



CZ2006 Software Engineering Lab 2 Report

Version 1.2

Prepared by: Runtime Error - Group SS7, Team 2

Datta Anusha

Ravishankar Amrita

Chua Clarita

Yeoh Sue Min Sarah

Lim Jun Hong

Kumar Mehul

2nd April, 2020

Table of Contents

Table of Contents.....	ii
Revision History.....	ii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions.....	3
1.4 Product Scope.....	4
1.5 References.....	4
2. Overall Description.....	5
2.1 Product Perspective.....	5
2.2 Product Functions.....	5
2.3 User Classes and Characteristics.....	6
2.4 Operating Environment.....	7
2.5 Design and Implementation Constraints.....	7
2.6 User Documentation.....	8
2.7 Assumptions and Dependencies.....	9
3. External Interface Requirements.....	10
3.1 User Interfaces.....	10
3.2 Hardware Interfaces.....	14
3.3 Software Interfaces.....	14
3.4 Communications Interfaces.....	15
4. System Features.....	16
5. Other Nonfunctional Requirements.....	29

5.1 Performance Requirements.....	29
5.2 Safety Requirements.....	29
5.3 Security Requirements.....	30
5.4 Software Quality Attributes.....	30
6. Other Requirements.....	31
Appendix A: Use Case Diagram and Description.....	
Appendix B: Analysis Models.....	
Appendix C: Future Works.....	

1. Introduction

1.1 Purpose

The Android and iOS mobile application, '**CrafTrip**' aims to ease the process of deciding on a travel destination. It is an app that helps users narrow down their options of which city to visit. It is a one-stop destination for all the confused travel enthusiasts out there. CrafTrip serves the purpose of:

1. Displaying important information about these cities:

The application allows the users to view the summary of a city at its destination card, in order to know more information about the city, such as flight details to the city, weather forecast of the city, the rate of exchange between Singapore Dollar and the currency used at the destination.

2. Generate travel recommendation according to users' previous right swipes:

The application will display a number of cities in the form of 'cards' and the user has the option to swipe right - indicating the user is interested in a city and left - indicating the user is not interested in visiting the city. Based on the user's right swipes, the application will generate and suggest the top 3 recommendations for the user

1.2 Document Conventions

The entire document is in font-style Arial. The main headings of the document are size 23. The subheadings of the document are size 17. The main and sub headings both have bold formatting. The rest of the document is regular, size 11. There are a total of 6 sections. Within each section, there are sub sections that are ordered in the form of {section number}.{sub-section.number}. "CrafTrip" has been referred to as the 'system' in the functional requirements. Priorities for higher-level requirements are assumed to be inherited by detailed requirements. The data documentation follows the below terminology:

<u>TERM</u>	<u>DEFINITION</u>
User	The traveler who is using the application to check various travelling information, like flight prices, weather and currency exchange rate.
Travel Destinations	The list of places the user can choose from, as a travel destination.

Forex Rate	Current Foreign Exchange rate of Singapore Dollar to any travel destination's currency chosen by the user.
Weather	The weather refers to the current climatic conditions of any place selected by the user, including temperature, humidity and general description.
Weather Icon	The weather icon describes the current weather conditions such as 'Cloudy', 'Sunny' or 'Rainy', of the selected travel destination.
Summary	The complete analysis includes the weather, currency exchange rates, and flight prices together as a single view for the destination selected by the user.
Cards	The swipe-able view of Travel Destinations that the user sees on the Travel Picks page which displays destination's name, photos and tags.
Tags	The one-word tags derived from the Google Vision API describing a travel destination.
Swipe - Like and Dislike	The user can 'swipe' on the cards containing the travel destinations. Left swipe is for 'dislike' which discards that particular destination, and a right swipe is for a 'like' which saves that particular destination in the 'History' page. The like and dislike can also be done via buttons.
Favourites	A page where the travel destinations that have been 'favorited' by the user, by selecting the 'heart' icon on the Travel Picks or History page, can be viewed.
View More	A button that expands the respective information tile and allows the user to dive deeper and view more information, like a 'view more' button for flight prices will show flight prices for all airlines and give the user an option to choose their travel dates as well.

History	A page which shows all the travel destinations that have been right-swiped or liked by the user on the Travel Picks page.
Tag Hit	The number of times the user has right-swiped or liked a travel destination which contains a particular tag. The total number of likes on a tag is the total tag hits for that particular tag.
Recommendations	The top 3 suggested travel destinations for the user, based on the tags which have the highest hits, and the travel destinations which have been assigned those tags by the Google Vision API.
Favourite and Unfavourite	The user can 'star' or 'un-star' any travel destination on the Travel Picks or History page, by using the 'heart' icon, which essentially adds or removes the travel destination from the Favourites page.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, users, clients, marketing directors and UI/UX designers. The above mentioned user groups are encouraged to read the documentation in sequential order and use the appendix given to fully understand the architecture and functioning of the 'CrafTrip' application.

The remaining of the SRS contains information regarding the assumptions made, design and implementation constraints, the various external interfaces (eg. software and hardware) and essential system feature details.

Below are the sections most pertinent to each user group:

- Developers: The operating environment, external interface requirements, system features and non-functional requirements.
- Quality assurance and testing team: Functional requirement to generate the use case scenarios. This group is advised to refer to the "System Features" section of the proposal.
- UI/UX designers: Design and implementation constraints and UI Mock-ups to fully understand the client's vision.
- Marketing team: UI Mock-ups along with the introduction section to generate the marketing plan based on those features.
- Users/Clients: System features to understand the exact working of the application.

1.4 Product Scope

CrafTrip is an Android mobile application that serves a large audience as a one-stop destination for deciding on a travel destination. We provide them insightful information instantaneously about the destinations to assist them in making a decision.

The application allows users to view flights from Singapore to travel destinations, to view the weather forecast of destinations and to check the currency exchange rate between Singapore Dollar (SGD) and the destination's currency. Users will be able to find money changers that are located in Singapore to exchange money as well. The system is based on Firebase and API calls to provide necessary information about every travel destination. Overall, the application allows indecisive travel enthusiasts to decide on a travel destination.

1.5 References

[1] "ISO/IEC/IEEE International Standard – Systems and software engineering – Life cycle processes –Requirements engineering," ISO/IEC/IEEE 29148:2011€, pp. 1–94, Dec. 2011.

[2] "Usage and limits | Firebase. (n.d.)" <https://firebase.google.com/docs/firestore/quotas> Mar.2020

2. Overall Description

2.1 Product Perspective

Craftrip is a new standalone application aimed towards people who want to travel out of Singapore. Craftrip feeds the user travel destinations in the form of a card allowing them to swipe on the cards according to whether they like it or dislike it. Login is required as the app has components unique to the user. We store the user's favourite destinations which are travel destinations that users particularly like and we store the user's preferences as well.

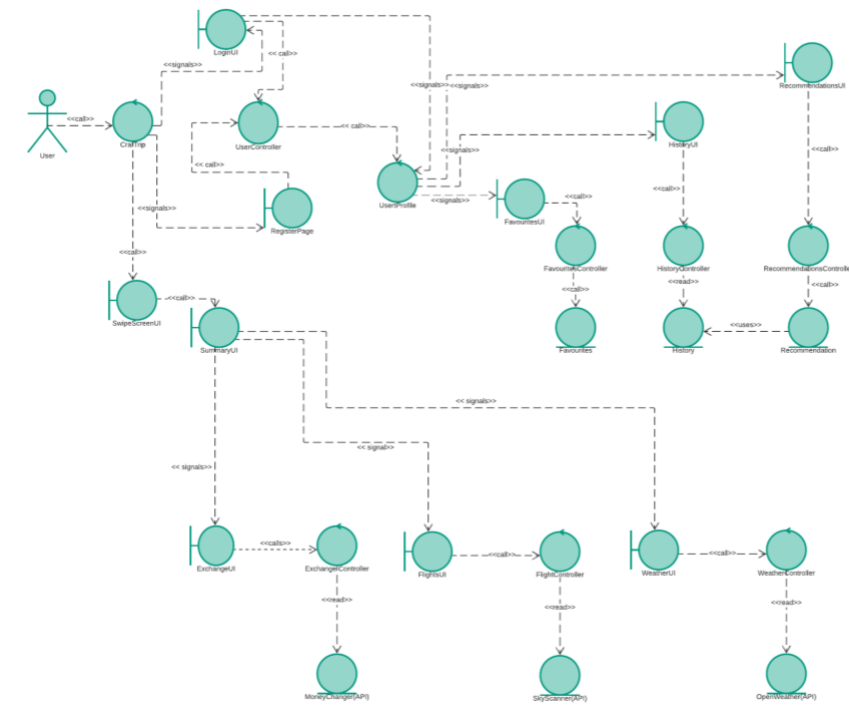


Fig 1. CrafTrip Conceptual Diagram

2.2 Product Functions

The major functions of the system are:

- 1. Display flight prices**
Craftrip will obtain flight prices via Skyscanner API for flights between a user's choice of travel destination and Singapore on dates selected by the user.
- 2. Display currency exchange rates**
Craftrip will display the current exchange rate of the travel destinations relative to Singapore and be able to view the nearest money changers around Singapore.

Craftrip will display the current weather at the travel destinations and weather forecast for the next 5 days obtained from OpenWeather API.

Craftrip will allow users to choose their preferred/desired tag their travel destinations should have eg. cultural.

Craftrip stores travel destinations that are tagged with different characteristics which are used in the system's algorithm to sort the initial arrangement of the travel destinations shown to the user according to their preference.

Craftrip will recommend to users travel destinations based on their past swipes to show them travel destinations they might be interested in.

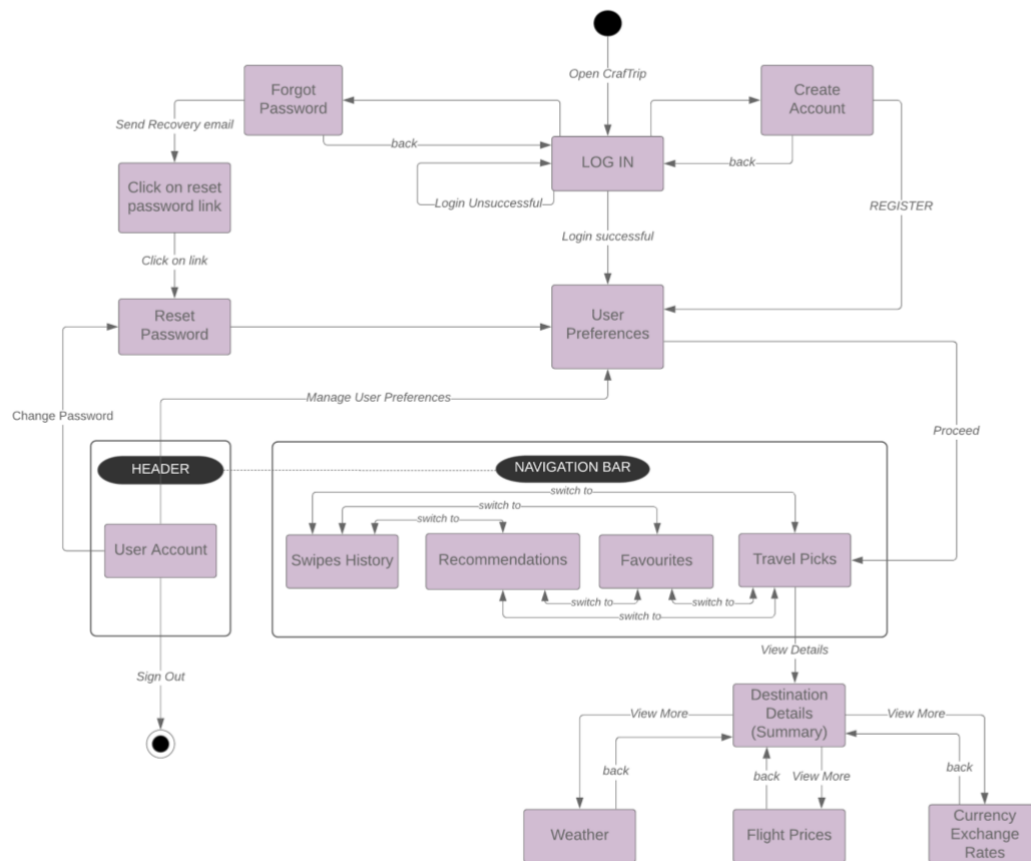


Fig 1. CrafTrip Dialog Map

2.3 User Classes and Characteristics

USER CLASSES	CHARACTERISTICS
Travel Bloggers	<p>Travel Bloggers travel frequently around the world throughout the year.</p> <p>Frequency: High as they would use this app more frequently than other user classes as they would need to travel more often based on their profession.</p> <p>Subsets of product functions used: Travel Bloggers usually want to know the aesthetics and the atmosphere of a travel destination before picking it, the travel destination's image and tags help with the same. Further the travel recommendations provided by our application help travel bloggers view other destinations that might match their individual style.</p>
Family Travellers	<p>Family travellers are those who travel with their family.</p> <p>Frequency: Usually these users only travel during peak holiday seasons, so their frequency of use of this app would be lesser than travel bloggers.</p> <p>Subsets of product functions used: They benefit the most from this application as they can immediately access the currency exchange rate, flight prices and weather for a destination through a single platform. These users usually want to know everything about the location so as to have a comfortable stay. The 'User Tutorial' within the app is most useful to these users as they are less experienced with the working of modern apps and their features.</p>
Student Travellers	<p>Student travellers like to take quick trips to destinations around their university city.</p> <p>Frequency: Similar to 'Family Travellers', these users travel during holiday seasons too. Thus their frequency of use is similar to Family Travellers.</p> <p>Subset of product functions used: Student travellers usually need to travel on a budget and would like a low-cost trip. CraftTrip provides them with instant access to the exchange rates and flight prices thereby allowing them to make decisions accordingly. Our application allows 'Flight prices' to be sorted from low-high, further assisting the users.</p>
Programmers	<p>Frequency: As and when required by the programmers, thus frequency would be less.</p>

	<p>Other programmers can develop the project in the future, to further improve on its features or extend it to other travel related applications or functionalities.</p> <p>Their technical expertise would be the highest compared to the other user sections.</p>
--	---

Amongst the above user classes, our most important user class would be the 'Travel Bloggers'. We would want to satisfy this user class the most as their frequency of use is the highest.

2.4 Operating Environment

The following table describes the operating environments in which the application will operate in:

OE-1	Application will require a stable Internet connection to perform all its functionalities without disruption.
OE-2	CrafTrip app currently supports Android OS running Android Lollipop (API level 21) and above.
OE-3	CrafTrip app currently supports iOS running iOS 9.0 and above.
OE-4	Application will utilize Firestore to generate, store as well as query data, in real-time.
OE-5	The platform used: Flutter, version: 1.12.13+hotfix.8

2.5 Design and Implementation Constraints

CrafTrip contains many features, mainly displaying weather information, exchange rates and flight information. Hence, these features require the support of API calls which would have certain timing requirements:

1. Weather - OpenWeatherAPI: 2 seconds
2. Exchange - exchangeratesAPI: 3 seconds
3. Flight - SkyscannerAPI: 5 seconds

Memory requirements are not needed because we retrieve information directly from the respective API and hence, storing of the various data would not be needed. Also, realtime

information is used so the API calls would be made every time the user requests for these information.

A firebase database and firebase authentication is still needed to store user information and preferences, as well as the travel destinations displayed as mentioned in section 2.1. The storage limitation on firestore is 1GiB which is still sufficient for current developments. However, this might hinder future improvements and scalability. Moreover, other limitations include:

1. Document reads - 50,000 per day
2. Document writes - 20,000 per day
3. Document deletes - 20,000 per day
4. Network egress - 10GiB per month

These limitations only hold for a free account and hence any future developments of CrafrTrip might require monetary cost to increase the size and functions of the database.

Furthermore, the collection of user data such as email, first name and last name needs to remain confidential and hence in order for CrafrTrip to be pushed into the market, the approval from relevant authorities on the confidentiality of customer information has to be obtained.

If the customer's organisation is responsible for maintaining the software, flutter SDK and dart programming language should be used.

2.6 User Documentation

To ensure users and stakeholders understand the functions of CrafrTrip, a tutorial will be shown after the user logs in. The tutorial consists of labels of the navigation bar and how to swipe a travel card. A demo video with detailed explanation of each feature is also provided to stakeholders for a deeper understanding of the entire process of CrafrTrip. The link will be provided when the application is downloaded

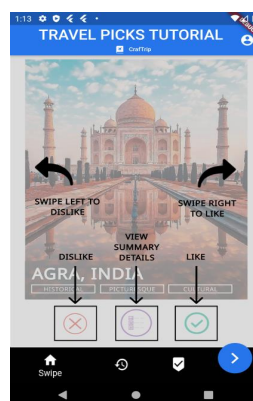


Fig 1. CrafrTrip Tutorial screenshot

2.7 Assumptions, Constraints and Dependencies

Assumptions:

1. Users have Internet access in order to use the application.
2. For the free version of the OpenWeather API, the maximum and minimum air temperature available is the maximum and minimum temperature across the city at the current time and not the maximum and minimum the temperature will reach in the entire day.
3. For currencies that were not available in the ExchangeRatesAPI, we assumed that US Dollar (USD) would be accepted in those cities.

Constraints:

1. Due to the present Covid-19 situation, the flights to various cities might be either unavailable or limited. However as the situation improves over the next few months, our application will serve its purpose to the fullest.
2. For the free version of the SkyScanner API, we are allowed to make 10 queries a day, that is, we can have live access to flight data only 10 times a day. Our back-up plan for the same is to make use of the old cache data made available by the API itself.
3. The application is currently available only in English.
4. All API calls retrieve data in JSON format, hence it becomes a restriction to developers formatting the data.

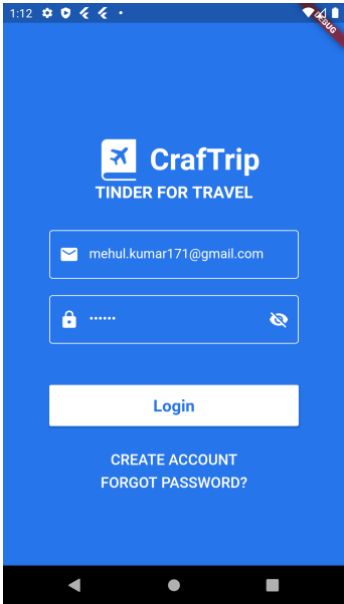
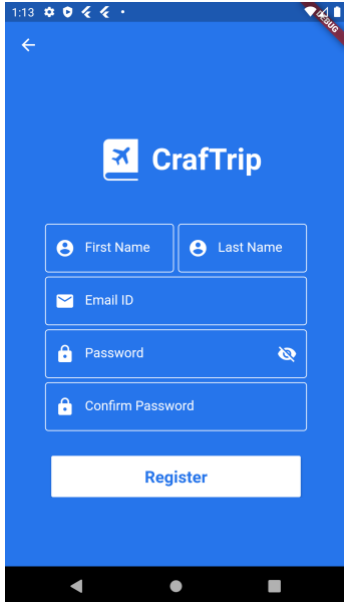
3. External Interface Requirements

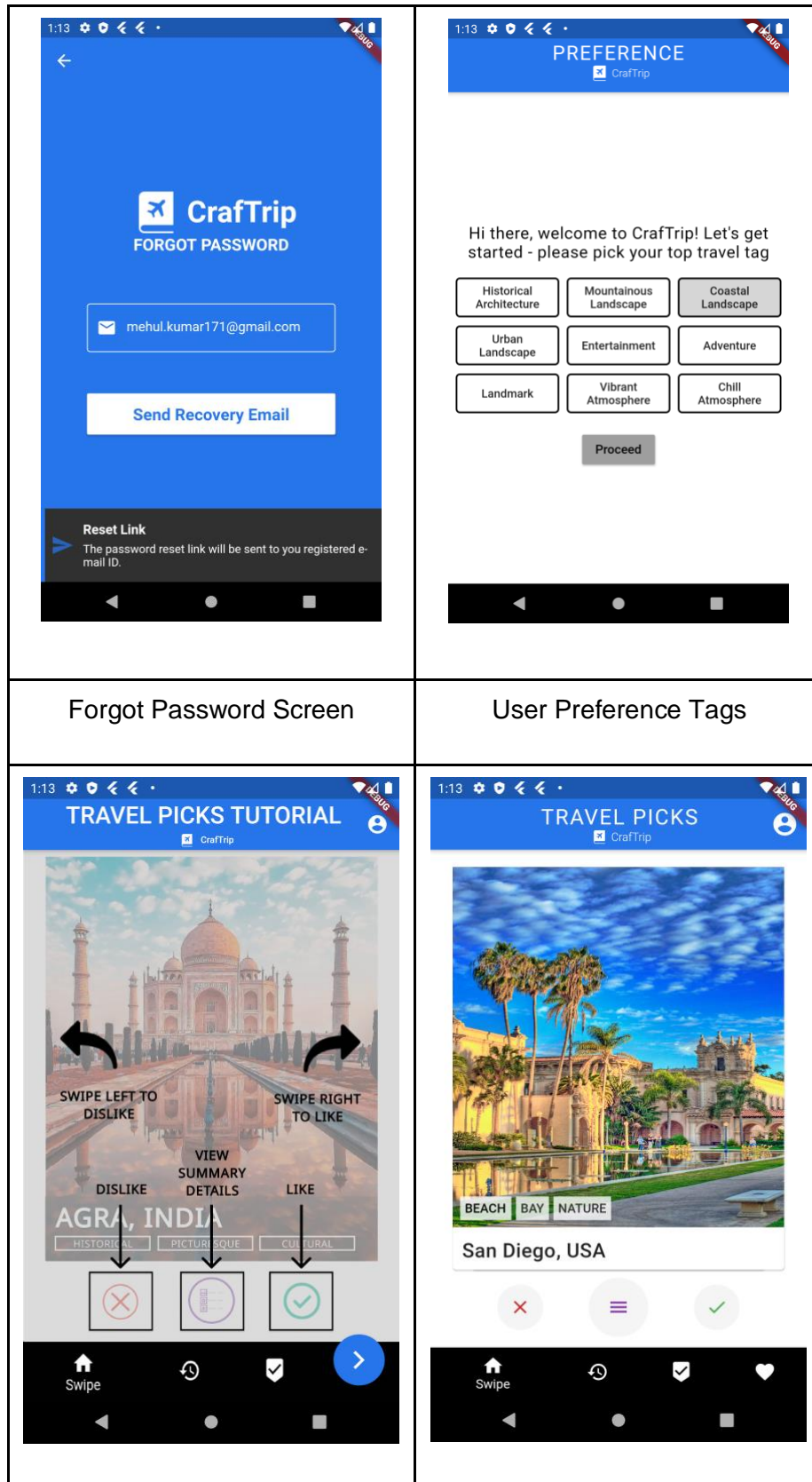
3.1 User Interfaces

The entire User Interface has been kept minimal and straightforward. The UI follows a blue and white theme all across the application. The app-bar at each screen is consistent and a back button exists at the top left corner on each screen. The bottom navigation bar allows users to switch between tabs at any point of time while using the app. The layout of the destination cards displayed within Favourites, Recommendations and History is consistent. The user interacts with the travel destination cards either by swiping left or right or by clicking on the cross and check button, to dislike or like the card.

The summary button in the middle has a 'list' icon on it thus making it intuitive for the user to understand that on clicking it the user will be able to view details about the destination. The buttons placed at the bottom of the swipe screen are easy to understand thus making the application easy to use.












On clicking on one of the travel destination cards within Favourites, Recommendations or History the user is redirected to the summary page of the travel destination, thereby allowing the user to view details instantly. The above allow the user to have a seamless and effortless user experience.

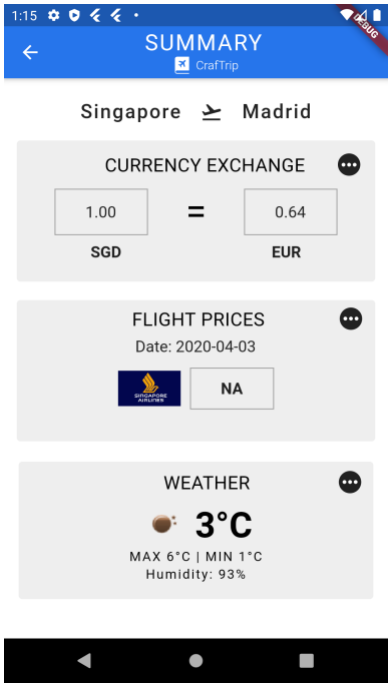


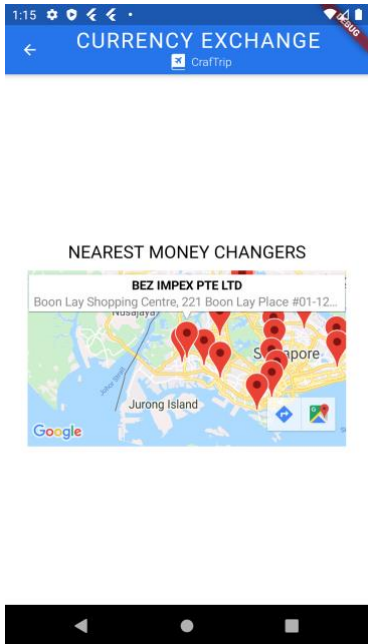
 <p>The login screen features a blue background with the 'CrafTrip' logo and tagline 'TINDER FOR TRAVEL' at the top. Below the logo, there is a text input field for the email ID containing 'mehul.kumar171@gmail.com', a password input field with a lock icon and a toggle for visibility, and a white 'Login' button. At the bottom, there are links for 'CREATE ACCOUNT' and 'FORGOT PASSWORD?'.</p>	 <p>The register screen has a blue background with the 'CrafTrip' logo at the top. It includes four input fields: 'First Name', 'Last Name', 'Email ID', and 'Password' (with a lock icon and visibility toggle), followed by a 'Confirm Password' field. A white 'Register' button is positioned at the bottom.</p>
Login Screen	Register Screen



Forgot Password Screen

User Preference Tags

<div>Tutorial Scene</div> <div><div><div>1:14</div><div>⚙️ 🔍 ⏪ ⏩ ⏴ ⏵</div><div>TRAVEL PICKS</div><div>CraTrip</div><div>👤</div></div><div><div>HARBOR CITY DOWNTOWN</div><div>Sydney, Australia</div><div>❌ ☰ ✅</div><div>Swipe ↺ ✔️ ❤️</div></div></div>	<div>Travel Picks Card</div> <div><div><div>1:14</div><div>⚙️ 🔍 ⏪ ⏩ ⏴ ⏵</div><div>HISTORY</div><div>CraTrip</div><div>👤</div></div><div><div><div>Athens</div><div>Temperature 13.0°C</div><div>Exchange 1SGD = 0.64EUR</div><div>❤️</div></div><div><div>Bali</div><div>Temperature 27.0°C</div><div>Exchange 1SGD = 11625.73IDR</div><div>❤️</div></div><div><div>Bangkok</div><div>Temperature 34.0°C</div><div>Exchange 1SGD = 23.02THB</div><div>❤️</div></div><div><div>Beijing</div></div><div>🏠 ↺ ✔️ ❤️ History</div></div></div>
<div>Travel Picks Card</div> <div><div><div>1:14</div><div>⚙️ 🔍 ⏪ ⏩ ⏴ ⏵</div><div>FAVOURITES</div><div>CraTrip</div><div>👤</div></div><div><div>FAVOURITES</div><div><div><div>Bangkok</div><div>Temperature 34.0°C</div><div>Exchange 1SGD = 23.02THB</div><div>❤️</div></div><div><div>Chicago</div><div>Temperature 3.0°C</div><div>Exchange 1SGD = 0.7USD</div><div>❤️</div></div><div><div>Dubai</div><div>Temperature 21.0°C</div><div>Exchange 1SGD = 0.7USD</div><div>❤️</div></div></div><div>🏠 ↺ ✔️ ❤️ Favourites</div></div></div>	<div>History Screen</div> <div><div><div>1:14</div><div>⚙️ 🔍 ⏪ ⏩ ⏴ ⏵</div><div>RECOMMENDATIONS</div><div>CraTrip</div><div>👤</div></div><div><div>Cultural City Urban</div><div>Top recommendations for you, based on your most swiped tags!</div><div><div><div>Tokyo</div><div>Temperature 19.0°C</div><div>Exchange 1SGD = 74.93JPY</div><div>❤️</div></div><div><div>New York</div><div>Temperature 6.0°C</div><div>Exchange 1SGD = 0.7USD</div><div>❤️</div></div><div><div>Madrid</div><div>Temperature 3.0°C</div><div>❤️</div></div></div><div>🏠 ↺ ✔️ ❤️ Recommendations</div></div></div>

<p>Favourites Screen</p> 	<p>Recommendations Screen</p> 
<p>Summary Screen</p> 	<p>Weather Screen</p> 
<p>Currency Screen Part 1</p>	<p>Currency Screen Part 2</p>

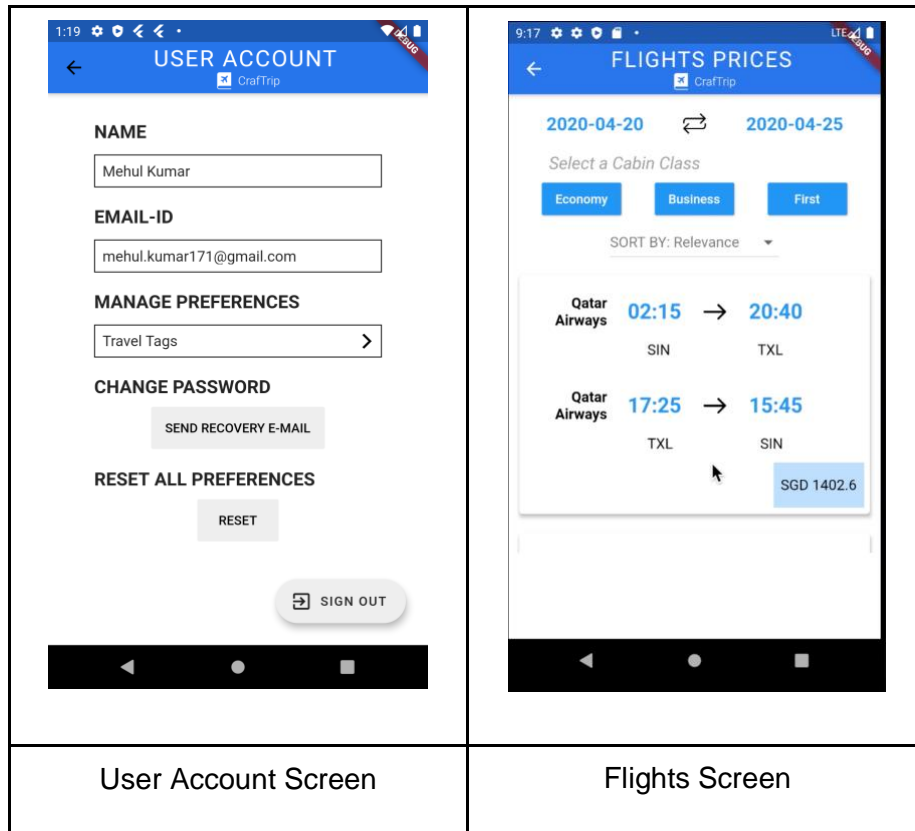


Fig 1. CraTriP UI

3.2 Hardware Interfaces

CraTriP app currently supports Android OS running *Android Lollipop (API level 21)* and above. CraTriP app also supports iOS running *iOS 9.0* and above. The application does not require any external hardware devices to run the application.

3.3 Software Interfaces

CraTriP makes use of Firebase for authentication, database and storage purposes.

- FirebaseAuth authentication function is used for user login, to create a new user account and to verify login credentials.
- Firestore database is used to store information about the travel destination
 - City name
 - Country
 - 'Tags' for the city
 - Currency used in the city
 - URLs for the image that gets displayed on the card.
- Firestore database is also used to store user information such as

- Users' first and last name
- Preferences
- Favourites, history and recommendations.
- Firestore is used for storing the images of the travel destination cards.

CrafTrip also makes use of the SkyScannerAPI, ExchangeRatesAPI and the OpenWeatherAPI. These APIs are used to retrieve important information about the cities.

- ExchangeRatesAPI makes use of the currency code stored in the Firebase database as a 3 letter code (e.g. USD or SGD) to make the API call.
- OpenWeatherAPI uses the city name as the parameter to make API calls.
- SkyScannerAPI is called by providing the cityID, arrival date and departure date.

3.4 Communications Interfaces

To allow the user to change their password, the application needs to communicate with the user's email through Firebase. To access the SkyScanner API, OpenWeather API and ExchangeRatesAPI, our application needs to make use of the HTTP communication standard. The user's passwords are encrypted and stored in the Firebase Database.

4. System Features

This section illustrates the functional requirements for the product organised by system features, the major services provided by the product. The organisation of this section is done based on the logical hierarchy and the order of appearance within the application.

4.1 Create New Account/ Login/Forgot Password

4.1.1 Description and Priority

The following feature is concerning the login and registering of a user. The functional requirements give a detailed description of how the same works. This particular feature is of High priority since it forms the foundation of our app and is the only entry point into our app. The specific priority ratings are as follows:

- Risk: Potential risks of this section could be that the user does not receive the email after clicking on “Send Recovery EMail” after his/her email address is entered.
- Advantage: In comparison with other applications, Crafter is able to notify the user specifically if the password entered is invalid or the email address is invalid. Other applications might only notify the user that username/password is invalid and hence the user might not be certain of the error.

4.1.2 Stimulus/Response Sequences

When the user first opens “Crafter”, the first screen he/she sees is the login page. If the user is an existing user he/she can just enter the login credentials and start using the app. However, if the user is a new user they would need to register. They can do so by clicking on the “Create Account” button.

4.1.3 Functional Requirements

4.1.3.1 The user must be able to create a new account/login using their email address and password.

4.2.3.1.1 The system must request the user to enter email address and password upon login.

4.2.3.1.2 The system must request the user to enter email address, first name, last name, password and confirm password fields upon creating a new account

4.1.3.2 The system must validate that both fields have been filled up.

4.2.3.2.1 The system must validate the email address and password are correct.

4.1.3.3 If the login information is incorrect, the system must display an error message.

4.1.3.4 The user must be able to request a change of password should he/she forget his/her password by clicking on 'Forgot Password'.

4.1.3.4.1 The system then sends the user a reset password link to the user's registered e-mail ID.

4.1.3.4.2 The user can then click on the link and reset his/her password.

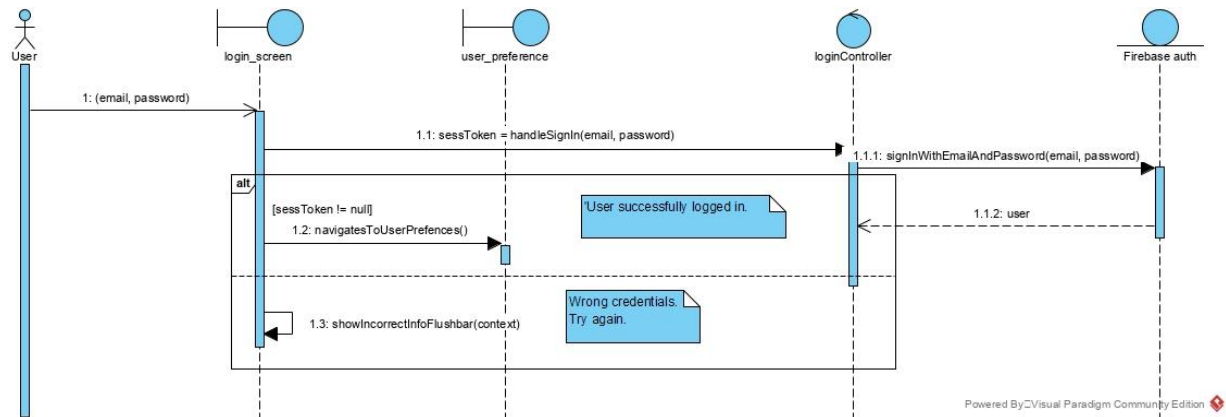


Fig 1. Login Sequence Diagram

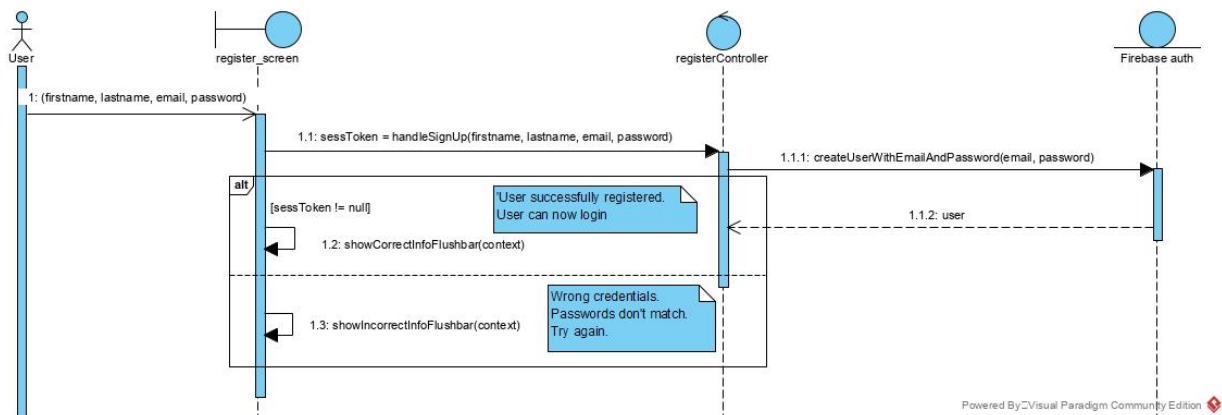
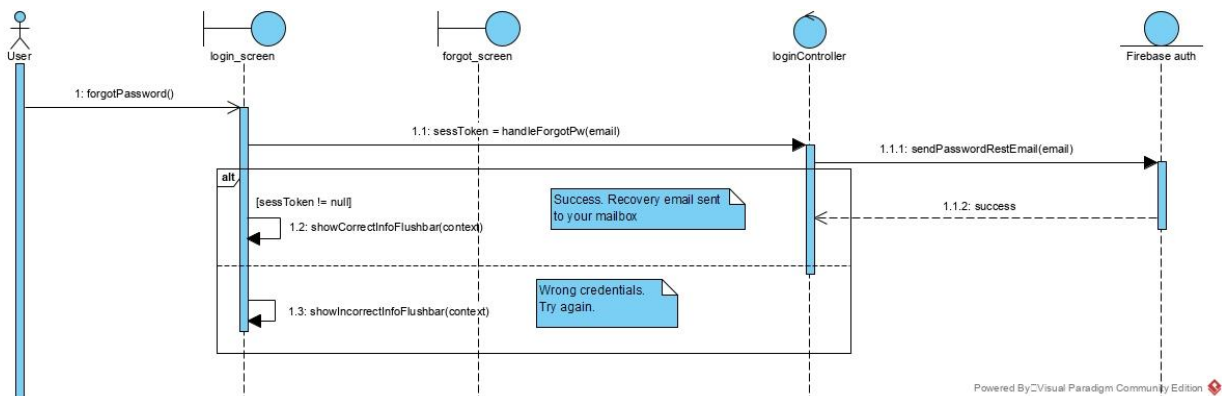


Fig 2. Register Sequence Diagram



4.2 Swipe Screen - Travel Destination Card

4.2.1 Description and Priority

This feature is one of the main features of the app. It is the core of our User Interface. It is the travel destination card which contains the city, country name and image. It allows the user to decide whether the user would be interested in visiting the destination based on the aesthetics of the destination. Further, the user is allowed to swipe left/right on each destination card. The priority of this feature is High.

The specific priority component ratings are as follows:

- Advantage: CraTriP allows the user to interact with the app by enabling the option to swipe when the user 'likes' the destination. This provides the user with a sense of satisfaction and provides the user with an assurance that the destination card has successfully been added to the 'like' section or the 'history'.
- Risk: Potential risk could be if the user made a mistake and chose to undo the action, This has not been implemented in CraTriP yet and should be considered in further improvements.

4.2.2 Stimulus/Response Sequences

Once the user has logged in, picked his/her favourite travel tag and clicked on proceed, the swipe screen page appears automatically.

4.2.3 Functional Requirements

4.2.3.1.1 The swipe screen page must contain two components: the destination card and the row of buttons.

4.2.3.1.1 Each travel destination card must satisfy the following:

4.2.3.1.1.1 A photograph of the city must occupy 75% of the screen.

4.2.3.1.1.2 The city name, followed by the country it's situated in.

4.2.3.1.1.3 Tags related to the city eg. vibrant, cultural. (3 tags per city)

4.2.3.1.2 The remaining 20% of the screen must contain a row with 3 buttons:

4.2.3.1.2.1 A cross button to dislike the travel destination.

4.2.3.1.2.2 A list button to view the summary of the travel destination.

4.2.3.1.2.3 A check button to like the travel destination.

4.2.3.1.3 Lastly, 5% of the screen must contain the navigation bar with 4 buttons:

4.2.3.1.3.1 A home button to navigate back to the swipe screen page

- 4.2.3.1.3.2 A 'history' button to display all the cards that the user has swiped right.
 - 4.2.3.1.3.3 A 'Recommendations' button to display 3 cards that CraTriP recommends users to visit based on the tags and history.
 - 4.2.3.1.3.4 A 'favourite' button to display all cards that the user has clicked the heart shaped icon of.
- 4.2.3.2 The user must be able to swipe left or select the 'green check' button on travel destinations cards that the user likes.
- 4.2.3.3 The user must be able to swipe right or select the 'red cross' on travel destinations cards that the user dislikes.
- 4.2.3.4 The user must be able to select the summary button at a particular travel destination card.
- 4.2.3.5 On selecting the summary button at a card, the system must display the following information in 3 separate rows:
 - 4.1.3.5.1 Latest exchange rate between SGD and the country's currency
 - 4.1.3.5.2 Flight information from Singapore to the city for tomorrow
 - 4.1.3.5.3 The current weather information at the city.
- 4.2.3.6 The user must be able to select the "view more" button to view more information about each of the following aspects: currency exchange, flight information and weather forecast.
- 4.2.3.7 The system must allow the users to favourite travel destinations using the favourites button.
- 4.2.3.8 The system must show only those travel destination cards on the swipe screen that have not been previously swiped by the user.

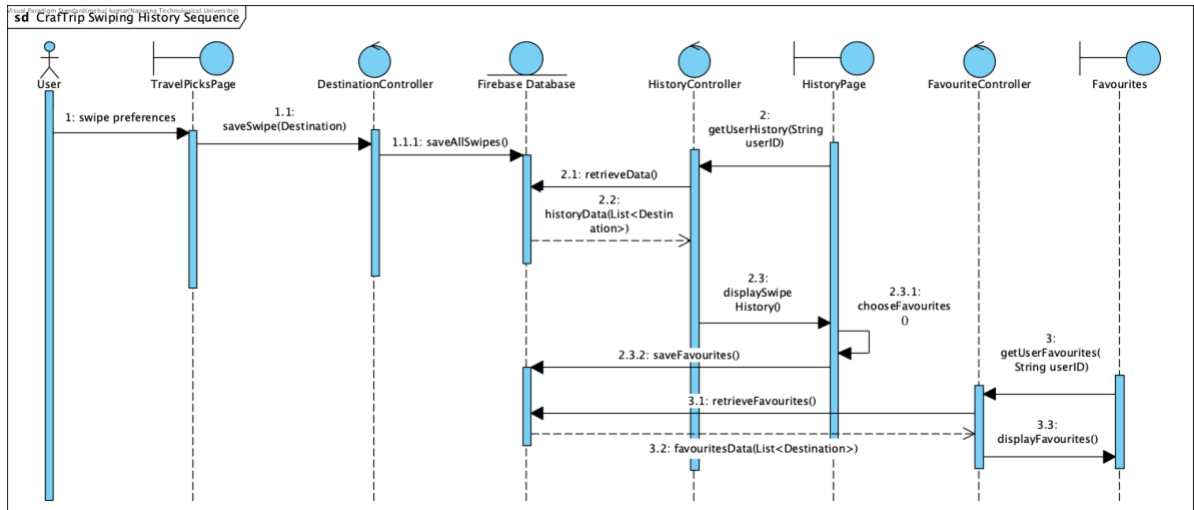


Fig 1. Swipe Screen Sequence Diagram

4.3 User Profile and Preferences

4.3.1 Description and Priority

This feature of 'CraffTrip' gives the user a personalised experience while using the app. The app allows the user to pick their favourite/desired travel tag, using which we customise the user's experience in the app. Further, the app stores all the destinations liked(right-swiped) by the user in the History tab, and all the travel destinations favourite-d by the user in the Favourites tab. Based on the history of the user, the app makes travel recommendations from the destinations not yet left swiped by the user. **Advantage:** This gives our app an edge over other applications in the market as it customises the user experience because of the user preference tags and travel destination recommendations. **Risk:** The computing costs of 'Recommendation' and customising the user experience is an added cost.

4.3.2 Stimulus/Response Sequences

After logging into the application, the tutorial is the first screen that appears for the user to pick their favourite/desired travel tag. To access History, Recommendations and Favourites, the user needs to click on the respective tabs in the bottom navigation bar. They can navigate to any of the tabs at any point of time while swiping travel destination cards.

4.3.3 Functional Requirements

4.3.3.1 Once logged in, the system must prompt the user to pick their favourite tag by clicking on tags. The system must allow the user to change this favourite tag in the future in the user account settings.

4.3.3.2 The system must then automatically navigate to the swipe screen page, displaying the travel destination cards.

- 4.3.3.3 The travel destination cards shown by the system must be cities based on the user's preference.
- 4.3.3.4 The system must allow the user to switch between tabs like Swipe Screen, History, Recommendations and Favourites using the bottom navigation bar.
- 4.3.3.5 Once done swiping, the system must display the following:
- 4.3.3.5.1 All the user's right swipes, under the History tab (destinations liked by the user)
 - 4.3.3.5.2 Recommendations of cities based on the user's right swipes' and tags of those cities, under the Recommendations tab.
 - 4.3.3.5.3 All the user's favourites swipes under the Favourites tab (destinations 'favourite-d' by the user)
- 4.3.3.6 The system must allow the user to 'favourite' destinations from the 'History' tab to add it to his/her favourite's list.
- 4.3.3.7 The system must allow the user to 'un-favourite' destinations from 'Favourites' list.
- 4.3.3.8 On clicking on any one of the travel destination cards within the 'History', 'Recommendation' or 'Favourite' tabs, the user is redirected to the summary page of the travel destination, thereby allowing the user to view details instantly.

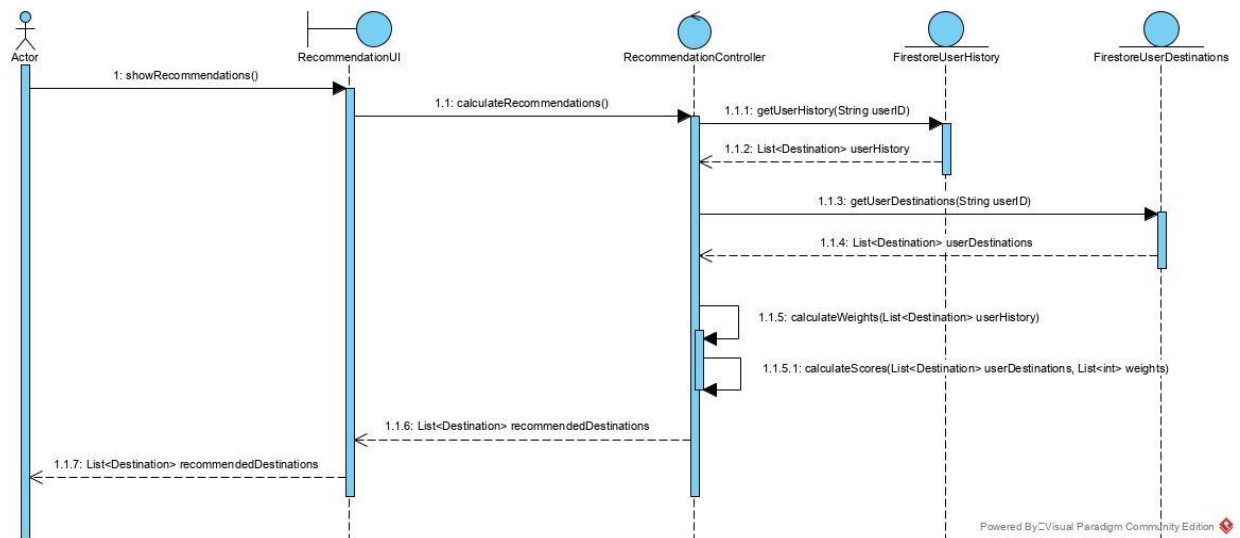


Fig 1. Recommendation Sequence Diagram

4.4 Retrieve Exchange Rate

4.4.1 Description and Priority

This feature allows the user to view details of the currency exchange rate between Singapore and the travel destination. It also allows the user to view the exchange rate trend across time from the start of 2020 using a line graph. Further it allows the user to view money changers located in Singapore displayed on a map.

Advantage: CrafTrip displays up to date information for exchange rates at the current time. It also provides a graph for users who are more concerned with how the exchange rate changes on whether to exchange money at this present time or wait awhile longer. Users can then predict if the exchange rate would drop or increase based on previous trends.

Risk: User's prediction of exchange rate might not follow according to the trend and hence the trend might cause the user to have a false impression on how the exchange rate might develop in the future.

4.4.2 Stimulus/Response Sequences

On selecting the 'View More' button of the Currency card of the summary page of a travel destination, the user will be able to view the currency exchange rates for that destination.

4.4.3 Functional Requirements

4.4.3.1 The system must display a line graph on the currency exchange rate from Singapore Dollar(SGD) to the travel destination's currency. The line graph must cover 50% of the screen.

4.4.3.2 The system must then display the current exchange rate from Singapore Dollar (SGD) to the travel destination's currency. (eg. SGD to USD)

4.4.3.3 The system must display a button to view money changer locations in Singapore. It will display a map with location markers of money changers.

4.4.3.4 The exchange rate prices must be displayed with two decimal places of precision. (eg. 1 SGD = 0.72 USD)

4.4.3.5 The system must be able to retrieve the exchange rates between SGD and other currencies using the ExchangeRatesAPI.

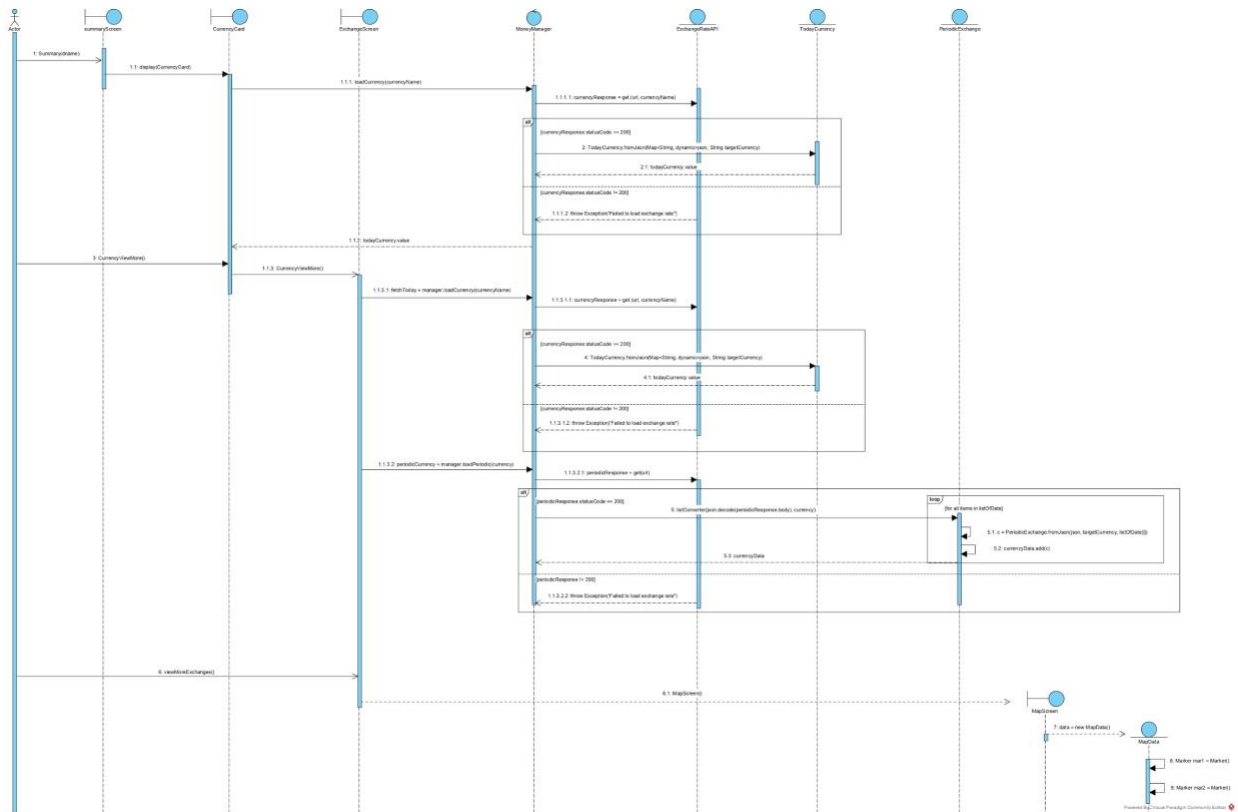


Fig 1. Exchange Rate Sequence Diagram

4.5 Retrieve Flight Information

4.5.1 Description and Priority

This feature allows the user to view details of round-trip flights to the travel destination through various airlines and at various times of the day. It also allows the user to sort the flights based on price. For an unlimited amount of queries, the cost of skyscanner api will be paid annually.

- Advantage: CrafterTrip helps users consolidate flight information including the flight prices for each cabin class and also provides the timing for each price. This helps users make an informed decision based on the flight and timing which are 2 important pieces of information when purchasing flight tickets.
- Penalty: Only round-trip flights are shown and this limits the user from buying 1-way tickets. This should be another area of improvement.

4.5.2 Stimulus/Response Sequences

On selecting the 'View More' button of the Flights card of the summary page of a travel destination, the user will be able to view the flight details for that destination.

4.5.3 Functional Requirements

- 4.5.3.1 The system must be able to retrieve flight information from Singapore to the travel destination one day after the current date in the summary page.
- 4.5.3.2 On selecting the “View More” button on the Flight card in the summary page, the user must be able to enter the date of departure and arrival. A calendar will be displayed and the user is required to select the dates. Then the following information for each airline must be displayed:
- 4.5.3.2.1 Flight prices
 - 4.5.3.2.2 Flight departure and arrival timings
 - 4.5.3.2.3 Airline
- 4.5.3.3 The user must be able to decide the criteria according to which he/she wants the flights sorted. Sorting optionality include:
- 4.5.3.3.1 Relevance
 - 4.5.3.3.2 Price - Low to High
 - 4.5.3.3.3 Price - High to Low
- 4.5.3.4 If flights with a certain set of specifications are not available, the system must display “No flights available.”
- 4.5.3.5 The flight prices must be in Singapore Dollar written as “SGD”. (e.g SGD 100).
- 4.5.3.6 The system must be able to retrieve the list of flight prices to a location using the Skyscanner API.

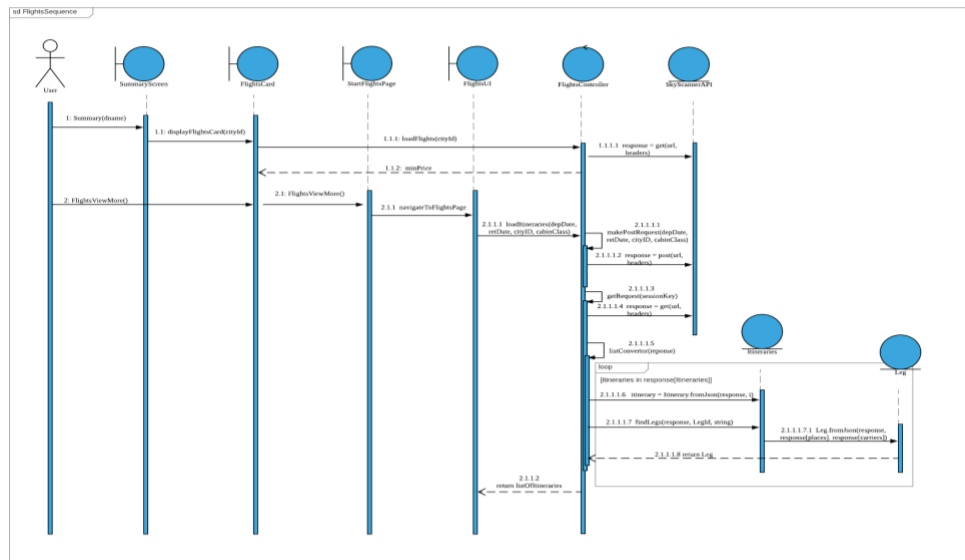


Fig 2. Flight Prices Sequence Diagram

4.6 Retrieve Weather Forecast

4.6.1 Description and Priority

This feature allows the user to view the current weather conditions at the travel destination. Further, it allows the user to view the forecast for the next 5 days, at intervals of 3 hours at the travel destination.

Advantage: CraTrip displays up to date information for weather forecast at the current time. It also provides the forecast for the next 5 days at intervals of 3 hours, making it a very useful feature for someone who is going to be reaching the destination within the next 5 days.

Risk: Weather forecast data acquired may not be a 100% accurate, thereby might mislead the users.

4.6.2 Stimulus/Response Sequences

On selecting the 'View More' button of the Weather card of the summary page of a travel destination, the user will be able to view the detailed weather information for that destination.

4.6.3 Functional Requirements

4.6.3.1 The user must be able to view the following in the Weather card in the summary page: the current air temperature, maximum and minimum air temperature, an icon describing the weather condition and humidity in the city.

4.6.3.2 On selecting the "View More" button on the weather card, the user must be able to retrieve the maximum and minimum air temperature at intervals of 3 hours for the next 5 days, at the travel destination.

4.6.3.3 The system must also display an icon that describes the weather condition at that particular time in the forecast. (eg. a cloud icon or a sunny icon)

4.6.3.4 The system must display the temperature in degrees centigrade in the range - 20°C to +50°C with no decimal place of precision. (eg. 25°C)

4.6.3.5 The system must be able to retrieve the weather forecast for the travel destination using the OpenWeatherMap API.

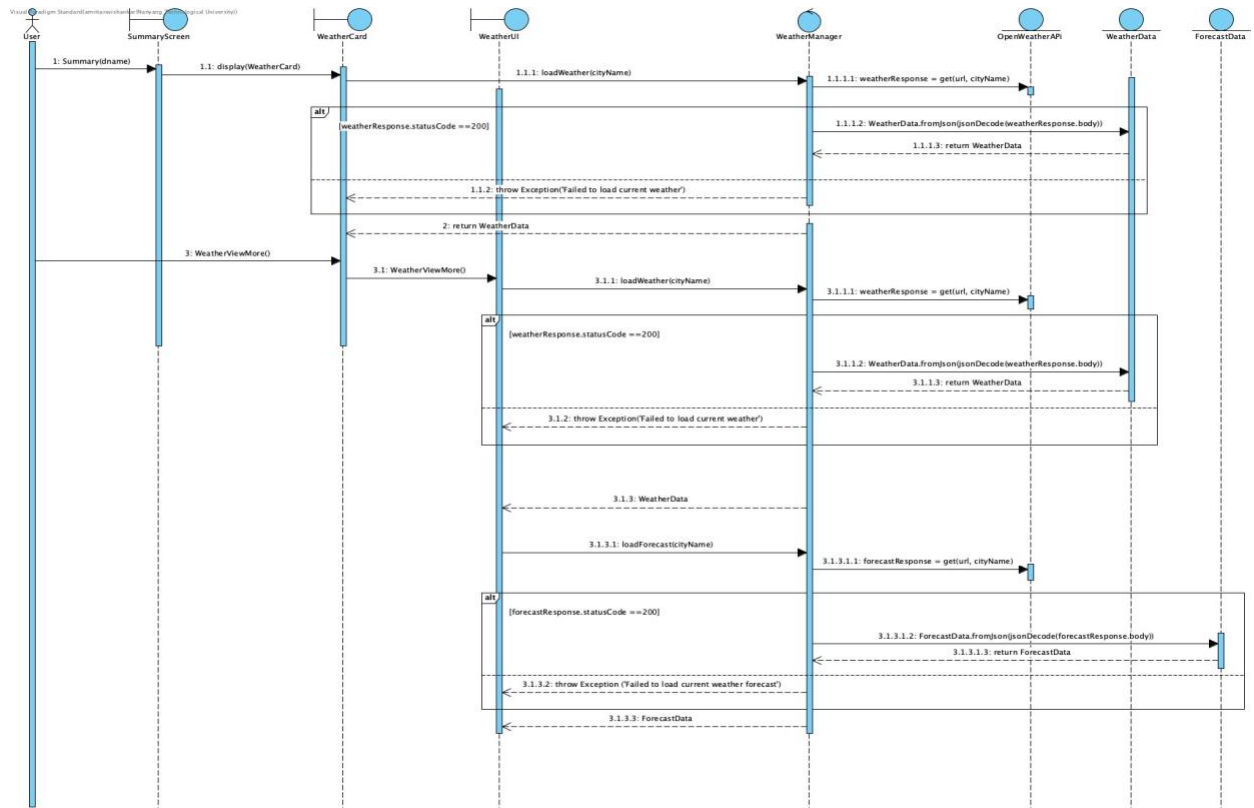


Fig 1. Weather Forecast Sequence Diagram

4.7 Summary Page

4.7.1 Description and Priority

This feature allows the user to view a summary of the above 3 features mentioned, namely exchange rate, flight information and weather forecast in a concise manner.

- Advantage: CrafterTrip allows users to view information in 1 screen allowing for ease of transition between screens if the user wants to factor in all 3 information to decide on the travel destination to visit.

4.7.2 Stimulus/Response Sequences

On selecting the 'Summary' button in the swipe screen page, the middle button, the user must be able to proceed to the summary page.

4.7.3 Functional Requirements

4.6.3.1 The user must be able to view the following in the summary page:

4.6.3.1.1 Latest exchange rate information between SGD and travel destination, with the corresponding currency name shown.

4.6.3.1.2 Flight prices for the next day between Singapore and the travel destination and the date of flight.

4.6.3.1.3 Latest weather information of the travel destination including current temperature, maximum and minimum temperature all in degrees celcius and the relative humidity in percentage.

4.6.3.2 Each of the above feature should have a button with 3 dots '...' corresponding to the 'view more' page for each feature

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- 5.1.1 It must take less than 5 seconds to display the next travel destination card after a user swipes right or left.
- 5.1.2 After the system reboots, the application must be restored/restarted within 5 minutes.
- 5.1.3 Summarized information about the travel destination must be displayed on an average of 1 minute when the user decides to view more information on the card.
- 5.1.4 It shall take less than 1 minute for the page of each feature to pop up when users select the “View More” button.
- 5.1.5 It must take less than 1 minute to load contents from the api to the main features.

5.2 Usability Requirements

The system must:

- 5.2.1 **Reduce short term memory load:** The implementation of dialog boxes helps users navigate seamlessly through the features and through components within the summarized information page. There is also a page for users to see their past swipes so users can keep a record of what destinations they have liked.
- 5.2.2 **Permit easy reversal of actions:** The user shall return to the summarized information page through the use of a ‘back’ button in the top left hand corner of each feature.
- 5.2.3 **Internal Locus of Control:** Swiping features of the card provide control to the user to “Like” or “Dislike” a travel destination card. Defined buttons and titles for each feature of the system also help make navigation and task activation clear for users. The user is also given the option to see and unsee password information in the event they want to check the password that was entered. This provides them with more control.
- 5.2.4 **Strive for consistency:** A consistent layout of the cards and its summarized information is achieved using similar font and color.

5.3 Security Requirements

5.3.1 The password must be hidden in the input field to protect users' data.

5.3.2 User's password must be encrypted by FirebaseAuth.

5.3.3 The user preferences and details of the user from the app such as the history of travel destinations, favourite destinations and recommended destinations must only be accessed by the user itself.

5.4 Software Quality Attributes

5.4.1 Supportability:

4.4.1.1 The mobile application must be supported by any mobile operating system such as iOS and android.

4.4.1.2 System must support the integration of the APIs and the database.

5.4.2 Maintainability:

4.4.2.1 The system must be updated daily to show relevant and current information of the travel destination searched through the retrieval of latest information from API calls.

5.4.3 Database

4.4.3 The Firestore database will store the travel destinations, favourite cities, history of users and user profile.

6. Other Requirements

Appendix A: Use Case Diagram and Description

6.1 Register for a new account

Use Case ID:	1.0		
Use Case Name:	Register for a new account		
Created By:	Sarah Yeoh	Last Updated By:	Sarah Yeoh
Date Created:	17/2/2020	Date Last Updated:	2/4/2020

Actor:	- User
Description:	Create a new account for a new user
Preconditions:	<ol style="list-style-type: none">1. User must not have a pre-existing account in the database2. Mobile phone must be connected to Wifi/Mobile data
Postconditions:	<ol style="list-style-type: none">1. The system must update its database to include an additional account

Priority:	High
Frequency of Use:	1 time
Flow of Events:	<ol style="list-style-type: none"> 1. User selects the 'Create Account' button 2. User enters a valid email address, first name, last name, password and confirm password fields. 3. User selects the 'Register' button 4. System validates that the email address entered is not currently found in the database 5. System checks if the password and confirm password fields are identical 6. System sends an email if account is successfully created 7. CraftTrip displays a pop-up box with the message 'Success! User successfully registered! Please login.'
Alternative Flows:	<p>AF-S5: Password and Confirm password fields are not identical</p> <ol style="list-style-type: none"> 1. System displays an error message "Incorrect Password" 2. Return to step 2
Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. Validate account availability
Special Requirements:	-
Assumptions:	No user verification email is needed

Notes and Issues:	-
-------------------	---

6.2 Validate account availability

Use Case ID:	2.0		
Use Case Name:	Validate account availability		
Created By:	Sarah Yeoh	Last Updated By:	Sarah Yeoh
Date Created:	17/2/2020	Date Last Updated:	2/4/2020

Actor:	- System
Description:	System checks if account is already existing in the database
Preconditions:	<ol style="list-style-type: none"> 1. User account must not already exist in the database 2. Mobile phone must be connected to WiFi/Mobile data
Postconditions:	<ol style="list-style-type: none"> 1. System will create a new account for the user in the database

	2. User will be able to login with the newly created account
Priority:	High
Frequency of Use:	0 - 3 times
Flow of Events:	<ol style="list-style-type: none"> 1. System validates the account availability 2. System creates a new account in the database
Alternative Flows:	<p>AF-S1: System detects that the account is already found in the database</p> <ol style="list-style-type: none"> 1. System displays an error message "Email address has an existing account." 2. User selects the "Create Account" button 3. User to enter a new email address, first name, last name, password and confirm password fields. 4. User selects the 'Register' button 5. System validates account availability
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	System contains database of all user's information
Notes and Issues:	-

6.3 Login

Use Case ID:	3.0		
Use Case Name:	Login		
Created By:	Sarah Yeoh	Last Updated By:	Sarah Yeoh
Date Created:	21/2/2020	Date Last Updated:	2/4/2020

Actor:	User
Description:	User login to existing account
Preconditions:	<ol style="list-style-type: none">1. User account must exist in the database2. Mobile phone must be connected to WiFi/Mobile data
Postconditions:	<ol style="list-style-type: none">1. User is able to choose favourite tag2. User is able to swipe cards3. User is able to view history, recommendations and favourites4. User is able to star and unstar favourites

Priority:	High
Frequency of Use:	0 - 20 times
Flow of Events:	<ol style="list-style-type: none"> 1. System displays email address and password fields 2. User enters email address and password 3. User selects the 'Login' button 4. System validates the account by checking the user's login credentials 5. User successfully login
Alternative Flows:	<p>AF-S3: System detects empty email or password fields</p> <ol style="list-style-type: none"> 1. System displays error message "Email address and password are required" 2. User enters email and password 3. User selects the 'login' button again 4. System validates the account by checking the user's login credentials <p>AF-S3: User enters wrong credentials</p> <ol style="list-style-type: none"> 1. System displays "Invalid email address/password" 2. User re-enters email and password 3. User selects the "login" button again 4. System validates the account by checking the user's login credentials
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-

Notes and Issues:	-
-------------------	---

6.4 Request for password change

Use Case ID:	4.0		
Use Case Name:	Request for password change		
Created By:	Sarah Yeoh	Last Updated By:	Sarah Yeoh
Date Created:	22/2/2020	Date Last Updated:	2/4/2020

Actor:	User
Description:	User forgets existing password and request to change it
Preconditions:	<ol style="list-style-type: none"> 1. User selects the "Forgot password?" button 2. Mobile phone must be connected to WiFi/Mobile data
Postconditions:	<ol style="list-style-type: none"> 1. System updates password for an existing account

Priority:	High
Frequency of Use:	0 - 20 times in a lifetime
Flow of Events:	<ol style="list-style-type: none"> 1. User selects the "Forgot Password?" button 2. System requests user to enter Email address. 3. User enters email address 4. System checks if email entered is an existing email address in the database 5. System sends a link to reset password 6. User enters new password and confirm password details 7. System to verify password and confirm password fields 8. System to verify new password entered is not the same as any of the old passwords 9. System to update new password into database
Alternative Flows:	<p>AF-S3: Email address entered is not an existing email address in the database</p> <ol style="list-style-type: none"> 1. System displays error message "Invalid email address" 2. Return to step 2 <p>AF-S6: New password and confirm password fields are not identical</p> <ol style="list-style-type: none"> 1. System displays error message "Password mismatch" 2. User re-enters password and confirm password fields. 3. System validates if both fields are identical 4. System updates database
Exceptions:	-

Includes:	-
Extend:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

6.5 Choose favourite tag

Use Case ID:	5.0		
Use Case Name:	Choose favourite tag		
Created By:	Sarah Yeoh	Last Updated By:	Sarah Yeoh
Date Created:	22/2/2020	Date Last Updated:	2/4/2020

Actor:	User
--------	------

Description:	User will choose a 'tag' that best describes the type of vacation they would prefer. This helps the system to narrow down to fewer cities that the user would prefer.
Preconditions:	<ol style="list-style-type: none"> 1. User must login 2. Mobile phone must be connected to WiFi/Mobile data
Postconditions:	<ol style="list-style-type: none"> 1. System must display a tutorial after every log in to show what the features of the app are for 2. System must display cards according to the tags chosen for subsequent visits to CrafTrip
Priority:	High
Frequency of Use:	0 - 3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. User login successfully 2. System displays 9 categories of tags to choose from 3. User selects a tag of their choice and selects the 'Proceed' button
Alternative Flows:	-
Exceptions:	-
Includes:	-
Extend:	-

Special Requirements:	-
Assumptions:	-
Notes and Issues:	- Only allowed to choose 1 tag

6.6 Swipe Cards

Use Case ID:	6.0		
Use Case Name:	Swipe cards		
Created By:	Sarah Yeoh	Last Updated By:	Sarah Yeoh
Date Created:	22/2/2020	Date Last Updated:	2/4/2020

Actor:	User
Description:	User to swipe cards right to indicate their interest to travel to that city, or left to indicate the lack of interest
Preconditions:	1. User to choose the favourite tag

	2. Mobile phone connected to Wifi/Mobile data
Postconditions:	<ol style="list-style-type: none"> 1. User is able to view exchange rate 2. User is able to view flight information 3. User is able to view weather information 4. User is able to select favourites
Priority:	High
Frequency of Use:	30 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. User selects a tag of their choice and selects the 'PROCEED' button 2. System displays the cards containing <ol style="list-style-type: none"> a. Name of city b. Picture that best describes the city c. Like button d. Dislike button e. "Summary" button 3. User swipes right or left 4. System will display another card under the same tag chosen by the user 5. Repeat step 2 until the user decides to stop by selecting another button in the navigation bar
Alternative Flows:	<p>AF-S2: User decides to select the "Summary" button</p> <ol style="list-style-type: none"> 1. System will display a summary of the <ol style="list-style-type: none"> a. Latest currency exchange rate b. Latest weather information c. Tomorrow's flight information 2. User selects the 'View more' button of <ol style="list-style-type: none"> a. Currency Exchange <ol style="list-style-type: none"> i. System must display a line graph of the currency exchange rate for the destination country from 1 Jan 2020 to the present date

- ii. System will display the current exchange rate between Singapore and the travel destination
- iii. User selects the "VIEW NEAREST EXCHANGES" button
- iv. System will display a map of all the money changers in Singapore
- v. User selects a marker on the map
- vi. System will display the address of the selected money changer
- b. Flight information
 - i. System will display a drop-down menu of:
 - 1. Date of Arrival
 - 2. Date of Departure
 - ii. User selects the arrival date and departure date from a pop-up calendar screen and selects 'ok' and then select "Check Prices" button
 - iii. System must display the prices of the flight sorted by Relevance with the following information:
 - 1. Flight price
 - 2. Date of departure
 - 3. Airline
 - iv. User is able to decide to sort the flights by:
 - 1. Price - Low to High
 - 2. Price - High to Low
 - 3. Flight time - Morning
 - 4. Flight time - Afternoon
 - 5. Flight time - Evening
 - 6. Flight time - Night
 - v. User must be able to select the cabin class of his choice
 - vi. System must display the updated list of flight based on the user's choice
- c. Weather information
 - i. System must display the current, minimum and maximum temperature and humidity of the country

Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. View exchange rate 2. View weather information 3. View flight prices
Extend:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

6.7 View exchange rate

Use Case ID:	7.0		
Use Case Name:	View exchange rate		
Created By:	Lim Jun Hong	Last Updated By:	Sarah Yeoh
Date Created:	23/02/2020	Date Last Updated:	2/4/2020

Actor:	<ul style="list-style-type: none"> - User - ExchangeRateAPI
Description:	Display currency exchange rate between SGD and travel destination
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged in 2. Mobile phone must be connected to the internet using WiFi/Mobile data 3. User must complete selecting their preferred tags
Postconditions:	<ol style="list-style-type: none"> 1. User must be able to view the specific currency exchange rate between SGD and the travel destination's currency 2. User must be able to see the nearest money changer
Priority:	High
Frequency of Use:	0 - 3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in successfully 2. System displays 'User Preferences' page 3. User selects a tag of their choice and selects the "PROCEED" button 4. System displays the cards containing <ol style="list-style-type: none"> a. Name of city b. Picture that best describes the city c. Like button d. Dislike button e. "Summary" button 5. User selects "Summary" button 6. System will display a summary of the: <ol style="list-style-type: none"> a. Latest currency exchange

	<ul style="list-style-type: none"> b. Tomorrow's flight information c. Latest weather information <ol style="list-style-type: none"> 7. User selects the "view more" button of the exchange rate as indicated by '...' 8. System must first display a line graph with the trend of the destination currency across time from 1 Jan 2020 to the current date. 9. System must display the current exchange rate between Singapore and the travel destination 10. User selects the "VIEW NEAREST EXCHANGES" button 11. System will display a map of all the money changers in Singapore 12. User selects a pin on the map 13. System will display the address of the selected money changer
Alternative Flows:	<p>AF-S4: User decides to look at their 'liked' travel destinations</p> <ol style="list-style-type: none"> 1. User selects the "History" button on the navigation bar 2. System displays travel destinations liked by user with: <ul style="list-style-type: none"> a. Latest currency exchange b. Latest weather information c. Favourite button 3. User selects a particular destination 4. System displays a summary of the travel destination <ul style="list-style-type: none"> a. Exchange rate b. Flight information c. Weather information 5. User selects the "view more" button of the exchange rate as indicated by '...' 6. System will display the current exchange rate between Singapore and the travel destination 7. User selects the "VIEW NEAREST EXCHANGES" button 8. System will display a map of all the money changers in Singapore 9. User selects a pin on the map

10. System will display the address of the selected money changer

AF-S4: User decides to look at their 'favourite' travel destination

1. User selects the "Favourites" button on the navigation bar
2. System displays travel destinations stored under "Favourites" with:
 - a. Currency Exchange
 - b. Weather information
3. User selects a travel destination
4. System display a summary of the travel destination with:
 - a. Currency Exchange
 - b. Flight information
 - c. Weather information
5. User selects the "view more" button of the exchange rate as indicated by '...'
6. System displays a line graph with the trend of the travel destination's currency exchange rate from 1 Jan 2020 to the present date.
7. System displays the latest exchange rate between Singapore and the travel destination
8. System displays a button to 'VIEW NEAREST EXCHANGES'
9. User selects the "VIEW NEAREST EXCHANGES" button
10. System must display a map of all the money changers in Singapore
11. User selects a pin on the map
12. System displays the address of the selected money changer

AF-S4: User decides to look at the 'Recommendation' page

1. User selects the "Recommendation" button on the navigation bar
2. System displays the 3 common tags that the API has retrieved after analysing the user's like and favourite history as well as displaying the 3 recommended travel destinations based on those tags, including their respective:

	<ul style="list-style-type: none"> a. Currency Exchange b. Weather information <ol style="list-style-type: none"> 3. User selects a particular travel destination 4. System displays a summary of the current exchange rate, current weather information and current flight information of the city <ul style="list-style-type: none"> a. Currency Exchange b. Flight information c. Weather information 5. User selects the “view more” button of the exchange rate as indicated by ‘...’ 6. System displays a line graph with the trend of the travel destination’s currency exchange rate from 1 Jan 2020 to the present date. 7. System displays the current exchange rate between Singapore and the travel destination 8. System must display a button to ‘VIEW NEAREST EXCHANGES’ 9. User selects the “VIEW NEAREST EXCHANGES” button 10. System must display a map of all the money changers in Singapore 11. User selects a pin on the map 12. System displays the address of the selected money changer
Exceptions:	-
Includes:	-
Extend:	-
Special Requirements:	-
Assumptions:	<ul style="list-style-type: none"> - For both alternative flows, user has at least one liked travel destination and favourite travel destination

	<ul style="list-style-type: none"> - Currency exchange rate are retrieved from ExchangeRate API
Notes and Issues:	-

6.8 View flight information

Use Case ID:	8.0		
Use Case Name:	View flight information		
Created By:	Lim Jun Hong	Last Updated By:	Sarah Yeoh
Date Created:	24/2/2020	Date Last Updated:	25/2/2020

Actor:	<ul style="list-style-type: none"> - User - Skyscanner API
Description:	Displays flight prices from Singapore and travel destination on the current date
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged in 2. Mobile phone must be connected to the internet using WiFi/Mobile data

	3. User has completed setting up their account
Postconditions:	<ol style="list-style-type: none"> 1. User will be able to view the flight prices from Singapore to the travel destination on the current date 2. User will be able to view flight prices between Singapore to the travel destination depending on their departure and arrival date
Priority:	High
Frequency of Use:	0 - 3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in successfully 2. System displays User Preferences page 3. User selects a tag of their choice and selects the "Proceed" button 4. System displays the cards containing <ol style="list-style-type: none"> a. Name of city b. Picture that best describes the city c. Like button d. Dislike button e. "Summary" button 5. User selects the "Summary" button 6. System will display a summary of the: <ol style="list-style-type: none"> a. Latest currency exchange b. Tomorrow's flight information c. Latest weather information 7. User selects the "view more" button of flight information as indicated by the '...' 8. System will display a drop-down menu of <ol style="list-style-type: none"> a. Date of Arrival b. Date of Departure 9. User selects the arrival date and departure date from a pop-up calendar screen and selects 'ok' and then select "Check Prices" button 10. System will display the prices of the flight sorted by Relevance

	<p>11. User can decide to sort the flights by:</p> <ul style="list-style-type: none"> a. Price - Low to High b. Price - High to Low c. Flight time - Morning d. Flight time - Afternoon e. Flight time - Evening f. Flight time - Night <p>12. User can decide to filter the flights by:</p> <ul style="list-style-type: none"> a. Economy b. Business c. First-Class <p>13. System will display the updated list of flight based on the user's choice</p>
Alternative Flows:	<p>AF-S5: User decides to see their liked travel destinations history</p> <ul style="list-style-type: none"> 1. User selects the "History" button on the navigation bar 2. System will display a list of the travel destinations that the user has previously liked or favorited along with their respective: <ul style="list-style-type: none"> a. Currency Exchange b. Weather information 3. User selects a particular travel destination 4. System will display a summary of the current: <ul style="list-style-type: none"> a. Currency Exchange b. Flight information c. Weather information 5. User selects the "view more" button of flight information 6. System will display a drop-down menu of: <ul style="list-style-type: none"> a. Date of Arrival b. Date of Departure 7. User selects the arrival date and departure date and selects the "CALCULATE PRICES" button 8. System will display the prices of the flight sorted by Relevance 9. User can decide to sort the flights in a drop-down menu by: <ul style="list-style-type: none"> a. Price - Low to High b. Price - High to Low

- c. Flight time - Morning
 - d. Flight time - Afternoon
 - e. Flight time - Evening
 - f. Flight time - Night
10. User can decide to filter the flights by selecting:
- a. Economy
 - b. Business
 - c. First-Class
11. System will display the updated list of flight based on the user's choice

AF-S5: User decides to view flight information of travel destination from a list of travel destinations that they have 'favourite'

- 1. User selects the "Favourites" button on the navigation bar
- 2. System will display user's "Favourite" travel destination
- 3. User selects a particular travel destination
- 4. System will display a summary of :
 - a. Latest currency exchange
 - b. Tomorrow's flight information
 - c. Latest weather information
- 5. User selects "view more" button of flight information
- 6. User selects the arrival date and departure date from a pop-up calendar screen and selects 'ok' and then select "Check Prices" button
- 7. System will display the prices of the flight sorted by Relevance
- 8. User can decide to sort the flights in a drop-down menu by:
 - a. Price - Low to High
 - b. Price - High to Low
 - c. Flight time - Morning
 - d. Flight time - Afternoon
 - e. Flight time - Evening
 - f. Flight time - Night
- 9. User can decide to filter the flights by selecting:
 - a. Economy
 - b. Business
 - c. First-Class

10. System will display the updated list of flight based on the user's choice

AF-S5: User decides to view flight information of travel destinations recommended by the app

1. User selects the "Recommendations" button on the navigation bar
2. System displays the 3 common tags that the API has retrieved after analysing the user's like and favourite history as well as displaying the 3 recommended travel destinations based on those tags, including their respective:
 - a. Currency Exchange
 - b. Weather information
3. User selects a particular travel destination
4. System displays a summary of the:
 - a. Latest currency exchange
 - b. Tomorrow's flight information
 - c. Latest weather information
5. User selects the "view more" button of flight information as indicated by '...'
6. System will display a drop-down menu of:
 - a. Date of Arrival
 - b. Date of Departure
7. User selects the arrival date and departure date and selects the "CALCULATE PRICES" button
8. System will display the prices of the flight sorted by Relevance
9. User can decide to sort the flights in a drop-down menu by:
 - a. Price - Low to High
 - b. Price - High to Low
 - c. Flight time - Morning
 - d. Flight time - Afternoon
 - e. Flight time - Evening
 - f. Flight time - Night
10. User can decide to filter the flights by selecting:
 - a. Economy
 - b. Business
 - c. First-Class
11. System will display the updated list of flight based on the user's choice

Exceptions:	-
Includes:	-
Extend:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	<ul style="list-style-type: none"> - User is only allowed to view flight information from the current date to 1 week in advance - Craftrip is only able to support up to 10 queries per day due to the limitation of “Free” plan from Skyscanner’s API host, RapidAPI - Skyscanner API is out of commission since 1st April 2020

6.9 View weather information

Use Case ID:	9.0		
Use Case Name:	View weather information		
Created By:	Lim Jun Hong	Last Updated By:	Sarah Yeoh

Date Created:	24/2/2020	Date Last Updated:	2/4/2020
---------------	-----------	--------------------	----------

Actor:	<ul style="list-style-type: none"> - User - OpenWeather API
Description:	Displays weather information of the travel destination
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged in 2. Mobile phone must be connected to the internet using WiFi/Mobile data 3. User has completed setting up their account
Postconditions:	<ol style="list-style-type: none"> 1. User must be able to see the weather of the travel destination
Priority:	High
Frequency of Use:	0 - 3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in successfully 2. System displays User Preferences page 3. User selects a tag and selects the "PROCEED" button 4. System displays the cards containing <ol style="list-style-type: none"> a. Name of city b. Picture that best describes the city c. Like button d. Dislike button e. "Summary" button

	<ol style="list-style-type: none"> 5. User selects the “Summary” button 6. System displays a summary of the current <ol style="list-style-type: none"> a. Currency Exchange b. Flight information c. Weather information 7. User selects the “view more” button of weather information as indicated in ‘..’ 8. System displays the weather information of the travel destination for the next 5 days
Alternative Flows:	<p>AF-S5: User decides to view weather information in favourites</p> <ol style="list-style-type: none"> 1. User selects the “Favourites” button on the navigation bar 2. System will display a list of the user’s “Favourites” travel destination as well as their respective: <ol style="list-style-type: none"> a. Currency Exchange b. Weather information 3. User selects a particular travel destination 4. System will display a summary of the: <ol style="list-style-type: none"> a. Latest currency exchange b. Tomorrow’s flight information c. Latest weather information 5. User selects the “view more” button of weather information as indicated in the ‘...’ 6. System displays the current, minimum and maximum temperature and relative humidity of the travel destination of 3 hours interval for the next 5 days <p>AF-S5: User decides to view weather information in history</p> <ol style="list-style-type: none"> 1. User selects the “History” button on the navigation bar 2. System will display the user’s history of the travel destinations that the user has liked or favorited previously as well as their respective <ol style="list-style-type: none"> a. Currency Exchange b. Weather information 3. User selects a particular travel destination 4. System will display a summary of the:

	<ol style="list-style-type: none"> a. Latest currency exchange b. Tomorrow's flight information c. Latest weather information <ol style="list-style-type: none"> 5. User selects the "view more" button of weather information as indicated in '...' 6. System displays the current, minimum and maximum temperature and relative humidity of the travel destination of 3 hours interval for the next 5 days <p>AF-S5: User decides to view weather information of travel destinations recommended by the app</p> <ol style="list-style-type: none"> 1. User selects the "Recommendation" button on the navigation bar 2. System displays the 3 common tags that the API has retrieved after analysing the user's like and favourite history as well as displaying the 3 recommended travel destinations based on those tags, including their respective: <ol style="list-style-type: none"> a. Currency Exchange b. Weather information 3. User selects a particular travel destination 4. System displays a summary of the: <ol style="list-style-type: none"> a. Latest currency exchange b. Tomorrow's flight information c. Latest weather information 5. User selects the "view more" button of weather information as indicated in '...' 6. System will display the current, minimum and maximum temperature and relative humidity of the travel destination of 3 hours interval for the next 5 days
Exceptions:	-
Includes:	-
Extend:	-

Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

6.10 Select favourites

Use Case ID:	10.0		
Use Case Name:	Select favourites		
Created By:	Lim Jun Hong	Last Updated By:	Sarah Yeoh
Date Created:	24/2/2020	Date Last Updated:	25/2/2020

Actor:	- User
Description:	User can 'favourite' the travel destinations that they show great interest in
Preconditions:	1. User must be logged in

	<ol style="list-style-type: none"> 2. Mobile phone must be connected to the internet using WiFi/Mobile data 3. User has completed selecting their tag preferences
Postconditions:	<ol style="list-style-type: none"> 1. User is able to save cards into their 'favourite' list
Priority:	High
Frequency of Use:	10 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in successfully 2. System displays 'User Preferences' page 3. User selects a tag of their choice and selects the "PROCEED" button 4. System displays the cards containing <ol style="list-style-type: none"> a. Name of city b. Picture that best describes the city c. Like button d. Dislike button e. Summary button 5. User selects the "History" button on the navigation bar 6. System displays the history of the cards liked by the user 7. User selects the "Favourite" button as indicated by a heart on the card 8. System will store the card in user's "Favourites" 9. System will display the collection of cards stored under "Favourites" with a pink heart on the top right hand corner of the card
Alternative Flows:	<p>AF-S5: User has not favourite a card</p> <ol style="list-style-type: none"> 1. If user selects the "favourites" button in the navigation bar before favoriting any card, the system will display an error message "No favourites"

Exceptions:	-
Includes:	-
Extend:	-
Special Requirements:	-
Assumptions:	<ul style="list-style-type: none"> - User has completed setting up their account with at least one tag chosen - User has at least one card in their “History” - User intends to “Favourite” a card
Notes and Issues:	-

6.11 View recommendations

Use Case ID:	11.0		
Use Case Name:	View recommendations		
Created By:	Lim Jun Hong	Last Updated By:	
Date Created:	24/2/2020	Date Last Updated:	

Actor:	- User
Description:	Allows user to view recommended cities based on the analysis of the tags of the cards that the user has liked
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged in 2. Mobile phone must be connected to the internet using WiFi/Mobile data 3. User has completed selecting their tag preferences
Postconditions:	<ol style="list-style-type: none"> 1. User must be able to view the cities recommended by the system 2. User must be able to store travel destinations in "Favourites"
Priority:	High
Frequency of Use:	0 - 3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in successfully 2. System displays User Preferences page 3. User selects a tag of their choice and selects the "PROCEED" button 4. System displays the cards containing <ol style="list-style-type: none"> a. Name of city b. Picture that best describes the city c. Like button d. Dislike button e. "Summary" button 5. User swipes right or left 6. System will display another card under the same tag chosen by the user

	<ol style="list-style-type: none"> 7. Repeat step 3 until the user decides to stop by selecting one of the buttons in the navigation bar 8. User selects the “recommendations” button on the navigation bar 9. System displays 3 recommended travel destinations based on 3 common tags in the user’s history
Alternative Flows:	-
Exceptions:	-
Includes:	-
Extend:	- Select favourites
Special Requirements:	-
Assumptions:	<ul style="list-style-type: none"> - User has completed setting up their account - User must have swiped cards
Notes and Issues:	-

6.12 View history

Use Case ID:	12.0
--------------	------

Use Case Name:	View history		
Created By:	Lim Jun Hong	Last Updated By:	Lim Jun Hong
Date Created:	24/02/2020	Date Last Updated:	2/4/2020

Actor:	- User
Description:	User can see the history of the travel cards they swiped right or favorited
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged in 2. Mobile phone must be connected to the internet using WiFi/Mobile data 3. User has completed selecting their tag preferences 4. Travel cards stored as favourites must appear in the history
Postconditions:	<ol style="list-style-type: none"> 1. User is able to view their history and differentiate those that are stored in "Favourites"
Priority:	High
Frequency of Use:	0 - 3 times a day

Flow of Events:	<ol style="list-style-type: none"> 1. User logs in successfully 2. System displays User Preferences page 3. User selects the tags of their choice and selects the "PROCEED" button 4. System displays the cards containing <ol style="list-style-type: none"> a. Name of city b. Picture that best describes the city c. Like button d. Dislike button e. "Summary" button 5. User selects the "History" button 6. System displays the list of cards liked/favorited by the user 7. Cards that have been "Favorited" are marked by a pink heart on the top right of the card
Alternative Flows:	-
Exceptions:	-
Includes:	-
Extend:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	<ul style="list-style-type: none"> - Cards that are favorited by the user initially, without swiping, are also stored in the 'History' list

6.13 View favourites

Use Case ID:	13.0		
Use Case Name:	View favourites		
Created By:	Lim Jun Hong	Last Updated By:	Lim Jun Hong
Date Created:	24/02/2020	Date Last Updated:	2/4/2020

Actor:	- User
Description:	User can view the cards they have stored as favourites
Preconditions:	<ol style="list-style-type: none">1. User must be logged in2. Mobile phone must be connected to the internet using WiFi/Mobile data3. User has completed selecting their tag preferences
Postconditions:	<ol style="list-style-type: none">1. User must be able to view the travel destinations they have marked as favourite
Priority:	High

Frequency of Use:	0 - 3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in successfully 2. System displays User Preferences page 3. User selects a tag of their choice and selects the "PROCEED" button 4. System displays the cards containing <ol style="list-style-type: none"> a. Name of city b. Picture that best describes the city c. Like button d. Dislike button e. "Summary" button 5. User selects the "Favourites" button on the navigation bar 6. System displays a list of the user's "Favourites" travel destination as well as their respective: <ol style="list-style-type: none"> a. Currency Exchange b. Weather information
Alternative Flows:	-
Exceptions:	-
Includes:	-
Extend:	-
Special Requirements:	-
Assumptions:	

Notes and Issues:	-
-------------------	---

6.14 Remove favourites

Use Case ID:	14.0		
Use Case Name:	Remove favourites		
Created By:	Lim Jun Hong	Last Updated By:	Lim Jun Hong
Date Created:	24/02/2020	Date Last Updated:	2/4/2020

Actor:	User
Description:	User removes a travel destination from “favourites”
Preconditions:	<ol style="list-style-type: none"> 1. User must be logged in 2. Mobile phone must be connected to the internet using WiFi/Mobile data 3. User has completed selecting their tag preferences 4. User has a travel destination stored in “Favourites”

Postconditions:	1. Travel destination that was stored in “Favourites” is removed
Priority:	High
Frequency of Use:	0 - 3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in successfully 2. System displays User Preferences page 3. User selects a tag of their choice and selects the “PROCEED” button 4. System displays the cards containing <ol style="list-style-type: none"> a. Name of city b. Picture that best describes the city c. Like button d. Dislike button e. “Summary” button 5. User selects the “Favourites” button on the navigation bar 6. System displays a list of the user’s “Favourites” travel destination as well as their respective: <ol style="list-style-type: none"> a. Currency Exchange b. Weather information 7. User selects the heart on the top right of a card 8. System will remove the card from “Favourites” and display the updated collection of cards stored under “Favourites” by the user
Alternative Flows:	-
Exceptions:	-
Includes:	-

Extend:	-
Special Requirements:	-
Assumptions:	- If the user removed a card that is both liked and favorited, the card will only be removed from the 'favourites' list and not from the 'like' history
Notes and Issues:	-

Appendix B: Analysis Models

1. Architecture Diagram

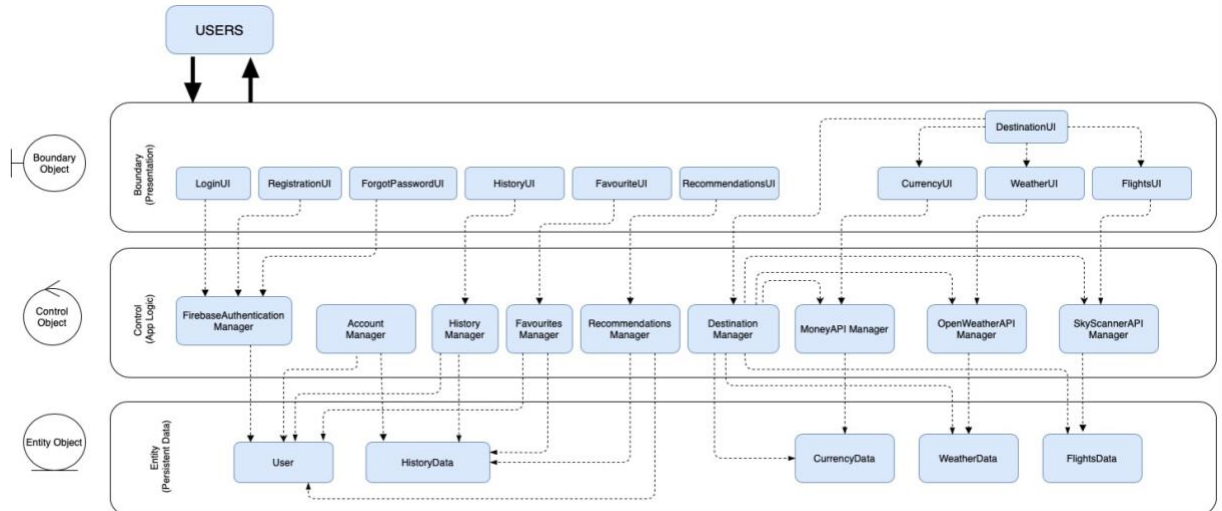
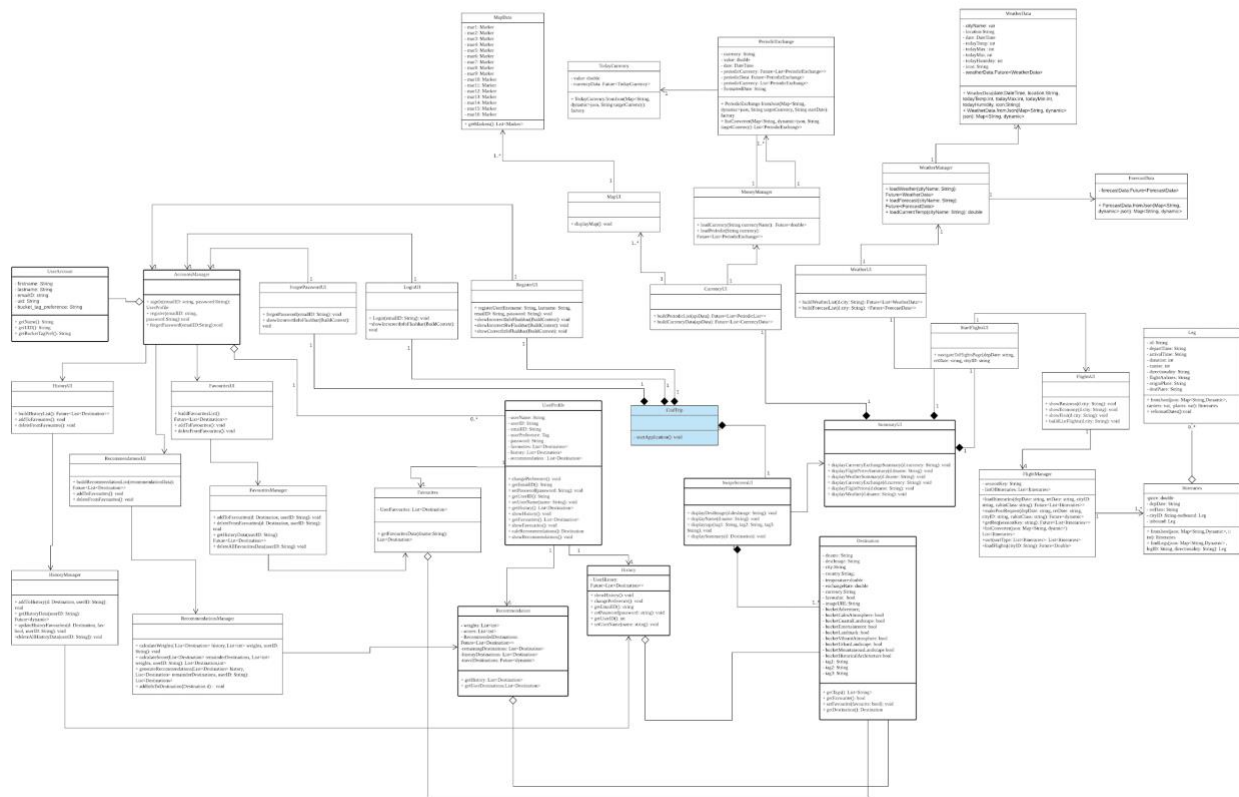


Fig 1. CrafTrip Architecture Diagram

2. Class Diagram



3. Classes and their descriptions

CLASSES	FUNCTIONALITIES
Account LoginScreen RegisterScreen ForgotPasswordScreen LoginManager RegisterManager ForgotPasswordManager UserAccountModel	This section deals with all the classes related to handling the account of a user such as Login, Register and Forgot Password.
Travel Swipes TravelSwipesScreen	This section contains all the classes that are needed to display the travel picks' swipe screen.

TravelSwipesCard TravelSwipesManager DestinationModel	
Weather WeatherScreen WeatherCard WeatherManager WeatherModel	These classes are all the classes required to display the current weather and the forecasted weather information
Flight Prices FlightsScreen FlightsManager FlightsModel	These classes are all the classes required to display the flight prices to and from the destination on a given departure and arrival date.
Currency Exchange CurrencyExchangeScreen ExchangesMapScreen CurrencyExchangeManager ExchangesMapManager CurrencyExchangeModel	These classes are all the classes required to display the current exchange rate and the exchange rate trend since the start of 2020.
History HistoryScreen DestinationCard HistoryManager HistoryModel	All the classes related to displaying the history of the user, that is all the users' right swipes, allowing them to access it in the future.
Favourites FavouritesScreen DestinationCard FavouritesManager FavouritesModel	These classes are all the classes required to display the user's favourite destinations, which he/she has favourite-d from the history tab.
Recommendations RecommendationScreen DestinationCard RecommendationManager DestinationModel	All the classes that need to be accessed in order to show the recommendations to the user.

Appendix C: Future Works

1. Include more travel destinations within the application
2. Include hotel bookings as a feature for every travel destination
3. Improve the performance of our application with respect to time required to load the flight screen in particular.
4. Add an undo button, in case the user accidentally left or right swipes on a destination.