

**Supermarket Saver**

James Redmond

C15339336

Submitted in partial fulfilment of the requirements for the degree of

**BSc (Hons) in Business Computing**

**Technological University Dublin**

May 2020

Supervisor: Dr Neil O’Connor

## **Contents**

Overview……………………………………………………………………………………………………………………4

Acknowledgements……………………………………………………………………………………………………5

Abstract..……………………………………………………………………………………………………………………5

Declaration………………………………………………………………………………………………………………..6

Project Introduction…..…………………………………………………………………….………………………..7

App Features……………………………………………………………………………………………………………..7

Business Rules……………………………………………………………………………………………………………8

Requirements…………………………………………………………………………………………………………….8

Use Case Diagram……………………………………………….……………………………………………………..9

**Database Object Summary……………………………………………………………………………………..10**

ER Diagram……………………………………………………………………………………………………10

**Technologies Used to create Supermarket Saver..………………………………………………….11**

Java……………………………………………………………………………………………………………….11

Build Tool………………………………………………………………………………………………………11

Frontend……………………………………………………………………………………………………….12

XML……………………………………………………………………………………………………………….12

Firebase…………………………………………………………………………………………………………13

**Other Technologies………..……………………………………………………………………………………….13**

Picasso…………………………………………………………………………………………………………..13

GitHub……………………………………………………………………………………………………………14

Android Studio……………………………………………………………………………………………….14

**Libraries/API’s…………………………………………………………………………………………………………15**

Google Maps API……………………………………………………………………………………………15

Google Places API…………………………………………………………………………………………..15

JSOUP…………………………………………………………………………………………………………….16

Google Sign-In………………………………………………………………………………………………..16

Firebase Cloud Messaging………………………………………………………………………………17

Stack………………………………………………………………………………………………………………17

**User Interface & App Instructions.…………………………………………………………………………..18**

Homepage………………………………………………………………………………………………………18

DrawerLayout…………………………………………………………………………………………………19

Login page………………………………………………………………………………………………………19

SignUp page…………………………………………………………………………………………………….20

Search Activity…………………………………………………………………………………………………21

View Products………………………………………………………………………………………………….22

Web view of products……………………………………………………………………………………..23

My shopping list………………………………………………………………………………………………24

Sharing a product…………………………………………………………………………………………….25

View supermarkets near me……………………………………………………………………………26

App in other language(s)…………………………………………………………………………………27

Test Cases..………………………………………………………………………………………………………………….28-31

Conclusion & Implementation Issues……………………………………………………………………………..32

References.…………………………………………………………………………………………………………………...32

## **Overview**

| **Project Details** | Edit photo |
| --- | --- |
| **Course** | BSc (Hons) in Business Computing |
| **Module** | Final Year Project - FPRJ4000 |
| **College** | [Technological University Dublin](https://www.tudublin.ie/) |
| **Student** | [James Redmond](https://www.linkedin.com/in/-jamesredmond/)  - C15339336 |
| **Project Supervisor** | Neil O'Connor |
| **Second Reader** | Thoa Pham |
| **Project Title** | Supermarket Saver |
| **GitHub Link** | [Supermarket Saver](https://github.com/jams100/SupermarketSaver) |

## **Acknowledgements**

I would first like to thank my Final year project advisor Dr. Neil O’Connor and second reader Thoa Pham of the college of Business in Technological University Dublin. The door to Dr. O’Connor was always open whenever I ran into a functional or technical problem. He consistently allowed this project to be my work but steered me in the right direction whenever he thought I needed it. I would like to thank the second reader of my project Thoa Pham, who along With Dr. O’Connor provided invaluable assistance as to how this application could be improved.

I must also acknowledge and thank all my lecturer’s I have had over my 4 years of study. I would like to particularly acknowledge Farrah Higgins whose Business Modelling module proved invaluable during my Third-year placement in SAP, where my role as a Scrum Master in an intern project was made simple through the practical element of her module.

I would like to acknowledge my great classmates who have provided many great moments over the last four years, particularly our class Rep Chris Kavanagh who has been a great representative for us and has always been able to find us an answer for any of our queries.

Finally, I must express my very profound gratitude to my parents and friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the development of my project. This accomplishment would not have been possible without them and everyone mentioned earlier. Thank you

## **Abstract**

This project is a mobile application for all ages who are looking for a helpful solution to finding the cheapest items in Supermarkets.

In this report I will demonstrate some of the project’s main features, how these features were designed, and the technologies involved in making the application.

I will also discuss the problems I encountered whilst developing the application and my reasoning for developing the application.

Declaration

This is original work.

All references and assistance are acknowledged

.

Signed: \_*James Redmond*\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_1/05/20\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Project Introduction**

This aim of this project was to create a mobile application that can compare the prices of products in Supermarkets to find the best prices. I was inspired to develop this project after working in Tesco for over two years and noticing the significant difference in prices of certain items between two shops that are just meters apart.

For this project I used data from Tesco and Supervalu and designed the app with a strong focus on user experience. This application is targeted towards savvy shoppers but can easily be adapted by anyone. There is no Android app like this for supermarket shoppers in Ireland currently as the nearest competitor would be supermarket.ie which is not a mobile app. I had some doubts before developing this application about whether I would have enough technologies or API’s, but I dismissed these fears and haven’t looked back since and I’m now looking forward to further developing this app after submission.

## **App Features**

* Users can Search for grocery products
* Users can Search for grocery products by voice
* Users can Create their own account
* Users can Log into their account
* Users can view products by category
* Users can add products to their shopping list
* Users can see their recent search history
* Users can see their nearest supermarkets
* Users can Share products
* Users can sort products by price, both Low to High & High to Low
* Users can use the application in multiple languages

## **Business Rules**

* Users can use app without registering signing up
* To add products to Shopping List a user must be logged in
* Each user can only have one account associated to his/her email
* User’s choices will not be monitored

## **Requirements**

*Actors*

The actors in my application are the User and the System

### User

Upon registering to the application, you are stored as a user. Users must register with email and password. A user’s email acts as their username and must be unique to every account. Once successfully registered, the email and password inputted can be used to access the user’s account. Once a user has logged into the app, they are brought to the home screen.

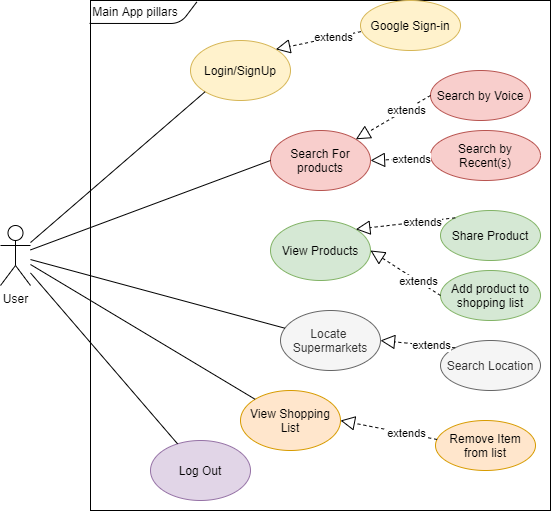
**Home Screen:** The user can search for products which are scraped from supermarket websites or search for products via the drawer layout where they can choose a specific category.

**Supermarkets Near me:** When a user chooses to click on this button, they are taken to a page that automatically locates their current position (provided location is on). They can then view supermarkets close to their current position.

### System

The system is required to accurately search for, scrape the websites and display the results back to the user in a clear and intuitive manner regardless of whether the search is by voice, previous search or a new search.

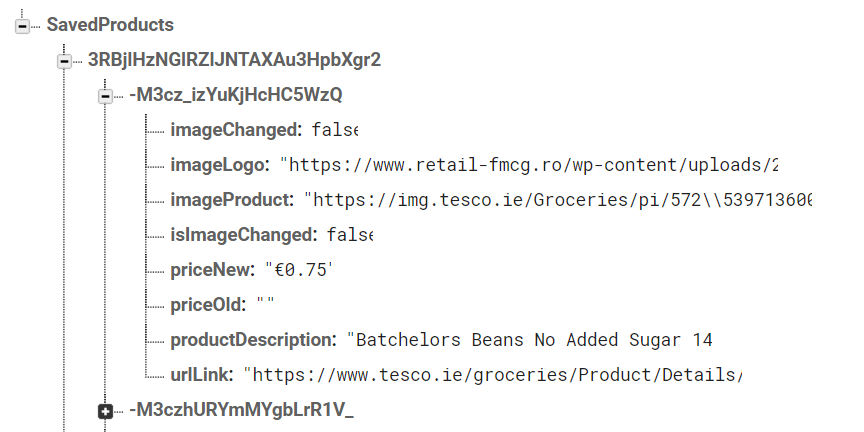
## **Use case diagram**



## **Database Object Summary**

**Saved Products**

|  |  |  |
| --- | --- | --- |
| imageLogo | String | Used for supermarket logo |
| ImageProduct | String | Stores image of product |
| PriceNew | String | Stores current price of product |
| PriceOld | String | Stores Previous price of product |
| productDescription | String | Stores Description of product |
| urlLink | String | Stores link to product on supermarket website |



*ER Diagram*

## 

## The database stores products and their associated information like the image of the product, its description and the price of the product. Users can have multiple products and products can have multiple users.

## **Technologies used to create Supermarket Saver**

*Java*

I chose to develop my project using Java because it was the first programming language I learned back in first year, hence it was the obvious choice as its my strongest programming language. It was gratifying to apply my four years of Java development experience into a project of my own.

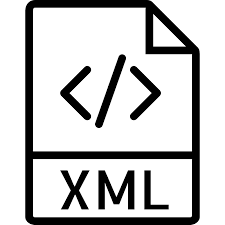
*Build Tool*



Gradle is a build system running on Android studio it’s a custom build tool used to build android packages (apk files) by managing dependencies and providing custom build logic. This made the overall process of building the application much simpler.

*Frontend*

When coming up with the idea for my app I was torn between developing a mobile or web application. I eventually decided to develop on android as the majority of users would be using the application from a mobile device anyway. I chose android over IOS as the number of android users significantly outweighs that of its main competitor IOS, giving my app a greater reach, in March 2020 the android operating system had a 78% market share. The basis of the app is for users to be able to quickly view grocery items and identify their nearest grocery stores no matter where they are, this is much more effective in a mobile application than on a laptop for example. Ultimately the potential and allure of possible future profitability was a big deciding factor.

*XML*

Using Android’s XML vocabulary, I could efficiently design UI layouts and the screen elements they contain, in the same way you create web pages in HTML. In my app the different types of xml files used include:

Animation XML files which define the process of creating shape and motion changes, this can be seen in when you open the application and the logo and other features gravitate into view.

Layout XML files which define the user interface of my app, it holds elements such as Buttons, Text Views, Layouts etc.

Drawable XML files which provide various logos and pictures to the elements of the application.

Style XML files which define different styles and looks for the UI, synchronising these styles across the different pages across the application gives the pages a related look and feel, greatly improving the user experience.

*Firebase Real-time Database*



Stores and syncs data in a NoSQL cloud Database. Data is synced across all clients in real time, and remains available when your app goes offline. For the style of the application the real tome synchronisation functionality is vital. Notable applications using this database include Shazam and SkyScanner.

## **Other Technologies**

*Picasso*



I wanted my app to be visually attractive and considering the volume of images that needed to be loaded when a user searches for a broad range of products I needed to use Picasso to resize and fit images in a less memory consuming fashion. Before using Picasso some of my UI was very sluggish so using it was of great benefit to me.

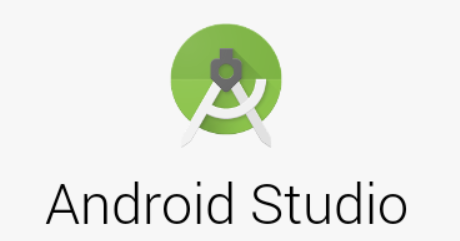
*GitHub*



I used GitHub as my version control system to keep my project up to date in the cloud. It provides access control and several collaboration features such as bug tracking, feature requests, task management and wikis for all projects.

GitHub also allowed me to use a simple text editor to write formatted documents. I used markdown for anyone who visits my GitHub repository so they can get a good overview about the application. I also found GitHub’s help section had articles for nearly any topic related to git that you can think of, making it an excellent source for documentation.

*Android Studio*



I chose android studio as my IDE to develop my mobile application as it is a much more efficient than other IDE’s when it comes to developing and building mobile applications such as eclipse. My first experience of using Android Studio was in our first semester mobile development module and I got the hang of it quickly. It’s built in git configuration and gradle integration were also a major help throughout development.

## **API’s & Libraries**

*Google Maps API*

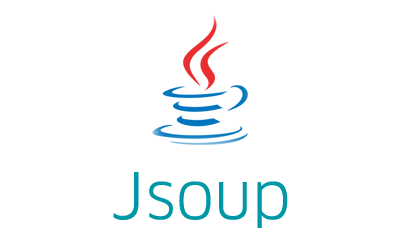


Supermarket Saver utilises this API to show the users location and through integrating the places API allows a user to view Supermarkets near their current location. They also have the option of searching for a specific location.

*Places API*

The Places API is a service that returns information about places using HTTP requests. Places are defined within this API as geographical locations or establishments. This was used to show users of the App Supermarkets near their current location.

*JSOUP*



JSOUP is an open source Java library which provides a convenient API for extracting and manipulating data using DOM, CSS and jQuery-like methods. It allows you to scrape and parse HTML from a URL. I used it for scraping both Tesco and Supervalu.

*Google Sign In*

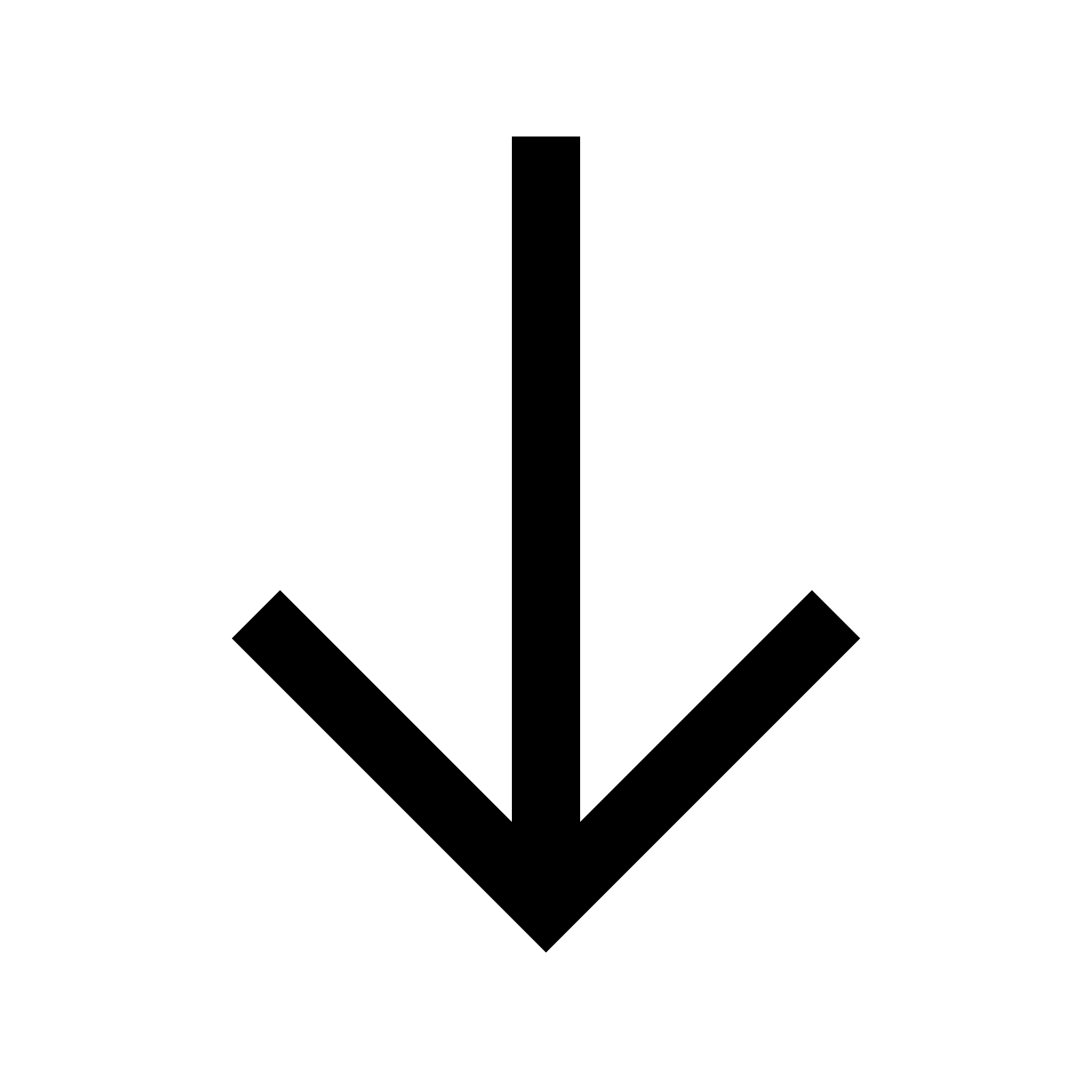


Google Sign-In is a secure authentication system that reduces the burden of logging into applications by enabling them to sign in with their Google Sign-In. It is also a gateway to connecting with Google’s users and services in a secure manner.

*Firebase Cloud Messaging*

I used firebase cloud messaging to send push notifications to users as its much more reliable and quicker than using stock android alarm and notification services. Users are reminded to check the app at every fixed time period. FCM allows for sending data in notifications also, it is much more reliable than other options because when a device is restarted the notifications will still come in whereas other solutions are hit and miss.

*Stack*

**Firebase**

**Android**

**UI Design & App Instructions**

|  |  |  |
| --- | --- | --- |
| *Homepage*  The user opens the app to be greeted with an animated start. This is the first indicator the user gets of a friendly UI.  The user can immediately start searching for products, view supermarkets by location or login/sign up if they want to add products to their shopping list. |  | |
| Homepage /MainActivity.java | |  |

|  |  |  |
| --- | --- | --- |
| *The Drawer Layout*  A user can navigate to certain features of the application such as Logging in, popular categories or view special offers using the drawer layout.  It is not visible by default and it needs to be opened either by sliding from left or clicking its icon in the Action Bar. The use of this layout greatly enhances the user experience. |  | |
| Drawer Layout from Homepage/ MainActivity.java | |  |

|  |  |  |
| --- | --- | --- |
| *Login page*  The user logs in with either their registered email and password or via Google.  A user needs to be logged in to be able to add items to their favourites/shopping list. |  | |
| Login Page /Login.java | |  |

|  |  |  |
| --- | --- | --- |
| *Signup Page*  The user can register an account with their email address. This a core function of the application.  Registering automatically logs the user in under their sign-up credentials.  If the user signs up successfully then they are automatically redirected to the homepage (MainActivity.java) |  | |
| Register Page /SignUp.java | |  |

|  |  |
| --- | --- |
| *Search Page* | *Search By Voice* |
| Search Page/Search.java | Search By Voice/Search.java |
| When a user clicks on the search button from the homepage they are taken to this page.  They can see a list of their previous searches; this has been implemented using Material search so as the user types in their product the suggestions become more relevant, this speeds up the efficiency of the searching process. | A user also has the option of searching for a product by voice. To do this they simply press on the microphone in the top right-hand corner and say the product they are looking for.  This is beneficial as it allows the user to multitask and, in many cases, allows users do things faster than other devices. (For example, if cooking searching by voice might be beneficial) |

|  |  |  |
| --- | --- | --- |
| *View Products (List layout)* | *View Products (Grid layout)* | |
| View Products/ProductList.java | |  |
| After a user searches for a product they are taken to the ProductList.java page automatically.  If users choose to view the product in a list view they will be presented with a layout like the above layout. | The other way of viewing products after searching for them is in a Grid view like the above, the benefit of this layout is it reduces the need to scroll.  The products are sorted from cheapest to most expensive by default, but the user can change this. | |

|  |  |
| --- | --- |
| Web View of product(s) | |
|  |  |
| If a user clicks into an item after searching for a product they are taken to the respective website of the supermarket of that product. This is useful if for example, a user wants to see if that item is gluten or lactose free. A user can also see other important information such as the nutritional info on said product from this page. | |

|  |  |
| --- | --- |
| *My shopping list (Saving a product)* | *Viewing my shopping list* |
| Saving a product/ProductList.java | My Shopping List /SavedProducts.java |
| A user can add a product to their shopping list by clicking on the saved button like above. This will save that product to their shopping list which they can view later.  A user needs to be Registered/Logged in if they wish to add products to their shopping list. | After adding products to their shopping list, a user can then navigate to their shopping list. From here they can view all their saved items. |

|  |  |  |
| --- | --- | --- |
| *Sharing a Product* | *View of Shared Product on Messenger* | |
| Sharing a product/ProductList.java | |  |
| A user can share a product with a friend if for example he or she identifies a good offer on a product that they were talking about with a friend a few days previous.  The product can be shared across almost any social media platform. | This is a view of how the receiver will view the message on their device. | |

|  |  |  |
| --- | --- | --- |
|  | Viewing Supermarkets near current location |  |
|  |  |  |
| GoogleMapsActivity.java | GoogleMapsActivity.java | GetNearbySupermarkets.java |
| When a user clicks the view nearest supermarkets button on the homepage they are taken to the GoogleMapsActivity.java page.  Location permission must be given for this feature to work. | When location permission is given the GoogleMapsActivity page automatically locates the users current position. | If a user clicks on the supermarkets button at the bottom of the page, then supermarkets within their current location are displayed.  A user can click onto one of these supermarkets to get more details and start navigating to it via Google Maps. |

|  |  |  |
| --- | --- | --- |
| *App in French* | *App in Spanish* | |
| Viewing app in French/Spanish – Strings.xml | |  |
| If a user from another country is using this app they can view it in French or Spanish. This is also beneficial if down the line the app was expanded into different regions in Europe. | |  |

## **Test Cases**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Test Data** | **Test Steps** | **Expected Result** | **Result** |
|  | *SignUp or Login Activity* | | | | |
| 1 | Check user registration | Email: james@gmail.com  Password: 123456  Confirm Password: 123456 | Launch the app.  Open Drawer view.  Click Login.  Click Signup.  Enter Email, Password & Confirm Password.  Click ‘Signup’ | User stored in firebase.  Signup complete.  Homepage appears. | **PASS** |
| 2 | Check user Login functionality | Email: john@gmail.com  Password: Monkey | Launch the app.  Click Login.  Enter Email.  Enter Password.  Click ‘Login’ | Login successful.  Homepage appears. | **PASS** |
| 3 | Check Google Login |  | Click ‘Google Sign In’ | User prompted to grant permissions.  Login Successful.  Homepage appears. | **PASS** |
|  |  |  | *MainActivity* |  |  |
| 4 | Check Drawer Layout Opens |  | Swipe from Left to right.  Or Click on Drawer Layout bar. | Drawer Layout opens | **PASS** |
| 5 | Click Logout Button |  | Press Logout | User is Logged out. | **PASS** |
| 6 | Click Search box |  | Click search box on homepage. | ‘SearchActivity’ launched.  Previous searches displayed (if applicable) | **PASS** |
| 7 | Click ‘Supermarkets Near me’ Button |  | Click on ‘Supermarkets Near me’ Button. | ‘GoogleMapsActivity’ launches successfully. | **PASS** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Test Data** | **Test Steps** | **Expected Result** | **Result** |
|  | *Search Activity* | | | | |
| 8 | Type a product and click search | Enter Product: ‘Bread’ | Click Search. | ‘ProductListActivity’ started, products displayed. | **PASS** |
| 9 | Search for product by voice | Vocalise product: ‘Water’ | Vocalise product to search for. | ‘ProductListActivity’ started, products displayed. | **PASS** |
| 10 | Search by clicking on a previous search. |  | Click on previously searched product ‘Milk’ | ‘ProductListActivity’ started, products displayed. | **PASS** |
|  |  |  | *ProductListActivity* |  |  |
| 11 | View products in list or grid format |  | Click the gird or list layout viewer. | Layout changes to list.  Layout changes to grid. | **PASS** |
| 12 | Sort products by price |  | Sort by price ‘Low to High’.  Sort by price ‘High to Low’ | Products are displayed in order of selected price. | **PASS** |
| 13 | Save a product to shopping list |  | Click on the save product button. | Save product to shopping list.  Display Toast message. | **PASS** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Test Data** | **Test Steps** | **Expected Result** | **Result** |
| 14 | Click on a displayed product |  | Click on: ‘Supervalu Still Water 2 litre’ | ‘WebViewActivity’ started, product displayed on supermarket website. | **PASS** |
| 15 | Share a product |  | Click on share button.  Choose platform to share product on.  Choose people to share with. | Product is shared with person(s) successfully.  Toast message is displayed. | **PASS** |
| 16 | View your shopping list |  | Click on saved products spinner in top right corner. | User is taken to their saved products page.  If not already logged in user is taken to login page. | **PASS** |
|  |  |  | *SavedProductsActivity* |  |  |
| 17 | Delete a product from shopping list |  | Click on Delete Button.  Confirm deletion. | Item removed from shopping list. | **PASS** |
| 18 | Share a product from shopping list |  | Click on share button.  Choose platform to share product on.  Choose people to share with. | Product is shared with person(s) successfully.  Toast message is displayed. | **PASS** |
| 19 | Click on a displayed product |  | Click on: ‘Batchelors Beans 141 gram’ | ‘WebViewActivity’ started, product displayed on supermarket website. | **PASS** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Name** | **Test Data** | **Test Steps** | **Expected Result** | **Result** |
|  | *GoogleMapsActivity* | | | | |
| 20 | Click on Location button |  | Click on location button. | The Maps interface moves to the user’s current location. | **PASS** |
| 21 | User presses the Supermarkets button |  | ‘Supermarkets’ button pressed. | Map moves to user’s current location.  Nearby Supermarkets displayed with beacon.  Toast message displayed. | **PASS** |
| 22 | User searches for a location | Location: ‘England’ | User Enters location into search bar.  User presses search. | Map moves to searched for location. | **PASS** |

## **Conclusion & Implementation Issues**

Overall, I am quite happy with the look and feel of the system that I have built. I am however quite disappointed that I did not get to add as much functionality to my project as I first planned.

An area where this project has helped me learn extensively is prioritisation of tasks and time management. Despite making regular commits to GitHub I feel retrospectively that I could have used a tool like JIRA to ensure that I made the most efficient use of my time.

One of the main implementation issues in this project was scraping the data from the Supermarket’s websites, whilst scraping the data itself took some time, it was presenting that data back to the user in a clear intuitive manner which proved more time consuming than I initially thought.

To conclude, there are many lessons to be learnt from building my first full scale application. One of the main things I would do differently if I could go back in time is spend more time planning out my application at the start of semester one, as I know I spent a lot of time in the past few months researching what should’ve been done during the early stages of the project.

Overall, I can see how this app with a little refinement could become an extremely useful app and I look forward to continuing working on it in the future.

## **References**

1. JSOUP Documentation, [Web]. (2020, January). Retrieved Jan 7th, 2020 from https://jsoup.org/
2. Android Developer Documentation, [Web]. (2020, January). Retrieved Jan 3rd, 2020 from https://developer.android .com/index.html
3. About Glide, [Web]. (2020, January). Retrieved Jan 20th, 2020 from https://bumptech.github.io/glide/
4. Google Maps SDK, [Web]. (2020, January). Retrieved March 20th, 2020 from https://developers.google.com/maps/documentation/android-sdk/intro
5. Authenticate Using Google Sign-In on Android, [Web]. (2020, January), Retrieved January 30th, 2020 from https://firebase.google.com/docs/auth/android/google-signin
6. Add Firebase to your Android project, [Web]. (2020, January), Retrieved February 15th, 2020 from https://firebase.google.com/docs/android/setup
7. Google Places API, [Web]. (2020, January), Retrieved March 21st, 2020 from https://developers.google.com/places/web-service/intro
8. Material Search View, [Web]. (2020, January), Retrieved February 20th, 2020 from http://miguelcatalan.info/2015/09/23/MaterialSearchView/