COMP-246 Term Project (Part-A+B+C)

GO BUS MOBILE APPLICATION

A picture containing clipart

Description generated with very high confidence

**Submitted By:**

Dr Md Fardoush Rahman (300958291)

Masudur Rahman (300578293)

Salim Sayeed (300933148)

Md Humayun Kabir (300527330)

Date: Fall 2018

**Submitted To:**

Professor: Mohamed Khan

Table of Contents

[GO Bus Mobile Application 4](#_Toc531795898)

[Problem Statement 4](#_Toc531795899)

[Vision description 4](#_Toc531795900)

[System features 4](#_Toc531795901)

[Benefits of the system 4](#_Toc531795902)

[Subsystems: 4](#_Toc531795903)

[Use Cases 6](#_Toc531795904)

[1.0 Registration Subsystem 6](#_Toc531795905)

[2.0 Service Subsystem 7](#_Toc531795906)

[3.0 Tracking Subsystem 8](#_Toc531795907)

[4.0 Notification Subsystem 9](#_Toc531795908)

[5.0 Feedback Subsystem 10](#_Toc531795909)

[User’s story 11](#_Toc531795910)

[Registration 11](#_Toc531795911)

[Login 11](#_Toc531795912)

[Display Bus location 11](#_Toc531795913)

[Display Arrival Time 12](#_Toc531795914)

[Display Route 12](#_Toc531795915)

[Track Bus 12](#_Toc531795916)

[Calculate Time 13](#_Toc531795917)

[Update Bus 13](#_Toc531795918)

[Update Route 13](#_Toc531795919)

[Add Comments 13](#_Toc531795920)

[View Ratings 13](#_Toc531795921)

[Workflows by subsystem – Activity UML Diagram 14](#_Toc531795922)

[1.0 Registration Subsystem 14](#_Toc531795923)

[2.0 Service Subsystem 14](#_Toc531795924)

[3.0 Tracking Subsystem 15](#_Toc531795925)

[4.0 Notification Subsystem 16](#_Toc531795926)

[5.0 Feedback subsystem 17](#_Toc531795927)

[Domain Class Diagram 18](#_Toc531795928)

[Description of Multiplicities 19](#_Toc531795929)

[ERD 19](#_Toc531795930)

[Revised Domain Class Diagram 20](#_Toc531795931)

[Detailed Design Class Diagram – packaged by subsystem 21](#_Toc531795932)

[1.0 Registration Subsystem 21](#_Toc531795933)

[2.0 Service Subsystem 22](#_Toc531795934)

[3.0 Tracking Subsystem 23](#_Toc531795935)

[4.0 Notification Subsystem 23](#_Toc531795936)

[5.0 Feedback Subsystem 24](#_Toc531795937)

[CRC Card 24](#_Toc531795938)

[System Sequence Diagrams 25](#_Toc531795939)

[Login 25](#_Toc531795940)

[Display Bus Location 25](#_Toc531795941)

[Track Bus 25](#_Toc531795942)

[Display Bus Route 26](#_Toc531795943)

[Display Arrival Time 26](#_Toc531795944)

[Sequence Diagram 26](#_Toc531795945)

[Login 26](#_Toc531795946)

[Display Bus Location 27](#_Toc531795947)

[State Machine Diagram 27](#_Toc531795948)

[Bus Status 27](#_Toc531795949)

[Commuter Status 27](#_Toc531795950)

[Mock-up UI 28](#_Toc531795951)

[Component & Deployment Diagram 29](#_Toc531795952)

[Technology tools used for Software Development 29](#_Toc531795953)

[Skeletal Code 30](#_Toc531795954)

[Gantt Chart 32](#_Toc531795955)

# GO Bus Mobile Application

# Problem Statement

## Vision description

The GO Bus needs to build an official mobile app. Most of time people use GO website to see the time and route of their desired transport, but passengers do not get the real-time data about their buses. Buses arrival time and locations varies and not fixed. So, commuters need to wait at the stop until the buses arrive. The waiting time on an average is 10 - 15 minutes. Moreover, some stops do not have sheds and passengers suffer during the extreme weather.

## System features

* The app will display a list of the buses that will arrive at the stop within an hour based on a specific bus stop.
* It will show the passengers how much time will need to get to the stops by walking.
* It will show all the different routes for desired destination.
* It will show real-time location of the buses.
* All the routes in the map will be color coded.
* Map will have the icon to show the directions of the routes.
* While active, the app will use the GPS of the passengers’ device to locate the position in the map.
* Passengers can search for the nearest stops.
* The app will send a notification if the buses have arrived at a designated bus stop, is out of service or take a detour.
* Passengers will receive advance notifications before arrival in desired stations.
* The app will show all the possible routes that the passengers search for.

## Benefits of the system

* Passengers will get the real-time information about the schedule of their transit.
* Passengers and drivers conflict can be minimized by reducing the waiting time.
* It will boost company’s reputation by increasing commuter’s satisfaction.
* Transit will be more efficient in terms of cost management
* If company will make more profit, government subsidy will be reduced.

# Subsystems:

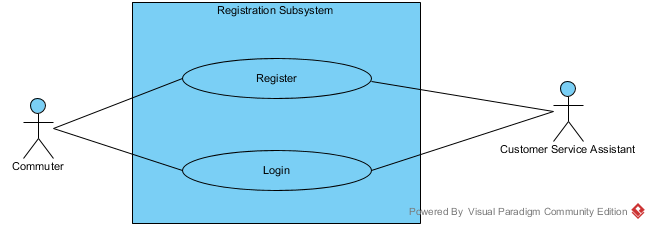
1. Registration subsystem

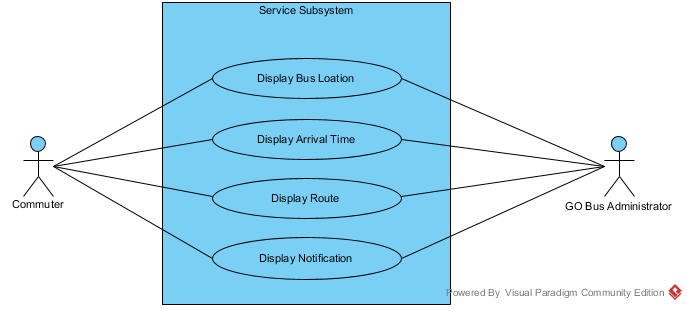
2. Service subsystem

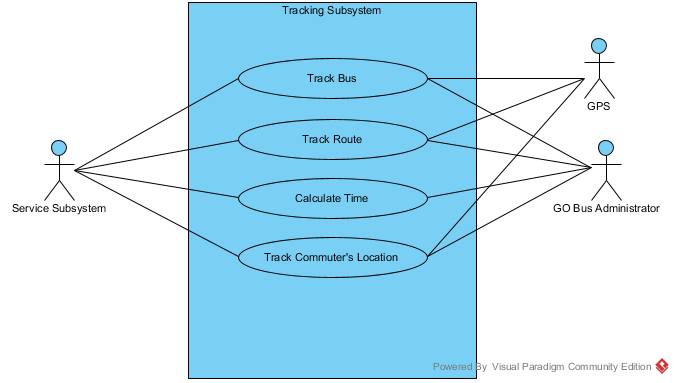
3. Tracking subsystem

4. Notification subsystem

5. Feedback Subsystem

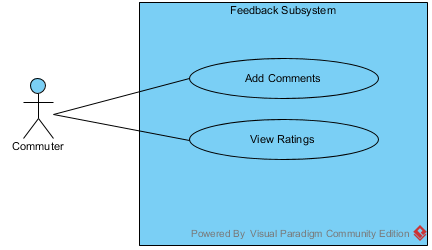






A close up of a map

Description generated with very high confidence



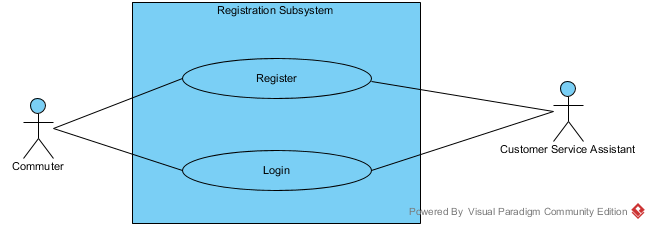
# Use Cases

## 1.0 Registration Subsystem

|  |  |
| --- | --- |
| User | User’s goal and resulting use case |
| Commuter | Register an account for GO Bus Mobile Application.  Login to an existing account to connect to the GO Bus Mobile Application. |
| GO Bus commuter service assistant | Register an account for a commuter for GO Bus Mobile Application.  Login to an existing account to connect to the GO Bus Mobile Application. |

|  |  |
| --- | --- |
| Use Case | Brief use case description |
| Register | Commuter/Go Bus commuter service assistant registers an account for the GO Bus Mobile Application. |
| Login | Commuter/Go Bus commuter service assistant logs into the system. |

|  |  |
| --- | --- |
| Registration Subsystem | |
| Use Cases | Users/Actors |
| Register | Commuter, GO Bus commuter service assistant |
| Login | Commuter, GO Bus commuter service assistant |

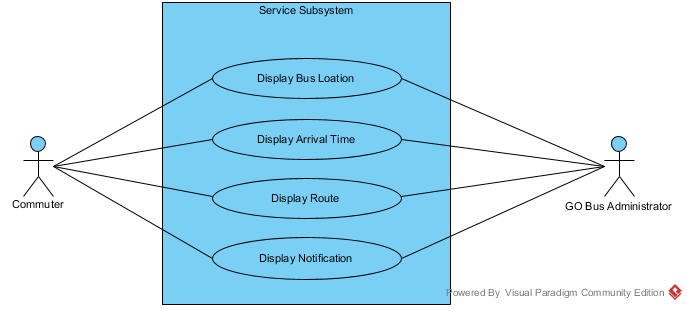


## 2.0 Service Subsystem

|  |  |
| --- | --- |
| User | User’s goal and resulting use case |
| Commuter | Login to an existing account to view the Display Bus Location, Display Arrival Time, Display Route, and Display Notification. |
| GO Bus Administrator | Update Bus Location, Arrival Time, Route and Notification for GO Bus Mobile Application. |

|  |  |
| --- | --- |
| Use Case | Brief use case description |
| Display Bus Location | Display the exact bus location at point in time. |
| Display Arrival Time | Display the arrival time at the any bus stoppage/station. |
| Display Route | Display the bus route at any particular location. |
| Display Notification | Display notification if any. |

|  |  |
| --- | --- |
| Service Subsystem | |
| Use Cases | Users/Actors |
| Display Bus Location | Commuter, GO Bus Administrator |
| Display Arrival Time | Commuter, GO Bus Administrator |
| Display Route | Commuter, GO Bus Administrator |
| Display Notification | Commuter, GO Bus Administrator |

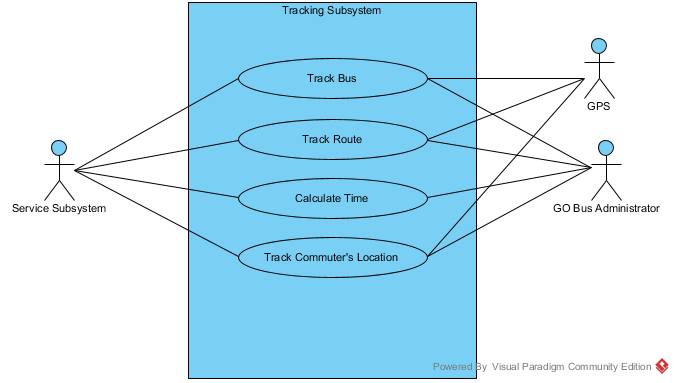


## 3.0 Tracking Subsystem

|  |  |
| --- | --- |
| User | User’s goal and resulting use case |
| Service Subsystem | Service subsystem will track the bus, route, calculate the estimated time and track the commuter location. |
| GO Bus Administrator | Go bus administrator can track bus, route and commuter location and calculate the estimated time for any route. |

|  |  |
| --- | --- |
|  | Brief use case description |
| Track Bus | Service subsystem/Go bus administrator tracks the bus location using GPS |
| Track Route | Service subsystem/Go bus administrator tracks the route of the buses using GPS |
| Calculate Time | Service subsystem/Go bus administrator calculates the estimated time for a bus to reach at a particular point. |
| Track Commuters Location | Service subsystem/Go bus administrator tracks the commuters location using GPS |

|  |  |
| --- | --- |
| Service Subsystem | |
| Use Cases | Users/Actors |
| Track Bus | Service Subsystem, GO Bus administrator, GPS |
| Track Route | Service Subsystem, GO Bus administrator, GPS |
| Calculate Time | Service Subsystem, GO Bus administrator |
| Track Commuter Location | Service Subsystem, GO Bus administrator, GPS |



## 4.0 Notification Subsystem

|  |  |
| --- | --- |
| User | User’s goal and resulting use case |
| Go Bus Administrator | Go Bus Administrator have right to add, remove, update for route and also bus depending on demand by commuters and Go Bus authority. |

|  |  |
| --- | --- |
| Use Case | Brief use case description |
| Add New Route | Go Bus Administrator will add new route for commuter satisfaction depending on their demand. |
| Remove Route | Go Bus Administrator will remove route to save money for Go Bus authority. |
| Update Route | Go Bus Administrator will update route time to time for not confusing time management for commuter. |
| Add New Bus | Go Bus Administrator will add bus for frequent service for satisfying commuters demand. |
| Remove Bus | Go Bus Administrator will remove bus to save money for the authority. |
| Update Bus | Go Bus Administrator will update bus to know when the bus will arrive or departure from the station. |

|  |  |
| --- | --- |
| Notification Subsystem | |
| Use Cases | Users/Actors |
| Add New Route | Go Bus Administrator |
| Remove Route | Go Bus Administrator |
| Update Bus | Go Bus Administrator |
| Add New Bus | Go Bus Administrator |
| Remove Bus | Go Bus Administrator |
| Update Bus | Go Bus Administrator |

A close up of a map

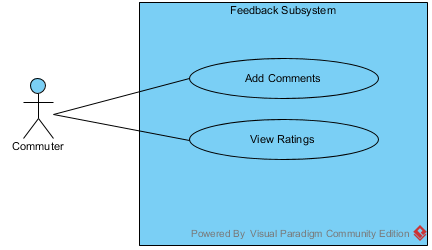
Description generated with very high confidence

## 5.0 Feedback Subsystem

|  |  |
| --- | --- |
| User | User’s goal and resulting use case |
| Commuter | Put comments and suggestions about the app.  Rating the services of GO Bus Mobile Application. |

|  |  |
| --- | --- |
| Use Case | Brief use case description |
| Comments | Commuters can add comments about the quality of the services of Go bus and about the price and routs. |
| Rating | Commuter gives the rating to the services of the application. |

|  |  |
| --- | --- |
| Feedback Subsystem | |
| Use Cases | Users/Actors |
| Comments | Commuter |
| Rating | Commuter |



## User’s story

### Registration

|  |
| --- |
| USER STORY  *As a commuter*, I want to *register for The GO Bus Application*, so that I can *use all the facilities* of the application.  ACCEPTANCE CRITERIA  1. System must be able to register me as a new commuter.  2. System must accept my details e.g., name, address, email etc.  3. System must validate if all my information is correct  4. System must give confirmation message when my registration is completed. |

### Login

|  |
| --- |
| USER STORY  *As a commuter*, I want to *login to The GO Bus Application*, so that I can *use all the facilities* of the application.  ACCEPTANCE CRITERIA  1. System must be able to log me in as an existing commuter.  2. System must validate if all user name and password is correct  3. System must display all the features of the application to me upon successful login. |

### Display Bus location

|  |
| --- |
| USER STORY  *As a commuter*, *I want to see all the buses available* in my desired route, so that *I can decide* which bus I will take to go to my destination.  ACCEPTANCE CRITERIA  1.The map must be available while the app is running.  2.The search feature must be available to search for a location.  3.Buses location must be sorted by time.  4. System must be able to provide at least two options to choose.  5. Routes must be highlighted.  6. The distance and time must be displayed accurately.  7. Store the choice and options of mine for the next time to save time. |

### Display Arrival Time

|  |
| --- |
| USER STORY  *As a commuter*, I want to *see all the bus’s arrival time* to my current location.  ACCEPTANCE CRITERIA  1. The map must be available while the app is running.  2. The search feature must be available to search for a location.  3. Buses location must be sorted by time.  4. System must be able to provide at least two options to choose.  5. Routes must be highlighted.  6. The distance and time must be displayed accurately.  7. Store the choice and options of mine for the next time to save time. |

### Display Route

|  |
| --- |
| USER STORY  *As a commuter*, *I want to see all the possible routes* for my destination, so that *I can decide* which route I might choose.  ACCEPTANCE CRITERIA  1.The route map must be available while the app is running.  2.The search feature must be available to search for a location.  3.Buses location must be sorted by time.  4. System must be able to provide at least two options to choose.  5. Routes must be highlighted.  6. The distance and time must be displayed accurately.  7. Store the choice and options of mine for the next time to save time. |

### Track Bus

|  |
| --- |
| USER STORY  *As a GO Bus Administrator*, *I want to track bus, route and commuter location*, so that *proper schedule will be maintained.*  ACCEPTANCE CRITERIA  1.System must be able to track bus, route and commuter’s location via GPS.  2.The search feature must be available to search for a location.  3.Buses location must be sorted by time.  4. System must be able to provide at least two options to choose.  5. Routes must be highlighted.  6. The distance and time must be displayed accurately.  7. Store the choice and options of mine for the next time to save time. |

### Calculate Time

|  |
| --- |
| USER STORY  *As a GO Bus Administrator*, *I want to calculate time,* so that *commuter can know the travel time.*  ACCEPTANCE CRITERIA  1.System must be able to track bus, route and commuter’s location via GPS.  2. System must be able to show arrival time on each stoppage.  3. The distance and time must be displayed accurately. |

### Update Bus

|  |
| --- |
| USER STORY  *As a GO Bus Administrator*, *I want to add, remove and update bus status,* so that *commuter can know the real time status of the bus arrival.*  ACCEPTANCE CRITERIA  1.System must be able to add new bus to a route according to commuter’s demand.  2. System must be able to remove a bus from the route to save money.  3. System must be able to update bus schedule accordingly. |

### Update Route

|  |
| --- |
| USER STORY  *As a GO Bus Administrator*, *I want to add, remove and update route,* so that *commuter can know the status of the present route.*  ACCEPTANCE CRITERIA  1. System must be able to add new route according to commuter’s demand.  2. System must be able to remove the route to save money.  3. System must be able to update route. |

### Add Comments

|  |
| --- |
| USER STORY  *As a commuter*, *I want to add comments,* so that *GO bus authority can take necessary initiatives to improve service.*  ACCEPTANCE CRITERIA  1. System must be able to add comments.  2. System must be able to give feedback for the comments. |

### View Ratings

|  |
| --- |
| USER STORY  *As a commuter*, *I want to view ratings,* so that *I can decide about taking their services.*  ACCEPTANCE CRITERIA  1. System must be able to ask ratings from commuters.  2. System must be able receive ratings from commuters.  3. System must be able to display ratings of other commuters. |

# Workflows by subsystem – Activity UML Diagram

## 1.0 Registration Subsystem

*Allows new users to register an account with GO Bus Mobile Application*

1.1 Commuter requests registration

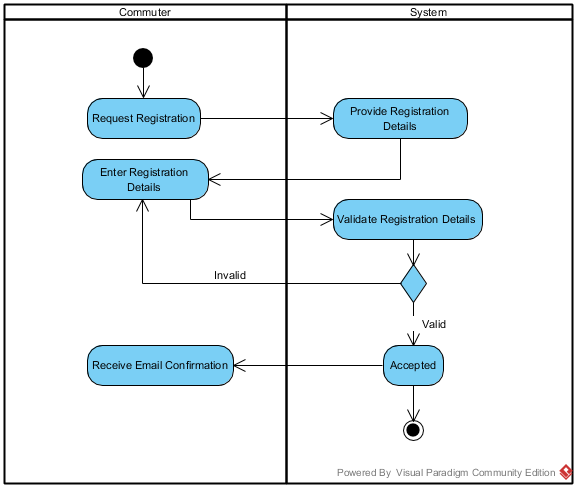
1.2 System provides registration page

1.3 Commuter enters registration details

1.4 System validates registration details

1.4.1 If details are invalid, prompt commuter for details again and proceed to 1.3; if not proceed to 1.5

1.5 Commuter receive email confirmation of acceptance



## 2.0 Service Subsystem

2.1 Commuter request for bus location, system will search location.

2.2 Commuter request for arrival time, system will display arrival time to commuter. Admin updates

arrival time continuously.

2.3 Commuter request for route, system will show the particular route. Admin always update the new

locations, drop old location if any.

2.4 System will display notification to the commuter.

A screenshot of a cell phone

Description generated with very high confidence

## 3.0 Tracking Subsystem

Tracking subsystem allows the administrator to locate the bus by the help of GPS on the route.

It will help to locate the client and estimate the expected time to reach the destination or to

arrive in a stoppage.

3.1 Administrator asks for the bus location on the route and time to reach the destination.

3.2 Global positioning system (GPS) will detect location of the bus on its route.

3.3 System will calculate the estimated time to reach the destination.

A screenshot of a cell phone

Description generated with high confidence

## 4.0 Notification Subsystem

4.1 GO bus administrator searches the route

4.2 System displays all routes

4.2.1 If the system does not find desired route, proceed to 4.1; if finds, proceed to 4.3

4.3 Go bus administrator enters the desired route name.

4.4 The system displayed the route name.

4.5 Go bus administrator edits the route.

4.6 The system will show the edited route.

4.7 Go bus administrator searches the route.

4.8 The system will show the all routes.

4.9 If the system does not find the desired route name the administrator will search the route again

if find, the desired route will be updated.

A picture containing screenshot

Description generated with very high confidence

## 5.0 Feedback subsystem

Feedback subsystem allows the commuters to submit the comments about the apps services,

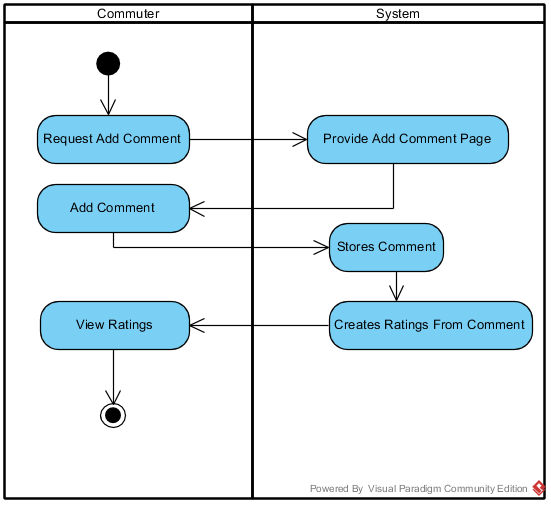
system evaluates the comments and rating the apps according to the commuter’s view.

5.1 Commuter adds comments about the apps service and quality.

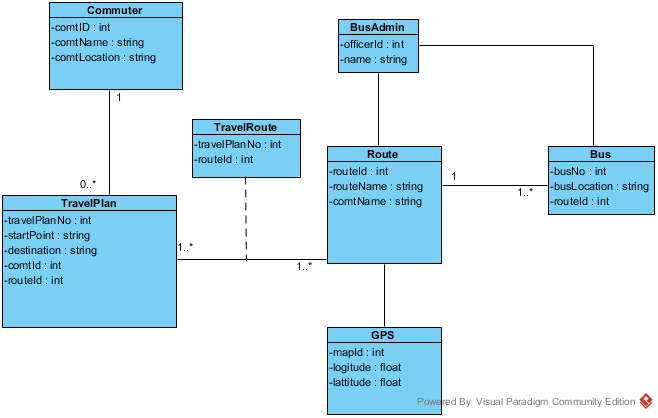
5.2 System receives and stores the comments

5.3 System rate the apps activities according to comments of the commuters.

5.4 System shows the rating for the commuters.



# Domain Class Diagram



## Description of Multiplicities

**Commuter and Travel Plan:**

One commuter can have zero or many travel plans but one travel plan must have assigned to one commuter.

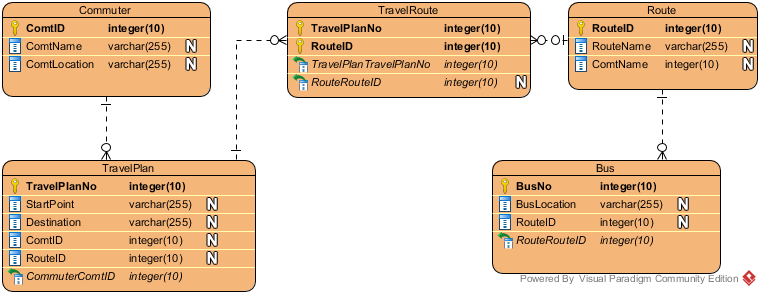
**Travel plan:**

One travel plan can take one or more routes through travel route and one or more routes can be taken by one travel plan through a travel route.

**Route and Bus:**

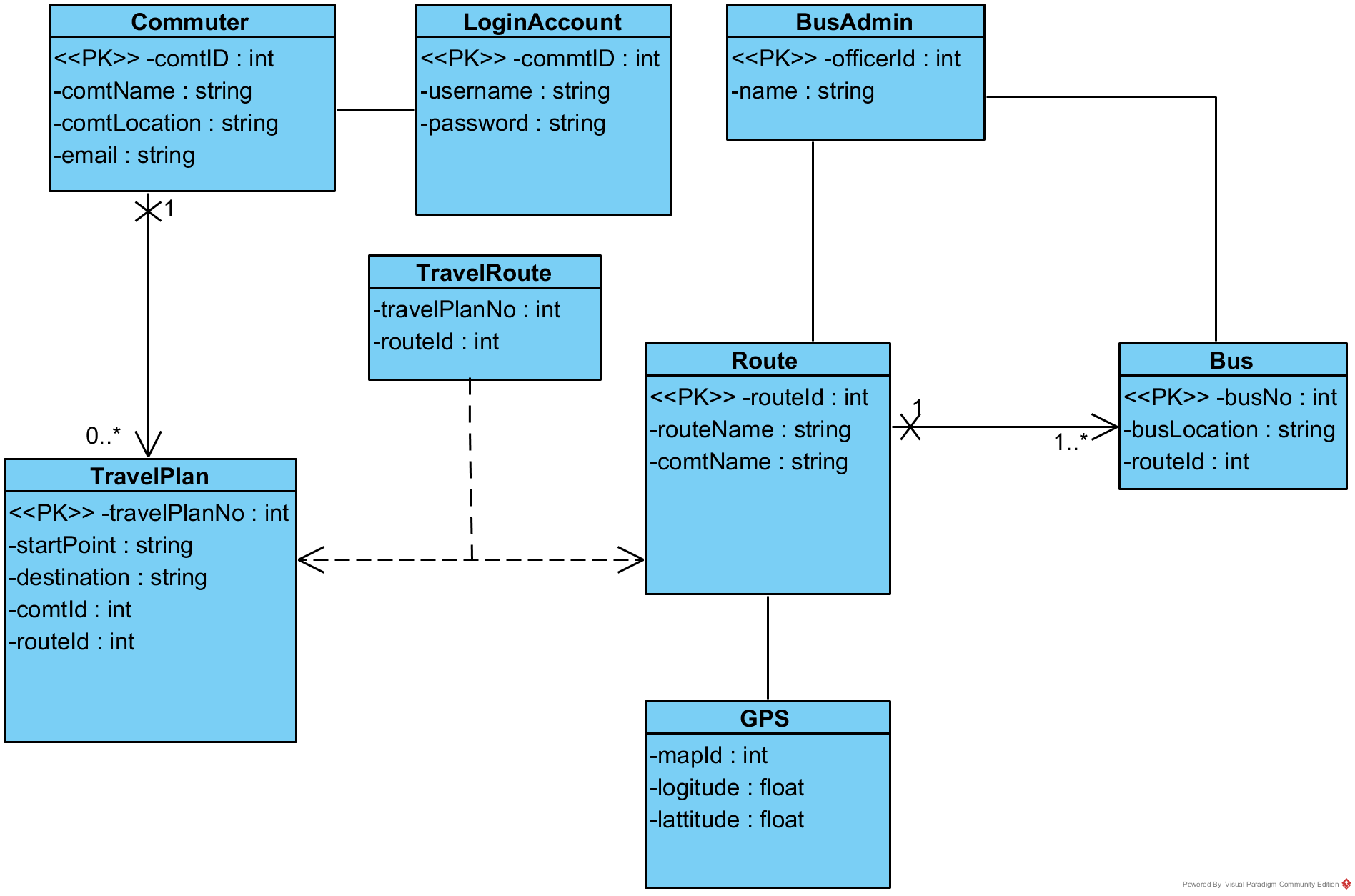
One route can have one or many buses, but one bus must have one route

# ERD



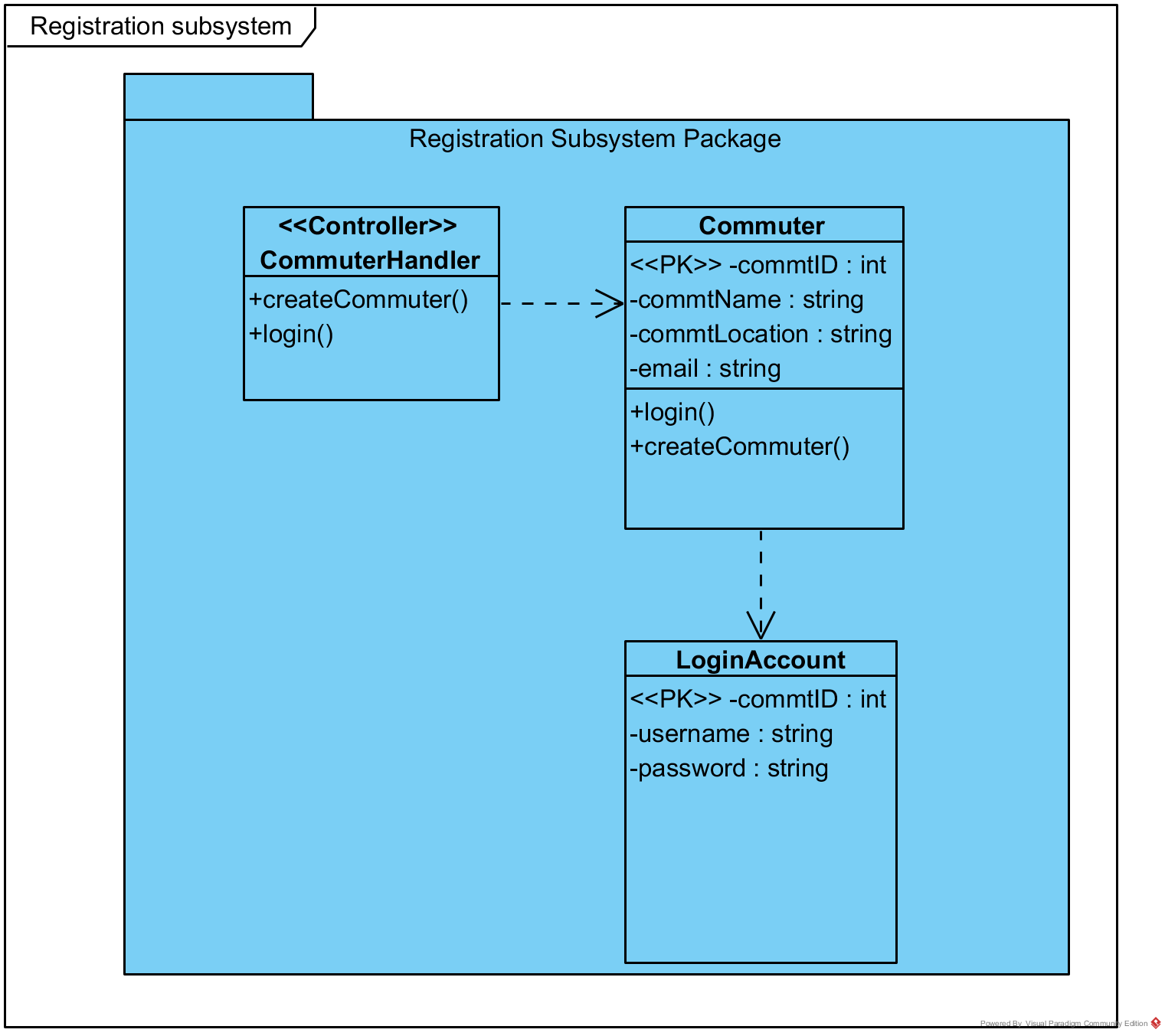
|  |  |
| --- | --- |
| Table | Attribute |
| Commuter | ComtID Integer(10)  ComtName varchar(255)  ComtLocation varchar(255) |
| TravelPlan | TravelPlanNo integer(10)  StartPoint varchar(255)  Deatination varchar(255)  ComtID integer(10)  RouteID integer(10)  CommuterComtID integer(10) |
| TravelRoute | TravelPlanNo integer(10)  RouteID integer(10)  TravePlan TravelPlanNo integer(10)  RouteRouteID intiger(10) |
| Route | RouteID integer(10)  RouteName varchar(255)  ComtName integer(10) |
| Bus | BusNo integer(10)  BusLocation varchar(255)  RouteID integer(10)  RouteRouteID integer(10) |

# Revised Domain Class Diagram

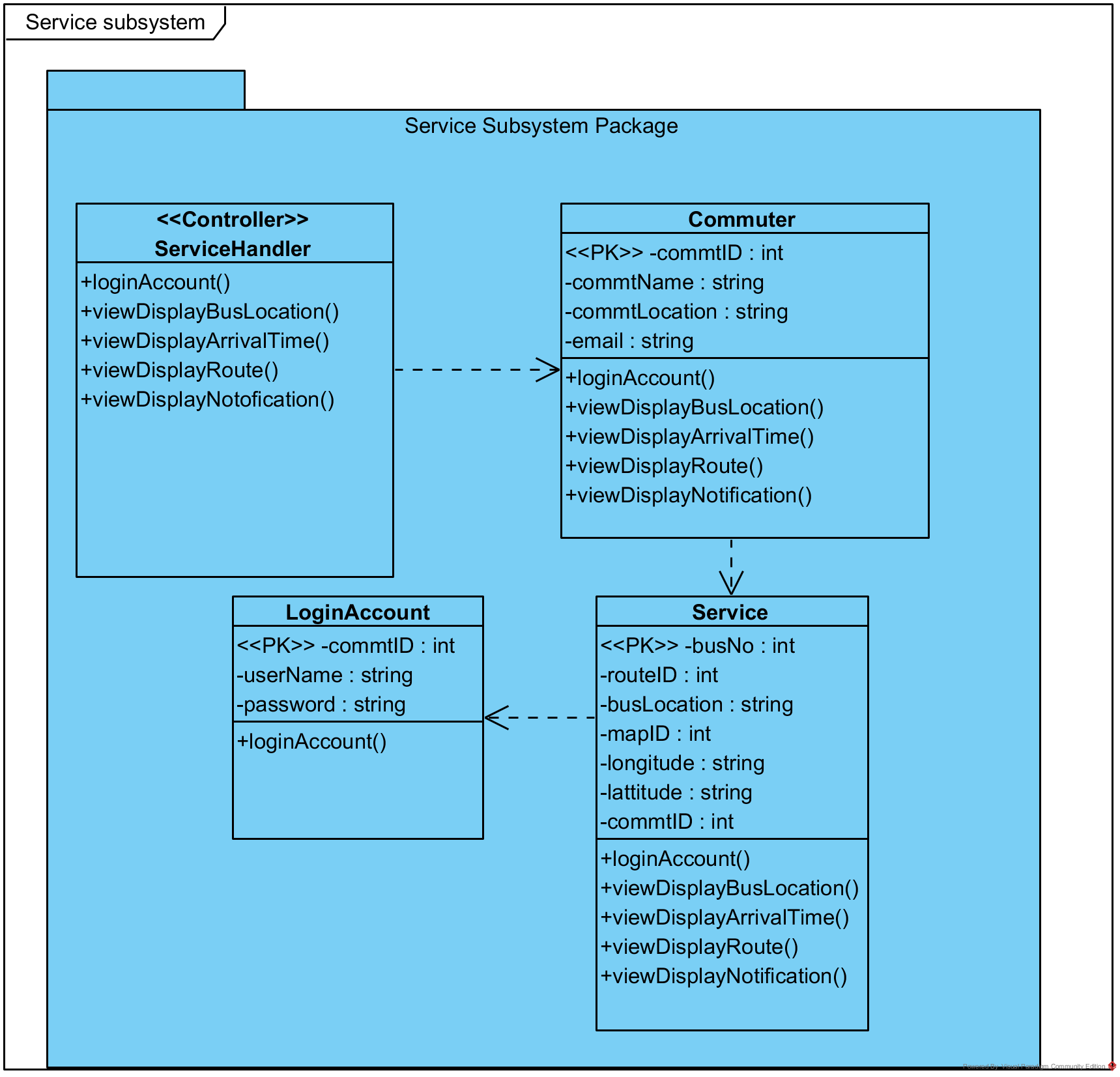


# Detailed Design Class Diagram – packaged by subsystem

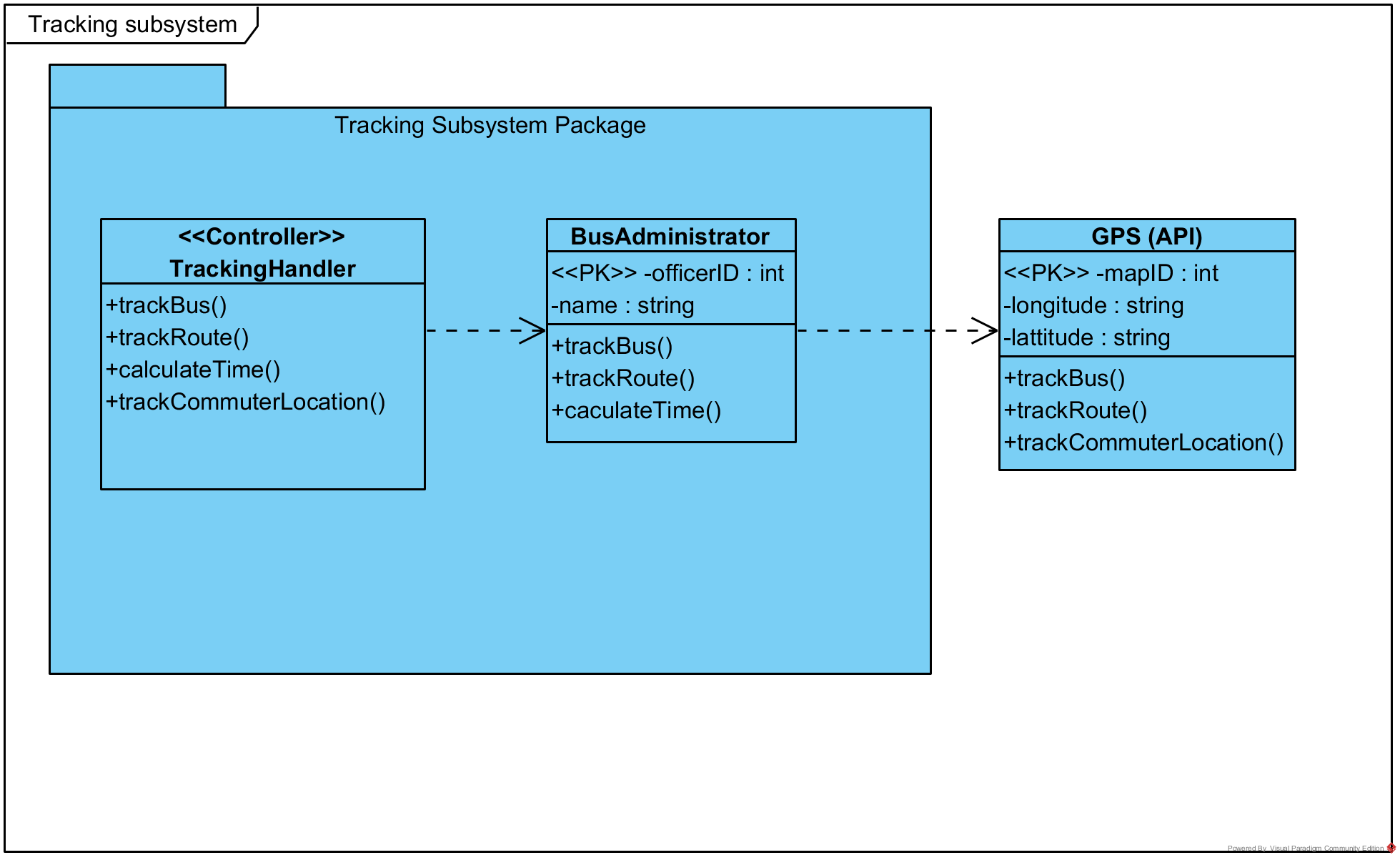
## 1.0 Registration Subsystem



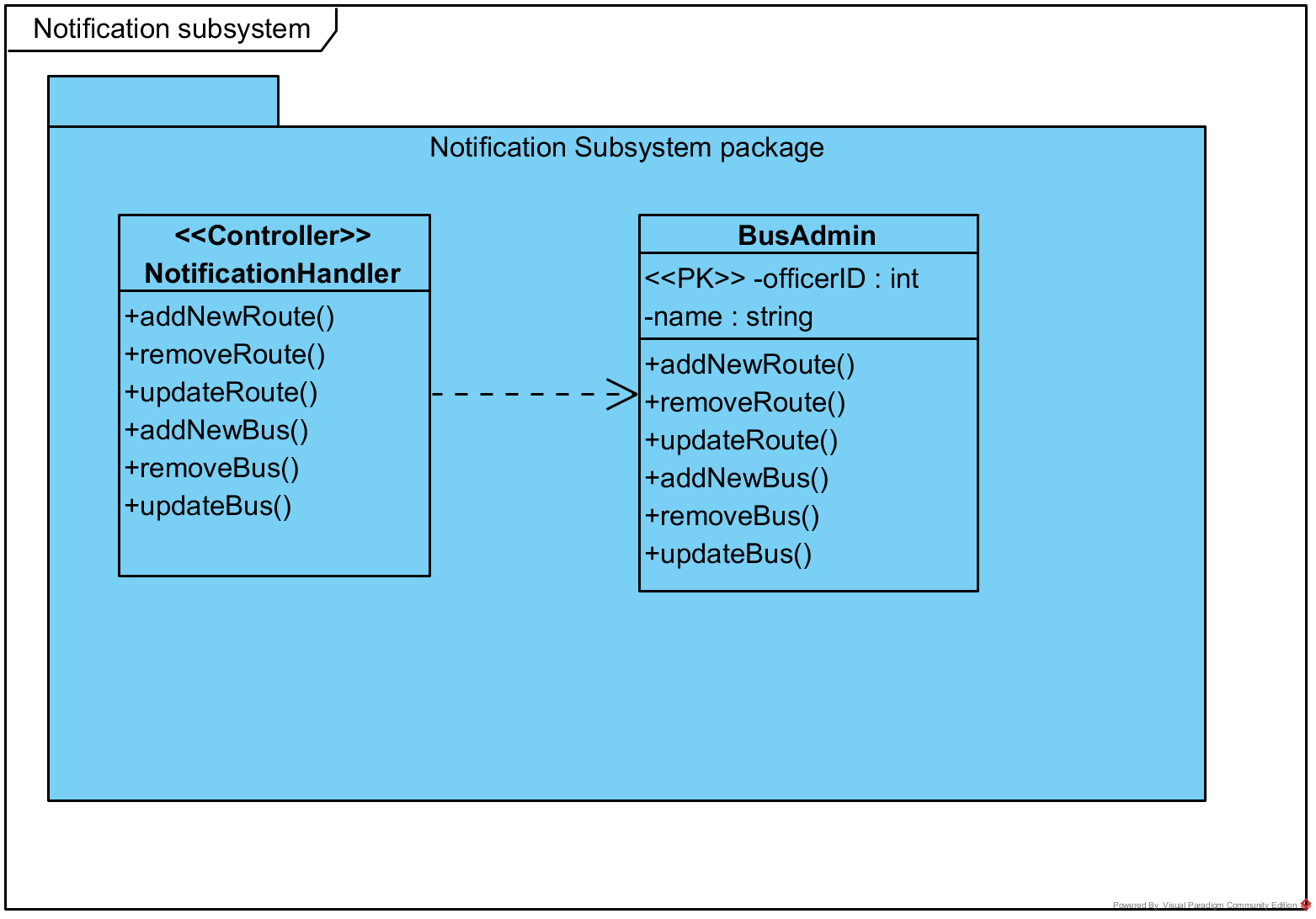
## 2.0 Service Subsystem



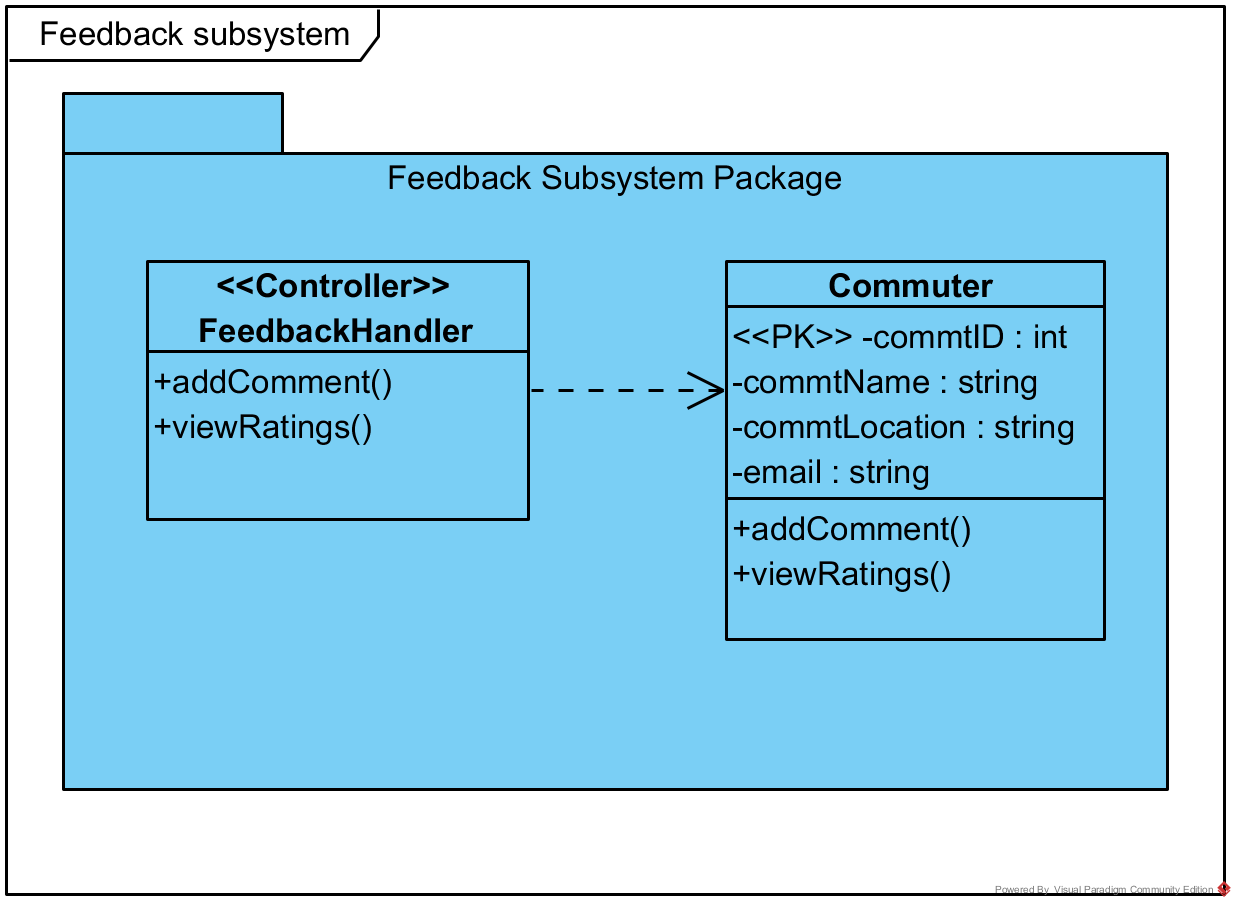
## 3.0 Tracking Subsystem



## 4.0 Notification Subsystem



## 5.0 Feedback Subsystem

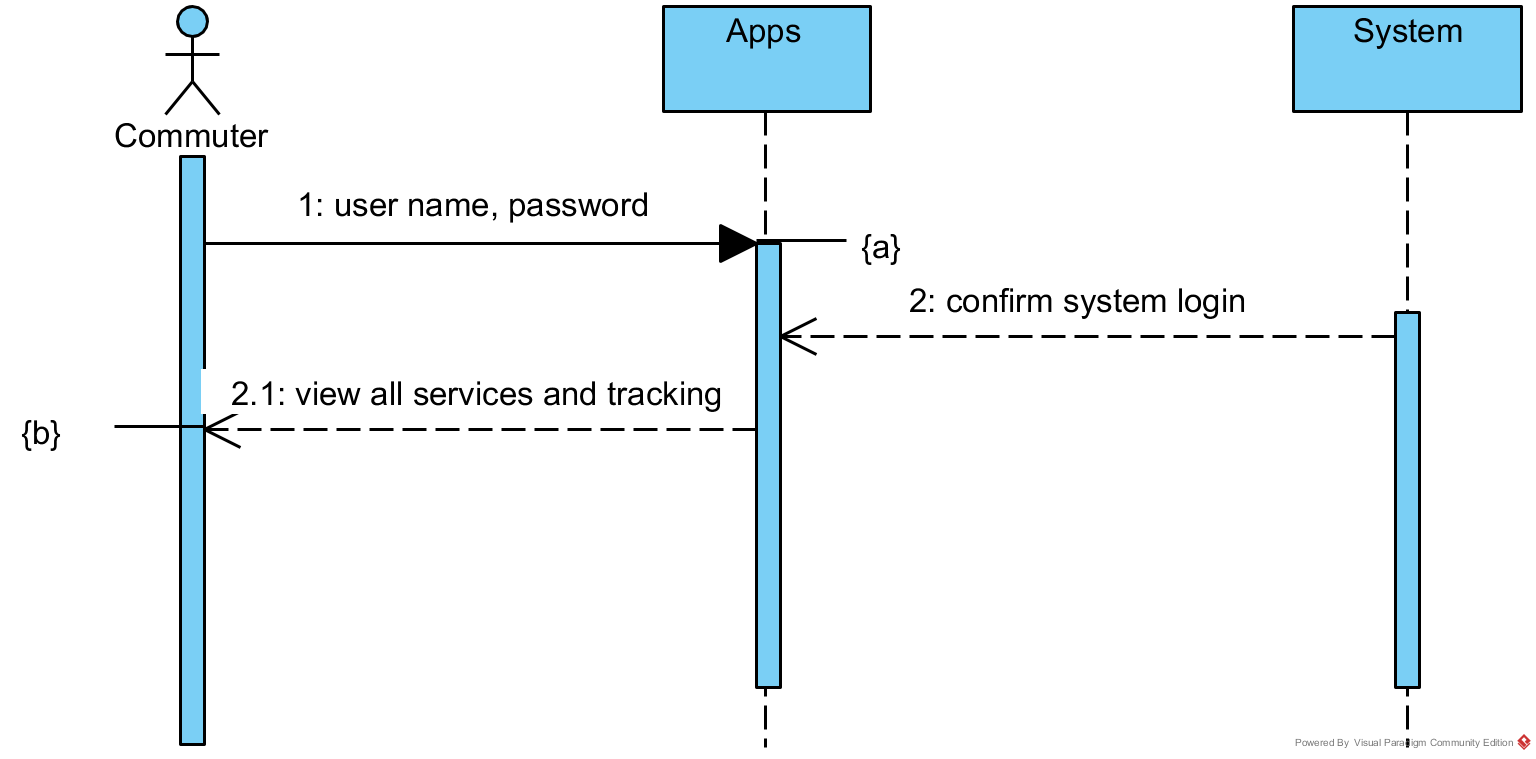


# CRC Card

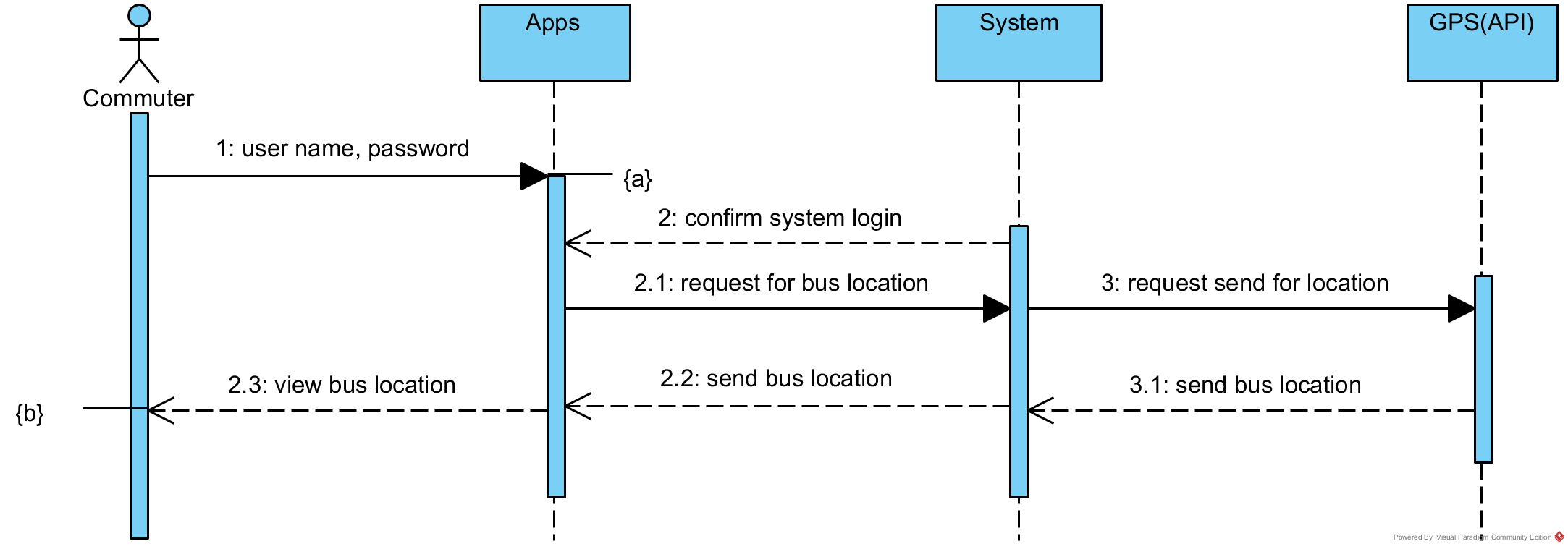
|  |  |  |
| --- | --- | --- |
| Commuter | CRUD  createCommuter()  login()  viewDisplayRoute()  viewDisplayBusLocation()  viewDisplayArrivalTime()  viewDisplayNotification() | LoginAccount, TravelPlan |
| TravelPlan | CRUD  viewDisplayRoute()  viewDisplayBusLocation()  viewDisplayArrivalTime() | TravelRoute, Route |
| Route | R  viewDisplayRoute()  trackBus()  trackRoute()  trackCommuterLocation()  calculateTime()  addNewRoute()  removeRoute()  updateRoute()  addNewBus()  removeRoute()  updateRoute() | GPS, Bus, BusAdmin |

# System Sequence Diagrams

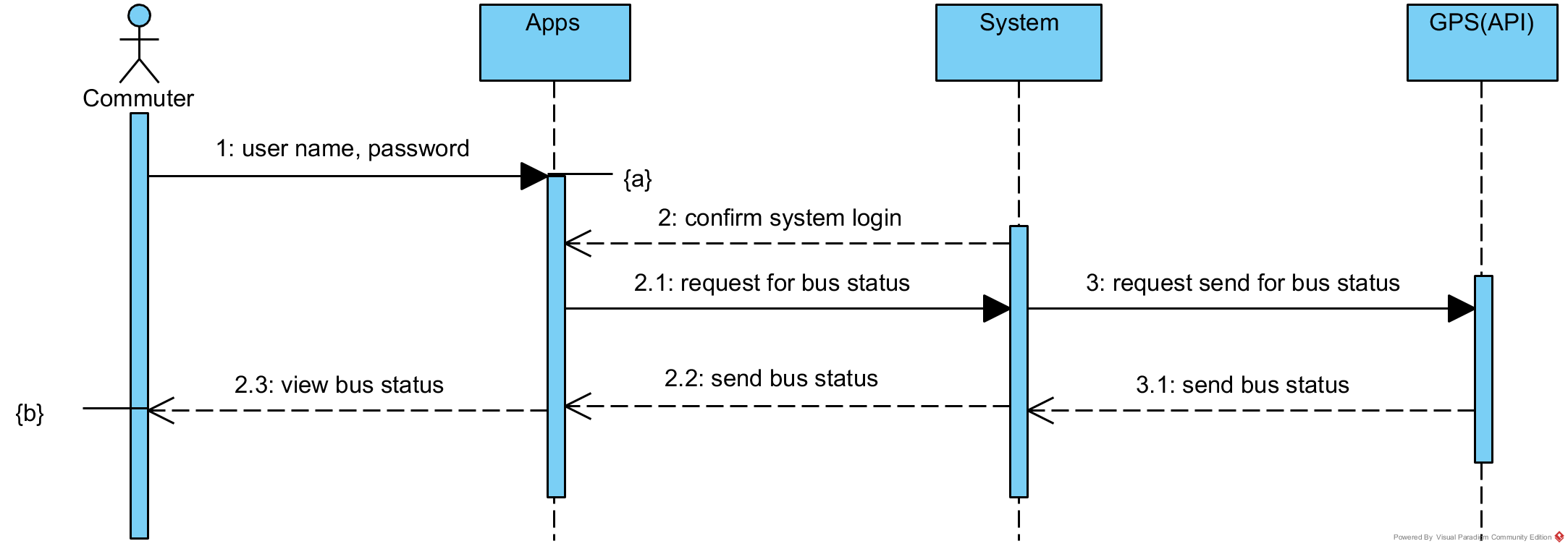
## Login



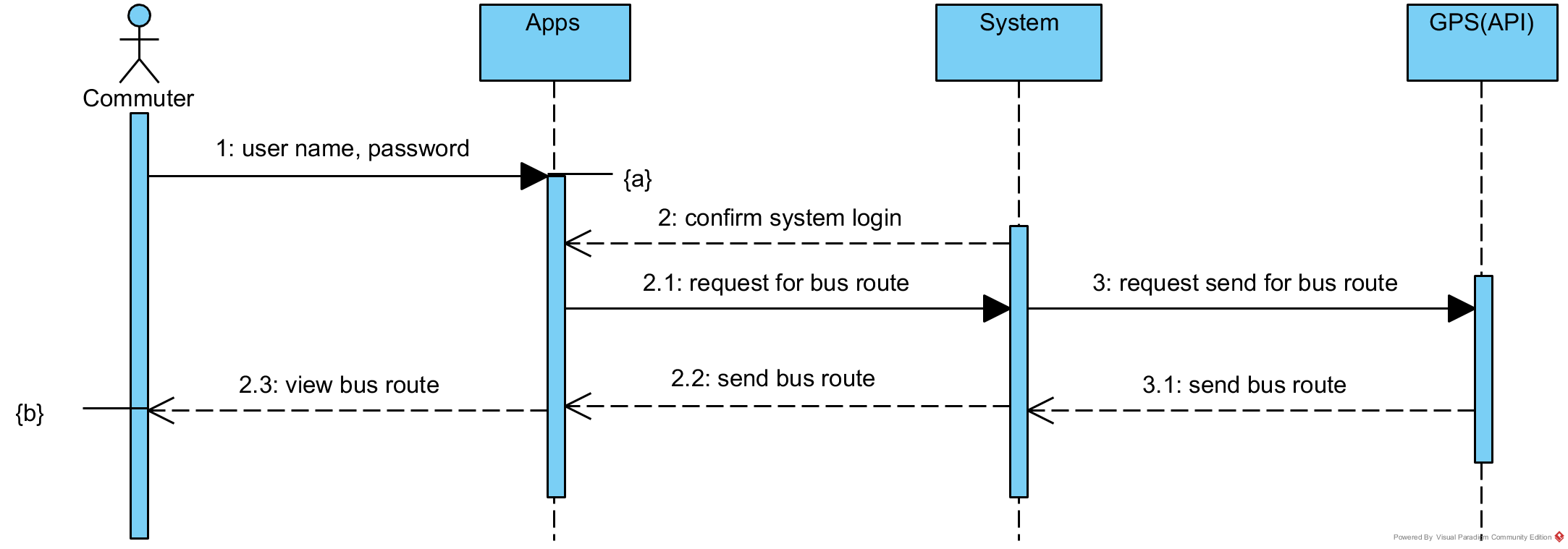
## Display Bus Location



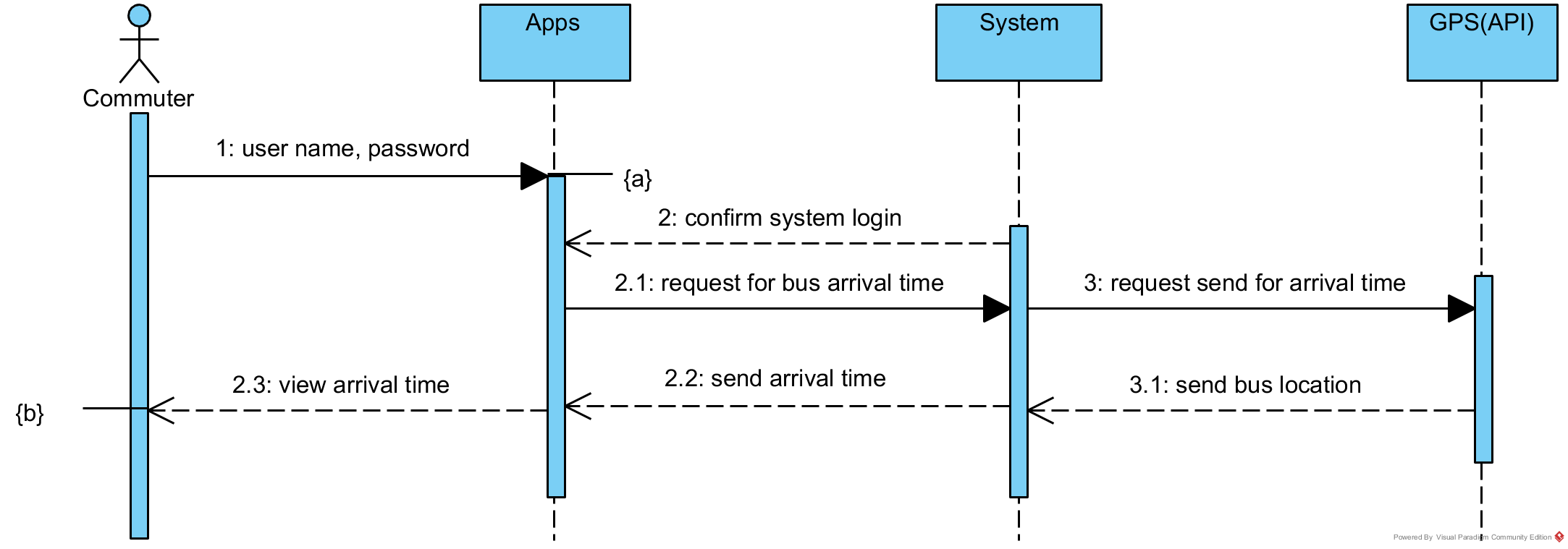
## Track Bus



## Display Bus Route

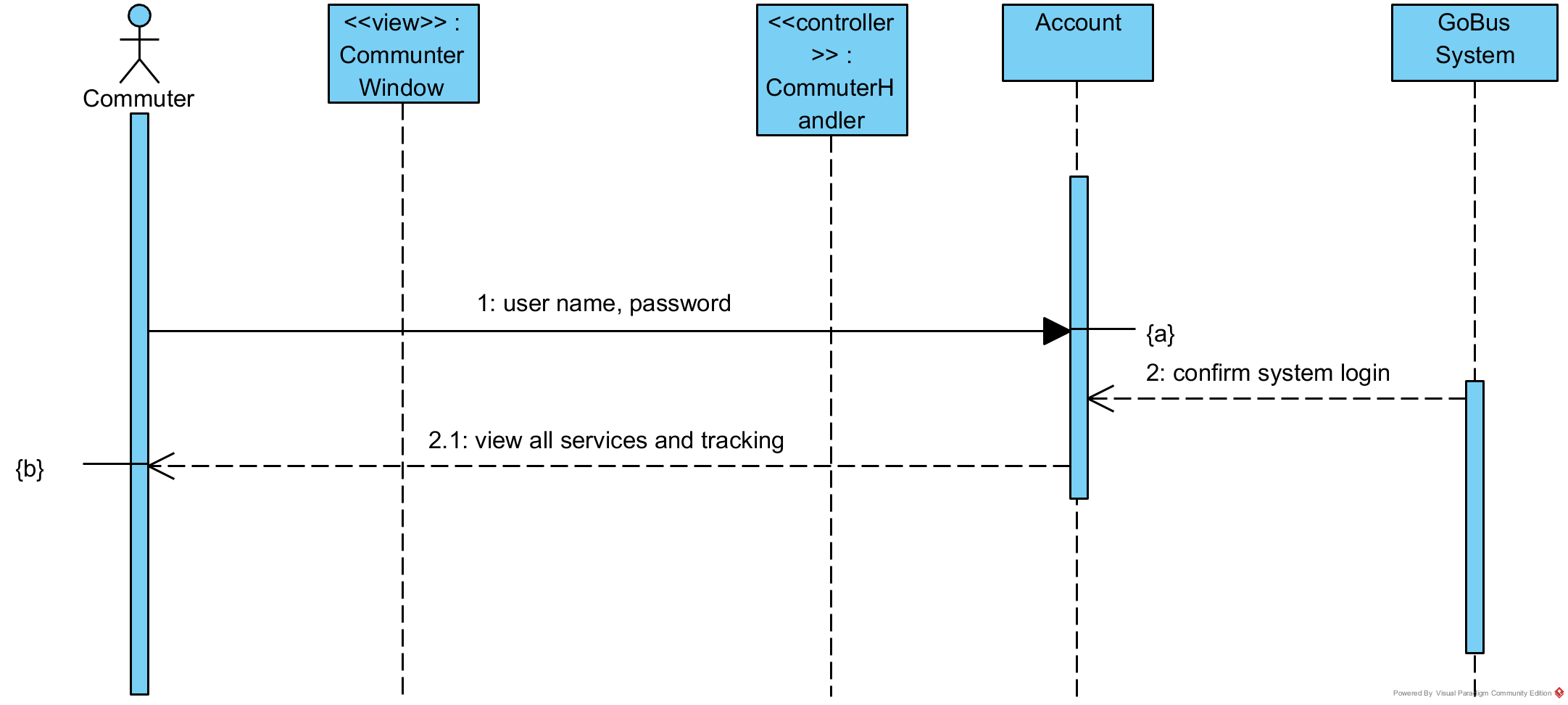


## Display Arrival Time



# Sequence Diagram

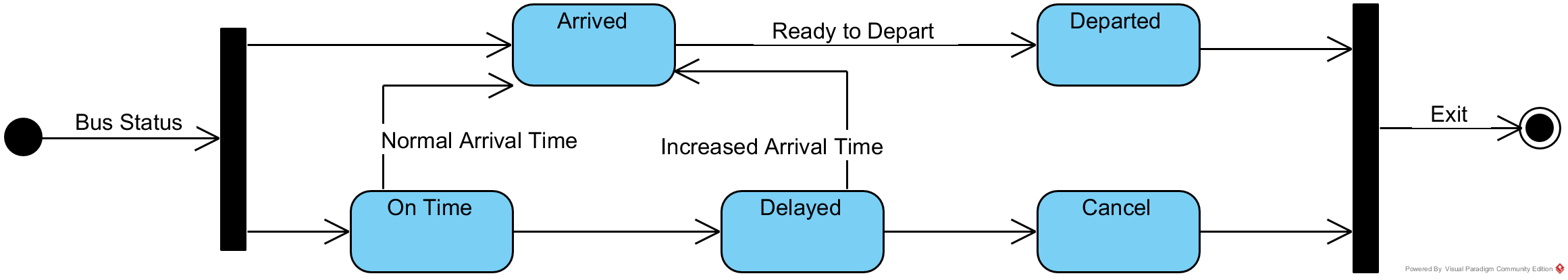
## Login



## Display Bus Location

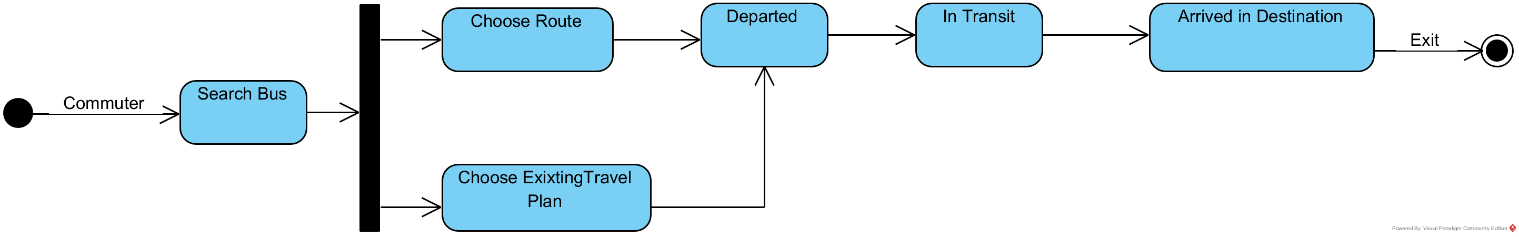
# State Machine Diagram

## Bus Status



|  |  |
| --- | --- |
| **State** | **Transition Causing Exit** |
| Ontime | Ontime arrival |
| Delayed | Delayed arrival |
| Cancelled | Bus cancelled |
| Arrived | Ready to depart |
| Departed | Ready to go for destination |

## Commuter Status



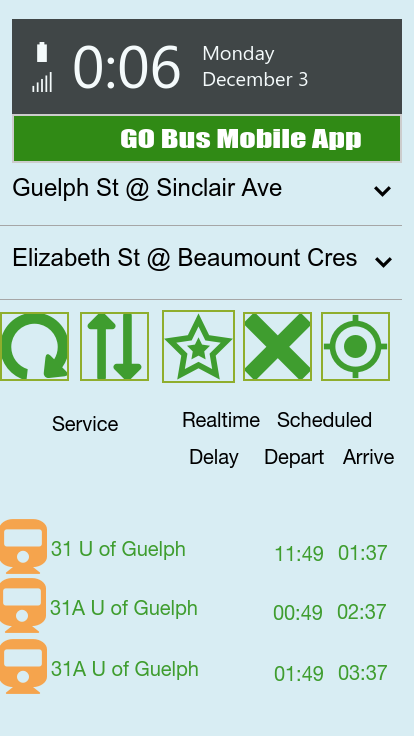
# Mock-up UI

A screenshot of a cell phone

Description generated with very high confidence

A close up of a map

Description generated with high confidence



# Component & Deployment Diagram

A screenshot of a cell phone

Description generated with very high confidence

# Technology tools used for Software Development

|  |  |  |
| --- | --- | --- |
| 3-Tier Architecture | Technology | Comments |
| Presentation | * HTML5 * CSS3 * JSP * JavaScript (ES6) * Java SE8 * Swift3 | User interface for the web view.  The user interface for the mobile application. |
| Application Server | * Java SE8 * Swift3 | Application Logic for the android and apple mobile applications. |
| Database Server | * Oracle 12C (SQL & PL/SQL) | Database and server-side processing. |

|  |  |  |
| --- | --- | --- |
| Category | Development Tool | Comments |
| Other | * Visual Studio 2015 EE * GitHub DVCS * SQL Developer 4.2 * IntelliJ * Visual Paradigm * JDBC 4.0 | Various Integrated Development Environments, Application Programming Interfaces and a Distributed Version Control System to facilitate efficient, iterative and incremental development sprints. |

# Skeletal Code

public class Commuter

{

private int commID;

private string commName;

private string commLocation;

private string email;

public void CreateCommuter()

{ //TODO- implement Commuter.CreateCommuter

throw new UnsupportedOperationException();

}

public void loginAccount()

{

throw new UnsupportedOperationException();

}

public void viewDisplayLocation()

{

throw new UnsupportedOperationException();

}

public void viewDisplayArrivalTime()

{

throw new UnsupportedOperationException();

}

public void viewDisplayRoute()

{

throw new UnsupportedOperationException();

}

public void viewDisplayNotification()

{

throw new UnsupportedOperationException();

}

public void addComments()

{

throw new UnsupportedOperationException();

}

public void viewRatings()

{

throw new UnsupportedOperationException();

}

}

public class Service

{

private int busNo;

private int routeID;

private string busLocation;

private int mapID;

private string longitude;

private string latitude;

private int commtID;

public void loginAccount()

{

throw new UnsupportedOperationException();

}

public void viewDisplayBusLocation()

{

throw new UnsupportedOperationException();

}

public void viewDisplayArrivalTime()

{

throw new UnsupportedOperationException();

}

public void viewDisplayRoute()

{

throw new UnsupportedOperationException();

}

public void viewDisplayNitification()

{

throw new UnsupportedOperationException();

}

}

# Gantt Chart