CA400 Airport Assistant
Technical Specification

Name: Shauna Moran

Student Number: 15381166

Supervisor: Ray Walshe

Abstract:

Airport Assistant is an Android Application to help users, primarily those who suffer from anxiety, with dealing with passing through an 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 that 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 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 there airline before the leave home. Similarly, features such as security tips and walk times to gates will help make the entire process less stressful.

Table of Contents

1. Introduction

1.1. Overview

Airport Assistant is an Android Application to help users, primarily those who suffer from anxiety, with dealing with passing through an 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 that 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 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 there airline before the leave home. Similarly, features such as security tips and walk times to gates will help make the entire process less stressful.

1.2. Glossary

- **AR**
  - Augmented reality (AR) is an interactive experience of a real-world environment where the objects that reside in the real-world are "augmented" by computer-generated perceptual information.
- **Accessible**
  - Easy to approach, reach, enter, speak with, or use.
- **Integration**
  - The act of combining or adding parts to make a unified whole.
- **Duty Free**

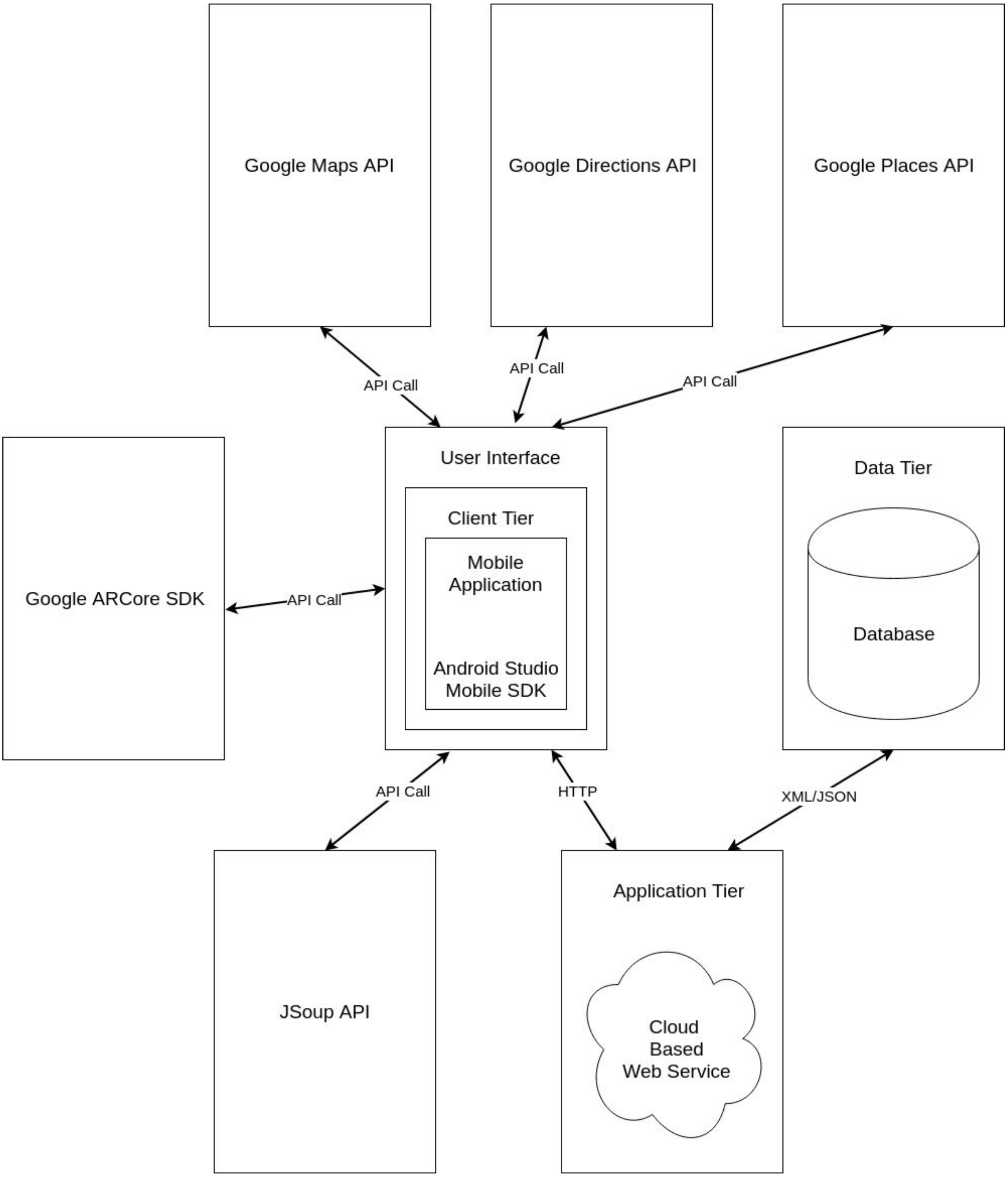
    - Duty-free shops (or stores) are retail outlets that are exempt from the payment of certain local or national taxes and duties, on the requirement that the goods sold will be sold to travelers who will take them out of the country.
- **Check-in**
    - The check-in process at airports enables passengers to check in luggage onto a plane.
- **Airport Security**
    - Airport security refers to the techniques and methods used in an attempt to protect passengers, staff, aircraft, and airport property from accidental/malicious harm, crime, and other threats.

2. System Architecture

2.1. Development Environment

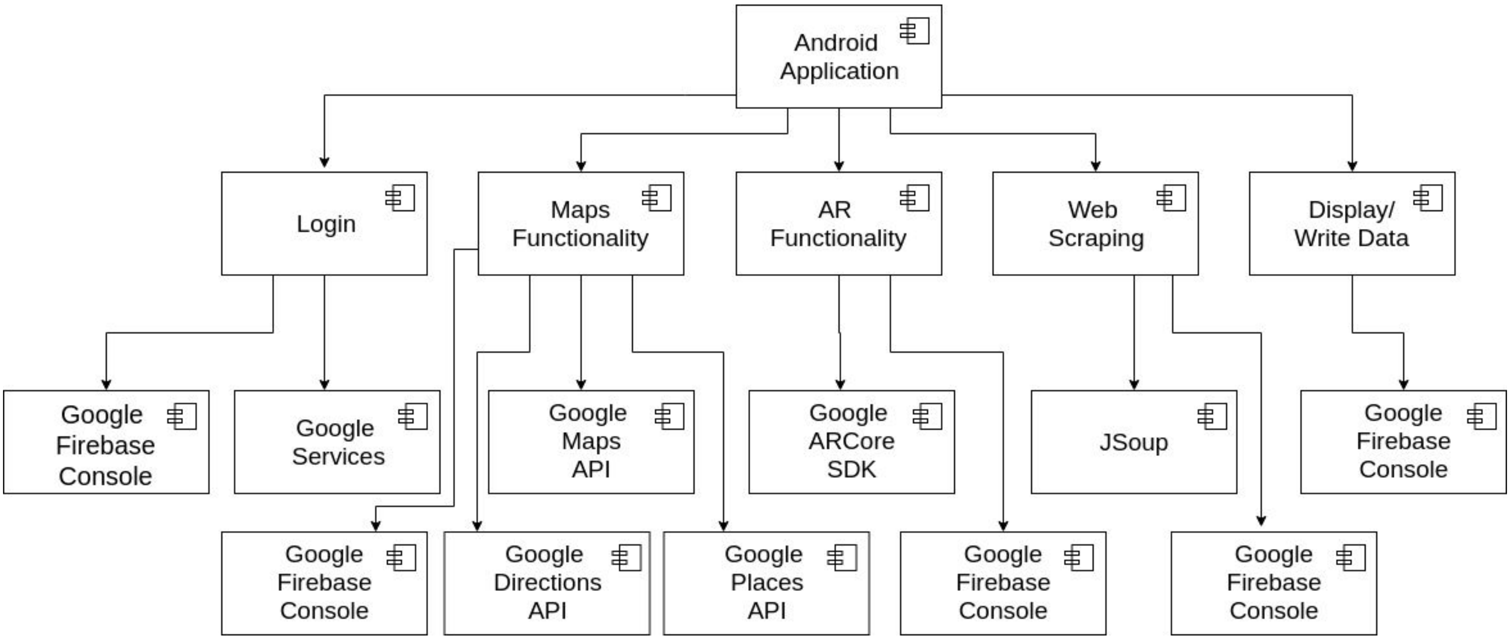
Airport Assistant was developed using Intelliji Ultimate version XXXXX. The application targets API level 27. This is due to the Google Sceneform requiring this API level. Devices must operate on this API to be allowed to utilise the emerging AR technology which requires the most up to date Android version to achieve it's groundbreaking functionality. It would not be feasible to develop this project on a lower API level if a user wished to utilise this functionality. This application consists of XXX classes and XXX XML files.

2.2. System Architecture Diagram

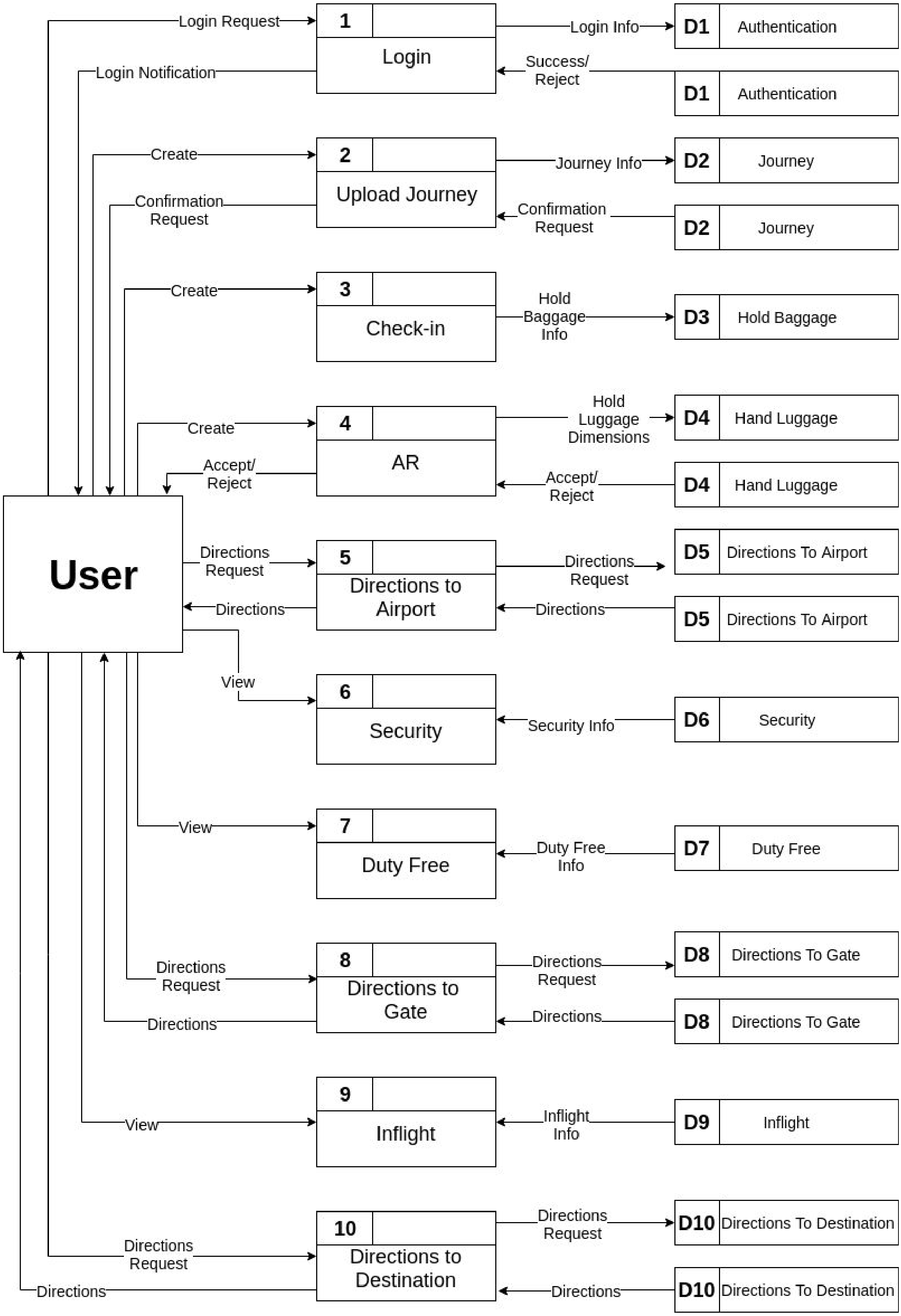


3. High Level Design

3.1. Component Model



3.2. Data Flow Diagram

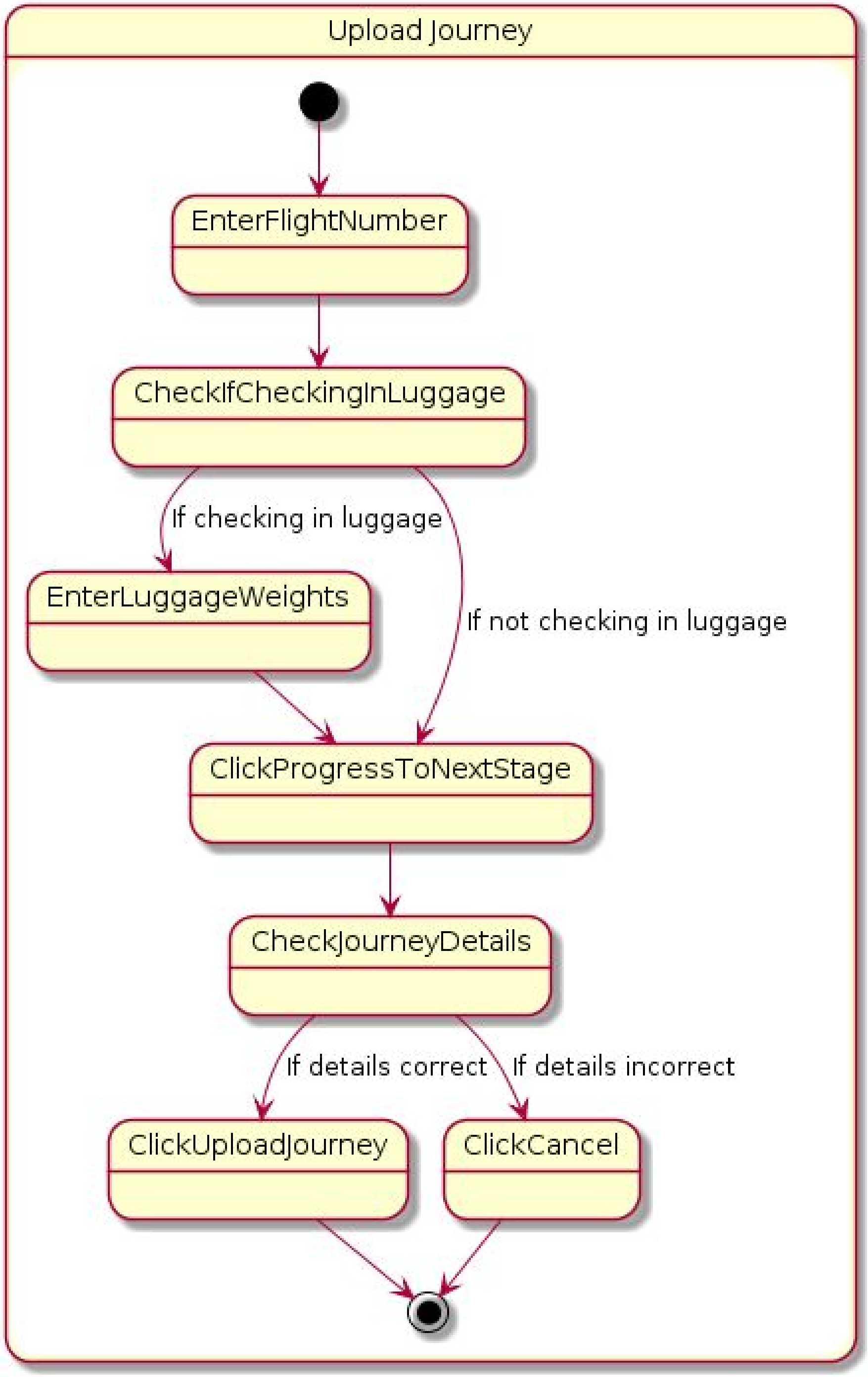


3.3. Class Diagram

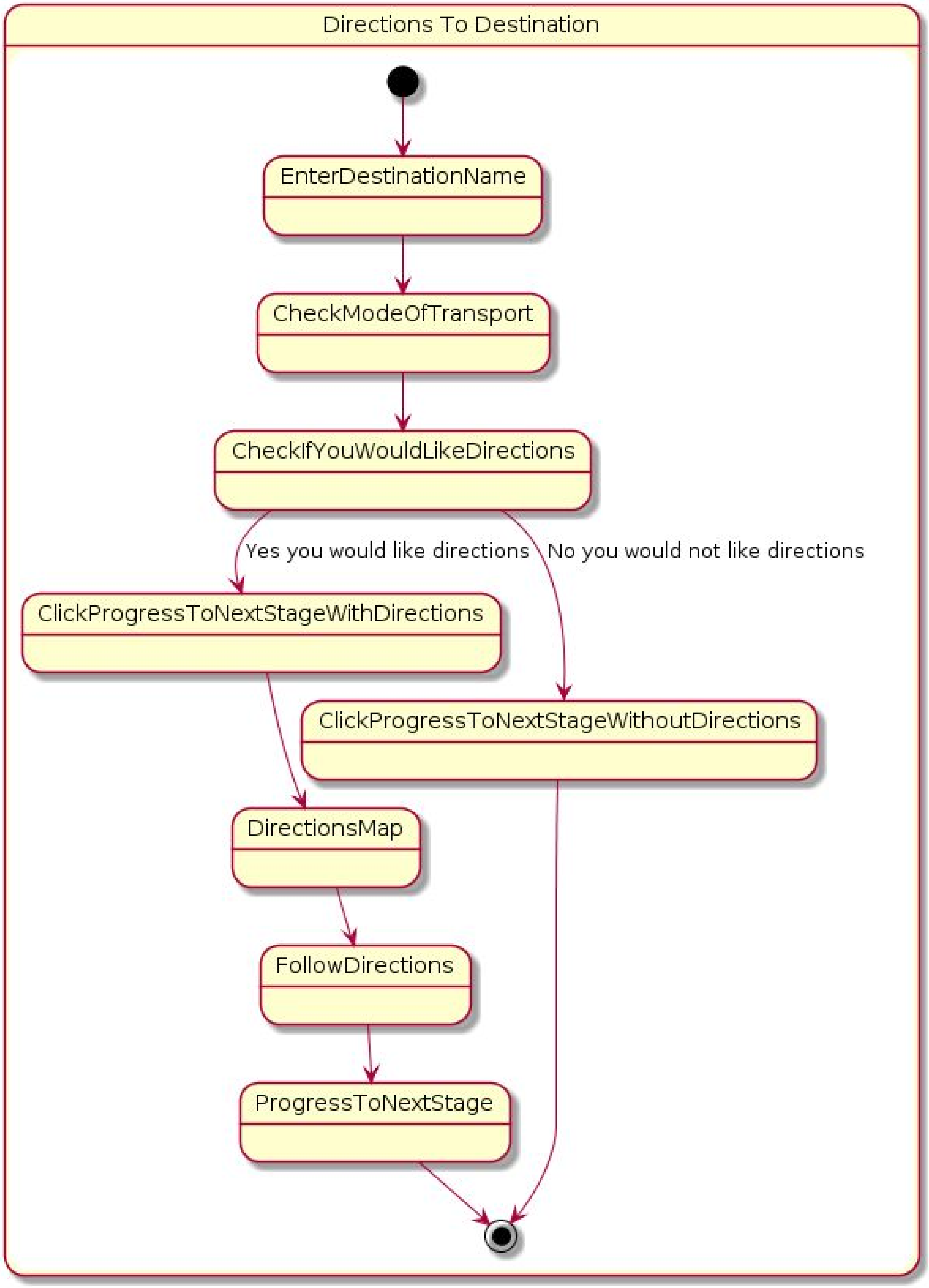
3.4. Object Diagram

3.5. State Machine

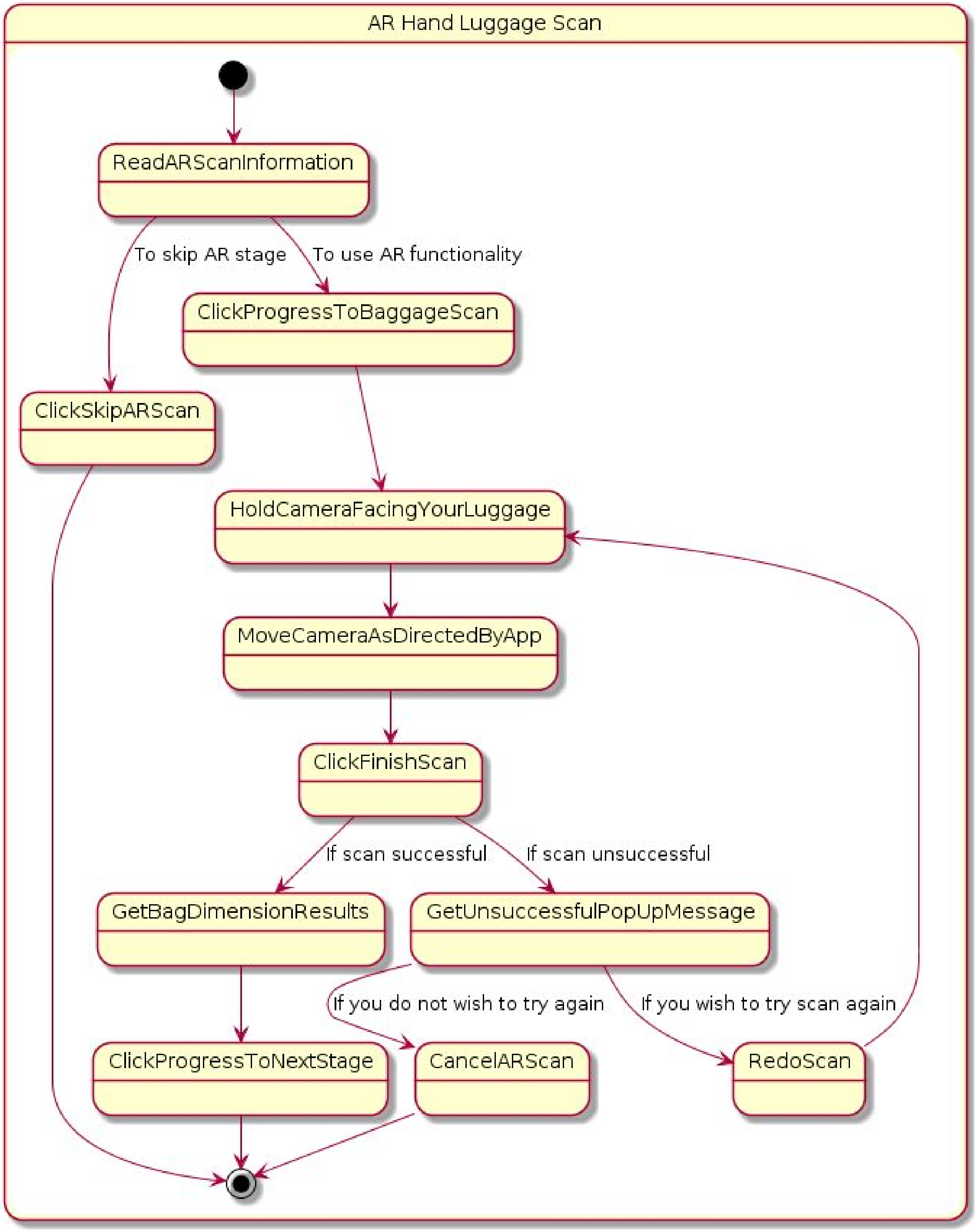
Upload Journey



Directions To Destination

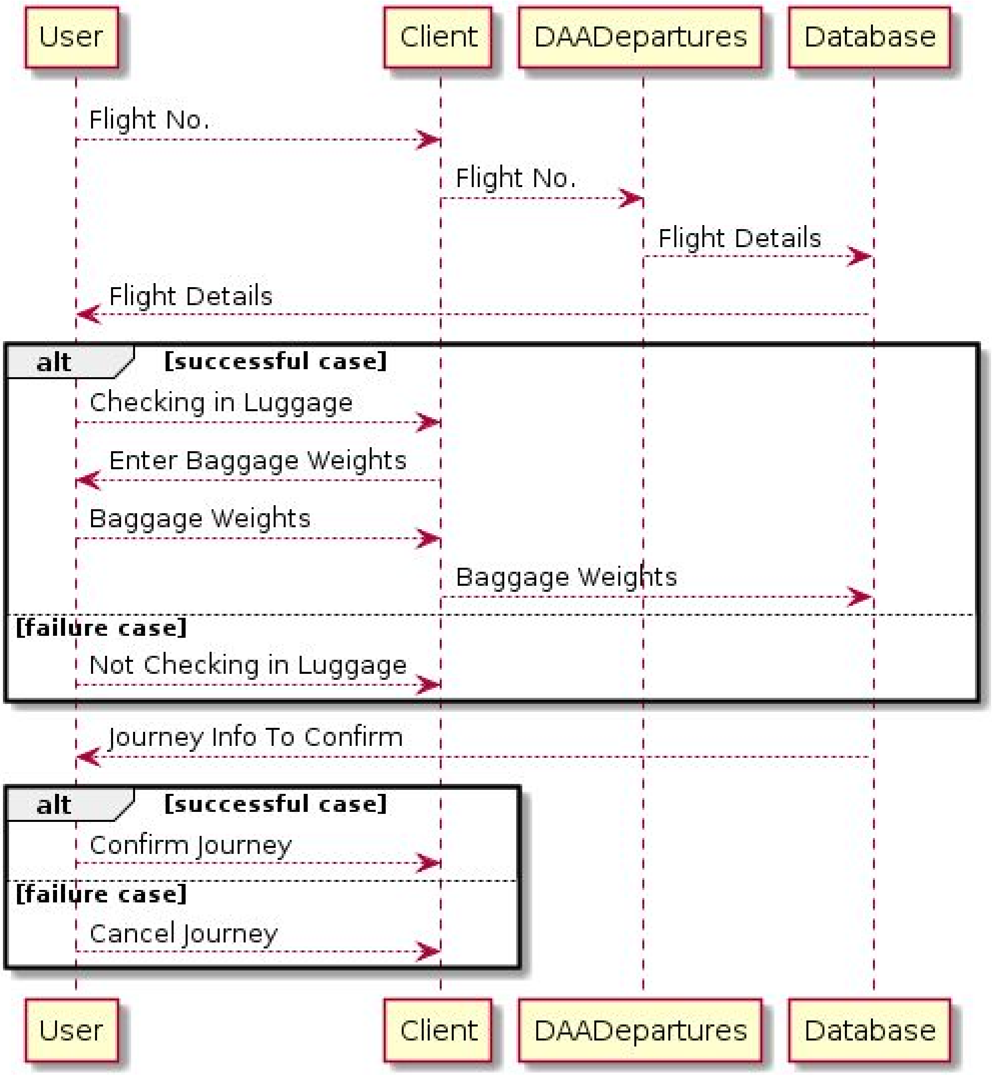

Directions To Destination
EnterDestinationName
CheckModeOfTransport
CheckIfYouWouldLikeDirections
Yes you would like directions
No you would not like directions
ClickProgressToNextStageWithDirections
ClickProgressToNextStageWithoutDirections
DirectionsMap
FollowDirections
ProgressToNextStage

AR Hand Luggage Scan

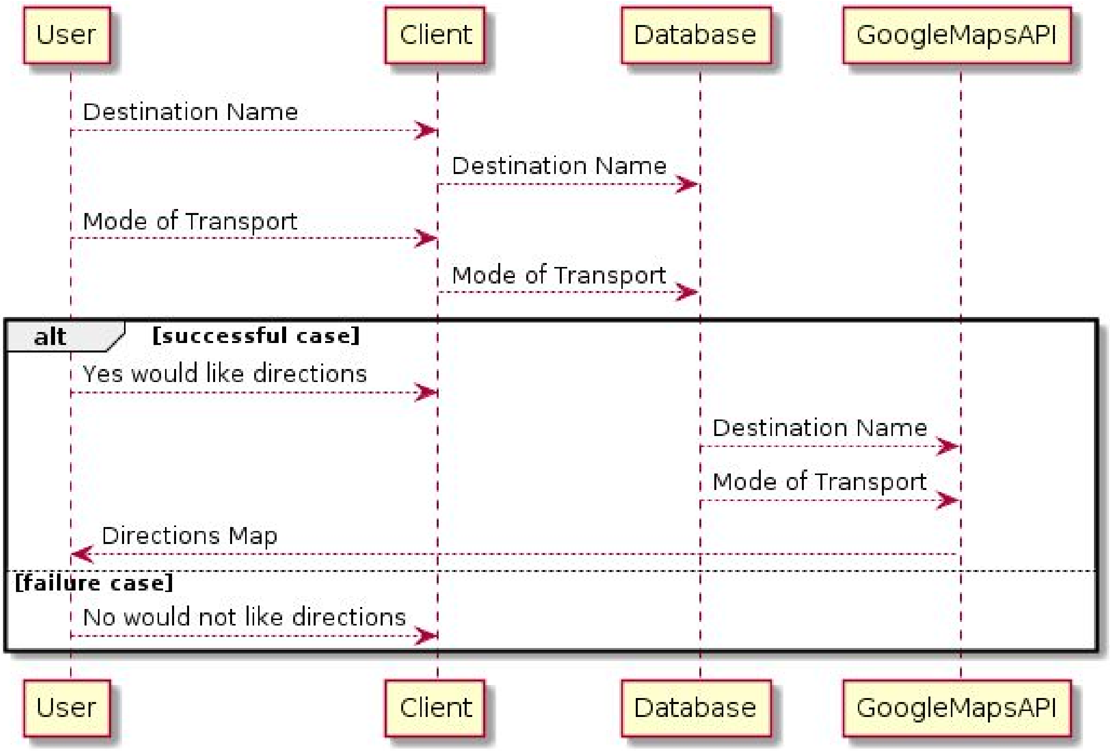


3.6. Sequence Diagram

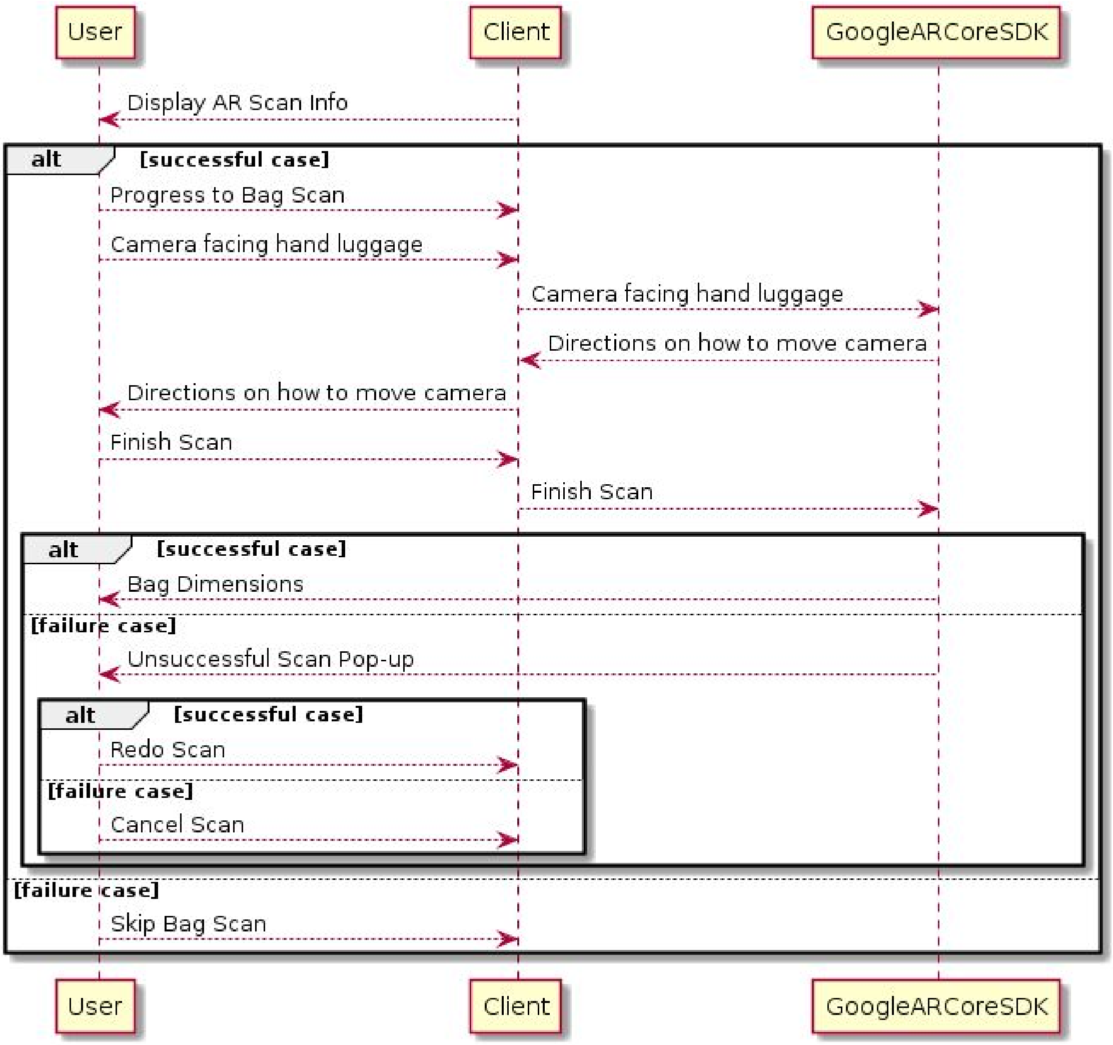
Upload Journey



Directions To Destination



AR Scan Info


User
Client
GoogleARCoreSDK
Display AR Scan Info
alt
[successful case]
Progress to Bag Scan
Camera facing hand luggage
Camera facing hand luggage
Directions on how to move camera
Directions on how to move camera
Finish Scan
Finish Scan
alt
[successful case]
Bag Dimensions
[failure case]
Unsuccessful Scan Pop-up
alt
[successful case]
Redo Scan
[failure case]
Cancel Scan
[failure case]
Skip Bag Scan
User
Client
GoogleARCoreSDK

4. Problems and Resolutions

- AR Sceneform Project Required API level
- AR beta version
- Google Pay for Passes API

AR Sceneform Project Required API level

Problem

Unfortunately, the AR Sceneform project which is used for working with the Google ARCore SDK requires Android API level 27. I have a Samsung S8 which is currently at API level 25 and my laptop is unable to run an emulator. This means I must use a device with the latest form of android. As of March this has been released in Ireland on the Samsung S9 but not yet on the S8. It is expected to be released in the next month.

I decided to use the Google Sceneform project as I believe it's the best way to progress with my project as it assists with certain aspects of working with ARCore.

Solution

To combat this issue I spent some time running the application on a friend's phone until it was possible to do so on my own. This limited me to running my application at certain times. I spent other times working on testing and documentation as to not waste time.

AR beta version

Problem

I ran into a number of issues using the Google ARCore SDK. Due to how new of a technology AR is, ARCore is still in its beta version. This means that there are a number of bugs present in the software that are still being flattened out. Unfortunately, working with a software that is in beta means that one is bound to run into such issues.

Solution

I started working with this technology early in the development process so had the time to work through such problems.

Google Pay For Passes Issue

Problem

Google Pay for Passes did not work as first expected. I initially expected it to act like the Google Pay API which can be integrated into applications. Unfortunately, the purpose of the API is to be built into applications, such as the Ryanair app, so that users can save their boarding pass to the Google Pay API and not for displaying boarding passes as I had first expected. This creates two issues. Firstly, I cannot display boarding passes and secondly, scanning boarding passes was how I intended to take in a user's flight information.

Solution

To resolve this issue I will ask a user the enter their flight number. I will then scrape the flight information from the Dublin Airport Departures site. This will mean that users will not have to enter all of their flight details which results in the application being easier to use. Unfortunately, I will not be able to display boarding passes in Airport Assistant.

5. Installation and Configuration

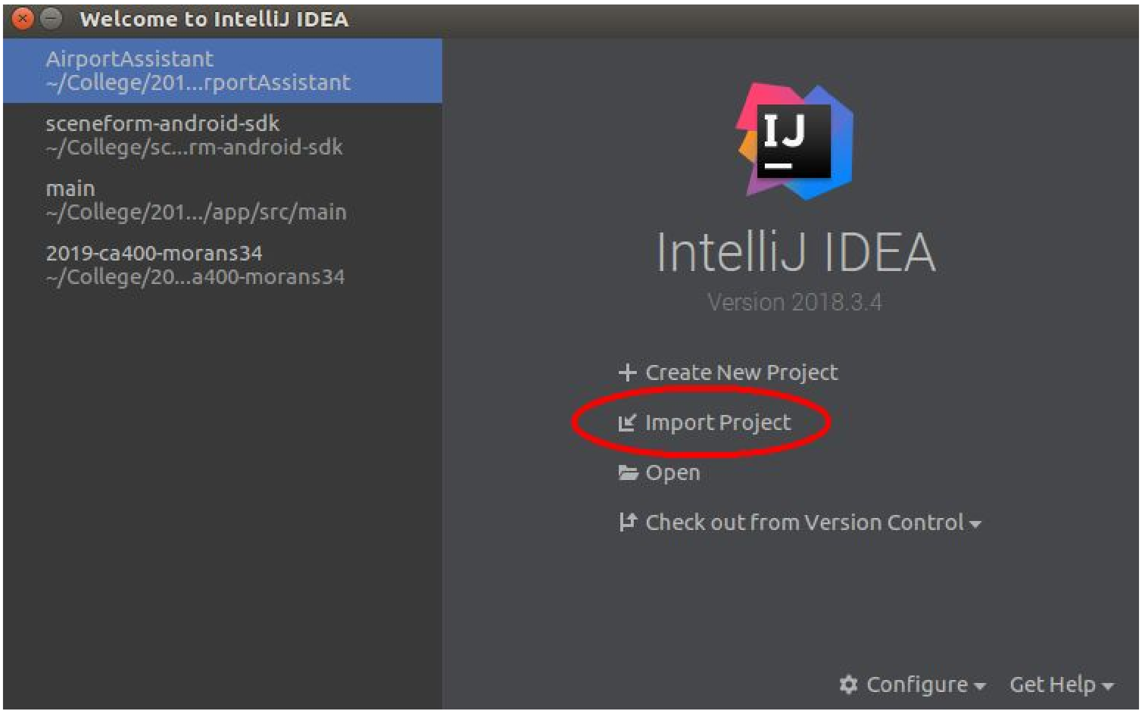
Required Software

- Intelliji Ultimate version XXX
- Java 8

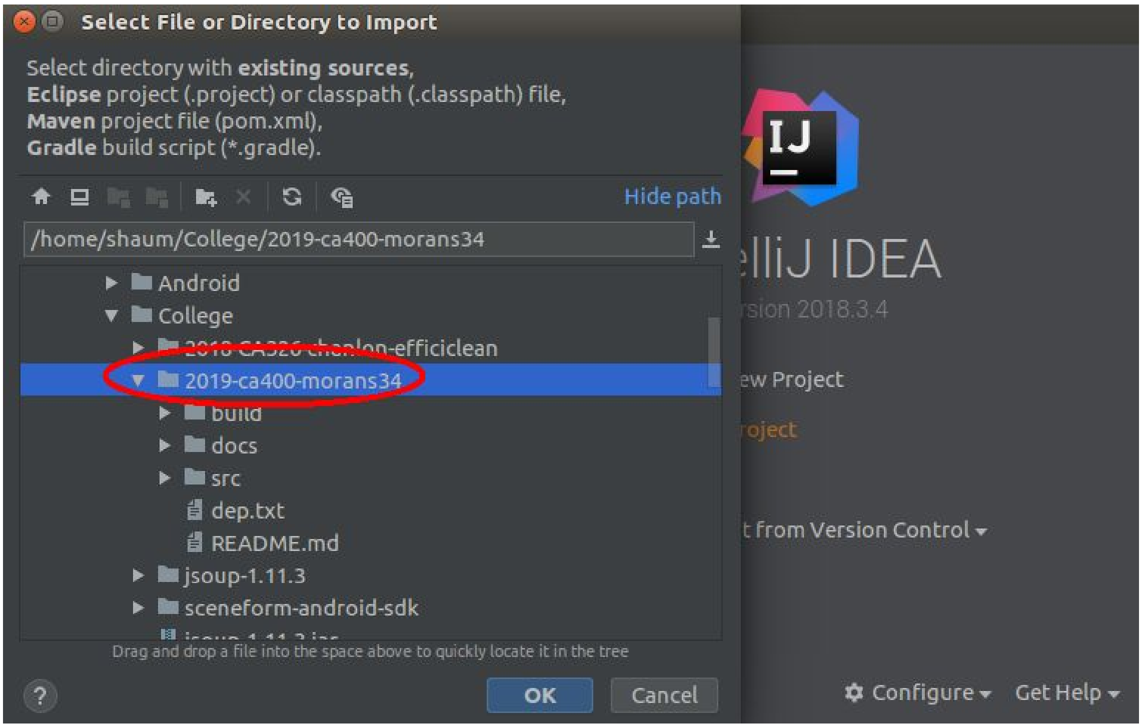
Instructions

Building the Android Project

- Create a folder where you would like the project to be stored
- Open a terminal
- Navigate to chosen folder
- Execute command: git clone https://gitlab.computing.dcu.ie/morans34/2019-ca400-morans34.git
- Open Intelliji
- Select Import Project as seen below



- Navigate to “C:\Users”YOUR CREATED DIRECTORY PATH” \2019-CA400-morans34”



Select File or Directory to Import

Select directory with **existing sources**,
Eclipse project (.project) or classpath (.classpath) file,
Maven project file (pom.xml),
Gradle build script (*.gradle).

/home/shaum/College/2019-ca400-morans34

Drag and drop a file into the space above to quickly locate it in the tree

- Click “Ok”
- The project will now build

Running the Application

- Connect your device to your laptop using a USB cable
- On the Toolbar at the top of the page click "Run"
- Select "Run App"
- Select your device
- Click "Ok"
- The application will now run