

SC2006: Software Engineering

Lab 2 Deliverables

WanderMap

Ang Yvette (U2220779L)

Chua Sym Eyan (U2221430B)

Tan Hui Ling (U2222280A)

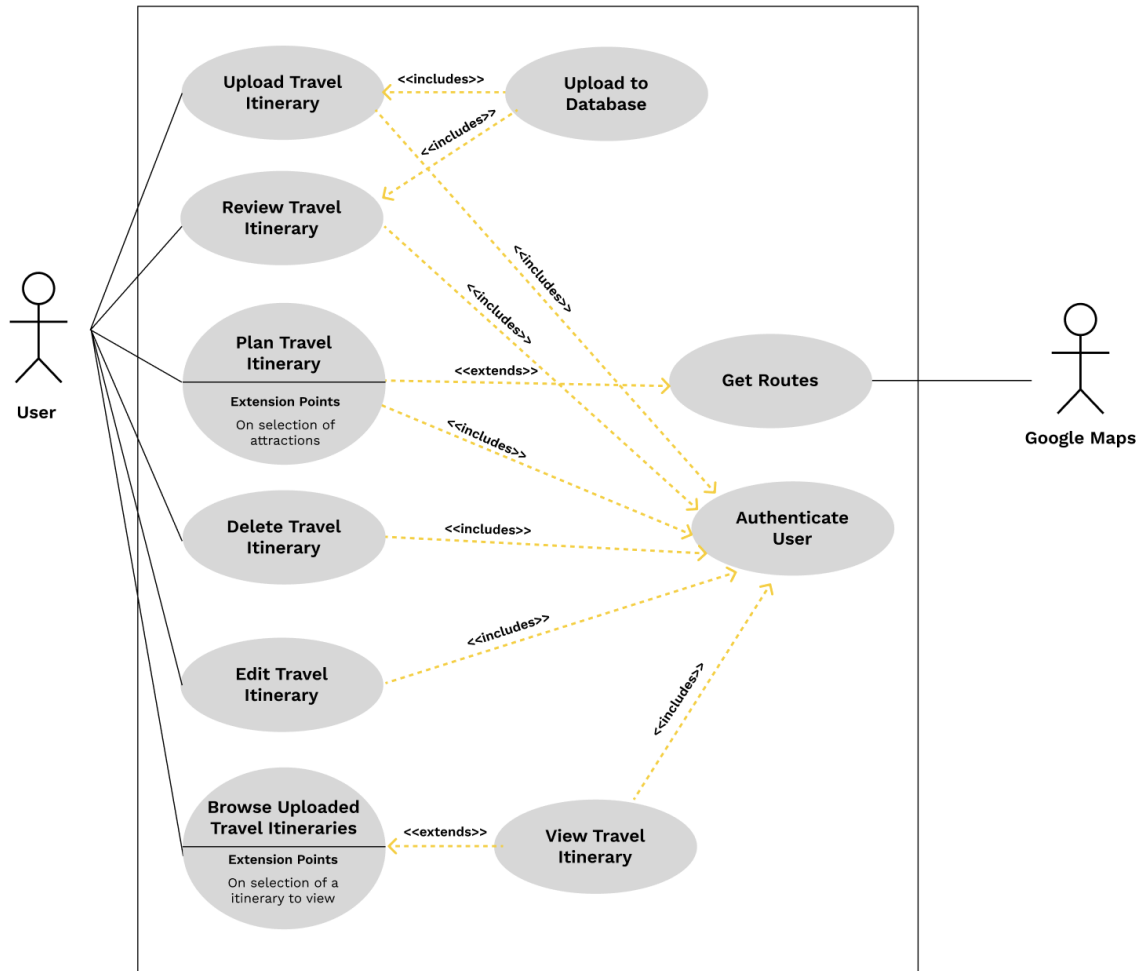
Dan He (N2304782D)

Content Page

Use Case Diagram.....	3
Use Case Diagram.....	3
Use Case Description.....	4
Conceptual Model.....	16
Class Diagram.....	16
Dynamic Model.....	17
Key Boundary Classes and Control Classes.....	17
Sequence Diagram.....	18
Login.....	18
Sign up.....	19
Review Itinerary.....	20
View Itinerary.....	21
Edit Itinerary.....	22
Post Itinerary.....	23
Create Itinerary.....	24
Browse Itinerary.....	25
Dialog Map.....	26

Use Case Diagram

Use Case Diagram



Use Case Description

Use Case ID:	001		
Use Case Name:	Authenticate user		
Created By:	Eyan Chua	Last Updated By:	Eyan Chua
Date Created:	01/02/2024	Date Last Updated:	01/02/2024

Actor:	Users of the website
Description:	All users must be able to log into their account before they are able to plan or post itineraries. This is to provide legitimacy for the itineraries uploaded and protect the quality of posted itineraries.
Precondition:	User has not been authenticated.
Postcondition:	User has successfully signed in or signed up.
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none">1. Users can choose to sign in if they have already created an account, or sign up if they are a first time user.2. If the user chooses to sign up, the app requests for a valid email and password.3. If the user chooses to sign in, the app requests for the email and password that was previously used to create the account.4. If the sign up credentials are valid, the system creates a new account for the user.5. If the sign in credentials are correct, the user is able to access their account.
Alternative Flows:	<p>AF - S4 If the sign up credentials are not valid, the app will display an error message and the app will return to step 2.</p> <p>AF - S5 If the sign in information is not found, the app will display an error message and the app will return to step 3.</p>
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and issues:	-

Use Case ID:	002		
Use Case Name:	Create and plan travel itinerary		
Created By:	Ang Yvette	Last Updated By:	Ang Yvette
Date Created:	01/02/2024	Date Last Updated:	01/02/2024

Actor:	User
Description:	User can input addresses of different attractions or activities he/she would like to visit. The website would then suggest an itinerary using the shortest path using Google Maps route API.
Precondition:	User has the website opened and an account created.
Postcondition:	User receives a suggested itinerary with estimated time required.
Priority:	Medium
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User selects 'Plan travel' option 2. User inputs the locations they would like to travel to. 3. If more then one attraction chosen, user chooses one attraction to be their starting point. 4. The app uses the Google Maps API to calculate and display an end-to-end route from the user's location to the date idea's location and other included locations. 5. The app estimates the time required for the planned date. 6. User selects as many categories they consider the route to be related to. 7. User can choose to discard, save or share the planned route.
Alternative Flows:	<p>AF - S3</p> <p>If the Google Maps API is unable to provide location for the User input location, an error message will be displayed. App returns to Step 3.</p>
Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. Google Maps Routes API to plan routes. 2. Google Maps Places API to get valid businesses.
Special Requirements:	-
Assumptions:	-
Notes and issues:	Time estimation will be based on expected duration at the intended time of travel. The time provided would be an estimation as it does

	not take into account how much time the user spends at each location.
--	---

Use Case ID:	003		
Use Case Name:	Get routes		
Created By:	Tan Hui Ling	Last Updated By:	Tan Hui Ling
Date Created:	01/02/2024	Date Last Updated:	01/02/2024

Actor:	Google Maps
Description:	Retrieve the optimal route between listed attractions in the travel itinerary on an integrated map feature. The optimal route shall start at the selected starting attraction and provide the shortest possible travel time between each attraction. This will help users save time on travelling and hence more effectively plan their day.
Precondition:	User has inputted a minimum of 2 attractions into their travel itinerary.
Postcondition:	System successfully displays the optimal route on the integrated map feature to the user.
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User inputs the addresses of the attractions they wish to visit through text input. 2. For every combination of 2 attractions, the system will input the addresses to Google Maps Routes API. 3. If Google Maps Routes API can find the imputed addresses and a valid route, it returns the details of route to the system. 4. The system will extract and store the travel time information for each combination of attractions. 5. The system will provide the shortest possible travel combination between all attractions, given the selected starting attraction. 6. The system will input the addresses of the attractions in order of the shortest travel combination to Google Maps Routes API. 7. If Google Maps Routes API can find the imputed addresses and a valid route, it returns the details of route to the system. 8. The system displays the route returned by Google Maps Routes API on the Integrated Map.
Alternative Flows:	<p>AF - S3</p> <p>If Google Maps Routes API cannot find a valid address for an attraction, the system will return an error message and prompt the user to re-enter the address. The system will return to step 2.</p>

	AF - S7 If Google Maps Routes API cannot find a valid route between attractions, the system will return an error message and prompt the user to re-enter all attractions. The system will return to step 2.
Exceptions:	-
Includes:	-
Special Requirements:	Google Maps Location API
Assumptions:	-
Notes and issues:	-

Use Case ID:	004		
Use Case Name:	Post travel itinerary		
Created By:	Tan Hui Ling	Last Updated By:	Tan Hui Ling
Date Created:	01/02/2024	Date Last Updated:	01/02/2024

Actor:	User
Description:	Users can upload their travel itineraries onto the website for other users to view and interact with. The travel itinerary includes information such as the rating of the route, addresses of attractions included in the route, and descriptions of the route and the attractions. This allows users to plan out their day with more confidence and efficiency.
Precondition:	User is logged into the system
Postcondition:	Travel itinerary is successfully uploaded onto the website
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User selects "Create new travel itinerary" on the homepage of the website. 2. User selects and adds the attractions they wish to visit or have visited into the itinerary. 3. The user selects to upload the travel itinerary. 4. The system processes the information. 5. The system uploads the travel itinerary and adds it to its database 6. The website displays a confirmation message to the user.
Alternative Flows:	AF - S4 If the system is unsuccessful in uploading the travel itinerary, the website will return an error message and prompt the user to try again. The system returns to step 2.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and issues:	-

Use Case ID:	005		
Use Case Name:	Edit travel itinerary		
Created By:	Eyan Chua	Last Updated By:	Eyan Chua
Date Created:	01/02/2024	Date Last Updated:	01/02/2024

Actor:	Users
Description:	User can edit unuploaded itineraries.
Precondition:	User is successfully logged in and has previously uploaded itineraries.
Postcondition:	Itinerary is successfully edited.
Priority:	Medium
Frequency of Use:	Seldom
Flow of Events:	<ol style="list-style-type: none"> 1. User selects an itinerary that they have created and saved but not yet uploaded. 2. User now has the option to edit the itinerary. 3. The user can choose to make amendments to the itinerary through adding attractions, deleting attractions, or editing the timing of attractions. 4. The app processes the edit request. 5. The app displays a success message if the request has been processed.
Alternative Flows:	AF - S4 If the app is unable to process the request, an error message will be displayed and the app returns back to step 3.
Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. Authentication of user as a user can only edit their own itineraries.
Special Requirements:	-
Assumptions:	-
Notes and issues:	-

Use Case ID:	006		
Use Case Name:	Delete travel itinerary		
Created By:	Eyan Chua	Last Updated By:	Eyan Chua
Date Created:	01/02/2024	Date Last Updated:	01/02/2024

Actor:	Users
Description:	User can delete their uploaded itineraries.
Precondition:	User is successfully logged in and has previously uploaded itineraries.
Postcondition:	Itinerary is successfully edited or deleted.
Priority:	Medium
Frequency of Use:	Seldom
Flow of Events:	<ol style="list-style-type: none"> 1. User selects an itinerary that they have previously uploaded. 2. User now has the option to delete the itinerary. 3. The app sends a confirmation message to delete the itinerary. 4. The app then processes the delete or edit request. 5. The app displays a success message if the request has been processed.
Alternative Flows:	AF - S2 If the app is unable to process the request, an error message will be displayed and the app returns back to step 2.
Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. Authentication of user as a user can only edit/delete their own itineraries.
Special Requirements:	-
Assumptions:	-
Notes and issues:	-

Use Case ID:	007		
Use Case Name:	Browse uploaded travel itineraries		
Created By:	Ang Yvette	Last Updated By:	Ang Yvette
Date Created:	01/02/2024	Date Last Updated:	01/02/2024

Actor:	User
Description:	User can browse travel itineraries uploaded by others. User can filter itineraries according to preset categories (attractions, nightlife, food, sightseeing, chill). User can click into each itinerary to view locations, date, duration and more information provided.
Precondition:	User has a registered account.
Postcondition:	User views a list of itineraries based on applied categories and views further information on the interested travel itinerary.
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User selects the "Browse Travel Ideas" option. 2. User applies categories (e.g., attractions, nightlife, food, sightseeing) to narrow down the list of itinerary ideas. 3. The app retrieves and displays itinerary ideas based on the applied categories. 4. User can click on an itinerary idea to view more information. 5. User views a map feature with locations for that itinerary and a list of information such as price and activities to be done.
Alternative Flows:	AF - S2 If no itinerary ideas match the applied filters, the app informs the user accordingly. App returns to Step 2.
Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. Get shared itinerary from other users
Special Requirements:	-
Assumptions:	-
Notes and issues:	-

Use Case ID:	008		
Use Case Name:	Delete travel itinerary		
Created By:	Eyan Chua	Last Updated By:	Eyan Chua
Date Created:	01/02/2024	Date Last Updated:	01/02/2024

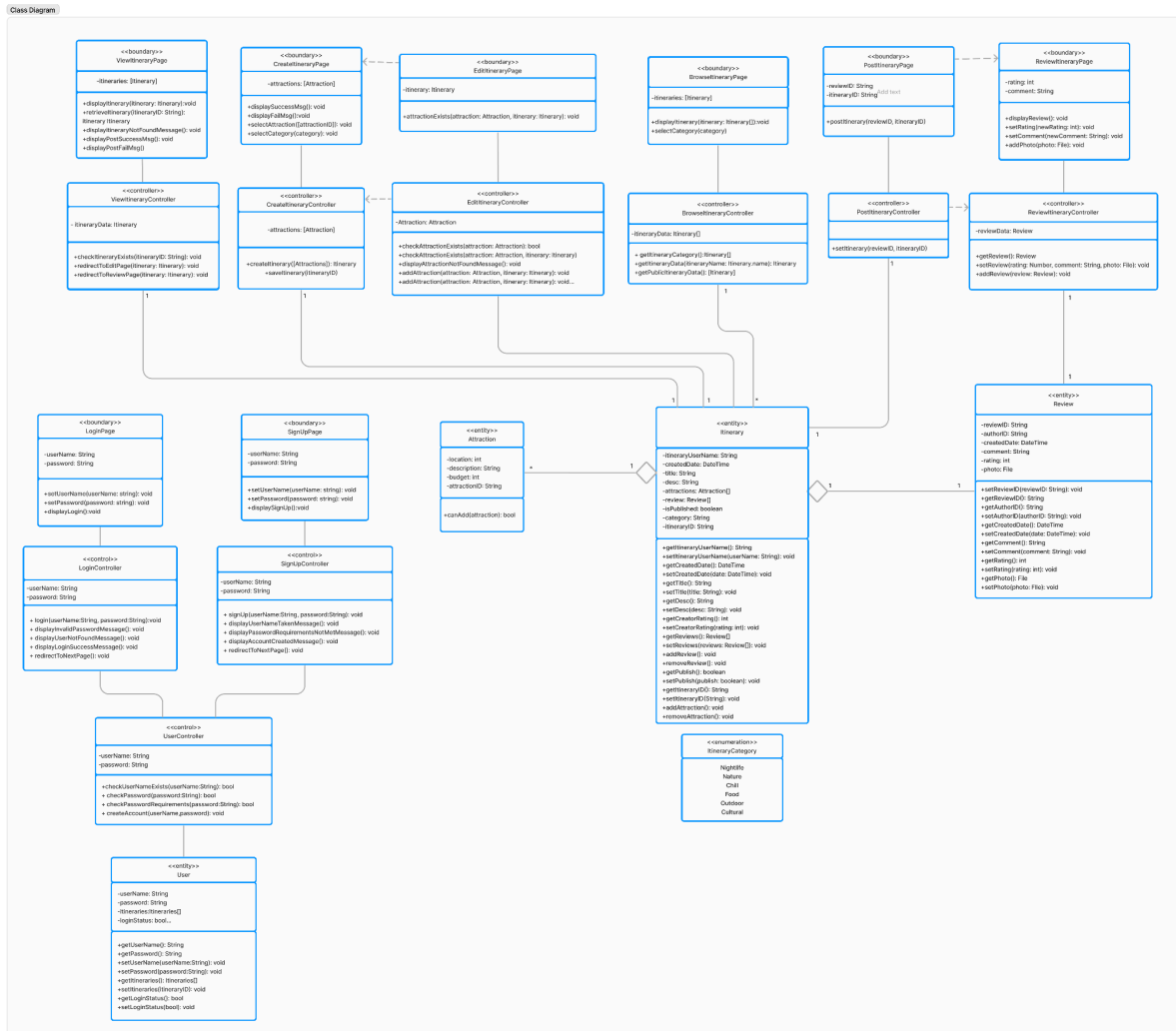
Actor:	Users
Description:	User can delete their uploaded itineraries.
Precondition:	User is successfully logged in and has previously uploaded itineraries.
Postcondition:	Itinerary is successfully edited or deleted.
Priority:	Medium
Frequency of Use:	Seldom
Flow of Events:	<ol style="list-style-type: none"> 1. User selects an itinerary that they have previously uploaded. 2. User has the option to delete the itinerary. 3. The app sends a confirmation message to delete the itinerary. 4. The app then processes the delete or edit request. 5. The app displays a success message if the request has been processed.
Alternative Flows:	AF - S2 If the app is unable to process the request, an error message will be displayed and the app returns back to step 2.
Exceptions:	-
Includes:	<ol style="list-style-type: none"> 1. Authentication of user as a user can only edit/delete their own itineraries.
Special Requirements:	-
Assumptions:	-
Notes and issues:	-

Use Case ID:	009		
Use Case Name:	Review travel itinerary		
Created By:	Eyan Chua	Last Updated By:	Eyan Chua
Date Created:	13/02/2024	Date Last Updated:	13/02/2024

Actor:	Users
Description:	User can review others' itineraries.
Precondition:	User is successfully logged in.
Postcondition:	Review is successfully posted and tagged to the itinerary.
Priority:	Medium
Frequency of Use:	Seldom
Flow of Events:	<ol style="list-style-type: none"> 1. User selects an itinerary that they see when browsing itineraries. 2. User now has the option to upload a text review, rating or photo. 3. If the user chooses to upload a review, they will be prompted to input text. 4. If the user chooses to upload a rating, they will be prompted to rate the itinerary out of 5, with 5 being the best and 1 being the worst. 5. If the user chooses to upload a photo, they will be prompted to upload a png or jpg file. 6. The app processes the review data and stores it in the database under the itinerary. 7. User receives a success message if the review has been posted.
Alternative Flows:	AF - S2 If the app is unable to process the review (e.g. wrong file type), an error message will be displayed and the app returns back to step 2.
Exceptions:	-
Includes:	Authentication of users as only registered users can upload reviews.
Special Requirements:	-
Assumptions:	-
Notes and issues:	-

Conceptual Model

Class Diagram



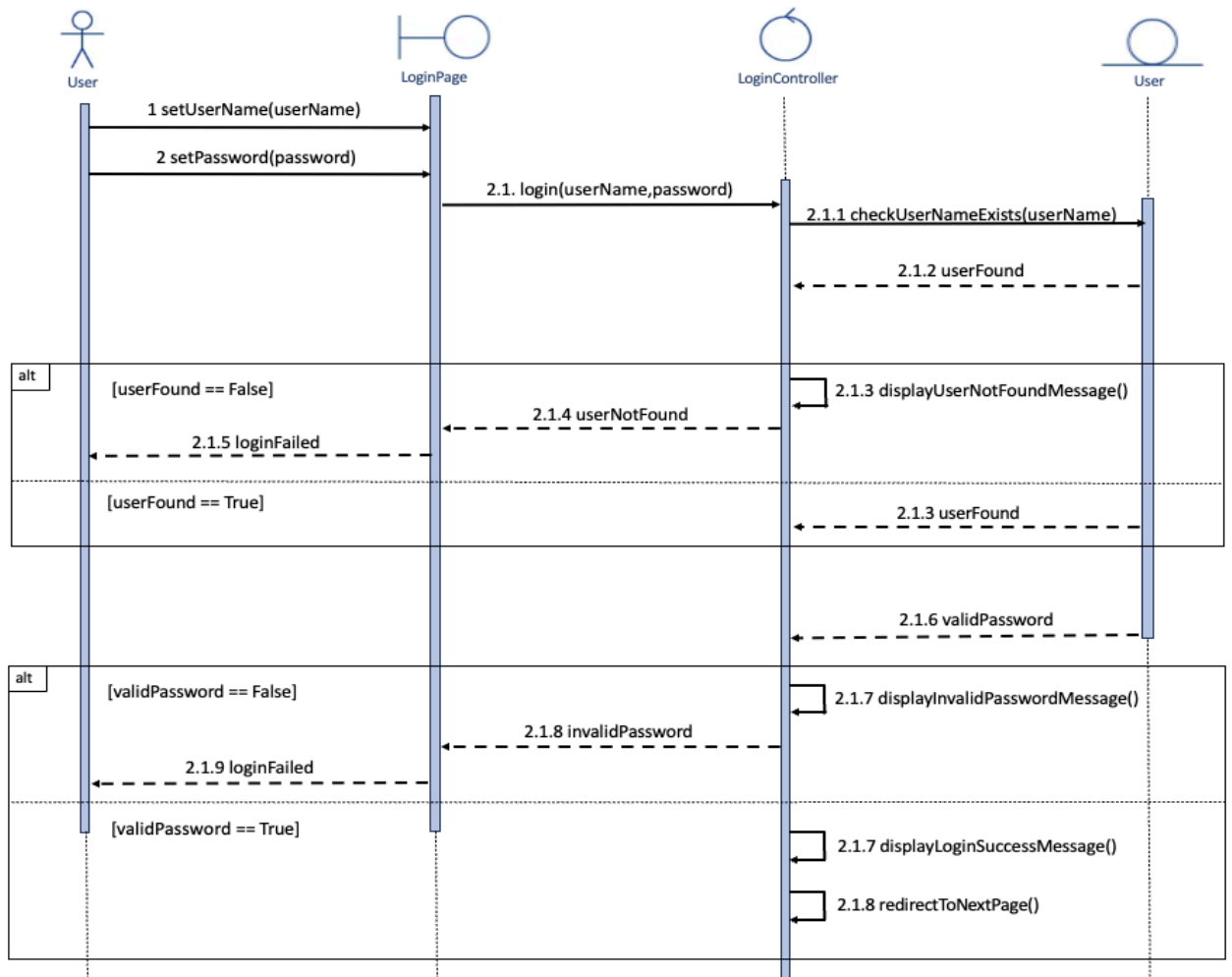
Dynamic Model

Key Boundary Classes and Control Classes

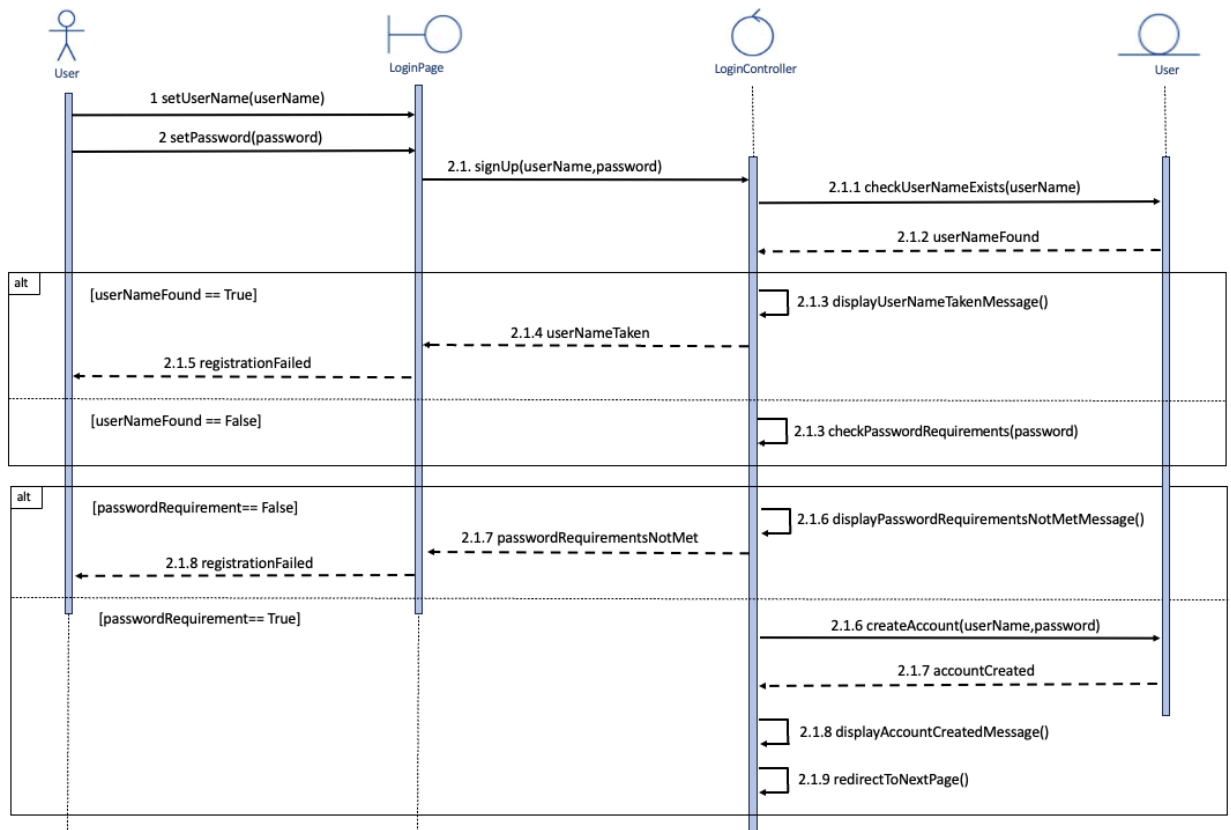
Boundary Classes	Control Classes
ViewItineraryPage	ViewItineraryController
CreateItineraryPage	CreateItineraryController
EditItineraryPage	EditItineraryController
BrowseItineraryPage	BrowseItineraryController
PostItineraryPage	PostItineraryController
ReviewItineraryPage	ReviewItineraryController
LoginPage	LoginController
SignUpPage	SignUpController
	UserController

Sequence Diagram

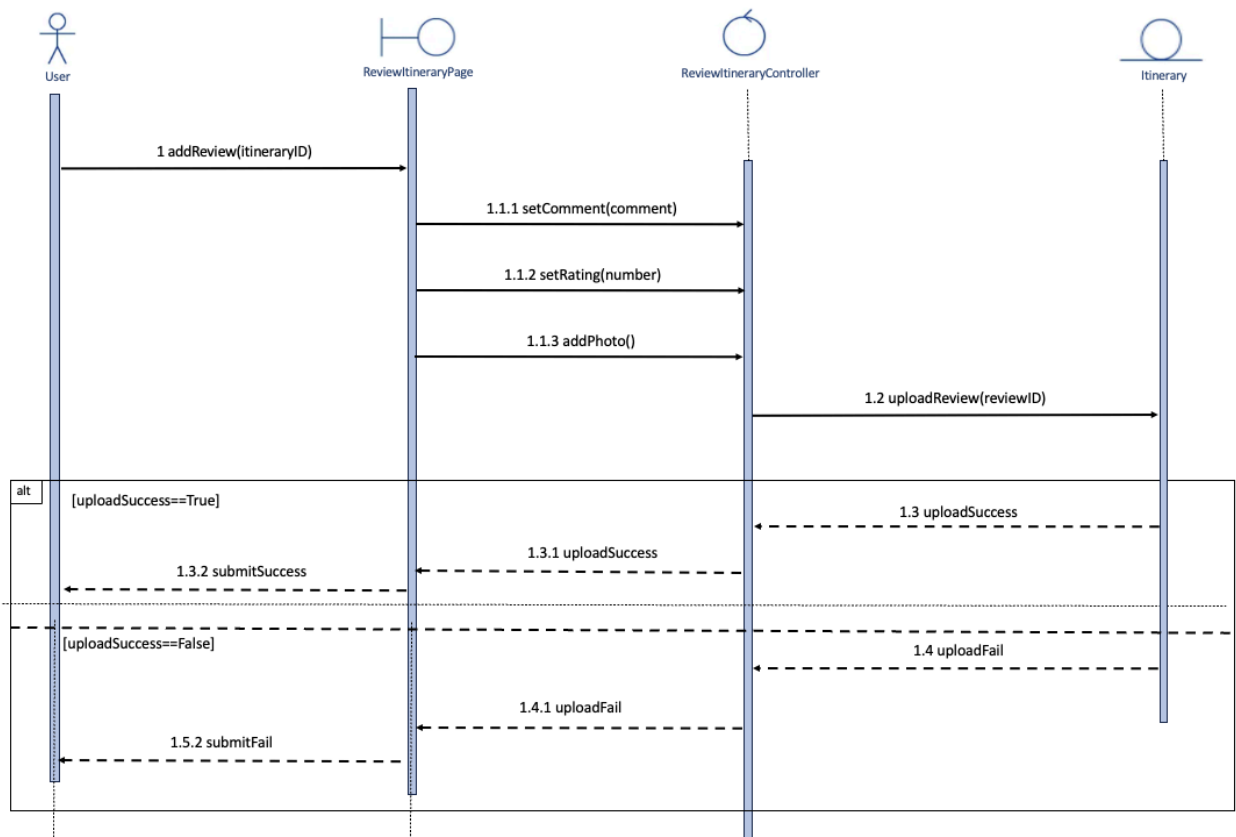
Login



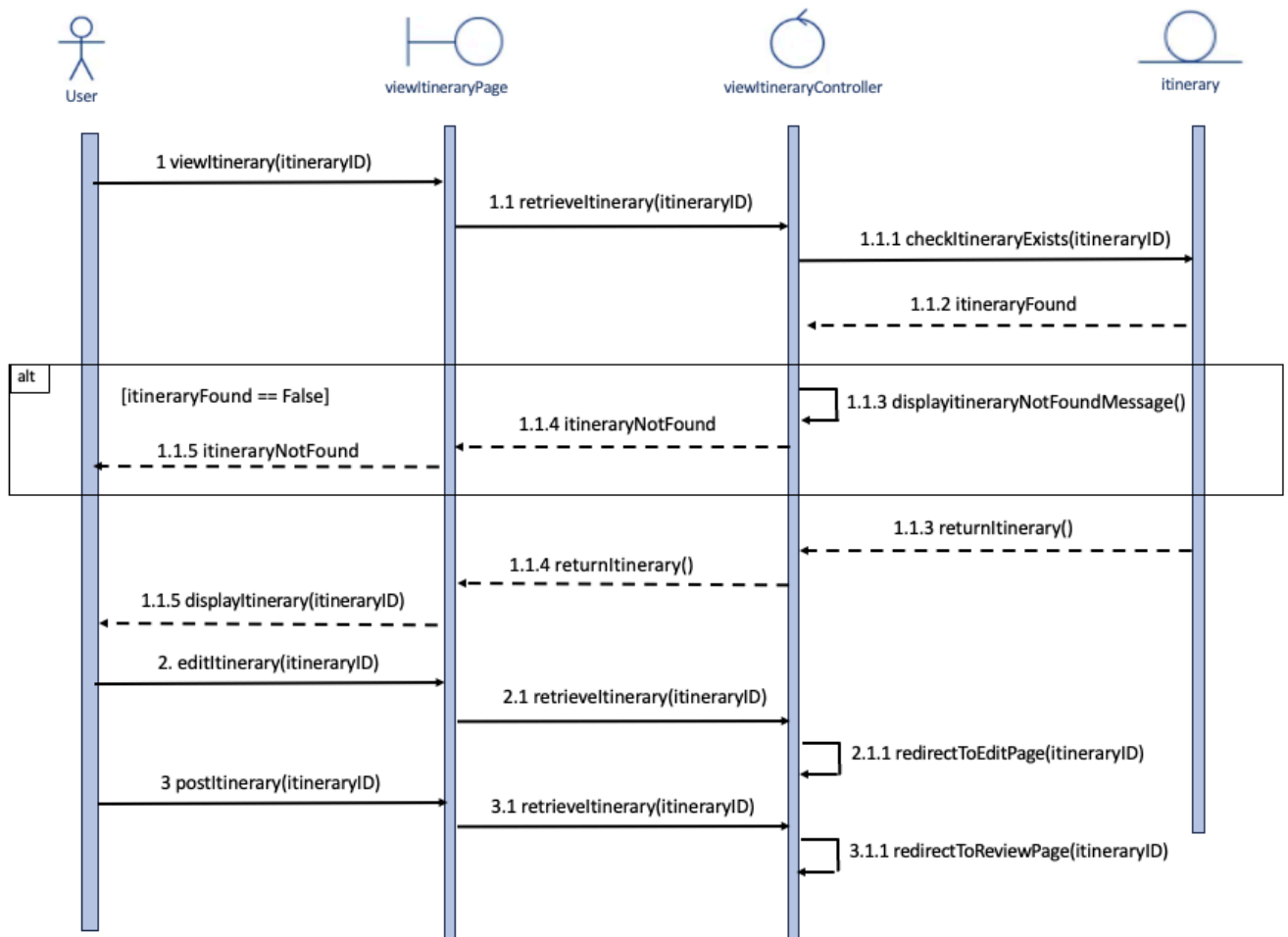
Sign up



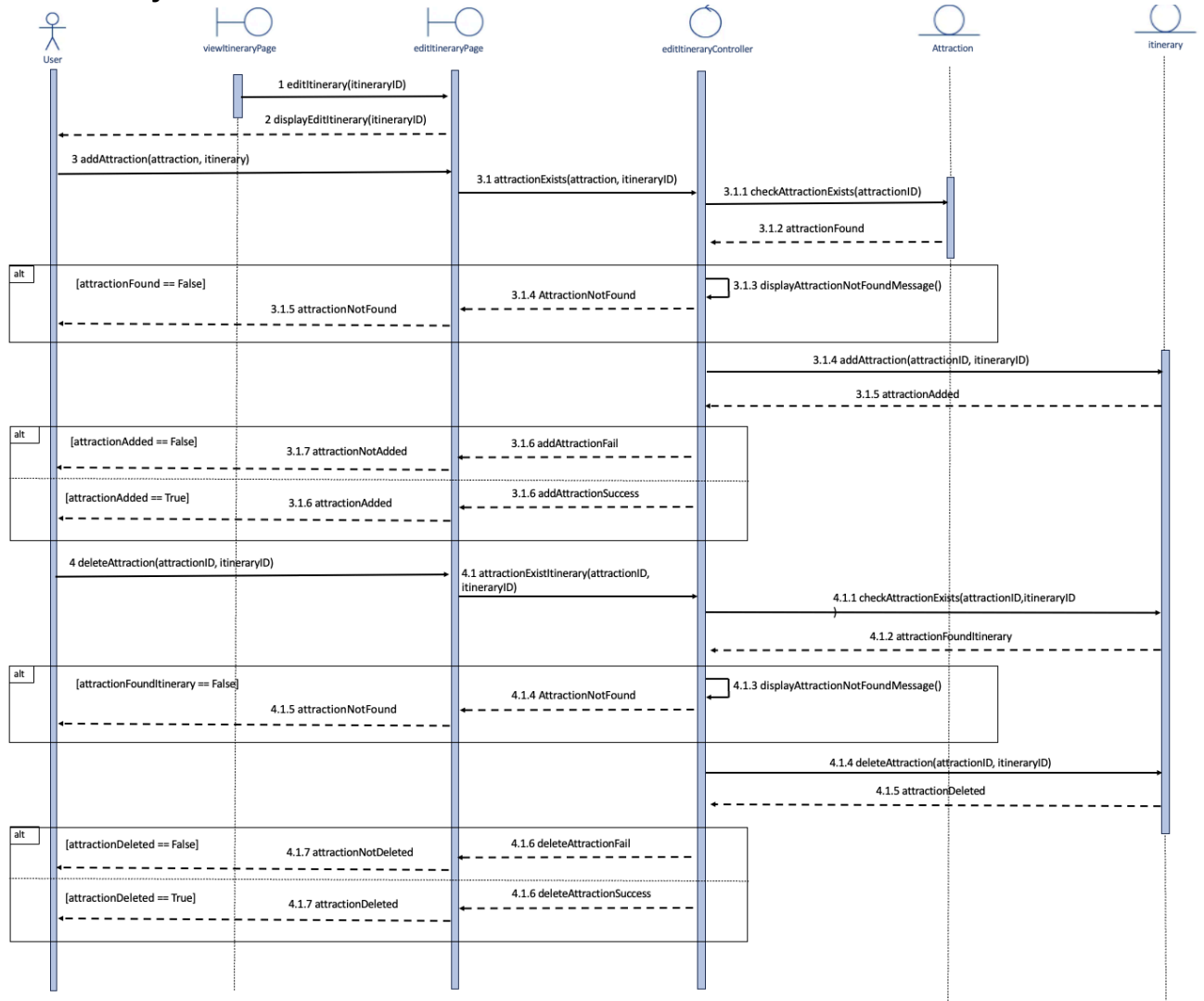
Review Itinerary



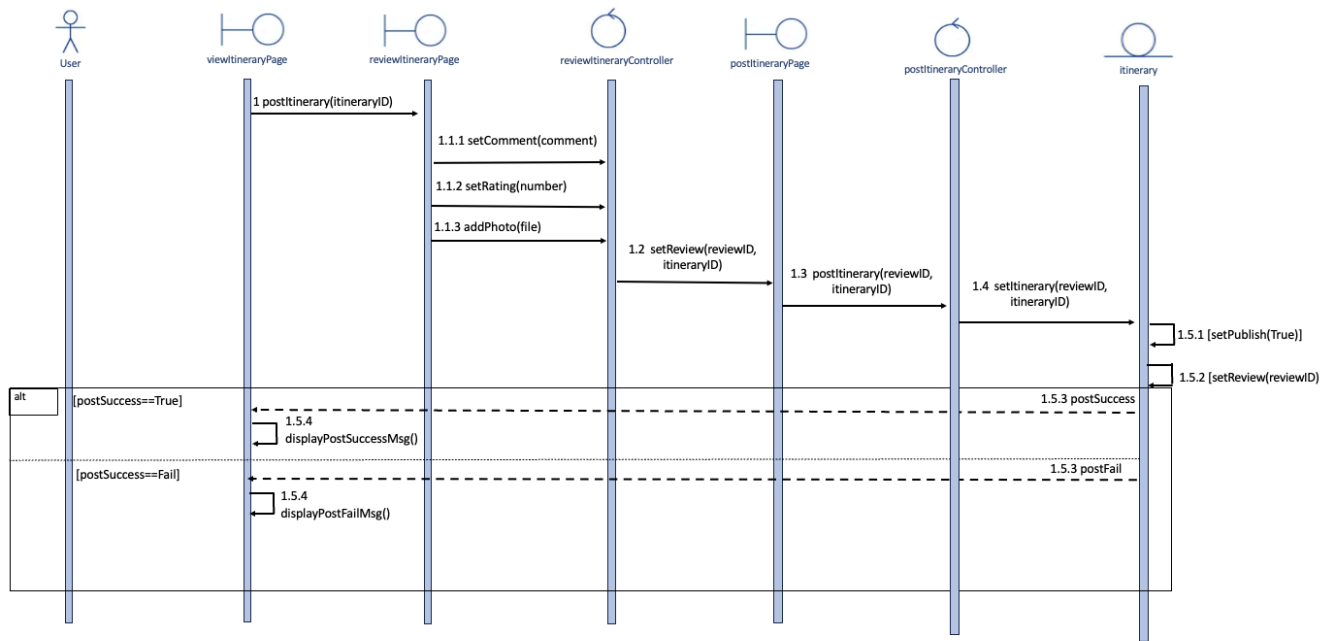
View Itinerary



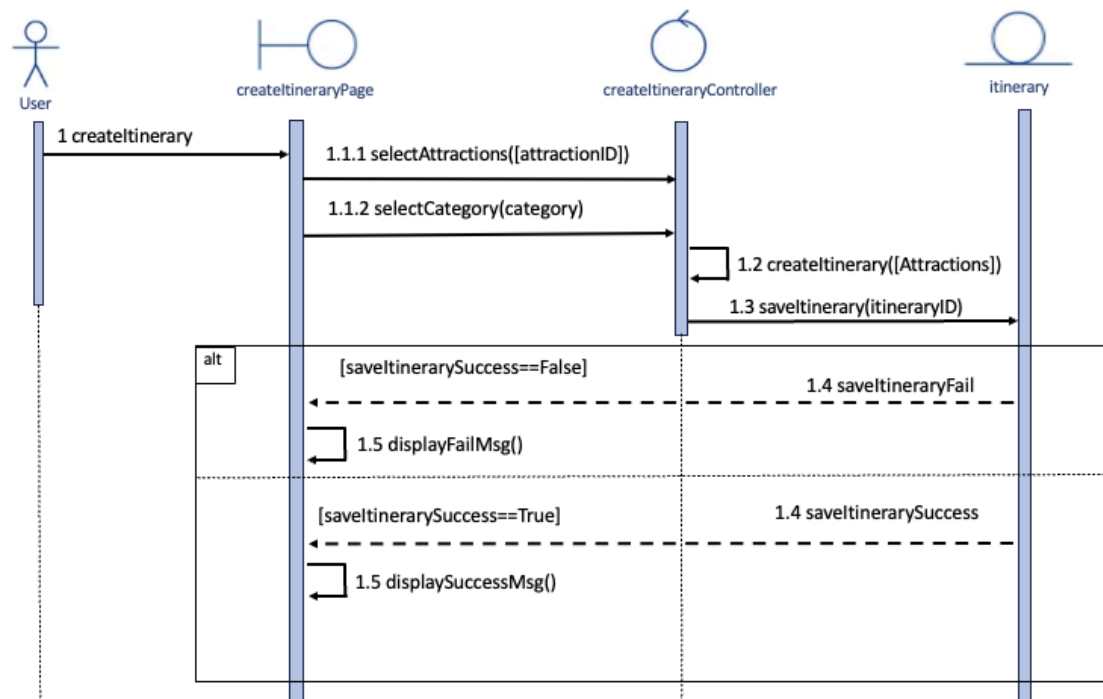
Edit Itinerary



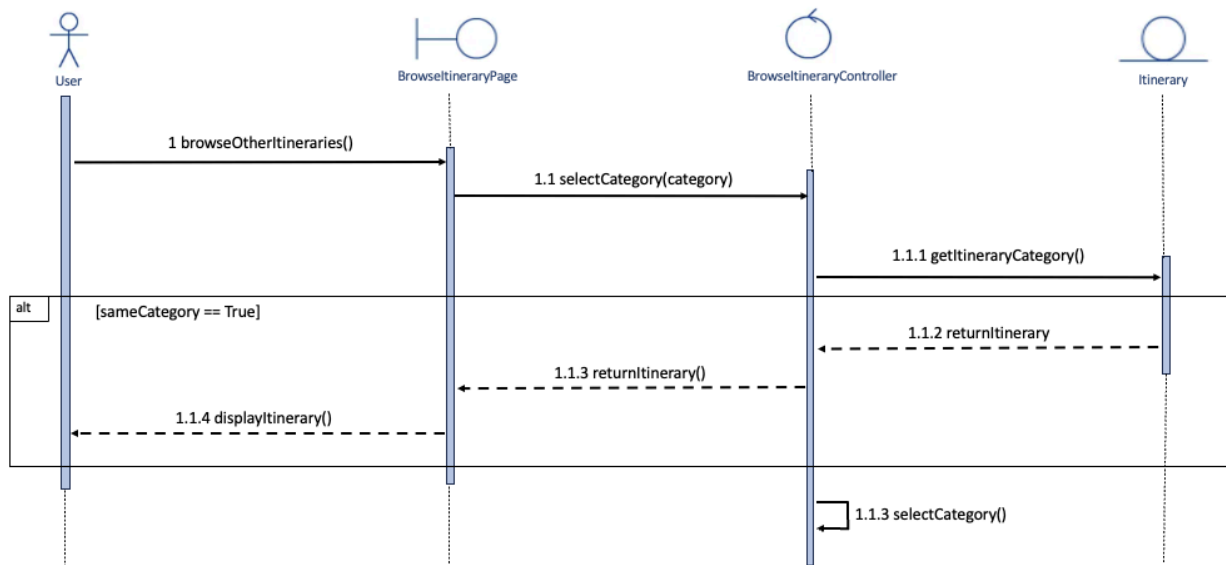
Post Itinerary



Create Itinerary



Browse Itinerary



Dialog Map

