|  |  |  |
| --- | --- | --- |
| **Use case Id:** | Plan Trip | |
| **Brief Description** | This use case should enable the use to generate a trip from a source to the desired destination, and the generated trip should be based on the real time data. | |
| **Primary actors** | Mobile Application User, OTP server | |
| **Trigger(s)** | The user plans a trip | |
| **Preconditions:**   1. The user has an origin and destination points. 2. The user is sharing his geolocation with the app. | | |
| **Post-conditions:**   1. Trip is generated successfully. | | |
| **Normal Scenario** | | |
| Actor Action | | System Response |
|  | | 1.The mobile application send its current geolocation information to the backend server. |
| 2. The user set origin, destination, max walking distance, date and time | | 3. request itineraries from OTP server |
| 4. OTP server send the itineraries response | |  |
| 5. The user selects an itinerary. | |  |
|  | | 6. <<include: Get station crowdedness >> |
|  | | 7. if itinerary was chosen << extend: Display itinerary on map >> on that itinerary |
|  | |  |
|  | |  |
| **Alternative flows:** | | |

|  |  |  |
| --- | --- | --- |
| **Use case Id:** | Get station camera feed | |
| **Brief Description** | Get the camera feed for a particular station. | |
| **Primary actors** | Station Crowdness Camera | |
| **Trigger(s)** | Timed every 30 seconds | |
| **Preconditions:**   1. The camera must be connected to the internet by a hardware | | |
| **Post-conditions:**   1. Station information is stored in the system DB. | | |
| **Normal Scenario** | | |
| Actor Action | | System Response |
| 1. collect footage from station | |  |
| 2. send footage | | 3. generate density map from the footage |
|  | | 4. calculate the number of crowed in the station |
|  | | 5. estimate the avg waiting time |
|  | | 6. store data in the DB to be accessed later by the user |
|  | |  |
|  | |  |
| **Alternative flows:** | | |

|  |  |  |
| --- | --- | --- |
| **Use case Id:** | Calculate bus Realtime location | |
| **Brief Description** | Estimate the Realtime bus location based on the GPS feed from the crowed. | |
| **Primary actors** | Mobile application users | |
| **Trigger(s)** | Timed every 30 seconds | |
| **Preconditions:**   1. The user is sharing his geo information 2. The user is connected to the internet | | |
| **Post-conditions:**   1. User receives live bus location information | | |
| **Normal Scenario** | | |
| Actor Action | | System Response |
| 1. send stream of geo information | | 2. filter the geo information data |
|  | | 3. cluster the data to detect buses in the system |
|  | | 4. track the discovered cluster through its movement in the system |
|  | | 5. identify the bus route id. |
|  | | 6. store the live information in the DB to be accessed later on. |
|  | |  |
|  | |  |
| **Alternative flows:** | | |

|  |  |  |
| --- | --- | --- |
| **Use case Id:** | Get station crowdedness | |
| **Brief Description** | Get the crowdedness data of a station. | |
| **Primary actors** | Mobile application users | |
| **Trigger(s)** | User planes a trip | |
| **Preconditions:**   1. Station crowdedness information is existence for the required station | | |
| **Post-conditions:**   1. User receives station crowdedness information | | |
| **Normal Scenario** | | |
| Actor Action | | System Response |
| 1. plan a trip | | 2. get the sequence of stations in the trip |
|  | | 3. query the DB for the station crowdedness information |
|  | | 4. update the map with the crowdedness information |
|  | |  |
| **Alternative flows:** | | |

|  |  |  |
| --- | --- | --- |
| **Use case Id:** | Get bus location | |
| **Brief Description** | Get and visualize in a map the current bus location for a selected bus service. | |
| **Primary actors** | Mobile application users | |
| **Trigger(s)** | User planes a trip | |
| **Preconditions:**   1. Information about bus exist | | |
| **Post-conditions:**   1. User can see the live bus location | | |
| **Normal Scenario** | | |
| Actor Action | | System Response |
| 1. plan a trip | | 2. get the bus location of buses inside the trip |
|  | | 3. query the DB for the live bus location |
|  | | 4. update the map with the live bus location |
|  | |  |
| **Alternative flows:** | | |

|  |  |  |
| --- | --- | --- |
| **Use case Id:** | Display itinerary on map | |
| **Brief Description** | Display the details of an itinerary on a map. | |
| **Primary actors** | Mobile application users | |
| **Trigger(s)** | User planes a trip | |
| **Preconditions:** | | |
| **Post-conditions:** | | |
| **Normal Scenario** | | |
| Actor Action | | System Response |
| 1. Plan a trip | | 1. OTP server Sends itineraries information to mobile application. |
| 1. Select an itinerary. | | 1. Send origin, destination and bus stops coordinates to Google Directions. |
|  | | 1. Receive the exact route coordinates for the selected itinerary. |
|  | | 1. Draw the coordinates of the itineraries on the map. |
|  | | 1. <<include: Get bus location >> |
| **Alternative flows:** | | |