PES UNIVERSITY	**Document Ref.:**	
	Version No.:	1
	Date:	4/2/2018
	Copy No.:	1

Project Name:	Porter Android Application
Project Code:	G42_AK
Status:	**Draft**
Document Type:	**Uncontrolled**

PORTER ANDROID APPLICATION

This Application will enable travellers to book coolies at railway stations across the country and aims at easing the burden off passenger's shoulders for a hassle free travel. The app has features such as ETA, type of coolies, price guide and coolie profiles, allowing users to see past ratings, reviews and more. The customer enters the PNR number details before reaching and the coolie appears within the expected time. Google Maps API is used to pinpoint the pickup location during the boarding process. The App aims to provide a safe, economic, quick and comfortable experience to all passengers of the Indian Railways.

Prepared By:		**Reviewed By:**	
Name	**Date**	**Name**	**Date**
Adithya Krishnan	3/2/2018	Dr. Anant Koppar	20/1/2018
		Dr, Shylaja S Sharath	20/1/2018
Chandini Velilani	3/2/2018	**Approved By:**	
		Name	**Date**
Ravneet Taneja	3/2/2018		

Distribution List

Project Representative(s)	Guide Representative(s)
1. <Name1>	4. Dr. Anant Koppar
2. <Name2>	5. Prof. Sathya Prasad (Help)
3. <Name3>	6. <Name4>

TABLE OF CONTENTS

Definitions, Acronyms and Abbreviations

1. IR : Indian Railways
2. PNR : Passenger Name Record
3. ETA : Expected Time of Arrival
4. GPS : Global Positioning System
5. API : Application Programming Interface
6. SDK : Software Development Kit

References

Books:

[1]Object Oriented Modeling and Design with UML - Book by James Rumbaugh

Industry standard methodology as per the above reference has been used for the design.

Research Papers:

[1] en-Yung Lin, Tuan-Anh Do, Bo-Kai Yang,"Design of Refrigerated Cargo Tracking Systems",2013 International Joint Conference on Awareness Science and Technology & Ubi-Media Computing.

[2] G.-H. Yang, K. Xu, V. O.K. Li, "Hybrid Cargo-Level Tracking System for Logistics," 2010 IEEE 71st Vehicular Technology Conference,(VTC 2010-Spring), pp. 1-5.

[3] Adewole Adewumi; Victor Odunjo; Sanjay Misra,"Developing a mobile application for taxi service company in Nigeria",2015 International Conference on Computing, Communication and Security (ICCCS)

Change History

This section describes the details of changes that have resulted in the current High-Level Design document.

#	Date	Document Version No.	Change Description	Reason for Change
1.	6.2.2018	1	Initial Commit	Initial Commit
2.	15.2.2018	2	Second Commit	Inputs from Guide
3.				

1.0 Introduction

1.1 Overview

Very few people use porter services mainly due to an unregulated system of pricing, issues with luggage capacity and other reasons. The porter industry still needs to invite technology in to build a solid infrastructure which is also customer friendly. Porter would be an app that would facilitate booking of porters with ease similar to the way we book cabs today using OLA and UBER. The passenger enters their PNR number and the porter would appear.

1.2 Scope

The scope of the project includes the evaluation of the platform. The architectural design of the system gives the platform sufficient room to scale up as per the increase in demand. The system design has been built across the pillars of modularity and low coupling. The application targets the online train ticket booking audience whose data can easily be fetched for the Railway System''s Database. The application focuses on the passenger's revenue enhancement and an improved and optimized service without the hassle of choosing a porter and bargaining the price as per the weight.

The application will be consuming a dummy payment portal since the utilization of a proper payment portal is out of scope of development and hence can only be applicable on a fully fledged application.

The application enables the user to:

- Book a porter at his/her choice of pick up point.
- Have a porter waiting at his/her coach during the time of arrival.
- Track the status of luggage.
- View historical listing of all the trips(past/upcoming) and notifications per date.
- View the previous reviews and ratings of the assigned porter.
- Have a private chat window with the porter.
- Carry out payment online.
- Incentives applicable to first time users.

More design features for future scope which can be added to the platform to make the application more effective and user friendly are:

- Adding filters related to price and popularity can help the users make a smart choice
- The application can serve as a common portal for multiple porter provider service application and hence, the user can choose the appropriate service as per the usage and rating.
- As per the previous trip experience and rating the user is assigned the next porter for the upcoming trip and similarly as per the porter's feedback the porter is assigned the next user for the upcoming trip.
- Users could gain points per trip which are added in the Point Basket and can be further utilized as money for the new porter bookings.

2.0 Design Constraints, Assumptions and Dependencies

This section clearly defines the constraints involved in the design.

- The system is bound to only android hosting phones and the OS version.
- The system is limited to HTTP/HTTPS Protocol.
- Clients pay the service fees in cash or via an e-wallet.
- The system does not support distributed database facility.

The Assumptions made are:

- The schema of Railway Database continues to be fixed.
- The pick up points outside railway station are well defined and fixed.
- The trains operates on the correct timings of arrival and departures.
- The willingness of the porter's to join the system and approval for the same from the Coolie Management Society.

The application depend on:

1. The google maps api to fetch the info about the pick- up points which users utilize to arrive at the location.
2. The social media api such as facebook and twitter api to post comments and tweets on the respective platforms.

3.0 Design Description

This section clearly defines the interfaces that exist between two or more modules/classes.

This could be represented diagrammatically for better understanding of the system.

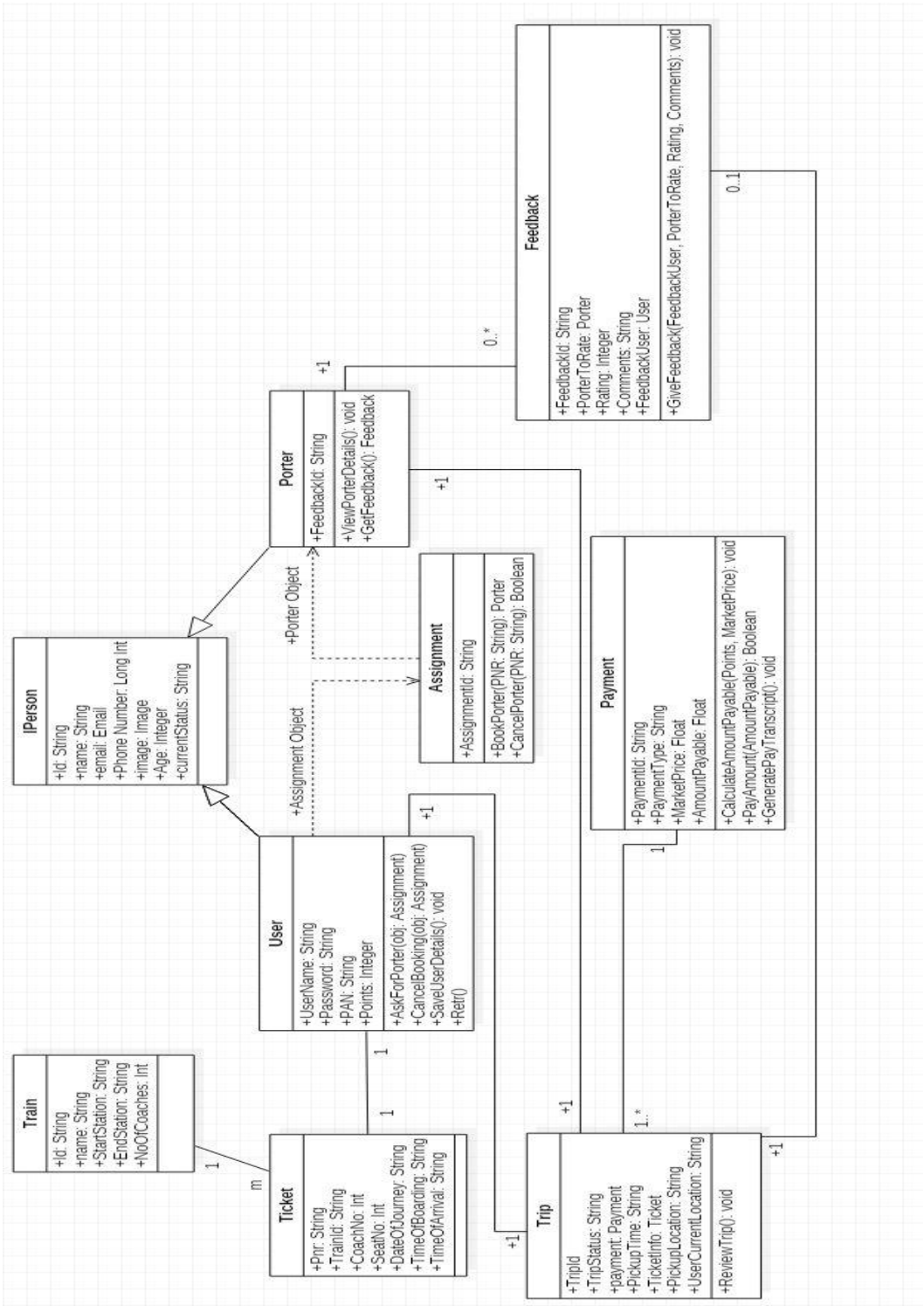
This section explains briefly about the major modules and classes.

The classes involved in the application are as follows:

- Train : This class contains information about the train booked by the user.
- Ticket : This class represents the booked train ticket held by the user
- IPerson : Serves as the interface for the User and Porter classes and holds the registered details of the person.
- User : Represents the user of the system and contains user specific information like PAN number.Enables the selection and cancellation of porter service.
- Porter : Represents the chosen porter.Enables viewing porter ratings and feedback.
- Assignment: This class performs the booking and cancellation processes.
- Trip:This class holds trip related information like the pickup location and time.Has functionality to review the service.
- Payment : This class implements the payment process and handles different types of transactions.
- Feedback : This class implements the feedback functionality and is used as a basis for improving the application.

3.1 Master Class Diagram

A class diagram of the entire system will be given at a high level and then broken down into sublevels in each of the classes below. The following outlines the class diagram for the entire system.


IPerson
+Id: String
+name: String
+email: Email
+Phone Number: Long Int
+image: Image
+Age: Integer
+currentStatus: String
Train
+Id: String
+name: String
+StartStation: String
+EndStation: String
+NoOfCoaches: Int
Ticket
+Pnr: String
+TrainId: String
+CoachNo: Int
+SeatNo: Int
+DateOfJourney: String
+TimeOfBoarding: String
+TimeOfArrival: String
User
+UserName: String
+Password: String
+PAN: String
+Points: Integer
+AskForPorter(obj: Assignment)
+CancelBooking(obj: Assignment)
+SaveUserDetails(): void
+Retr()
Porter
+FeedbackId: String
+ViewPorterDetails(): void
+GetFeedback(): Feedback
Assignment
+AssignmentId: String
+BookPorter(PNR: String): Porter
+CancelPorter(PNR: String): Boolean
Feedback
+FeedbackId: String
+PorterToRate: Porter
+Rating: Integer
+Comments: String
+FeedbackUser: User
+GiveFeedback(FeedbackUser, PorterToRate, Rating, Comments): void
Payment
+PaymentId: String
+PaymentType: String
+MarketPrice: Float
+AmountPayable: Float
+CalculateAmountPayable(Points, MarketPrice): void
+PayAmount(AmountPayable): Boolean
+GeneratePayTranscript(): void
Trip
+TripId
+TripStatus: String
+payment: Payment
+PickupTime: String
+TicketInfo: Ticket
+PickupLocation: String
+UserCurrentLocation: String
+ReviewTrip(): void
Porter Object
Assignment Object
1
m
1
1
+1
+1
+1
+1
0..*
1
1..*
+1
0..1

3.1.1 Description

Detailed description about each component of the class diagram:
Train : This class contains information about the train booked by the user

Ticket : This class represents the booked train ticket held by the user

IPerson : Serves as the interface for the User and Porter classes and holds the registered details of the person.

User : Represents the user of the system and contains user specific information like PAN number. Enables the selection and cancellation of porter service.

Porter : Represents the chosen porter. Enables viewing porter ratings.

Assignment : This class performs the booking and cancellation processes.

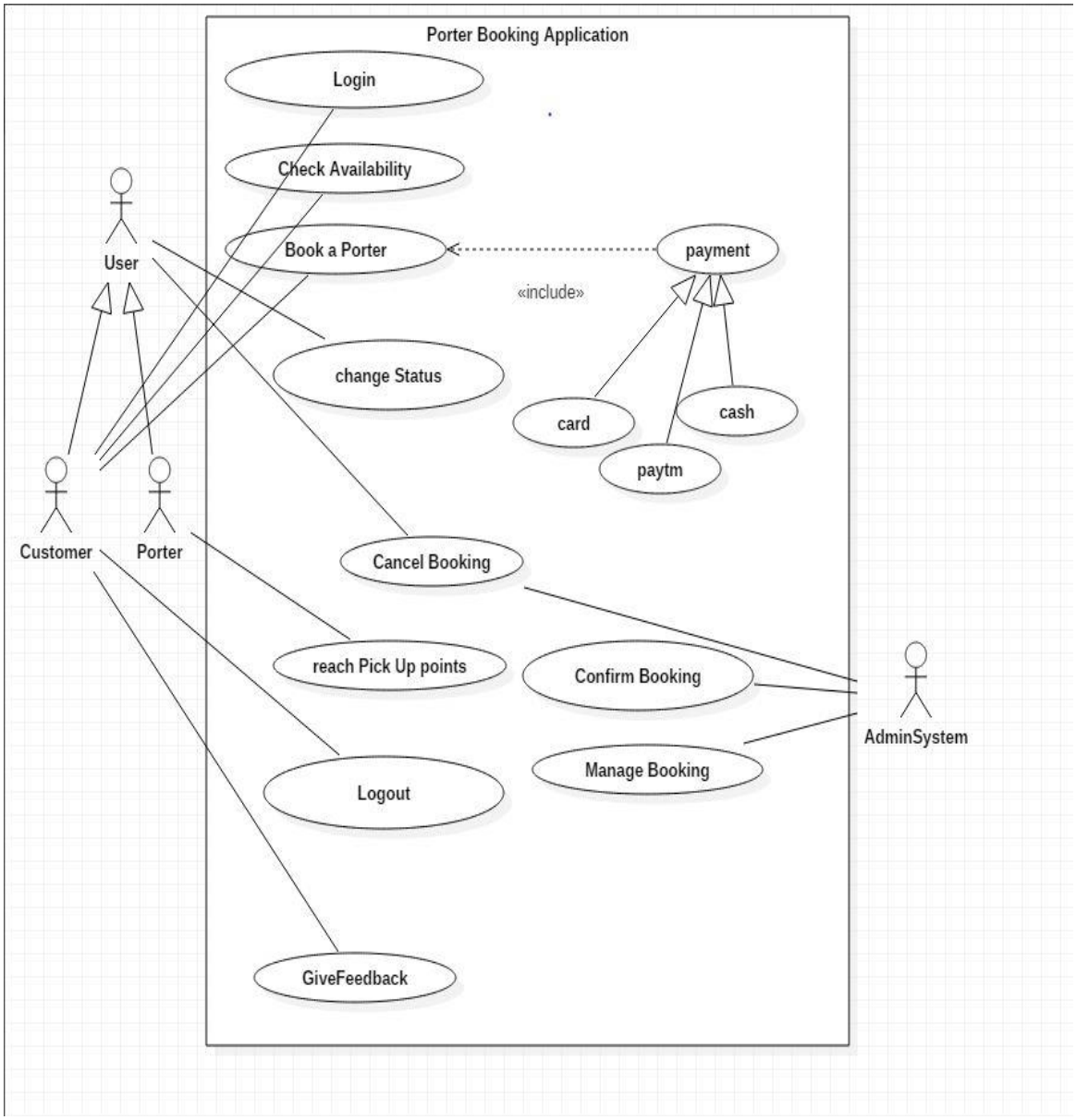
Trip : This class holds trip related information like the pickup location and time.Has functionality to review the service.

Payment : This class implements the payment process and handles different types of transactions.

Feedback : This class implements the feedback functionality

3.1.2 Use Case Diagram

This section shall depict the use-case diagram or reference to the CRS shall be made if it is defined in CRS. The diagram shall be broken up into multiple levels based on the need.

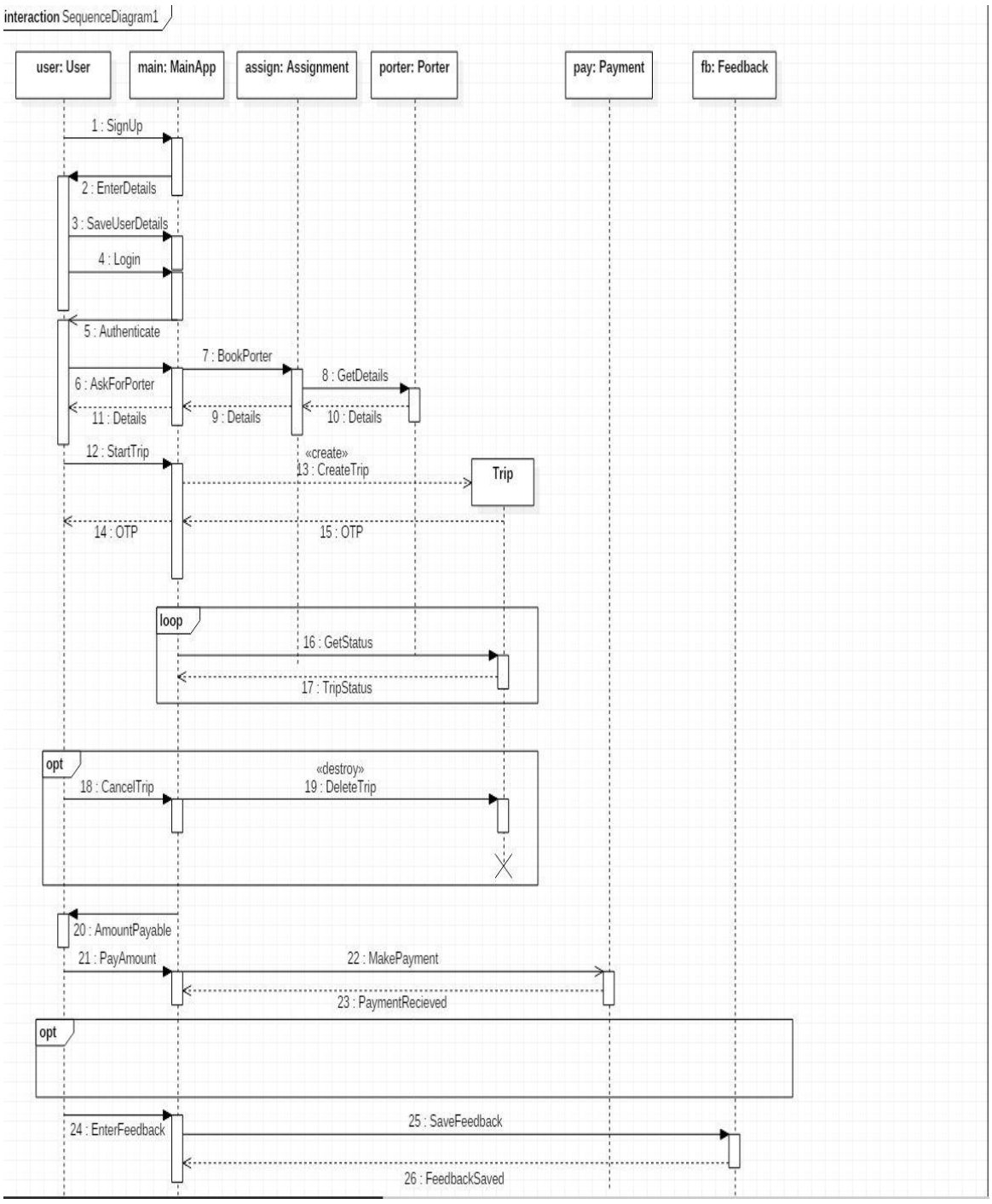


Use Case Item	Description

Login	Enables the customer to login to his home page.
Check Availability	Enables the customer to check porter availability.
Book a Porter	Enables the customer to book the chosen porter.
Change Status	Enables the user of the system to switch status between OnService and Available.
Cancel Booking	Enables the customer to cancel the booking.
Reach Pickup Points	Guides the customer to reach the specified pickup point during onboard.
Logout	Enables the customer to logout from his account.
GiveFeedback	Enables the customer to give feedback about the porter service
Payment	Enables the customer to pay for porter service through three modes card,cash or paytm.
Confirm Booking	Enables the System Admin to confirm the porter service booking after checking porter status.
Manage Booking	Enables the System Admin to manage porter bookings by ensuring no conflicts in the booking process between service requests and also update service logs periodically.

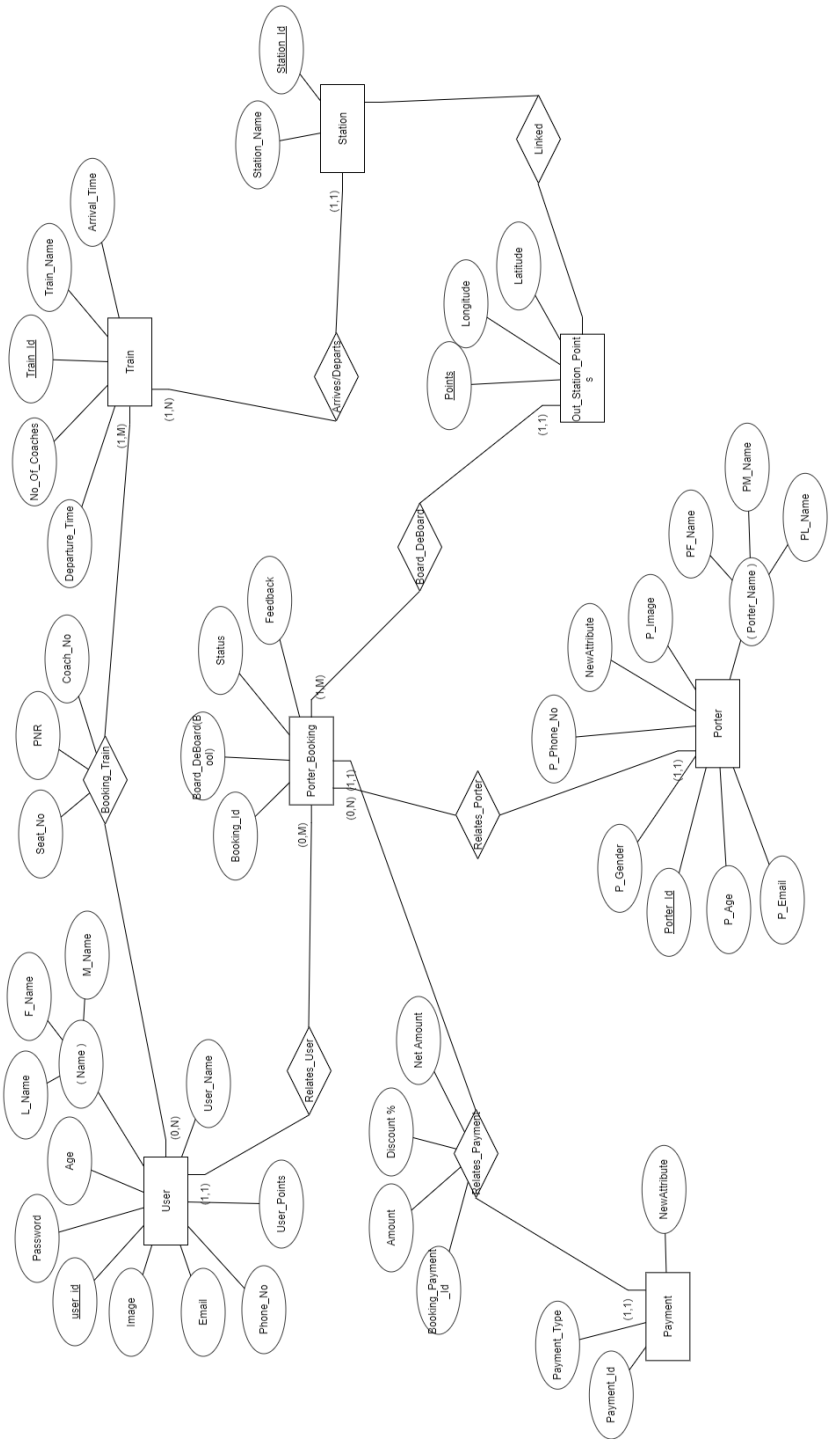
3.1.3 Sequence Diagram

The Sequence diagram for each module will be presented here.



4.0 ER Diagrams

This section will include the ER Diagram. The following table shall be filled for details of the entities and their data elements / attributes.


Station_Id
Station_Name
Station
Linked
(1,1)
Arrives/Departs
Arrival_Time
Train_Name
Train_Id
No_Of_Coaches
Departure_Time
Train
(1,N)
(1,M)
Longitude
Latitude
Points
Out_Station_Point
s
(1,1)
Board_DeBoard
Feedback
Status
Board_DeBoard(B
ool)
Booking_Id
Porter_Booking
(1,M)
(0,M)
(0,N)
(1,1)
Relates_Porter
Coach_No
PNR
Seat_No
Booking_Train
PM_Name
PF_Name
PL_Name
(Porter_Name)
P_Image
NewAttribute
P_Phone_No
Porter
(1,1)
P_Gender
Porter_Id
P_Age
P_Email
M_Name
F_Name
(Name)
L_Name
Age
Password
User_Id
Image
Email
Phone_No
User_Name
User_Points
User
(0,N)
(1,1)
Relates_User
Net Amount
Discount %
Amount
Booking_Payment
_Id
Relates_Payment
NewAttribute
Payment
(1,1)
Payment_Type
Payment_Id

#	Entity	Name	Definition	Type
ENTITIES				
1.	Passenger	User	User of the App	Class
2.	Coolie	Porter	Transports bags	Class
#	**Attribute**	**Name**	**Definition**	**Type (size)**
DATA ELEMENTS				
1.	User_Name	Passenger Name	Name of User	String
2.	Porter_Id	Porter License	Unique ID	Integer

5.0 User Interface Diagrams

A brief description of the screen will be given here. Screens will have references to the appropriate CRS section.

Figure 1 : LOGIN/SIGN-UP

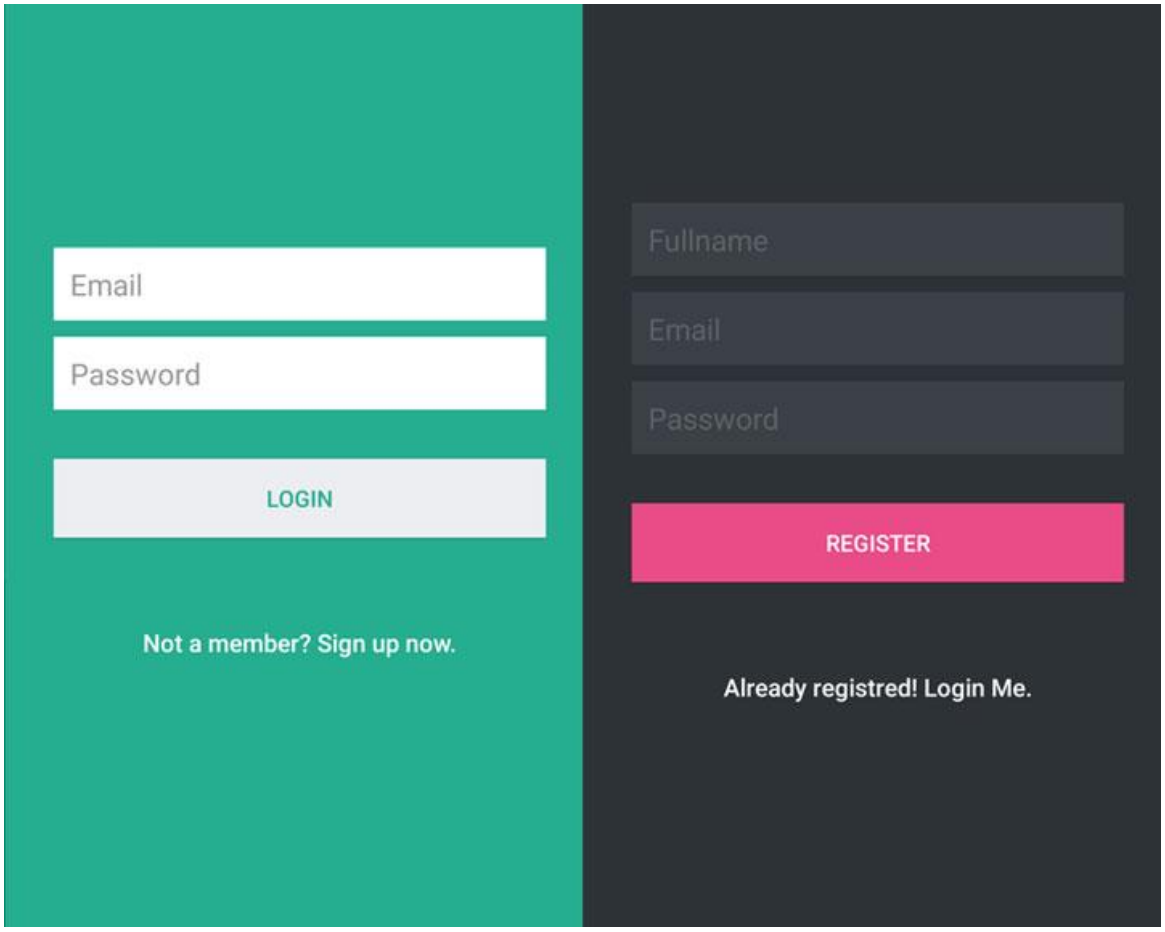


Figure 2 :BOOK PORTER

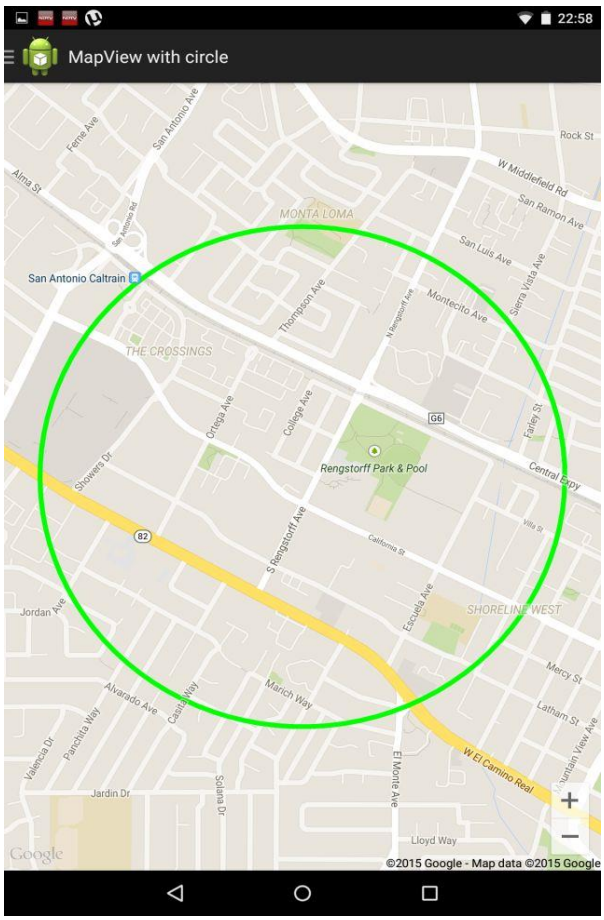


Figure 3: CHOOSE DROP LOCATION

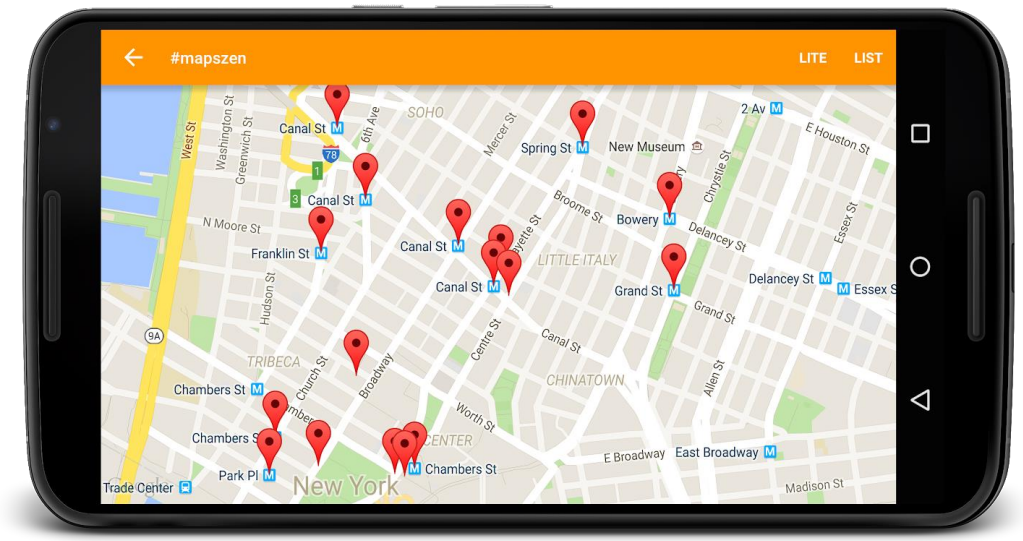
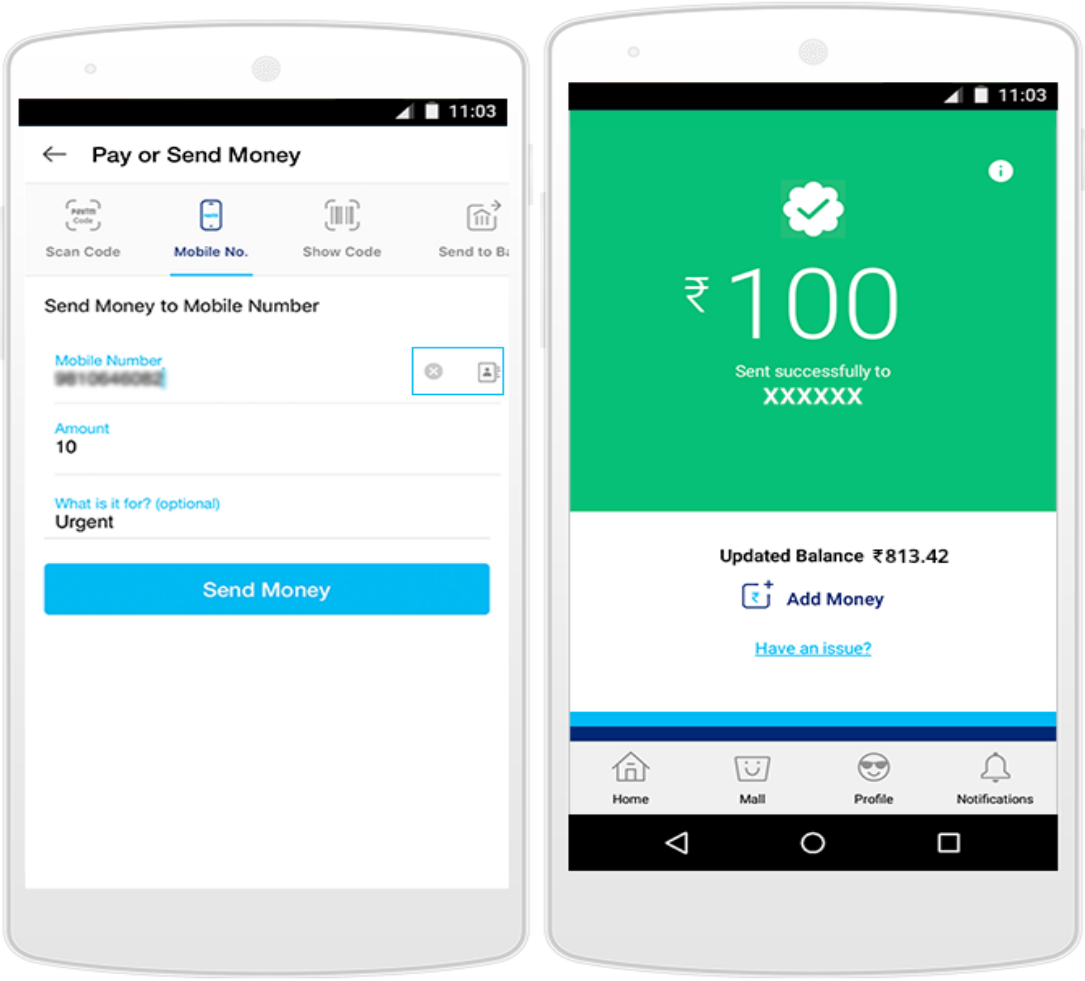


Figure 4 : MAKE PAYMENT



SignUp : This screen enables customer to sign up to a new account.It consists of text boxes to input his username/email and provide password for authentication.Popup appears when the username is taken or password is invalid asking the username to enter new values.

Login : The login screen enables the customer to login to the account .It consists of two query text boxes asking for username/email and password.

UserHome : This screen presents the users home page consisting of his current balance in payment account, previous rides, best rated porters, new railway stations added for service, user personal details etc.The mentioned options would be present as vertical tabs on the screen's left end with BookPorter option on the screen's bottom section.The central part consists of design layouts and images.

SelectPorter: This screen enables user to choose a porter depending on various parameters such as whether the service is enabled on the chosen railway station, currently available porters, industry price, discounts based on current points collected etc.The layout is similar to UserHome with the mentioned options as side tabs vertically on the screen's left end and the central portion indicating the details of the selected porter such as name,age,weight,current location,ratings, availability etc.The bottom section consists of BookService option button along with a text box to enter user PNR allowing the user to accept the porter and book the service.

Service : This screen depicts the ongoing service.It indicates that the chosen porter has accepted the user request and displays his contact details, estimated time of arrival and displays a map where the user can setup pickup points if he is onboarding the train.The map guides the user to the pickup point.

Payment : This screen enables payment options.It consists of text query fields which are centrally aligned asking for payment mode,payment details and popups indicating payment status,discount availed on payment.

Review : This screen enables user to review the service.It consists of a rating option allowing user to rate on a five point scale choosing the star icons displayed and a dropdown asking for any comments after rating.A thank you message to be displayed as popup after review.

6.0 Report Layouts

This section will include a description of the report, which will cover selection criteria, sorting and grouping criteria and the tables used to generate the report. The actual report layout will be put into an appendix and a reference to the same will be provided here.

If the administrator so wishes, they should be able to view statistics gathered by the system regarding bookings. These statistics should be displayed on a page with individually expandable sections, such as extending the number of bookings from the past year to the past two years.

The selection criteria, sorting and grouping should be based on

- The porter
- The customer
- Time range
- Ratings
- Feedback
- frequency of booking

The admin can directly query the database to collect the information,generate reports and run regression analysis to further optimize the system.

The tables the database to generate the report includes USER, PORTER, PORTER_BOOKING, FEEDBACK, PAYMENT.

The following table shows the expected report layout

- Channel Specific traffic

Organic Search	Referrals	Social Media	Email Marketing	Paid Search	Direct Traffic

- Porter (Qualitative Description as per the ratings given by customers)
-

Bad(below 2.5)	Average (2.5<rating<=3.5)	Good (3.5<rating<=4.5)	great(above 4.5)

- User(Qualitative Description as per the porter generated scores)

Bad(below 2.5)	Average (2.5<rating<=3.5)	Good (3.5<rating<=4.5)	great(above 4.5)

- Peak Hours (Frequency of Booking based on the time the trip was initiated at per Railway Station)

Above 70 trips	50<=No. of Trips<70	30<=No. of Trips<50	Below 30

A revenue report can also be generated for a full fledged application.

7.0 External Interfaces

7.1 Hardware Requirement

Hardware Requirements

Device capabilities:TouchScreen, Internet connectivity, GPS functionality.

NETWORK	Technology	GSM / HSPA / LTE
DISPLAY	Resolution	480 x 800 pixels, 5:3 ratio (~217 ppi density)
	Multitouch	Yes
PLATFORM	OS	Android 4.0.4 (Ice Cream Sandwhich),
	CPU	Dual-core 1.3 GHz Cortex
MEMORY	Internal	4 GB, 512 MB RAM
COMMS	WLAN	Wi-Fi 802.11 b/g/n, Wi-Fi Direct
	GPS	Yes, with A-GPS
FEATURES	Messaging	SMS(threaded view)
BATTERY	Talk time	Up to 10 h (4G)

7.2 Software Requirement

The below section specifies the software requirements:

- **Android SDK**:It is a set of development tools used to develop applications for Android platform. The Android SDK includes the required android libraries,Debugger,Emulator,Documentations etc.

Recommended Version for this project is Ice Cream Sandwich or Above.

Source : https://developer.android.com/studio/index.html

- **Java SE:** Java SE API provides the core functionality of the Java programming language. It defines everything from the basic types and objects of the Java programming language to high-level classes that are used for networking, security, database access, graphical user interface (GUI) development, and XML parsing**.**

Recommended version for this project is Version 7 or above**.**

Source:
http://www.oracle.com/technetwork/java/javase/downloads/index.html

- **SQLite Database**: SQLite is a relational database management system, similar to Oracle, MySQL, PostgreSQL and SQL Server.The source code for SQLite exists in the public domain and is free for both private and commercial purposes.

Recommended version is 3.6 and above.

Source : https://www.sqlite.org/

- **PHP Web Service**: We will create a Web Service in PHP, read from the SQLite database, and let the Android Application connect with the Web Service and send data.The data interchange between web service to app and web service to SQLite happens in JSON format.The communications make use of REST API.

Source : http://phpforandroid.net/doku.php

7.3 Communication Interfaces

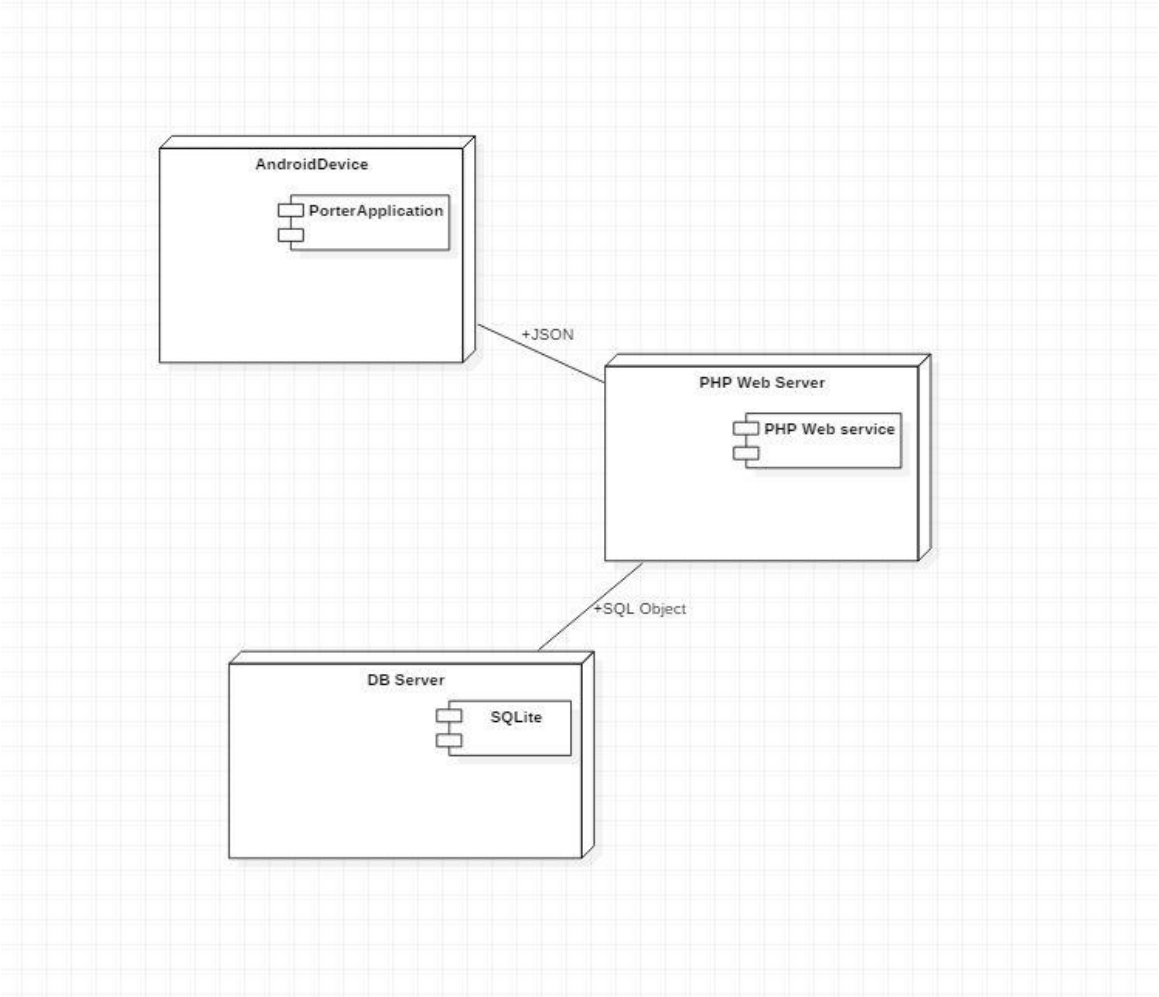
The provided interfaces are applicable across India.

2G capabilities : GSM 900, GSM 1800
3G capabilities : UMTS 900, UMTS 2100
4G capabilities : LTE 850(5), LTE 1800(3), LTE 2100(1), LTE 2300(40), LTE 2500(41).

8.0 Packaging and Deployment Diagrams

The packaging and deployment diagrams for the system shall be presented here.

The following figure is the Deployment Diagram for the system.



9.0 Help

This section shall describe the help planned for the system like, online / context sensitive help and other documentation (e.g. User Manual, Technical Manual) planned, to aid in the usage of the system.

The application will include a Help feature to reach out to the Porter team and directly chat with a customer care agent.

Upon the installation of the application, the application will walk the user through basic setup and usage steps.(This could be a simulated video).

The application will also include an error tracking log that will help the user understand what error occurred when the application crashed along with suggestions on how to prevent the error from occurring again.

10.0 Alternate Design Approach

The design approach used for the application is Model View Controller(MVC).An alternative design approach can be using Model View Presenter(MVC) design pattern. Unlike the MVC design pattern, the presenter refers back to the view due to which mocking of the view easier that leverages the MVP design pattern over MVC. There are two variations of this design

1. Passive View -- in this strategy, the view is not aware of the model and the presenter updates the view to reflect the changes in the model.
2. Supervising Controller -- in this strategy, the view interacts with the model directly to bind data to the data controls without the intervention of the presenter. The presenter is responsible for updating the model. It manipulates the view only if needed -- if you need a complex user interface logic to be executed.

Since the user interface is quite simple and the code flow can be easily followed and tested in the MVC architecture, The MVC architecture is preferred for this application.

11.0 Reusability Considerations

This section shall describe the reusability considerations planned for the project. They may comprise of the following:

- All the components are reusable and easily replaceable due to the independency among all.
- With the increase in request distribution and new worker role servers can be introduced in the system.
- MVC architecture ensures the modularity and ease of flow in the system.

12.0 Traceability Matrix

CRS Reference Section No. and Name.	DESIGN / HLD Reference Section No. and Name.
Section 3.1 Figure 1.1	Section 5.0 User Interface Diagram Figure 1
Section 3.1 Figure 1.1	Section 5.0 User Interface Diagram Figure 2
Section 3.1 Figure 1.1	Section 5.0 User Interface Diagram Figure 4
Section 3.1 Figure 1.1	Section 5.0 User Interface Diagram Figure 2

- SignUp/Log In : Enables SignUp/Login functionality.
- Book Porter : Enables the customer to book the chosen porter.
- Make Payment : Enables the customer to pay for porter service through three modes card,cash or paytm
- Help/Cancel Booking : Enables the customer to cancel the booking.