CT-Fastrack – System Requirements

Team: Scrum Masters

- **Functional Overview of System:**

The functional requirements for the system are divided into four main categories: Trip Planning, Route Schedules, Next Departures and Alerts, and Online Bus Ticketing and Pass System.

Trip Planning:

- The System shall plan trip for user based on Current location, popular landmarks and specific address.
- The System shall generate information including actual walking distance to bus stop, expected travel time to destination.

Route Schedules:

- The System shall generate point to point schedule within a route.
- The System shall generate maps by filtering route option.

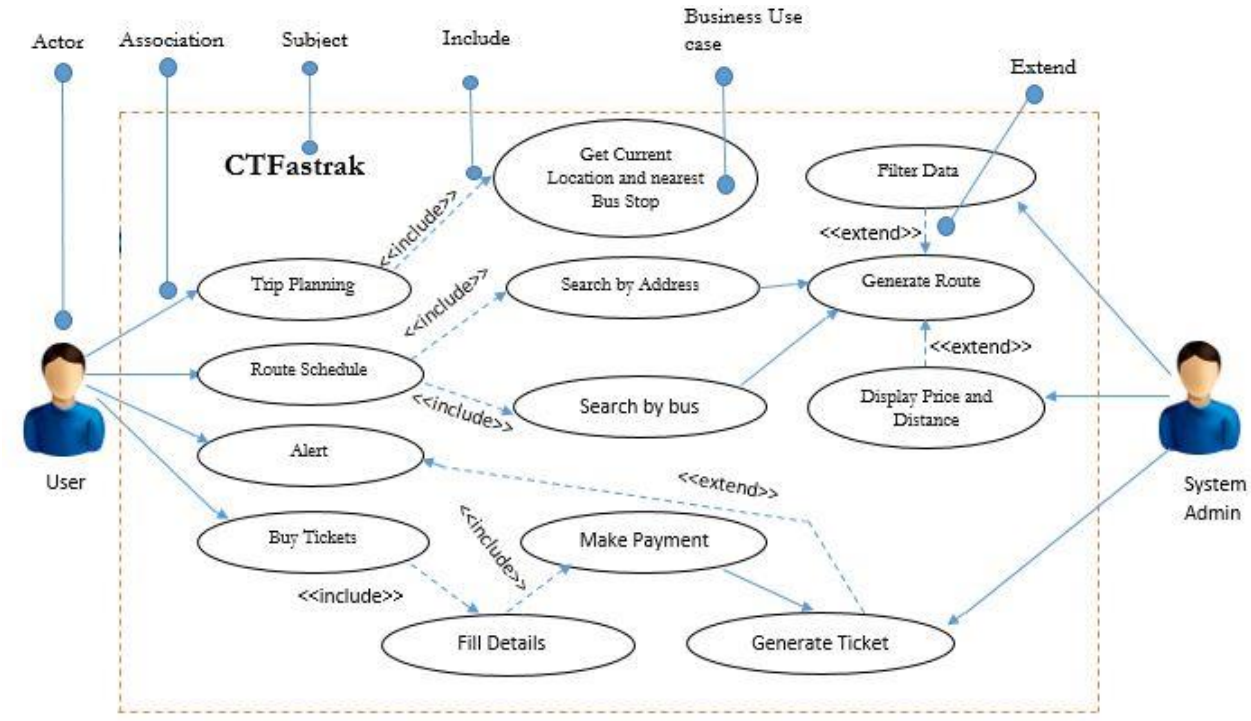
Next Departures and Alerts:

- The System shall provide estimated real-time departure times for all FastTrack buses and provides scheduled departure times for CT-Transit and Amtrak at a customer selected stop.
- The System shall also provide Park and Ride details with Alert System.

Online Bus Ticketing and Pass System:

- The System shall provide online ticket purchase and eTicket generation facility.

Use Case Diagram:



User Stories with Pre-Conditions and Post-Conditions:

User Story	Goal	Reason/Benefit	Pre and Post Conditions
1.User -> Trip Planning-> Get current locations and nearest bus stops	Trip Planner Function is used to generate the route to the destination and find the nearest bus stop and get the travel start and end time	User will be able to get the bus timings and nearest bus stop to travel to a specific location in less than 3 secs	Preconditions: - Current Location is tracked by the system - User should enter the destination address Post condition: - User is able to see the bus timings and the route details

2.User -> Route Schedule -> Search by bus or Search by address	Route Schedule Function is used to generate bus schedule for the user	User will be able to download a Bus Timings PDF file in less than 3 secs	Precondition: - Current location is tracked by system - User should enter the bus number Post condition: - PDF file will be generated by the system
3.User ->Alert	Alert Function is used to set alert/ notification for the user	The user will be notified for the set alert.	Precondition: - User should have used Trip Planner Function - User should have route schedule function - User should have used Buy ticket function Post condition: - Alert or notification will be sent to user by the system
4.User -> Buy Ticket ->Fill Details -> Make Payments	Buy ticket function is used to buy online tickets for the user.	The user is able to buy ticket online and get an e-ticket.	Precondition: - User should have used Trip planner function - User should have filled credit/debit PayPal details Post condition: - User will get an e-ticket copy on his device
5. System Admin ->Display Price	System Admin uses Display Price to display ticket price	User will be able to see the ticket price	Precondition:

			- User should use the Buy ticket function Postcondition: - System Admin will display ticket
6. System Admin -> Generate Route	System Admin uses Generate Route function to call Google API at the backend of the application to generate best bus routes	User will be able to see the bus timings with the best routes displayed	Precondition: - User should use Trip planning and Route schedule function Postcondition: - User will get the best route and bus timing
7. System Admin -> Generate Ticket	System Admin uses the Generate Ticket function to generate e-ticket to the user	User will get the e-ticket via email	Precondition: - User should use the Buy ticket function Postcondition: - User will get e-ticket in email

- **Complex Stories:**

  User (Students, Office workers, General Commuters, Handicapped people, Blind people) can use Voice Control Function for all Main functions like Trip Planning, Route Schedule, Alert, Buy Tickets. Depending on the option user chooses the User Story for each function will vary if the Voice Control Function is used. These user stories need to be broken down.

- **Non-Functional Requirements:**

  The functional requirements for the system are divided into four main categories: Performance requirements, logical database requirements, design constraints, standard compliance, availability, and Security.

Performance Requirements:

- The load time for user interface screen shall no longer than two seconds.
- Queries shall return results within five seconds.

Logical Database Requirements:

- The database should retain the following information:
  Online ticket booking information: Passenger Name, Passenger ID Number, Phone Number, Travel Details, Payment Type, and Total bill.

Design Constrains:

- The application shall be a stand-alone system running on Android and IOS environment.

Standard Compliance:

- There shall be specific consistency in variable names within the system. The graphical user interface is designed to have consistent look and feel.

Availability:

- The System shall be available for 24/7.

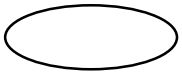
Security:

The System shall be protected against threats and hackers by using firewalls and network security tools.

❖ Glossary of Term Definitions:

Use Case Notations

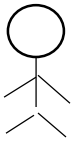
1.



Use Case

A use case represents a user goal that can be achieved by the system or software application.

2.



Actor

Actors are used to representing the users of system

3.

Association

Actor and use case can be associated to indicate that the actor participates in that use case, is described by Association.

4.

<<include>>	*reuse* of functionality (i.e. the included functionality is used or could be used elsewhere in the system).

5.

<<extend>>	adding *new* features/capabilities to a use case

6.

----------► Dependency	A dependency relationship represents that a model element relies on another model element for specification

Other Terms:

User Story:	capture a description of a software feature from an end user perspective and helps to create a simplified description of a requirement.
Pre-Conditions:	Pre-conditions are tests that must prove true before the use case is allowed to proceed.
Post-Conditions:	Post-conditions identify the items that the use case must handle before terminating.