ECM3401 Individual Literature Review and Project

Halal Finder: Safe Eating Solution for the UK Muslim Population.

Acknowledgements

I wish to express my sincere gratitude to Professor Leon Danon for providing me with continuous support and motivation during the development of Halal Finder. To my Mother, thank you for encouraging me in all of my pursuits and pushing me to follow the path of academia. To my Father who spent the last 30 years working endlessly so his children could live the life he dreamt, thank you. To the College of Engineering, thank you for proving a platform for academia which provides students with endless opportunities.

Abstract

There are over 3.5 million Muslims living in the United Kingdom. Muslims eat Halal food as part of their religious practice, which requires them to locate genuine Halal outlets. Currently, it is difficult for Muslims in the UK to locate outlets selling genuine Halal food in their local areas. This can result in them feeling disconnected from their local communities. This report details the research and development behind Halal Finder, an application which aims to connect Muslims to genuine Halal outlets nearby. The application has three main requirements: locating Halal outlets, ensuring these outlets are genuinely Halal, and providing the user with a good experience. There are a wide variety of technologies available for creating Halal Finder, with several decisions to be made at each step in the process. This includes the selection of application platform, map interface, Optical Character Recognition (OCR) technology and recommendation system. Having examined the final application with Muslim users, it was concluded that Halal Finder meets the critical needs of the Muslim community when locating Halal outlets, and thus a successful application has been developed.

I certify that all material in this dissertation which is not my own work has been identified.

Contents

1. Introduction

1.1 Background Knowledge

Members of the Islamic faith only consume meat which is categorised as Halal. The word Halal is an Arabic word which translates to 'permissible'; in the context of food this means meat which Muslims can consume under Sharia law [1]. Animals which are not slaughtered according to Sharia law are Haram (forbidden) for Muslims to consume. The Halal meat industry is worth over £4.5 Billion [2], and several Halal certifying bodies have established themselves in order to regulate and monitor businesses serving Halal food. Muslims currently form more than 5% of the UK population [3] and with this percentage rising, the Halal meat industry plays a vital role in serving the growing needs of the Muslim community.

1.2 The identified problem faced by British Muslims

At present, it is very challenging for UK Muslims to locate Halal outlets such as grocery stores, butcheries and eateries since the majority of large franchises do not openly advertise that they sell Halal products, and small local businesses have little exposure. Being unaware of Halal outlets within their local area can lead to Muslims feeling disconnected from their community, and often results in them travelling to other areas to purchase Halal products. Not only does this cause inconvenience, but also has adverse effects on the local community as members are not able to support local businesses.

Fraud is a major issue in the Halal meat industry, with many outlets falsely claiming to serve Halal meat in order to gain a larger customer base. Halal certifying bodies have attempted to tackle this problem by only issuing certificates to genuine Halal serving outlets which follow correct protocols. The recent rise in the number of non-Halal outlets falsely claiming to hold legitimate certification, in combination with several scandals involving certain Halal certifying bodies has resulted in the Muslim population losing their trust in Halal food providers. In 2017, it was reported that 85% of products labelled as Halal could not be verified [4]. The head of the UK's global supply chain development Phil Hadley has recently stated that *"The Muslim community both in the UK and overseas is growing at a faster rate than the population in general, therefore the relative significance of the Halal markets is only likely to increase"* [2]. From a business perspective, the growth of the Muslim population is healthy for the economy but does mean that businesses are responsible for providing genuinely Halal food to the Muslim consumers. Thus, it is more important than ever before to successfully validate whether an outlet is selling genuine Halal food.

1.3 The proposed solution: Halal Finder

The aim of my project is to create an application which tackles the current challenges faced by Muslims when locating Halal food. The application will have three key focuses to ensure a complete solution to the problem is developed: locating Halal outlets, verifying these Halal outlets are genuine and providing a good user experience. The application will be called Halal Finder.

Halal Finder will provide a platform for people to discover Halal outlets including grocery stores, butcheries and eateries in their surrounding area, thus making use of geolocation [5]. The application will allow users to view outlets on an interactive map in the form of tags, which will hold information about the outlet. Users will be encouraged to share information with other users by adding new outlets to the map using geotagging [5]. In order to ensure that the outlets displayed by Halal Finder are genuine, certificates issued by a regulation body will be processed. There are several Halal certifying bodies and for the purpose of this application we will only accept certification from the Halal Monitoring Committee (HMC) [6], an independent, non-profit registered charity which monitors, inspects and certifies Halal products. HMC are currently the most trusted UK Halal certifying body and have a respectable reputation in comparison to other bodies.

For a user to successfully add a new outlet to Halal Finder, they will be required to upload a photograph of the outlet's HMC certificate. The application will use Optical Character Recognition (OCR) [7] to extract data from the certificate image, this data will then be passed to an algorithm to determine whether the certificate is valid. Only outlets with valid HMC certificates will be added to Halal Finder. The application will also incorporate other essential features to improve the user experience: users will be able to rate and review

their experiences at outlets and a recommendation system will be implemented to make tailored eatery suggestions to users. Having conducted research, I have found a small number of applications which also attempt to tackle the challenges faced when locating genuine Halal outlets. Thoroughly investigating each application has revealed that they all share common problems which Halal Finder aims to solve.

1.4 State of the art

After conducting thorough research on current mobile applications used to locate genuine Halal food, I have gained a clear understanding of the five main applications used in the UK: Halal Spot [8], Halal Trip [9], Halal local [10], Halal Food Finder [11] and Halal LookApp [12]. All the applications investigated shared common problems which result in each application not fully solving the problem of locating genuine Halal outlets. Therefore, there is still a gap in the application market to provide an application which can locate genuine Halal outlets. Below are some of the common problems I have discovered with the five applications:

- **Data Integrity** – Halal Spot and Halal Trip use outdated information, claiming outlets are Halal despite holding expired HMC certificates. This is a major concern as these outlets are officially classified as 'non-Halal' thus the applications are misguiding their users. One of the several checks Halal Finder will run when processing HMC certificates will be to read its expiry date and use this information to inform users whether outlets hold valid HMC certification, thus resolving this issue of data integrity.

- **Incorrect Verification Techniques** – All of the applications investigated are accepting verbal confirmation of HMC certification. The businesses themselves contact the application providers and verbally confirm that they are selling HMC certified products, without presenting any valid HMC certificate. Outlets will often claim to be Halal in order to be promoted on the applications and gain business, raising a serious issue of trust. By accepting verbal authentication, applications are providing a loophole for business owners to mis-sell products to Muslim customers. My application aims to solve the issue of including bogus Halal outlets by requiring a picture of the Halal certificate to be uploaded in order to add an outlet. Several checks will then be conducted on the certificate picture to determine whether the outlet is added/rejected.

- **User Experience** – The majority of applications investigated [9], [11], [12] do not provide the user with a good experience when using the application to achieve specific tasks. This was primarily due to very poorly-built and unresponsive Graphical User Interfaces (GUIs). Furthermore, the design of certain applications [8], [9], [10] make it difficult for the user to view information. For example, outlets displayed in a list format make it challenging for the user to assess the distance from their location to an outlet. Moreover, applications such as Halal Local and Halal Trip are crammed with additionally functionality such as prayer time alarms and calendars, making the user lose focus on the main purpose of the application. My application will aim to solve all these problems by providing the user with a high-quality GUI which utilises a simple map on which users can clearly view and locate outlets. Furthermore, I will ensure that additional features which do not add value to locating Halal outlets are not implemented.

Halal Finder will utilise a recommendation system to make eatery recommendations to users. These recommendations will be based on the user's eating habits and the ratings they have allocated to previously visited restaurants. Currently, no application which I have investigated holds a recommendation feature. However, applications such as Google [13] and Trip advisor [14] are able to make eatery recommendations to users. My application will better fulfill the Muslim communities' needs than Google and TripAdvisor as I will ensure the outlets recommended are HMC certified, a feature neither of the applications possess.

1.5 Market Research

Conducting market research on the dietary needs of the Muslim community was crucial prior to the development of the application. The research allowed me to determine the scale of the problem, identify my target audience and develop the application to best meet user requirements. As part of my market research, I distributed a survey with five YES/NO style questions at Bristol Jamia Mosque. Table 1 shows the questions

asked, the purpose of these questions and the results obtained by all 40 candidates who completed the survey (24 male/16 Female).

Question	***Purpose***	***Results***
1. Do you try to ensure the food you eat is from Halal outlets?	To gain the importance of Halal eating to the Muslim community.	100% - YES 0% - NO
2. Are you 100% confident that the food you buy, stated as Halal, is genuine?	To understand if the candidate is aware of businesses selling ingenuine Halal goods.	15% - YES 85% - NO
3. Have you ever used a mobile application that is able to locate Halal food?	To ascertain the popularity of Halal food applications.	25% - YES 75% - NO
4. Do you find it difficult to locate Halal outlets in Bristol or other UK cities?	To understand the scale of the problem when locating Halal outlets.	90% - YES 10% - No
5. Would you consider using an application that is able to locate genuine Halal outlets in the UK?	To understand the demand for Halal Finder.	90% - YES 10% -NO

Table 1: Shows the questions asked during market research, their purpose and the results gained.

Results from question 1 of the survey show that 100% of candidates make a conscious effort to consume Halal meat, demonstrating the importance of locating genuine Halal outlets to Muslims. The results from question 2 show that Muslims are well aware of the fraud in the Halal meat industry, with 85% of candidates being doubtful as to whether the Halal food they source is genuine. A quarter of the candidates surveyed had previously used an application to locate Halal Food and through engaging in conversation with these 10 candidates, I learnt that the application they had all used was 'Halal Spot'. Having asked how frequently they use the application, the majority of candidates answered "not often", they explained this was because the application did not contain up-to-date information.

The response of candidates to question 4 is meaningful to this project; we understand that there is an obvious need for an application that is able to locate genuine Halal outlets, with 90% of the candidates surveyed finding it difficult to locate genuine Halal outlets. Results from question 5 clearly show a strong demand for an application to locate Halal outlets as 90% of candidates would consider such an application. The results from question 5 were further analysed by looking at the age of the candidates to specifically determine the target audience of Halal Finder. Figure 1 shows the age distribution of the 36 candidates that would consider using an application to locate Halal food. The results gathered were insightful, showing a strong interest from 15-36-year old's and a significant lack of interest from the 37-50+ age group.

Through conversing with candidates who engaged with the survey, I generally found that the older members of the Muslim community were not attracted to the idea of using a mobile application, assuming they would find it 'confusing' and would not be able to operate the application correctly. In contrast to this, I found the younger generation were excited by the idea of Halal Finder and were confident they would utilise the application as long as it is "easy to use" and can "correctly locate genuine outlets". From the research conducted, it is clear that the target market for Halal Finder is 15-36 year olds. However, by developing a simple and easy-to-operate application, I believe that I will be able to increase the interest shown by the 37-50+ age group.

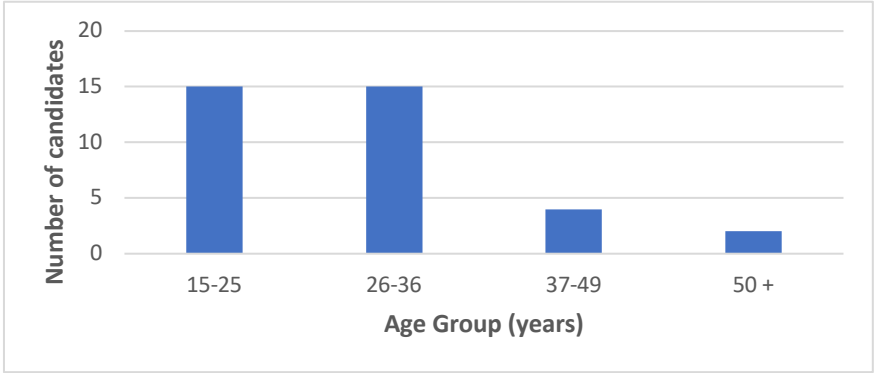


Figure 1- Bar chart showing age distribution of candidate who responded 'Yes' to Question 5.

2. Summary of Literature Review

This section will be dedicated to summarising the research and analysis of a variety of sources which are relevant to this project. This research has established which methods and technologies I have utilised during the development of Halal Finder.

2.1 Related sources to Mobile platform choice

There are two main approaches to developing a mobile application; designing a native application or designing a cross-platform application. Native applications are built for a single specific platform. Android and iOS are the dominant mobile platforms in the UK mobile market with Android penetrating 49.69% of the market and iOS penetrating 49.24% as of September 2018 [15]. There are several advantages of building native applications in comparison to cross-platform applications. Since native applications are developed for a specific platform, they effectively utilise the system through accessing private native Application Programming Interfaces (APIs) in the phone's operating system, such as the geolocation and camera API this has been shown in studies such as [16],[17]. The main disadvantage of native applications is fragmentation [19]. Since iOS and Android are two separate operating systems, there is no cross-compatibility meaning Android application code will not run on iOS and vice versa. With iOS penetrating 49.24% of the market and Android penetrating 49.69% [15], building a native application will significantly isolate a large portion of potential users from being able to use Halal Finder.

Cross-platform applications are compatible with multiple operating systems. There are two main approaches to cross-platform applications; web and hybrid applications. The main advantage of cross-platform applications in comparison to native applications is their ability to run on multiple different platforms, allowing the application to be available to significantly more users [18]. A common problem with all cross-platform applications is the user experience the application provides [17]. Table 1 gives an overall comparison of native and cross-platform approaches. The table demonstrates that although native applications limit the number of potential users, they provide a higher quality application and excellent user experience in comparison to cross-platform applications. It was decided that Halal Finder would be a native Android application as a key focus of Halal Finder is to provide the user with a high-quality application.

Decision Criterion	Native Approach	Cross-platform Approach
Quality of UX	Excellent	Not as good as native apps
Quality of apps	High	Medium to low
Potential users	Limited to a particular platform	Large – as it reaches to users of different platforms

Table 2: Comparison of Native and Cross-platform approaches [19].

2.2 Map Implementation

Halal Finder will utilise an interactive map which displays the user's position and the outlets around them, allowing them to easily explore the outlets in their surroundings. Outlets will be displayed on the map as tags, users can tap on tags to view more details about outlets. Google Maps has over one billion monthly users in over 200 countries, making Google Maps the most used map system globally [20]. Furthermore, Statista has reported that as of June 2017, Google Maps had a 98.33% market reach on the UK Android platform [21], demonstrating that this interface is very popular amongst UK Android mobile users. Therefore, it would be advantageous to incorporate Google Maps in Halal Finder as the majority of users will be familiar with interacting with the map interface. Additionally, as a result of the popularity of Google Maps, there is a lot of documentation for developers which would be beneficial during development.

2.3 Recommendation System

Halal Finder will implement a recommendation system which will allow eatery recommendations to be made to the user. Recommendation systems rely on processing data from their user base to gain an understanding of users, which can then be used to make recommendations to them. Collaborative filtering (CF) systems are based on the principle that if two users have a similar rating history then they will behave similarly in the future [22]. Recommendations of items are made to a user based on those items highly rated by other users

similar to them [23]. Implementing a CF system is applicable to Halal Finder as it can produce serendipitous eatery recommendations to users, encouraging them to explore eateries.

2.4 OCR API

Halal Finder will use an OCR API to extract text from the image of the HMC certificate uploaded by the user. The HMC certificate contains two key pieces of information that need to be extracted by the OCR API: expiry date and certificate authoriser. The extracted text will be used by an algorithm to verify whether the certificate is valid/invalid. Google's Cloud Vision API [24] is a powerful image analysis tool which is able to recognise text, objects and logos. Through the use of advanced machine-learning models utilising convolutional neural networks [24], the Cloud Vision API can provide very accurate results. Table 3 compares the performance of the Cloud Vision API and another popular OCR API; Microsoft Computer Vision API ascertained by a study conducted by Playment [25]. The results from this study demonstrate that the Cloud Vision API is significantly more precise.

	Google's Cloud Vision	**Microsoft Computer Vision**
Total Images Processed	3001	3001
Extracted data	2762	1328
Correctly extracted	1839	463
Incorrect extraction	923	865
Precision	66%	34%

Table 3: Study of Google's Cloud Vision API VS Microsoft Computer Vision API.

3. Project Specification

This section will define both user requirements and system requirements. The purpose of Halal Finder is to locate genuine Halal outlets. My research on current state of the art, covered in section 1.4, highlights the key issues with current applications, from which I have learnt that Halal Finder must achieve the following three objectives: provide the user with accurate outlet information, correctly verify the Halal status of outlets and provide a good user experience. Several functional and non-functional requirements need to be met in order to successfully achieve these objectives.

3.1 Functional Requirements

Functional requirements specify what the system should do and how this will be achieved. Halal Finder has three primary functional requirements: locating Halal outlets, ensuring these outlets are genuinely Halal and enhancing the user experience.

3.1.1 **Locating Halal outlets** – A map interface will be implemented to enable users to locate Halal outlets, displaying the user's current position and surrounding outlets as tags. Users can interact with the map to explore their local surroundings or different areas of the UK. Users can view an outlet by tapping on a tag, which will then provide information about the outlet such as the name, address and image of the Halal certificate.

3.1.2 **Verifying Halal outlets** –When a new outlet is added to Halal Finder, a photo of the outlet's HMC certificate is required to be taken by the user. An OCR system will extract data from the certificate image, this will be passed to an algorithm which will conduct several checks on the data to ensure the certificate is valid, including a check of the expiry date. This will ensure only Halal outlets with valid HMC certificates are added to the application. If the certificate fails to meet any of the criteria, it will be rejected and the outlet will not be added to the application. Adding outlets in this manner ensures that all outlets displayed in Halal Finder have valid certificates.

3.1.3 **Enhancing user experience** – For Halal Finder to provide the user with a good experience the application has to be built to a high standard with clear and reliable functionality. Additional functionality to improve user experience will be implemented such as a recommendation system

to make eatery recommendations to the user. Furthermore, users will be able to review outlets and read the reviews of other users.

3.2 Non-Functional Requirements

A non-functional requirement is a requirement that specifies a criterion which can be used to assess the operation of a system. Halal Finder intends to achieve three non-functional requirements; these are: usability, extensibility and performance.

3.2.1 **Usability** – Halal Finder must be simple and easy to use, which will be achieved by implementing a user-centered design. Halal Finder will provide the user with a simple, non-cluttered GUI, making it easy to navigate around the application. Reliability of functionality is another important element that will enhance usability therefore it will be crucial to ensure that there are no issues in the general use of the application, including uploading certificate images and locating outlets.

3.2.2 **Product extensibility** – Halal Finder is a native Android application; to increase the product extensibility I will eventually build an iOS version of the application to run on Apple devices; significantly increasing the number of potential users. Although this will not be achieved within the scope of this project it will be considered as future work.

3.2.3 **Performance** – Slow and irresponsive applications tend to be disregarded by users [26] thus application performance is a key element to consider when developing applications. I will need to ensure that Halal Finder performs functionality without significant latency and does not over use the phone's resources.

4. Evaluation Criteria

Halal Finder can be evaluated through breaking down the requirements stated in section 3 into specific features and then examining whether these features function correctly. In this section I have mapped the application requirements and their corresponding evaluation criteria. The outcome of the project will be deemed successful if all the criteria below have been achieved to a high standard.

4.1 Locating Halal Outlets

To evaluate whether users can locate outlets I will examine whether the following has been achieved:

A) The map is implemented correctly and responds well to user interaction.
B) The user's location and outlet locations are accurately displayed on the map in the form of tags.
C) On selection of a tag, outlet information including the HMC certificate image is clearly displayed to the user.

4.2 Ensuring outlets are genuinely Halal

To evaluate whether Halal Finder can verify the Halal status of outlets added to the application, the following will be examined:

A) The application can access the device's camera to take a photograph of the certificate.
B) The implemented OCR system can extract text from the certificate image with a high confidence rate.
C) The algorithm which takes in the text retrieved from the OCR system and classifies whether a certificate is valid or invalid by conducting several checks is correctly implemented. The OCR system and algorithm will be tested by feeding the application both valid and invalid certificates and checking whether outlets with valid certificates are added and those with invalid certificates are rejected. The OCR system and algorithm will be classed as successful if the application has a high accuracy rate of differentiating between valid and invalid certificates.

4.3 Enhancing the User experience

The main feature of the application which will increase user experience is making eatery recommendations to the user, the examination criteria for which is the following:

A) Implementation of a collaborative filtering system to recommend eateries to the user. A group of test users will be asked whether they would consider visiting the recommended eatery suggested by the application. The recommendation system will be considered a success if the vast majority of users express that they would consider visiting the recommended eatery.

4.4 Usability

The usability of the application will be marked against the following criteria:

A) The application has a clear GUI which is easy for users to understand and execute functionality. Users will be asked to comment on the usability of the application, this criterion will be seen as successful if the majority of users believe that using the application functionality is simple and clear.

B) There are no bugs in the application which cause unexpected results or the application to shut down.

4.5 Performance

Application performance will be marked using the following criteria:

A) Users can conduct functionality such as locating and adding outlets in a timely manner.

B) The application does not overuse the phone's resources and in turn, drain the phone's battery. To measure this, diagnostics will be run on the phone to assess the resource levels used by Halal Finder.

5. Product Design

In this section, the chosen design for each component of the application will be described as defined in the specification. Through my market research and studying competitor applications, I learnt usability and user experience were vital factors I had to consider when designing the application. The design of the application aims to provide an intuitive interface, incorporating an array of features to maximise user experience. Having researched various design methodologies, I decided to adopt a user-centered design (UCD) approach [27]; this is an iterative process where each design decision is heavily focussed on the user's needs and experiences, tailoring the application specifically to its audience. An important consideration of UCD is the idea that every object and function has a clear purpose, which can be achieved by eradicating any redundant aspects of the application.

5.1 Overview of the Application Structure

Having established the design focus for Halal Finder, the next step was to design the structure of the application; including the layout of pages and the interactions between them through application functionality. Figure 2 depicts the application structure. Following UCD principles, I ensured the layout was as intuitive as possible for users. On opening the application the user arrives at a login/ register page which requires them to create an account or sign in. Once logged in, users are welcomed to the application homepage and presented with three clear options; 'Explore', 'Add Outlet' and 'Recommend'. Figure 2 demonstrates that the 'Explore' page contains the map interface, to which other pages are re-directed to. Minimising the number of windows in the application to five, each with their own distinct functionality, has resulted in easier user navigation and execution of functionality. In the following sub-sections I will break down the design of each of the five pages and the design decisions made.

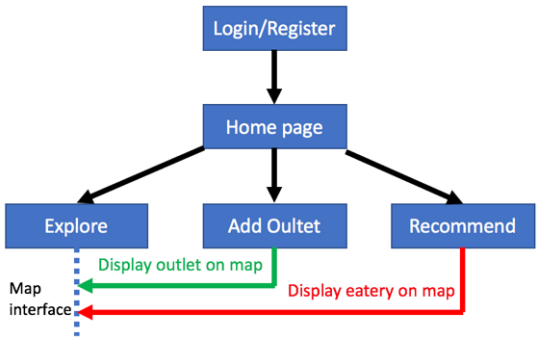


Figure 2: Structural diagram of the layout of pages in Halal Finder and the interaction between them.

5.2 Login/Register page

The Login/Register page is the first page users view on opening the application and therefore, strongly influences the user's first impression of Halal Finder. Having decided to take a UCD approach to the design of Halal Finder, I designed the page based around the functionality it was trying to achieve - logging the user in. Thus, I made the decision to display the username and password field immediately as opposed to placing a login button on the page which loads a form. This design choice means users have to navigate less through the application and the expected behaviour from the application is immediately translated to the user.

A register button is also displayed on this page. On selection, rather than opening a new window I chose to implement a pop-up window which requires the user's email and login. Once the registration form is completed, the pop-up window closes and the user returns to the login/Register page. An email is sent to the registered email with a link to choose their password, after which their account will be created. The design of the registration form includes form validation to avoid user mistakes and carelessness.

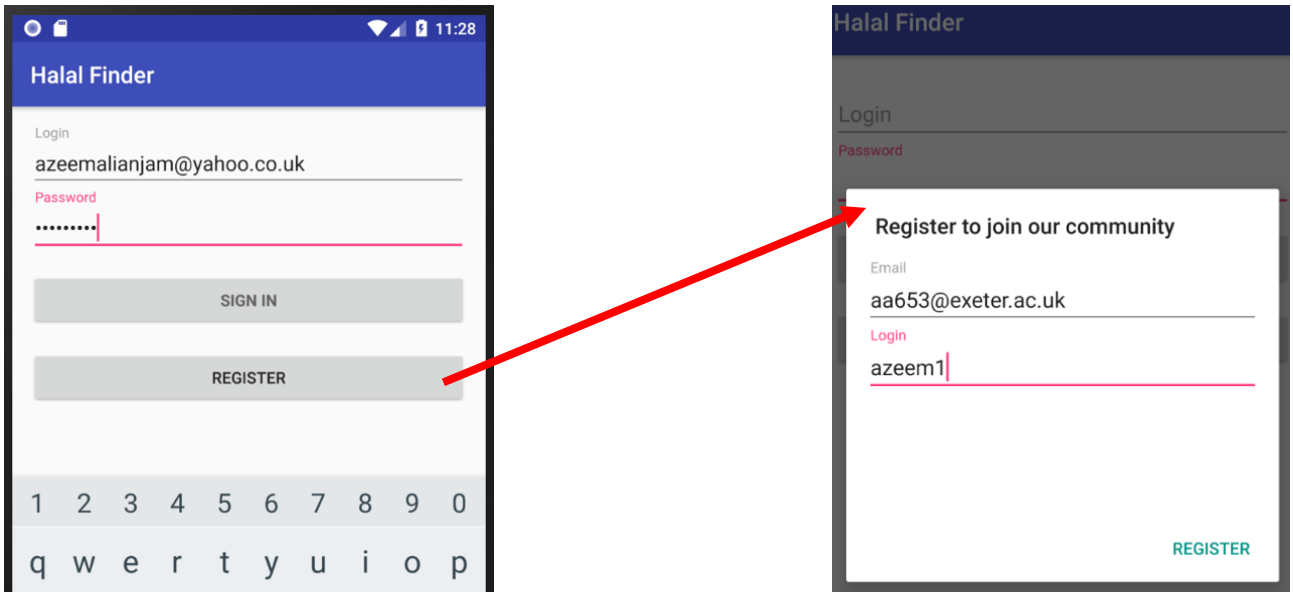


Figure 3 –Login page and the register pop-up window shown when pressing the 'REGISTER' button.

5.3 Application Homepage

Once the user has successfully logged in to Halal Finder they will be welcomed to the application homepage as seen in Figure 4. The aim of this page is to clearly present the user with three options: 'Explore', 'Add Outlet' and 'Recommend', which are represented by three distinct buttons on the homepage. Initially, it was planned that the homepage would open on a map interface displaying the user's location and nearby outlet tags. However, after investigating and reading user feedback of Halal LookApp, the homepage of which is a map interface, it is evident that many users have found it difficult to navigate from the map to other application features such as the Qibla locator. Therefore, a minimalistic design approach was taken when constructing Halal Finder's homepage to allow users to easily understand and navigate to the desired feature.

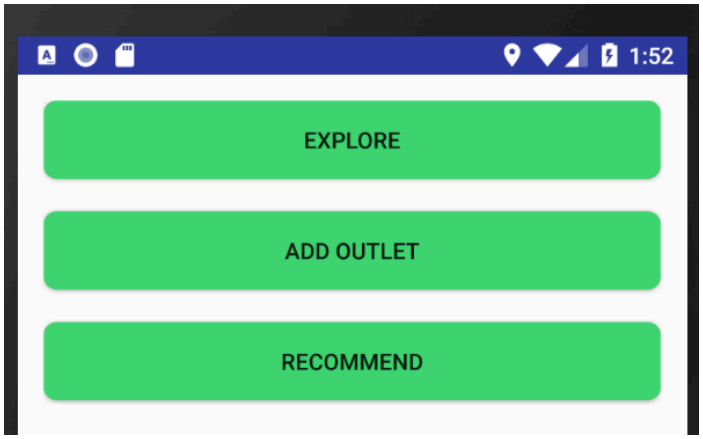


Figure 4 – Homepage of Halal Finder.

5.4 Explore

The 'Explore' page holds the most important functionality within the application; the ability to locate Halal outlets. Many key design decisions had to be made when constructing this page to ensure that it was easy for users to find Halal outlets. The main consideration when designing this page was the way in which outlets would be displayed to users. The traditional design method used by many competitor applications is to implement a list structure on the page, where outlets at the top of the list are nearest to the user's current location. However, I believe that implementing a list structure does not provide a good user experience of exploring outlets and decided that Halal Finder would utilise an interactive map.

When opening the 'Explore' page of Halal Finder, an interactive map fills the entire screen and clearly indicates the user's current position and Halal outlets as tags. Users can interact with the map through gestures such as pinch and swipe, allowing them to explore outlets in an easy and enjoyable fashion and in turn providing them with an engaging experience. To allow users to navigate quickly around local areas, the map will automatically adjust its bearings around the user's location, with their location tag being displayed in the centre of the screen, saving users time navigating around the UK map to find their location. A 'Current location' button is provided to users which returns them to their current location, saving them time navigating back to their current location on the map.

An important aspect of the interactive map is the way in which it translates the necessary information about Halal outlets to the user. Initially, the Google Maps default tag was used to display all outlets, with a small text box below each tag displaying the outlet name. After adding several neighbouring outlets, I presented the map interface to members of the Bristol Islamic society to gain feedback. The majority of feedback received was negative, the main concern being that the map interface was very hard to understand in areas with a high density of outlets. Figure 5 shows the previous design of the map interface, the map is indeed very clustered and difficult to interpret with the names of neighbouring outlets overlapping.

Taking this feedback into consideration, I took a different design approach. I decided to use three distinct icons to represent eateries, grocery stores and butcheries and only these icons would be placed on the map instead of the names of the outlets. This is because when a user wants to explore Halal outlets, it is likely that they will be searching for a type of outlet rather than a specific outlet, thus displaying the outlet name was not deemed necessary. After implementing the three different tags, the appearance of the map became more concise and easier to understand even in areas with several outlets. Figure 6 shows the new map interface in a heavily populated area. Figure 7 shows the icons used to represent a grocery, butchery and eatery respectively. When designing these tags much thought was given to the graphics used to ensure they were distinct and easy to interpret.

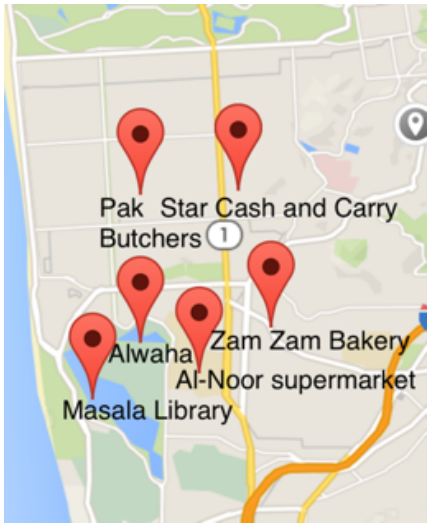


Figure 5 – Previous map design.

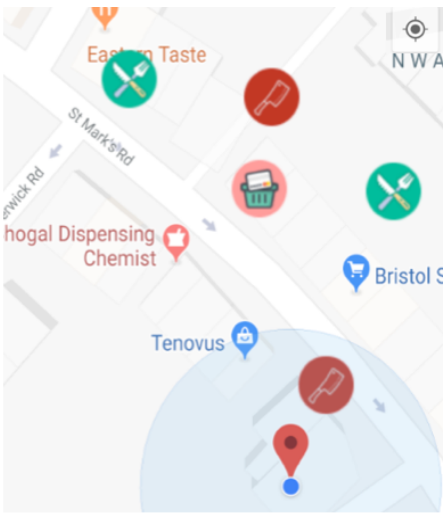


Figure 6 – Refined map design.

Figure 7 – Re-designed outlet tags.

Interaction with outlet tags is a key feature of the map interface, allowing users to gain relevant information about outlets. One tap on a tag will display the outlet name below the icon and two taps on a tag will open

a pop-up window revealing further information about the outlet. Pop-up windows are used to display outlet information since the window can be layered over the map. Thus, the user can tap on the map in the background after reading the outlet description and continue to explore other outlets, making exploring outlets a fast process. In contrast, if outlet information was displayed in a new window, the user would have to redirect themselves to the map, making the process of exploring time-consuming and ineffective. A minimalistic design approach was taken when constructing the layout of the pop-up window displaying outlet information, ensuring only key information which would help users make a decision was displayed. The pop-up window contains the name of the outlet, address, outlet type and ratings and reviews left by users.

5.5 Add Outlet

A key premise of Halal Finder is that it is a user-based application; users are encouraged to contribute to the content of the application through adding new outlets, to provide a better experience to fellow users. When designing the mechanism through which users add new outlets, simplicity was crucial in order to ensure the process was easy and required little time and effort, to encourage users to add new outlets. When the user opens the 'Add Outlet' window, they are presented with a simple form requesting basic information about the outlet which should be very quick and easy for the user to find out. The form requires the following information about an outlet: name, type (grocery, butchery, eatery), address and a picture of the HMC certificate. Figure 8 shows the form users are required to complete to add an outlet.

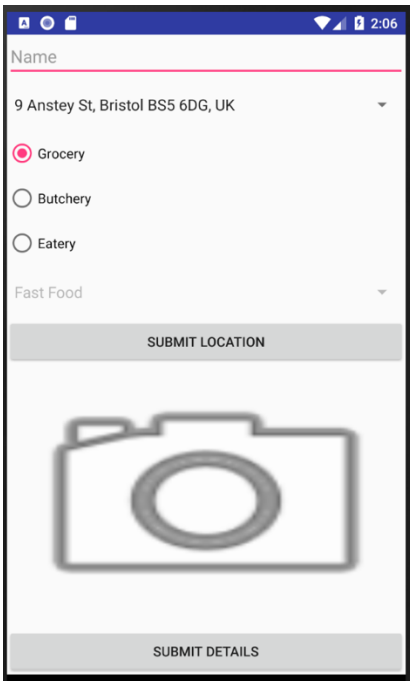


Figure 8 – Form users are required to complete to add an outlet.

Below the form are two buttons, one to retrieve the outlet's location and the other to upload an image of the HMC certificate. In some cases, the user may have to make some effort to find the address of the outlet and then fill this out, to overcome this problem it was decided to place a 'Submit Location' button on the 'Add Outlet' page. When pressed, this submits the user's current position as the outlet's address. This mechanism works with the general flow of the application as users are required to be in the outlet in order to add the business as they have to upload the HMC certificate displayed in the outlet. The camera button opens the device's rear-facing camera which allows the user to take a photograph of the outlet's HMC certificate, once the photo has been taken the user is given the option to 'Retake' or 'Use' the photo. On selecting 'Use', an algorithm will conduct OCR and extract the contents of the certificate. Several checks will then be conducted on the certificate to ensure that it is genuine and can be added to Halal Finder. Table 4 shows the specific checks conducted and the reasoning behind them.

Check Conducted	Reasoning
Check "HMC" is the certificate authoriser.	To ensure all outlets are HMC certified.
Check the certificate contains the phrase "Has attained the HMC Approval for selling HMC approved products".	To ensure the certificate is a valid HMC certificate as all valid certificates contain this phrase.
Check the expiry date is valid.	To ensure the certificate is valid and the outlet is abiding up-to-date HMC legislations at present.
Check for the presence of HMC's email address.	All HMC certificates contain this information, thus it can be used as a criterion to accept certificates.

Table 4: Checks conducted on HMC certificate image and respective reasoning behind each check.

Once the 'Add Outlet' form has been completed, location button selected and the HMC certification photo has successfully passed examination, the user can press the 'Submit details' button at the bottom of the page to add the outlet to the application.

5.6 Recommendation System

As the development for Halal Finder was time constrained, I organised development such that the most

crucial requirements that would strongly affect the application would be completed first and least important tasks would be completed during the finishing stages of development. When starting development, I knew that although making eatery recommendations was a requirement that needed to be completed it was not a priority. As when Halal Finder would be first released to the public it would not be possible for me to gain sufficient amount of user and eatery data such that good recommendations could be made to the users. With this in mind I decided to design and develop my recommendation system during the final stages of development.

A collaborative based approach was taken to develop the recommendation system, this was done using the main principles of the k-nearest neighbors (K-NN) algorithm [22]. Slight adjustments were made to the K-NN algorithm to tailor the recommendation system to what I believe would generate better recommendations given that user and outlet data was minimal.

Collaborative filtering systems (CF) are based on finding similarities between user behaviour, thus if two users have a similar rating history it is assumed that they will behave similarly in the future [23]. Using CF systems seemed logical as they have the potential to generate new and interesting eatery recommendations for users and not solely just on what the user likes, encouraging the user to explore. To implement the CF system the KNN algorithm was chosen as it is simple to implement and can be easily adapted to fit different requirements [28]. The K-NN algorithm works by merging users into groups called 'neighbourhoods', where users in each neighbourhood have similar ratings of items. The target user (the user who is being recommended an eatery) will then be examined on the ratings he has given eateries in order to determine which neighbourhood he belongs to. From this information, the highly rated eateries of other users in that neighbourhood will be recommended to the target user [28].

From the data gathered from Halal Finder I have a labeled set of users and a labelled set of eateries, with the aim of the recommendation system being to use this data to find similar behaviour amongst users and in turn generate recommendations for the target user. To accomplish this, the ratings of each user for every eatery present in the application is represented in a vector format. In the case that a user has not been to an outlet a rating of three is placed in the index for that outlet's rating in the vector rather than 0. Placing a 0 would cause great divergence even between similar users in later steps when calculating the similarity, thus it was decided a middle rating of three would be given. Once we have computed all the rating vectors for each user in the application, we have to determine the similarity between the target user and all other users. This is done by calculating the Euclidean distance. The Euclidean distance is a similarity measure which calculates the straight-line distance between two points in n-dimensional space and through this calculation it is able to measure the similarity between users [29]. The Euclidean distance is calculated between the target user's rating vector and the rating vector of each other user. From this we order all the distances from the smallest to greatest. The smaller the distance the more similar the behaviour of the user is to the target user and the greater the distance the more different the behaviour of the user is to the target user. It was decided that the three smallest distances would be picked thus we would be choosing the three closest 'neighbours' to the target user. The reasoning behind choosing the three closest users is that within the three respective rating vectors for the users there is a high probability of a sufficient number of highly rated eateries, thus a good number of eateries can be recommended to the user.

Once we have filtered the user base to only the three closest users who show similar behaviour we process each of the vectors for these three users to find a suitable eatery for the target user. Two criterions were put in place to ensure the recommendations generation was best for user experience. Criteria 1 is that only eateries that the user has currently not been to would be recommended, criteria 2 is that all eateries with a rating of three or above would be recommended. Criteria 1 is used to provide the user with a new eatery increasing user experience as it encourages them to try something new. Criteria 2 is used to ensure that the eateries recommended are to a certain high standard as I did not want to recommend eateries that were poorly rated by users with similar behaviour. There was a possibility that none of the eateries in the three vectors were three or above. As I always want to generate a recommendation to the target user, in this particular case the recommendation system would output the highest rated eatery in the application. This

made sense as if the majority of users highly ranked a specific eatery there is a strong probability that the target user would like it.

To clearly show the eateries which were recommended it was decided that the user would be re-directed to the 'Explore' window with the interactive map only showing the recommended eateries and the users current location as tags. Displaying the recommended eateries on a map rather than in a list better translates to the user how close/far the eatery is.

6. Development

6.1 Methodology to Development

The primary consideration that had to be made prior to development was which development methodology should be adhered to and to do this, I looked at what was required from the end solution. After conducting market research, it was clear that users were interested in the Halal Finder application but only if it sufficiently met their needs. Thus, during development it was crucial to ensure that all functionality built was in accordance to user preferences, meaning users would have to periodically examine application functionality throughout this stage, allowing the application to fine-tuned. With this in mind, the only logical approach to software development was Agile development [30]. Agile development is a method of software development by which the solution constructed transforms through the collaborative participation of developers and the end user. Other approaches such as the Waterfall approach [31] were not considered as they would mean users would have no interaction with the product during development, but rather only after development had been completed. Thus, leaving no time to implement important user feedback which could potentially improve the application. The main benefit of using Agile development is that it allows functionality to rapidly appear in early stages of development with the ability to verify that it is working correctly without having the complete solution. Having functionality appear quickly through development allowed me to frequently test the functionality with end users and implement their feedback where necessary.

Once my approach to development was chosen, I had to establish a back-log of tickets and organise the development of Halal Finder into sections. A key part of Agile development is constructing a back-log containing high level tickets [30], where each ticket represents a specific task, for example, "Implement Explore page". Before each development stage several tickets were taken, further refined and then implemented. Establishing a back-log of tickets allowed me to carefully plan development, and refinement allowed me to pragmatically think about how each specific task would be implemented. Figure 9 depicts the process of taking tickets from the backlog and fine tuning them before the development stage.

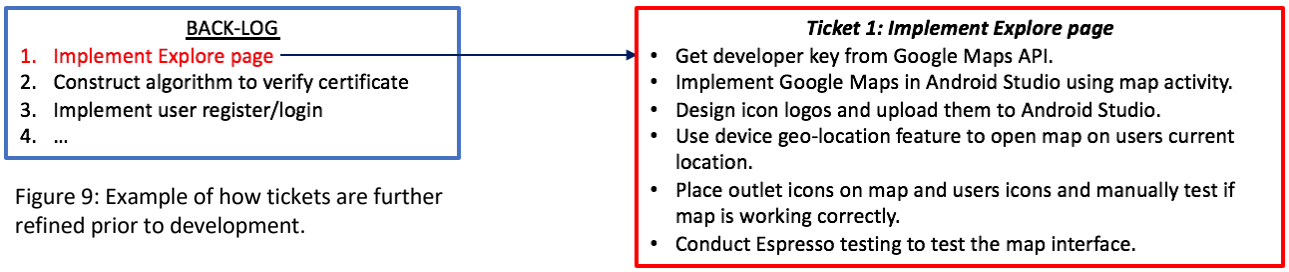


Figure 9: Example of how tickets are further refined prior to development.

Sprints are used in the Agile methodology to organise development. A sprint is a timeboxed period in which a set amount of work is to be accomplished [30]. I broke down the development of Halal Finder into ten two-week sprints; in each sprint 6-8 tickets were taken from the backlog and implemented. Decomposing development into small units allowed a manageable work load to be established, ensuring product quality.

6.2 Application Technologies

Halal Finder is a full-stack application meaning it consists of both a front-end and back-end. Two separate frameworks were used to create Halal Finder: Android Studio [32] and JHipster [33]. These two frameworks were chosen primarily because they utilise the Java programming language, which I am very familiar with.

6.2.1 Client Side

The client Halal Finder is a native Android application, thus it made sense to use Android Studio which is built specifically for constructing native Android applications. The framework offers a reliable UI (user interface) developer which provides a wide array of UI components, allowing the application to be fully customised. Furthermore, the framework includes a wide array of emulators which allow the application to be run on different Android devices, allowing for scalability. Halal Finder utilises various APIs such as the Google Maps API and Google Cloud Vision API. Android Studio offers various libraries which easily interact with these APIs and require little effort allowing for swift and hassle-free interaction.

The design patterns that govern best practices for Android is to execute and manage screens as Activities [34]. Typically, each screen offering dedicated functionality should be portioned into its own Activity. Halal Finder has five Activities these are: *LoginActivity*, *MainActivity*, *ExploreActivity*, *AddOutletActivity* and *RecomendationActivity*. The *MainActivity* in the application is the homepage which provides the user with the three options 'Explore', 'Add Outlet' and 'Recommendation'. Each Activity is loosely coupled from other activities thus, there is minimal dependencies between Activities, providing a cohesive user experience in Halal Finder. Each individual Activity needs to be managed, thus the application is required to react when the phone is idled, sleeping or when the application has been placed in the background, Activities pave the way to deal with all these situations and help preserve the battery life of the device. Appendix A shows a UML diagram which illustrates the main components that build the Android client application

6.2.2 Server Side

The JHipster framework was used to architect the back-end of Halal Finder and provides the Application Server [35]. Utilising this framework allowed me to incorporate user authentication, database access, transactions and administration rights. The framework incorporates the PostgreSQL database [36], allowing me to store user and outlet credentials securely. The databases were set up through constructing a hybrid UML design [37]. JHipster provides an end-to-end design generator which converts the UML design into database entities and maps their respective interactions for me to utilise. The database created contains three tables: User table, Outlet table and Review table. The User table contains information about users and consists of: unique user ID, username and hashed password. The Outlet table contains information about outlets and consists of: unique outlet ID, outlet name, address, type and HMC certificate image. The Review table contains the rating and reviews users have given outlets and consists of: user ID, outlet ID, rating and review.

One of the major components supported by JHipster is the Spring Boot framework [38], this is leveraged in Java providing inversion of control for large enterprise applications. Communication between the client and the Application Server is done using the REST (Representational state transfer) protocol, this is a style of software architecture of the World Wide Web which results in a maintainable and better performing software architecture [39]. The marshalling of data is done using JavaScript Object Notation (JSON) [40] since it is human readable and therefore, easy to use and understand. All client communication with REST endpoints are done asynchronously, so that the application does not freeze whilst the request is being served by the Application Server.

6.2.3 Version Control

Prior to coding it was important to set up a version control mechanism that would allow me to record changes to the code base. The benefit of using version control is that it allows easy identification of bugs that have been introduced into the code base by recalling previous versions of the file and examining what changes caused the bug. Furthermore, utilising a version control system allows me to keep copies of the entire code base, which would prove hugely beneficial in a case of file corruption or loss. The GitHub framework [41] was utilised to implement version control, since I have used it throughout university and my industrial year and thus am very familiar with its working. Several branches were made in order to keep track of changes. A master branch was used to keep a thoroughly tested version of the code base and several other branches were created which held individual functionality of the application. Once the individual functionality in each of the branches was completed and thoroughly tested, it was merged into the master branch. Using version control in this manner allowed development to be organised and clear from a developer's perspective.

6.2.4 Deployment

The JHipster environment provides ease of use when deploying the Application Server artefacts (i.e. binaries) to our cloud provider. Heroku [42] was chosen as the cloud solution for this project, however JHipster provides abstraction for deployment meaning we are agnostic to any vendor and can chose other providers in the future. This may be required based on the growth of our solution; Amazon Web Services [43] or Microsoft Azure Cloud Services [44] could be future possibilities for deployment. JHipster provides a pipeline-based deployment to the cloud, it provisions the code, executes tests and then updates the artefacts on the cloud. Delta database changes are updated accordingly using Liquibase [45]. Deploying the Android client application was straightforward and required a signed .apk file that is then deployed on the Google Play platform [46], where users can download the application.

6.3 Development of Functionality

The following sub-sections explain how individual functionality was achieved this includes: logging-in/registering, exploring outlets, adding outlets and creating a recommendation system that is able to recommend eateries.

6.3.1 Login/Register

When first opening the application, the user is required to register their credentials or login with registered credentials, which will allow them to use application functionality. The Android client has a *LoginActivity* which houses this capability. To register a user a *register_user(…)* function was created, the function is executed upon the user filling in their email and username and clicking the "Register" button. The *register_user(…)* function takes the email and username string and places this in a request alongside a JWT token provided by the Android client. A JWT token is a JSON web token which provides a secure way of representing information between two parties, in this case the client and the Application Server [47]. The request is sent to the Application server *Register(…)* REST endpoint, the user management layer of the Application Server first authenticates whether a valid client is requesting to add a user using the JWT token provided. If the JWT token is correct, the Application Server makes a request to the end user via email, confirming acknowledgement of their registration request. The email contains a link through which the user is directed to the JHipster website to set their password, utilising the angular aspect of the JHipster framework. Once the user has entered their password, the password is hashed and stored with the username in the User table in the database.

To login a user, a *login_user(…)* function was created, the function is executed upon the user entering their username and password and clicking the "login" button. The *login_user(..)* function takes the username and password string and places them in a request, which is then sent to the authentication layer of the Application Server using the *Login(…)* REST endpoint. The password string is hashed and if it matches the hashed password stored for the given username in the database a JWT token is issued to the user. The JWT token allows the user to login and conduct further transactions in the session. Figure 10 depicts a high-level sequence of interaction between the user, Android client and Application Server when registering and logging in.

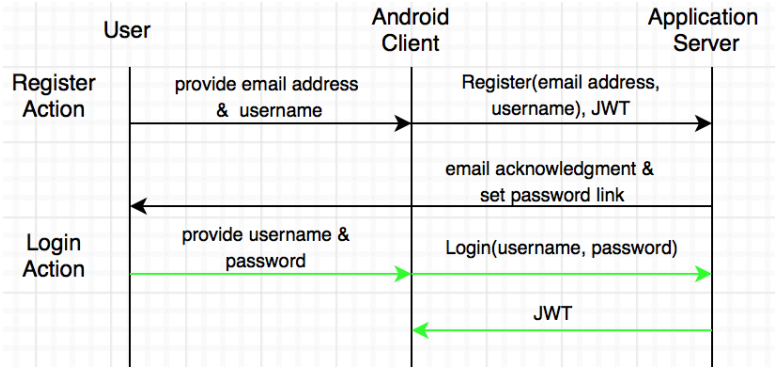


Figure 10: Diagram showing high-level sequence of interaction between user, Android Client and Application server during registration and login. Black arrows represent the process of registering and green arrows represent process of logging in.

6.3.2 Explore

When the user clicks on the "Explore" button from the *MainActivity* the *ExploreActivity* is run. The *ExploreActivity* was created using the default *MapActivity* template provided by Android Studio. To utilise the provided map I needed a Google Maps API token, this token was retrieved by creating an account with the Google Maps API. I then utilised my token, providing me with a fully functioning map interface.

The next task was to find the user's current position, so that this can be displayed on the map. Android Studio provides different levels of fidelity of positional data, which are mainly driven on battery consumption. In order to find the user's location quickly and without using significant computational resources, it was decided that as soon as the application receives an accurate enough positional lock of the user, that geolocation will be used to approximate the user's position. Although it may not be the user's exact location, the position pinned by the application is accurate enough based on the data received from the user's device. To retrieve the device's location a declaration of permission had to be established so that the user acknowledges his location is being used by the application. This was achieved by implementing the following configuration in the manifest file:
"<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>".

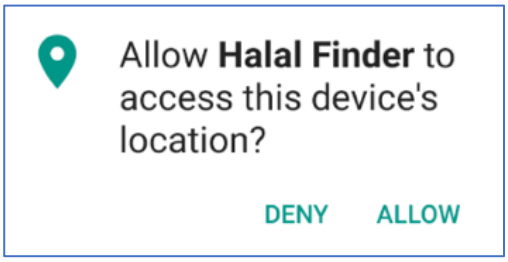


Figure 11 - Output message displayed when first opening 'Explore Page'.

Next an instance of the *LocationManager* class is created which provides the necessary functionality of using location services in Android and allows us to retrieve the device's location. A *LocationListener* is implemented so that updates in the device's location can be received. Once the user's position has been pinned on the map, the next step was to pin the neighbouring outlets on the map. Initially, all outlets in the database were displayed on the map, however it was observed that by doing this, the map interface would become unresponsive for 2-3 seconds whilst all the outlets were being added. This was problematic as it did not offer users a fluid experience whilst navigating around the map. After conducting research about effective ways to populate a map using geospatial queries [48], it was decided that not all outlets in the database would be immediately placed on the map and this is for two key reasons: firstly, because it is computationally expensive in terms of the number and size of requests that have to be made from the client; the underlying reason for the map interface freezing. Secondly, because many outlets displayed will be redundant as the user will not be able to view all outlets immediately having been zoomed in to their current location.

The explore functionality was reconstructed so that only local outlets around the user are shown, this was achieved by creating an invisible bounding box on the visible map. The bounding box is expanded to an appropriate ratio to give a geo-rectangle, the co-ordinates of this rectangle are then sent to the Application Server using REST endpoints. A geospatial query is then run on the Outlets table which returns those outlets intersecting the rectangle defined. As the user navigates around the map, the co-ordinates of the geo-rectangle change, thus respective geospatial queries are made to the database and more outlets are displayed on the map. The result of using a bounding box and making queries as when needed, provides a more fluid UI and computationally effective process. The final step before populating outlets on the map is to process each outlet by its "Type" field, as this is used to assign a different pin type to the outlet on the map.

When the user finds an outlet they may be interested in they can click on its pin for more details. As mentioned in the design phase, I wanted to provide the user with outlet information in an intuitive and simple fashion, with precise and unambiguous detail. To achieve this, a custom pop-up window layered on top of the map is displayed which is populated with the following outlet information: name, address, type, HMC certificate image, overall rating and individual customer ratings and reviews. The outlet's HMC certificate can

be viewed by pressing the 'information' button located on the window. Figure 12 shows the outlet information window.

Information for the outlet is retrieved from the database using two separate REST calls. The first call is made when the *ExploreActivity* is first run and the local outlets are retrieved to be displayed on the map; in this call basic outlet information is returned from the Application Server such as outlet name, address and type. The second REST call is made when the user double clicks on the outlet pin for more information, this call returns the outlet's customer ratings and reviews and the HMC certificate image. Using two separate REST calls to retrieve outlet information is an efficient way of minimising data bandwidth since additional outlet information is only being retrieved when needed. The average outlet rating is calculated by conducting a mean calculation on all the individual customer ratings and is displayed in the pop-up window.

One challenge I did face when populating the outlet information pop-up window was displaying the certificate image so that it could be readable. The certificate image was retrieved from the database in the JSON response as a Base64 string, to convert the Base64 string to an image I had to first convert it into an array of bytes and then convert the byte array to a bitmap. The bitmap was then modified so that the image could be human readable.

On the outlet information pop-up window the user is given the option to rate and write a review by pressing the "Add Review" button, this launches a pop-up window as shown in Figure 13 containing a text field for users to enter their review and give their rating in the form of stars. The Android RatingBar library [49] was used to construct the rating bar of stars, the library gave me flexibility in terms of choosing the total number of stars and incremental step size. Having entered content in the text fields and given a rating, users can press the "Submit Review" button which executes the in-built getNumStars() to convert the user's star rating in to a floating value. The floating value along with the text in the text field, the user ID and outlet ID is then sent as a JSON object to the Application Server which stores the rating and review for the appropriate outlet in the Review table. Once the request is sent the window is closed and the user is returned to the outlet information pop-up window.

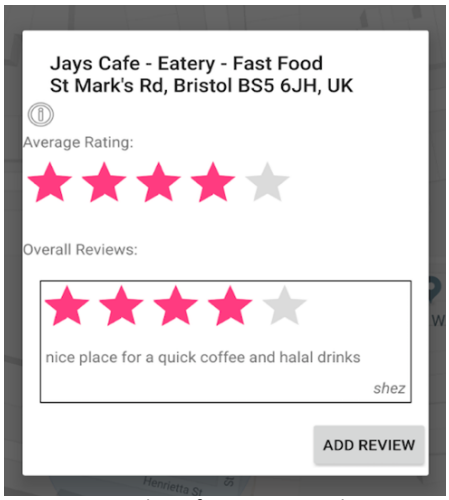


Figure 12- Outlet information window.

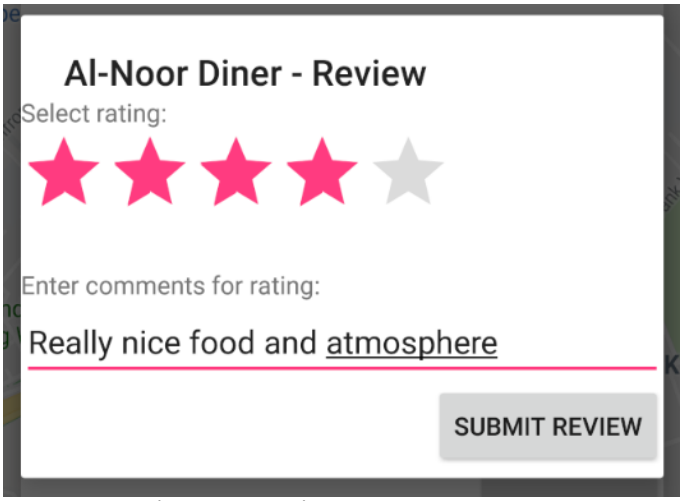


Figure 13 – Outlet review window.

6.3.3 Add Outlet

The *AddOutletActivity* is primarily focussed on form entry and uses three key technologies to add a genuine outlet to the application, these are: Android Geocoder API [50], Google Cloud Vision API [24] and a verification algorithm. As mentioned in section 5.5 the decision was made to use the user's geolocation to submit the location of the outlet rather than having the user manually fill the address of the outlet as this requires effort from the user. To automatically fill in the outlet address the Android Geocoder API was used. Upon clicking the 'Submit Location' button on the 'Add Outlet' page the geolocation position of the device is sent in a request to the Geocoder API for processing. The API converts the geolocation position made up of longitude and latitude to either a single address or an array of possible addresses in the vicinity. This data is then sent back to the application and displayed. If one address was returned the address form is filled with

that specific address, if multiple addresses are returned a dropdown is used to display the possible addresses and the user is required to select one.

One of the key requirements when adding an outlet is to upload an image of the HMC certificate. To allow the application to have access to the device's camera a declaration of permission had to be established, so the user acknowledges the camera is being used by Halal Finder. This was achieved by placing the following configuration in the manifest file:

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
```

Once permission to use the camera has been achieved, the device's camera can now be used to take images. The next step was to implement the Google Cloud Vison API, to conduct optical character recognition of the certificate image. An account was first made with the Google Cloud Vision service and a token was received and implemented in Android Studio. When the outlet form has been completed and the image certificate has been taken the user is now able to press the 'Submit Details' button. Upon pressing the 'Submit Details' button, the certificate image is encoded into a Base64 string. This was achieved by first converting the image into a bitmap, which is then compressed into a ByteArrayOutputStream. The ByteArrayOutputStream is then converted in to an array of bytes which is then subsequently converted in to a Base64 string. The Base64 string is encapsulated inside a JSON object this is then placed in a request along with the Google Cloud Vision API token. The request is sent from the client to the Application Server. The Application Server forwards the request to the Google Cloud Vision server which uses OCR processing to decode the certificate image in to text and returns the result in a JSON object. I then tokenise the JSON object received and extract the text. The text received from conducting OCR is converted into a string and then fed in to a verification algorithm, whose purpose it is to ensure the certificate image is genuine by conducting several checks.

Developing the verification algorithm was a crucial part of adding a genuine outlet. To do this the design of the HMC certificate had to be studied to see what elements of the certificate could be used to verify it was a genuine HMC authorised certificate. Once I had established what parts of the certificate I wanted to extract, I started developing the verification algorithm. The algorithm takes in a string containing the extracted text generated by Google Cloud Vision API and conducts 3 asserts statements to check the existence of certain phrases and one calculation. Figure 14 shows the certificate and highlighted are the parts extracted. The bullet points below describe the implementation of each check:

- Check 1: This check is used to ensure the issuing body of the certificate is the Halal Monitoring Committee, this is very crucial as the application should only add outlets which are HMC authorised. The check is done by parsing over the input string and asserting the existence of the sub-string "HALAL MONITORING COMMITTEE UK".

- Check 2: All HMC certificates contain the phrase "Has attained the HMC Approval for selling HMC approved products", thus it made sense to verify the existence of this phrase. The check is done by passing over the input string and asserting the existence of the sub-string "Has attained the HMC Approval for selling HMC approved products".

- Check 3: All HMC certificates contain the email address of the HMC body, thus it was decided that this was another criteria the verification algorithm could check for. The check is done by parsing over the input string and asserting the existence of the sub-string "info@halalhmc.org". It is important to note that the phone number was not used as a criterion to validate the certificate as phone numbers are prone to changing, whereas organisational emails are very unlikely to change.

- Check 4: All HMC certificates include an expiry date thus, it is crucial to ensure that all outlets in the application do not contain expired certificates, making them genuinely Halal. Development of this check was more complicated than the other three check. To extract the date posted on the certificate I had to ensure their was a consistent date format used across all certificates. An HMC representative was contacted and confirmed that dates are always inserted in the following format 'DD.MM.YYYY'. With this

information, the expiration date is extracted as follows, the sub-string “Expiry of validation:” is searched for in the input string, once found the following 10 characters are extracted, in the example of the certificate provided this would give me the string “09.11.2019”. From the user’s device I then get the current date using the *Date* Class [51]. Finally, a comparison is done to see if the expiration date on the certificate is greater than the current date, if the expiration date is greater the check will pass otherwise it will fail. Appendix B shows the code used to implement the conditional logic for validating a certificate.

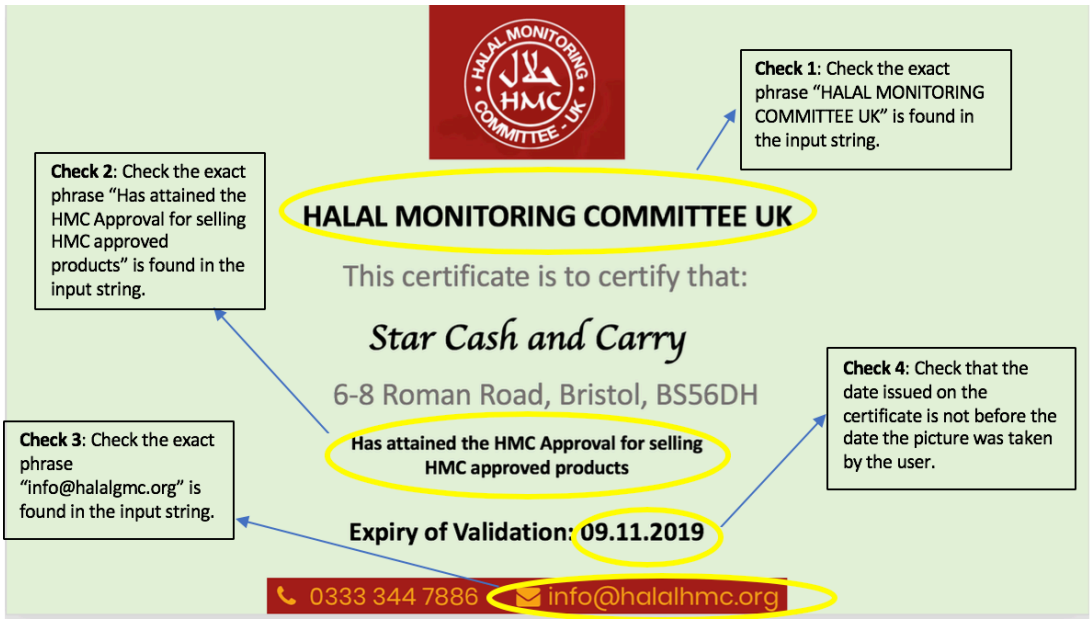


Figure 14: Image of the HMC certificate for ‘Star Cash and Carry’ annotated with checks the verification algorithm conducts.

If any of the four checks conducted by the verification algorithm fail the outlet will not be added and the application will output the message “Outlet cannot be added”. If all the checks have passed then a request containing the outlet’s name, address, type, certificate image and expiry date is sent to the Application Server which then stores this information in the Outlet table in the database. It is important to note that we store the outlet’s certificate expiration date so that when the certificate becomes expired the outlet is automatically removed from the application. This ensures that at any given time, Halal Finder always displays genuine Halal outlets.

6.3.4 Recommendation System

To generate eatery recommendations for the logged in user also referred to as the target user, the first step was to extract all user and eatery data from the database. To do this a single REST endpoint ‘*recommend(…)*’ call is made to the Application Server. The data retrieved consists of all the users in the User table, all the outlets in the Outlet table and a mapping of users and their allocated ratings to outlets gained from the Review table.

We then iterate through each user by their unique ID and check what ratings they have given outlets. For each user a list of tuples is constructed containing the outlet IDs visited and the rating given. We cross-match the outlet ID with the ID of eateries in the database. If the rated outlet is an eatery it remains in the tuple and if it is not an eatery it is removed from the tuple. The results of this is a list of tuples for each user containing the IDs of the eateries visited and the respective rating given. We then construct a vector of size *X* for each user, where *X* represents the total number of eateries in the database. Each index in the vector relates to a specific eatery this is done in relation to ordering the eatery IDs, for example the eatery with the smallest eatery ID number will be at index 0 of the vector. We the iterate over each element of the list of tuples for the user placing the rating of the eatery in to the corresponding position of the vector. This vector is known as the ratings vector and is created for each user in the application. Note if a user has not rated an eatery a default rating of three is given to the eatery to ensure that there are no NULL values in the vector.

Once all users have had their eatery ratings placed in to a vector, we need to compute the similarity between the target user and all other users. To do this the Euclidean distance is calculated between each user's rating vector and the target user's rating vector. Figure 15 shows the code to calculate the Euclidean distance. The values obtained from calculating the Euclidean distances are placed in a tuple along with the respective user ID. These tuples are then sorted in terms of the smallest distance. The top three users with the smallest Euclidean distances and therefore the most similar behaviour to the target user are then chosen as the closest 'neighbours'. We then look at the rating vector for each of each of three closest users and transfer the ID of any eatery rated three or above to an array which will contain the recommended eateries. Any duplicate IDs are then removed from the array as we do not want to recommend the same eatery more than once to the user. Next, we remove the IDs of the eateries the target user has visited from the array, this is to ensure that recommendations made to the target user are eateries that he has not visited before. Finally, the array is ordered so that the eateries from the user with the smallest Euclidean distance are first.

```
private long euclideanDistance(List<RatingItem> userA, List<RatingItem> userB) {

    long total = 0;

    for (int i = 0; i < userA.size(); i++) {

        long diff = userA.get(i).review - userB.get(i).review;

        diff = diff * diff;
        total += diff;

    }

    return Math.round(Math.sqrt(total));
}
```

Figure 15: Code for calculating Euclidean Distance.

The array containing the IDs of the recommended eateries is then iterated over and REST calls are made to the Application Server to get the information of those eateries from the Outlet table in the database. The user is then directed to the 'Explore' page where the recommended eateries are shown along with the user's current position.

7.Testing

Testing is a crucial part of development as it ensures the application functions correctly and efficiently. Agile development was chosen for this project primarily because it allows the system to frequently undergo various testing procedures throughout development, allowing problems to be identified early on during development, rather than in later stages when it becomes harder to locate and fix problems. Halal Finder is a consumer-oriented application therefore it was vital to test the application's user interface thoroughly.

7.1 UI Testing

Following the study of current solutions in section 1.4 and market research in section 1.5, it was clear that a simple and easy-to-use UI would provide a good user experience. To ensure this was achieved, two main technologies were used: Pencil and Espresso.

7.1.1 Pencil

Pencil is a prototyping tool used to create wireframes and UI prototypes [52]. Once I had gaged what my target audience sought from the user interface, I constructed sketches of how the application features could be visually implemented, these were then developed in to wireframes. Conducting early stage UI development enabled me to showcase my wireframes to potential users and gather valuable feedback, after which I significantly refined the user interaction with the application. Combining both the wireframes and the user feedback, I was able to develop high precision UI mockups utilising Pencil. Figure 16 shows the wireframe developed for the 'Add Outlet' window, Figure 17 shows the high precision UI mock-up of the 'Add Outlet' window developed from the wireframe and user feedback. The UI mock-ups were examined by a different set of users to gain their feedback and if necessary appropriate adjustments were made. Allowing

users to be heavily involved in the UI design has ultimately resulted in a better user experience of Halal Finder since its design is heavily related to the needs of the user.

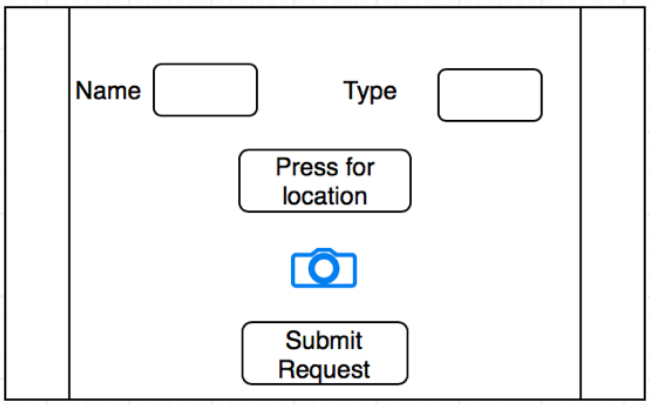


Figure 16- Initial wireframe developed for 'Add Outlet' window.

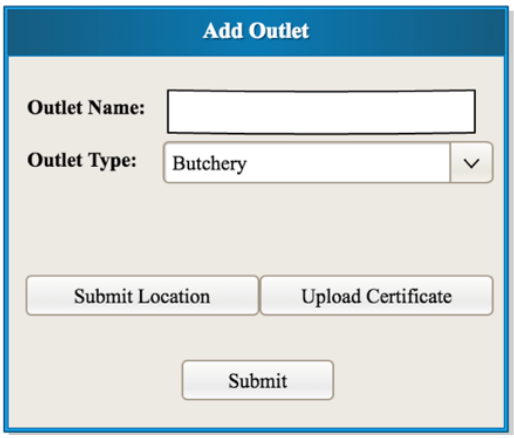


Figure 17- High precision UI mock-up of 'Add Outlet' window.

7.1.2 Espresso

Espresso is a testing framework provided in Android Studio, designed to provide a fluent API for writing concise and reliable UI tests based on user actions [53]. The framework operates by recording interactions with the UI and converting them into executable code, which can then be run frequently throughout development to ensure the UI is functioning correctly. Espresso tests allow you to examine each individual component of the UI, it achieves this through assertion statements that check whether the UI mirrors the anticipated behavior and state after user interaction is conducted. Using Espresso was extremely useful during the development of Halal Finder as it allowed me to test the UI with the possible paths users could take when using the application. The following sequence is one possible path a user of Halal Finder could take:

Figure 18: Example of an application path in Halal Finder.

Using Espresso, the path described in Figure 18 was tested as follows: At stage 1 the user is located at the login/register window where four assertion statements are placed to verify the existence of the two text boxes which accept the username and password and the two buttons for logging in and registering. Once the user has logged in they will select the 'Explore' button on the application homepage, at which point an assertion is run to check the existence of the map interface and an interaction is conducted to check the user can zoom in and out of the map. In stage 3 the user clicks on an outlet on the map; this will run an assertion to check the existence of a text box on the screen, to test that the clicked outlet's name is being displayed.

In stage 4, the user double clicks on an outlet, the expected response from the application being to display a pop-up window describing the outlet. To test the application responds in the correct way, an assertion is run to check the presence of a pop-up window and several text boxes which hold information about the outlet. Espresso proved extremely useful in breaking down the paths users of Halal Finder would take in the application and testing specific UI elements of each of the windows visited in the path.

7.2 Functionality Testing

Functionality testing was frequently conducted throughout the development of Halal Finder to ensure the system functions according to the functional requirements of the project. Four widely established levels of software testing were implemented to test the application rigorously and consistently; Unit testing, Integration testing, System testing and User Acceptance testing.

7.2.1 Unit Testing

Unit testing examines individual components of the system to ensure they are functioning as required. In my code base, each function is typically coupled with a unit test. Since the application was built using Android Studio, the built-in JUnit test framework was used to create unit tests and evaluate the coverage of the code being tested. Unit tests are small tests which examine small parts of the program, meaning a large number of unit tests can be ran very quickly, allowing the application to be tested frequently throughout development. Performing unit testing was very useful during development as it allowed me to implement new features in the application and then run previously written tests to ensure other parts of the system continued to function correctly, resulting in a fast pace of development. All unit tests that were written passed and an overall code coverage rate of 71.6% was achieved, this value shows that a 'Good' level of testing has been conducted according to industrial practices [54].

Conducting unit testing on the recommendation system was particularly important as it allowed me to examine whether the recommendation algorithm was correctly extracting similarities between users to generate logical recommendations. The unit tested consisted on adding three new users to the application A, B, and C. User A had visited five eateries and both B and C had visited 7 eateries. All the eateries user A had visited user B had visited and both users gave the identical ratings, showing a strong similarity in their behaviour. User C had also visited the same eateries as User A but gave very different ratings to the eateries. The unit test ran the scenario where user A requests a recommendation. The desired output of the unit test was the highest rated eatery of user B that user A had not visited, the unit test generated this output, showing the recommendation system works correctly.

7.2.2 Integration Testing

Integration testing is a form of testing where individual components of the program are joined together and tested as a group. Halal Finder can be broken down into several components, thus integration testing was utilised to ensure different elements of the application worked together without conflict or errors. For example, during the development phase I first built the 'Explore' window and tested all of its functionality with unit tests. Subsequently, I developed the 'Add Outlet' window which adds an outlet on the map located on the 'Explore' window and also used unit tests to examine the self-contained functionality of the 'Add Outlet' window. Integration testing was then utilised to test the interaction between both the 'Explore' and 'Add Outlet' windows in order to ensure the outlet was successfully added on the map.

The main advantage of integration testing is that it allows faults in the interaction of separate components to be easily identified. During development, one integration test that failed was the interaction between the REST call which retrieves the text of the certificate image from the Google Cloud Vision API and the verification algorithm which then processes the text. This particular test is technically known as a 'flaky' test because, in some cases the verification algorithm would correctly receive all the data and go on to process it, but sometimes the algorithm would receive a NULL value and fail. The root cause of the issue lay with the Google Cloud Vision API not completing the response correctly, as the application did not wait for the correct amount of time before executing the verification algorithm. The test having revealed this underlying issue in the pipeline of calls to the verification algorithm, allowed the resolution by utilising a semaphore that signaled the end of the processing from the Vision API and then commencing on to the verification algorithm. Overall, all integration tests were passed with a 100% success rate. Having combined individual components and testing their interactions the next step was to test the system as a whole unit.

7.2.3 System Testing

System testing tests a whole integrated system to verify the system functions in accordance to the functional requirements. To conduct system testing it was decided the built-in Android Studio emulator would not be used to mimic an Android smart phone as it is not how users would run the application. Instead, a smartphone running Android OS was purchased and a developer's version of the application was installed on the phone. System testing was then conducted by imitating a user's behaviour as if they were navigating through the application, utilising the various features as set out in the specification in section 3. To do this, several possible user flows were noted and then tested. Figure 19 shows an example of one user flow.

Figure 14: User flow describing a specific path a user could take in the application.

The user flow described in Figure 19 requires the user to: log in to the application, visit the 'Add Outlet' window, add an outlet, view the outlet on the map and submit a review for it. For each user flow that was created, a form was produced to assess whether the functionality of individual components worked correctly. The form for the user flow described in Figure 19 is shown in Table 5. All 17 application flows that were tested passed, demonstrating that the application can conduct functionality successfully and bug-free. System testing was performed by myself and although results did prove the application met the specification's requirements, it was important to get real users to test the application.

Functionality	**YES/NO**
Successful login if credentials are correct	YES
OCR working correctly	YES
Algorithm correctly processing extracted certificate text	YES
Outlet correctly being displayed on map in correct location	YES
Review and rating correctly being assigned to outlet	YES

Table 5: Checks for the user flow described in Figure 14.

7.2.4 User Acceptance Testing

User Acceptance Testing (UAT) is conducted by real users of the product who test the software, allowing any bugs or criticisms to be identified and resolved prior to application deployment. UAT for Halal Finder provided an insight into how users would utilise the application as well as their feedback on what worked well and aspects that required improvement, this in turn allowed crucial changes to be made prior to release. Two important elements had to be established before conducting UAT; where to source test users and how to deploy the application to these users. The decision was made to release an alpha version of Halal Finder for test users to run on their phones. Releasing an alpha version meant that I would be able to gage how Halal Finder operates on different variants of real Android smartphones. Furthermore, this closed release meant I could select for which users Halal Finder becomes downloadable for from the Google Play store, preventing the application from being available to the general public.

It was decided that UAT would be conducted by members of the University of Bristol's Islamic society because they were based locally and could be easily be contacted. I arranged a meeting with members of the society and chose 10 candidates who possessed different types of Android smartphones, ensuring testing would be done on different types of hardware. All candidates were between the ages of 18-24 since my market research in section 1.5 established this age group is within my main target audience. Having identified my test sample, I sent an invitation link to each candidate via the Google Play store allowing them to download the application. The test user group were given a brief covering activities to carry out and a scoring system to rate these activities; a score of 0-10 was to be given to each activity with 0 indicating the task was difficult to perform and 10 indicating that the task was easy and logical to conduct. Users were instructed to complete and score the following tasks: register credentials, login to the application, explore outlets using the 'Explore' page, add an outlet to the application, write a review for the added outlet and use the application to generate an eatery recommendation for themselves. The test user group were given one week to complete testing of the application. The bar chart in Figure 20 shows the average score of each task received from the test candidates.

Overall, the results gained were positive with areas for improvements highlighted. Tasks such as registration, logging in and reviewing/rating outlets were easy to conduct for users and therefore considered successful, whereas tasks such as exploring outlets, adding outlets and finding a recommendation were less successful, with users finding these tasks more difficult. Having reviewed the results, I arranged a meeting with candidates to gain a better understanding of what they found problematic when exploring Halal outlets,

adding outlets and generating recommendations. The problems identified and the respective solutions implemented are discussed below.

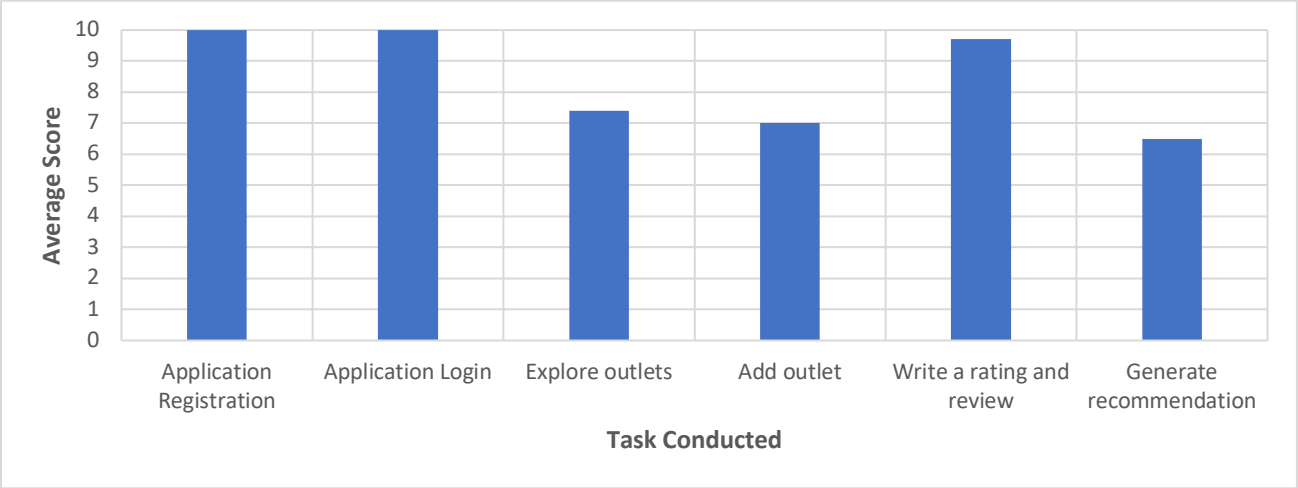


Figure 20: Shows the average score of each task received from the test candidates

7.2.4.1 Explore Outlets

The task of exploring outlets gained an average score of 7.4 from all 10 candidates, the modal score being 7. Through discussion with the user group it was discovered that all 10 users really enjoyed exploring outlets using the map interface and found the use of distinct icons to separate the type of entities "really useful". However, they all shared the same common problem, multiple users had added the same outlet to the application, causing an overlap in outlet icons on the map and multiple information windows opening when clicking on a duplicated outlet icon. This issue was immediately resolved by implementing a check before an outlet is added to determine whether it exists in the same location, if the outlet does previously exist it will not be added.

7.2.4.2 Add Outlet

The task of adding an outlet gained an average score of 7 from all 10 candidates with the modal score being 7. Through discussion with the group it became apparent there was a mis-communication between the application and user. All users were under the impression the purpose of uploading a certificate image was for it to be displayed in the information box for the outlet, not for outlet validation. Thus, they thought the application contained a flaw when certain outlets were rejected from the application. I explained to them that the certificate image is used to check whether the Halal outlet is genuine and that this is done via using OCR and a verification algorithm. All users were very surprised as they were not aware that the application housed this capability and suggested that the outlet should state what it is doing with the image certificate. I have altered the 'Add Outlet' window so that as soon as the 'Submit Details' button is pressed a message box is shown stating 'Certificate image is now being processed to verify the Outlet'.

7.2.4.3 Generate Recommendation

When receiving the results for the 'Generate recommendation' activity I was quite surprised as it gained the lowest average score of 6.5 with a modal score of 5. When speaking to users it became apparent that all users strongly liked the eatery being recommended and would consider it, however they disliked the mechanism by which the application made recommendations. As explained in section 5.6, to find a recommendation the user is required to press the 'Recommend' button, this action will run the recommendation algorithm and the recommended eateries will be displayed in the 'Explore' window. All test group users agreed that the eateries that were displayed should be ordered in terms of what the system believes is the best match, furthermore they believed a list should be used to display the recommended eateries as it immediately shows the outlet information rather than the map interface which shows an icon.

Several ideas were debated for the design of the recommendation interface such as using the map but changing the icons to numbered labels which represent the order of the recommended eateries. Another

suggestion made was to use a simple list which displays the recommended eateries in order of the highest recommended. Following discussion with the group, we came to the conclusion to reconstruct the recommendation interface as follows: when the user presses the 'Recommend' button, a horizontally-split window will appear. The upper half of the window will display the interactive map from the 'Explore' window, showing the recommended eateries as tags, and the lower half of the window will display the recommended eateries in a list ordered by the highest recommended. Clicking on a different eatery in the list will re-format the map to show the location of this eatery. This new mechanism accommodates the user needs discussed with test user group, resulting in a more successful feature. Figure 16 shows the new split-screen recommendation window.

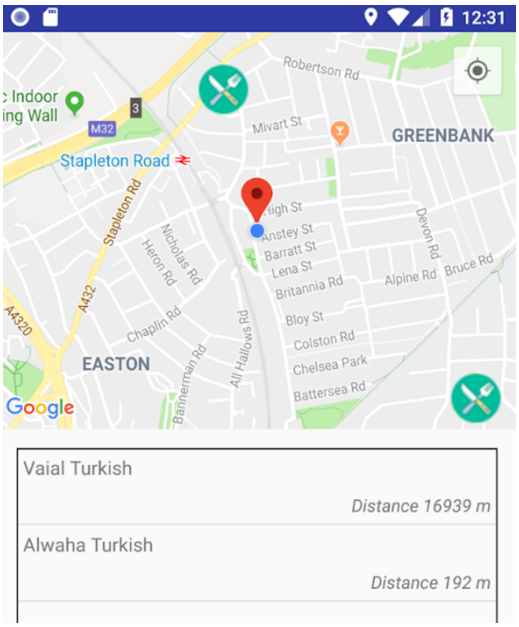


Figure 21: Refined interface for 'Recommendation' window.

8. Description and evaluation of the final product

Having completed development of Halal Finder, it was essential to review the final product. Evaluation was achieved through a combination of self-reflection, user feedback throughout the project, and post-market research. As with any form of development, internal and external factors may cause the final product to slightly differ from the original speciation, therefore it is important to note any deviations from the initial specification and explain why they were made.

Halal Finder has achieved all the requirements laid out in the project's specification. Taking on user-feedback throughout the design and development phase and during user acceptance testing, has allowed the end product to be refined to provide users with the best experience whilst achieving the ultimate goal of locating genuine Halal outlets. The application has been broken down into its key components to help describe the final product and my assessment of each part is given.

8.1 Usability

A user-centered approach was taken to design the application's UI, whereby each window in the application is tailored to the needs of the user. This has meant that users are clearly able to navigate around the application and can easily understand the functionality each window of the application is trying to achieve. When first logging in to the application the application homepage provides users with three clear options: 'Explore', 'Add Outlet' and 'Recommend'. By providing three distinct options each with their own individual window meant that users can quickly navigate from one activity to another. Leading on from the homepage each of the three activities have been carefully designed and developed around user feedback, resulting in a more user-focussed application. As a result of carefully designing and developing the application around user-feedback criteria 4.4.A of the evaluation criteria has been achieved. This can also be highlighted by the

results of the UAT whereby all individual components of the application gained an average score of 8.4/10, demonstrating that users were strongly satisfied with the usability of the application.

8.2 Registration/Login

On starting the application, the user is presented with a simple login form to sign in to the application using their username and password. If the credentials provided are correct the user is able to successfully login to the application to explore, add and rate outlets and generate eatery recommendations for themselves. A registration option is provided to new users, this allows users to register their details with Halal Finder to start using functionality. To register, a user is required to provide their email address and username. The Application Server then sends an email to the user, containing a link to set their password. Once the user password has been provided the user's details are stored in the application database and the user is able to successfully login. UAT has shown that the activity of registering to the application and logging-in is highly successful with 100% of users rating these two activities 10/10.

8.3 Locating Halal outlets

On choosing the 'Explore' option, the *ExploreActivity* is run and an interactive map is displayed to the user. The user's position is marked on the map and surrounding outlets are plotted each with a distinct icon depending on the type of outlet; grocery, butchery or eatery. The user can interact with the map by zooming in and out and swiping on the map. A button is located on the page to re-position the user to their current location to avoid users manually navigating back to their location once they have finished exploring. When tapping on an outlet icon once, the outlet's name appears and when tapping on the icon twice, a pop-up window populated with outlet information appears. This window is layered on top of the map and includes: outlet name, address, type, HMC certificate image, overall rating and individual reviews and ratings. The outlet window is organised in a manner such that all information is displayed clearly to the user for e.g. the certificate image is scaled to a readable size. A 'Submit Review' button is located at the bottom of the outlet window, when selected, the user is presented with a rating bar containing stars and a text box to write an optional review about the outlet. When trialling the explore functionality with users in the UAT phase all candidates enjoyed the process of locating outlets. The feedback obtained illustrated that users were able to navigate around the map easily, their position was located correctly and outlet data was displayed to them clearly. Thus, we can conclude that criteria 4.1 and all of its sub-sections have been achieved to a high standard.

8.4 Adding outlets

Having selected the 'Add Outlet' button, the *AddOutletActivity* is run and a form is displayed to the user, containing compulsory fields the user is required to fill in about the outlet such as name, address, outlet type and a photo of the HMC certificate. To submit the address of the outlet, the user is required to press the 'Submit location' button, setting their current position as the outlet address. This is because the user is required to also submit the outlet's HMC certificate, located inside the outlet thus the assumption is made that when the user is adding a new outlet they are located at the outlet.

A button with a camera logo is placed on the 'Add Outlet' window. Pressing this button opens the device's back camera, allowing users to upload a certificate image. Once the picture has been uploaded, two key technologies are utilised to ensure the certificate is genuine and the outlet is Halal; Google Vision API and a verification algorithm. The photo is first sent to the Google Cloud Vision API where OCR is carried out to read the text from the certificate. The extracted text is then passed to the verification algorithm which conducts several checks on the certificate e.g. checking the authorising body is HMC and the date has not expired. If all the checks are passed then the outlet is successfully added to Halal Finder, if one or more of the checks fail, the outlet is rejected. Conducting system testing on the application during which I submitted real and fake HMC certificates to the application has verified that the application successfully differentiates between genuine and non-genuine Halal outlets and adds or rejects them accordingly. Ten example of genuine and fake certificates were provided to Halal Finder. All ten genuine outlets were successfully added first time and all ten fake certificates were successfully rejected first time, showing the application has a high rate of correctly classifying outlets. Halal Finder successfully achieves criterion 4.2 as the application is able to utilise

the phone camera, the OCR system correctly extracts text and the verification algorithm is able to add/reject genuine/fake outlets respectively.

8.5 Eatery Recommendations

Having pressed the 'Recommend' button on the homepage, the *RecommendationActivity* is run which executes the implemented recommendation algorithm. The recommendation algorithm is based on a collaborative filtering system; if two users have a similar history of rating eateries then it is predicted that they will behave similarly in the future. Concepts from the K-nearest neighbour algorithm were used to implement a system which could make recommendations to a target user. Each user from the set of users is assigned a vector which contains their ratings from 0-5 for each eatery in the application. The Euclidean distance is calculated between the target user's rating vector and every other user's rating vector to determine the similarity measure between users. A list of users is then created ordered by smallest distance to the target user. The rating vectors for the three users which have the smallest distance are processed to generate recommendations. The algorithm recommends any eatery in each of the three vectors that is rated three or above and has not been previously visited by the target user. Due to the time constraints of the project I did not have time to fully encapsulate all the principles of the K-nearest neighbour algorithm into my recommendation system but did extract the main concept of understanding the similarity between users in order to generate appropriate recommendations. The design of the recommendation window is split horizontally, with the top half displaying the map interface with the recommended eateries and the bottom half showing a list of the recommended eateries with additional information; the eatery's name and distance from the user. This gives users the option to scroll down the list or navigate around the map to find an eatery. The list and map are both in sync, thus, if the user clicks on a specific eatery in the list then the map interface focuses on that eatery pin.

Having manually tested the recommendation algorithm through unit testing it has been shown that the system functions successfully and is able to correctly calculate similarity between users and from this filter the appropriate eateries to recommend to the target user. This is further supported by UAT where it has been demonstrated that the recommendations generated were seen as useful by all users during testing thus, confirming the algorithm can appropriately find similar behaving users from the user set. It is clear that criterion 4.3 has been met since an algorithm which uses collaborative filtering principles has been implemented and the majority of users during testing were highly satisfied by the eatery recommendation made to them.

8.6 Post-launch research

After successfully deploying the application publicly on the Google Play store, post-launch research was conducted. Post-launch research was crucial as it gave users the opportunity to provide a final verdict on the application, allowing the final application to benchmarked with user needs.

To conduct this research, the forty members of Easton Jamia Masjid involved in the initial market research were contacted again for their participation. Using members from the initial market research was important since their initial feedback helped direct the way in which the application was designed, thus they would be able to make an accurate assessment of functionality. The final sample size for post-launch research was 30, having subtracted members who did not want to participate in market research for a second time and those who did not own an Android smartphone. Emails were sent to all 30 members with a link to Halal Finder and a feedback form, which was designed to cover the evaluation criteria in section 4. The form was made up of the following five components: locating Halal outlets, adding outlets, eatery recommendations, general application usability and application performance. Members were told to utilise the application freely for two weeks with no other specific instructions, encouraging them to use the application as naturally as possible. The feedback form asked users to give a ranking of 'Poor', 'Satisfactory', 'Good' or 'Excellent' to each of the five components and comment on their opinion of the application.

8.6.1 Results achieved

The results obtained are shown in Table 6 and were extremely positive overall; a rating of 'Excellent' dominated all five sections demonstrating that Halal Finder was well liked by users.

	Percentage of Candidates % (Total candidates: 30)			
Criteria	*Poor*	*Satisfactory*	*Good*	*Excellent*
Locating Halal outlets	0	0	10	90
Adding Outlets	0	0	3	97
Eatery Recommendation	0	10	10	80
Overall application usability	0	0	0	100
Application Performance	0	0	7	93

Table 6- Shows the results achieved from post-market research (Percentages have been rounded to the nearest whole number where appropriate).

The activity of locating Halal outlets relates to criterion 4.1 in the evaluation criteria. 27 users out of the 30 users rated this activity as 'Excellent', with one user commenting "The whole process of exploring outlets is really clear and informative, I never knew there were so many Halal places here in Bristol". The results obtained strongly indicate that the functionality of locating Halal outlets and seeking information about outlets using the map interface works correctly whilst providing the user with a good experience.

The activity of adding outlets related to criterion 4.2 in the evaluation criteria. Adding outlets achieved the highest overall results with 97% of users rating it as 'Excellent'. Having analysed the outlet database at the end of the post-launch research I discovered 13 new outlets had been added, demonstrating that the mechanism for adding outlets functions correctly. Manually analysing the HMC certificate of each outlet showed that every certificate was valid, therefore we can conclude that the verification algorithm functions correctly. From reading the comments left by users, I found that 20% of users wanted to be provided with a more specific error message when the application rejects a certificate. There are two scenarios in which the application rejects a certificate; scenario 1 is where the HMC certificate is not genuine, thus the text extracted from the OCR API is correct and the certificate fails the verification checks. Scenario 2 is where the image taken by the user is unclear, thus the text extracted from the OCR API is incorrect and in turn, the certificate fails the verification checks. In both cases the application outputs the message "Outlet cannot be added". Users requested for a more specific error message to be provided; in scenario 1 they wanted to be informed that the Halal certificate had not passed verification checks and in scenario 2, they wanted to be informed that the picture taken was unclear and encouraged to re-take the photo. This is because a clearer image may result in the outlet successfully being added to Halal Finder.

From a developer's perspective, differentiating between an invalid certificate and the OCR API not being able to correctly process information from an unclear photo is very difficult. This is because the Google Vision API does not include any form of confidence measure, indicating the service's level of confidence in correctly extracting text from the certificate image. After receiving feedback, the only minor change I was able to make to the application was reformatting the application output to "Certificate is incorrect/ if picture taken is unclear please take again" when a certificate has failed to be accepted. A potential future improvement would be to replace the Google Cloud Vision API with Microsoft's Computer Vision API. The Computer Vision API contains a confidence measure which could be used to set a confidence threshold to help provide users with more informative error messages. For example, when an outlet has been rejected, if the API has a confidence of less than 70% indicating that the certificate text has not been extracted well, the user could be requested to re-take the image. On the contrary, if the API generates a confidence rating of 70% or above, the user could be notified that the outlet is invalid and cannot be added to the application.

The activity of obtaining an eatery recommendation related to criterion 4.3 in the evaluation criteria. This activity was heavily modified after the low score and feedback obtained in UAT and subsequently, the results achieved post-launch were significantly better, with the majority of the users rating the activity as 'Excellent'. Comments revealed that users did consider and in some cases, visit the recommended eateries, demonstrating that the recommendation system does correctly utilise user behaviour to recommend

eateries. However, three users rated the activity as satisfactory and indicated that the recommended eatery did not suit their requirements. It is important to note that at the time of conducting post-launch research, the application held 17 outlets and 33 users, only 13 of whom submitted ratings of outlets. The small amount of data held by Halal Finder at present is thought to be the reason why some users may not be receiving satisfactory recommendations. As more users, eateries and ratings are added to Halal Finder, significant improvements can be made to the recommendation system to generate better recommendations for the user. For example, instead of recommending all the eateries rated three or above by the three users that display the most similar behavior to the target user, we could further process the eateries and recommend only those containing features similar to the eateries the target user has rated highly. One feature that could be used is the 'Type' of eatery, for example, if the target user has highly rated an Italian eatery, we could only recommend highly rated Italian eateries from the eateries obtained from the three closest users.

Application usability related to criterion 4.4 in the evaluation criteria. Users found the application easy to use and logical, reflected by the extremely positive results shown in Table 6. This is as a result of the user-centered approach taken when designing each window of the application and the interaction between different pages. Furthermore, user feedback regarding improvements to usability were implemented throughout development to ensure Halal Finder best suited users. To assess criterion 4.4.B all users were asked whether they had experienced any unexpected functionality; shutdowns, geolocation faults or camera faults whilst using the application. 100% of users in the research sample had not encountered any issues whilst using Halal Finder. This feedback is very positive and demonstrates that the application has been built to a high standard with no present bugs and furthermore, the application is able to correctly utilise the phone's geolocation and camera API.

The final component of post-launch research was application performance. This section was related to criterion 4.5 in the evaluation criteria. Users were asked to review the performance of Halal Finder, this included commenting on the speed at which functionality was executed for example, whether they believed the application was timely or took too long when processing a 'Add Outlet' request. 93% of users rated application performance as 'Excellent', with the remaining users rating it as 'Good', thus we can conclude that the performance of the application meets the needs of the users when conducting functionality thus, criterion 4.5.A has been met. To assess criterion 4.5.B, users were asked to review the application's consumption of battery power. This was completed by requesting users to open "Battery Usage" in their phone's settings and reporting whether their device had identified Halal Finder as an application that consumes excess battery power. 22 out of the 30 users examined the battery consumption of Halal Finder; they all expressed that the battery consumption of Halal Finder was normal and the application did not overly consume battery power. Thus, it can be concluded that Halal Finder is efficient in utilising the phone's resources.

9. Final discussion and conclusion

Developing Halal Finder has been an enjoyable experience and has allowed me to familiarise myself with the entire software development cycle, from conducting initial market research to gather requirements, right through to designing the application, development and then testing the application. In the early stages of the project it was important for me to really understand the depth of the problem of locating genuine Halal outlets, as this would provide me with the exact requirements Halal Finder would have to fulfill. Through examining current state of the art, I discovered problems with current applications aiming to tackle the same problem of locating genuine Halal outlets. Furthermore, through conducting market research I took a pro-active approach, meeting with the Muslim community and discussing the problems faced when locating genuine Halal outlets. This research provided me with the exact requirements Halal Finder needed to fulfill: successfully locating Halal outlets, verifying these outlets are genuinely Halal and providing the user with a good experience. The research also established that the target audience of Halal Finder would be 15-36 year old Muslims.

During the design stage of Halal Finder, it was established that a user-centered design approach would be taken as this heavily focusses on the user's needs and experiences, tailoring the application specifically to its audience. To design the UI of Halal Finder an iterative process was set up where I would create wireframes showing the design of each window in the application, the wireframes were then shown to potential users and were modified appropriately using the feedback received to help develop high precision UI mock-ups. The high precision UI mock-ups were then assessed by a different set of users to confirm the UI was of high standard and if necessary further feedback was implemented. By repeating this process for each window in the application, has resulted in an easy-to-use application, reflected in the results from the post-market research where 100% of the users rated the usability of Halal Finder as 'Excellent'.

It was decided that Halal Finder would be developed using the Agile methodology as it allows users to periodically examine application functionality throughout development. After the implementation of each core functionality, for example exploring outlets, users were asked to test the feature. If any consistent criticism was found amongst users during reviews then this feedback was implemented. Using the Agile methodology has allowed end-users to fine tune the application, resulting in the development of a high-quality application. This again can be demonstrated by the outcomes of the post-market research where for each core functionality; locating Halal outlets, adding outlets and eatery recommendation, the dominant rating given by users was 'Excellent'.

After completing post-market research, I regularly spent time using Halal Finder. I found the activity of exploring Halal outlets enjoyable as it allowed me to discover new Halal outlets that I had not come across before, in a simple and clear manner. However, after using the application to add outlets, I recognised that a handful of outlets I visited placed their HMC certificate at the back of the premise, thus there was a poor internet connection and this in turn caused great latency when uploading the image of the certificate. Although this problem was not raised during UAT and post-market research, I consider it vital to tackle in order to ensure genuine Halal outlets can be added. A future solution would be to implement an off-line mode in Halal Finder, which stores the certificate image locally on the phone and when a strong internet connection becomes available the application carries out the request to add the outlet in the background.

Through analysing the results from the post-market research and UAT, the recommendation system can be considered successful for the vast majority of user. In the post-market research, 80% of users rated the recommendation feature as 'Excellent' and all 10 users in UAT liked the eatery that was recommended to them. After post-market research, the user base of Halal Finder continued to grow, resulting in more outlets in more UK cities to be added. During UAT and post-market research, all outlets in Halal Finder were located in Bristol since all users were from there, thus all recommended eateries were local to the users. However, I have recently found that having made requests for eatery recommendations, the recommended eateries are in other cities such as London, too far a distance for me to consider. A future improvement for the application would be to provide the user with a distance measure that they can set and only eateries within that distance will be recommended to them.

As mentioned in section 5.6, the recommendation system was developed in the final sprints of development thus slight adjustments were made to the system such that good recommendations could still be generated for users despite little data being available to the system. One such modification that was made to the system was that when a new user who has no history of rating eateries requests a recommendation, the recommendation system recommends the most highly rated eatery in the application to the user. Although this mechanism has shown to work, when considering the positive results provided during UAT and post-market research, I believe it could be improved so that new users also get recommended eateries that are tailored to them. In order to do this, the application would need to extract more features about eateries such as 'price, 'atmosphere' and 'service standard', where each feature has subsequent options. Prior to receiving recommendations, a new user would be required to complete an initial form which would gather information about features which the user finds important in an eatery, for example for the 'price' feature, a user might prefer the option 'expensive'. Subsequently, we would recommend the outlets which hold these features, allowing tailored recommendations to be made to the user despite them not having reviewed any outlets before.

Within the planned and consistent testing structure used during the development of Halal Finder, there were elements of testing that were very useful. UAT provided the most beneficial feedback which led to drastic re-design and development of the application. For example, as mentioned in section 7.2.4.3, users disliked the way in which the recommended eateries were displayed on a map and wanted them to be ordered and presented more clearly. This led me to re-design the entire recommendation window in accordance to their feedback and the new split screen used for displaying recommendations received very positive feedback during post-market research. Overall, it can be stated that the application was thoroughly tested since the evaluation of the final product revealed all sections of the evaluation criteria were met successfully and this is further supported by the post-market research, where no issues/bugs were identified by users. However, after reflecting on the entire testing phase, I believe there were ways in which testing Halal Finder could have been improved. Although Halal Finder was completed successfully, having met all requirements, the application was developed with too much haste because a significant amount of time was spent waiting for user feedback; firstly during UAT to respond to the questionnaire and secondly, to meet me for discussions on improvements. The wait for users to provide their feedback could have been avoided by implementing an additional form window in the alpha version of the application. Users could have been asked to submit their feedback using the form provided in the application and on submission, the application would have sent the feedback to me via email. By doing this user feedback could have been implemented sooner rather than later.

Increasing user reach is one of the main long-term goals of Halal Finder as stated in requirement 3.2.2. Halal Finder is a native Android application, thus it suffers from fragmentation and in turn, there is no cross-compatibility with iOS. Future work includes developing an iOS version of Halal Finder to run on Apple devices as this would allow the application to be available to a wider audience. The iOS application would be developed using Apple's Xcode framework [55] using Objective-C and Swift.

9.1 Conclusion

During post-market research, I spoke to a lot of people who were pleased that they could finally locate genuine Halal outlets, encouraging them to explore their local cities. I believe that Halal Finder has achieved something much greater than just serving the Muslim community. The principles the application utilises for locating and adding outlets can be expanded to serve other communities such as the Jewish community. Jewish people consume meat which is categorised as Kosher; as with Halal food there are certifying bodies which issue certificates to genuine Kosher selling outlets. The long-term goal of Halal Finder is to completely re-brand the application so that it is able to be used by the UK's Muslim and Jewish community to locate genuine Halal and Kosher food.

Finally, I am very proud of the work I have achieved. Halal Finder has successfully achieved the three key requirements it has set out to do: locate genuine Halal outlets, verify that the outlets added are genuinely Halal and provide the user with a good experience. Meeting the Muslim community during my market research revealed to me the scale of the problem of locating genuine Halal outlets, with many Muslims not feeling confident in the food they were purchasing. When presenting my application to the Muslim community during post-market research, I was pleased that they saw Halal Finder not just as a complete solution to locate genuine Halal outlets but rather a technology that will allow them to integrate more within their local communities by encouraging them to explore their surroundings.

10.References

[1] Geaves, Ron. *Aspects of Islam*. Darton, Longman and Todd, 2005.

[2] Food Manufacture. (2017). *Halal Food is a bigger market than China.* [online] Available at: https://www.foodmanufacture.co.uk/Article/2017/03/15/Halal-food-is-a-bigger-market-for-the-UK-than-China [Accessed 6 January 2019].

[3] Office for National Statistics. (2018). *Freedom of Information (FOI): Muslim Population in the UK.* [online] Available at: https://www.ons.gov.uk/aboutus/transparencyandgovernance/freedomofinformationfoi/muslimpopulationintheuk/ [Accessed 1 November 2018].

[4] Fuseini, Awal, et al. "Halal meat fraud and safety issues in the UK: a review in the context of the European Union." *Food Ethics* (pp. 127-142). 2017.

[5] Djuknic, Goran M., and Robert E. Richton. "Geolocation and assisted GPS." *Computer* 34.2 (pp. 123-125). 2001.

[6] Halal Monitoring Committee. (2018). [online] Available at: https://halalhmc.org [Accessed 2 November 2018].

[7] Mori, Shunji, Hirobumi Nishida, and Hiromitsu Yamada. *Optical character recognition*. John Wiley & Sons, Inc., 1999.

[8] Halal Spot. (2018). [online] Available at: http://www.halalspot.co [Accessed 3 November 2018].

[9] Halal Trip. (2018). [online] Available at: https://www.halaltrip.com [Accessed 3 November 2018].

[10] Google Play. (2018). *Halal Local*. [online] Available at: https://play.google.com/store/apps/details?id=com.halallocal.app&hl=en_GB [Accessed 2 November 2018].

[11] Google Play. (2018). *Halal Food Finder*. [online] Available at: https://play.google.com/store/apps/details?id=com.halal.foodfinder&hl=en_GB [Accessed 2 November 2018].

[12] Halal LookApp. (2018). [online] Available at: http://halallookapp.com/index.php [Accessed 2 November 2018].

[13] Google Search. (2018). *How Search Works*. [online] Available at: https://www.google.com/intl/en_uk/search/howsearchworks/ [Accessed 2 November 2018].

[14] Trip Advisor. (2018). *Restaurants*. [online] Available at: https://www.tripadvisor.co.uk/Restaurants [Accessed 2 November 2018].

[15] Statista. (2018). *Market Share held by mobile operating systems in the United Kingdom (UK) from December 2011 to September 2018.* [online] Available at: https://www.statista.com/statistics/262179/market-share-held-by-mobile-operating-systems-in-the-united-kingdom/ [Accessed 2 November 2018].

[16] Christensen, Henrik Bærbak. "Cross Platform Mobile Development". *University of Aarhus (*pp. 98-106). 2015.

[17] Medium. (2017). *Why we are not cross-platform developers*. [online] Available at: https://medium.com/pixplicity/why-we-are-not-cross-platform-developers-fd7ef70e976d [Accessed 2 November 2018].

[18] Heitkötter, Henning, Sebastian Hanschke, and Tim A. Majchrzak. "Evaluating cross-platform development approaches for mobile applications." *International Conference on Web Information Systems and Technologies* (pp. 120-138). Springer, Berlin, Heidelberg, 2012.

[19] Dalmasso, Isabelle, et al. "Survey, comparison and evaluation of cross platform mobile application development tools." *2013 9th International Wireless Communications and Mobile Computing Conference* (pp. 323-328). IEEE, 2013.

[20]. Google Cloud. (2018). *Google Maps Platform*. [online] Available at: https://cloud.google.com/maps-platform/ [Accessed 5 November 2018].

[21] Statista. (2017). *Market reach of the most popular Android apps in the UK as of June 2017.* [online] Available at: https://www.statista.com/statistics/516285/uk-market-reach-of-the-most-popular-android-smartphone-apps/ [Accessed 5 November 2018].

[22] Molina, Leidy Esperanza, and Sandjai Bhulai. "Recommendation System for Netflix." *University of Amsterdam* (pp. 14-20). 2018.

[23] Schafer, J. Ben, et al. "Collaborative filtering recommender systems." *The adaptive web* (pp.291-324). Springer, Berlin, Heidelberg, 2007.

[24] Google Cloud. (2018). *Optical Character Recognition*. [online] Available at: https://cloud.google.com/vision/docs/ocr [Accessed 7 November 2018].

[25] Data Sciences US. (2018). *Comparing The Top Computer Vision APIs For OCR*. [online] Available at: https://www.datascience.us/comparing-the-top-computer-vision-apis-for-ocr/ [Accessed 7 November 2018].

[26] Charland, Andre, and Brian Leroux. "Mobile application development: web vs.native." *Communications of the ACM* (pp. 49-53), 2011.

[27] Kangas, Eeva, and Timo Kinnunen. "Applying user-centered design to mobile application development." *Communications of the ACM* (pp. 55-59). 2005.

[28] Cui, Bei-Bei. "Design and Implementation of Movie Recommendation System Based on Knn Collaborative Filtering Algorithm." *ITM Web of Conferences* (pp. 32-48). EDP Sciences, 2017.

[29] Qian, Gang, et al. "Similarity between Euclidean and cosine angle distance for nearest neighbor queries." *Proceedings of the 2004 ACM symposium on Applied computing* (pp. 156-212). 2004.

[30] Martin, Robert C. *Agile software development: principles, patterns, and practices*. Prentice Hall, 2002.

[31] Balaji, S., and M. Sundararajan Murugaiyan. "Waterfall vs. V-Model vs. Agile: A comparative study on SDLC." *International Journal of Information Technology and Business Management* (pp. 26-30). 2012.

[32] Android Studio. (2018). [online] Available at: https://developer.android.com/studio/ [Accessed 2 November 2018].

[33] JHipster. (2018). [online] Available at: https://www.jhipster.tech [Accessed 2 November 2018].

[34] Android Studio Documentation. (2019). [online] Available at: https://developer.android.com/docs [Accessed 2 January 2019].

[35] Introduction: Application servers. (2019). [online] Available at: https://www.ibm.com/support/knowledgecenter/en/SSAW57_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/welc_servers.html [Accessed 2 January 2019].

[36] PostgreSQL. (2019). [online] Available at: https://www.postgresql.org [Accessed 2 January 2019].

[37] Stevens, Perdita, and Rob Pooley. *Using UML: software engineering with objects and components*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[38] Walls, Craig. *Spring Boot in action*. Manning Publications, 2016.

[39] Battle, Robert, and Edward Benson. "Bridging the semantic Web and Web 2.0 with representational state transfer (REST)." *Web Semantics: Science, Services and Agents on the World Wide Web* (pp. 61-69). 2008.

[40] Peng, Dunlu, Lidong Cao, and Wenjie Xu. "Using JSON for data exchanging in web service applications." *Journal of Computational Information Systems* (pp. 7-10). 2011.

[41] GitHub. (2019). [online] Available at: https://github.com [Accessed 2 January 2019].

[42] Heroku. (2019). [online] Available at: https://dashboard.heroku.com/apps [Accessed 2 January 2019].

[43] Amazon Web Services AWS. (2019). [online] Available at: https://aws.amazon.com [Accessed 4 January 2019].

[44] Microsoft Azure Cloud Services. (2019). [online] Available at: https://azure.microsoft.com/en-gb/free/cloudservices/search/?&OCID=AID719823_SEM_a39wF7As&lnkd=Google_Azure_Brand&dclid=CJX9_7GY6-ECFUrjGwodXnsI_w [Accessed 4 January 2019].

[45] Liquibase. (2019). [online] Available at: https://www.liquibase.org [Accessed 2 January 2019].

[46] Google Play. (2019). [online] Available at: https://play.google.com/store?hl=en_GB [Accessed 2 January 2019].

[47] Jones, Michael, Brain Campbell, and Chuck Mortimore. *JSON Web Token (JWT) profile for OAuth 2.0 client authentication and authorization Grants* (pp.2-5). RFC 7523, 2015.

[48] Working with Geospatial Data. (2019). [online] Available at: https://www.fulcrumapp.com/blog/working-with-geodata/ [Accessed 22 January 2019].

[49] Android Developers – Rating bar. (2019). [online] Available at: https://developer.android.com/reference/android/widget/RatingBar [Accessed 22 January 2019].

[50] Android Geocoder. (2019). [online] Available at: https://developer.android.com/reference/android/location/Geocoder [Accessed 22 January 2019].

[51] Android – Date. (2019). [online] Available at: https://developer.android.com/reference/java/util/Date [Accessed 22 January 2019].

[52] Pencil. (2019). [online] Available at: https://pencil.evolus.vn [Accessed 23 January 2019].

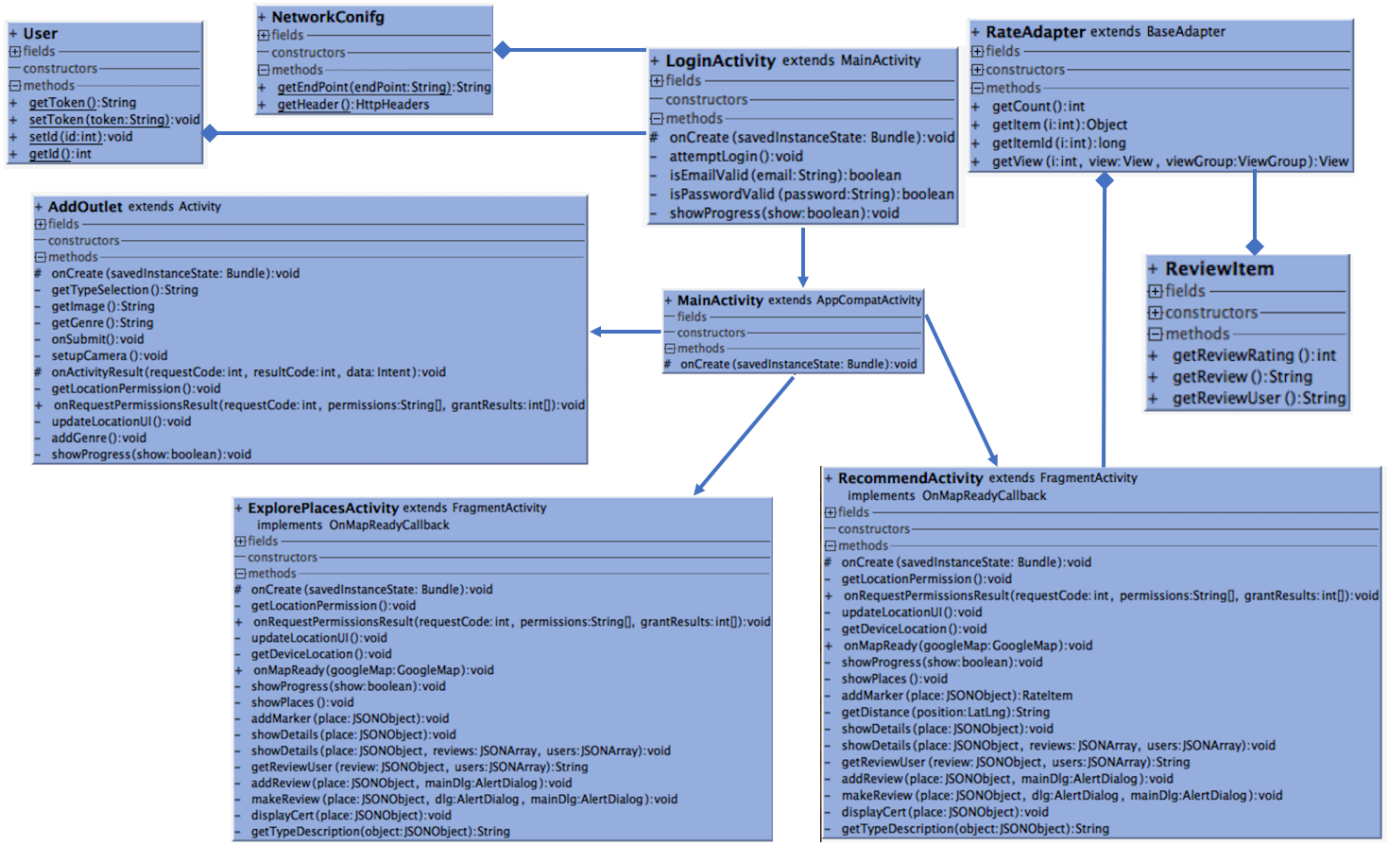
[53] Android – Espresso. (2019). [online] Available at: https://developer.android.com/training/testing/espresso [Accessed 25 January 2019].

[54] Johnson Controls International. “Internal Document: Software Practice – Application Development: Testing”. 2012.

[55] Xcode. (2019). [online] Available at: https://developer.apple.com/xcode/ [Accessed 25 March 2019].

11. Appendices

A. Class diagram displaying the important components of the Android Client Application.



B. Code showing the conditional logic used during certificate validation.

```
//Ensure HMC wording is correct
if (l.contains("HMC approved products")) {
    hmcCheck = true;
}

if (l.contains("Validation:")) {
    outlet.setCertImgContentType(l);

    String[] words = l.split( regex: " ");
    String dateWord = words[words.length - 1];

    //The date has to be in this format
    SimpleDateFormat sdf = new SimpleDateFormat( pattern: "dd.MM.yyyy");
    Date date = sdf.parse(dateWord);

    //check that the certificate has not expired
    if (date.after(new Date())) {
        expiryCheck = true;
    }

    break;
}

//Ensure approval wording is correct
if(l.contains("Has attained the HMC Approval for selling HMC approved products")) {
    attainmentCheck = true;
    break;
}

//Ensure email to source is correct
if(l.contains("info@halalhmc.org")) {
    emailCheck = true;
    break;
}
```