

MSc Final Project

School of Computing & Information Engineering

Name:	Emma Lynch
Student Number:	B00444529
Project Title:	Shopping Centre App
Supervisor:	Dr Daniel Kelly
Second Marker:	Dr Joan Condell
Submission Date:	01/09/2016

Contents

Chapters	5
Abstract:	6
Chapter 1: The Problem	7
1.0 Introduction:	7
1.1 E-commerce	8
1.2 Social Media	9
1.3 Summary	10
1.4 Problem Statement:	10
Chapter 2: The Solution	12
2.1 Development of a Shopping Centre App	12
2.2 Why develop an App?	12
2.3 Creating consumer trust	13
2.4 Review of Shopping Apps and Websites	13
2.5 Business Case	20
2.6 Costs and Benefits to Stakeholders	21
2.7 Project Risks	22
2.8 Ethical Considerations	23
Chapter 3: Design and Implementation	25
3.1 Summary	25
3.2 Why Create a Native Android Application?	25
3.3 Functional Requirements	25
3.4 Non-Functional Requirements	26
3.5 Android key design principles	28
3.6 Implementation	31
3.7 Android Studio	31
3.8 Architecture	33
3.9 Why utilise SQLite Database?	33
3.10 SQL Scripts	33
3.13 SQLiteOpenHelper class	40
3.14 Testing Data Storage	41
3.15 Black Box Testing	41
3.16 White Box Testing	42
3.17 Emulator and Testing on an Android Device	42
Chapter 4: Conclusion and Future Plans	44
4.1 Changes made to the application	44
4.2 Challenges	44

4.3 The future for GoHere.....	45
Bibliography.....	46

I declare that this is all my own work and that any material I have referred to has been accurately referenced. I have read the University's policy on plagiarism and understand the definition of plagiarism. If it is shown that material has been plagiarised, or I have otherwise attempted to obtain an unfair advantage for myself or others, I understand that I may face sanctions in accordance with the policies and procedures of the University. A mark of zero may be awarded and the reason for that mark will be recorded on my file.

Chapters

Chapter 1: The Problem

An in depth analysis of the problem faced by many retailers today who struggle to stay in business with the increase in popularity of online shopping.

Chapter 2: The Solution

A description of the solution of an Android application proposed to tackle the problem faced by retailers. An in depth insight into the requirements and an overview of similar pieces of software.

Chapter 3: Design and Implementation

The design of how the application's image and how this application will be used by retailers to store their data.

Chapter 4: Conclusion

Challenges faced in the development of the application and what the future has in store

Abstract:

As the Internet continues to play a huge role in our day to day lives, we are increasingly purchasing goods online instead of visiting local stores. The internet offers a wide range of products and the convenience of browsing through a variety of stores whilst sitting comfortably in our own homes.

The impact this has had on traditional stores has made it difficult for some shopping centres to retain tenants as their profits are rapidly decreasing.

This document outlines a solution to the problem through the development of an Android Application to ensure that customers can be given an easy to use shopping application which allows them to browse the stock available in local stores. This means that they are still offered the convenience of a shopping website and helps ensure that local businesses and stores will remain profitable and open.

Chapter 1: The Problem

1.0 Introduction:

In recent years the decline of the UK High Street has been quite topical both in the media and amongst academia. The reasons behind this have been well documented, with numerous themes emerging, primarily that the number of empty retail units has increased annually to around 46,000 in 2013 (Musso 2014.) In 2013, the Department of Communities and Local Government stated that the costs of renting and maintenance of a store has made it increasingly difficult for retailers to manage and remain competitive offline. E-commerce will be analysed further later in this chapter in order to explore why it is attractive for consumers to purchase goods online instead of keeping their money in local city centres.

There is no single reason why consumers are not shopping on their High Streets as they did previously and academics and researchers have put forward many varied arguments. Hall (2013) suggests that the blame lies in “the economic downturn and the continued growth in the number of out-of-town shopping centres, supermarkets and retail parks, which are attracting customers away from town centres.”(Musso, 2013)

It is important to state that other European Countries are not facing the same problems regarding retail as the UK. The EMEA stated in 2013 that most European customers continue to prefer the convenience of shopping locally especially when purchasing clothes or groceries. (Musso, 2013)



(figure 1(a) problem diagram)

1.1 E-commerce

According to Bowen (2014), 'firms today have realised the benefit of e-commerce and use it in a variety of activities.' The shopping experience for consumers is different in the online environment however as they are not able to physically touch, see or try on the end product. They may experience trust issues particularly when the vendor is a small unknown company and customers run an additional risk of receiving a product of poor quality. It is difficult for e-retailers to build intentional trust with their customers and unless the company is well established with a large consumer base already, they may not obtain the online results that they envisage when they create their own website.

However, e-commerce 'represents an expanding and significant opportunity for the e-retailers' e-shopping sites' (Lim et al. 2014). The internet has provided the customer with the means of searching for particular products, comparing prices across a range of different websites and purchasing their desired item online 'using mobile or computer based information searching devices.' (Lim et al. 2014)

Bamfield (2013) argues that as online retail is argued to amount to 21.5% of retail sales by 2018, retailers will gain some success through the development of their own website which will increase footfall as 96% of customers consult the Internet prior to making purchases (Lim et al. 2014.) The power of the Internet in retail is further cemented with the statistic that 30% of UK shoppers use their mobile device to search for a retailers' closest store (Lim et al. 2013).

1.2 Social Media

Due to the increased popularity of social media, retailers are creating and customizing their social media accounts 'and facilitating the creation of virtual communities to reach, interact and collaborate with current and potential e-shoppers' (Lim et al, 2014.) These retailers use social media as a tool for communicating with customers.

A typical example of a store embracing social media to connect with customers is the e-retailer Teavana which provides links for its shoppers on many different social media avenues. Its customer can 'join conversation with Facebook, follow the e-retailer with Twitter, pin with Pinterest, learn and explore with YouTube and join the circle with Google plus' (Lim, 2014) This shows the lengths retailers are going to connect with the public and the costs they incurring to have people on their payroll to communicate on social media. This is in the hope that they are making their company very convenient for the consumer to access.

E-commerce sales increased 14% in 2014 and now according to Kelkoo they have reached over £50bn (Teuber, 2012.) However, social media is not the clear-cut answer to pushing customers to spend their money. Just because a company looks successful on social media and receiving many 'likes' or 'retweets' doesn't mean they are getting any financial gain. These social media users may not be actually going out and purchasing the products. A social strategy needs to lead to increased desire for products and then drive sales.'(Teuber, 2012)

Many shopping centres do offer social media sites for their customers to access information about upcoming events, special in store promotions and to announce new store openings. According to the research above, this is not enough and they need to do more. A recent survey by Ad Age found remarkably that just '17% of marketing executives believe generating sales leads and pushing a social commerce strategy with Facebook advertising is important.'(Teuber, 2012) It is also very time

consuming to build a network of customers over social media as each customer first has to 'like' the page.

Trust plays a pivotal role in any retail transaction. US research by JC Penny and GAP which promoted use of the Facebook 'buy' button was suspended shortly after studies began due to a lack of trust as 75% of Facebook users do not believe it is a secure enough platform to make purchases on (Teuber, 2012). What social media does offer is the ability to find out what the consumer wants, through listening to customers, retailers can amend their 'offerings and products in order to meet their customers' desire which will mean they're more likely to sell'(Teuber, 2012.)

Studies by YouGov (Williams, 2014(b)) found that out of 1096 UK consumers, when asked how they felt about retailers with a social media site, around 25% said it made them trust the seller, with half of all participants claiming social media presence made them trust the retailer less and 42% stating that social media sites such as Facebook and Twitter are for socialising with friends, not for shopping. Its Research Director, James McCoy (Ready, 2014) explains that "the rise of social networks ... coincided with the growth of online shopping and many retailers have tried to harness the power of sites such as Facebook to increase sales." However, consumers prefer to separate their shopping and social lives.

1.3 Summary

As technology changes, retailers need to adapt their strategies and marketing to ensure they remain connected with their customers. As explained above, many retailers have taken to social media and/or developed websites to move with the times. However, it is unclear if this is helping their offline stores show any profit. The problem is maintaining customer loyalty to stores in the UK High Streets especially local stores, and how through either convenience or choice, customers in the UK more than in any other European city are preferring to surf for their products online.

1.4 Problem Statement:

The High Street was significantly and adversely affected by the recent economic recession. Many retailers are trying to hold on to customers through the

development of websites whilst keeping their traditional offline stores open. It is difficult to say if this has given them the trading results that they were aiming for. Similarly, retailers are opening social media accounts as a way of connecting with consumers when they are on sites like Facebook and Twitter. But locally, they may still be struggling to sell items in town centres. This project aims to find a technical solution that could be adopted locally to maximise business potential for retailers and to keep money in the local area.

Although this will provide a service for the big retailing companies, it should also give local businesses the same prospects.

Chapter 2: The Solution

2.1 Development of a Shopping Centre App

In order to alleviate some of the difficulties faced by many retailers I decided to create and implement an Android application.

2.2 Why develop an App?

Over 60% of customers use their mobile device to help them shopping in store (Williams, 2013(b)). The growth of open software platforms and the decrease of the costs of mobile hardware has made it easier for apps to be developed (Marshall, 2013) our increased use of technology has additionally 'impacted consumer buying behaviour in how we shop, where we shop and when we shop' (Marshall, 2013.) As consumer interest in technology grows it becomes imperative to reach out to consumers differently to how they were approached in the past. Many retailers attempt to combine both the offline and online connection with their customers, this is known as "multi-channelling" an example of this is giving customers to ability to buy online and call into the store to collect their order. This decreases the amount of time the customer has to wait to receive their parcel, and may therefore prove to be more convenient for that customer, it also brings the consumer into the physical offline store in the hope that the atmospherics present will entice them to purchase more than just the package they are collecting.

Studies by Google last year found that UK consumers use their smartphones for making purchases far more than in any other European country. Its studies show that 32% of people living in the UK buy something on their smartphones every month 'compared to just 8 per cent in France, 15% in Germany and 19% in Sweden.'(Williams, 2013(a)) In the same research '83 per cent of 1,000 UK consumer surveyed said they use the internet daily, compared to 75 per cent last year' (Williams, 2013(a).) Google's Director of Performance Solutions and Innovation, Ian Carrington stated: "people are increasingly using their smartphones for researching products, locating stores and making purchases. That's why it is now essential for businesses to ensure their sites are mobile-ready." (Williams, 2014(a)) The table below in Fig 2.0 displays the growth of smartphone sales and how they will continue to steadily rise predicting that in 2017 there will be 2.5billion sales globally. An IDC study in 2013 found that Android was dominating 81% (Kerr, 2013) of the

smartphone market. A similar study a year later (2014) found that Android iOS had 96.4% of smartphone sales with Windows phones decreasing by almost 10% (Mayton, 2014).

Smartphone Users and Penetration Worldwide, 2012-2017

	2012	2013	2014	2015	2016	2017
Smartphone users (billions)	1.13	1.43	1.75	2.03	2.28	2.50
—% change	68.4%	27.1%	22.5%	15.9%	12.3%	9.7%
—% of mobile phone users	27.6%	33.0%	38.5%	42.6%	46.1%	48.8%
—% of population	16.0%	20.2%	24.4%	28.0%	31.2%	33.8%

Note: individuals of any age who own at least one smartphone and use the smartphone(s) at least once per month
Source: eMarketer, Dec 2013

166980 www.eMarketer.com

(figure 2(a) (eMarketer 2014) An Analysis of Growth of Smartphone Users)

2.3 Creating consumer trust

No payment or personal details given by user in the app

Security as no purchase made through app

Efficiency through constant update of stock levels

Consistency in user experience to provide consumer loyalty and increasing chance of them using the application again.

2.4 Review of Shopping Apps and Websites

As there have been more shopping websites and shopping centre social media sites available for comparison reasons I have decided to review a broad range of these in order to ensure that the shopping centre app will be functional, on the same level as other consumer based products and as efficient as possible. The app being proposed for the purpose of the specified problem has some unique points that cannot be found in any of the shopping apps that are already available. These qualities include; the option of searching for stock in a local shopping centre instead of an online warehouse (therefore minimising the time it takes a customer to locate the product - given that the item is in stock at the time of their query.)

The applications researched below have their pros and cons which are identified. Although they do provide good shopping assistance, they do not give the user the opportunity to browse through the stock in a local High Street store and do not lead to increased footfall in the stores based on what stock they have.

The shopping centre (Rushmere and Trinity Leeds) display a list of stores available on location and directions on how to find these stores but offer more information on centre events and parking facilities.

The app proposed **to conquer** this solution focuses more on showing the user what is available to them in each store and how convenient it is for them to first locate the item, then visit the store, purchase it and use it straight away.

ASOS Website and shopping App

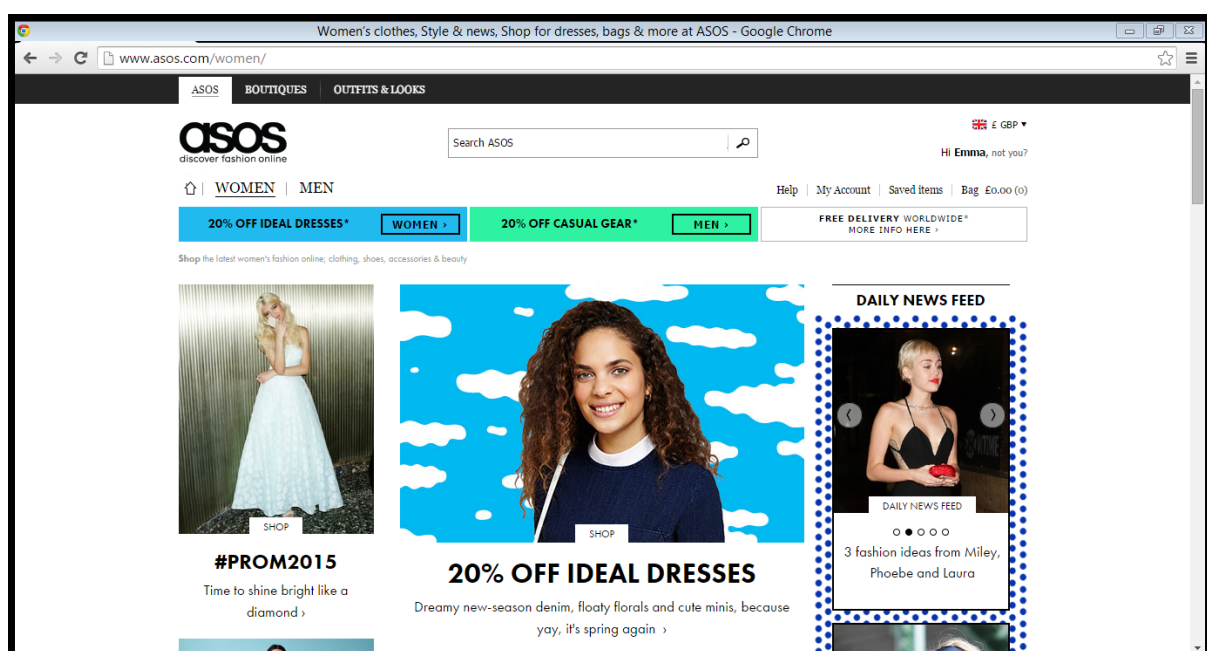
Displays a wide range of brands and clothing categories

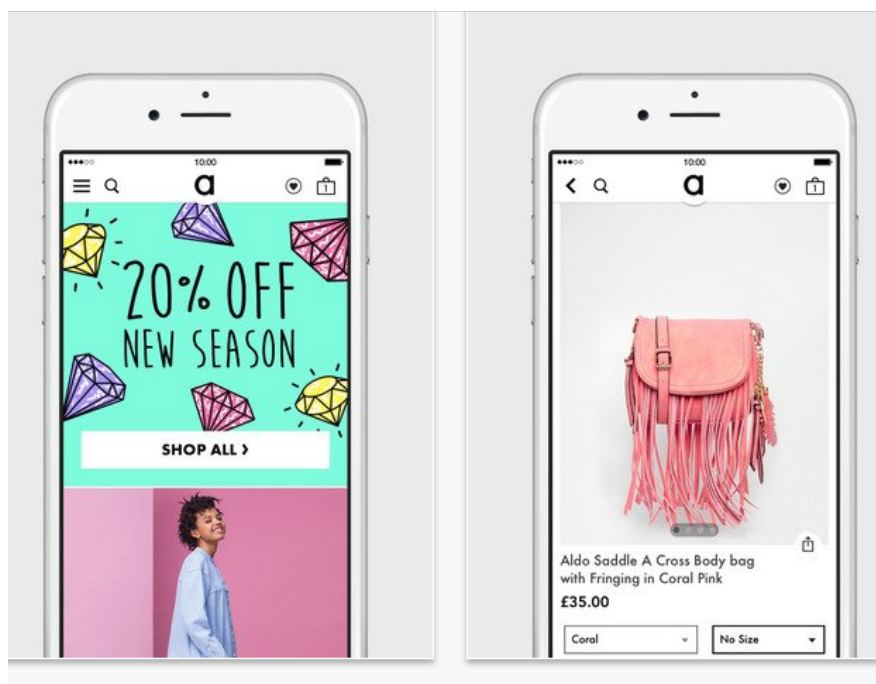
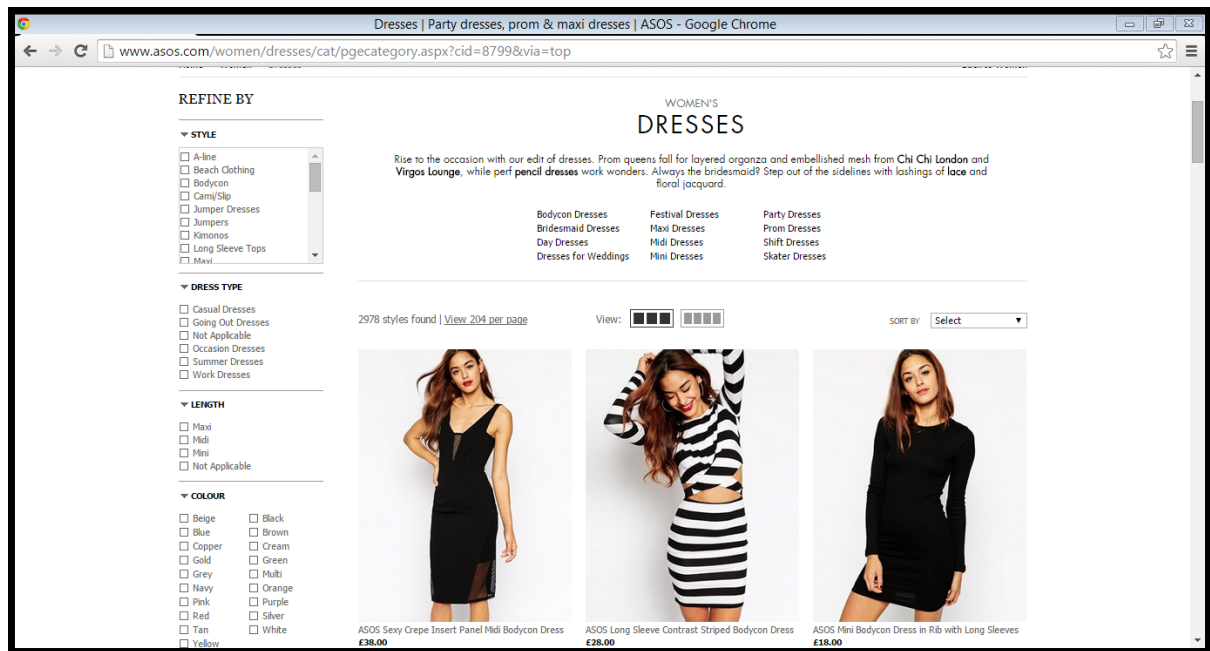
User can browse through site or enter a search

Products displayed with description, brand details and price

Offers displayed on main page

Pros	Cons
<ul style="list-style-type: none"> - Retailers can sell their products on ASOS and/or their own website - Customers can locate products from multiple stores on the same website - Customers have option of paying with PayPal increasing their trust in the security of the site 	<ul style="list-style-type: none"> - Customers must wait for delivery to receive their products - Retailers must pay a fee to advertise their products on the site





Asap 54

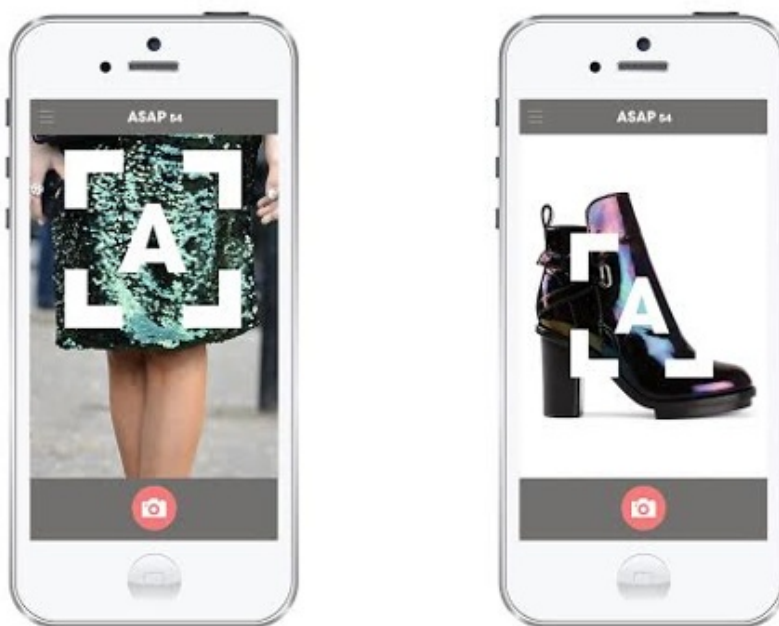
“The Shazam of shopping” (Sedghi, 2014)

Users can take a photograph of clothes on a person when walking down the street and the App produces similar items until the correct product is found using visual recognition technology

Links to relevant store's website

No payment option available on the App

Pros	Cons
<ul style="list-style-type: none"> - Customers can easily access information about the item and locate where it was purchase - user is directed to appropriate retailer's website 	<ul style="list-style-type: none"> - It may be more difficult for it to establish if an item is from an older season of stock - No payment option, app acts as a middleman service between store and customer



Rushmere App

The first App of its kind in Northern Ireland

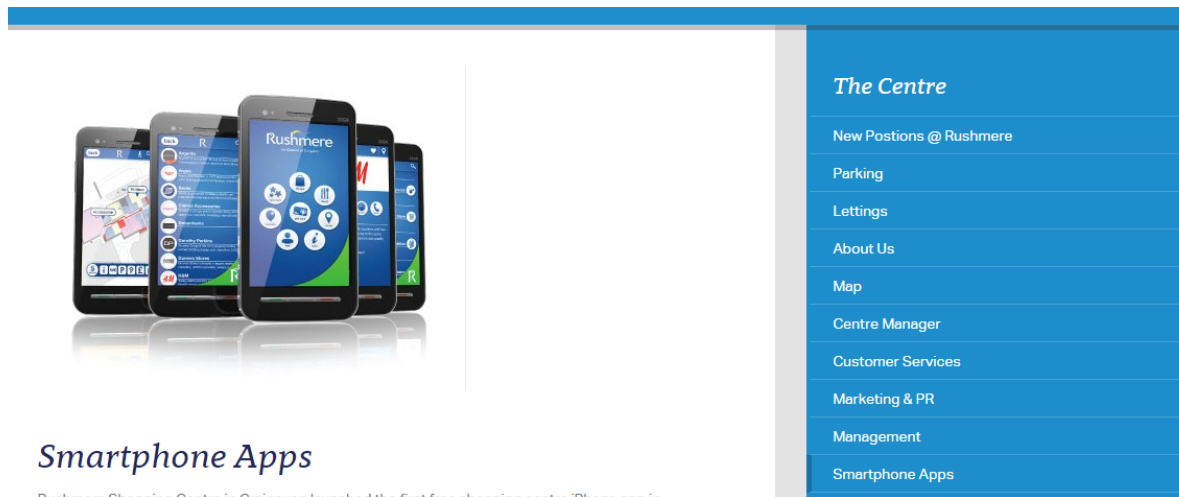
Featuring details about the centre's location and parking facilities

Store details are available for users

Products are not displayed

Pros	Cons
<ul style="list-style-type: none"> - Promotion of shopping centre and publicity for stores that are located in the centre 	<ul style="list-style-type: none"> - App is less personal to the actual stores. They only advertise their location in the centre - Products are not displayed

- gives customers details on parking facilities making it easier for them to locate the car park when they shop
- details about centre location and parking facilities may help tourists who do not know area very well



MetroCentre App

User can make use of interactive map available in the MetroCentre App to get around the centre

Special offers and discounts are displayed

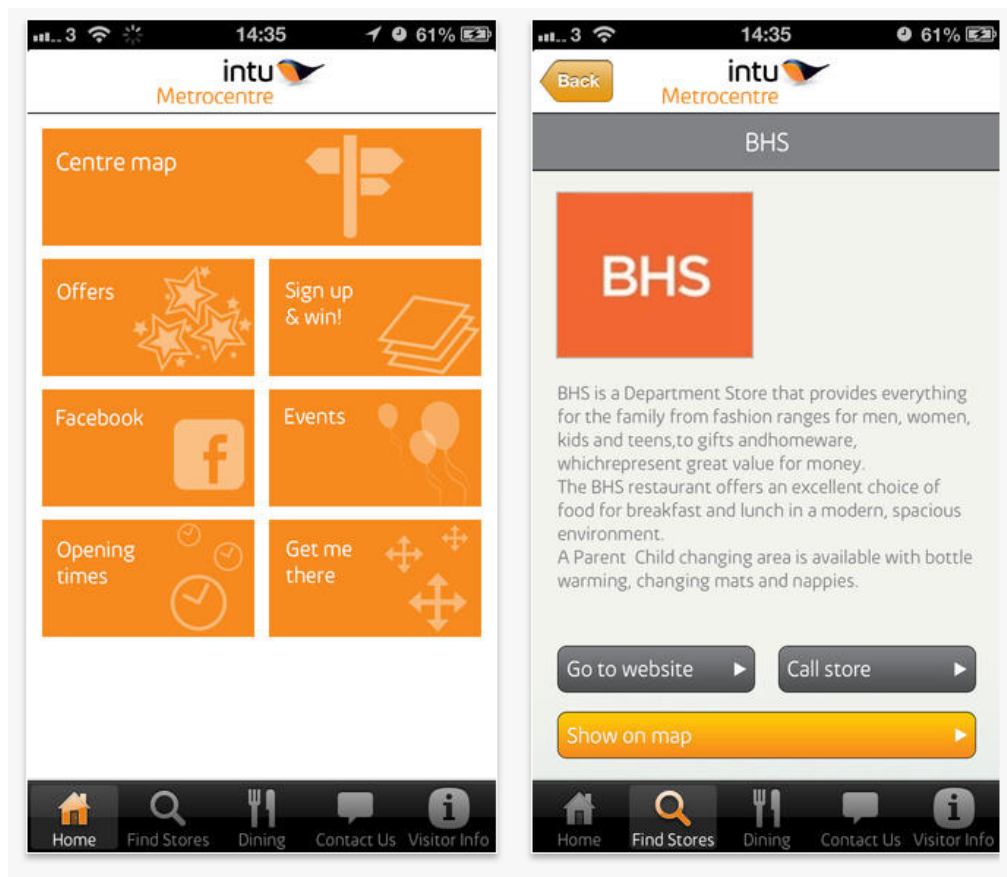
Upcoming events are promoted

User can save their parking spot and find their way back to their car at the end of a shopping trip

Cinema listings and dining options are also displayed

Pros	Cons
<ul style="list-style-type: none"> - Shopping centre details are displayed including location and dining facilities - Interactive map for customer to input car park space and find it later 	<ul style="list-style-type: none"> - Retailers do not have the option of displaying products

- Interactive map to show customer how to get around the centre or find a store
- Centre events are advertised
- Special offers displayed



Trinity Leeds App

Store Map and option to browse through stores available in the Trinity Leeds shopping centre

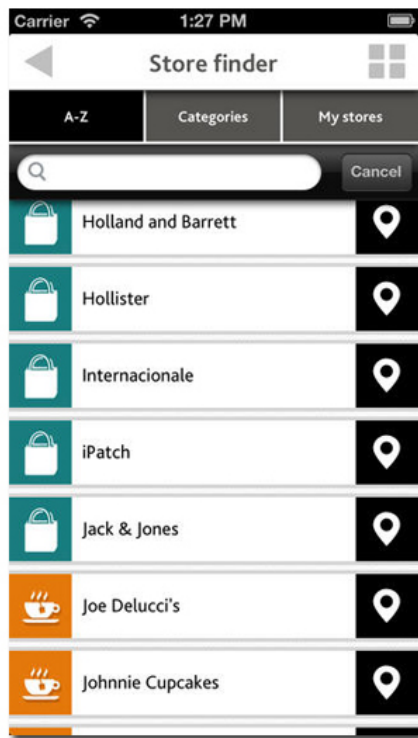
Link to customer service team

Interactive map that directs user to store they want to find

User can sign up to access information about special offers in centre

Pros	Cons
- Shopping centre advertises which stores are available	- Stores do not advertise products

- Users can browse through the available stores
- Interactive map
- Users can sign up to mailing list to receive information about promotions



Name	Cost	Functionality	Products Displayed	Payment Option	Features
ASOS	Free	√√	√√√	√√√	Search Option
Rushmere	Free	√			
MetroCentre	Free	√√			Saves car park information

ASAP 54	Free	√√	√√		Visual Recognition Technology
Trinity Leeds	Free	√√			Interactive Map

2.5 Business Case

The design and implementation of a multi functioning application aimed at consumers to allow them to search for particular items on a search engine unique to a particular shopping centre. They can then locate items quicker, making their experience more convenient.

Stores input their own products into the database

Information about events in the shopping centre or special offers can be displayed to the users

A second interface will be used by the retailers to make it easier for them to input their product information

Although a prototype will be used for presentation purposes the application will be made so that it can be applied to any shopping centre when they decide to adopt it

Payments will not be processed through the application in order to avoid the trust issues consumers may or may not have with using a shopping application.

“Go Here” will initially be an Android application making it available to most smartphones and tablets. Later, it will be available on any smartphone or tablet.

2.6 Costs and Benefits to Stakeholders

Stake Holder	Benefits	Costs
Consumer	Convenience in location of products Quicker shopping experience Save time waiting on deliveries from products ordered online No cost for customer	Time using application
Retailers	Easier to reach customers Marketing of products to customers and stock availability Control of how their information is displayed in the app Increase in sales and	Small fee to advertise products on App Small fee if their special offers will appear on first screen when user accesses the retailer's shopping centre
Shopping Centres	Promotion of shopping centre Promotion of events within centre Less retail units empty; more rent being received No initial cost to shopping centre	Small fee to advertise events happening in the centre

The key stake holders identified are the consumers, retailers and shopping centres. The retailers and centres are struggling to get the customers to notice them when they have so many other purchasing options online. This app would help them to give the customer a new dimension to both shopping offline and shopping using their mobile device. It will then create increased footfall in the centres and increased profit for the retailers.

This will result in a decrease in the amount of empty retail units in the centre and therefore will be to the shopping centre's advantage also.

The customer's shopping experience will be more convenient and they will be able to go straight to the shops to purchase their products without having to buy the item online.

2.7 Project Risks

Risk	P	L	Threat	Contingency Plans
Not being able to incorporated new software correctly to make app interactive	3	2	6	Start working early to develop skills needed for the work
Losing work	2	5	10	Ensure work is always being backed up
Software problem program not running	3	4	12	Ensure that problems in software are expected and that there are always back up plans that can be exercised and that there is adequate time to provide for these problems in the project management phase of the project.

Goal is to make minimalist piece of software primarily that functions for June to be expanded into more advanced version by September.

2.8 Ethical Considerations

This project will conduct research under Category Z defined by Ulster University. It will not involve any invasive procedures disclosure of data.

The project will be conducted at all times in accordance with these guidelines

Chapter 3: Design and Implementation

Design and develop an application written in Java using Android Studio.

Design views and layouts so that it can be easily used. Also design app icon.

Write code starting with the creation of the interface and then the layout using XML

Inbuilt database using SQLite in the Android Studio application to save and retrieve data.

3.1 Summary

In this section of the report the reader will find a description of the system requirements, images of the intended layout of the application specifically each screen/view and an overview of how the data will be sent and retrieved from the database.

3.2 Why Create a Native Android Application?

Native apps are developed for use on a specific mobile device with the ability to take advantage of the functionalities of the specific device, including file storage and the camera. Because of platform heterogeneity one size fits all approach is not realistic. Android developers get a lot of support throughout the development, implementation, testing and production cycle.

In the case of 'GoHere' there are a lot of factors to take into consideration; the needs of the user will be different to the needs of the retailers. Their needs must be seen as having equal weight in importance in order to provide the best service possible to both parties.

3.3 Functional Requirements

- (1) The ability to display data stored in the database in the Application

To fulfil the promises made to clients the app must have the ability to access data stored in the database and fill relevant fields in the app's UI with the data. This will require a lot of testing to ensure that the data is available;

- (a) In the correct place on the UI screen
- (b) That photographs showing the product are the correct size and quality
- (c) The prices are with the correct item to avoid any problems for the retailers
- (2) Inputting data to the database

This process needs to be as simple as possible to avoid any errors. If inputting data is proven to be too technical or difficult, retailers may opt out of using the app's services in the future to avoid any problems that they may face through data errors.

3.4 Non-Functional Requirements

- (1) Security- Security is of huge importance both to the reputation of the app and of the retailers advertising their products. It is essential that all avenues are crossed to ensure that this app keep the retailers' passwords hidden to provide keep clients and to ensure that the app develops a good reputation to gain new clients.
- (2) Accessibility and Compatibility - as the app is being made as a native Android app it will be accessible to all Android devices. Upon recognition of its popularity a few months after its deployment it will be made available for other key platforms such as iOS.
- (3) Portability - the key selling point and the main reason why this app has such commerciality is the fact that it can be used on the go. The app therefore must give the user the information they require anytime and anywhere and the database used must always be accessible.

3.5 User Stories

ID	Title	As a...	I want...	Because...	Acceptance Criteria
1	Login or Register	Developer	Separate users with login credentials to those who need to register their account.	Users that have login credentials will not have to be redirected to register a new account	Selection of a button to login or register will permit user to progress to a new screen

1	Login or Register	Retailer	Sign into my account or register if I am a new user	I would like to access my account with sign in data to validate that I am who I say I am	Select button
2	Registration	Developer	To make registration a quick and simple process	I will make this simple in order for the user to then login and start adding products	Enter username, password and confirm password before selecting the 'Register' button
2	Registration	Retailer	To register	To create my account so that I can start to use this product	Enter details and save, then progress to the login screen
3	Login	Developer	To allow retailer with accounts to login	Quick and efficient process with no fuss	Easy login process through checking variables against SQLite database
4	Add/ Delete Product	Developer	To provide a form for the retailer to enter information related to their product. To provide an easy fool-proof way to delete a product	Easy to understand form, retailer can enter all the data they would like to be stored to add a new product. Retailer can also enter the product code of an existing product to delete all related data from the database.	Data is checked to ensure it is in correct format when adding a product and that all fields are populated. Product code is checked against the retailer's database to check for a match
4	Add/ Delete Product	Retailer	To successfully add or delete products	Easy to use format, foolproof one click stop to add a new product. Quick and efficient way to delete products to ensure database replicates the stock I have in store	All fields populated and the correct respective button is selected
5	Add Image	Developer	Access user's camera and files to upload image to database	Quick process of accessing files and storing them for retrieval by users of the application	One field must be populated to store the image name, an image must be selected and then the image is saved
5	Add Image	Retailer	Upload an image to show users my product	Quick and easy access to files and camera functionality so that I can go back and add more photos	Access files, access camera

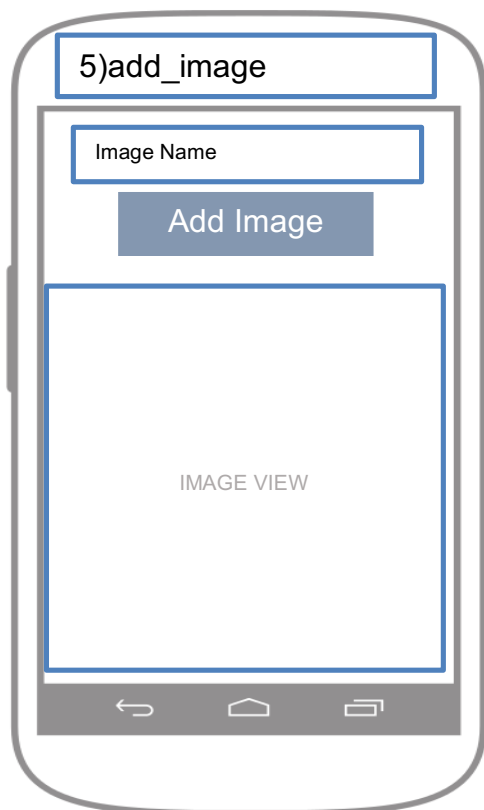
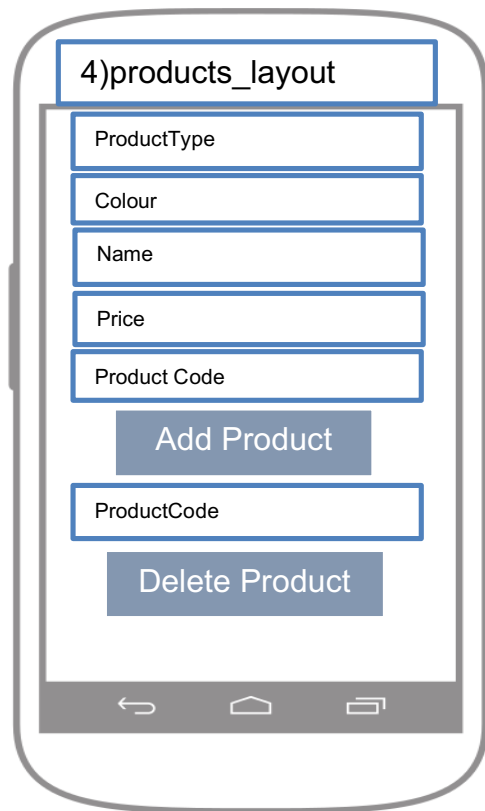
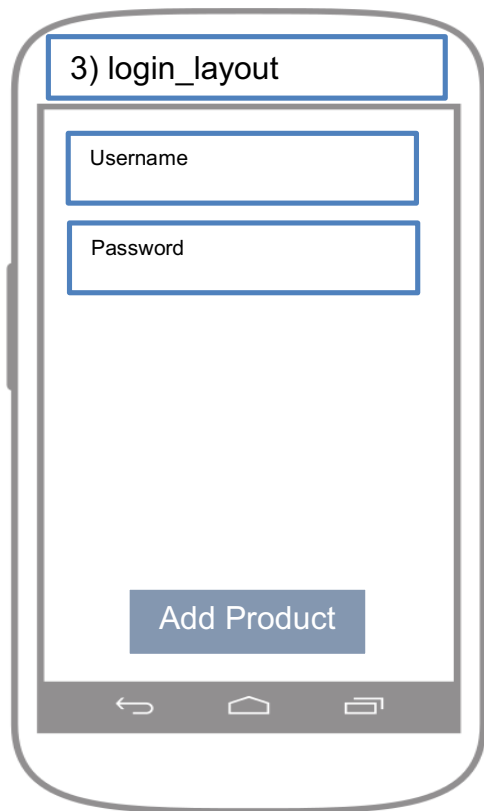
3.5 Android key design principles

When designing for a native platform it is important to take that platform's design principles into consideration. The developer researched Android design principles before preparing the screen designs shown below.

Google prefers Android applications to be developed for the 'worst case scenario' meaning that they are made so that absolutely anybody can use them, regardless of their experience with using applications already on the market and of their levels of technical knowledge. Therefore, it believes that sometimes 'real objects are more fun than buttons and menus.' The developer has tried to include this principle in the design of this application.

The 'only show me what I need to know when I need it' guideline is put into practice later in the app through the search engine which will search through the products of the relevant centre by product type and/or colour to only show the user a selection from the centre's range of products. (This will be amended in future versions to allow user to choose based on a wider range of options such as size and style.)





Screen designs have been created for the User Interface and are displayed above.

Screen 1:

The retailer can select to login if they have saved credentials or register if they do not by selecting the relevant button this will then bring them to screen 2 or screen 3 depending on their selection.

Screen 2:

New users can register here by entering their username and password (then confirming their password in the box below to ensure it is entered correctly) if this username does not exist with the corresponding password the new account is created and the user is brought to the login screen (3) upon successful registration

Screen 3:

The login screen for existing users accessible from both screen 1 and 2. Here the retailer is asked to enter their username and password, the database is then searched to see if these values exist. Upon successful login the user is presented with a message to welcome them and they are brought to screen 4 to enter product details

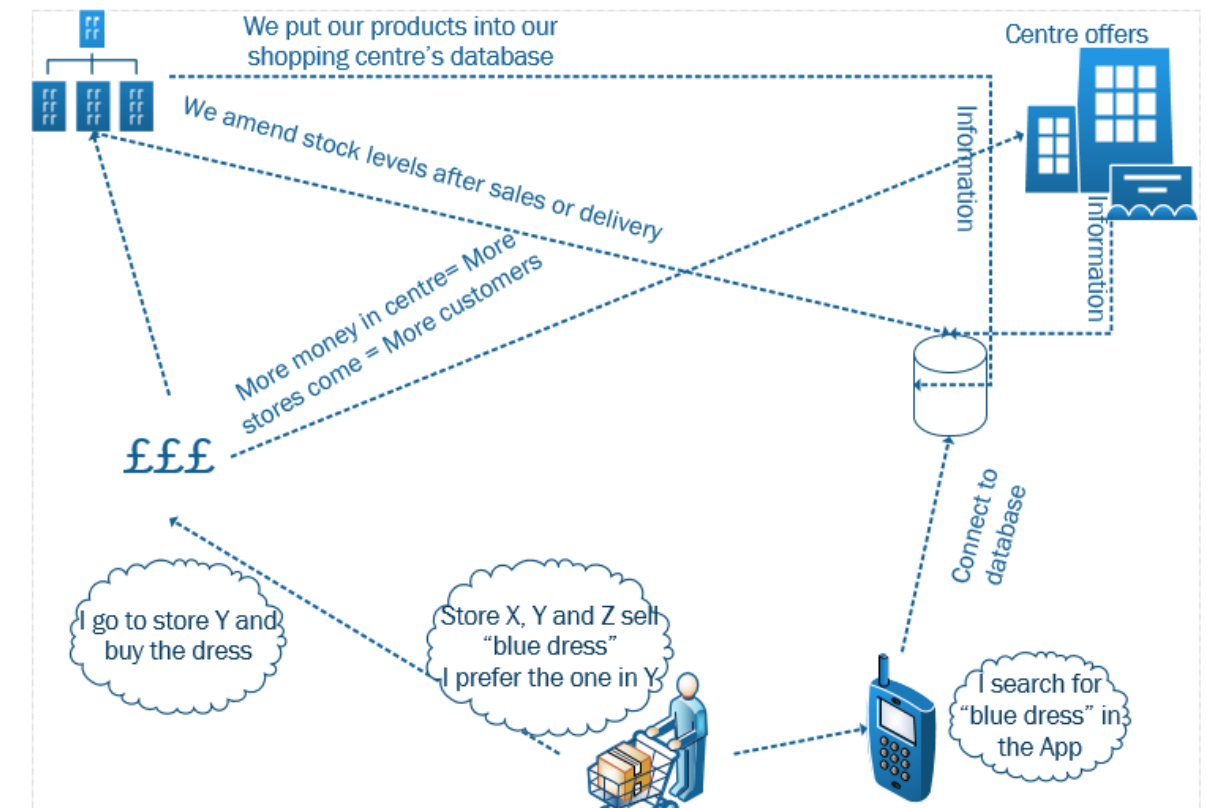
Screen 4:

Where the user enters product details and can also enter a product code to delete a product.

Screen 5:

If a user decides to enter a product they are then brought to Screen 5 to name and upload a corresponding image.

3.6 Implementation



(figure 3(a) implementation)

The android application was developed using XML and Java in Android Studio. The development of 'GoHere' was executed using a combination of the following:
SQLite Database created in Android Studio to with many corresponding Read and Write methods

Separate classes for each XML layout to start the new activity and define the course of action

Implementation of scripts to create the SQL databases in the DatabaseOperations class

3.7 Android Studio

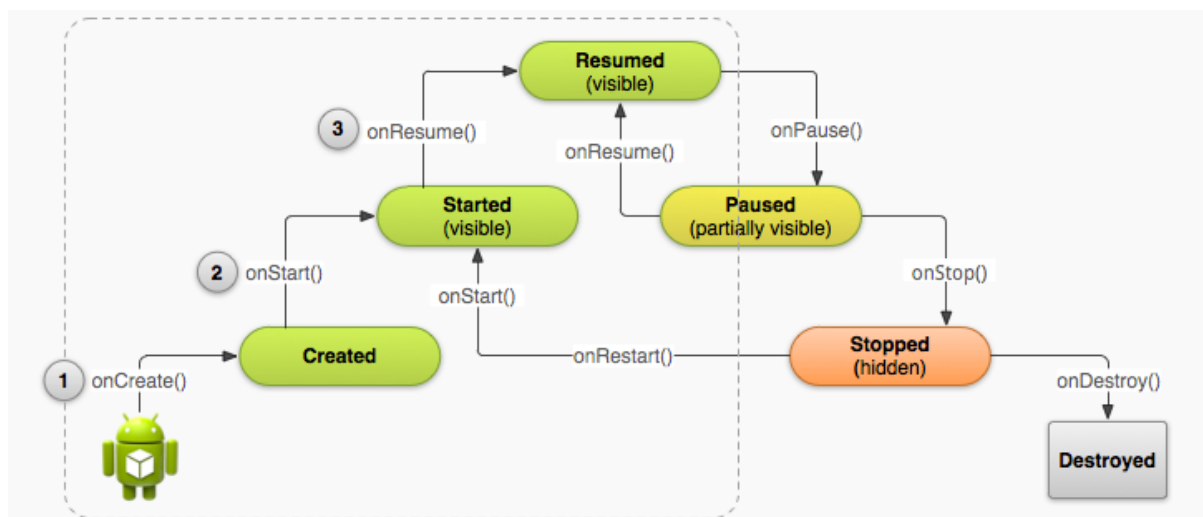
Android provides a software platform and application package through the Software Stack. Developers can write the majority of their program in Java. An Android application is being developed to provide a User Interface for the users and a

separate interface for the retailers. It is important to ensure that both interfaces are simple to use and guide the relevant user through efficiently to the end of the process.

Usually apps are launched through the deployment of a `main()` method, in Android this is not the case. Instead, an Activity instance is started through invoking call-backs. These call-backs are linked to specific stages in the Activity's lifecycle. As the system creates a new activity, the call-back method rotates to another stage in the cycle. The main page in Android is its Activity which is where the app starts running. Every other screens' information runs as an extension to the main Activity each therefore citing 'implements Activity' at the start in its class definition.

The 'GoHere' app is created in Android Studio through the use of Java and the Extensible Mark-up Language (XML.) Each screen declared in Java also has a corresponding XML file which provides a very efficient way for the developer to foresee exactly how the app will look and work physically to the user. This visualisation characteristic allows any programming amendments to be seen in action very quickly and can be very helpful in the process of UI design and debugging.

Although all of the core methods may not be necessary in the 'GoHere' app, to ensure good programming and optimum user experience it is essential to think about these methods for the implementation stage to eradicate any errors. When the methods are implemented correctly, the user is able to resume where they last left off if for example they accept a phone call while using the app. Correct implementation will also ensure resources are not being used by the app when it is no longer in use.



(Figure 3(b) the Android Method Lifecycle (Stack Overflow, 2013)

3.8 Architecture

GoHere's Android Studio package provides a self-containing application with an in-built database accessible through a series of methods which are stored in the 'DatabaseOperations' class and accessible to each XML view's respective Java class.

3.9 Why utilise SQLite Database?

SQLite has many attractive features for Android Developers most of which stem from its self-containing features which make it very easy to configure and debug. Due to its open source and light weight nature it uses just a small amount of disk space and therefore the start-up time and memory consumption are significantly reduced.

It provides a platform for Software Developers to test their application without the reliance of connection to a remote database with the utilisation of an in-process library. This therefore enables quicker emulator performance for testing purposes when implementing a new piece of code.

SQLite databases like MySQL databases are accessed through powerful SQL queries, such as the creation of the SQLite relations.

The application which creates the database is the only application with access as it is built in to the application's package in Android Studio, therefore no parsing is required which reduces the complexity of the code required to read and write to the database.

3.10 SQL Scripts

Scripts are implemented in the 'DatabaseOperations' class to createSQL tables where the Android Application performs many subsequent functions such as Read, Write, Delete and Update in later classes. DatabaseOperations implements the SQLiteOpenHelper activity which enables the class to carry out scripting functions.

Each table is defined in the 'TableData' Java class which depicts the variables necessary to be input by a user for the relevant criteria to be met (to successfully store data in the respective tables.)

The creation of the reg_info table which stores the retailer's login details is displayed below. It is from here that any piece of the retailer's products' data can be reached due to primary key functionality given to the variable C_ID (customer ID.)

For reference purposes the details for 'TableInfo' have also been included from the 'TableData' class.

```
public String CREATE_QUERY= "CREATE TABLE "+ TableData.TableInfo.TABLE_NAME+"(" +
TableData.TableInfo.C_ID+ " INTEGER PRIMARY KEY AUTOINCREMENT, " +
TableData.TableInfo.USER_NAME+ " TEXT," + TableData.TableInfo.USER_PASS+ " TEXT);";
```

```
public static abstract class TableInfoimplements BaseColumns{
public static final String C_ID = "c_id";
public static final String USER_NAME="user_name";
public static final String USER_PASS="user_pass";
public static final String DATABASE_NAME="gohere_info";
public static final String TABLE_NAME="reg_info";

}
```

```
public String PRODUCT_QUERY= "CREATE TABLE "+ TableData.ProductInfo.TABLE_NAME+"("
+TableData.ProductInfo.COMPANY_ID+ " INTEGER," + TableData.ProductInfo.PRODUCT_TYPE+ " TEXT,"
+TableData.ProductInfo.PRODUCT_ID+ " INTEGER PRIMARY KEY AUTOINCREMENT,"
+TableData.ProductInfo.PRODUCT_TYPE+ " TEXT, "
+ TableData.ProductInfo.PRODUCT_COLOUR +" TEXT," +
TableData.ProductInfo.PRODUCT_NAME+ " TEXT,"
+ TableData.ProductInfo.PRODUCT_PRICE+" DOUBLE," +
TableData.ProductInfo.PRODUCT_CODE+" TEXT);";
```

```
public static abstract class ProductInfoimplements BaseColumns{
public static final String COMPANY_ID=TableInfo.C_ID;
public static final String PRODUCT_ID="p_id";
public static final String PRODUCT_TYPE="product_type";
public static final String PRODUCT_COLOUR="product_colour";
public static final String PRODUCT_NAME="product_name";
public static final String PRODUCT_PRICE="product_price";
public static final String PRODUCT_CODE="product_code";
public static final String DATABASE_NAME="gohere_info";
public static final String TABLE_NAME="product_info";
```

```
}
```

```
public String IMAGE_QUERY= "CREATE TABLE "+ TableData.ImageInfo.TABLE_NAME+"("
+TableData.ImageInfo.P_ID+ " INTEGER,"
+TableData.ImageInfo.I_ID+ " INTEGER PRIMARY KEY AUTOINCREMENT,"
+TableData.ImageInfo.IMAGE_NAME+ " TEXT, "
+ TableData.ImageInfo.IMAGE_LOCATION+" TEXT);";
```

```
public static abstract class ImageInfoimplements BaseColumns{
public static final String I_ID = "i_id";
public static final String P_ID="p_id";
public static final String IMAGE_NAME="image_name";
public static final String IMAGE_LOCATION="image_location";
public static final String DATABASE_NAME="gohere_info";
public static final String TABLE_NAME="image_info";
}
```

The display below shows in simple terms how the database is composed, with reference made to the Primary Keys in each table and the variables that are required:

Reg_info

C_ID (PK)	Integer Auto-Increment
USER_NAME	Varchar
USER_PASS	Varchar

Product_info

COMPANY_ID	Integer (FK)
PRODUCT_ID (PK)	Integer Auto-Increment
PRODUCT_TYPE	VARHCAR
PRODUCT_COLOUR	VARCHAR
PRODUCT_NAME	VARCHAR
PRODUCT_PRICE	DOUBLE
PRODUCT_CODE	VARCHAR

<i>Image_info</i>	
<i>P_ID</i>	Integer (FK)
<i>I_ID (PK)</i>	Integer Auto-Increment
<i>IMAGE_NAME</i>	VARCHAR
<i>IMAGE_LOCATION</i>	BLOB

3.11 Java Classes

Not only do the Java Classes define the variables required for the user to progress to the next XML view in the application session, they also provide the implementation functionality and the creation and management of the SQLite Database.

Each class with a corresponding XML view has the onCreate() method in order to define the purpose of each section of the view when it starts and manage how to process any data that is input by the user before progressing to the next View (or screen.)

When all required information is entered by the retailer and the user clicks a button to carry on to the next screen in the app, the onClick method is applied which talks to the 'DatabaseOperations' class to perform some read or write queries on the database.

A good example is the LoginActivity.java class which checks that both the username and password are correct through checking if the username and password match to details stored in the database before:

```

Login.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(getApplicationContext(), "Please wait...", Toast.LENGTH_LONG).show();
        username = USERNAME.getText().toString();
        userpass= USERPASS.getText().toString();
        DatabaseOperations DOP = new DatabaseOperations(cts);
        Cursor CR = DOP.getInformation(DOP);
        CR.moveToFirst();
        booleanloginstatus = false;
    }
}

```

```

        String NAME = "";
        //creation of the cursor object using a do..while loop
        do {

            //0 and 1 below represent and point to the correct column index for each type of data
            if (username.equals(CR.getString(0)) && (userpass.equals(CR.getString(1)))) {
                //login is successful
                loginstatus = true;
                NAME = CR.getString(1);

            }
        } while (CR.moveToNext());
        //IF loginstatus is true
        if (loginstatus) {
            Toast.makeText(getBaseContext(), "Login Success..... \n Welcome " + NAME,
                Toast.LENGTH_LONG).show();

            Intent intent= new Intent(LoginActivity.this, ProductsActivity.class);
            startActivity(intent);
            setContentView(R.layout.products_layout);
            try {
                Thread.sleep(100);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            else {
                Toast.makeText(getBaseContext(), "Login Failed..... \n Try again ", Toast.LENGTH_LONG).show();
            }

        }
    });

```

3.12 Methods within the 'DatabaseOperations' class

In addition to providing the gateway for the creation of the Database, the 'DatabaseOperations' class is where the methods which are implemented on the database are stored; for both Read and Write purposes.

This functionality enables retailers to write their products' details to the database for them to be displayed to the end user when they request to perform a read or 'get' function on the database.

The method used to input or write a new product to the database is displayed below, which defines the necessary variables and the format in which they should be saved to the database:

```
public void putProductInformation (DatabaseOperations dop, int c_id, int p_id, String type, String colour, String name, Double price, String code){
```

```
    SQLiteDatabase SQ = dop.getWritableDatabase();
    ContentValues cv = new ContentValues();
    //cv.put (TableData.ProductInfo.COMPANY_ID,);
    cv.put(String.valueOf(TableData.ProductInfo.COMPANY_ID),c_id);
    cv.put(String.valueOf(TableData.ProductInfo.PRODUCT_ID),p_id);
    cv.put (TableData.ProductInfo.PRODUCT_TYPE, type);
    cv.put(TableData.ProductInfo.PRODUCT_COLOUR, colour);
    cv.put (TableData.ProductInfo.PRODUCT_NAME, name);
    cv.put (TableData.ProductInfo.PRODUCT_PRICE, price);
    cv.put (TableData.ProductInfo.PRODUCT_CODE, code);
    long j= SQ.insert(TableData.ProductInfo.TABLE_NAME, null, cv);
    Log.d("Product operations", "Product inserted");
}
```

The above method 'putProductInformation' is accessed in a separate Java class 'ProductsActivity' when the retailer enters the required information to the product form and hits the 'save' button.

In the onCreate() class displayed below, the values that are entered by the retailer are converted to a String resource from their original 'EditText' before being pushed through to the 'putProductInformation' class on application of the 'onClick' method.

When a product is successfully added, a toast message is displayed to the retailer and they are then brought to a new screen to upload a corresponding image accessible either through their Android phone's files or by use of camera functionality.

```
protected void onCreate (Bundle savedInstanceState){
    super.onCreate(savedInstanceState);
    setContentView(R.layout.products_layout);
    //all variables essential for adding an item

    PRODUCT_TYPE=(EditText)findViewById(R.id.p_type);
    PRODUCT_COLOUR=(EditText)findViewById(R.id.p_colour);
    PRODUCT_NAME=(EditText)findViewById(R.id.p_name);
    PRODUCT_PRICE=(EditText)findViewById(R.id.p_price);
    PRODUCT_CODE=(EditText)findViewById(R.id.p_code);

    //product code to quote when deleting an item
    DEL_CODE=(EditText)findViewById(R.id.p_code_del);

    Add= (Button)findViewById(R.id.AddProduct);
    Add.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            //get data from all edittexts

            type=PRODUCT_TYPE.getText().toString();
            colour=PRODUCT_CODE.getText().toString();
            name=PRODUCT_NAME.getText().toString();
            price=Double.parseDouble(PRODUCT_PRICE.getText().toString());
            code=PRODUCT_CODE.getText().toString();

            DatabaseOperations DB= new DatabaseOperations(ctx);
            c_id++;
            p_id++;
            DB.putProductInformation(DB, c_id, p_id, type, colour, name, price, code);
            Toast.makeText(getBaseContext(),"product added now add a photo", Toast.LENGTH_LONG).show();
            Intent intent= new Intent(ProductsActivity.this, AddImageActivity.class);
            startActivity(intent);
            setContentView(R.layout.add_image);
            //finish();
        }
    });
}
```

```
});
```

Retailers also have an option to delete products once logged in by entering its product code. Again, the user will receive information via a Toast message when the product has been successfully deleted:

```
Delete= (Button)(findViewById(R.id.DelProduct));
Delete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        DatabaseOperations DB= new DatabaseOperations(ctx);
        Cursor CR = DB.getProductCode(DB, code);
        booleanregistered = false;
        code=PRODUCT_CODE.getText().toString();
        //get the value of the columns from the cursor
        CR.moveToFirst();
        do {
            if (registered) {
                DB.deleteProduct(DB, code);
                Toast.makeText(getBaseContext(), "product removed", Toast.LENGTH_LONG).show();
                finish();
            }
        } while (CR.moveToNext());
        Toast.makeText(getBaseContext(), "invalid product code, try again", Toast.LENGTH_LONG).show();
        finish();
    }
});
```

3.13 SQLiteOpenHelper class

SQLiteOpenHelper is a class called from the DatabaseOperations class to manage database creation; opening the database if it does exist and creating the database if it does not (version management.)

The subclasses within this class help developers implement onCreate smoothly with a default template

SQLiteOpenHelper ([Context](#) context,
[String](#) name,
[SQLiteDatabase.CursorFactory](#) factory,
int version)

Context **Creates database or Opens if it is already created**

Name	The String name of the file holding the database
Factory	Creates cursor objects through the SQLiteDatabase.CursorFactory library
Version	Version control which starts at 1 (integer) <pre>public static final int <i>database_version</i> = 1;</pre>

3.14 Testing Data Storage

Android Studio permits the developer to test and debug the backend locally with the Android app that has been implemented. It also very conveniently provides a service for the app to be deployed from Android Studio directly with a live backend to make it accessible for users.

Testing is a very important stage during and post implementation to ensure that the application does what it is supposed to do while taking into consideration the possibility of system error and/or users not doing what the programmer expects. Several tests have been developed to attempt to cover all eventualities: white box testing and black box testing are the two types of testing that are most commonly used. Although Database Management Systems do include the means to test data query, integrity and processing, to ensure a professional application has been developed there are further steps that must be taken before deployment.

3.15 Black Box Testing

Black Box Testing is used to test the functionality of the code this period of testing is not concerned with how the software was made or how the project was built. Its key

purpose is to test if the expected outputs are displayed when the user inputs certain search requests.

This testing was useful to ensure retailers received the correct information at the correct time and that validation throughout the application is working for example when the retailer enters data the application will check that all fields are populated and in the instance of registering a new user this will ensure that both passwords match before inputting their data in the database.

3.16 White Box Testing

An essential element to testing software. This is predominantly exercised by the software developer who knows the project inside out. It is important at this stage that any problems regarding accurate syncing of the database and the application are realised by the tester. The possibility of a database deadlock through being accessed by too many users or retailers at the one time is high and will produce the biggest problems for this application if any problems are not caught at this testing stage.

It is essential that white box testing is carried out using a methodological approach to ensure that all areas are covered and to avoid problems down the line especially to avoid an access deadlock.

This tests the 'internal function of the software [and] uses specific knowledge of the code.' This is an imperative stage in the testing of any piece of software especially when a database is a key component where 'testing is performed often manually' and errors are deliberately set in place to ensure that the program realises there is a problem and can cope properly (Akram et al. 2015.)

3.17 Emulator and Testing on an Android Device

Throughout the implementation process, the application will be tested on a Nexus Galaxy Emulator in the Android Studio environment both periodically and as soon as new significant changes are made to the code.

In addition to this once the database has been linked to the application project, the application will then be tested on a real device. This is essential for creating an app to see the quality of user experience and how efficiently the program is running. This

is an imperative stage 'later in the sprint to validate critical performance ... and usability (Correia et al, 2015).'

Chapter 4: Conclusion and Future Plans

4.1 Changes made to the application

Through the course of the development of GoHere a few of the original plans have been scaled back to ensure optimum performance.

The plan to have an Android application accessible by all shopping centres and retailers is still very much on the forefront however for testing version 1 a study case has been adopted to focus on just one shopping centre focusing on the customer that will be paying for the service.

Originally the developer had created and implemented an Android Application that used a BackgroundTask class to open and close connections to a database stored in phpmyadmin and accessible through methods in the Java code with the assistance of php scripts.

Although this is still an option for the further development of GoHere in the future and in different versions when it becomes live to the market, it was not a feasible way of testing locally.

The start-up time was becoming persistently slow and it became almost impossible to run and therefore debug the application.

The decision to implement a SQLiteDatabase which resides locally in the Android Studio application diminishes the risk of delays due to start up time and greatly improved the period of time taken for the application to load which greatly assisted in the running and debugging of GoHere from a developer's stand point.

This is mainly due to the ability of SQLite to utilise the database concept from within the application package itself.

4.2 Challenges

Throughout the development of the android application there were quite a number of challenges faced and these challenges are reflected in many of the decisions to change the course and implementation of the GoHere application.

When initially creating this application, the aim was to create a MySQL database which would be stored in phpmyadmin and accessible through an XAMPP server. This ran smoothly for the first few attempts, however as the application got bigger it would take longer and longer to connect to the database which became increasingly difficult to debug resulting in poor emulator performance and timeouts.

Further problems arose when using this method as Android Studio itself started crashing, restarting the computer used to code and test the application and therefore the process was becoming very difficult to test.

The decision to implement an SQLiteDatabase was not taken lightly and only after a series of tests were conducted to try to salvage the original plan such as; completely deleting the Android Studio package that was installed, rebuilding from scratch after a new download starting with small steps of ensuring onClick methods were operational before putting in the code and methods.

After many attempts and tests the decision to move and use SQLite seemed like the correct thing to do to provide a version of the application for the deadline.

4.3 The future for GoHere

This prototype can be used internationally when shopping centres decide to sign up to allow its retailers to use the service. When all stores are onboarded users will then be able to browse through the variety of products from the comfort of their own home.

To ensure that security is not an issue for the users the application will not allow users to pay through it. Instead users will be able to have an item saved as when this application is fully deployed it will show users the size range that is currently in stock. This means that they will conveniently and quickly locate the product that they are searching for and know that they will get it in their size in X store in X shopping centre.

GoHere will therefore offer users an experience that is not currently available to them as consumers and the software will enable them to browse as they would if they were searching on a retailer e-commerce application whilst ensuring that they locate the product quickly and efficiently that same day. The customer can then go to the store and make a purchase therefore cutting out the time spent waiting for a delivery.

Bibliography

- Android Studio Developer Tools <http://developer.android.com/tools/studio/index.html><accessed 20/05/15>
- Akram M, Waqas W (2015) Analytical study of black box and white box testing for database applications <accessed 25/05/15>
- Bowen, G. (2014) *Computer Mediated Marketing Strategies Social Media and Online Branding* I-GI Global
- Correia I, Santos A (2015) Mobile testing in Software industry using agile: challenges and opportunity IEEE 8TH Conference on Software Testing, Verification and Validations (April, 2015) <accessed 29/05/15>
- eMarketer (2014) *Smartphone users worldwide will total 1.75 billion in 2014* <http://www.emarketer.com/Article/Smartphone-Users-Worldwide-Will-Total-175-Billion-2014/1010536><accessed 21/03/15>
- Kerr, D. (2013) *Android dominates 81 percent of world smartphone market* <http://www.cnet.com/uk/news/android-dominates-81-percent-of-world-smartphone-market/><accessed 23/03/15>
- Lim, J.S. Al-Alai, A. Heinrichs, H. (2014) *Impact of Satisfaction with E-Retailers Touch Points on Purchase Behaviour: The Moderating Effect of Search and Experience Product Type Marketing Letters*
- Marshall, A., (2013) *Designing telemedicine apps that health commissioners will adopt*, Open Innovations Association (FRUCT), 2013 14th Conference
- Mayton, J. (2014) *Windows phone RIP? Android, iOS account for 96.4 per cent of global smartphone sales* Tech Times <http://www.techtimes.com/articles/13252/20140818/windows-phone-rip-android-ios-account-for-96-4-percent-of-global-smartphone-sales.htm><accessed 20/03/15>
- Musso, F. (2014) *Handbook of research on Retailer-Consumer Relationship Development* I-GI Global
- Ready, B. (2014) *Your mobile is the new high street* The Telegraph <http://www.telegraph.co.uk/technology/news/10668543/Your-mobile-is-the-new-high-street.html><accessed 20/03/15>
- Sedghi, A. (2014) *New app Asap54 is 'Shazam of shopping'* The Guardian <http://www.theguardian.com/technology/2014/mar/04/asap54-iphone-app-shazam-shopping-visual-recognition><accessed 20/03/15>
- Teuber, B. (2012) *The rise of social media in retail* The Guardian <http://www.theguardian.com/media-network/media-network-blog/2012/jul/25/social-retailing><accessed 21/03/15>
- Williams, R. (a) (2014) *UK consumers buy more via smartphone than rest of Europe* The Guardian <http://www.telegraph.co.uk/technology/news/10693221/UK-consumers-buy-more-via-smartphone-than-rest-of-Europe.html><accessed 20/03/15>
- Williams, R. (b) (2014) *Social media presence makes retailers appear less trustworthy* The Guardian <http://www.telegraph.co.uk/technology/news/10668053/Social-media-presence-makes-retailers-appear-less-trustworthy.html><accessed 19/03/15>