Use Cases

for

MyTripDiary

Version Alpha 3 approved

Prepared by
Tai Chen An
Tan Ming Rui, Ezra
Goel Armaan
Sim Guanyu
Xu Yinfeng
Nepal Aaradh

Nanyang Technological University Team No Idea

23 Mar 2023

Revision History

Name	Date	Reason For Changes	Version
Goel Armaan	15/02/2023	Edit use cases VT01, ET02	Alpha 2
Goel Armaan	23/03/2023	Fill in missing details	Alpha 3

Use Cases

Use Case ID:	AT01		
Use Case Name:	Add Trip		
Created By:	Tan Ming Rui, Ezra	Last Updated By:	Tan Ming Rui, Ezra
Date Created:	31/01/23	Date Last Updated:	31/01/23

Actor:	User
Description:	Adds a trip, which is defined by the trip's start point and end point. Each trip also has a name.
Preconditions:	1. The user must be logged in.
Postconditions:	1. The user is able to Execute trip
Priority:	High
Frequency of Use:	Low
Flow of Events:	1. User fills in the name, start point, and end point of the trip they wish to create. 2. User clicks "Save".
Alternative Flows:	-
Exceptions:	EX-S2: User clicks "Cancel" 1. System returns to homepage
Includes:	GP01
Special Requirements:	The system must be able to synchronize data to the online database within 3 seconds.
Assumptions:	1. The connection of the app and the Database is established
Notes and Issues:	-

Use Case ID:	RR01		
Use Case Name:	Register		
Created By:	Tan Ming Rui, Ezra	Last Updated By:	Tan Ming Rui, Ezra
Date Created:	31/01/23	Date Last Updated:	31/01/23

Actor:	User
Description:	The user registers an account with their email address and password.
Preconditions:	1. The User's device must be connected to WiFi/Cellular Data. 2. The email address given is not registered in the System.
Postconditions:	1. The registered user account is stored in the database. 2. The User can login to the System using the registered account.
Priority:	High
Frequency of Use:	Once per User
Flow of Events:	1. The System requests the User to input the following information fields: • Email address • Password • Confirm Password 2. The System validates the required fields. 3. User clicks on the Register button 4. The System sends an email with a confirmation link to the registered email address. 5. If the User clicks on the confirmation link, the user account is successfully created. 6. Return to User Login menu after confirmation of account creation.
Alternative Flows:	AF-S4: Text fields are not filled in appropriately 1. The System will prompt the User to fill in the required fields. 2. The System returns to Step 1. AF-S2: Email already exists in the database 1. The System will prompt an error message "Email address is already in use" AF-S2: Username already exists in the database 1. The System will prompt an error message "Username is already in use"
Exceptions:	-
Includes:	-

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

Use Case ID:	LL01		
Use Case Name:	Login		
Created By:	Tan Ming Rui, Ezra	Last Updated By:	Tan Ming Rui, Ezra
Date Created:	31/01/23	Date Last Updated:	31/01/23

Actor:	User
Description:	Login to System through verification of User's account email and password.
Preconditions:	1. The User's device must be connected to WiFi/Cellular Data. 2. The System currently does not have a User logged in. 3. User has an existing and verified account associated with the email stored in the database. 4. The System must be able to communicate with the Database. 5. The hosted Database must be online.
Postconditions:	-
Priority:	High
Frequency of Use:	Low
Flow of Events:	1. The User inputs the email address and password 2. The User clicks the "Login" button. 3. The System verifies the login credentials. 4. If the login credentials are verified, the User successfully logs in. 5. The User can proceed to use the System.
Alternative Flows:	AF-S4: If the login credentials are invalid 1. The System informs the User the login has failed. 2. The System returns to Step 1.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	DT01		
Use Case Name:	Delete Trip		
Created By:	Nepal Aaradh	Last Updated By:	Nepal Aaradh
Date Created:	01/02/2023	Date Last Updated:	20/03/2023

Actor:	User
Description:	Delete a trip that has been saved
Preconditions:	1. The user must be logged in 2. The trip being requested to be deleted must exit
Postconditions:	The trip is no longer existent
Priority:	High
Frequency of Use:	Low
Flow of Events:	1. The user selects a trip to delete from a list of previously saved trips 2. The user confirms that the deletion should take place
Alternative Flows:	1. The user does not confirm the deletion and instead decides to cancel the delete request
Exceptions:	-
Includes:	-
Special Requirements:	The system must be able to synchronize data to the online database within 3 seconds.
Assumptions:	1. The connection of the app and the Database is established
Notes and Issues:	-

Use Case ID:	ET01		
Use Case Name:	Edit Trip		
Created By:	Nepal Aaradh	Last Updated By:	Nepal Aaradh
Date Created:	01/02/2023	Date Last Updated:	20/03/2023

Actor:	User
Description:	Edit the name of a previously saved trip
Preconditions:	1. The user must be logged in 2. The trip being requested to be edited must exit
Postconditions:	The trip name has been updated
Priority:	High
Frequency of Use:	Low
Flow of Events:	1. The user selects a trip to edit from a list of previously saved trips 2. The user enters the new name for the trip with which to replace the current trip name 3. The user confirms the change
Alternative Flows:	-
Exceptions:	EX1: If user cancels while editing 1. System returns to “Manage trips” screen
Includes:	-
Special Requirements:	The system must be able to synchronize data to the online database within 3 seconds.
Assumptions:	1. The connection of the app and the Database is established
Notes and Issues:	-

Use Case ID:	ST01		
Use Case Name:	Star Trip		
Created By:	Nepal Aaradh	Last Updated By:	Nepal Aaradh
Date Created:	01/02/2023	Date Last Updated:	20/03/2023

Actor:	User
Description:	‘Star’ a trip (indicate a trip as a frequently used trip)
Preconditions:	1. The user must be logged in 2. The trip being requested to be starred must exit
Postconditions:	The trip is pinned to top of the list of previously saved trips making it easily accessible to the user
Priority:	Low
Frequency of Use:	Low
Flow of Events:	1. The user selects a trip to star from a list of previously saved trips
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	The system must be able to synchronize data to the online database within 3 seconds.
Assumptions:	1. The connection of the app and the Database is established
Notes and Issues:	-

Use Case ID:	ET02		
Use Case Name:	Execute Trip		
Created By:	Goel Armaan	Last Updated By:	Goel Armaan
Date Created:	31/01/23	Date Last Updated:	23/03/23

Actor:	User
Description:	Records an instance of a saved trip with the user's selected mode of transport.
Preconditions:	1. User is logged in 2. User has the trip saved
Postconditions:	1. An instance of the saved trip is added to the user's trip history
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User clicks a saved trip 2. User chooses their preferred mode of transport 3. User reviews the estimated price 4. User clicks the execute trip button 5. An instance of the saved trip is recorded in the system
Alternative Flows:	-
Exceptions:	EX1: User cancels execution 1. User can choose not to execute a trip, at which point the use case exits at either step 2 or step 3 EX2: Trip cannot be executed 1. If a chosen mode of transport is unavailable, the user cannot move on to review estimated prices. 2. The user is informed of this limitation and is asked to choose another mode of transportation.
Includes:	GP01
Special Requirements:	The system must be able to synchronize data to the online database within 3 seconds.
Assumptions:	The user executes a trip right before they embark on it
Notes and Issues:	-

Use Case ID:	GR01		
Use Case Name:	Get Route		
Created By:	Goel Armaan	Last Updated By:	Goel Armaan
Date Created:	31/01/23	Date Last Updated:	23/03/23

Actor:	Google Maps API
Description:	Retrieves the route between a trip's starting point and destination point for the chosen mode of transport
Preconditions:	1. User is logged in 2A. User is adding a new trip or 2B. User is executing a saved trip 3. User device has a working WiFi or Cellular connection
Postconditions:	1. The correct route from the trip's starting point to the trip's destination point with the chosen mode of transport is displayed
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User has entered the starting and destination point for a new trip 2. The API is queried and the route for a private car is displayed by default.
Alternative Flows:	AF-S1: User is executing a saved trip 1. If the user is executing a saved trip, they do not have to enter the starting and destination point as required in step 1. AF-S2: User chooses another mode of transport 1. If the user chooses another mode of transport, the API is queried again and a route with the new mode of transport is displayed.
Exceptions:	EX1: No route can be found 1. If there is no route between the start and destination point for the selected mode of transport, a route cannot be displayed. 2. The user is informed of this limitation and is asked to try another mode of transportation EX2: API is unavailable 1. If the Google Maps API is unavailable or cannot be queried, a route cannot be displayed. 2. The user is informed of this limitation and is asked to try again later.
Includes:	GA01

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

Use Case ID:	GP01		
Use Case Name:	Get Price		
Created By:	Goel Armaan	Last Updated By:	Goel Armaan
Date Created:	31/01/23	Date Last Updated:	23/03/23

Actor:	Car Park API, Public Transport API, Taxi API
Description:	Calculates an estimated price of a route with the chosen mode of transportation
Preconditions:	1. User is logged in 2A. User is adding a new trip or 2B. User is executing a saved trip 3. The trip's route is valid 4. User device has a working WiFi or Cellular connection
Postconditions:	1. A well-estimated price of a route with the chosen mode of transportation is displayed
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User selects a valid mode of transportation for a trip 2. The respective API is queried 3. The price for the trip's route with the chosen mode of transportation is displayed
Alternative Flows:	AF-S2: 1. If the mode of transportation is biking or walking, no API is queried. The price is 'free'.
Exceptions:	EX1: API is unavailable 1. If the API used in calculating the price for a chosen mode of transportation is unavailable, then the estimated price cannot be calculated 2. Instead, a message is shown to the user stating the estimated pricing for the chosen mode of transportation is currently unavailable
Includes:	GR01, GP02
Special Requirements:	The system must be able to calculate the fare cost of a public transport route within 3 seconds.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	VS01		
Use Case Name:	View Stats		
Created By:	Xu Yinfeng	Last Updated By:	Xu Yinfeng
Date Created:	01-02-2023	Date Last Updated:	01-02-2023

Actor:	User, Database
Description:	System get analytics data from database and display on the analytics dashboard
Preconditions:	1. User is connected to WiFi/Cellular Data 2. User has logged into their user account 3. User has completed at least 1 trip
Postconditions:	-
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User completes a trip, trip recorded in trip history 2. System functions calculate/store trip information statistics 3. System get trip statistics and display in the analytics dashboard
Alternative Flows:	AF1: 1. If user has not completed any trips, display “No Stats Available”
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	VH01		
Use Case Name:	View Trip History		
Created By:	Xu Yinfeng	Last Updated By:	Xu Yinfeng
Date Created:	01-02-2023	Date Last Updated:	01-02-2023

Actor:	Database, User
Description:	Get user trip history, display user trip history
Preconditions:	1. User is connected to WiFi/Cellular Data 2. User has logged into their user account
Postconditions:	-
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User login to their user account 2. User clicked on "Trip History" side bar 3. System retrieve user trip history from database 4. System display trip history by user’s chosen order
Alternative Flows:	AF1: 1. If user has not completed any trips, display “No trip history”
Exceptions:	-
Includes:	EP01
Special Requirements:	-
Assumptions:	User has logged into their account
Notes and Issues:	-

Use Case ID:	EP01		
Use Case Name:	Edit Past Trip Price		
Created By:	Xu Yinfeng	Last Updated By:	Xu Yinfeng
Date Created:	01-02-2023	Date Last Updated:	01-02-2023

Actor:	User, Database
Description:	User enter/edit trip price information upon completion of a trip
Preconditions:	1. User is connected to WiFi/Cellular Data 2. User has logged into their user account 3. User has completed at least 1 trip
Postconditions:	-
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User completes a trip, trip recorded in trip history 2. User goes into trip history and click “edit” button 3. User changes the value of “trip price” for trip 4. User click “confirm edit” 5. System send post request to update database trip price
Alternative Flows:	AF1: 1. If user has not completed any trips, display “No trip available”
Exceptions:	-
Includes:	-
Special Requirements:	The system must be able to synchronize data to the online database within 3 seconds.
Assumptions:	-
Notes and Issues:	-

Use Case ID:	VT01		
Use Case Name:	View Saved Trips		
Created By:	Nepal Aaradh	Last Updated By:	Nepal Aaradh
Date Created:	02/02/2023	Date Last Updated:	20/03/2023

Actor:	User
Description:	Displays the previously saved trips to the user
Preconditions:	1. The user must be logged in 2. The user must have saved from trips previously
Postconditions:	1. The user can view the previously saved trips
Priority:	-
Frequency of Use:	-
Flow of Events:	1. The user selects to view saved trips 2. The user is displayed previously saved trips
Alternative Flows:	-
Exceptions:	-
Includes:	DT01, ST01, ET01
Special Requirements:	-
Assumptions:	1. The connection of the app and the Database is established
Notes and Issues:	-

Use Case ID:	GA01		
Use Case Name:	Get Carpark Availability		
Created By:	Xu Yinfeng	Last Updated By:	Xu Yinfeng
Date Created:	28/01/2023	Date Last Updated:	28/01/2023

Actor:	Carpark API, System, User
Description:	System gets carpark availability information based on user destination
Preconditions:	1. Users need to be connected to WiFi/Cellular data 2. Users need to enter a valid destination 3. Users need to tick “yes” for travelling by car
Postconditions:	-
Priority:	High
Frequency of Use:	High
Flow of Events:	1. Users select destination and preference ranking 2. Users tick “yes” for travelling by car 3. Upon route search, System should send get request to Carpark API to get carpark availability information
Alternative Flows:	-
Exceptions:	EX1: Invalid destination 1. Display error message and prompt user to re-enter destination EX2: Carpark API Query failure 1. System prompts error message 2. “Retry” button enabled for retrying API query
Includes:	-
Special Requirements:	Car park availability in an area must be loaded within 3 seconds.
Assumptions:	1. Carpark API is online and queryable
Notes and Issues:	-

Use Case ID:	GP02		
Use Case Name:	Get Parking Price		
Created By:	Xu Yinfeng	Last Updated By:	Xu Yinfeng
Date Created:	28/01/2023	Date Last Updated:	28/01/2023

Actor:	Carpark API, System, User
Description:	Get parking rate information of carparks and calculate parking fees based on parking duration and time of the day
Preconditions:	1. Users need to be connected to WiFi/Cellular data 2. Users need to enter a valid destination 3. Users need to tick “yes” for travelling by car
Postconditions:	-
Priority:	High
Frequency of Use:	High
Flow of Events:	1. Users select destination and preference ranking 2. Users tick “yes” for travelling by car 3. Users enter duration of stay and time of arrival 4. System gets carpark rate from Carpark API 5. System calculates total parking fee based on carpark rate, time of day and length of stay
Alternative Flows:	-
Exceptions:	EX1: Invalid destination 1. Display error message and prompt user to re-enter destination EX2: Carpark API Query failure 1. System prompts error message 2. “Retry” button enabled for retrying API query
Includes:	-
Special Requirements:	-
Assumptions:	1. Carpark API is online and queryable
Notes and Issues:	-