

CS 335 GROUP PROJECT

GROUP 55: TOURISM

BY: MAC KENZIE DRAKE PASCUAL (), NIKITA RADZEVICIUS (22368276), AND RANDON REDDY (22252352)

INTRODUCTION

Being tasked to create a software system with a theme in tourism, we decided to create a software system in the form of a tourism app and is based in cities with big tourism industries. The users of our app would want features to enable easy access tourism across cities and allow businesses to attract potential customers.

Upon discussion, we decided to focus on three main different actors (Tourist, Accommodation Provider [AP], and App Developer [AD]) and how they would interact with our app based of their respective purposes. Our starting basis is to create User Stories which are used to develop sets of UML diagrams for our system in our next step. We then created a UI Mockup of what the app would look like which concludes with System tests being performed. (R)

Each person's contribution to this project will be denoted by:

- Report, System Tests, Use Case Diagram – Randon Reddy (R)
- Wireframe, Architecture Diagram, Video Presentation - Nikita Radzevicius (N)
- PowerPoint Presentation, Class Diagram - Mac Kenzie Drake Pascual (M)

User Stories and Sequence diagrams were spilt amongst us. Description indicates who has contributed what.

PART I – USER STORIES

- As an adventurous tourist, I would like a feature that lets me get pictures of unknown objects, landmarks, or destinations and instantly provides me with details about them as I scan them, such as their names, historical significance, and nearby attractions. With that, I can explore and learn more about the places I go. (M)
- As a global tourist, I want a world clock time converter feature in my travel app. This will enable me to quickly inspect the differing time zones in various areas, confirming that I remain prepared and on time when traveling, wherever I may be in the world. (M)

- iii. As a budget tourist, I want an app with a converting/savings system. I would like to be able to effortlessly manage my spending across currencies using real-time exchange rates. I can monitor my spending and receive notifications about potential savings on flights and accommodation. I can also see what the exchange rate is between different currencies. (M)
- iv. As a tourist coming to Dublin for the first time, I would like an app with some sort of accommodation feature, where I can see a list of the local hotels, inns and B&Bs along with the prices so I can compare them and get the best value alongside the closest hotel to the activities I want to go to. (N)
- v. As a tourist I want an app that has features translating text and images on the go in multiple languages, as well as having an offline mode as I don't speak English very well and I may not always have Wi-Fi or data, having this would be helpful for my holidays so I can better navigate the city and shops. (N)
- vi. As a tourist going abroad by myself can be lonely, having an app that features a social media or a feature that allows for meetup of other tourists or locals in Dublin and communication with messages and photos, so that I can find others and make friends on holidays. (N)
- vii. As a tourist, getting around in a big city can be tiring when walking, it would be helpful to have an app where transport such as taxis, bus and trains would be available to view along with their prices and timetables, so that I can make it to my destination on time and in the most convenient fashion to me. (N)
- viii. As an accommodation provider, I would like a platform to be able to post listing of available rooms to occupy in the hotel, manage bookings, and offer specials to attract tourist. (R)
- ix. As an accommodation provider, I would like a platform to communicate with potential customers to answer any questions or inquires. (R)
- x. As an app the developer, I would like to access to app to manage suspicious accommodation listings, moderate tourist messenger groups and to provide safety and security features to users. (R)

PART 1 CONCLUSION:

As a group we discussed and exchanged ideas about different User Stories that could apply to our software system and what type of consumer would be interested in a tourism app and what features they would like or enjoy. The User Stories created above leads us to Part II in which we construct sets of UML diagram for our system. (R)

PART II – UML DIAGRAMS

USE CASE DIAGRAM

In general, Use Case Diagrams are used to represent end goals and functionality of a system and to show how different users would interact with the system. In the diagram below we have 3 primary actors: Tourist, Accommodation providers, and the app developers. In the diagram we use generalization to group Tourist and APs together as they both be able to access the same various actions and activates while however APs have access to certain functionality based on their needs as stated in the User Stories. (R)

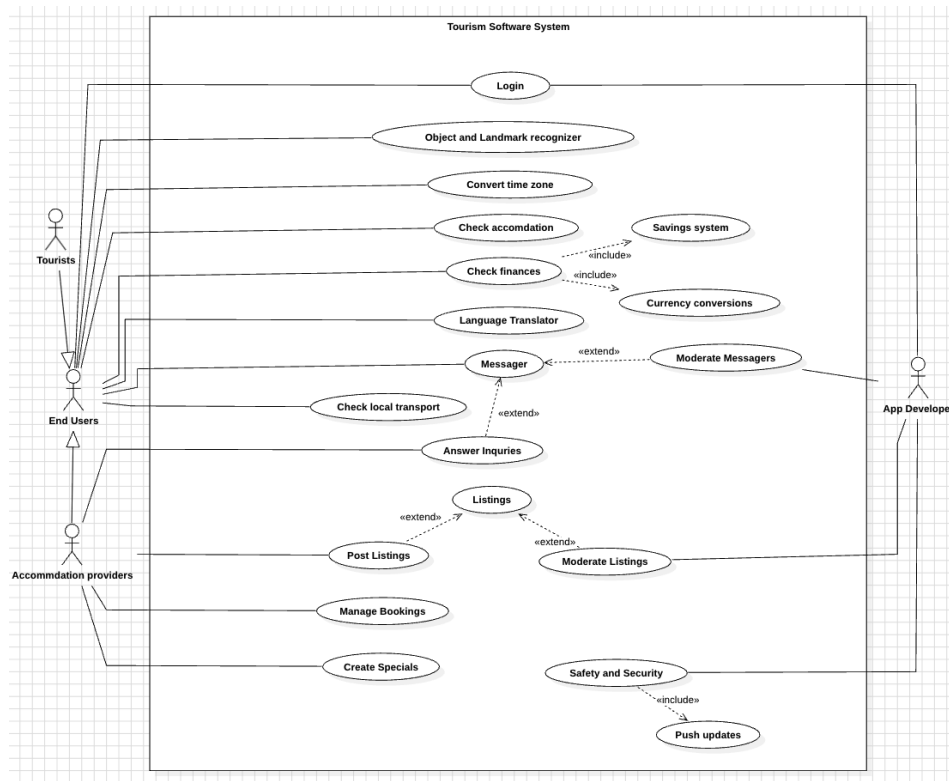


Figure 2.1: Use Case Diagram for TravelBuddy

SEQUENCE DIAGRAMS

Tourist Photo Scanner: As the user opens the application, they click on the “Photo Scanner Tab” as an *AdventurousTourist* object being made. Once selected, the object asks the app to get a scan on unknown artefacts or holiday places. The app would then give the request to the *TouristPhotoScanner* class to take a photo shot verify the image. Once it starts scanning, it processes the information and puts it placed in the servers so the app can display them in return for the *AdventurousTourist* object to finally see the results to the user. If the object takes more than one photo, the class will have looped its functions.

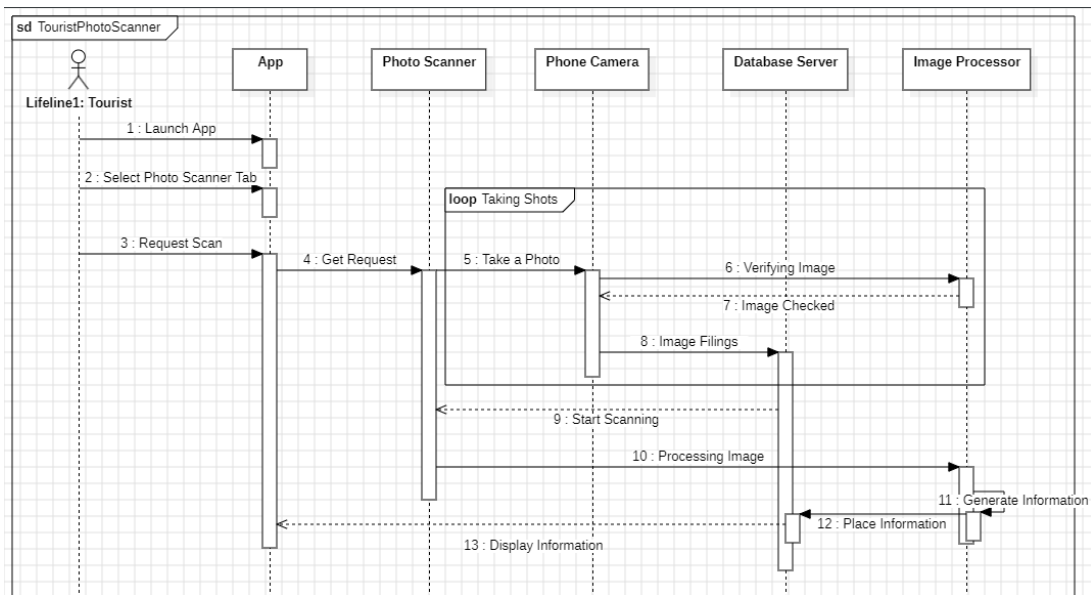


Figure 2.2: Sequence Diagram for User Story 1

World Clock Converter: When the app has been opened, the user selects the “World Clock Time Zones Tab” to the *GlobalTourist* object. The object then makes a request to convert their current time to their chosen regional time zone of their preferences. The *TimeConversion* class receives the message and does the conversions from the APIs. The class would do a loop if the *GlobalTourist* object asks for more regional times. Once done, the database gets the information to be shown for the app to the object. It would lastly prompt the object to save the converted time zones on their tab menu. The class would let the app show recent updates of the time slots.

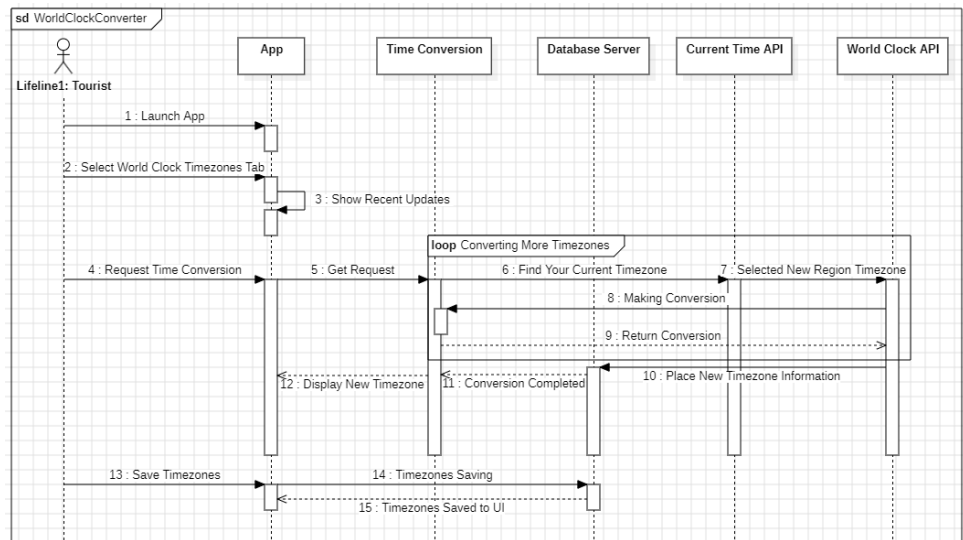


Table 2.3: Sequence Diagram for User Story 2

SEQUENCE DIAGRAMS

Money Converter Saver: The Money Converter Saver tab is used by the *BudgetTourist* object to request money conversions or deposits. The app sends a message to the *MoneyExchanger* class, which verifies the card's validity and then prompts the object to choose the currency. The *CurrencyConverter* class calculates the latest exchange rate. The newly exchanged cash is sent to the server, and the app displays the results. If the user chooses the latter option, the Bank API saves the amount to the database server for display in the app. The app posts updates to the user if the object requests one automatically.

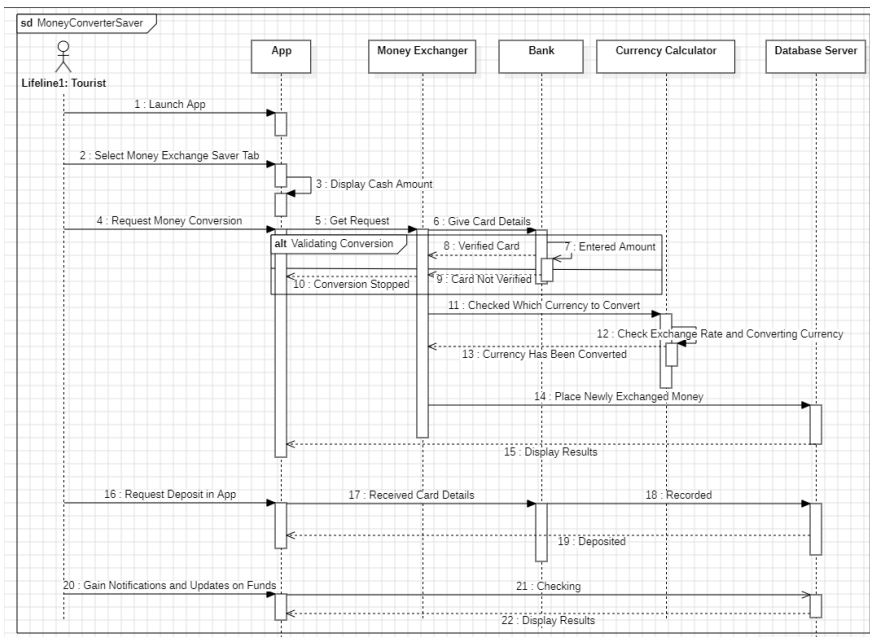


Table 2.4: Sequence Diagram for User Story 3

Accommodation Booking: For the booking of accommodation, users will be able to search for accommodations on the app according to their necessary requirements and filters, the app will then return a list of results. From there a user can then book accommodations, the app will take in the booking which gets sent to the Accommodation List class where bookings are stored, from there that booking is passed on to the accommodation provider for notification. The booking will then be removed from the List and the app will display the confirmed booking to the user.

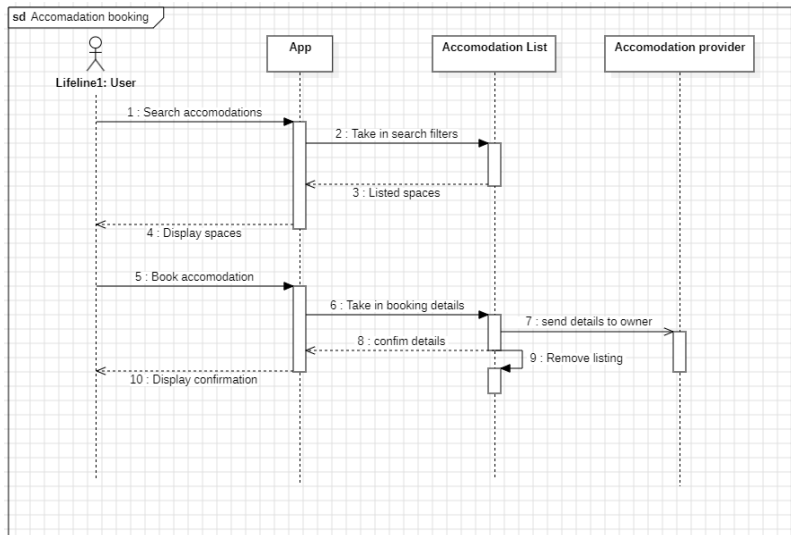


Table 2.5: Sequence Diagram for User Story 4

SEQUENCE DIAGRAMS

Translation: The user opening the app can input text to translate, this text gets sent through the app to the translation database which checks the language and then translates it in the corresponding dictionary. The user can also send a photo to translate, this image then gets sent to the image processor system which extracts the text, this text is then examined for its language and is then translated and sent back to the user. The user can also input feedback based on the translation which gets sent to the translation database.

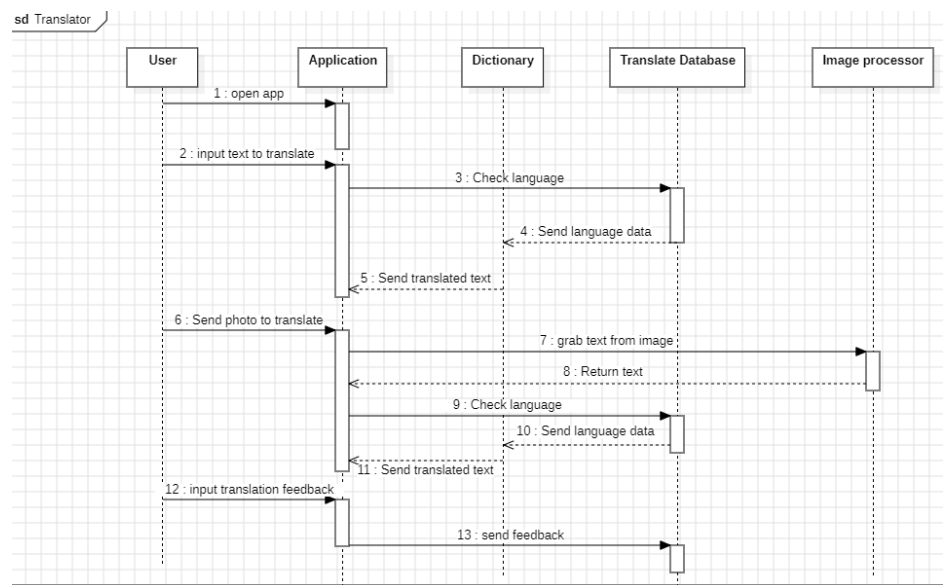


Table 2.6: Sequence Diagram for User Story 5

Social media: The user upon submitting their log in details onto the app, gets their details sent to the server which logs them in. Then the server automatically sends their corresponding feed with all the updated posts and displays it to the user. The user can then create a new post which gets sent from the app to the server, which is then checked by an admin, if a rule is broken then the post is deleted. A message can also be inputed and sent to another user, which gets sent to the messages system. Any replies are then returned from Messages back to the user.

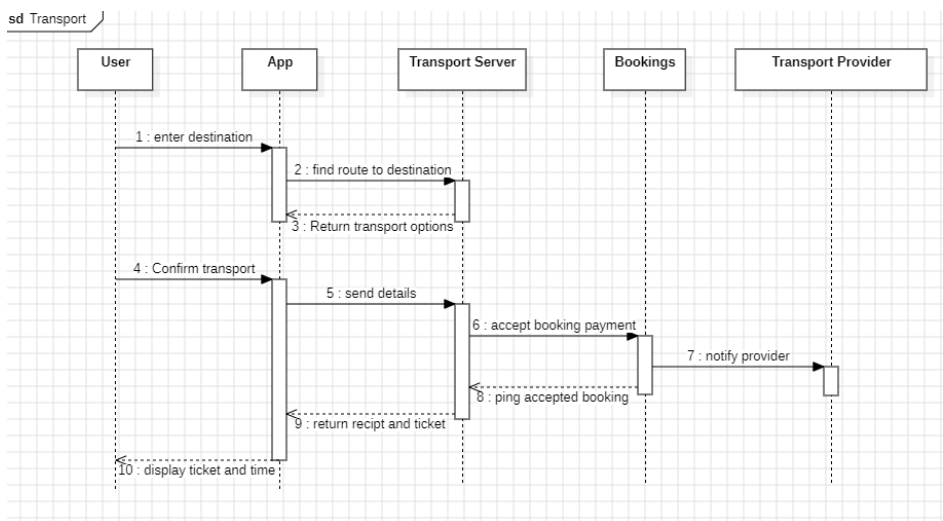


Table 2.7: Sequence Diagram for User Story 6

SEQUENCE DIAGRAMS

Transport: The user can send their destination to the app, which relays that to the transport server to find the best route and returns it to the user. The user can then confirm the selected form of transport and send their booking details to the server, which then accepts their payment and notifies the transport provider. The confirmed booking is pinged back to the server and back to the app from there, displaying the confirmed booking and ticket if applicable to the user, along with the time for their transportation.

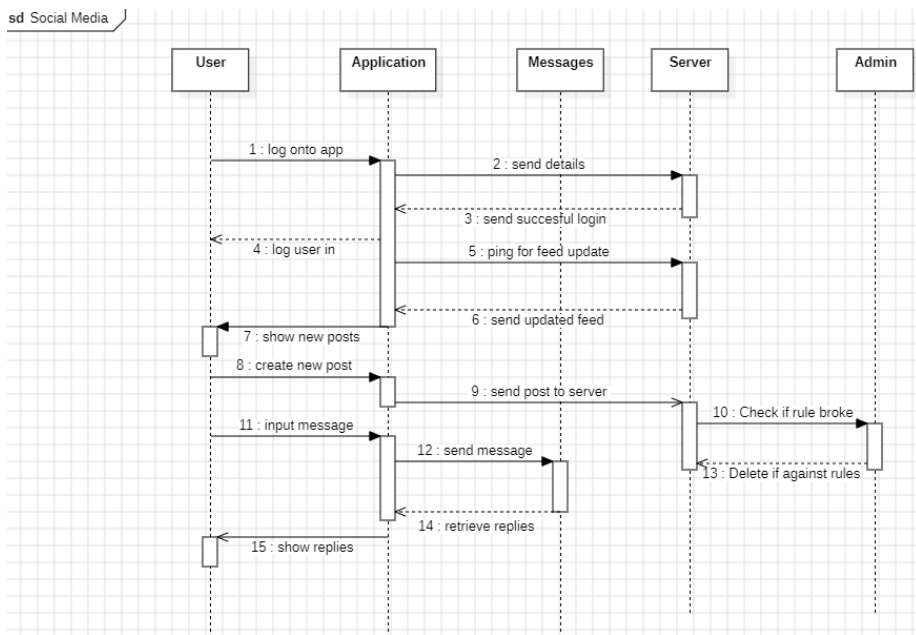


Table 2.8: Sequence Diagram for User Story 7

[Accommodation Provider]: On opening of the app a new user may register a new Accommodation Provider account or login into an existing account, which uses the *AccommdationProvider* object to either register a new accommodation provider user or to find an existing account. Once logged in as an *AP*, the *AccommdationProvider* object may: post or manage listings by calling on the *Listings* object and return the results, manage bookings by requesting it from the *BookingsManager* class and returning the results, post and manage specials by calling the *Specials* class and returns the respected results, and can message with potential customers, requested from the *Messenger* class, with inquires and returns messages.

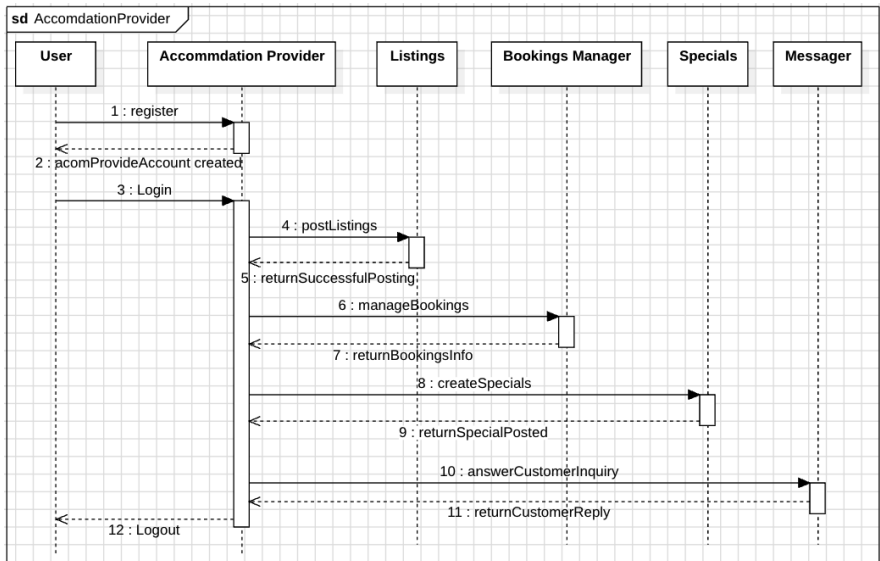


Table 2.9: Sequence Diagram for User Story 8 & 9

SEQUENCE DIAGRAMS

[App Developer]: On app launch, App Developers use their developer account details to login allowing them to access to AD Settings from the *AppDeveloper* object. Once an AD is logged then they can perform actions such as Listing moderation (which is called on from the *Listings* class and returns suspicious listings to the AD), Message moderation (requested from *Messenger* class and returns users who violate community guidelines), and Safety and Security updates and maintenance (pushes security and safety updates to users via *SafetySecuirty* class and returns if the operation was successful or not

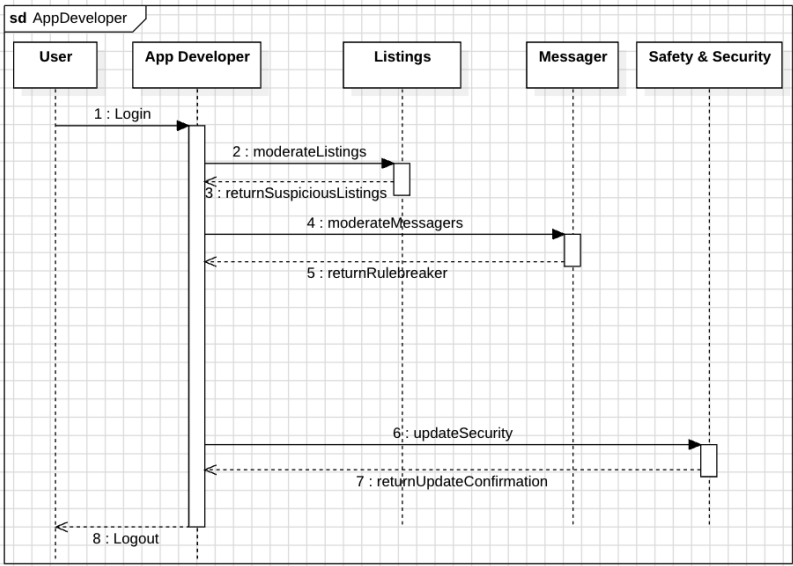


Table 2.10: Sequence Diagram for User Story 10

CLASS DIAGRAM

The travel app's structure is depicted in this class diagram, in which this lists classes and their relationships. The *TravelApp* class is portrayed as the main class with attributes like currency conversion and time management. It communicates with other classes and objects in *AccommodationProvider*, *AppDeveloper* and *Tourist* that all use the app, connecting to its various feature methods like *AccommodationFeature*, *CurrencyConverter*, etc. Most links have been aggregated and associated with the main class as either 1-1 or Many-1, depending on how many people regularly visit the app and navigate inside it. (M)

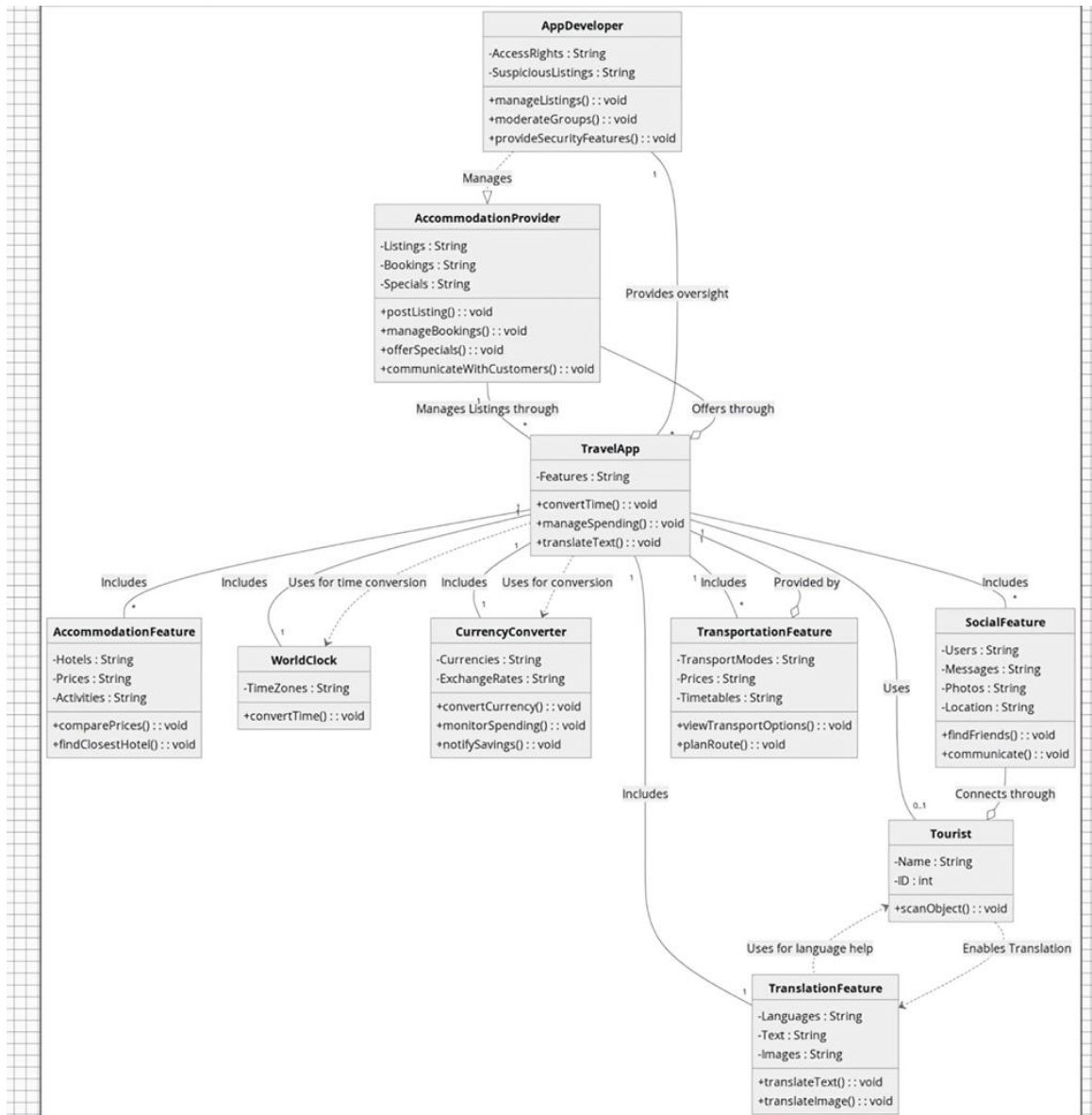


Figure 2.11: Class Diagram for TravelBuddy

ARCHITECTURE DIAGRAM

An architectural diagram is used to represent a system as an organized set of communicating components, in this diagram it is composed of 6 sub-systems: The transport sub-system, translation sub-system, social media sub-system, currency exchange sub-system, time conversion sub-system and smart camera sub-system. These sub-systems are connected to the application backend which hosts the databases and servers to provide data to the app. (N)

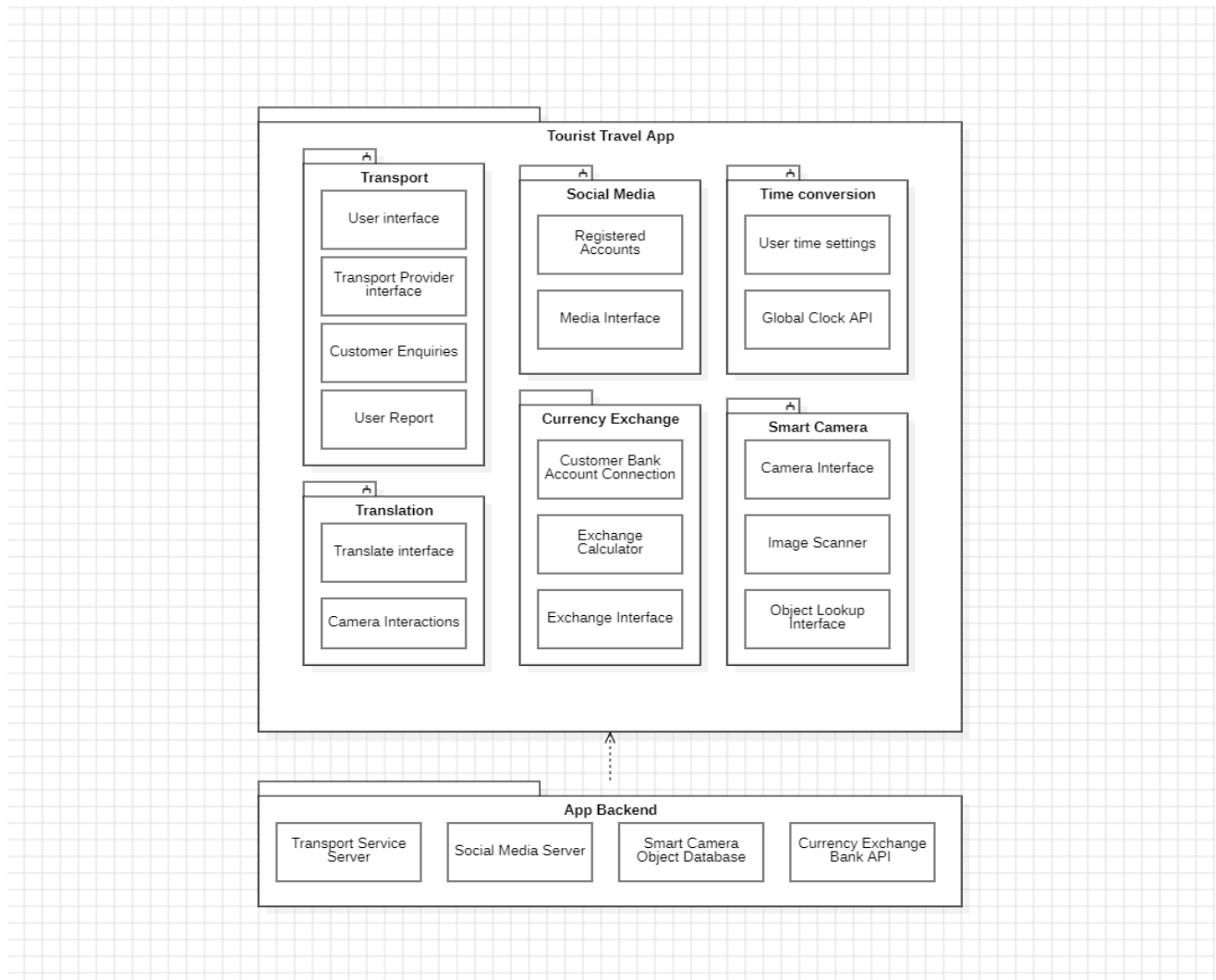


Figure 2.12: Architecture Diagram for TravelBuddy

PART III – USER INTERFACE MOCKUP

For the mockup, I created an app wireframe, which is composed of the login, main homepage, and the four main features accessible at the bottom of the screen at any time which are: the accommodation search and book feature; the transport search and book feature; the text, photo, and image translator; the tourist social media feature; and a currency converter with deals and coupons available. The user can easily access all of these with a few presses with limited user journey required to reach their goal. The user can also edit their profile and details within the settings button accessible from the home page. Ultimately the purpose of the app is to be accessible, simplistic, and useful to tourists and people coming abroad to limit confusion and to better help their experience in a new country. A clickable Wireframe is included inside the project pack. (N)

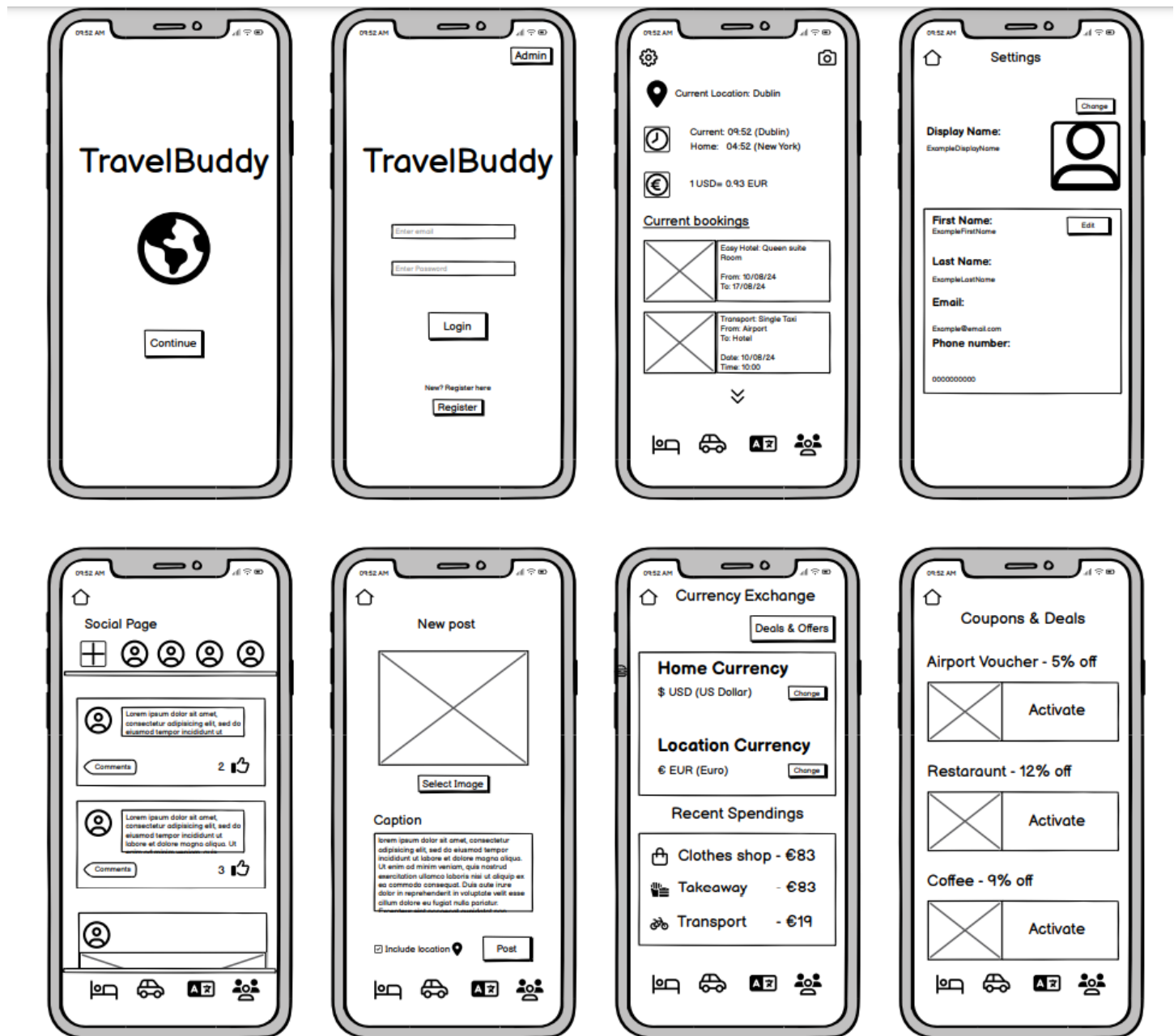


Figure 3.1: TravelBuddy UI Mockup 1



Figure 3.2: TravelBuddy UI Mockup 2

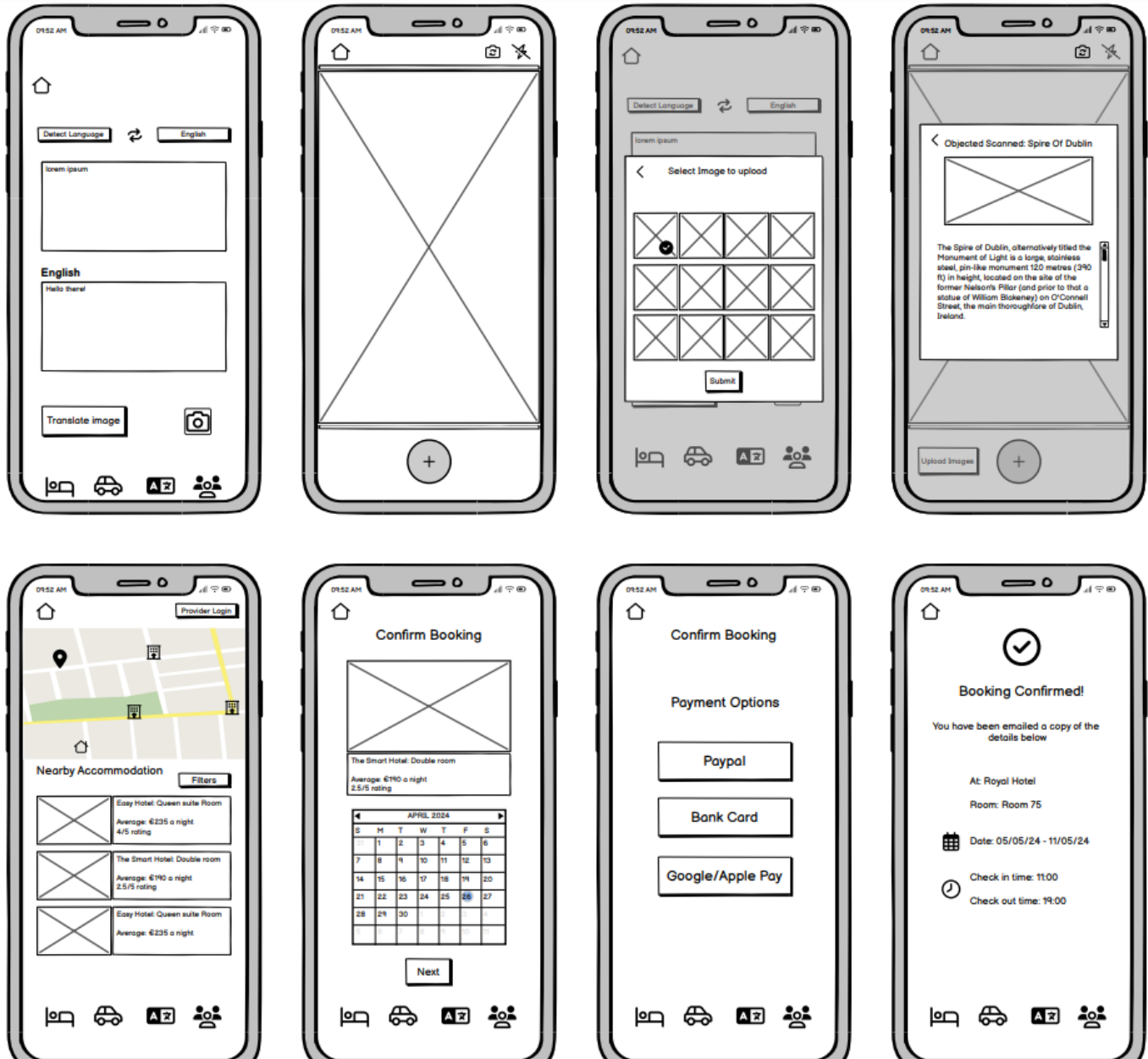


Figure 3.3: TravelBuddy UI Mockup 3

PART IV – SYSTEM TESTS

The importance of System Testing is to show that the system in question (in this case the app we created) does what is intended to do and detect any faults/errors in the process. The way we accomplish this is by conducting tests with two different objectives. Validation testing demonstrates that the system meets its requirements where a successful test is shows the system operated as intended whilst Defect testing determines faults within the system which tests behavior that is abnormal to the system. A successful test shows that the system performs incorrectly to show faults within said system. Below is a table consisting of a select number of both types of system tests (with * indicating defect tests). (R)

Test No.	Scenario	Test Step	Expected Result	Actual Outcome
1	As a tourist, verify that details about landmarks from pictures is received	Take picture of landmark and upload	App should return summarized information about pictured landmark	App returns summarized information of landmark
2	As a tourist, verify that time zones can be converted to other time zones	Input time in current time zone to desired one	App should return current time in desired time zone	App returns converted time in desired time zone
3	As a tourist, verify that exchange rate between two currencies is visible	Input amount to convert into current currency input field	App should return the equivalent amount in desired currency	App returns the equivalent amount in desired currency
4	As a tourist, verify that accommodation around a desired location is visible	Input desired location in input field	App should return lists of accommodation postings/availability	App returns lists of accommodation postings/availability
5*	As a tourist, verify login successful into App Developer settings	Log in into App Developer settings with tourist login	App should not allow access to AD settings and returns unauthorized login attempt to user	App allows login and returns AD settings
6	As tourist, verify messengers send/receive to other users	Input text message in input field	App should return delivered symbol on sent message	App returns delivered symbol on sent message
7	As a tourist, verify public transport is shown and available	Input user current and desired location	App should return public transportation that's nearby	App returns public transportation that's nearby

8	As accommodation provider, verify that postings/listings uploads are successful	Input user's listing	App should return successful posting notification	App returns successful posting notification
9	As accommodation provider, verify messengers send to tourist	Receive tourist inquiry and input user message	App return delivered symbol on sent message	App returns delivered symbol on sent message
10	As App Developer, verify access to security and safety features and moderation tools	Input user login details	App should return successful login message and present tools	App returns successful login message and present tools

Table 4.1: System Tests for TravelBuddy

CONCLUSION

With creation of User stories, a set of UML diagrams, User Interface, and System tests we learnt how we would set the basic groundwork to create a software system in the corporate world and how to collaborate with different people on a shared project. This project allowed us to experience what working will be like in teams and gave us valuable insight on how to work with others.

All figures will be included in the same file as this report.