The Shopping Helper Application

Final Report for CS39440 Major Project

Author: Richard Henry Chowne (rch15@aber.ac.uk)
Supervisor: Dr. Antonio Manuel Anjos (ara11@aber.ac.uk)

1st April 2017

Version 1.0 (Final)

This report is submitted as partial fulfilment of a BSc degree in
Computer Science Vision and Graphics (G450)

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB
Wales, UK

Declaration of originality

I confirm that:

- This submission is my own work, except where clearly indicated.
- I understand that there are severe penalties for Unacceptable Academic Practice, which can lead to loss of marks or even the withholding of a degree.
- I have read the regulations on Unacceptable Academic Practice from the University's Academic Quality and Records Office (AQRO) and the relevant sections of the current Student Handbook of the Department of Computer Science.
- In submitting this work, I understand and agree to abide by the University's regulations governing these issues.

Name …………Richard Henry Chowne………………………………

Date ………………18th/03/2017……………………………

Consent to share this work

By including my name below, I hereby agree to this dissertation being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Name …………………Richard Henry Chowne………………………

Date …………………21/03/2017…………………………

Acknowledgements

I am grateful to my lecturers for guiding me, being patient with me and always making me want to better myself as well as apply all that I have learnt into the real world. I am also grateful to my supervisor Dr. Antonio Anjos for guiding me and helping me in my struggles in my final project.

I would like to specifically thank my dissertation supervisor Dr. Antonio for giving me the opportunity to work on this project, As well as the computer science department for preparing me and teaching me all that I need to know to be prepared in my career.

Also like to be thankful to my friends for helping me throughout the time of my final Project, as well as broadening my understanding of my field and always teaching me new things.

Abstract

The Project is about constructing an application for shoppers to help make their daily tasks for shopping go much easier due to us living in a technological era. The following project will allow me to build the said app from scratch by coding it in Android studio.

This app will be made to assist shoppers in constructing a grocery list on their phones, which will save them time; rather than looking through various items and spending an unnecessary amount of time in the supermarkets. The user can also edit the shopping list as well as scan products with the help of the built-in scanner, to help make their shopping transition become smoother.

The application shall be programmed in android studio as it offers much more flexibility, in areas such as the preparation of the application as an APK as well as the user program is much easier to use with no restrictions.

When loading the application, the user will be presented with three buttons: ***Add shopping list***, ***Load shopping List*** and a ***How to*** use section. Within the "*Add shopping list"* the user will be presented with a box and three drop down buttons where the user can add products to the shopping list manually or scan them into the shopping list. As well as an edit button in case the user wants to modify the amount or change the name and a Delete button to remove the products.

The "*scan Products"* will enable the camera in the user's phone to become a scanner so that the user can scan the barcode on the product, it will check the database to see if the product being scanned is in it and if it is not it will ask the user to add it to the database so they won't have to in the future, then it shall be added to the users shopping list.

I will include within in the application a how to use section as that might become helpful for those who do not know how to use the application or to those who need help. The methodology I am using is FDD: Feature driven design which will be explain in the further sections in depth.

Contents

1. Background, Analysis & Process

For the following sections *Background, Analysis and process,* I will be going into what I did for the project in each section in as much detail as I can so that the user can get a better understanding of how I have made my application as well as what I was thinking when doing these changes or applying certain features. Hopefully, each sub section will be explained in detail. For Background, I will be discussing the research that I have done for my application and what was needed to accomplish it in terms of research as well as in terms which portions of my project I will be following. As well as the end goal that I must accomplish in this assignment.

I will also brush over the other comparisons of shopping applications in the background section and how I can make mines stand out, as well as choosing the proper programming environment for me. Also, a brief description on the API I am using as well as the kind of library I am using for my scanner.

For the analysis part of my report I will mention all that I have done in regard to the setup of my application as well as the setup of the programming application Android Studio, as well as mention the alternative programming environments available to me to use. I will mention if the requirements have been met as well as if there are any security issues in regard to my application. (Shopping Helper)

For the Process section of my report, I will mention the methodology briefly and what the other types of methodology I have used in my report and finalised with as well as how I started the project.

1.1. Background

The subsequent project is about constructing an app for shoppers to help make their daily tasks of shopping much easier due to us living in the twenty-first century, which has also been referred to as the era of technology.

Likewise, individuals can access the internet through their fingertips using their electronic smart-phones that are with people; either at most or all times. The following project will allow me to build the said app from scratch by coding it in Android Studio.

The main aspect that I will be focusing on during the construction process of this app, will be the scanner aspect as well as the shopping list, as I find that to be the most difficult bit that is focused on making sure that individuals are able scan objects in, as well as remove them from the desired shopping list; thus, making sure that it registers properly.

With the use of the applications built in scanner the user can scan products, as well as providing the choice of editing the items within the electronic shopping list, such as removing products that the buyer no longer wishes to have on his/her list.

Furthermore, the said app will be saving the saving the buyer a significant amount of time from standing and waiting and deciding if they need that item or not, this makes the task of shopping go much quicker as you know what you want already before heading to checkout.

There was a lot of preparation as mentioned there was a lot of research online looking for papers on how to help with building the app, android developers website was an immense help with the research and help in how to get started in building my application and setting up android studio.

The other factor was also comparing other shopping helper applications and seeing how I could make mines better within the time frame of this project, so a lot of comparisons between shopping apps was needed. As well as asking people what makes a shopping Application the way it is and if they had a choice how would they want to design it, this gave me a starting point on what to look for as well as research.

The other research I had to do was find and setup a free products API [1] to hold the products as well as testing the connection to make sure that I could create as well as add onto the database and retrieve that data.

As well as look at how to use an API what were the proper requests and commands to use in order to get it working well so that involved a lot of research on google on how it works which I did find online as well as some great APIs but they all cost money, which I will mention in the implementation section below.

The motivation I had in this project was that at the end of this project I would have the knowledge of building apps as well as I would have built my very own android application, as for the interest in this project I wanted to create something like how amazon built their own shop with no customers, as everything revolved around their phones as well as in Japans stores.

How people just use the shopping basket application in their phones to scan products and then add it to the basket physically then checkout minimizing the time expectancy of shopping in general. As it takes out the middle man i.e. (the cashier).

This app will be made to assist shoppers in constructing a grocery list on the spot, which will save them time; rather than looking through various items and spend an unnecessary amount of time in the supermarkets.

The final goal for the following project, is to produce an efficient and entirely serviceable operating application that is user friendly in assisting the shopper with their essential tasks, in which it helps shoppers with their essential tasks by making sure that the application is competent enough to register products when scanned through.

As well as the creation of multiple shopping lists. As well as editing and deleting of the shopping lists as well as products entered manually or scanned in that are displayed into the shopping lists.

1.2. Analysis

There were many problems in the begging to get setup with how I could not get android studio to setup on my laptop as well as how to get the virtual android to setup which held no result, but eventually after further reading and searching online for resources I have managed to get everything setup and working fine.

I could have used eclipse to code my application or NetBeans android plug-in, but I have decided on using Android Studio as I was determined to work on android and release on android as well as there were plenty of help online.

The objective of this project was to get a shopping helper application that would help the user create shopping list that can be saved and that they can add products manually or scan the barcode of the product, via the camera as well as edit and delete products. Which I have mentioned above in the background section.

In most cases, the agreed objectives and requirements have been met, if given more time I would have accomplished more features such as share as well as a simpler looking GUI and implementing more features such as QR code scanning and adding of coupons. As well as a save products scanned which I have mentioned in the critical evaluation section of this report.

Regarding possible security issues there will not any issues regarding this project, the only security issue will have to be the permission the user will have to grant when the scanner button is clicked, the camera loads up other than that there is no other issue, everything is saved locally on the device from shopping lists and products.

In which I have mentioned below as well as when the user removes the application everything with the application is also deleted (mentioned below), which makes no security issue for my project other than the above-mentioned permission. Also, the permission for internet access to connect to the API to send products barcodes and retrieve the names as well as add it to the database so no more scanning is needed for the same product and going through the same step, (which I have mentioned below in the proper sections)

1.3. Process

The type of research for my project is a software engineer based one, the process I used for my project was feature driven development (FDD) where it is a mix of lightweight and agile methods the purpose of this is to give me working software in a timely manner which I felt that would be perfect for this project.

The Project life cycle of Feature Driven Development [2] are:

- Developing an overall model.
- Building of a feature List.
- Plan by feature.
- Design by feature
- Build by feature.

The Project lifecycle of the Waterfall Life cycle [3] are:

- Having a Requirement Analysis.
- A System design.
- Implementation.
- Testing.
- Deployment.
- Maintenance.

What I have used in this project was a mix of both of the above-mentioned methodologies. I started off my project with the waterfall process to wrap my head around everything since I have been accustomed to it before and thought it would fit perfectly with this project as well in some respects. Then I later on adapted the Feature driven development one in the later stages of my project and continued to use that.

I began by drawing out an overall model to see how they could be combined and be programmed, the second step was to build my features list that hold activities for my design which will be mentioned in the other sections, after the features list was finished I had to plan by feature to assign ownership to each feature.

Before the decisive step of building the feature, I had to design them so that I could program each feature to do what I wanted them to do and test that everything was working fine.

For the decisive step, I built the feature as well as test it again to make sure that it does everything I wanted it to do. That is my process as mentioned I will go over what features and their purposes are as well as locations in the implementation section below.

List of features:

- let the user add, delete and edit a product.
- Let the user create a shopping list, edit the shopping list, delete the shopping list, view the shopping list and add Products to the shopping List.
- Add products, delete products, edit products.
- Scan products barcodes edit scanned products names and change the quantity, delete the scanned product.
- Loading of a shopping list
- Looking up the product on the database (done by the system)
- Saving of the shopping list and its products (done by changes to the shopping list)3

2. Design

In the following subsections, I will explain on the following designs choices of my application for my final project.

The *overall architecture,* I will be explaining how my application will look like as well as show how everything links together as well as mention the methodology used. As well as the overall design of my application by the end of the project.

For the *Detailed design section,* I will talk about the refined design and plans and why I have decided to do it that as well as important issues of my design as well as justifying why.

For the *User interface design*, I will be explaining in detail on how I focused on maximizing the user experience as well as the changes to the initial design of my load menu to the latest one that I have ended up with, and all the other additions and button layouts as well as new activities as.

For each of the user interfaces I will mention there will be a corresponding Appendix as there is a clear picture of how everything looks like in the appendices. I will go into the reasoning as to why I have done everything the way it is in the following subsections. As well as justifications to the designs I have chosen.

There were a lot of revisions to the design as I had to go back and change the look of things as well as apply a few changes such as the shopping list that displays the products.

2.1. Overall Architecture

The most important aspects of the overall architecture were how things would be presented and if they would be properly programmed and linked well with one another.

Initially in the beginning, I did not have much to show but had been given a lot of ideas as well as searching online for resources and ideas.

I decided that it would have to make sense when the user would open an activity and jump from activity to activity so I started by structuring how each activity would look like.

I had adopted the waterfall design method as I was most conformable with it as a starting point. For the overall architecture, I had to design each section in stages for my project so I would not get confused as well as to know the execution order from the top of my head. Then I changed it back to the FDD (feature driven development) design by the middle of my project.

I had started with the menu screen as well as how things would progress from there when the user clicks on each button where it will take them and how things would look.

A new layout design was needed for the add shopping list as well as for the load list.

The overall design for this project is to allow the user to be able to create a shopping list and then access it in the load shopping list, as well as edit it or delete it and access the shopping list to add products in two ways.

where the user can manually add products by inputting the name and quantity and submitting it to the list or scanning products in and just inputting the quantity which adds it to the scanners API and to the shopping list. See (*Appendix A)*

2.2. Detailed Design

For the detailed design aspect of my project I have come to find that the most critical issue is the httpsGetProduct and httpsAddProduct as they are the classes that are involved with how to add and retrieve products from the products database. Which is using Outpan API [1] which my supervisor recommended to me.

I will be going into the aspect of my scanner as to how it is able to make GET and Post requests to the products database below in as much detail as I can.
Get: retrieves the product that the user has specified via the GTIN.
POST: adds the product that the user has specified onto the database with the help of the GTIN.

When I came across the idea of storing products I struggled on how to do that hence why my supervisor advised me, as well as myself searching online for product APIs which was to no avail. The products API I am using is from Outpan [1] and its made storing the data much more easier and recalling it smoother. As everything is being done with a get and post request using the barcodes of the products which again makes everything smoother as I don't have to make my own database which would have taken too long and make me miss my deadline.

All the user does is as soon as he scans the barcode of a product, it collects the GTIN and sends that as a Post request to adding into the API products database. Then the GET request returns the products name that the user has defined. Which the user then can add the quantity.

As well as the next time the user will not have to scan the same product again as well as re add it to the database as it saves the product name.

If the product scanned Is registered on the database all it will do is gather the information and then the next activity will load which will take you to the product name section and quantity directly. To be added to the shopping list.

This was with the help of using Postman which made me see how everything is being done as well as how I can test is which is why I will mention it in the testing section below.

2.3. User Interface Design

For the user interface design I had initially gone with a very simple layout with three buttons: Add list and scanner and a how-to see *Appendix B*. That was the initial design but after a few revisions and as I progressed in the project I realised that it was flawed and had to start all over again from scratch.

I will provide more Screenshots and a brief description on each picture and the functionality and purpose of each button in their corresponding Activities, which will be available in the Appendixes section below.

I changed the design from the pictures in *Appendix B* to *Appendix C* which is the more current one. There are slight different changes that have been applied from text darkening and button renaming. As well as for the why the buttons have been built like that it is because it is very simple and not that padded as for the positioning I thought that putting them in the centre gives it a better look for presentation.

See *(Appendix B)* The initial interface design I had submitted when I started with my project but as mentioned after further development I have improved on my design and changed the layout. The very begging of my designs which I believed at the time was the right one but after further research and analysis as well as structuring I decided it was not needed and an upgraded one was required.

See *(Appendix C***)** The new user interface is much more better as it offers clear instructions to the user as well as the execution of orders is much easier now. You can see the that the buttons names are different as well as the text is more bolder and much easier to spot. This is the new menu page I have decided to go with. My justification for why the buttons have be renamed are as follows, it is a good form of execution in my head as when the user loads the screen he can see a create and load buttons in from as well as a how to get used to the program.

See (*Appendix D*) The load shopping list is where the user can see the shopping lists that they create. It also can be manipulated with editing the name and deleting the shopping list. It also can be viewed and then the user can add products to the shopping list. The justification behind having the loading screen display the shopping lists like so is that it is easier to spot as well as I have put in a scroller so if the user has multiple shopping lists they can scroll as well as why I have made the edit and delete button and view separate as I think that it would better if the user could be able to separately do each task as well as view each list.

See (*Appendix E)* The *products list* of one of the *shopping lists* where you can see the products that have been added into the shopping list. I have made it so that when the user scanned a product it will display with “air quotes” over the product i.e. “fish” so it is easier to detect and notice which is scanned in and which is added manually.

As for the shopping lists product layout, I have made it so that the users who input products in that are more than a certain amount of characters it will make it become "…" as I had an issue which I will mention in the implementation section regarding product display.

The buttons that are beside the products *(Appendix E)* are there to manipulate them. The edit and delete buttons in the products list to delete the desired product as well as edit the product such as change the name or give it a nickname and change the quantity. Initially they were supposed to be borderless to make it less cramped but I decided to keep them in as they would be easy to press as well as the user will not make a mistake and hit the wrong thing as most people have made that mistake.

Initially there was a Load button within the shopping list (the old design), but I removed it after going through many changes and to make the scan products in the add products activity rather than within in the shopping list. Which will be mentioned in the respected sections. As it made more sense as well as looks much more better with the load being in the front page rather than inside the shopping List.

See *(Appendix F)* when the user scans a product and it is in the database it pulls the product name and lets the user input the quantity. If the product is not in the database then (Appendix G) will be triggered alerting the user that it's not in the database and if they would like to add it then click yes or no to scan again. The justification behind keeping the layout as it is because I had initially kept the design before as Product name now I changed it to a hint text so the user knows to put the product name as well as the quantity I had initially made it a hover text so when the user is moved to the proper section they would know what to put into it as a message would pop up.

(Appendix H) which will prompt the user to add the name of the scanned product by either clicking "Yes" or "No" if "NO" is clicked then the user is taken to ScanBack again if the user clicks "Yes" then the user gets taken to the next page to add the product to the database by inputting the name and after hitting submit. The user will notice that the quantity field has been added in (appendix I) where they input the amount they want of the product and will find it in the shopping list. Which is in *(Appendix J).*

Initially, I had thought about before letting the user ad products they would be presented with a message to tell them that this is the place to add products in. Although that was the initial idea I have made it so that there is a clear message telling the user that the product that they have scanned is no registered and if they would like to add it they can choose yes or no if yes then they go through the proper steps if no then they can scan a different product.

As for the submit buttons in the create section and the adding of a product as well as from the manual add products as well as the scan products I have designed them and out them on the bottom corners as most applications that I have researched online have the buttons on the bottom, as well as when editing the button layouts, it automatically made the positioning of the submit buttons under which was a good thing for me. As the user can clearly see it when confirming their choices.

The save [4] is done as soon as a change is detected in the shopping list. The save is done internally so when the user deletes the application everything is gone with it.

Saving internally [4] was an immense help as they were only accessible by my app. Also, one of the benefits with internally saving is when the user uninstalls my application everything is also gone with the uninstallation. All of the files and cache everything is completely wiped. As well as the only one who has access to the application is the user who downloads it. Compared to externally saving and recalling which was more time consuming as well as not beneficial for this project as well as not what I wanted to do.

3. Implementation

The issues that I have encountered through this project were a quiet a few in the most complicated sections of my project, I managed to fix them with a lot of online research to help me with finding a solution to my problems as well as rechecking my code and asking my colleagues and supervisors in the meetings for assistance.

During the course of the project, I found out that elements of my design were overly complex and that I could not understand what was wrong. As the logic behind the programming was sound it was then that I decided on testing my work after I had made some progress after implementation of certain changes, Then I tested it in on an actual phone and noticed that things were going well and everything was working correctly for the initial stages of building my application. Such as the GUI and how everything was connecting to one another.

Eventually, as I progressed with my application I came across a lot of small problems. For instance, when I came across adding products or the creation of the shopping list the application crashed and I thought that I broke something but after running in debug mode I managed to find out why and fix the problem.

The other issue was when products were being added they were not being displayed properly as they would display weird symbols and asterisks but small errors like that were corrected after debugging and some corrections in a few lines of my code.

Before building the layout, there was an issue with renaming the application which I had struggled with hence why I kept the name of the application as “Shopping Application “rather than “Shopping Helper” which I initially wanted to do the project broke for some reason so I kept the name as is.

The other minor issue would have to the aesthetics of the applications specifically the theme of the application as well as the image of the application which I will mention in my critical evaluation. I tried to change the icon in the image to one that I had designed but it did not work as well as when I wanted to change the theme of the app from traditional blue to a much more easier appealing colour like gray or ashy brown but it didn’t work as well as I didn’t have time to do that as mentioned there was a lot of other issues.

There were a few minor issues that were building up in the layout of the design since I was using Android studio, when making the layout for the appearances of the buttons and the texts the positioning of them was fixed which didn't allow me to move them freely until I did some further alterations to the positioning of the buttons as well as layouts for headers and boxes to hold the shopping lists and products.

Before starting to get used to API I had struggled with searching for an API as most of the APIs I searched online were charging a fee which not what I was looking for as well as it was for a broader service as well hence why I decided to use Outpan [1]. This was a legitimate free API that did what I wanted it to do as well as I could see people adding on the site products building it up from brands and what not which helped me in my case as when I scan a product it adds it to the database as I mentioned below.

One of the many issues that I had was when setting up the API [1] get and post requests of products, it always broke on run in the virtual machine when testing it out on my computer which was very hard. That was because it was not scanning properly until testing on a live machine then it worked fine, but only after making sure that the barcode reader was reading the barcode correctly which involved going online to find a better barcode reader to import as the one I was using was connected to a library of books rather than products one hence why I used Zxing (zebra crossing) [5]

To help me understand a little of what was going on I used Postman to help me test the API and see if products were correctly being sent there as well as being returned from there. Everything was working fine and I have added a few products with the use of Postman as a test which returned all the answers I was looking for.

One of the many issues were when I was trying to load the shopping lists (*section c)* it would always let you create two shopping lists but it always overrode the first one just leaving the second one. After checking the save and load method in my code and my load list I applied those changes.

There was an issue with how products will be displayed if the product being scanned in was too long i.e. Swan Menthol Filter Tips. The Product will be displayed with the full length of the text and the quantity and the delete and edit buttons would be pushed to the side making it impossible to press. The solution to this was to make a character display word limit, which I have set to be 15 character and if its more than that it will display it as "Swan Menthol …" which would make things easier to be viewed which I have mentioned in my report. Which I have also showed in the appendices in *(Appendix E*) as a fix to the issue.

The other issue was when I was trying to edit the products (*section d)* and try to delete them. Whenever I tried to delete the product on shopping list one, I would go back to the load list menu and click the second shopping list to add a product as soon as going back, I had found out that the load list only showed products name instead of the shopping list names.

The values had been switched, the output of the products as the names of the list and the name of the shopping list as products. The other issue was about saving and how to delete a specific product which was difficult. As whenever the user deletes a product the whole shopping list gets removed but after fixing it so that it would not happen again within the shoppingList class.

As for how the shopping list is being read, the shopping list reads the information from shopping list and what it holds as a number and it iterates by name of list and amount the list holds i.e. so the name of the shopping list will be read and the number of products it holds and the products as well as each product and their quantity and a new line for other products and it will stop the iteration after it hits another shopping list.

There was also an issue with the scanner sending and getting requests when the user would add a product see *(section a in code samples)* after applying changes the get and post were working well. See (*section b in code samples).*

Regarding the methodology for this project I have done a feature list following FDD [2](feature driven development) design to help me start on certain features and develop them and test them.

I will mention below my feature list for this project:

- Edit: to modify the name of the shopping list or the shopping list.
- Delete: the shopping list and as well as the products within the shopping list.
- Add shopping list: takes the user to the create section where he makes a new shopping list to be viewed and edited in the Load List section.
- Load list: where the created list is, accessible and can be edited and deleted.
- View: to view the shopping list that the user decides and can begin to add products to that shopping list.
- Add products: takes the user to an activity where there are 2 buttons where one is scan product and manually add product.
- Scan product: will open the camera and prompt the user to give permissions to have access to the camera, then the user scans the product and will be prompted with a message saying that the product is not in the database and if they would like to scan back or click yes to continue. Then the user will be taken to an activity where he will input the name and quantity after the data is being pulled from the database. Then become added to the shopping list.
- Manually add product: will take the to an activity where he will have to input the name of the product he wants and the quantity.
- Submit: processing the initial activity in the following activities in Add List and when adding of a product is being done and quantity as well as scanning a product, when the name and quantity is being confirmed.

After scanning a product and the error message appears prompting the user with two choices.

- Yes: takes the user to the next activity which enables him to add the name and quantity of the scanned product to be displayed to the shopping list.

- NO: which takes the user if pushed back to the scanner to re- scan a new product to be added.

After applying all the tests and repeating the same thing repeatedly with a minor change for each feature for the desired classes. I managed to create everything that I needed and tested each button to make sure it connects to the corresponding activity and does what it is supposed to do correctly.

All these problems were a major delay on the time frame I had given myself as well as the time frame of the project. Work was at a halt until issues were all solved and eventually the work was finished and everything was working as it should be.

There were a lot of revisions and changes that were made, by nearing the end of this project, All the changes have been implemented and bugs have been dealt cleared and everything was running smoothly and the work was completed within the time frame of this project deadline.

As of writing this report the finished project has all been fixed with its changes and has all been implemented meeting the requirements of the project description as well as meeting the deadline.

There was panic that I would not be able to make it with all the errors in the beginning of this project, and that it would slow the final result but as mentioned above through a lot of rewriting my code and removing of useless methods everything was sorted by the end of the project.

4. Testing

Below I will go into further detail on the testing of my application. The testing that I have run are on four things which will be mentioned below with sample code in the appendices section.

I will also include below why I have not included other forms of Testing in my application as I was running short on time as well as there were other factors such as getting things to work well, before running the other forms of testing which have been mentioned below.

4.1. Overall Approach to Testing

The approach to testing that I have done is a Feature driven development one, where each method before creation is testing and then applied to the build after running through many checks.

The testing that has been performed were for the creation of a shopping list as well as loading of a shopping list. The other test was for manipulation of the shopping list with the edit, delete of a product. In the appendices will be sample code of the tests. The type of tests I have run are a Junit testing and user testing.

As for the other forms of testing such as automated testing, I have not done any automated tests. Also, if given more time I would like to implement automated testing in my program and see how it fares. I just didn’t have any time to do any of those tests as I was rushing for time.

The other form of testing that I have not included in my report is Integration testing as I ran out of time and everything that I needed to test was working fine as is it should.

I also have not included Stress testing as I believe that through programming this project I have stress tested all the methods and classes and variables I needed to and by the end of it as mentioned above everything was working as it should be.

I may have no extra test classes but I have tested everything before finalising my choices for design as I was programming the application, as well as for code to get to where I am in this project which is a completed project which met the requirements asked which I have mentioned above as well as in my critical evaluation section.

4.1.1. Unit Tests

As mentioned in the overall subsection I have run unit tests on a few methods within my application, the methods I have run the tests are on the creation and manipulation of products within the shopping list and the Loading of a shopping list as well as the editing and deleting of a shopping List in the Load List.

The first test was to check the load method, to make sure that when users create shopping lists it would save and load the lists with the products inside them still intact and not missing see (*section f in code samples).*

The second test was to check if the user could add products into the shopping list and if the app closed will the product be still available after reloading the application, which was tested and was available. See *(section e in code samples)*

After creation of a product was tested, I tested how the user can edit the product within the shopping list so the edit test was run and succeeded which is also in the appendices in *(section e)*

the final test ran was the deletion (delete Test) of a product within the shopping list, where the user would delete the product the wanted and go back to the menu page and close or load the shopping list again and find that the product was not deleted or it deleted the wrong product overwriting the previous product.

After a few runs with the debugger I managed to fix the delete method I missed a few extra params, I also seemed to have to have forgotten to put the location of the test class was in the wrong place. The test succeeded. The sample code for the delete portion of this code is below in the appendices in the code sample *(section e of code samples).*

The other test that I have also performed was to make sure that the scanner was working well when scanning products as well as adding scanned products. The result for the test was when the user scans a product that Is not in the database, it will prompt the user to add it to the database which is mentioned in the implementation section as an issue.

The test passed as the user scanned something for the first time it will add the product to the database so that the user will not have to scan it again. As it will just pull the data from the products API and display it in the name of the product and become added to the shopping list.

4.2. User Testing

The User testing that I have performed was to make sure that the user could create a shopping list and be able to manipulate the shopping list, by editing the name of the shopping list as well as deleting it and creating multiple lists and being able to access them.

The other user test was to make sure that the user could be able to add products to their desired shopping list and manipulate the data within it such as change the name of the product and the quantity as well as deleting a product.

The tests all passed as the user could do all the things mentioned above, only after I have changed a few variables as well as fixing the add shopping list method to creating multiple shopping lists and trying to edit a shopping list.

When the user goes back to look in the load list the products the user data that always was being displayed in both of the shopping lists, was the name of the products instead of the shopping list name and the shopping list names were in the products section.

This involved a lot of fixing with the buffer reader to make sure that they were displaying it all properly, as well as features for the scanner were fixed to make sure the input and output stream readers were properly reading the information and returning actual products.

The testing was done by family and friends as well as colleagues, who shared their input as well as gave me more insight to what I could improve on and further improve on my code.

5. Critical Evaluation

For my critical analysis, there were a lot of parts of the project that were going perfectly the way I wanted it to. In areas of building everything as well as programming the buttons and adding the shopping list and building the scanner as well as the connection to the API so scanned products can be returned with names.

I enjoyed building the layouts and the activities for each section everything was working well no errors there. I did struggle a bit to get used to the IDE I was using as this is the first time I have used Android studio which I mention in my report, I have had to teach myself to use a brand-new IDE which was also interesting in its own aspect as I now enjoy using Android Studio and look forward to doing more personal projects with it.

One of the other things that worked well was the fact that I could send products to the Products database which was also a first for me as I have not used an API before but was a good learning experience for me. I would improve on the Products section in my code by making it so that it saves the first 100 scans or I would let the user decide to either let the system do it or they can save products they want manually so when they scan it everything will be called from the first time scan they have done in the beginning. I would do this as in the future or I could have done it in my application if I was given more time, or as a future plan for the application.

The other thing would have to the save method and load which I am glad is working which was a hassle to implement as it would sometimes work and for some weird reason it would break even though there was nothing wrong and I would have to clear the log with a line of code which I have marked out and kept in my class which I will be fixing in the future as well probably letting the user decide where they can save their shopping list and send it off which I have mentioned below.

If there were any areas that I would mention that I need to improve on it would have to be more thorough testing in the sense that I have not performed any UI testing, as I was short on time and if given more I believe that I could have presented a more complete project.

In areas such as the testing as well implementation which I have mentioned in the above sections, putting in a share method which I initially thought was possible if everything was accomplished before the deadline. Although I never got to apply all that I wanted to this application there is a lot of room for improvement.

I would also try and make as an improvement for future work a more modern design and more features, for the future work on this application I would like to add more features as well as revamp of the layout as well as implement a few things such as a total for the products within the shopping list.

As for the areas within my project that could be improved on if given more time I believe that I could have made applied a few more tests such as automated testing, and UI testing as well as stress testing. As well as make it so that users could share their shopping lists with other people through Facebook and Twitter as well as other social media platforms or by email.

Also in the add products section I could make it so when the users are in manually adding section where they add the product, they can have an autofill in case they forget half the word, or for those who don’t know what the product is called so a drop down section will be available under the name section with a few products names matching the users input all sorted alphabetically which will help them with their choices.

I would also like to be able to implement a QR scanner as some products have a QR code on them as that might be useful to some shoppers. As well as a way for them to add the discounts from the shops they visit to be applied to the shopping products to give them a total in case they are under financial stress, so they know how much to spend on future products the next time they go shopping. Which will make shopping go much smoother.

I would also like to make the waiting page period in my application be a little bit better designed as I believe it could have been done better if given more time and a lot more research online for sources. I would improve on it by making the waiting screen with a loading bar which will be much more appealing as well as be more modern.

For the products name I would like to make it so that there is a hover over text notifying to the user what the user should do in the case of creation of the shopping list, as well as in the editing of a shopping list and in a product guiding them to either edit the name or quantity of the product or shopping list.

It would be between implementing that or I would like to implement a small icon in the top left. When the user clicks(triggers) it everything within the shopping list would be marked and the user can edit or delete just like in the iPhone marking and deletion of programs, but in this case, it will be of the products.

For the products layout if given more time I would improve on functionality in the aspects of adding a product and deleting as well as I would make it so that the user can move products around. I would either implement the iPhone method of deleting and moving or I would make it so that there were icons in the products, so that the user would see on the top right a marker and when clicked he can freely choose what to edit delete and move around.

Another improved idea for my application would be to make it so that when the edit is clicked the user can click on a product quantity and instead of opening up the edit page they can go ahead and do the edit from the same page with a little dropdown number to help them or they can delete the number and re-write it again.

I also would like to implement a notification for the user notifying them that they have products within their shopping lists to either add or save as well as notify them about the certain discounts that can be applied to their shopping lists.

It also allows you to add products to the shopping lists which can be renamed as well as deleted. These changes can also be done to the shopping lists the user creates. when the user adds a product through the scanner it scans the barcode of the product and sends the data (barcode number) and returns the product name all done through an internet connection and if not you will be prompted with a message saying no internet or that the API is down.

If all the requirements are met then the product name will be returned so that the user can add the quantity to be displayed and further edited in the shopping list.

One of the improvements that I would implement if I had more time in the project is to make more Testing for my application if given more time, as I believe that if I had more testing methods my report would have been better in content aspects but for the future work I will be testing in the following, Automated Testing, User Interface testing, Stress Testing as well as a more detailed User Based Testing after I have applied all the above mentioned changes and applied them to the application, so that I may be given some extra feedback and apply them to a future update of the application..

Also in the case of making the application better if I had more time I could have made it so that the users could see a total of products as well as product prices from the desired shops but I did not have much time to apply that to my program as there were a lot of other things taking precedence.

In the ways of improving the application I could make it so that the users save scanned products which I have mentioned above. I will be applying that to the future of this project.

As I mentioned above about further improvements I plan to keep on making more updates on this application and apply all the changes that I have mentioned to making this application much more better.

I thoroughly enjoyed doing this project as I have learned a lot throughout this project and If given the opportunity to do this again I would certainly take the chance as well as apply all that I have learnt in this project and apply it to the future ones.

There are a lot of things I could do to improve on this application which I have mentioned above in much detail as well as how the application could have been done better, I plan to apply a lot of changes and maybe in an update after this is all handed in, as once again I would like to say I really enjoying working on this project as I have found that I enjoy working on applications on this scale as well as in this field so I might take this up on my free time or as a career choice down the line.

The last change that I thought I could complete as I have mentioned in the implementation is that if I could professionally package the application as having an icon that shows that is the shopping helper application rather than the stock android image but there was not time as well as there were complications with doing it.

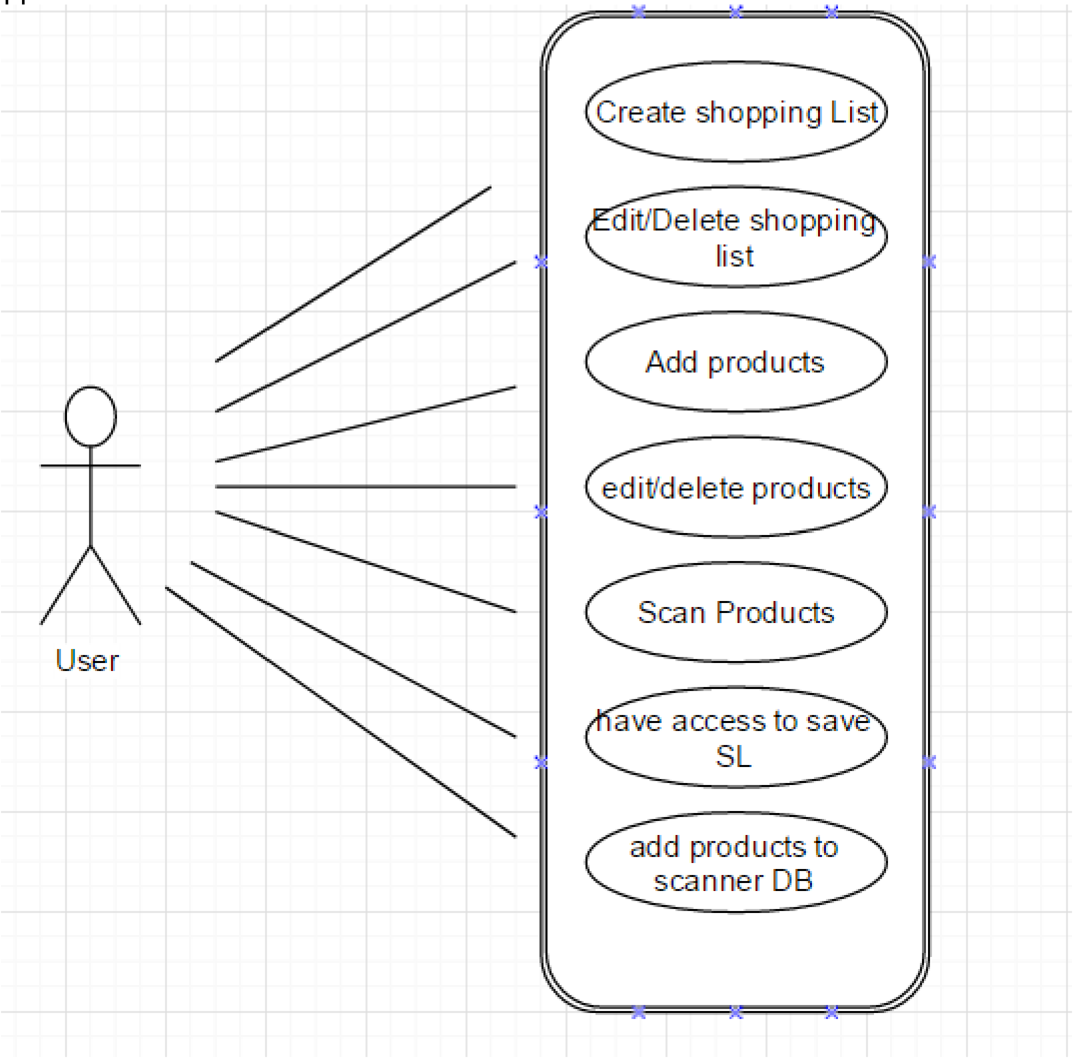
To conclude I have met all the requirements of my projects description, as well as have met my deadline of this project. My application does the following. It is a shopping helper which lets you create as hoping list as well as multiple shopping lists. It lets you name the shopping list as well as edit its name, it also lets you delete the shopping list.

You can also add products in two ways either by scanning in using the camera to scan barcodes or to manually input the name and quantity of the product, after which will be displayed depending on either of your choices the result will be displayed in the shopping list of your choosing to be manipulated.

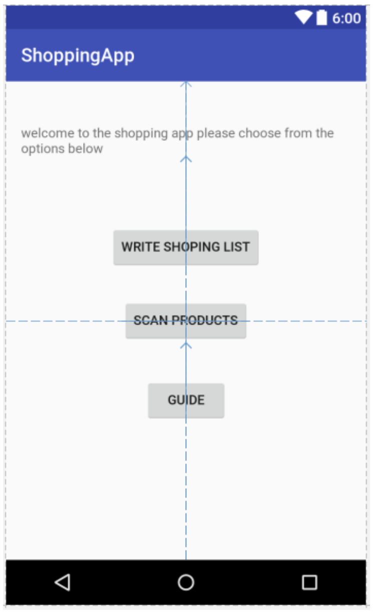
Once again, I would like to mention that everything is saved internally so when the user decides that they don’t want the application they can uninstall it and everything regarding the application will be removed from your system leaving nothing behind.

6. Appendices

Appendix A: which shows what the user can have access to.

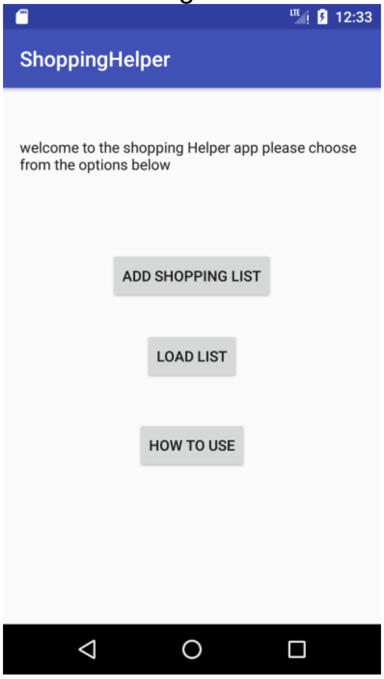


Appendix B: which holds the initial design which was submitted during the start of the project



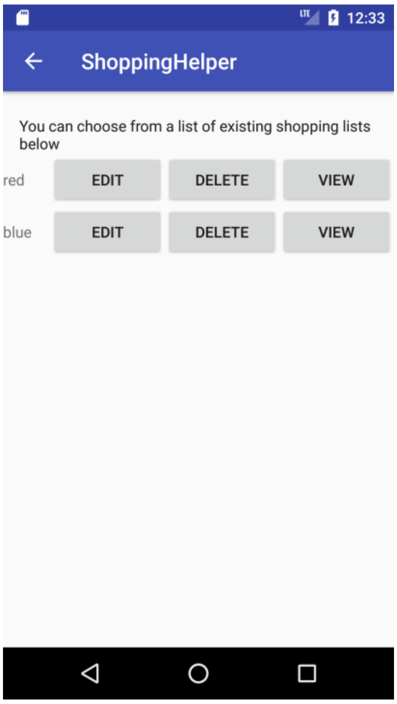
welcome to the shopping app please choose from the options below

Appendix C: which holds the modern design choice as well as latest changes.



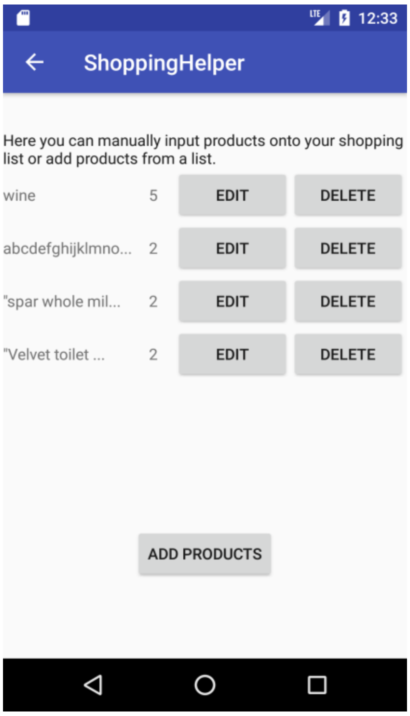
welcome to the shopping Helper app please choose from the options below

Appendix D: shows the shopping list names within the load list section and can be accessed and manipulated.



You can choose from a list of existing shopping lists below

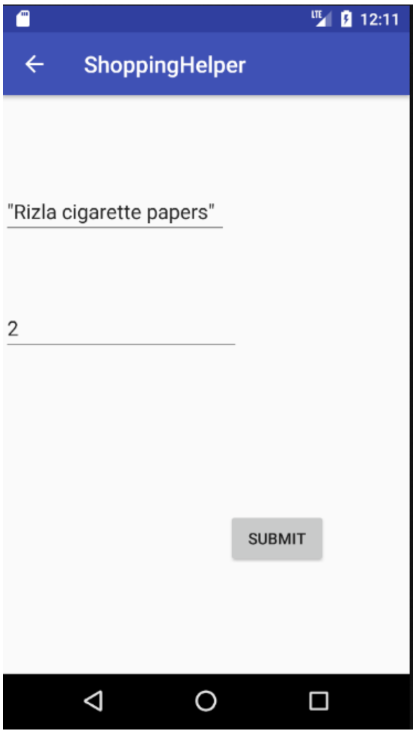
Appendix E: the products list when the shopping list is accessed. Where the user can add products to be edited or deleted.



Here you can manually input products onto your shopping list or add products from a list.

wine	5	EDIT	DELETE
abcdefghijklmno...	2	EDIT	DELETE
"spar whole mil...	2	EDIT	DELETE
"Velvet toilet ...	2	EDIT	DELETE

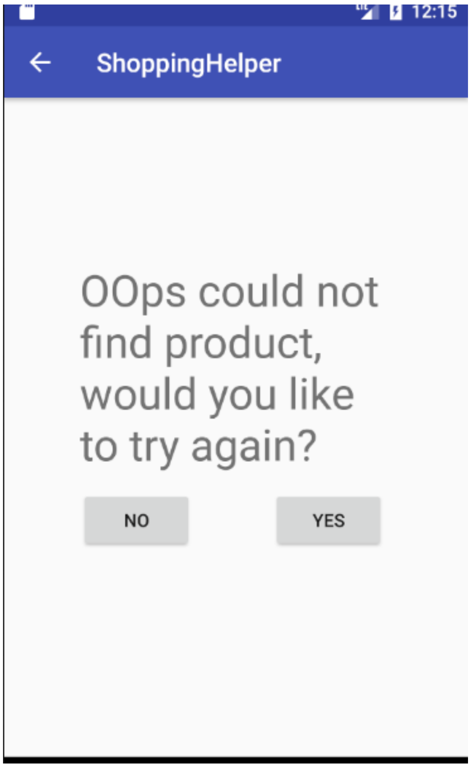
Appendix F: when scanned product is pulled from the database just add the quantity.



"Rizla cigarette papers"

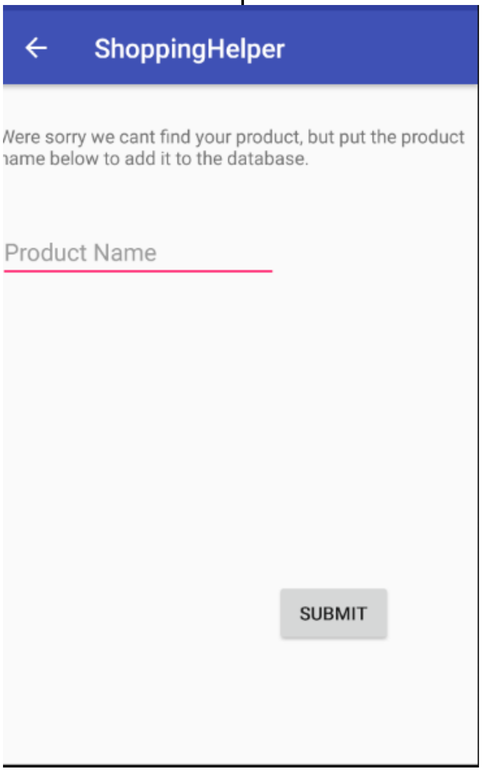
2

Appendix G: when product that is not recognised by the database tells the user to rescan and add it to the database when clicking yes. If no then it takes u back to the scanner.



OOps could not find product, would you like to try again?

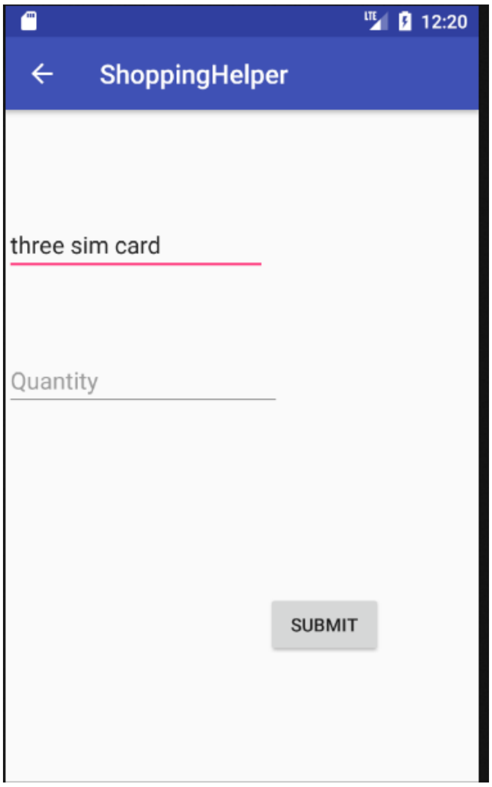
Appendix H: Lets the user add the scanned product if user clicks yes to the database so when scanning again it remembers the product.



Were sorry we cant find your product, but put the product name below to add it to the database.

Product Name

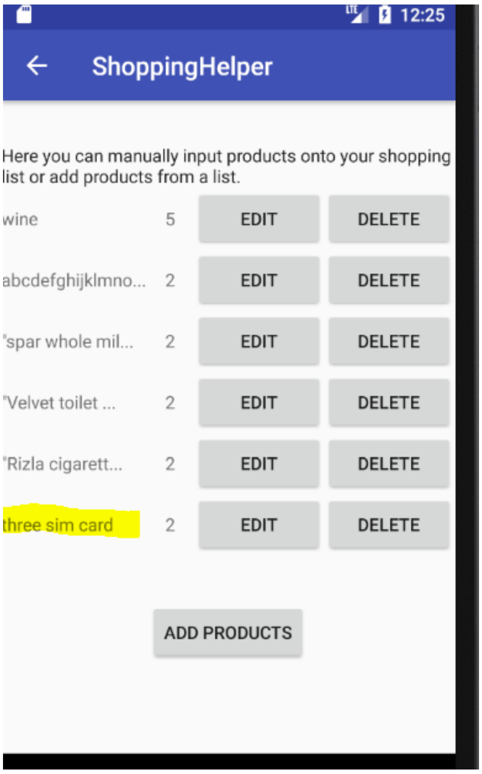
Appendix I: when scanned product is sent to the database it retrieves the name as well as has the quantity field added to it. Then adds it to the shopping list.



three sim card

Quantity

Appendix J: you can see the product that the user has scanned and added has been displayed below.

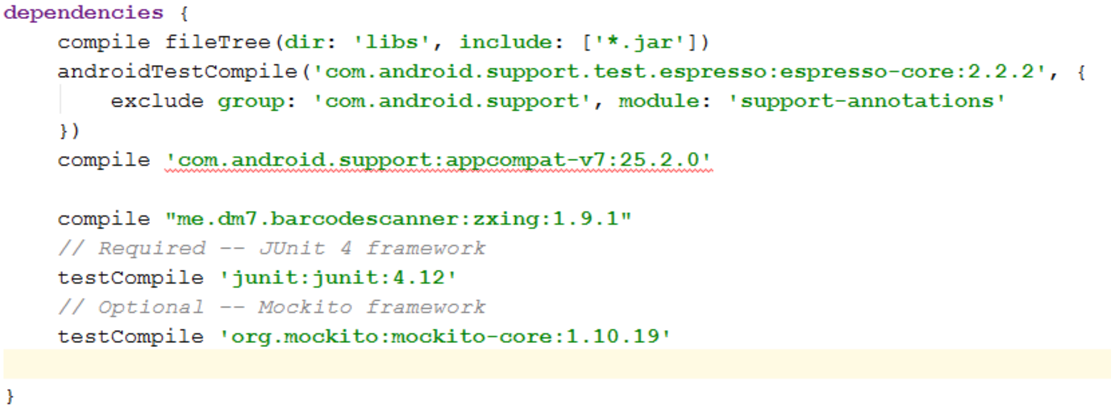


Here you can manually input products onto your shopping list or add products from a list.

wine	5	EDIT	DELETE
abcdefghijklmno...	2	EDIT	DELETE
'spar whole mil...	2	EDIT	DELETE
'Velvet toilet ...	2	EDIT	DELETE
'Rizla cigarett...	2	EDIT	DELETE
three sim card	2	EDIT	DELETE

A. Third-Party Code and Libraries

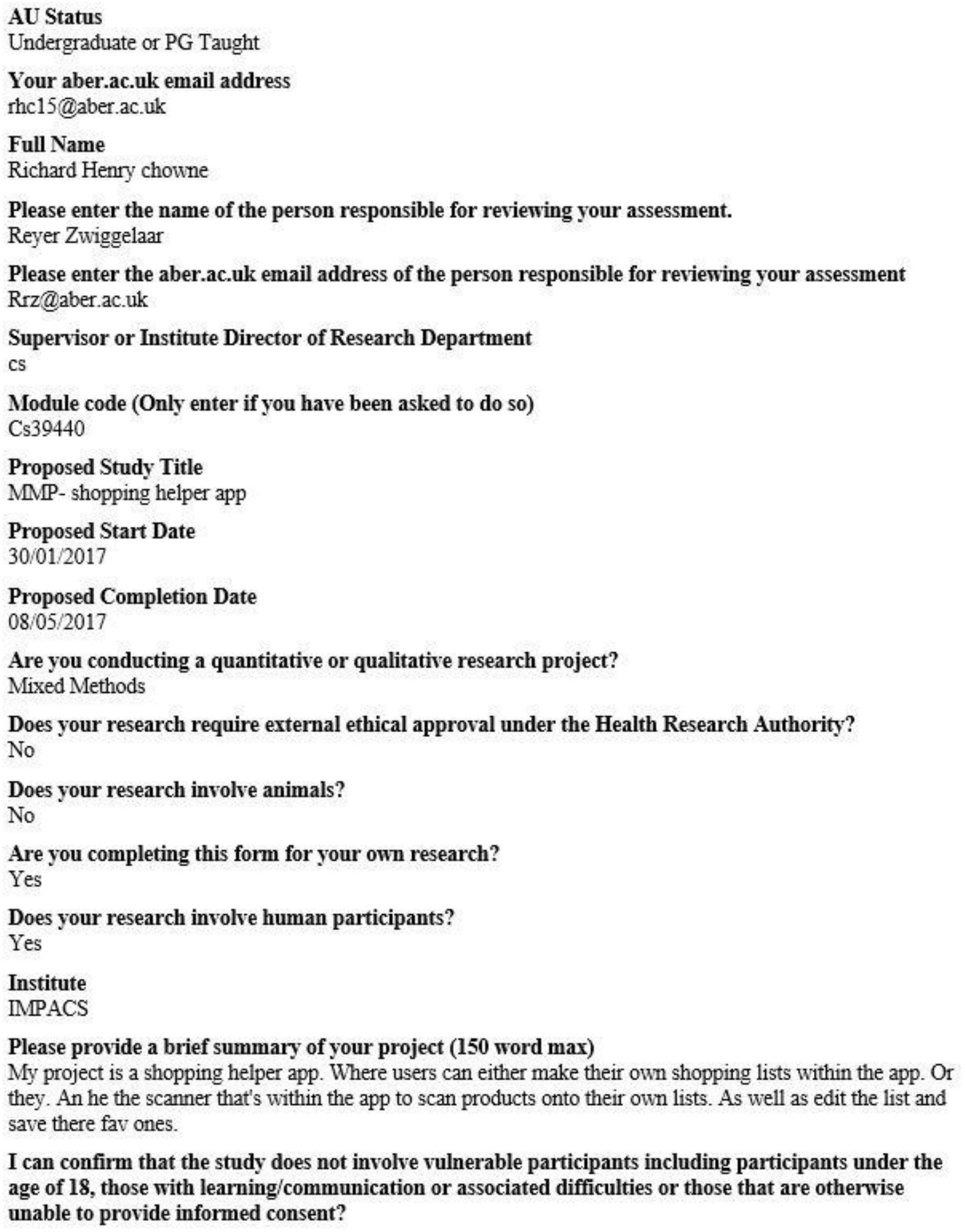
(Section A): using Zxing for the barcode scanner.



```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.2.0'

    compile "me.dm7.barcodescanner:zxing:1.9.1"
    // Required -- JUnit 4 framework
    testCompile 'junit:junit:4.12'
    // Optional -- Mockito framework
    testCompile 'org.mockito:mockito-core:1.10.19'

}
```

B. Ethics Submission

This appendix includes a copy of the ethics submission for the project.
Ethics Application Number: 6714



AU Status
Undergraduate or PG Taught

Your aber.ac.uk email address
rhc15@aber.ac.uk

Full Name
Richard Henry chowne

Please enter the name of the person responsible for reviewing your assessment.
Reyer Zwiggelaar

Please enter the aber.ac.uk email address of the person responsible for reviewing your assessment
Rrz@aber.ac.uk

Supervisor or Institute Director of Research Department
cs

Module code (Only enter if you have been asked to do so)
Cs39440

Proposed Study Title
MMP- shopping helper app

Proposed Start Date
30/01/2017

Proposed Completion Date
08/05/2017

Are you conducting a quantitative or qualitative research project?
Mixed Methods

Does your research require external ethical approval under the Health Research Authority?
No

Does your research involve animals?
No

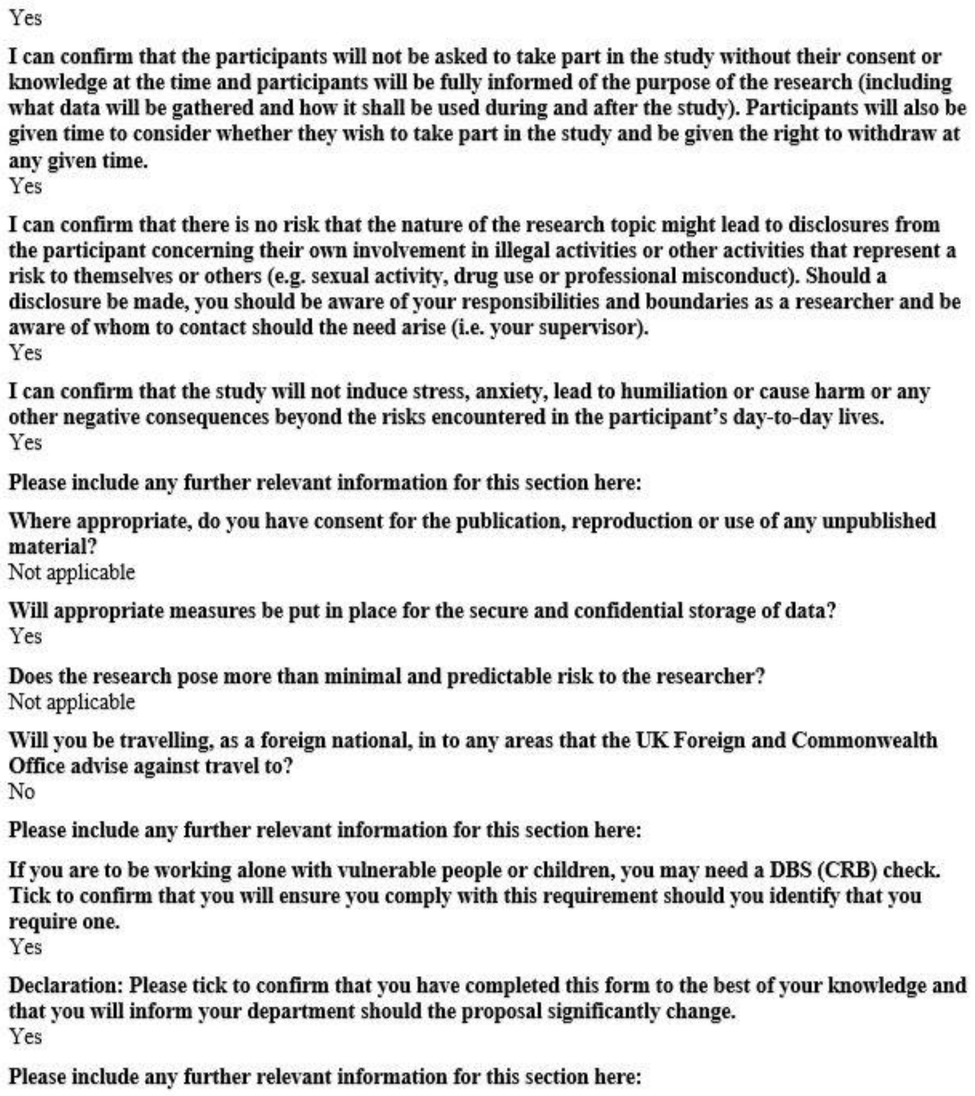
Are you completing this form for your own research?
Yes

Does your research involve human participants?
Yes

Institute
IMPACS

Please provide a brief summary of your project (150 word max)
My project is a shopping helper app. Where users can either make their own shopping lists within the app. Or they. An he the scanner that's within the app to scan products onto their own lists. As well as edit the list and save there fav ones.

I can confirm that the study does not involve vulnerable participants including participants under the age of 18, those with learning/communication or associated difficulties or those that are otherwise unable to provide informed consent?



Yes

I can confirm that the participants will not be asked to take part in the study without their consent or knowledge at the time and participants will be fully informed of the purpose of the research (including what data will be gathered and how it shall be used during and after the study). Participants will also be given time to consider whether they wish to take part in the study and be given the right to withdraw at any given time.
Yes

I can confirm that there is no risk that the nature of the research topic might lead to disclosures from the participant concerning their own involvement in illegal activities or other activities that represent a risk to themselves or others (e.g. sexual activity, drug use or professional misconduct). Should a disclosure be made, you should be aware of your responsibilities and boundaries as a researcher and be aware of whom to contact should the need arise (i.e. your supervisor).
Yes

I can confirm that the study will not induce stress, anxiety, lead to humiliation or cause harm or any other negative consequences beyond the risks encountered in the participant's day-to-day lives.
Yes

Please include any further relevant information for this section here:

Where appropriate, do you have consent for the publication, reproduction or use of any unpublished material?
Not applicable

Will appropriate measures be put in place for the secure and confidential storage of data?
Yes

Does the research pose more than minimal and predictable risk to the researcher?
Not applicable

Will you be travelling, as a foreign national, in to any areas that the UK Foreign and Commonwealth Office advise against travel to?
No

Please include any further relevant information for this section here:

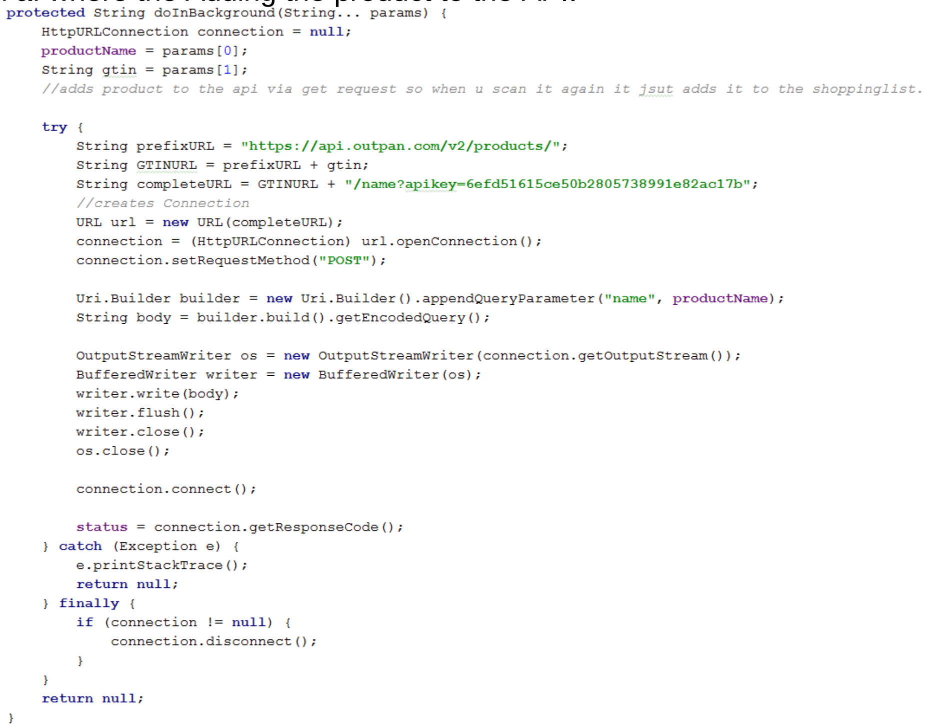
If you are to be working alone with vulnerable people or children, you may need a DBS (CRB) check. Tick to confirm that you will ensure you comply with this requirement should you identify that you require one.
Yes

Declaration: Please tick to confirm that you have completed this form to the best of your knowledge and that you will inform your department should the proposal significantly change.
Yes

Please include any further relevant information for this section here:

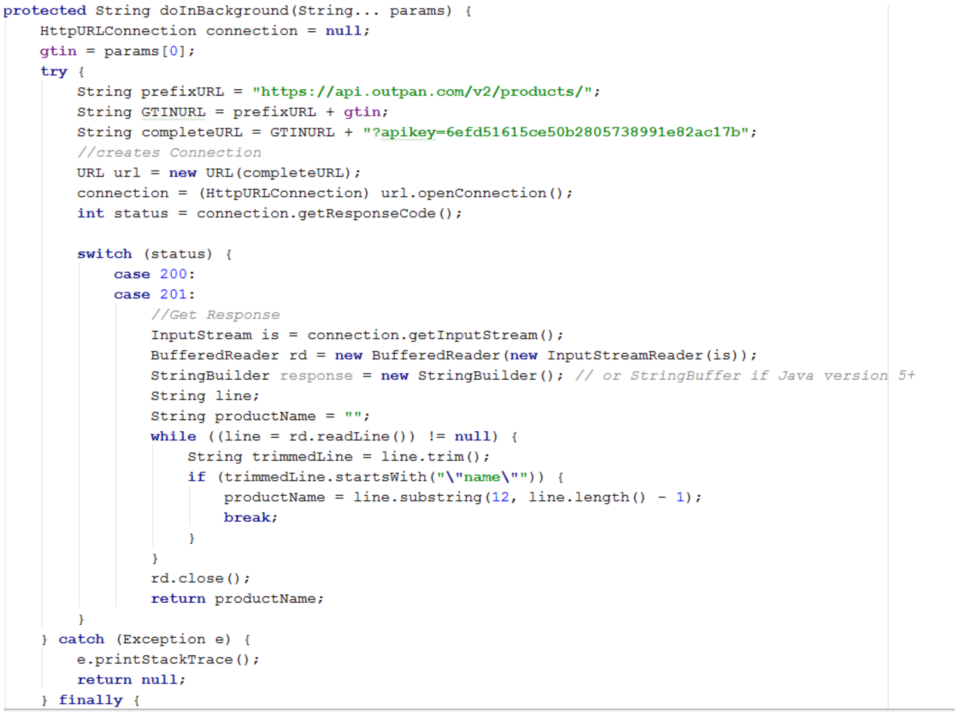
C. Code Samples

Section a: where the Adding the product to the API.



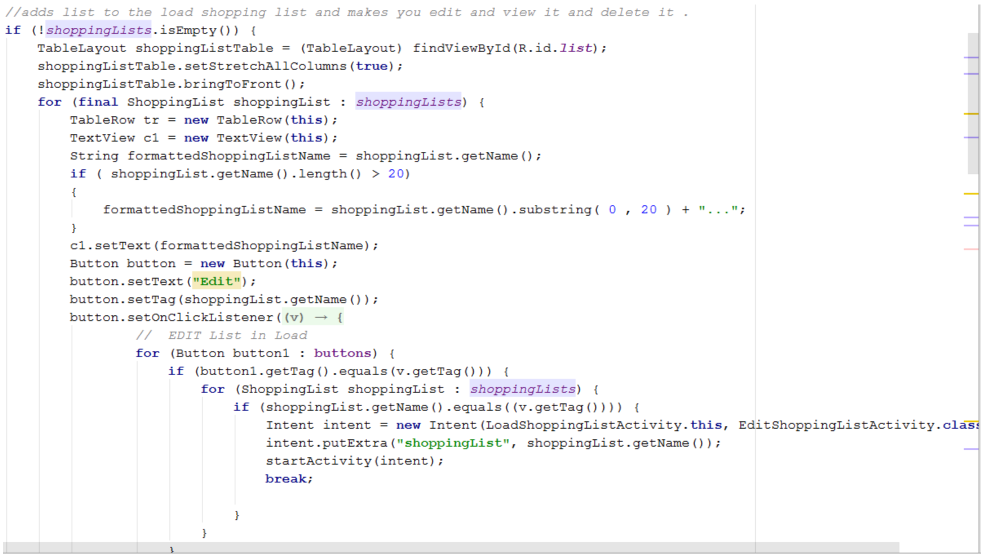
```
protected String doInBackground(String... params) {
    HttpURLConnection connection = null;
    productName = params[0];
    String gtin = params[1];
    //adds product to the api via get request so when u scan it again it jsut adds it to the shoppinglist.

    try {
        String prefixURL = "https://api.outpan.com/v2/products/";
        String GTINURL = prefixURL + gtin;
        String completeURL = GTINURL + "/name?apikey=6efd51615ce50b2805738991e82ac17b";
        //creates Connection
        URL url = new URL(completeURL);
        connection = (HttpURLConnection) url.openConnection();
        connection.setRequestMethod("POST");

        Uri.Builder builder = new Uri.Builder().appendQueryParameter("name", productName);
        String body = builder.build().getEncodedQuery();

        OutputStreamWriter os = new OutputStreamWriter(connection.getOutputStream());
        BufferedWriter writer = new BufferedWriter(os);
        writer.write(body);
        writer.flush();
        writer.close();
        os.close();

        connection.connect();

        status = connection.getResponseCode();
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    } finally {
        if (connection != null) {
            connection.disconnect();
        }
    }
    return null;
}
```

Section b: Getting the product from the API.

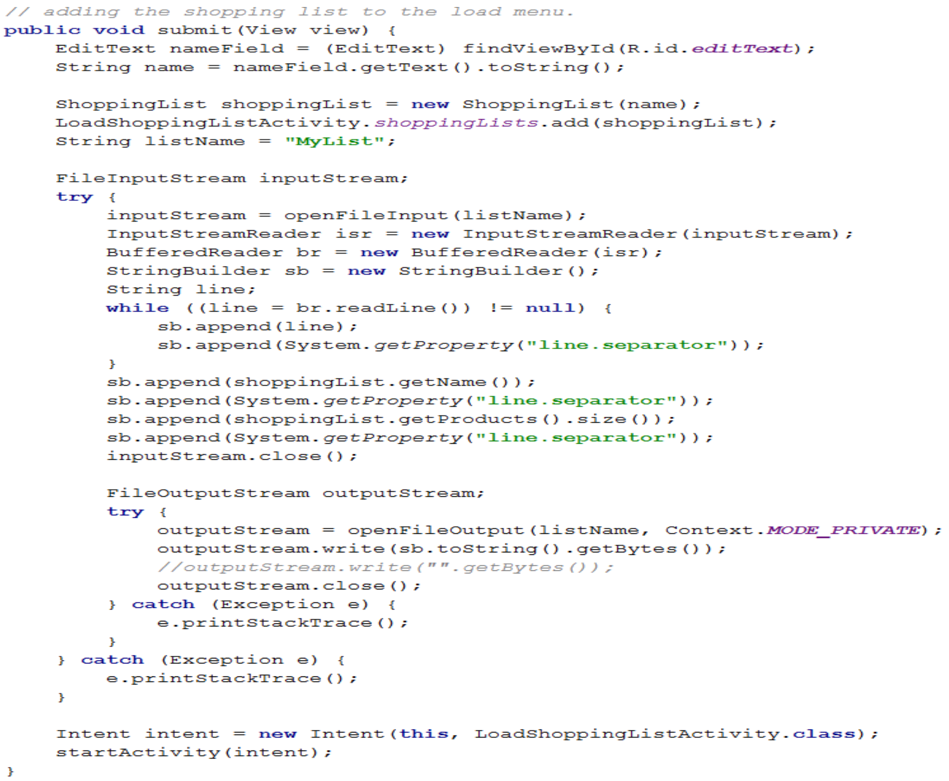


```
protected String doInBackground(String... params) {
    HttpURLConnection connection = null;
    gtin = params[0];
    try {
        String prefixURL = "https://api.outpan.com/v2/products/";
        String GTINURL = prefixURL + gtin;
        String completeURL = GTINURL + "?apikey=6efd51615ce50b2805738991e82ac17b";
        //creates Connection
        URL url = new URL(completeURL);
        connection = (HttpURLConnection) url.openConnection();
        int status = connection.getResponseCode();

        switch (status) {
            case 200:
            case 201:
                //Get Response
                InputStream is = connection.getInputStream();
                BufferedReader rd = new BufferedReader(new InputStreamReader(is));
                StringBuilder response = new StringBuilder(); // or StringBuffer if Java version 5+
                String line;
                String productName = "";
                while ((line = rd.readLine()) != null) {
                    String trimmedLine = line.trim();
                    if (trimmedLine.startsWith("\"name\"")) {
                        productName = line.substring(12, line.length() - 1);
                        break;
                    }
                }
                rd.close();
                return productName;
        }
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    } finally {
```

Section c: the fixed method for the load list to add shopping list to be displayed and add shopping list to load.

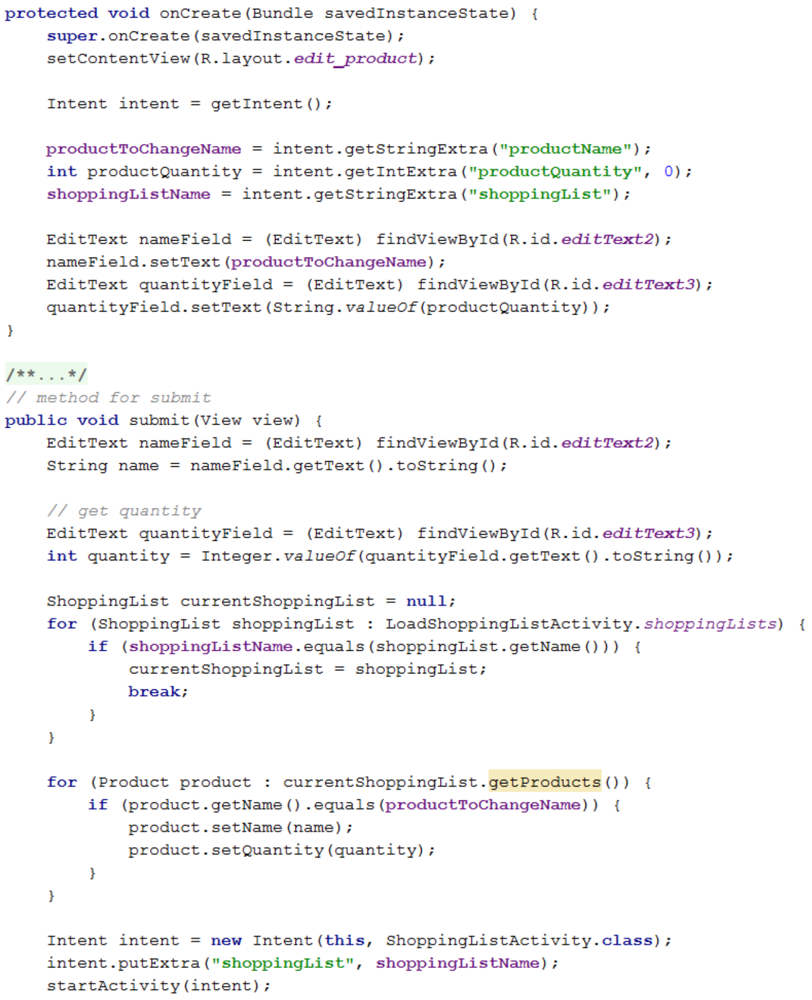


```
//adds list to the load shopping list and makes you edit and view it and delete it .
if (!shoppingLists.isEmpty()) {
    TableLayout shoppingListTable = (TableLayout) findViewById(R.id.list);
    shoppingListTable.setStretchAllColumns(true);
    shoppingListTable.bringToFront();
    for (final ShoppingList shoppingList : shoppingLists) {
        TableRow tr = new TableRow(this);
        TextView c1 = new TextView(this);
        String formattedShoppingListName = shoppingList.getName();
        if ( shoppingList.getName().length() > 20)
        {
            formattedShoppingListName = shoppingList.getName().substring( 0 , 20 ) + "...";
        }
        c1.setText(formattedShoppingListName);
        Button button = new Button(this);
        button.setText("Edit");
        button.setTag(shoppingList.getName());
        button.setOnClickListener((v) → {
                //  EDIT List in Load
                for (Button button1 : buttons) {
                    if (button1.getTag().equals(v.getTag())) {
                        for (ShoppingList shoppingList : shoppingLists) {
                            if (shoppingList.getName().equals((v.getTag()))) {
                                Intent intent = new Intent(LoadShoppingListActivity.this, EditShoppingListActivity.clas
                                intent.putExtra("shoppingList", shoppingList.getName());
                                startActivity(intent);
                                break;

                            }
                        }
                    }
```



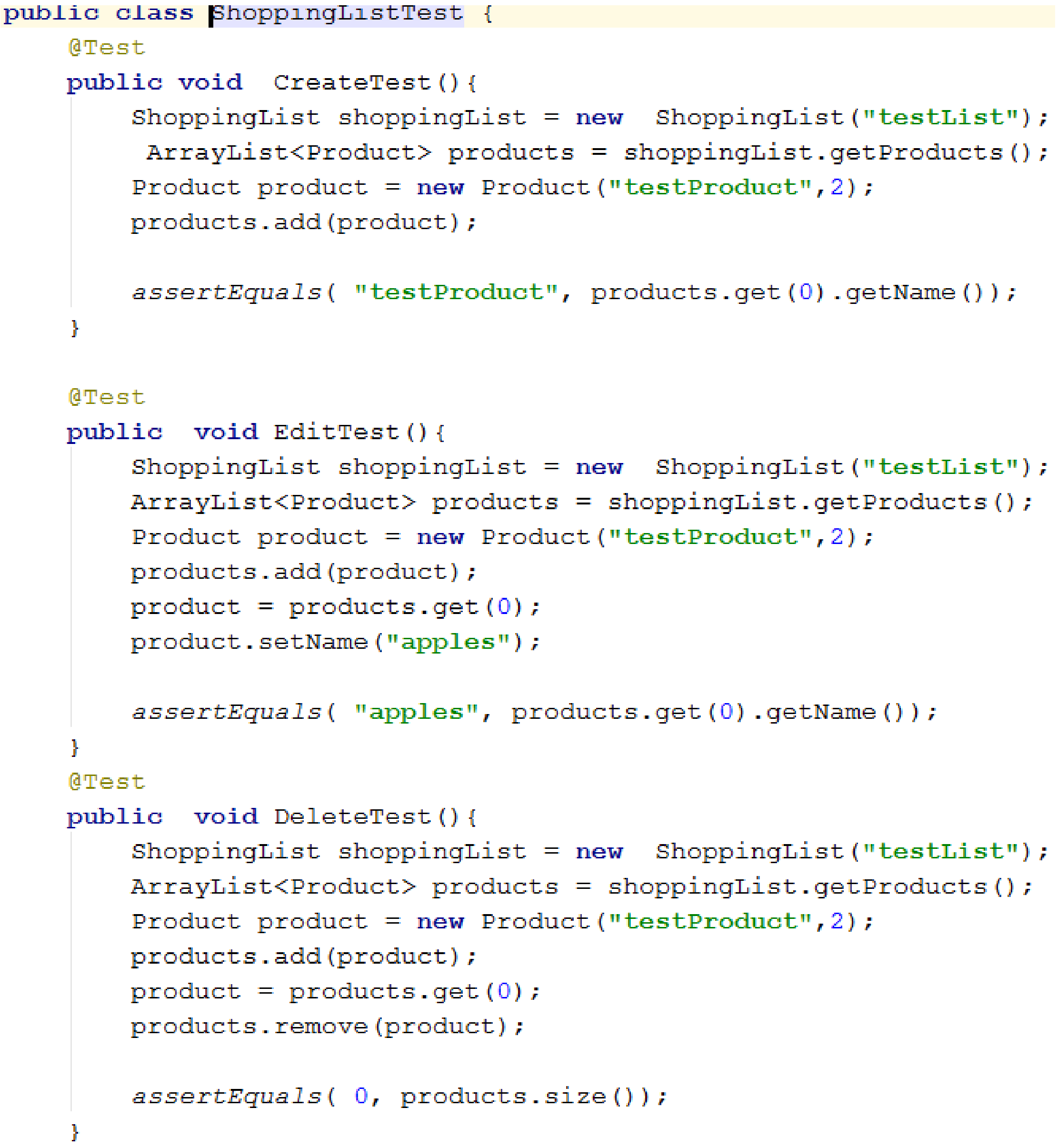
```
// adding the shopping list to the load menu.
public void submit(View view) {
    EditText nameField = (EditText) findViewById(R.id.editText);
    String name = nameField.getText().toString();

    ShoppingList shoppingList = new ShoppingList(name);
    LoadShoppingListActivity.shoppingLists.add(shoppingList);
    String listName = "MyList";

    FileInputStream inputStream;
    try {
        inputStream = openFileInput(listName);
        InputStreamReader isr = new InputStreamReader(inputStream);
        BufferedReader br = new BufferedReader(isr);
        StringBuilder sb = new StringBuilder();
        String line;
        while ((line = br.readLine()) != null) {
            sb.append(line);
            sb.append(System.getProperty("line.separator"));
        }
        sb.append(shoppingList.getName());
        sb.append(System.getProperty("line.separator"));
        sb.append(shoppingList.getProducts().size());
        sb.append(System.getProperty("line.separator"));
        inputStream.close();

        FileOutputStream outputStream;
        try {
            outputStream = openFileOutput(listName, Context.MODE_PRIVATE);
            outputStream.write(sb.toString().getBytes());
            //outputStream.write("".getBytes());
            outputStream.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }

    Intent intent = new Intent(this, LoadShoppingListActivity.class);
    startActivity(intent);
}
```

Section d: the fixed edit method that lets user edit products.



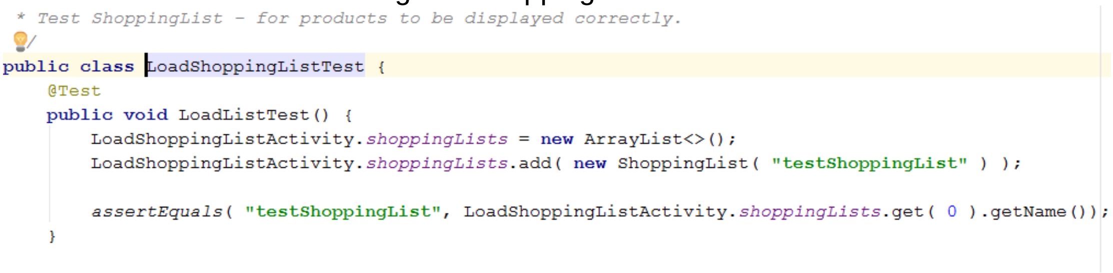
```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.edit_product);

    Intent intent = getIntent();

    productToChangeName = intent.getStringExtra("productName");
    int productQuantity = intent.getIntExtra("productQuantity", 0);
    shoppingListName = intent.getStringExtra("shoppingList");

    EditText nameField = (EditText) findViewById(R.id.editText2);
    nameField.setText(productToChangeName);
    EditText quantityField = (EditText) findViewById(R.id.editText3);
    quantityField.setText(String.valueOf(productQuantity));
}

/**...*/
// method for submit
public void submit(View view) {
    EditText nameField = (EditText) findViewById(R.id.editText2);
    String name = nameField.getText().toString();

    // get quantity
    EditText quantityField = (EditText) findViewById(R.id.editText3);
    int quantity = Integer.valueOf(quantityField.getText().toString());

    ShoppingList currentShoppingList = null;
    for (ShoppingList shoppingList : LoadShoppingListActivity.shoppingLists) {
        if (shoppingListName.equals(shoppingList.getName())) {
            currentShoppingList = shoppingList;
            break;
        }
    }

    for (Product product : currentShoppingList.getProducts()) {
        if (product.getName().equals(productToChangeName)) {
            product.setName(name);
            product.setQuantity(quantity);
        }
    }

    Intent intent = new Intent(this, ShoppingListActivity.class);
    intent.putExtra("shoppingList", shoppingListName);
    startActivity(intent);
```

Section e: test method for creating a product as well as deleting and editing it from the shopping list.



```
public class ShoppingListTest {
    @Test
    public void  CreateTest(){
        ShoppingList shoppingList = new  ShoppingList("testList");
         ArrayList<Product> products = shoppingList.getProducts();
        Product product  = new Product("testProduct",2);
        products.add(product);

        assertEquals( "testProduct", products.get(0).getName());
    }

    @Test
    public  void EditTest(){
        ShoppingList shoppingList = new  ShoppingList("testList");
        ArrayList<Product> products = shoppingList.getProducts();
        Product product  = new Product("testProduct",2);
        products.add(product);
        product = products.get(0);
        product.setName("apples");

        assertEquals( "apples", products.get(0).getName());
    }
    @Test
    public  void DeleteTest(){
        ShoppingList shoppingList = new  ShoppingList("testList");
        ArrayList<Product> products = shoppingList.getProducts();
        Product product  = new Product("testProduct",2);
        products.add(product);
        product = products.get(0);
        products.remove(product);

        assertEquals( 0, products.size());
    }
```

Section f: Test to check loading of a shopping list.



```
 * Test ShoppingList - for products to be displayed correctly.
 */
public class LoadShoppingListTest {
    @Test
    public void LoadListTest() {
        LoadShoppingListActivity.shoppingLists = new ArrayList<>();
        LoadShoppingListActivity.shoppingLists.add( new ShoppingList( "testShoppingList" ) );

        assertEquals( "testShoppingList", LoadShoppingListActivity.shoppingLists.get( 0 ).getName());
    }
```

7. Bibliography

[1] API Website of everything, “Website: Outpan,” [Online]. Available: www.outpan.mixnode.com. [Accessed 4 May 2017].

This is the website that has the API that I have used for my project. It holds the products API that I am using in my project as it is required because it holds the database of products as well as where the get and post requests are sent to

[2] S. W. Ambler, “Agile modeling,” Ambysoft Inc., [Online]. Available: http://agilemodeling.com/essays/fdd.htm. [Accessed 15 April 2017].

Where the information I have gathered from The FDD as well as explaining it in a way that I have understood as well as with visual interpretation.

[3] SDLC Waterfall Model, “Tutorials Point,” [Online]. Available: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm. [Accessed 5 May 2017].

Holds the waterfall design information in the sense that I used this as a starting point in my project to get started but then changed to FDD.

[4] Android Saving files, “ Developer Andriod,” Developers, [Online]. Available: https://developer.android.com/training/basics/data-storage/files.html#WriteInternalStorage. [Accessed 4 May 2017].

The site that contains information on the save states as well as the benefits of internal saving vs external which I used in my project

[5] Zxing masterfile, “Github,” Zxing masterfile download, [Online]. Available: https://github.com/zxing/zxing. [Accessed 19 04 2017].

The library used in my project that holds the barcode scanner feature on my project. I used this site to get the information.