Julian Siddiqui

DE211 – Assessment 2

**Use Case 1**

**Case** **Description**: The user adds a trip.

**Actor**: The user

**Pre-Conditions**

1. The app/program is working.
2. Add function is working.

**Main Flow**

1. The user adds a trip.
2. The trip goes into the system input.

**Alternative Flows**

**A1**

1. If the user wants to go certain trips then the user needs to filterTrips.
2. User can also use the search criterion.

**Exception Flows**

**E1**

1. The user finally find a specific Trips that the user wants to go.
2. The use case ends.

**Use Case 2**

**Case** **Description**: User sorts a trip.

**Actor**: The user

**Pre-Conditions**

1. The app/program is working.
2. Sort function is working.

**Main Flow**

1. The user sorts a trip.
2. The user chooses any of the sort options.
3. The system will sort the trips.
4. The app shows the sorted trips. **(A1, A2, A3, A4)**

**Alternative Flows**

**A1**

1. The user chooses the ‘best match’ option.
2. The app sorts from the ‘best match’ trips.

**A2**

1. The user chooses the ‘most convenient’ option.
2. The app sorts from the ’convenient’ ‘.

**A3**

1. The user chooses the ‘lowest’ option.
2. The app sorts from the ‘lowest’.

**Exception Flows**

**E1**

**1.** The user can delete selected Trips put some other trips as desirable.

**2.** The use case ends.

**Use Case 3**

**Case** **Description**: User filters a trip.

**Actor**: The user

**Pre-Conditions**

1. The app/program is working.
2. Filter function is working.

**Main Flow**

1. The user filters a trip.
2. The user chooses any of the filter options.
3. The app shows the filtered trips.

**Alternative Flows**

**A1**

1. The user chooses the ‘Speed’ option.
2. The app filters only the ‘Speed’.

**A2**

1. The user chooses the ‘Calorie Loss’ option.
2. The app filters only the ‘Calorie Loss’.

**A3**

1. The user chooses the ‘Distance’ option.
2. The app filters only the ‘Distance’.

**A4**

1. The user chooses the ‘Routes’ option.
2. The app filter only the ‘Routes’.

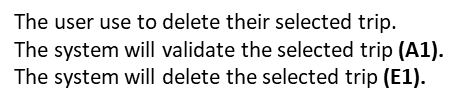
**Use Case 4**

**Case** **Description**: deleteASelectedTrip.

**Actor**: The user

**Pre-Conditions:**

1. The app/program is working.
2. Search function is working.

**Main Flow**

**Alternative Flows**

**A1**

The selection is invalid.

The case goes back to step 1.

**Exception Flows**

**E1**

1. The user successfully deletes a selected trip.
2. The use case ends.

**Use Case 5**

**Case** **Description**: User edits a trip from added trip.

**Actor**: The user

**Pre-Conditions:**

1. The app/program is working.
2. Edit function is working.

**Main Flow**

1. The user selects a trip to edit.
2. The app will show the selected trip.
3. The user edits the selected trip.
4. The system will validate the input. **(A1)**
5. The edited trip gets saved to LocalStorage. **(E1)**

**Alternative Flows**

**A1**

1. The input is invalid.
2. The case goes back to step 1.

**Exception Flows**

**E1**

1. The user successfully edited the selected trip.
2. The use case ends.

**Use Case 6**

**Case** **Description**: User saves a trip to LocalStorage.

**Actor**: The user

**Pre-Conditions:**

1. The app/program is working.
2. Save function is working.

**Main Flow**

1. The user saves a trip to LocalStorage.
2. the system will check the given input**.(A1)**
3. The trip goes into the LocalStorage**.(E1)**

**Alternative Flows**

**A1**

1. The user input a invalid trips
2. The trip goes back to previous stage.

**Exception Flows**

**E1**

1. The user put valid trips.
2. The Use case finish.

**Use Case 7**

**Case** **Description**: User loads a trip from LocalStorage

**Actor**: The user

**Pre-Conditions:**

1. The app/program is working.
2. Load function is working.

**Main Flow**

1. The user loads a trip from LocalStorage.
2. The system will get all data from LocalStorage. **(A1)**
3. The system loads the selected trip. **(E1)**

**Alternative Flows**

**A1**

1. There is no trip to load.
2. The case goes back to case 1.

**Exception Flows**

**E1**

1. The user successfully loaded a trip.
2. The use case ends.

**Use Case 8**

**Case** **Description**: User update trip

**Actor**: The user

**Pre-Conditions:**

1. The app/program is working.
2. Update function is working.

**Main Flow**

1. The user can update their trips.
2. The app shows list of updated Trip.(A1)
3. The updated trips saves in localStorage.(E1)

**Alternative Flows**

**A1**

1. The user can choose updated trips.
2. The app shows other updated trips.

**Exception Flows**

**E1**

1. The user successfully update a trip.
2. The use case ends.

**Use Case 9**

**Case** **Description**: User revert trip

**Actor**: The user

**Pre-Conditions:**

1. The app/program is working.
2. Revert function is working.

**Main Flow**

1. The user can revert their trips.
2. The user can use list of revert possibility Trip back to previous position(A1)
3. The revert trips saves in localStorage.(E1)

**Alternative Flows**

**A1**

1. The user can back to their previous stage .
2. The app shows revert trips.

**Exception Flows**

**E1**

1. The user successfully revert a trip.
2. The use case ends.

**Use Case 10**

**Case** **Description**: User validate trip

**Actor**: The user

**Pre-Conditions:**

1. The app/program is working.
2. Validate function is working.

**Main Flow**

1. The user can validate their trips.
2. The app shows list of validate Trips.(A1)
3. The app shows validate trips. (E1).

**Alternative Flows**

**A1**

1. The user put invalid trips.
2. The app goes to previous stage.

**Exception Flows**

**E1**

1. The user input validate trips.
2. The use case ends.

**Use Case 11**

**Case** **Description**: User calculation withInATrip

**Actor**: The user

**Pre-Conditions:**

1. The app/program is working.
2. WithInATrip function is working.

**Main Flow**

1. The app shows calculation suchas (fastest speed, avg speed).(A1)
2. The app shows validate trips. (E1).

**Alternative Flows**

**A1**

1. The app shows wrong put calculations.
2. The user should hit the reset button to default values.

**Exception Flows**

**E1**

1. The app showing correct calculations of the trip.
2. The use case ends.

**Use Case 12**

**Case Description:** User calculationAcrossManyTrips

**Actor:** The user

**Pre-Conditions:**

1. The app/program is working.

2. acrossManyTrips function is working.

**Main Flow**

1. The app shows calculation of each different trips (fastest speed, avg speed).(A1)

2. The app shows validate trips. (E1).

**Alternative Flows**

**A1**

1. The app shows wrong put calculations.

2. The user should hit the reset button to default values.

**Exception Flows**

**E1**

1. The app showing correct calculations each of different trip.

2. The use case ends.

**Use Case 13**

**Case Description**: User calculation provideDefaultValues

**Actor**: The user

**Pre-Conditions:**

1. The app/program is working.

2. provideDefaultValues function is working.

**Main Flow**

1. The user can hit the default value button to get default values of calculation suchas (fastest speed, avg speed).
2. The app can show the wrong values (A1)
3. The app shows the correct default values trips. (E1).

**Alternative Flows**

**A1**

1. The app shows wrong put calculations.

2. The app can go back to their previous stage of default values.

**Exception Flows**

**E1**

1. The app showing correct default values of the trip.

2. The use case ends.

**Use Case 14**

**Case** **Description**: User trip given a search criterion

**Actor**: The user

**Pre-Conditions:**

1. The app/program is working.
2. search criterion function is working.

**Main Flow**

1. The user can search their trips using search input bar
2. The app can check trips of the given search input.(A1)
3. The app search the correct Trips(E1).

**Alternative Flows**

**A1**

1. The User put invalid search criterion.
2. The app goes back to their previous stage.

**Exception Flows**

**E1**

1. The app showing correct trips with the correct search criterion.
2. The use case ends.

**Use Case 15**

**Case** **Description**: User Get all trips

**Actor**: The user

**Pre-Conditions:**

1. The app/program is working.
2. Get all trips function is working.

**Main Flow**

1. The user can get all their trips history from the day the user started to use this app
2. The app can check all user trips .(A1)
3. The app shows all correct Trips(E1).

**Alternative Flows**

**A1**

1. The app can shows wrong user trips.
2. The app goes back to their previous stage.

**Exception Flows**

**E1**

1. The app showing correct all user trips.
2. The use case ends.