

CS551

Advance Software Engineering

Final Project Report

PASEO Team #2

By

Megha Sai Reddy

Amulya Pindi

Vinutha Muthyala

Naresh Goud Pogakula

Table of Contents

User Manual

Introduction.....	3
Sign In.....	3
Sign Up.....	4
Home Screen.....	5
Navigation Menu.....	6
Emergency.....	7
Give a Ride.....	7
Take a Ride.....	9
View my Rides.....	10
Settings.....	11
Error Recognition and Handling.....	12
Known Bugs and Deficiencies.....	12
Project management report.....	13
Final project evaluation.....	17
Project Proposal.....	19
Project Plan.....	21
First Increment Report.....	22
Second Increment Report.....	37
Third Increment Report.....	62
Fourth Increment Report.....	90
Presentation slides	121
GitHub URL.....	124
Youtube Project Video URL.....	124

User Manual:**Introduction:**

PASEO is a carpooling android application with security measures to ensure that the user feels secured all times. Carpooling is a sharing of car journeys such that more than one person travel together in a car towards the same destination. Carpooling has proved to be a major triumph in reducing the emissions of greenhouse gases as it reduces the number of cars on road, further it has also proved to be a blessing to those who do not own a private car.

Launch App:

To launch the app, tap the application icon on the home screen of your Android device.

Note: This app requires your Android device to be connected to the Internet and It runs on Android version 21 above.



The Paseo Application icon

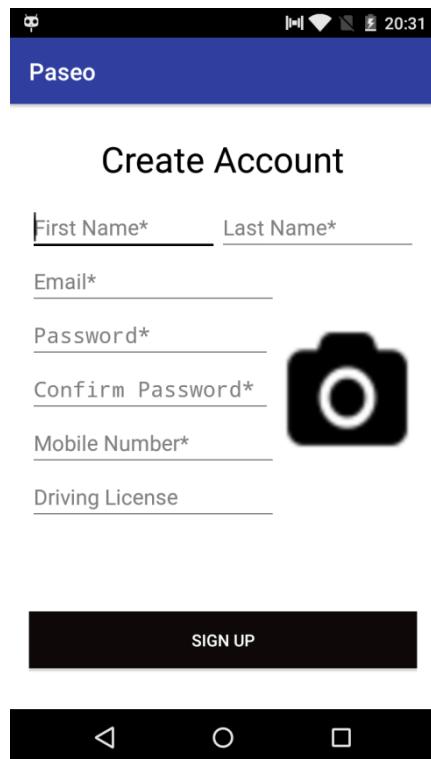
Sign In:

You can sign in using your email and password or you can also login with your Gmail account. Once you enter valid credentials, you will be redirected to home screen.



Sign Up

If you are new user, tap on the New User? Register button on previous screen to register your details.

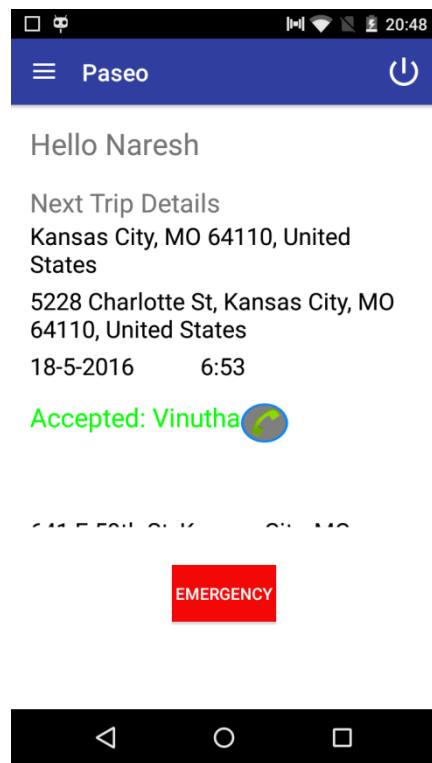


The asterisk(*) symbol indicates that the fields are mandatory. Enter all the details and you can also upload your profile picture by tapping on capture

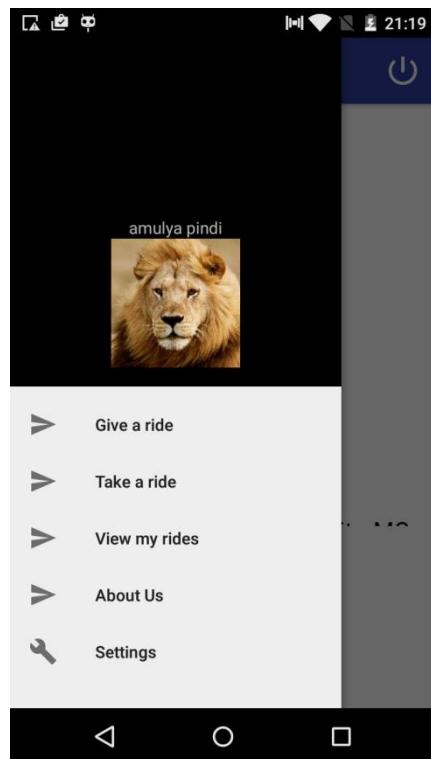
image. Once you filled your details, tap on Sign up to register your user account in application.

Home Screen:

After signing into application, the home screen appears. You can sign out from the application by tapping on the button which is at top-right corner of the screen. You can see your upcoming trip details on the home screen. When you are in Emergency, you can send an alert message to your emergency contacts by tapping on the Emergency Button. You can access the side menu by tapping on the top-left of the screen.



Navigation Menu:



You can see your name and your profile picture as shown in above screen. If you tap on Give a ride, you can able to post your ride details. You can see list of available rides by tapping on Take a ride. To see all your rides information, tap on View my rides. You can update your password, mobile number and emergency contact details using Settings page.

Emergency :



Send SMS To Emergency Contact

hello I'm in danger



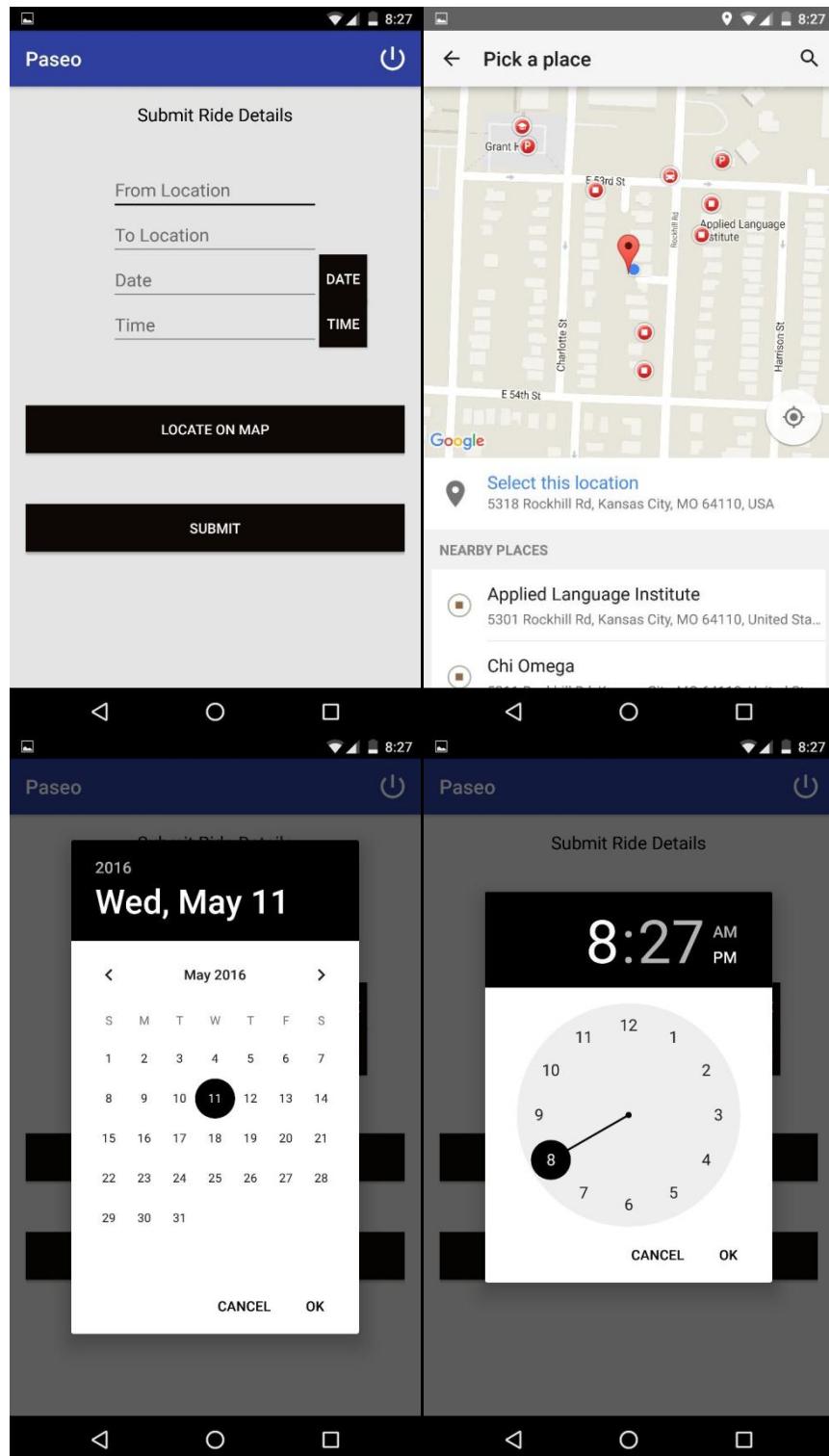
SMS sent.



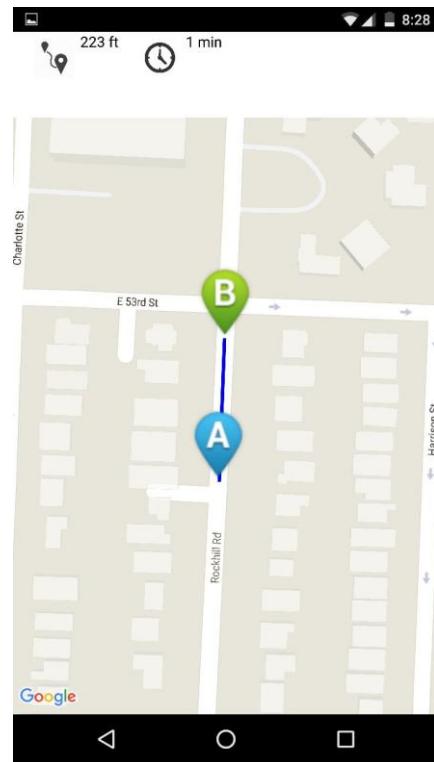
If you tap on Emergency Button on Home screen, a default alert message will be sent to your emergency contacts and another screen opens. There you can again send messages by just tapping on Speech To Text button which will record your voice and will automatically send it as an alert text message to your contacts.

Give a Ride:

You can post your ride details using this screen. You can enter From & To location directly using keyboard or You can pick the location from the Google map by double tapping on the From and To location text fields. You can set the date and time of your journey by tapping on date and time buttons. The following screens give you better understanding.

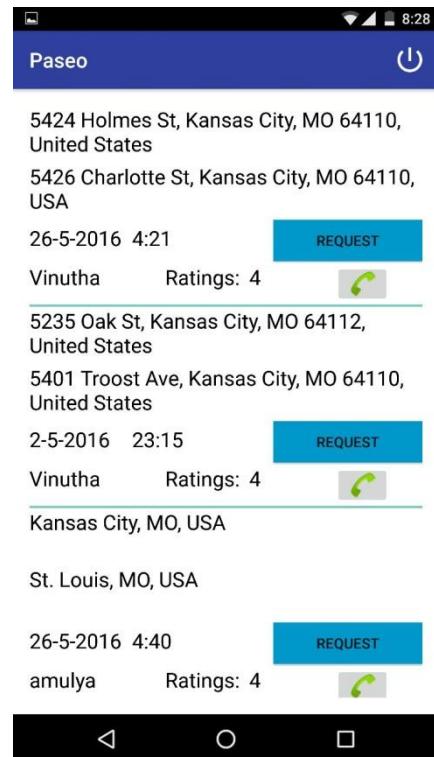


You can also check the distance and the time between source and destination by clicking on the Locate On Map , it also points both the places on the map and shows the path between them.



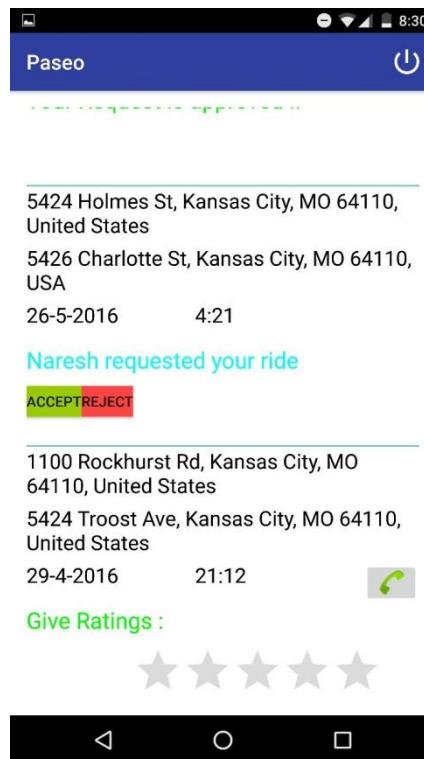
Take a Ride:

You can see all the available rides posted on this screen. You need to scroll down to see all the rides. On every post, from & to location, date & time and rider name & ratings are displayed.

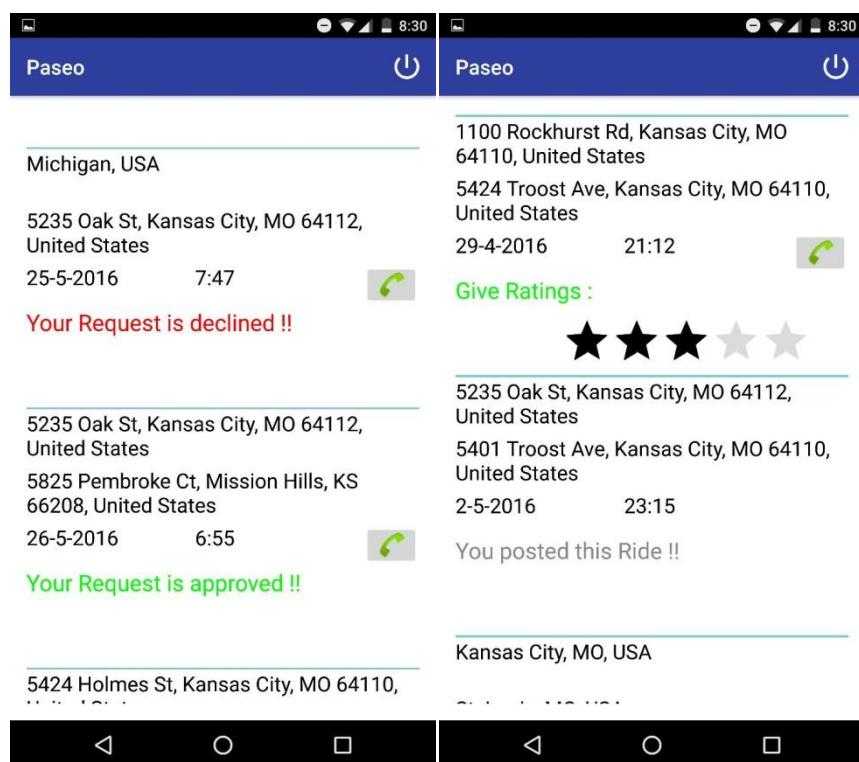


You can request for a particular ride by tapping on Request button. You can also contact the rider by tapping on the call button.

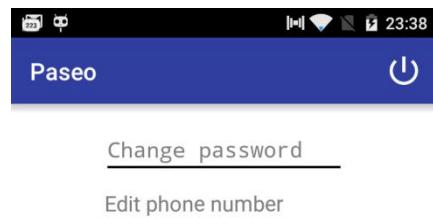
View My Rides:



View my rides contains all the rides that you have posted and requested. You can see one information message on every such ride. If other user requested your ride then you will see the message saying that other user requested your ride. You can accept or reject rides requested by other users by tapping on Accept or Reject Button. You can rate the driver after you finish the trip.



Settings:



**Emergency
Contacts**

Name	Number
Name	Number

SAVE



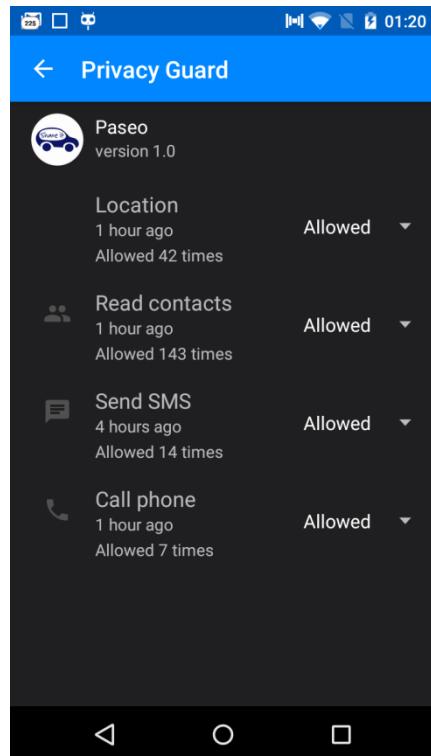
You can update the password and mobile number using this settings screen. You can also add or update emergency contact details. After giving the input, tap on SAVE to update your user details.

Error recognition and Handling

For Android devices 6.0 above, If you find unfortunately application has stopped, make sure that all permissions are granted to application.

To do that manually:

Go to Settings->Apps->Paseo->Permissions



Known bugs and Deficiencies

1. When the user clicks on view my rides tab, sometimes the data loads a tad slow.

This is the only bug that we've encountered so far, we will correct this in next release.

Project Management Report

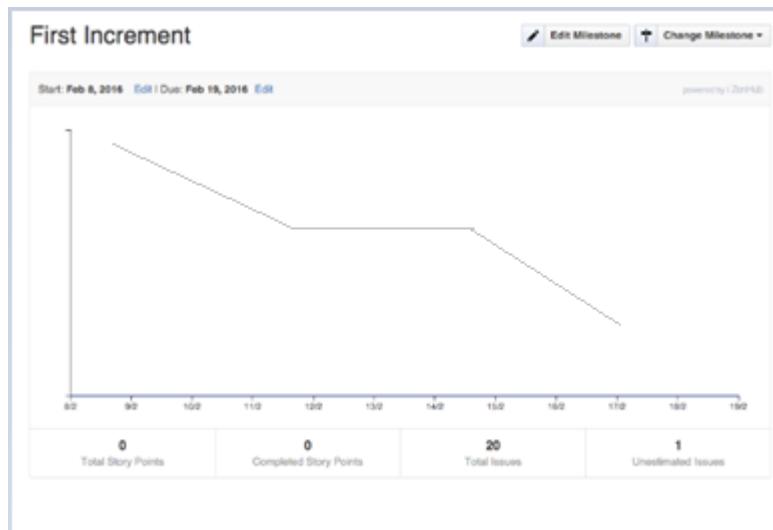
We have used Zen hub to track all our issues and tasks. We've created four milestones for four increments. For each increment, we have assigned tasks to each team member and updated them once we've completed them.

Increment 1:

a.Task Board:

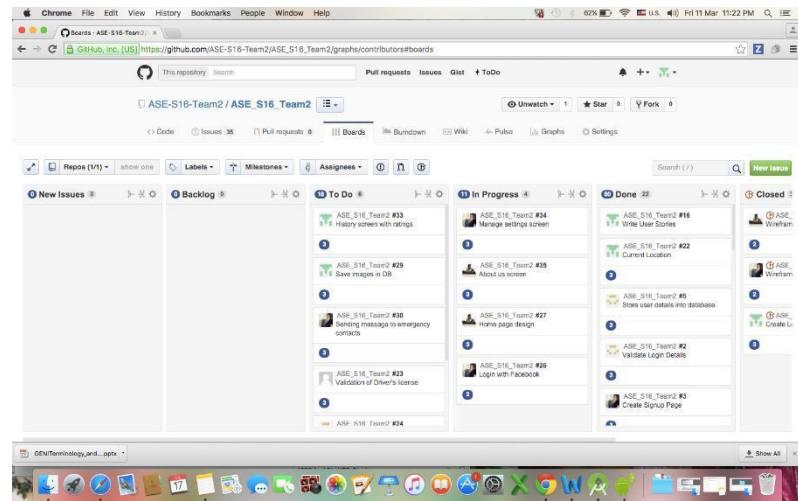
The screenshot shows the Zen Hub interface for the 'ASE_S16-Team2' repository. The task board is divided into several columns: 'New Issues', 'Backlog', 'To Do', 'In Progress', 'Done', and 'Closed'. The 'In Progress' column contains 11 items, while the 'Done' column contains 22 items. Each item has a small thumbnail, a title, and a progress bar indicating completion status. The 'Done' column includes user icons for team members like ASE_Wireframe, and ASE_Wireframe. Below the task board, there is a 'Burndown chart' tab and a 'Burdwn' tab. The bottom of the screen shows a Mac OS X desktop with various application icons in the Dock.

b.Burndown chart:

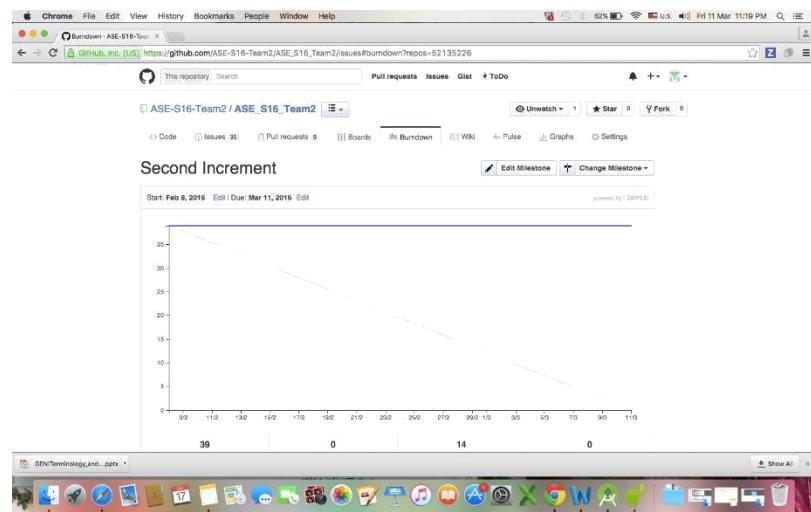


Increment 2

a.Task board

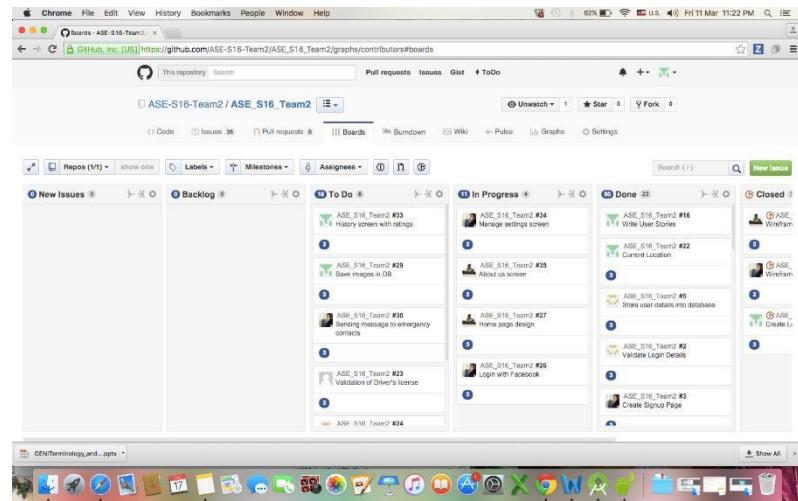


b.Burndown Chart:

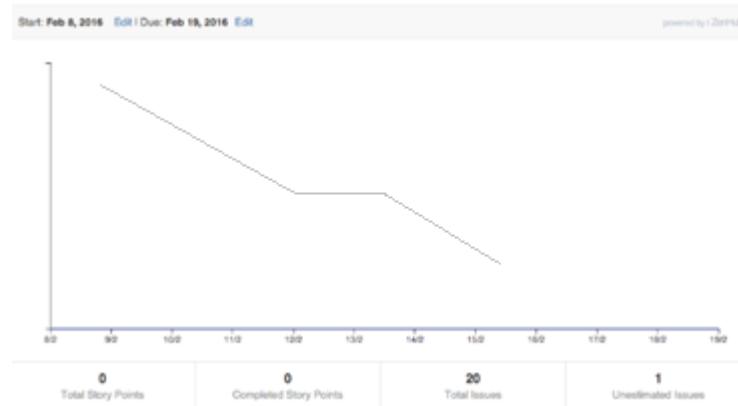


Increment 3

a.Task board

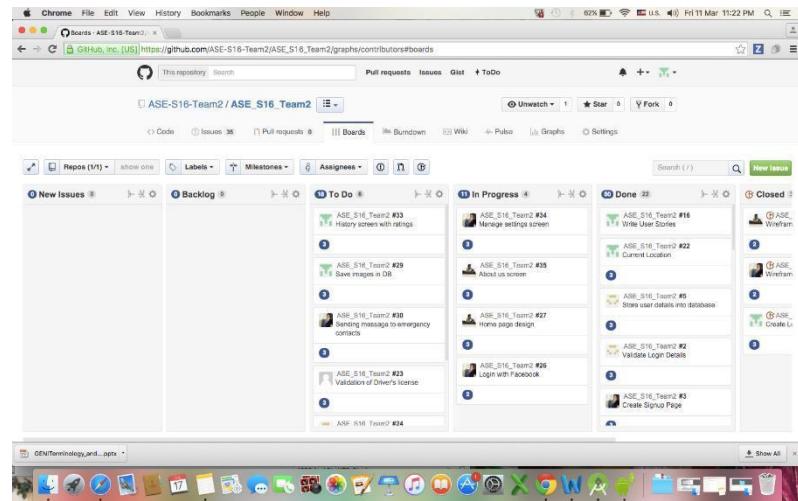


b.Burndown Chart:

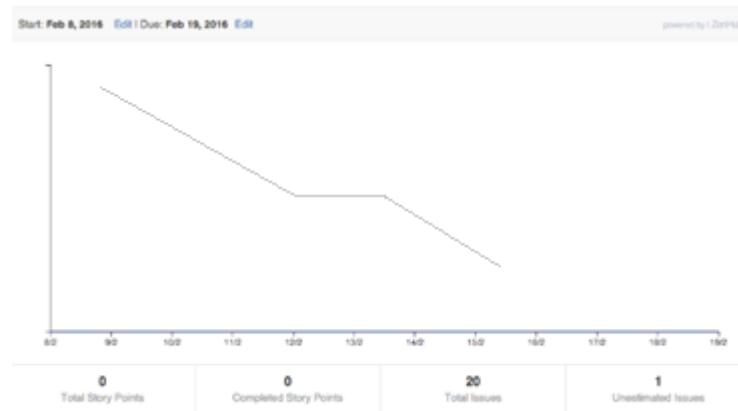


Increment 4

a.Task board



b.Burndown Chart:



Project Evaluation:

Our final increment of the project “PASEO” considerably satisfies our original requirement specifications, in fact it has exceeded our original specifications in some aspects credit to the agile process that was adopted. We initially wanted to create a car pooling application with enhanced security features that could be effortlessly used by the end user and we succeeded in doing that. We basically divided the work on the team member’s strengths and were continuously communicating with each other till final design was done and from there on it was smooth sailing. The continuous increments ensured us to continuously evolve our practices, practice, add/remove, enhance features and change the plan according to the feedback. The project management tools like Zenhub helped us understand the status of our project and reminded us of the tasks that were completed and that need to be completed. It kept us on our feet to meet the deadline. The agile process basically made us to create a better product within the stipulated time with its feedback mechanism. Coming to the management in our group, we basically divided the work on each other’s strength then we created separate issues that were to be addressed by each member with the next deadline so that we could keep track of the tasks that need to be addressed. Our team has addressed all the issues on time and we did a smooth sailing in getting everyone work.

Individual Contribution to Project:

Amulya Pindi: 25%

- Sign-in with Gmail
- Calling and request ride features for accepting rides
- Updating the details to mongo database
- Giving feedback to the riders
- UI of the settings page

Vinutha Muthyala: 25%

- UI part of the home page
- Sending SMS and notifying the user which are a part of emergency
- logout functionality
- retrieving mongo database details for the emergency features
- email implementation
- Voice to text conversion as a part of emergency.

Megha Sai: 25%

- Manage setting features, place pickers for the locations
- Date and time pickers
- Map marker implementations and Google Map services.
- Architecture/use case diagrams
- User Interface,Design of Take Ride screens, wireframes

Naresh Goud: 25%

- To store and retrieve the data from Mongo database,
- Real time scenario of booking and taking ride.
- manage settings features,
- List view for the view ride screen,
- Directions implementation between source and destinations

PROJECT PREPROPOSAL

- GROUP NUMBER: Team-2
- PROJECT TITLE: Ride it.. Share it! (Car Pooling System)

- GROUP MEMBERS:

Vinutha Muthyala
Amulya Pindi
Meghasai Reddy Bodimani
Naresh Goud Pogakula

- PROJECT GOAL AND OBJECTIVES:

Motivation: The hassles faced by a person to travel in colossal cities by own transport make them annoying. Having a feasible car-pooling system in vicinity would make them solace. It also helps in pollution control, reduce traffic congestion and fuel usage.

Significance: This application enables the user to find the people going on the same way so that the commute can be shared. The trips are logged in the user's account to share the costs incurred.

Objectives: To develop a robust and reliable car-pooling application with cogent results and user-friendly methodologies, enabling them to find the pickup and drop points easily and also, to ensure the safety of the users with a brief background check.

System Features:

1. This application allows the users to register themselves either for ride sharing or car-sharing.
2. The user details like name, contact no., e-mail, address, start point and end point etc., are captured and used for further communication among the users.
3. The car-sharing users would update the details of their ride and wait for ride sharing users to contact them.
4. The ride sharing users can request for a car pool specifying the details of their route and location, which allows the car-sharing users to plan accordingly.
5. The users can also link other users about the rides.

- RELATED WORK:

1. International Journal of Advanced Research in Computer Science and Software Engineering
2. A Survey to Justify the Need for Carpooling

- BACKUP PROJECT:

LOCATION ALARM

The objective is to remind the user of a task at a particular location. We generally have different works at different places and we quite often tend to forget certain primitive things like groceries, take away from a certain restaurant, drop by a friend etc. Our app counters this problem. Basically it asks the user to enter any tasks at particular locations and takes its location and adds it to his account. Then we track the GPS of the user and if he is within a close proximity of any location that he has set up, the app sends a notification saying that he needs to carry on a certain task / work at that particular location.

- BIBLIOGRAPHY:

http://www.ijarcsse.com/docs/papers/Volume_3/4_April2013/V3I3-0385.pdf

<http://www.ijsce.org/attachments/File/v5i2/B2612055215.pdf>

<https://en.wikipedia.org/wiki/Carpool> https://en.wikipedia.org/wiki/Real-time_ridesharing <http://www.carpoolworld.com/>

PROJECT PLAN

We have planned our project in four phases

First phase:

We will have to do a lot of research on different carpooling applications and we basically designed the whole project. We need to search different API's that can be used for storing database, different google API's. We plan to develop basic user screens like the sign up, register and sign in pages. We plan to develop the overall plan of the project and the screens for all user stories. We created various tasks in tasks in zenhub that are addressed to different members of the team with deadlines.

Second phase:

We plan to implement API's like the MongoDb to store, retrieve and update the data. We plan to implement the database part, connect it to the signup, register and login page to store the user details in the database. We further plan to add additional screens like view my rides, take rides and manage settings page . We plan to make the implement the take my ride and view my ride page. We also plan to add login with gmail feature

Third phase:

We plan to implement time and date pickers to the UI. We plan to update the profile picture of the user in the database, add Google places api such that a user a view his place on the map. An emergency button such that the user can send a message during emergency. We also plan on implementing the text to speech for messaging and an additional feature of rating the driver once the trip has been completed

Fourth phase:

We plan to add a call button so that the user can contact the driver, further an additional button for the driver such that he can accept or reject rides. In app notifications for the user id the ride has been accepted. We plan to show the locations on the map , the time and distance between them.

CS551 Advanced Software Engineering

PROJECT INCREMENT-1

Project Name: PASEO

Group Details: Team-2

Amulya Pindi

Vinutha Muthyala

Naresh Pogakula

Megha Sai Reddy Bodimani

I INTRODUCTION:

Carpooling is sharing of car journeys such that more than people travel together in a car towards the same destination. Carpooling has proved to be a major triumph in reducing the emissions of greenhouse gases as it reduces the number of cars on road, further it has also proved to be a blessing to those who do not own a private car.

The application “PASEO” is a carpooling android application with security measures to ensure that the user feels secured all times. We register the rider and the passenger and validate the details of the users by checking their passport number, Driver’s license, car details etc. The rider gives the timings of his ride and the passenger picks up the ride based on his convenience. We further ensure full security to the users by providing unique features like sharing current location, tracking the ride in case of emergency providing them a button to give alerts based on location and time.

II PROJECT GOAL AND OBJECTIVES

GOAL:

The main intention of our project is to make carpooling more feasible and secure to both the parties that is the rider and the passenger. We essentially want to bridge the gap of security which plays a major spoilsport and makes people more skeptical towards opting carpooling. We want to ensure maximum security and anonymity by introducing unique features such as “send current location”, “alert when there’s an accident nearby” and “declaring emergency on a click”.

OBJECTIVES (Problem Statement):

There are a lot of impediments faced by people who travel to different parts of the city in their own transport which kind of makes them irksome and there are also people who do not have their own transport and pay a bomb to commute in the city. The amount of environmental pollution that is caused by single driver cars is phenomenal and is on the rise. On an average American spends 40 hours each year stuck in traffic which further adds to our woes.

According to a survey by carpooling just twice a week about 1,600 pounds of greenhouse gases can be kept out of the air each year. Further carpooling cuts down the cost of both the rider and the passenger, makes their life and also for the generations to come.

SPECIFIC FEATURES:

- REGISTER AND LOGIN

User can register themselves as a rider or a passenger through the application by giving their details.

- VALIDATION

The application validates certain details of the rider by checking his history, checks if he has a valid license and checks certain details of the car.

- ANONYMITY

The application ensures possible anonymity of both the rider and the passenger as only their names will be revealed. They can contact each other through the app to ensure certain safety.

- RIDE TRACKING

The user can track his ride through the app and can also share his location on the click of a single button to his desired friends who can also track the users ride to ensure to safety.

- EMERGENCY ALERT

The user can just pop out an emergency signal to the police and his family in case of any accident or in danger on the click of a button or on the utterance of a certain keyword.

- SAFETY METER

The user can see how safe an area is during different times of the day in which they are travelling, so they can be alert at each and every point. The application pops out a warning or alert on case if the area is highly unsafe.

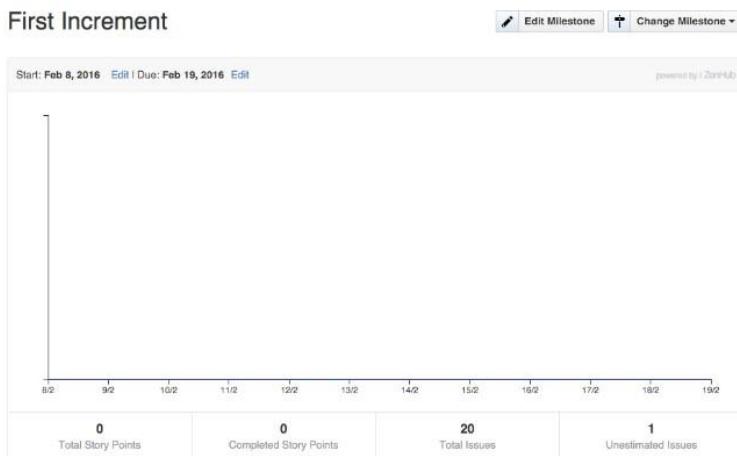
SIGNIFICANCE:

Our application is a perfect blend of carpooling with the right amount of security which should attract a user. It helps people with no own transport to travel at great discounted prices and the ones with own transport to cut down their costs by a large margin and to avoid the daily humdrum of driving in traffic .It further helps to cut down the emission of the greenhouse gases and to ensure a better future for the present and generations to come.

III PROJECT PLAN:

BURNDOWN SCREEN SHOTS:

First Increment Burndown Graph:



Second Increment Burndown Graph:



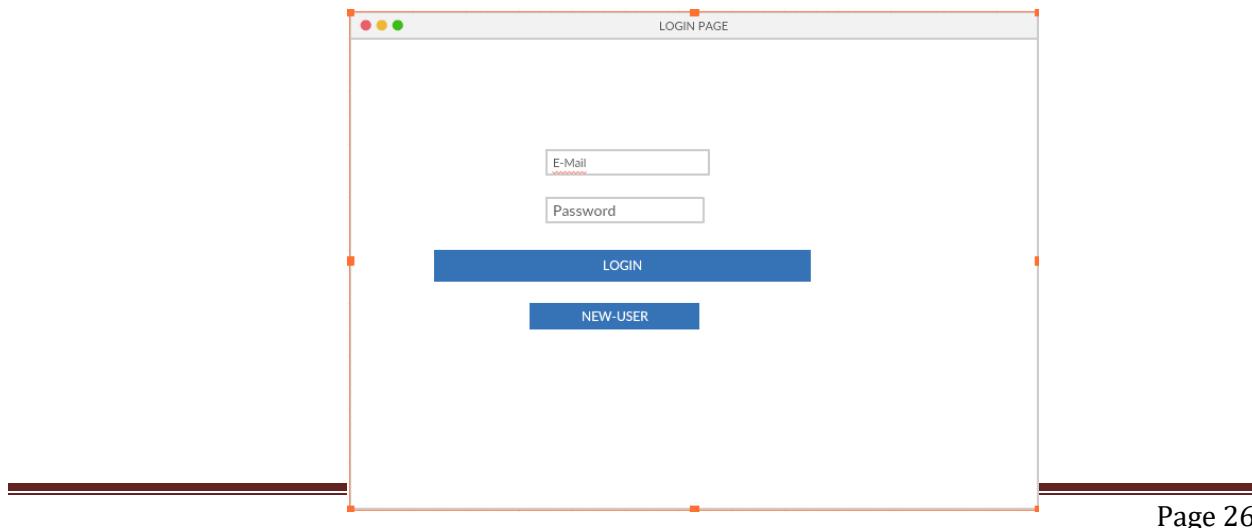
IV FIRST INCREMENT REPORT:

EXISTING API's

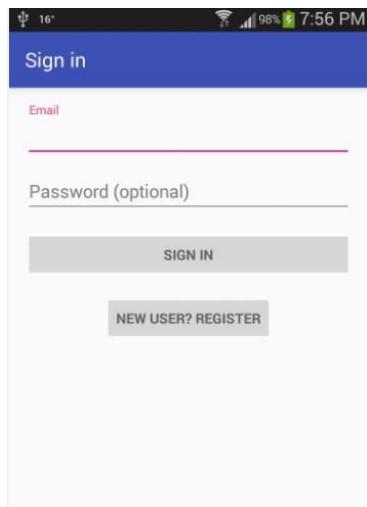
1. Google Maps API
2. Mongo DB API

WIREFRAMES AND MOCKUPS

1. The login Page is used by the users to login the application and New-Users can click register to create an account in the application.



MOCKUP FOR THE ABOVE WIREFRAMES is



2. The Registration page is used by the New-Users to enter their details and save them to database for creating the account.

A registration form titled "REGISTRATION PAGE". It features several input fields: "FIRST NAME", "LAST NAME", "E-MAIL", "PASSWORD", "CONFIRM PASSWORD", and "PASSPORT No.". To the right of these fields is a large rectangular "PROFILE PHOTO" placeholder. Below the input fields is a "DRIVING LICENSE" placeholder. To the right of the "PROFILE PHOTO" placeholder is a blue "UPLOAD PHOTO" button. At the bottom center is a blue "SIGN UP" button.

MOCKUP FOR THE REGISTRATION PAGE

Create Account

First Name*

Last Name*

Email*

Password*

Confirm Password*

Passport No*

Driving Licence



3. Home Page is used by existing users to give in their ride details, request ride details and view location.

HOME PAGE

REQUEST RIDE

FROM LOCATION

TO LOCATION

DATE

TIME

MOCKUP SCREEN FOR THE HOME PAGE:

HomeActivity

REQUEST RIDE LOGOUT

Submit Ride Details

From Location _____

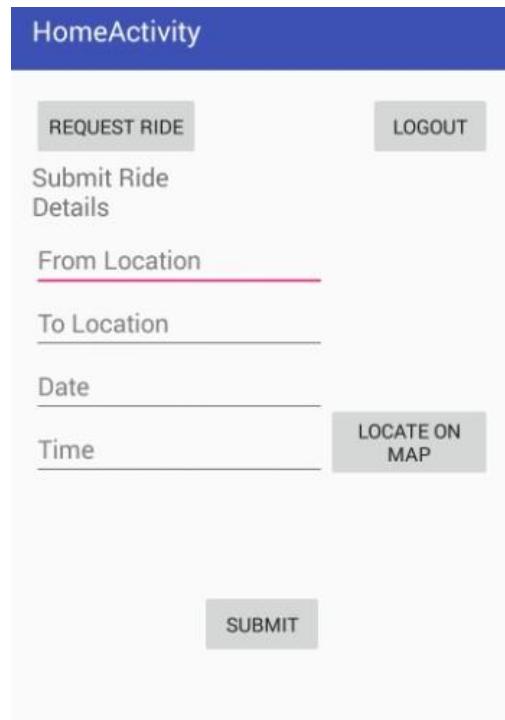
To Location _____

Date _____

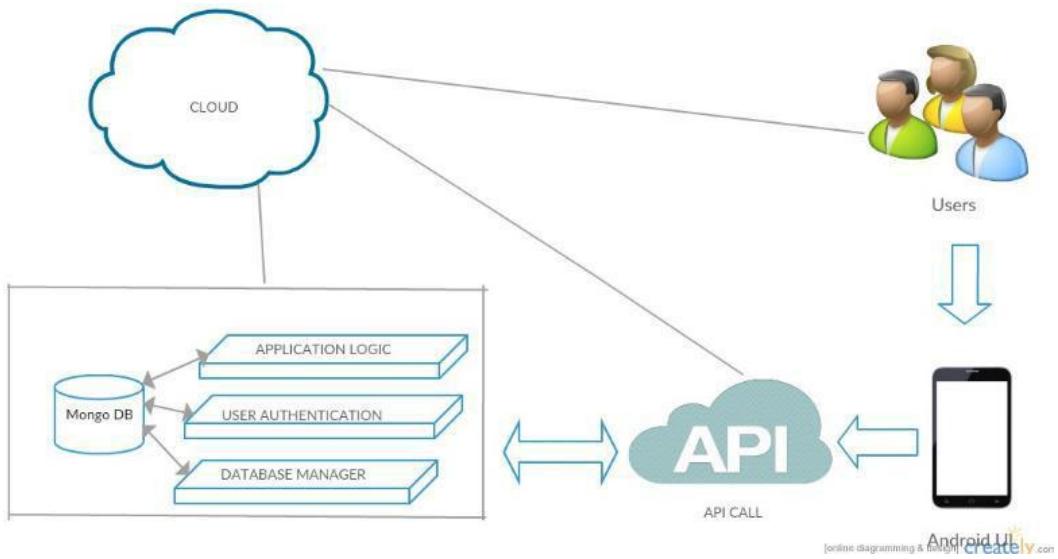
Time _____

LOCATE ON MAP

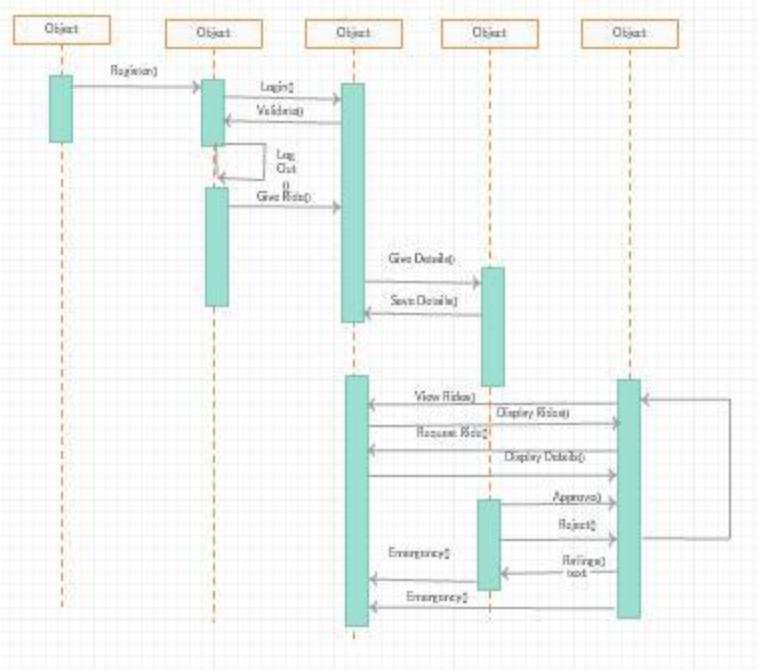
SUBMIT



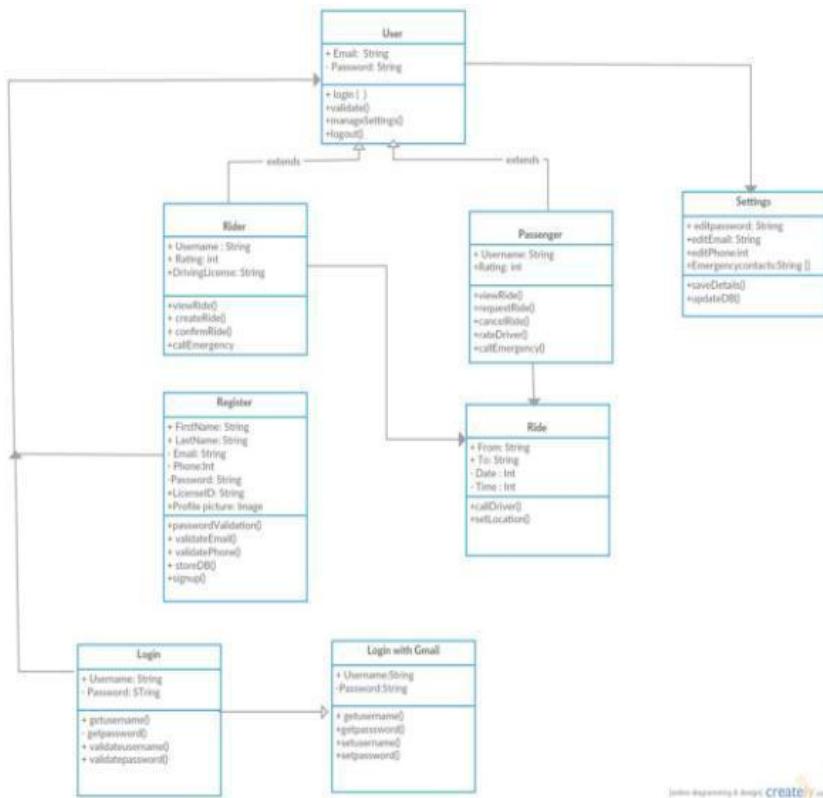
ARCHITECTURE DIAGRAM:



SEQUENCE DIAGRAM:



CLASS DIAGRAM:



USER STORIES:

1. US-1 (Login details of the Application)

User-Story Description: The user should be able to view the signing page enabling him to either login if an existing user or registering if new user.

Actor: Users

Requirement Specification: The screen should consists of two fields accepting the Email and password and two buttons login and New-user.

- Should have an input text field to accept Email
- Should have an input password field to accept password
- Button named Login, when clicked by the user he should be directed to US-2
- Button named New-user, when clicked by the user he should be directed to US-3
- The details of the user are authenticated, if he/she are existing users.

Assumptions: User should be able to perform the login and registration successfully.

2. US-2 (Home screen of the Users)

User-story Description: As a registered user, he should be able to either request a ride or give the ride details and submit.

Actor: Users

Requirement Specifications: The screen should consists of fields enabling user to provide details of his ride and submit. Should also consists of three buttons for requesting ride, showing present location and logout.

Ride details should consists of four input fields.

- Should have an input text field to accept the From location
- Should have an input text field to accept the To location
- Should have a date field to select the desired date.
- Should have a Time field to select the desired time.
- Button named 'Submit' at the bottom of the page to submit the ride details
- Button named 'Locate me' to provide the present location of the user
- Button named 'Logout' to come out of application.
- On clicking submit the ride details should be saved.
- Button named 'Request Ride' to enable user to request a ride

Assumptions: User should be able to easily submit the ride details and request ride.

3. US-3 (Registering the new-users)

User-story Description: Enabling the user to register into the application by taking the basic details. Actors: Users

Requirement Specifications: As a New-user, he should be able to register with the application by filling the details of the user.

- Screen should consist of an input text field to accept the first name of the user
- Should consist of an input text field to accept the last name of the user.
- Should consists of an email field to accept the email.
- Should have a set password field enabling the user to set password
- Should have a confirm password field to verify the password.
- Should have an input text field to enter the passport number.
- Should have an input text field to enter the user's driver's license.
- Should have a facility for the user to select a profile picture.
- Button to submit the details to database.

The following fields should be mandatory:

First name, email, Password, passport.

Assumptions: New-users should be able to create an account in the application.

4. US-4 (Showing the current location)

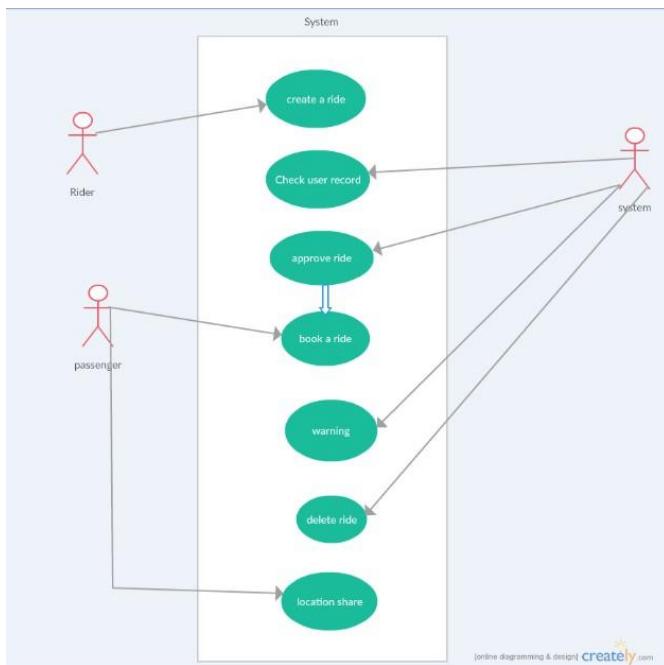
User-story Description: As a registered user, the user should be able to see his present location. Actors: Users

Requirement Specifications: User should be able to locate his current location.

- The button would be placed in the registration page and once the user clicks the button his current location would be displayed. (This would be updated in further US)

Assumptions: User should be able to locate his present location.

USE CASE:



TESTING DOCUMENTS:

1. Login details of the Application

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to login/New-user should be able to register	The user would be able to enter the E-mail and password and click login to enter to application	User should be able to login	User is logged-in.
2.	User should be able to login/New-user should be able to register	New-User should click the Sing-up button to register, to navigate to registration page	New-users are directed to register page.	New-Users are able to view the registration page.
3.	User should be able to login/New-user should be able to register	User should enter E-Mail and Password and are validated for the users.	User details are authenticated.	User is able to login only when accurate details are submitted
4.	User should be able to login/New-user should be able to register	User is entered wrong details an error would pop up.	User credential are to be validated and if wrong a pop-up would be displayed	User would be able to view a message if incorrect details are

2. Registration screen of the Users

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	New-User should be able to register by giving the basic details.	New-Users should fill all the fields and click submit.	New-users should be successfully register.	New-Users are registered.
2.	New-User should be able to register by giving the basic details.	New-User should enter all the mandatory fields. First name, email, password, passport	New-users are registered once the user enters the mandatory	New-Users are able register.
3.	New-User should be able to register by giving the basic details.	New-Users would be notified with validation regarding the mandatory fields.	New-Users are able to view an error if fields are left blank.	New-users are notified with error.
4.	New-User should be able to register by giving the basic details.	On submit the details entered are saved to the Data Base.	New-User details are save to the data base once they click the submit button.	Details of new users are save to data base.

3. Home Screen of the Users

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	Registered User should be able to either request a ride or give in ride details.	Users can fill in the ride details and submit their details.	Users should be able to fill the ride details to provide a ride	Users can successfully provide their ride
2.	Registered User should be able to either request a ride or give in ride details.	Users can click locate me button to view their current location.	Users can view their present location.	Users are able to view their location.
3.	Registered User should be able to either request a ride or give in ride details.	User can click request ride to view the existing trips.	Users are able to view the trip details if they request ride.	Users are able to check the trip details.
4.	Registered User should be able to either request a ride or give in ride details.	Users can click on logout to come out of the application.	Users are redirected to the login page if they select the logout.	Users are logged out successfully.

4. Showing the current location

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	Registered User should be able to locate their location.	Users are able to vies their location on clicking locate me button.	Users are to view their current location.	Present location of the user is displayed.

BIBLIOGRAPHY:

1. <https://en.wikipedia.org/wiki/Carpool>
2. https://en.wikipedia.org/wiki/Real-time_ridesharing
3. <http://www.carpoolworld.com/>
4. http://www.ijarcsse.com/docs/papers/Volume_3/4_April2013/V3I3-0385.pdf
5. http://www.ijscce.org/attachments/File/v5i2/B26120_55215.pdf

CS551 Advanced Software Engineering

PROJECT INCREMENT-2

Project Name: PASEO

Group Details: Team-2

Amulya Pindi

Naresh Pogakula

Vinutha Muthyala

Megha Sai Reddy Bodimani

I INTRODUCTION:

Carpooling is sharing of car journeys such that more than people travel together in a car towards the same destination. Carpooling has proved to be a major triumph in reducing the emissions of greenhouse gases as it reduces the number of cars on road, further it has also proved to be a blessing to those who do not own a private car.

The application “PASEO” is a carpooling android application with security measures to ensure that the user feels secured all times. We register the rider and the passenger and validate the details of the users by checking their passport number, Driver’s license, car details etc. The rider gives the timings of his ride and the passenger picks up the ride based on his convenience. We further ensure full security to the users by providing unique features like sharing current location, tracking the ride in case of emergency providing them a button to give alerts based on location and time.

II OBJECTIVES

There are a lot of impediments faced by people who travel to different parts of the city in their own transport which kind of makes them irksome and there are also people who do not have their own transport and pay a bomb to commute in the city. The amount of environmental pollution that is caused by single driver cars is phenomenal and is on the rise. On an average American spends 40 hours each year stuck in traffic which further adds to our woes.

According to a survey by carpooling just twice a week about 1,600 pounds of greenhouse gases can be kept out of the air each year. Further carpooling cuts down the cost of both the rider and the passenger, makes their life and also for the generations to come.

FEATURES:

- REGISTER AND LOGIN

User can register themselves as a rider or a passenger through the application by giving their details.

- VALIDATION

The application validates certain details of the rider by checking his history, checks if he has a valid license and checks certain details of the car.

- ANONYMITY

The application ensures possible anonymity of both the rider and the passenger as only their names will be revealed. They can contact each other through the app to ensure certain safety.

- RIDE TRACKING

The user can track his ride through the app and can also share his location on the click of a single button to his desired friends who can also track the users ride to ensure to safety.

- EMERGENCY ALERT

The user can just pop out an emergency signal to the police and his family in case of any accident or in danger on the click of a button or on the utterance of a certain keyword.

- SAFETY METER

The user can see how safe an area is during different times of the day in which they are travelling, so they can be alert at each and every point. The application pops out a warning or alert on case if the area is highly unsafe.

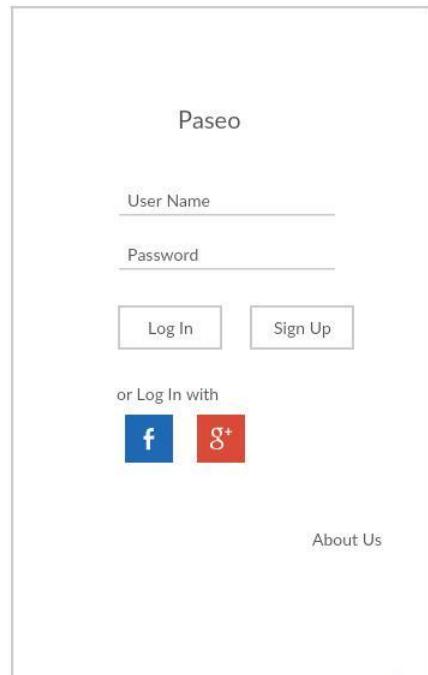
EXISTING API's:

1. Google Maps API
2. Mongo DB API

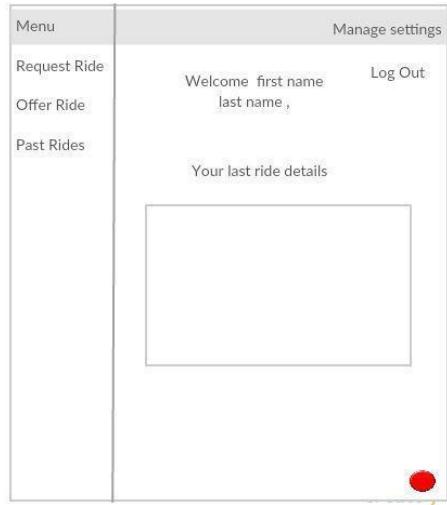
DETAIL DESIGN OF FEATURES:

WIREFRAMES AND MOCKUPS

1. The login Page is used by the users to login the application and New-Users can click register to create an account in the application. Users can also login with either Gmail or Facebook. A link About Us is provided to give user information about the app.



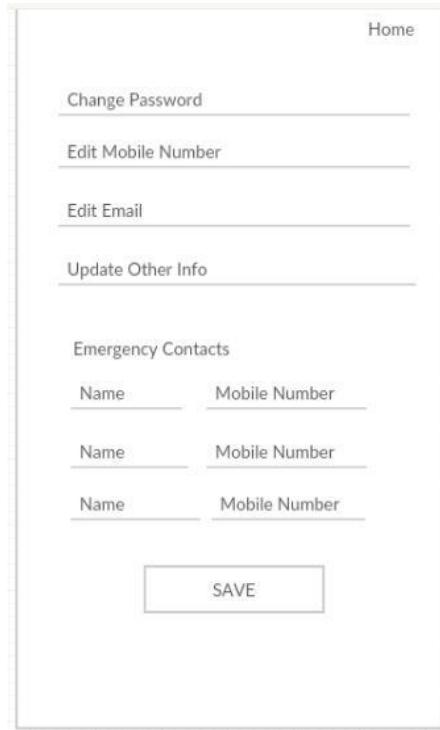
2. Registered user would be navigated to the Home screen as below where the user can perform multiple functions.



3. The Registration page is used by the New-Users to enter their details and save them to data base for creating the account.

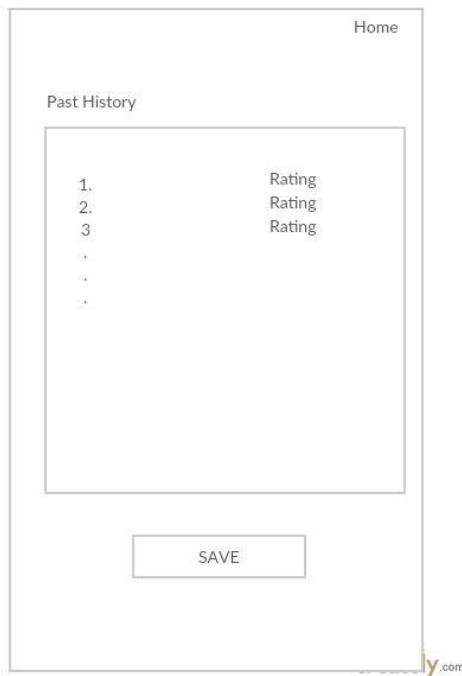
A registration form titled 'Welcome'. It includes fields for 'First Name*' and 'Last Name*', both marked with an asterisk. There are also fields for 'Email*', 'User Name*', 'Password*', 'Confirm Password*', and 'Mobile Number*'. Below these are three file upload fields: 'Driving Licence' with a 'Browse' button, 'Other proofs' with a 'Browse' button, and 'Upload Pic' with a 'Browse' button. At the bottom is a large 'Sign Up' button.

4. Settings page is used by the user to update details like password, phone number and other details



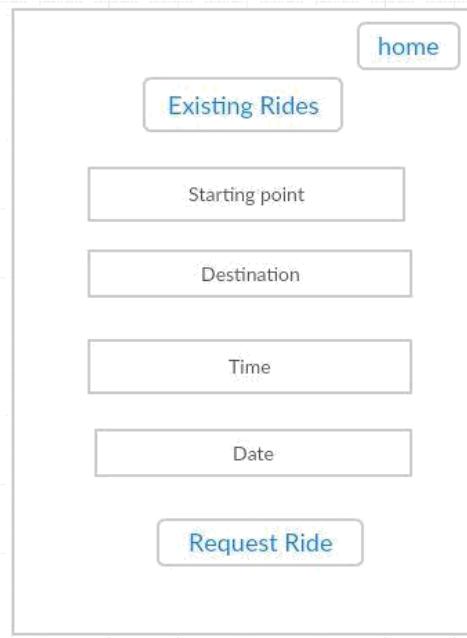
A wireframe of a settings page. At the top right is a 'Home' button. Below it are four horizontal input fields: 'Change Password', 'Edit Mobile Number', 'Edit Email', and 'Update Other Info'. Underneath these is a section titled 'Emergency Contacts' containing three rows, each with 'Name' and 'Mobile Number' fields. At the bottom is a large rectangular 'SAVE' button.

5. History page would display the past rides of that user which allows the user to rate the rider.



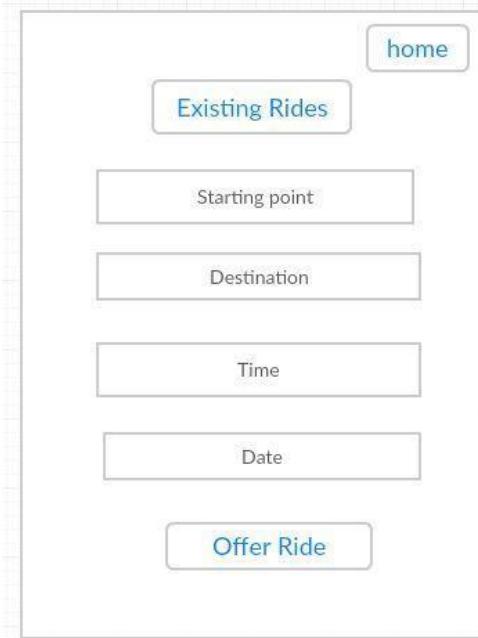
A wireframe of a history page. At the top right is a 'Home' button. Below it is a section titled 'Past History' containing a list of past rides. The list includes items 1, 2, 3, and an ellipsis, followed by 'Rating' repeated three times. At the bottom is a large rectangular 'SAVE' button.

6. Request ride page is used for the user to search/ create rides.



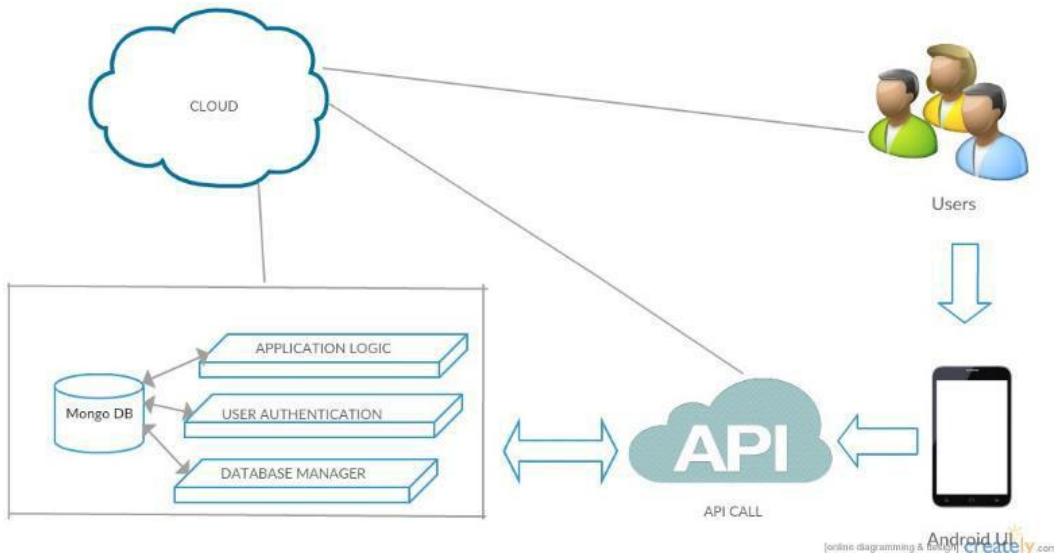
The diagram shows a user interface for requesting a ride. At the top right is a 'home' button. Below it is a 'Existing Rides' button. The main area contains four input fields: 'Starting point', 'Destination', 'Time', and 'Date'. At the bottom is a large 'Request Ride' button.

7. Offer ride page is used by the user to give/search rides.

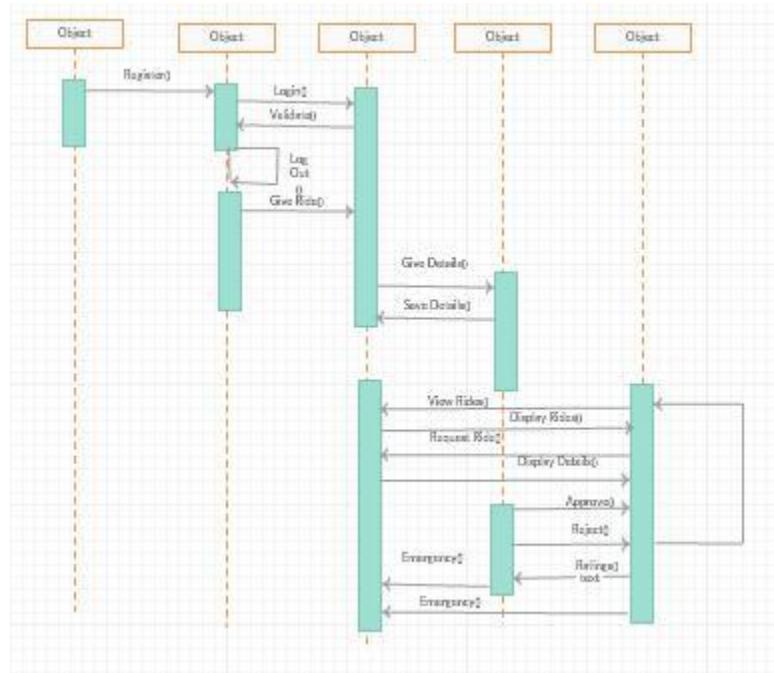


The diagram shows a user interface for offering a ride. It has a similar layout to the request page, with a 'home' button at the top right, an 'Existing Rides' button, and four input fields for 'Starting point', 'Destination', 'Time', and 'Date'. However, the bottom button is labeled 'Offer Ride'.

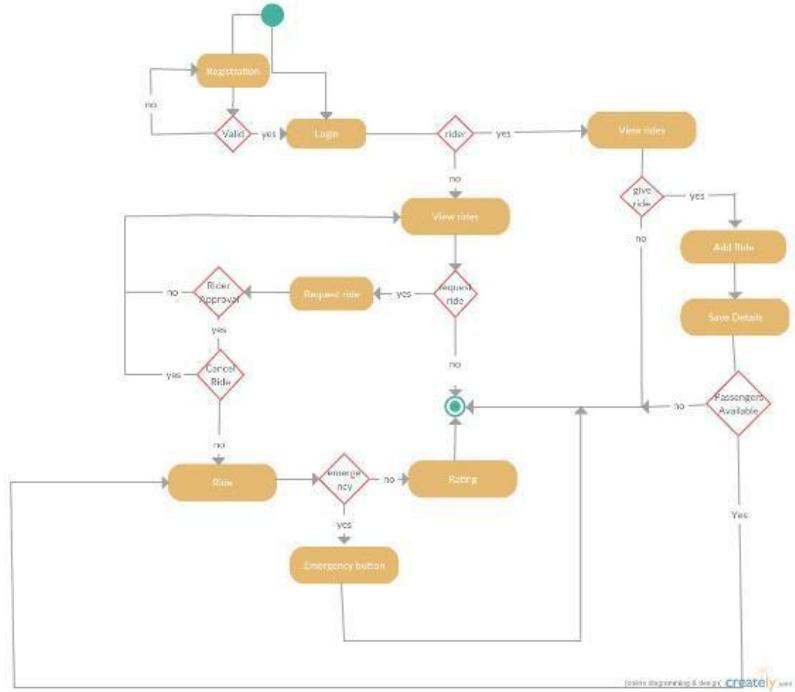
ARCHITECTURE DIAGRAM:



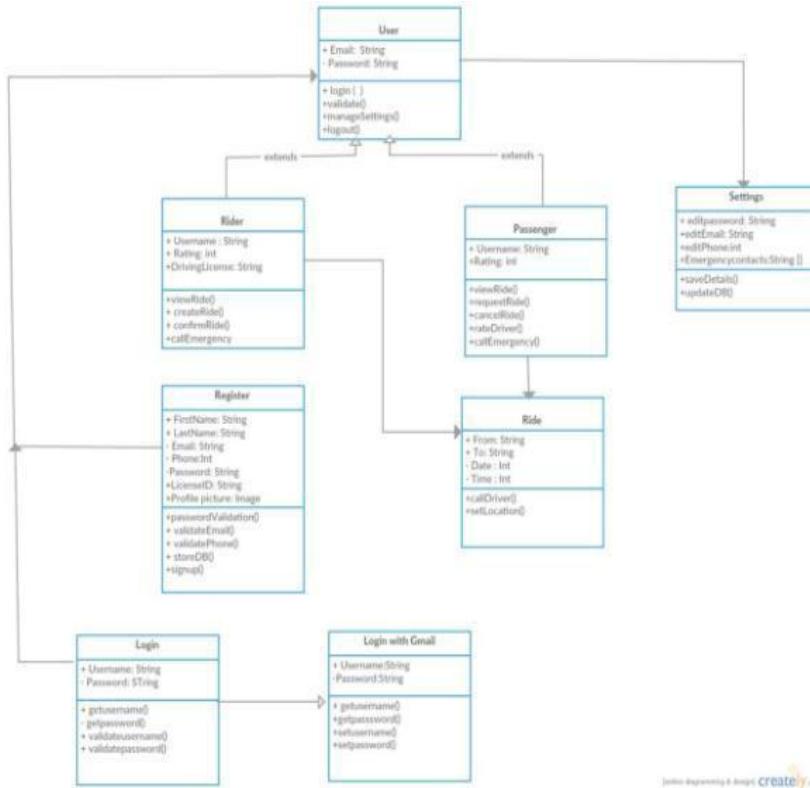
SEQUENCE DIAGRAM:



ACTIVITY DIAGRAM:



CLASS DIAGRAM:



USER STORIES:

1. US-1 (Login details of the Application)

User-Story Description: The user should be able to view the signing page enabling him to either login if an existing user or registering if new user. User can login with Gmail or Facebook.

Actor: Users

Requirement Specification: The screen should consists of two fields accepting the Email and password and two buttons login and New-user. Gmail and Facebook login buttons as well.

- Should have an input text field to accept Email
- Should have an input password field to accept password
- Button named Login, when clicked by the user he should be directed to US-2
- Button named New-user, when clicked by the user he should be directed to US-3
- The details of the user are authenticated, if he/she are existing users.
- Page should consists of Gmail and Facebook buttons enabling the user to login with existing accounts.

Assumptions: User should be able to perform the login and registration successfully.

2. US-2 (Home screen of the users)

User-Story Description: The user can see his active rides if he has currently booked any and he can further choose if he wants to give a ride or take a ride. He can logout at any time and can send a message on the hit of a button during emergency. User can also manage his settings.

Actor: Users

Requirement Specification: The screen should consist of fields displaying the users past rides, it should further have buttons to request rides, offer rides, past rides, manage settings, logout and an emergency button.

- Should have a side menu to display different buttons.
- Button named Manage Settings when clicked by the user should be directed to US-4.
- Button named Logout to come out of the application.
- Button named Request Ride when clicked by the user should be directed to US-5.
- Button named Offer Ride when clicked by the user should be directed to US-6.
- Button named Past History when clicked by the user should be directed to US-7.
- Button named Emergency to send a message to contacts during emergency.

Assumptions: User should be able to perform the login and use functionalities successfully.

3. US-3 (Registering the new-users)

User-story Description: Enabling the user to register into the application by taking the basic details. Actors: Users

Requirement Specifications: As a New-user, he should be able to register with the application by filling the details of the user.

- Screen should consist of an input text field to accept the first name of the user
- Should consist of an input text field to accept the last name of the user.
- Should consists of an email field to accept the email.
- Should consist if an input text filed to accept username.
- Should consist of an input filed to accept mobile number.
- Should have a set password field enabling the user to set password
- Should have a confirm password field to verify the password.
- Should have a button to upload other proofs.
- Should have button to upload the user's driver's license.
- Should have a button to upload the users profile photo.
- Button to submit the details to database.

The following fields should be mandatory:

First name, email, Password, passport, mobile number, username

Assumptions: New-users should be able to create an account in the application.

4. US-4 (Update users profile)

User-Story Description: The user should be able to change his password, mobile number, email and he can also add the contact details of his family to send them a message during emergency.

Actor: Users

Requirement Specification: The screen should consists of fields accepting the Email, password, mobile number, other info and two buttons login and New-user.

- Should have an input text field to enter new password.
- Should have an input field to enter new phone number.
- Should have an input field to enter new email.
- Should have an input filed to update other info.
- Should have 3 input fields to enter name and mobile number.
- Should have a button save to successfully update the details to the database.
- Should have a button home to enable the user to move to home page.

Assumptions: User should be able to perform the update and save details successfully.

5. US-5 (Request ride)

User-story Description: The user should be able to request ride between two

places. **Actor:** Users

Requirement Specifications: The screen should consist of the following fields and buttons

- Should have an input text field to enter the starting point.
- Should have an input filed to enter the destination.
- Should have an input field to enter the time.
- Should an input filed to enter the date of journey.
- Should have a button Request Ride to request a ride to the rider.
- Should have a link which on click should display the existing rides available between the two locations.
- Should have a button Home to redirect to the home page.

Assumptions: User should be able to request the ride successfully.

6. US-7 (History)

User-story Description: The user should be able to view his past rides and give ratings to rides.

Actor: Users

Requirement Specifications: The screen should consist of the following fields and buttons

- Should have an input text box to display the users past rides.
- Should have a field to rate the driver of his previous ride.
- Should have a Save button to save the updated rating.
- Should have a home to populate user to home page.

Assumptions: User should be able to view all his rides.

7. US-6 (Offer Ride)

User-story Description: The user should be able to offer ride details and view the active rides available for him

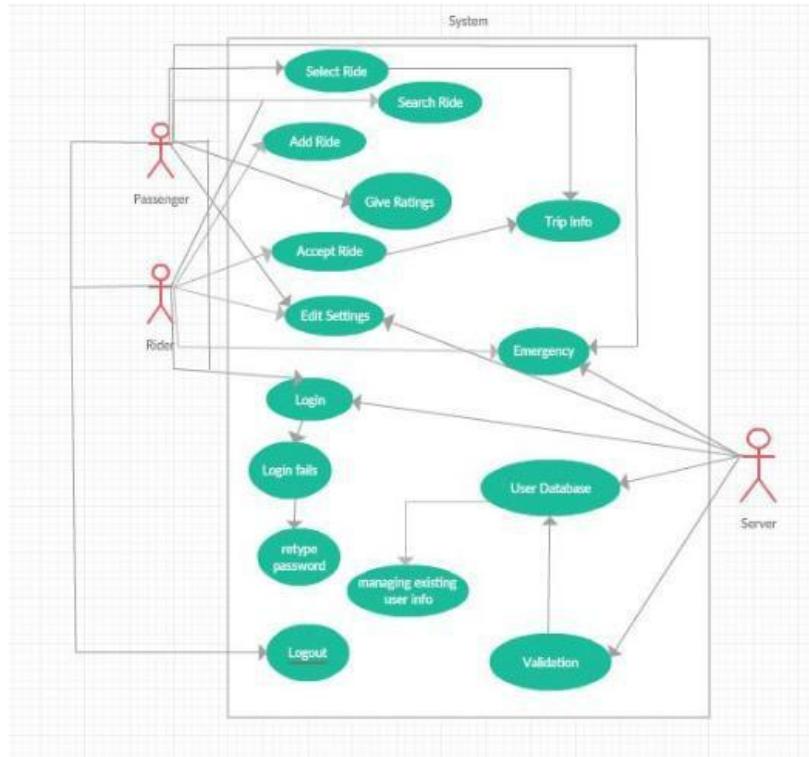
Actors: Users

Requirement Specifications: The screen should consists of the following inputs.

- Should have an input text field to enter the starting point.
- Should have an input filed to enter the destination.
- Should have an input field to enter the time.
- Should an input filed to enter the date of journey.
- Should have a button Offer Ride to request a ride to the rider.
- Should have a link which on click should display the existing rides available between the two locations.
- Should have a button Home to redirect to the home page.

Assumptions: User should be able to request the ride successfully.

USE CASE DIAGRAM:



SERVICE DESCRIPTION:

1. Google API is used to locate the present location of the user and also enabling the New-users to login with Gmail.
2. Used Mongo DB to store the user details and to retrieve the user details when needed.

TESTING DOCUMENTS:

1. Login details of the Application

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to login/New-user should be able to register	The user would be able to enter the E-mail and password and click login to enter to application	User should be able to login	User is logged-in.
2.	User should be able to login/New-user should be able to register	New-User should click the Sing-up button to register, to navigate to registration page	New-users are directed to register page.	New-Users are able to view the registration page.
3.	User should be able to login/New-user should be able to register	User should enter E-Mail and Password and are validated for the users.	User details are authenticated.	User is able to login only when accurate details are submitted

4.	User should be able to login/New-user should be able to register	User is entered wrong details an error would pop up.	User credential are to be validated and if wrong a pop-up would be displayed.	User would be able to view a message if incorrect details are entered.
5.	User should be able to login/New-user should be able to register	User would be able to sign-in with Gmail or Facebook.	User should be successfully logged in.	New-users are able to login.
6.	User should be able to login/New-user should be able to register	User can read about the application following that link	User can view the information about the application.	User is able to view the application related data.

2. Registration screen of the Users

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	New-User should be able to register by giving the basic details.	New-Users should fill all the fields and click submit.	New-users should be successfully registered.	New-Users are registered.
2.	New-User should be able to register by giving the basic details.	New-User should enter all the mandatory fields. First name, email, password, passport	New-users are registered once the user enters the mandatory fields	New-Users are able register.
3.	New-User should be able to register by giving the basic details.	New-Users would be notified with validation regarding the mandatory fields.	New-Users are able to view an error if fields are left blank.	New-users are notified with error.
4.	New-User should be able to register by giving the basic details.	On submit the details entered are saved to the Data Base.	New-User details are save to the data base once they click the submit button.	Details of new users are save to data base.

3. Ride

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	Registered User should be able to either request a ride or give in ride details.	Users can fill in the ride details and submit their details.	Users should be able to fill the ride details to provide a ride	Users can successfully provide their ride details.
2.	Registered User should be able to either request a ride or give in ride details.	Users can click request ride to view available rides	Users can view available rides	Users are able to view available rides

3.	Registered User should be able to either request a	User can click offer to create a ride	Users should be able to create rides	Users are able to create the rides
----	--	---------------------------------------	--------------------------------------	------------------------------------

	ride or give in ride details.			
4.	Registered User should be able to either request a ride or give in ride details.	Users can click on logout to come out of the application.	Users are redirected to the login page if they select the logout.	Users are logged out successfully.

4. Home screen

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to choose different services	Users should click request ride to redirect to ride page.	Users should be able to redirect to the ride page	Users are successfully redirected.
2.	User should be able to choose different services	Users should click offer ride to redirect to the ride page	Users should be able to redirect to the ride page	Users are successfully redirected.
3.	User should be able to choose different services	Users should click past rides to redirect to the previous rides page	Users should be redirected to the past rides page	Users are successfully redirected to the page.
4.	User should be able to choose different services	Users can click on logout to come out of the application.	Users are redirected to the login page if they select the logout.	Users are logged out successfully
5.	User should be able to choose different services	Users can click manage settings button to redirect to manage settings page	Users should be able to redirect to the page	User are successfully redirected

5. Manage settings screen

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to update his details	Users should enter any field that he wants to be update	Users should be able to update the information	Users have successfully updated the information.
2.	User should be able to update his details	Users should be able to redirect to home page	Users should be able to redirect to the ride page	Users are successfully redirected.
3.	User should be able to update his details	User can enter the emergency contacts and click save	Users should be able to redirect to home page	Users are successfully redirected to home page

6 .Past rides

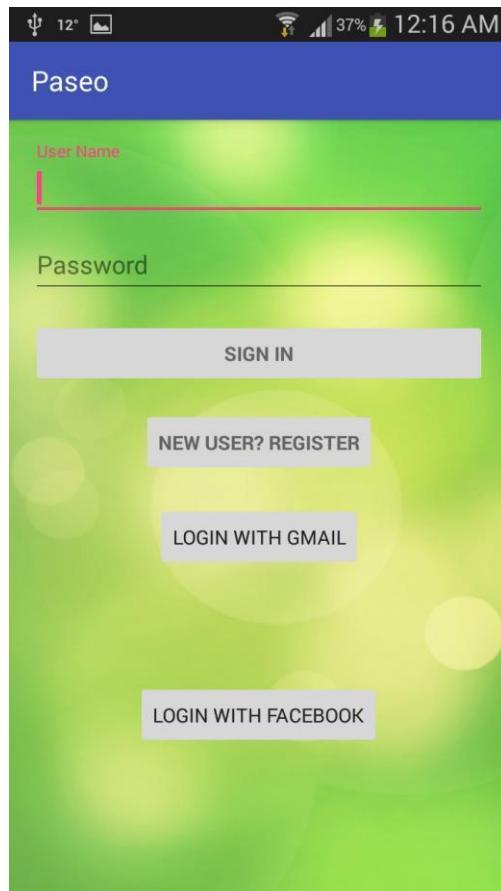
S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to rate their previous rides	Users should rate their previous ride and click save.	Users should be able to redirect to the home page	Users are successfully redirected.

IMPLEMENTATION:

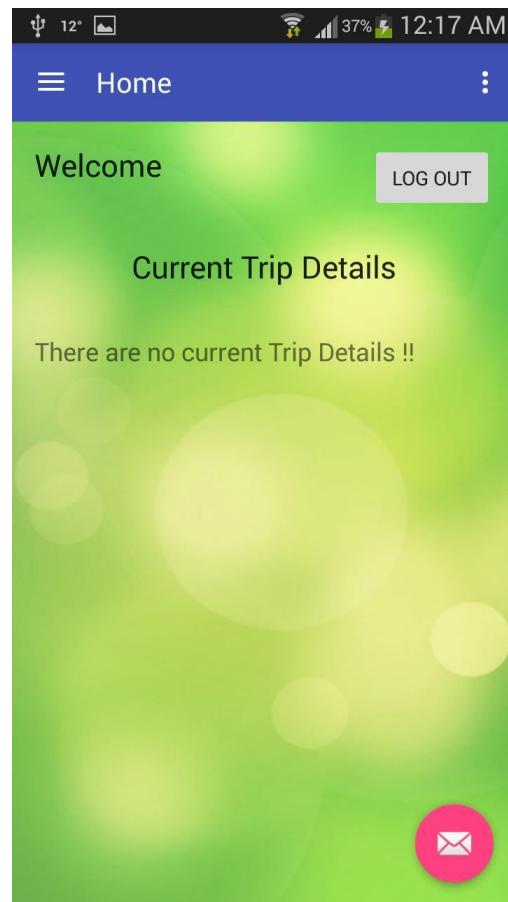
- ② Mobile client Implementation: Implementation in mobile is feasible and is easy to use as most of the present world runs on smart phones. We have implemented the project using Android.
- ② Server Implementations: Used Mongo database to store and retrieve the details of the user about his ride and profile info.

DEPLOYMENT:

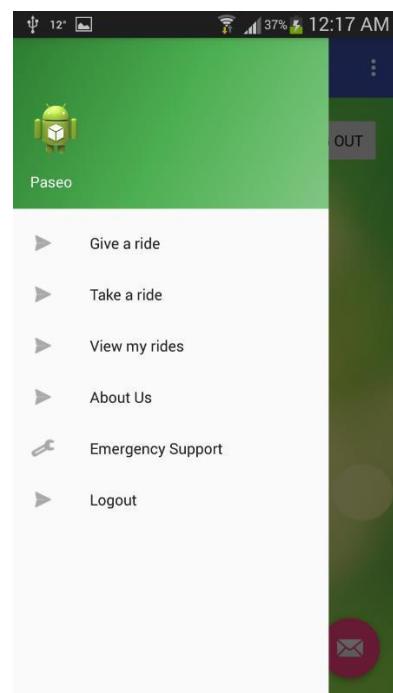
1. Login Page:



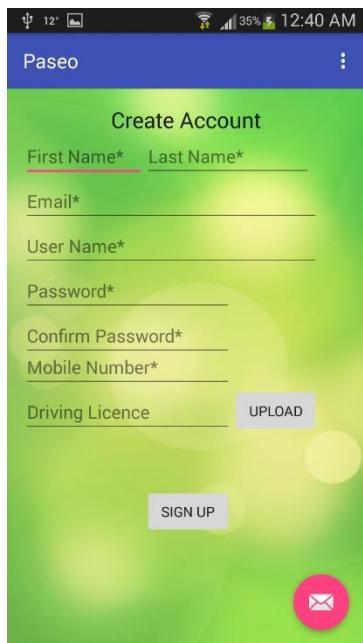
2. Home Page:



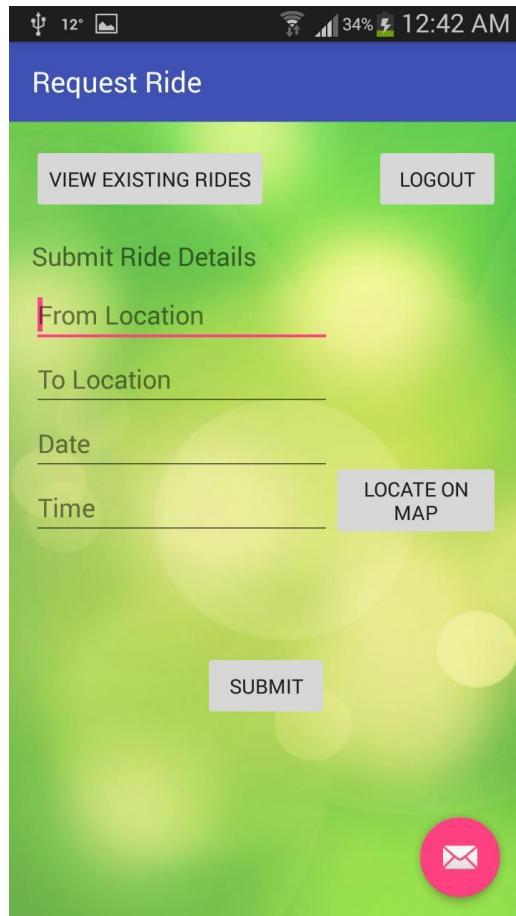
Menu Screen:



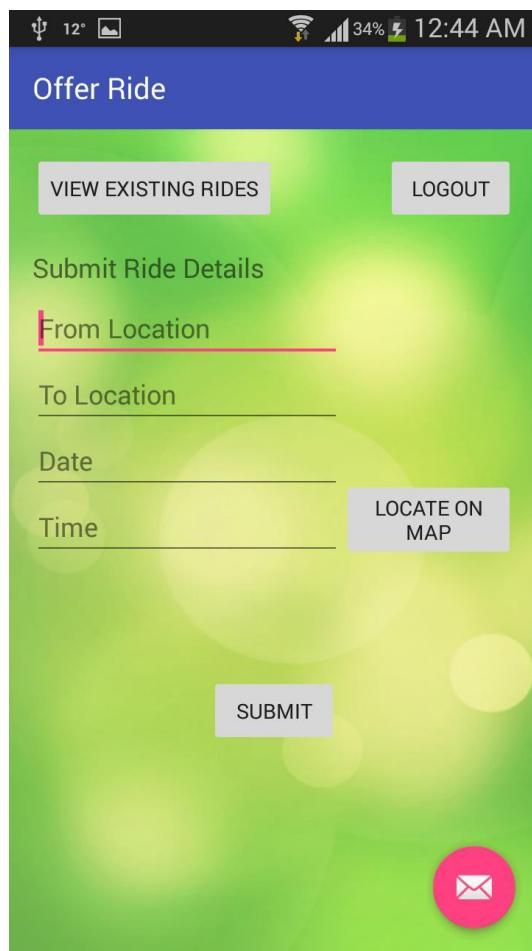
3. Registration Page:



4. Ride Request:



5. Offer Ride:



PROJECT MANAGEMENT:

- ❑ **Project Timeline:** March 11th 2016.
- ❑ **Members:** Amulya1, VinuthaMuthyala, Naresh, Megha Sai.
- ❑ **Task Responsibilities:** Issues are created in github for each members and every member have collectively worked to complete them in given time. Have shared the work equally among us.

Work Completed:

Description:

- ❑ **Stories:** The login page, Home Page, Registration page are completed with the UI and functionality.
- ❑ **Service Design:** The Google service is implemented to notify the user with the current location and Mongo database service to save and retrieve the information of the user and ride, they are implemented to enhance the functionality of the user.

- ② **Service Implementation:** Google API and Mongo database and are being worked successfully.

Responsibilities:

We all have worked together to provide an effective and clear implementation of the project.

Amulya Pindi: Sign-in with Gmail, UI of the settings user page, documentation.

Vinutha Muthyala: UI part of the home page, Functionality of retrieving the data from Mongo database. Google Map services.

Megha Sai: Architecture/use case diagrams, UI of the login page, design of request screens, wireframes.

Naresh: To store the data to Mongo database, UI of the registration screen.

Time Taken and Contributions: All have contributed 5hrs each almost every alternate day and almost 10hrs completely the last three days.

Work to be completed:

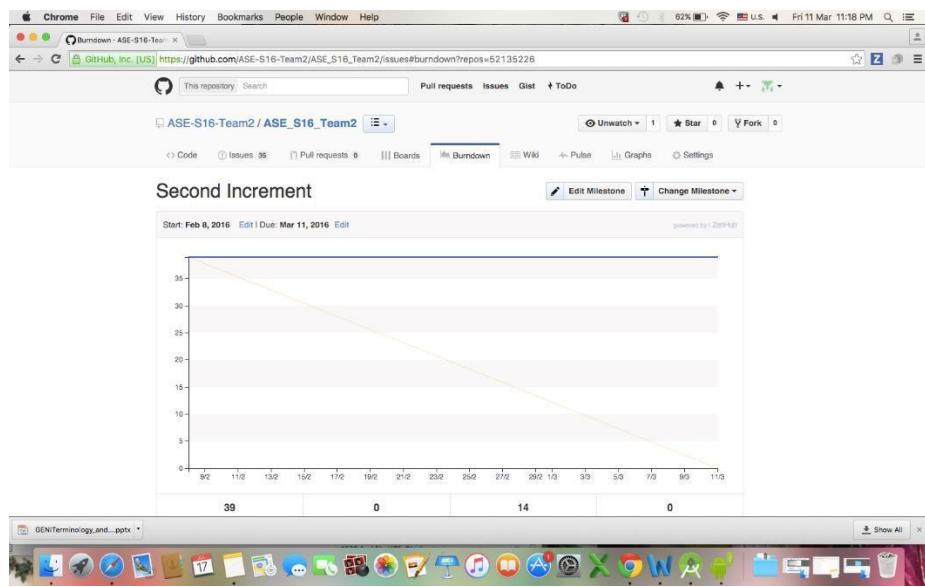
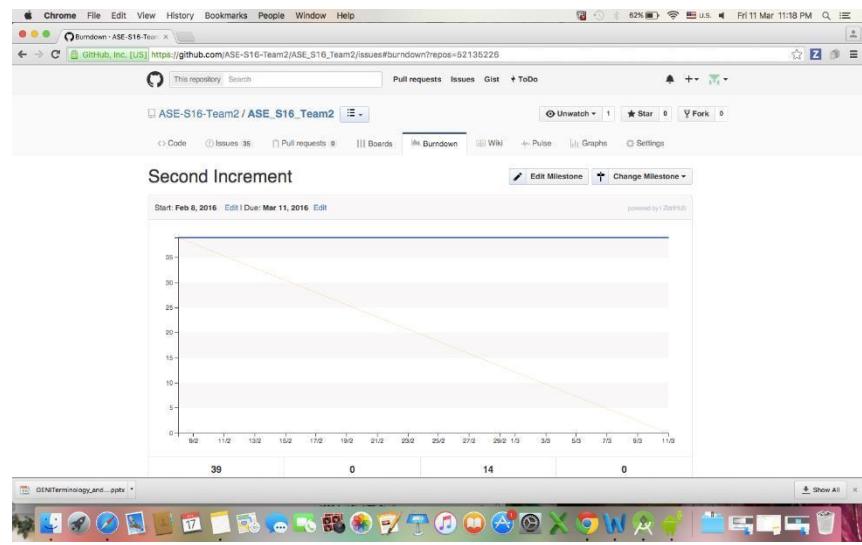
Description:

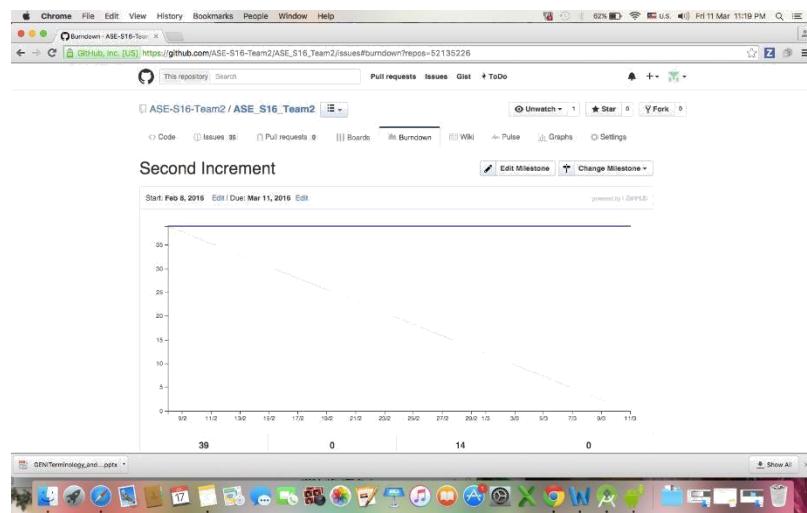
The validation in fields, the Database updates with respective to each page, displaying the security features for the user.

Issues:

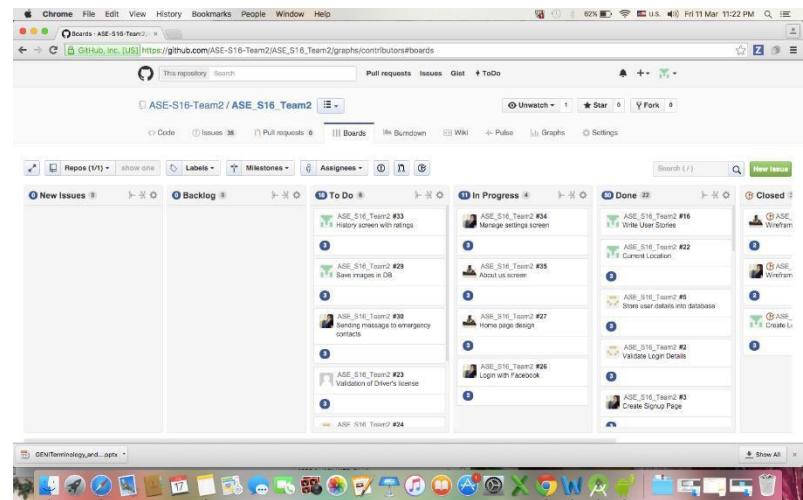
Experienced few issues in implementing the validations and also while updating the information in mongo database.

Burndown Graphs:





DASH BOARD:



BIBLIOGRAPHY:

1. <https://en.wikipedia.org/wiki/Carpool>
2. https://en.wikipedia.org/wiki/Real-time_ridesharing
3. <http://www.carpoolworld.com/>
4. http://www.ijarcsse.com/docs/papers/Volume_3/4_April2013/V3I3-0385.pdf
5. <http://www.ijscce.org/attachments/File/v5i2/B2612055215.pdf>

CS551 Advanced Software Engineering

PROJECT INCREMENT-3

Project Name: PASEO

Group Details: Team-2

Amulya Pindi

Naresh Pogakula

Vinutha Muthyalu

Megha Sai Reddy Bodimani

I INTRODUCTION:

Carpooling is sharing of car journeys such that more than people travel together in a car towards the same destination. Carpooling has proved to be a major triumph in reducing the emissions of greenhouse gases as it reduces the number of cars on road, further it has also proved to be a blessing to those who do not own a private car.

The application “PASEO” is a carpooling android application with security measures to ensure that the user feels secured all times. We register the rider and the passenger and validate the details of the users by checking their passport number, Driver’s license, car details etc. The rider gives the timings of his ride and the passenger picks up the ride based on his convenience. We further ensure full security to the users by providing unique features like sharing current location, tracking the ride in case of emergency providing them a button to give alerts based on location and time.

II OBJECTIVES

There are a lot of impediments faced by people who travel to different parts of the city in their own transport which kind of makes them irksome and there are also people who do not have their own transport and pay a bomb to commute in the city. The amount of environmental pollution that is caused by single driver cars is phenomenal and is on the rise. On an average American spends 40 hours each year stuck in traffic which further adds to our woes.

According to a survey by carpooling just twice a week about 1,600 pounds of greenhouse gases can be kept out of the air each year. Further carpooling cuts down the cost of both the rider and the passenger, makes their life and also for the generations to come.

FEATURES:

- REGISTER AND LOGIN

User can register themselves as a rider or a passenger through the application by giving their details.

- VALIDATION

The application validates certain details of the rider by checking his history, checks if he has a valid license and checks certain details of the car.

- ANONYMITY

The application ensures possible anonymity of both the rider and the passenger as only their names will be revealed. They can contact each other through the app to ensure certain safety.

- RIDE TRACKING

The user can track his ride through the app and can also share his location on the click of a single button to his desired friends who can also track the users ride to ensure to safety.

- EMERGENCY ALERT

The user can just pop out an emergency signal to the police and his family in case of any accident or in danger on the click of a button or on the utterance of a certain keyword.

- SAFETY METER

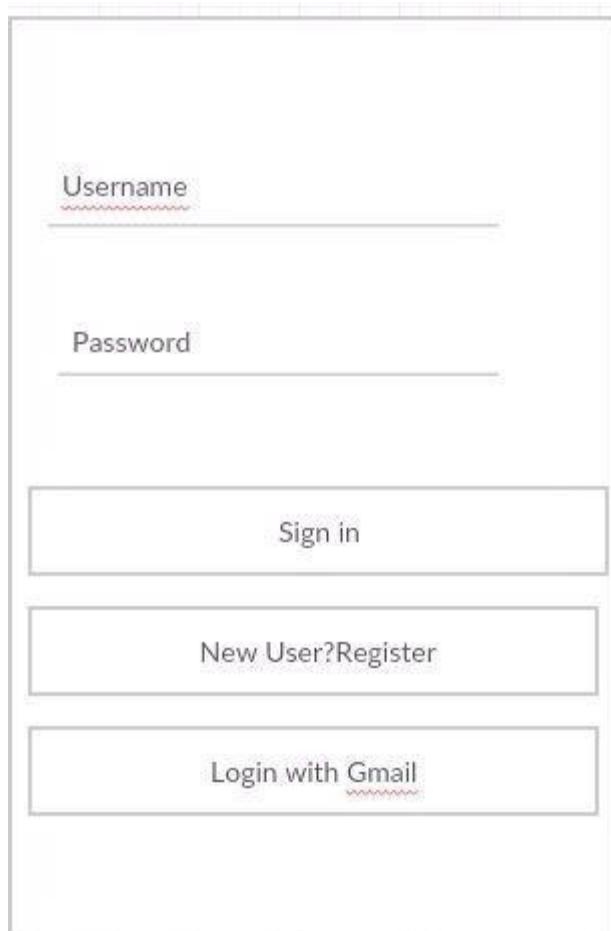
The user can see how safe an area is during different times of the day in which they are travelling, so they can be alert at each and every point. The application pops out a warning or alert on case if the area is highly unsafe.

EXISTING API's:

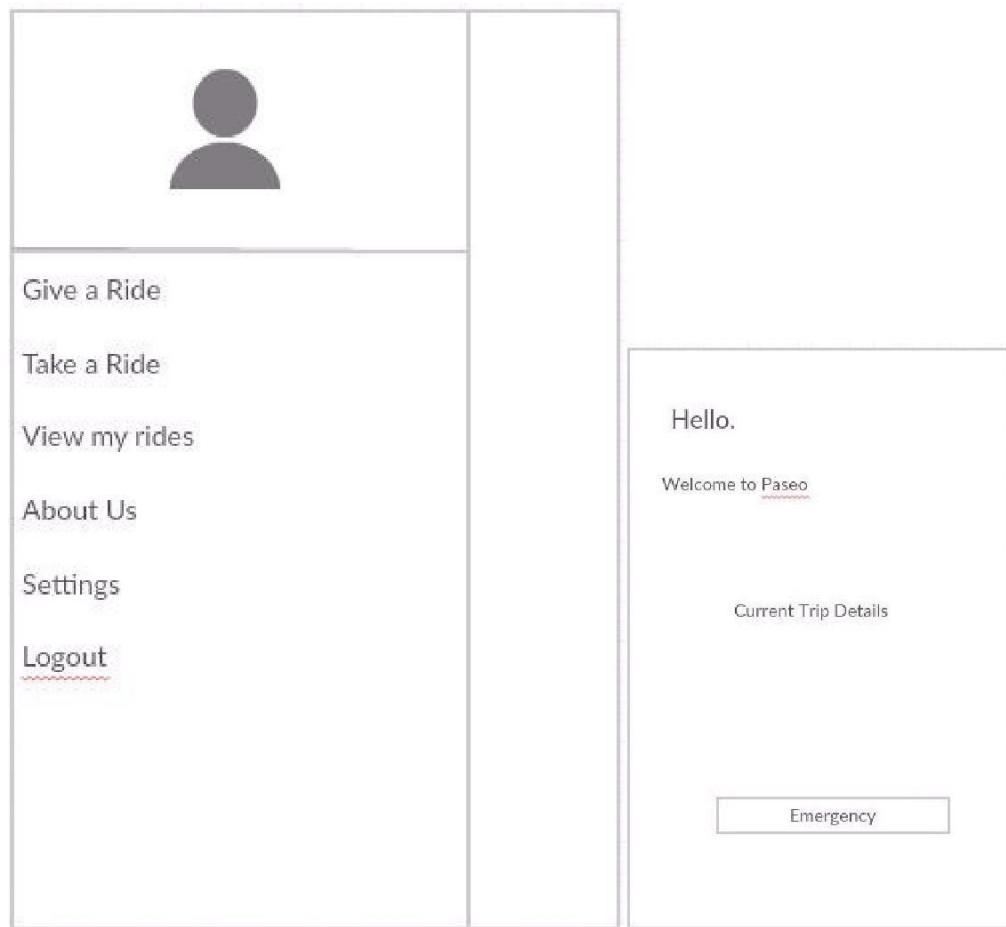
3. Google Maps API
4. Mongo DB API

DETAIL DESIGN OF FEATURES:**WIREFRAMES AND MOCKUPS**

2. The login Page is used by the users to login the application and New-Users can click register to create an account in the application. Users can also login with either Gmail or Facebook. A link About Us is provided to give user information about the app.



3. Registered user would be navigated to the Home screen as below where the user can perform multiple functions.



4. The Registration page is used by the New-Users to enter their details and save them to data base for creating the account.

The image shows a wireframe of a registration form. The title "Welcome" is at the top. The form consists of several input fields with validation asterisks: "First Name*", "Last Name*", "Email*", "User Name*", "Password*", "Confirm Password*", "Mobile Number*", and "Driving Licence" with a "Browse" button. Below these are two more sections: "Other proofs" with a "Browse" button and "Upload Pic" with a "Browse" button. The entire form is enclosed in a dark grey border.

4. Settings page is used by the user to update details like password, phone number and other details

The wireframe shows a vertical list of settings options:

- Change password
- Edit Phone Number
- Edit Email
- Other Details (button)
- Emergency Contacts
- Name _____ Number _____
- Name _____ Number _____
- Save (button)

- ② Take Ride and View existing Rides page would display the past rides of that user which allows the user to rate the rider.

The wireframe shows a page titled "Past History" under "Home". It displays a list of past rides with a rating field next to each:

1.	Rating
2.	Rating
3.	Rating
.	
.	
.	

At the bottom is a "SAVE" button.

6. Request ride page is used for the user to search/ create rides.

View Existing Rides

Submit Ride Details

From Location

To Location

Date

Date

Time

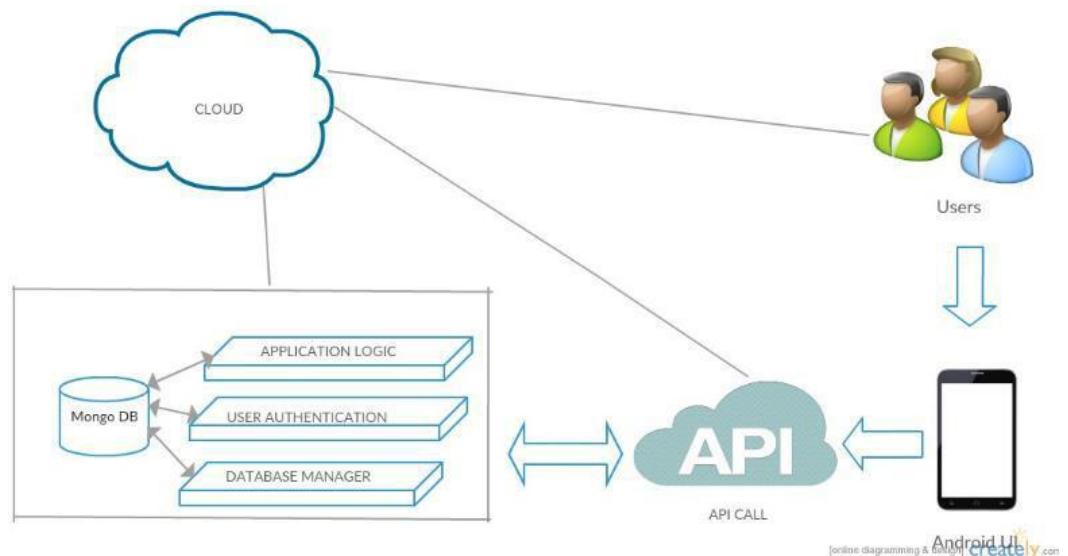
Time

Locate on Map

Submit

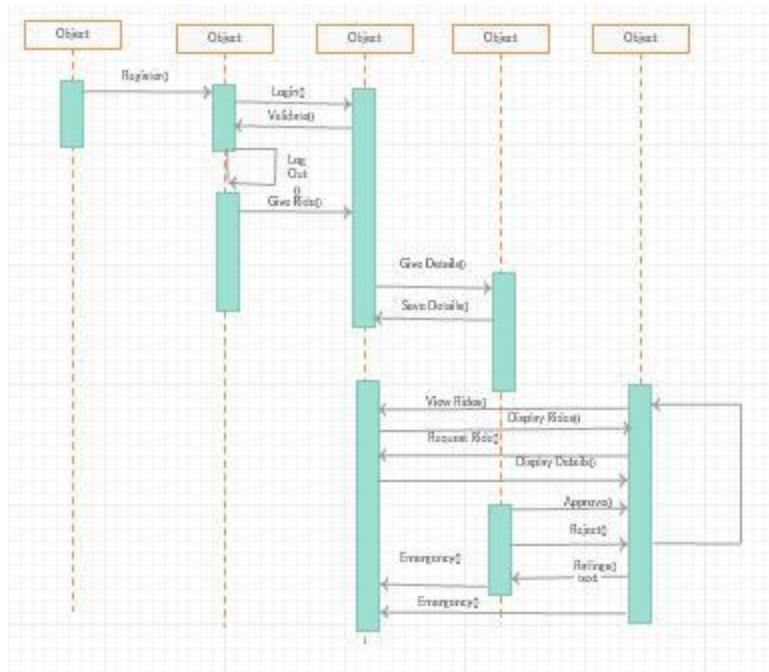


ARCHITECTURE DIAGRAM:

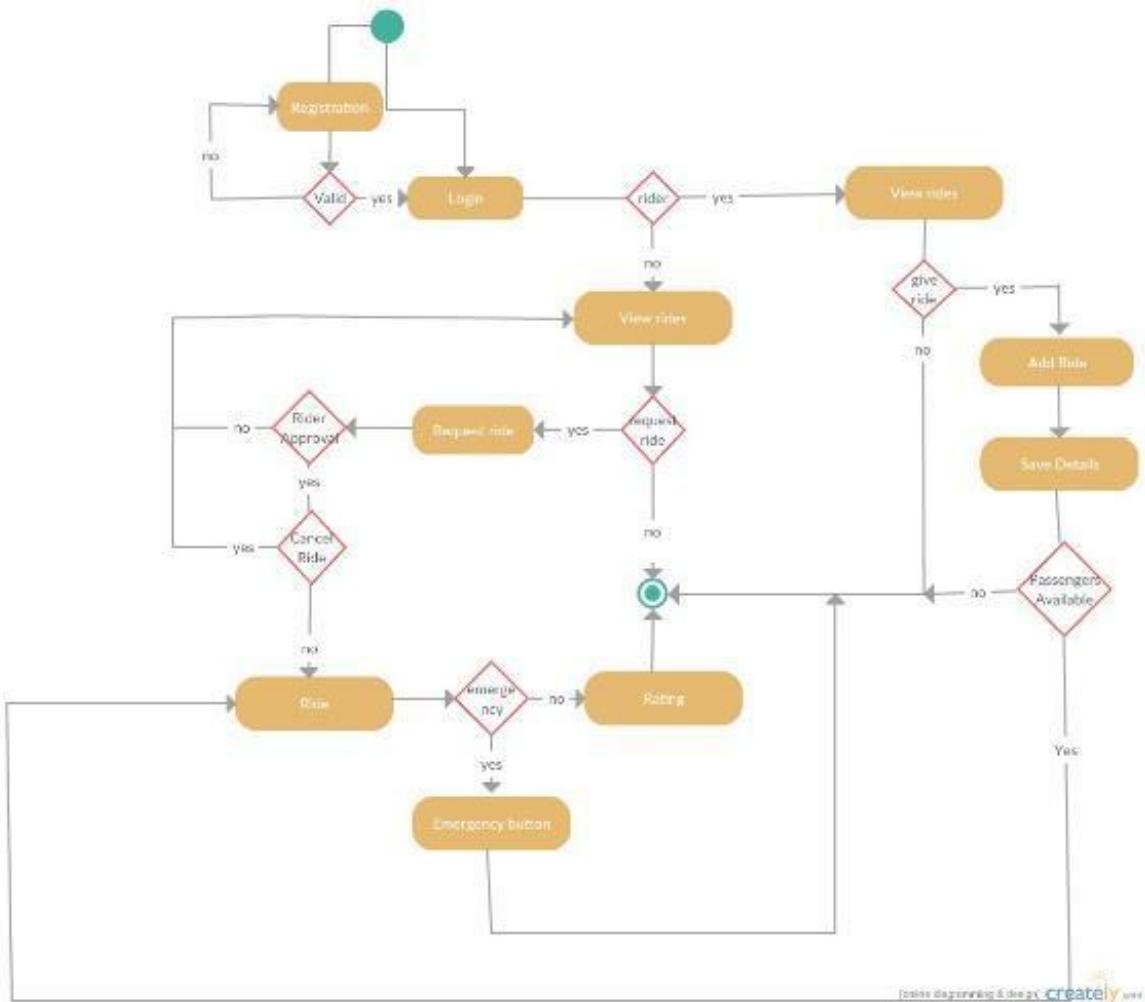


Android UI Create .com

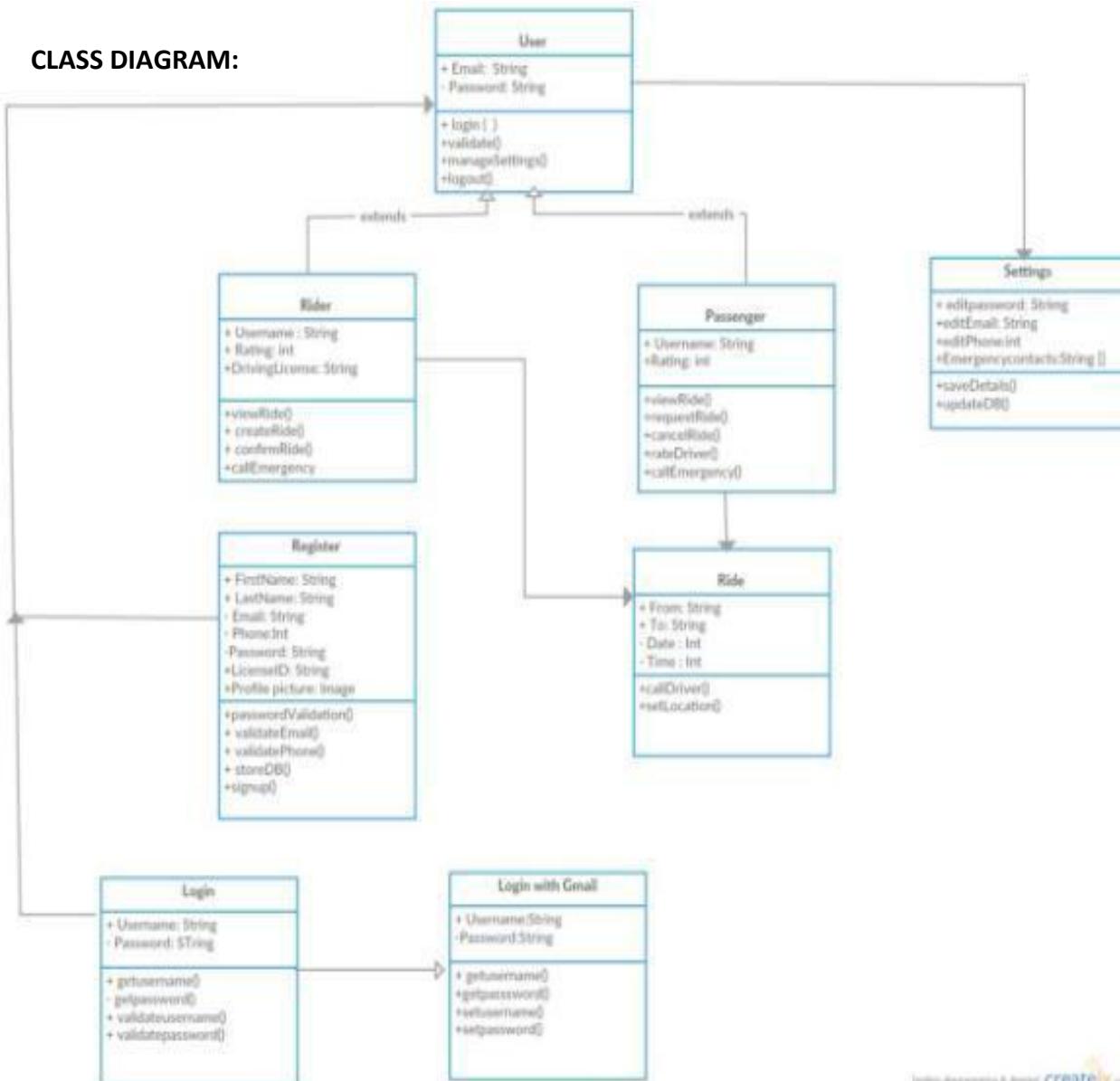
SEQUENCE DIAGRAM:



ACTIVITY DIAGRAM:



CLASS DIAGRAM:



USER STORIES:

1. US-1 (Login details of the Application)

User-Story Description: The user should be able to view the signing page enabling him to either login if an existing user or registering if new user. User can login with Gmail or Facebook.

Actor: Users

Requirement Specification: The screen should consists of two fields accepting the Email and password and two buttons login and New-user. Gmail and Facebook login buttons as well.

- Should have an input text field to accept Email
- Should have an input password field to accept password
- Button named Login, when clicked by the user he should be directed to US-2
- Button named New-user, when clicked by the user he should be directed to US-3
- The details of the user are authenticated, if he/she are existing users.
- Page should consists of Gmail and Facebook buttons enabling the user to login with existing accounts.

Assumptions: User should be able to perform the login and registration successfully.

2. US-2 (Home screen of the users)

User-Story Description: The user can see his active rides if he has currently booked any and he can further choose if he wants to give a ride or take a ride. He can logout at any time and can send a message on the hit of a button during emergency. User can also manage his settings.

Actor: Users

Requirement Specification: The screen should consist of fields displaying the users past rides, it should further have buttons to request rides, offer rides, past rides, manage settings, logout and an emergency button.

- Should have a side menu to display different buttons.
- Button named Manage Settings when clicked by the user should be directed to US-4.
- Button named Logout to come out of the application.
- Button named Request Ride when clicked by the user should be directed to US-5.
- Button named Offer Ride when clicked by the user should be directed to US-6.
- Button named Past History when clicked by the user should be directed to US-7.
- Button named Emergency to send a message to contacts during emergency.

Assumptions: User should be able to perform the login and use functionalities successfully.

3. US-3 (Registering the new-users)

User-story Description: Enabling the user to register into the application by taking the basic

details. Actors: Users

Requirement Specifications: As a New-user, he should be able to register with the application by filling the details of the user.

- Screen should consist of an input text field to accept the first name of the user
- Should consist of an input text field to accept the last name of the user.
- Should consists of an email field to accept the email.
- Should consist if an input text filed to accept username.
- Should consist of an input filed to accept mobile number.
- Should have a set password field enabling the user to set password
- Should have a confirm password field to verify the password.
- Should have a button to upload other proofs.
- Should have button to upload the user's driver's license.
- Should have a button to upload the users profile photo.
- Button to submit the details to database.

The following fields should be mandatory:

First name, email, Password, passport, mobile number, username

Assumptions: New-users should be able to create an account in the application.

4. US-4 (Update users profile)

User-Story Description: The user should be able to change his password, mobile number, email and he can also add the contact details of his family to send them a message during emergency.

Actor: Users

Requirement Specification: The screen should consists of fields accepting the Email, password, mobile number, other info and two buttons login and New-user.

- Should have an input text field to enter new password.
- Should have an input field to enter new phone number.
- Should have an input field to enter new email.
- Should have an input filed to update other info.
- Should have 3 input fields to enter name and mobile number.
- Should have a button save to successfully update the details to the database.
- Should have a button home to enable the user to move to home page.

Assumptions: User should be able to perform the update and save details successfully.

5. US-5 (Request ride)

User-story Description: The user should be able to request ride between two places. **Actor:** Users

Requirement Specifications: The screen should consist of the following fields and buttons

- Should have an input text field to enter the starting point.
- Should have an input filed to enter the destination.
- Should have an input field to enter the time.
- Should an input filed to enter the date of journey.
- Should have a button Request Ride to request a ride to the rider.
- Should have a link which on click should display the existing rides available between the two locations.
- Should have a button Home to redirect to the home page.

Assumptions: User should be able to request the ride successfully.

6. US-7 (View existing rides)

User-story Description: The user should be able to view his past rides and give ratings to rides.

Actor: Users

Requirement Specifications: The screen should consist of the following fields and buttons

- Should have an input text box to display the users past rides.
- Should have a field to rate the driver of his previous ride.
- Should have a Save button to save the updated rating.
- Should have a home to populate user to home page.

Assumptions: User should be able to view all his rides.

7. US-6 (Offer Ride)

User-story Description: The user should be able to offer ride details and view the active rides available for him

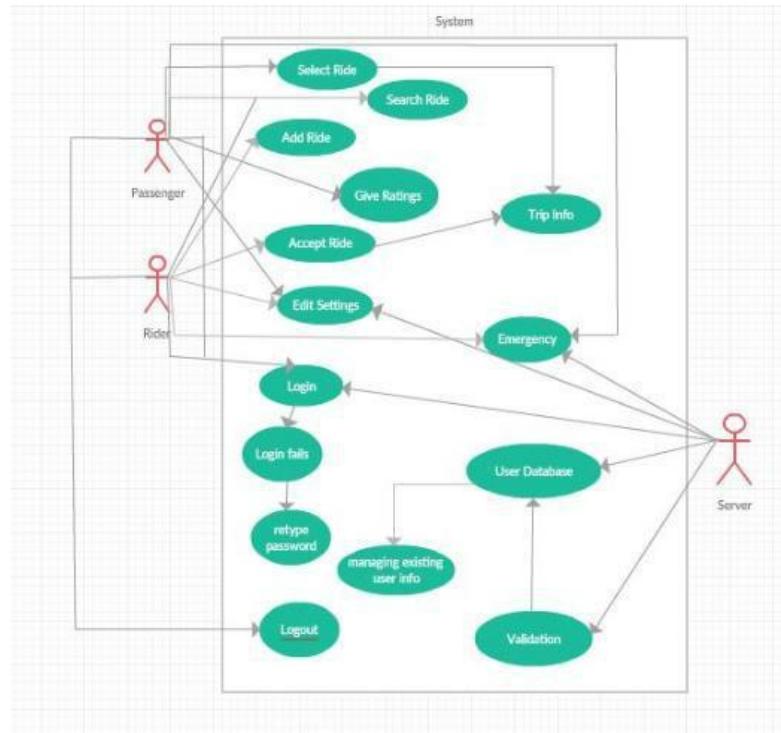
Actors: Users

Requirement Specifications: The screen should consists of the following inputs.

3. Should have an input text field to enter the starting point.
4. Should have an input filed to enter the destination.
5. Should have an input field to enter the time.
6. Should an input filed to enter the date of journey.
7. Should have a button Offer Ride to request a ride to the rider.
8. Should have a link which on click should display the existing rides available between the two locations.
9. Should have a button Home to redirect to the home page.

Assumptions: User should be able to request the ride successfully.

USE CASE DIAGRAM:



SERVICE DESCRIPTION:

- ❑ Google API is used to locate the present location of the user and also enabling the New-users to login with Gmail.
- ❑ Used Mongo DB to store the user details and to retrieve the user details when needed.

TESTING DOCUMENTS:

1. Login details of the Application

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to login/New-user should be able to register	The user would be able to enter the E-mail and password and click login to enter to Application	User should be able to login	User is logged-in.
2.	User should be able to login/New-user should be able to register	New-User should click the Sing-up button to register, to navigate to registration page	New-users are directed to register page.	New-Users are able to view the registration page.
3.	User should be able to login/New-user should be able to register	User should enter E-Mail and Password and are validated for the users.	User details are authenticated.	User is able to login only when accurate details are submitted

4.	User should be able to login/New-user should be able to register	User is entered wrong details an error would pop up.	User credential are to be validated and if wrong a pop-up would be display	User would be able to view a message if incorrect details are entered.
5.	User should be able to login/New-user should be able to register	User would be able to sign-in with Gmail or Facebook.	User should be successfully logged in.	New-users are able to login.
6.	User should be able to login/New-user should be able to register	User can read about the application following that link	User can view the information about the application.	User is able to view the application related data.

2. Registration screen of the Users

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	New-User should be able to register by giving the basic details.	New-Users should fill all the fields and click submit.	New-users should be successfully register.	New-Users are registered.
2.	New-User should be able to register by giving the basic details.	New-User should enter all the mandatory fields. First name, email, password, passport	New-users are registered once the user enters the mandatory fields	New-Users are able register.
3.	New-User should be able to register by giving the basic details.	New-Users would be notified with validation regarding the mandatory fields.	New-Users are able to view an error if fields are left blank.	New-users are notified with error.
4.	New-User should be able to register by giving the basic details.	On submit the details entered are saved to the Data Base.	New-User details are save to the data base once they click the submit button.	Details of new users are save to data base.

3. Ride

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	Registered User should be able to either request a ride or give in ride details.	Users can fill in the ride details and submit their details.	Users should be able to fill the ride details to provide a ride	Users can successfully provide their ride details.
2.	Registered User should be able to either request a ride or give in ride details.	Users can click request ride to view available rides	Users can view available rides	Users are able to view available rides
3.	Registered User should be able to either request a	User can click offer to create a ride	Users should be able to create rides	Users are able to create the rides

	ride or give in ride details.			
4.	Registered User should be able to either request a ride or give in ride details.	Users can click on logout to come out of the application.	Users are redirected to the login page if they select the logout.	Users are logged out successfully.

4. Home screen

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to choose different services	Users should click request ride to redirect to ride page.	Users should be able to redirect to the ride page	Users are successfully redirected.
2.	User should be able to choose different services	Users should click offer ride to redirect to the ride page	Users should be able to redirect to the ride page	Users are successfully redirected.
3.	User should be able to choose different services	Users should click past rides to redirect to the previous rides page	Users should be redirected to the past rides page	Users are successfully redirected to the page.
4.	User should be able to choose different services	Users can click on logout to come out of the application.	Users are redirected to the login page if they select the logout.	Users are logged out successfully
5.	User should be able to choose different services	Users can click manage settings button to redirect to manage settings page	Users should be able to redirect to the page	User are successfully redirected

5. Manage settings screen

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to update his details	Users should enter any field that he wants to be update	Users should be able to update the information	Users have successfully updated the information.
2.	User should be able to update his details	Users should be able to redirect to home page	Users should be able to redirect to the ride page	Users are successfully redirected.
3.	User should be able to update his details	User can enter the emergency contacts and click save	Users should be able to redirect to home page	Users are successfully redirected to home page

6 .Past rides

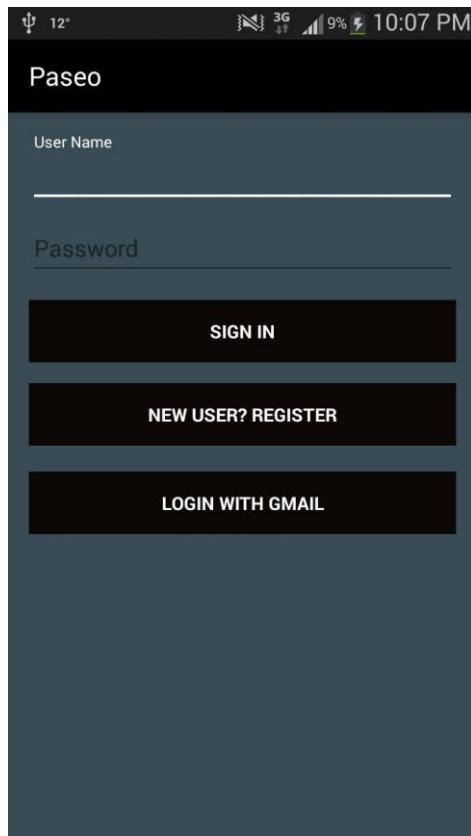
S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to rate their previous rides	Users should rate their previous ride and click save.	Users should be able to redirect to the home page	Users are successfully redirected.

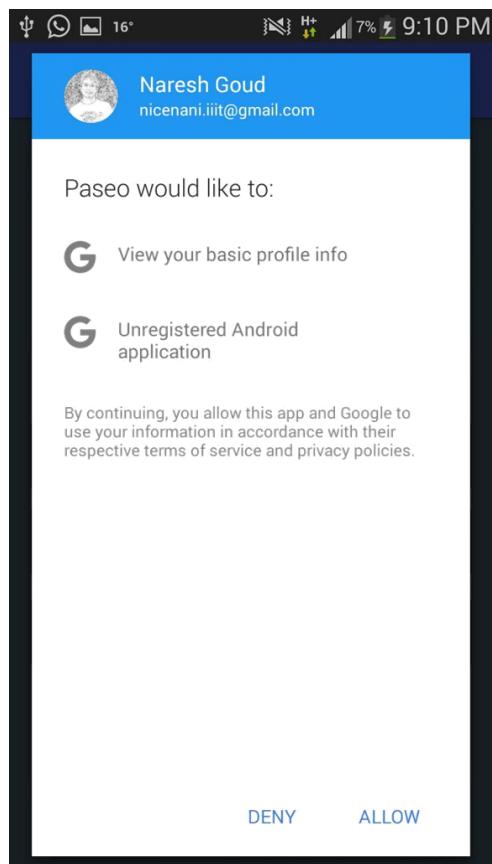
IMPLEMENTATION:

- ② Mobile client Implementation: Implementation in mobile is feasible and is easy to use as most of the present world runs on smart phones. We have implemented the project using Android.
- ② Server Implementations: Used Mongo database to store and retrieve the details of the user about his ride and profile info.

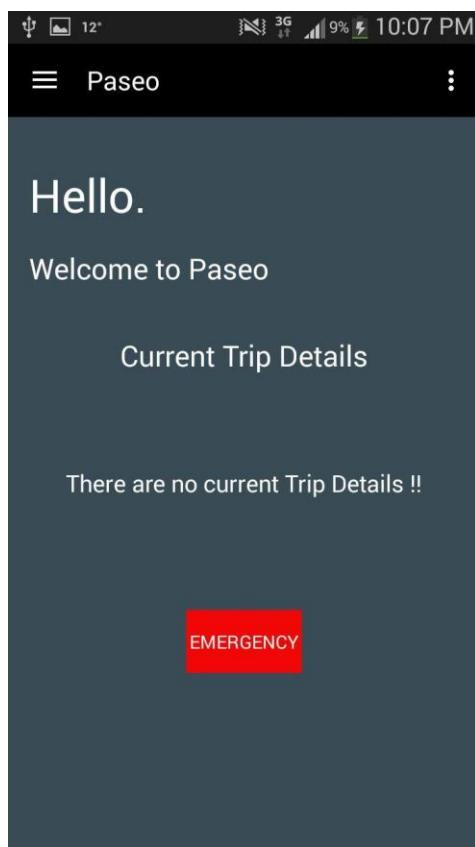
DEPLOYMENT:

1. Login Page:

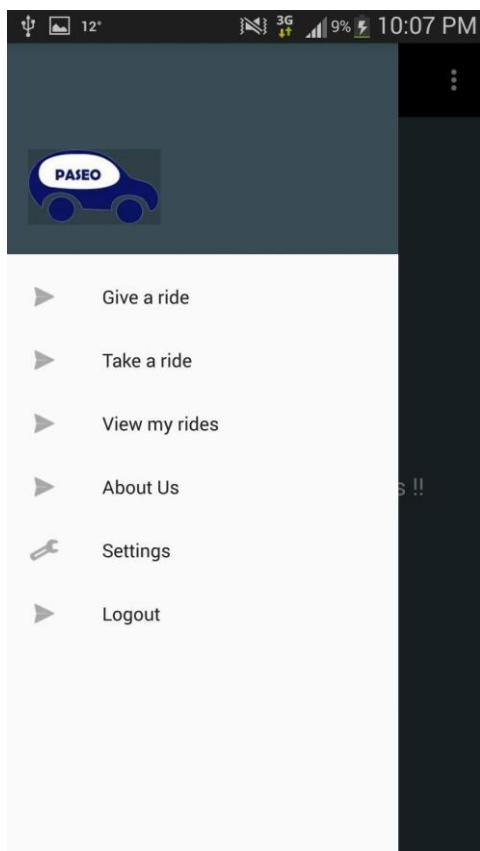




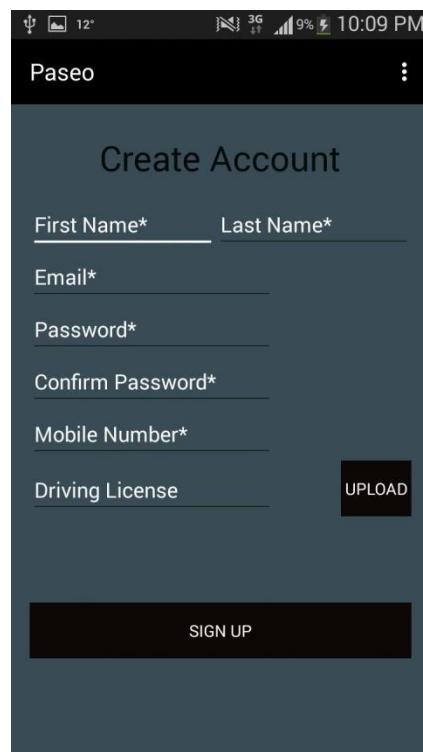
2. Home Page:



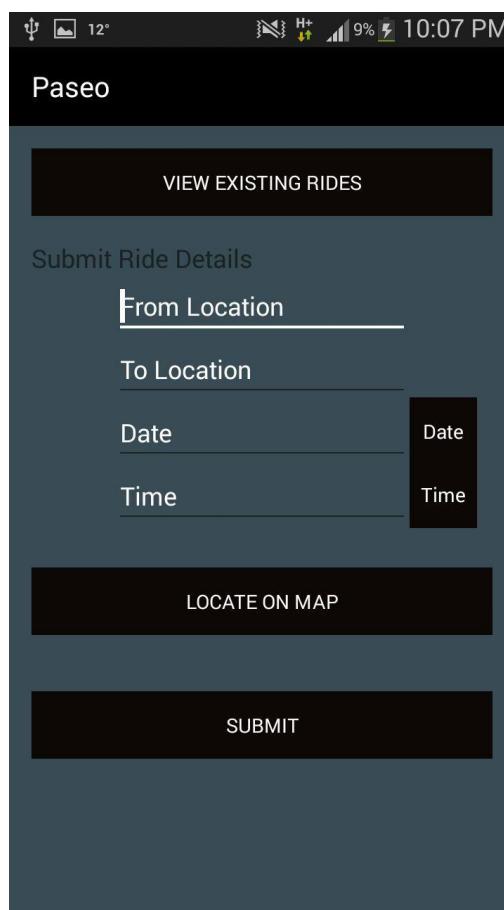
Menu Screen:

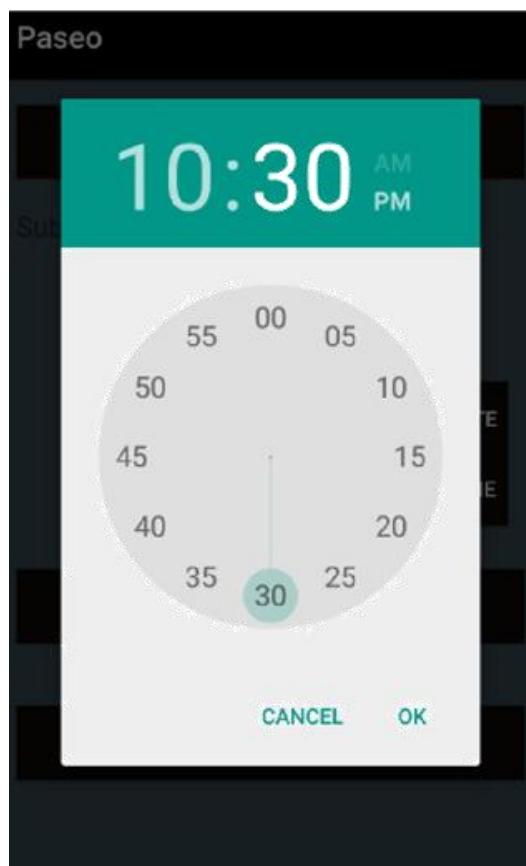


3. Registration Page:



4. Ride Request:

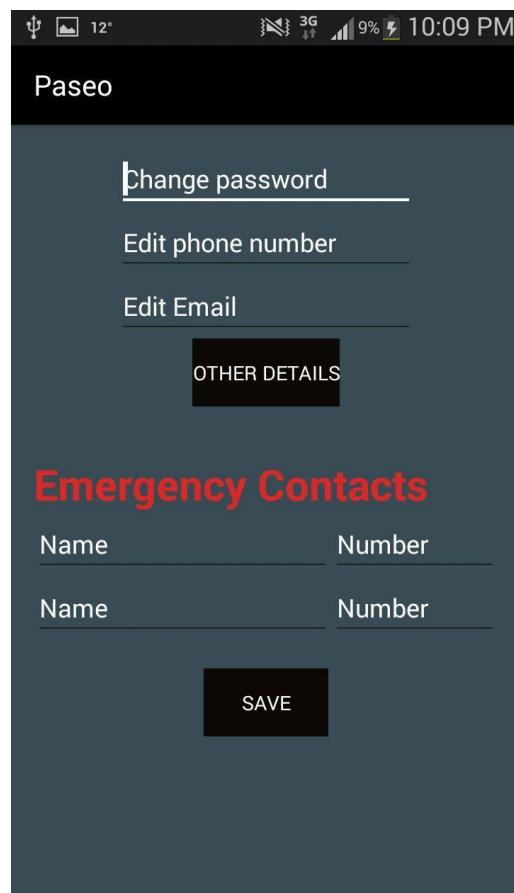




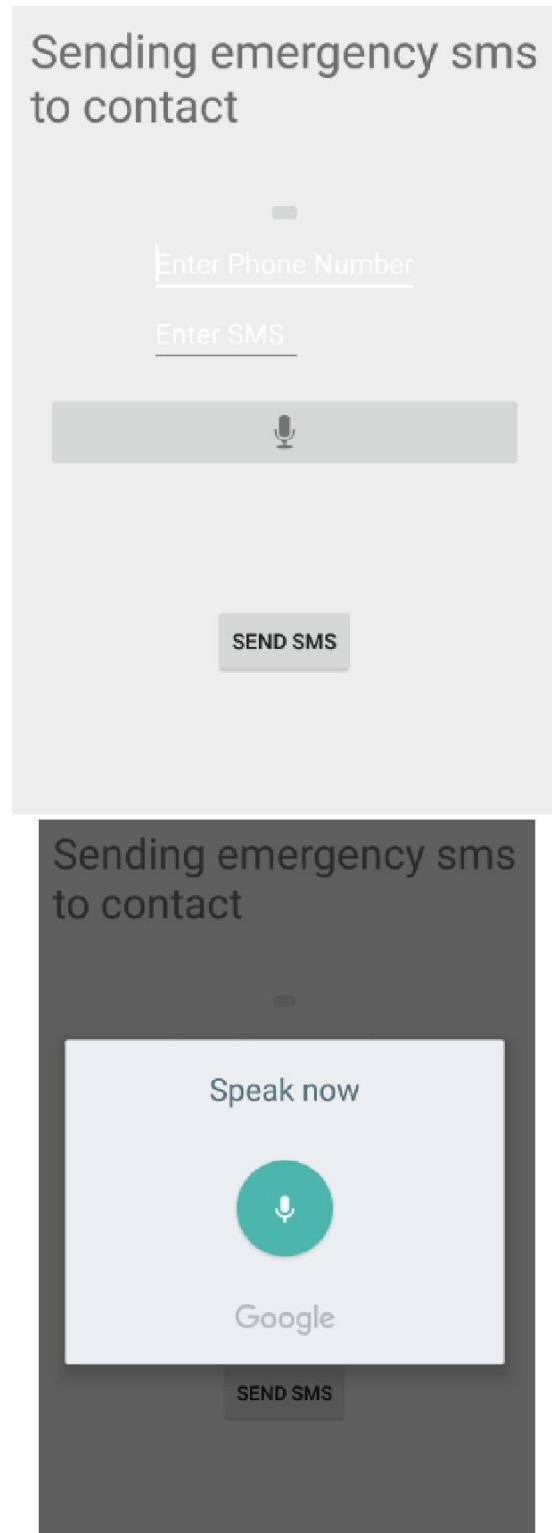
5. Offer Ride:



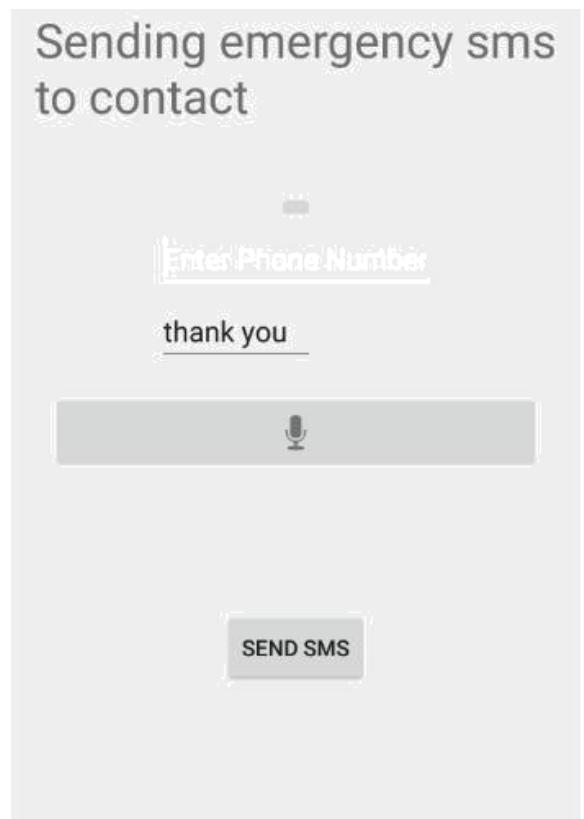
6. Settings



7. Security features:



Sending emergency sms to contact



PROJECT MANAGEMENT:

4. **Project Timeline:** 6th April 2016.
5. **Members:** Amulya1, VinuthaMuthyala, Naresh, Megha Sai.
6. **Task Responsibilities:** Issues are created in github for each members and every member have collectively worked to complete them in given time. Have shared the work equally among us.

Work

Completed:

Description:

- **Stories:** UI is improved, security alert notifications, emergency contacts, voice to text and the feedback rating is implemented.
- **Service Design:** The Google service is implemented to notify the user with the current location and Mongo database service to save and retrieve the information of the user and ride, they are implemented to enhance the functionality of the user.
- **Service Implementation:** Google API and Mongo database and are being worked successfully.

Responsibilities:

We all have worked together to provide an effective and clear implementation of the project.

Amulya Pindi: Sign-in with Gmail, sending SMS and notifying the user which are a part of emergency, giving feedback to the riders, UI of the settings page, mandatory fields in application and documentation.

Vinutha Muthyalu: UI part of the home page, Voice to text conversion as a part of emergency, Functionality of retrieving the data from Mongo database, Google Map services and driver's document validation.

Megha Sai: Architecture/use case diagrams, UI, design of request screens, wireframes and manage setting features.

Naresh: To store the data to Mongo database, UI, Activity diagram and manage settings features.

Time Taken and Contributions: All have contributed 5hrs each almost every alternate day and almost 10hrs completely the last three days.

Work to be completed:

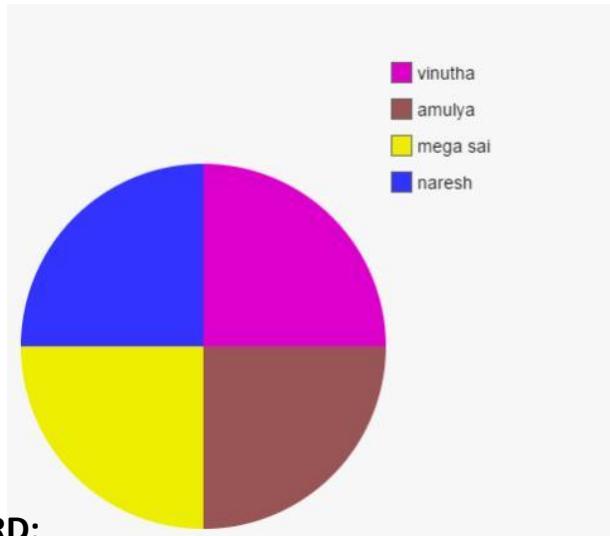
Description:

Payment features, location alarm implementation, directions in google maps, enhancements in security features.

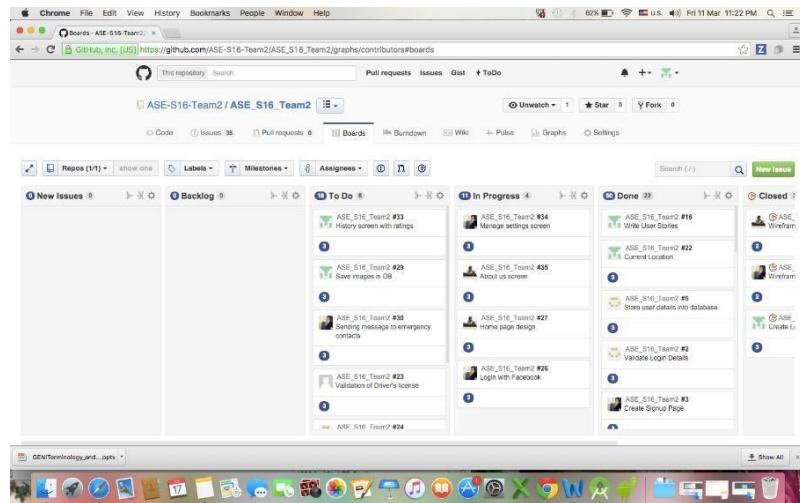
Issues:

Experienced few issues in implementing the notifications in security, route implementation in google maps and about the payment feature implementation.

Burndown Graphs:



DASH BOARD:



BIBLIOGRAPHY:

1. <https://en.wikipedia.org/wiki/Carpool>
2. https://en.wikipedia.org/wiki/Real-time_ridesharing
3. <http://www.carpoolworld.com/>
4. http://www.ijarcsse.com/docs/papers/Volume_3/4_April2013/V3I3-0385.pdf

CS551 Advanced Software Engineering

Project Increment-4

PROJECT NAME: PASEO

TEAM-2

Amulya Pindi (46)
Vinutha Muthyala (36)
Naresh Pogakula (47)
Megha Sai Reddy Bodimani(6)

INTRODUCTION:

Carpooling is a sharing of car journeys such that more than people travel together in a car towards the same destination. Carpooling has proved to be a major triumph in reducing the emissions of greenhouse gases as it reduces the number of cars on road, further it has also proved to be a blessing to those who do not own a private car.

The application “PASEO” is a carpooling android application with security measures to ensure that the user feels secured all times. We register the rider and the passenger and validate the details of the users by checking their Driver’s license, car details etc. The rider gives the timings of his ride and the passenger picks up the ride based on his convenience. We further ensure full security to the users by providing unique features like sharing current location, tracking the ride in case of emergency providing them a button to give alerts based on location and time.

OBJECTIVES:

There are a lot of impediments faced by people who travel to different parts of the city in their own transport which kind of makes them irksome and there are also people who do not have their own transport and pay a bomb to commute in the city. The amount of environmental pollution that is caused by single driver cars is phenomenal and is on the rise. On an average American spends 40 hours each year stuck in traffic which further adds to our woes.

According to a survey by carpooling just twice a week about 1,600 pounds of greenhouse gases can be kept out of the air each year. Further carpooling cuts down the cost of both the rider and the passenger, makes their life and also for the generations to come.

FEATURES:

REGISTER AND LOGIN

User can register themselves as a rider or a passenger through the application by giving their details.

VALIDATION

The application validates certain details of the rider by checking his history, checks if he has a valid license and checks certain details of the car.

ANONYMITY

The application ensures possible anonymity of both the rider and the passenger as only their names will be revealed. They can contact each other through the app to ensure certain safety.

RIDE TRACKING

The user can track his ride through the app and can also share his location on the click of a single button to his desired friends who can also track the users ride to ensure to safety.

EMERGENCY ALERT

The user can just pop out an emergency signal to the police and his family in case of any accident or in danger on the click of a button or on the utterance of a certain keyword.

SAFETY METER

The user can see how safe an area is during different times of the day in which they are

travelling, so they can be alert at each and every point. The application pops out a warning or alert on case if the area is highly unsafe.

EXISTING API's:

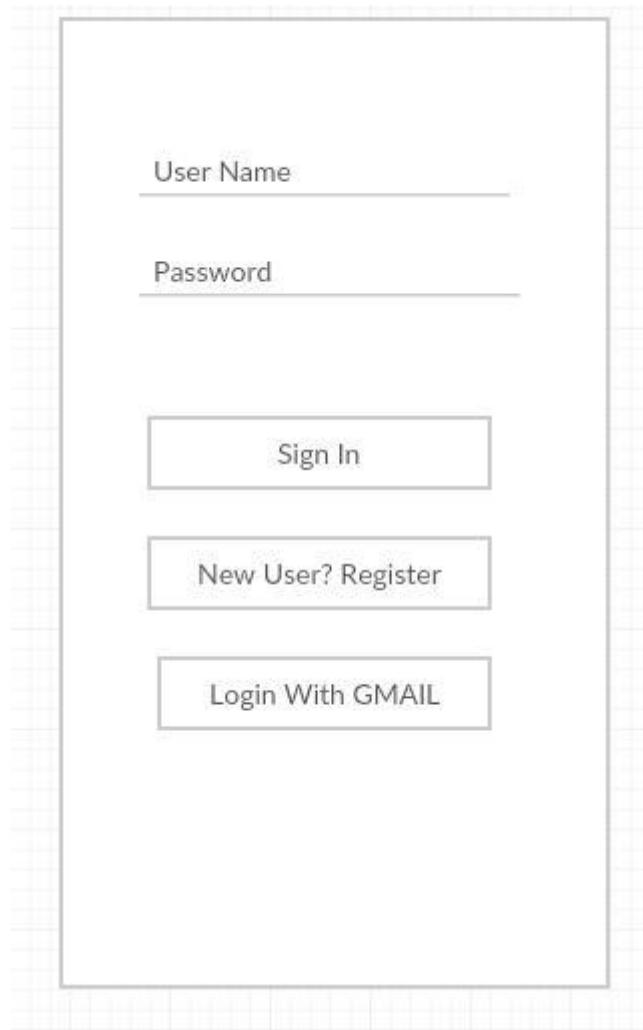
Google Maps API

Mongo DB API

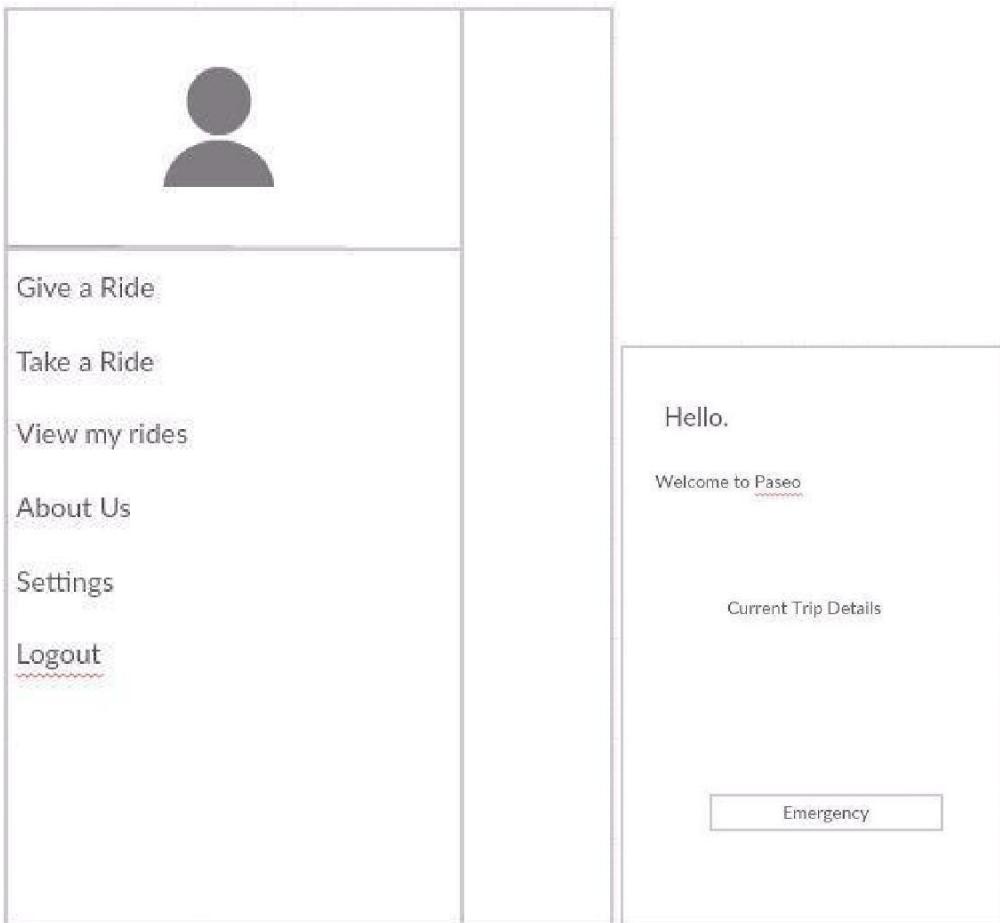
DETAIL DESIGN OF FEATURES:

WIREFRAMES AND MOCKUPS

- The login Page is used by the users to login the application and New-Users can click register to create an account in the application. Users can also login with Gmail.



- Registered user would be navigated to the Home screen as below where the user can perform multiple functions.



3. User can register from the following page.

Create Account

Profile Pic

First Name Last Name

Email

Password

Confirm Password

Mobile Number

Driving License No.

Sign UP

- Settings page is used by the user to update details like password, phone number and other details

The wireframe shows a vertical list of options:

- Change password
- Edit Phone Number
- Edit Email
- Other Details** (button)
- Emergency Contacts

Under Emergency Contacts, there are two rows for entering names and numbers:

Name	Number
Name	Number

A large "Save" button is located at the bottom.

- Take Ride and View existing Rides page would display the past rides of that user which allows the user to rate the rider.

The wireframe shows a list of past rides, each represented by a card:

From Location	To Location	Date	Time	Call
				+
From Location	To Location	Date	Time	Call
				+

The cards are repeated twice. A horizontal bar is at the bottom.

6. Give a ride page is used for the user to search/ create rides.

View Existing Rides

Submit Ride Details

From Location

To Location

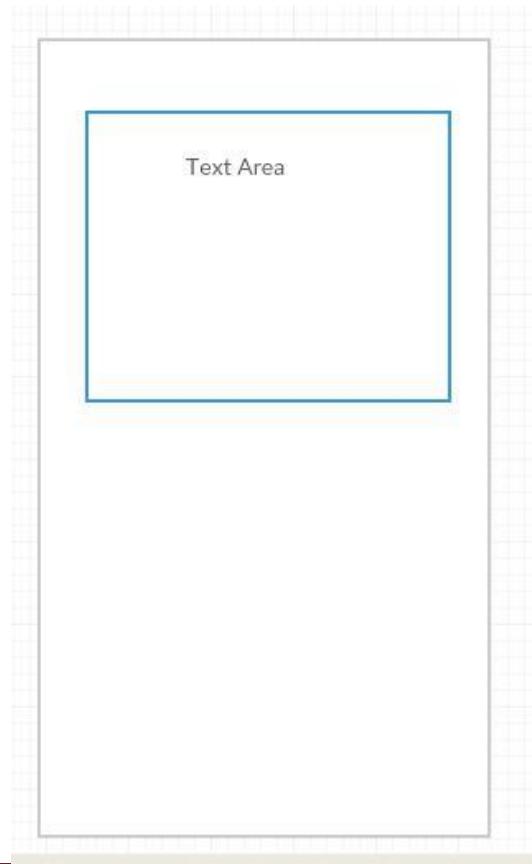
Date Time

Time

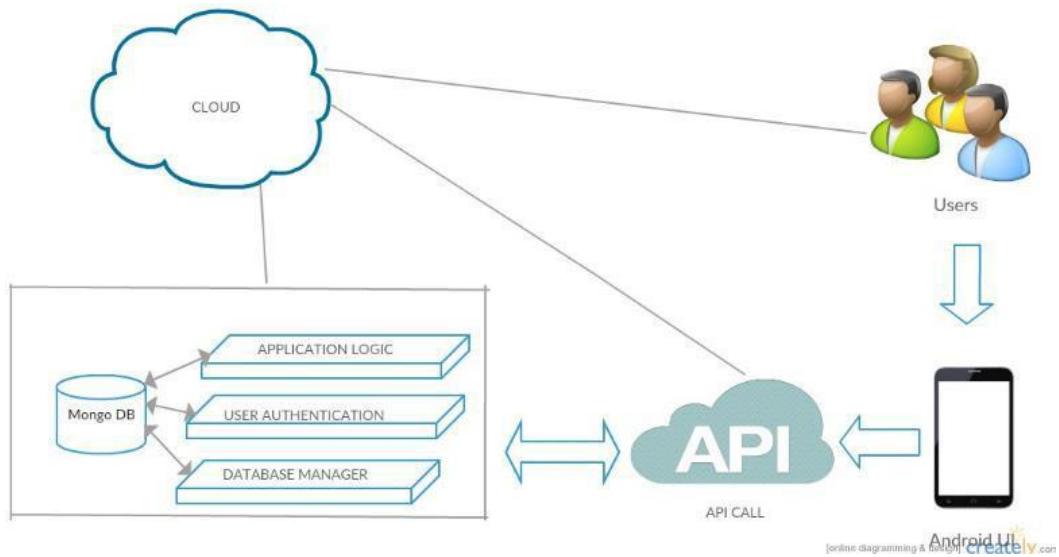
Locate on Map

Submit

7. About us page to acknowledge user about application.

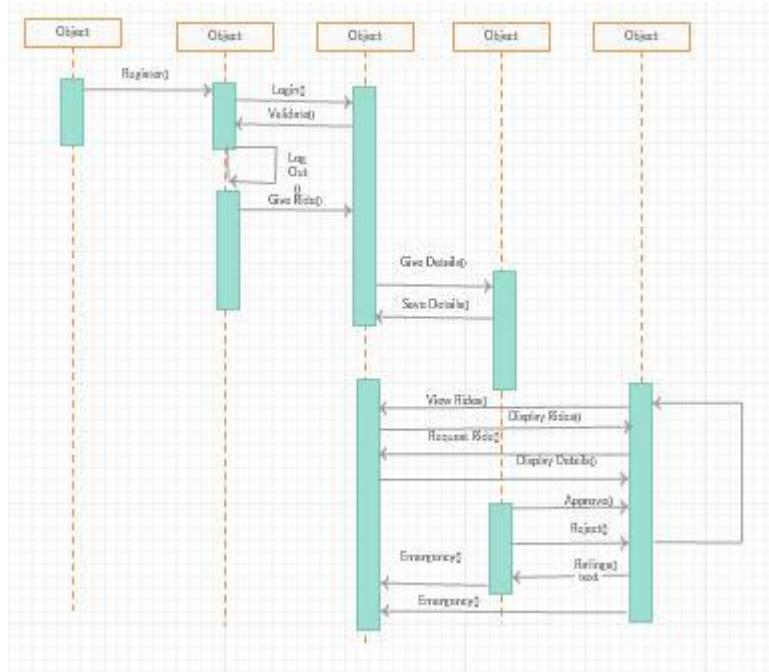


ARCHITECTURE DIAGRAM:

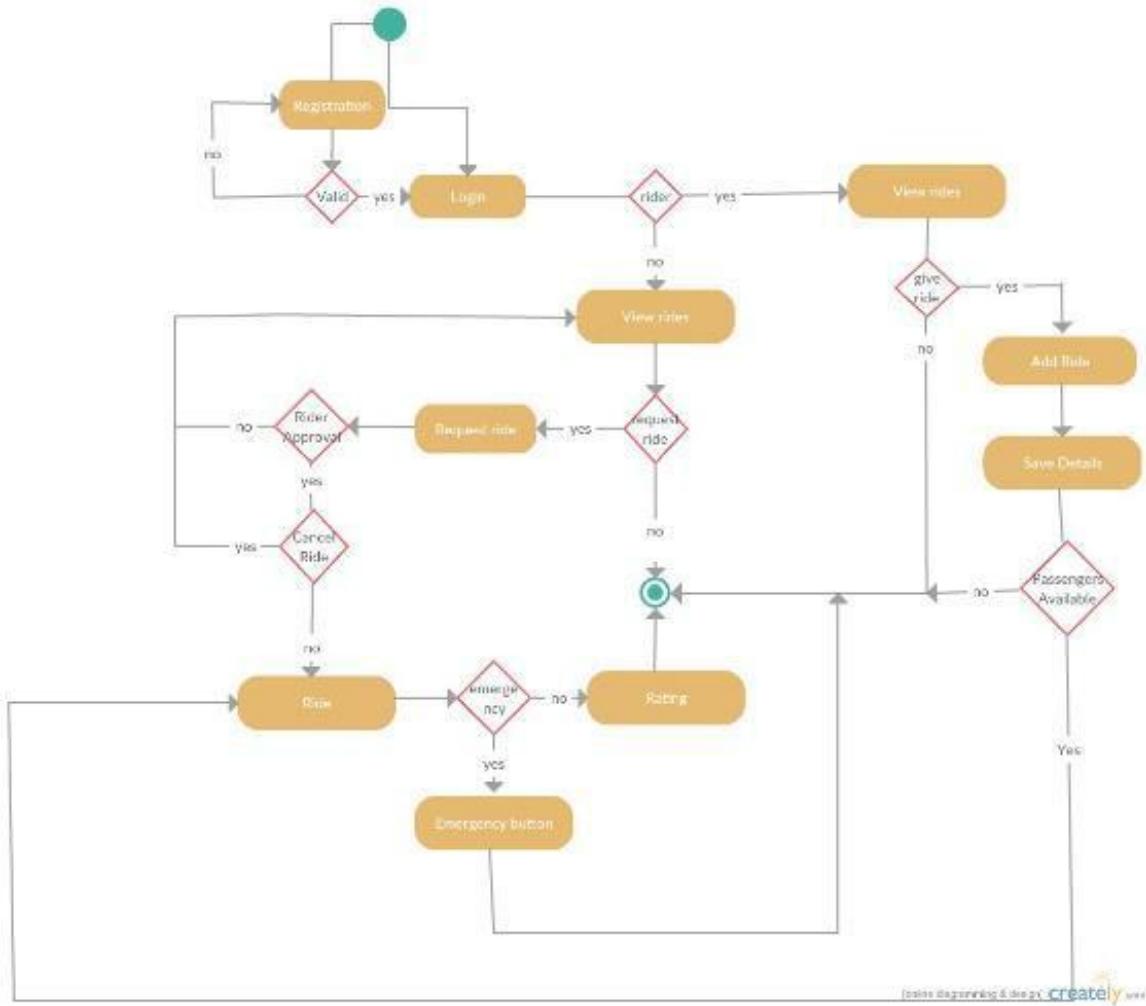


Android UI & CreateIV.com

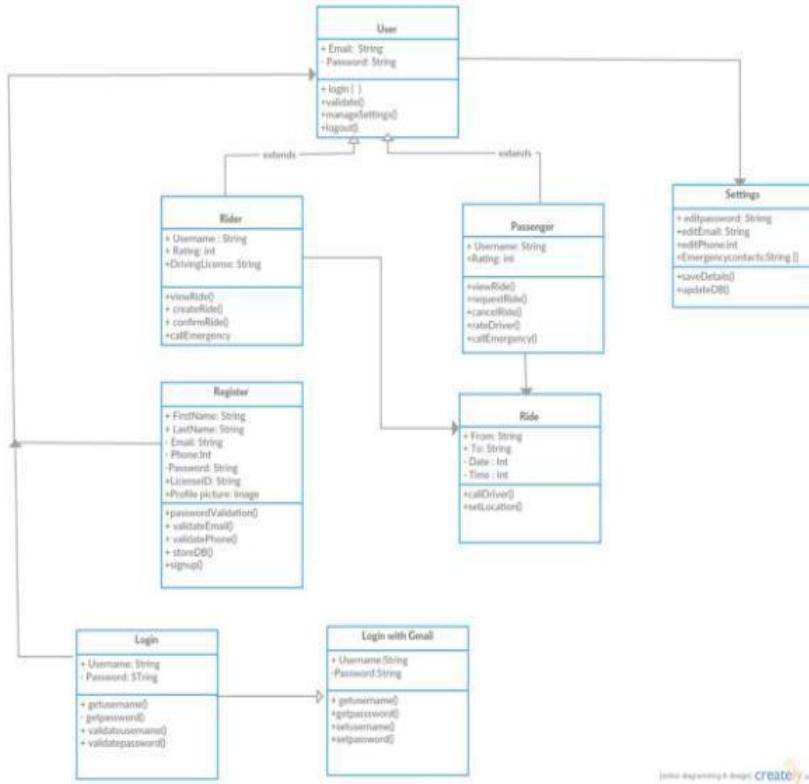
SEQUENCE DIAGRAM:



ACTIVITY DIAGRAM:



CLASS DIAGRAM:



Join us on [Engineering & Design](#) at [CreateMy.com](#)

USER STORIES:

1. US-1 (Login details of the Application)

User-Story Description: The user should be able to view the signing page enabling him to either login if an existing user or registering if new user. User can login with Gmail.

Actor: Users

Requirement Specification: The screen should consists of two fields accepting the Email and password and two buttons login and New-user. Gmail login button as well.

Should have an input text field to accept Email stated as user Name

Should have an input password field to accept password

Button named Sign IN, when clicked by the user he should be directed to US-2 Button named New-user, when clicked by the user he should be directed to US-3 The details of the user are authenticated, if he/she are existing users.

Page should consists of Gmail button enabling the user to login with existing accounts.

Assumptions: User should be able to perform the login and registration successfully.

2. US-2 (Home screen of the users)

User-Story Description: The user can see his active rides if he has currently booked any and he can further choose if he wants to give a ride or take a ride from side menu. He can logout at any time and can send a message on the hit of a button during emergency. User can also manage his settings.

Actor: Users

Requirement Specification: The screen should consist of fields displaying the users current rides, it should further have buttons to request rides, offer rides, past rides, manage settings, logout in side menu and an emergency button. Rating stars should be present.

Should have a side menu to display different features.

Side menu should have Settings when clicked by the user should be directed to US-4.

Side menu should have take a ride when clicked by the user should be directed to US-5.

Side menu should have give a ride button when clicked by the user it should be directed to US-6.

Side menu should have View my rides and log out when clicked by the user should be directed to US-7.

Button named Emergency to send a message to contacts during emergency. Side menu should have a About Us to display details of the application.

Rating stars to provide user to give rating for his/her rides taken.

Logout to allow the user to come out of the application.

Assumptions: User should be able to perform the login and use functionalities successfully.

US-3 (Registering the new-users)

User-story Description: Enabling the user to register into the application by taking the basic details.

Actors: Users

Requirement Specifications: As a New-user, he should be able to register with the application by filling the details of the user.

Screen should consist of an input text field to accept the first name of the user

Should consist of an input text field to accept the last name of the user.

Should consists of an email field to accept the email.

Should consist of an input filed to accept mobile number.

Should have a set password field enabling the user to set password. Should have a confirm password field to verify the password.

Should have input field to enter user's driver's license number. Should have a button to upload the users profile photo.

Button to submit the details to database.

The following fields should be mandatory:

First name, Last name, email, Password, mobile number

Assumptions: New-users should be able to create an account in the application.

US-4 (Update user's profile)

User-Story Description: The user should be able to change his password, mobile number, email and he can also add the contact details of his family to send them a message during emergency.

Actor: Users

Requirement Specification: The screen should consists of fields accepting the Email, password, mobile number, other info and two buttons login and New-user.

Should have an input text field to enter new password.

Should have an input field to enter new phone number.

Should have an input field to enter new email.

Should have an input filed to update other info.

Should have 3 input fields to enter name and mobile number.

Should have a button save to successfully update the details to the database.

Assumptions: User should be able to perform the update and save details successfully.

US-5 (Take a ride)

User-story Description: The user should be able to request ride between two places. Actor: Users

Actor: Users

Requirement Specifications: The screen should consist of the following fields and buttons

Screen should display the existing rides available between the two locations.
Passenger should be given a functionality to call the rider to confirm the ride.

Assumptions: User should be able to request the ride successfully.

US-7 (View existing rides)

User-story Description: The user should be able to view all existing rides.

Actor: Users

Requirement Specifications: The screen should consist of the following fields and buttons
User should be able to view all the existing rides.

Assumptions: User should be able to view all his rides.

US-6 (Give a Ride)

User-story Description: The user should be able to offer ride details and view the active rides available for him

Actors: Users

Requirement Specifications: The screen should consists of the following inputs.

- Should have an input text field to enter the starting point.
- Should have an input filed to enter the destination.
- Should have an input field to enter the time.
- Should an input filed to enter the date of journey.
- Should have a button submit to save a ride to the rider. Should have locate on map button.

Assumptions: User should be able to request the ride successfully.

8. US-8 (Emergency details)

User-story Description: The user should be able to send emergency information to his three contacts.

Actor: Users

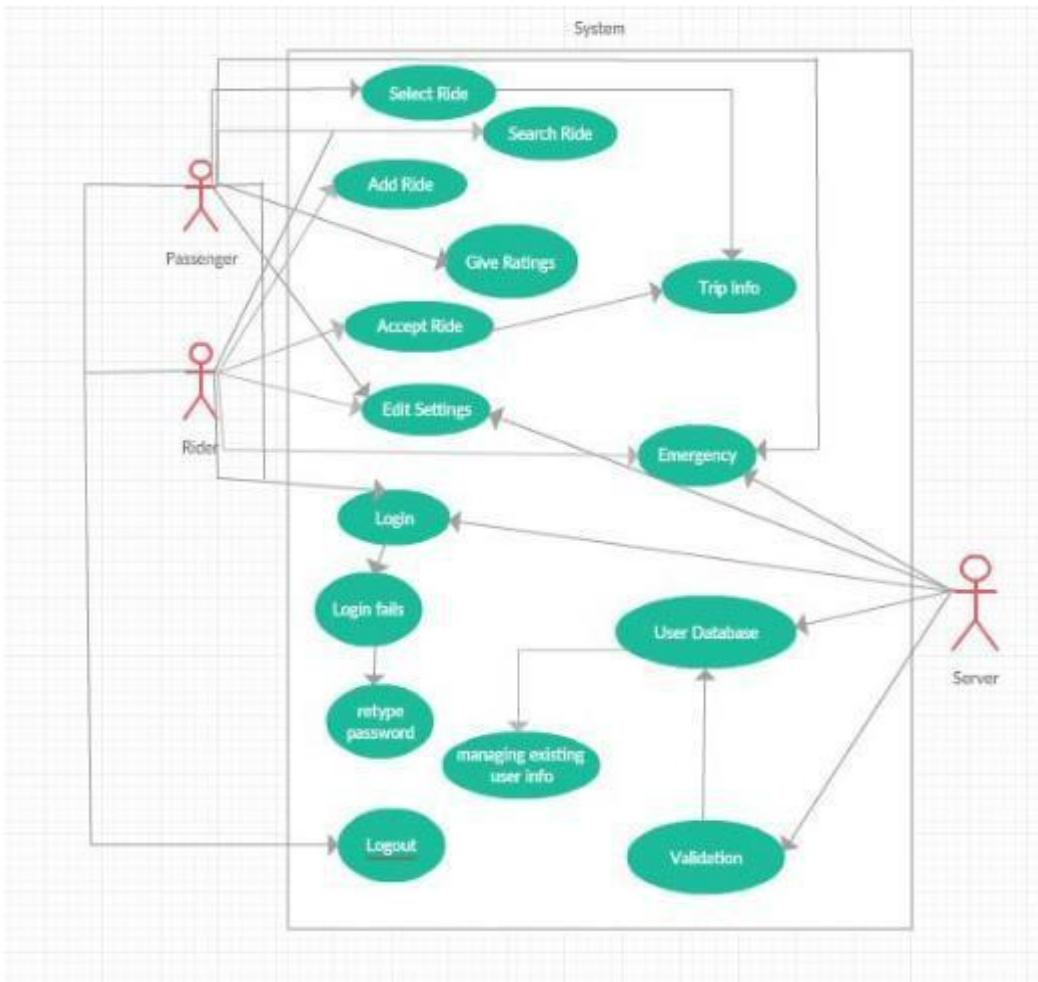
Requirement Specifications: The screen should have the below functions on clicking the emergency button.

SMS should be sent to the emergency contacts with default message.

User should be allowed to send a voice message to his emergency contacts.

Assumptions: User should be able to successfully send the necessary details to the emergency contacts.

USE CASE DIAGRAM:



SERVICE DESCRIPTION:

- ② Google API is used to locate the present location of the user, to get directions, to use the place holder and also enabling the New-users to login with Gmail.
- ② Used Mongo DB to store the user details and to retrieve the user details when needed.

TESTING DOCUMENTS:

Login details of the Application:

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to login/New-user should be able to register	The user would be able to enter the E-mail and password and click login to	User should be able to login	User is logged-in.

		enter to application		
2.	User should be able to login/New-user should be able to register	New-User should click the Sing-up button to register, to navigate to registration page	New-users are directed to register page.	New-Users are able to view the registration page.

3.	User should be able to login/New-user should be able to register	User should enter E-Mail and Password and are validated for the users.	User details are authenticated.	User is able to login only when accurate details are submitted
4.	User should be able to login/New-user should be able to register	User is entered wrong details an error would pop up.	User credential are to be validated and if wrong a pop-up would be	User would be able to view a message if incorrect details
5.	User should be able to login/New-user should be able to register	User would be able to sign-in with Gmail.	User should be successfully logged in.	New-users are able to login.

Registration screen of the Users

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	New-User should be able to register by giving the basic details.	New-Users should fill all the fields and click submit.	New-users should be successfully register.	New-Users are registered.
2.	New-User should be able to register by giving the basic details.	New-User should enter all the mandatory fields. First name, email, password.	New-users are registered once the user enters the mandatory fields	New-Users are able register.
3.	New-User should be able to register by giving the basic details.	New-Users would be notified with validation regarding the mandatory fields.	New-Users are able to view an error if fields are left blank.	New-users are notified with error.
4.	New-User should be able to register by giving the basic details.	On submit the details entered are saved to the Data Base.	New-User details are save to the data base once they click the submit button.	Details of new users are save to data base.

Give Ride

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	Registered User should be able to either request a ride or give in ride details.	Users can fill in the ride details and submit their details.	Users should be able to fill the ride details to provide a ride	Users can successfully provide their ride details.
2.	Registered User should be able to either request a	Users can click request ride to view available rides	Users can view available rides	Users are able to view available

	ride or give in ride details.			rides
3.	Registered User should be able to either request a	User can click offer to create a ride	Users should be able to create rides	Users are able to create the rides

	ride or give in ride details.			
4.	Registered User should be able to either request a ride or give in ride details.	Users can click on logout to come out of the application.	Users are redirected to the login page if they select the logout.	Users are logged out successfully.

Home screen

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to choose different services	Users should click request ride to redirect to ride page.	Users should be able to redirect to the ride page	Users are successfully redirected.
2.	User should be able to choose different services	Users should click offer ride to redirect to the ride page	Users should be able to redirect to the ride page	Users are successfully redirected.
3.	User should be able to choose different services	Users should click past rides to redirect to the view rides page	Users should be redirected to the past rides page	Users are successfully redirected to the page.
4.	User should be able to choose different services	Users can click on logout to come out of the application.	Users are redirected to the login page if they select the logout.	Users are logged out successfully
5.	User should be able to choose different services	Users can click manage settings button to redirect to manage settings page	Users should be able to redirect to the page	User are successfully redirected
6.	User should be able to choose different services.	User can click on the emergency button to	Emergency SMS would be send to the user's contact	User is able to send the message

Manage settings screen

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to update his details	Users should enter any field that he wants to be update	Users should be able to update the information	Users have successfully updated the information.
2.	User should be able to update his details	User can enter the emergency contacts and click save	Users should be able to redirect to home page	Users are successfully redirected to home page
3.	User should be able to	User can send SMS to the	User is able to	User is able to

	send SMS to emergency contacts.	emergency contacts by voice or a default message.	send message	successfully send message
--	---------------------------------	---	--------------	---------------------------

6 Take rides

S.No	Test Description	Steps to Follow	Expected Result	Actual Result
1.	User should be able to rate their previous rides	Users should rate their previous ride and click save.	Users should be able to redirect to the home page	Users are successfully redirected.
2.	User should be able to call or request the rider for ride	User should be able to make a call to the rider or send request for acceptance	User is able to make call or request.	User is successfully contacting rider for ride.

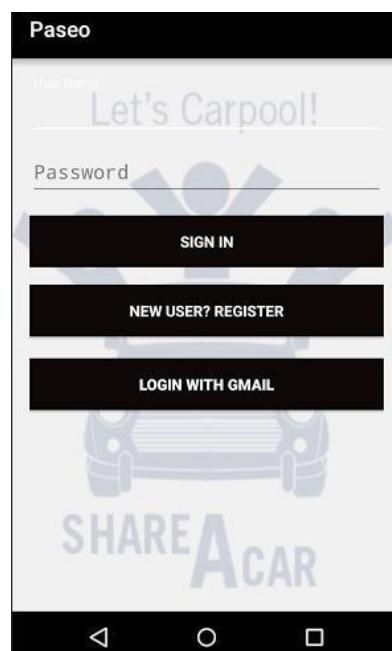
IMPLEMENTATION:

Mobile client Implementation: Implementation in mobile is feasible and is easy to use as most of the present world runs on smart phones. We have implemented the project using Android.

Server Implementations: Used Mongo database to store and retrieve the details of the user about his ride and profile info.

DEPLOYMENT:

Login Page:



Home Page:



Hello

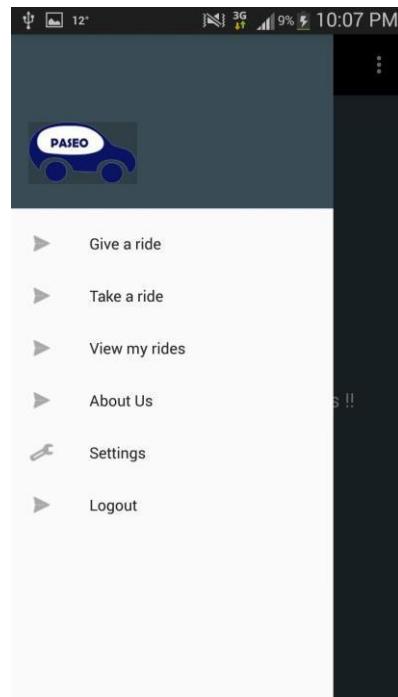
Welcome to Paseo

Current Trip Details

There are no current Trip Details !!



Menu Screen:



Registration Page:

Paseo

Create Account

First Name* Last Name*

Email*

Password*

Confirm Password*

Mobile Number*

Driving License

UPLOAD

SIGN UP

◀ ○ □

Ride Request:

Paseo 

VIEW EXISTING RIDES

Submit Ride Details

From Location

To Location

Date **DATE**

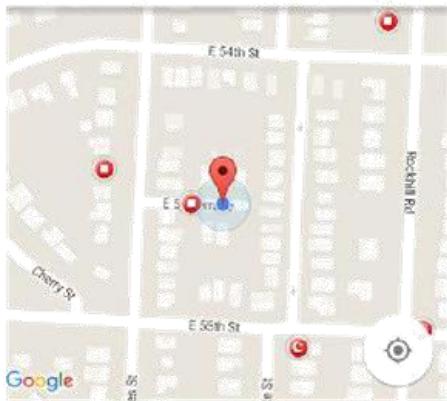
Time **TIME**

LOCATE ON MAP

SUBMIT

◀ ○ □

← Pick a place



Select this location

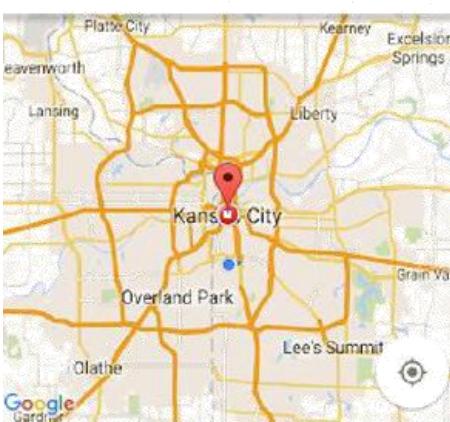
706 E 54 Terrace, Kansas City, MO 64110, USA

NEARBY PLACES

The National Museum of Toys and...

5235 Oak St, Kansas City, MO 64112, United...

← Kansas City, MO, United St...



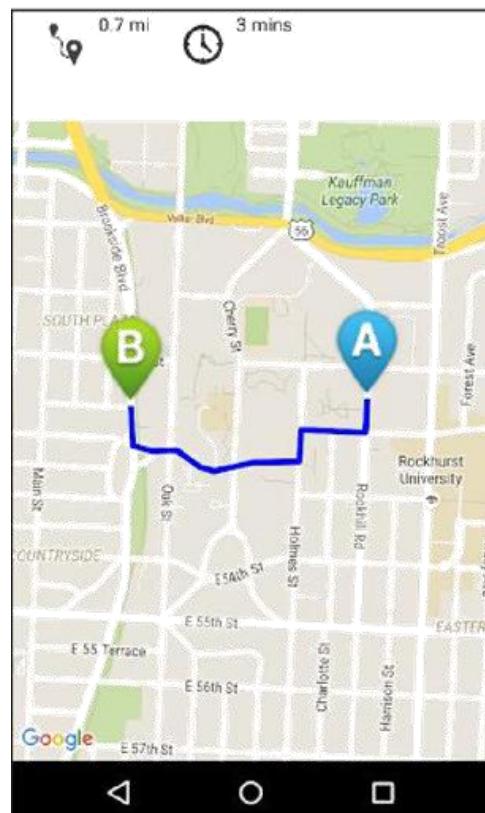
Kansas City

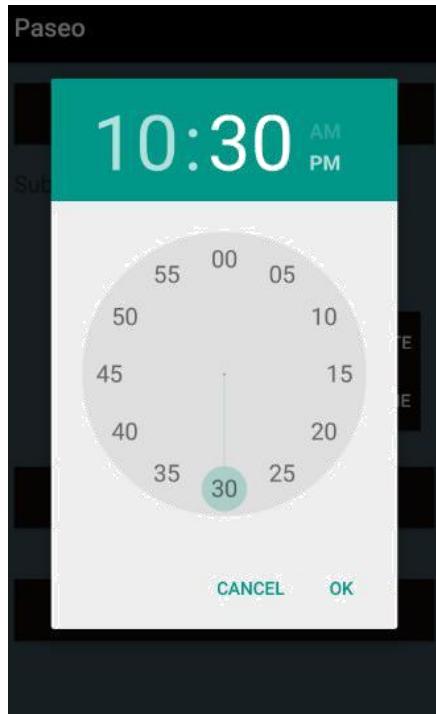
Kansas City, MO, USA

SEARCH RESULTS

Kansas City

Kansas City, MO, USA





Offer Ride:



Kansas City, MO, USA

Texas, USA

26-5-2016



21:33

Your Request is approved !!

Chicago, IL, USA

Washington, DC, USA

2-6-2016



14:40

Your Request is declined !!



Settings



Change password

Edit phone number

Edit Email

Emergency Contacts

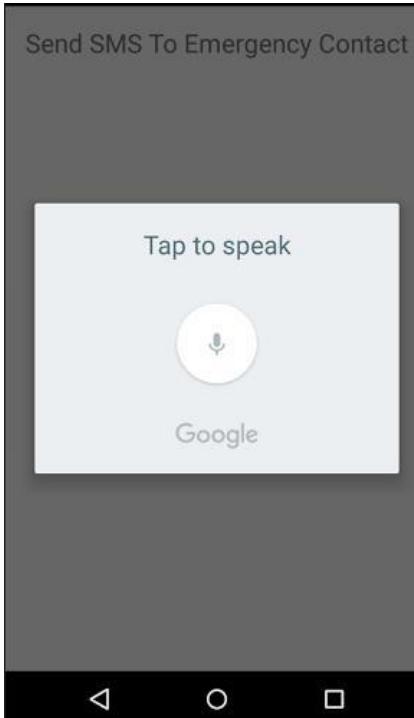
Name _____ Number _____

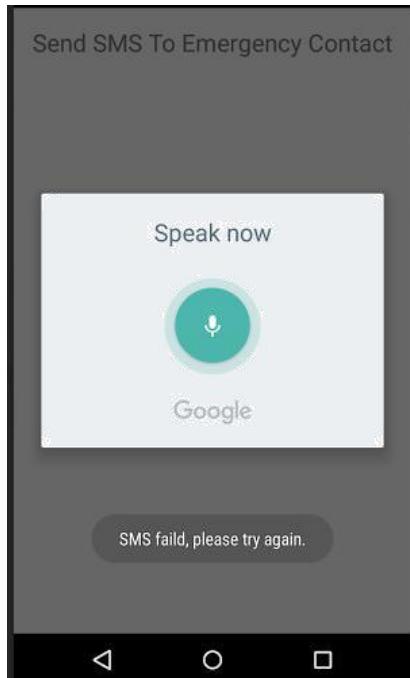
Name _____ Number _____

SAVE



Security features:





PROJECT MANAGEMENT:

Project Timeline: 29th April 2016.

Members: Amulya1, VinuthaMuthyala, Naresh, Megha Sai.

Task Responsibilities: Issues are created in github for each members and every member have collectively worked to complete them in given time. Have shared the work equally among us.

Work Completed:

Stories: UI is improved, security alert messages sent on a click, list view for rides and a feature to call and request ride, Real time implications of call and SMS, Place picker for locations, Direction markers for source and destination, Picture implementation and multiuser implementations.

Service Design: The Google service is implemented to notify the user with the current location and also to pick locations. Mongo database service to save and retrieve the information of the user and ride, they are implemented to enhance the functionality of the user.

Service Implementation: Google API and Mongo database and are being worked successfully.

Responsibilities:

We all have worked together to provide an effective and clear implementation of the project.

Amulya Pindi: Sign-in with Gmail, sending SMS and notifying the user which are a part of emergency, calling and request ride features for accepting rides, updating the details to mongo database, giving feedback to the riders, UI of the settings page, mandatory fields in application and documentation.

Vinutha Muthyala: UI part of the home page, logout functionality, retrieving mongo database details for the emergency features, email implementation, Voice to text conversion as a part of emergency.

Megha Sai: Architecture/use case diagrams, UI, design of request screens, wireframes, manage setting features, place pickers for the locations, map marker implementations and Google Map services.

Naresh: To store the data to Mongo database, UI, Activity diagram, manage settings features, list view for the view ride screen, directions implementation between source and destinations and real time scenario of booking ride.

Time Taken and Contributions: All have contributed 5hrs each almost every alternate day to make the project a success.

BIBLIOGRAPHY:

<https://en.wikipedia.org/wiki/Carpool> https://en.wikipedia.org/wiki/Real-time_ridesharing <http://www.carpoolworld.com/>
<http://www.ijjsce.org/attachments/File/v5i2/B2612055215.pdf>
http://www.ijarcsse.com/docs/papers/Volume_3/4_April2013/V3I3-0385.pdf

Presentation Slides

PASEO



TEAM:

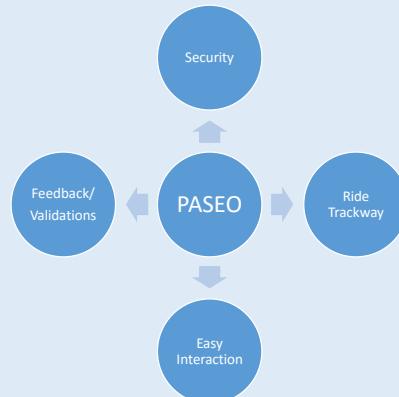
Amulya Pindi 46
Naresh Goud 47
Vinutha Muthyala 36
Meghasai Reddy Bodimani 6

OBJECTIVE

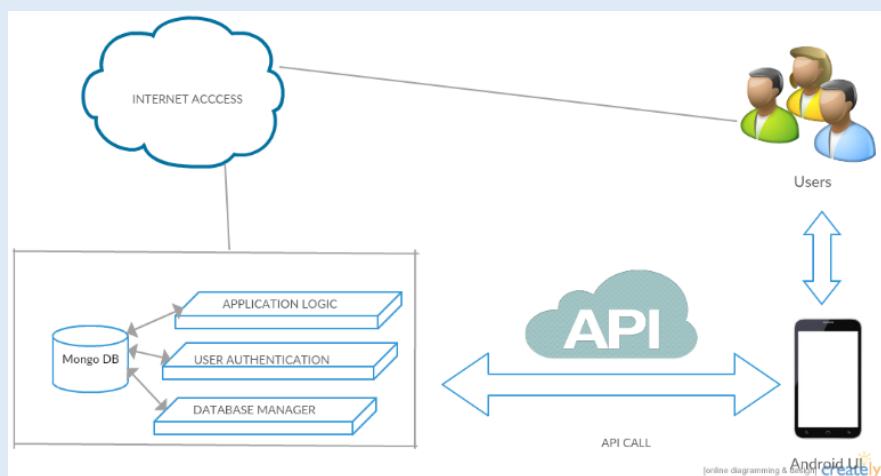
- To design a feasible application enabling the users to quickly access the available rides mentioned in application and to easily communicate with the user.
- To notify and provide security to the users of the application to have a safe journey.

SIGNIFICANT FEATURES

- Riders trackway.
- Security.
- Feedback.
- User Communications.
- Login with gmail.
- Validations.
- Notifying the Users.



Architecture (PASEO)



Github URL

https://github.com/SCE-UMKC/ASESP16_Paseo_2

YouTube Project Video URL

<https://www.youtube.com/watch?v=zdv2X4cAGbw&feature=youtu.be>