Requirement specification document

HAMZAH MALIK – ACSF375

Contents

[**Purpose and scope of this document** 2](#_Toc30740694)

[**Intended audience of this document** 2](#_Toc30740695)

[**Current systems in place and their issues.** 2](#_Toc30740696)

[Paper-Based Menus 2](#_Toc30740697)

[Tablets at tables 3](#_Toc30740698)

[**Proposed system** 3](#_Toc30740699)

[**Essential Requirements:** 3](#_Toc30740700)

[**Non-essential requirements** 4](#_Toc30740701)

[**Volere templates – Functional and Non-Functional requirements.** 5](#_Toc30740702)

[**Things the system won’t do** 8](#_Toc30740703)

[**Notes** 9](#_Toc30740704)

[**References** 9](#_Toc30740705)

[Bibliography 9](#_Toc30740706)

# 

# **Purpose and scope of this document**

For my final year project at university, I decided to develop an Augmented Reality (AR) app aimed at both restaurant owners and consumers alike. This application intends to incorporate AR within a restaurant’s menu-based system by displaying interactable items in AR, where users can see what their order may look like and valuable information about such products.

Extensive research into existing mobile applications has led me to consider all the requirements of my application, as well as external non-functional requirements and the process behind mitigating risks that correlate with my project. Mapping out such requirements has also led me to consider what features I deemed most important, and what could be considered “desired” but not essential to my project.

# **Intended audience of this document**

This document is intended to be read by restaurant owners, who will be my proposed clients during this project. The layout of requirements will not only give them insight into my project, but also prove to be useful to look at for reference during the development stages of this project.

# **Current systems in place and their issues.**

## Paper-Based Menus

For most restaurants, the handling of menus is traditionally done by receiving a paper-based menu where customers can view a selection of items, sometimes with the aid of a few images to help sway their choice in chosen items from a menu.

The structure of this process varies from restaurant to restaurant, with someone handing out multiple menus in the form of main courses, drinks and even desserts; whilst others may encapsulate this information into one large menu.

The current process for handling the addition or removal of items to a menu can be extremely meticulous for restaurant owners. No owner wants to keep items on the menu that don’t sell well, but there are costs attached to changing a menu in the form of both time and menu. Furthermore, the restaurant industry is focused on customer service and satisfaction. It can be detrimental to a restaurants credibility by having to apologize to customers wanting a product that the item is “no longer available”.

Based on research conducted by OpenTable, “86% of customers regularly check out menus online before dining out” (OpenTable, 2015). In a restaurant environment, customers often don’t read paper menus, rather than quickly scanning over the item. This means menus often have a small amount of time to make a big impact.

When the ingredients run out for a popular dish, staff need to remember to tell each new customer that it’s “sold out”, which often can be disappointing for customers who will need to spend even more time deciding on what to order.

One of the biggest reasons restaurant owners use paper-based menus are for the “low costs” attached to printing them out. However, the costs attached to having to reprint menus or updating price lists aren’t as small as owners like to think. In a 1997 study, store-level data from five multi-store supermarket chains were examined to measure menu costs. The study found that “menu costs per store averaged more than 35 percent of net profit margins” (Keton, 2019). This meant that profitability of items needed to drop by 35% to simply justify updating the final price of items.

Restaurants often only display images attached to certain images in a menu. Having a menu which allocates space for images of every single item on a paper menu would take up too much space to manage. Here we find another problem: those who do not speak English very well often struggle to understand a restaurant menu without visual aid, which could lead to lower revenue for restaurant owners, with “1.3% of the population of England and Wales (726,000 people)” not speaking English very well. (Gov.UK, 2018).

### Tablets at tables

Another form of system in place is to use small tablets/readers on tables to use for menu assistance with items. Whilst these tablets solve some of the functionality problems of paper-based systems, they instead bring their own problems. These solutions are still relatively new, meaning only a few chain restaurants incorporate these. Another problem is that they’re likely to not be an affordable business expense for smaller restaurants and cafes for every table. Replacing any tablets that get stolen, dropped or broken is also an expense restaurant owners need to mitigate somehow. Batteries will also start to deplete overtime, so there would need to be staff dedicated to overseeing the status of tablets.

So how can restaurants incorporate the functionality of what tablets provide, without the substantial costs attached?

# **Proposed system**

The system I propose to develop is an augmented reality (AR) based approach to a menu-based system that builds upon the functionality of the previous systems mentioned above.

The app focuses on using augmented reality to display different items from different menus in which customers can view their product within an AR plane, where they can click on them to see more information about the item. The availability of viewing products in AR also removes the worry of language barriers, where customers can view products without having to question what the item is.

Another major benefit to using my mobile application is the accessibility being able to view a menu without being in the restaurant itself. The users simply need the app installed to be able to select what menu they’d like to check on-the-go. For example, a customer can use the app to see a range of menus to select from. Picking one, the user is then welcomed to a screen where they can select from a list of items on a menu and then view them in AR. They can then either click on the product again to see more information about it or select another item. The system will also have functionality to clear instances of AR models on a plane or search for specific items from a list. Popular mobile applications such as ‘UberEats’ and ‘JustEat’ are founded on the idea of scalable large-scale systems with accessibility to menus regardless of location.

My application also has the capabilities to manually add or remove items to a menu or a list of menus without the cost attached to paper-based systems.

Below is the summary of the requirements of my application:

# **Essential Requirements:**

* Downloadable on Android 7+ devices
* This is the minimum requirements for the ARFoundation framework to run on, therefore all Android devices must satisfy this condition.
* Home menu with selection of other menus to select from.
* This will be a layout for different restaurants where users can select from
* Categorically separated to help identify the type of restaurant.
* Filter/Search system to search for specific items/restaurants.
* Should allow users to search for restaurants by name.
* More complex filter by searching for a product name and displaying all restaurants containing that item (by name).
* Selectable items from a ‘menu’.
* The ‘menu’ itself, users can see items from a list of mains, sides and desserts to choose from, alongside an associated price.
* AR plane for users to place 3D assets.
* Plane will provide users an understanding of where they can place AR objects, depending on their environment.
* ARFoundation framework handles the detection of environmental mapping.
* Set limit to the number of instances of each item displayed on a plane
* Circumvents AR Display clogging on associated device.
* Users must be able to clear all AR instances using a “clear button”
* A hard “reset” button to remove 3D models in existence.
* App must provide functionality that enables the navigation between menus, items and other pages
* This will be in the form of buttons.
* App should provide an instructional guide explaining how to use the app.

# **Non-essential requirements**

These are features of my mobile application that could be implemented but are not considered essential from the research done into essential requirements. Due to time constraints imposed by the project, these may also be difficult to implement as part of the prototype. If enough time is left after the development of the core functionality, then these features may be seen, otherwise they won’t appear.

* Implementation of ‘Admin’ user who can add in items.
* Whilst this sounds like a useful feature to have on-paper, it isn’t a necessity as the addition of items onto the app will most likely be done by me (the developer)
* Once I have agreed on working with a client, it’s unlikely they would want to handle the addition of items or their own menu into my app, as they would expect the developer to handle such a requirement.
* Using a database to store assets.
* Will be useful when handling a mass amounts of assets by limiting the size of an app, but under a controlled prototype environment, it’s not a necessity.
* The likelihood my app will have a large installation size is low as I’m aware of what assets I’m using and their size.
* Implementation of a database means a slower application overall, as the app needs to fetch assets stored into a database and then display them.
* Instead, software like DigitalOcean’s spaces can be used if this application is commercialised, which is an “Object storage with a built-in content delivery network (CDN) meets your infrastructure needs with a simpler architecture”.

# **Volere templates – Functional and Non-Functional requirements.**

The Volere Requirements Specification Template has been chosen to map out my functional and non-functional requirements as it provides a useful template and guide for writing specifications in a high-level form. According to the Volere website, the template has been downloaded “in excess of 20,000 times” (Volere, 2018).

|  |  |  |
| --- | --- | --- |
| **Requirement ID:** 1 | **Requirement Type:** Constraint | **Event/Use case #:** |
| **Description:** The application must be compatible with an Android device running Android 7+ | | |
| **Rationale:** Users should be able to download the app and run it straight away without fail. | | |
| **Source:** The initial statement of requirements | | |
| **Fit criteria:** The application will be installed and run on an Android device running Android 7 and up, as this is the minimum version of Android required to run AR Foundations framework. The application will be installed and run on a range of Android devices ranging from Android 7 to Android 10.0 | | |
| **Customer satisfaction:** 4 | | **Customer dissatisfaction:** 5 |
| **Priority:** Essential | | **Conflicts:** None |
| **Supporting Material:** The initial statement of requirements | |  |
| **History:** New requirement | |

|  |  |  |
| --- | --- | --- |
| **Requirement ID:** 2 | **Requirement Type:** FR | **Event/Use case #:** |
| **Description:** Users should be presented with a home screen consisting of menus to choose from. | | |
| **Rationale:** A user must be able to load up the application and be presented with a home menu that offers them a range of menus to click on, alongside the restaurant name above each. Restaurants should be categorised based on cuisine speciality such as “Mediterranean” and fall under two categories if they satisfy both. | | |
| **Source:** The initial statement of requirements | | |
| **Fit criteria:** Ability to click on restaurant menus from each sub-category of cuisine type. Each of these should lead to the relevant augmented reality menu where users then select from a menu of items. | | |
| **Customer satisfaction:** 5 | | **Customer dissatisfaction:** 5 |
| **Priority:** Essential | | **Conflicts:** None |
| **Supporting Material:** The initial statement of requirements | |  |
| **History:** New requirement | |

|  |  |  |
| --- | --- | --- |
| **Requirement ID:** 3 | **Requirement Type:** FR | **Event/Use case #:** |
| **Description:** Users must able to search or filter for specific menus and items. | | |
| **Rationale:** A user should be able to load up the application and then click on the search bar to search up a restaurant. They should also be able to search for an item and the application will return all the restaurants who provide that set product. | | |
| **Source:** The initial statement of requirements | | |
| **Fit criteria:** Searching for different productsfrom differentmenus and checking if they return the correct restaurant who supply such a product. | | |
| **Customer satisfaction:** 4 | | **Customer dissatisfaction:** 4 |
| **Priority:** Essential | | **Conflicts:** None |
| **Supporting Material:** The initial statement of requirements | |  |
| **History:** New requirement | |

|  |  |  |
| --- | --- | --- |
| **Requirement ID:** 4 | **Requirement Type:** FR | **Event/Use case #:** |
| **Description:** Users must be able to select from items in an interactive AR menu. | | |
| **Rationale:** A user should be able to load up the application, select a specific menu and then be led to a page where they can select items from a menu. This menu should be from a drop-down list and they can then place items onto a plane. | | |
| **Source:** The initial statement of requirements | | |
| **Fit criteria:** Selecting items from a menu should bring up an AR plane. | | |
| **Customer satisfaction**: 4 | | **Customer dissatisfaction:** 5 |
| **Priority:** Essential | | **Conflicts:** None |
| **Supporting Material:** The initial statement of requirements | |  |
| **History:** New requirement | |

|  |  |  |
| --- | --- | --- |
| **Requirement ID:** 5 | **Requirement Type:** FR | **Event/Use case #:** |
| **Description:** Users must be able to select from items in an interactive AR menu. | | |
| **Rationale:** A user should be able to load up the application, select a specific menu and then be led to a page where they can select items from a menu. This menu should be from a drop-down list and they can then place items onto a plane. | | |
| **Source:** The initial statement of requirements | | |
| **Fit criteria:** Selecting items from a menu should bring up an AR plane. | | |
| **Customer satisfaction**: 4 | | **Customer dissatisfaction:** 5 |
| **Priority:** Essential | | **Conflicts:** None |
| **Supporting Material:** The initial statement of requirements | |  |
| **History:** New requirement | |

|  |  |  |
| --- | --- | --- |
| **Requirement ID:** 6 | **Requirement Type:** FR | **Event/Use case #:** |
| **Description:** Users must be able to place items onto an AR plane. | | |
| **Rationale:** A user can place items onto an augmented reality plane after selecting it from the given menu. The AR plane will provide users an understanding of where exactly they can place such AR objects, depending on their given environment. The ARFoundation framework handles any environmental plane detection and mapping, as well as the appearance of 3D models once selected. | | |
| **Source:** The initial statement of requirements | | |
| **Fit criteria:** Selecting an item from a menu and then placing it onto a given AR plane will provide a visual of the 3D model in Augmented Reality. | | |
| **Customer satisfaction**: 5 | | **Customer dissatisfaction:** 5 |
| **Priority:** Essential | | **Conflicts:** None |
| **Supporting Material:** The initial statement of requirements | |  |
| **History:** New requirement | |

|  |  |  |
| --- | --- | --- |
| **Requirement ID:** 7 | **Requirement Type:** FR | **Event/Use case #:** |
| **Description:** Users must only be able to place a certain number of instances for each item on an AR plane. | | |
| **Rationale:** Users should have a parameterised limit to the number of instances of each item that can be displayed on a plane. This circumvents the chance of the AR display being clogged up on the associated android device. | | |
| **Source:** The initial statement of requirements | | |
| **Fit criteria:** Users can place items onto a plane. After reaching a set limit, the first instance of the item placed will be replaced with the very latest one. | | |
| **Customer satisfaction**: 5 | | **Customer dissatisfaction:** 5 |
| **Priority:** Essential | | **Conflicts:** None |
| **Supporting Material:** The initial statement of requirements | |  |
| **History:** New requirement | |

|  |  |  |
| --- | --- | --- |
| **Requirement ID:** 8 | **Requirement Type:** FR | **Event/Use case #:** |
| **Description:** Users must be able to clear all instances of AR models on the given plane. | | |
| **Rationale:** After placing items onto an AR plane, they should be able to click on a ‘clear’ button within the application to clear all instances of AR models placed. | | |
| **Source:** The initial statement of requirements | | |
| **Fit criteria:** Clicking the clear button will lead to the successful clearance of AR items on a set plane. | | |
| **Customer satisfaction**: 5 | | **Customer dissatisfaction:** 4 |
| **Priority:** Essential | | **Conflicts:** None |
| **Supporting Material:** The initial statement of requirements | |  |
| **History:** New requirement | |

|  |  |  |
| --- | --- | --- |
| **Requirement ID:** 9 | **Requirement Type:** FR | **Event/Use case #:** |
| **Description:** Users must only be able to place a certain number of instances for each item on an AR plane. | | |
| **Rationale:** Users should have a parameterised limit to the number of instances of each item that can be displayed on a plane. This circumvents the chance of the AR display being clogged up on the associated android device. | | |
| **Source:** The initial statement of requirements | | |
| **Fit criteria:** Users can place items onto a plane. After reaching a set limit, the first instance of the item placed will be replaced with the very latest one. | | |
| **Customer satisfaction**: 4 | | **Customer dissatisfaction:** 4 |
| **Priority:** Essential | | **Conflicts:** None |
| **Supporting Material:** The initial statement of requirements | |  |
| **History:** New requirement | |

|  |  |  |
| --- | --- | --- |
| **Requirement ID:** 10 | **Requirement Type:** FR | **Event/Use case #:** |
| **Description:** Functionality must be provided to navigate through the mobile application. | | |
| **Rationale:** The AR app must provide functionality that enables the user to navigate through the application’s menus, products and other pages within the app. This should be done in the form of in application button. | | |
| **Source:** The initial statement of requirements | | |
| **Fit criteria:** Users will click on buttons and successfully be led to a new page or previous page, depending on the button pressed. | | |
| **Customer satisfaction**: 5 | | **Customer dissatisfaction:** 5 |
| **Priority:** Essential | | **Conflicts:** None |
| **Supporting Material:** The initial statement of requirements | |  |
| **History:** New requirement | |

|  |  |  |
| --- | --- | --- |
| **Requirement ID:** 11 | **Requirement Type:** FR | **Event/Use case #:** |
| **Description:** App should provide an instructional guide explaining how to use the app. | | |
| **Rationale:** The application must have a section dedicated to how to use the app and how to place items onto an AR plane. | | |
| **Source:** The initial statement of requirements | | |
| **Fit criteria:** When users click on the ‘how to use’ section, they will be sent to relevant section. | | |
| **Customer satisfaction**: 5 | | **Customer dissatisfaction:** 5 |
| **Priority:** Essential | | **Conflicts:** None |
| **Supporting Material:** The initial statement of requirements | |  |
| **History:** New requirement | |

|  |  |  |
| --- | --- | --- |
| **Requirement ID:** 12 | **Requirement Type:** NFR (Performance) | **Event/Use case #:** |
| **Description:** The mobile application should take no longer than 5 seconds to load up assets when selected and attempted to be placed onto an AR plane. | | |
| **Rationale:** The system should be able to respond quickly enough, regardless of device and hardware attached to the Android device. The requirements do not consider the size of the models being displayed, and the difference between low-poly and hi-poly count models. | | |
| **Source:** New Requirement | | |
| **Fit criteria:** The response time should be quick based on the size of the models. Consistent model sizes will lead to similar loading times, whereas drastically different sizes could affect performance. | | |
| **Customer satisfaction**: 5 | | **Customer dissatisfaction:** 5 |
| **Priority:** Essential | | **Conflicts:** None |
| **Supporting Material:** The initial statement of requirements | |  |
| **History:** New requirement | |

|  |  |  |
| --- | --- | --- |
| **Requirement ID:** 13 | **Requirement Type:** NFR (Scalability) | **Event/Use case #:** |
| **Description:** The mobile application should scale accordingly to a range of Android devices, depending on the size of the device’s screen. This should apply to Android tablets too. | | |
| **Rationale:** The system should be able to scale up or down with the UI buttons, main menu and selection options also scaling accordingly. For larger devices, these buttons will appear larger, and smaller for phones. This will be done using the Canvas Scaler component which can be added to my root canvas for my menus. The Canvas scaler is a Game Object with a Canvas component on it, which all the UI elements are children of. In the Canvas Scaler component, you can set its UI Scale Mode to “Scale With Screen Size”. | | |
| **Source:** New Requirement | | |
| **Fit criteria:** Users should be able to see clear UI in the application regardless of device. This will be tested on phones ranging from 5-6inches in screen size to tablets that range from 7-10 inches. These buttons should also still be clickable and not overlap with each other. | | |
| **Customer satisfaction**: 5 | | **Customer dissatisfaction: 4** |
| **Priority:** Essential | | **Conflicts:** None |
| **Supporting Material:** The initial statement of requirements | |  |
| **History:** New requirement | |

|  |  |  |
| --- | --- | --- |
| **Requirement ID:** 14 | **Requirement Type:** NFR (Usability) | **Event/Use case #:** |
| **Description:** The mobile application should use a colour scheme that is legible and does not confuse the end user. | | |
| **Rationale:** The contrast between text and background/foreground colouration should be made clear, and easy to read in both daytime and night-times. The app shouldn’t have text that is hard to decipher based on the colour pallet used. | | |
| **Source:** New Requirement | | |
| **Fit criteria:** During the testing phase of my application, have a set of different colour schemes for end users to see (in the form of either wireframes or actual implementation) and decide on a final colour choice using the data received as feedback. | | |
| **Customer satisfaction**: 5 | | **Customer dissatisfaction:** 5 |
| **Priority:** Essential | | **Conflicts:** None |
| **Supporting Material:** The initial statement of requirements | |  |
| **History:** New requirement | |

# **Things the system won’t do**

There are some features that I have decided to currently not incorporate due to a lack of time and resources for my given project.

Currently, the application will only be deployed on Android devices, with the users required to have access to a suitable android device. In the future, I will be able to export this project to iOS versions of the application or a web-driven application, so access to my application is not restricted by device.

The ARFoundation framework handles mitigating cross-platform code across iOS and Android devices, and theoretically the only thing holding me back is a lack of access to an Apple Mac and XCode to deploy my application onto iOS. However, this is still something that can be done if given access to both.

A web-driven application could be developed, if the ARFoundation framework ever support web-AR. Google are still testing the capabilities of WebXR (Google Developers, 2018) and whilst it’s not commercially ready, once released, could definitely be used to adapt my app to a web-application.

The mobile application will also not include any form of payment options – this application has been intended to be an addition to restaurants, and not a platform to buy or sell items, but simply to see what products each restaurant have available.

# **Notes**

The requirements for the development of my application serve as a guideline and basis for the development milestones of my project. I aim to have all my essential requirements implemented as an evident prototype. If there is enough time left after the core development has been completed, I will also attempt to implement my non-essential requirements.

I understand that if they are not implemented, it wouldn’t likely to be an issue for potential clients of this project, if taken further.

# **References**

# Bibliography

Google Developers. (2018, May 01). *Welcome to the immersive web*. Retrieved from Google Developers: https://developers.google.com/web/updates/2018/05/welcome-to-immersive

Gov.UK. (2018, February 08). *English Language Skills*. Retrieved from Gov.Uk: https://www.ethnicity-facts-figures.service.gov.uk/uk-population-by-ethnicity/demographics/english-language-skills/latest

Keton, W. (2019, September 09). *Menu-Costs*. Retrieved from Investopedia: https://www.investopedia.com/terms/m/menu-costs.asp

OpenTable. (2015, July 16). *OpenTable Research Reveals What Diners Actually Want from Technology*. Retrieved from OpenTable: https://press.opentable.com/news-releases/news-release-details/opentable-research-reveals-what-diners-actually-want-technology

Volere. (2018, 01 01). *Volere Requirements Specification Template*. Retrieved from Volere.org: https://www.volere.org/templates/volere-requirements-specification-template/