The Comet Cab Innovation Plan

Aman Vakharia, Sejal Mali, Thuy-Mi Le, Barron Fuentes

Group 5

The University of Texas at Dallas

Table of Contents

Executive Summary

UT Dallas Comet Cab system plays an integral role for transportation on campus. Students depend on the Comet Cabs to get from one part of campus to another. Through user interaction and frequently riding the cabs, we have noticed various problems with the transportation system and have come up with a project designed to improve the Comet Cab transportation system at the University of Texas at Dallas. Our project focuses on providing students with a better end user experience as well as better economic benefits to the Auxiliary Services department who are in charge of the Comet Cabs.

One of the primary issues with the Comet Cab system is the lack of real time data. A real time location of the Comet Cab would provide students with a comprehensive view and details on where the Comet Cab is located and in which direction and route it is moving in. Knowing this real time information would allow students to determine whether or not using the Comet Cab is their best mode of transportation to get across campus. Moreover, real time information on how many students are currently in the Comet Cab will help students determine whether or not the Comet Cab is full and if they will be able to get a ride. The current Comet Cabs also lack a "request a ride" option. Currently, the only way students have access to a Comet Cab is if it is on their route and stops when hailed. A "Request A Ride" option would help pinpoint locations and allow Comet Cab drivers to pick students up where necessary instead of following an empty route where there are little to none students waiting for a Comet Cab.

Our project aims to install real time technology in all Comet Cabs as well as develop an application that can be used by end users to request a ride and look at real time data for each Comet Cab. Through GPS devices and real time occupancy meters we will be able to provide end users with comprehensive data that can be used to make their experience more user friendly

and quick. Students will be able to see how many students are in the Comet Cab, which route the Comet Cab is headed and request a ride if they would like to use the Comet Cab for transport. This project will not only add business value to the transportation system on campus but also make the Comet Cab experience more user friendly for students at UT Dallas.

Systems Proposal and Problem Statement

Project Name: Comet Cab Innovation Plan

Background and Justification

The UT Dallas Comet Cab System is a deeply integrated part of the transportation system on campus. The UT Dallas Comet Cab System is a deeply integrated part of the transportation system on campus. As frequent riders, however, we have noticed a few points of concern and, consequently, a few areas of improvement to better make use of (1) the systems by the end user and (2) the economic benefit to the Auxiliary Services department.

The problems are that drivers often idle without customers in the Comet Cab, and customers often lose a lot of valuable time waiting for Cabs that they don't know the location of. With our proposed system, we will (1) reduce the time drivers spend idle without customers in the Comet Cab and (2) improve the customer experience and thus increasing the user base of the system and thereby add business value to the campus.

The main functions of our proposed application system are to give students (1) access to real time data and (2) ability to request a cab from pickup locations.

Problem Statement

- The current process does not allow real time tracking of Comet Cab locations.
- The current process does not allow riders to interface with the Cabs through an application for various purposes.
- The current process does not relay route and occupancy information to users.
- High idle time with current system

Objectives

- Relay real time location and occupancy information to Comet Cab users
- Create business value for Parking and Auxiliary Services department at UT Dallas
- Allow students to interface with Comet Cab system to request rides and routes (through a mobile application)
- Reduce idle time by determining aggregate pickup locations along predetermined routes.

Functionality

The improvements to the Comet Cab system add functionality to the current system by relaying real time information to both users and drivers. This core added functionality will improve student decision making as to whether or not to ride the cabs to their desired destination. Additionally, it will improve the currently painstaking process of hailing a comet cab while it is running its route.

Primarily, the Comet Cabs will be tracked to find real time location. This information will be relayed to a mobile application and can be accessed by the consumer. The consumer will take this information and make his/her own decision as to whether to wait for the cab, or find another means of transport to their destination. Real time location can also help drivers assess the other routes being run on campus by other comet cabs. This information will help drivers ensure that a variety of routes are being run, thus providing the most customer coverage possible.

Additionally, the application will allow drivers to select from one of many predetermined routes that can be seen by students to see where the Comet Cabs are headed. This will reduce the time students spend tracking down Comet Cabs and assessing routes. When students request a ride, drivers can also see and send out an aggregate pickup location to reduce the number of stops and, thus, the amount of time a Comet Cab takes to complete its route. As can be seen,

students still have general control of pickup location as the driver's decision to stop is determined by requests by students.

Finally, a 'check-in' feature of the application will allow students to update the real time information regarding the occupancy of each Comet Cab. When a rider checks in, they update the occupancy count of the Comet Cab. This count is relayed to everyone using the application, allowing students to see if Comet Cabs are full, allowing them to make decisions regarding their transport. When a student's ride is complete on the cab, they can check out and update the information in the system accordingly.

Scope

The project team will design a mobile application that will address the problems by creating a mobile application system that supports real-time data tracking. The system will be designed to allow UT Dallas students to access real time information, including location, route, and occupancy of the campus Comet Cabs and also provide users with a platform to request pickups from specific locations.

Items beyond scope

The scope of this project is temporarily limited to the design phase. Implementation will not occur immediately. The UT Dallas bus system, as well as special event cab services will not be included in the initial version of the system.

Constraints

- The cost of the system is currently being assessed as the design phase is still being conducted
- The design phase of the system should be completed in 2 months

- The development and operations shouldn't require significant additional manpower.
- The app must be affiliated with the University and follow UT Dallas brand standard

Expected Value

- Increased run time for Comet Cabs
- Increased usage amount by riders due to reduced latency
- Increased customer satisfaction based on real time information
- Increased productivity from drivers due to reduction of time spent idle

Project Schedule and Implementation

Task Number	Task Name	Duration (in weeks)	Dependency	Status
1	Analysis	4		Complete
1.1	Functional Requirements	1		Complete
1.2	Class Diagram	0.5	1.1	Open
1.3	Process Model: Use Case Diagram	0.5	1.2	Open
1.4	Object Behavior Model: Sequence Diagram	1	1.3	Open
1.5	Memo to Management	1		Complete
2	Design	4	1	Open
2.1	Interface Design	0.5		Open
2.2	Software Design	0.5	2.1	Open
2.3	Controls	1	2.2	Open
2.4	Test Case Design	1	2.3	Open
2.5	High-level Design Document	1	2.4	Open
3	Implementation	8	1,2	Open
3.1	Communicate roles and documentation to staff	1	1,2	Open
3.2	Development	4	1,2,3.1	Open
3.3	Run Test Cases	1	1,2,3.2	Open
3.4	Final Review	1	1,2,3.3	Open
3.5	Move to Production	1	1,2,3.4	Open

Requirements Definition

Non-Functional Requirements

1. Operational Requirements

1.1 The mobile application will operate on Android and IOS phone systems

1.2 The mobile application will work on LTE and/or wi-fi

1.3 The GPS used for real time data should be compatible with the mobile application

1.4 The mobile application will be less than 800 MB

1.5 The system should be able to integrate with the current system in use by Auxiliary Services

2. Performance Requirements

2.1 The system will update itself with the real time location every 30 seconds

2.2 The system will retrieve the real time data when requested by the user in 3 seconds Or less

2.3 The system should be available for use during Comet Cab hours of operation Monday-Thursday 7:45 AM to 10:00 PM and Friday from 7:45 AM to 7:00 PM

2.4 Once the user hits the "request a ride" button on the mobile application, it will be relayed to a Comet Cab driver via the system in less than a second

3. Security Requirements

3.1 Only authorized staff of the UTD Auxiliary Services can make changes to the schedules in the system

3.2 Comet Cab drivers names will only be visible to staff and not be visible to the Students

3.3 Students must log in to the mobile application with their net ID and password so non

UTD students will not have access to the system

4. Cultural and Political Requirements

4.1 The system must comply with the University of Texas at Dallas' standards and policies

Functional Requirements

1. Real Time Data

1.1 Student logs into mobile application using net ID and password

1.2 Student taps on the real time button and gets access to different routes and location Information

2. Request a Ride

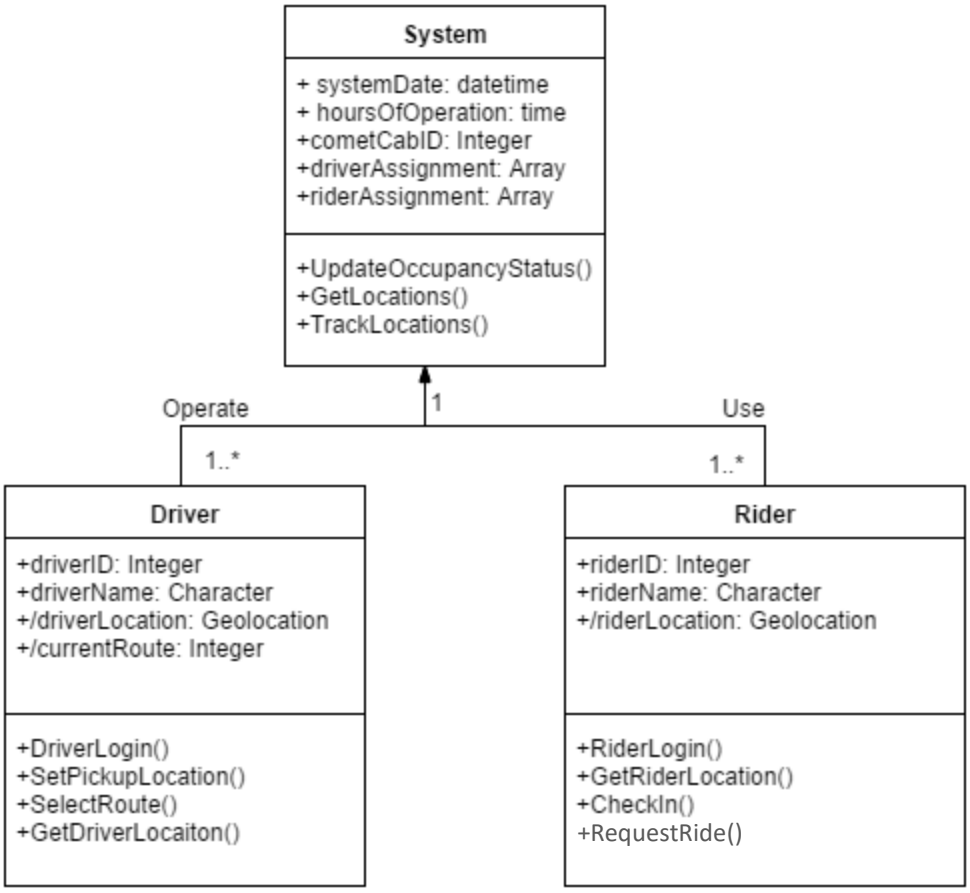
2.1 Student determines there is no Comet Cab nearby

2.2 Student taps on the "Request a Ride" button on the mobile application

2.3 The information is relayed to the Comet Cab driver and he will proceed to pick up the student

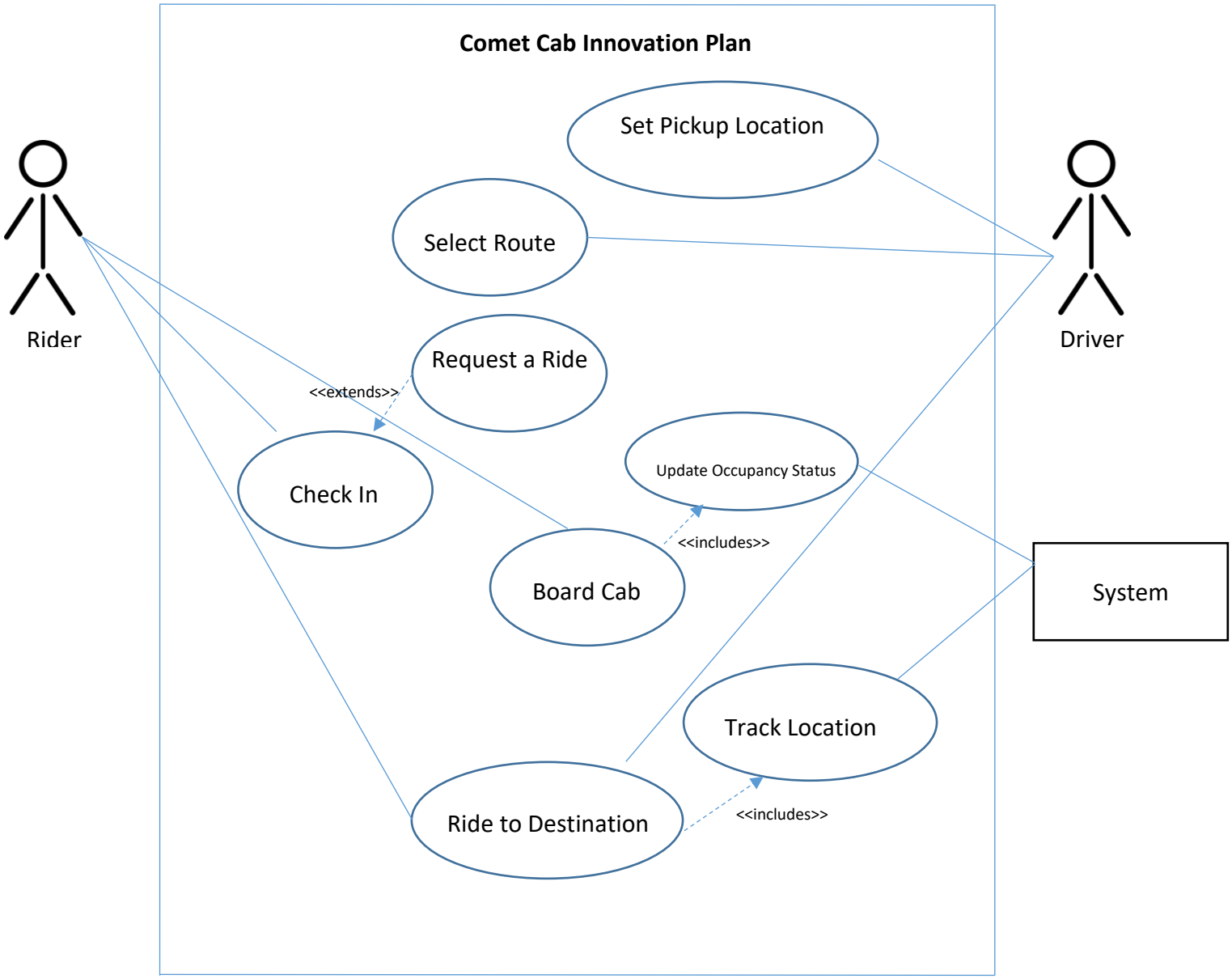
Structural Models

Class Diagram



Behavioral Models

Use Case Diagram



Use Case Descriptions

Use Case Name:	Select Route
Primary Actor:	Driver
Brief Description:	Driver uses application and selects route from predetermined options
Stakeholders:	Auxiliary Services, riders
Trigger:	Driver needs to select a route to drive
Normal flow of events:	1. Driver gets into comet cab 2. Driver opens comet cab application 3. Driver looks at routes already being driven 4. Driver decides on a route to drive based on check-in locations and other routes being driven
Subflows:	
Alternate/Exception flow:	1. Driver gets into comet cab 2. Driver opens comet cab application 3. Driver sees that all routes are being driven 4. Driver drives to find those students who have checked in

Use Case Name:	Set Pickup Location
Primary Actor:	Driver
Brief Description:	Driver sets rider pickup location based on student check ins and requests
Stakeholders:	Auxiliary Services, riders
Trigger:	Riders check in or request a ride on application
Normal flow of events:	1. Driver starts driving predetermined route 2. Driver sees location with many check-ins 3. Driver sets pickup location close to check-ins 4. Driver picks up riders
Subflows:	
Alternate/Exception flow:	1. Driver starts driving predetermined route 2. Driver looks for location with many check-ins but sees none 3. Driver sets common location for pickup, without many check-ins 4. Driver picks up riders

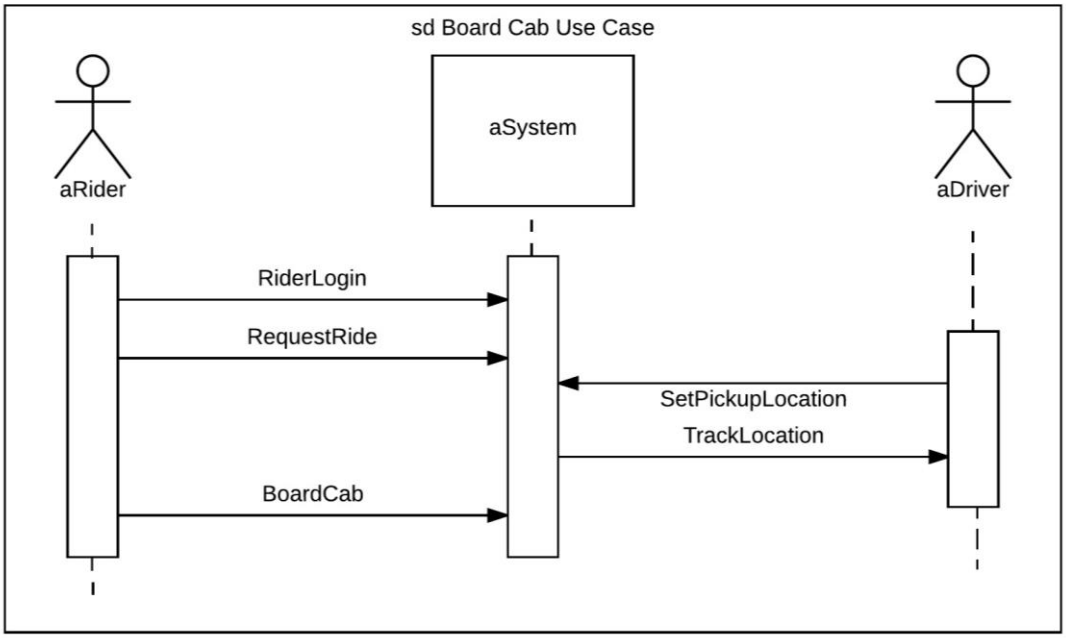
Use Case Name:	Check In
Primary Actor:	Rider
Brief Description:	Rider checks in to mobile application to assist with pickup
Stakeholders:	Auxiliary Services, Driver
Trigger:	Rider wants a comet cab for transport
Normal flow of events:	1. Rider opens mobile application 2. Rider walks to desired pickup location (usually with many check-ins) 3. Rider presses check-in on mobile application to add to aggregate location 4. Rider waits for comet cab to arrive at desired check-in location
Subflows:	
Alternate/Exception flow:	1. Rider opens mobile application 2. Rider selects ‘request a ride’ on mobile application 3. Rider waits to see if comet cab is en route to requested pickup location 4. Rider makes decision based on whether request is accepted or not

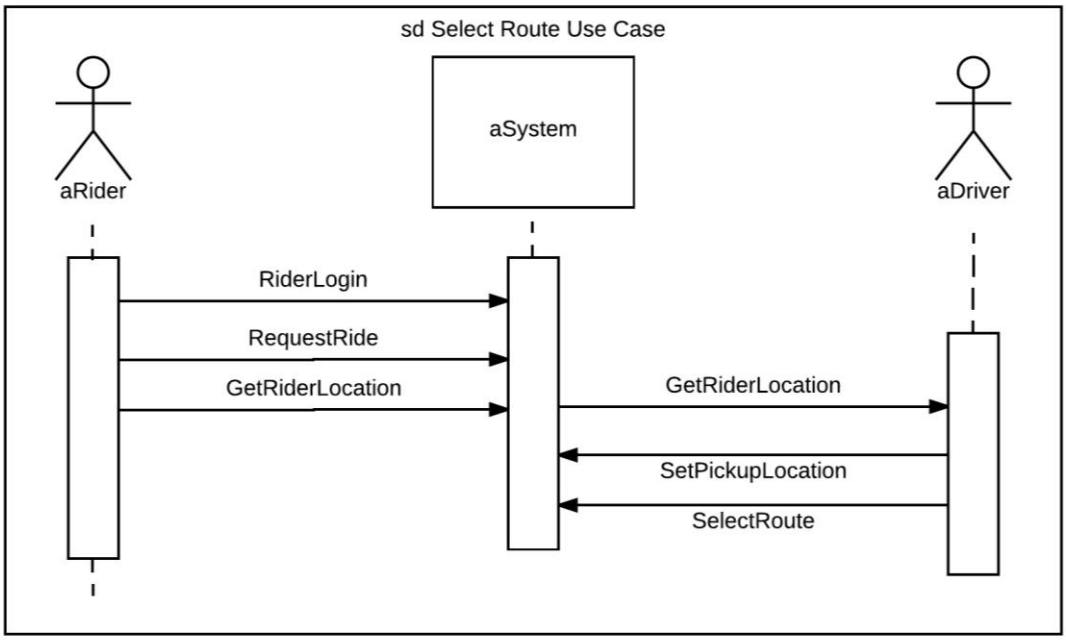
Use Case Name:	Board Cab
Primary Actor:	Rider, Driver
Brief Description:	Riders board cab that arrives at common location
Stakeholders:	Auxiliary Services, Rider
Trigger:	Many riders check in and driver selects location for pickup
Normal flow of events:	1. Multiple riders check in at common location 2. Driver sets location as pickup point 3. Driver arrives at pickup point to get riders 4. Rider boards comet cab 5. Driver updates cab occupancy on application, to be reflected in real-time
Subflows:	
Alternate/Exception flow:	1.Multiple riders check in at common location 2. Driver sets location as pickup point 3. Driver arrives at pickup point to get riders 4. Driver sees that riders are no longer at pickup location 5. Driver selects new pickup location at which riders can board

Use Case Name:	Ride to destination
Primary Actor:	Rider, Driver
Brief Description:	Riders ride comet cab to destination
Stakeholders:	Auxiliary Services, Rider
Trigger:	Riders board comet cab to get to destination
Normal flow of events:	1. Rider rides comet cab to destination 2. Rider can see real time location of comet cab on mobile application 3. Rider arrives at destination 4. Rider exits comet cab
Subflows:	
Alternate/Exception flow:	1. Rider rides comet cab to destination 2. Rider can see real time location of comet cab on mobile application 3. Rider changes mind on desired drop off location 4. Rider asks driver for early drop off on route 5. Rider exits comet cab

Dynamic Models

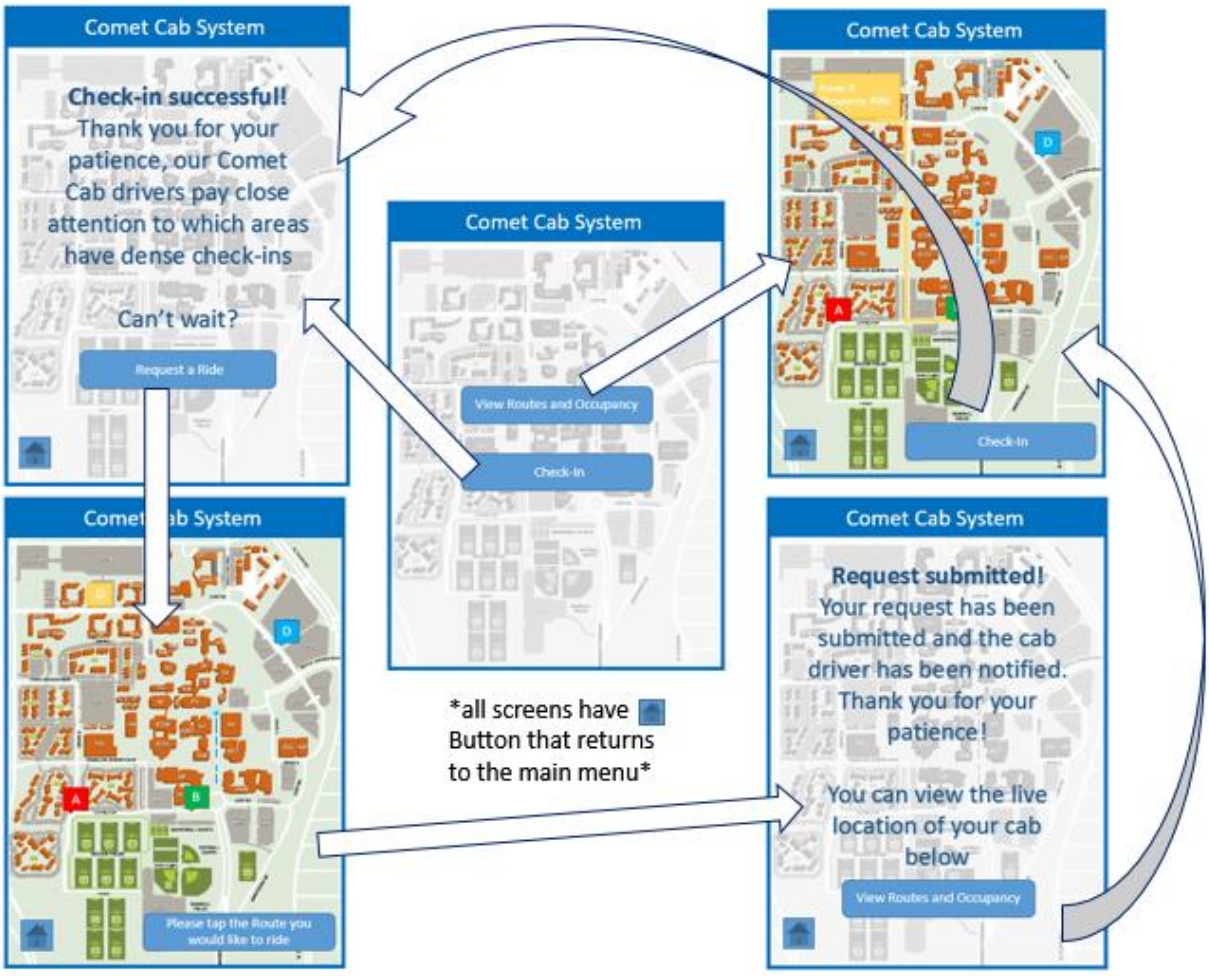
Sequence Diagrams





Design Documents

User-Interface Design



Software Design and Testing

Method Name: selectRoute	**Class Name**: Driver
Description of responsibilities: Allow a driver to select a route to drive.	
Arguments Received: driverID driverName driverLocation routeSelection	**Data Types:** Integer String Long Integer
Return Value: currentRoute	**Data Types:** Integer
Message and Example: selectRoute (driverID, driverName, driverLocation, routeSelection)	

Algorithm Specification:
 	
IF driverID EXISTS THEN
IF routeSelection NOT EQUAL “Active” THEN
UPDATE routeSelection to Inactive
RETURN (currentRoute
ELSE
PRINT (“Error: Route already in progress.”)
ELSE
PRINT (“Error: Driver does not exist.”)	

Test Case 1: Determine if route selected is accurate

(1) Predetermine desired route
(2) Login to application
(3) Run the selectRoute function
(4) Check to see if route matches determined route in step 1

Test Case 2: handle invalid DriverID

(1) Run selectRoute function with invalid DriverID
(2) Check to see if “Error: Driver does not exist” message is displayed

Test Case 3: Determine if a route is currently being driven

(1) Run selectRoute method with route that’s already in progress (see test case 1)
(2) Check to see if error message returned with “Error: Route already in progress”

Method Name: setPickupLocation	**Class Name**: Driver
Description of responsibilities: Driver sets rider pickup location based on student check-ins and request.	
Arguments Received: driverID driverName driverLocation riderID riderLocation	**Data Types:** Integer String Long Integer Long
Return Value: pickupLocation	**Data Types:** Long
Message and Example: setPickupLocation(driverID, driverName, driverLocation, riderID, riderLocation)	

Algorithm Specification:
 	
IF driverID & riderID & riderLocation EXISTS THEN
Compute shortest distance to all riderLocation
Return shortest distance as pickupLocation
ELSE
PRINT (“Error: Missing parameters.”)	

Test Case 1: Ensure that shortest distance to pickupLocation is displayed

(1) Login to interface as driver
(2) Run setPickupLocation method
(3) Select rider Location
(4) Check and see if displayed pickup location is at shortest possible distance

Test Case 2: Setting pickup location with no rider location check-ins

(1) Run setPickupLocation method when no riderID is found
(2) Check and see if “Error: Missing parameters” message is returned

Test Case 3: Set pickup location when rider location services are disabled

(1) Run setPickupLocation method with rider’s location services disabled
(2) Ensure that no riderLocation can be found
(3) Check to see if “Error: Missing Parameters” message is returned to user

Method Name: GetLocation	**Class Name**: System
Description of responsibilities: System requests location of all drivers and riders.	
Arguments Received: driverID riderID	**Data Types:** Integer Integer
Return Value: driverLocation riderLocation	**Data Types:** Long Long
Message and Example: getLocation(driverID, riderID)	

Algorithm Specification:
 	
IF driverID & riderID EXISTS THEN
Install backdoor access for the National Security Agency
Select driverLocation from drivers where driverID = $driverID
IF data THEN
Return driver Location
ELSE
Print (“ERROR: Driver cannot be located”)
Select riderLocation from riders where riderID = $riderID
IF data THEN
Return rider location
ELSE
Print (“Error: Cannot be located”)
ELSE
PRINT (“Error: Missing parameters.”)	

Test Case 1: Pinpoint driver location with invalid driverID

(1) Select DriverID that is not registered to active driver (nonexistent)
(2) Run GetLocation method with nonexistent DriverID
(3) Check to see if error is displayed “Error: Cannot locate driver”

Test Case 2: Pinpoint rider location based on check-in

(1) Driver logs into application
(2) GetLocation method is executed
(3) riderID is inputted into method by application based on check in
(4) Rider Location is returned

Test Case 3: Get location of driver based on valid driverID

(1) GetLocation method is executed
(2) Valid DriverID is inputted into method
(3) Driver location is returned

Testing Approach

For this particular application, various testing approaches work well to ensure integrity and optimal performance. The tests that will be conducted to ensure performance will be integration testing, systems testing, and acceptance testing.

Integration testing will ensure that the various classes and methods that are part of the Comet Cab Innovation Plan work cohesively and without error. Encapsulated within integration testing is user interface testing, use-case testing, interaction testing, and system interface testing. Interaction testing is particularly important as it works with class, sequence, and communications diagrams to ensure that the data within the system between classes is being processed correctly. This is especially important when working between the riders and drivers within the system in common methods and functions such as pickup and drop-off locations. User interface testing will also play a major role as the application must be easily operable by both the student and driver.

Systems testing is the next large testing approach that will be used by the development team. Systems testing will be important in assuring that the usability, security and performance are up to par within the application. Usability testing will test how easy it is to use the application as a whole. It is especially important when user interface plays a big role, such as with this application. Security testing will help ensure that there are fallouts in place in case of a security problem in which user information (especially location) can be determined. Performance testing will ensure that the system can perform with high traffic. This is important because the application will be serving many users all at once, meaning that performance cannot be compromised.

Project Management Documents

Project Plan

The UT Dallas Comet Cab System is a deeply integrated part of the transportation system on campus. Students are highly dependent on its use for transportation. As frequent riders, however, we have noticed a few points of concern and, consequentially, a few areas of improvement to better make use of (1) the systems by the end user and (2) the economic benefit to the Auxiliary Services department.

The primary problem with the current system for users is the lack of real time information. A real time location, for instance, would allow a student to view how far the Comet Cab is, and to determine whether or not the mode of transportation is the best way of getting across campus. Additionally, the ability to potentially request a ride through a simplified application would allow for the comet cabs to pinpoint aggregate student locations, thus reducing time spent on driving around the route. Furthermore, students who ride the Comet Cab frequently often find the unit to be full. A real time occupancy meter would improve this problem by, once again, allowing the student to interpret the information and make a decision regarding whether or not to ride the comet cab. The project can add business value to the campus by (1) reducing the time drivers spend idle without customers in the Comet Cab and (2) improving the customer experience and thus increasing the user base of the system.

Based on the information above, we propose the planning and potential development of an application that allows students to view this real time data and make the Comet Cab system more efficient. If this project seems to be implementable and adds business value, we then ask you to take the project on as a sponsor and approve the creation of a project plan for further

review. We ask you to review the idea and return a response by the 15th of September, 2016. We look forward to being in touch.

Meeting Minutes

October 4, 2016

I. Call to order

Barron Fuentes called to order the regular meeting of the Systems Analysis and Design Project at 11:16AM on October 4, 2016 at the Davidson Management Honors Program Lounge.

II. Roll call

Barron Fuentes conducted a roll call. The following persons were present: Barron Fuentes, Sejal Mali, Thuy-Mi Le

III. Approval of minutes from last meeting

Barron Fuentes read the minutes from the last meeting. The minutes were approved as read.

IV. Open issues

a) Write Phase 2 System Proposal

Divided the Proposal into the following parts: Executive summary, Project name, Background and justification, Functionality, Project scope, Expected value, Project schedule, and implementation.

Discussed what is included in each of the sections. Went over the example provided. Split up the sections between each member

V. New business

a) Create Google doc.

b) Start individual drafts.

Barron: Schedule, Project name

Sejal: Executive summary, implementation

Aman: Functionality, Expected value

Thuy-Mi: Background and justification, Project scope

c) Meet back one week from today.

VI. Adjournment

Barron Fuentes adjourned the meeting at 11:50AM.
Minutes submitted by: Thuy-Mi
Minutes approved by: Group

October 13, 2016

I. Call to order

Barron Fuentes called to order the regular meeting of the Systems Analysis and Design Project at 10:15AM on October 4, 2016 at the Davidson Management Honors Program Lounge.

II. Roll call

Barron Fuentes conducted a roll call. The following persons were present: Barron Fuentes, Sejal Mali, Aman Vakharia, Thuy-Mi Le

III. Approval of minutes from last meeting

Barron Fuentes read the minutes from the last meeting. The minutes were approved as read.

IV. Open issues

a) We need to put together the entire proposal.

Spent the entire meeting putting it together and making the proposal cohesive.

V. New business

b) No new business until next assignment is announced.

c) Meeting times scheduled for Tuesday or Thursday mornings the week before assignments are due.

VI. Adjournment

Barron Fuentes adjourned the meeting at 11:30AM.
Minutes submitted by: Thuy-Mi
Minutes approved by: Group

November 1, 2016

I. Call to order

Barron Fuentes called to order the regular meeting of the Systems Analysis and Design Project at 1:00 PM on November 1, 2016 at the Undergrad Lounge in JSOM.

II. Roll call

Barron Fuentes conducted a roll call. The following persons were present: Barron Fuentes, Thuy-Mi Le

III. Approval of minutes from last meeting

Barron Fuentes read the minutes from the last meeting. The minutes were approved as read.

IV. Open issues

a) Assign parts for Milestone 3

Divided the Proposal into the following parts: Requirements section (10 pts), use case diagram (15 pts), class diagram (10 pts), sequence diagrams (15 pts), and meeting minutes.

Discussed what is included in each of the sections. Went over the example provided. Split up the sections between each member

V. New business

b) Create Google doc.

c) Start individual drafts.

Requirements section: Sejal

Use-case diagram and descriptions: Aman

Class diagram: Barron

Sequence diagram: Thuy-Mi

Meeting minutes: Thuy-Mi

d) Meet back this Thursday, November 3.

VI. Adjournment

Barron Fuentes adjourned the meeting at 2:00 PM.
Minutes submitted by: Thuy-Mi
Minutes approved by: Group

November 3, 2016

I. Call to order

Barron Fuentes called to order the regular meeting of the Systems Analysis and Design Project at 11:00 AM on November 3, 2016 at the Undergrad Lounge in JSOM.

II. Roll call

Barron Fuentes conducted a roll call. The following persons were present: Aman Vakharia, Thuy-Mi Le

III. Approval of minutes from last meeting

Barron Fuentes read the minutes from the last meeting. The minutes were approved as read.

IV. Open issues

a) Compile all individual parts for final assignment

V. New business

b) Schedule meeting time sometime during the next couple weeks for the final assignment

VI. Adjournment

Barron Fuentes adjourned the meeting at 11:30AM
Minutes submitted by: Thuy-Mi
Minutes approved by: Group

November 17, 2016

I. Call to order

Barron Fuentes called to order the regular meeting of the Systems Analysis and Design Project at 11:00 AM on November 17, 2016 at the Undergrad Lounge in JSOM.

II. Roll call

Barron Fuentes conducted a roll call. The following persons were present: Aman Vakharia, Sejal Mali, Barron Fuentes, Thuy-Mi Le

III. Approval of minutes from last meeting

Barron Fuentes read the minutes from the last meeting. The minutes were approved as read.

IV. Open issues

a) Discussed all the topics for Milestone 4

b) Divided the work for each person to complete on his/her own

c) User Interface Design and Meeting Minutes – Thuy-Mi

d) Software Design – Barron

e) Testing – Aman

f) Combine final report and presentation - Sejal

V. New business

g) Meet on November 29 to make final tweaks and edits.

VI. Adjournment

Barron Fuentes adjourned the meeting at 11:45AM
Minutes submitted by: Thuy-Mi
Minutes approved by: Group

November 29, 2016

VII. Call to order

Barron Fuentes called to order the regular meeting of the Systems Analysis and Design Project at 1:00 PM on November 29, 2016 at the Undergrad Lounge in JSOM.

VIII. Roll call

Barron Fuentes conducted a roll call. The following persons were present: Aman Vakharia, Sejal Mali, Barron Fuentes, Thuy-Mi Le

IX. Approval of minutes from last meeting

Barron Fuentes read the minutes from the last meeting. The minutes were approved as read.

X. Open issues

h) Last minute tweaks in the final combination of the project

XI. New business

i) Study for the exam!!!!!!

XII. Adjournment

Barron Fuentes adjourned the meeting at 1:45PM
Minutes submitted by: Thuy-Mi
Minutes approved by: Group

Lessons Learned

Barron Fuentes

Using the software development lifecycle to create a replacement system for handling the UT Dallas CometCab system was an effective framework. During the process, primarily in the detailed software design, I found that the initial designs we created did not match up entirely with the final versions. The longer I spent thinking about design of the system, the more changes that I wanted to make to the initial designs. An example of this is added features, database storage optimization, and required attributes for certain objects. Something I would do differently next time around would be to use the throwaway prototyping methodology for each step in the design process. During this project I went forward with the first version of designs that later needed to be updated. If I had made a few versions of the designs before moving to the next steps, then I would've identified more errors or improvements.

During this project I learned of my group member's strengths in different areas of the development lifecycle. Some members were stronger in idea generation, aesthetic design, and business processes. Other members were strong in the detailed technical design and functionality. Success was dependent on coordination between both of these skills. This is the most important lesson that I learned from this project was the group dynamic of working with multiple individuals who all had different strengths and weaknesses. It is important to learn how to work in a group dynamic like this because in the professional world, almost all of the projects I will be working on will involve working in groups similar to this one.

Thuy-Mi Le

I learned a lot from this project, it was a lot of fun for me to be working on such a big design throughout the entire semester. I found great value in the project management aspect of

working on this project with a small team. I learned a lot about how to work effectively with team members and set deadlines and whatnot. Additionally, the project really helped me apply what we were learning in the class and really see all the aspects of a system project come together, which was all really cool. My favorite part of all the parts of the milestones was definitely creating all the use cases and diagrams. My personal favorite was the user experience diagram that I was able to put together for this final milestone. I felt like it really brought together all aspects of the rest of the diagrams and gave kind of a physical face to what we've been working hard on all semester.

I definitely see myself using what I learned in this class in the future. I've already been able to talk about the SDLC in several interviews I've been a part of and even implement it in group projects for other classes. Additionally, this final project is something that I've become really proud of and can use as an example of planning, teamwork, and a deliverable both in interviews and general discussions about my education. I'm very thankful for the rich experience that this class had, both in lecture, in-class activities, and the hands-on experience of the project. I look forward to applying everything I've learned about teamwork and communication (both verbally with the team, and written for proposals) to the rest of my time at UT Dallas and in my future career. Thank you for the wonderful class and project!

Sejal Mali

Aman Vakharia