“Comet Transit”

Youtube Link: https://youtu.be/rpz7d0wPqgI

MIS 6308.001 SAPM - Fall 2019

Under the guidance of:

Mr Srinivasan Raghunathan

Group 4:

Ayush Singh

Nishi Shah

Prasanth Ramanathan

Shruthi Gopal

Surya Varma

Table of Contents

Executive Summary

The University of Texas at Dallas provides residential options within the campus. Students also have the option of living outside the campus with a lot of residential options available around the campus. The influx of students has been increasing each year and has seen a rapid increase in the number of students who live off-campus. Students who have opted to reside in the apartments outside the campus rely upon different modes of transport to commute to the university. One such mode of transport is the DART Comet Cruiser which is a free shuttle to the university and covers a portion of the areas around the campus. The students who do not possess their own vehicles prefer to use the UTD shuttle for their daily commute.

The UTD Shuttle has several types of buses plying to and from the university. On an average, the UTD shuttle has a capacity of carrying 65 passengers (no. of seats and standing capacity included). The students using the UTD Shuttle sometimes face a problem where the buses tend to get completely full during peak hours. Inevitably, it causes a situation where the bus driver would not be able to allow any more passengers who are waiting at subsequent stops to enter. This might be a problem because most of the passengers using the UTD Shuttle are students and they might be running late for their classes. Since the buses currently operate on a fixed schedule and the next bus reaches the university roughly in twenty minutes, there is a possibility that the students might not be able to reach their classes on time. The idea of providing additional buses for each and every route and timing would not be economically feasible, and it is not the optimal solution as well.

To address this problem, we have decided to utilize the UTD students' app to indicate the crowd status of each bus. [CROWD MONITORING IDEA] Based on the crowd, the app will indicate the crowd status inside the bus. The crowd status will be displayed by different colors indicating the availability of space inside the bus. The red color would indicate that the bus is full. Once the crowd status turns red, the DART officials will be automatically notified.

DART will have two additional vans specifically to solve this issue. Once a notification is received indicating that a bus is full, DART would operate a van in that route and based on the GPS coordinates of the bus, the immediate location of the bus would be identified. The van would then be directed to the specific stop from which the bus was found to be full.

The app will save students' class schedules and accordingly notify the students during the days in which they have classes. Based on daily crowd data, the app would provide suggestions to the students to choose the optimal bus timing to reach the university. This will be based on the time when the class begins and the number of students that usually commute in that particular bus timing.

Problem Statement

Some commute policies at UTD are recommended to be updated based on student's travel patterns - an area that can grab the advantage of advancement in technology. For instance, deployment of sensors to detect student capacity in the bus, determining the need of a vehicle for peak hours, and so on. Through this project, we encourage systematic and optimized travel arrangements that accommodate timely student capacity. We seek to fill the knowledge gap and design better policies by addressing three challenges as under:

The UTD mobile app allows students to monitor the location of the bus. However, this doesn't allow them to see the occupancy capacity that is whether it's completely filled or partially filled.

The current system doesn't offer any customized student transit plan. If implemented then this would notify the students on how to plan or take the bus for a particular day.

Another problem with the current system is that it doesn't automatically do effective crowd management. If the bus gets filled then the students have to wait for the next bus to arrive. This makes the students late for classes. Moreover, for students who are running late, it worsens the situation. Lastly, this also burns a student's pocket if they have a scheduled exam on the very day. Also, considering the management's perspective they don't have any access to information about how many people are left to be picked so as to deploy any backup van or bus.

Business Needs -

Below are the business needs which highlight the need for a new system

- Longer waiting time for students
- Buses are often overcrowded
- No immediate service to take care of left-out students
- Lack of customized commute plan reminder

Objectives -

Objectives of the proposed system are as follows

- Create an improved transit system which allows students to track occupancy in the bus
- Create a new system which notifies students about a recommended transit plan
- Create a new system which deploys vans/buses to take care of left-out students

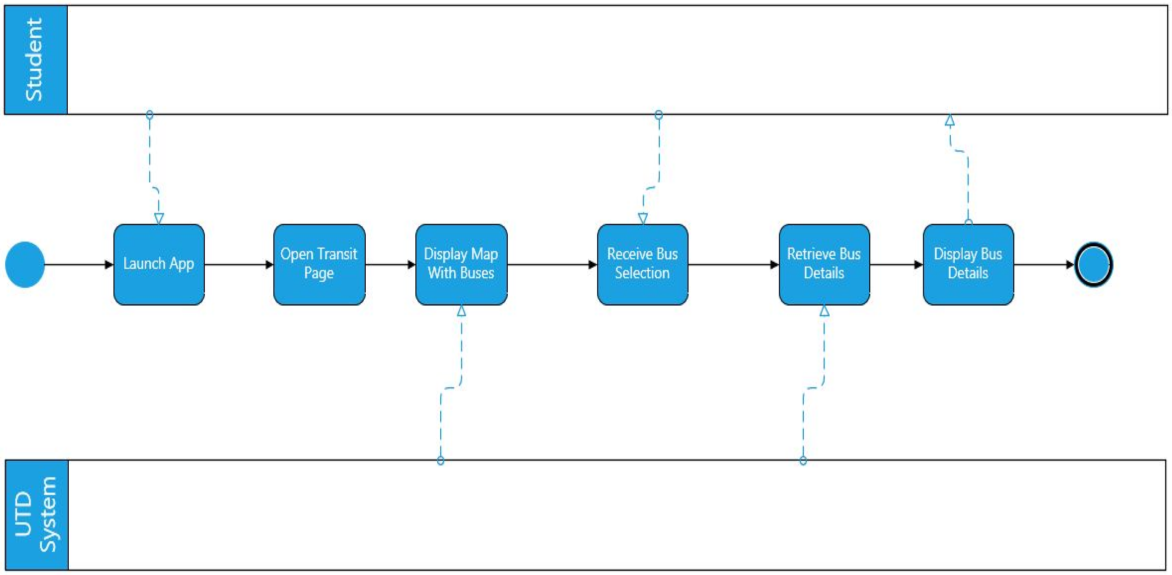
Scope -

The scope of the new proposed system is as follows

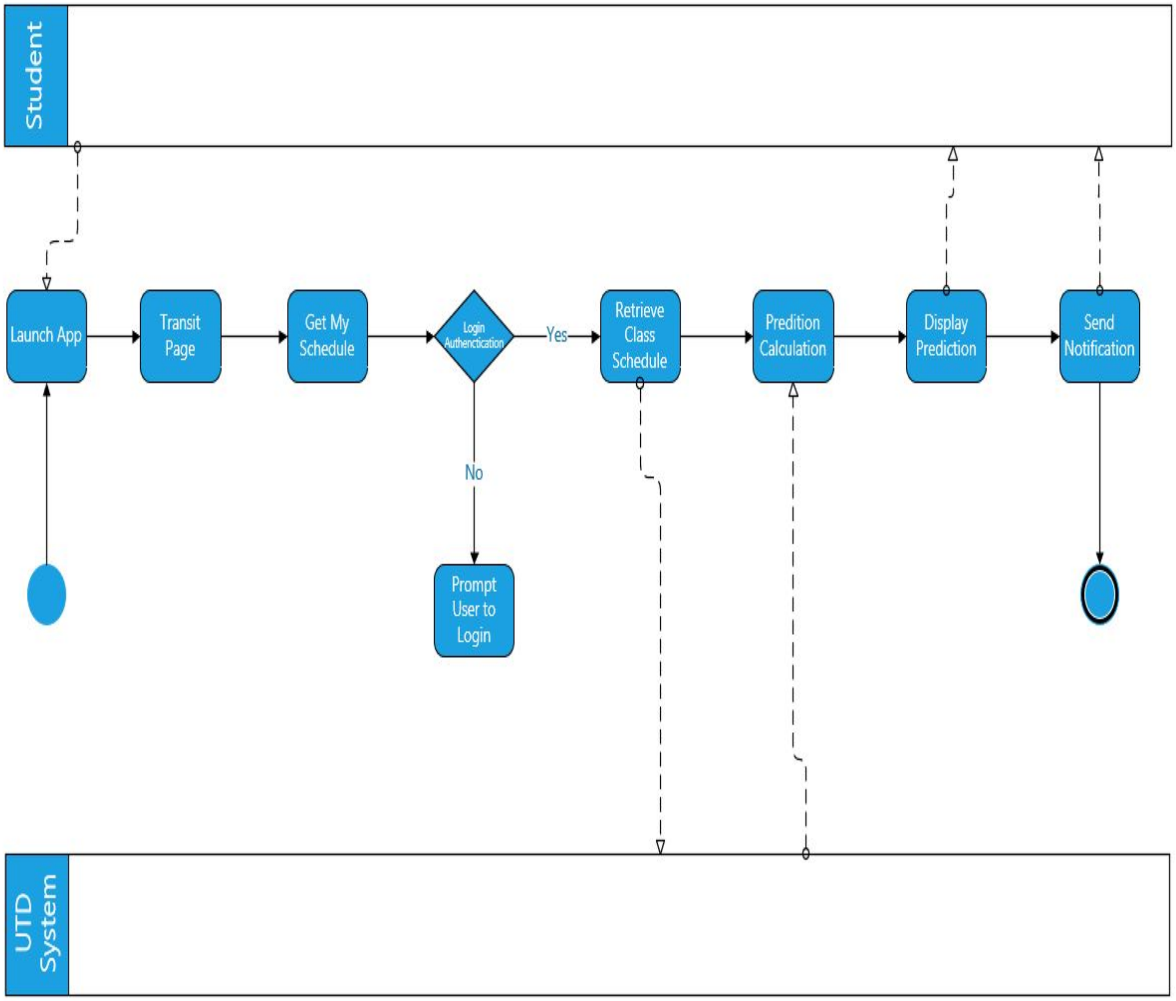
- It will target the UTD students who avail DART/ Comet cruiser services
- It will also improve transit management for UTD Dart officials
- Timeline to reach production will be 3 months
- Estimate cost for including sensors will be $600 for 6 buses

BPMN Models

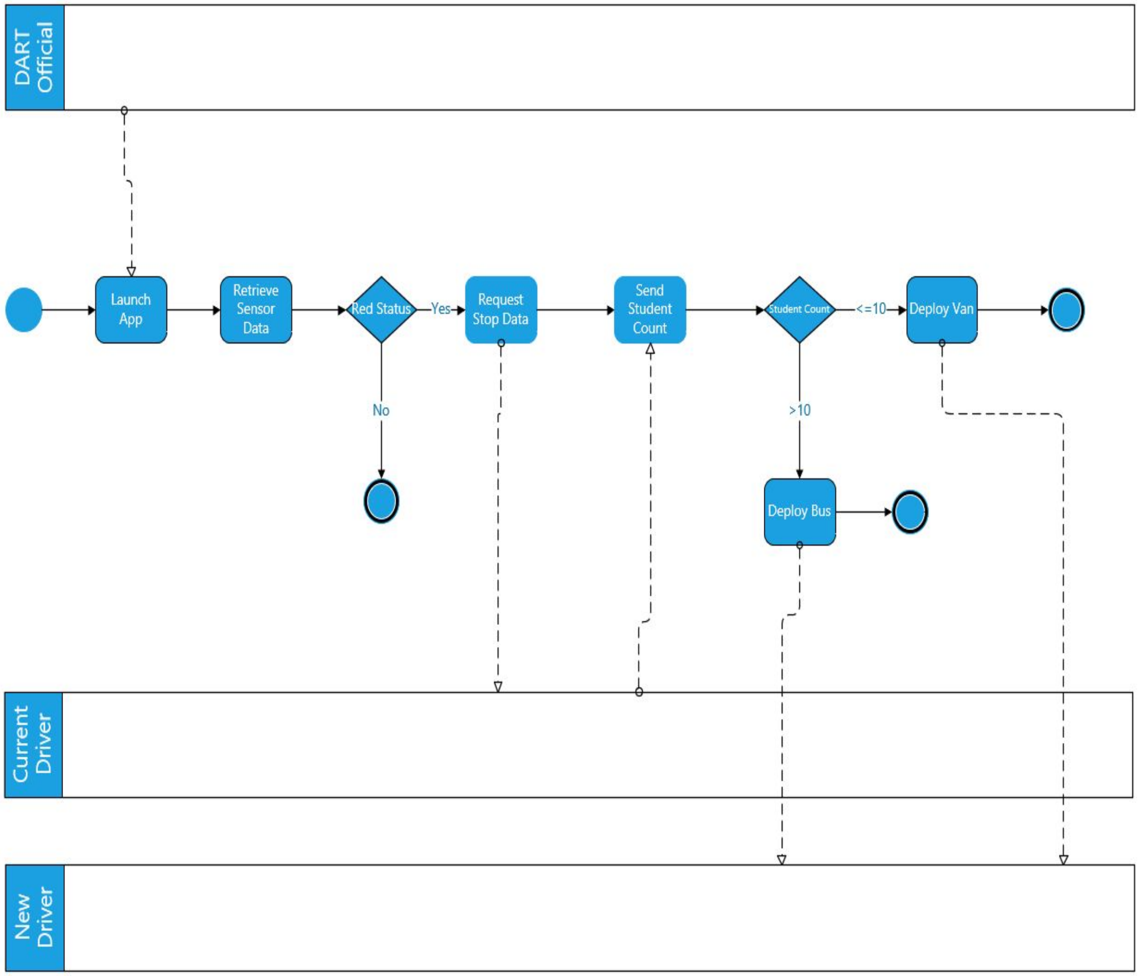
1. **Display Bus Status**



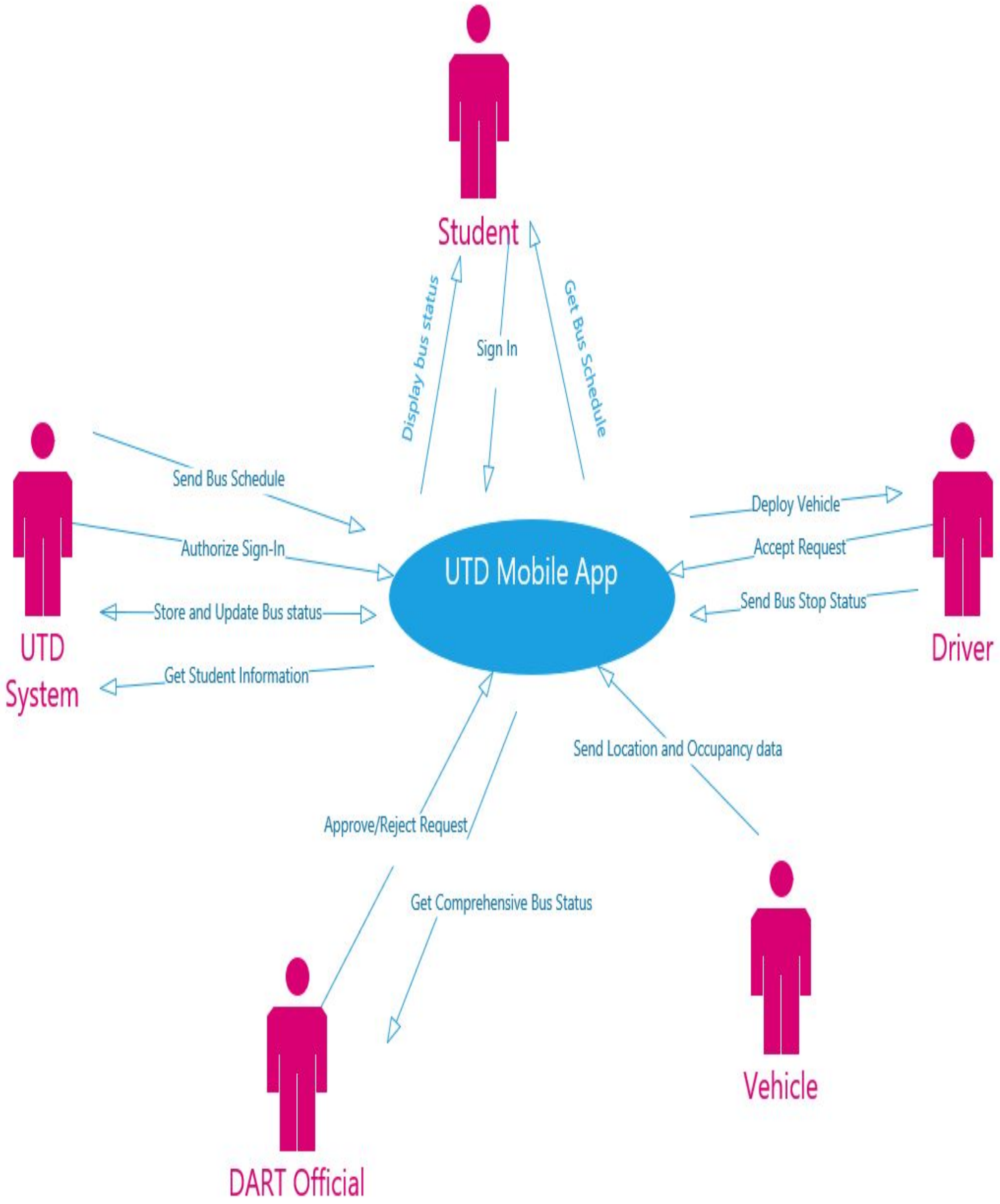
2. Get Bus Schedule/Notification



3. Deploy Vehicle

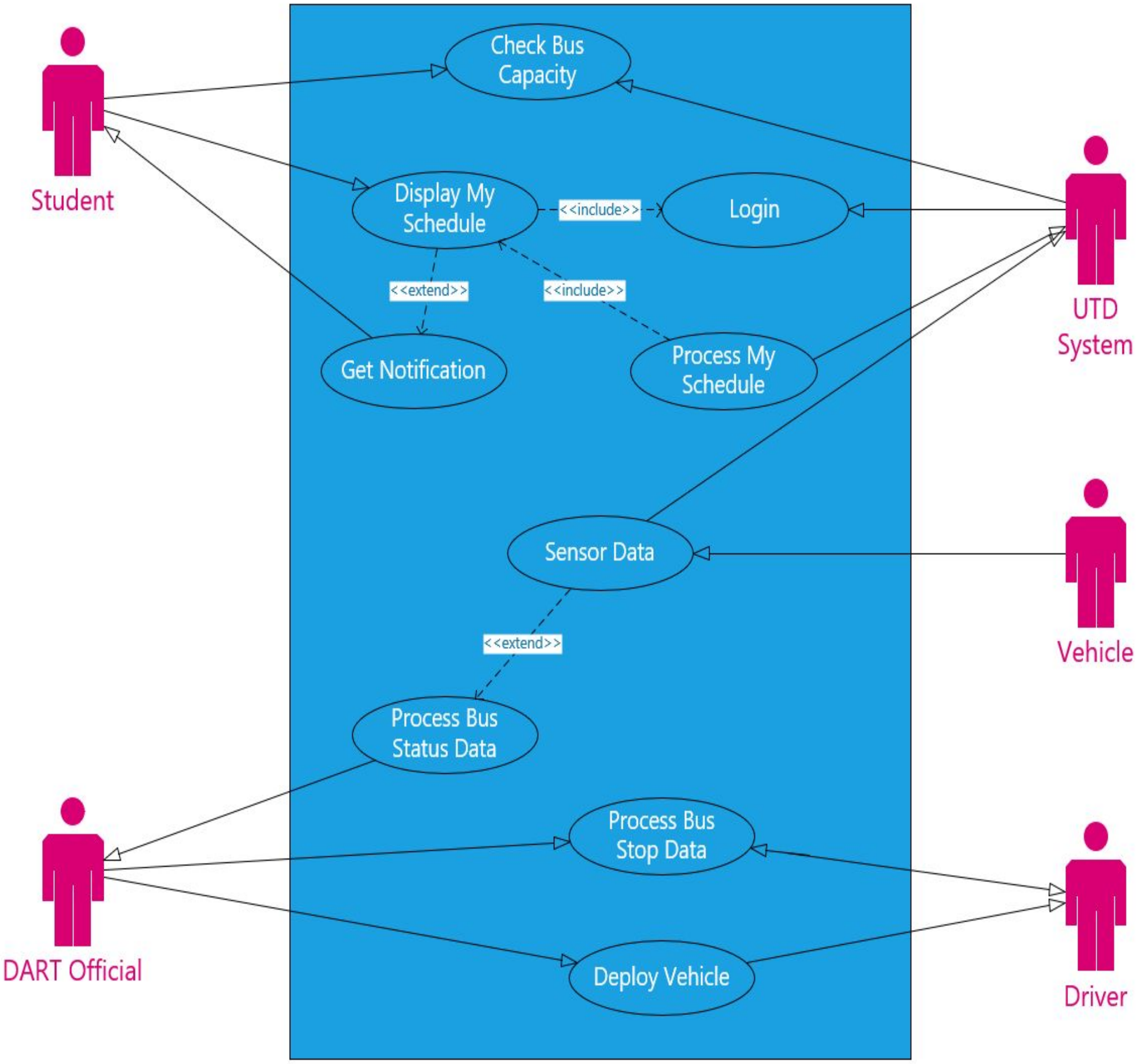


Context Diagram



Process Model: Use-Case Diagram and Use Case Descriptions

Use-Case Diagram



Use Case Descriptions

1.

Use Case Name: Check Bus Capacity

Primary Actor: Student

Stakeholders: Student, UTD System

Description: To check the location and occupancy of the buses

Trigger: Student accesses the transit page in the UTD mobile App

Relationships:

Includes:

Extends:

Normal flow of events:

1. Open the UTD mobile App

2. Open the transit page

3. System generates map page

4. User selects the bus

5. System displays the bus information

Exceptional flow of events:

2.

Use Case Name: Login

Primary Actor: Student

Stakeholders: Student, UTD System

Description: Students sign in and a request is automatically sent to the UTD System to fetch class schedule

Trigger: Student signs in on the transit page with NetID and Password

Relationships:

Includes:

Extends:

Normal flow of events:

1. Login information is submitted

2. System sends NetID and password to UTD system for authentication

3. Fetch class time table request is received by the system

Exceptional flow of events:

3.

Use Case Name: Process my Schedule

Primary Actor: UTD System

Stakeholders: UTD System

Description: UTD System is fetching the student class schedule and matches it to the previous bus data

Trigger: Member login

Relationships:

Includes: Display my Schedule

Extends:

Normal flow of events:

1. Execute login use case
2. System fetches the class schedule from UTD System
3. Process the class schedule with previous bus data.

Exceptional flow of events:

4.

Use Case Name: Display my Schedule

Primary Actor: Student

Stakeholders: Student, UTD System

Description: UTD System displays the processed prediction

Trigger: Student clicks on the Display my Schedule option on the transit page

Relationships:

Includes: Login

Extends: Get Notification

Normal flow of events:

1. Click on the Display my Schedule section in the transit page
2. Execute the login use case
3. Execute the Process my Schedule use case
4. Display prediction
5. Prompt the user to subscribe for a notification

Exceptional flow of events:

3A. Authentication failed, display login error screen for invalid username and password

5.

Use Case Name: Get Notification

Primary Actor: Student

Stakeholders: Student

Description: Prompt user to subscribe for prediction notification

Trigger: User accesses Display my Schedule page

Relationships:

Includes:

Extends:

Normal flow of events:

1. Execute Display my Schedule use case

2. Send notification to the user

Exceptional flow of events:

2A. If user declines the subscription, do not send the notification

6.

Use Case Name: Sensor Data

Primary Actor: Vehicle

Stakeholders: Vehicle, UTD System

Description: Sensor data sent from Vehicle to UTD System

Trigger: Students boarding the bus

Relationships:

Includes:

Extends: Process bus status data

Normal flow of events:

1. Gather motion sensor data when students board or leave the bus

2. Update the UTD System with sensor data

Exceptional flow of events:

7.

Use Case Name: Process Bus Status Data

Primary Actor: Dart Official

Stakeholders: Dart Official

Description: Process sensor data and notify Dart Official when it turns red

Trigger: The bus gets full and crowd status in the sensor turns red

Relationships:

Includes:

Extends:

Normal Flow of events:

1. Bus status updated to red when bus gets fully filled

2. Notify Dart Official

Exceptional flow of events:

1A. Bus is not fully filled

2A. Do not notify Dart Official

8.

Use Case Name: Process Bus Stop Data

Primary Actor: Driver

Stakeholders: Dart Official, Driver

Description: Driver reports the extra students waiting at bus stops to the Dart Official

Trigger: Bus gets filled and there are more students at the bus stops to be accommodated

Relationships:

Includes:

Extends:

Normal flow of events:

1. Bus status turns red

2. Report the number of students waiting at the bus stops

Exceptional flow of events:

2A. Do not respond if there are no additional students waiting at the bus stops

9.

Use Case Name: Deploy Vehicle

Primary Actor: Dart Official

Stakeholders: Dart Official, Driver

Description: To notify the new driver to take a bus/van when the old bus gets filled

Trigger: Dart Official notifies new driver and deploys a new vehicle

Relationships:

Includes and Extends:

Normal flow of events:

1. Get the data regarding the number of students waiting at the bus stops

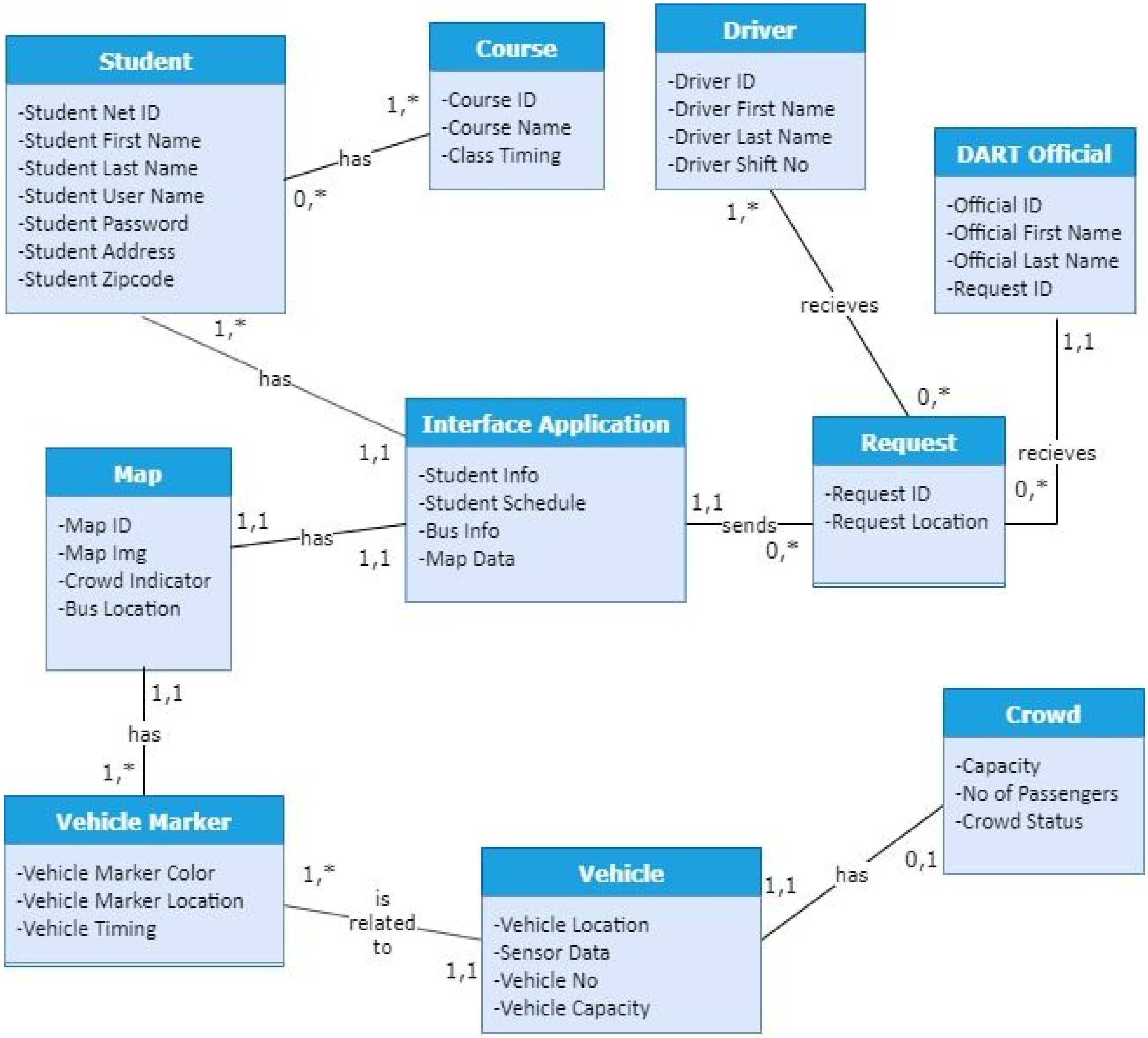
2. Deploy bus when more than 10 students are waiting and deploy van otherwise

3. New driver notified

4. New driver takes new vehicle

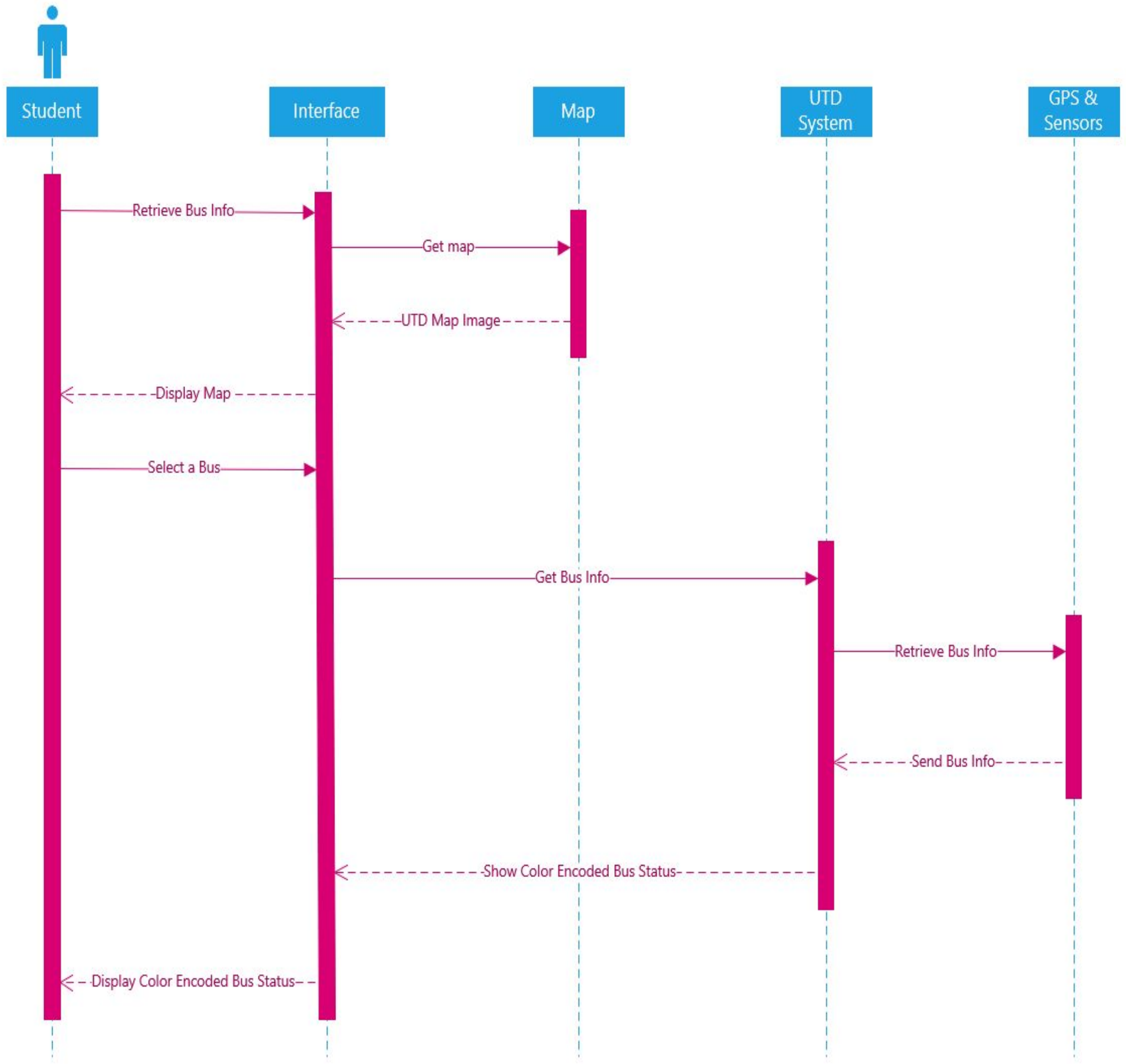
Exceptional flow of events:

Data Model: A class diagram without methods for the proposed system

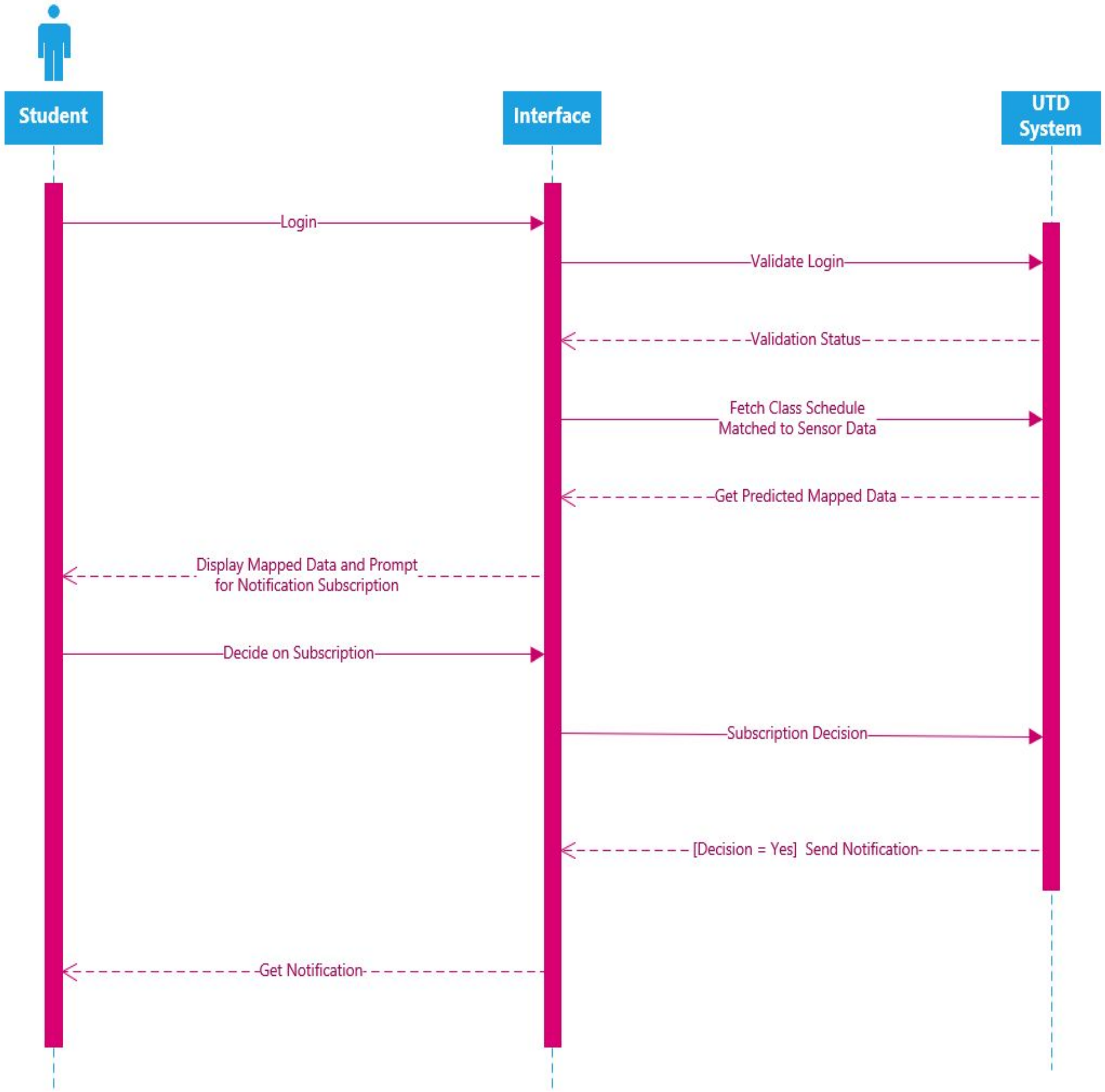


Object Behavior Model: A Sequence Diagram

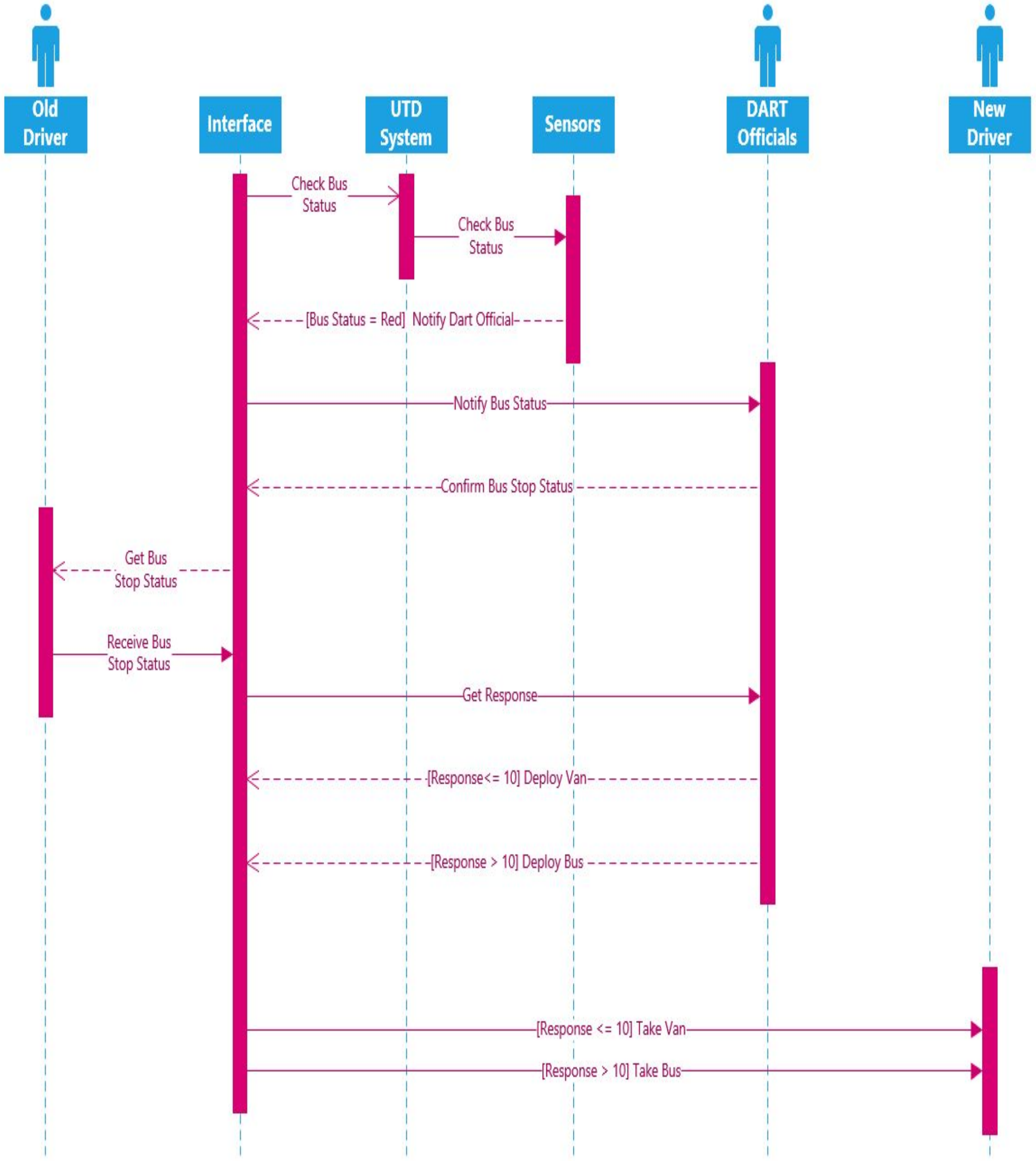
1. Display Bus Capacity



2. Prediction and Notification


Student
Interface
UTD System
Login
Validate Login
Validation Status
Fetch Class Schedule Matched to Sensor Data
Get Predicted Mapped Data
Display Mapped Data and Prompt for Notification Subscription
Decide on Subscription
Subscription Decision
[Decision = Yes] Send Notification
Get Notification

3. Deploy Vehicle



Data Dictionary

Login info: Username + Password

Username=NetID

Password=Data element

Bus colors=Sensor data+filter

Map page=UTD system road map + legend + Filter

Filter= (Red) + (Yellow) + (Green)

Legend=Bus colors

Prediction = {classes scheduled} + {previous bus reports of the same day of the week}

Notification = Username + Prediction

Class schedule = Data element

Bus reports = Data element

Sensor data = Data element

#people in each bus = sensor data

People waiting at the stop = Data element

UTD System Road Map =Data element

DartNotification=Bus Color1

Bus color1=Red

Driver1 notification=DartBusStopRequest

Driver1 input=DartBusStopRequest+Driver1Response

DartBusStopRequest=Data element

Driver1Response=Data element

Driver2 notification=DartDeployRequest

DartDeployRequest=(Bus) + (Van)

Functional Specification Document

- The proposed system will help the students to track the bus on route. The system works by recording real-time onboarding and offboarding data via sensors attached on the bus. These sensors in turn send the occupancy data to the University of Texas at Dallas's system. This data is processed and color codes are assigned depending on the occupancy.

  If number of students = 45 (Color turns Red)
  If number of students < 45 (Color turns Yellow)
  If number of students < 30 (Color turns Green)

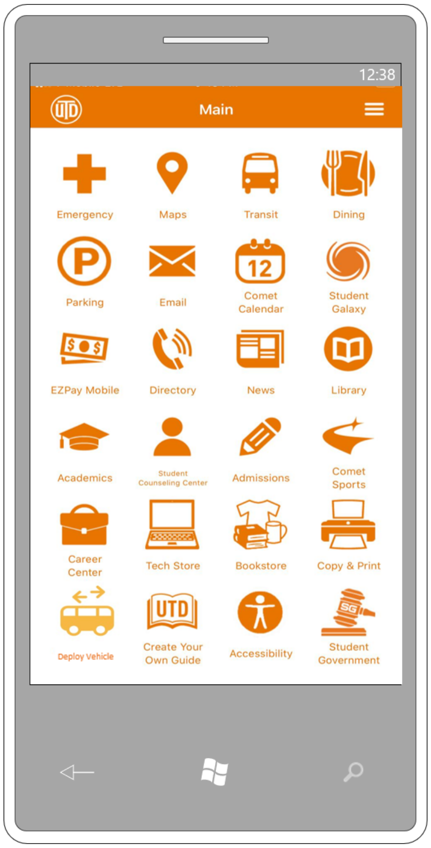
  The students can see these color coded data in the transit page of UTD mobile app and determine whether to board the current bus or not.

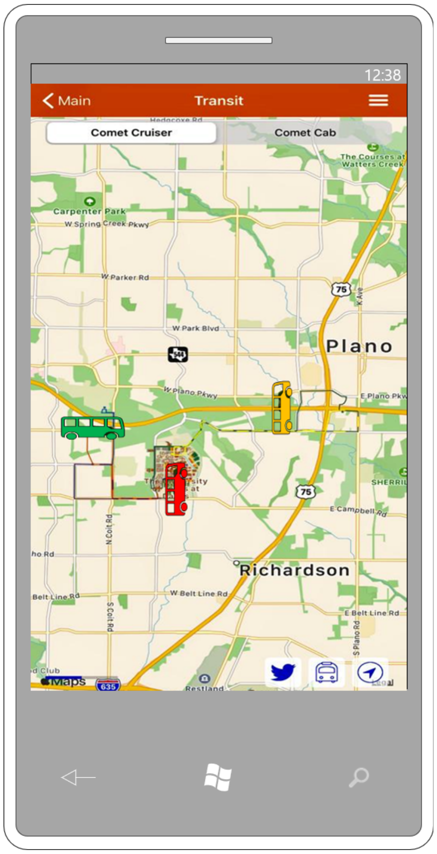
- Another proposed system is to introduce customized alert/notification system depending on a student's class schedule. The system will make use of the past travel data to come up with a prediction and help student suggest an ideal boarding time. The system will be sending out notifications once in the morning(by default) and it will be available to view in the UTD mobile app. To access this feature, the students must login in the app and allow the UTD system to access their class schedule. Once it is done, the system will automatically fetch data and work on prediction algorithm. It can also be setup to send custom notifications depending on the time of the day.

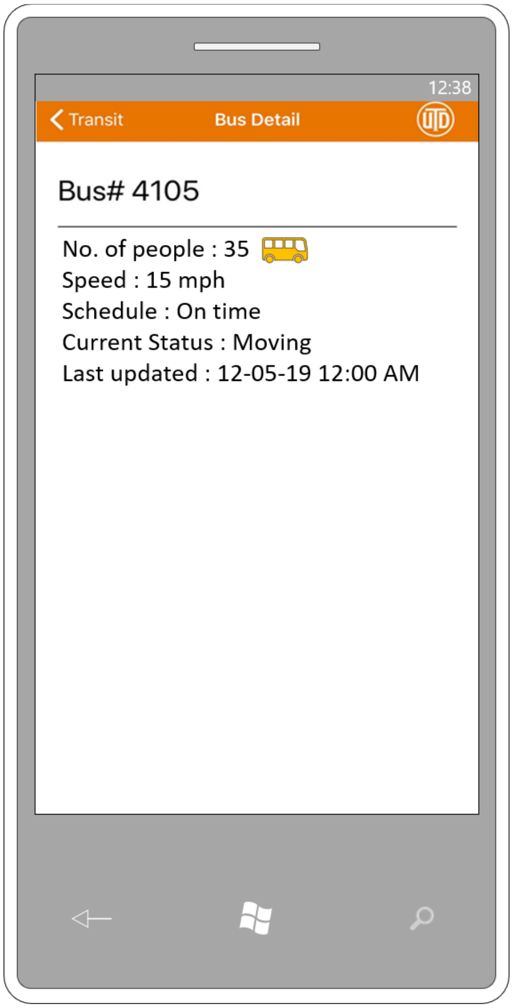
- The third proposed system is to introduce a vehicle deployment system. This system is a novel one and will work towards catering the needs of missed out students. In the current scenario, often students get left out due to overcrowded buses and there is no system to take care of the left out systems. This new system will automatically notify the DART officials when the bus becomes filled and it will also get another confirmation from the bus driver to ensure whether there is actually a need or not. Once the previous said conditions are met a new driver will be contacted and a vehicle will be deployed. The deployed vehicle will be either a bus or a van depending on the number of students waiting at the bus stops.

If number of students < 10, a Van will be deployed
If number of students > 10, a Bus will be deployed

Interface Design







Bus# 4105

No. of people : 35

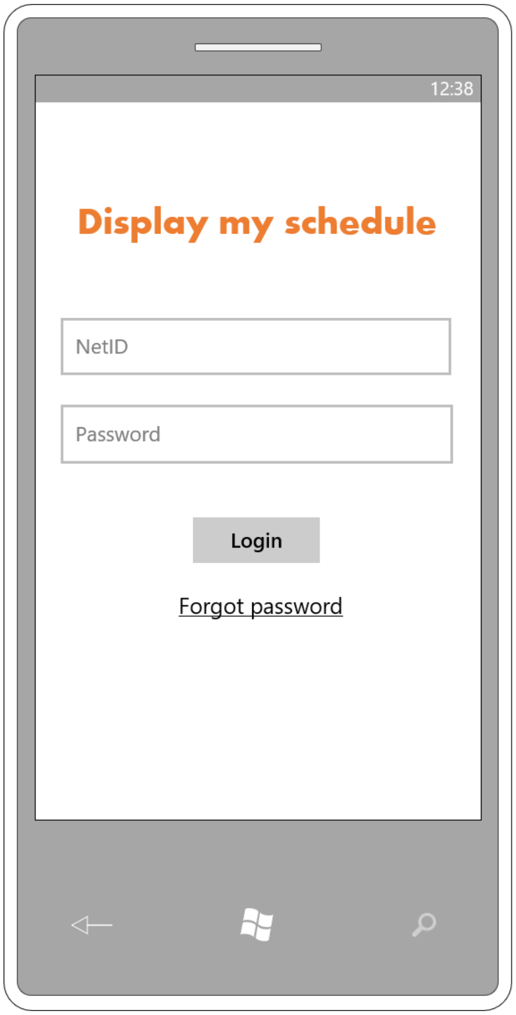
Speed : 15 mph

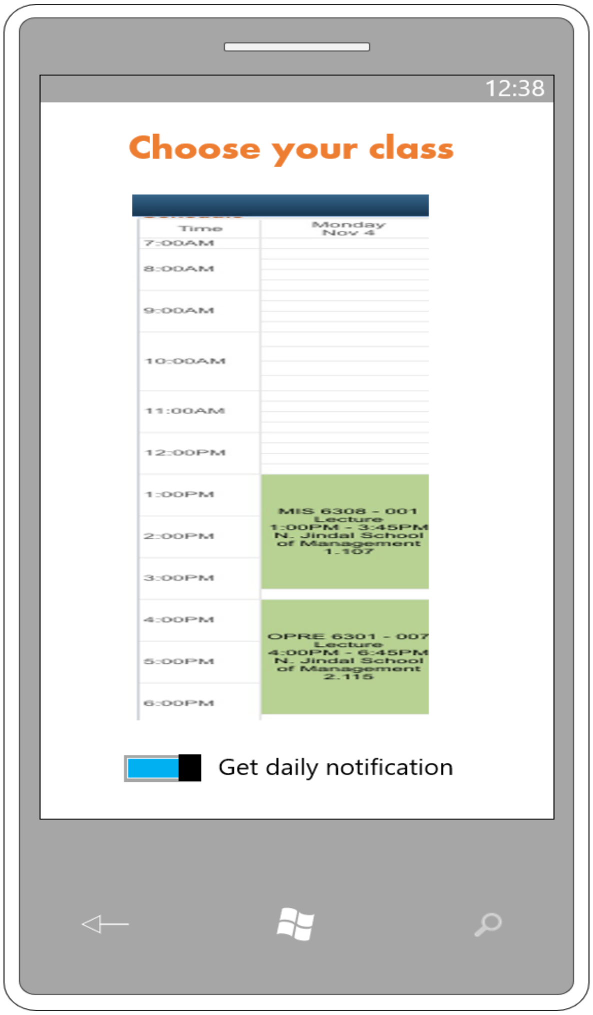
Schedule : On time

Current Status : Moving

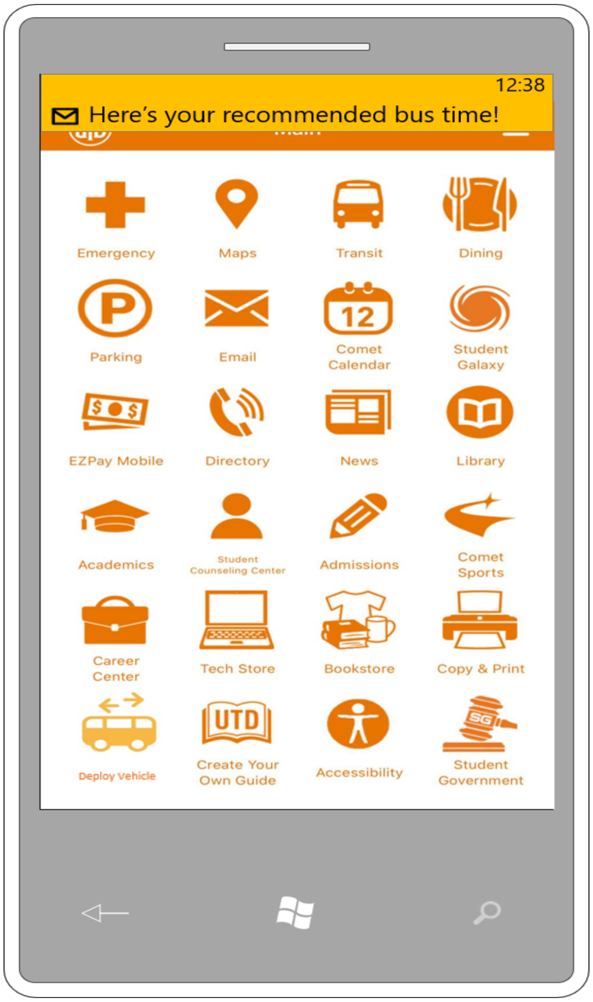
Last updated : 12-05-19 12:00 AM

Group - 4



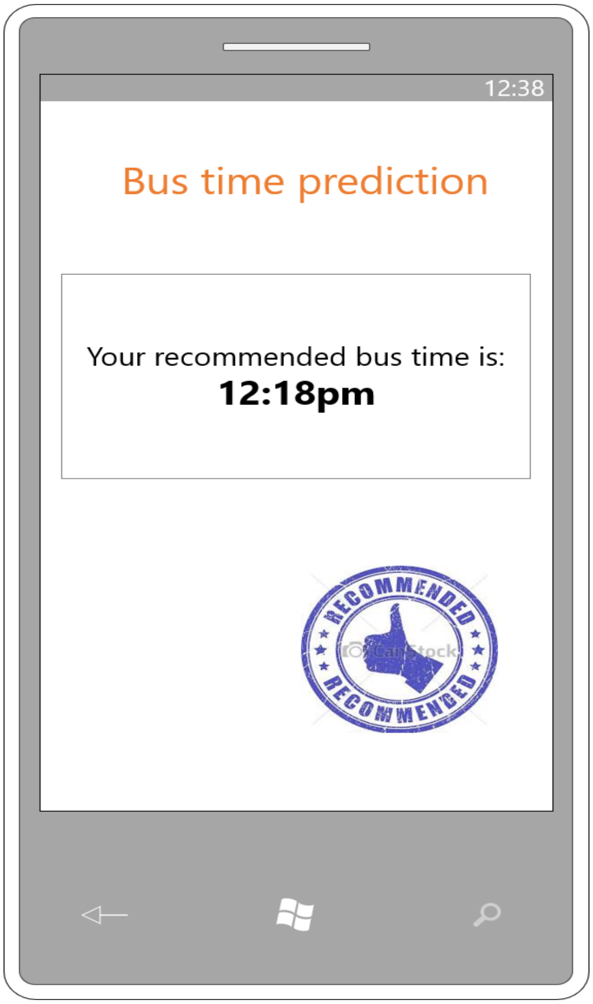


Time	Monday Nov 4
7:00AM	
8:00AM	
9:00AM	
10:00AM	
11:00AM	
12:00PM	
1:00PM	MIS 6308 - 001 Lecture 1:00PM - 3:45PM N. Jindal School of Management 1.107
2:00PM	
3:00PM	
4:00PM	OPRE 6301 - 007 Lecture 4:00PM - 6:45PM N. Jindal School of Management 2.115
5:00PM	
6:00PM	



12:38

Here’s your recommended bus time!

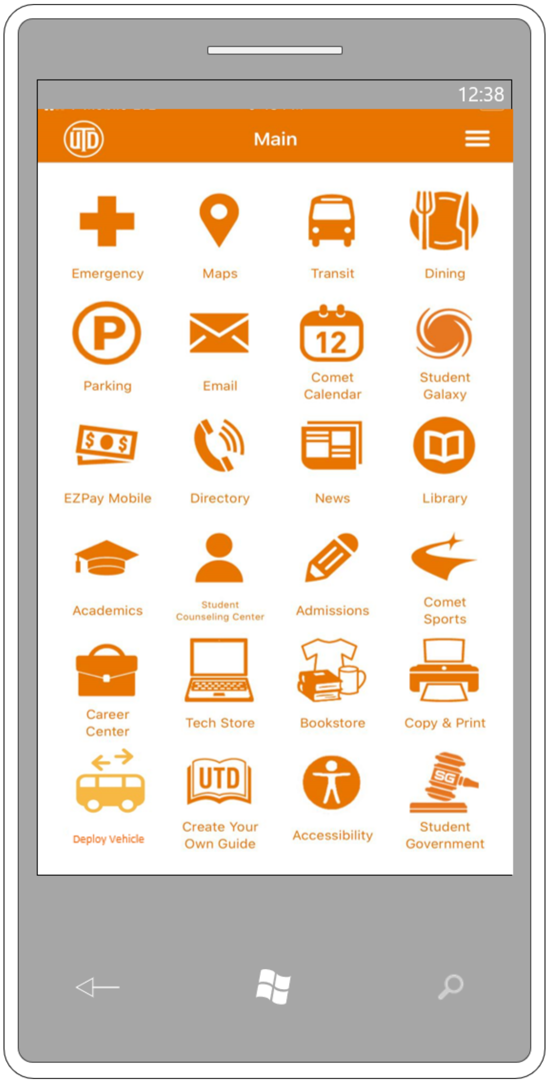


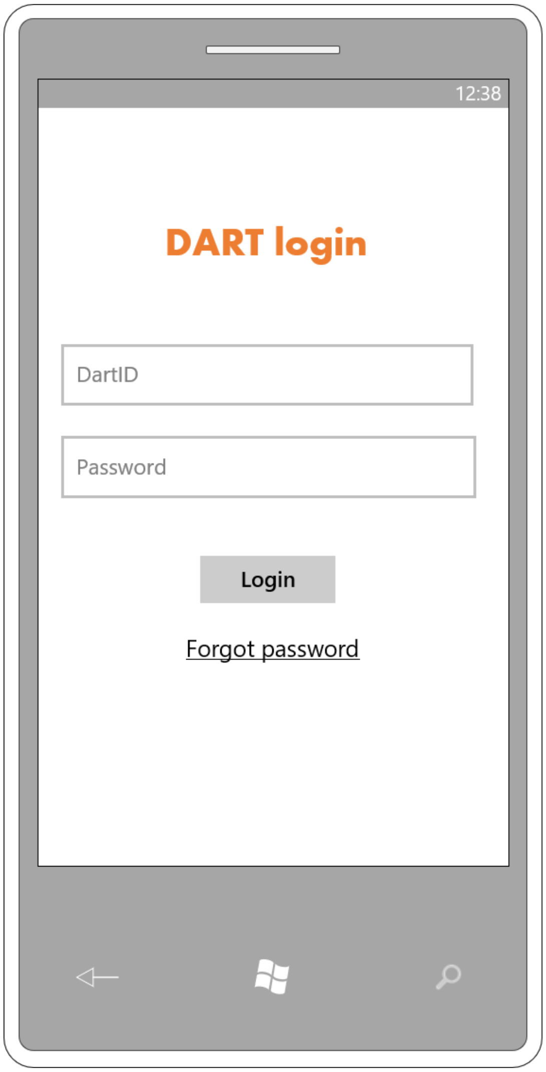
12:38

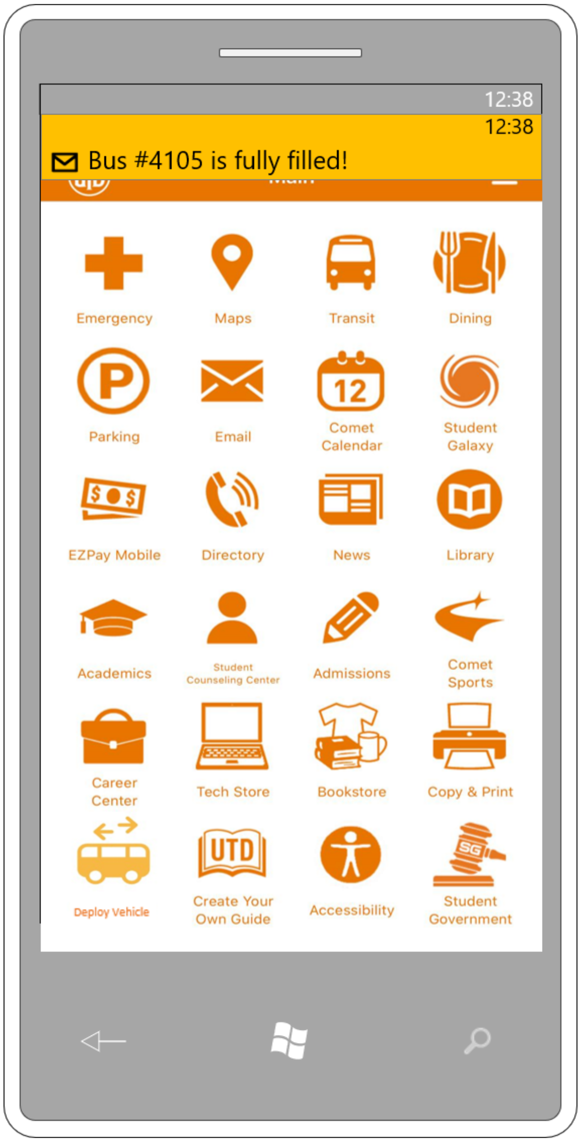
Bus time prediction

Your recommended bus time is:
12:18pm

Group - 4


12:38
Main
Emergency
Maps
Transit
Dining
Parking
Email
Comet Calendar
Student Galaxy
EZPay Mobile
Directory
News
Library
Academics
Student Counseling Center
Admissions
Comet Sports
Career Center
Tech Store
Bookstore
Copy & Print
Deploy Vehicle
Create Your Own Guide
Accessibility
Student Government

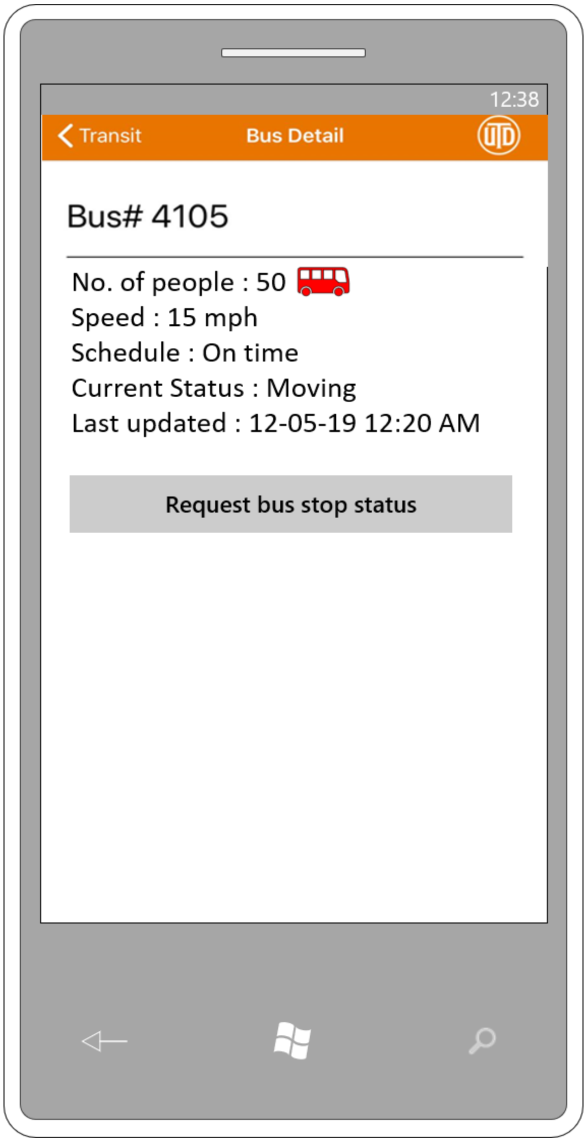

12:38
DART login
DartID
Password
Login
Forgot password



12:38

12:38

Bus #4105 is fully filled!



12:38

Bus# 4105

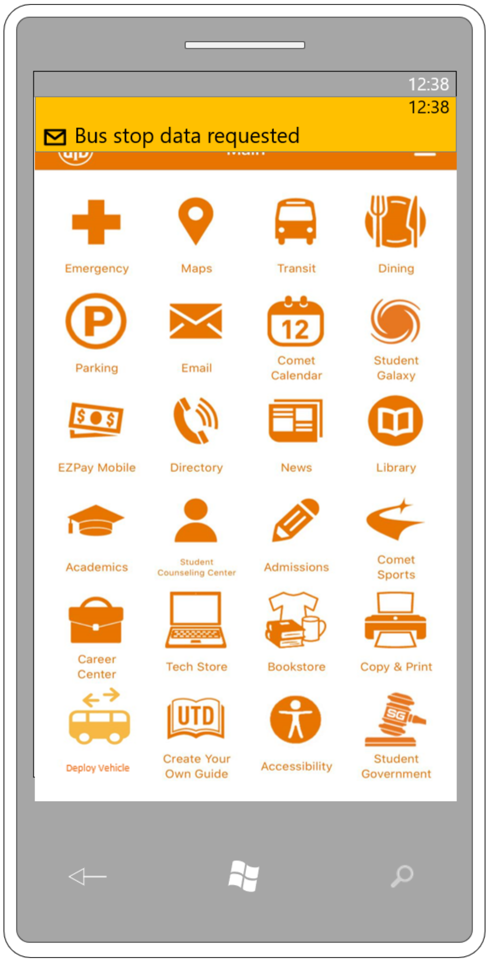
No. of people : 50

Speed : 15 mph

Schedule : On time

Current Status : Moving

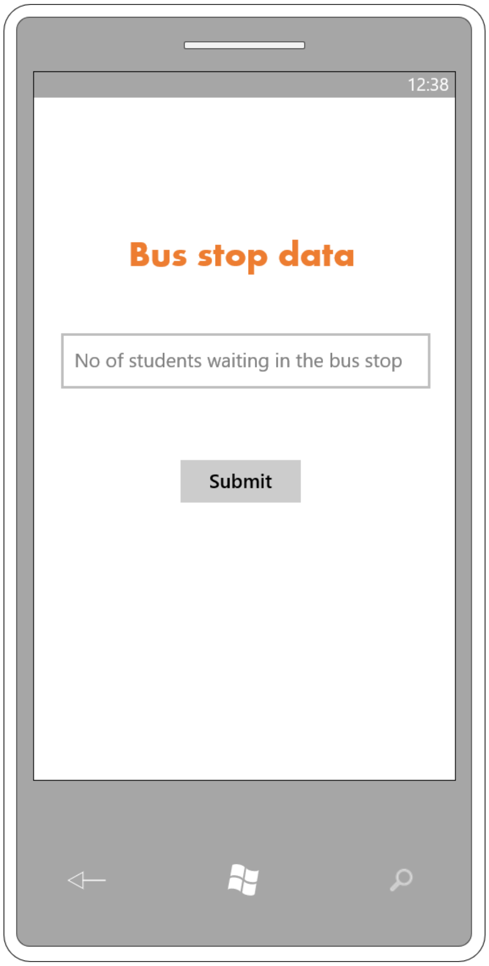
Last updated : 12-05-19 12:20 AM



12:38

12:38

Bus stop data requested

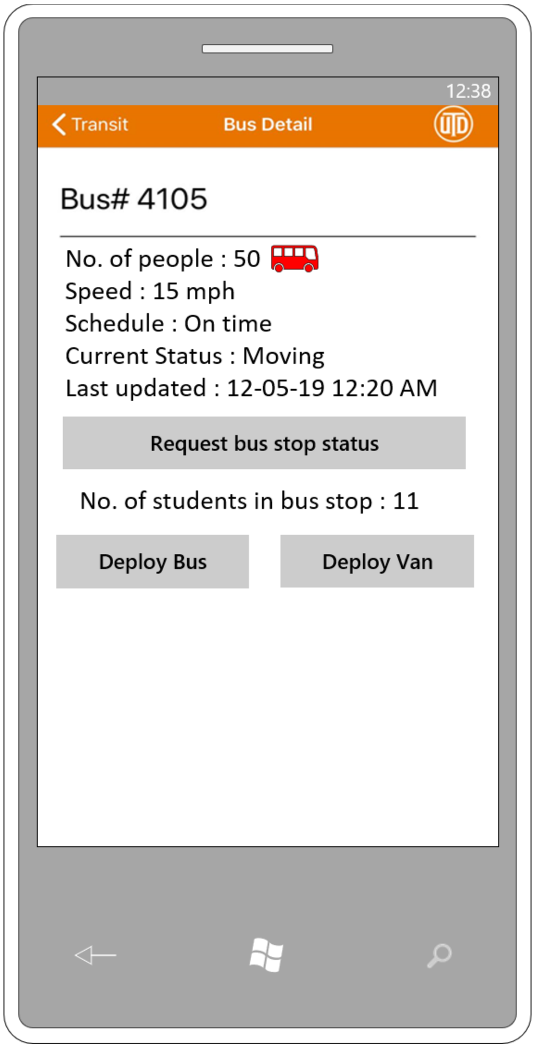


12:38

Bus stop data

No of students waiting in the bus stop

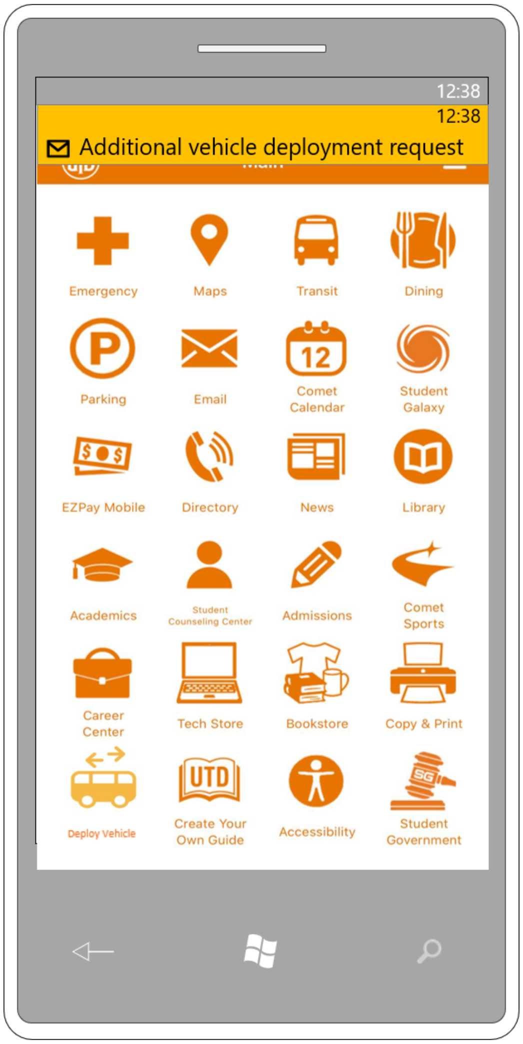
Group - 4



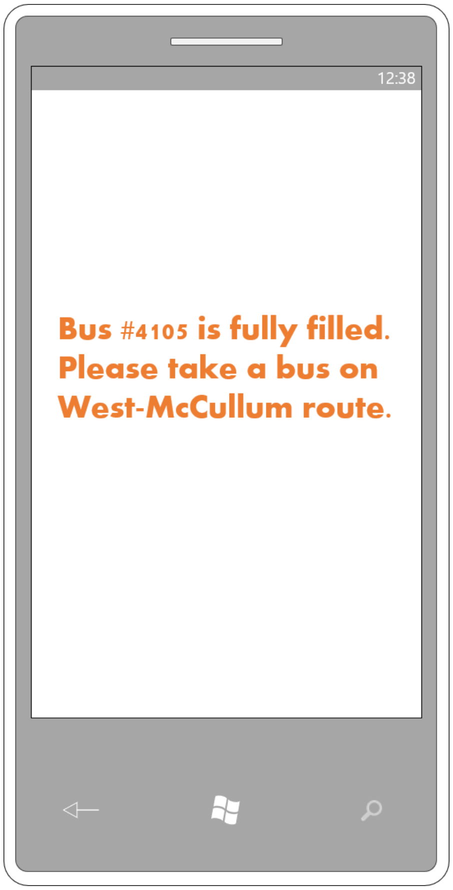
Bus# 4105

No. of people : 50
Speed : 15 mph
Schedule : On time
Current Status : Moving
Last updated : 12-05-19 12:20 AM

No. of students in bus stop : 11



Additional vehicle deployment request



Bus #4105 is fully filled.
Please take a bus on
West-McCullum route.

Database Design

Student(StudentNetID, StudentFirstName, StudentLastName, StudentAddress, StudentZipcode)

Primary key - StudentNetID

StudentAccount(StudentNetID, StudentUserName, StudentPassword, BusInfo)

Primary key - StudentUserName should be unique and non null.
Foreign key - StudentNetID must be present in the Student table.
BusInfo column can be null till it gets updated with the bus schedule if the student opts for it.

StudentSchedule(StudentScheduleID, StudentNetID, CourseID)

PrimaryKeys - StudentNetID and CourseID

The combination of both the keys should be unique.

CandidateKey - StudentScheduleID

Course(CourseID, CourseName, ClassTiming)

PrimaryKey - CourseID

Driver(DriverID, DriverFirstName, DriverLastName)

DriverID is the primary key

Schedule(DriverShiftNo, DriverID, Timing, VehicleNo, BusRoute)

Primary key - DriverShiftNo

Foreign key - DriverID, VehicleNo

DARTOfficial(OfficialID, OfficialFirstName, OfficialLastName)

Primary key - Official ID

Vehicle(VehicleNo, VehicleCapacity, SensorID)

Primary key - VehicleNo

Foreign key - Sensor ID

Sensor(SensorID, SensorStatus, PurchaseDate, SensorData)

Primary key - SensorID

VehicleRequest(RequestID, VehicleNo, RequestLocation, OfficialID, Status)

Primary key - Request ID

Foreign keys - VehicleNo, OfficialID

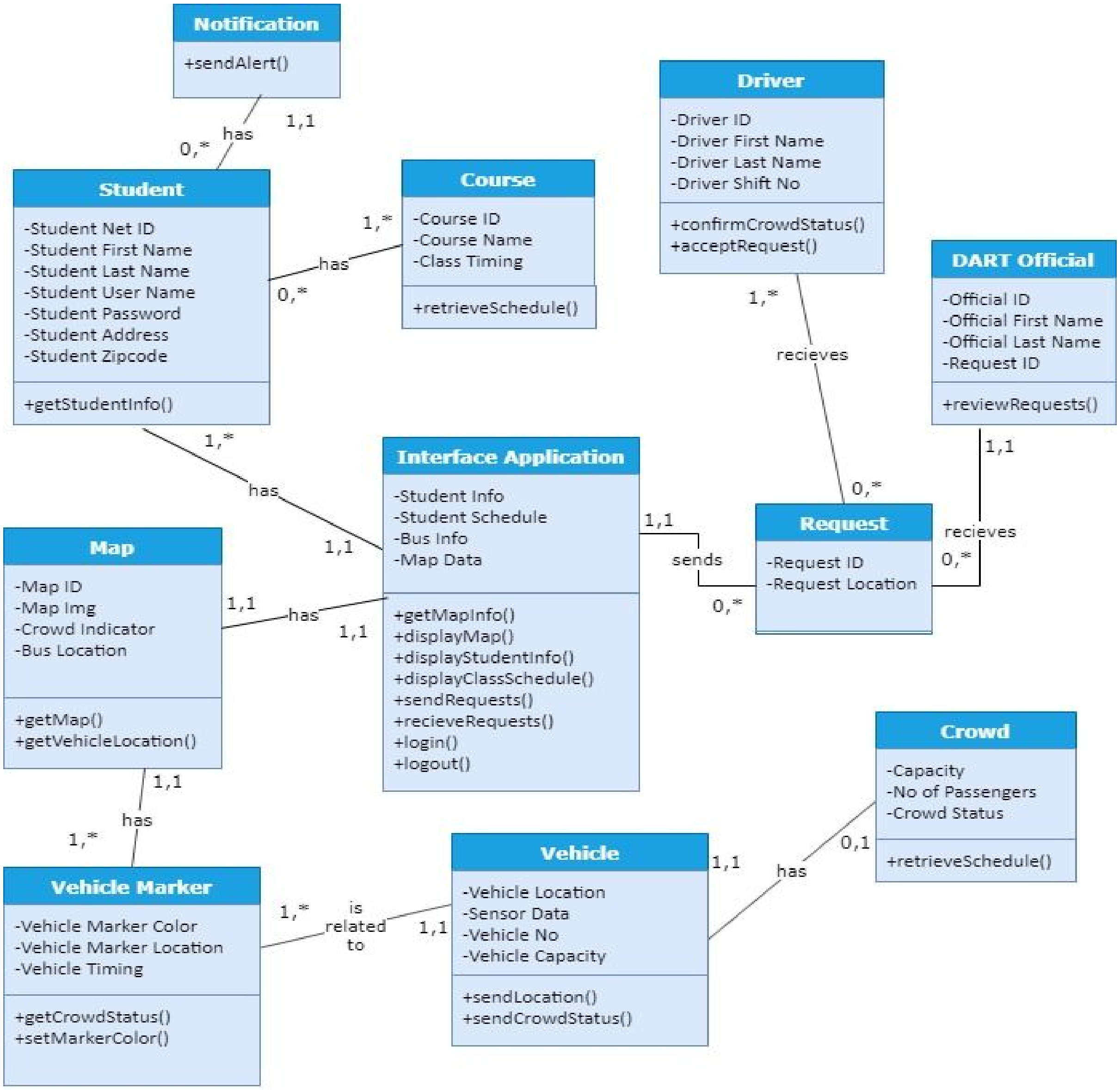
VehicleMarker(MarkerEntryID, VehicleNo, VehicleMarkerColor, VehicleLocation, CurrentNoOfPassengers)

Primary key - MarkerEntryID

Foreign key - VehicleNo

This table is recorded for every colour change for a particular vehicle.

Complete Class Diagram



Software Design

A) Method Name: getStudentInfo()

Class name: Student

Clients (Consumers): Students, UTD Interface Application

Associated Use Cases: Display My Schedule

Description of Responsibilities: Obtains the weekly schedule of a particular student based on the courses that the student has taken

Arguments received: Course ID, Course Name, Class Timing, Student ID

Type of Value Returned: String Array

Pre-conditions:

1) Student has enrolled to at least one course

Post-Conditions:

1) The student's course schedule will be sent to the UTD interface app

Psuedocode:

```
Student Course Timing = retrieveSchedule()

IF Number of Courses >= 1

{

FOR All the Courses Selected

{
```

```
ADD Student Course Timing INTO Course Schedule
}
}
ELSE
{
Student Not eligible to Receive Notifications
}
RETURN Course Schedule
```

B) Method Name: determineCrowdStatus()

Class name: Crowd

Clients (Consumers): Passengers, Vehicle

Associated Use Cases: Sensor Data

Description of Responsibilities: Obtains the number of people in the bus and determines the crowd status based on the capacity of the bus

Arguments received: Capacity, Number of Passengers

Type of Value Returned: String

Pre-conditions:

1) Sensor counts the number of people entering and exiting the bus

2) Based on the people entering and exiting, the count in the bus needs to be determined

Post-Conditions:

1) The color of the Vehicle Marker will be determined based on crowd status

Psuedocode:

```
IF Number of Passengers == Crowd Capacity
{
Crowd Status = Full
}
ELSE IF Number of Passengers >= 0.75*Crowd Capacity AND <= Crowd Capacity
{
Crowd Status = Almost Full
}
ELSE IF Number of Passengers < 0.75*Crowd Capacity AND >= 0
{
Crowd Status = Empty
}
Return Crowd Status
```

C) Method Name: setMarkerColor()

Class name: Vehicle Marker

Clients (Consumers): Interface Application, Students

Associated Use Cases: Process Bus Status Data

Description of Responsibilities: Obtains crowd status data and accordingly modifies the color of the bus icon

Arguments received: Crowd Status

Type of Value Returned: String

Pre-conditions:

1) Sensor counts the number of people entering and exiting the bus and provides the info

2) Colors for different crowd status' need to be finalized

Post-Conditions:

1) The color of the Vehicle Marker will be determined based on crowd status

Psuedocode:

```
Switch(Crowd Status)

Case: Full

{

Color = Red;

}

Case: Almost Full

{

Color = Yellow

}

Case: Empty

{

Color = Green

}

Case: Default

{

Color = Black
}

Return Color
```

D) Method Name: sendRequests()

Class name: Vehicle Marker

Clients (Consumers): Interface Application, DART Officials

Associated Use Cases: Process Bus Status Data

Description of Responsibilities: Obtains crowd status data and sends the request from the interface application to the DART Official to deploy an additional bus

Arguments received: Crowd Status, DART Official ID

Type of Value Returned: String

Pre-conditions:

1) Sensor counts the number of people entering and exiting the bus and provides the info

2) DART Official contact details needs to be retrieved

Post-Conditions:

1) The DART official will be contacted

Psuedocode:

```
If (Crowd Status == Full)

{

ProcessRequest();

}
```

E) Method Name: getMapInfo()

Class name: Interface Application

Clients (Consumers): Interface Application, Students

Associated Use Cases: Process Bus Stop Data

Description of Responsibilities: Updates map data everytime map info is obtained. Obtains bus location, crowd status and displays the bus location on the map. It also shows the colour of the bus depending upon the crowd capacity

Arguments received: Map Img

Type of Value Returned:

Pre-conditions:

1) Crowd status must be calculated and sent to the map to be processed

2) Bus location should be sent to the map

Post-Conditions:

1) The map should display the bus location live and the crowd status updated regularly

Pseudocode:

```
Define MapObject

IF(MapData != NULL)

{

mapObject = mapData.getUpdatedInfo();

}

Return mapObject;
```

Future enhancements

1. Install sensors at major bus stops to automate the crowd density check process.
2. Improve the system by predicting bus part failures. This will avoid any bus breakdowns.
3. Improve transit management interface for UTD Dart Officials by creating a standalone app.

Project Management Deliverables

The following project management details should be included in the report as well.

1. **Project Activities:**

   Project Problem statement:
   Project Planning:
   BPMN Models:
   Context Diagram
   Process Model : Use case diagram and Use case descriptions
   Data Model without functions
   Object Behavior Model : A sequence Diagram
   Data Dictionary:
   Functional Specification Document:
   Interface Design:
   Database Design:
   Complete Class Diagram:
   Software Design:
   Minutes of Meeting:
   Presentation:

2. **Allocation of activities:**

   Project Problem statement: Everyone
   Project Planning: Everyone
   BPMN Models: Ayush
   Context Diagram: Shruthi
   Process Model :

Use case diagram : Surya
Use case descriptions : Nishi, Shruthi
Data Model without functions : Prashanth
Object Behavior Model : Sequence Diagrams : Nishi
Data Dictionary: Surya
Functional Specification Document: Surya, Ayush
Interface Design: Nishi
Database Design: Shruthi
Class Diagram: Prashanth
Software Design: Ayush Singh
Minutes of Meeting: Shruthi, Surya
Presentation: Prashanth

3. **Planned timeline:**

September:
Discuss the project ideas
Finalize our project idea with the professor
Get the project started with the functional specifications

October:
Prepare the BPMN and context diagrams
Prepare the Use case diagram and descriptions
Ready the database design and class diagram
Prepare the data dictionary for the project

November:
Prepare the contracts for the database
Prepare the sequence diagrams of the project
Prepare the Interface design of the project
Start preparing the project report

December:
Finish the project report
Prepare the PPT and record our presentation

4. **Execution timeline:**

September:
Discuss the project ideas

Finalize our project idea with the professor
Get the project started with the functional requirements

October:

Prepare the BPMN and context diagrams
Prepare the Use case diagram and descriptions

November:

Prepare the data dictionary for the project
Ready the database design and class diagram
Prepare the contracts for the database
Prepare the sequence diagrams of the project
Prepare the Interface design of the project
Start preparing the project report

December:

Finish the project report
Prepare the PPT and record our presentation

Minutes of Meeting

Meeting No: 1
Date: Sept 11 - 2019
Time Spent: 3 hours

Attendees

Ayush Singh, Nishi Shah, Prashanth Ramanathan, Shruthi Gopal, Surya Varma

Agenda

Project selection

- Come up with different project ideas and choose one which satisfies the SAPM project guidelines.

Activities

Project selection

- Each team member gave inputs on possible ideas for the project.
- Brainstormed on different project ideas and analyzed each one of them with project guidelines.
- We shortlisted to three ideas which were equally good.
- Documented the problem statements for the project ideas to discuss further with Professor.

Next Meeting plan:

September 16, 2019.

Next Meeting Agenda Items

- Meet with the professor to present the three ideas and to fix the final one .

Meeting No: 2
Date: Sept 16 - 2019
Time Spent: 1 hour

Attendees

Ayush Singh, Nishi Shah, Prashanth Ramanadhan, Surya Varma

Agenda

Project selection

- Meet with the professor to present the three ideas and to fix the final one .

Activities

Project selection

- Presented the three problem statements for the project ideas and had a discussion with the professor on those fulfilling the project guidelines.
- We decided to go with (-topic-) with 3 major functionalities.
- We also added extra features to our system which would further enhance our project.

Next Meeting plan:

September 29, 2019.

Next Meeting Agenda Items

- Functional requirement of the projects.

Meeting No: 3
Date: Sept 29th - 2019
Time Spent: 2 hours

Attendees

Ayush Singh, Nishi Shah, Prashanth Ramanathan, Shruthi Gopal, Surya Varma

Agenda

Functional Requirements

- The team came together and brainstormed on the functional requirements.
- The team completed the functional requirements.

Next Meeting

October 15th, 2019

Next Meeting Agenda Items

- BPMN and Context diagrams.

Meeting No: 4
Date: October 15th - 2019
Time Spent: 2 hours

Attendees

Ayush Singh, Nishi Shah, Prashanth Ramanathan, Shruthi Gopal, Surya Varma

Agenda

BPMN diagrams

- Come up with BPMN diagram for the 3 features that we are trying to incorporate in our system

Context Diagram

- The team worked together to come up with the context diagram.

Next Meeting

October 25th, 2019

Next Meeting Agenda Items

- Use case diagram and use case descriptions.

Meeting No: 5
Date: October 25th - 2019
Time Spent: 2 hours

Attendees

Ayush Singh, Nishi Shah, Prashanth Ramanathan, Shruthi Gopal, Surya Varma

Agenda

Use case diagram

- The team worked together to come up with a use case diagram.

Use case description

- Started tackling the major use cases.
- Documented the major use cases

Next Meeting

November 1st, 2019.

Next Meeting Agenda Items

- Continue with use case description
- Start working on Data dictionary

Meeting No: 6
Date: November 1st - 2019
Time Spent: 3 hours

Attendees

Ayush Singh, Nishi Shah, Prashanth Ramanathan, Shruthi Gopal, Surya Varma

Agenda

Use Case Description

- Completed the use case descriptions.
- Completed the data dictionary.

Next Meeting

November 20th, 2019.

Next Meeting Agenda Items

- Class diagram

Meeting No: 7
Date: November 20th - 2019
Time Spent: 2 hours

Attendees

Ayush Singh, Nishi Shah, Prashanth Ramanathan, Shruthi Gopal, Surya Varma

Agenda

Class Diagram

- The team sat down and designed the class diagram. Every team member contributed to the completion of the diagram.

Next Meeting

November 23rd, 2019

Next Meeting Agenda Items

- Sequence Diagram, Database design

Meeting No: 8
Date: November 23rd - 2019
Time Spent: 2 hours

Attendees

Ayush Singh, Nishi Shah, Prashanth Ramanathan, Shruthi Gopal, Surya Varma

Agenda

Sequence Diagram

- The team started and completed the sequence diagram.
- The team arranged a meeting with the professor to verify the class and sequence diagrams.

Database Design

- The team worked on and completed the database design.

Next Meeting

November 25th, 2019.

Next Meeting Agenda Items

- Contracts

Meeting No: 9
Date: November 25th - 2019
Time Spent: 2 hours

Attendees

Ayush Singh, Nishi Shah, Prashanth Ramanathan, Shruthi Gopal, Surya Varma

Agenda

Contracts

- The team worked on the contracts. Each team member contributed to the completion of the contracts.

Next Meeting

November 26th, 2019.

Next Meeting Agenda Items

- Interface design

Meeting No: 10
Date: November 26th - 2019
Time Spent: 3 hours

Attendees

Ayush Singh, Nishi Shah, Prashanth Ramanathan, Shruthi Gopal, Surya Varma

Agenda

Interface Design

- The team worked on the interface design.
- The team completed the mockups.

Next Meeting

December 1st, 2019.

Next Meeting Agenda Items

- Create PowerPoint Presentation.

Meeting No: 11
Date: December 1st- 2019
Time Spent: 3 hours

Attendees

Ayush Singh, Nishi Shah, Prashanth Ramanathan, Shruthi Gopal, Surya Varma

Agenda

PowerPoint Slides

- The team working on creating the PPT slides.

Next Meeting

December 2nd, 2019.

Next Meeting Agenda Items

- Continue with creating PowerPoint Presentation.

Meeting No: 12
Date: December 2nd - 2019
Time Spent: 2 hours

Attendees

Ayush Singh, Nishi Shah, Prashanth Ramanathan, Shruthi Gopal, Surya Varma

Agenda

PowerPoint Slides

- The team completed creating the PPT slides.

Next Meeting

December 4th, 2019.

Next Meeting Agenda Items

- Presentation Practice.

Meeting No: 13
Date: December 4th - 2019
Time Spent: 1 hour

Attendees

Ayush Singh, Nishi Shah, Prashanth Ramanathan, Shruthi Gopal, Surya Varma

Agenda

Presentation Practice and Recording

- The team recorded the presentation.

Next Meeting

No more meetings.

Next Meeting Agenda Items

- None