

**Design and implementation system for passenger Positioning System**

Table of content: .

[**USE CASE**](#_b1bmvr5rwyby) **2**

[Various Actors and their Use Cases](#_vmb5yovad9jr) 3

[1) Passenger](#_pukwcz70tbbo) 3

[2) Transportation Agency](#_fn6gse8jnno0) 3

[3) System](#_yhubbyts738j) 3

[4) GPS](#_8fk5tlfjih2q) 4

[**Class diagram**](#_ixpe6t9l77wf) **4**

[Main classes for class diagram](#_fbqj3v7sjhpo) 5

[**Sequence diagram**](#_aa5nyeccnna2) **7**

[Sequence Diagram for Register](#_mp6zv7fugodm) 7

[Sequence Diagram for login Use case](#_r9srdw1x5bul) 9

[Sequence Diagram for View Passengers Location](#_lh4se0ppg9v4) 9

[Sequence Diagram for Allow Access To Location](#_c0isyaa5l2de) 11

[**Activity Diagram**](#_pbsdszxp1i1u) **11**

[Main scheme or flow](#_ya2n0z1inj3f) 12

# 

# 

# **USE CASE**

A use case is a description of a specific interaction between a user (or actor) and the system to achieve a specific goal. A use case must yield an observable result that is of value to the user of the system. An actor represents a role of a user that interacts with the system that you are modeling

As mentioned it consists of actors of different priorities to the system. The use cases in this diagram solve the problem of the functional requirements which was mention in the system requirement document don earlier

Usecase have three main type of actors but for our mobile application, we have two main actors which are the primary and secondary actors, which are located at the left and right of the use case diagram respectively

For a passenger positioning system, the actors in the system are .

**Actors**

* Passenger(Primary)
* Transportation Agency(Primary)
* System(Secondary)
* GPS(Secondary)

### Various Actors and their Use Cases

#### 1) Passenger

- Register

- Allow Access To their Location

- View Others passengers Location

- Login

#### 2) Transportation Agency

- Register

- Login

- View Passengers Location

#### 3) System

- Send Notification

- Regard Update

- Regard Alert

- Provide Relevant Info

- Calculate ETA

- Generate Report

- Provide Analysis

#### 4) GPS

- Takes Passenger Location

- Provide Passenger Location

The figure below is the sample use case diagram for the aforementioned system

# 

# **Class diagram**

This is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

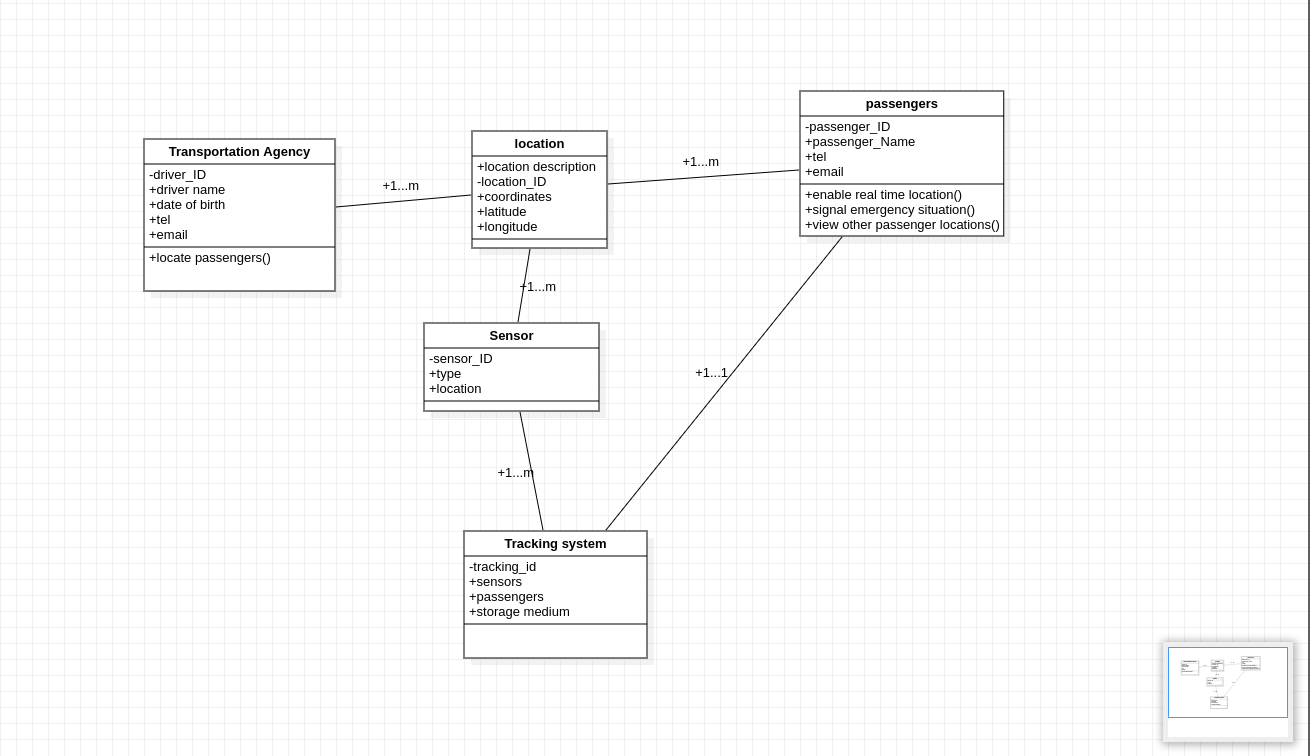
For the system we were required to model, we identified the following classes:

Implementing a class diagram for Passenger Positioning system:

After analyzing the system we were able to come out with the various classes for our class diagram:

## Main classes for class diagram

1. **Passenger**: Represents a passenger who is being tracked by the system.
2. **Location**: Represents a location within the transportation vehicle or facility.
3. **Sensor**: Represents a device that is used to detect the location of passengers.
4. **Tracking System**: Represents the main system that coordinates the tracking of passengers and communicates with the sensors.
5. **Transport Agency**: They are drivers or companies who locate passengers on the map and get to them thereby minimizing fuel consumption



From the diagram above, we can view the various relationships among the classes:

* The location class is linked to the transport agency by a one to many relationship that is a transportation agency may have one or many locations viewed on the map
* Passenger class too share a common relationship with the location class which is also in the sense that passenger can have one or multiple locations on the map
* For each passenger in the system the can have one and only one tracking system per passenger for their device
* Sensors class share a one to many relationship with both the location class and the tracking system class
* The tracking class makes sure it get details for each passenger based on the sensor on their various devices

# **Sequence diagram**

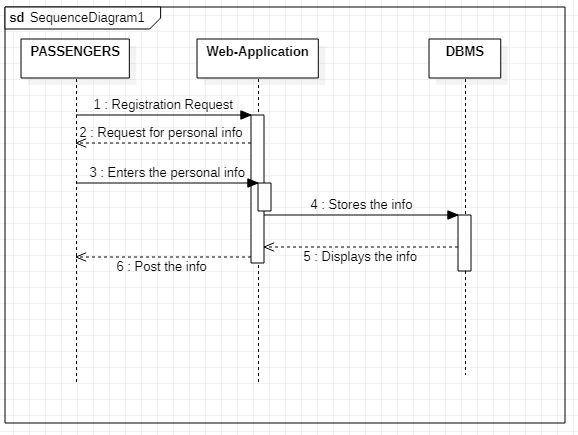
A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction. A sequence diagram shows the sequence of messages passed between objects. Sequence diagrams can also show the control structures between objects.

Sequence diagrams always come up as a result of use cases of an application.

Even though not all of the use cases were used to provide the sequence diagram of the entire application, below are a series of sequence diagrams for some of the major use cases of the system.

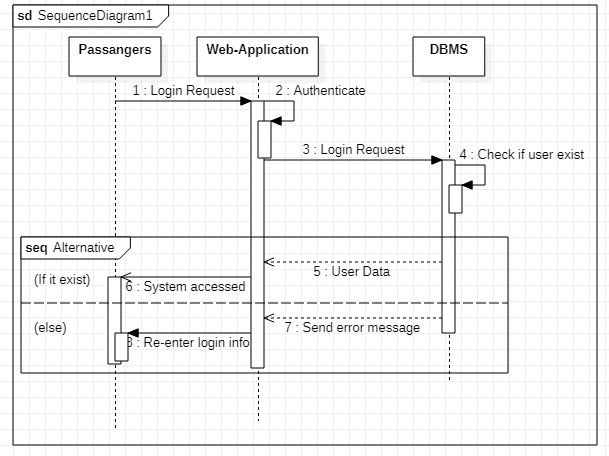
### Sequence Diagram for Register

* Firstly we have a sequence diagram which came to existence from use case called Register



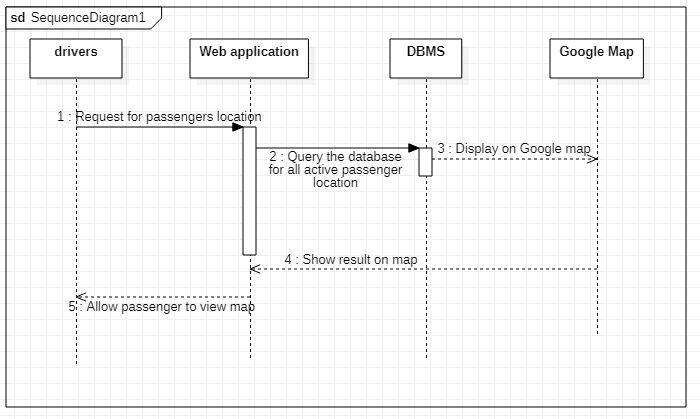
* Here Both primary users, which are the passengers and transport agency register their personal information on the web application which in turn processes the data and sends it in the database which is being hashed and stored since it’s their confidential information..
* Then a request is sent back to the web app signaling the users that the account has been successfully created.

### Sequence Diagram for login Use case

* Secondly this sequence diagram is as a result of the Login use case which was assigned to the two primary actors that is the Passenger and the Agency.
* The scheme of flow is rather straightforward, in that before a passenger or Agency login their credentials are to be verified on the database.
* The Database checks if the information entered is correct and if correct returns a positive feedback else an error message to the system.

### Sequence Diagram for View Passengers Location

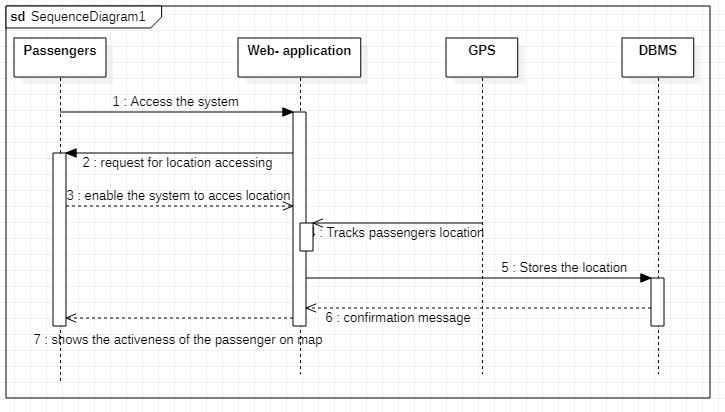
* Thirdly this sequence diagram is as a result of the View Passengers Location use case which was assigned to just one of the actors called the Transportation Agency though bought of the primary actors which are the passengers and the agency are able to view passengers location.



* This sequence represents the flow of the transport agency locating users on the map via geolocation.
* Here travel agencies requesting for passenger location is queried on the web app and the details are sent to the data base which retrieves the information and sends the location details to the map API which inturns return the output on the web application.
* Also passengers will be able to see other passengers location which may drive them to go where there are many passengers in other to have more chances of having a taxi.

### Sequence Diagram for Allow Access To Location

The following sequence diagram gives us a brief overview of the passenger enabling real time location on the web application.



* Here, the passengers enable their location immediately after accessing the web app for real time monitoring of the passengers location.

# **Activity Diagram**

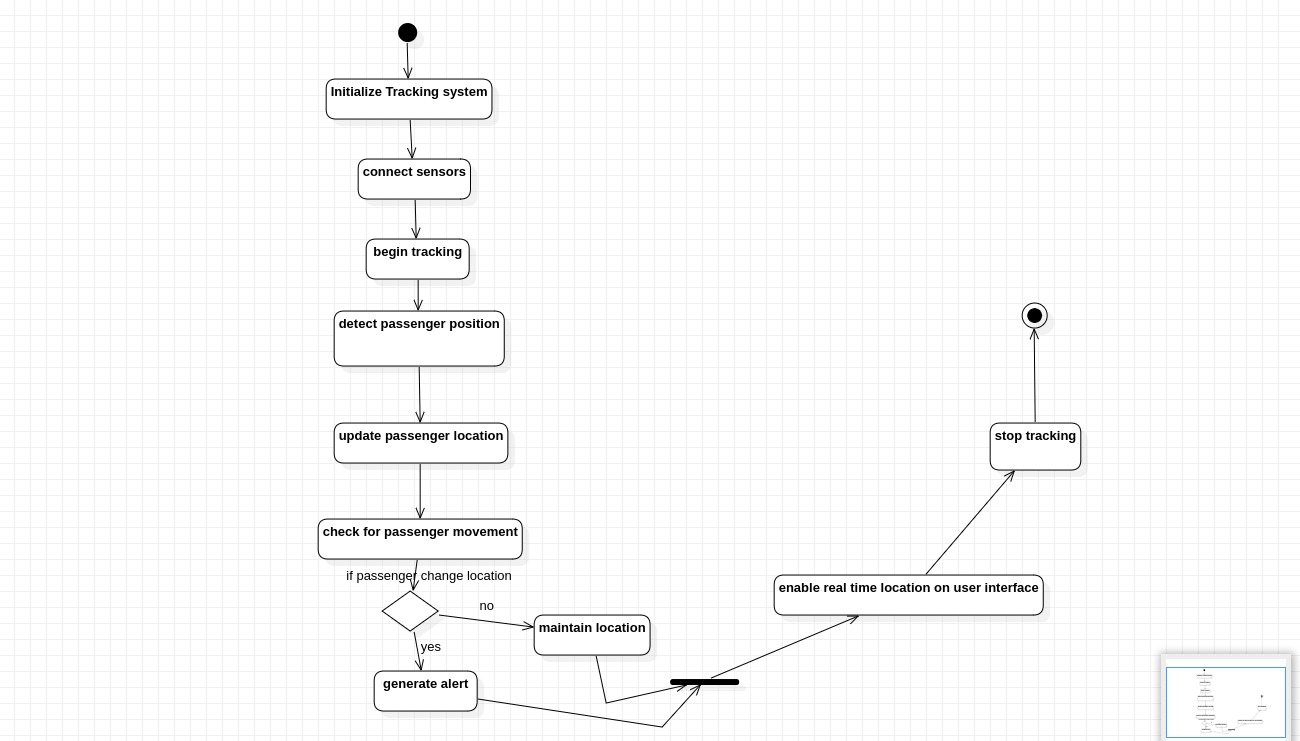
This visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent.

The activity diagram illustrates the basic flow of events in a passenger positioning system. The system starts by initializing the tracking system and connecting the sensors. It then begins tracking the passengers, detecting their location and updating it as necessary. The system checks for any unexpected movement by the passengers and generates an alert if necessary. The system also stores the passenger location data and provides real-time location data to the user interface. Reports on passenger movement can be generated as needed. Finally, the system stops tracking when the user disables on his phone and the process ends.

So these are the various stages or activities in our system:

### Main scheme or flow

1. Start
2. Initialize Tracking System
3. Connect Sensors
4. Begin Tracking
5. Detect Passenger Location
6. Update Passenger Location
7. Check for Passenger Movement
8. Generate Alert if Passenger Moves from Expected Location
9. Store Passenger Location Data
10. Provide Real-Time Location Data to User Interface
11. Generate Reports on Passenger Movement
12. Stop Tracking
13. End



So that is the basic systematic flow of events relative to the flow of our system.

And its pretty straight forward form the diagram.

This summarizes the basic diagrams required for our system.