

Airport Assistant

Shauna Moran

1. Introduction

1.1 Overview

Airport Assistant is an Android Application to help users who suffer from anxiety in airports with dealing with navigating through the airport. According to travelweekly.co.uk a study conducted by CPP found that one third of people find a working week to be less stressful than taking a flight and a quarter say it would be less stressful to move house than to take a flight. The purpose of this application is to take the stress out of air travel and make it more accessible to people who avoid travelling due to fear of the panic it will cause.

Airport Assistant works by, instead of overloading users with information about the airport, bringing them through the process step by step, completing checks, giving advice and providing important information. It is almost like someone holding the user's hand and being their assistant throughout the journey.

The stages of this process will include...

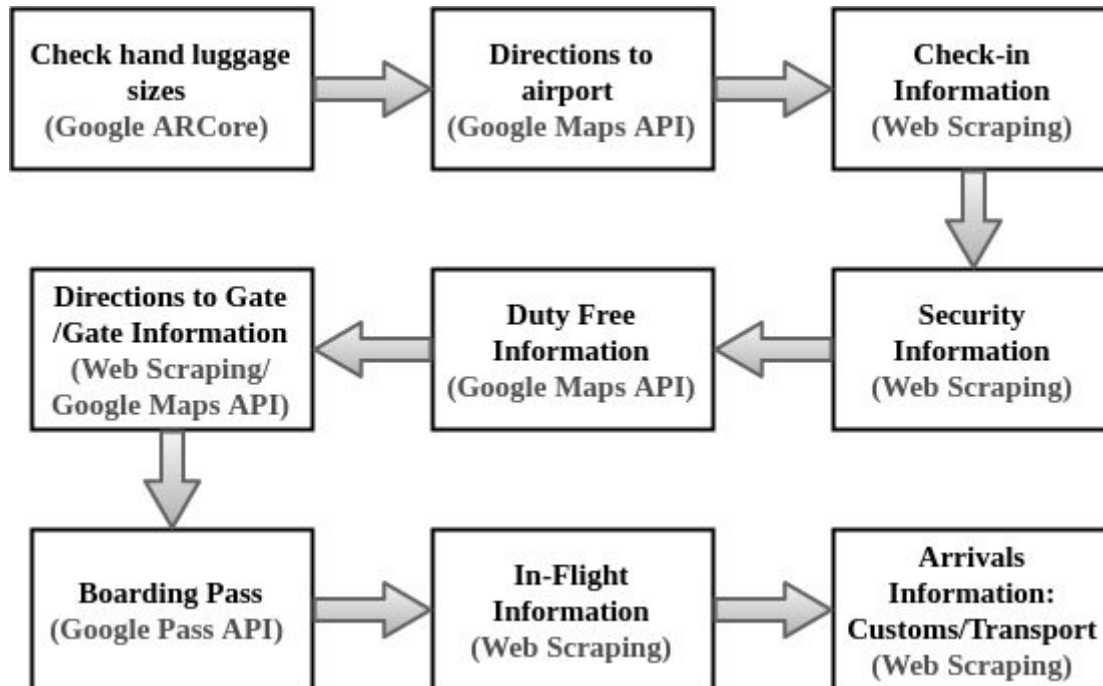
- Checking hand luggage sizes
- Directions to the airport
- Check-in information
- Security information
- Duty free information
- Directions to gate/ Gate information
- Display boarding pass
- In-flight information
- Arrivals information

And will include technologies such as...

- AR (Augmented Reality)
- Web Scraping
- Google Pay API for Passes
- Google Maps API
- Firebase database
- Android application

The primary feature of the application is the AR (Augmented Reality) functionality for measuring luggage sizes. This will help ease the worries of users who are concerned that their hand luggage may not meet the requirements of their airline before they leave home.

A diagram of how Airport Assistant would work...



1.2 Business Context

This application would be used in airports to allow users to create a more comfortable and less nerve-wrecking atmosphere when passengers are passing through the airport. This application would help keep passengers informed and remind them to complete tasks such as prepare for security, check hand luggage sizes and progress to their gate which would attempt to take pressure off the staff in the airport. The application could also contain advertising using location based services for the shops in the Duty Free area of the airport.

1.3 Glossary

- **AR:** Augmented reality is a technology that works on computer vision based recognition algorithms to augment sound, video, graphics and other sensor based inputs on real world objects using the camera of your device.
- **Web Scraping:** Web Scraping is a technique employed to extract large amounts of data from websites whereby the data is extracted and saved to a local file in your computer or to a database in table (spreadsheet) format.
- **API:** An application programming interface (API) is a set of routines, protocols, and tools for building software applications.
- **SDK:** A software developer's kit (SDK) is a set of programs used by a computer programmer to write application programs.
- **QR Code:** A matrix barcode that is read by photographing it with the camera of a smartphone or other mobile device that is equipped with a bar-code reader.

2. General Description

2.1 Product / System Functions

Checking hand luggage sizes

The primary feature of this application is the AR functionality which allows users to check the size of their hand luggage. This will be developed using the Google ARCore SDK. ARCore is Google's platform for building augmented reality experiences. Using different APIs, ARCore enables your phone to sense its environment, understand the world and interact with information. It does this using Motion Tracking, Environmental Understanding and Light Estimation. I will use this technology which detects 2D planes to detect the front and side of the hand luggage and using a measurement algorithm, that I will write, to measure the dimensions of the hand luggage the user has. I will then compare this measurement to the dimensions which are allowed by a user's airline.

Directions to the airport

Users asked how they are travelling to the airport and will then be given directions to the airport for their specific mode of transport. This will be built using the Google Maps API. This will also give users an estimate of how long it will take them to travel to the airport so they are aware when they should begin their journey.

Check-in information

When a passenger are entering a journey into the application they will scan the QR code on their boarding pass and the flight information will be taken into the application using the Google Pay API for Passes. They will also enter whether or not they are checking in luggage. If a user is checking in luggage this stage will give users information on what they cannot place in their check luggage and what the weight restrictions are which will be fed to the application via web scraping.

Security information

This section of an application will give users information on the process of passing through security. This will include the rules of passing through security, tips and tricks and for certain airports where the information is available, security wait times. This information would be gathered using web scraping.

Duty free information

The Duty free section of the application would provide two different functions, one being that they can use a map to find certain stores in the Duty free area and also that stores in Duty free can offer advertising using location based services. This functionality will be built using the Google Maps API.

Directions to gate/ Gate information

Again using the Google Maps API, I will provide information to users on how far they are from their gate. Unfortunately, gate information is not freely available online so I will have to ask a user to enter their gate number which I will then use to direct them to their gate on Google Maps and tell them the distance of the walk.

Display boarding pass

As passengers will have already scanned their boarding passes using the Google Pay API for Passes, the application will at this stage display the customers boarding pass so that they do not have to either print their boarding passes or have to use multiple applications throughout the process.

In-flight information

While a user is on their flight I hope to provide offline content such as flight information which will be provided via web scraping so that user's feel that they are informed throughout the process. I will also look into providing specific airline flight information such as airline magazines and menus in this section.

Arrivals information

When a user lands in their desired location the application will provide them with information on their current location, transport options from this airport and baggage claim information if it is available. This will be completed using web scraping and the Google Maps API.

2.2 User Characteristics and Objectives

Airport Assistant is aimed at users who are passing through the airport, primarily those who suffer from anxiety when completing this process. As this application will be used by people of all ages and technical abilities, the user interface must be easy to use and intuitive. It must provide information clearly and notify users when tasks must be completed. The user interface must be accessible as it may be used by people with accessibility requirements such as screen readers.

As users will be brought through the process step by step, the application will be divided into a number of stages with a progress bar present on the screen so that users feel that they are making progress as they move through the airport. I intend to keep the user interface as minimalistic as possible, while also providing all necessary information and functionality.

2.3 Operational Scenarios

Scenario One: Login

Current System State:

User chooses to login with either username and password, Google or Facebook.

Informal Scenario:

If user is logging in conventionally they must type in their username and password and click enter. Otherwise, they must select the Google or Facebook icons and enter required fields.

Next Scenario:

The user will then be successfully logged into the application and will be presented with the home screen.

Scenario Two: Create Account

Current System State:

User does not currently have an Airport Assistant account and wishes to create an account.

Informal Scenario:

They click "Create Account" button and it progressed to the next page where they will enter required fields and click "Confirm" button.

Next Scenario:

User now has an account on Airport Assistant and can login to the application.

Scenario Three: User uploads a journey

Current System State:

User selects the "Upload Journey" button on the homepage of the Airport Assistant.

Informal Scenario:

Passengers will be presented with a QR scanner which they will use to scan the QR code on their boarding pass. This information will be used to upload a user's flight information, check what airline they are travelling with so hand luggage sizes can be calculated correctly and for uploading the boarding pass itself so that it can be presented later.

Next Scenario:

The passengers journey is now uploaded to the application and can be seen on the home screen of the application.

Scenario Four: User Selects a Saved Journey

Current System State:

A user has already uploaded a journey and wants to begin this journey.

Informal Scenario:

The user clicks the journey they wish to begin and are asked to confirm if this is the correct journey.

Next Scenario:

The passenger begins the process of stepping through this journey.

Scenario Five: User uses AR screen to check hand luggage sizes

Current System State:

User has began a selected journey. They now wish to check the dimensions of their hand luggage. As a user has scanner their boarding pass and therefore provided airline information, this allows us to check the size of the user's bag compared to what luggage size is permitted on their flight.

Informal Scenario:

The passenger users the AR functionality to scan their luggage from two sides so that the length, width and depth of the bag can be calculated. Once the scanning is completed the user is informed whether the bag passes or fails for the specific airline. If the bag passes, the user is progressed to the next stage and if not, the user is asked if they wish to try another bag or skip this stage.

Next Scenario:

The user now knows if there baggage is the correct size and can move on to the next stage with confidence that their bags are okay.

Scenario Six: User gets directions to airport

Current System State:

The passenger is asked how they intend on travelling to the airport, they are then asked if they would like directions for their chosen mode of transport.

Informal Scenario:

The user can is then either presented with google maps directions to the airport from their current location.

Next Scenario:

The user is now at the airport and can progress to the next stage of the process.

Scenario Seven: Check- in phase**Current System State:**

If the user selected after login that they will have luggage to check in they will be presented with the check-in information page, if not they will progress directly to the next stage.

Informal Scenario:

Users will be shown information on how to check for their gate, what their permitted luggage weight is and what their bag weight is (this information was entered earlier). They will also be informed of warnings of items that are not allowed to be placed in check-in luggage.

Next Scenario:

The user has now checked in their hold luggage and will be progressed onto the security stage.

Scenario Eight: Security**Current System State:**

Users are about to pass through security and are looking for information on this process.

Informal Scenario:

The application will inform users of how long the security wait times are in the airport the user is travelling from if this specific airport provides this information online. Information about the process and tips and tricks are also provided here.

Next Scenario:

The user has now passed through security and can progress to the next stage of passing through the airport.

Scenario Eight: Duty Free**Current System State:**

The user has passed through security and is now in Duty Free and is looking for information.

Informal Scenario:

Passengers will be given the option to view a map of the duty free, search a specific store or view deals which are currently being offered in the Duty Free.

Next Scenario:

Users have accessed this information and will progress onto the next stage of getting directions and time estimates to their gate.

Scenario Nine: Directions to gate**Current System State:**

The passenger will then be asked to check the departures board for their gate number as this information is not readily available outside of the airport.

Informal Scenario:

Once the user enters their gate number they will be presented with the estimated walking time to their gate and asked if they would like directions to their gate.

Next Scenario:

The user has now progressed to their gate and is ready to board their flight with their boarding pass.

Scenario Ten: Boarding Pass**Current System State:**

The user has progressed through all previous stages and is now at their gate.

Informal Scenario:

The application displays the passengers boarding pass on the application to be scanned when boarding the plane.

Next Scenario:

The user is now onboard the flight and inflight information can now be presented.

Scenario Eleven: Inflight Info**Current System State:**

The user is now onboard the flight and is looking for inflight information which can be accessed offline.

Informal Scenario:

The application will display information on how long the journey will take, specific airline information and other important information.

Next Scenario:

The user has completed their flight and is now looking for information on their destination airport.

Scenario Twelve: Arrivals Info**Current System State:**

The user has landed in their destination airport and now is looking for information on transport options.

Informal Scenario:

The application will provide users with a list of their transport options and ask them where they intend on travelling to.

Next Scenario:

The user is now finished with this journey and has arrived at their destination.

2.4 Constraints**Time Constraints**

As this project is due on the 19th of May the main constraint I will face in the process of developing this project will be getting the application completed in the time frame. I will have to consistently consider this constraint when deciding what functionality I wish to implement in Airport Assistant.

Integration Constraints

Many technologies are intended to be utilised in the development of this project including an Android application, a Firebase database, web scraping and AR functionality. These numerous turning cogs in the application will make it difficult to integrate this entire application and will make it difficult to combine the application into one finished project.

Security Requirements

As I will be storing boarding passes and passenger information I will have to be very careful to secure user data in a safe manner. I will also be tracking user location using the Google Maps API so I will have to be sure this information is also secured.

User Requirements

Although I may have ideas of what I would like to see implemented in Airport Assistant, I will have to include users throughout the process to ensure that the application is as user friendly as possible.

3. Functional Requirements

3.1 External Requirements

Hardware:

Passengers who wish to use Airport Assistant will need a Android smartphone with a camera so that they can scan the QR code on their boarding pass and the application can read in this information.

Software:

To run the Airport Assistant application the user must be running Android 7.0 or later. This is as the ARCore functionality will not run on devices without this software. These versions of Android supported however, include over one billion Android devices.

Environment:

As this application will be used in the airport it is a requirement that the passenger is connected to the airports wifi or has a strong mobile data connection in the airport. They will also need to connect to the wifi in the airport they are flying to or have roaming data available on their mobile.

3.2 Functional Requirements

3.2.1 Create Account

Description:

When a user wants to create an account on Airport Assistant they will have the option of creating an account using their forename, surname, username, email and password or creating an account using Google or Facebook.

Criticality:

Users must be able to create accounts on the application so that only they can access their private information

Technical Issues:

The create account page must interact closely with the database to create this user in the system.

Dependencies with other requirements:

This functionality does not depend on any other part of the application.

3.2.2 Login**Description:**

When a user wishes to login to the application they will enter their username and password which is then validated with the information in the database.

Criticality:

The login functionality is critical as a user must be able to enter too see their previously uploaded journeys. If users could log into any account they wished, they could obtain other users boarding passes.

Technical Issues:

The login page must interact correctly with the database, so it is essential that the data is correct.

Dependencies with other requirements:

The login functionality depends on the create account functionality. Without this functionality the user would not have an account to log into.

3.2.3 Journey Upload**Description:**

Passengers wish to upload a journey they will take soon. They scan the QR code on their boarding pass which will be used to extract flight and passenger information.

Criticality:

Without this functionality a user could not start a journey so could not use the application to take them through this process through the airport.

Technical Issues:

This functionality must interact with the Google Pay API for Passes to abstract the information and the database to store the information.

Dependencies with other requirements:

This functionality depends on a user being able to login to the application and store their information in a secure account that can only be accessed by one specific user.

3.2.4 AR functionality for checking hand luggage sizes

Description:

The user can use the AR functionality present in the application to check the dimensions of their hand luggage. They do this by scanning the front and side of the luggage using the ARCore functionality present in the application which will be used to measure the depth, width and height of the baggage. This information will then be compared with the dimensions which this airline allow on their aircraft and the user is told whether their baggage is the correct size or not.

Criticality:

This is one of the primary functions involved in reducing the stress levels of passengers moving through the airport. This functionality will not affect the rest of the application but allows users with the piece of mind that their bags are the correct size.

Technical Issues:

This functionality involves the use of a number technologies including Googles ARCore SDK and Web Scraping. The main issue with this functionality will be integrating these technologies together with the database to check the measured sizes to the scraped sizes and then returning a result to the passenger.

Dependencies with other requirements:

This is dependent on a user uploading their journey so that the dimensions of the users specific airline can be scraped and compared with the measurements gathered from the AR scanning of the hand luggage.

3.2.5 Transport to airport

Description:

The purpose of this functionality is to direct users to the airport using their desired method of transport using the Google Maps API.

Criticality:

This is not a crucial functionality as it is the choice of the passenger whether they wish to be directed to the airport or not. This functionality will not affect the rest of the application if a user decides to skip this stage.

Technical Issues:

This functionality will be developed using the Google Maps API. Integrating the application with the Google Maps API will be the toughest technical issue to overcome while developing this feature.

Dependencies with other requirements:

This functionality will not depend on any other features within the application.

3.2.6 Check In

Description:

If a user has previously selected that they will have luggage which they will be checking in, they will be provided information at this stage about checking in this luggage. They will be informed how to check on the board for their check in desk, and what their baggage weighs compared to what their baggage allowance is to reassure the user that they have the correct amount of baggage with them. It also provides them with information on what items they are not allowed in their hold luggage.

Criticality:

This feature is important to ensure that the user checks in their bags weight and is aware of what items they cannot leave in their baggage.

Technical Issues:

This will be developed using web scraping and querying the database.

Dependencies with other requirements:

This functionality will be dependent the users entering information earlier on whether they will be checking in luggage or not and if yes, what weight their hold luggage is.

3.2.7 Security

Description:

This feature will give users information on the process of passing through security. This will include the rules of passing through security, tips and tricks and for certain airports where the information is available, security wait times.

Criticality:

This stage is crucial as all passengers passing through the airport will have to progress through this stage of the airport. This is also one of the most stressful aspects of passing through the airport so will be vital to ensure users can remain calm during the process.

Technical Issues:

Web Scraping will be used to implement this aspect of the application. Not all airports will have security wait times readily available online so I will have to scrape this airport off sites where it is readily available.

Dependencies with other requirements:

This functionality is not dependent on any of the other features present in this application.

3.2.8 Duty Free

Description:

This functionality is split into two sections one being that they can use a map to find certain stores in the Duty free area and also that stores in Duty free can offer advertising using location based services.

Criticality:

It is not crucial as it is an optional part of passing through the airport and can be skipped by users if they wish to continue on to their gate.

Technical Issues:

The Google Maps API will be used to develop this functionality for the application.

Dependencies with other requirements:

Yet again, this functionality will not depend on any other aspects of the application.

3.2.9 Directions to gate

Description:

This feature allows users to check how many minutes walk away their gate is and get directions to their gate. The user will have to manually enter their gate number as this information is not available outside of the airport to be provided with this information.

Criticality:

It is a crucial stage of the process as passengers need to know how far they are from their gate to ensure that they progress to their gate with enough time to board their flight.

Technical Issues:

This functionality will be developed using the Google Maps API so will also need to be integrated with this technology.

Dependencies with other requirements:

This functionality will depend on users entering their gate number so that they can be directed to the correct gate.

3.2.10 Boarding Pass

Description:

The application will at this stage display the customers boarding pass so that they do not have to either print their boarding passes or have to use multiple applications throughout the process.

Criticality:

This functionality is extremely important so that users can present their boarding pass at their gate.

Technical Issues:

The Google Pay for Passes API will be used to present the boarding pass at this phase.

Dependencies with other requirements:

This phase is dependent on a user scanning their boarding pass when they are uploading their journey to the application.

3.2.11 Onboard Information

Description:

The application will display information on how long the journey will take, specific airline information and other important information.

Criticality:

This stage is crucial to keeping users informed while on the flight and hence relaxed while they complete their journey.

Technical Issues:

This feature of the application will be gathered using web scraping.

Dependencies with other requirements:

This functionality will also depend on a user scanning their boarding pass when they upload their journey to the application. This information is necessary so that the application can provide passenger specific information in relation to their airline.

3.2.12 Arrival Information

Description:

When a user lands in their desired location the application will provide them with information on their current location, transport options from this airport and baggage claim information if it is available.

Criticality:

This is crucial as users will be in a new place that they may not be familiar with so users need to be kept informed so that they can remain calm during this process.

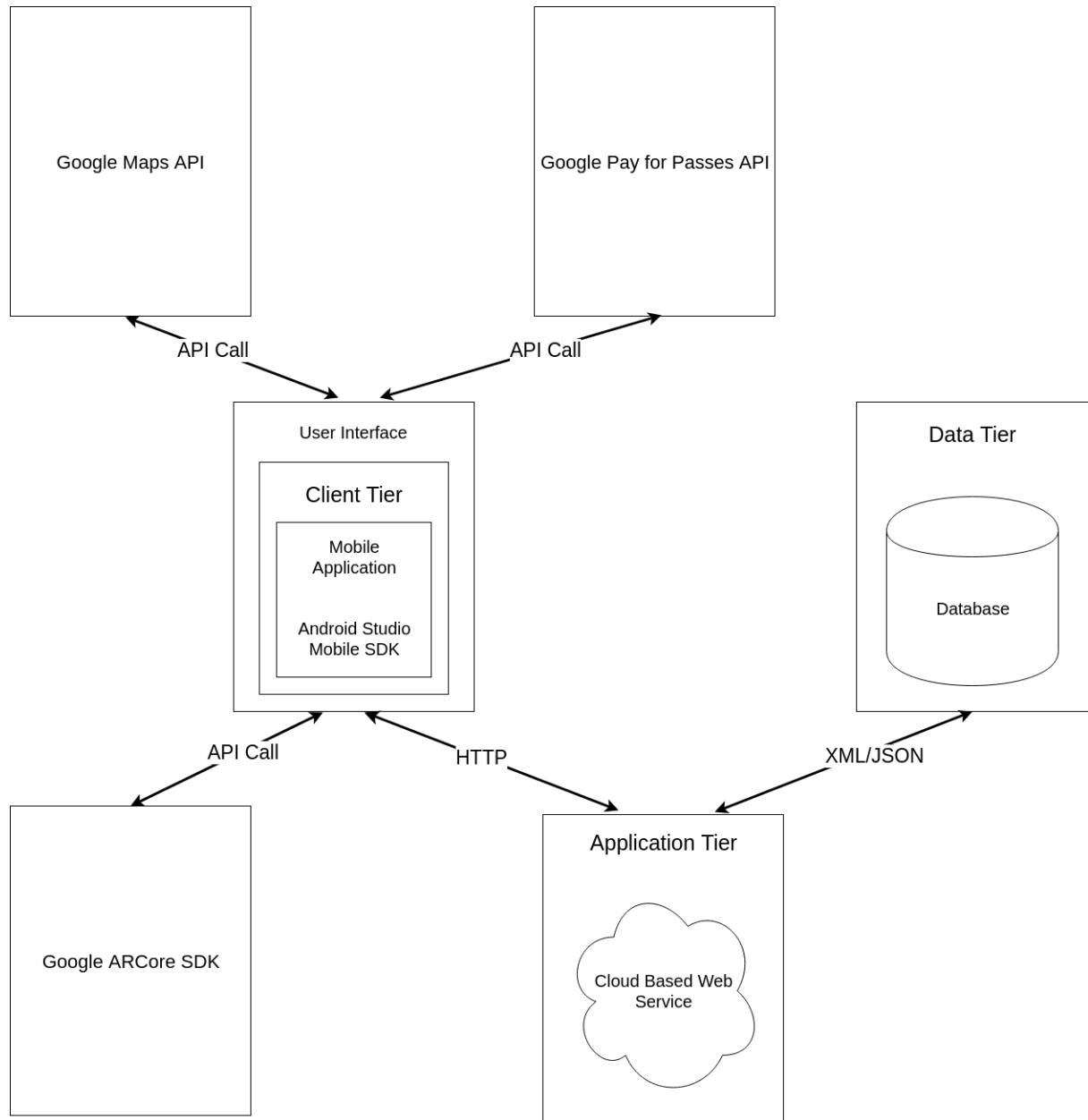
Technical Issues:

This will be developed using Web Scraping and the Google Maps API. Combining these two technologies will be difficult but must be done to achieve this functionality.

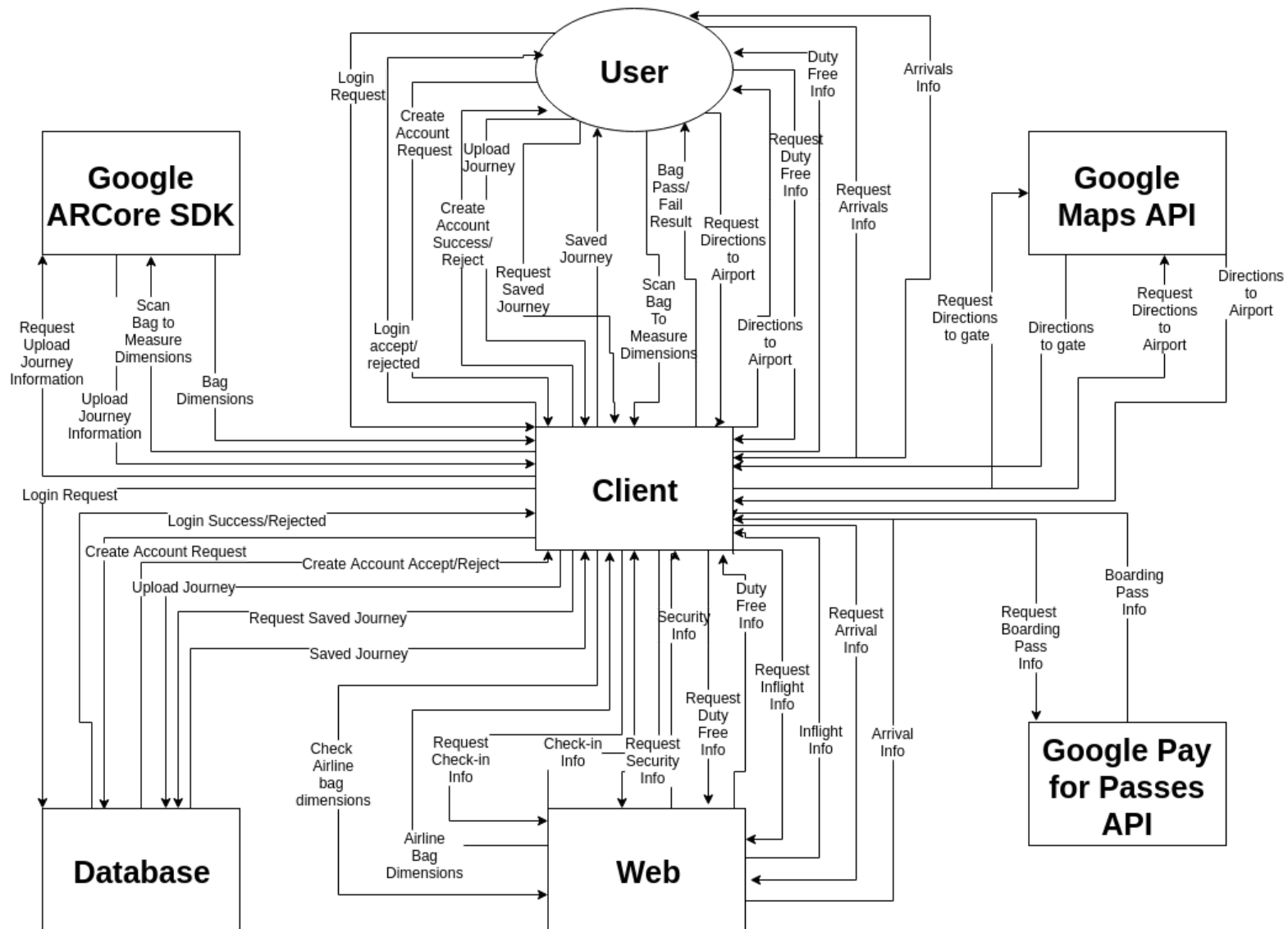
Dependencies with other requirements:

This functionality does not depend on any other features of this application.

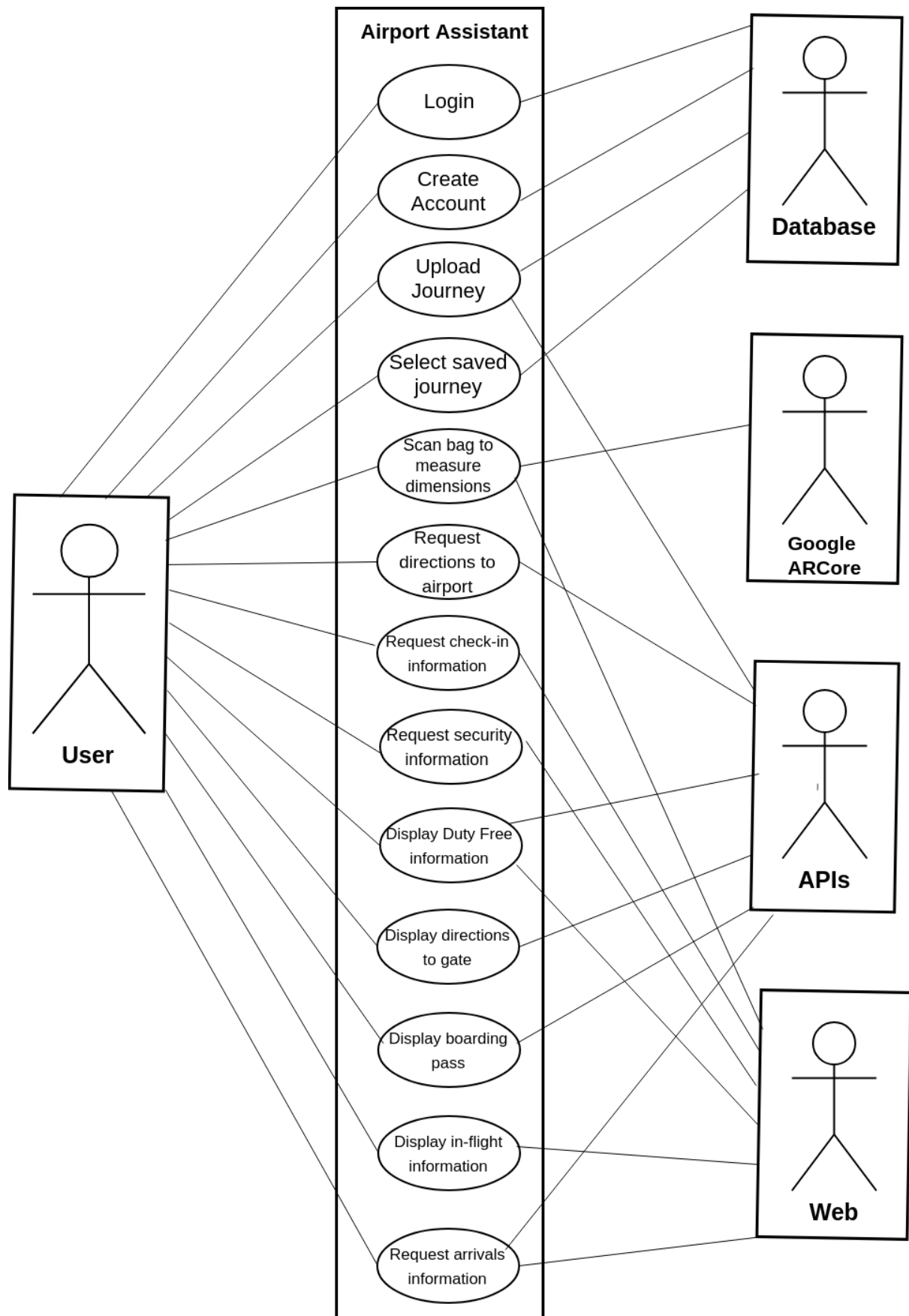
4. System Architecture



Context Diagram



Use Case Diagram



Use Case Descriptions

USE CASE 1	Create Account	
Goal in Context	User enters forename, surname, username, email address and password and creates an account	
Preconditions	User has downloaded the application	
Success End Condition	User has successfully created an account	
Failed End Condition	User cannot create an account	
Primary, Secondary Actors	User Database	
Trigger	User inputs their details at the create account screen	
DESCRIPTION	Step	Action
	1	User is presented with create account screen
	2	User enters their forename, surname, username, email address and password
	3	User clicks create account button
	4	Client verifies user's details.
	5	User has successfully created an account
EXTENSIONS	Step	Branching Action
	4a	User already has an account on the application and is asked to log into this account

USE CASE 2	Login to application	
Goal in Context	User inputs details to access their account	
Preconditions	User has an account on Airport Assistant	

Success End Condition	User has successfully logged in	
Failed End Condition	User fails to log into the application	
Primary, Secondary Actors	User Database	
Trigger	User inputs their details at the login screen	
DESCRIPTION	Step	Action
	1	User is presented with the login screen.
	2	User enters their username and password.
	3	User clicks login button.
	4	Client verifies user's details.
	5	User is successfully logged in.
EXTENSIONS	Step	Branching Action
	2a	User does not have an account and chooses to create an account.
	4a	User enters incorrect login details and is asked to try again.

USE CASE 3	Upload journey to application	
Goal in Context	A user wishes to upload a journey they wish to take to their account.	
Preconditions	User has logged into the application and has a boarding pass with a QR code that they can scan to upload the journey.	
Success End Condition	User has successfully uploaded a journey to the application	
Failed End Condition	User cannot upload a journey to the application	

Primary, Secondary Actors	User Database	
Trigger	User clicks the “Upload Journey” button	
DESCRIPTION	Step	Action
	1	User is on the home screen of the application.
	2	User clicks the “Upload Journey” button
	3	User is presented with a QR scanner
	4	They must scan the QR code on their boarding pass
	5	The journey will then be uploaded to the application and presented on the home screen
EXTENSIONS	Step	Branching Action
	4a	If the boarding pass is out of date the journey will not be uploaded

USE CASE 4	Select saved journey	
Goal in Context	A user wants to begin a journey which they have previously uploaded	
Preconditions	User has logged into the application and uploaded a journey successfully	
Success End Condition	User successfully starts a journey	
Failed End Condition	User cannot start a journey.	
Primary, Secondary Actors	User Database	
Trigger	User selects a specific journey to begin.	
DESCRIPTION	Step	Action

	1	User selects the journey they wish to begin
	2	User is asked to confirm that this is the journey they wish to begin
	3	User selects to confirm this choice
	4	User successfully starts the journey of their choice
EXTENSIONS	Step	Branching Action
	1a	User's journey has not yet been uploaded so they must select to "Upload Journey"
	3b	User has selected an incorrect journey and selects to cancel the journey selection.

USE CASE 5	User scans hand luggage to check bag sizes	
Goal in Context	A user wishes to check the size of their hand luggage so they know if it is the correct size	
Preconditions	User has successfully started a journey	
Success End Condition	User has successfully checked the dimensions of their hand luggage and passed this check	
Failed End Condition	User cannot measure their bag or fails the hand luggage size check	
Primary, Secondary Actors	User ARCore Web Scraper	
Trigger	User has started a journey	
DESCRIPTION	Step	Action
	1	User selects to check the size of their hand luggage
	2	The AR camera opens and the user scans the front and side of their bag to check the dimensions
	3	Users bag dimensions are checked against their airline requirements.
	4	Users are informed that their baggage has passed the

		baggage check
	5	Users progress onto the next stage successfully
EXTENSIONS	Step	Branching Action
	4a	A user's baggage fails the baggage check. The user is then asked to re-check their baggage or move onto the next stage.

USE CASE 6	User gets directions to airport	
Goal in Context	User wishes to get directions to the airport from their current location	
Preconditions	User has successfully started a journey and has progressed through the previous stages.	
Success End Condition	User has successfully been directed to the airport	
Failed End Condition	User has not been directed to the airport	
Primary, Secondary Actors	User Google Maps API	
Trigger	User selects that they wish to be directed to the airport	
DESCRIPTION	Step	Action
	1	User selects that they wish to be directed to the airport
	2	User is directed to the airport using the Google Maps API
	3	User arrives at the airport
	4	They move onto the next stage successfully
EXTENSIONS	Step	Branching Action
	1a	User does not wish to be directed to the airport and progresses to the next stage of the application

USE CASE 7	Check-in Information	
Goal in Context	User wishes to view check in information on the application	
Preconditions	User has began and journey and completed all steps up to this phase	
Success End Condition	The user's information need about the check in process has been satisfied	
Failed End Condition	The user is lacking information about the check in process	
Primary, Secondary Actors	User Web	
Trigger	User completes previous stage successfully.	
DESCRIPTION	Step	Action
	1	User is displayed information on their check-in gate and information regarding check-in bags
	2	User chooses to progress on from this stage when they are finished with the check-in process

USE CASE 8	Security Information	
Goal in Context	Application displays security information to the user.	
Preconditions	User has progressed through all previous stages of the journey.	
Success End Condition	User has passed through security successfully and their information need is met.	
Failed End Condition	User is unable to pass through security as they are lacking information from the application.	
Primary, Secondary Actors	User Web Scraper	
Trigger	User has completed the previous stages in the journey.	
DESCRIPTION	Step	Action

	1	User is displayed information about the security process
--	---	--

USE CASE 9	Duty Free Information	
Goal in Context	User wishes to view information about Duty Free, view a map of Duty Free or search where stores in are.	
Preconditions	User has progressed through all the previous stages of the journey process.	
Success End Condition	User obtains all the information they are looking for about the Duty Free process.	
Failed End Condition	User is left wanting information about the Duty Free Process	
Primary, Secondary Actors	User Google Maps API Web Scraping	
Trigger	User has passed through the security section of the application	
DESCRIPTION	Step	Action
	1	User is presented with the Duty Free area of the application
	2	The user can search for a specific store or view offers which may be available at the Duty Free.
	3	Once a user searches a store they will be presented with the Google Maps API where they will be given directions to a specific store.
EXTENSIONS	Step	Branching Action
	3a	User may choose not to search a specific store and skip through this functionality.

USE CASE 10	Display directions to gate	
Goal in Context	User wishes to be directed to their gate and find out how far from their gate they are.	

Preconditions	User has progressed through previous stages of the journey process and has entered their gate number.	
Success End Condition	User has been successfully directed to their gate with sufficient assistance from the application.	
Failed End Condition	The user has not been directed to their gate.	
Primary, Secondary Actors	User Google Maps API	
Trigger	User has entered their gate number and confirms it.	
DESCRIPTION	Step	Action
	1	User enters their gate number which is displayed on the departures board in the airport.
	2	The user is informed how far their gate is from their current location.
	3	They select to view this information on a map.
	4	The user is then directed to the gate using the Google Maps API.
EXTENSIONS	Step	Branching Action
	3a	The user can select not to be directed to the gate and progress onto the following stage of the journey.

USE CASE 11	Display Boarding Pass
Goal in Context	User has arrived at their gate and wishes to display their boarding pass to get onto their flight.
Preconditions	User has progressed through the earlier stages of the journey process.
Success End Condition	User can display their boarding pass successfully.

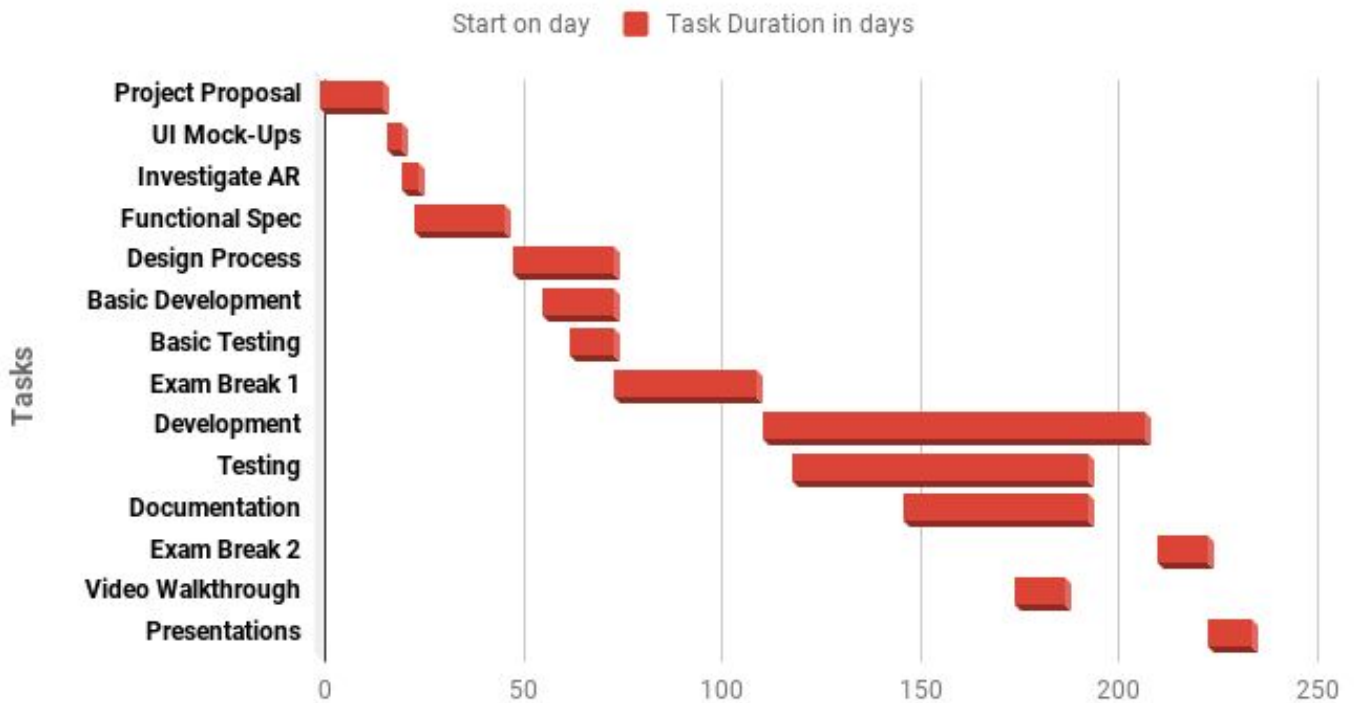
Failed End Condition	User cannot display their boarding pass on the application.	
Primary, Secondary Actors	User Google Pay for Passes API Database	
Trigger	User has progressed to this stage after completing the earlier journey stages.	
DESCRIPTION	Step	Action
	1	Application displays the boarding pass.
	2	User selects to move onto the next stage of the journey process.

USE CASE 12	Display In-Flight Information	
Goal in Context	User is now aboard the airplane and requires offline information about the journey they are about to take.	
Preconditions	User has progressed through previous stages of the journey process.	
Success End Condition	User is successfully presented with inflight information	
Failed End Condition	User is not presented with the information on the flight.	
Primary, Secondary Actors	User Database Web Scraping	
Trigger	User has completed the boarding pass stage of the journey.	
DESCRIPTION	Step	Action
	1	User is displayed with inflight information about their journey.
	2	User progresses onto the next stage once they have completed their flight.

USE CASE 13	Display Arrivals Information	
Goal in Context	User has arrived in their desired location and wished to receive transport information about their current airport.	
Preconditions	User has completed all previous steps in the journey process.	
Success End Condition	User has all obtained all information which they wish about the transport in their current location.	
Failed End Condition	User has not obtained there desired information about their current location.	
Primary, Secondary Actors	User Google Maps API Web Scraping	
Trigger	User has progressed through the previous stage of the application.	
DESCRIPTION	Step	Action
	1	User is welcomed to the country they are currently present in.
	2	User is shown a list of transport options for their current airport.
	3	User is then displayed with the Google Maps API to direct them to their destination in their desired mode of transport.
	4	User reaches their desired location
EXTENSIONS	Step	Branching Action
	3a	User wishes to not be transported to their destination and skips this stage

6. Preliminary Schedule

Start date and Task durations



Critical Tasks			
Critical Task Name	Start Date	End Date	Duration (days)
Project Proposal	08/10/2018	24/10/2018	16
Draw up User Interface Mockups	25/10/2018	29/10/2018	4
Investigate AR functionality	29/10/2018	01/11/2018	4
Functional Specification	01/11/2018	23/11/2018	23
Design Process	26/11/2018	21/12/2018	25
Basic Development	03/12/2018	21/12/2018	18
Basic Testing	10/12/2018	21/12/2018	11
Exam Break 1	21/12/2018	26/01/2019	36
Development	28/01/2018	04/05/2019	96
Testing	04/02/2019	20/04/2019	75

Documentation	04/03/2019	20/04/2019	47
Exam Break 2	07/05/2019	20/05/2019	13
Video Walkthrough	01/04/2019	14/04/2019	13
Presentations	20/05/2019	01/06/2019	11

7. Appendices

Firebase API

- <https://firebase.google.com/>

Google ARCore SDK

- <https://developers.google.com/ar/>

Google Maps API

- <https://cloud.google.com/maps-platform/>

Google Pay for Passes API

- <https://developers.google.com/pay/passess/>

Dublin Airport Site

- <https://www.dublinairport.com/>