

UTD Ride Share Service



MIS 6308 – Summer 2015 Project – Group 11

Arundhati Chadaga

Preeta Borkar

Rupashree Bhattacharya

Sejal Shah

Shivroopa Raghuwanshi

Table of Contents

Executive Summary.....	4
Problem Statement.....	5
The Business Need	5
Functionality / Objectives	5
Project Assumptions	6
Context Diagram	7
Use Case Diagram	8
Use Case Description	9
Driver Registration Use Case.....	9
Passenger Registration Use Case	9
Process Login Use Case	10
Place Ride Use Case	10
Driver and Passenger Match Use Case	11
Booking Use Case	11
Cancel Ride Use Case	12
Process Payment Use Case	13
Data Dictionary	14
Registration Use Case	14
Login Use Case	14
Place Ride Use Case	14
Driver & Passenger Match	14
Booking Use case	14
Cancel ride Use Case	15
Payment Use Case.....	15
Class Diagram - Without Methods.....	16
Class Diagram - With Methods	17
Sequence Diagrams.....	18
Passenger Registration Use Case	18
Driver Registration Use Case.....	18
Login Use Case	19
Trip Details Use Case.....	19

Driver & Passenger Match Use Case.....	19
Booking Use Case.....	20
Cancel Ride - Driver Use Case	20
Cancel Ride - Passenger Use Case.....	20
Payment Use Case.....	21
Functional Specification Document for the Proposed System	22
Entity Relationship Diagram	23
Database Constraints.....	24
a) Student Table.....	24
b) Driver Table.....	24
c) Passenger Table	24
d) Account Table	24
e) Rides_List Table.....	25
f) Booking Table.....	25
g) Invoice Table	25
h) Payment Table	26
Interface Designs	27
Home Screen.....	27
Registration Page of Driver	28
Registration Page of Passenger.....	29
Login Page	30
Driver Trip details page.....	31
Passenger Trip details page	32
Driver Profile page	33
Passenger Profile page.....	34
Screen for placing rides.....	35
Passenger Bill	36
Software Design: Method Contracts and Pseudocode	37
Control Mechanisms	41
Conclusion.....	43
Project Management Deliverables	44

Executive Summary

Are you new to Dallas or USA, feeling perplexed on how to get around the city? Need a ride to your school or the neighboring Walmart but too lazy to wait for the next available Comet cruiser? Want to avoid an extra expenditure on a student car this semester? Well, a UTD-Ride Share service will definitely brighten your day. This ride-share service will be owned by UTD and operated by UTD students/ staff hence uniting us with a sense of belongingness to UTD. The service will help the passengers ride with your fellow comet drivers, hence assuring security. The service will make it more convenient for student to get around the city, earn some extra bucks to support their student expenses, be more eco-friendly and save the worry of not being able to park in the overflowing parking lots.

Real-time ridesharing is a win-win approach to match drivers and riders for carpooling. We have seen a splurge in the number of ridesharing apps in past few years, who try to use different concepts for mapping and target different audience to create their niche. The big names in the industry include Uber, Lyft and sidecar whereas we see many more like Car2go who promise to be eco-friendly, Zimride- owned by bigger rental companies like Enterprise and some companies like Carma set them apart by targeting their audience like working professional.

UTD-Ride share is a university driven carpool service where the drivers publish their itinerary on the application and riders can choose to ride with the drivers who fall in their route. The additional feature of allowing 3 riders to share a ride brings down the cost of travel per passenger, hence catering to our student audience. The unique selling point for our service is its affiliation with the university, low cost of travel due to multiple passengers and critical background check of the drivers ensuring safety.

Currently student rely on Facebook pages, favors and comparatively expensive means like Uber/ Lyft for their transportation needs that cannot be taken care of by Comet Cruisers. As a part of our project we have presented our deep analysis of the proposed UTD-Ride share through context diagram, use case diagram, class diagram and sequence diagram that gives the broad perspective. The intricate details like database design, control mechanism, software design and user interface design give a holistic overview of the feasibility and implementation. Systematic project planning and implementation schedule makes the proposed vision achievable.

Problem Statement

The Business Need

- There is currently no dedicated platform within UTD for students to pool cars or share rides with others on campus.
- Students who want to pool cabs with others have no means to find out if someone else is travelling to the same location. They are thus, limited to asking friends or posting on Facebook pages for ride requests.
- For people who do find rides via Facebook or other networks, there is no way to ensure the Driver's credibility and validate his identity and license.
- For rides that can be sought via Facebook or word-of-mouth, it is currently not possible to reserve a seat or learn about any last minute cancellations.
- The lack of any common platform to post rides also prevents any prospective drivers who would want to share a ride from informing their peers.
- Students relying on ZipCar or Uber for personal use leads to increased traffic around campus as well as high pollution levels.
- There is also a dearth of available on campus job opportunities.

Functionality / Objectives

- In this project, we plan to create a UTD based mobile application dedicated to facilitating ridesharing/ carpooling amongst UTD students.
- This project proposes a mobile application which will let students search for all rides voluntarily posted by drivers on the designated day and time.
- This application will be officially supported by UTD. To be able to use this ride as either the driver or passenger, one should be a UTD student. He/ She will have to first register on this application. To register as a driver, one will be required to produce a valid driver's license, proof of vehicle insurance and an owned vehicle.
- Once a passenger selects a ride, a request will be sent to the driver for approval. Once the driver approves, the passenger's ride will be secured.
- The passenger will pay the billed amount to the driver at the end of the journey using a credit card. The amount will be billed by UTD rideshare which will in turn pay the drivers a cumulative sum at the end of a stipulated period.
- This ride is also intended to give an opportunity to everyday commuters to reduce their fuel costs by sharing the ride with a colleague.
- Carpooling is expected to help reduce traffic congestion in and around campus and is an environmentally responsible initiative.

- This application is also expected to provide job opportunities to students at times of their convenience.

Project Assumptions

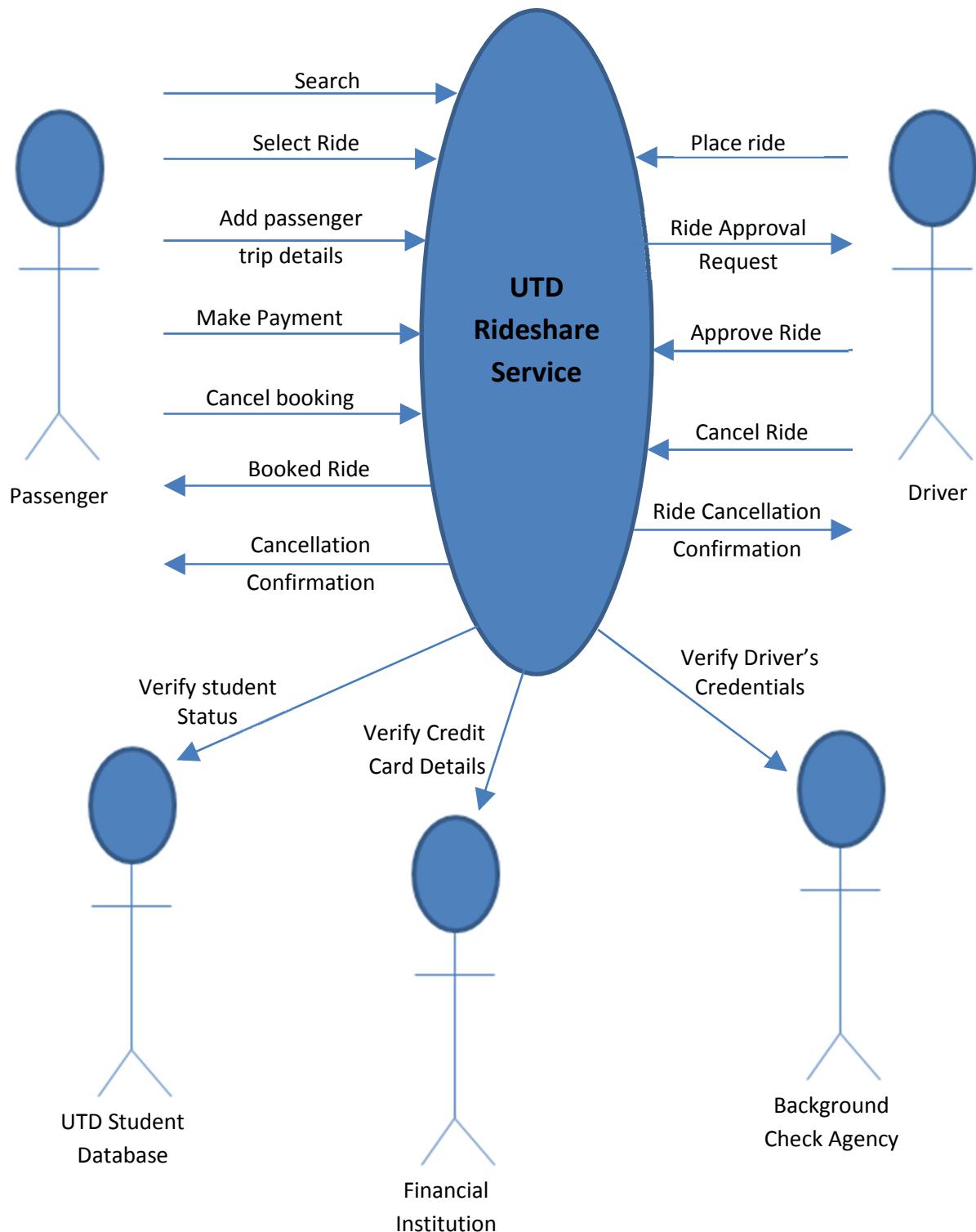
In this project, we primarily focus on how rides can be shared by a driver and a prospective passenger. Drivers place rides to particular locations along with the date and time of departure. These rides are visible to passengers. Once a passenger requests a ride, a driver has the option to approve or deny it. The rides are billed by UTD. Here, we mainly concentrate on the process of requesting a ride and booking it from the passenger's point of view. To focus on the process, we have introduced the constraint that only the driver can post a ride. The passenger can only select rides from the list available to him/her.

We have assumed here, that the driver cannot be a passenger or vice versa. Also, since passenger needs to authenticate himself/herself as a UTD student, he/she can only book one seat per ride.

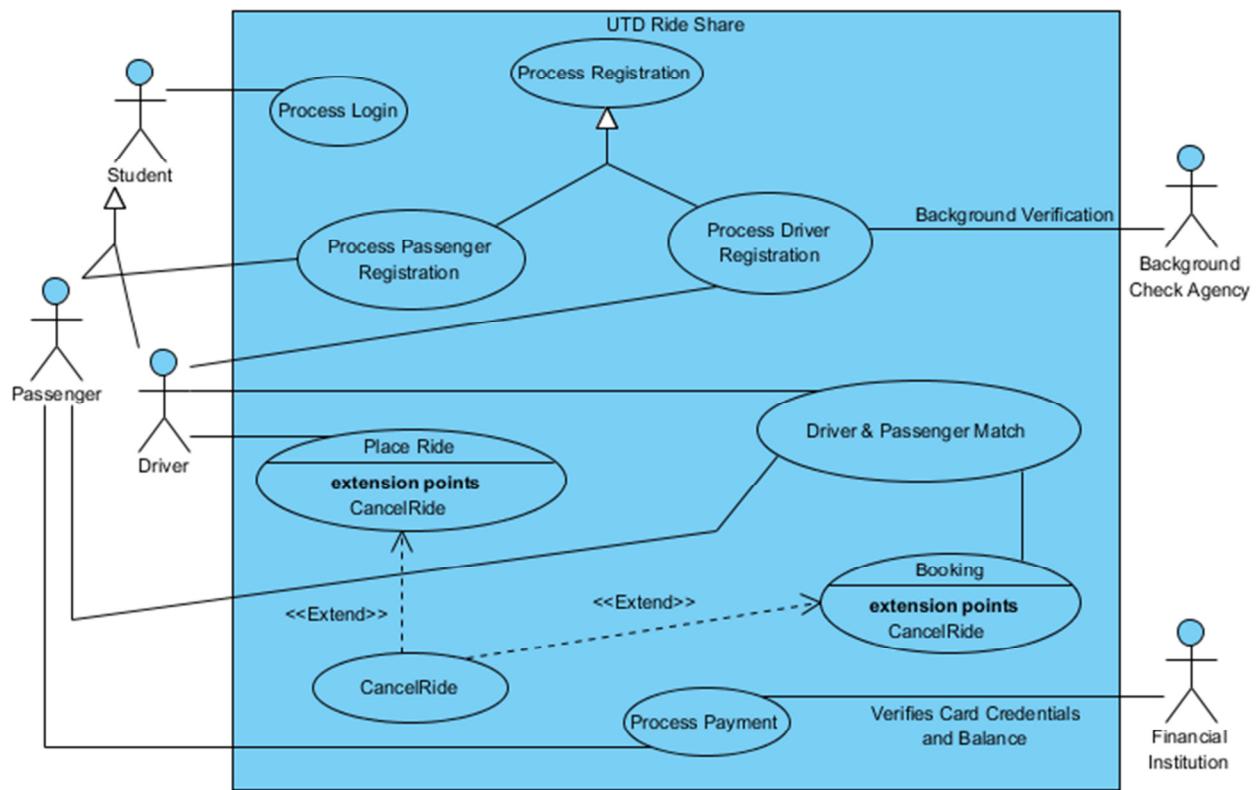
For simplification purposes, we have not included GPS location determination but let the passenger choose the rides from the available list. Also, we assume that the payment is determined by the School authorities and so we have not incorporated mechanisms to calculate the billed amount as per miles.

For Driving License, Insurance and Vehicle verification we depend on an external Background Verification Company. A passenger must book a ride at least 1 hour before the journey.

Context Diagram



Use Case Diagram



Use Case Description

Driver Registration Use Case

Use Case Name	Driver Registration
Primary Actor	Driver
Trigger	Click on the Register button and Select Register as Driver option.
Normal flow of events	<ol style="list-style-type: none">1. For each person who clicks on Register as Driver Button DO2. Display the UTD Rideshare Driver <u>Registration Page</u> for the driver to fill in.2. The driver enters the registration details.3. The UT Database will verify if this is an existing student using <u>NetID, Password</u>4. The Background Check Agency will authenticate the driving credentials.5. Notify the driver about the <u>login details</u>.6. The system displays the Login Page to the driver.6. End Do
Exceptional Flow	<ol style="list-style-type: none">3.1 IF the UT Database cannot verify the student THEN display Student Verification Error. TERMINATE.4.1 IF the driver fails the Background Check, THEN display authentication error. TERMINATE

Passenger Registration Use Case

Use Case Name	Passenger Registration
Primary Actor	Passenger
Trigger	Click on the Register button and Select Register as Passenger option.
Normal flow of events	<ol style="list-style-type: none">1. For each person who clicks on Register as Passenger Button DO2. Display the UTD Rideshare Passenger <u>Registration Page</u> for the passenger to fill in.2. The passenger enters the <u>registration details</u>.3. The UT Database will verify if this is an existing student using

	<p><u>NetID, Password</u></p> <p>5. Notify the passenger about the <u>login details</u>.</p> <p>6. The system displays the Login Page to the passenger.</p> <p>6. End Do</p>
Exceptional Flow	<p>3.1 IF the UT Database cannot verify the passenger as student THEN DISPLAY Student Verification Error. TERMINATE.</p> <p>4.1 If the driver fails the Background Check, THEN DISPLAY Authentication Error. TERMINATE.</p>

Process Login Use Case

Use Case Name	Login
Primary Actor	Driver/Passenger
Trigger	Click on the Login button
Normal flow of events	<p>1. For each person who clicks on LOGIN Button DO</p> <p>2. Display the UTD Rideshare Passenger <u>Login Page</u></p> <p>2. The passenger enters the <u>NetID</u> and <u>Password</u></p> <p>3. Validate the login details.</p> <p>4. Display the Account Page to the User.</p> <p>6. End Do</p>
Exceptional Flow	3.1 IF NETID and Password do not match; Display Error. Return to Home Page.

Place Ride Use Case

Use Case Name	Place Ride
Primary Actor	Driver
Trigger	On clicking Place Ride Button on Driver's home page
Normal flow of events	<p>1. On click of Place Ride Button Do</p> <p>2. Open Place Ride page for Driver to enter <u>Trip</u> details</p> <p>3. Validate all mandatory fields required to Place Ride</p> <p>5. Set <u>Ride Status</u> as 'In Process'</p> <p>6. Notify driver about his added ride</p> <p>7. Display home page</p> <p>8. End Do</p>
Exceptional Flow	7. IF driver selects cancel ride button on My Trip page THEN

	Allow driver to cancel ride and Execute Cancel Ride Use case
--	--

Driver and Passenger Match Use Case

Use Case Name	Driver & Passenger Match
Primary Actor	Passenger
Trigger	The Passenger enters a Search Request
Normal flow of events	<ol style="list-style-type: none"> 1. Retrieve <u>Trip</u> details based on <u>Search Request</u> entered by Passenger 2. Display <u>Search Result</u> 3. Add <u>Selected Trip</u> to <u>Shopping Cart</u> 4. Send <u>Trip Confirmation by Driver</u> message to Driver with an option to accept or deny the ride request 5. After the Driver accepts the ride request, execute Booking use case
Exceptional Flow:	<ol style="list-style-type: none"> 1. IF no rides match the <u>EnteredDate</u> and <u>EnteredTime</u> in the <u>Search Request</u> THEN Display “No Rides Available Currently” 4. IF Driver denies the ride request THEN Delete <u>Selected Trip</u> from <u>Shopping Cart</u>

Booking Use Case

Use Case Name	Booking
Primary Actor	UTD Ride Share system
Trigger	On driver's acceptance for any ride from shopping cart
Normal flow of events	<ol style="list-style-type: none"> 1. For each ride accepted by driver Do 2. Show <u>Terms of Agreement</u> to passenger to <u>Accept/ Deny</u> 3. Assign a unique <u>Itinerary Number</u> 4. Set <u>Booking_Status</u> to ‘Booked’

	<p>5. Update <u>NoOfAvailableSeats</u> of the booked ride to <u>NoOfAvailableSeats -1</u></p> <p>6. Display list of accepted rides on <u>Booking Page</u> with assigned <u>Itinerary Number</u></p> <p>7. Notify the passenger about <u>Billing information</u> with ride details and the amount to be paid as <u>TotalFare</u> and <u>Tax</u></p> <p>8. The system displays a screen for <u>Payment</u></p> <p>9. End Do</p>
Exceptional Flow	<p>5. IF driver/ passenger select cancel ride button on my trip page THEN</p> <p>Allow them to cancel ride and Execute Cancel Ride Use case</p>

Cancel Ride Use Case

Use Case Name	Cancel Ride
Primary Actor	Driver / Passenger
Trigger	On click of Cancel Ride Button on Driver/ Passenger respective home page
Normal flow of events	<p>1. If passenger cancels ride</p> <p>1.a. notify driver of respective ride about the cancellation.</p> <p>1.b. update <u>NoOfAvailableSeats</u> to <u>NoOfAvailableSeat +1</u></p> <p>1.c. Set <u>Booking Status</u> as 'Cancelled'</p> <p>2. If driver cancels ride</p> <p>2.a. notify booked passenger of respective ride about cancellation</p> <p>2.b. Set <u>Booking Status</u> for all respective bookings as 'Cancelled'</p> <p>2.c. Set <u>Ride Status</u> as 'Cancelled'</p>
Exceptional Flow	

Process Payment Use Case

Use Case Name	Payment
Primary Actor	Passenger
Trigger	Swipe of credit/Debit card
Normal flow of events	<ol style="list-style-type: none">1. Passenger swipes the credit/debit card2. The Financial Institution verifies the card details3. Assign a unique <u>TransactionNumber</u> to the payment process4. Set <u>Payment Status</u> as 'Complete'5. Set <u>Booking Status</u> as 'Completed'
Exceptional Flow	<ol style="list-style-type: none">2. If card details are invalid or passenger account does not have sufficient funds debar the passenger.

Data Dictionary

Registration Use Case

Registration Page = NetID + FirstName + LastName + [Registering as]

Registering as = [Driver / Passenger]

Driver Registration page = DriverLicenseNumber + InsuranceNumber + BankAccountNumber

Login Use Case

Login Credentials = Net ID + Password

Place Ride Use Case

Trip_Info= NetID + 0{Trip}

Trip = Ride_No + Ride_Date + DepartureTime + NoOfAvailableSeats + DriverSource + DriverDestination + Ride_Status

Driver & Passenger Match

Search Request = EnteredDate + EnteredTime

Search Result = 0{Trip}

Trip = Select Option + Ride_Date + DepartureTime + DriverSource + DriverDestination + NoOfAvailableSeats

Select Option = [Yes | No]

Passenger Trip Details = PassengerSource + PassengerDestination

Shopping Cart = 0{Selected Trip}

Selected Trip = Ride_No + Ride_Date + DepartureTime + PassengerSource + PassengerDestination

Trip Confirmation by Driver = RideConfirmation + Ride_Date + DepartureTime + PassengerSource + Passenger Destination

RideConfirmation = [Yes|No]

Booking Use case

Booking page = 0{Itinerary} + Terms of Agreement

Itinerary = Itinerary_Number + Ride_Date + DepartureTime + PassengerSource + PassengerDestination + Booking_Status + Ride_No

Terms of Agreement = [Accept|Deny]

Billing information = InvoiceNumber + InvoiceDate + Ride_Date + DepartureTime + PassengerSource + PassengerDestination + Ride_No + Itinerary_Number + TotalFare + Tax

Cancel ride Use Case

Driver_Cancel= Ride_No + Select option + Ride_Date+ DepartureTime + DriverSource + DriverDestination

Select option = [yes|no]

Passenger_Cancel= Itinerary_Number + Select option + Ride_Date+ DepartureTime + PassenegerSource + PassenegerDestination

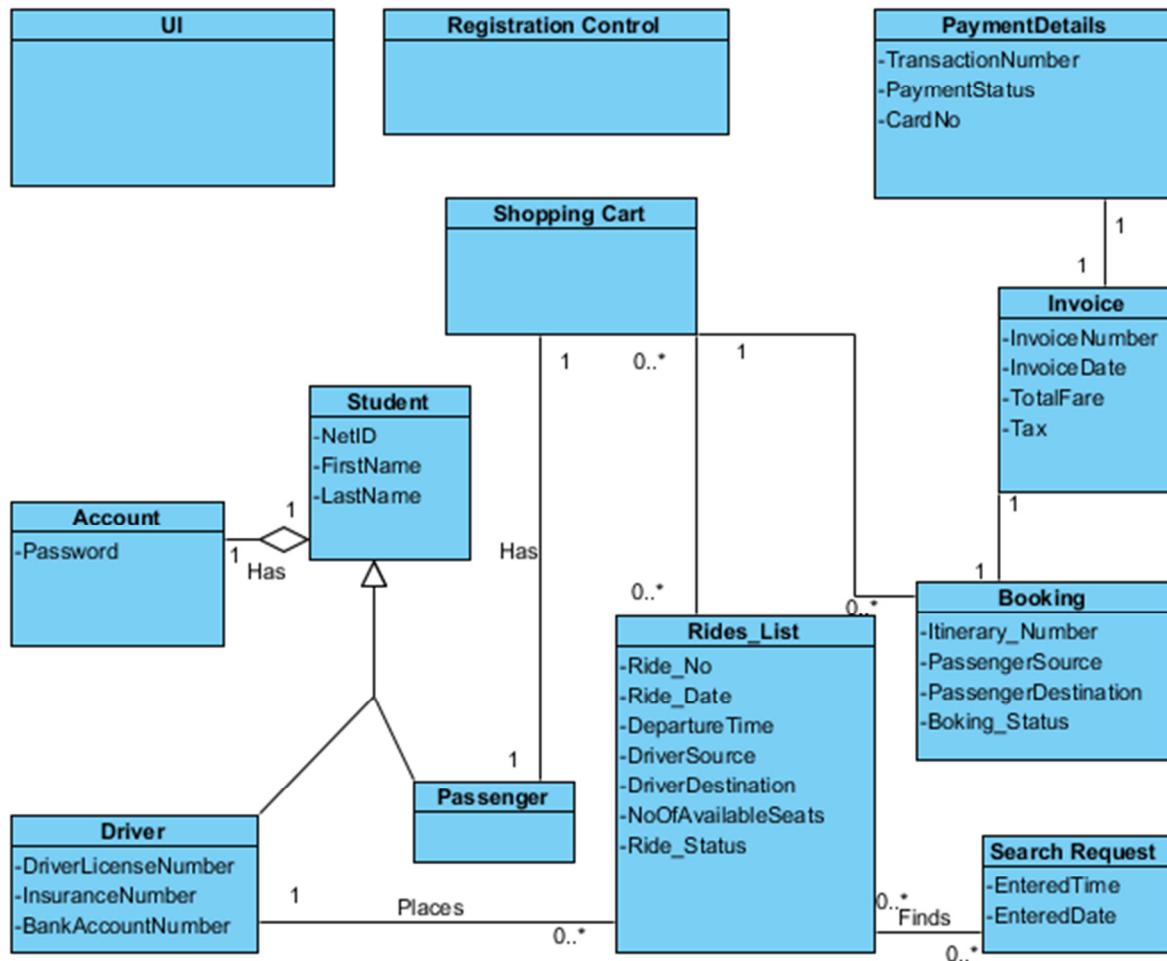
Select option = [yes|no]

Payment Use Case

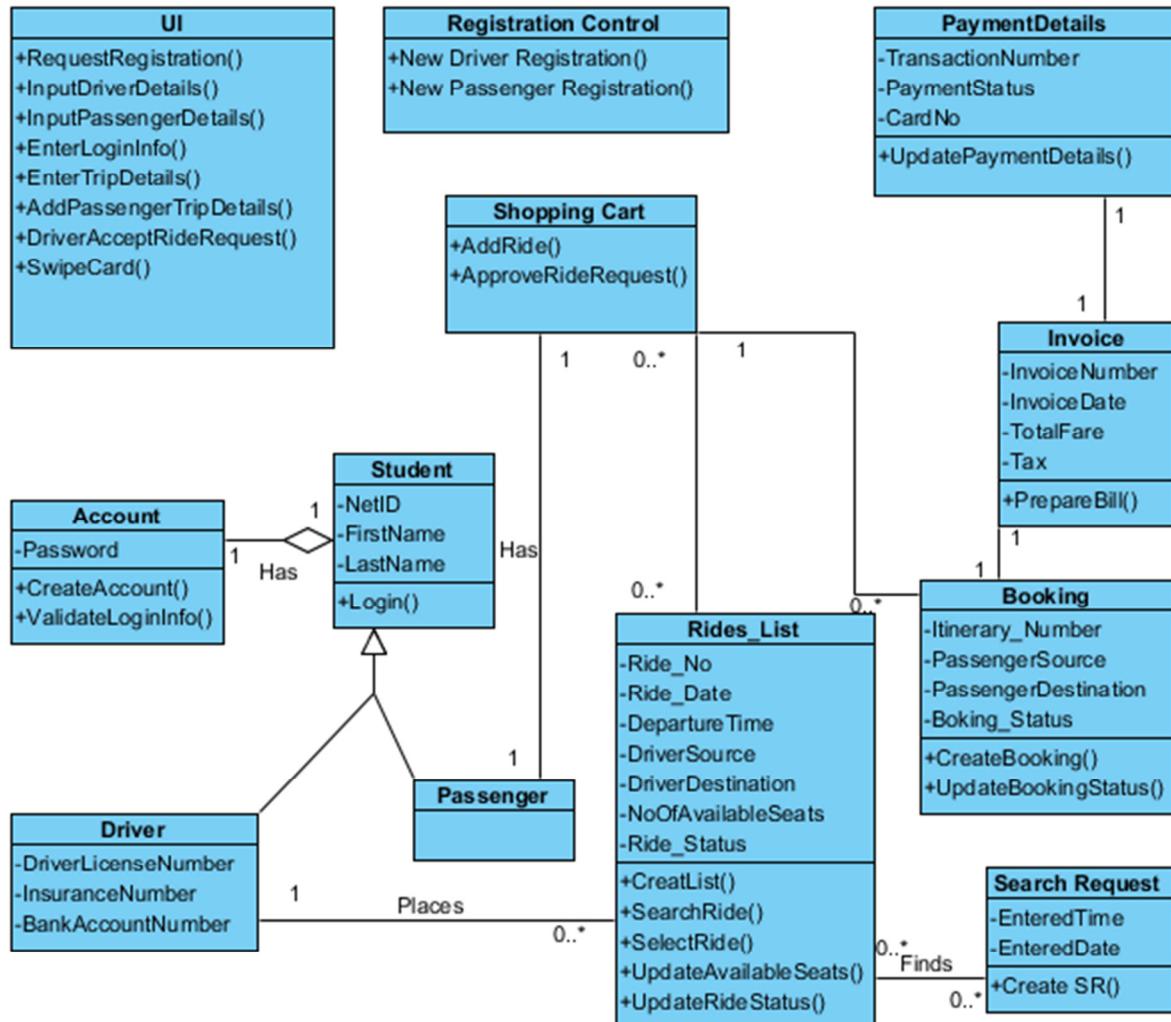
Payment= TransactionNumber + Itinerary_Number + NetID+ CardNo + Agree_amounts + PaymentStatus

Agree_Amount = [Yes|No]

Class Diagram - Without Methods

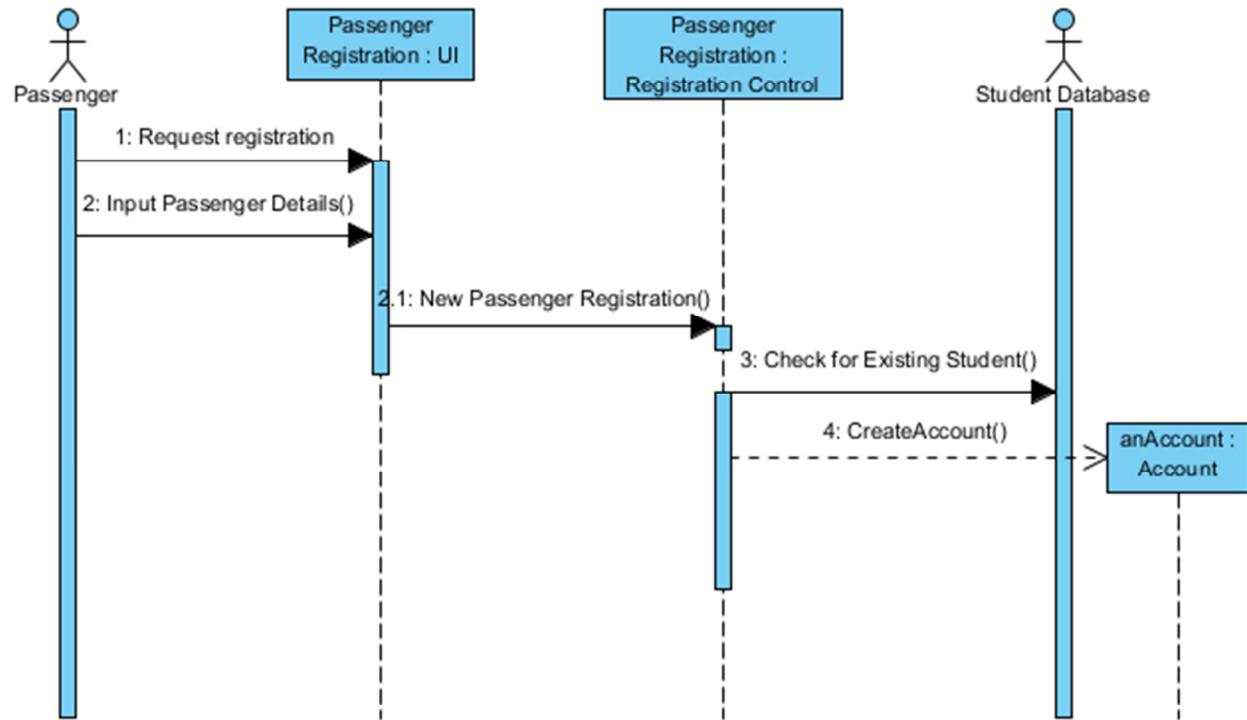


Class Diagram - With Methods

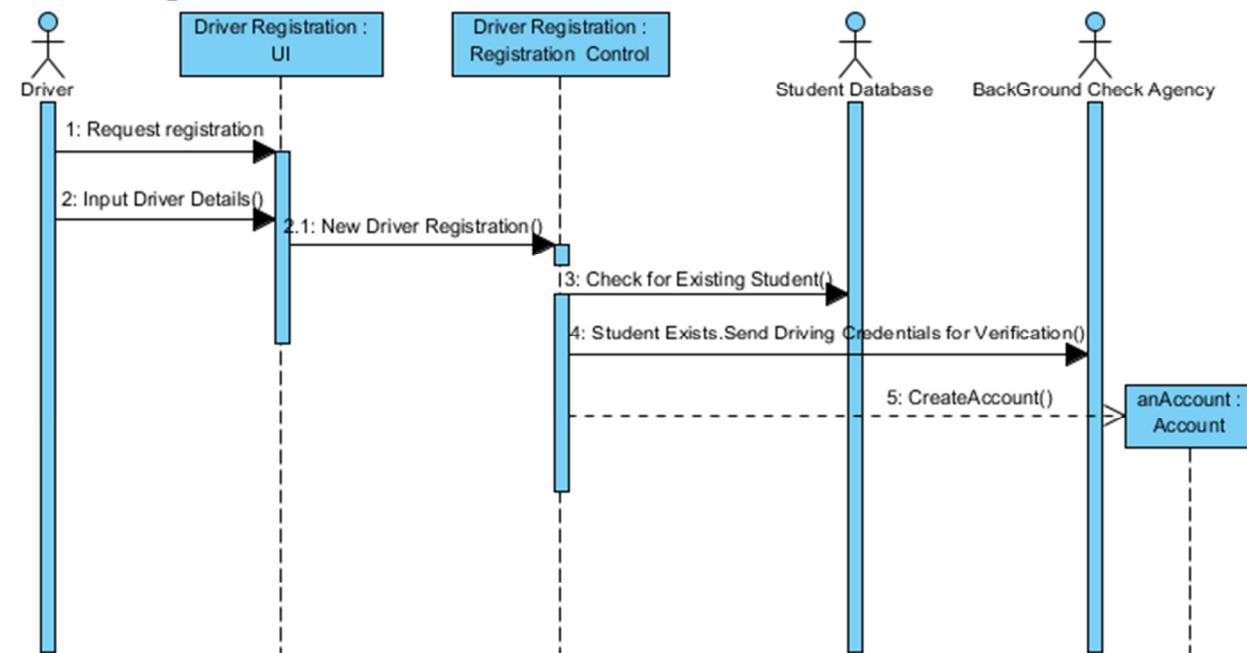


Sequence Diagrams

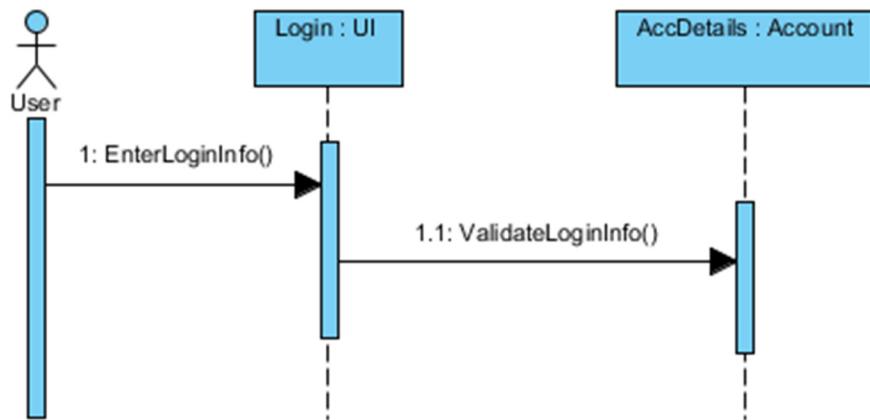
Passenger Registration Use Case



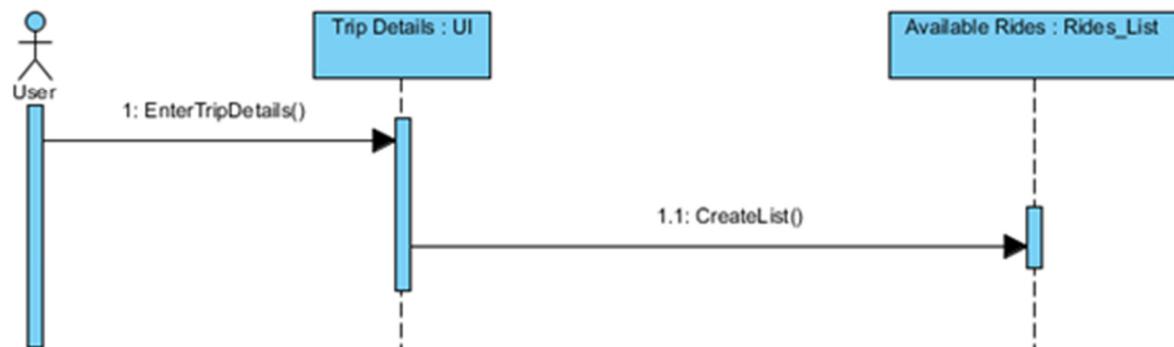
Driver Registration Use Case



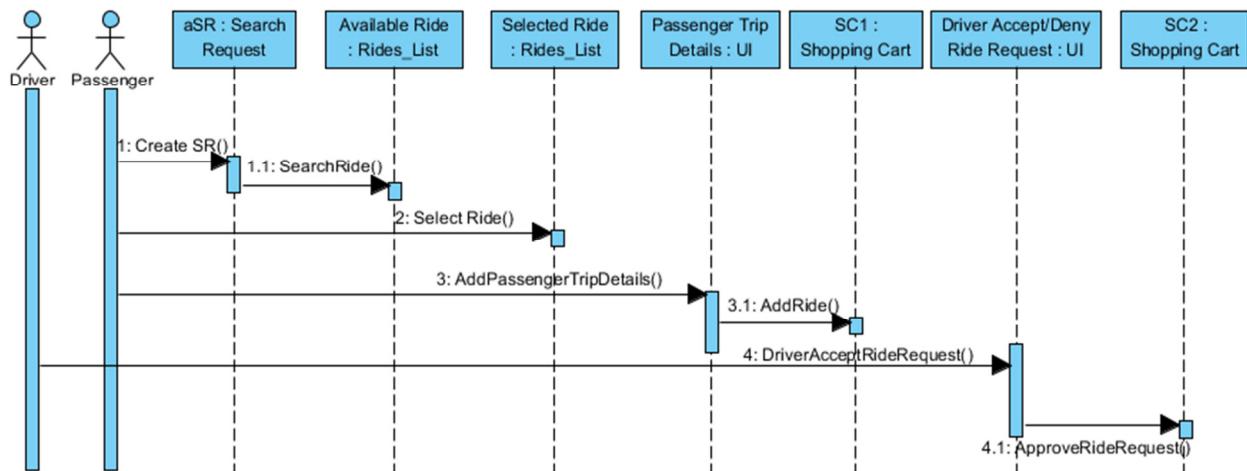
Login Use Case



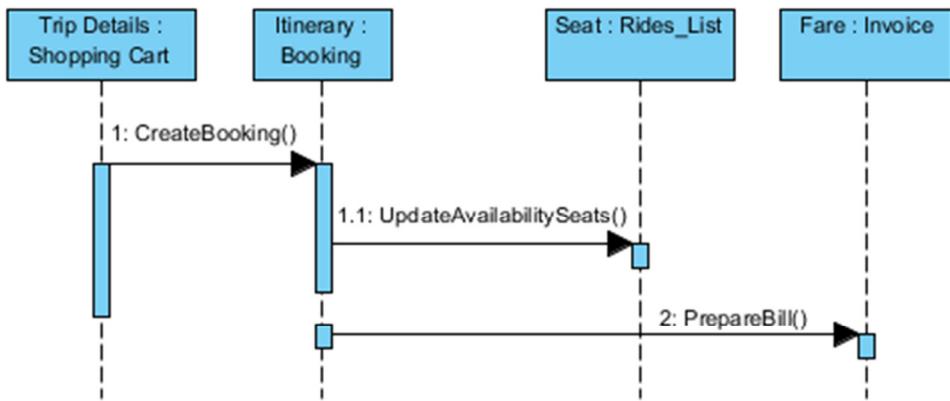
Trip Details Use Case



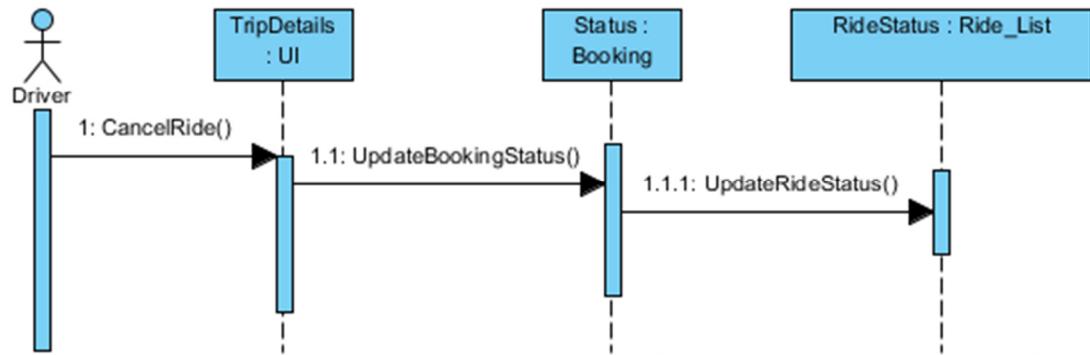
Driver & Passenger Match Use Case



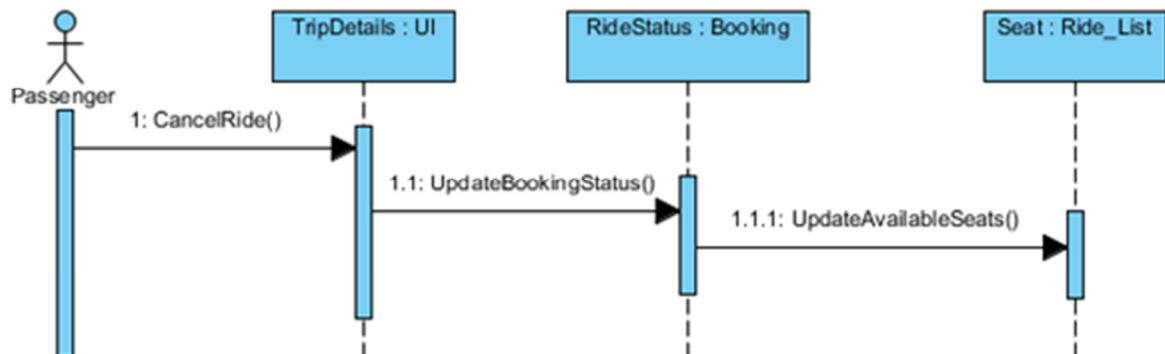
Booking Use Case



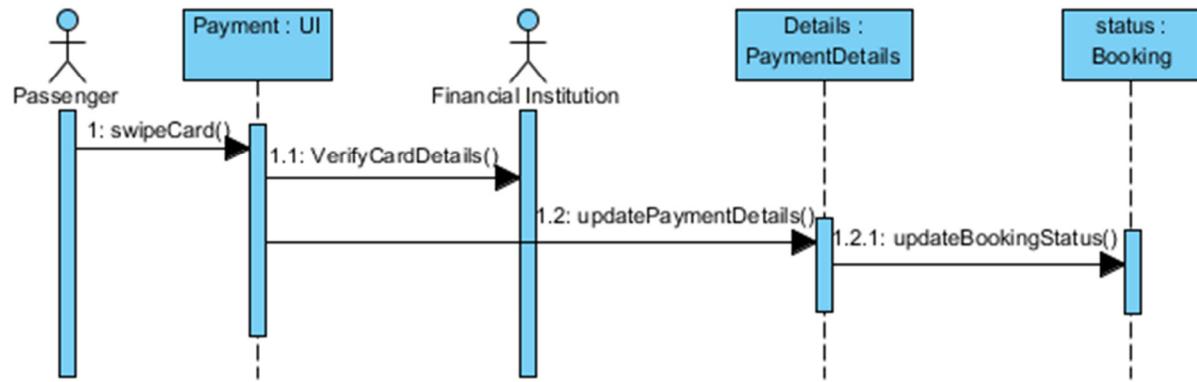
Cancel Ride - Driver Use Case



Cancel Ride - Passenger Use Case



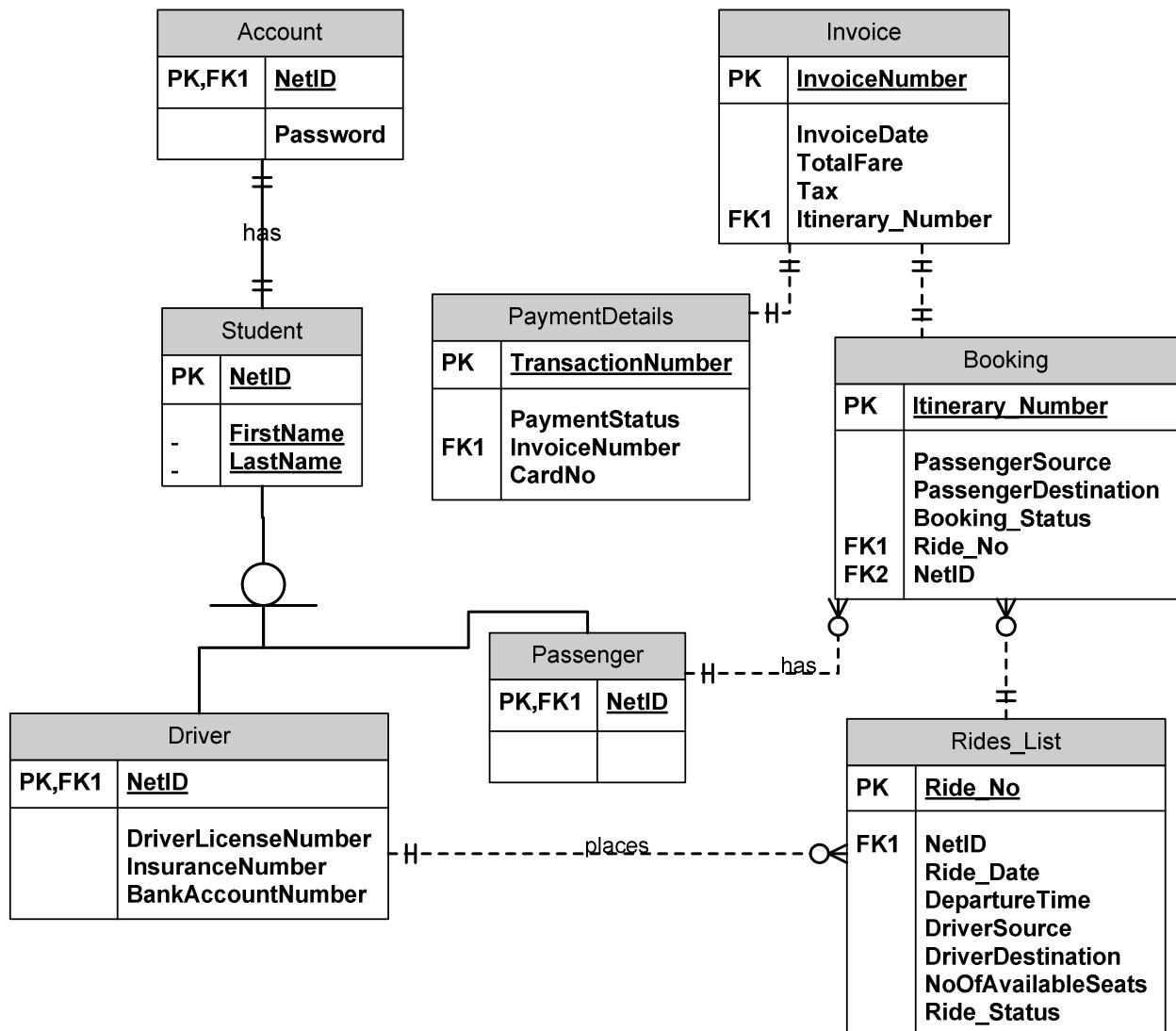
Payment Use Case



Functional Specification Document for the Proposed System

1. The proposed system is a UTD based mobile application dedicated to facilitating ridesharing/carpooling amongst UTD students.
2. This application will be officially supported by UTD. To be able to use this ride as either the driver or passenger, one should be a UTD student. He/ She will have to first register on this application. Additionally, to register as a driver, one will be required to produce a valid driver's license, proof of vehicle insurance and an owned vehicle.
3. This system will allow registered drivers to voluntarily place rides by entering their source and destination points along with the departure time and number of available seats in the car.
4. Registered passengers can search rides posted by drivers based on day and time criteria.
5. Once a passenger selects a ride and enters his/her actual pick-up and drop-off points which would generally be on route of the selected ride, the System will add it to the shopping cart. The system will then send the ride to the driver for approval. If the driver approves, the system will book the ride for the passenger and generate a unique itinerary number. The Available seat Count for the particular ride will be reduced by 1.
6. The passenger will pay the billed amount to the driver at the end of the journey by swiping a credit/debit card. The system will outsource card verification to an external Financial Institution. The System will note if the Payment is successful or has failed.
7. If the Payment fails due to insufficient fund in the card, the passenger will be debarred from using the UTD Rideshare app again.
8. This system will also allow drivers or passengers to cancel rides. A driver can only cancel rides that he is driving and a passenger can only cancel rides that he has booked.
9. If a driver cancels any ride, all passengers booked for that ride will be notified and their bookings get cancelled. The ride status of the particular ride will be set to 'Cancelled'. If a passenger cancels a booked ride, his booking shall be cancelled, the driver shall be notified and the Available Seat Count for that ride will be increased by 1.

Entity Relationship Diagram



Database Constraints

a) Student Table

i) Primary Key Constraint /Integrity Constraint:-

Primary Key ‘NetID’ should not be NULL

ii) Uniqueness Constraint:-

Primary Key ‘NetID’ should be unique

b) Driver Table

i) Primary Key Constraint /Integrity Constraint:-

Primary Key ‘NetID’ should not be NULL

ii) Uniqueness Constraint:-

Primary Key ‘NetID’ should be unique

iii) Referential Integrity Constraint:-

All ‘NetID’ which is the Foreign Key in Driver table should exist as the Primary Key in Student table.

Every Driver should have a NetID

c) Passenger Table

i) Primary Key Constraint /Integrity Constraint:-

Primary Key ‘NetID’ should not be NULL

ii) Uniqueness Constraint:-

Primary Key ‘NetID’ should be unique

iii) Referential Integrity Constraint :-

All ‘NetID’ which is the Foreign Key in Passenger table should exist as the Primary Key in Student table.

Every Passenger should have a NetID

d) Account Table

i) Primary Key Constraint /Integrity Constraint:-

Primary Key ‘NetID’ should not be NULL

ii) Uniqueness Constraint:-

Primary Key ‘NetID’ should be unique

iii) Referential Integrity Constraint:-

All 'NetID' which is the Foreign Key in Account table should exist as the Primary Key in Student table.

Every Account should have a NetID

e) Rides_List Table

i) Primary Key Constraint /Integrity Constraint:-

Primary Key 'Ride_No' should not be NULL

ii) Uniqueness Constraint:-

Primary Key 'Ride_No' should be unique

iii) Referential Integrity Constraint:-

All 'NetID' which is the Foreign Key in Rides_List table should exist as the Primary Key in Driver table.

Every Ride in the Rides_List table should have the associated driver's NetID

f) Booking Table

i) Primary Key Constraint /Integrity Constraint:-

Primary Key 'Itinerary_Number' should not be NULL

ii) Uniqueness Constraint:-

Primary Key 'Itinerary_Number' should be unique

iii) Referential Integrity Constraint:-

All 'NetID' which is the Foreign Key in Booking table should exist as the Primary Key in Passenger table.

All 'Ride_No' which is the Foreign Key in Booking table should exist as the Primary Key in Rides_List table.

Every Booking in the Booking table should have the associated passenger's NetID

Every Booking in the Booking table should have the associated ride's Ride_No

g) Invoice Table

i) Primary Key Constraint /Integrity Constraint:-

Primary Key 'InvoiceNumber' should not be NULL

ii) Uniqueness Constraint:-

Primary Key 'InvoiceNumber' should be unique

iii) Referential Integrity Constraint:-

All 'Itinerary_Number' which is the Foreign Key in Invoice table should exist as the Primary Key in Booking table.

Every Invoice in the Invoice table should have the associated Booking's Itinerary number

h) Payment Table

i) Primary Key Constraint /Integrity Constraint:-

Primary Key 'TransactionNumber' should not be NULL

ii) Uniqueness Constraint:-

Primary Key 'TransactionNumber' should be unique

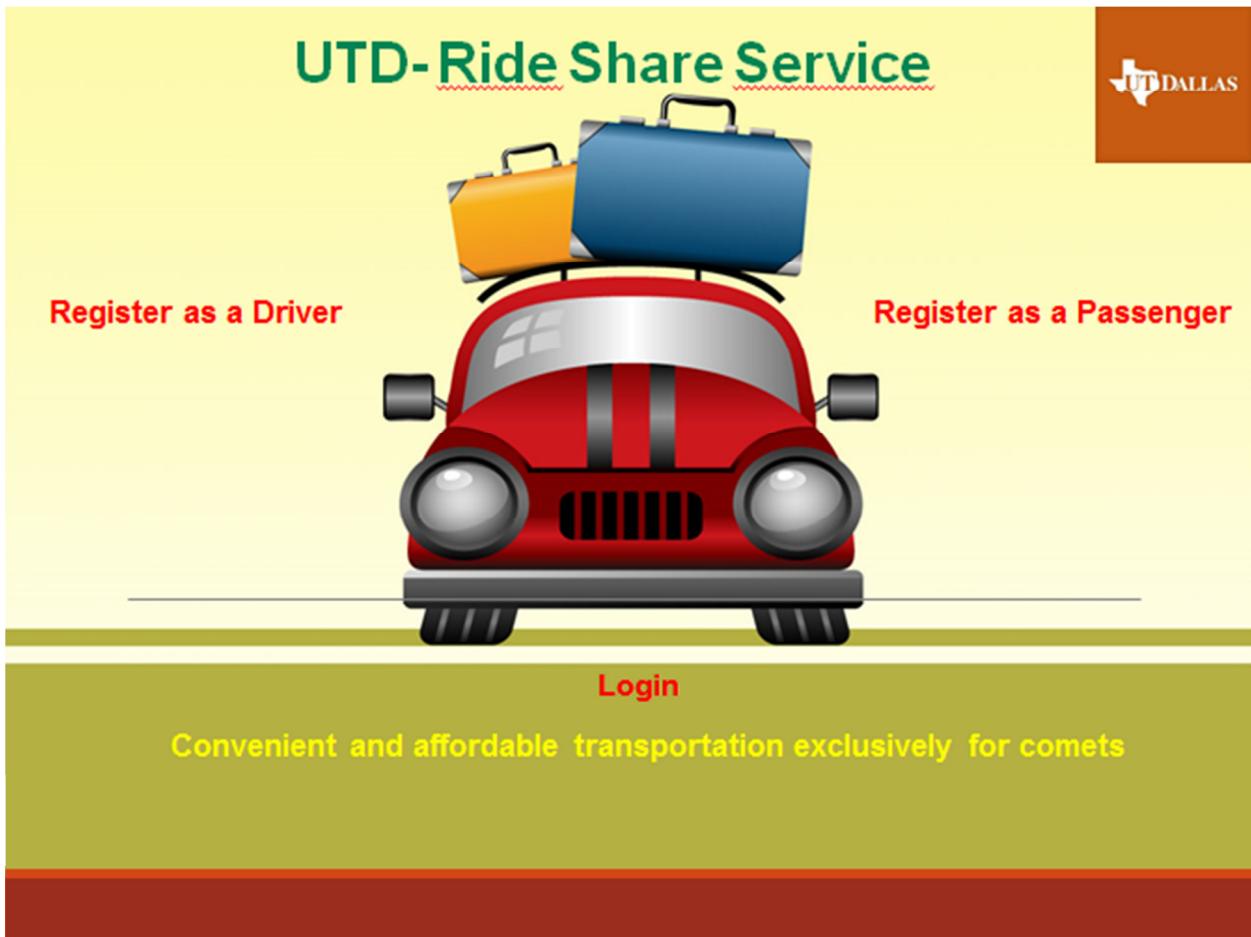
iii) Referential Integrity Constraint:-

All 'InvoiceNumber' which is the Foreign Key in Payment table should exist as the Primary Key in Invoice table.

Every Transaction in the Payment table should have the associated Invoice's Invoice Number

Interface Designs

Home Screen



Registration Page of Driver



UTD-Ride Share Driver Registration Page

First Name*	<input type="text"/>	Last Name*	<input type="text"/>
Net ID*	<input type="text"/>	Bank Account Number*	<input type="text"/>
Insurance Number*	<input type="text"/>	Driver's License Number*	<input type="text"/>
Password*	<input type="password"/>		
Confirm Password*	<input type="password"/>		
Car Insurance papers*	<input type="button" value="Attach Documents"/>		
Car Registration papers*	<input type="button" value="Attach Documents"/>		
Bank Account Check*	<input type="button" value="Attach Documents"/>		
<input type="checkbox"/> Accept terms and conditions			
<input type="button" value="Submit"/>			

Note: For UTD-drivers, verification and background check process can take 5-10 business days. Verification email will be sent to your UTD email address on creation of driver's account. Please login using the credentials to access your account after confirmation.

Registration Page of Passenger



UTD-Ride Share Passenger Registration Page

First Name*

Last Name*

Net ID*

Password*

Confirm Password*

Accept [terms and conditions](#)

Submit

Note: For UTD-riders, the verification process can take 3-5 business days. Verification email will be sent to UTD email address on creation of passenger's account. Please login using the credentials to access your account after confirmation.

Login Page

 UTD DALLAS

UTD-Ride Share Login Page

Net ID

Password

Log in

[Remember me](#)

[Forgot password?](#)

Driver Trip details page



UTD-Ride Share Driver -Trip details Page

[My Trips](#) [Pace Ride](#) [Profile](#) [Sign out](#)

Ride_No	Pickup Date	Pickup Time	Source	Destination	Fare \$	Ride_Status	
1234	6/12/2015	11:00 AM	JSOM, UTD	McCallum, Dallas	6.10	Completed	Cancel Ride
1235	6/13/2015	2:00 PM	McCallum, Dallas	Walmart, Coit Road	7.50	Completed	Cancel Ride

Passenger Trip details page



UTD-Ride Share Passenger – Trip details Page

[My Trips](#) [Pace Ride](#) [Profile](#) [Sign out](#)

Itinerary No	Pickup Date	Pickup Time	Pickup Address	Drop-off Address	Fare \$	Booking Status	
1105	6/12/2015	11:00 AM	JSOM, UTD	McCallum Blvd, Dallas	6.10	Completed	Cancel Ride
1589	6/13/2015	2:00 PM	McCallum Blvd, Dallas	Walmart, Coit Road	7.50	Completed	Cancel Ride

Driver Profile page


UTD-DALLAS

UTD-Ride Share Driver Profile Page

[My Trips](#) [Pace Ride](#) [Profile](#) [Sign out](#)

General Information
[\(Click to edit\)](#)


[Upload Profile Picture](#)

First Name	Luis	Last Name	Mathew
Net ID	lxm420854	Insurance Number	24585136
Bank Account Number	24578966321215	Driver's License Number	TX-245789

[Update](#)

Note: Any update will take 3-5 days to get updated, after verification process. Confirmation email will be sent to UTD email address once change is approved.

Passenger Profile page



UTD-Ride Share Passenger Profile Page

[My Trips](#) [Pace Ride](#) [Profile](#) [Sign out](#)

General Information
[\(Click to edit\)](#)

First Name	<input type="text" value="Jessica"/>
Last Name	<input type="text" value="Brad"/>
Net ID	<input type="text" value="jxb140250"/>

[Update](#)

Note: Any update will take 3-5 days to get updated, after verification process. Confirmation email will be sent to UTD email address once change is approved.

Screen for placing rides



UTD-Ride Share Place Ride Page

[My Trips](#) [Pace Ride](#) [Profile](#) [Sign out](#)

Please enter the following details:

Date of ride*	<input type="text"/>
Time of departure*	<input type="text"/>
Source Location*	<input type="text"/>
Destination Location*	<input type="text"/>
Number of available seats*	<input type="text"/>

Submit



Passenger Bill



UTD-Ride Share Bill

Bill Number	005486
Date	7/8/2015
Ride Number	2455
Journey Start time	5:00 PM
Passenger Source	UTD University Village
Passenger Destination	Collins Creek Mall
Total Fare	\$8.08
Tax Amount	\$1.65

Pay

Software Design: Method Contracts and Pseudo code

Method Name: UpdatePaymentDetails	Class name: PaymentDetails	ID: RS101
Clients (Consumer): UI		
Associated Use case: Payment		
Description of responsibilities: Updates payment details for a given passenger		
Arguments Received: PaymentStatus, verifyCardDetails		
Type of Values returned: UpdateFlag (Success/Failure)		
Pre- Condition: UserID should be unique, payment amount should be valid number, financial institution should return success for verify card details		
Post-Condition: updates the payment details in the database		

Pseudo code:

```
Generate TransactionNumber  
IF verifyCardDetails = "SUCCESS"  
    IF SwipeCard = "SUCCESS"  
        SET PaymentStatus = "COMPLETE"  
    ELSE  
        SET PaymentStatus = "INCOMPLETE"  
ELSE  
    PaymentStatus = DEBARRED  
EXIT
```

Method Name: UpdateSeatCount	Class name: Ride_List	ID: RS102
Clients (Consumer): Booking		
Associated Use case: Booking		
Description of responsibilities: Once the passenger books the ride and the driver accepts this booking, the number of seats available in the car will be updated		
Arguments Received: Ride_No		
Type of Values returned: UpdateFlag (Success/Failure)		
Pre- Condition: Ride_No should be unique		
Post-Condition: updates the payment details in the database		

Pseudo code:

```
Read Ride_No from Booking  
Read NoOfAvailableSeats  
SELECT CASE  
CASE 1 (Booking Status = "BOOKED")  
If NoOfAvailableSeats > 0  
    a. Inform Passenger if Booking success  
    b. NoOfAvailableSeats = NoOfAvailableSeats - 1
```

Else

- Inform Passenger of Booking Failure
- Delete Itinerary_Number from Booking

CASE 2 (BookingStatus="CANCELLED")

- Inform Passenger about booking cancellation
- NoOfAvailableSeats = NoOfAvailableSeats + 1

END CASE

Method Name: SearchRide Class name Ride_List ID: RS103
Clients : Search Request
Associated Use case: Driver & Passenger Match
Description of responsibilities: Searches a list of rides from the Rides_List that confirms to the Search criteria
Arguments Received: Ride_Date , DepartureTime
Type of Values returned: 0{Trip} (List of Trip)
Pre- Condition: User is authenticated. Date and time are both not-null.
Post-Condition: List of trips beginning half an hour before and after the desired time is generated on the given date.

Pseudo code:

```

RECEIVE Ride_Date and DepartureTime from SearchRequest
DO while there are rides remaining in the Rides_list
Start time = DepartureTime from SearchRequest - 30 minutes
End Time = DepartureTime from SearchRequest + 30 minutes
BEGIN IF
IF Ride_Date from SearchRequest = Ride_Date in Ride_List AND
    Start Time < DepartureTime in Ride_List < EndTime
THEN add to Selected Ride:Ride_List
END-IF
END DO WHILE

```

Method Name: UpdateBookingStatus Class name: Booking ID: RS104
Clients : Payment Details , TripDetails
Associated Use case: Process Payment, Cancel Ride Driver, Cancel Ride Passenger
Description of responsibilities: Updates the old booking status to a new one.
Arguments Received: New Booking Status, Ride_No, Itinerary_Number
Type of Values returned: New Booking Status
Pre- Condition: The new Booking Status is determined
Post-Condition: The Booking status is updated.

Pseudo code:

SELECT CASE

CASE 1 (New Booking Status = CANCELLED AND If Cancelled_By_Driver = True)

DO WHILE there are available rides in the RIDE_LIST

BEGIN IF

IF RIDE_NO = Cancelled_Ride_NO

Set Booking Status = CANCELLED

END-IF

END DO WHILE

CASE 2 (New Booking Status = CANCELLED AND If Cancelled_By_Passenger = True)

DO WHILE there are available rides in the RIDE_LIST

IF Itinerary_Number = Cancelled_Itinerary_Number

THEN Set Booking Status = CANCELLED

END DO WHILE

CASE 3 (New Booking Status = PAID)

DO WHILE there are available rides in the RIDE_LIST

IF Itinerary_Number = Successful_Booking_Itinerary_Number

THEN Set Booking Status = PAID

END DO WHILE

END CASE

Method Name: ValidateLoginInfo Class name: Account ID: RS105
Clients : UI
Associated Use case: Login
Description of responsibilities: Authenticates the user to login
Arguments Received: Net ID, Password
Type of Values returned: Boolean (1 – Verified 0 – Login Failed), Error Type
Pre- Condition: An account is created for this NetID
Post-Condition: Account details are displayed.

Pseudo Code

```

RECEIVE Net ID and Password from Login:UI
BEGIN IF
IF RECEIVED Net ID Exists in Account List AND IF Received Password = Account Password
THEN
Set Verified = 1
Error = NULL
ELSE IF RECEIVED Net ID Exists in Account List AND IF Received Password NOT EQUAL TO
Account Password THEN
Set Verified = 0
Set Error = Incorrect Password
ELSE IF RECEIVED Net ID Does not Exist
THEN
SET Verified = 0
SET Error = Account does not exist.
ENDIF

```

Control Mechanisms

Some of the control measures for the system can be as follow:

- a) If a Driver cancels a ride within 30 minutes of scheduled departure time, he/she would be charged a fine of \$3 per booking that his/her ride has. All the passengers who had booked the particular ride and would face inconvenience due to the cancellation would be compensated by being granted a \$3 discount on their next ride. The system allows the discount from the fine that the driver of the cancelled ride pays. Similarly if a Passenger cancels his/her booking within 30 minutes of scheduled departure time, he/she would be charged a fine of \$3 which would be paid to the Driver to compensate for a missed opportunity. The fine would be charged on the student's UTD account. This can be incorporated as a control mechanism that would prevent last minute cancellations.
- b) If a Driver is made to wait for a Passenger for more than 10 minutes of the scheduled Departure time at the pickup location, the booking of the Passenger would get cancelled and he/she would be charged the cancellation penalty of \$3. This mechanism would prevent delay in journey and restrict no-show by passengers.
- c) After selecting a particular ride and adding it to the shopping cart, when a Passenger enters his/her exact pickup and drop-off point on the basis of which the Driver accepts/denies the ride request, the system would accept only Texas zip codes. There would be some rides where the driver would be travelling to outside Dallas locations like to Austin, Houston etc. for which the driver can place a ride in the system, but it would not be likely that a ride would travel outside Texas. Hence this mechanism would filter out invalid ride requests at the input phase thus saving processing time of the system.
- d) Once a Passenger selects a ride and adds it to the shopping cart, an alert message goes to the Driver asking him/her to accept or deny the request. If the ride is of the same day, the driver has 30 minutes to respond to the request before the request expires and the ride gets deleted from the shopping cart of the Passenger prompting him/her to select another ride. This pre-determined time frame would limit the waiting period for a booking to get confirmed and thus prevent uncertainty in the Passenger regarding the booking status. If the ride is not of the same day, the time limit is 3 hours from the time the Driver gets the ride request. However this time limit of 15 minutes or 3 hours is not valid between 11.00 PM and 8.00 AM.
- e) To reconfirm the bank account information, system asks driver to upload a copy of check.

- f) In case, the number of hours driven by driver reaches to threshold level of 20 hours per week for Fall and Spring semester and 40 hours per week for Summer semester, system sends an alert to driver for not taking any more rides. This control is being set to avoid any legal consequences.
- g) In case of insufficient fund in passenger's account. The fare amount will be added to passenger's Galaxy account which s/he has to pay later with an option to remove the debarred status by paying a 15\$ one time penalty.
- h) System does not provide an option to book multiple rides for the same time and same date to avoid any confusion at the time of giving an acceptance by driver.

Conclusion

From the deep study of the entire system and business of UTD-Ride share, we strongly felt there is great potential for the implementation of this service which will not only help students commute and earn some extra bucks but also help them network with fellow Comets and increase their affiliation to the university. There's always room for improvisation of this process but starting with our design will definitely be a great kick start for UTD-Ride share.

To summarize, this service should be marketed as an affordable, safe and convenient University transportation service. The drivers must go through various checks like affiliation to UTD, valid driver's license and insurance. The key to success of this service lies in attracting more and more drivers, that way there will always be some ride available for the riders when they need it. Perks for drivers and powerful marketing activities on UTD website and Facebook will prove useful.

On the other hand the riders will go through UTD Net ID authentication. Following which they are free to use the service. The key to longevity of this service relies on the restricting unwanted cancel of rides after both the parties agree to it. Also payment defaults can be one more potential loophole that will cause loss of trust and driver attrition. Like most Ride-share services we are not storing the card information of the riders from the beginning. This is considering their student status. Exception flows and control mechanisms in this document cover some potential checks we can use in order to avoid some of the loopholes.

With the documented analysis and design steps for UTD-Ride share coupled the exception flows and strong project execution plan; we believe there is enough insight to realize the service.

Project Management Deliverables

Meeting 1			
Project Activities	Allocation of Activities	Planned timeline	Execution Timeline
Group Introduction	All Team members	-	-
Create a project tracker in Google doc	Arundati Chadaga	6/9/2015	6/9/2015
Update tracker with project ideas	Rupashree Bhattacharya, Sejal Shah, Preeta Borkar, Arundati Chadaga, Shivroopa Raghuvanshi	6/10/2015	6/10/2015

Minutes of Project Meeting				
Date	Time	Attendees	Topics Discussed	Conclusion
6/8/2015	9:00 - 10:00 PM	Rupashree Bhattacharya, Sejal Shah, Preeta Borkar, Arundati Chadaga, Shivroopa Raghuvanshi	Group Introduction	Project Manager decided

Meeting 2			
Project Activities	Allocation of Activities	Planned timeline	Execution Timeline
Design a short problem statement for UTD ride share system	Arundati Chadaga	6/16/2015	6/16/2015
Design short Problem Statement on Airport Parking System	Sejal Shaha	6/16/2015	6/16/2015
Design a short Problem Statement on JSOM Waiver and Transfer System	Rupashree Bhattacharya	6/16/2015	6/16/2015
Meet Dr. Raghunathan	Preeta Borkar &	6/23/2015	6/23/2015

Srinivasan for feedback on Project topics	Shivroopa Raghuwanshi		
---	-----------------------	--	--

Minutes of Project Meeting				
Date	Time	Attendees	Topics Discussed	Conclusion
6/8/2015	9.00 - 9.40 PM	Rupashree Bhattacharya, Sejal Shah, Preeta Borkar, Arundati Chadaga, Shivroopa Raghuwanshi	Brainstorming on various Project Topics	Following 3 topics shortlisted:- 1) UTD Ride Sharing 2) Airport Parking System 3) UTD Waiver and Transfer System"

Meeting 3			
Project Activities	Allocation of Activities	Planned timeline	Execution Timeline
Brain storming on UTD ride share system	All	6/22/2015	6/22/2015
Design initial draft of context diagram	Preeta Borkar & Shivroopa Raghuwanshi	6/27/2015	6/29/2015
Design initial draft of Use case diagram	Rupashree Bhattacharya, Arundati Chadaga,	6/27/2015	6/27/2015

Minutes of Project Meeting				
Date	Time	Attendees	Topics Discussed	Conclusion
6/22/2015	9.00 - 9.40 PM	Rupashree Bhattacharya, Preeta Borkar, Arundati Chadaga, Shivroopa Raghuwanshi, Sejal	Finalized the Project for UTD-Ride services and brainstorm of functionalities	1. Brainstormed on actors, databases, system functionality and processes that happen in between actor/database and

		Shah		system. 2. Draw a draft of context diagram and the use case diagram based on the discussion.
--	--	------	--	---

Meeting 4			
Project Activities	Allocation of Activities	Planned timeline	Execution Timeline
Meet professor for feedback on diagrams	Preeta Borkar & Shivroopa Raghuvanshi	7/8/2015	7/8/2015
If diagram gets approved, then start work on class diagram	All	7/8/2015	7/8/2015

Minutes of Project Meeting				
Date	Time	Attendees	Topics Discussed	Conclusion
7/6/2015	9.30 – 10:30 PM	Rupashree Bhattacharya, Preeta Borkar, Arundati Chadaga, Shivroopa Raghuvanshi, Sejal Shah	Discuss changes in Context Diagram Build use case diagram	1. Made changes to the designed context diagram 2. Built a new use case diagram

Meeting 5			
Project Activities	Allocation of Activities	Planned timeline	Execution Timeline
Discuss changes to Use case diagram and context diagram	All	7/8/2015	7/8/2015

Minutes of Project Meeting				
Date	Time	Attendees	Topics Discussed	Conclusion
7/8/2015	9.30 – 10:30 PM	Rupashree Bhattacharya, Preeta Borkar, Arundati Chadaga, Shivroopa Raghuvanshi, Sejal Shah	Discussed modifications in Use case diagram and context diagram as suggested by professor	Made changes to Use Case diagram and context diagram

Meeting 6			
Project Activities	Allocation of Activities	Planned timeline	Execution Timeline
Design Registration Sequence Diagram, Problem Statement, Scope	Sejal Shah	7/16/2015	7/16/2015
Design sequence diagram for Payment, Login, Trip Details, Cancel Ride Driver, Cancel Ride Passenger	Preeta Borkar	7/16/2015	7/16/2015
Design sequence diagram for driver passenger match Use case Write Use case description for driver passenger match use case	Rupashree	7/16/2015	7/16/2015
Write data dictionary for Place a ride, Payment and Process cancellation use case. Write use case description for the same	Arundhati, Shivroopa	7/16/2015	7/16/2015

Minutes of Project Meeting				
Date	Time	Attendees	Topics Discussed	Conclusion
7/14/2015	9.30 – 10:00 PM	Sejal Shah, Rupashree Bhattacharya, Preeta Borkar	Creation of Sequence Diagrams, Data Dictionary and Problem Statement	Decided to allocate task each team member

Meeting 7			
Project Activities	Allocation of Activities	Planned timeline	Execution Timeline
Discuss all diagrams and use cases to verify the process flow	All	7/16/2015	7/16/2015

Minutes of Project Meeting				
Date	Time	Attendees	Topics Discussed	Conclusion
7/16/2015	10.00 - 11.20 PM	Rupashree Bhattacharya, Preeta Borkar, Arundati Chadaga, Shivroopa Raghuvanshi, Sejal Shah	Review and brainstorming on the Use Case Diagram Review and feedback discussion on all the sequence diagrams	1. 2 new use cases - Process Cancel Request and Login added to the use case diagram 2. Various modification done on the sequence diagrams based on feedback from team

Meeting 8			
Project Activities	Allocation of Activities	Planned timeline	Execution Timeline
Use Case Diagram and Sequence diagram for Driver & Passenger Match use case on Visual Paradigm, Data Dictionary for Driver & Passenger Match use case and Login Use Case	Rupashree Bhattacharya	7/21/2015	7/21/2015
Design sequence diagram for registration use case on visual paradigm	Sejal Shah	7/21/2015	7/21/2015
Write final data dictionary for Place Ride, Process Cancellation and Process Payment Use Cases	Arundati Chadaga	7/20/2015	7/20/2015

Draw Sequence diagram for Booking use case on Visual Paradigm, Data Dictionary for Booking Use Case and Registration Use Case	Shivroopa Raghuvanshi	7/21/2015	7/21/2015
Draw Sequence diagram for Place Ride, Login, Process Cancellation and Process Payment use cases on Visual Paradigm	Preeta Borkar	7/21/2015	7/21/2015

Minutes of Project Meeting				
Date	Time	Attendees	Topics Discussed	Conclusion
7/18/2015	10.00 PM - 11.45 PM	Rupashree Bhattacharya, Preeta Borkar, Arundati Chadaga, Shivroopa Raghuvanshi, Sejal Shah	Complete sequence diagram and use case diagram corrections Finalize the data description for each use-case.	1. Problem Statement & Scope finalized 2. Completed corrections to sequence diagrams and completion of use case diagram 3. Complete discussion on each data dictionary

Meeting 9			
Project Activities	Allocation of Activities	Planned timeline	Execution Timeline
Getting approval on all sequence diagrams and class diagram form professor	Preeta Borkar and Shivroopa	7/23/2015	7/23/2015
Reviewed the available documentation	All	7/22/2015	7/22/2015

Minutes of Project Meeting				
Date	Time	Attendees	Topics Discussed	Conclusion
7/22/2015	9:30 - 10:30 PM	Rupashree Bhattacharya, Preeta Borkar, Arundati Chadaga, Shivroopa Raghuwanshi, Sejal Shah	Final review of sequence diagram preparation of class diagram overall layout	1. Final sequence diagram for all use cases 2. Listing of all classes and relations amongst them 3. Rough diagram of class diagram

Meeting 10			
Project Activities	Allocation of Activities	Planned timeline	Execution Timeline
Prepare class diagram and ER Diagram	Rupashree Bhattacharya	7/30/2015	7/30/2015
Software Design	Preeta and Sejal	7/30/2015	7/30/2015
Interface design	Arundati Chadaga, Shivroopa Raghuwanshi	7/30/2015	7/30/2015

Minutes of Project Meeting				
Date	Time	Attendees	Topics Discussed	Conclusion
7/26/2015	3:00 - 3:45 PM	Rupashree Bhattacharya, Preeta Borkar, Arundati Chadaga, Shivroopa Raghuwanshi, Sejal Shah	Distribution of work - Class Diagram, ER Diagram, Software Design, Interfaces	Distributed work among all team members.

Meeting 11			
Project Activities	Allocation of Activities	Planned timeline	Execution Timeline
Control Design	Rupashree Bhattacharya, Shivroopa Raghuwanshi	8/2/2015	7/30/2015
Project Management Track Sheet based on previous MOMs and task distribution	Preeta Borkar	8/2/2015	8/2/2015
Executive Summary Conclusion	Arundati Chadaga	8/2/2015	8/2/2015
Functional Specification Context Diagram	Sejal Shah	8/2/2015	8/2/2015

Minutes of Project Meeting				
Date	Time	Attendees	Topics Discussed	Conclusion
7/30/2015	8:30 - 10:30 PM	Rupashree Bhattacharya, Preeta Borkar, Arundati Chadaga, Shivroopa Raghuwanshi, Sejal Shah	Review of Use Case Descriptions, Class Diagram, ER Diagram, Database constraints, Method Signatures and Structured English, User Interfaces	1. Class Diagram and ER diagram Reviewed and Finalized 2. Use Case Descriptions for all Use Cases Reviewed and Finalized 3. Method Signatures (Contract Form) and Pseudo Code finalized. 4. User Interfaces finalized.

Meeting 12			
Project Activities		Allocation of Activities	Planned timeline
Execution Timeline			
Project Collation and review	All	8/3/2015	8/3/2015

Minutes of Project Meeting				
Date	Time	Attendees	Topics Discussed	Conclusion
8/2/2015	10.00 - 10:30 PM	Rupashree Bhattacharya, Preeta Borkar, Arundati Chadaga, Shivroopa Raghuwanshi, Sejal Shah	Review of Control Mechanisms and context diagram	1. Control Mechanisms finalized 2. Context Diagram finalized