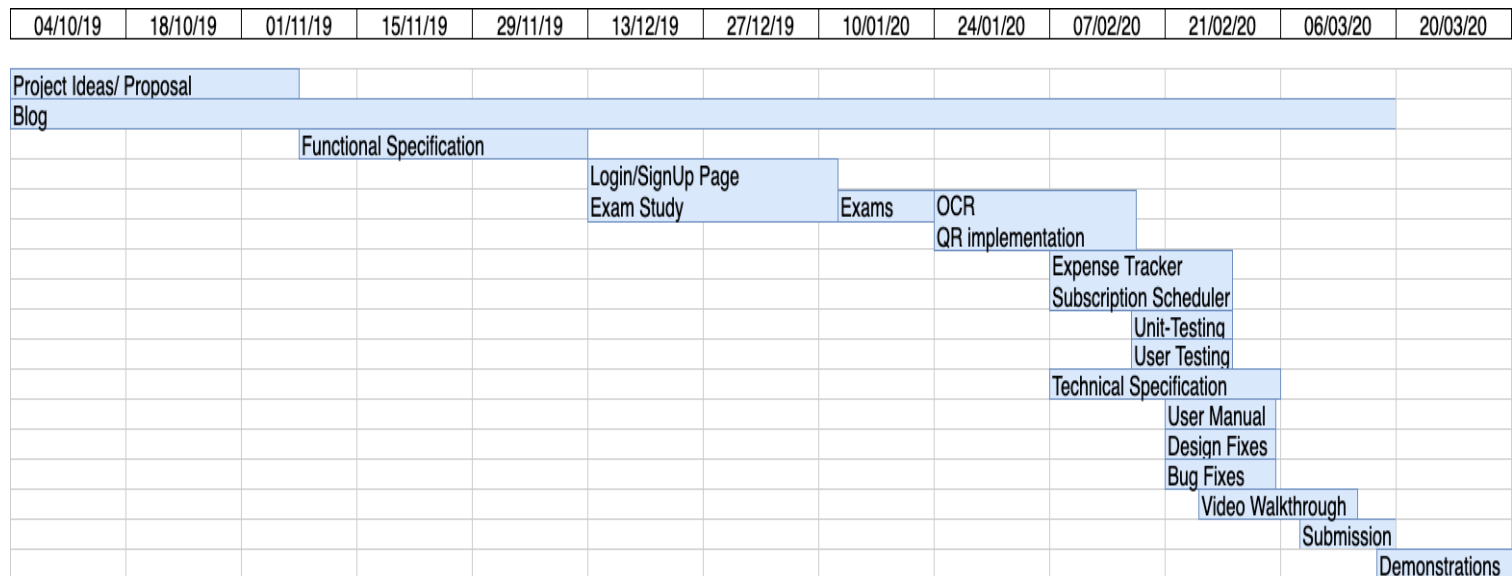
5.3 Data Flow Diagram



The above Data Flow Diagram is used to illustrate the data flow of each step the user may take when using the Application. It shows how data is retrieved and then made user friendly through the likes of formatting and generating reports.

6. Preliminary Schedule

Gantt Chart



As this is a project with a specific deadline, personal targets must be set throughout the development process. We have created a Gantt Chart in order to ensure all personal targets and deadlines will be met. As displayed above, all important stages of the development process are given a start time and end date. As well as the use of this chart we will be having regular meetings with our supervisor Dr. Ray Walshe to ensure sufficient progress is being made throughout each week. We know all areas of the development may not go smoothly therefore we have allowed some extra time in some cases to allow for any issues in the development process.

7. Appendices

Amazon Web Services Documentation:

<https://aws.amazon.com/>

Tesseract (OCR):

<https://github.com/tesseract-ocr/tesseract>

ZXing Library (QR codes):

<https://pypi.org/project/zxing/>