

# **CS551 Advanced Software Engineering**

**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:

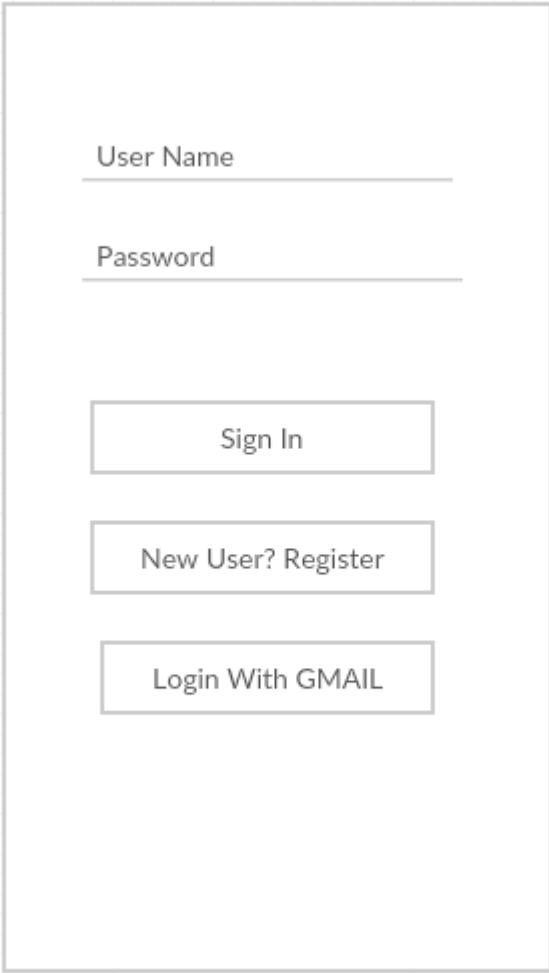
Google Maps API

Mongo DB API

## DETAIL DESIGN OF FEATURES:

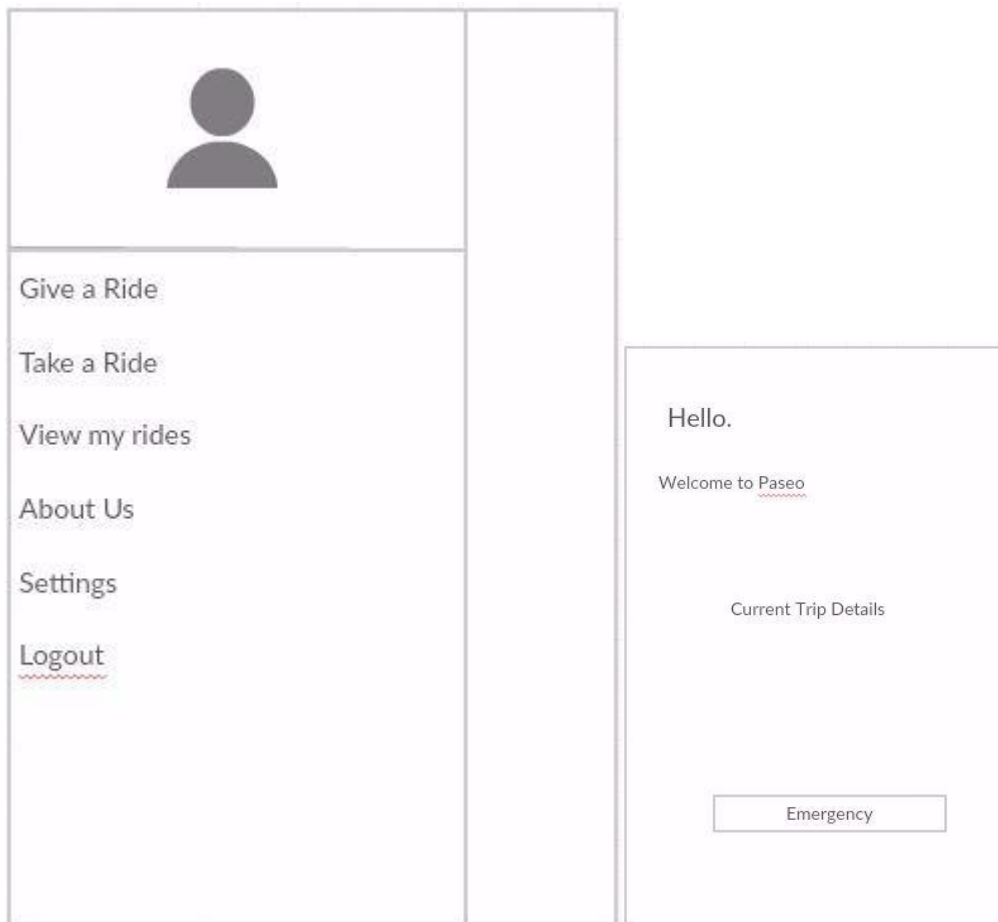
## WIREFRAMES AND MOCKUPS

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 Gmail.

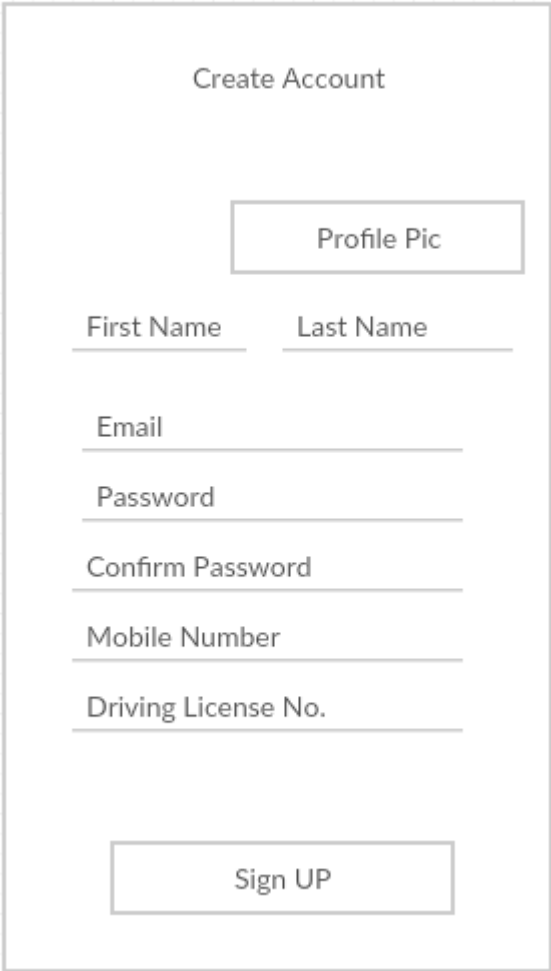


A wireframe of a login page, centered on a light gray grid background. The page is enclosed in a thin gray border. It features two input fields at the top: 'User Name' and 'Password', each with a horizontal line below the text. Below these fields are three buttons stacked vertically: 'Sign In', 'New User? Register', and 'Login With GMAIL'. All text and button labels are in a simple, sans-serif font.

2. 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.



The image shows a registration form titled "Create Account" on a light gray background with a subtle grid pattern. The form is enclosed in a thin gray border. It contains several input fields and a button. At the top, the title "Create Account" is centered. Below it is a rectangular box labeled "Profile Pic". Underneath the profile picture box are two input fields: "First Name" and "Last Name". Following these are five single-line input fields: "Email", "Password", "Confirm Password", "Mobile Number", and "Driving License No.". At the bottom of the form is a rectangular button labeled "Sign UP".

Create Account

Profile Pic

First Name Last Name

Email

Password

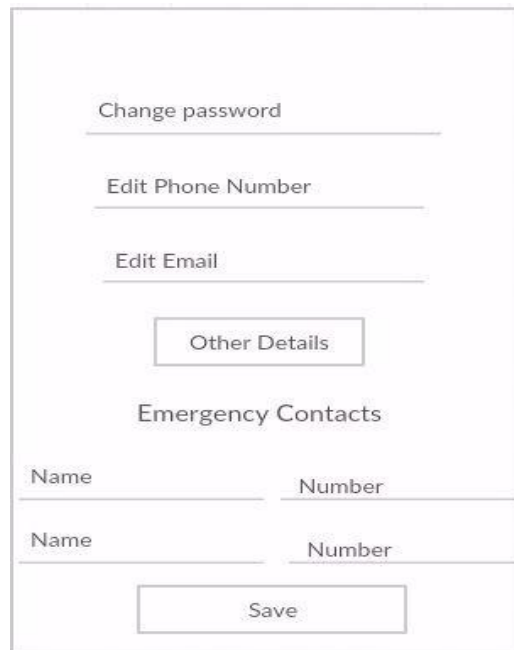
Confirm Password

Mobile Number

Driving License No.

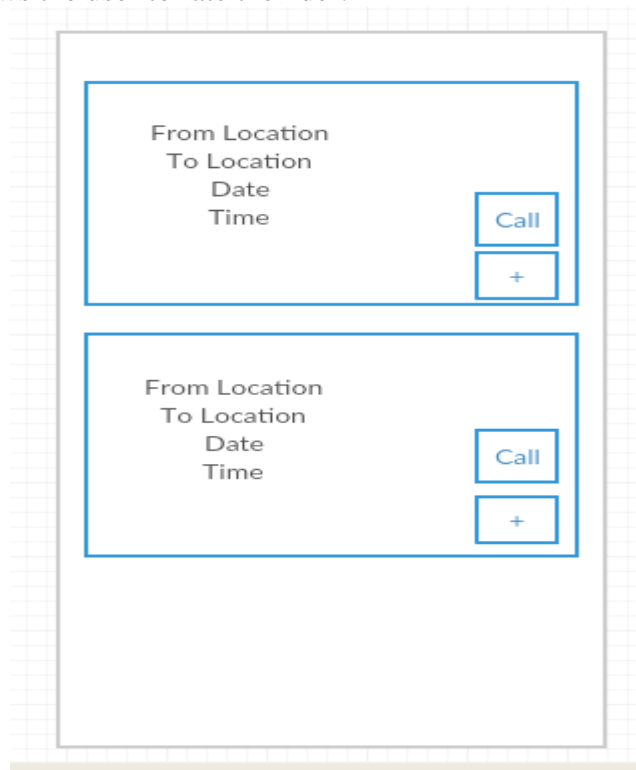
Sign UP

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



The settings page UI mockup is a vertical rectangle with a light purple background. It contains the following elements from top to bottom: a 'Change password' label with a horizontal line below it; an 'Edit Phone Number' label with a horizontal line below it; an 'Edit Email' label with a horizontal line below it; a rectangular button labeled 'Other Details'; a section header 'Emergency Contacts'; two rows of input fields, each with a 'Name' label and a 'Number' label; and a rectangular button labeled 'Save' at the bottom.

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



The 'Take Ride and View existing Rides' page UI mockup is a vertical rectangle with a light gray background. It features two identical ride entry boxes, each outlined in blue. Each box contains the labels 'From Location', 'To Location', 'Date', and 'Time' stacked vertically. To the right of these labels are two blue buttons: 'Call' and '+'. The entire page is set against a light gray grid background, and there is a solid tan horizontal bar at the very 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 Date

Time Time

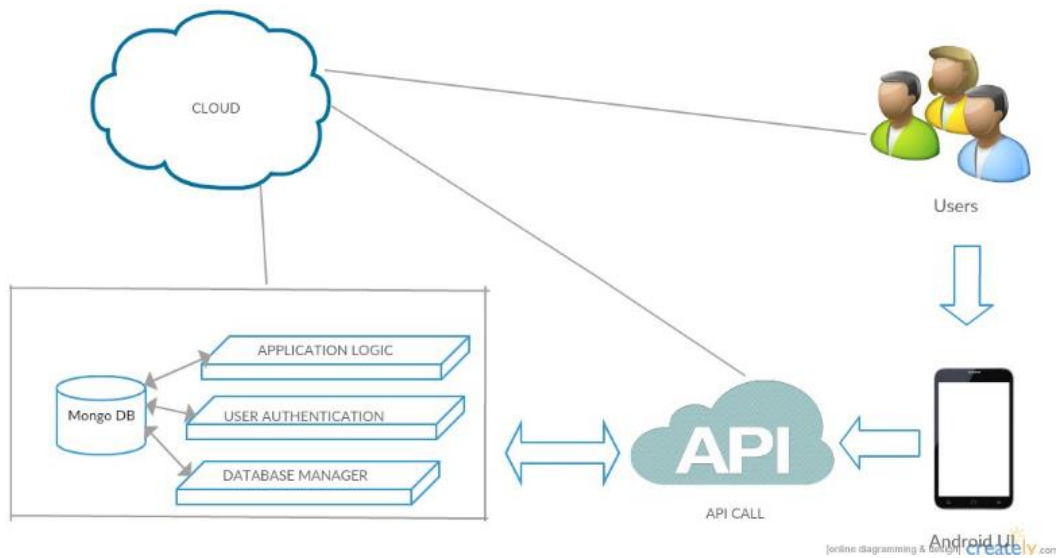
Locate on Map

Submit

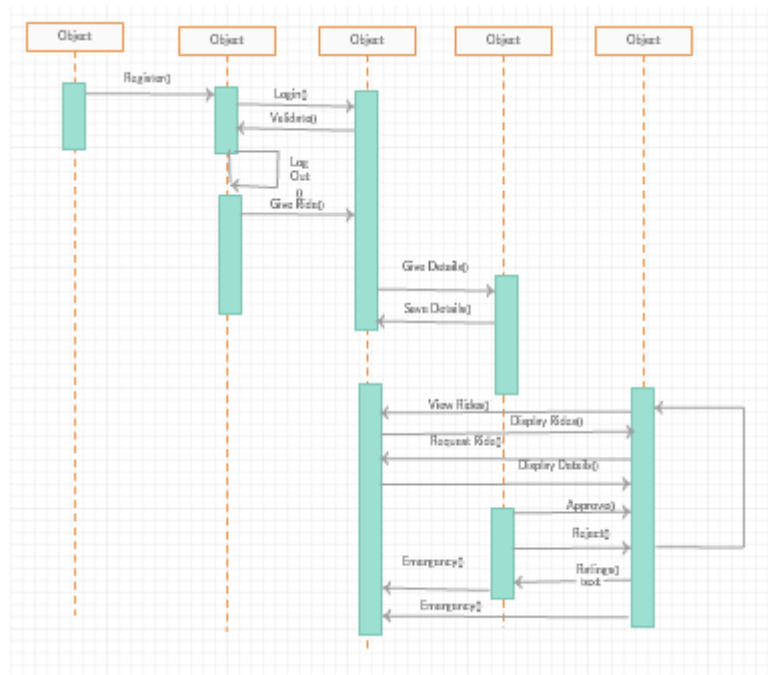
7. About us page to acknowledge user about application.

Text Area

## ARCHITECTURE DIAGRAM:

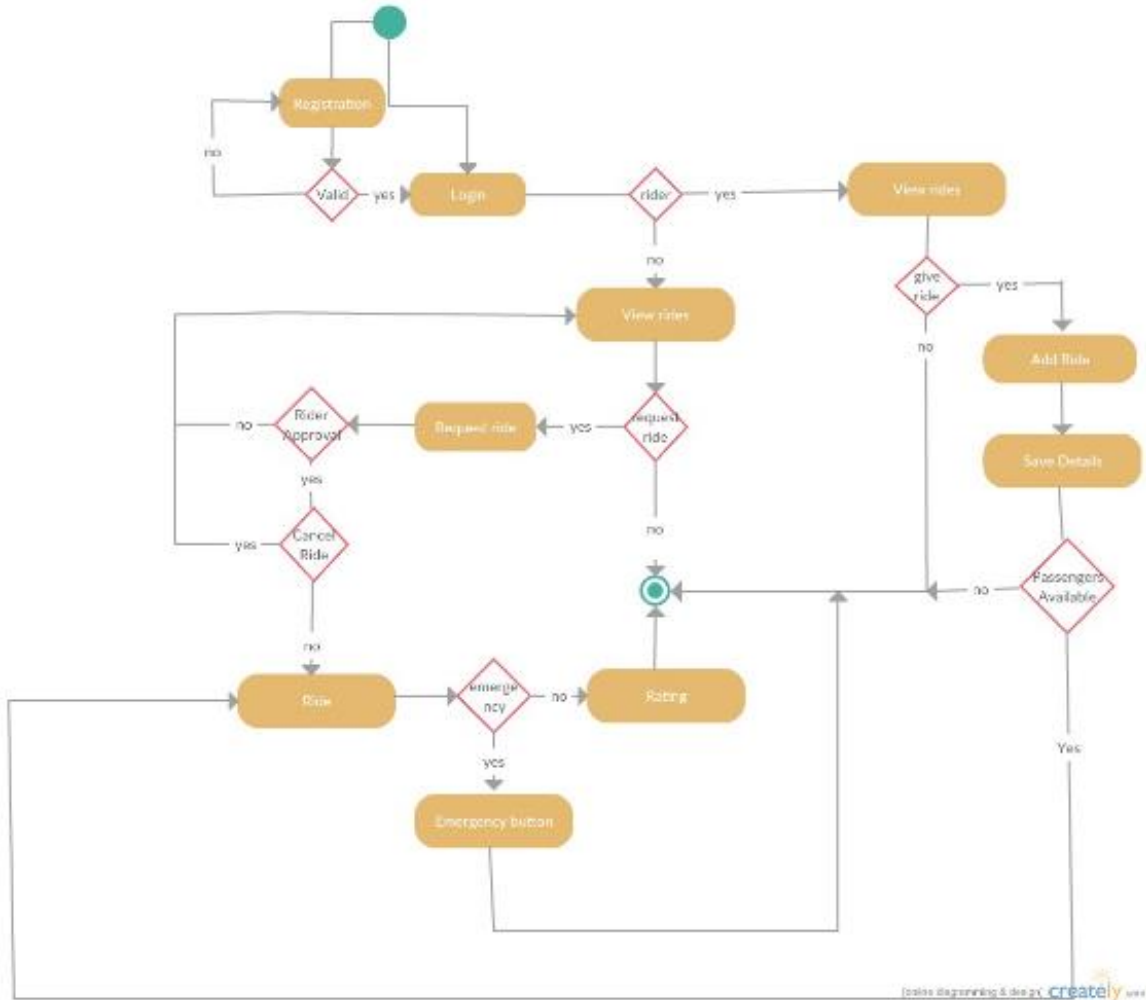


## SEQUENCE DIAGRAM:

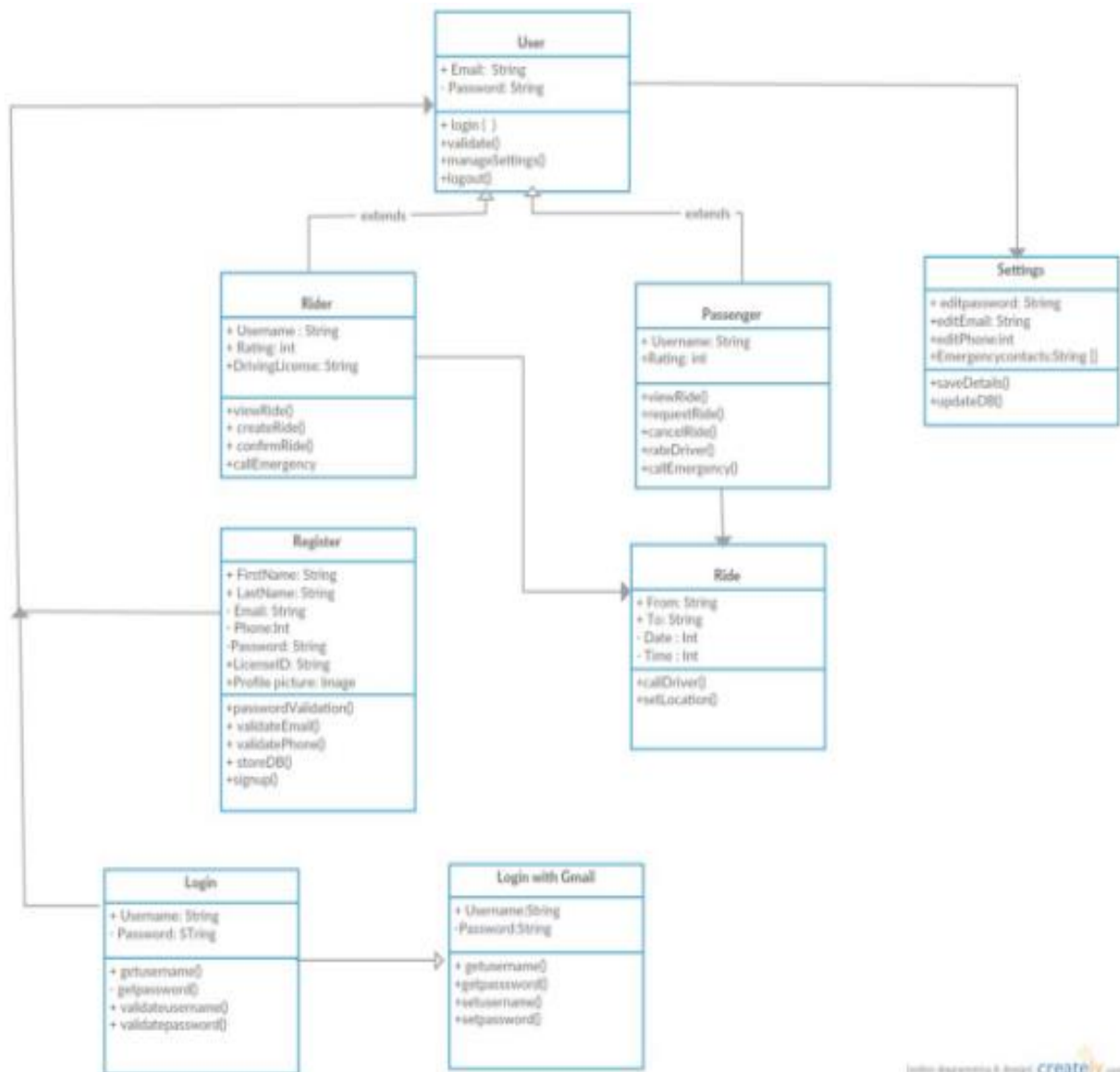




## ACTIVITY DIAGRAM:



## 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.

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.

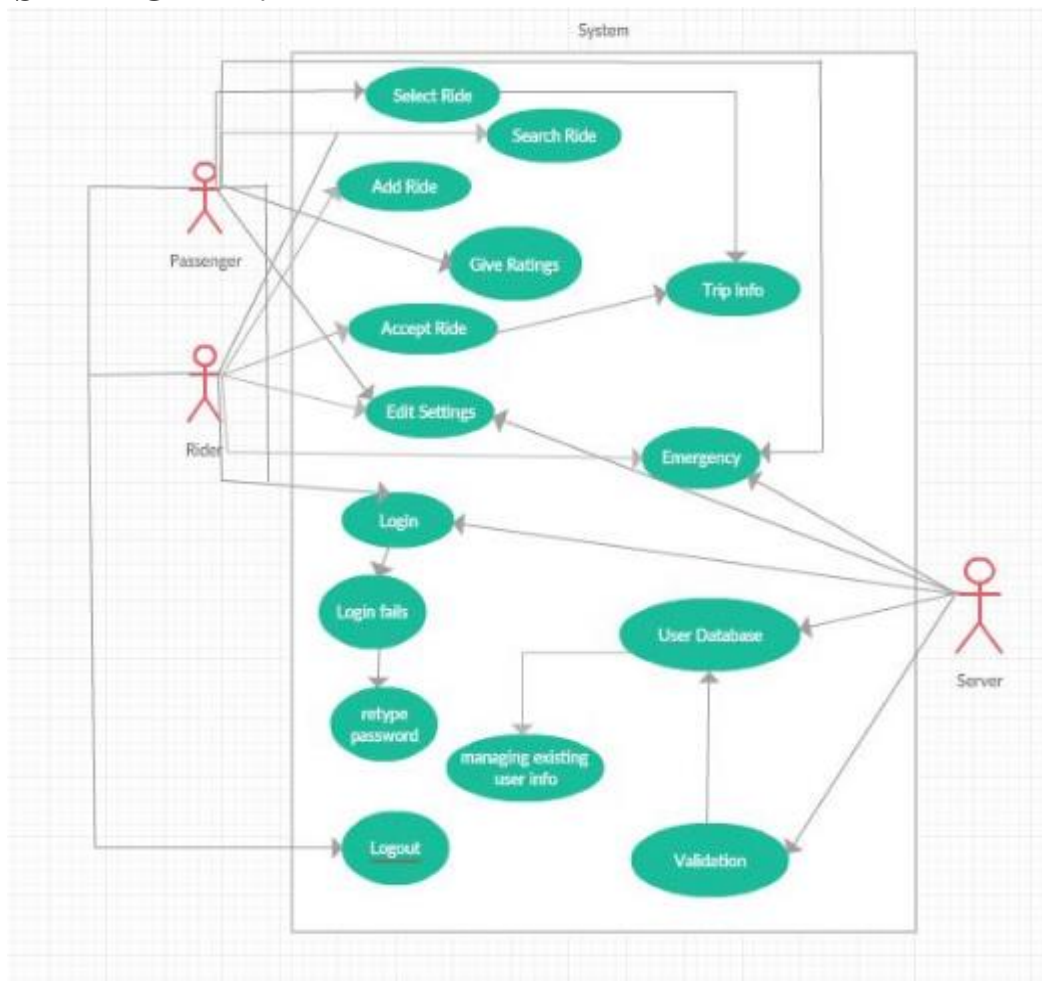
Actor: Users

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:

1. 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 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 non-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 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.

## 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 send SMS to emergency contacts.	User can send SMS to the emergency contacts by voice or a default message.	User is able to send message	User is able to 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.

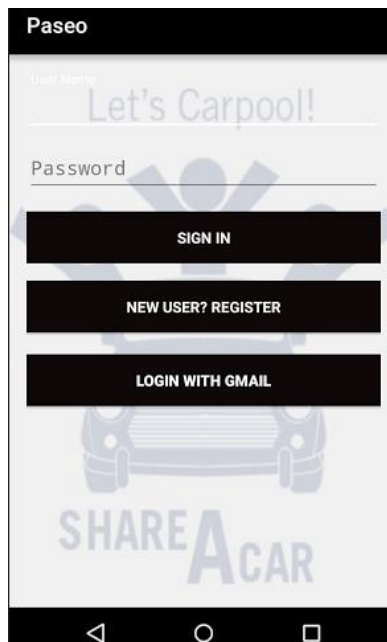
## IMPLEMENTATION:

**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.

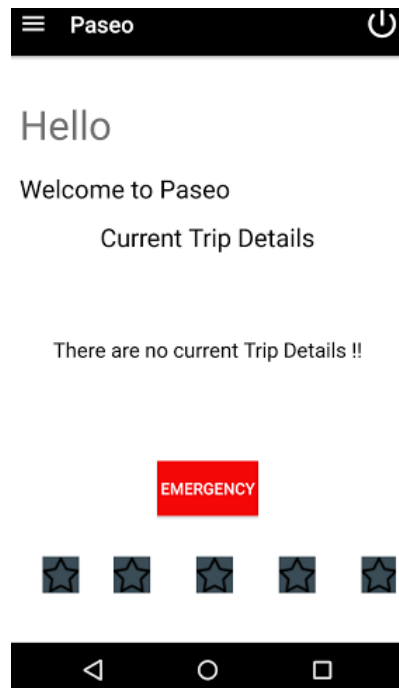
**Server Implementations:** Used Mongo database to store and retrieve the details of the user about his ride and profile info.

## DEPLOYMENT:

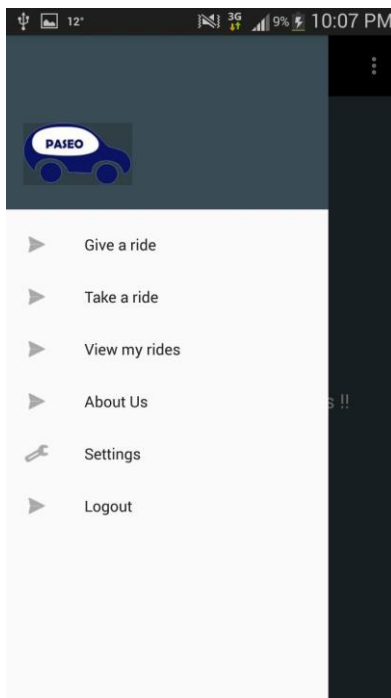
Login Page:



Home Page:



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

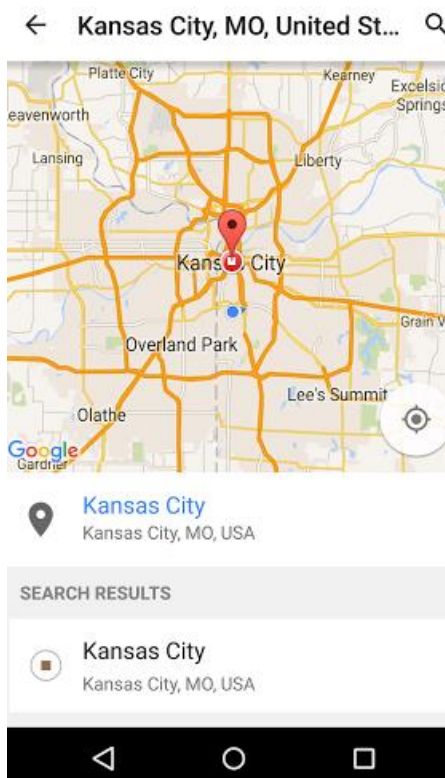
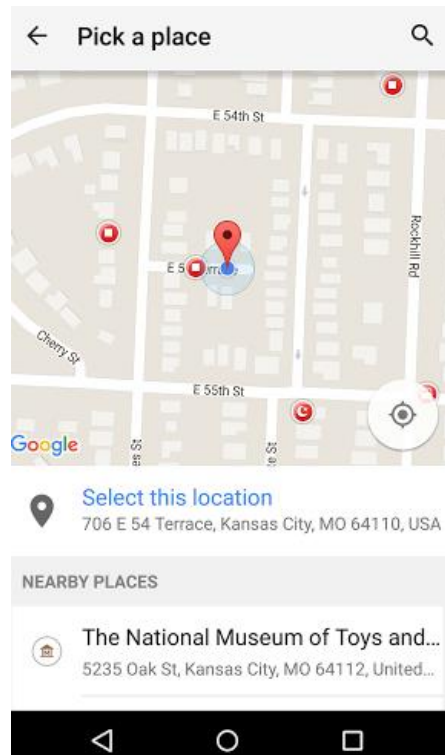
DATE

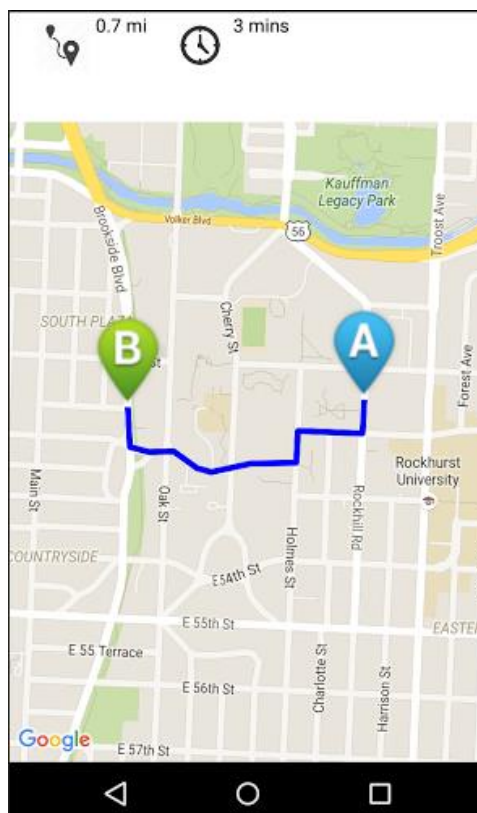
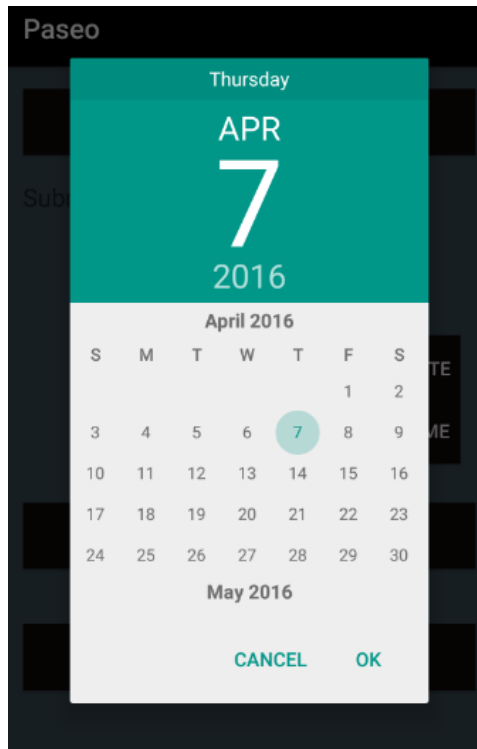
Time

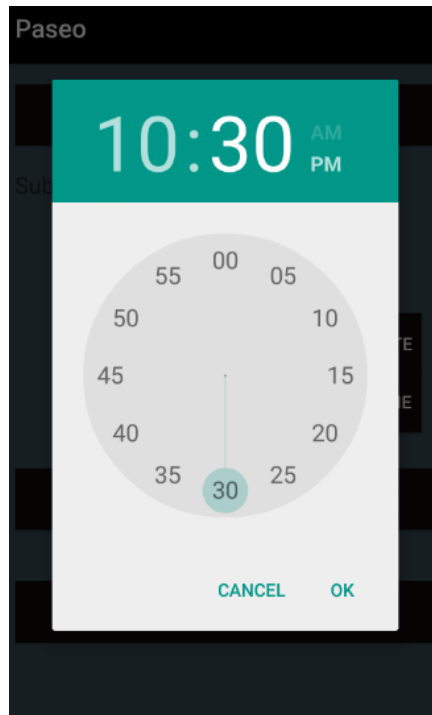
TIME

LOCATE ON MAP

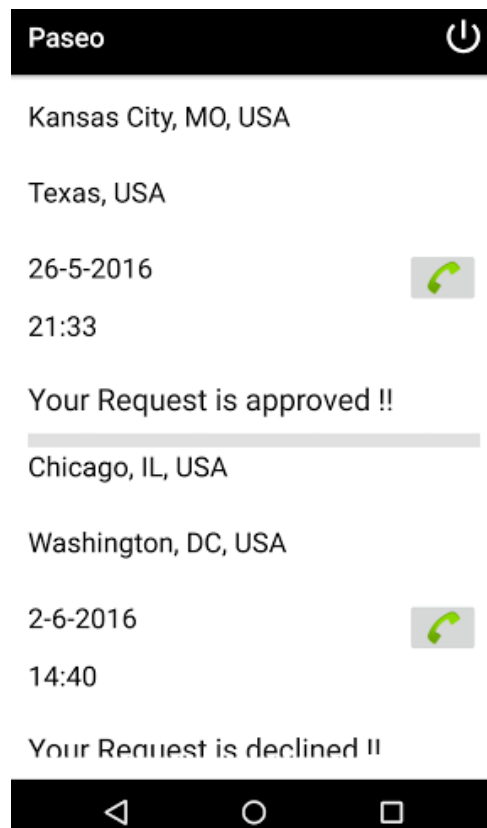
SUBMIT







Offer Ride:



## Settings

Paseo

Change password

Edit phone number

Edit Email

Emergency  
Contacts

Name

Number

Name

Number

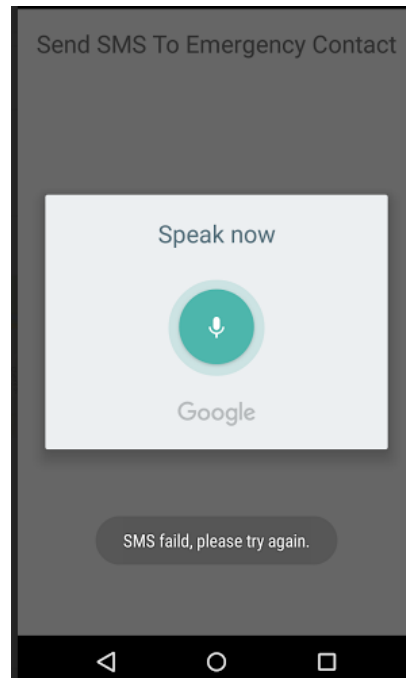
SAVE

## Security features:

Send SMS To Emergency Contact

Tap to speak

Google



## PROJECT MANAGEMENT:

Project Timeline: 29<sup>th</sup> 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](https://en.wikipedia.org/wiki/Real-time_ridesharing)

<http://www.carpoolworld.com/>

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

[http://www.ijarcsse.com/docs/papers/Volume\\_3/4\\_April2013/V3I3-0385.pdf](http://www.ijarcsse.com/docs/papers/Volume_3/4_April2013/V3I3-0385.pdf)