



A digital menu application for accessibility in restaurants

Student Name: Angela Peng	Student ID: C15402002	Supervisor Name: Dr. Oisin Creaner
Link to Software Repository:	https://github.com/C15402002/FYP	

Table of Contents

Table of Figures	4
1. Project Statement.....	5
2. Research	6
2.1 Background Research.....	6
2.2 Similar Application	7
2.2.1 <i>OrderWizard</i>	7
2.2.2 <i>eHopper POS</i>	9
2.3 Technologies Researched	11
2.3.1 <i>Android V. IOS</i>	11
2.3.2 <i>Front End Research</i>	13
2.3.3 <i>Back End Research</i>	15
2.3.4 <i>Application Programming Interface and Libraries</i>	16
2.4 Relevant Research.....	17
2.4.1 <i>Customer Relationship Management (CRM)</i>	17
2.4.2 <i>Localizing User Interface</i>	17
2.4.3 <i>PayPal</i>	18
2.4.4 <i>Vision Impairment</i>	18
3. Approach and Methodology	19
3.1 Waterfall Model.....	19
3.1.1 <i>Requirements</i>	19
3.1.2 <i>Design</i>	20
3.1.3 <i>Implementation</i>	20
3.1.4 <i>Verification/Testing</i>	20
3.1.5 <i>Maintenance</i>	21
3.2 Priorities of Accessibility Functions.....	21
3.2.1 <i>Priorities set for Users – Main Task</i>	21
3.2.2 <i>Priorities set for MANAGER – Medium Task</i>	22
3.2.3 <i>Priorities set for FLOOR/KITCHEN STAFF – Low Task</i>	22
4. Design.....	23

4.1 Technical Architecture Diagram.....	23
4.2 Diagrams and iterations	24
4.2.1 <i>The First Iteration</i>	24
4.2.2 <i>The Second Iteration</i>	27
4.2.3 <i>The Third Iteration</i>	29
5. Prototyping and Development	32
5.1 Front End Development	32
5.1.1 <i>Restaurant Application User Activity Prototype</i>	32
5.1.2 <i>Point of Sale Management Activity Prototype</i>	35
5.2 Back End Development.....	39
6. Testing	40
6.2 Unit Testing.....	41
6.3 Integration Testing.....	41
6.4 System Testing	41
6.5 Acceptance Testing.....	42
7. Issues and Risks	42
7.1 Data Storage	42
7.2 User Interface assistance	42
7.3 Third Party APIs	43
8. Plan and Future Work	44
Bibliography	46

Table of Figures

FIGURE 2-1 ORDER WIZARD QR SCANNER (6).....	7
FIGURE 2-2 ORDER WIZARD MENU ITEM (6).....	7
FIGURE 2-3 ORDER WIZARD EDIT ORDER ITEM (6)	8
FIGURE 2-4 ORDER WIZARD PAYPAL (6)	8
FIGURE 2-5 EHOPPER SALES ANALYTICS (7).....	9
FIGURE 2-6 EHOPPER EMPLOYEE MANAGEMENT (7).....	10
FIGURE 2-7 EHOPPER PRODUCT MANAGEMENT (7)	10
FIGURE 3-1 WATERFALL MODEL (17).....	19
FIGURE 4-1 MODEL-VIEW-CONTROLLER (18).....	23
FIGURE 4-2 USE CASE OF FIRST ITERATION	25
FIGURE 4-3 SEQUENCE DIAGRAM OF FIRST ITERATION	26
FIGURE 4-4 USE CASE OF SECOND ITERATION	28
FIGURE 4-5 USE CASE OF THIRD ITERATION	30
FIGURE 4-6 SEQUENCE DIAGRAM OF THIRD ITERATION	31
FIGURE 5-1 APPLICATION COLOUR SCHEME.....	32
FIGURE 5-2 COLOURBLINDGO COLOUR BLINDNESS ASSISTANCE.....	33
FIGURE 5-3 APPLICATION LOGIN PAGE PROTOTYPE.....	33
FIGURE 5-4 APPLICATION REGISTER PAGE PROTOTYPE	34
FIGURE 5-5 MY PROTOTYPE QR SCANNER	34
FIGURE 5-6 POS COLOUR SCHEME	35
FIGURE 5-7 POS STAFF LOGIN PAGE.....	35
FIGURE 5-8 POS MANAGEMENT PAGE.....	36
FIGURE 5-9 ENTITY RELATIONSHIP DIAGRAM	39
FIGURE 0-1 PROJECT PLAN GANTT CHART.....	44

1. Project Statement

This project is for the creation of an accessible restaurant ordering application for mobile devices. The method of communication will be in two parts one for the user which will be built using android system and the management system which will be controlled by the restaurant manager that will be built by a web application.

The customer user interface which will be designed to assist users, will contain the functions to create an account to capture customer details, add, edit, delete order, series of function and abilities to perform certain operations.

The second part will be the user interface used by the restaurant manager will contain the ability to perform the CRUD functions for menu items, analysis of sales, management of restaurant staff and overall management of the restaurant.

In this report a detailed background research, project approach, design, prototyping, testing, issues and risks, and the future plan are elucidated.

2. Research

2.1 Background Research

Quick Response codes also known as QR codes are becoming extremely popular in the technology industry. QR codes is defined by the oxford dictionary (1) as a machine-readable code consisting of an array of black and white squares, typically used to store URLs or other information for reading by the camera of a smartphone.

Quick Response codes were first introduced by Denso Wave during 1994 (2). It was used to enhance decoding speed and the ability to hold large amount of information. These codes are growing rapidly in the marketing and ecommerce community. According to an article by OpenJaw (3) the rise of QR code usage is targeted in Asian countries especially China. Unlike any other country China has 900 million active consumers a month using a popular worldwide application called WeChat (4). This application contains many functions one in particular is its use of QR codes for payment transactions, marketing and user identification. A downside to this application would be its functionality outside of China. This app will only operate at its full potential if the user has a Chinese credit card and a Chinese phone number otherwise it is just another social media application.

For a business, it is important to understand its market and acknowledge the connection with its consumers. Flexibly is the key to a successful business, by implementing technology into the food and drink industry we are able to establish this connection. An application that captures user detail and functions subjected to flexibility like diary free, gluten free, vegetarian or vegan. We can build a trusted foundation between the producer and consumer.

Point of Sale (POS) systems (5) contains tools that are essential to businesses that handles sales reporting, customer management, Inventory management and employee management. POS can provide store manager with information that increases business efficiency.

2.2 Similar Application

2.2.1 OrderWizard

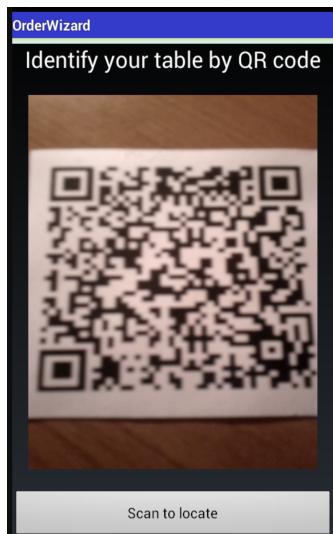


Figure 2-1 Order Wizard QR scanner (6)

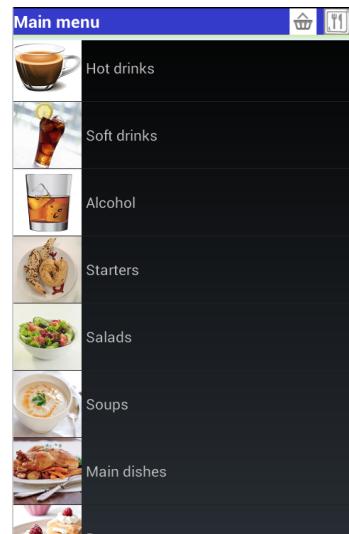


Figure 2-2 Order Wizard Menu Item (6)

DT282 B.Sc. (Hons.) in Computer Science International
Interim Progress Report 2018/19

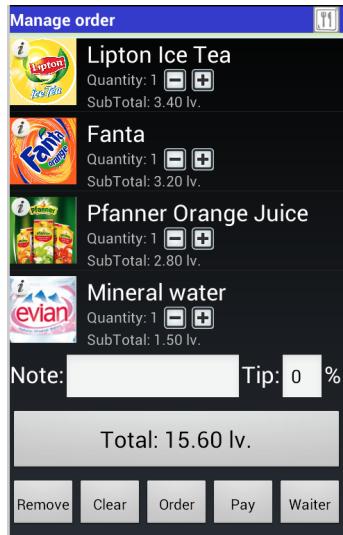


Figure 2-3 Order Wizard Edit Order Item (6)

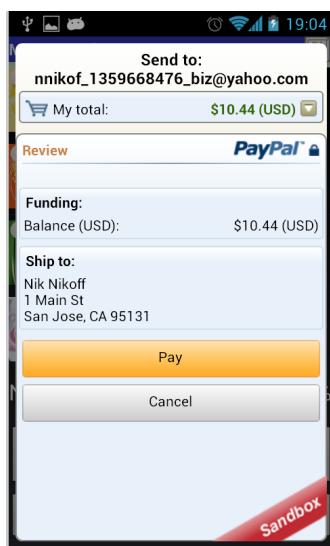


Figure 2-4 Order Wizard PayPal (6)

OrderWizard (6) has similar functions to this project including QR code scanning. This application supports both android and IOS system. The system will ask the user to scan a QR code to retrieve a restaurant, then depending on the restaurant a menu will display. From the menu, the user is able to add items to order, when they are finished they can then edit, delete or add extra details. Payments are compensated by PayPal, offline cash or offline card transaction.

A benefit that this application takes is users can call the waiter if they want to process the transaction with cash.

A disadvantage is that any extra details that the user adds is not by item but by the overall order as seen in fig. 2.3. This app also does not ask for customer details this may cost a loss in clientele and security breach. User authentication provides secure transactions and personal information without this it becomes a vulnerability. The system will become exposed to cybercriminals from gaining personal information.

This application is available on google play but there haven't been any updates since 2013 when the application was first created.

2.2.2 eHopper POS

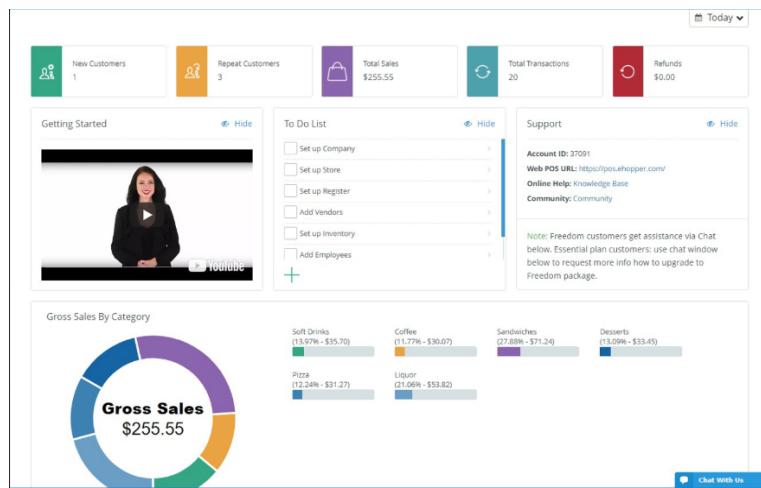
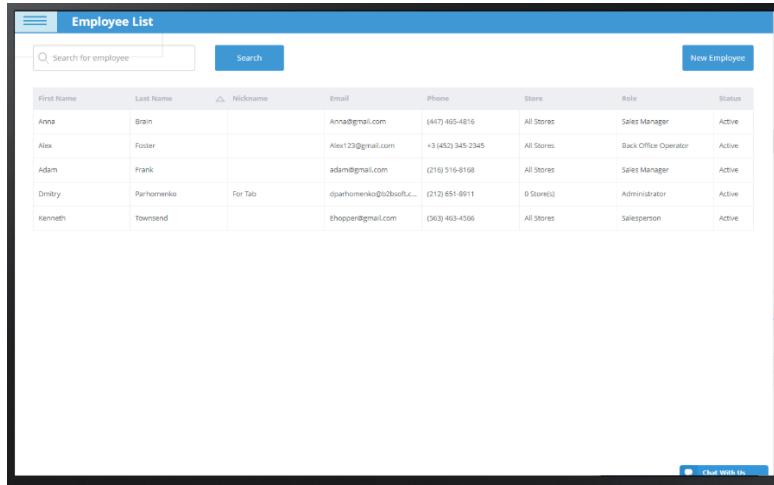


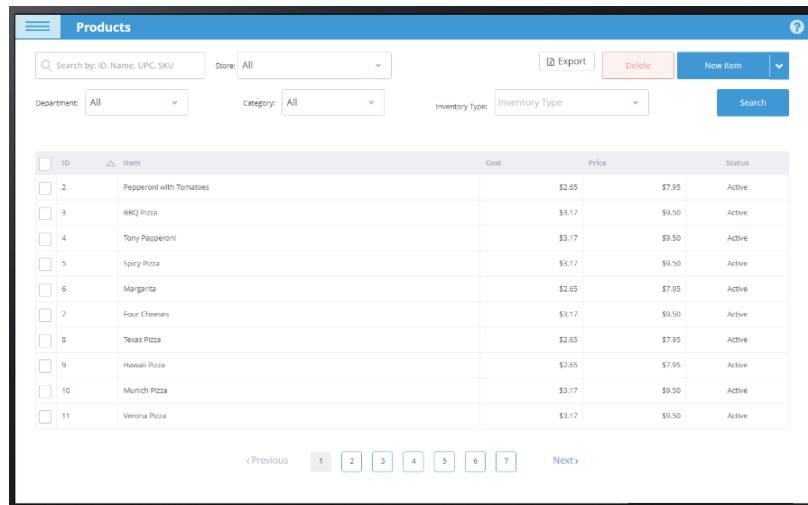
Figure 2-5 eHopper Sales Analytics (7)

DT282 B.Sc. (Hons.) in Computer Science International
Interim Progress Report 2018/19



First Name	Last Name	Nickname	Email	Phone	Store	Role	Status
Anna	Brain		Anna@gmail.com	(447) 405-4816	All Stores	Sales Manager	Active
Alex	Foster		Alex123@gmail.com	+31(40)2 345-2345	All Stores	Back Office Operator	Active
Adam	Frank		adamt@gmail.com	(210) 516-8168	All Stores	Sales Manager	Active
Dmitry	Parhomenko	For Tab	dparhomenko@z2bizsoft.c...	(212) 651-8911	0 Store(s)	Administrator	Active
Kenneth	Townsend		thepper@gmail.com	(563) 403-4566	All Stores	Salesperson	Active

Figure 2-6 eHopper Employee Management (7)



ID	Item	Cost	Price	Status
2	Pepperoni with Tomatoes	\$2.65	\$7.95	Active
3	BBQ Pizza	\$3.17	\$9.50	Active
4	Tony Pepperoni	\$3.17	\$9.50	Active
5	Spicy Pizza	\$3.17	\$9.50	Active
6	Margarita	\$2.65	\$7.95	Active
7	Four Cheeses	\$3.17	\$9.50	Active
8	Texas Pizza	\$2.65	\$7.95	Active
9	Hawaii Pizza	\$2.65	\$7.95	Active
10	Munich Pizza	\$3.17	\$9.50	Active
11	Verona Pizza	\$3.17	\$9.50	Active

Figure 2-7 eHopper Product Management (7)

eHopper (7) is a point of sale management system. This program is available in IOS, android and windows operating systems. According to their webpage and documentation this system possesses many operations from Multi-store management, POS reports, employee management, order tracking to tax management and ingredient management.

An advantage of this POS is it provides the business with an extensive amount of functions to manage their performance. The system supports many devices and it provides a self-explanatory user interface.

A disadvantage of eHopper is that for a business to access its full potential, expenditure of the system increases.

2.3 Technologies Researched

2.3.1 Android V. IOS

Both platforms are supported on many devices and SDKs. While android and IOS are at a constant competition which one another, there are pros and cons that needs to be considered when developing an accessible application. Below discuss the differences between Android and IOS in modern applications.

2.3.1.1 Android Platform

Market share that android set out is for global usage eminently for areas of lower income. About 75% of the population (8) worldwide use an android platform device, as android is available to many different manufacturers even with different SDK versions consumers can experience various hardware devices but still remain as a loyal customer to android. This is also known as Device Fragmentation (9). While developing applications developers must consider limited devices that the application can support otherwise it will cause complexity.

Advantages of Android Platform:

- Various of devices available

- Can be developed on different machines
- Written in Java very popular programming language
- Flexible

Disadvantages of Android platform:

- Takes a lot of time for development which then costs a lot more money
- Lots of device fragmentation

2.1.1.2 IOS Platform

Unlike android, IOS tends to cater consumer that are willing to spend money on applications.

Disadvantage developing IOS:

- Applications can only be developed on mac
- Written in Swift language which has constant updates, developers find it difficult to manage changes
- Compatibility issues with Objective-C

Advantages developing IOS:

- Little device fragmentation
- Cost less and little time in development
- Less configuration

Having analysed the difference between both android (java) and IOS (swift), I decided to implement this project using Android studios as I have acquired prior experience with this platform. Undeniably, android is still evidently more popular and cheaper than developing an IOS application.

2.3.2 Front End Research

Developing a front-end user interface, I considered the use of native, web and hybrid application. Below discusses its pros and cons of utilizing these three types of front-end development.

2.3.2.1 Native application

As this project is concerning the development of an android application, android API's native contains many tools that can be directly accessed through the IDE. For android studio, it be dependent on XML files for activities. (10)

Advantages of using native application:

- Fast and responsive. As the application is native to its platform, data is fetched from a web server rather the application
- Internet access is not an essential requirement, though some application do expect to use it
- Specific UI design for an overall better user experience
- More interactive experience because they are running on the device built-in features
- Runs much smoother for user input and output

Disadvantage of using native application:

- Need an extensive amount of knowledge in the programming language during development and maintenance. Native apps need constant software updates to fix any bugs
- Needs vast amount of time to implement with the large amount of work that is expected to develop a native application the cost sequentially increases

2.3.2.2 Web Application

Web application hosted by web browsers are condensed information from the website to produce a more functional application. A wide range of technologies can be used to implement a web application this includes, open source platforms like PHP or MYSQL. (11)

Advantages of using web application:

- Easy to maintain without user having to constantly update the application
- Accessible to all platforms through an internet connection which allows cross-platform application service
- Security is a priority in web applications. Unlike other simple websites, web applications are utilized on the machines dedicated server

Disadvantage of using web application:

- Needs internet access at all time, without it users cannot experience the application to its full potential
- Much slower than native apps
- Less interactive for client side.

2.3.2.3 Hybrid Application

Hybrid applications is a combination of a native app and a web app. Built using JavaScript, HTML, CSS wrapper in a native container like, web view. (12)

Advantages of using hybrid application:

- Easy to develop and time for development is significantly faster than native apps.
- Allows access to different device platforms to be integrated this is cost efficient

- Consistent maintenance can be done without having the user update the application except for any major changes

Disadvantages of hybrid application:

- Much slower than native apps
- Less interactive, navigation to different activity could be lagged
- Cannot experience access to native features, this can be a downfall for UI design and programmatic operations

2.3.3 Back End Research

2.3.3.1 Firebase

Cloud hosted services which can be used by application development as a backend server. Firebase offers many abilities to perform client-server interaction. (13)

Benefits:

- Offers storing and synchronising real time data to any connected devices of all platform.
- Offers the use of google analytics
- Easy to implement with little back end development
- Can implement many ways to authenticate users including email and password, Facebook, google account authentication. Making developing apps identifying users efficiently
- Allows notifications to be sent to targeted users using the notification console GUI. This is a free service that firebase provides.

Limitations:

- Usage limitations, firebase free version only allows up to 50 connections
- Doesn't work like ordinary relational DBMS. Therefore, normal SQL queries are no use in firebase

2.3.3.2 SQLite

SQLite is an open source database service and it can be accessed using android studio IDE. No installation is required for this database service because it is an in-process relational DBMS.

Benefits:

- Lightweight structured file using query language, SQL to retrieve data
- Doesn't rely on server which means no configuration is required and can be portable
- Communicates directly with the application for this instance Android studio IDE

Limitations:

- Can only handle low to medium traffic which is not suitable for large data storing application
- Scalability limitations

2.3.4 Application Programming Interface and Libraries

2.3.4.1 Google Play services

With the introduction of google play service 7.8, Mobile vision API (14) released a new API which enables barcode scanner to read and decode its contents.

Google barcode API is able to detect barcodes in real time on devices in any orientation. Implementing this with permission to use Camera API, an android framework which supports various camera features. Allows the application to access the device camera and scanning any barcode it detects.

2.3.4.2 Zxing Library

Also known as “Zebra Crossing” (15) is a google operated open source service which is used to read and decode barcodes using an image processing library.

2.4 Relevant Research

2.4.1 Customer Relationship Management (CRM)

Mobile customer relationship management (CRM) (16) is extremely important in today's business industry. With the increase in the use of technology and internet services businesses ensures customers are receiving the best service at all time. Capturing client details allows employees to provide real-time data and information to clients immediately.

Benefits of customer relationship management is its ability to deliver performance reports, analytics in sales and organizational activities in businesses.

2.4.2 Localizing User Interface

Advancement of technology made it possible to communicate with individuals of different nationality to interact with each other at ease. For instance, if a user that has many allergies is travelling and ordering food in a foreign restaurant becomes a difficulty an application that supports different languages can benefit these individuals. Therefore, the language barrier is not a problem when customers go to a non-English spoken country.

Android studio (17) is able to localize the user interface to support different languages. By utilising translation editor in the IDE, it will allow the application

to change from one language to another and not effecting the systems default language.

2.4.3 PayPal

The most secure way of electronic transactions is through PayPal (18). PayPal released BrainTree (19) in 2014 making it effortless to develop payment services on Android, IOS and web platforms. BrainTree accepts payments using credit card, PayPal, Google Pay and Apple Pay. By using HTTP based RESTFUL APIs, we can retrieve real-time data to authenticate credentials. Responses and requests are structured in JSON.

2.4.4 Vision Impairment

In an article by Invision App (20) it talks about how “Making your app accessible to visually impaired users benefits **all** your users”. There is an estimate of 285 million people globally who are vision impaired. Developing an application that allows flexibility and change in user interface can ease any conflict and intimidation for individuals with vision impairment.

Colour schemes, amount of information displayed and navigation should all be considered when developing.

3. Approach and Methodology

3.1 Waterfall Model

The waterfall model methodology (21) is a simple structured organizational method. The advantage of using this method is its linear structure making this approach understandable and manageable. By following each step, I was able to envision a distinct final product and goal. Below describes a thorough explanation of each steps taken into utilizing this structure in the project.

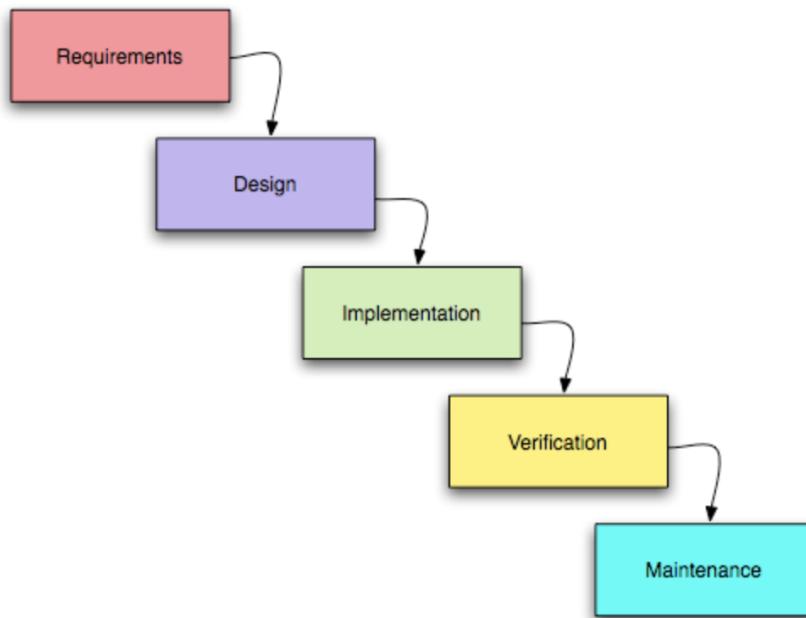


Figure 3-1 Waterfall Model (21)

3.1.1 Requirements

This phase gathers information of what the overall system should be able portrayal. Information is written down and documented in detail that includes any research and methodology. This is a crucial part of project planning as it is the foundation of the whole product. Without this phase carefully theorized it will cause a lot of complications throughout the whole process of development.

During my requirement stage I investigated on similar applications of this project and the use of QR codes in modern technology.

3.1.2 Design

During design phase focuses on technical architecture design, programming language used and diagrams to assist in implementation. In this project, I constructed MVC, use case, sequence and entity relationship diagrams which gives an insight to the project objectives.

3.1.3 Implementation

The coding stage of the project takes the above two phases into action. Creating and implementing the requirements which includes both front-end and back-end development. For this project, I will be implementing a user friendly interactive application that asks users for inputs which will be stored in a back-end server. QR code IDs that contains menu information are also stored. I hope to implement feasible accessibility operations.

3.1.4 Verification/Testing

This phase is for acceptance testing and system testing to resolve any issues that emerges. Certainly, testing is performed continuously during the early stages of implementation. Following later stages, testing should be carefully conducted on unbiased terms and well documented. In section 6. Testing, a detailed description of each stage of testing is discussed.

3.1.5 Maintenance

Last phase of the project, application is ready to be formalized to the market. System refurbishment and debugging are still maintained and kept under control by system updates. Hybrid application updates only concerns major changes. Other modification can be done with internet access and won't cause a major effect on usability and performance.

Administrator is always monitoring the system looking for bugs that arises and user feedback on particular functions that needs readjustments. User feedback and testimonials are a huge part of maintenance, helping the developers to enhance details in their system for more efficiency

3.2 Priorities of Accessibility Functions

I have divided different tasks that needs to be implemented into three parts from Main, Medium to Low. Depending on what is expected of the final product setting out priorities from high to low will help me to manage time and functions importance. Implementing particular features in Users activity to this project will define its uniqueness between other similar applications.

3.2.1 Priorities set for Users – Main Task

HIGH Priorities for USERS include:

- Ability to scan QR code with the device's camera
- User authentication
- Ability to use PayPal/Credit card payments
- Ability to resize font size for clear text visual

MEDIUM Priorities for USERS include:

- Ability to change language on application

DIT

**DT282 B.Sc. (Hons.) in Computer Science International
Interim Progress Report 2018/19**

- Ability to call for waiter
- Filter menu items according to user dietary information

LOW Priorities for USERS include:

- Ability to adjust colour for users with vision impairment
- Text-to-Speech properties

3.2.2 Priorities set for MANAGER – Medium Task

High Priorities for MANAGER include:

- Ability to add, delete, edit menu items
- Ability to manage staff, employee rota, employee details

MEDIUM Priorities for MANAGER include:

- Ability to manage products, delivery, stock
- Ability to see performance report

LOW Priorities for MANAGER include:

- Ability manage customer relationship management

3.2.3 Priorities set for FLOOR/KITCHEN STAFF – Low Task

High Priorities for STAFF include:

- Ability to see customers' orders
- Staff authentication

MEDIUM Priorities for STAFF include:

- Ability to retrieve real time data from customers

LOW Priorities for STAFF include:

- Ability to send notifications to customers

4. Design

4.1 Technical Architecture Diagram

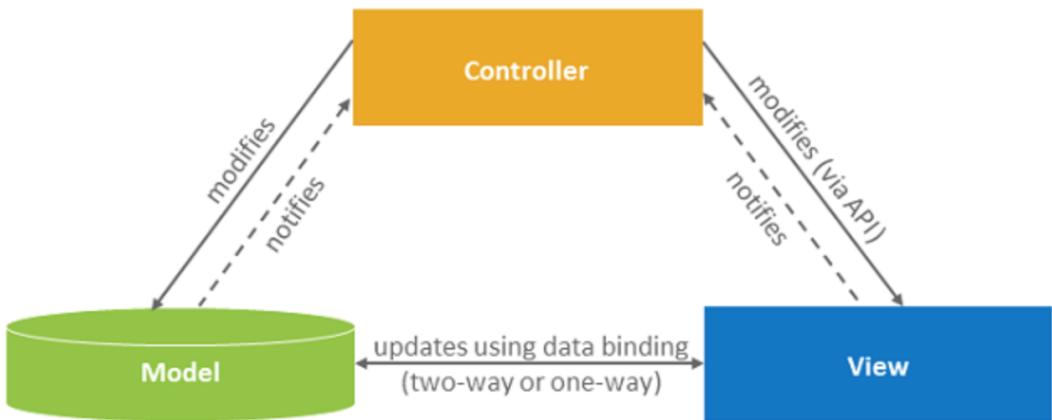


Figure 4-1 Model-View-Controller (22)

The model view controller architecture (MVC) (22) assisted me to design an interactive system that can be flexible to change. MVC consists of three parts, the first would be the model, this contains the data of the application any changes made to the model correspondingly changes the view to show the user of a new update. The second part is the view, this is the user interface displayed to the user. The last part is the controller, this controls the user input and according to the inputs the view and/or model might changes.

From this architecture, I am able to design my system which will consist of the customer (view), the manager (view), database(model). The controller will update any data that has been changed by both views to model, once the update is done the views will automatically update. With the three environments working together it creates a smooth application.

4.2 Diagrams and iterations

This section comprises of use case diagrams, sequence diagrams and their corresponding iterations. I think that this is a simple way to understand each actors' roles and what the system entails in a sequential order.

4.2.1 The First Iteration

The first iteration includes actions customers are able to perform when using this application. The process of customer upon registering and making an order will involve at least 2 actors, the customer itself and the restaurant staff.

4.2.1.1 Use Case Diagram

Main Flow

- Customer Creates Account
- Customer allows application to access device camera to scan QR code
- Customer can view menu
- Customer makes order
- Customer can view status of their order
- Waiter can view order status
- Manager can view menu

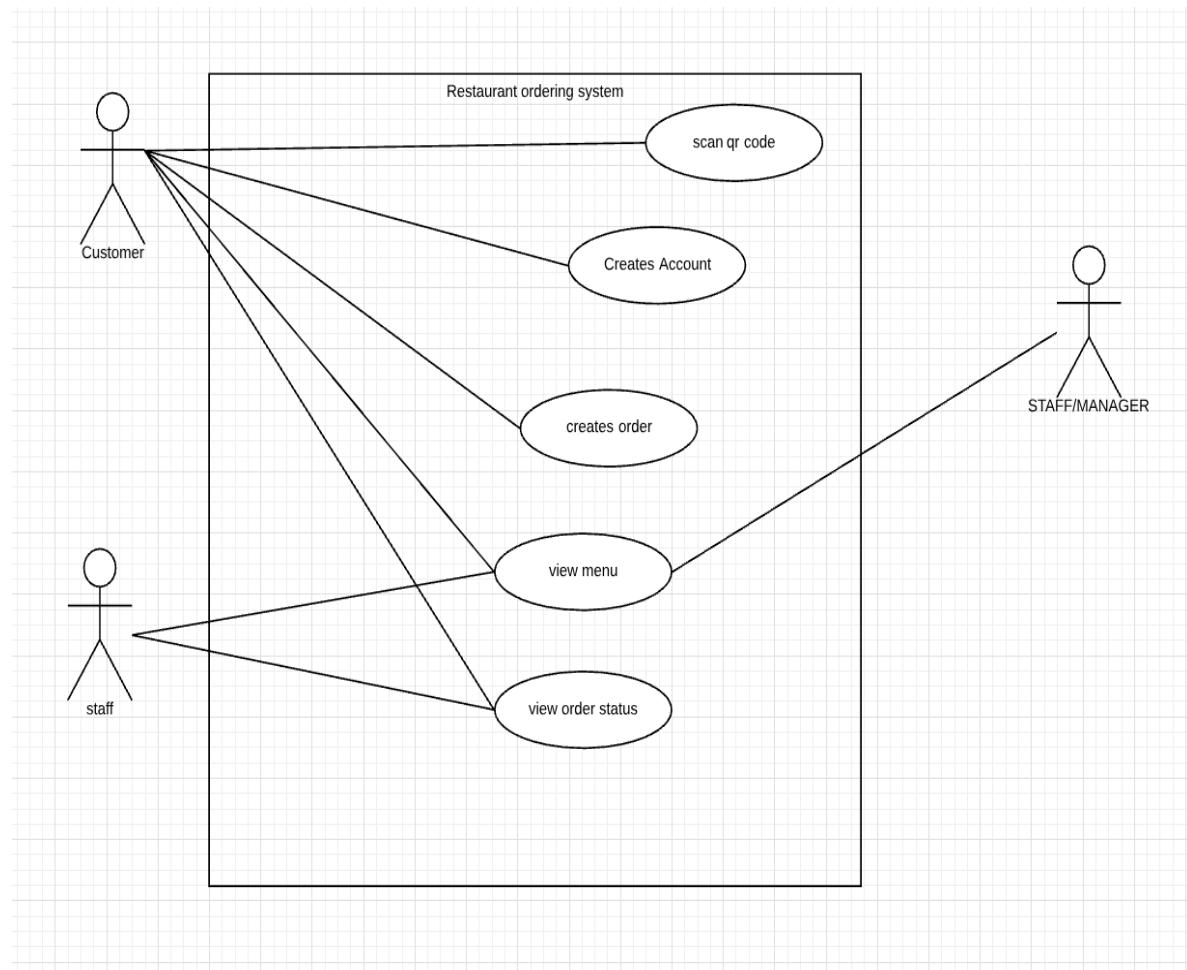


Figure 4-2 Use Case of First Iteration

Description: From the first iteration, we can see that the Customer is able to scan QR, create account, create order, view menu and view order status. The staff actor is consisting of waiters and chefs.

Pre-condition: we know nothing about the user.

Post-conditions: we know information about the user, we know their allergies and we know if they have a vision impairment.

Exception Errors: The user may only use card transaction when using this application.

Alternative variations: Users are validated by unique email and a corresponding password.

4.2.1.2 Sequence Diagram

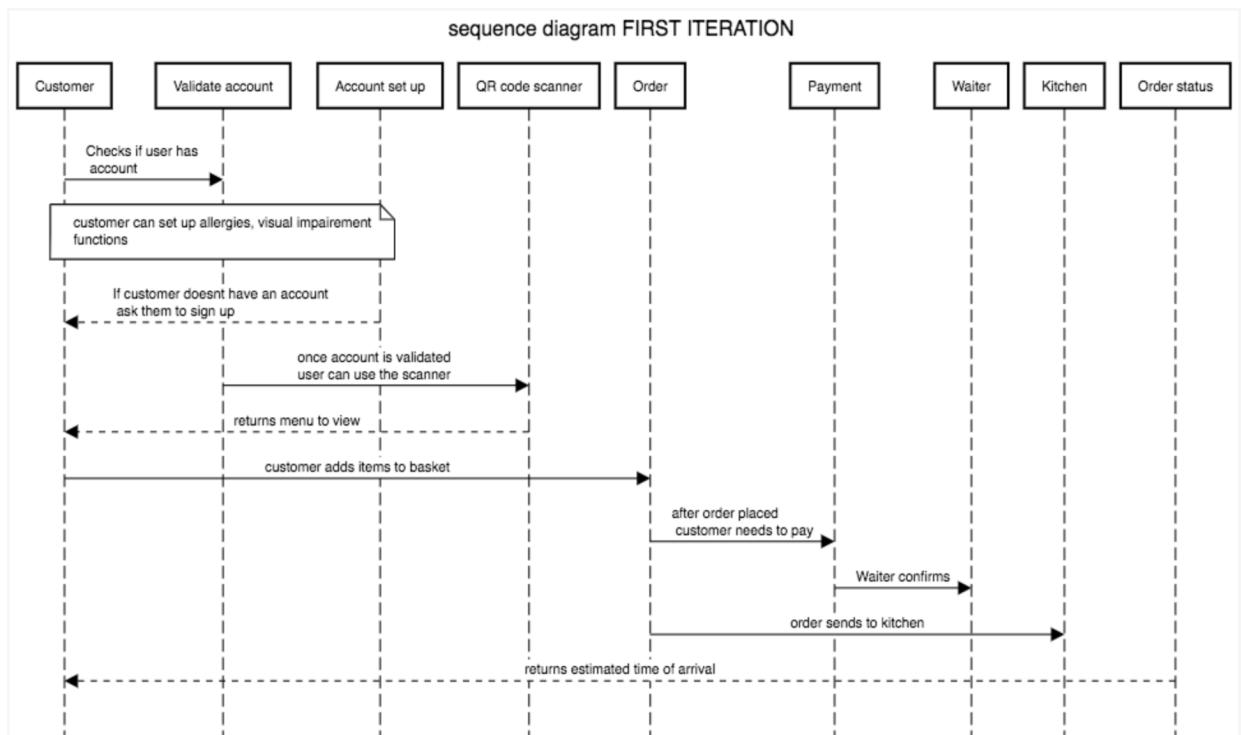


Figure 4-3 Sequence Diagram of First Iteration

This sequence diagram shows the sequence of actions performed by the customer when using this system. The user upon using the application must log into an already existing account or create a new account. The user inputs will be validated based on firebase authentication on email and password. Once the user is able to access their account they are able to use the QR code scanner. They must first allow the application access to their mobile device camera to perform

the camera function. After scanning the QR code, the activity page will show the restaurant menu from here the user can add items on the menu into their basket to add to the order. Once the user is finished with their order they must pay the total amount that is owed before the order goes to the kitchen. The application will ask the user permission to enter card details. Subsequently, the transaction is successful order will be sent to the kitchen staff and waiter. The user from here will be able to see the estimated time of arrival to their table.

4.2.2 The Second Iteration

Second iteration is the actions performed by waiter/kitchen staff actor. They have limited access to information about the restaurant itself and customers. Another actor that is involved in the second iteration is the administrator. Administrators has unlimited access, they would know a certain amount of knowledge regarding how the system works and are able to perform maintenance and updates to the software when needed.

4.2.2.1 Use Case Diagram

Main Flow:

- Staff depending on their role a waiter can view transaction and order
- Kitchen staff can view order only
- Administrator performs system maintenance

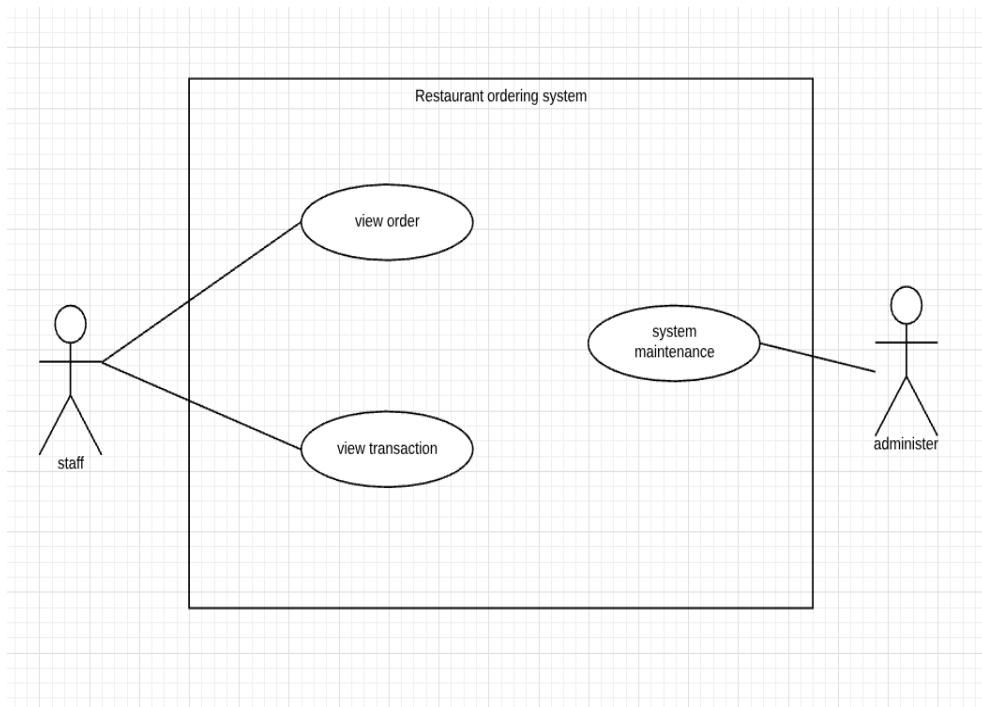


Figure 4-4 Use Case of Second Iteration

Description: The second iteration shows the staff and administrator. The staff can see the order placed by customers and transaction progress. Administrators maintains the system and keeps the system updated.

Pre-condition: Each staff are given roles by manager, we know information about each staff. We know information about the administrator. Staff doesn't know customer orders.

Post-condition: Waiters are able to see order after customer has placed one.

Exception Error: The chef can only see the order if the transaction process is completed.

Alternative Variables: Waiter and chef must log into the staff account, role and position is saved.

4.2.3 The Third Iteration

The third iterations involve the manager of the restaurant only. They also have unlimited access to information about everything in the business. They are able to perform the CRUD operations to menu items, manage employees, manages stocks and review analytic reports. The manager runs the restaurant therefore, have the ability to control the environment.

4.2.3.1 Use Case Diagram

Main Flow:

- Managers can add new items to the menu
- Managers can delete items from the menu
- Managers can edit items on the menu
- Managers can view sales analytics/reports
- Managers manages stock
- Managers manages staff

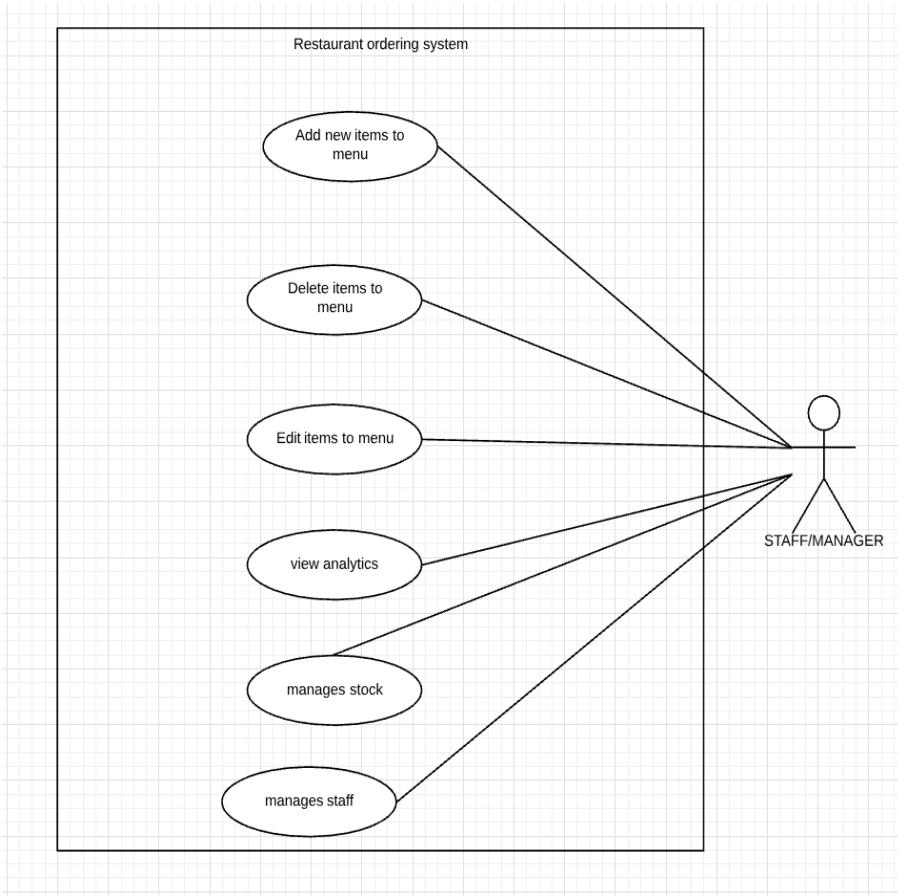


Figure 4-5 Use Case of Third Iteration

Description: The third iteration will contain the details of roles the restaurant manager. The manager can create, view, update and delete menu items.

Managers are also responsible for staff management and stock management. By capturing customer details in the first iteration managers can view weekly, monthly and annually analytics of target sales.

Pre-condition: Managers knows information about menu, staff and products.

Post-condition: Managers can view targets and what products needs to be ordered.

Exceptional Errors: Manager is the only one who can view the analytics

Alternative Variables: Manager logs into staff page, the role and position must be clearly stated

4.2.3.2 Sequence Diagram

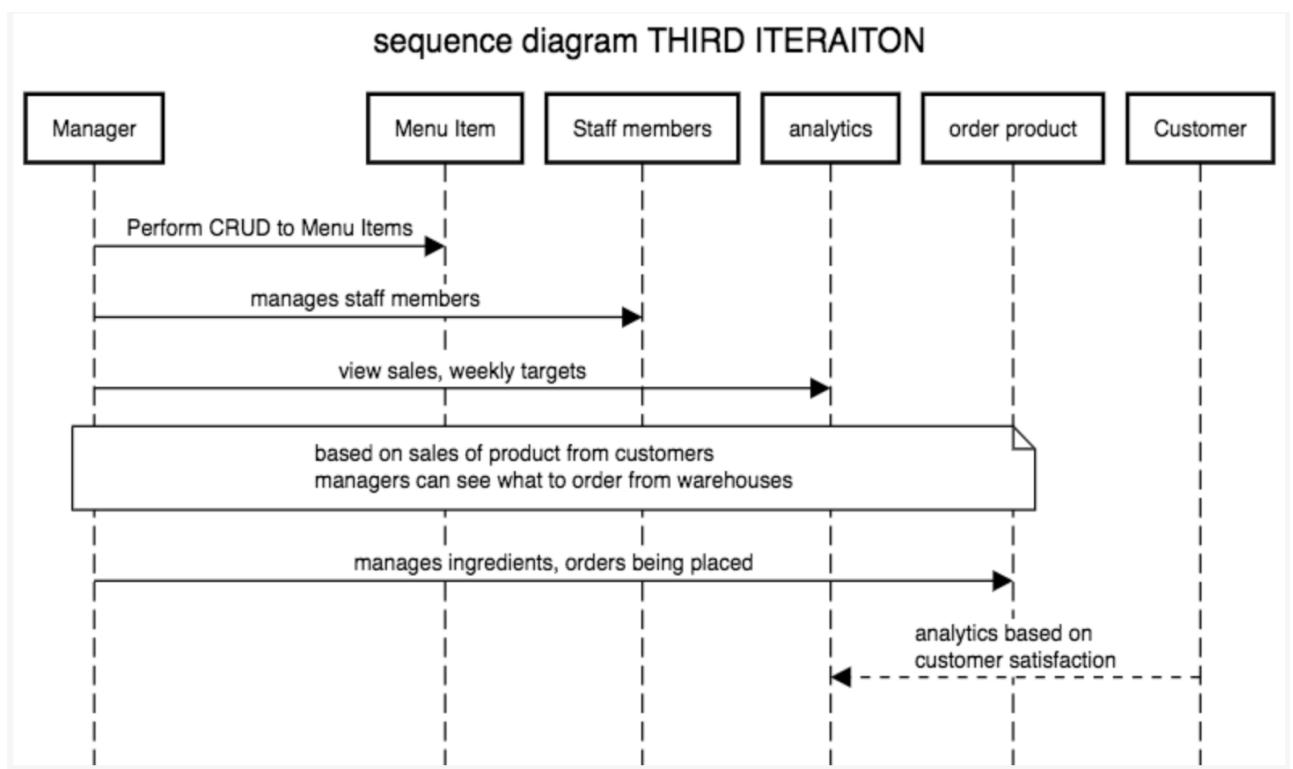


Figure 4-6 Sequence Diagram of Third Iteration

Managers can perform create, read, update and delete (CRUD) operations on menu items. Managers can manage staff members by adding new employees and organizing roster. Based on customer data and sales transactions manager can see weekly/monthly reports which will let manager learn how much stock needs to be ordered. Managers are the only actors that can access this information.

5. Prototyping and Development

5.1 Front End Development

Using Adobe XD CC (23) I was able to design a GUI to show the design of the final product. I chose a simple colour scheme because with colour blindness accessible feature in mind I decided to go against any colour that would be too harsh. The three colours that most of the population in the world can see would be white, grey and black. Some cases a tint of yellow could be recognized to vision impaired also.

5.1.1 Restaurant Application User Activity Prototype

Considering the possible colour schemes, I decided to use warm tones. Warm tones are appealing to the eye and monochromatic tones suggest clarity. I will use the below colour scheme for the application user part.



Figure 5-1 Application Colour Scheme

I also tested this colour scheme with an application colourblindgo that allows normal vision users to see what colour-blind individuals could see. In Figure 5.2 we can see 4 different types of colour blindness and the colour scheme it shows. I will be using this application throughout the GUI designing phase.

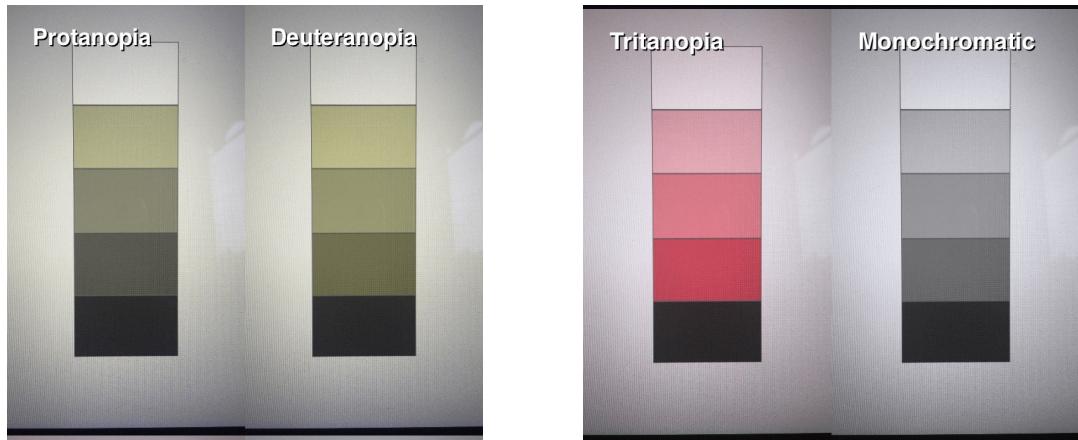


Figure 5-2 ColourBlindGo Colour Blindness Assistance

Using the same software, I am able to demonstrate how to navigate through the application. Certainly, more development on the user interface will be implemented as the project progresses.

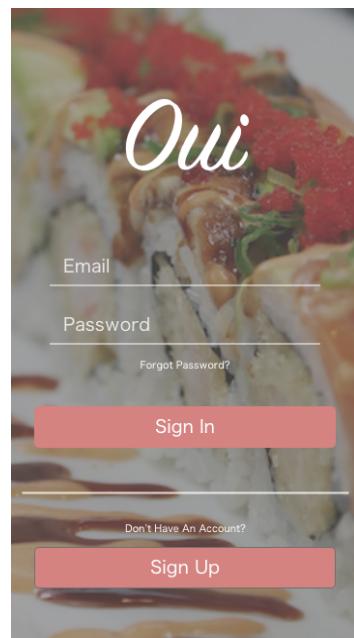


Figure 5-3 Application Login Page Prototype

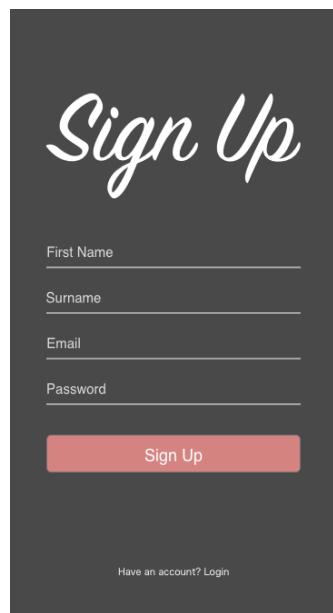


Figure 5-4 Application Register Page Prototype

Adobe XD delivers simple straightforward design mechanism that allows developers to visualize the end product. Here in Figure 5-3 and 5-4 are my design of this projects user authentication activities.



Figure 5-5 My Prototype QR scanner

Using Zxing library I was able to implement a functional QR code scanner activity. Notice the yellow dots as this is how the information is stored then decoded and displayed as output.

5.1.2 Point of Sale Management Activity Prototype

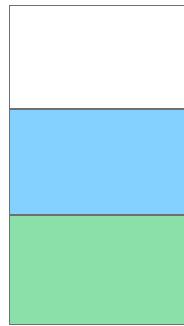


Figure 5-6 POS Colour Scheme

For the manager POS part I will use the triple colour scheme consists of green representing wealth and tranquillity, blue which stimulates productivity and white for neutrality. In the figures below shows a prototype implemented by adobe XD of a graphical user face designed for the business side.

Figure 5-7 POS Staff Login Page

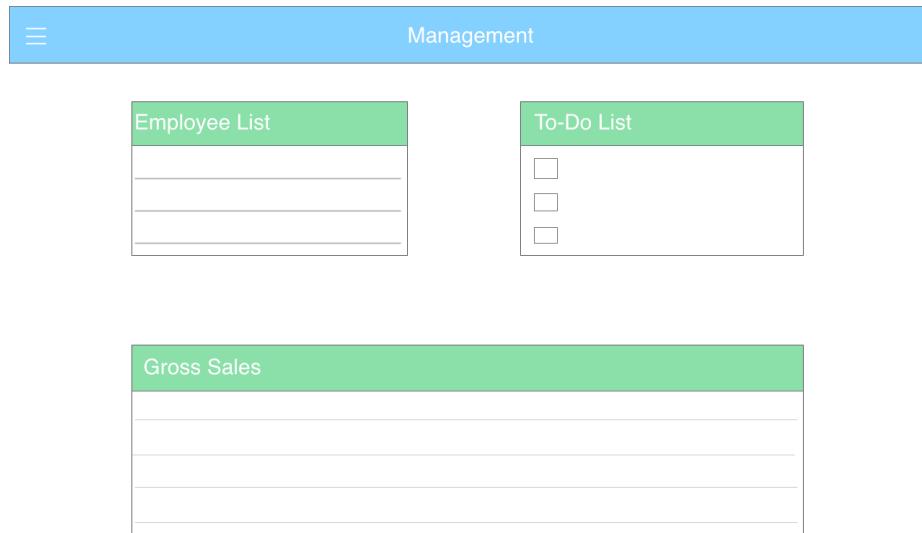


Figure 5-8 POS Management Page

Figure 5-7 and 5-8 shows the appearance of point of sale management activity. It will be a clean and minimalist front-end design allowing for efficiency and smooth navigation.

5.1.3 Heuristics Evaluation

Heuristics evaluation (24) is a usability inspection technique in which certain criteria are to be met. This evaluation demonstrates clear step by step rules to follow while making interactions between man and machine understandable. There are ten principles that needs to be considered when design a graphical user interface

5.1.3.1 Visibility of system status

The application should keep the users up-to-date through appropriate notifications within moderate time. In the case of this project, real-time notification can be sent to users to make them aware that the order was received or is in preparation. This allows interaction between the user and the business.

5.1.3.2 Match between system and real world

The application should be able to speak the language of the user. Allowing information to appear in a natural and orderly manner matching real world scenarios. The interaction between the user and the mobile device is essentially that the mobile device is a “personal waiter”. This personal waiter should cater the user accordingly for instance different languages used.

5.1.3.3 User control and freedom

Supporting undo and redo. User should have control on navigation throughout the application without any complexity. Modifying of orders or deleting items should all be accessible to users, giving them a certain management. Likewise, for manager POS, this application will allow managers to perform tasks that gives them certain responsibilities.

5.1.3.4 Consistency and standards

User interface should have a sense of consistency and sequential actions. Having prior knowledge of the used of the app users will adapt to the layout without any trouble. They will be able to guide them self naturally.

5.1.3.5 Error prevention

Before committing to another activity, a confirmation message is shown to users for assurance. Eliminate any error prone conditions to prevent unnecessary errors. This can be implemented when a transaction is taken place, preventing any false charges.

5.1.3.6 Recognition rather than recall

Users should not need to remember information from one part of the application to another. Instructions for usability should be evident while using the system.

Format of the application is consistent with little information displayed, users also have control and freedom within the application allowing them to review certain history data. For instance, payment history or order history.

5.1.3.7 Flexibility and efficiency in use

Functions in the system should accommodate both experienced and non-experienced users. Navigation will be a smooth and sequential transition allowing efficiency in use. Accessible functions allow users to customize a flexible user interface like different languages and text size enlargement.

5.1.3.8 Aesthetic and minimalist design

Information displayed should be kept at a minimal and only include materials that is relevant. Just like the manager prototype above, keep the layout basic and straightforward so that the user won't have a hard time finding information. The main activity should show simple information regarding restaurant details, employees and target for the day.

5.1.3.9 Assist users recognize, diagnose, and recover from errors

Error message and solution in plain text should be outputted to users. A query that is understandable to both experienced and non-experienced users. Users should be able to fix any problems that arise when navigating through the system in a non-programmatically way. Clear and short instructions will provide the user with guidance when deploying the errors.

5.1.3.10 Help and documentation

May be necessary to include help or documentation to assist users in using any hidden features. The help or information icon can be implemented into the

activities with an apprehensible structured down documentation detailing the functions that are available in the application.

5.2 Back End Development

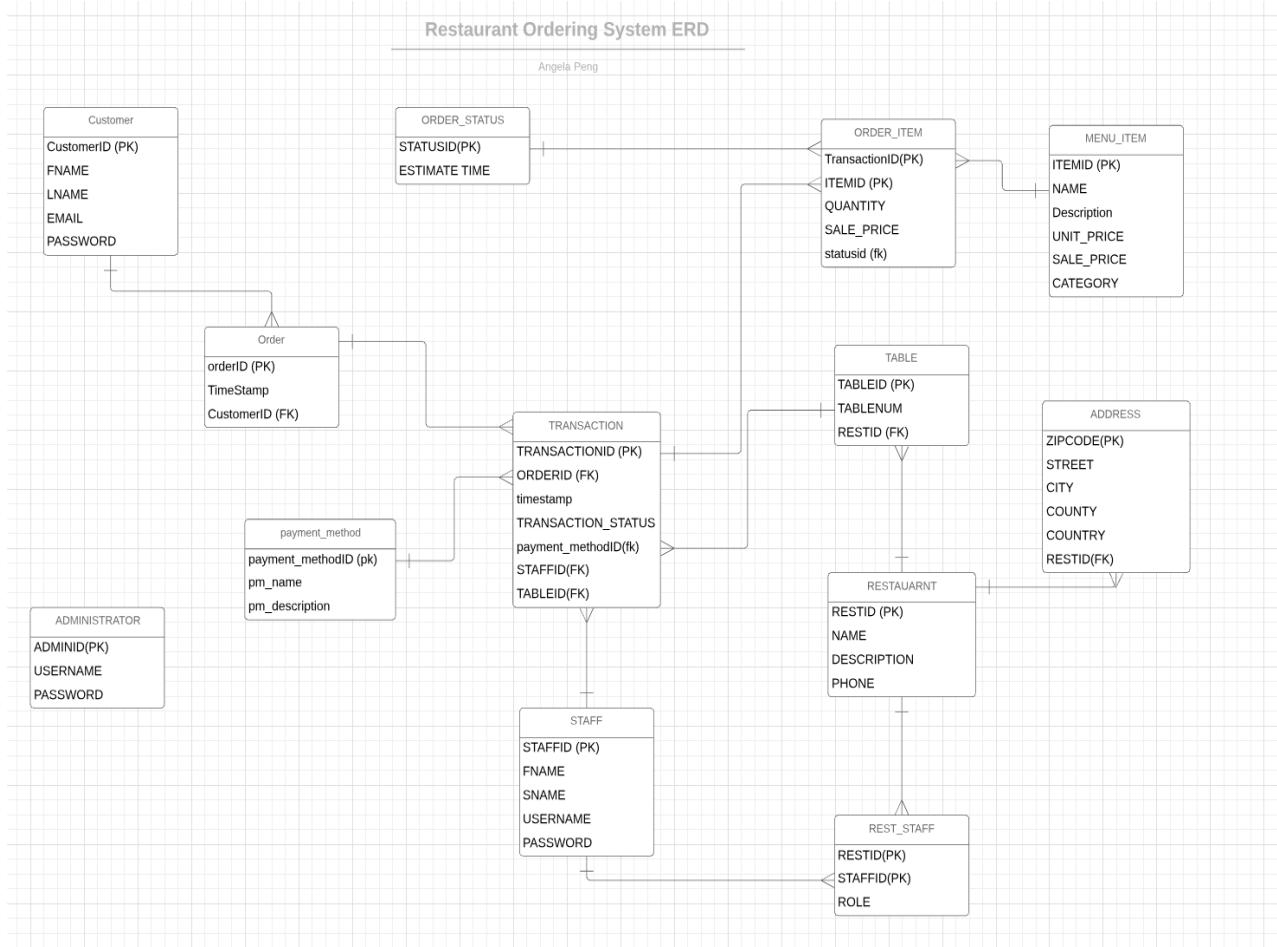


Figure 5-9 Entity Relationship Diagram

Entity relationship diagram represents relationships between entity sets that are stored on a database. An entity set is a collection of similar entities which can contain attributes that illustrate its properties. Firebase can perform basic create, read, update and delete (CRUD) functions smoothly making implementation easier and effectively.

In figure 5-8 we can see the relationships between each entity of this project. As I will be implementing this project using google firebase database service there will be no need for the use of foreign keys. Firebase is a cloud based NoSQL database server, data is stored in the structure of JavaScript Object Notation (JSON).

For my prototype, I developed a user authentication Login/Registration activity. Using firebase email and password authentication I was able to implement this function easily. When a new user is added to the database, firebase will show the new user on real-time.

Using firebase authentication user `sendEmailVerification()` function, email verification is sent to the users email account to verify if the user is authentic. Once the email is received by the user they are able to click on the attached link to prove of their verification.

6. Testing

My plan for system testing will be divided into four parts (25) ; Unit testing, Integration testing, System testing and Acceptance testing with additional UI evaluation technique. Each phase of testing will be documented for references in the event the same issue arises. In the process of performing these tests usability, functionality, loading and regression are considered. Overall testing will be carried out continuously throughout the project to ensure absolute functionality.

6.2 Unit Testing

In this phase, any implementation during implementation process is tested. Once a function or a block of code is produced I will be testing each part correcting and eliminating redundant code. Testing will be done on both virtual and physical mobile device. Because of the unfamiliarity of using firebase I will be testing new code regularly to monitor the performance.

6.3 Integration Testing

Integration testing should be done during and after a new activity is added to the main package. This testing shows the smoothness of performing a task, the flow of each activity and gives a general overview of how the application's final product. Integration testing can also eliminate any problems that may arise with conflicted code and makes implementing more efficient during development.

6.4 System Testing

After the development stage, I will gather three to five students from third year computer science. With a possibility of two have some form of vision impairment and two of which their native language is not English. Here I can see if this project can reach its potential with vision impairment and translation functionality. The reason I chose especially computer science student is because I would like to conduct the heuristics evaluation, in order to receive relevant feedback system testing should be performed by individuals with some knowledge of software development. This could provide me with a clear understanding of what aspects of the application needs maintenance before finalising anything.

6.5 Acceptance Testing

During this phase of testing will be done by potential end users. I will assemble a group of ten individuals which includes; 3 with vision impairment, 3 with mother tongue of a different language, 3 individuals with perfect vision with English proficiency, 1 individual with technical knowledge. This allows an unbiased

7. Issues and Risks

There are many challenges to face when developing this project. Below I will give a detailed description of each challenge that are unresolved or future risks that may occur and my approach to solving them.

7.1 Data Storage

As this is my first experience working with firebase database, there are many risks and challenges that will arise during the course of development. Firebase is formatted in JSON and does not comprehend SQL queries. This is a major drawback for me as my experience with data storage is querying. That being said firebase also have limits, as the application builds up and more data is added firebase will lag and performance will down grade. My approach to this issue is to research ways to prevent this from happening and alternative ways to implement around this challenge.

7.2 User Interface assistance

With the hope of incorporating unique aspect to this project being for users with vision impairment, I realised upon research the extent of complexity that may be required to implement this operations may take up a large amount of time. With

the certain amount of time to implement these operations, at least two function that will be included in the final product.

7.3 Third Party APIs

An extremely important risk factor is the use of third party APIs, especially with android studios there are many different APIs and libraries. This can change the whole implementation or design if installation of these APIs goes wrong.

Carefully conducted tests can help resolve some of these issues and implementing different sections on different workspace can also limit this issue from happening. This is a great challenge to overcome as I will be working with many APIs.

8. Plan and Future Work

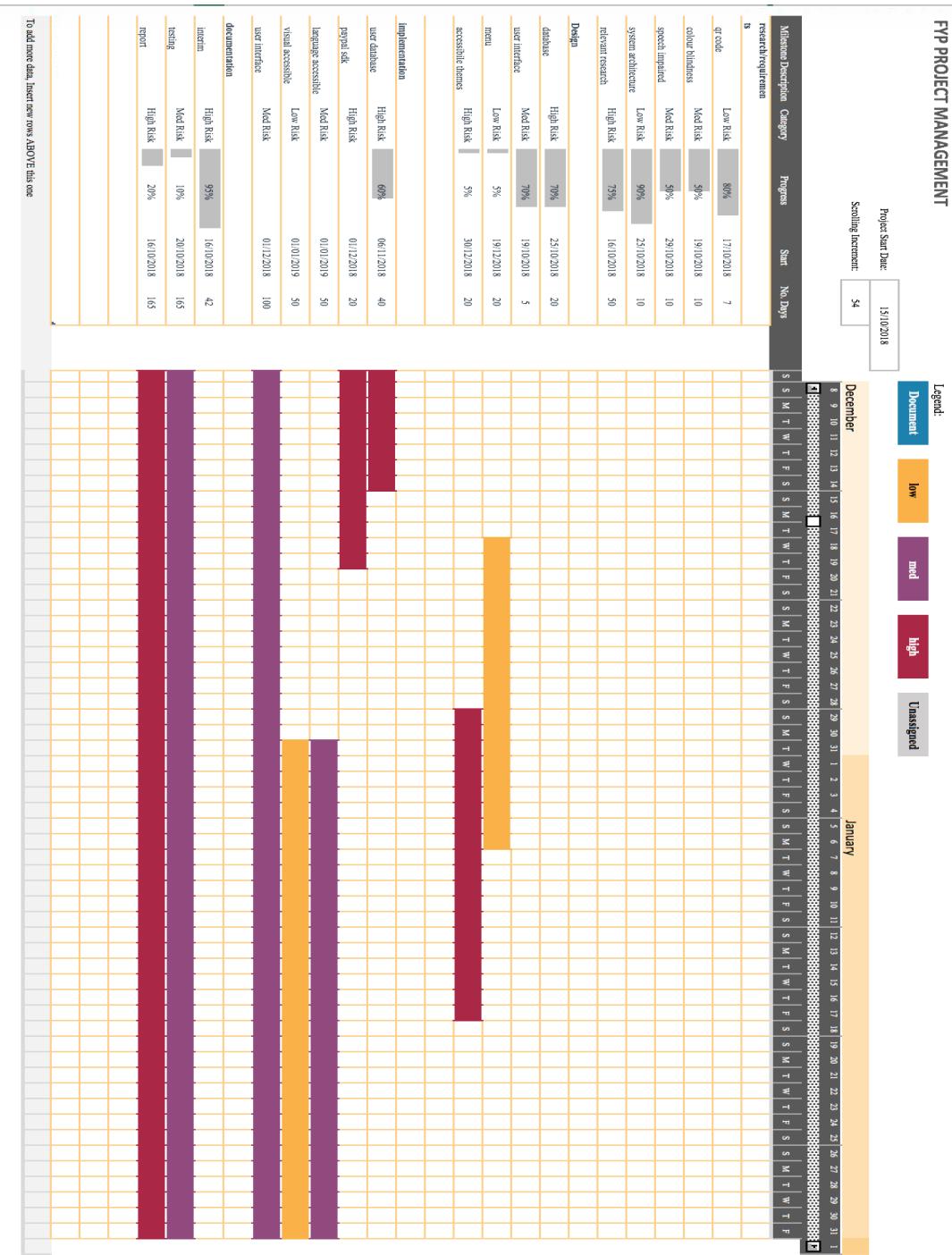


Figure 0-1 Project Plan Gantt Chart

The above figure shows my future plans for this project. It shows that I will be performing different tasks simultaneously, this can be time efficient. I will continue with my research as the project progresses. I aim to follow the initial design for the user interface but there may be slight changes that may fit to assist the user more advantageously.

I will continue with implementing and testing the application as I develop. At the same time, I will be documenting my experience, meeting logs with my supervisor, additional research and any issues or challenges that may occur. I understand that when facing with challenges or errors, it should be dealt with appropriately and at the same time expanding my knowledge.

I am determined to follow this plan progressively in a timely manner to complete a favourable final product. I expect to implement and incorporate as much accessible functions into this project as possible to create a comfortable environment.

Bibliography

1. **OED.** Oxford. *Oxford dictionary*. [Online] https://en.oxforddictionaries.com/definition/qr_code..
2. **Denso Wave, Corporate.** QR code. *Denso Wave*. [Online] <http://www.qrcode.com/en/history/>.
3. **OpenJaw.** The Rise of QR Code in China and its Effect on the Travel Industry. *OpenJaw Tech*. [Online] 10 April 2017. <http://www.openjawtech.com/qr-code-travel-industry/>.
4. **Wechat, Tencent.** Wechat. *Wechat*. [Online] <https://www.wechat.com/en/>.
5. **Guinn, Justin.** What is a Point of Sale System? Software Advice. *Software Advice*. [Online] [https://www.softwareadvice.com/resources/what-is-a-point-of-sale-system/..](https://www.softwareadvice.com/resources/what-is-a-point-of-sale-system/)
6. **rangelov@orderwizard.mobi.** OrderWizard. *Google Play Store*. [Online] 19 September 2013. https://play.google.com/store/apps/details?id=mobi.orderwizard&hl=en_US.
7. **eHopper.** eHopper POS Feature. *eHopper*. [Online] <https://ehopper.com/product-features/>.
8. **R, Daria.** iOS vs Android Development: Which One is Best for Your App? *RubyGarage*. [Online] <https://rubygarage.org/blog/ios-vs-android-development>.
9. **Rouse, Margrate.** mobile device fragmentation. *TechTarget*. [Online] <https://searchmobilecomputing.techtarget.com/definition/mobile-device-fragmentation>.
10. **Khan, Uzair.** The pros and cons of native apps. *Clutch*. [Online] Clutch, 12 June 2018. <https://clutch.co/app-developers/resources/pros-cons-native-apps>.
11. **Parmar, Bhavin.** What is a web app? Pros & Cons of using them. *imobdevtech*. [Online] iMOBDEV Technology PVT. LTD., 1 July 2016. <https://www.imobdevtech.com/Blog/what-is-a-web-app-pros-cons>.
12. **Lorenc, Katarzyna.** 05 DEC 2016, KATARZYNA LORENC MOST POPULAR KEY PROS AND CONS OF NATIVE AND HYBRID APPS. *ready4S*. [Online] ready4S, <https://ready4s.com/key-pros-and-cons-of-native-and-hybrid-apps>.

- 5 December 2016. <https://www.ready4s.com/blog/key-pros-and-cons-of-native-and-hybrid-apps/>.
13. **Manchanda, Amit.** A Quick Guide to Firebase: An Innovation in Mobile App Development. *Dzone*. [Online] DZone, 18 April 2018.
<https://dzone.com/articles/a-quick-guide-to-firebase-an-innovation-in-mobile>.
14. **Vision, Mobile.** Release Notes. *Mobile Vision Developers, Google*. [Online] 20 December 2017. <https://developers.google.com/vision/android/release-notes>.
15. **zxing, github.** ZXing 3.3.3 API. *github. ZXing API*. [Online]
<https://zxing.github.io/zxing/apidocs/>.
16. **SalesForce.** Mobile CRM. *Salesforce*. [Online]
<https://www.salesforce.com/eu/learning-centre/crm/mobile-crm/>.
17. **Android, Developer.** Localize the UI with Translations Editor. *Developers*. [Online] 16 April 2018.
<https://developer.android.com/studio/write/translations-editor#editaddtext>.
18. **PayPal.** PayPal Developer. *PayPal*. [Online] <https://developer.paypal.com/>.
19. **BrainTree.** BrainTree A PayPal Service. *BrainTree Payments*. [Online]
<https://developers.braintreepayments.com/>.
20. **Zafar, Ayesha.** How to design mobile app experiences for the visually impaired. *Inside Design*. [Online] InVision, 2017 December 2017.
<https://www.invisionapp.com/inside-design/mobile-design-visually-impaired/>.
21. **Hughey, Douglas.** The Traditional Waterfall Approach. *Comparing Traditional Systems Analysis and Design with Agile Methodologies*. [Online] 2009.
<http://www.umsl.edu/~hugheyd/is6840/waterfall.html>.
22. **SAP.** Model View Controller (MVC). *SAP Documentation*. [Online]
https://help.sap.com/doc/saphelp_uiaddon20/2.05/en-US/91/f233476f4d1014b6dd926db0e91070/frameset.htm.
23. **Adobe, Creative.** Design already ahead of its time. *adobe XD CC*. [Online]
<https://www.adobe.com/ie/products/xd.html..>

DIT

DT282 B.Sc. (Hons.) in Computer Science International

Interim Progress Report 2018/19

24. **Nielsen, Jacob.** 10 Usability Heuristics for User Interface Design. *Nielsen Norman Group*. [Online] 1 January 1995.

<https://www.nngroup.com/articles/ten-usability-heuristics/>.

25. **Inflectra.** Software Testing Methodologies - Learn The Methods & Tools. *Inflectra*. [Online] 1 March 2018.

<https://www.inflectra.com/ideas/topic/testing-methodologies.aspx>.