Traverse

A Sophisticated Travel Application

By

ALEX GREIG

Software Design and Development - Task 2
ST AUGUSTINE'S COLLEGE

A project proposal to Big Time Software for the development of *Traverse*. A mobile application that helps to traverse the complexities of travel by utilising artificial intelligence and sophisticated databases to organise, book and construct an itinerary personalised to each unique individual.

8 MARCH 2022

TABLE OF CONTENTS

CHAPTER 1

PROBLEM DEFINITION

Resurfacing after the tribulations of COVID-19, travel is becoming more prominent than ever as people from all around the world wish to explore and experience the world. Planning and organising holidays has always been time consuming, stressful and confusing. Traverse is a sophisticated travel application that utilises machine learning to enable the user to develop a personalised itinerary within a target budget; and discover and book different transport, activities, restaurants and hotels based off independent reviews and ratings. Complementing the rating system, which is not completed by all travellers, the artificial intelligence will build profiles of different businesses and the frequency and polarity of visits and use that to recommend different activities. The mobile application allows users to store photos within the holiday timeline on their profile and share their experiences with only friends or the public domain. The user can also get inspiration for holiday ideas from friends and family, or the wider community of individuals and groups who travel the world.

Through answering a small number of questions, such as the priorities of travelling (food, culture, adrenaline inducing experiences), the machine learning model will recommend activities associated with a certain destination to the user, allowing for discovery of new places easily and efficiently. Another issue that is prevalent when travelling is the struggle in trying to book accommodation and flights, and the dilemma of which one to book first. Traverse will solve this problem, through simultaneous booking, enabling the user to book both the flights and accommodation without stressing over availability. In conjunction to this, the artificial intelligence will ensure that their is no overlaps in the schedule or excess waiting time for transportation, such as connecting flights, so that the travelling experience will be seamless. Thus the machine learning will use the user preferences, budget, time frame and destination to efficiently build a

personalised itinerary.

The mobile application will give the user the ability to use flight numbers to retrieve flight information from the International Air Transport Association (IATA) and the International Civil Aviation Organization (ICAO) API which can then be input into the itinerary so that all the updated times and locations of each event, for example delayed flights, are in one place. The application will also use bus, ferry and train timetables to specify in their schedule the different options the user can use to get between each location, highlighting the most time and cost effective. Therefore, the traditional approach of navigating through websites and convoluted booking sites to try and plan travel is significantly improved by Traverse, as all aspects of planning travel are located in one application, allowing smooth travel that will lead to new experiences and adventures.

CHAPTER 2

IPO DIAGRAM

Table 2.1: IPO Table

Input	**Processes**	**Output**
Button input of sign up or login	The application will take the user to the requested landing page so that they can login or sign up to the application. If the user has requested for their login details to be remembered in a past session, the login process will be completed using the stored credentials on the mobile so the user will be taken straight to the main page.	The landing page that was requested will be loaded for the user; this will be the verification to the user that the process has worked.
New username, password, email	The application will take the inputs from the signup page and validate the length, complexity of the password and validate that the username is unique and does not exist in the database. If the username and password are validated successfully then the application will create a new record in the database with the user information, however the boolean that stores the email validation status will be false. After this a confirmation email will be sent to the user inputted email, which when seen and confirmed will change the boolean to true.	The user will see a success message on the screen when the username and password are validated and a reminder to go their email and confirm it is the right email. Once their email is confirmed, they will be taken to the home page of the application.

Continued on next page

Table 2.1: *(Continued)* IPO Table

Input	Processes	Output
Username, Password	The application will take the user input from the login landing page and check against a database to validate the login details. The application will check if the username exists within the database, then crosscheck the entered password against the encrypted password within the database.	If the password and username are successfully validated then the application will output a success message to the screen and the user will be taken to their home page on the mobile application.
The user inputs the location they want to travel to and the time frame of the holiday. Using a input bar, the user enters the budget they have for travel and the user also inputs a float between 0 and 1 using a slider on a range of questions that assess the priorities of the trip.	The budget that the user entered is validated to make sure that it is a float and does not contain any characters other than numbers or a decimal place. The application utilises artificial intelligence, more specifically, machine learning to build profiles of the user and use this information to recommend and build a customised itinerary within the budget and time frame that they specified. Then the application will find flights and accommodation, using the flights api and database, based upon the user profile. The application will retrieve information from its database pertaining to the location that the user inputs and make decisions on whether to include the activities in the itinerary. This will be based upon the budget and priorities that the user has entered. The application will check the availability of each activity and make sure that all the times line up	The output to the user will be the different transportation, hotels and activities that are recommended to the user when building their itinerary. The sum of the costs will be shown at the bottom next to the specified budget. At the end, the whole itinerary will be displayed and if the user wishes they can make specific additions or change their priorities / budget and let the machine learning model shift their itinerary to meet their new wishes.

Continued on next page

Table 2.1: *(Continued)* IPO Table

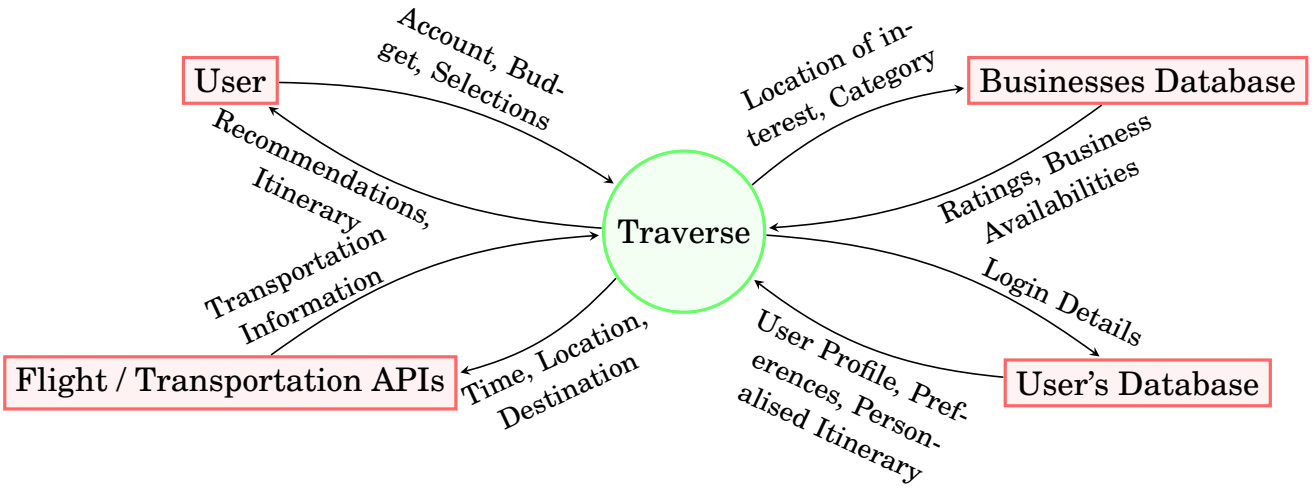
Input	Processes	Output
Confirm booking button	At the end this button will signal the application to book all the events in the schedule. Before this is done it will ensure that all events can be booked and there is availability.	Success message will be shown when all events are booked, while it is completing this process there will be a loading animation to inform the user that the process is happening. If there are certain events that don't fit in the schedule or are not available, new recommendations will be given or they will ask the user to change them.

CHAPTER 3

CONTEXT AND DATAFLOW DIAGRAMS

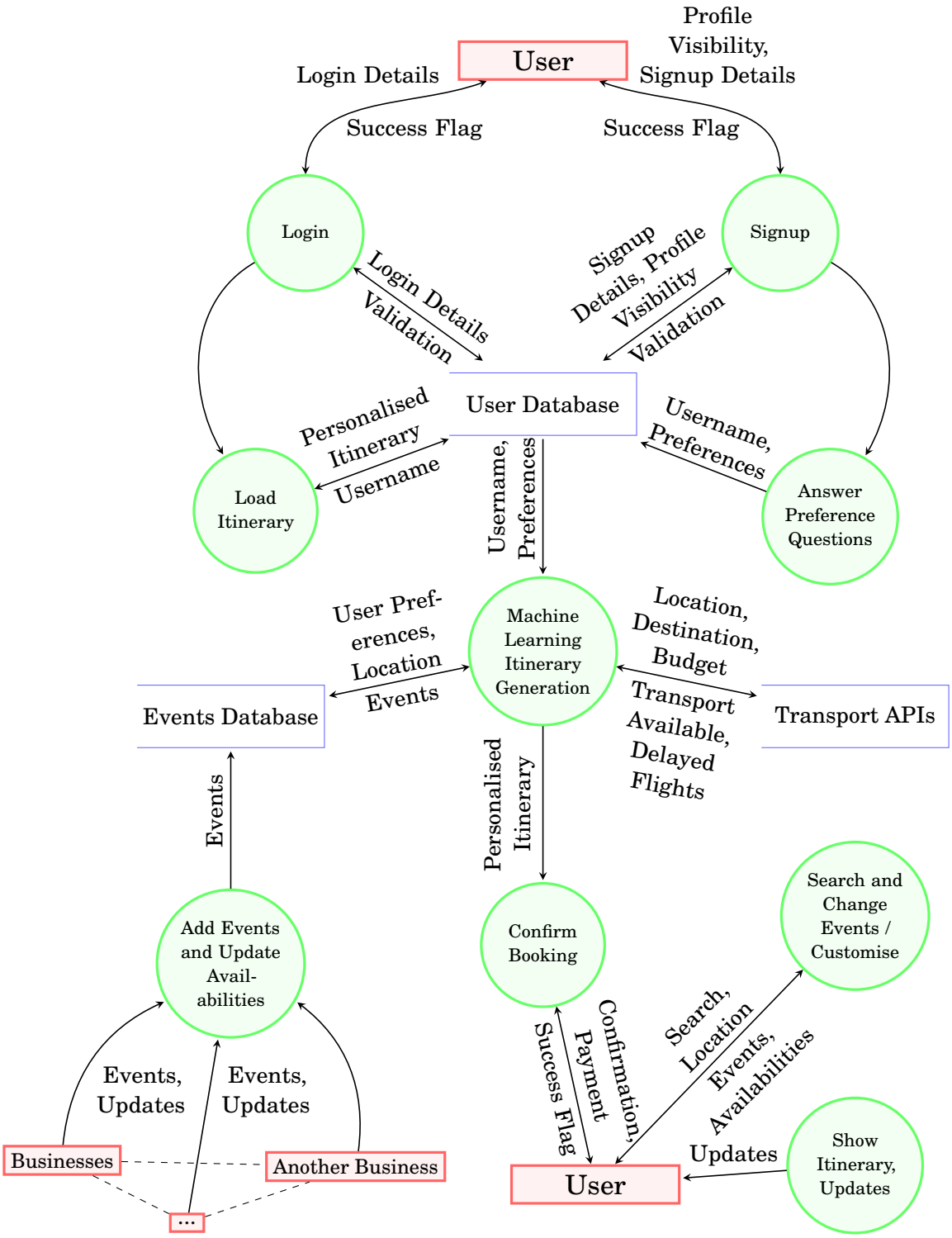
3.1 Context Diagram

Figure 3.1: Context Diagram



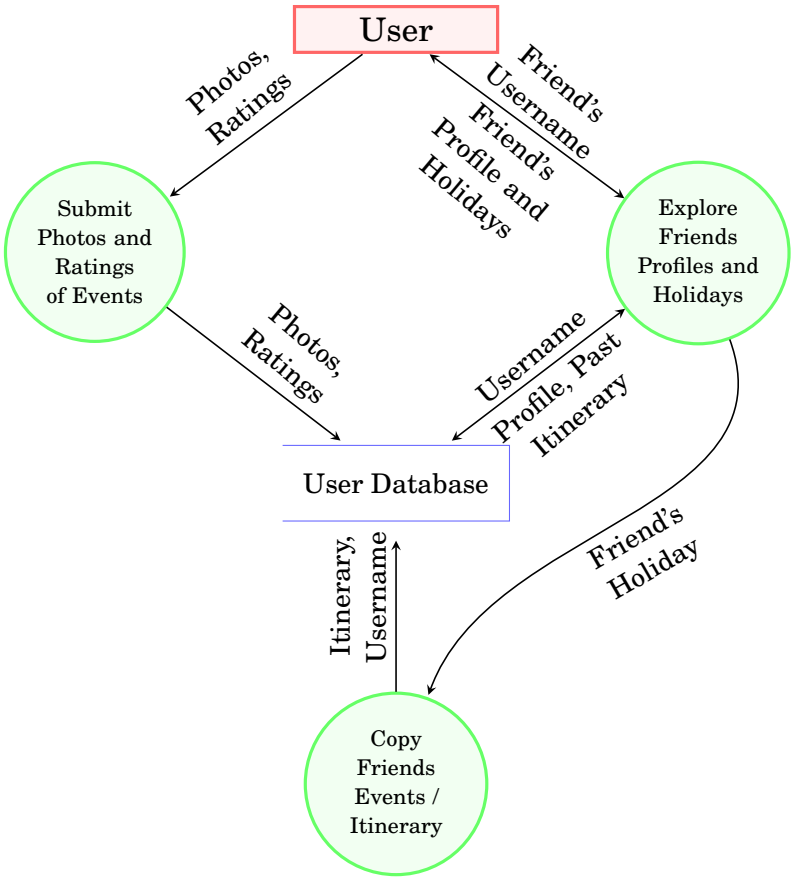
3.2 Data Flow Diagram - Core Functionality

Figure 3.2: Data Flow Diagram - Core Functionality



3.3 Data Flow Diagram - Social Functionality

Figure 3.3: Data Flow Diagram - Social Functionality



CHAPTER 4

ALGORITHMS

The core algorithms that will be used in the application are displayed below, written in the pseudo-code syntax:

4.1 Main Program

Listing 4.1: Main Program

```
1  BEGIN MAINPROGRAM
2     LoadLandingPage() // Instantiates the program and loads the landing
          page with buttons for the user's selection.
3     IF SignUp.pressed() == True THEN
4        LoadSignUp() // Loads the sign up
5        INPUT username, password, email, confirm_password
6        IF username.entered() == True AND email.entered() == True AND
             password.entered() == True AND password == confirm_password
             THEN
7           IF username.validate() == True AND password.validate() == True
                THEN
8              Display tickIcon
9              SendValidationEmail() // Sends a validation email to the
                   email address that was inputted by the user, so that it
                   can be verified.
10             Display "Verification Email Sent!"
11             LoadPreferences() // Loads the first question and the
                   associated input sliders within them.
12             WHILE question <> last_question DO
```

```
13              INPUT answer // Answer is inputted through slider so the
                    answer will be a float.
14              IF next_question.pressed() == True AND question_answered
                    () THEN
15                  LoadNextQuestion()
16                  UpdateUserData(question, username, answer) // Data
                        will be entered into the database based upon the
                        type of data so that the machine learning model
                        can use the information to predict recommendations
                         that the user will like. This is done so that
                        this function can be reused to update the User
                        Profile Database.
17
18              ELSE IF skip_question.pressed() == True THEN
19                  LoadNextQuestion()
20              ENDIF
21          ENDWHILE
22
23          INPUT answer // Answer is inputted through slider so the
                answer will be a float.
24          IF next_question.pressed() == True AND question_answered()
                THEN
25              UpdateUserData(username, question, answer)
26          ENDIF
27          LoadTripPlanning()
28          INPUT budget, from, to, duration, group_size // from and to
                 are the beginning and ending destinations.
29          business_records = ReadBusinessData() // Interacts with
                Business Database and reads the availabilities and
                different business events into memory.
30          user_data = ReadUserData() // Reads user data into memory
                to input into the machine learning model for
                recommending and optimising itinerary.
31          transport_data = TransportAPI(duration) //Retrives
                transport data within the duration period which can be
                used to decide the method of transport between locations
                 when travelling.
32          ml_input_data = business_records, user_data, budget, from,
                to, duration, group_size, transport_data // Data that
                the machine learning model will use to make
                sophisiticated decisions about events to recommend the
                user. This datatype is a tuple as it contains multiple
                different datatypes.
```

```
33              ml_recommendations =
                    MachineLearningRecomendations(ml_input_data)
34              LoadFlights(ml_recommendations) // Parameters of the
                    function are the ml recommendations as these
                    recommendations will then be loaded for the user.
35              INPUT flights_chosen
36              UpdateUserData("Flights", username, flights_chosen) //
                    Parameters are the dataID, data and the username of the
                    User.
37
38              LoadHotels(ml_recommendations)
39              INPUT hotels_chosen
40              UpdateUserData("Hotels", username, hotels_chosen)
41
42              LoadActivities(ml_recommendations)
43              INPUT activities_chosen
44              UpdateUserData("Activities", username, activities_chosen)
45
46              LoadDiscover() // Loads the discover page which lets the
                    user customise their itinerary.
47              INPUT extra_activities
48              UpdateUserData("Activites", username, extra_activities)
49
50              LoadPayment()
51              INPUT payment_method
52              ConcurrentBooking(payment_method, user_data)
53              DISPLAY Success
54
55              HomeItinerary() // Function that loads the itinerary and
                    other options for the user. This will take the user to
                    the main page and enable them to explore other
                    activities, look at their flights and interact with the
                    social options of the mobile application.
56
57          ELSE
58              Display crossIcon
59          ENDIF
60      ENDIF
61      IF username.entered() == True THEN
62          If username.validate() == True THEN
63              Display tickIcon
64          ELSE
65              Display crossIcon
```

```
66          ENDIF
67       ENDIF
68    ELSE IF SignIn.pressed() == True THEN
69       LoadSignIn() // Loads the sign in functionality
70       IF username.entered() == True AND password.entered() THEN
71          If username.validate() == True THEN
72             Display tickIcon
73             HomeItinerary() // Function that loads the itinerary and
                   other options for the user. This will take the user to
                   the main page and enable them to explore other
                   activities, look at their flights and interact with the
                   social options of the mobile application.
74
75          ELSE
76             Display crossIcon
77          ENDIF
78       ENDIF
79    ENDIF
80
81 END MAINPROGRAM
```

4.2 Update User Data

Listing 4.2: Update User Data

```
1  BEGIN UpdateUserData(dataID, username, data)
2     OPEN UserDatabase for relative access
3     dataIDs = array of identifiers of all questions // These should
          correlate to dataID passed into function when retrieving the user
           preferences.
4     FOR i=0 to users.len() DO
5        IF users[i].username == username THEN
6           FOR i=0 to dataIDs.len() DO
7              IF dataIDs[i] == dataID THEN
8                 WRITE UserData.users[i] from data using dataID
9              ENDIF
10          NEXT i
11       ENDIF
12    NEXT i
13    CLOSE UserDatabase
14 END UpdateUserData
```

Machine learning algorithms in recommender systems are typically classified into two categories — content based and collaborative filtering methods although modern recommenders combine both approaches. Collaborative filtering looks for patterns in the user activity in relation to other users, to produce user specific recommendations whereas Content-based filtering recommends items with similar content (e.g. metadata, description, topics, keywords) to the items the user has liked or indicated to like in the past.

For this application the approach that would be taken based upon the preferences that are retrieved from the beginning would be a content-based filtering the user has indicated to some level their priorities when travelling and has given data through answering questions about preferences and other inputs such as budget. First the algorithm will find similarities between the events and activities within the business database and use the answers from the user to generate a similarity matrix. From this, the machine learning model will create Bayesian networks to model the links between the different events and the weightings of similarity, enabling recommendations to be made.

4.3 Machine Learning Recommendations and Optimisation

Listing 4.3: Machine Learning Recommendations and Optimisation

```
1  BEGIN MachineLearningRecomendations(ml_input_data)
2     business_records, user_data, budget, from, to, duration, group_size,
          transport_data = ml_input_data // Data that the machine learning
          model will use to make sophisticated decisions about events to
         recommend the user. This is decompressing the tuple into multiple
          variables for utilisation.
3     FOR i=0 to user_data.len() DO
4        FOR j=0 to business_records.events.len() DO
5           similarity_matrix[i][j] = cosine_similarity(user_data[i],
               budget, business_records.events[i]) // Constructs the
               similarity matrix based upon the similarity of user
               preferences and target budget with the different events.
6        NEXT j
7     NEXT i
8
9     bayesian_network =
         BayesianGeneration(similarity_matrix, transport_data) //Transport
          data is used in the Bayesian network generation to link events
         not only by similarity and budget, but also the ability to be
         transported between the different locations.
10
11    availabilities = business_records.availability
```

```
12      recommendations = GradientDescentOptimisation(availabilities,
            duration, user_data, bayesian_network) //Optimisation of the
            timing of each event to make sure that the Itinerary lines up.
            Works by making steps towards a perfectly fitted timeslot by
            moving towards a minima and evaluating its performance based on a
             reward function. Utilises the events that are linked with the
            highest weighted similarity on the Bayesian network.
13
14      RETURN recommendation // Data type of recommendation is an event
            from business_records.events
15
16
17  END MachineLearningRecomendations(ml_input_data)
```

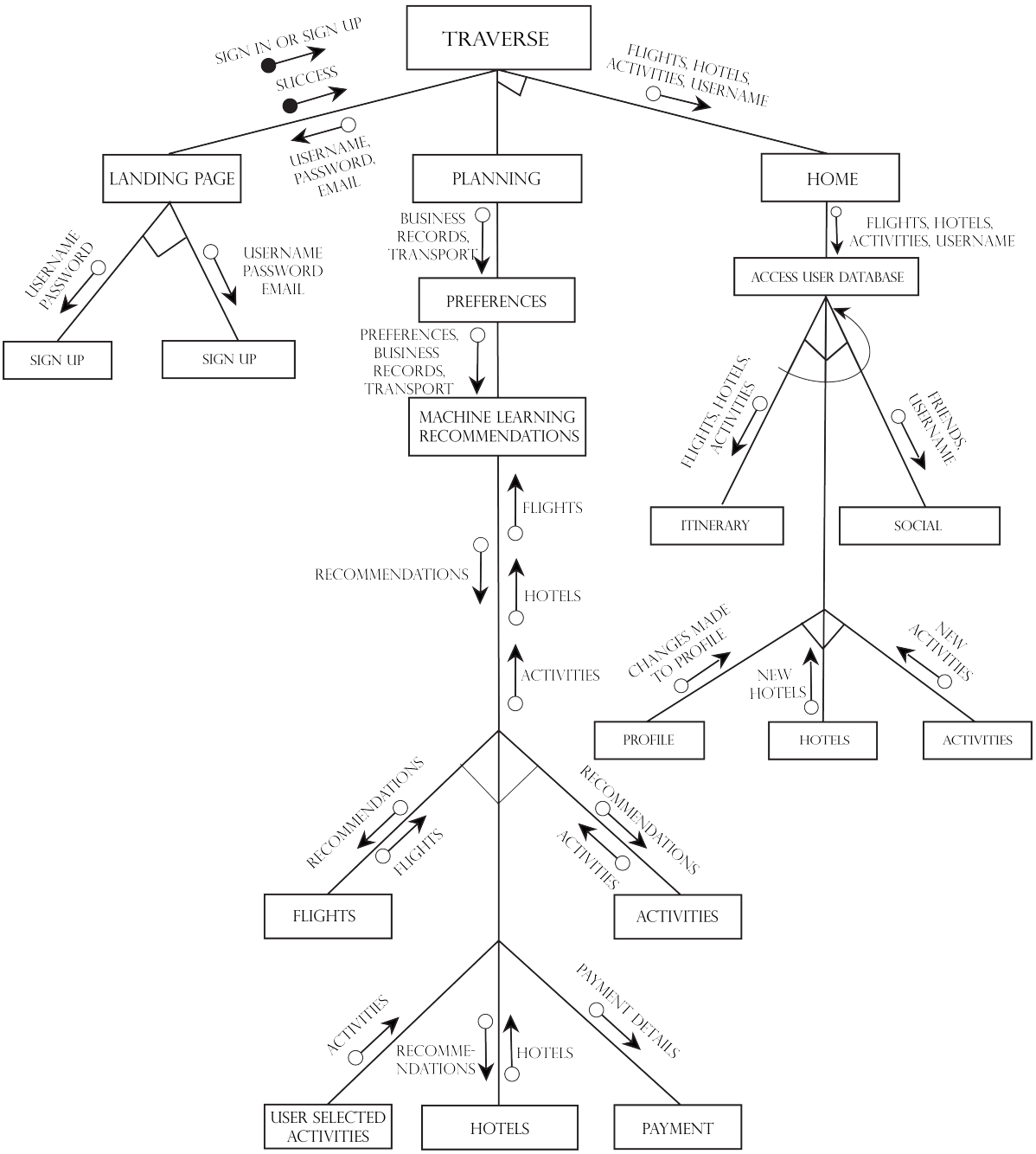
4.4 Interaction with Transport APIs

Listing 4.4: Interaction with Transport APIs

```
1  BEGIN TransportAPI(duration)
2      transport_services = multidimensional array of n elements of [
           transportID, api_key, query_address] // All the api_keys needed
           to access the APIs of different transportation APIs, the
           multidimensional array of 3 by n, is so that each api key is
           coupled with a transport ID and query address so that each
           transport service can be identified and queried.
3
4      transport_data = empty multidimensional array // This array will
           contain each transport and the times of arrival and location for
           each transport service.
5      FOR i=0 to transport_services.len() DO
6          transport_data[i] = HttpRequest(transport_services[i][0],
               transport_services[i][1], transport_services[i][2])
7      NEXT i
8      RETURN transport_data
9  END TransportAPI
```

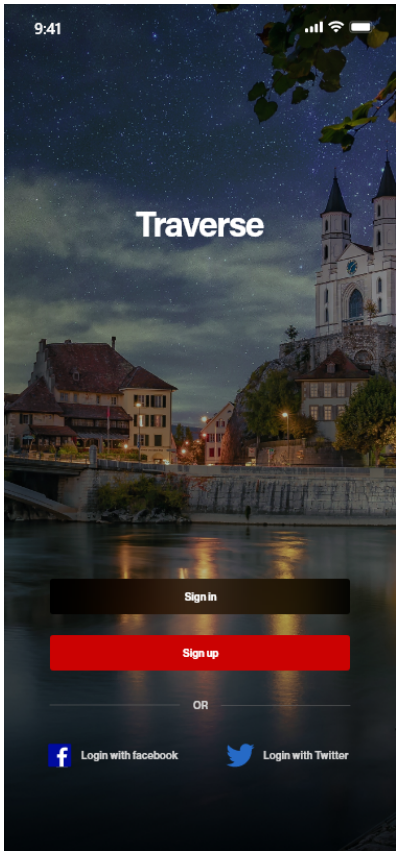
CHAPTER 5

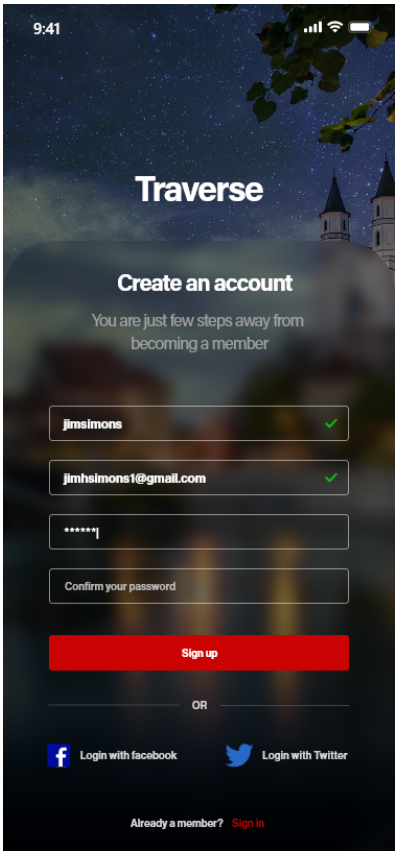
Structure Chart

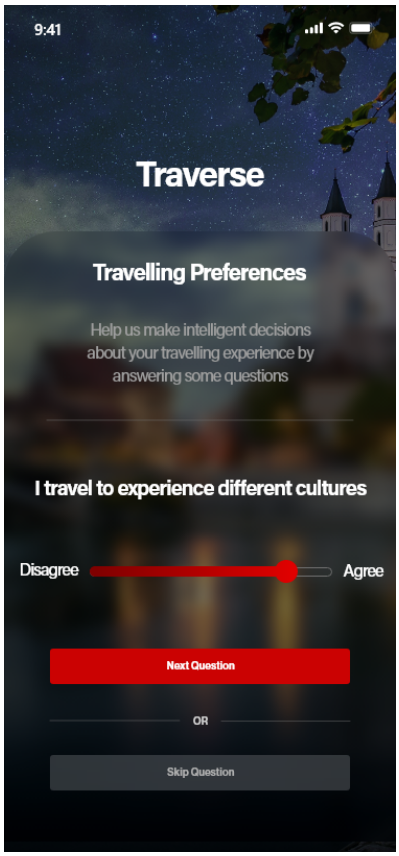

TRAVERSE
SIGN IN OR SIGN UP
SUCCESS
USERNAME, PASSWORD, EMAIL
FLIGHTS, HOTELS, ACTIVITIES, USERNAME
LANDING PAGE
PLANNING
HOME
USERNAME PASSWORD
USERNAME PASSWORD EMAIL
SIGN UP
SIGN UP
BUSINESS RECORDS, TRANSPORT
PREFERENCES
PREFERENCES, BUSINESS RECORDS, TRANSPORT
MACHINE LEARNING RECOMMENDATIONS
FLIGHTS
RECOMMENDATIONS
HOTELS
ACTIVITIES
FLIGHTS, HOTELS, ACTIVITIES, USERNAME
ACCESS USER DATABASE
FLIGHTS, HOTELS, ACTIVITIES
FRIENDS, USERNAME
ITINERARY
SOCIAL
CHANGES MADE TO PROFILE
NEW HOTELS
NEW ACTIVITIES
PROFILE
HOTELS
ACTIVITIES
RECOMMENDATIONS
FLIGHTS
RECOMMENDATIONS
ACTIVITIES
FLIGHTS
ACTIVITIES
ACTIVITIES
RECOMME-NDATIONS
HOTELS
PAYMENT DETAILS
USER SELECTED ACTIVITIES
HOTELS
PAYMENT

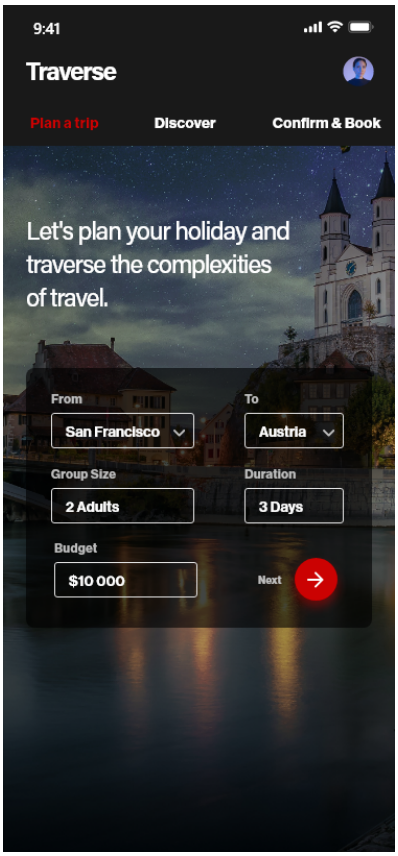
CHAPTER 6

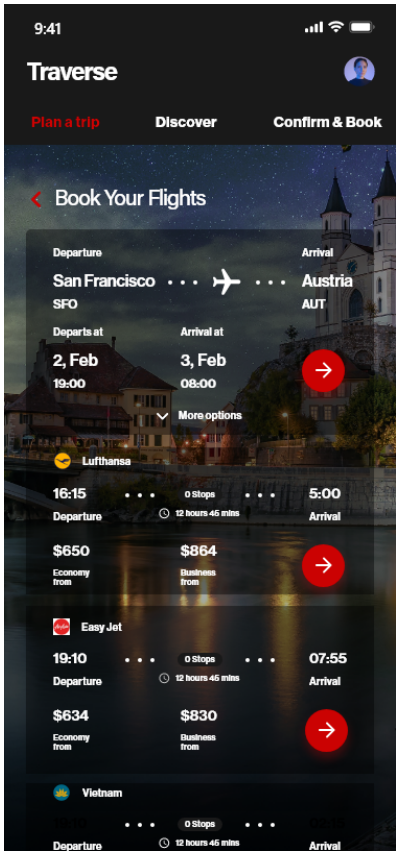
Interface Design

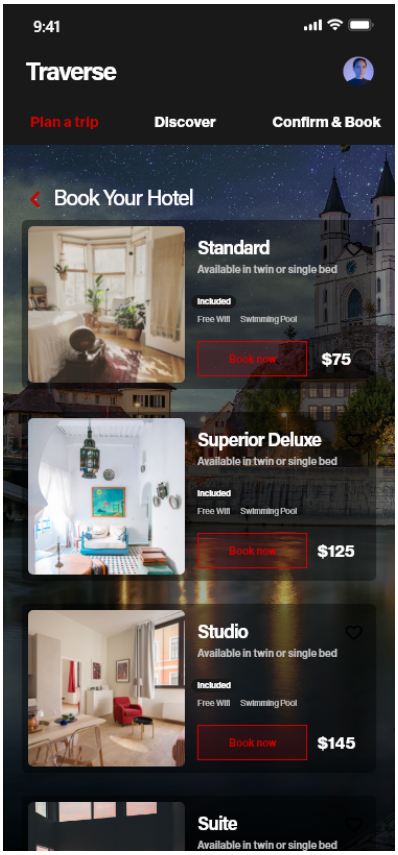

9:41
Traverse
Sign in
Sign up
OR
Login with facebook
Login with Twitter

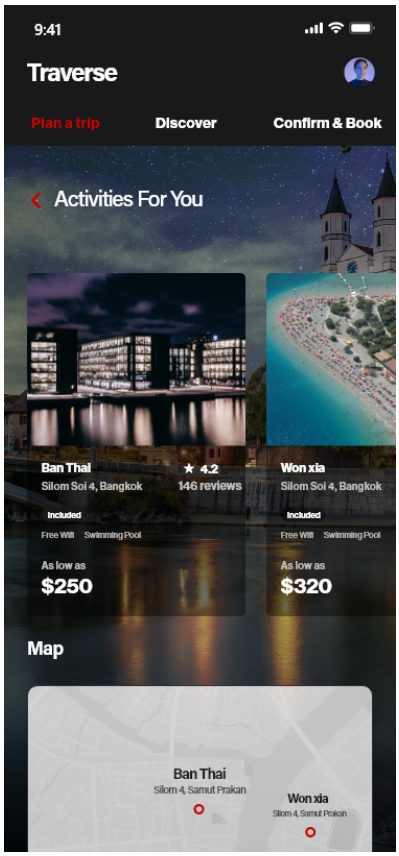

9:41
Traverse
Create an account
You are just few steps away from becoming a member
jimsimons
jimhsimons1@gmail.com
Confirm your password
Sign up
OR
Login with facebook
Login with Twitter
Already a member? Sign in

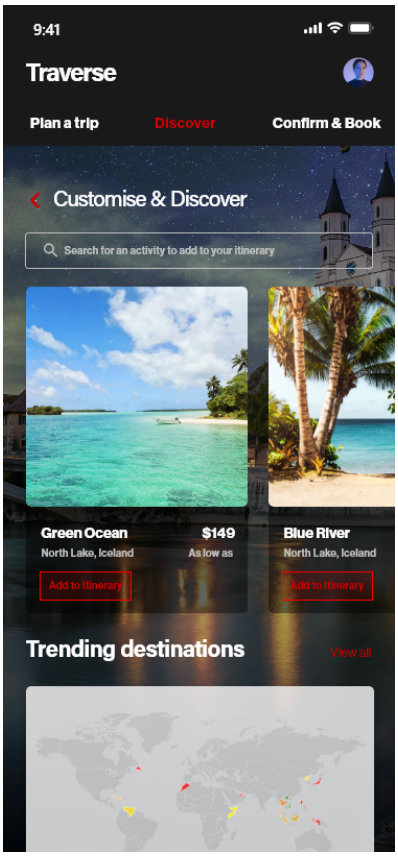

9:41
Traverse
Travelling Preferences
Help us make intelligent decisions about your travelling experience by answering some questions
I travel to experience different cultures
Disagree
Agree
Next Question
OR
Skip Question

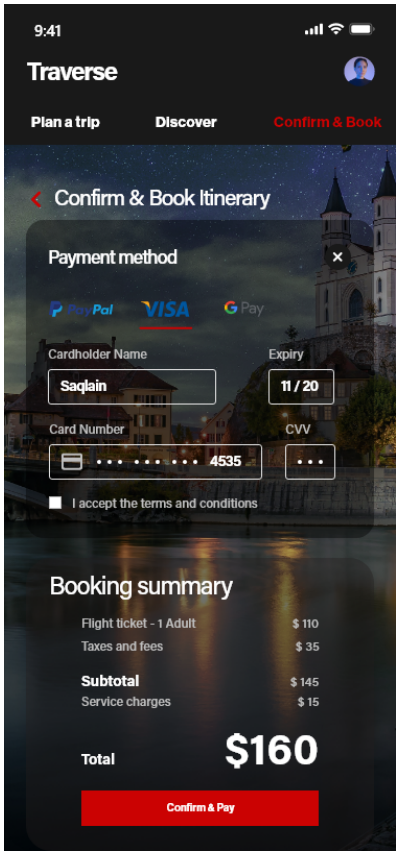

9:41
Traverse
Plan a trip
Discover
Confirm & Book
Let's plan your holiday and traverse the complexities of travel.
From
San Francisco
To
Austria
Group Size
2 Adults
Duration
3 Days
Budget
$10 000
Next

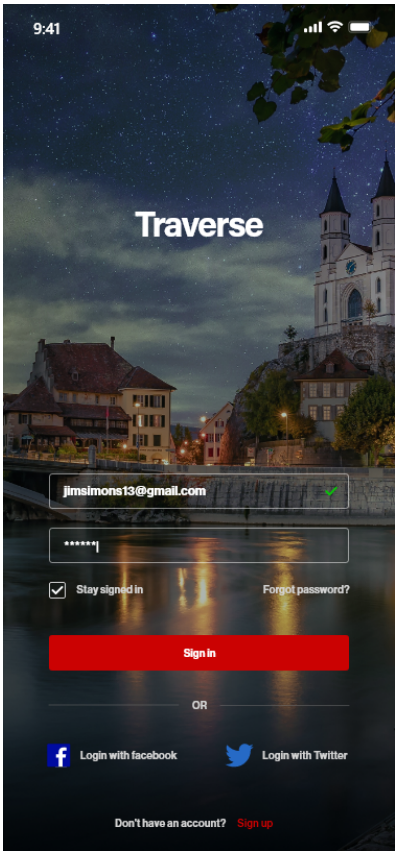

9:41
Traverse
Plan a trip
Discover
Confirm & Book
Book Your Flights
Departure
Arrival
San Francisco
SFO
Austria
AUT
Departs at
Arrival at
2, Feb
19:00
3, Feb
08:00
More options
Lufthansa
16:15
0 Stops
5:00
Departure
12 hours 45 mins
Arrival
$650
Economy from
$864
Business from
Easy Jet
19:10
0 Stops
07:55
Departure
12 hours 45 mins
Arrival
$634
Economy from
$830
Business from
Vietnam
0 Stops
Departure
12 hours 45 mins
Arrival

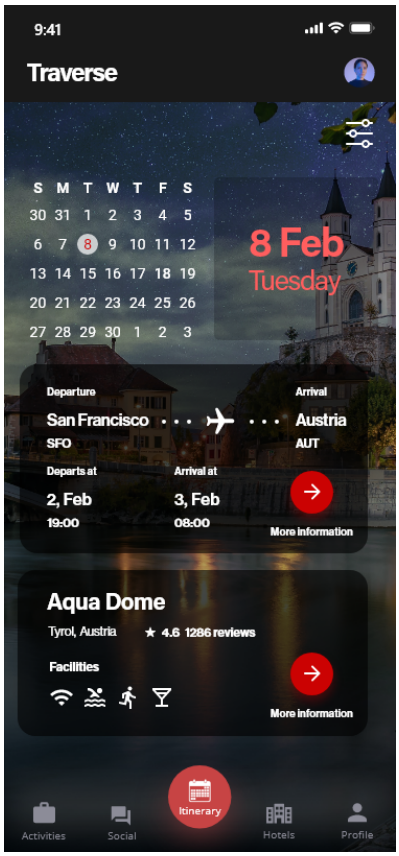

9:41
Traverse
Plan a trip
Discover
Confirm & Book
Book Your Hotel
Standard
Available in twin or single bed
Included
Free Wifi
Swimming Pool
Book now
$75
Superior Deluxe
Available in twin or single bed
Included
Free Wifi
Swimming Pool
Book now
$125
Studio
Available in twin or single bed
Included
Free Wifi
Swimming Pool
Book now
$145
Suite
Available in twin or single bed

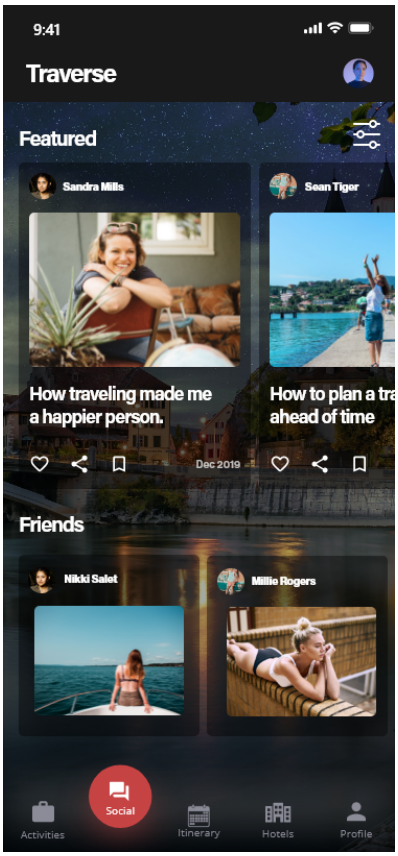

9:41
Traverse
Plan a trip
Discover
Confirm & Book
Activities For You
Ban Thai
Silom Soi 4, Bangkok
4.2
146 reviews
Included
Free Wifi
Swimming Pool
As low as
$250
Won xia
Silom Soi 4, Bangkok
Included
Free Wifi
Swimming Pool
As low as
$320
Map
Ban Thai
Silom 4, Samut Prakan
Won xia
Silom 4, Samut Prakan


9:41
Traverse
Plan a trip
Discover
Confirm & Book
Customise & Discover
Search for an activity to add to your itinerary
Green Ocean
North Lake, Iceland
$149
As low as
Add to itinerary
Blue River
North Lake, Iceland
Add to itinerary
Trending destinations
View all


9:41
Traverse
Plan a trip
Discover
Confirm & Book
Confirm & Book Itinerary
Payment method
PayPal
VISA
G Pay
Cardholder Name
Saqlain
Expiry
11 / 20
Card Number
4535
CVV
I accept the terms and conditions
Booking summary
Flight ticket - 1 Adult
$ 110
Taxes and fees
$ 35
Subtotal
$ 145
Service charges
$ 15
Total
$160
Confirm & Pay


9:41
Traverse
jimsimons13@gmail.com
Stay signed in
Forgot password?
Sign in
OR
Login with facebook
Login with Twitter
Don't have an account?
Sign up

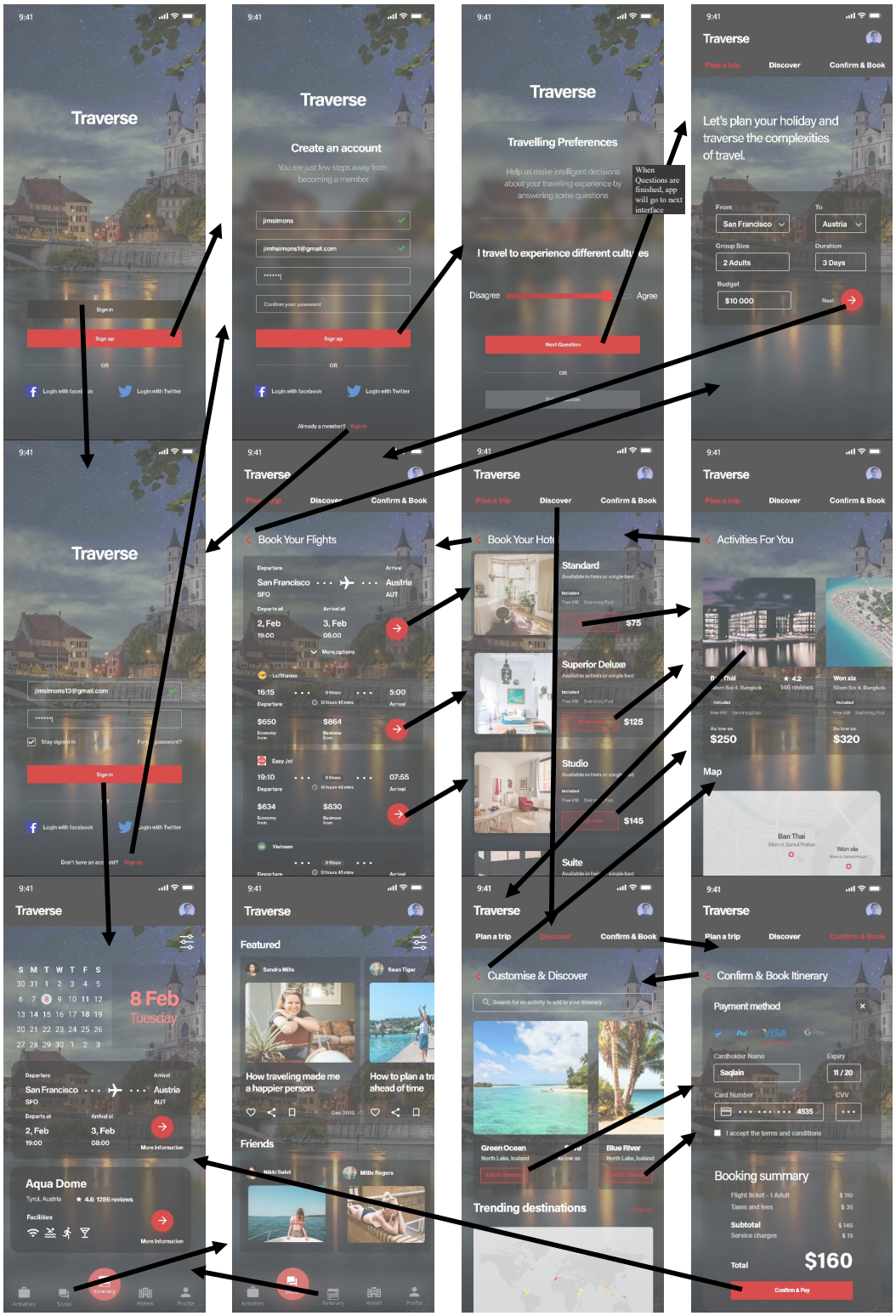

9:41
Traverse
S M T W T F S
30 31 1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 1 2 3
8 Feb
Tuesday
Departure
San Francisco
SFO
Arrival
Austria
AUT
Departs at
2, Feb
19:00
Arrival at
3, Feb
08:00
More information
Aqua Dome
Tyrol, Austria
★ 4.6 1286 reviews
Facilities
More information
Activities
Social
Itinerary
Hotels
Profile


9:41
Traverse
Featured
Sandra Mills
Sean Tiger
How traveling made me a happier person.
Dec 2019
Friends
Nikki Salet
Millie Rogers
Activities
Social
Itinerary
Hotels
Profile

CHAPTER 7

STORYBOARD

Shown below is the typical navigation of the user interface:



CHAPTER 8

SOCIAL AND ETHICAL ISSUE

The development and design of software solutions carries the weight of being socially and ethically responsible. The social and ethical issues surrounding this application are privacy and inclusivity. Privacy is about protecting an individual's personal information. Personal information is any information that allows others to identify you. Privacy is a fundamental principle of our society, we have the right to know who holds our personal information. In Australia, privacy is legally protected under the Privacy Act 1988 and its subsequent amendments. This act contains ten National Privacy Principles, that set standards that organisations are required to meet when dealing with personal. As the Application is developed and operated from Australia, it needs to follow these privacy laws and obligations. This issue applies to this application the collection of data such as location, preferences, budgets are private and the distribution of that data, such as locations that are being visited needs to be carefully monitored and restricted. The data from this application must be encrypted within a database so that unwanted access to the user database would not mean that all user data is exposed. Another very significance appearance of privacy issues within Traverse is the transference of payment details. These financial details would need to be encrypted and communicated over a TLS stream to ensure security. As this is extremely sensitive information any critical issue within the software needs to be resolved in a timely, accurate and efficient manner.

Another social and ethical issue is the inclusivity of the application as software design should include functionality that allows software to be used and accessed by a wide range of users. People who are visually or physically impaired need to have accessibility functions that allow them to utilise the software. This means that certain transport or activities need to consider the physical abilities of the individual before recommending it to them.