MIS 6308.002

SYSTEM ANALYSIS AND PROJECT MANAGEMENT

Group Project

Group No: 2

Group Members: Musham Chaturvedi

Shubham Dave

Sruti Seshadri

Sriram Prasath Subbiah Pandiarajan

Tanya Bhardwaj

Contents

1. Project Presentation YouTube Link

The link to team presenting the project: YouTube Link

2. Executive Summary

As regular users of the Comet Cruiser service, we analysed and proposed improvements to the existing system. We applied the basic understanding of System Analysis and Project Management to decipher the system in terms of System Development Life Cycle.

The current system provided all the data points related to Comet Cruisers to Transportation Department of UT Dallas as well as Public Information Office. However, it was not directly available to one of the most important stakeholders, commuters. Commuters (majorly consisting of students of UT Dallas) need to have the real-time data of the bus and its movements every day and yet it was almost impossible for them to know where is their bus in real-time or if there is enough space available for them or if they need to catch the next one and so on.

This project is intended to solve these problems present in the current system of Comet Cruisers for UT Dallas. The new system proposed uses IOT and Web 2.0 to collect data from different sources like students, bus system, etc. After collecting the data, it processes it and disseminates quality information to bus commuters in real-time using which they can make informed decisions about their commute. Giving this information to commuters will make system more efficient in terms of serving its customers as well as it will also help in getting data for improving system by use of data analytics.

The project deliverables are:

1. Request for extra bus during peak hours.
2. Live tracking of the buses and average wait time of arrival for commuters.
3. Real-time bus & bike stand occupancy data for commuters

Internet of Things Deliverables:

1. Component health data for buses
2. Regulating temperature inside the bus automatically

These are the features which can be leveraged using crowd-sourced data and IOT to improve the efficacy of the system and help commuters as well as UTD Transport Department.

3. Problem Statement

In the existing system, data regarding number of passengers and time of arrival at each stop for all trips of all different routes. However, this data was not being collected in real-time and due to it, it was difficult for the UTD Transportation Department to disseminate the information to commuters. Without knowledge about current location or current occupancy of the bus, the commuters had to wait at the bus stop.

Inadvertently, the commuters might be too late for the bus (if the bus was early than scheduled arrival time at a stop) and missed it or were too early and had to wait for it till the time it arrived (which could also be after the scheduled time in case of delays).

Also, during peak hours, the stops that come first in the route may fill up the bus and the commuters at stops later in the route would have to wait for another bus to come by. Even, the next bus could be filled up during peak hours, which would further waste time of commuters and all this just cause, the stop is positioned later in the scheduled route.

4. Business Requirements

Due to the problems as mentioned above, this new system is being proposed which would help solve following business requirements:

- Tracking of bus in real-time to know if bus is delayed due to any traffic delays or breakdowns
- Providing the occupancy information of buses to commuters
- Collecting commuter data for different routes & timings and after collecting enough data over time, using it to increase the efficiency of buses by changes in route
- Managing occupancy in peak hours by use of extra bus on special routes
- Maintaining temperature inside the bus uniform without manual intervention to make the commute a better experience for commuters
- Collecting vehicle health data for different components on regular basis to track and alert UTD Transportation Department for low health before the bus breaks down

5. Objectives

Objectives to be achieved by the new proposed system are as follows:

- Display average wait time of arrival for bus in real-time
- Display current position of the bus in real-time
- Display current occupancy of the bus in real-time
- Improve the availability of buses in timely manner during peak hours
- Improve the experience for users by maintaining uniform temperature inside the bus
- Improve the efficiency of Comet Cruiser system by managing occupancy and stops based on commuter data
- Reduce the number of breakdowns by alert system for low health for different components of the bus

6. Scope

The scope of the new proposed system can be defined as follows:

- The proposed system would collect the data of commuters entered or exited at a stop and provide timely update for current occupancy & location of the bus.
- The proposed system would allow the driver to request for extra bus during peak timings (if the current bus is at its capacity)
- The proposed system would collect the data for commuters utilizing different routes of Comet Cruiser service and would provide better estimate of peak timings as well as modifications to routes required, if any.
- The proposed system would reduce the number of break downs by alerting the relevant authorities in cases of low health of bus components.

7. Data Collection

A public information request was sent to UTD Transportation Department to collect the data about Comet Cruiser service. Apart from this, data was also collected by conducting interview of UTD Comet Cruiser drivers and commuting students.

Based on both these activities, we found the following data points:

- How many drivers are available for Comet Cruisers?
    - 13 drivers
- What are the shifts for each of the drivers?
    - 2 shifts
        - 6:00 a.m. to 3:00 p.m.
        - 2:00 p.m. to 12:00 a.m.
- How many total buses are in circulation for Comet Cruisers?
    - 7 buses in circulation
- How many spare buses are available for Comet Cruisers?
    - 3 spare buses
- What are the number of trips for each bus per week?
    - On average 100 to 112 loops per bus weekly
- What is the average no. of passengers per route per week?
    - On average: 35,000 per week
- What is the fuel type for each bus?
    - Diesel fuel
- How much fuel is required by each bus per month?
    - This information is not available. Fuel consumption is kept by the vendor that runs the 883 services.
- How many breakdowns occurred for each of the Comet Cruiser routes?
    - 2 to 3 per week
- What is the distance between stops of each of the Comet Cruiser routes?
    - It varies for each stop.
- What is the average maximum capacity in terms of passengers for each bus
    - 60 to 65 passengers

8. Functional Specifications

- Proposed functionality will provide the live tracking facility for UTD Comet Cruiser for all the routes and average waiting time using an app.
- Proposed functionality will provide real-time information of bus and bike stand occupancy using IOT.
- Proposed functionality will provide alerts to driver and UTD Transport Department for low health metric of different components of the bus.
- Proposed functionality will automatically adjust the temperature inside the bus by using collecting data using temperature sensors inside and outside the bus.
- Proposed functionality will allow the driver to request the transportation department to avail a spare bus during rush timings.

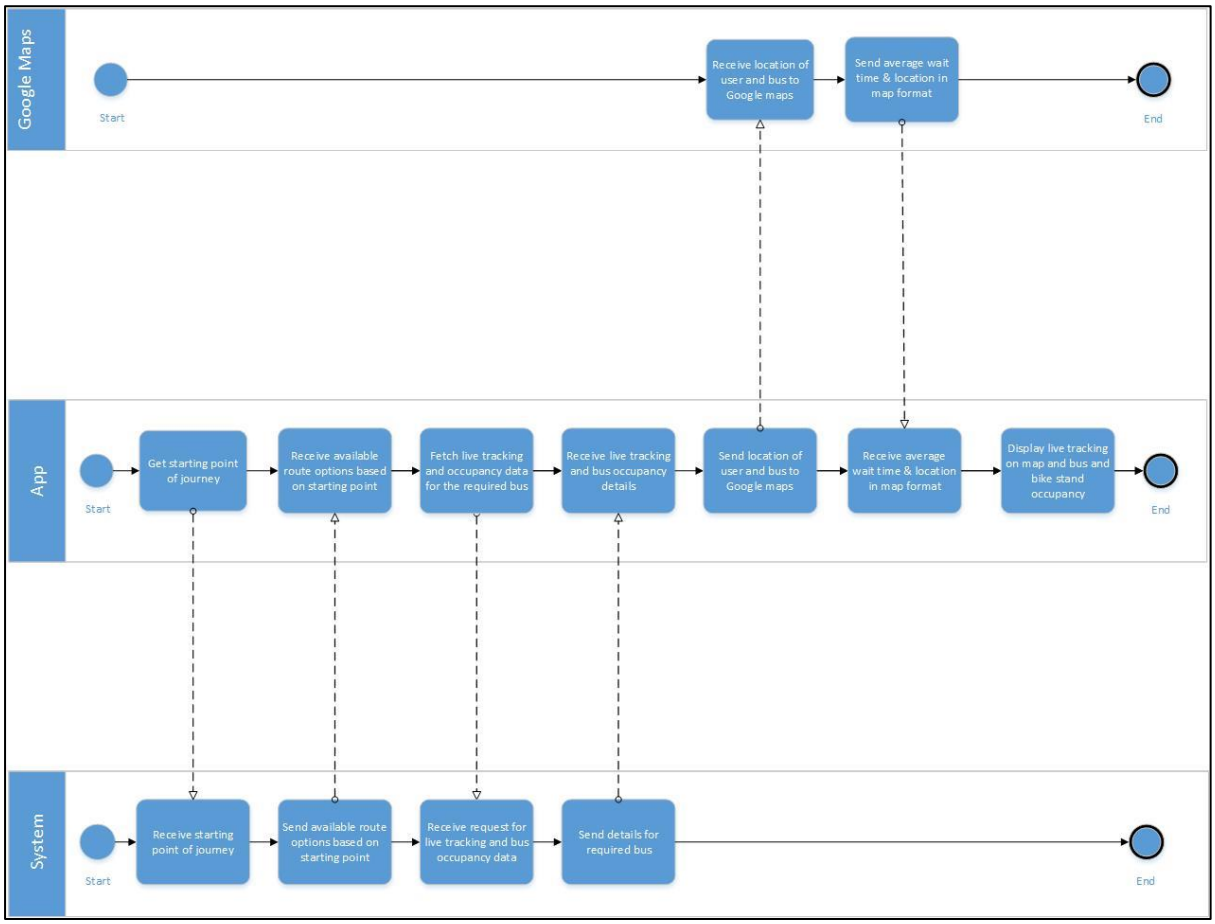
9. Proposed Solution

Using IOT Sensors and App for Comet Cruiser commuters, following functionalities are proposed:

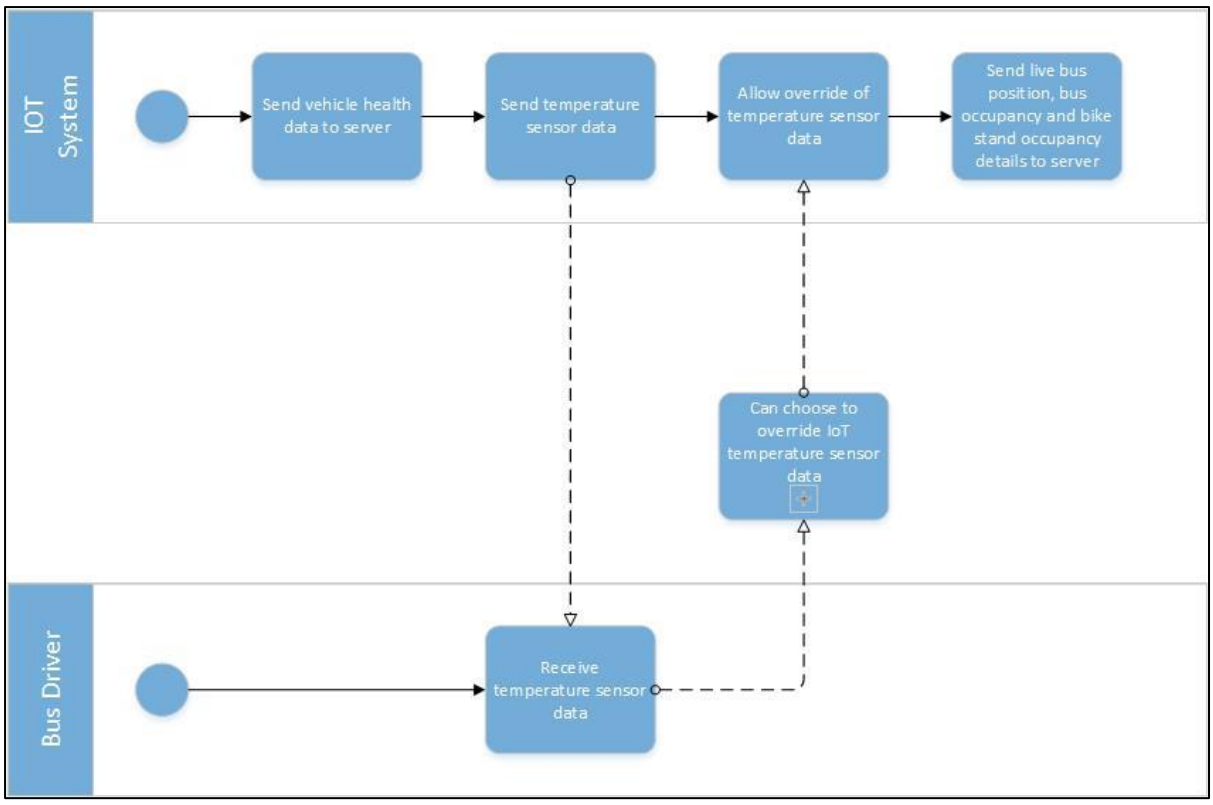
- **Live tracking & average wait time of bus**: Using GPS sensor for the buses, exact location of buses running on each route will be available. The location of each bus will be available to commuters via the app. The users will be given an option to select the start point for their journey and the route they want to travel on (since for hubs like UTD or McCallum community stop, multiple buses could be boarded). On selection of these options, the app will display the current location of the bus on a map interface using Google Maps API. Also, based on the starting point of the journey and current location of the bus, average wait time will be displayed by leveraging Google Maps API.
- **Current bus and bike stand occupancy**: Using sensors at entry/exit points of the bus, bus occupancy details will be captured. Also, a sensor will be kept for bike stand which will capture data of how many stands are in use for the bus. This information will be relayed to the backend servers by using IOT device after each stop. Using it and the total count of commuters on previous stop, the occupancy of the bus & bike stand will be calculated and displayed on the app for commuters by selecting respective route.

- **Bus health alerts**: Health of different components of the bus will be tracked by using sensors. These sensors will relay the health metric information to IOT device and will be processed in the backend servers. Backend servers will contain health metric thresholds for each component of different models of the buses. During processing, if health metric value for one of the component of the bus will go below the threshold, bus driver and UTD Transportation Department will be notified via alerts. This will help in bringing in the bus before it breaks down and sending extra bus to the location from where the bus (with low health) returned.
- **Automatic temperature balance**: Sensors will capture data of temperature within the bus and outside the bus and based on it, temperature of the bus will be balanced. This system could be overridden by the bus driver, in case of malfunction of the sensors, to make sure that the bus doesn't heat up or cool down too much.

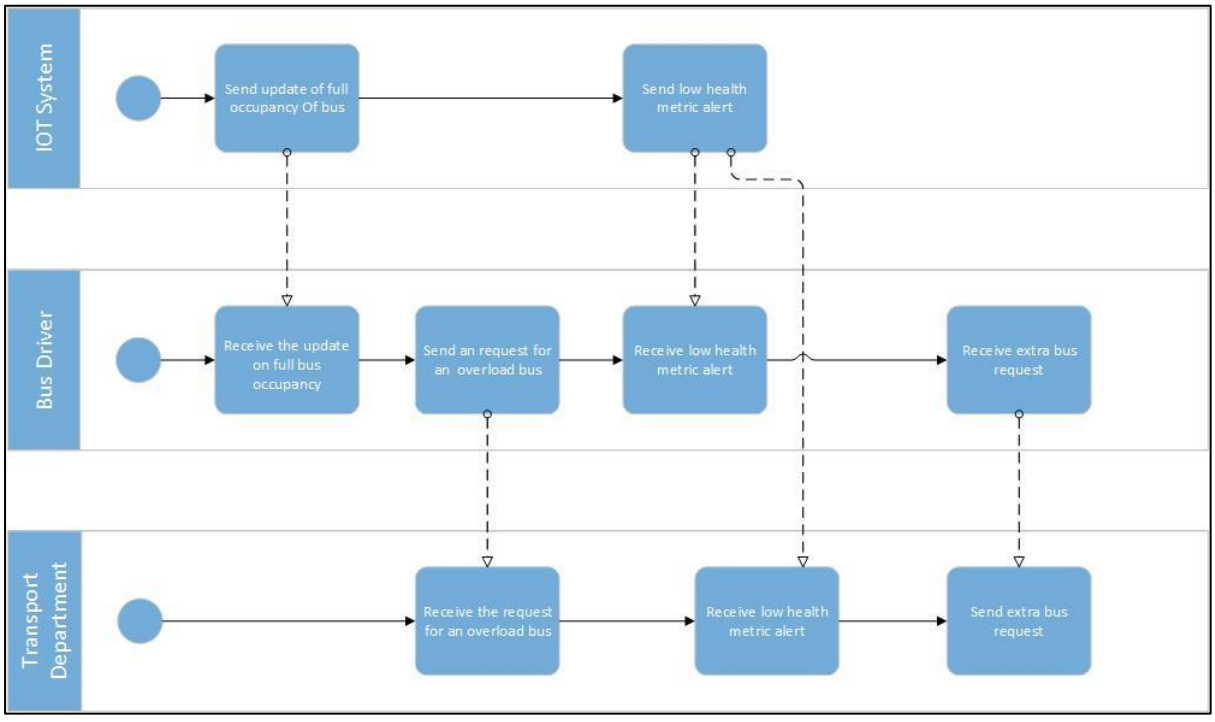
10. Business Process Model Notation (BPMN) Diagrams



BPMN diagram for displaying live tracking and bus occupancy details in app

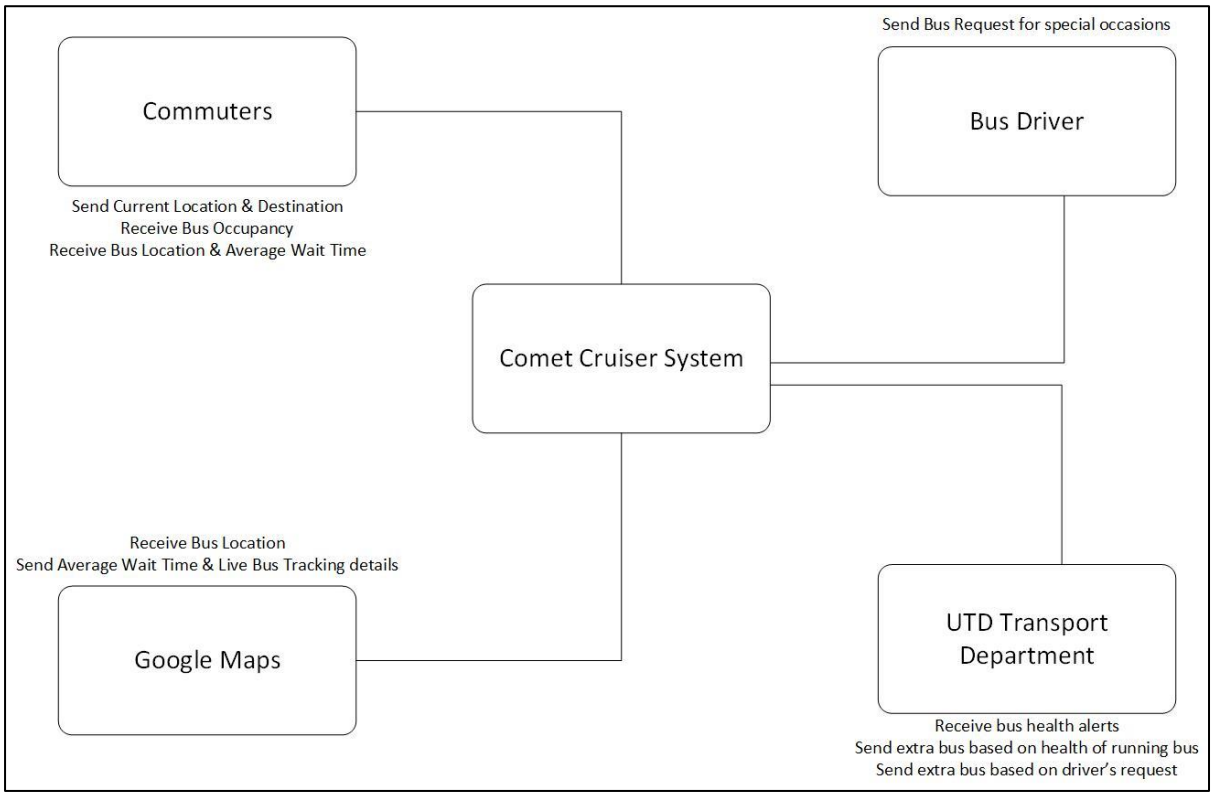


BPMN diagram for capturing and storing data from various sensors



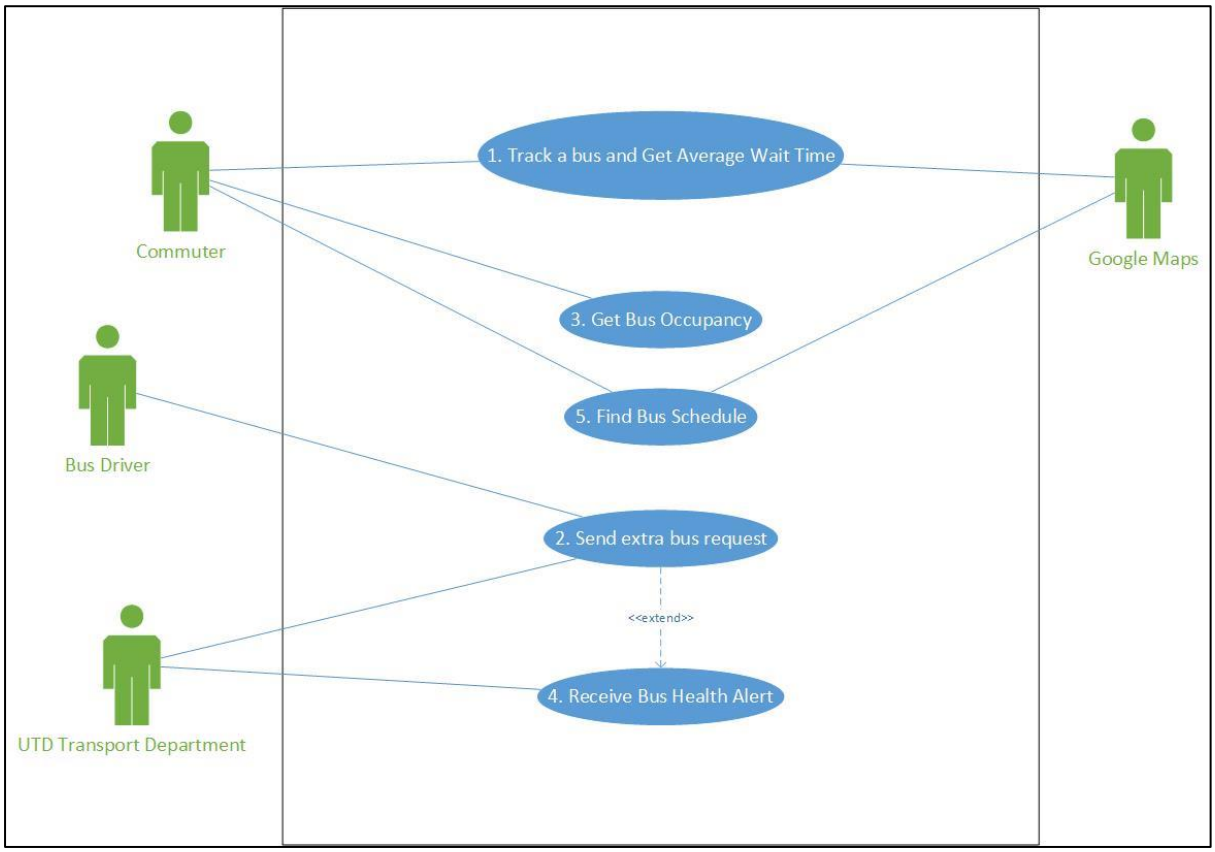
BPMN diagram for capturing full occupancy or low health alert

11. Context Diagram



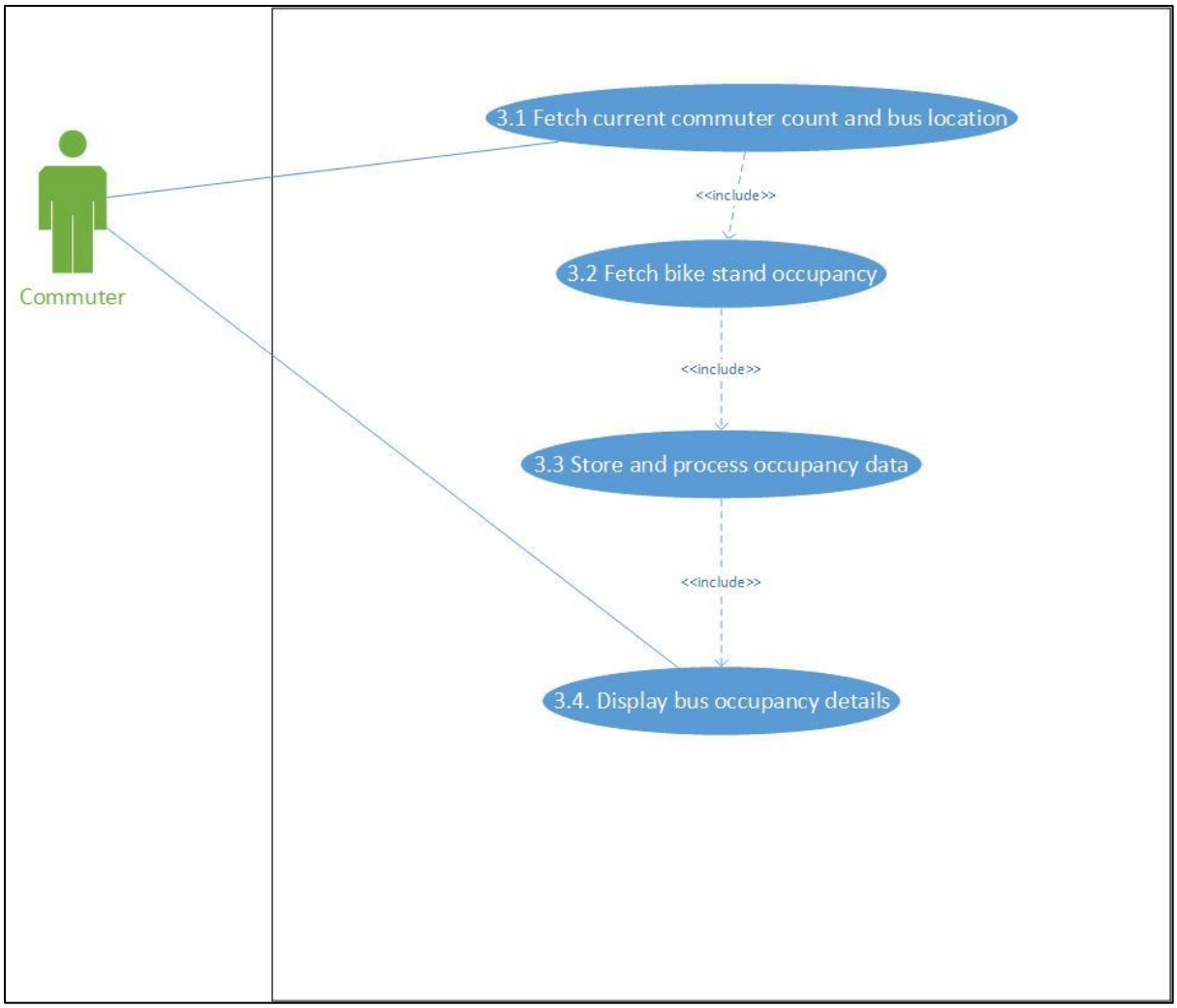
Context Diagram

12. Use Case Diagram

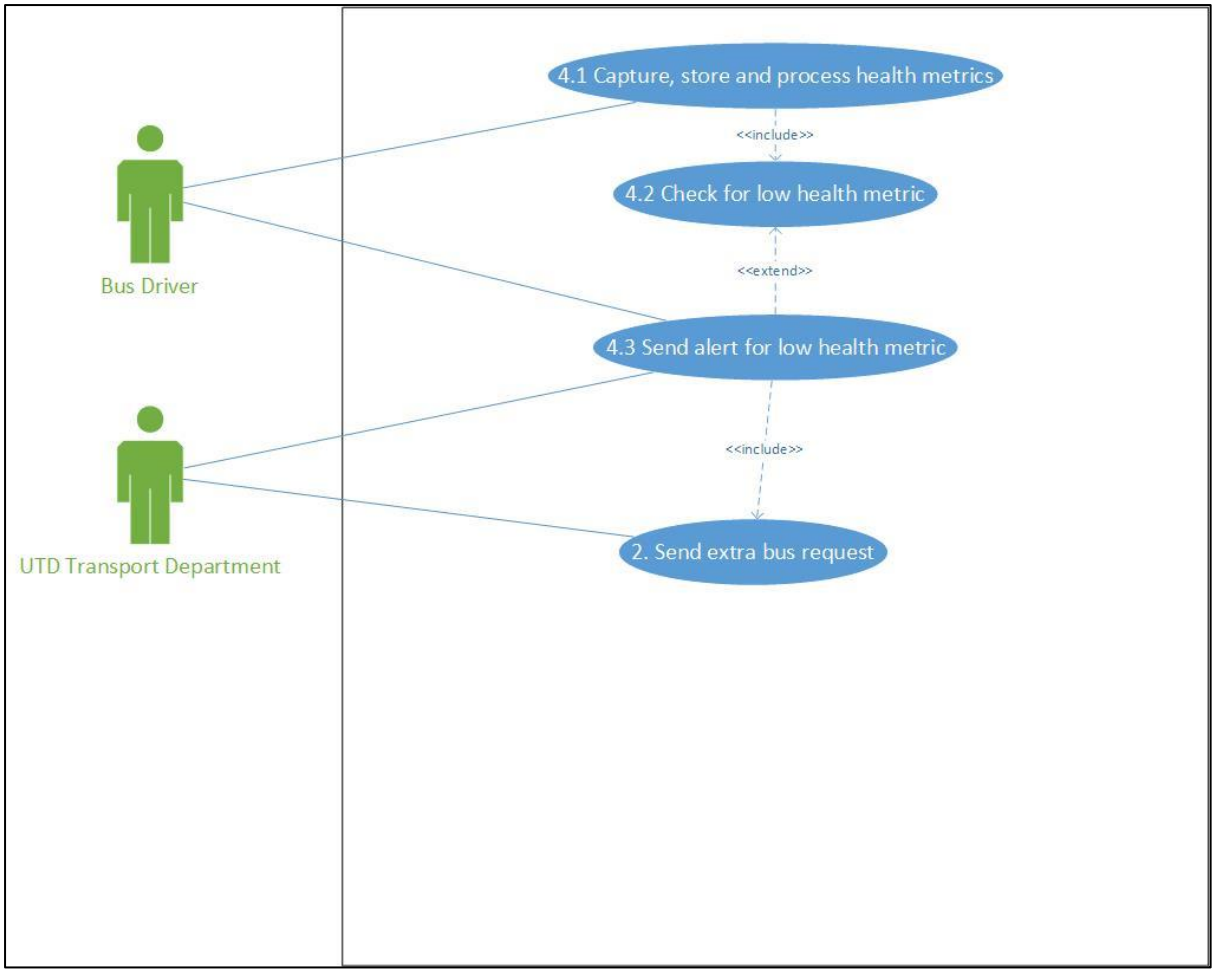


Use Case Diagram – Level 1

Use Case Diagram for “Track Bus and Average Wait time” - Level 2



Use Case Diagram for “Get Bus Occupancy” - Level 2



Use Case Diagram for “Receive Bus Health Alert” - Level 2

Use Case Diagram for “Find Bus Schedule” - Level 2

13. Use Case Descriptions

Use Case Name	Track Bus and Get Average Wait Time
Use Case ID	1
Primary Actors	Commuter and Google Maps
Stakeholders & Interests	Commuter – wants to know the current location of the bus and average wait time for bus to reach their location
Brief Description	This use case describes the process of live bus tracking and average wait time for commuter
Trigger	Commuter opens the app/website to know about bus's current location and average wait time
Trigger Type	External Trigger
Relationships	
Association	Commuter and Google Maps
Include	
Extend	
Generalization	
Normal Flow of Events	
1. Get *source* and *destination* details from the commuter. 2. Display all *route* options for the selected *source* and *destination*. 3. Get *route* selected by the commuter. 4. Based on the *route* selected, fetch *current location* and *map* of bus running on that *route*. 5. Get average *wait time* for *source* from Google Maps based on *current location* of bus and *destination* as user's selected *source*. 6. Display *bus tracking details* on app/website to the commuter.	
Sub Flows	
Alternate/Exception Flows	

Use Case Name	Get Bus Occupancy
Use Case ID	3
Primary Actors	Commuter
Stakeholders & Interests	Commuter – wants to know the amount of vacancy for seats as well as bike stand in the current bus for a route
Brief Description	This use case describes the process of getting bus seat and stand occupancy data
Trigger	Bus health becomes lower than safety threshold
Trigger Type	Internal Trigger
Relationships	
Association	Commuter
Include	
Extend	
Generalization	
Normal Flow of Events	
1. Get current *commuter count* based on commuters entered and exited on last stop and current *bus location*. 2. Get *bike stand occupancy*. 3. Store and process *occupancy data* in server. 4. Display bus and bike stand *occupancy data* in the app/website.	
Sub Flows	
Alternate/Exception Flows	

Use Case Name	Receive Bus Health Alert
Use Case ID	4
Primary Actors	Bus Driver and UTD Transport Department
Stakeholders & Interests	UTD Transport Department – wants to know if bus health is low and send extra bus before bus breaks down
Brief Description	This use case describes the process of getting alert if bus health is low
Trigger	Bus health becomes lower than safety threshold
Trigger Type	External Trigger
Relationships	
Association	Bus Driver and UTD Transport Department
Include	
Extend	
Generalization	
Normal Flow of Events	
1. Process and compare to check if any component's health is below *safety threshold* from backend. 2. Send *alert* to UTD Transport Department and Bus Driver to bring the bus in for maintenance. 3. Send extra bus to the *location* where the called-off bus was present.	
Sub Flows	
Alternate/Exception Flows	

Use Case Name	Find Bus Schedule
Use Case ID	5
Primary Actors	Commuter and Google Maps
Stakeholders & Interests	User – wants to get schedule of buses during their required time interval
Brief Description	This use case describes the process of finding bus schedule between given time constraints
Trigger	User clicks on "Find Bus Schedule" in app/website
Trigger Type	External Trigger
Relationships	
Association	Commuter and Google Maps
Include	
Extend	
Generalization	
Normal Flow of Events	
1. The commuter opens Comet Cruiser app/website and clicks on *"Find Bus Schedule"* functionality. 2. The commuter enters the *source*, *destination*, *start time* and *end time* for the bus schedule he/she needs. 3. Based on these inputs, the system finds the *bus route details*. 4. These *bus route details* are displayed on the app/website.	
Sub Flows	
Alternate/Exception Flows	
4a – Display "No bus is available for given time" if no buses are available for the given schedule.	

14. Data Dictionary

Use Case "*Track Bus and Get Average Wait Time*"

- Bus Stop = {Stop ID + Street Name + City + State + Zip Code}
- Source = {Bus Stop}
- Destination = {Bus Stop}
- Route= {Route Name + Route ID}
- Bus Location = Latitude + Longitude
- Map = {Map Objects} + Map Legend
- Bus Tracking Details = Bus Location + Map + Wait Time
- Wait Time = Data element storing average wait time in minutes
- Stop ID = Data element storing Stop ID of the route
- Street Name = Data element storing street names
- City = Data element storing city names
- State = Data element storing state codes
- Zip Code = Data element storing zip codes
- Latitude = Data element storing latitude for bus location
- Longitude = Data element storing longitude for bus location
- Map Objects = Data element storing the map image with buildings, roads, etc.
- Map Legend = Data element storing the bus image to show bus on the map

Use Case "*Get Bus Occupancy*"

- Commuter Count = #Commuter Entered + i
- Bus Occupancy = Data element calculated based on Last Occupancy Data and Commuter Count
- Bus Location = Latitude + Longitude
- Bike Stand Occupancy = [0|1|2]
- Occupancy Data = Bus Occupancy + Bike Stand Occupancy

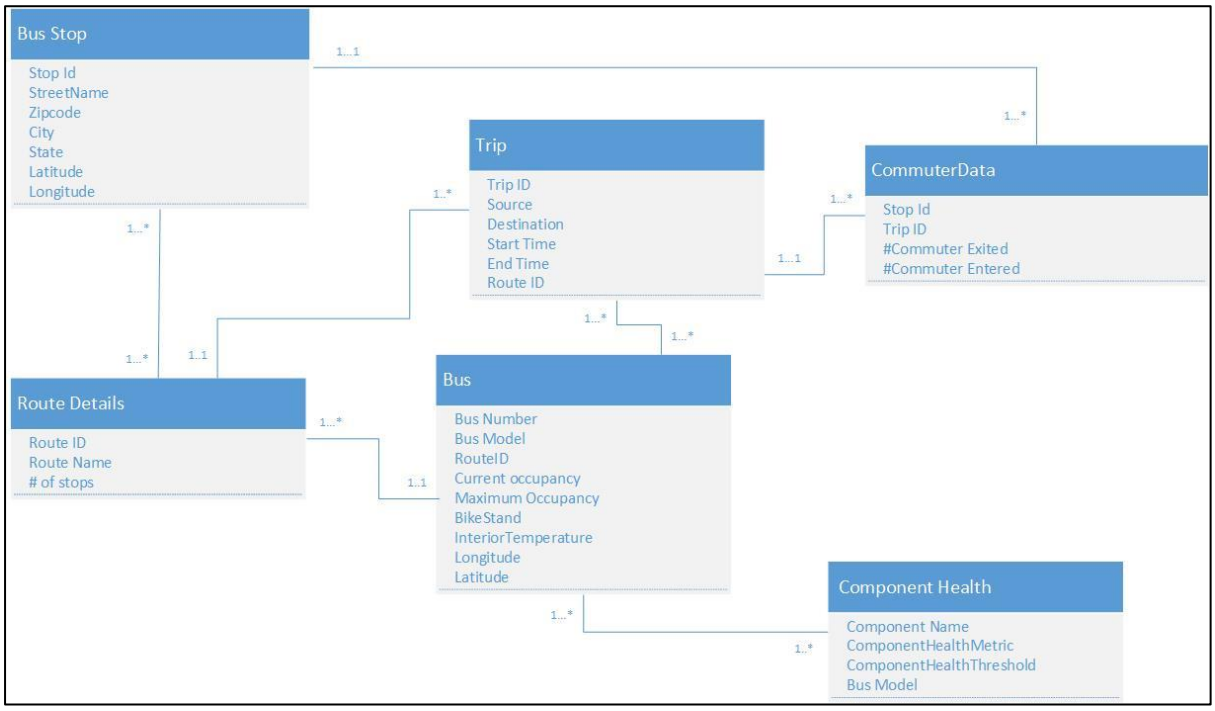
Use Case "*Receive Bus Health Alert*"

- Metric = Component Name + Metric Value
- Health Metrics = {Metric}
- Safety Threshold = {Metric}
- Alert = Metric + Safety Metric + Action
- Safety Metric = Metric
- Action = Data element storing action for bus driver to get bus to shed
- Bus Location = Latitude + Longitude

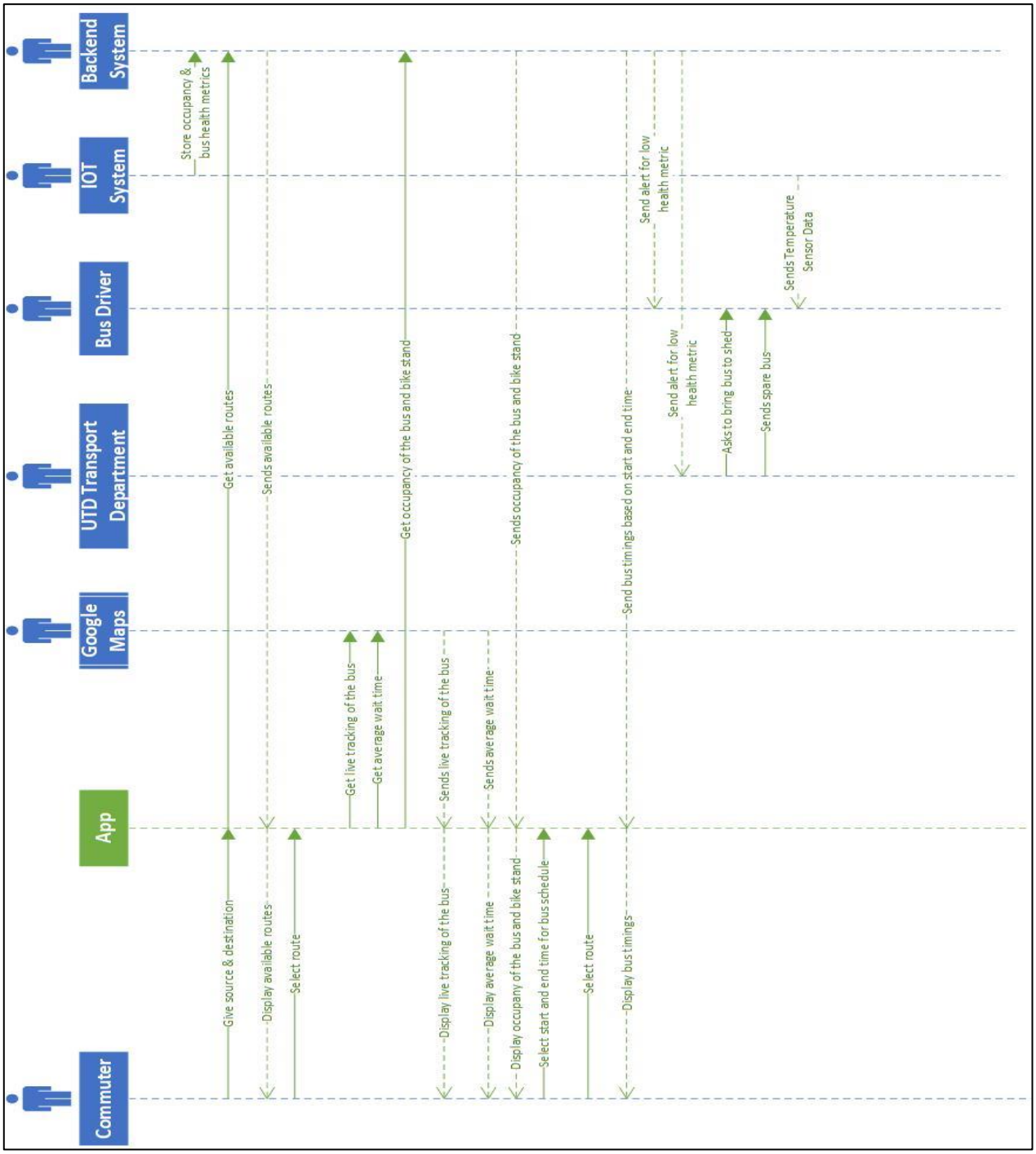
Use Case "*Find Bus Schedule*"

- Bus Stop = {Stop ID + Street Name + City + State + Zip Code}
- Source = {Bus Stop}
- Destination = {Bus Stop}
- Start Time = Data element storing start time for schedule to be searched for
- End Time = Data element storing end time for schedule to be searched for
- Route Details = Bus Route + {Start Time + End Time} + Route Name + Route ID
- Bus Route = {Map Objects}
- Map Objects = Data element storing the map with route and with buildings, roads, etc.

15. Class Diagram (without methods)



16. Sequence Diagram

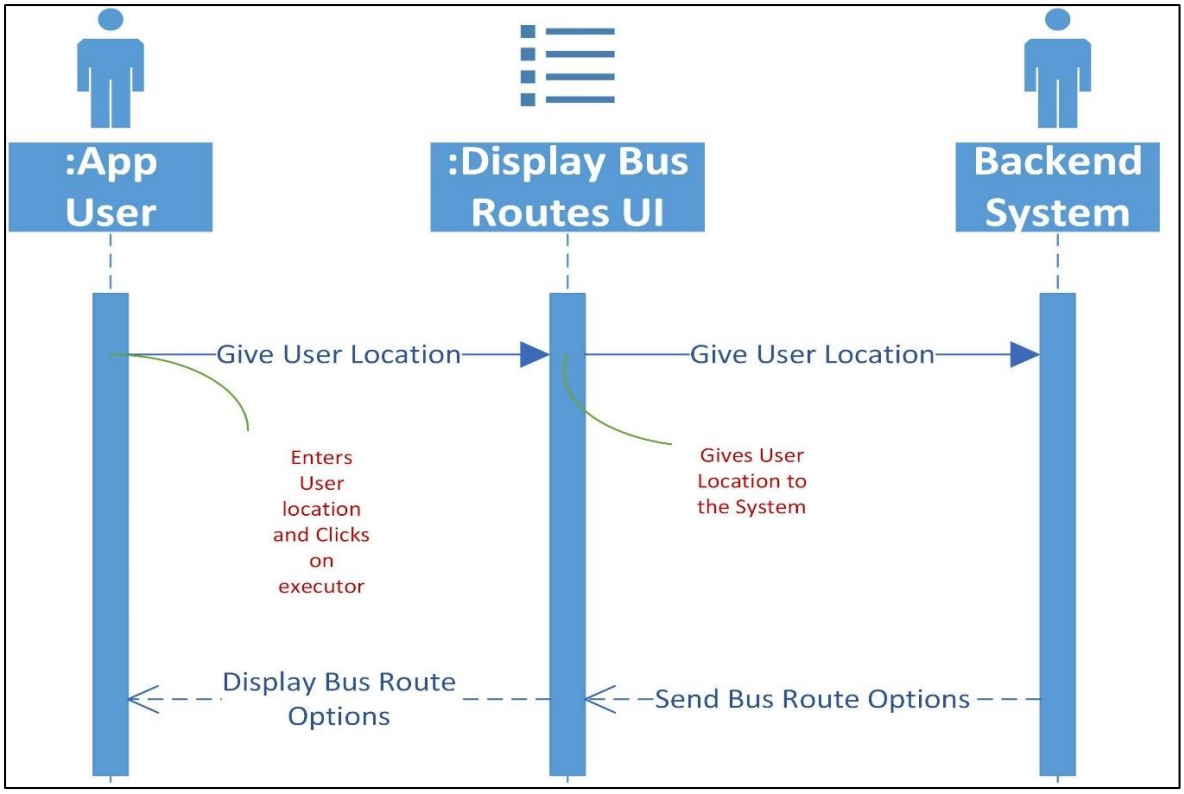


Sequence diagram for functionalities of the proposed system

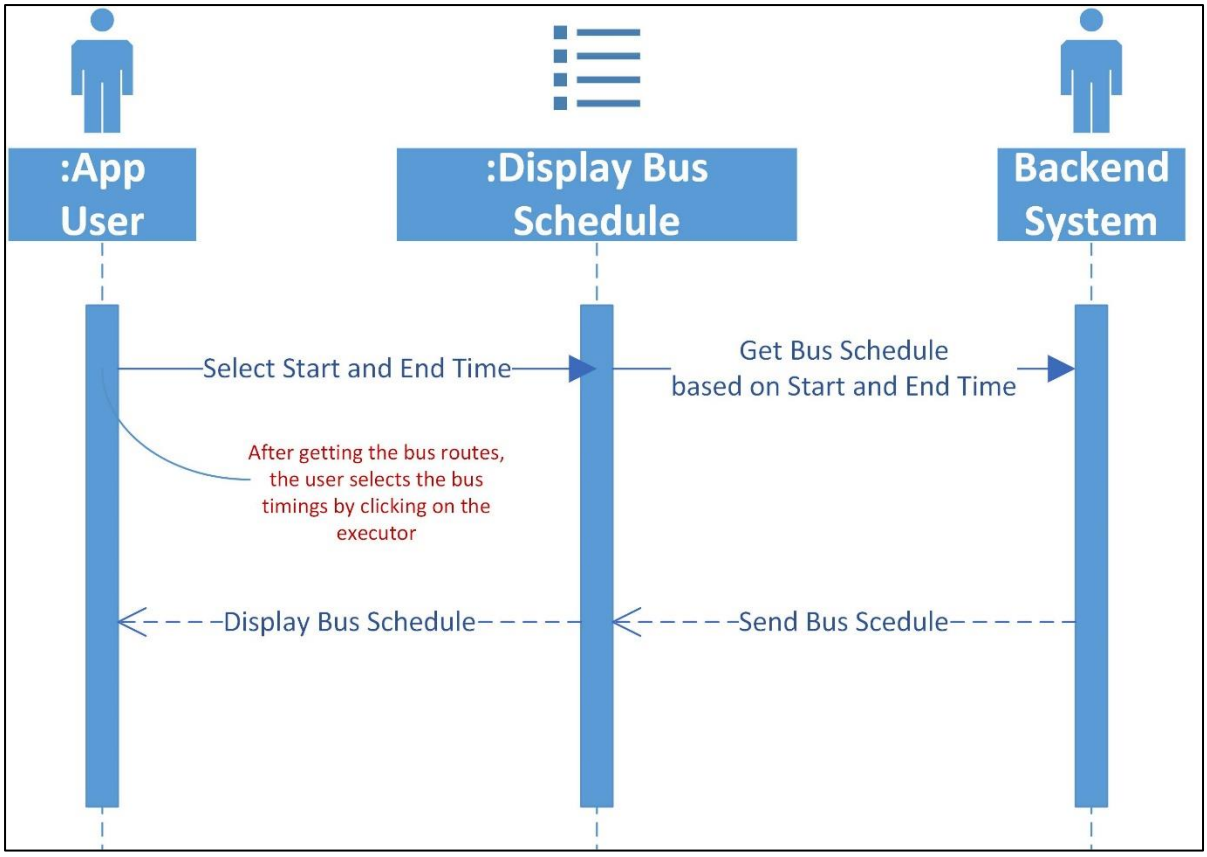
17. Interface Design

UI for Login of Comet Cruiser App

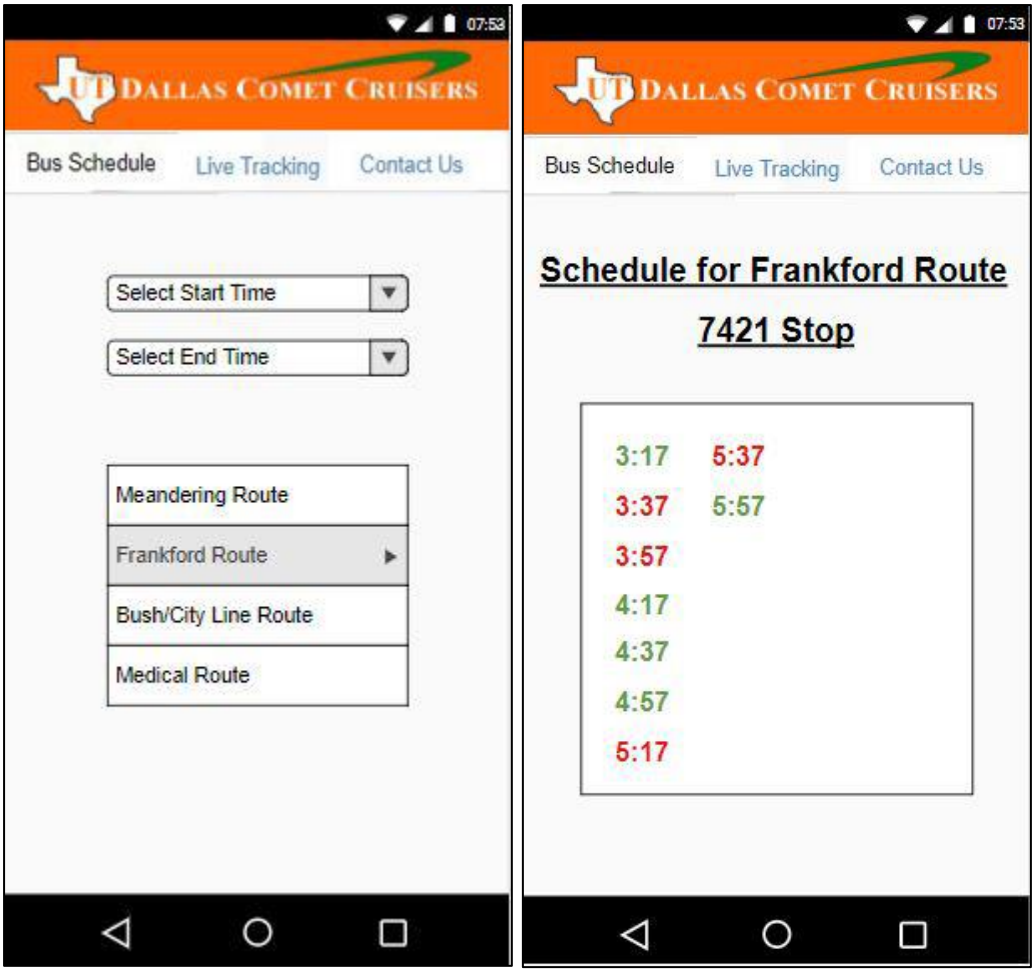
This login page is directly using existing API for login using UTD ID



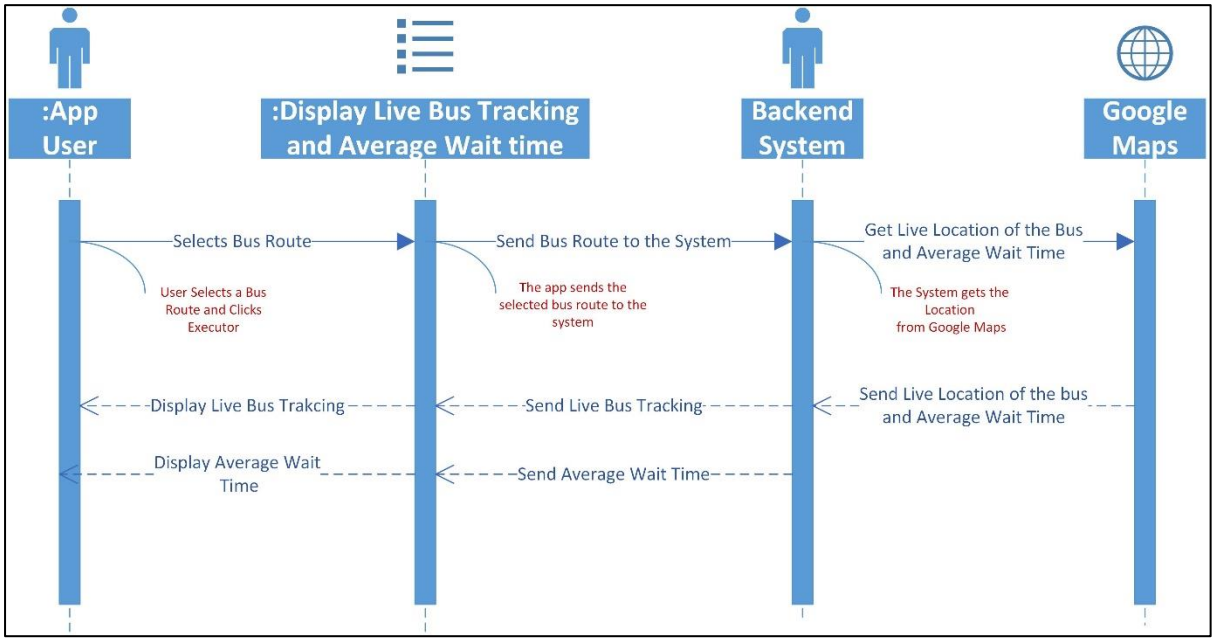
Display Bus Routes UI Sequence



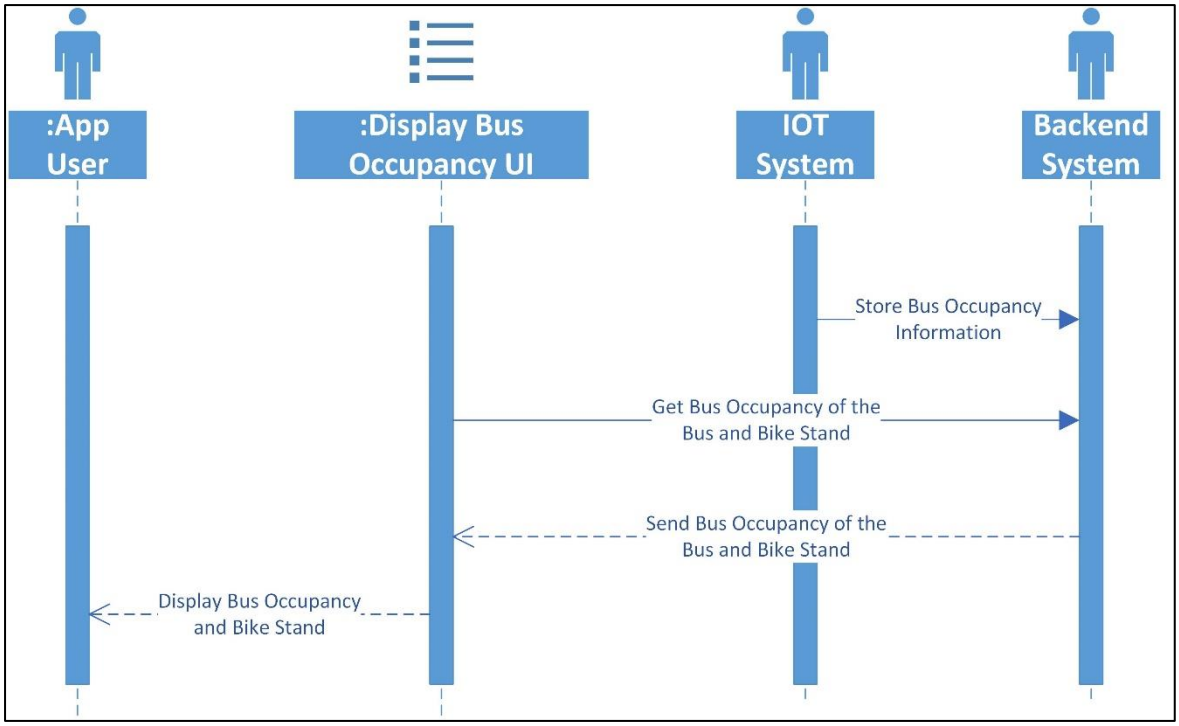
Display Bus Schedule UI Sequence



UI for Bus Schedule functionality



Display Live Tracking and Average Wait Time UI Sequence

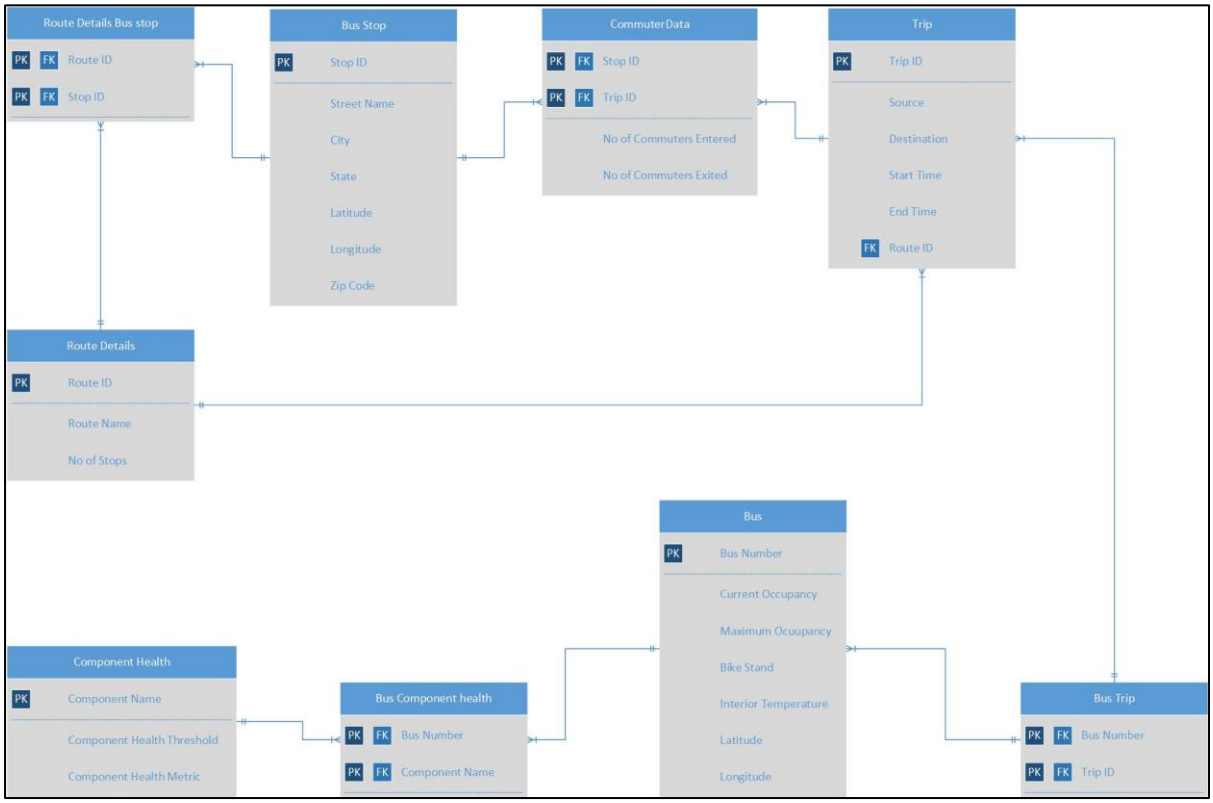


Display Bus Occupancy UI Sequence

UI for Live Tracking and Bus Occupancy functionality

18. Database Design

18.1 Entity Relationship Diagram



ER Diagram for the proposed system

18.2 Design of tables

- Bus Stop (Stop ID, Street Name, Zip code, City, State, Latitude, Longitude)
- Trip (Trip ID, Source, Destination, Start Time, End Time, Route ID)
- Route Details (Route ID, Route Name, # of stops)
- Route Details Bus Stop (Route ID, Stop ID)
- Bus (Bus Number, Route ID, Current Occupancy, Maximum Occupancy, Bike stand, Interior temperature, Longitude, Latitude)
- Bus Trip (Bus Number, Trip ID)
- Component Health (Component Name, Component Health metrics, Component Health Threshold, *Stop ID, Trip ID*)
- CommuterData (Stop ID, Trip ID, # of Commuter exited, # of Commuter entered)
- Bus Component Health (Bus Number, Component Name)

18.3 Database Constraints

- Bus Stop (Stop ID, Street Name, Zip code, City, State, Latitude, Longitude)
    a) Primary key – Stop ID
    b) Foreign Key – N/A
    c) Unique Constraint – Stop ID
    d) Not Null Constraint – Stop ID, Latitude, Longitude

- Trip (Trip ID, Source, Destination, Start Time, End Time, Route ID)
    a) Primary key – Trip ID
    b) Foreign Key – Route ID must exist in Table Route Details
    c) Unique Constraint – Trip ID
    d) Not Null Constraint- Trip ID, Route ID

- Route Details (Route ID, Route Name, # of stops)
    a) Primary key – Route ID
    b) Foreign Key – N/A
    c) Unique Constraint – Route ID
    d) Not Null Constraint - Route ID

- Route Details Bus Stop (Route ID, Stop ID)
    a) Primary key – Route ID Stop ID
    b) Foreign Key – N/A
    c) Unique Constraint – Route ID Stop ID
    d) Not Null Constraint – Route ID Stop ID

- Bus (Bus Number, Route ID, Current Occupancy, Maximum Occupancy, Bike stand, Interior temperature, Longitude, Latitude)
    a) Primary key – Bus Number
    b) Foreign Key – Route ID, must exist in Route Details table
    c) Unique Constraint – Bus Number
    d) Not Null Constraint – Bus Number, Route ID

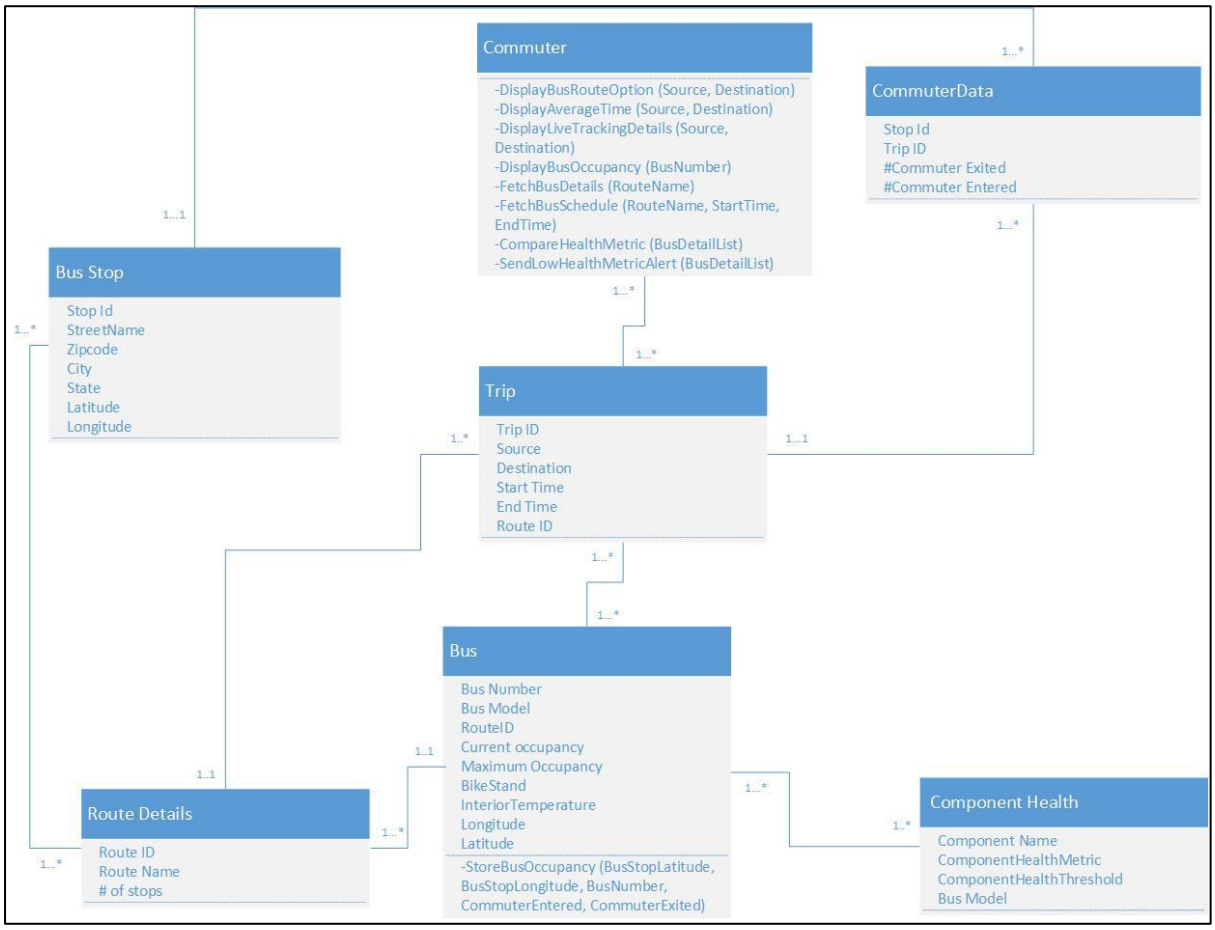
- Bus Trip (Bus Number, Trip ID)
    a) Primary key – Bus Number Trip ID
    b) Foreign Key – N/A
    c) Unique Constraint – Bus Number Trip ID
    d) Not Null Constraint – Bus Number Trip ID

- Component Health (Component Name, Component Health metrics, Component Health Threshold)
    a) Primary key – Component Name
    b) Foreign Key – Stop ID, Trip ID must exist in Bus Stop table and Trip table respectively.
    c) Unique Constraint – Component Name
    d) Not Null Constraint – Component Name

- Bus Component Health (Bus Number, Component Name)
    a) Primary key – Bus Number Component Name
    b) Foreign Key – N/A
    c) Unique Constraint – Bus Number Component Name
    d) Not Null Constraint – Bus Number Component Name

- CommuterData (Stop ID, Trip ID, # of Commuter exited, # of Commuter entered)
    a) Primary key – Stop ID Trip ID
    b) Foreign Key – Stop ID Trip ID must exist in Bus Stop table and Trip table respectively
    c) Unique Constraint – Stop ID, Trip ID
    d) Not Null Constraint – Stop ID, Trip ID

19. Class Diagram (with methods)



20. Software Design

Signature:

Method name	Display bus route options
Class name	
ID	
Clients (Consumers)	Commuters
Associated use cases	Track Bus and Get Average Wait Time
Description of responsibilities	Fetch the options of routes based on source and destination
Arguments received	Source and Destination for the user
Type of value returned	List of Route Names
Pre-conditions	Both source and destination value can't be blank
Post-conditions	If no route names are available, it will return an empty list. Handling needs to be done to display "No routes available" for such scenarios. Otherwise, display the list of route names as clickable text boxes.

Logic:

- CREATE empty list
- FETCH Stop IDs for Source and Destination
- FETCH Route IDs from Route Stop Details table where Route ID has Stop IDs of source and destination both.
- IF (0 Route IDs are obtained in above step)
    - RETURN empty list
- ELSE
    - FETCH Route Name based on Route IDs obtained in above step and ADD them to the empty list
    - ADD Route Names obtained in above step to the empty list
    - RETURN list

Signature:

Method name	Display average wait time
Class name	
ID	
Clients (Consumers)	Commuters
Associated use cases	Track Bus and Get Average Wait Time
Description of responsibilities	Fetch the average wait time based on source and destination
Arguments received	Source (Current location of bus) and Destination (Current location of user)
Type of value returned	Time in minutes
Pre-conditions	Both source and destination value can't be blank
Post-conditions	

Logic:

- CONNECT to Google Maps API and send Source and Destination as parameters
- CONVERT JSON response into minutes format
- RETURN value obtained in above step

Signature:

Method name	Display live bus tracking
Class name	
ID	
Clients (Consumers)	Commuters
Associated use cases	Track Bus and Get Average Wait Time
Description of responsibilities	Display the bus location on map with respect to the user's location and route of bus to reach the user
Arguments received	Source (Current location of bus) and Destination (Current location of user)
Type of value returned	Google Map graphic format
Pre-conditions	Both source and destination value can't be blank
Post-conditions	

Logic:

- CONNECT to Google Maps API and send Source and Destination as parameters
- CONVERT JSON response into Google Map graphic format with source displayed as bus icon and destination displayed as person icon and path connected between them as route
- RETURN value obtained in above step

Signature:

Method name	Fetch bus details
Class name	
ID	
Clients (Consumers)	Commuters
Associated use cases	Track Bus and Get Average Wait Time
Description of responsibilities	Fetch bus details running on the route selected by the user
Arguments received	Route Name
Type of value returned	Composite value made of average time in minutes and live bus tracking as a map
Pre-conditions	Route name can't be blank
Post-conditions	If empty value is returned, handling needs to be done to display "No buses running on selected route" for such scenarios. Otherwise, display the average time in minutes and map of bus route.

Logic:

- FETCH bus doing trip on the route selected by user
- IF (no bus is doing trip)
    - RETURN empty value
- ELSE
    - STORE average wait time in minutes by calling *"Display average wait time"* method
    - STORE bus tracking details in map format by calling *"Display live bus tracking"* method
    - CREATE composite of average wait time and bus tracking details
    - RETURN composite value

Signature:

Method name	Store bus occupancy
Class name	
ID	
Clients (Consumers)	UTD Transport Department
Associated use cases	Get Bus Occupancy
Description of responsibilities	Based on no. of commuters entered and exited on current stop and last commuter total, store and calculate bus occupancy
Arguments received	Bus stop latitude & longitude, bus number, no. of commuters entered the bus and no. of commuters exited the bus
Type of value returned	Composite of current occupancy and maximum occupancy of the bus and current occupancy of bike stand as integers
Pre-conditions	All arguments can't be blank
Post-conditions	

Logic:

- FETCH bus stop ID based on latitude & longitude of the bus stop
- FETCH maximum bus occupancy of the bus based on bus number
- FETCH commuter total for last stop based on bus number and bus stop ID
- FETCH no. of bikes on bike stand using bike stand sensor
- CALCULATE current occupancy of the bus by adding commuter total with no. of commuters that entered the bus and subtracting commuters that exited the bus
- STORE current occupancy

Signature:

Method name	Display bus occupancy
Class name	
ID	
Clients (Consumers)	Commuters
Associated use cases	Get Bus Occupancy
Description of responsibilities	Based on bus number, display bus occupancy
Arguments received	Bus number
Type of value returned	Composite of current occupancy and maximum occupancy of the bus and current occupancy of bike stand as integers
Pre-conditions	Bus number can't be blank
Post-conditions	

Logic:

- RETURN composite value of maximum occupancy and current occupancy of the bus and current occupancy of the bike stand (out of maximum of 2) based on bus number

Signature:

Method name	Fetch bus schedule
Class name	
ID	
Clients (Consumers)	Commuters
Associated use cases	Find Bus Schedule
Description of responsibilities	Fetch bus schedule details based on the route & start and end time selected by the user
Arguments received	Route name, schedule start time and schedule end time
Type of value returned	List of bus timings
Pre-conditions	All parameters can't be blank
Post-conditions	If empty list is returned, handling needs to be done to display "No buses running for selected schedule" for such scenarios. Otherwise, display the various timings of the bus route.

Logic:

- CREATE empty list
- FETCH bus timings based on selected start & end timings for the selected route
- IF (no bus timings are available)
    - RETURN empty list
- ELSE
    - STORE the values of bus timings returned in the above step in the list
    - RETURN list

Signature:

Method name	Compare health metric
Class name	
ID	
Clients (Consumers)	UTD Transport System and Bus driver
Associated use cases	Receive Bus Health Alert
Description of responsibilities	Compare health metric values with their respective threshold and send low health metric alert if component has health metric value below threshold
Arguments received	List of bus number, its components, health metric values for each component and respective health metric threshold
Type of value returned	None
Pre-conditions	Data for health metric values should be present
Post-conditions	Alerts will be sent for all components for which health metric value is less than the threshold

Logic:

- CREATE an empty list of objects
- ITERATE through list of input parameters
    - IF (health metric value is less than threshold)
        - Add the bus number, component name, health metric value and health metric threshold to the empty list
- If (list of objects is not empty)
    - Call *"Send low health metric alert"* method with parameter as list of objects

Signature:

Method name	Send low health metric alert
Class name	
ID	
Clients (Consumers)	UTD Transport System and Bus driver
Associated use cases	Receive Bus Health Alert
Description of responsibilities	Send the alert when a component of the bus has low health metric
Arguments received	List of bus number, component name, health metric value for it and respective health metric threshold
Type of value returned	Text message
Pre-conditions	At least one of the components has low health metric value for one of the buses
Post-conditions	Replacement bus would be required

Logic:

- ITERATE through the list of input parameters
    - CREATE message for UTD Transport Department and Bus Driver including component name, health metric value, health metric threshold
    - FETCH Bus driver details based on bus number
    - SEND message to UTD Transport Department and respective bus driver

21. System Control Design

- Login used for the Comet Cruiser app is using UTD ID and Password, to ensure that this app is used only by UTD authorized commuters.
- Option to get additional bus is reserved for UTD Transportation Department and Bus Driver only.
    - UTD Transportation Department will be able to send extra bus by analysing the commuter data and understanding the peak timings.
    - Bus Driver can request for additional bus only if current bus is filled up to its maximum capacity.
- Override option is available with the driver for temperature controls, in case the temperature sensors malfunction, to prevent over-heating or over-cooling of the bus temperature.
- Average wait-time is being taken from Google Maps API. This will ensure that while calculating the ETA, it considers all the factors including traffic, weather, speed of the bus, distance to travel, etc. and returns a precise estimate.
- Three sensors are being used on each door to get precise information about number of commuters entered/exited.

22. Weekly Project Timeline

WEEKLY SCHEDULE		TASKS
23rd Aug 2017	26th Aug 2017	Systems Analysis and Project Management 1st session
27th Aug 2017	2nd Sep 2017	Group members are assigned
3rd Sep 2017	9th Sep 2017	Brainstorming session & merit of brainstormed ideas
10th Sep 2017	16st Sep 2017	Formulating ideas
17th Sep 2017	23rd Sep 2017	Finalizing project idea
24th Sep 2017	30th Sep 2017	Formulating questions for data collection
1st Oct 2017	7th Oct 2017	Analysis of existing system
8th Oct 2017	14th Oct 2017	Analysis of existing system
15th Oct 2017	21st Oct 2017	Discussion of functionalities in proposed system
22nd Oct 2017	28th Oct 2017	Analysis and discussion on scope of the project
29th Oct 2017	4th Nov 2017	Design of proposed system
5th Nov 2017	11th Nov 2017	Use Case formulation, report review, creation of use case diagrams & descriptions and data dictionary notations
12th Nov 2017	18th Nov 2017	Creation of data dictionary notations and sequence diagrams, review of system design and BPMN class diagrams
19th Nov 2017	25th Nov 2017	Creation of UI interfaces, BPMN diagrams and software design
26th Nov 2017	2nd Dec 2017	Report editing, system control design, creation of UI interfaces, DB design, PPT details and report deliverables
3rd Dec 2017	9th Dec 2017	Recording the presentation and final report review

23. Minutes of Meeting

Meeting Number	1
Meeting Date	3rd September 2017
Meeting Time	12:00 – 14:30
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj, Sruti Seshadri
Late Attendee	Sriram Prasath Subbiah Pandiarajan
Discussion	Brainstorming session
Discussion was carried out on different ideas for the project. Ideas included Self-help kiosk, energy converting steps, parking optimization systems, food service system, mobile goods movement system.	
Task	**Performed By**
Self-help kiosk	Shubham Dave
Mobile goods movement system	Chaturvedi Musham
Food service system	Tanya Bhardwaj
Energy converting steps	Sruti Seshadri
Parking optimization systems	Sriram Pandiarajan
Conclusion	Brainstormed and recorded the best ideas

Meeting Number	2
Meeting Date	5th September 2017
Meeting Time	12:45 – 13:00
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Tanya Bhardwaj, Sruti Seshadri, Sriram Prasath Subbiah Pandiarajan
Late Attendee	
Discussion	Merit of brainstormed ideas

The ideas under consideration were discussed and respective pros and cons were listed.

Task	**Performed By**
Self-help kiosk – pros and cons	Shubham Dave
Mobile goods movement system – pros and cons	Chaturvedi Musham
Food service system – pros and cons	Tanya Bhardwaj
Energy converting steps – pros and cons	Sruti Seshadri
Parking optimization systems – pros and cons	Sriram Pandiarajan

Conclusion	Collected more details regarding each of the ideas

Meeting Number	3
Meeting Date	16th September 2017
Meeting Time	12:00 – 14:00
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj, Sruti Seshadri
Late Attendee	Sriram Prasath Subbiah Pandiarajan
Discussion	Formulating ideas

Each member had to formulate and give a basic idea of what is to be achieved and why it helps using their idea.

Task	**Performed By**
Self-help kiosk – draft document	Shubham Dave
Mobile goods system – draft document	Chaturvedi Musham
Food service system – draft document	Tanya Bhardwaj
Energy converting steps – draft document	Sruti Seshadri
Parking optimization systems – draft document	Sriram Pandiarajan

Conclusion	Created documents of ideas to present to the professor

Meeting Number	4
Meeting Date	19th September 2017
Meeting Time	12:45 – 13:45
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj, Sruti Seshadri, Sriram Prasath Subbiah Pandiarajan
Late Attendee	
Discussion	Finalizing project idea

Defined the scope of each of our ideas and presented the ideas to the professor. Finalized on one of the ideas of improving Comet Cruiser system.

Task	**Performed By**
Self-help kiosk – scope and presentation	Shubham Dave
Mobile goods system– scope and presentation	Chaturvedi Musham
Food service system – scope and presentation	Tanya Bhardwaj
Energy converting steps – scope and presentation	Sruti Seshadri
Parking systems – scope and presentation	Sriram Pandiarajan

Conclusion	Finalized on the optimization of the Comet Cruiser system

Meeting Number	5
Meeting Date	30th September 2017
Meeting Time	Online
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj, Sruti Seshadri, Sriram Prasath Subbiah Pandiarajan
Late Attendee	
Discussion	Accumulated questions for data collection

Discussed about what is required to understand the existing system and came up with the questions which we need to ask in the public information request to analyse the existing system.

Task	**Performed By**
Existing system – data collection	Shubham Dave
Existing system – data collection	Chaturvedi Musham
Existing system – data collection	Tanya Bhardwaj
Existing system – data collection	Sruti Seshadri
Existing system – data collection	Sriram Pandiarajan

Conclusion	Finalized the questions to be included for Public Information Request

Meeting Number	6
Meeting Date	11th October 2017
Meeting Time	Online
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj, Sruti Seshadri Sriram Prasath Subbiah Pandiarajan
Late Attendee	
Discussion	Analysis of existing system

Document and discussed the merit of the existing system using images.

Conclusion	Documented about incorporation of functionalities and optimization of existing system

Meeting Number	7
Meeting Date	24th October 2017
Meeting Time	12:40 – 13:15
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Shubham Bhushanbhai Dave, Tanya Bhardwaj, Sruti Seshadri
Late Attendee	
Discussion	Discussion of functionalities

Discussed about adding IOT to bus stops and bus doors.
Discussed about new routes that might help commuters in resolving current problems.

Task	**Performed By**
Proposed solution – IOT inside vehicle	Shubham Dave
-	Chaturvedi Musham
Proposed solution – Step sensors	Tanya Bhardwaj
Proposed solution – New route designs	Sruti Seshadri
-	Sriram Pandiarajan

Conclusion	Functions and additional features of proposed system were noted

Meeting Number	8
Meeting Date	28th October 2017
Meeting Time	Online
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj, Sruti Seshadri
Late Attendee	Sriram Prasath Subbiah Pandiarajan
Discussion	Analysis and scope of the project
Finalized the scope of the project and the design of each component that was proposed.	

Task	**Performed By**
Proposed solution – scope and parameters	Shubham Dave
Proposed solution – scope and parameters	Chaturvedi Musham
Proposed solution – scope and parameters	Tanya Bhardwaj
Proposed solution – scope and parameters	Sruti Seshadri
Proposed solution – scope and parameters	Sriram Pandiarajan

Conclusion	Created first draft of project deliverables

Meeting Number	9
Meeting Date	31st October 2017
Meeting Time	13:00 – 14:30
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj, Sruti Seshadri
Late Attendee	Sriram Prasath Subbiah Pandiarajan
Discussion	Design of proposed system

Features and incorporation of features for the proposed system	
Task	**Performed By**
Context diagram – draft & first draft of full report	Shubham Dave
Use Case diagram – draft	Chaturvedi Musham
Relative merits of features	Tanya Bhardwaj
Vehicle IOT features	Sruti Seshadri
Scope of system features	Sriram Pandiarajan
Conclusion	First draft of context diagram and project report

Meeting Number	10
Meeting Date	5th November 2017
Meeting Time	14:00 – 16:00
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj
Late Attendee	Sriram Prasath Subbiah Pandiarajan
Discussion	Use Case formulation and report review

Designed the use case diagrams and created first draft of report.	
Task	**Performed By**
Use Case Diagram 1 – design	Shubham Dave
Use Case Diagram 2 – design	Chaturvedi Musham
Use Case Diagram 3 – design	Tanya Bhardwaj
-	Sruti Seshadri
Scope of system features	Sriram Pandiarajan

Conclusion	Created first draft of Use Cases
Meeting Number	11
Meeting Date	7th November 2017
Meeting Time	13:00 – 16:00
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj, Sruti Seshadri
Late Attendee	
Discussion	Vehicle health data and Use case diagram and descriptions
Researched vehicle health IOT and started the design of use case descriptions	
Task	**Performed By**
Use case description 1	Shubham Dave
Use case description 2	Chaturvedi Musham
Use case description 3	Tanya Bhardwaj
Vehicle Health IOT implementation	Sruti Seshadri
-	Sriram Pandiarajan
Conclusion	Created Visio diagrams of use cases

Meeting Number	12
Meeting Date	9th November 2017
Meeting Time	Online
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj, Sruti Seshadri
Late Attendee	
Discussion	Formulated questions for 2nd public information request
Formulated the questions to be submitted for public information request	
Conclusion	Recorded the questions and data required from the public information request

Meeting Number	13
Meeting Date	11th November 2017
Meeting Time	10:30 – 12:30
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj, Sruti Seshadri
Late Attendee	
Discussion	Created use case description and data dictionary notations

Completed use case descriptions and started data dictionary notations	
Task	**Performed By**
Data dictionary notations	Shubham Dave
Use case description review	Chaturvedi Musham
Data dictionary notations	Tanya Bhardwaj
Use case description and diagram review	Sruti Seshadri
-	Sriram Pandiarajan

Conclusion	Completed use case descriptions

Meeting Number	14	
Meeting Date	13th November 2017	
Meeting Time	13:00 – 17:00	
Meeting Type	SAPM Project Group 2 Meeting	
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj, Sruti Seshadri	
Late Attendee		
Discussion	Created data dictionary notations and sequence diagrams	
Completed data dictionary notations and started sequence diagrams		
Task		**Performed By**
Data dictionary notations		Shubham Dave
Sequence diagrams		Chaturvedi Musham
Data dictionary notations		Tanya Bhardwaj
Sequence diagrams		Sruti Seshadri
-		Sriram Pandiarajan
Conclusion	Completed data dictionary notation and started design of sequence diagrams	

Meeting Number	15
Meeting Date	18th November 2017
Meeting Time	10:00 – 12:00
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj, Sruti Seshadri
Late Attendee	
Discussion	Review of system control design, BPMN diagrams and revision of sequence diagrams

Divided work after discussion of deliverables.
Reviewed system control design and BPMN diagrams

Task	**Performed By**
Review of system control design	Shubham Dave
Sequence diagram review	Chaturvedi Musham
Class diagram – with methods	Tanya Bhardwaj
BPMN diagrams	Sruti Seshadri
-	Sriram Pandiarajan

Conclusion	Updated the report with all deliverables completed till date.

Meeting Number	16	
Meeting Date	26th November 2017	
Meeting Time	18:00 – 22:00	
Meeting Type	SAPM Project Group 2 Meeting	
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj, Sruti Seshadri	
Late Attendee		
Discussion	UI interfaces, BPMN design, software design and UI interfaces	
Review and changes on each member's completed designs		
Task		**Performed By**
Software design		Shubham Dave
UI Interface design		Chaturvedi Musham
Class diagram – without methods		Tanya Bhardwaj
BPMN diagram		Sruti Seshadri
-		Sriram Pandiarajan
Conclusion	Completed UI interfaces, BPMN diagrams, class diagrams and software design	

Meeting Number	17
Meeting Date	28th November 2017
Meeting Time	13:00 – 15:00
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj, Sruti Seshadri, Sriram Prasath Subbiah Pandiarajan
Late Attendee	
Discussion	Report editing, system control design, UI interfaces, Database design
Review and final format documentation of each member's work	

Task	**Performed By**
System control design & report editing, documentation – references, proposed solution, functional specifications	Shubham Dave
UI interface design	Chaturvedi Musham
DB design and ER diagram	Tanya Bhardwaj
MoM & weekly project timeline	Sruti Seshadri
	Sriram Pandiarajan
Conclusion	Completed system control design, UI interfaces and DB design.

Meeting Number	18
Meeting Date	1st December 2017
Meeting Time	Online – conference call
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj, Sruti Seshadri
Late Attendee	
Discussion	PPT details and report deliverables
Details of PPT deliverables	
Conclusion	Time and place of PPT and voice over meeting

Meeting Number	19
Meeting Date	6th December 2017
Meeting Time	20:00 – 22:00
Meeting Type	SAPM Project Group 2 Meeting
Attendees	Shubham Bhushanbhai Dave, Chaturvedi Musham, Tanya Bhardwaj, Sruti Seshadri, Sriram Prasath Subbiah Pandiarajan
Late Attendee	
Discussion	Recording of final presentation and final report review

Task	**Performed By**
Project report – review	Shubham Dave
PPT – review	Chaturvedi Musham
PPT – completion	Tanya Bhardwaj
PPT – first draft	Sruti Seshadri
	Sriram Pandiarajan

Conclusion	Project PPT and voice recordings completed

24. References

- "*Object-Oriented Systems Analysis and Design*" by Jeff Hoffer, Joey George, and Joe Valacich, Pearson Prentice-Hall, Second Edition, 2006
- Data obtained from UTD Transportation Department through public information request.
- *"Who owns the road? The IOT-connected car of today—and tomorrow"* – https://dupress.deloitte.com/dup-us-en/focus/internet-of-things/IOT-in-automotive-industry.html
- *"Telecoms versus carmakers in race to get connected"* – https://www.ft.com/content/6c1b7f60-a9d3-11e7-93c5-648314d2c72c?segmentid=acee4131-99c2-09d3-a635-873e61754ec6
- *"List of temperature sensors"* – https://en.wikipedia.org/wiki/List_of_temperature_sensors
- *"How sensor technology will shape the cars of the future"* – https://www.melexis.com/en/insights/knowhow/how-sensor-technology-shape-cars-future
- *"EGR Flap Position Sensing in Powertrain (Automotive)"* – https://www.melexis.com/en/applications/egr-flap-position-sensing
- "*People Counters, Retail Traffic Counting & Pedestrian Door Counters - Traf-Sys*" – https://www.trafsys.com/people-counting/