



Bilkent University

---

Department of Computer Engineering

# Senior Design Project

*Project short-name: LodeStar*

## Analysis Report

Barış Poyraz, Berk Evren Abbasoğlu, Çelik Köseoğlu, Efe Ulaş Akay Seyitoğlu, Hüseyin Beyan

Supervisor: Halil Bülent Özgür

Jury Members: Çiğdem Gündüz Demir and Uğur Güdükbay

Progress Report

Nov 6, 2017

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/1.

# Contents

1. Introduction	1
2. Current System	2
3. Proposed System	3
3.1.Overview	4
3.2.Functional Requirements	5
3.2.1.Data Resources	5
3.2.2.User-Specific Requirements	7
3.3.Non-Functional Requirements	8
3.3.1.Reliability	8
3.3.2.Usability	8
3.3.3.Accessibility	9
3.3.4.Extensibility	9
3.3.5.Portability	10
3.3.6.Efficiency	10
3.4.Pseudo Requirements	10
3.5.System Models	10
3.5.1.Scenarios	10
3.5.2.Use Case Model	24
3.5.2.1.Login Use Case	26
3.5.2.2.Create Account Use Case	27
3.5.2.3.Create Account Using Google Use Case	28
3.5.2.4.Create Account Using Facebook Use Case	28
3.5.2.5.Change Account Settings Use Case	29
3.5.2.6.Change Account Privacy Preferences Use Case	30
3.5.2.7.Select the City of Destination Use Case	31
3.5.2.8.Travel the Destination City Use Case	31
3.5.2.9.Select the Destination Airport Use Case	32
3.5.2.10.Travel the Destination Airport Use Case	32
3.5.2.11.Scan Flight Ticket Barcode Use Case	32
3.5.2.12.View Destination Weather Information Use Case	33
3.5.2.13.View Flight Information Use Case	34
3.5.2.14.View Flight Information Use Case	34
3.5.2.15.Manually Enter Flight Information Use Case	35
3.5.2.16.View Transportation Suggestions Use Case	35

3.5.2.17.Flight Time Notification Use Case	36
3.5.2.18.During Flight Virtual Exploration Use Case	36
3.5.2.19.During Flight Virtual Exploration Use Case	37
3.5.2.20.During Flight Reading Exploration Use Case	37
3.5.2.21.View Destination City Videos Use Case	38
3.5.2.22.View City Touristic Spots Use Case	38
3.5.2.23.View City Accommodation Use Case	39
3.5.2.24.View City Events Use Case	39
3.5.2.25.View Comments of Places Use Case	40
3.5.2.26.Enter Comment for Places Use Case	40
3.5.2.27.View Living Expenses Use Case	41
3.5.2.28.View Currency Rates Use Case	41
3.5.3.Object and Class Model	42
3.5.4.Dynamic Models	44
3.5.4.1.Sequence Diagrams	44
3.5.4.2.Activity Diagrams	48
3.5.5.User Interface	50
3.5.5.1.Login Page	50
3.5.5.2.Sign Up Page	51
3.5.5.3. Home Page	52
3.5.5.4.Trip Page	53
3.5.5.5.Weather Information Page	55
3.5.5.6.Flight Information Page	56
3.5.5.7.Currency Rates Page	57
3.5.5.8.Me Page	58
3.5.5.9.Preferences Page	59
4. References	60

# 1. Introduction

In today's world, what is the most valuable currency? American Dollars? Gold? Airline or credit card points? Even though they all offer value in their own ways, unarguably the most important currency for everyone is time. So, saving time should offer the greatest profit. Especially where people are forced to waste time due to unavoidable circumstances. One such circumstance is waiting at the airport, possibly for hours at a time.

Imagine booking a flight to a foreign country. The day of the exciting getaway comes knocking on the door. As the punctual person you are, you go to the airport hours before the departure time. You check-in, hand over your luggage, go through passport control and sit next to your gate, waiting for the flight. You check your watch, and then monitor the expected departure time of your flight. Which is two hours away... To make matters worse, it has also been delayed for another hour and a half. The thrill of the holiday hides behind the curtains of a dreaded airport wait. Sounds familiar?

LodeStar aims to help such passengers to utilize this otherwise wasted time efficiently. Thanks to the development of technology, it is now possible to obtain information about most countries. Favorite places to visit, must-see events, historic restaurants and so forth. While such information is already presented in many popular applications already, LodeStar diverges from such applications with one unique feature. With the help of Google Street View, LodeStar will offer users the 360° views of the airport they will be traveling to. Hours before reaching their destinations, the users will be able to see where they can buy a SIM card, rent a car, get on the train, or claim their luggage. LodeStar will show them how to go from the landing site to the mentioned places with directions, and the 360° pictures will etch the path into their minds. If the user possesses virtual reality glasses, these images will also be displayed in virtual reality for an immersive experience.

This report's departure point is an overview of the development strategy of Project LodeStar. Information about the current technologies will be given as well as the difference LodeStar will make in contrast to these ones. Functional and non-functional requirements will be listed. The system models of LodeStar will be presented. Some use cases and possible scenarios will be explained. Following this, object models of the planned system and activity diagrams will be presented. At the end of the report, LodeStar's user interface mockups will be showcased.

## 2. Current System

There are already a few applications available for LodeStar to fetch data from. Below, you can find such applications, their details, and how LodeStar will incorporate their functions to its own.

### FourSquare<sup>[1]</sup>

- shows trending, popular, or interesting places nearby
- allows users to comment on and rate these places
- allows users to check-in to these places

### Google Street View<sup>[2]</sup>

- shows 360° images of many places around the world
- allows for quick travel between nearby places, like walking down a road

### Google Maps<sup>[3]</sup>

- displays roads and streets of many places around world in map form
- calculates distance between two places
- calculates viable routes between two places
- calculates the time required to reach from one place to another depending on means of transport

### Trip Advisor<sup>[4]</sup>

- gives recommendations on travel ideas
- recommends places to eat, spend time at, and visit
- shows major attractions of a city

### Yelp<sup>[5]</sup>

- shows places like cafes and restaurants nearby
- allows users to comment and rate these places
- gives information about the food offered in a specific place

### **Weather.com<sup>[6]</sup>**

- shows weather conditions in all cities around the world in real time
- shows likelihood of rain, snow etc.
- shows the sunset and sunrise hours

### **Google CardBoard<sup>[7]</sup>**

- allows a simple VR experience via a smart phone
- unlocks the full potential of VR based applications

### **FlightAware<sup>[8]</sup>**

- gives information about ongoing flights
- shows departure and arrival points
- lists the duration of flights

Instead of switching between lots of websites and applications to gather simple but distinct information, LodeStar will offer all of them under one roof. Having to keep track of many different things across many different applications can easily become tiresome. Users might also forget the information they receive more quickly. By providing all the information in one common, united application, LodeStar will ensure the user can get what he wants immediately, and easily.

To build upon these listed features of already established applications, LodeStar adds a few more. For instance, guiding a user to the a SIM card store or rent-a-car store is nonexistent in any preceding application. Moreover, there aren't any applications which explains the user how to get a ticket for bus or subway upon arrival, and which method to choose according to his/her needs and constraints. With a unified source for vital information, LodeStar will be the application for all who travel.

## **3. Proposed System**

In this section, the overview of the LodeStar application will be presented. On top of that, the necessary requirements (functional & non-functional) for the application will be specified.

### **3.1.Overview**

When traveling by plane, especially internationally, customers must go to the airports early in order to check-in, register their luggage, go through customs etc. Since any or all of these things could potentially demand a lot of time due to queues or poor management, most people go to the airport very early, and have to wait a long time before boarding the plane. With LodeStar, we aim to help people value their time which would otherwise be wasted at the gates of airports by giving them a glance of the destination airport and city. This should also help alleviate some stress as the passengers will have a clearer idea of what to expect upon arrival.

Some of LodeStar's features are available in other apps. Many, many other apps... One must use Foursquare to check the places of interest, TripAdvisor to mark a route, Google Maps to follow, a weather app to check the forecast, a currency app to see the current transaction rates, and search through many webpages or blogs to find whether public transport or taxi should be their preferred choice.

LodeStar uniquely combines all of these features, and then adds some. With the API's of the mentioned applications, which are free and ready for use, LodeStar merges their capabilities. To build upon that, LodeStar shows how to navigate through the destination airport, where to get a SIM card, rent a car, buy a train ticket or find a taxi and even suggests which one to pick depending on your time and money constraints. All of these functionalities are offered in one app, all from the comfort (or discomfort) of your seat at the gate.

In addition to airports, these features can also be used in favor of tourism companies. When customers who want to make an international trip and go to the offices of such companies, they can be shown, in virtual reality, the cities of their interest. The places to visit, major attractions, or staff-picked videos can be showcased. This will allow the customers to make a better decision for themselves, and increase the likelihood of them booking a ticket.

The dreaded long waits at the airport will be history. Journey into the chaotic unknown of a foreign city will be unknown to the users of LodeStar. A picture is worth a thousand words, and the 360° guides of LodeStar will outmatch any written or verbal description.

## **3.2.Functional Requirements**

In this section, functional requirements will be discussed. “Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. “ [9]. As part of functional requirements, we will discuss data resources requirements and user-specific requirements.

### **3.2.1.Data Resources**

- The application should fetch data from many other sources using several APIs.
- The application should fetch:
  - normal or 360° images from Google Street View
  - maps from Google Maps
  - flight information from Flight Radar
  - tourist attraction information from FourSquare
  - weather from [weather.com](http://weather.com)
  - currency exchange rates
- The application should show the opening and closing hours of the airports and tourist attractions.
- The application should provide real-time information about weather conditions of the destination point.
- application should show the available restaurants, cafes, and stores in the airport with their opening and closing hours. In the future, the system might favor some places over others and might list promoted items.
- The application should provide Facebook integration.
- The application will provide VR view of destination airport that shows the precise location of airport facilities and important stores for travelers, for example telecommunication shops SIM card purchase.
- The application should include taxi, subway and bus prices. Furthermore, the system should calculate the time and money needed to reach from the airport to target location using either of these methods.
- The application should be able to retrieve flight information from scanning the QR code on the boarding pass with a smart phone camera.

- The application should guide the user to target location in the airport, whether via VR images or else. Target locations can be for SIM card purchases, rent-a-car stores, baggage claim, or simply exit the airport.
- The application should guide the user to target location outside the airport with the help of Google Maps.
- The application should display necessary documents for Visa application if the user must get the Visa in the target country.
- The application should provide an estimation of the cost of general items in the target country. The countries can be ranked like expensive, moderately expensive, cheap etc.
- The application should display flight information like duration, set speed, plane attributes and the name of the captain.
- The application will gather data for city-wide events, special days and activities. The application might display additional graphics for these festivities in the VR images. This can include fireworks for New Year's Eve.
- The application should display helpful information for those with flight anxiety. Displayed information can include tips on how to avoid nausea or vomiting.
- The application should display images and information on any country during flight if the plane offers Wi-Fi service. This can be extended to show the galaxy via Google Earth.
- The application should show FourSquare comments and recommendations for airport, stores and touristic city locations. The application should allow a user to add his/her comment and rating as well.
- Facebook integration should be able to show the users information on their friends' activities on their trips using the application.
- The application can be enhanced to include 360° video clips of tourist attractions put together by staff. This can help tourism companies help their customers before choosing where to go.
- The application can show the food menus available during flight.
- The application can guide step by step on how to buy a bus or subway ticket.
- The application can have built-in games with VR support.
- In the case of contracted companies, the application should be able to guide user to those shops in the airport and see the available promotions.

### 3.2.2.User-Specific Requirements

- The user should be able to create accounts or use their preexisting Facebook information to log in.
- The user should be able to enter the name of the target country and city in order to receive information about both the airport and the tourist attractions. The user should be able to scan the QR code of his/her ticket with a smart phone camera in order to receive the same information.
- The user should be able to enter all the planned trip before going to the target country and receive guidance. That is to say, the user should be able to receive guidance on how to get from the airport to the hotel before departure. The system should save this and show it later in the target country, without requiring internet access.
- The user should be able to enter their budget. According to the entered budget, the system should recommend whether to take the taxi or the bus. Available time can be another variable here.
- According to their budget, the user will be recommended with different places to visit.
- The user should be able to see, step by step, how to buy a metro or bus ticket in the target city if such information is available.
- The user should be able to see where to eat before departure, or where the duty-free stores are.
- The user should be able to retrieve all the information for all the airports, not just the target one. This can help if they take more than one flight during the trip.
- The user should receive notifications or alarms as the flight time approaches.  
Notifications can be like: “30 minutes to boarding. Go to gate.” To build upon this, the system must acquire real time flight information like delays or canceled flights.
- The users should be able to view 360° images of their target location if they have a Google Cardboard and the images are available from Google Street View.
- The user should be able to see comments or ratings about a given place. Should s/he want it, the user should be able to add comments or ratings as well.
- The application should notify the user if a Facebook friend of theirs had been to the target country and/or city as well.
- The application might recommend specific cities or countries to visit depending on the user’s entered budget and time constraints.
- The user might be able to unlock achievements for doing various tasks while using the application. The achievements could be like, “Visit Turkey 5 times.”

- In the case of contracted companies, the user can unlock achievements like “Fly with Turkish Airlines 10 times.”
- Integrated small games for VR view, like Brick Breaker, will be provided for especially children who will use the application.

### **3.3. Non-Functional Requirements**

In this section, non – functional requirements will be discussed. There are many definitions of non – functional requirements. “Colloquially speaking, NFRs have been referred to as “-ilities” (e.g., usability) or “-ties” (e.g., integrity), i.e., words ending with the string “-ility” or “-ity” [10] . A simpler definition is to think of them as quality attributes of the software. Non-functional requirements can be very challenging to handle but are crucial for successful implementation. [11] Because of this, we will spend a lot effort to make them as good as possible. Below you can find information about LodeStar’s non-functional requirements such as reliability, usability, accessibility, extensibility, portability, and efficiency.

#### **3.3.1. Reliability**

- The application should be updated frequently in order to not show any misinformation about the airports or tourist attractions.
- The application should collect real time data regarding the flights such as delays or cancellations.
- The application should precisely display the routes to targets chosen by the user in images, and if available, in VR view.
- The application should have information about real time traffic to keep travel times accurate via taxi or bus.
- The application should give recommendations specific to the user.
- The application should provide accurate transportation recommendations, considering both the allocated budget and time.

#### **3.3.2. Usability**

- The application should be easy to use for all types of users and provide a user-friendly and a easy-to-use interface.

- The application should provide necessary information on how to put on and use Google Cardboard for VR view.
- The application should show as much information as it can by taking minimum information from the user.
- The application should give recommendations according to the user's needs and specifications.
- The application should display the comments and ratings of airport or tourist attractions clearly.
- The application should present simply on how to get to common locations like SIM card stores or rent-a-cars.

### **3.3.3. Accessibility**

- The application should provide a satisfying experience to users without using Google Cardboard as well.
- The application should remember the previously entered routes of the user (like airport to hotel map) after reaching the destination and show them without requiring further internet access.
- The application size should be sufficiently small to be downloaded while at the departure airport.

### **3.3.4. Extensibility**

- The application should be able to include new features with ease, so it should be developed keeping this in mind.
- The system should be open to future additions, such as integration of bus terminals and train stations.
- The system should be available for web integration in the future.
- The system should be able to be used with VR glasses other than Google Cardboard for a more satisfying experience.

### **3.3.5.Portability**

- The application should be able open to use on different hardware and software platforms, in our case on both Android and iOS devices.

### **3.3.6.Efficiency**

- The application should respond to user and gather travel information quickly while using the least possible amount of system resources and internet bandwidth as possible.
- The app should not take a lot of time while switching to Virtual Reality mode.
- The application should provide navigation through the application with as few steps as possible.
- The application should calculate taxi or bus routes quickly.

## **3.4.Pseudo Requirements**

- The application will have a server and will be available on App Store & Google Play

## **3.5.System Models**

To start off the system model, the report will be explaining the scenarios that a user in our application may go through. In this report twenty-six scenarios that a user may encounter are included. To explain each scenario in detail, they are divided into sub-sections as: Use Case Name, Actors, Entry Conditions, Exit Conditions and Main Flow of Events.

Secondly, the system model will include a use-case diagram. The use-case diagram will be used to describe the events and actions between the user and the system. Furthermore, each use-case is explained in detail to for further explanation.

### **3.5.1.Scenarios**

In scenarios section, we will discuss how our app will function during different user-program interactions. Importance of this section is demonstrating the storyline of the application, as well as detecting possible alternative solutions to problems we are trying to solve.

## **Scenario 1**

**Use Case Name:** LoginAsJack

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Login screen

**Exit Conditions:**

- User Jack is on Home page

**Main Flow of Events:**

1. User Jack enters his email as "jack@mail.com."
2. User Jack enters his password.
3. User Jack taps the "Login" button.
4. LodeStar checks if the username and the password is correct
5. LodeStar confirms that the credentials are correct.
6. LodeStar navigates to the Home page of the user Jack.

## **Scenario 2**

**Use Case Name:** CreateNewAccountForJack

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Login screen

**Exit Conditions:**

- User Jack is on Home page

**Main Flow of Events:**

1. User Jack taps on the "Sign Up Using Email" button on the screen.
2. LodeStar navigates to Create New Account page.
3. User Jack enters his username as "jack."
4. User Jack enters his email address as "jack@mail.com."
5. User Jack enters his choice of password twice.
  - a. User Jack taps the "Register" button.
6. LodeStar checks if the username "jack" exists in the database.
7. LodeStar checks if the mail address "jack@mail.com" exists in the database.
8. LodeStar checks if the rest of the information is accurate and complete.
9. LodeStar creates a new account.
10. LodeStar navigates to the Home page.

## **Scenario 3**

**Use Case Name:** CreateNewAccountForJackUsingGoogleCredentials

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Login screen

**Exit Conditions:**

- User Jack is on Home page

**Main Flow of Events:**

1. User Jack taps on the “Sign In With Google” button on the screen.
2. LodeStar navigates to Google Login website.
3. User Jack enters his email address as "jack@gmail.com."
4. User Jack enters his password.
5. User Jack taps the “Sign In Using Google” button.
6. LodeStar checks if the mail address "jack@gmail.com" exists in the database.
7. LodeStar creates a new account.
8. LodeStar navigates to the Home page.

## **Scenario 4**

**Use Case Name:** CreateNewAccountForJackUsingFacebookCredentials

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Login screen

**Exit Conditions:**

- User Jack is on Home page

**Main Flow of Events:**

1. User Jack taps on the “Sign In With Facebook” button on the screen.
2. LodeStar navigates to Facebook Login website.
3. User Jack enters his email address as "jack@facebookmail.com."
4. User Jack enters his password.
5. User Jack taps the “Sign In With Facebook” button.
6. LodeStar checks if the mail address "jack@facebookmail.com" exists in the database.
7. LodeStar creates a new account.
8. LodeStar navigates to the Home page.

## **Scenario 5**

**Use Case Name:** ChangePreferencesOfJack

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Me screen

**Exit Conditions:**

- User Jack is on Me screen

**Main Flow of Events:**

1. User Jack taps the “Gear” icon on the top right corner.
2. LodeStar navigates to the Preferences page.
3. User Jack views preferences for his LodeStar account.
4. User Jack navigates to the “Change Username” page of Preferences.
5. User Jack changes his username.
6. User Jack navigates to the “Privacy” page of Preferences.
  - a. User Jack opts out of app analytics.
  - b. User Jack selects “Do Not Share My LodeStar Activity on Facebook.”
  - c. User Jack selects “Do Not Notify My Facebook friends about my activity.”
7. User Jack User Jack taps the “Clear History” button.
8. User Jack taps the “Save Preferences” button.
  - a. LodeStar checks if all the given information are accurate and valid.
  - b. LodeStar saves the changes of User Jack’s account.
  - c. LodeStar navigates to the Homepage of the User Jack.

## **Scenario 6**

**Use Case Name:** ScanFlightTicketQR

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Home page

**Exit Conditions:**

- User Jack is on Trip screen

**Main Flow of Events:**

1. User Jack taps on the “Scan Flight Ticket” button on the screen.
2. LodeStar navigates to the “Flight Ticket Scanner” screen.
3. User Jack allows LodeStar to use his device’s camera.
4. User Jack points his device’s camera to his flight ticket.

5. LodeStar reads the barcode on the flight ticket.
  - a. LodeStar gets the flight number.
  - b. LodeStar acquires the flight information using the flight number.
  - c. LodeStar gets additional information about the destination city.
  - d. LodeStar compiles the information into a readable format.
6. LodeStar navigates to the “Trip” page.

**Use Case Name:** ViewLodeStarDestinationWeatherInformation

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Trip screen

**Exit Conditions:**

- User Jack is on Trip screen

**Main Flow of Events:**

1. User Jack taps on the “Weather” button on the screen.
2. LodeStar navigates to the “Destination Weather” screen.
3. LodeStar gathers information about the weather of the destination city.
4. LodeStar acquires information about temperatures.
5. LodeStar acquires information about weather conditions (rain, fog, etc.).
6. LodeStar acquires information about wind and humidity.
7. User Jack reads and digests this information.
8. User Jack taps the “Back” button.
9. LodeStar navigates back to the “Trip” page.

## Scenario 8

**Use Case Name:** ViewLodeStarFlightInformation

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Trip screen

**Exit Conditions:**

- User Jack is on Trip screen

**Main Flow of Events:**

1. User Jack taps the “Flight Information” button on the screen.
2. LodeStar navigates to the “Flight Information” page.
3. LodeStar gathers information about the flight using the flight number.
  - a. LodeStar acquires departure and arrival time.

- b. LodeStar acquires aircraft brand and model.
  - c. LodeStar acquires the pilot's name, if available.
  - d. LodeStar acquires estimated flight speed, if available.
  - e. LodeStar acquires flight distance.
4. User Jack reads and digests this information.
  5. User Jack taps the "Back" button.
  6. LodeStar navigates back to the "Trip" screen.

## **Scenario 9**

**Use Case Name:** ViewLodeStarSIMCardInformation

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Trip screen

**Exit Conditions:**

- User Jack is on Trip screen

**Main Flow of Events:**

1. LodeStar displays a menu option called "How to Buy a SIM Card" if the destination airport sells SIM Cards in the airport.
2. User Jack taps on the "How to Buy a SIM Card" button.
3. If the destination Airport has Google Street View 360 images available, LodeStar shows a "View In Virtual Reality" button .LodeStar shows the route from exit gates to SIM card shop in VR view.
4. User Jack travels in the route and digests the route information.
5. User Jack taps the "Back" button.
6. LodeStar navigates back to the "Trip" screen.

## **Scenario 10**

**Use Case Name:** ManuallyEnterFlightInformation

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Homepage

**Exit Conditions:**

- User Jack is on LodeStar Home screen

**Main Flow of Events:**

1. User Jack taps on the "Manually Search for Destination" button on the screen.

2. LodeStar navigates to the Destination Cities list.
3. User Jack selects his destination city in this list.
4. LodeStar lists available airports in that city (optional).
5. User Jack selects his destination airport in the list.
6. LodeStar navigates to the “LodeStar Flight” screen.

## Scenario 11

**Use Case Name:** ViewLodeStarTransportationSuggestion

**Actors:** Jack

**Entry Conditions:**

- User Jack is on City screen

**Exit Conditions:**

- User Jack is on City screen

**Main Flow of Events:**

1. User Jack taps on the “Transportation Suggestions” button on the screen.
2. LodeStar navigates to Transportation Suggestions screen.
3. LodeStar asks User Jack the location he wishes to go.
4. LodeStar gathers information about transportation selections that are available to user at the time of the arrival and their approximate costs.
5. LodeStar displays the details of transportation choice.
  - a. LodeStar acquires location of the transportation choice.
  - b. LodeStar acquires the approximate price to use the transportation choice.
6. User Jack reads and digests this information.
7. User Jack presses the “Back” button.
8. User Jack navigates back to the Trip screen.

## Scenario 12

**Use Case Name:** ViewLodeStarAirport

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Destination screen

**Exit Conditions:**

- User Jack is on Destination screen

**Main Flow of Events:**

1. User Jack taps on the “Destination Airport” button on the screen.

2. LodeStar navigates to Destination Airport screen.
3. LodeStar asks User Jack about their destination airport he wishes to see.
4. LodeStar gathers information about the destination airport.
5. LodeStar displays the details of destination Airport.
  - a. LodeStar displays the destination Airport virtually.
  - b. User Jack can navigate in Virtual Reality.
  - c. LodeStar displays the airport in Google Maps.
6. User Jack navigates back to the Destination screen.

### **Scenario 13**

**Use Case Name:** TravelLodeStarAirport

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Destination screen

**Exit Conditions:**

- User Jack is on Destination screen

**Main Flow of Events:**

1. User Jack taps on the “Destination Airport” button on the screen.
2. LodeStar navigates to Destination Airport screen.
3. LodeStar asks User Jack about the destination airport he wishes to see.
4. LodeStar gathers information about the destination airport.
5. LodeStar displays the details of destination city.
  - a. LodeStar displays the destination city virtually.
  - b. LodeStar displays the city in Google Maps.
6. User Jack navigates back to the Destination screen.

### **Scenario 14**

**Use Case Name:** TravelLodeStarCity

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Destination City screen

**Exit Conditions:**

- User Jack is on Destination City screen

**Main Flow of Events:**

1. User Jack taps on the “Destination City” button on the screen.

2. LodeStar navigates to Destination City screen.
3. LodeStar asks User Jack about the destination city he wishes to see.
4. User Jack enters the destination city.
5. LodeStar gathers information about the destination city.
6. LodeStar presents the information.
7. User Jack taps the “View In Virtual Reality” button.
8. LodeStar displays the destination city in Virtual Reality.
9. User Jack taps the arrows in the virtual environment to roam in the destination city.
10. User Jack navigates back to the Destination screen.

## **Scenario 15**

**Use Case Name:** FlightTimeNotification

**Actors:** Jack

**Entry Conditions:**

- User Jack is on any page

**Exit Conditions:**

- User Jack taps the “Dismiss” button

**Main Flow of Events:**

1. User Jack is on any page.
2. 15 minutes remain until boarding time.
3. LodeStar gives a notification via an alarm to Jack.
4. User Jack taps the “Dismiss” button of the notification and goes to the plane.

## **Scenario 16**

**Use Case Name:** DuringFlightVirtualExploration

**Actors:** Jack

**Entry Conditions:**

- User Jack is on board with Wi-Fi access

**Exit Conditions:**

- User Jack closes the “During Flight” screen

**Main Flow of Events:**

1. User Jack navigates to During Flight screen.
2. LodeStar gives information about the city the plane is flying over.
3. User Jack selects the Explore Virtually option.
4. User Jack gets a peek of the city’s points of interest in a virtual reality environment.

5. User Jack closes the During Flight screen.

## **Scenario 17**

**Use Case Name:** DuringFlightVideoExploration

**Actors:** Jack

**Entry Conditions:**

- User Jack is on board with Wi-Fi access

**Exit Conditions:**

- User Jack closes the “During Flight” screen

**Main Flow of Events:**

1. User Jack navigates to During Flight screen.
2. LodeStar gives information about the city the plane is flying over.
3. User Jack selects the Explore by Video option.
4. User Jack gets a peek of the city’s points of interest in the form of a video.
5. User Jack closes the During Flight screen.

## **Scenario 18**

**Use Case Name:** DuringFlightReadingExploration

**Actors:** Jack

**Entry Conditions:**

- User Jack is on board with Wi-Fi access

**Exit Conditions:**

- User Jack closes the “During Flight” screen

**Main Flow of Events:**

1. User Jack navigates to During Flight screen.
2. LodeStar gives information about the city the plane is flying over.
3. User Jack selects the Explore by Text option.
4. User Jack gets a peek of the city’s points of interest in the form of textual description.
5. User Jack closes the During Flight screen.

## **Scenario 19**

**Use Case Name:** ViewDestinationCityVideos

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Destination City screen

**Exit Conditions:**

- User Jack is on the Destination City screen

**Main Flow of Events:**

1. User Jack is on Destination City screen.
2. User Jack presses the “Videos From The City” button.
3. LodeStar shows the most popular Youtube videos featuring the destination city.
4. User Jack selects one of the Youtube videos.
5. User Jack watches the video.
  - a. If the video supports 360 view, Jack can watch it virtual reality.
  - b. Jack can also watch the video through in app Youtube player.
6. User Jack closes the video.
7. User Jack returns to Destination City screen.

**Scenario 20**

**Use Case Name:** ViewCityTouristAttractions

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Destination City page

**Exit Conditions:**

- User Jack is on Destination City page

**Main Flow of Events:**

1. User Jack is taps on the button “Tourist Attractions” on City page.
2. LodeStar lists the popular spot suggestions in a list.
3. Jack taps on a listed suggestion.
4. LodeStar shows the VR view of the city at the chosen location.
5. LodeStar displays some information about the location.
6. User Jack taps the “Back” button.
7. LodeStar navigates to Destination City screen.

**Scenario 21**

**Use Case Name:** ViewCityAccommodation

**Actors:** Jack

**Entry Conditions:**

- User Jack is on LodeStar Destination City page

**Exit Conditions:**

- User Jack is on the Destination City page

**Main Flow of Events:**

1. User Jack taps on the “Accommodation View” button.
2. Jack types the words of the Hotel or address of the intended accommodation.
3. LodeStar shows the place on the map.
4. Tapping on the place on the map will open the VR view of the place(if available)or nearby street.
5. User Jack taps the “Back” button
6. LodeStar navigates to Destination City screen.

**Scenario 22****Use Case Name:** ViewCityEvents**Actors:** Jack**Entry Conditions:**

- User Jack is on Destination City screen

**Exit Conditions:**

- User Jack is on Destination City screen

**Main Flow of Events:**

1. User Jack taps “City Events” button on Destination City screen.
2. LodeStar shows the active and upcoming city events on the map.
3. User Jack selects one of the events by tapping on it.
4. LodeStar opens the VR view of the event location.
5. LodeStar gathers information about the events and shows it to Jack.
6. User Jack taps the “Back” button
7. LodeStar navigates to Destination City screen.

**Scenario 23****Use Case Name:** ViewCommentsOfPlaces**Actors:** Jack**Entry Conditions:**

- User Jack is on Destination City screen

**Exit Conditions:**

- User Jack is on Destination City screen

**Main Flow of Events:**

1. User Jack makes a selection of a place to visit on the Destination City screen
2. LodeStar displays the targeted place and shows a “Comments” button, if available comments exist.
3. User Jack taps on the “Comments” button.
4. LodeStar displays the comments.
5. User Jack taps the “Back” button.
6. LodeStar navigates to Destination City screen.

## **Scenario 24**

**Use Case Name:** EnterCommentForPlaces

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Destination City screen

**Exit Conditions:**

- User Jack is on Destination City screen

**Main Flow of Events:**

1. User Jack selects some places to visit on the Destination City screen.
2. LodeStar displays the targeted place and shows a “Comments” button, if available comments exist.
3. User Jack taps on the “Comments” button.
4. LodeStar displays the comments.
5. User Jack taps “Add A Comment” button.
6. LodeStar displays a title and text area.
7. User Jack enters the new comment for the place.
8. User Jack taps the “Submit” button.
9. LodeStar adds the new comment to its database and displays it.
10. User Jack taps “Done” button.
11. LodeStar navigates to Destination City screen.

## **Scenario 25**

**Use Case Name:** ViewLivingExpenses

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Destination City screen

**Exit Conditions:**

- User Jack is on Destination City screen

**Main Flow of Events:**

1. User Jack taps the “See Living Expenses” button on the Destination City screen.
2. LodeStar gathers data for the city.
3. LodeStar displays the living expenses on the city:
  - a. LodeStar acquires daily restaurant costs,
  - b. LodeStar acquires rent prices,
  - c. LodeStar acquires daily approximate grocery costs,
  - d. LodeStar acquires daily approximate transportation costs.
4. User Jack taps on the “Back” button.
5. LodeStar navigates to Destination City screen.

**Scenario 26**

**Use Case Name:** ViewCurrencyRates

**Actors:** Jack

**Entry Conditions:**

- User Jack is on Destination City screen

**Exit Conditions:**

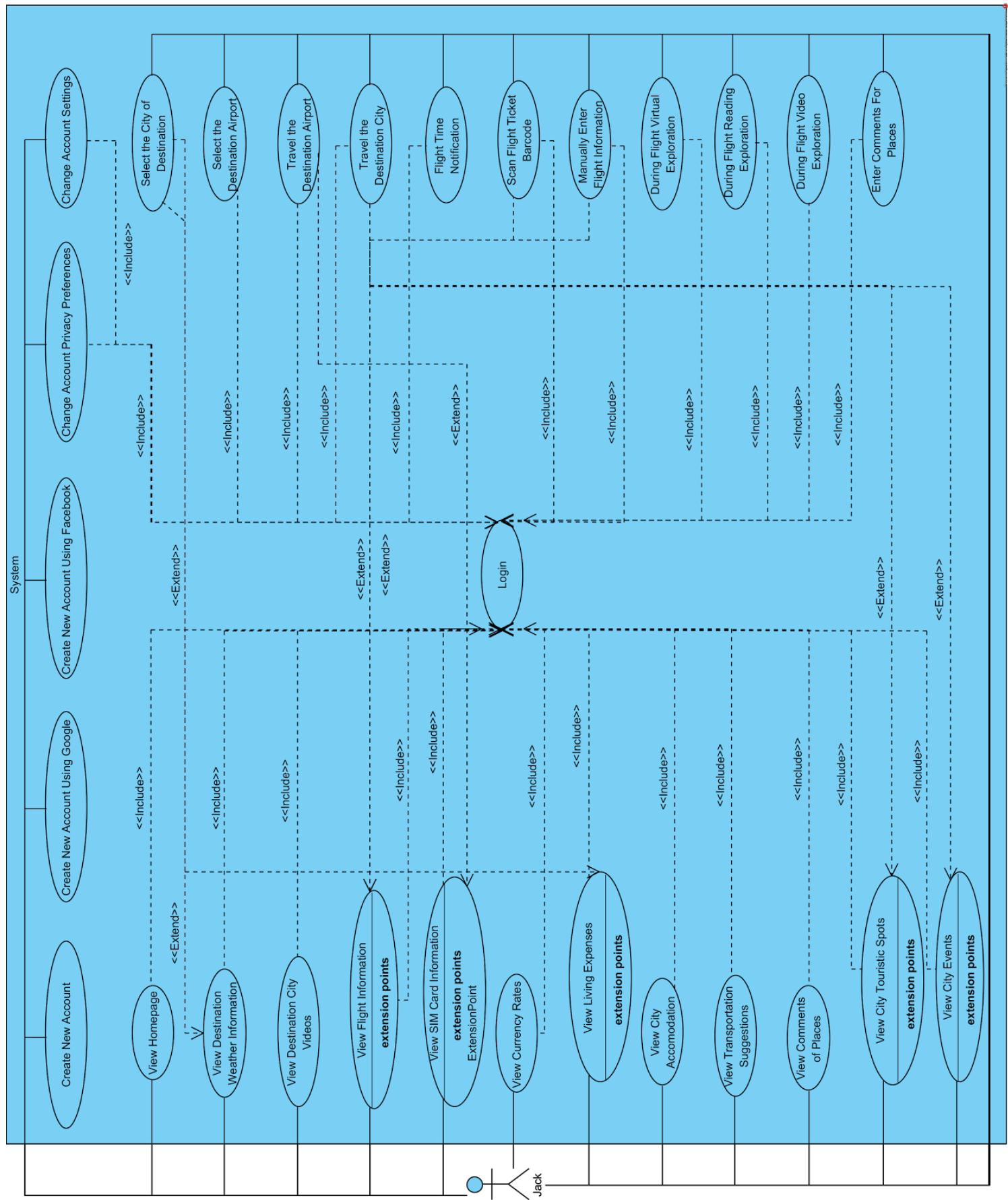
- User Jack is on Destination City screen

**Main Flow of Events:**

1. User Jack taps the “See Currency Rates” button.
2. LodeStar gathers real time exchange data.
3. LodeStar displays the currency exchange rates between the visited country and User Jack’s country.
4. User Jack taps on the “See Currency Exchange Shops” button on the screen.
5. LodeStar displays the currency exchange shops on a map.
6. User Jack taps on one of the displayed places.
7. LodeStar displays VR view of the shop if available.
8. User Jack taps on the “Back” button.
9. LodeStar navigates to Destination City screen.

### **3.5.2. Use Case Model**

In the following section, we will introduce use-case models of our system. A use case is a methodology used in system analysis to identify, clarify, and organise system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal [12]. This is important for showing our user-system interactions.



### **3.5.2.1.Login Use Case**

**Use Case Name:** Login

**Actors:** User

**Entry Conditions:**

- User is on Login screen

**Exit Conditions:**

- User is on Home page
- Authentication failed and user is on Login screen

**Main Flow of Events:**

1. User enters email.
2. User enters password.
3. User taps the “Login” button.
4. LodeStar checks if the email and the password are correct.
5. LodeStar confirms that the credentials are correct.
6. LodeStar navigates to the Home page of the user.

**Alternative Flow of Events:**

- A. User enters wrong email and/or wrong password.
  - a. LodeStar detects that the credentials are incorrect.
  - b. LodeStar navigates to Login screen.
  - c. LodeStar informs user about the wrong entries.
- B. User presses “Login with Google” button
  - a. LodeStar looks for the user’s Google credentials.
  - b. LodeStar navigates to the Home page of the user.
- C. User presses “Login with Facebook” button
  - a. LodeStar looks for the user’s Facebook credentials.
  - b. LodeStar navigates to the Home page of the user.

### **3.5.2.2.Create Account Use Case**

**Use Case Name:** Create New Account

**Actors:** User

**Entry Conditions:**

- User is on Login screen

**Exit Conditions:**

- User is on Home page

**Main Flow of Events:**

1. User taps the “Login Using Email” button.
2. LodeStar navigates to “Sign Up” page.
3. User enters username.
4. User enters email address.
5. User enters password twice.
6. User taps the “Sign up Using Email” button.
7. LodeStar checks if there is an already user registered with that username.
8. LodeStar checks if there is an already user registered with that mail address.
9. LodeStar checks if the information is accurate and complete.
10. LodeStar creates a new account for the user.
11. LodeStar navigates to the Home page of the user.

**Alternative Flow of Events:**

- A. User entries are inaccurate or missing.
  - a. LodeStar returns to Login screen.
  - b. LodeStar informs the user about missing and/or inaccurate entries.
  - c. User enter the missing and/or inaccurate informations again.
  - d. LodeStar creates a new account for the user.
  - e. LodeStar navigates to the Home page of the user.

### **3.5.2.3.Create Account Using Google Use Case**

**Use Case Name:** Create New Account Using Google

**Actors:** User

**Entry Conditions:**

- User is on Login screen

**Exit Conditions:**

- User is on Home page

**Main Flow of Events:**

1. User presses on the “Login Using Google” button.
2. User enters his email address.
3. User enters his password.
4. LodeStar checks if the mail address already exists in the database.
5. LodeStar creates a new account.
6. LodeStar navigates to the Home page of the user.

**Alternative Flow of Events:**

- A. User has already logged into Google.
  - a. User presses on “Login Using Google” button
  - b. LodeStar checks the credentials.
  - c. LodeStar creates a new account.
  - d. LodeStar navigates to the Home page of the user.
- B. User has already logged in using Google
  - a. User presses on “Login Using Google” button.
  - b. LodeStar checks the credentials.
  - c. LodeStar navigates to the Home page of the user.

### **3.5.2.4.Create Account Using Facebook Use Case**

**Use Case Name:** Create New Account Using Facebook

**Actors:** User

**Entry Conditions:**

- User is on Login screen

**Exit Conditions:**

- User is on Home page

**Main Flow of Events:**

1. User taps the “Sign in with Facebook” button.
2. User enters email address.

3. User enters password.
4. LodeStar checks if the mail address already exists in the database.
5. LodeStar creates a new account.
6. LodeStar navigates to the Home page of the user.

**Alternative Flow of Events:**

- C. User has already logged into Facebook.
  - a. User taps on “Sign in Using Facebook” button.
  - b. LodeStar checks the credentials.
  - c. LodeStar creates a new account.
  - d. LodeStar navigates to the Homepage of the user.
- D. User has already logged in using Facebook.
  - a. User presses on the “Login Using Facebook” button.
  - b. LodeStar checks the credentials.
  - c. LodeStar navigates to Home page of the user.

### 3.5.2.5.Change Account Settings Use Case

**Use Case Name:** Change Account Settings

**Actors:** Jack

**Entry Conditions:**

- Change Account Setting includes Login use case
- User is on Me screen

**Exit Conditions:**

- User is on Me screen

**Main Flow of Events:**

1. User taps on the “Gear” icon.
2. LodeStar navigates to the Preferences screen.
3. User changes his username.
4. User changes his email address.
5. User changes his password.
6. User taps the “Save Preferences” button.
7. LodeStar checks if all the given information are accurate and valid.
8. LodeStar saves the changes of user’s account.
9. LodeStar navigates to the Me screen of the user.

**Alternative Flow of Events:**

- A. 1.User has signed in using Google.
  - a. User cannot change his email address (Skip Step 5).
- B. 2.User has signed in using Facebook.
  - a. User cannot change his email address (Skip Step 5).
- C. User does not change his username (Skip Step 4).
- D. User does not change his email address (Skip Step 5).
- E. User does not change his password (Skip Step 6).

### **3.5.2.6.Change Account Privacy Preferences Use Case**

**Use Case Name:** Change Account Privacy Preferences

**Actors:** Jack

**Entry Conditions:**

- Change Account Privacy Preferences includes Login use case
- User Jack is on Me screen

**Exit Conditions:**

- User Jack is on Me screen

**Main Flow of Events:**

1. User taps on the “Gear” icon.
2. LodeStar navigates to the Preferences screen.
3. User views his preferences.
4. User navigates to the “Privacy” screen of Preferences.
  - a. User opts out of app analytics.
  - b. User selects “Do Not Share My LodeStar Activity on Facebook.”
  - c. User selects “Do Not Notify My Facebook Friends About My Activity.”
5. User taps the “Clear History” button.
6. User taps the “Save Preferences” button.
7. LodeStar saves the new preferences of the user.
8. LodeStar navigates to the Me screen of the user.

**Alternative Flow of Events:**

- A. 1.User does not opt out of app analytics (Skip Step 4.a).
- B. 2.User does not select “Do Not Share My LodeStar Activity on Facebook” (Skip Step 4.b).
- C. 3.User does not select “Do not Notify My Facebook Friends About My Activity” (Skip Step 4.c).

### **3.5.2.7.Select the City of Destination Use Case**

**Use Case Name:** Select the City of Destination Use Case

**Actors:** Jack

**Entry Conditions:**

- Select the City of Destination includes Login use case
- User Jack is on Trip screen

**Exit Conditions:**

- User Jack leaves Trip screen

**Main Flow of Events:**

1. User taps the “Select Destination” button.
2. LodeStar navigates to Trip screen.
3. User selects the destination city.

**Alternative Flow of Events:**

- A. User does not select the destination city. (In this case nothing is loaded).

### **3.5.2.8.Travel the Destination City Use Case**

**Use Case Name:** Travel the Destination City Use Case

**Actors:** Jack

**Entry Conditions:**

- Travel the Destination City includes Login use case
- Travel the Destination City includes Select Destination use case
- User Jack is on Trip screen

**Exit Conditions:**

- User Jack leaves Trip screen

**Main Flow of Events:**

1. User taps the “Select Destination” button.
2. User selects the City of Destination.
3. User is redirected to the Trip screen.
4. User travels the tourist attractions of the Destination City with Google Street View.

**Alternative Flow of Events:**

- A. User selects the virtual reality option in the 4th event to travel the tourist attractions

### **3.5.2.9. Select the Destination Airport Use Case**

**Use Case Name:** Select the Destination Airport Use Case

**Actors:** Jack

**Entry Conditions:**

- Select the Destination Airport includes Login use case
- User Jack is on Trip screen

**Exit Conditions:**

- User Jack leaves the Trip screen

**Main Flow of Events:**

1. User taps the “Select Destination Airport” button.
2. User selects the Destination Airport.
3. User is redirected to the Trip screen.

### **3.5.2.10. Travel the Destination Airport Use Case**

**Use Case Name:** Travel the Destination Airport Use Case

**Actors:** User

**Entry Conditions:**

- Travel the Destination City includes Login use case
- Travel the Destination City includes Select Destination Airport use case
- User Jack is on Trip screen

**Exit Conditions:**

- User Jack leaves the Trip screen

**Main Flow of Events:**

1. User taps the “Select Destination Airport” button.
2. User selects the Destination Airport
3. User is redirected to the Trip screen.
4. User travels the Destination Airport with Google Street View.

**Alternative Flow of Events:**

- A. User selects the virtual reality option in the 4th event to travel the destination airport

### **3.5.2.11. Scan Flight Ticket Barcode Use Case**

**Use Case Name:** Scan Flight Ticket Barcode Use Case

**Actors:** User

**Entry Conditions:**

- Scan Flight Ticket Barcode includes Login use case
- User is on Home page

**Exit Conditions:**

- User is on LodeStar Flight screen

**Main Flow of Events:**

1. User taps the “Scan a Boarding Pass” button.
2. LodeStar navigates to the Flight Ticket Scanner screen.
3. User allows LodeStar to use his device’s camera.
4. User points his device’s camera to his flight ticket.
5. LodeStar reads the barcode on the flight ticket.
6. LodeStar gets the flight number and flight information using the flight number.
7. LodeStar gets additional information about the destination city and displays into a readable format.
8. LodeStar navigates to the Trip screen.

**Alternative Flow of Events:**

- A. User doesn’t give permission to use device camera (skip steps 4-6).

### 3.5.2.12. View Destination Weather Information Use Case

**Use Case Name:** View Destination Weather Information

**Actors:** User

**Entry Conditions:**

- View Destination Weather Information includes Login Use case
- User is on Trip screen

**Exit Conditions:**

- User is on LodeStar Trip screen

**Main Flow of Events:**

1. User taps the “Weather” button.
2. LodeStar navigates to the “Weather Information” screen.
3. LodeStar loads information about temperatures, weather conditions (rain, fog, etc.), and wind.

### **3.5.2.13. View Flight Information Use Case**

**Use Case Name:** View Flight Information

**Actors:** User

**Entry Conditions:**

- View Flight Information includes Login Use case
- User is on Trip screen

**Exit Conditions:**

- User is on Trip screen

**Main Flow of Events:**

1. User taps the “Flight Information” button.
2. LodeStar opens “Flight Details” screen
3. LodeStar retrieves information about the flight using the flight number.
4. LodeStar loads departure and arrival time, aircraft brand, model and flight distance, if available.

### **3.5.2.14. View Flight Information Use Case**

**Use Case Name:** View SIM Card Information

**Actors:** User

**Entry Conditions:**

- View SIM card Information includes Login Use Case
- User is on Trip screen

**Exit Conditions:**

- User is on Trip screen

**Main Flow of Events:**

1. LodeStar displays a menu option called “How to Buy a SIM Card” if the destination airport sells SIM Cards in the airport.
2. User taps the “How to Buy a SIM Card” button.
3. LodeStar lists the related shops.
4. If the destination Airport has Google Street View 360 images available, LodeStar shows a “View with VR” button next to the shops in the lists.
5. LodeStar shows the route from exit gates to SIM card shop in VR view.
6. User travels in the route in VR View.

**Alternative Flow of Events:**

- A. Destination airport doesn't have SIM Card shops (skip 2-5).
- B. Destination airport doesn't have Street View images available (skip 4-5).

### **3.5.2.15.Manually Enter Flight Information Use Case**

**Use Case Name:** Manually Enter Flight Information

**Actors:** User

**Entry Conditions:**

- Manually Enter Flight Information includes Login use case
- User is on Home page

**Exit Conditions:**

- User is on Trip screen

**Main Flow of Events:**

1. User taps the “Manually Search for Destination” button.  
LodeStar navigates to the Destination cities list.
2. User selects his destination city in this list.  
LodeStar displays available airports in the city.
3. User selects his destination airport in this list.
4. LodeStar displays “Trip” screen.

**Alternative Flow of Events:**

- A. Destination city aren't included in LodeStar database (skip 2-6).
- B. Destination airport aren't included in LodeStar database (skip 4-6).

### **3.5.2.16.View Transportation Suggestions Use Case**

**Use Case Name:** View Transportation Suggestion

**Actors:** User

**Entry Conditions:**

- View Transportation Suggestion includes Login use case
- User is on Trip screen

**Exit Conditions:**

- User is on Trip screen

**Main Flow of Events:**

1. User taps the “Transportation Options” button.
2. LodeStar opens to Transportation Options page.

3. LodeStar asks user about their location in the city that they want to go.
4. User types the name or address of the location.
5. LodeStar gathers and lists information about transportation selections that are available to user at the time of the arrival and their approximate costs.
6. User selects one of the options.
7. LodeStar displays the details of transportation choice.
  - a. LodeStar loads location of the transportation choice.
  - b. LodeStar loads the approximate price to use the transportation choice.

**Alternative Flow of Events:**

- A. User typed location cannot be found in the city.
  - a. LodeStar asks user again to type the information (return to step 4).
- B. No available transportation suggestion found (skip 5-7 ).

### **3.5.2.17.Flight Time Notification Use Case**

**Use Case Name:** Flight Time Notification

**Actors:** User

**Entry Conditions:**

- Flight Time Notification includes Login use case
- User is on any screen

**Exit Conditions:**

- User taps the Dismiss button

**Main Flow of Events:**

1. User is on any screen.
2. LodeStar detects only 15 minutes remain until boarding.
3. LodeStar gives a notification via an alarm to User.
4. User taps the “Dismiss” button.

### **3.5.2.18.During Flight Virtual Exploration Use Case**

**Use Case Name:** During Flight Virtual Exploration

**Actors:** User

**Entry Conditions:**

- During Flight Virtual Exploration includes Login use case
- User is on board with Wi-Fi access

**Exit Conditions:**

- User closes the travel During Flight screen

**Main Flow of Events:**

1. User is on any page.
2. LodeStar gives information about the city the plane is flying over.
3. User selects the “Explore In Virtual Reality” option.
4. User gets a peek of the city’s points of interest in virtual reality.

### **3.5.2.19.During Flight Virtual Exploration Use Case**

**Use Case Name:** During Flight Video Exploration

**Actors:** User

**Entry Conditions:**

- During Flight Video Exploration includes Login use case
- User is on board with Wi-Fi access

**Exit Conditions:**

- User is on the travel during flight page

**Main Flow of Events:**

1. User is on any page.
2. LodeStar gives information about the city the plane is flying over
3. User selects the “Explore by Video” option
4. User gets a peek of the city’s points of interest in video form.

### **3.5.2.20.During Flight Reading Exploration Use Case**

**Use Case Name:** During Flight Reading Exploration

**Actors:** User

**Entry Conditions:**

- During Flight Reading Exploration includes Login use case
- User is on board with Wi-Fi access

**Exit Conditions:**

- User is on the travel during flight page

**Main Flow of Events:**

1. User is on any page.
2. LodeStar gives information about the city the plane is flying over.

3. User selects the “Explore by Text” option.
4. User Jack gets a peek of the city’s points of interest in text form.

### **3.5.2.21. View Destination City Videos Use Case**

**Use Case Name:** View Destination City Videos

**Actors:** User

**Entry Conditions:**

- User is on Trip screen

**Exit Conditions:**

- User Jack is on the Trip screen

**Main Flow of Events:**

1. User Jack is on Trip screen.
2. LodeStar shows most popular Youtube videos about that city.
3. User selects one of the Youtube videos.
4. User watches the video.
  - a. If the video supports 360 view, user can watch it in virtual reality.
  - b. User can watch the video through in app Youtube player.

### **3.5.2.22. View City Touristic Spots Use Case**

**Use Case Name:** View City Tourist Attractions

**Actors:** User

**Entry Conditions:**

- View City Touristic Spots includes Login use case
- User is on Trip screen

**Exit Conditions:**

- User is on the Trip screen

**Main Flow of Events:**

1. User taps the “Tourist Attraction” button on City screen.
2. LodeStar lists the popular spot suggestions in a list.
3. User taps on a listed suggestion.
4. LodeStar shows the VR view of the city at the chosen location.
5. LodeStar displays some information about the location.

**Alternative Flow of Events:**

- A. No suggested touristic spot is found (skip 2-5).

### **3.5.2.23. View City Accommodation Use Case**

**Use Case Name:** View City Accommodation

**Actors:** User

**Entry Conditions:**

- View City Accommodation includes Login use case
- User is on Trip screen

**Exit Conditions:**

- User is on the Trip screen

**Main Flow of Events:**

1. User taps the “Accommodation” button.
2. User types the words of the Hotel or address of the intended accommodation.
3. LodeStar shows the place on the map.
4. Tapping on the place on the map will open the VR view of the place (if available) or nearby street.

**Alternative Flow of Events:**

- A. Typed words don't match with any city location, LodeStar asks the user to try again (return step 2).
- B. No available panorama Google Street View is found near the area (skip 4).

### **3.5.2.24. View City Events Use Case**

**Use Case Name:** View City Events

**Actors:** User

**Entry Conditions:**

- View City Events includes Login Use case
- User is on Trip page

**Exit Conditions:**

- User is on the Trip page

**Main Flow of Events:**

1. User taps the “City Events” button on Trip screen.
2. LodeStar shows the active and upcoming city events on the map.
3. User selects one of the events.
4. LodeStar will open the VR view of the event location.

5. LodeStar displays the information about the chosen event.

**Alternative Flow of Events:**

- A. No Google Street View is available around the event location(skip 4-5).
- B. No available information is found about the event (skip 5).

### **3.5.2.25.View Comments of Places Use Case**

**Use Case Name:** View Comment Of Places

**Actors:** User

**Entry Conditions:**

- View Comment Of Places includes Login use case
- User is on Trip screen

**Exit Conditions:**

- User is on the Trip screen

**Main Flow of Events:**

1. User taps the “Places to See” button.
2. User makes a selection any place to visit on the “Places to See” screen.
3. When places are listed, “Comments” button are listed under the places.
4. User taps the “Comments” button of one of the places.
5. LodeStar gathers data of the place and displays the comments.

**Alternative Flow of Events:**

- A. No available comment is found for the place (skip 3-4).

### **3.5.2.26.Enter Comment for Places Use Case**

**Use Case Name:** Enter Comment For Places

**Actors:** User

**Entry Conditions:**

- Enter Comment For Places includes Login use case
- User is on Trip screen

**Exit Conditions:**

- User is on the Trip screen

**Main Flow of Events:**

1. User taps the “Places to See” button.
2. User makes a selection to visit on the “Places to See” screen.

3. When places are listed, “Comments” button are listed under the places, if available comments exists.
4. User taps on the Comments button of one of the places.
5. LodeStar gathers data of the place and displays the comments.
6. User enters a new comment for the place.
7. User taps the “Send” button.
8. LodeStar adds the new comment in the place data.
9. LodeStar navigates to Destination City screen.

**Alternative Flow of Events:**

- A. User exceeds character limit when commenting (return step 4).

### 3.5.2.27. View Living Expenses Use Case

**Use Case Name:** View Living Expenses

**Actors:** User

**Entry Conditions:**

- View Living Expenses includes Login use case
- User is on Trip screen

**Exit Conditions:**

- User is on the Trip screen

**Main Flow of Events:**

1. User taps the “Living Expenses” button.
2. LodeStar gathers and displays the living expenses on the city:
  - a. LodeStar shows daily restaurant costs.
  - b. LodeStar shows average rent prices.
  - c. LodeStar shows daily approximate grocery costs.
  - d. LodeStar shows daily approximate transportation costs.

### 3.5.2.28. View Currency Rates Use Case

**Use Case Name:** View Currency Rates

**Actors:** User

**Entry Conditions:**

- View Currency Rates includes Login use case
- User is on Trip screen

**Exit Conditions:**

- User is on the Trip screen

### Main Flow of Events:

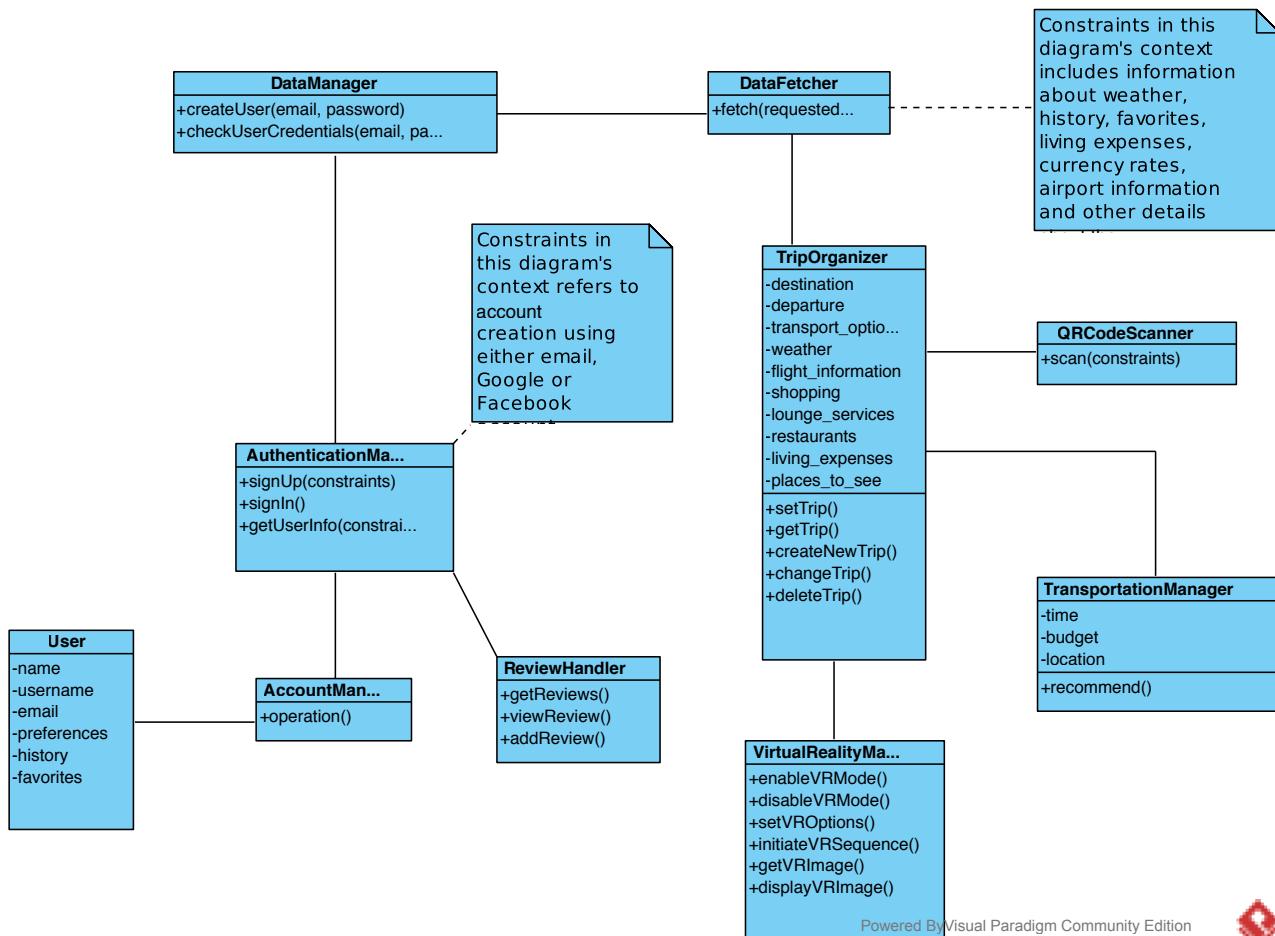
1. User taps the “See Currency Rates” button.
2. LodeStar gathers and displays the currency exchange rates between the visited country and user’s country.
3. User taps on the button “See Currency Exchange Shops” on the screen.
4. LodeStar displays the currency exchange shops on a map.
5. User taps on one of the displayed places.
6. If available, VR view of the nearby shops are displayed.

### Alternative Flow of Events:

- A. No shop for currency exchange is available (skip 5-7).

### 3.5.3.Object and Class Model

For demonstrating our initial design for program architecture, we will introduce object class diagram in this section. The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction[13].



**TripOrganizer:** This class will control and maintain trip related information. It will communicate with DataManager to fetch information related to the specified airport.

**DatabaseManager:** This class will be responsible for all database communications.

**DataFetcher:** This class will fetch required information about the trips. Will contain all API communication code. As an example, the fetch function in this class can be called using a flight no as an argument to get specific information about a flight.

**User:** This class is a representation of a user in LodeStar. The user class will contain distinguishing information like name-surname, email, preferences, history and person

**VirtualRealityManager:** The task of this class is to manage the virtual reality aspects of our application. It will be responsible for initiating the Virtual Reality mode, fetching the VR image, displaying the image and adjust any preferences the user might have when the VR image is shown.

**AuthenticationManager:** The duty of this class is to enable the authentication via Google+ and Facebook. This class will give the user the option to sign in using one of these social media platforms.

**Airport:** This class will hold the critical information about an airport. It will hold the name of the airport, the city the airport is located, the country the airport is located in and the working hours of the airport.

**TransportationManager:** This class will contain information about transportation related activities such as time, location and budget constraints

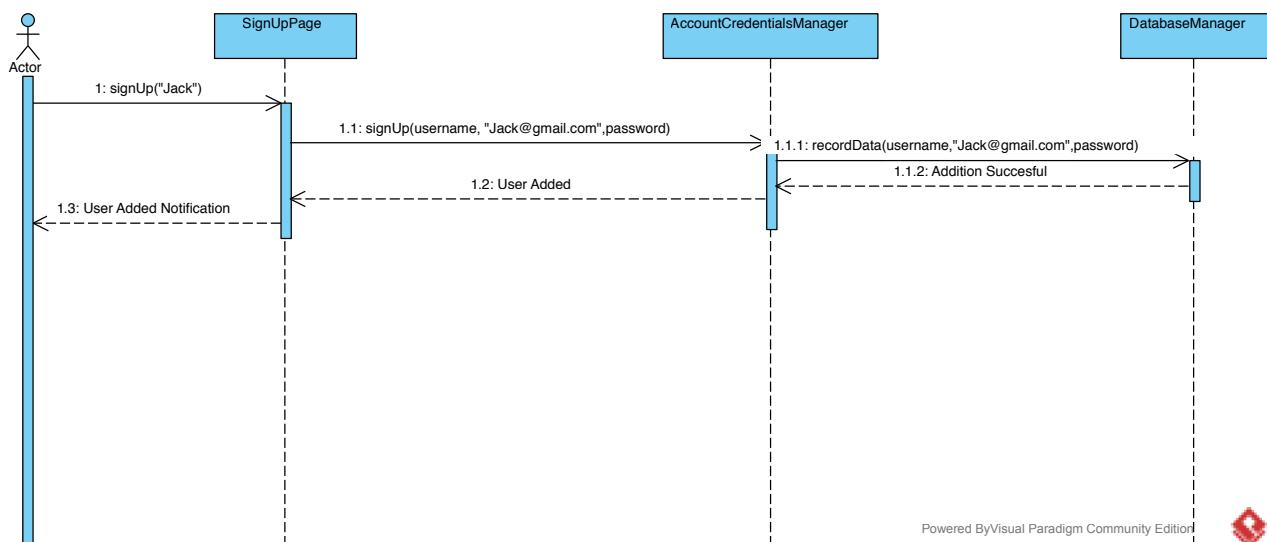
### 3.5.4.Dynamic Models

Dynamic modelling of a system demonstrates the function calls and interaction between the system components, thus shows the inner processing of the system.

#### 3.5.4.1.Sequence Diagrams

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order with which function calls [14]. We introduced following six example sequences for our system.

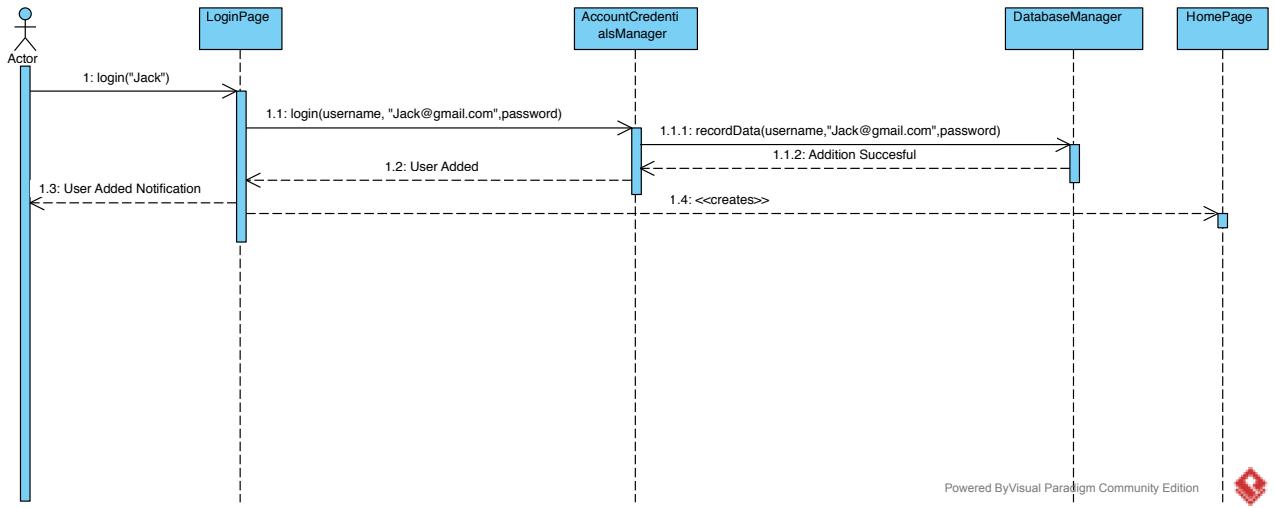
##### 3.5.4.1.1.Sign Up Sequence Diagram



The figure shows information about how “Jack” signs up to use LodeStar. Initially Jack will need to provide his credentials. To do that, the SignUpScreen will be prompted to him. Having entered the credentials, the manager will contact the database to update it for the new user’s credentials. After updating the database, a notification to the user will be given.

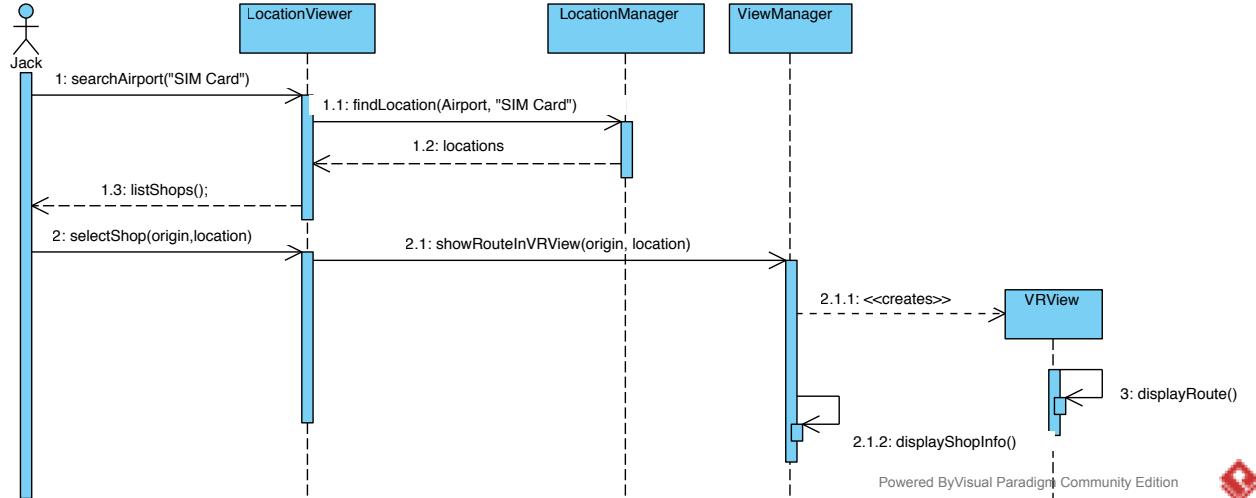
##### 3.5.4.1.2.Log In Sequence Diagram

The figure shows information about how user “Jack” logs in to use LodeStar. Initially Jack will need to provide his credentials. To do that, the LoginScreenView will be prompted to him. Having entered the credentials, the manager will contact the



database to check the user's credentials. After receiving a verification from the database, LoginScreenView will direct Jack to Home Page.

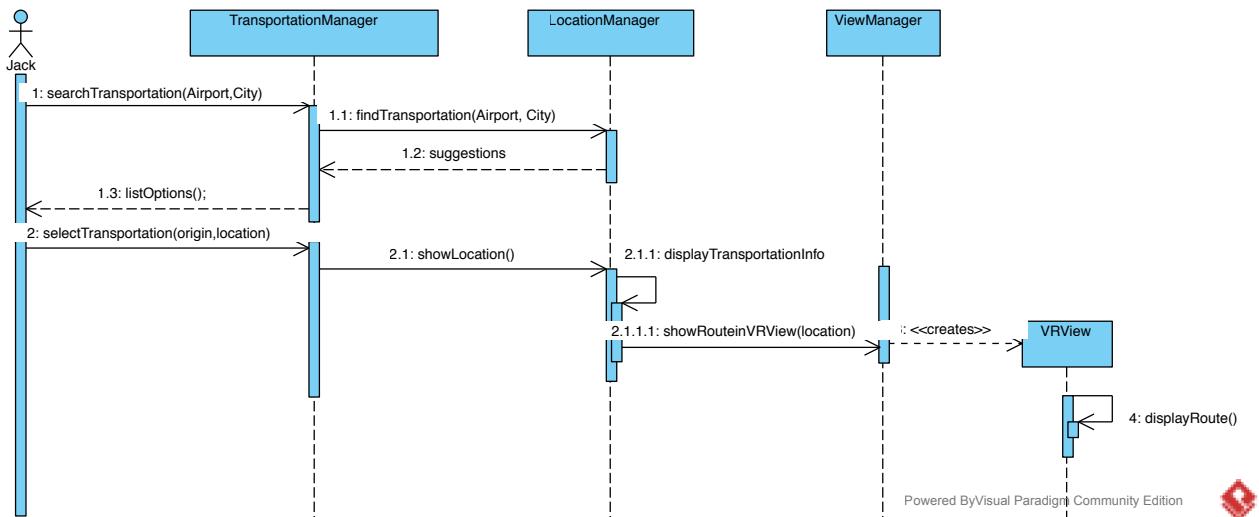
### 3.5.4.1.3.Find SIM Card Information Sequence Diagram



The above sequence diagram displays the program call sequence when a user looks for a SIM card shop during their travel. First, the 'LocationViewer' will search the airport for desired criteria. To accomplish this, it will call 'LocationManager', which will return the locations of SIM card shops in the chosen airport. 'LocationViewer' will list them to the user. Then the user will select a shop which will cause 'ViewManager' to show a route to that shop in VR view. Following this, 'ViewManager' will create a

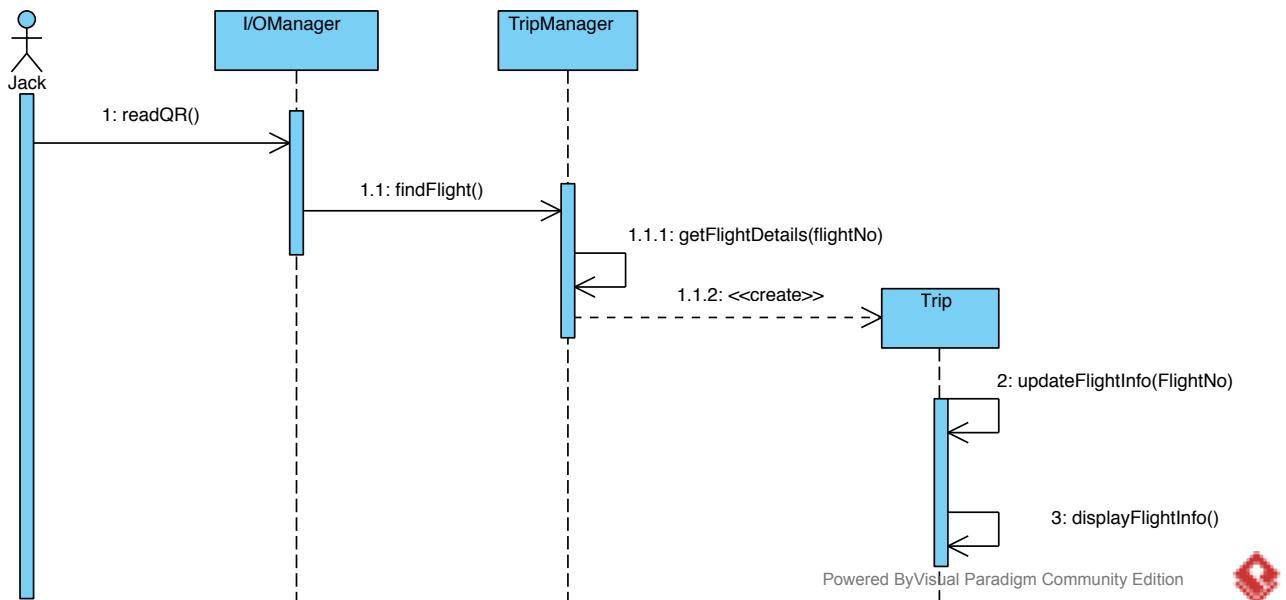
VRView, that will display the route to the shop from a chosen origin point in airport. It will also display information about the shop to the user.

### 3.5.4.1.4. View Transportation Suggestion Sequence Diagram



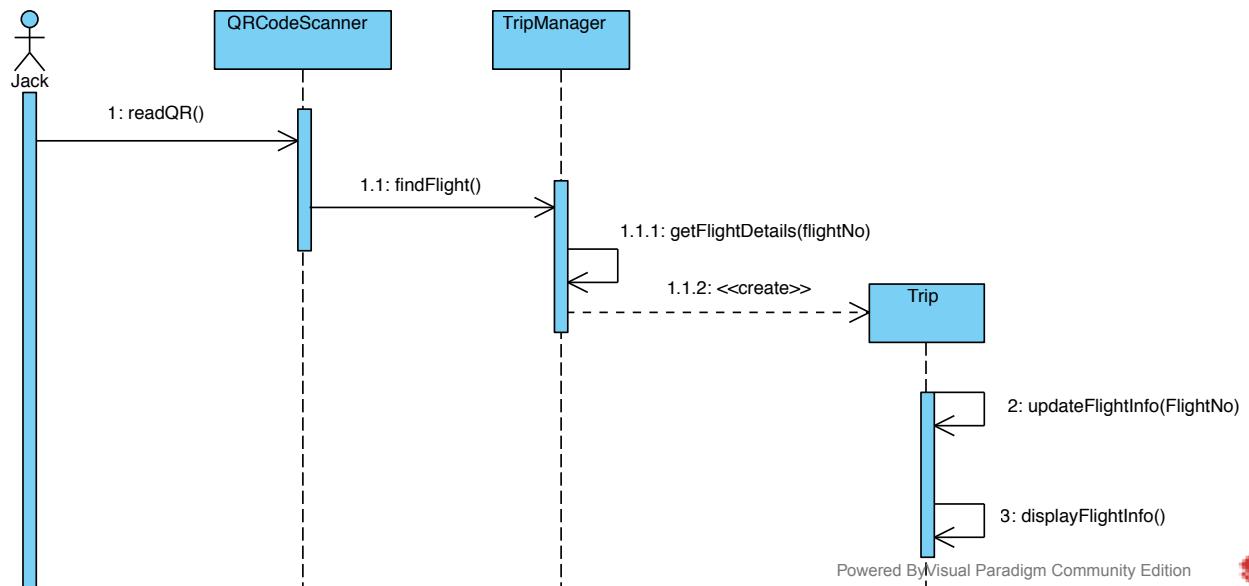
This sequence diagram depicts the program functions when a user requires a transportation suggestion during their travels. First, TransportationManager will be called for possible means of transportation in a certain city for the user's airport. LocationManager will search the stops and terminals that user can choose and returns a list of suggestions. TransportationManager will show them to the user along with their details, like costs. The user will select one of these options. LocationManager will locate the chosen option, display related information and call ViewManager to show the route in VR view. A new VRView will be created and it will display the route to the user.

### 3.5.4.1.5.Travel City Sequence Diagram



The above sequence diagram is intended for the user to travel the tourist attractions at the city of destination. The user initially searches the city using the LocationViewer. Then the manager finds the chosen city and returns the corresponding info (it's Google Street View). LocationViewer will also give the user the option to travel the city virtually.

### 3.5.4.1.6.Scan Flight Ticket Sequence Diagram

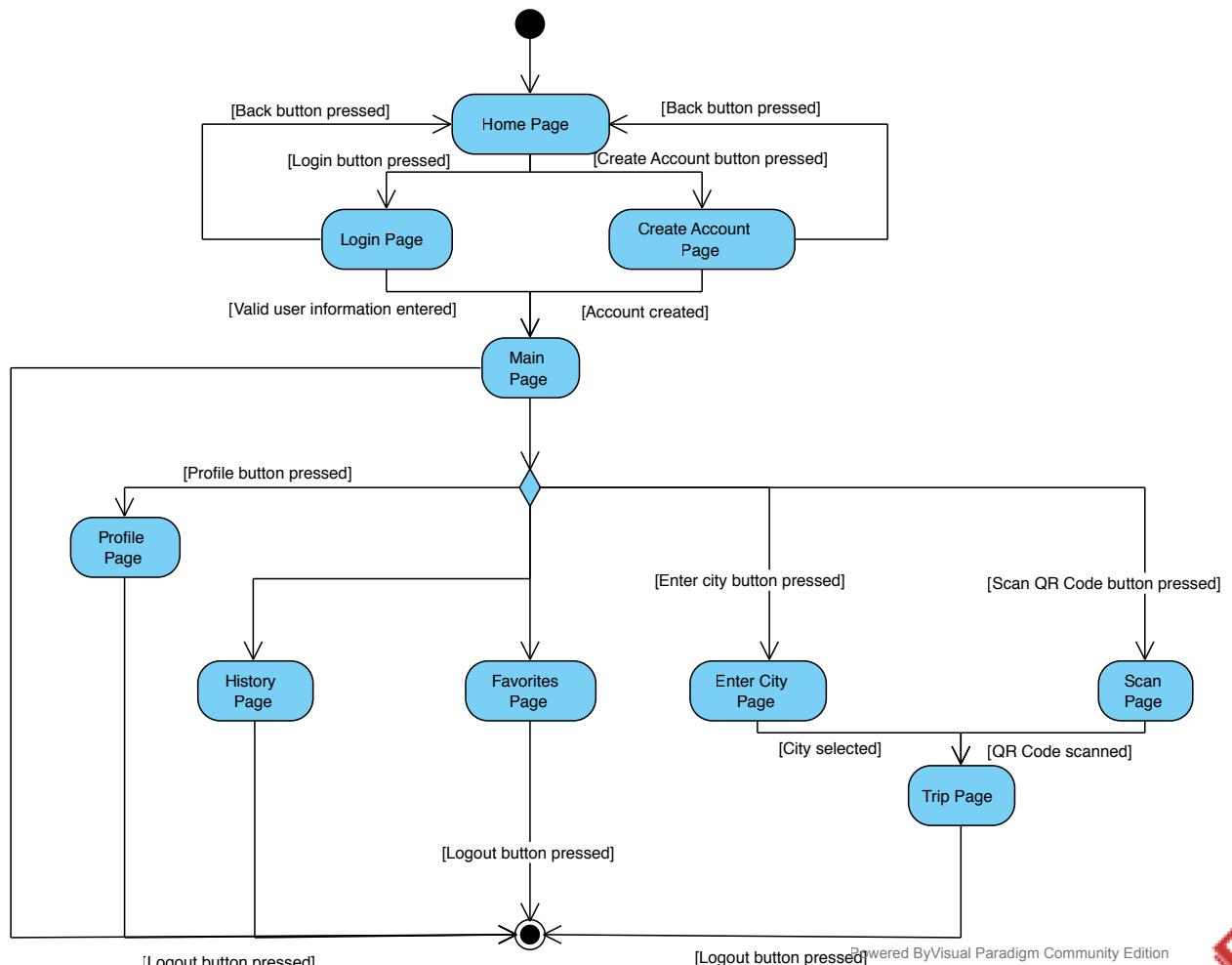


This sequence diagram demonstrates the calls when users scan QR codes on their tickets at the beginning of their trip. When user wants to read their ticket, I/OManager call reads the QR code. Scanned information is sent to the TripManager to find the flight number and other details using the code. TripManager gets the flight details and creates a new Trip. Newly created Trip will be used in the application to update the flight information, so that the user will be kept up to date on delays and such. Also, it will display the flight info for other application functions to use, such as flight notifications.

### 3.5.4.2. Activity Diagrams

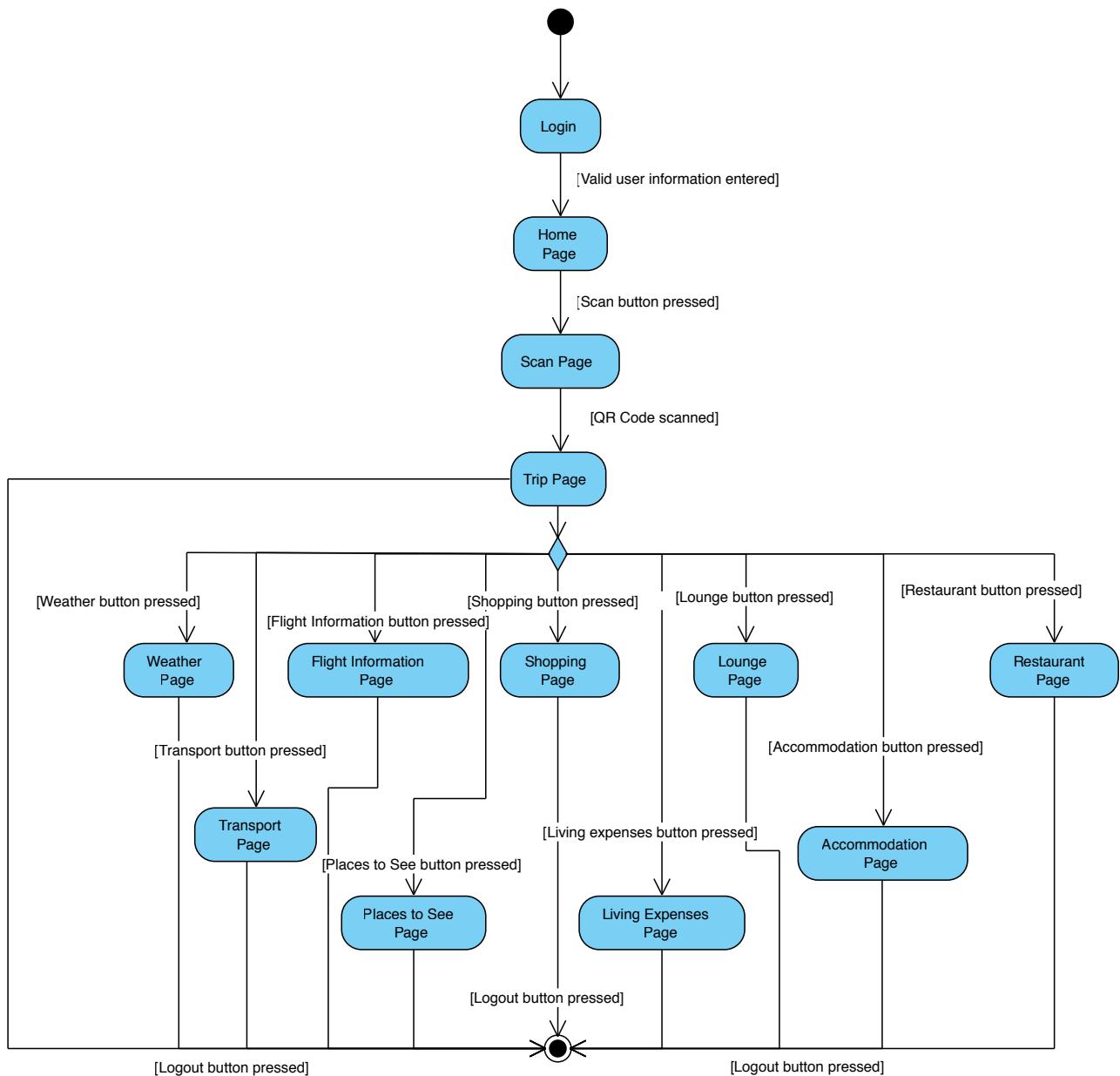
Activity diagram is another important diagram in unified modelling language to describe the dynamic aspects of the system that represents the flow from one activity to another activity[15]. We included following two examples of activity sequences for our system.

#### Application Flow Activity Diagram



LodeStar greets the user in its home page upon starting the application. From there, the user can either login with his/her credentials or set up a new account. Following successful login, LodeStar guides the user to the main page of the application. The user here is granted to choose between managing user profile, visiting history or favorites pages, or viewing information about an upcoming trip. To get to this trip screen, the user needs to either enter the destination city's name or scan the QR Code of a valid ticket. From any screen, it is possible to logout.

## Trip Creating Activity Diagram



This activity flow commences with a successful login into the system by the user. There, the user scans the QR Code of their plane ticket by using his/her smart phone's camera. Then, they are taken to the trip page. Here, the possibilities are endless. The user can check the weather of the destination city. S/he can check the living expenses, look for accommodation, get recommendations about transport or review the places to see. About the airport of interest, the user can see the restaurants or shopping stores available as well as the lounges present. For the flight, they can view all information like duration and expected landing time. From any of the screens, the user can logout.

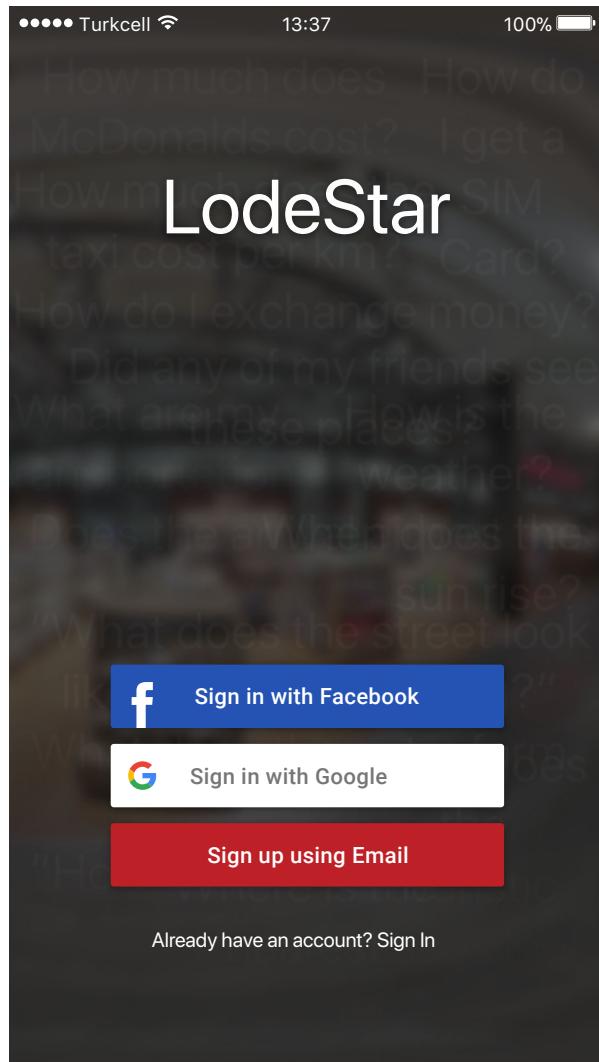
### **3.5.5.User Interface**

In this part of the report, the user-interface of the application will be presented. Each sketch is provided with an explanation to better specify how it address the requirements mentioned above.

#### **3.5.5.1.Login Page**

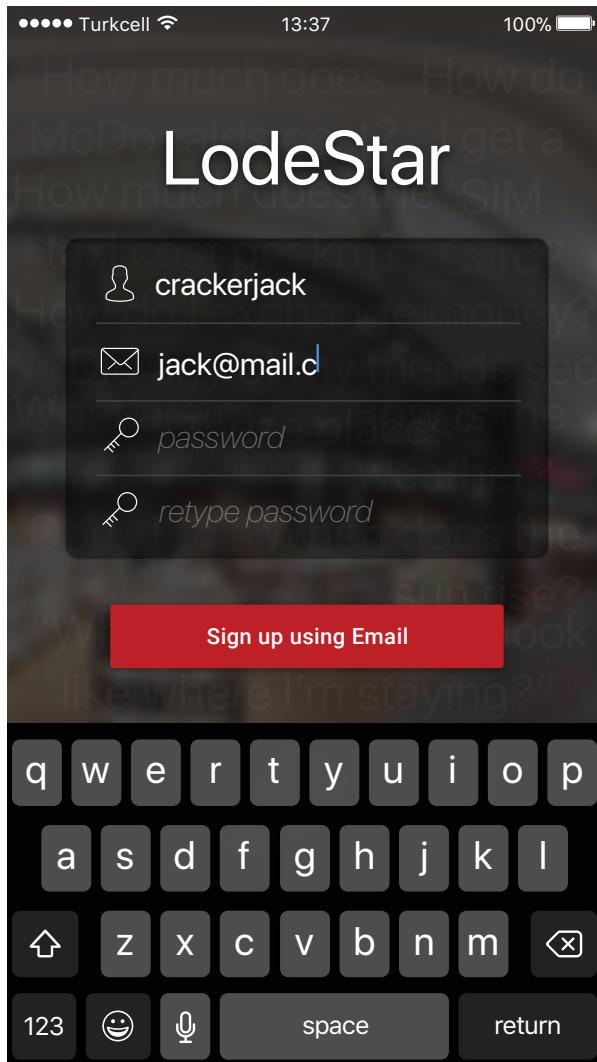
The user will log in to LodeStar from the Login page as showcased below. The user will have three different options. The user will also have an option to sign in using their social media accounts.

If the user already has an account, he will be able to sign into his account after tapping “Already have an account? Sign In” label from the previous page. After doing that, he will be navigated to the following page.



### 3.5.5.2. Sign Up Page

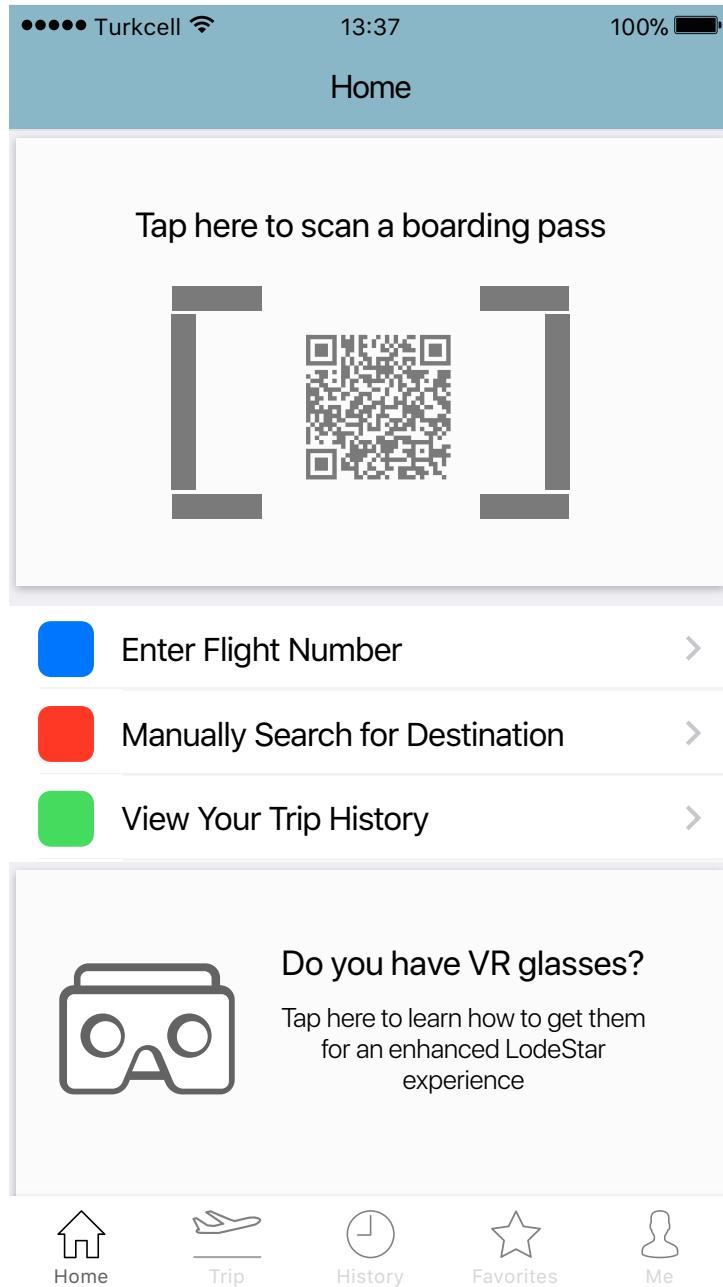
If the user does not wish to sign up using their social media accounts and wants to sign up using their personal email address, then he will be navigated to the following page.



### 3.5.5.3. Home Page

The user will be navigated to this page once they pass the log-in page. The page will have a big button at the top instructing the user to scan a flight card using their phone camera. Or the user will be able to enter a flight number manually if he does not have a flight card yet. (If the user did online check-in or the user basically wishes to view another flight). The user will also be able to manually search for cities just to learn more about them. We do not want to alienate users without flight cards. There might be users who want to learn about different cities without traveling there.

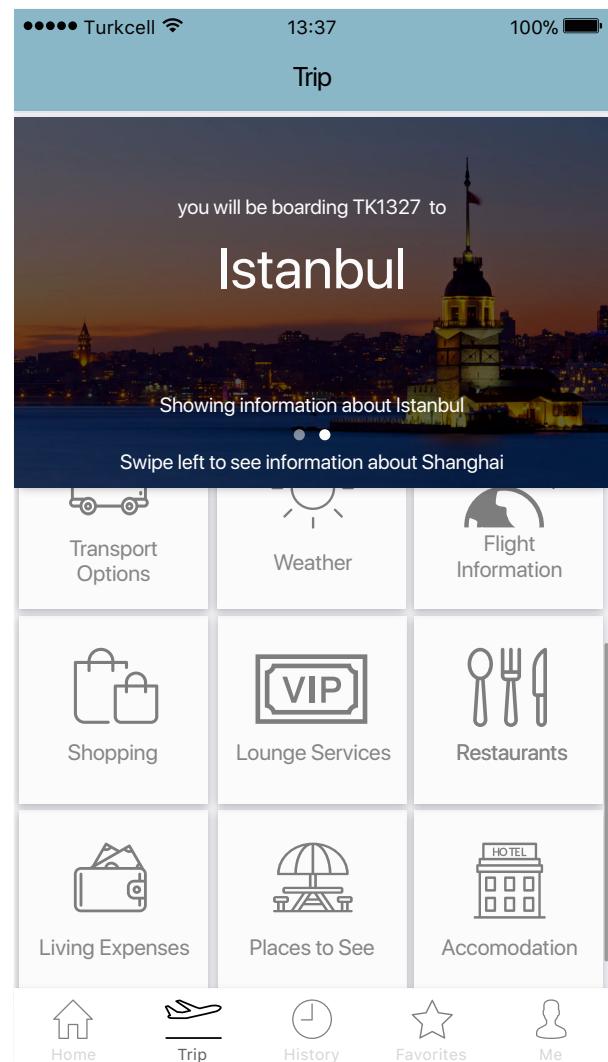
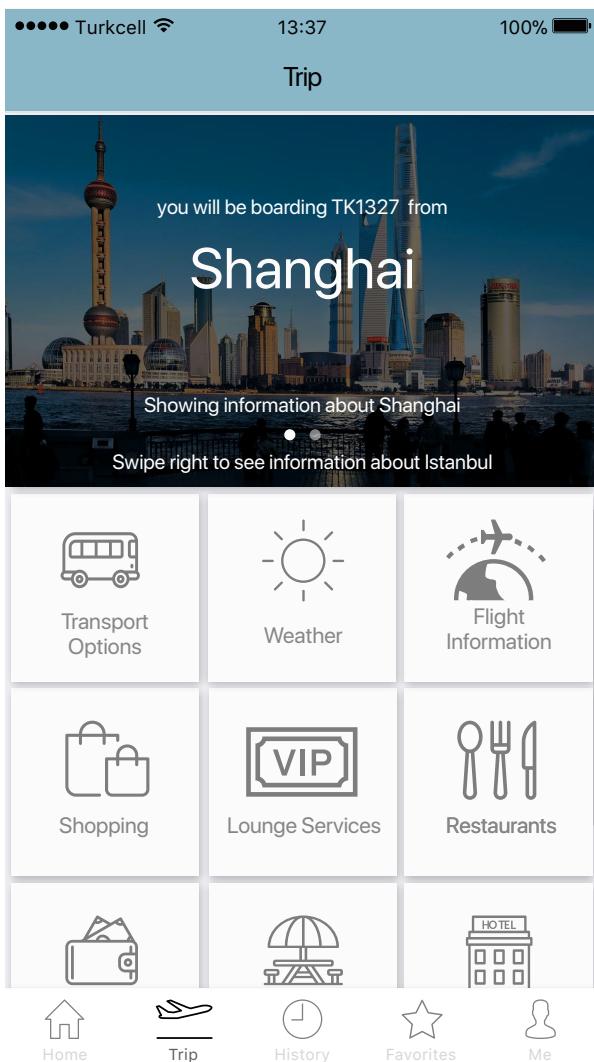
Towards the bottom of the page, a simple card will be shown to inform the user about the fact that LodeStar experience is enhanced with VR glasses. LodeStar will also educate the user about VR glasses and tell them where to get one.



### 3.5.5.4. Trip Page

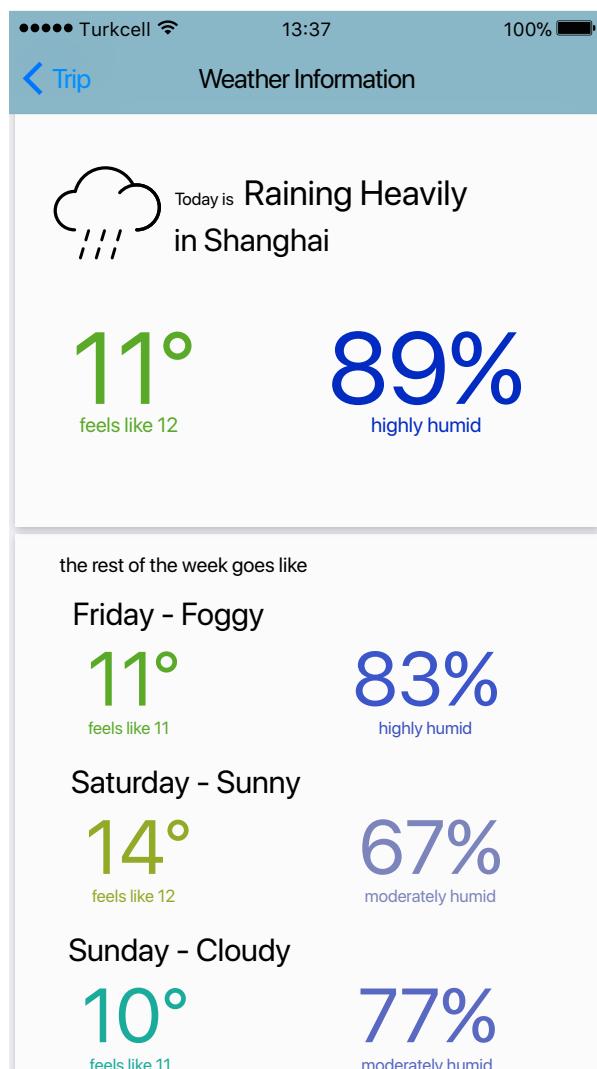
This page will be navigated upon a successful flight card scan. Using the QR code located on the flight card, the departure and arrival cities of the corresponding flight can be learnt. This way, LodeStar will fetch details about these cities and display them to the user in a user-friendly way.

The user will start seeing cards from the departure city. After swiping right from anywhere on the page, the user will start seeing cards about the destination city.



### 3.5.5.5. Weather Information Page

If the user taps on the “Weather” card from the Trip page (see previous page), the user will see the weather information about the corresponding city. Weather information will be shown in a minimalist and simple way, without any clutter. The user only needs to know if they need a coat or an umbrella in the city they are traveling. They do not need to be informed about the speed of wind or how much rain falls per square meter per year.



### 3.5.5.6.Flight Information Page

This page will be navigated to when the user taps on the “Flight Details” card from the “Trip” page. This screen will gather all available data that can be gathered by scanning a flight card. Also, it will display a simple weather summary. Tech savvy users which are interested in the technical details of a flight will be able to learn the model of the plane and view the pictures of the aircraft.

The screenshot shows a mobile application interface for flight information. At the top, there is a header bar with the network provider (Turkcell), time (13:37), battery level (100%), and a back arrow labeled "Trip". The main title is "Flight Details". Below the header, the flight information is displayed in two columns:

SHANGHAI, CHINA LEAVING FROM GATE 208 Shanghai Pudong Int'l - PVG	ISTANBUL, TURKEY ARRIVING TO TERMINAL 1 Istanbul Ataturk Int'l - IST
-------------------------------------------------------------------------	----------------------------------------------------------------------------

Flight times and distances:

11.50PM CST 7.50AM GMT+3	Direct Distance: 4,994mi Planned Speed: 480 mph	10.30AM CST 6.30PM GMT+3
-----------------------------	----------------------------------------------------	-----------------------------

Average Delay: 10-20 minutes

Weather forecast for Istanbul:

It will be Raining Moderately when you arrive Istanbul

Temperature: 16° feels like 15

Humidity: 52% not very humid

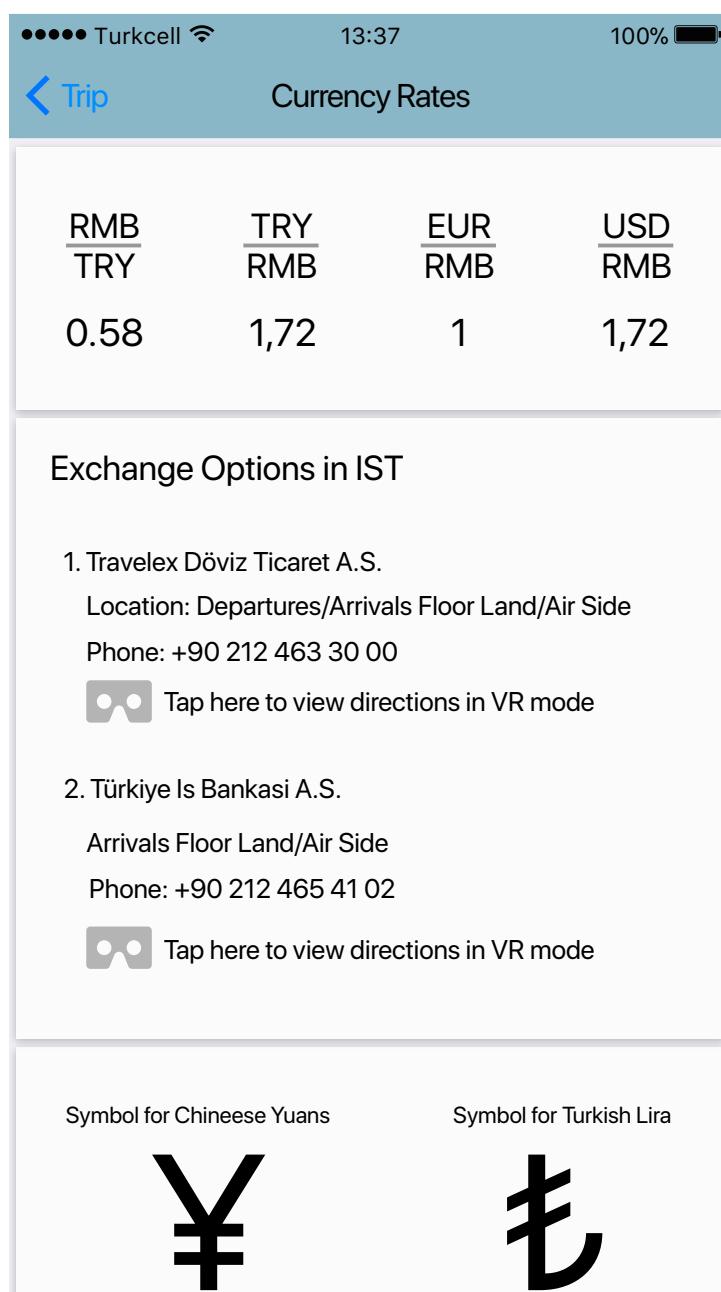
Technical Details:

Aircraft Type	Boeing 777-300ER (twin-jet)
View Aircraft Pictures Online	<a href="#">Tap Here</a>
Planned Altitude	33,000ft

Information from: <https://flightaware.com/live/flight/THY26>

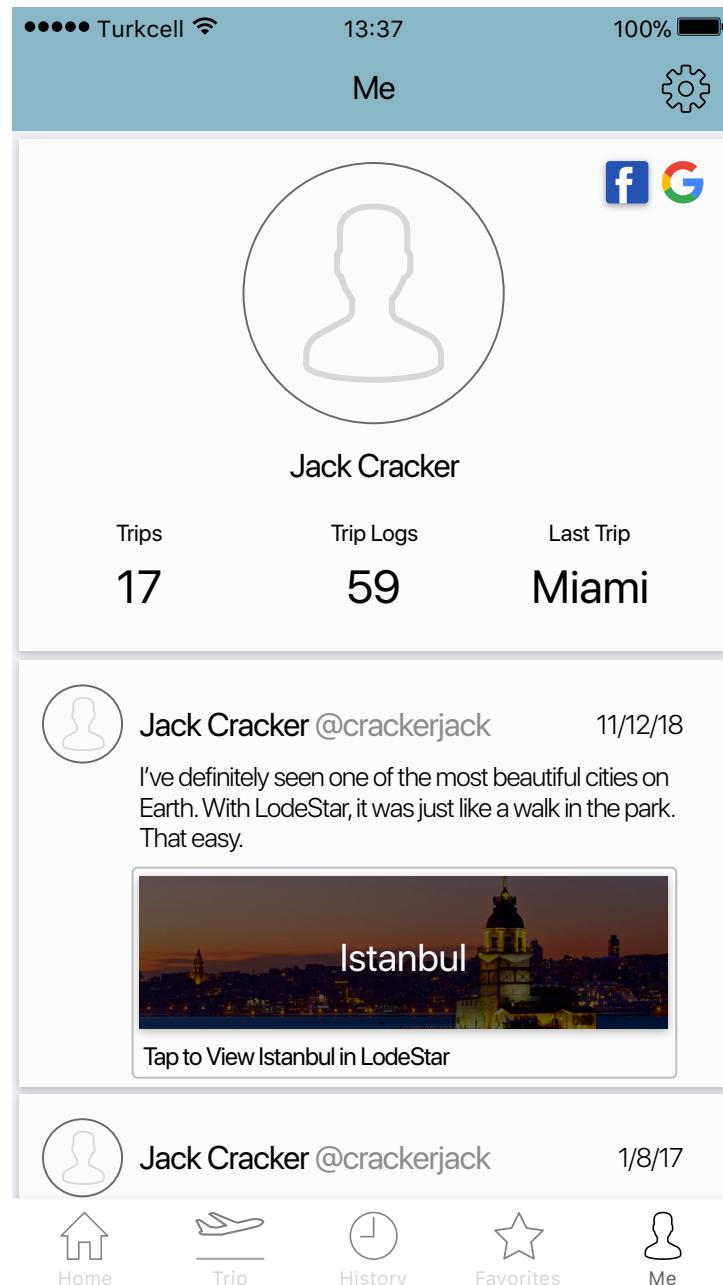
### 3.5.5.7. Currency Rates Page

This page will be navigated to when the user taps on the “Currency Rates” card from the “Trip” page. This screen will show the user cross currency rates between the cities they are traveling. Also, it will display the currency exchange rates for internationally known currencies such as USD and EUR. Towards the bottom of the page, symbols for the currencies will be shown just to inform the user about their symbols.



### 3.5.5.8. Me Page

This page will contain a basic account summary at the top. Google and Facebook icons will be shown if the user has linked their corresponding accounts with LodeStar. Upon tapping the gear icon on the top right, the user will be taken to the preferences page.(see next page)



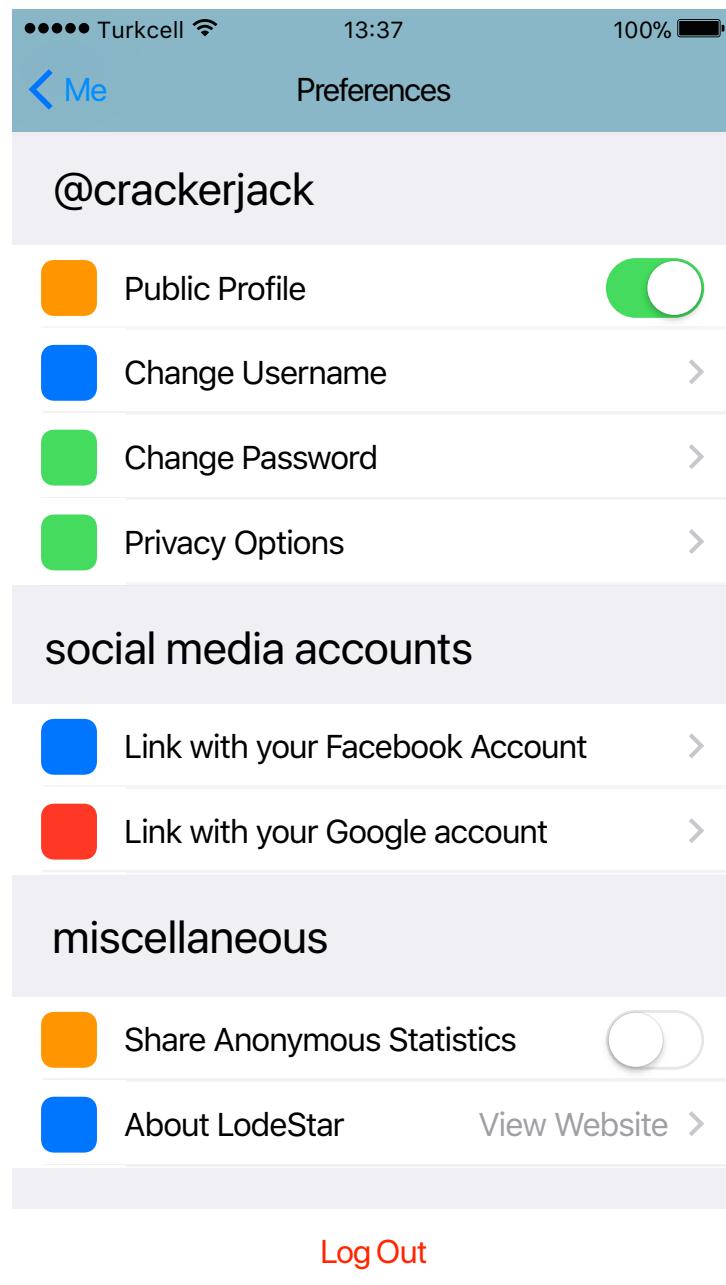
**Trips:** how many times jack scanned flight cards with LodeStar

**Trip Logs:** these will be comments that the user made during their trips. Just like tweeting, the user will write comments about the things they've seen, done, etc.

**Last Trip:** Destination city for the last flight card Jack has scanned.

### 3.5.5.9.Preferences Page

This page will be navigated to once the user taps the gear icon on the “Me” page. This page will allow user to change basic preferences.



## 4. References

- [1] "Food, Nightlife, Entertainment," FourSquare. [Online]. Available: <https://foursquare.com>. [Accessed: 05-Nov-2017].
- [2] "Instant Street View," Google Street View. [Online]. Available: <https://www.instantstreetview.com>. [Accessed: 05-Nov-2017].
- [3] "Maps and Driving Information," Google Maps. [Online]. Available: <https://www.google.com.tr/maps/@39.9033766,32.7627648,11z?hl=en> [Accessed: 05-Nov-2017].
- [4] "Top Things To Do Near Me," TripAdvisor. [Online]. Available: <https://www.tripadvisor.com/Attractions>. [Accessed: 05-Nov-2017].
- [5] "Yelp," Yelp. [Online]. Available: <https://www.yelp.com>. [Accessed: 05-Nov-2017].
- [6] "National and Local Weather Radar, Daily Forecast," Weather.com. [Online]. Available: <https://weather.com/en-GB/>. [Accessed: 05-Nov-2017].
- [7] "Google Cardboard," Google Cardboard – Google VR. [Online]. Available: <https://vr.google.com/cardboard/>. [Accessed: 05-Nov-2017].
- [8] "Flight Status API / Flight Tracking API / FlightAware API → Commercial Services → FlightAware," FlightAware. [Online]. Available: <https://flightaware.com/commercial/flightxml/>. [Accessed: 09-Oct-2017].
- [9] Malan R., Bredemeyer D. Functional Requirements and Use Cases. Architecture Resources For Enterprise Advantage. 2001. [Online]. Available: [http://www.bredemeyer.com/pdf\\_files/functreq.pdf](http://www.bredemeyer.com/pdf_files/functreq.pdf). [Accessed: 06-Nov-2017].
- [10] Chung L., Leite J.C.S.D.P. On Non-Functional Requirements in Software Engineering. Conceptual Modeling: Foundations and Applications. Lecture Notes in Computer Science. 2009. [Online]. Available: [https://doi.org/10.1007/978-3-642-02463-4\\_19](https://doi.org/10.1007/978-3-642-02463-4_19). [Accessed: 06-Nov-2017].
- [11] Chung L., et al. Non-Functional Requirements in Software Engineering. Springer Science+Business Media New York. 2000.
- [12] SearchSoftwareQuality. (2017). What is use case? - Definition from WhatIs.com. [online] Available at: <http://searchsoftwarequality.techtarget.com/definition/use-case> [Accessed 4 Nov. 2017].
- [13] www.tutorialspoint.com. (2017). UML - Class Diagram. [online] Available at: [https://www.tutorialspoint.com/uml/uml\\_class\\_diagram.htm](https://www.tutorialspoint.com/uml/uml_class_diagram.htm) [Accessed 4 Nov. 2017].
- [14] En.wikipedia.org. (2017). Sequence diagram. [online] Available at: [https://en.wikipedia.org/wiki/Sequence\\_diagram](https://en.wikipedia.org/wiki/Sequence_diagram) [Accessed 2 Nov. 2017].

[15] www.tutorialspoint.com. (2017). UML - Activity Diagrams. [online] Available at: [https://www.tutorialspoint.com/uml/uml\\_activity\\_diagram.htm](https://www.tutorialspoint.com/uml/uml_activity_diagram.htm) [Accessed 6 Nov. 2017].