A Major Project Report On

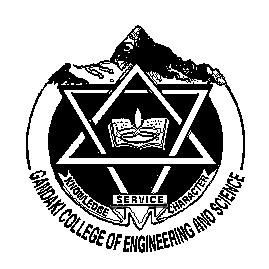
**eTaxi Service**

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Software Engineering at Pokhara University

***By***

**Aashim Bajracharya (01)**

**Jharana Gurung (20)**



**Department of Research and Development**

**GANDAKI COLLEGE OF ENGINEERING AND SCIENCE**

Lamachaur, Kaski, Nepal

**(November, 2017)**

A Major Project Report On

**eTaxi Service**

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Software Engineering at Pokhara University

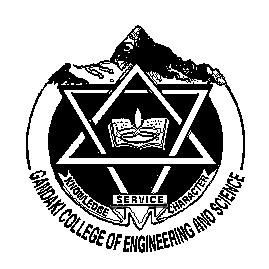
***By***

**Aashim Bajracharya (01)**

**Jharana Gurung (20)**

***Supervisor***

**Er. Sujan Tamrakar**

****

**Department of Research and Development**

**GANDAKI COLLEGE OF ENGINEERING AND SCIENCE**

Lamachaur, Kaski, Nepal

**(November, 2017)**

BONAFIDE CERTIFICATE

This is to certify that this project titled “**eTaxi-Service**” in partial fulfillment of the requirements for the degree of BACHELOR OF ENGINEERING IN SOFTWARE ENGINEERING is a bona fide work of Aashim Bajracharya, Jharana Gurung under the supervision of Mr./Er. Sujan Tamrakar. It is further certified that this work doesn’t form part of any other project work on the basis of which a degree or award was conferred on any earlier occasion on this by any other candidate.

**Date of Evaluation**: 9th November, 2017

|  |  |  |
| --- | --- | --- |
| ……………………………...…  Er. Sujan Tamrakar  Supervisor |  | ……………………………...…  Er. Sujan Tamrakar  Project Head |
|  |  |  |
|  |  |  |
|  |  |  |
| ……………………………...…  Ashok Raj Parajuli  Vice Principal |  | |  | | --- | | ……………………………...…  Er. Bidhur Devkota  Western Regoinal Campus(WRC)  (External) | |
|  |  |  |

**ACKNOWLEDGEMEMT**

We would like to express our inmost appreciation to all those who provided us the possibility to complete this research report. A special gratitude we would like to give to our Project Supervisor, Er. Sujan Tamrakar whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project especially in writing this research report.

Furthermore, we would also like to acknowledge with much appreciation to the staff of Gandaki College of Engineering and Science, who gave the permission to use all required equipment and the necessary materials to complete the android project “eTaxi-Service”. Last but not the least, many thanks goes to the Head of the Research and Development, Er. Sujan Tamrakar, who have invested his full effort in guiding the team in achieving the goal. We have to appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills thanks to their comment and advices.

**TABLE OF CONTENTS**

APPROVAL CERTIFICATE ………….…………………….... i

ABSTRACT …………………………….…………………..... ii

1. INTRODUCTION ……………………………………………….... 1
   1. BACKGROUND ………………………………………... 1
   2. PROBLEM STATEMENT ……………………….……... 1
   3. OBJECTIVES …………………………………….……... 2
   4. IMPLICATION ………­…………………………….……. 2
2. LITERATURE REVIEW…………………………………………… 3
3. TOOLS & METHODOLOGY.…………….……………………….. 8
   1. REQUIRED TOOLS ………………………………… 8
   2. DESIGNS ……...…………………………………….. 9
4. TIMELINE CHART …….…………………………………...…… 31
5. BIBLIOGRAPHY …………………………………………............ 33

Chapter 1

**INTRODUCTION**

* 1. **BACKGROUND**

The taxi service in Pokhara is availed since late 20th century. This public transportation is an effective means of transportation used by the local people as well as tourists in Pokhara. There are mainly two kinds of taxi available in Pokhara city and they are Maruti and Toyota. But now Toyota taxi has been rarely used due to its high taxi fare than another. The taxi service in Pokhara has been flourishing effectively. There are many taxi stops available in the city such Bindhyabasini Temple, Taxi Chowk, Mahendrapool, Nayabazar, Srijana Chowk, Prithvi Chowk, Amarshing Chowk,etc.

* 1. **PROBLEM STATEMENT**

In our day to day life, it is a burden to go to nearby taxi and reach to the destination. In today’s world of time value of money, we feel tedious to hire a taxi and make a bargain with related taxi fare. Even though there is standard taxi fare but it is not practiced fairly. There is no transparency in the taxi fare management. Suppose if any of the factor hampers taxi service, we have to face high rate of taxi fare which is out of standard deviation between effect of factor and the rate of increment of the fare. Hence there is no standard rule and management in the public taxi management.

* 1. **OBJECTIVES**
* To provide on the spot taxi service to the passenger.
* To select a desire destination location by clicking on map interface by passenger.
* To provide passenger an interface to view nearby drivers on the map.
* To select one of the driver nearby by the passenger.
* To get review of duration and taxi fare for the certain destination on the basis of type of the road(i.e highway or city road)
* To notify nominated driver about booking done by certain passenger nearby him/her.
* To cancel the booking done by passenger.   
  1. **IMPLICATION**

Our application is a beneficial for the people of Pokhara Valley. Our application provides on the spot service to the time valued people with transparent standard taxi fare. Our application can be used by all of the people as our application provides user friendly interface with interactive service. We pre-calculate the standard taxi fare according to the destination of the passenger. Due to this, passenger can have unbiased decision to utilize the service easily. Our application can be well applicable for the students who can afford to use taxi service to reach his/her destination.

Chapter 2

**LITERATURE REVIEW**

**Uber** is a ridesharing app for fast, reliable rides in minutes; day or night. It may be most popular application. With Uber, you just tap to request a ride, and it’s easy to pay with credit or cash in select cities. Requesting Uber is easy, here’s how it works:

* Just open the app and tell us where you’re going.
* The app uses your location so your driver knows where to pick you up.
* You’ll see your driver’s picture, vehicle details, and can track their arrival on the map.
* Payment can be made by credit card, cash in select cities, Android Pay, PayPal, and more.
* After the ride, you can rate your driver and provide feedback to help us improve the Uber experience. You’ll also get a receipt by email.

**Ola** is Indian based app for hiring cabs. Book a taxi in just a few taps. Here’s how it works:

* Set your pickup location (E.g. Home, Office, Airport, Railway station or your current location)
* See different types of cabs or taxis in your location, displayed on a map
* Pick the type of cab you want and tap RIDE NOW
* Get instant confirmation with ride details
* Track your cab as it reaches you

Once your trip ends, you can pay by cash or keep your Ola Money recharged to ride cash-free! You can also reduce the taxi fare up to 50% by choosing Ola Share. We match you with other Ola users travelling along your route so you can share a cab. You can also choose to travel exclusively with your colleagues or classmates.

**Lyft** is also alternatives to hiring a cab. It is similar to Uber app. After downloading Lyft and signing up, simply open the app and request a ride. Never worry about having cash for a cab or the bus again with Lyft, you pay through your phone. It’s easy, fast, and secure. Easily split the cost of a ride with friends through the app. No more IOU’s (I Owe You). For drivers, all Lyft drivers pass comprehensive background and DMV checks before being approved for the service, and Drivers are rated by passengers and only the highest-rated drivers are allowed on the road.

**Eddy Cab** is Nepali based app for hiring cab. It works as a taxi providing agency that connects taxi drivers and passengers. Downloading the app, you can quickly choose your route from the available drivers and can request a taxi. Taking only a few seconds to complete, you can arrange your dedicated ride in a fraction of the time. The taxi will arrive to pick you up. Auto detects your current location and will drop you to your destination with satisfaction. For drivers, they will have to visit nearest Agent with following document to get registered as Eddy Cab Driver:

* Eddy Cab KYC( know your customer ) form need to be filled up
* Driving License Copy
* Taxi Blue Book Copy
* Driver’s Citizenship Paper or Passport

As soon as driver will be registered from the same time will be listed as Driver service provider in our system.

**Onver** is another Nepali based app for hiring cab. When you open the application, a map appears with a red pin defining your current location. If you press on the Green Smart Icon, your pickup location will be your current location by default. Should you desire the taxi to meet you at a nearby different location than your current\default one, just long-press on the map on the desired point; a “Blue” pin will appear and when you press on the “Green Smart Icon” to confirm your request, the position of the Blue pin will be sent to the taxi as your desired pickup location. Once your request is accepted by a taxi, you will be notified & the information of your taxi will appear on the top of your screen. The info contains driver’s photo, driver’s name, taxi number, model & color. A calling button appears on the left of the taxi info banner. An orange pin will appear on your map showing the position of your booked taxi. A green line will appear between your pickup location & the taxi’s position, depicting the best route for the taxi to reach your pickup location. You will be able to track in real-time the position of the taxi as he drives towards you. You can also read the ETA inside the smart icon. At any time in the request you feel the need to call your taxi, just click on the “Calling Icon” on the left-top of the screen and a call to your driver will be initiated.

**Uber** and **Lyft** showed up in the news a lot during its popularity. There have been scandalous news stories related to all sorts of things including how the startup treats its customers and drivers, how it activates surge pricing, how it doesn't operate without a license around the world, and all sorts of other terrible things.

**Eddy Cab** and **Onver** focused only on Kathmandu valley and it doesn’t estimate the price of ride.

Our application **eTaxi Service** is also similar to these app mainly focusing on people of Pokhara valley for safety ride maintaining standard price as possible. Moreover our application provides emergency service to needed passenger with transparent standard taxi fare. The passenger can also book or can share a booked ride at a resolute date and time remarking his/her location so that the taxi can pick up. There are different discount schemes for new app users or a regular user. For more safety, there will be emergency tap for customer if there is any dangerous situation so that our system will get notification.

Chapter 3

**TOOLS & METHODOLOGY**

**3.1. REQUIRED TOOLS**

During the development of the system, we require various tools essential for the project. It would have been a nightmare without those tools. Here are some list of tools used in the project.

* 1. Sublime Text Editor
  2. Android Studio
  3. PHP web framework
  4. Cacoo Wireframe
  5. Dia
  6. Adobe PDF Reader
  7. Microsoft Office
  8. Github
  9. Atleast 2 Android Device

Sublime Text is a proprietary cross-platform source code editor with a Python application programme interface(API). We use Sublime Text editor for the purpose to create various kinds of documents such as text documents for temporary use. Moreover we write programming language in this text editor.

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. Android Studio is the platform or the tool we use to develop the application. We compile, execute and test the code during the development of the project.

Laravel is a free, open-source PHP web framework that is for the development of web applications following the model–view–controller (MVC) architectural pattern. Some of the features of Laravel are a modular packaging system with a dedicated dependency manager, different ways for accessing relational databases, utilities that aid in application deployment and maintenance, and its orientation toward syntactic sugar. We create Laravel api for Android implementation to handle the database and to control the database from the web.

The Cacoo Wireframe is used for creating the wireframe model of our product. Cacoo is a web-based design tool, that allows real-time collaboration on a the same diagram, at the same time. With this application you can create network charts, site maps, UML and wireframes.

Dia is free and open source general-purpose diagramming software and uses a controlled single document interface (SDI) similar to GIMP and Inkscape. Dia is used to create various designs and artifacts for the project such as use case diagrams, system sequence diagrams, domain diagrams, class diagrams etc.

Adobe PDF Reader is a family of Adobe Acrobat for Application software and web services to view, create, manipulate, print and manage files in Portable Document Format (PDF). The basic Acrobat Reader is available for several desktop and mobile platforms.

Micosoft Word is a word processor included in Microsoft Office and some editions of the now-discontinued Microsoft Works. It implemented the proprietary doc format as its primary format. Microsoft Office are used for the purpose of the documentation.

GitHub is a Web-based Git version control repository hosting service. It is mostly used for computer code. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project. GitHub offers both plans for private and free repositories on the same account which are commonly used to host open-source software projects. Moreover we use Github to synchronize our task during the project development.

* 1. **DESIGNS**

We have created use case for our application as mentioned below in Fig 1.1. Similarly, we have designed system sequence diagram (SSD) for main use cases such as “notify status”, “request service” and “process passenger request with nearby driver” below. Also we have designed the ER diagram mentioned below.

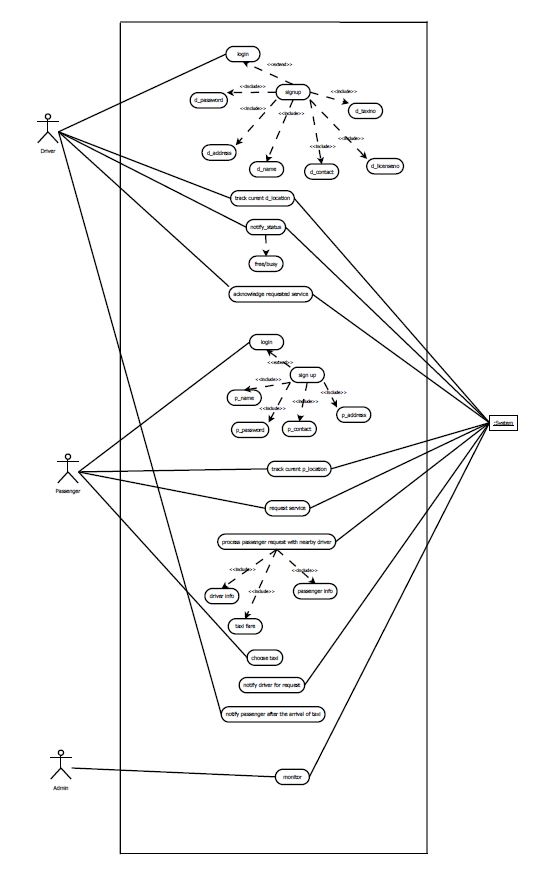


Fig 3.1. Use Case

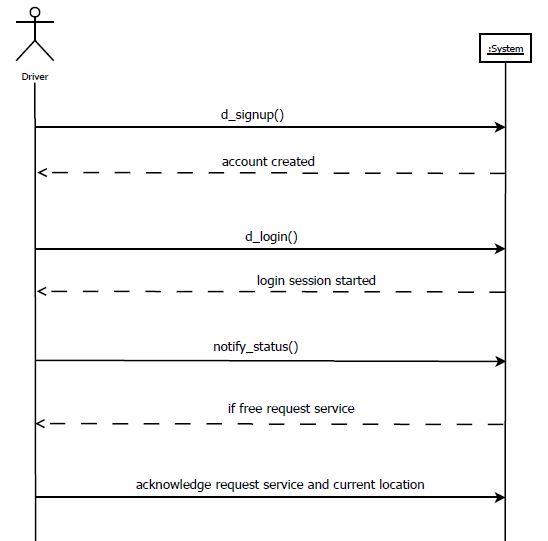


Fig 3.2. System Sequence Diagram for use case “notify status”

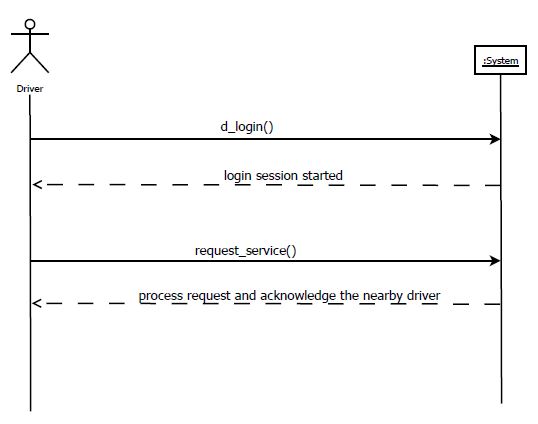


Fig 3.3. System Sequence Diagram for use case “process passenger request”

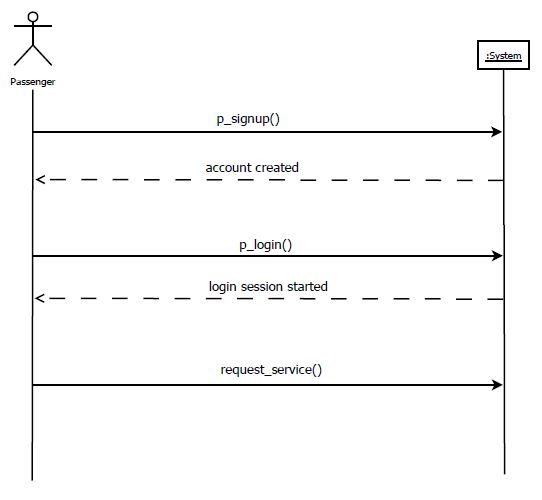


Fig 3.3. System Sequence Diagram for use case “request service”

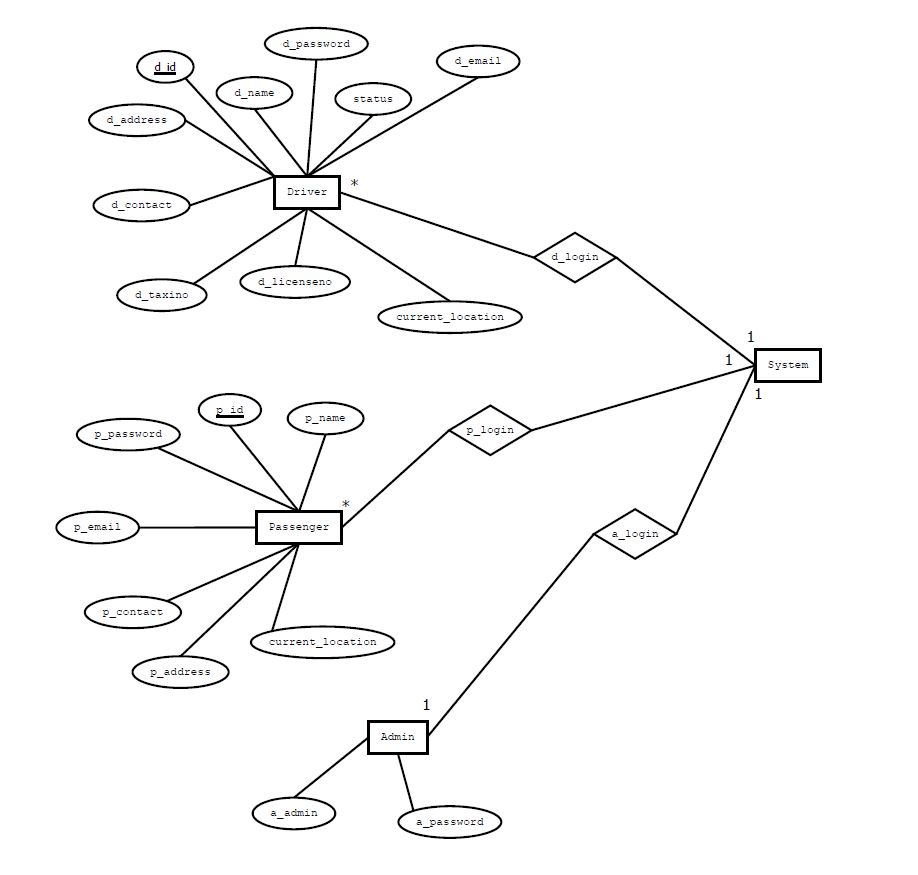


Fig 3.4. E-R Diagram

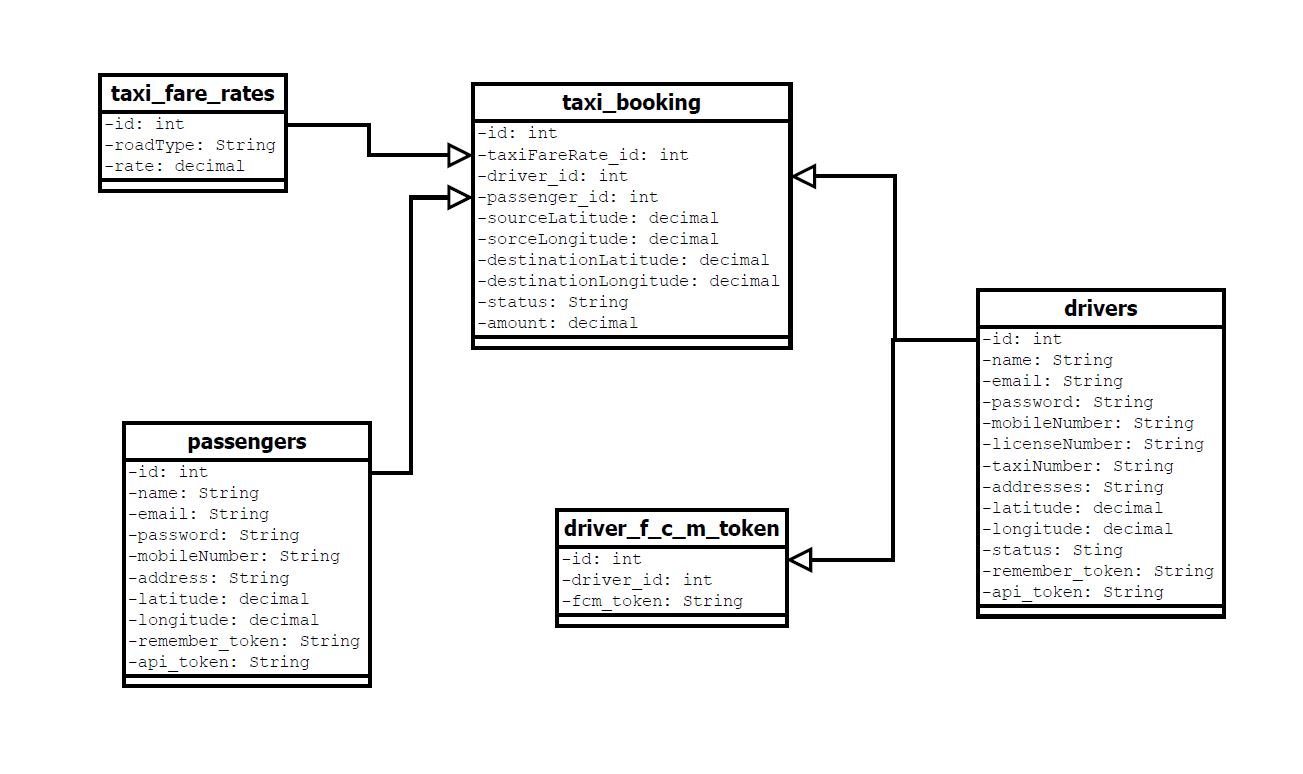


Fig 3.5.Domain Model Diagram

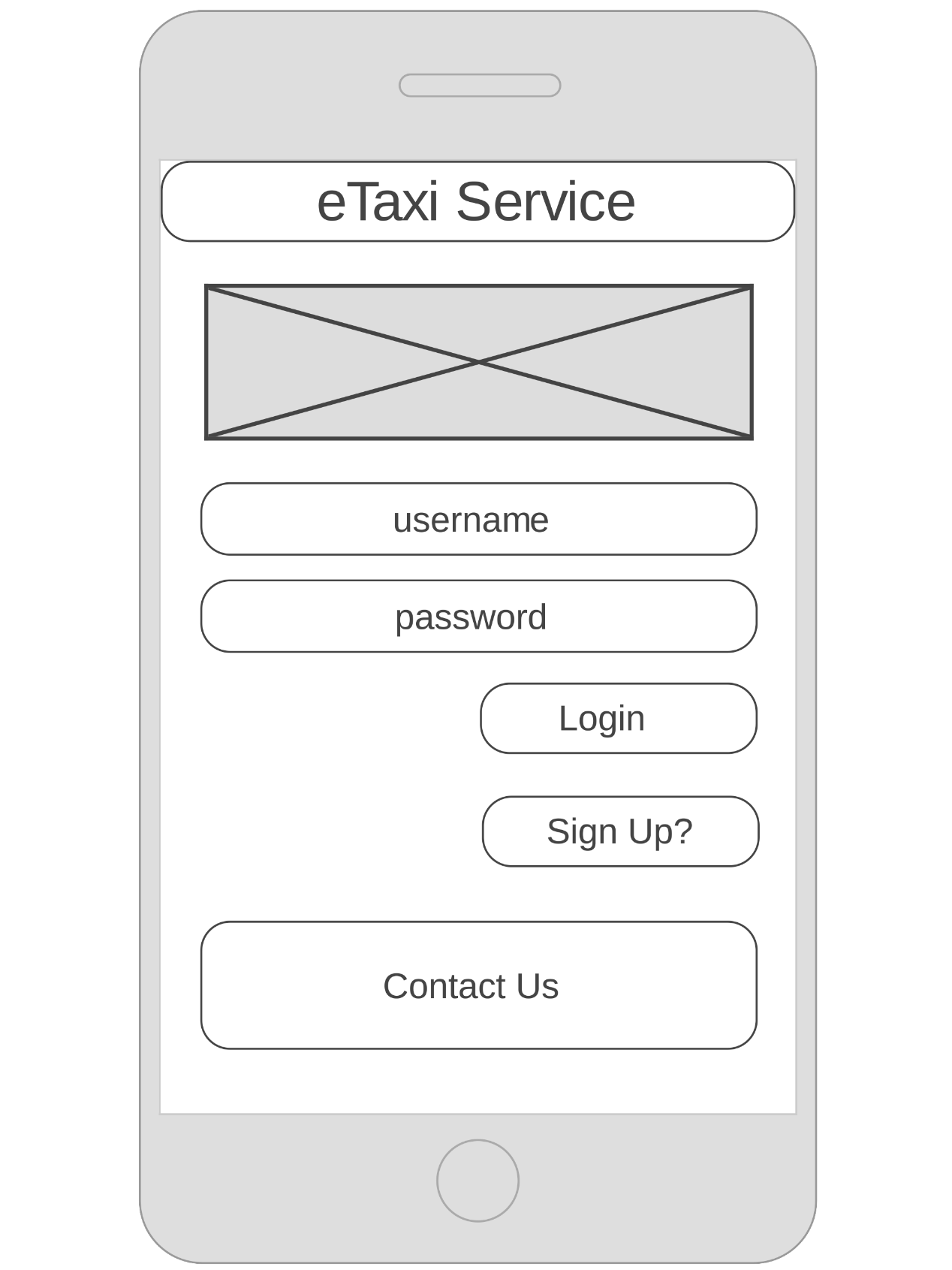


Fig 3.6. eTaxi Service Login & Sign Up Wireframe

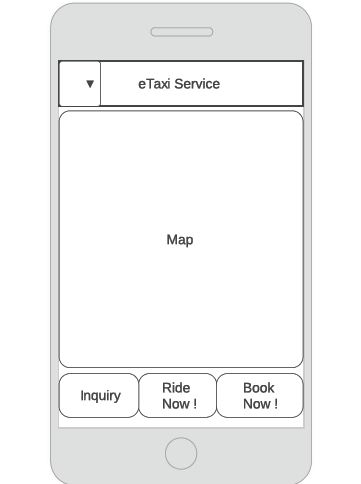


Fig 3.7. eTaxi Service Passenger Dashboard Wireframe



Fig 3.8. Driver Dashboard­ Wireframe

The use case contracts are mentioned below:

**Use Case UC1: d\_login**

**Primary Actor:** Driver

**Stakeholders:**

* **Driver:** Wants to login to the system.

**Postconditions:** Driver sign up the new account by providing full information in the system.

**Basic flow:**

* Driver sign up the new account.
* System provides interface to login the account.
* Drive enters the username and password to login to the system.

**Alternate flow:**

* System provides interface to login the account.
* Drive enters the username and password to login to the system.
* If driver fails to login the system, his username and password is reset or system provides the interface to sign up the new account.

**Use Case UC2: track d\_location**

**Primary Actor:** System

**Stakeholders:**

* **System:** Wants to track the current location of the driver to process the passenger service if that driver is nearby.

**Preconditions:** Driver must login to the system.

**Postconditions:** Driver notifies the system if he is free or busy in providing the service.

**Basic flow:**

* Driver sign up the new account.
* System provides interface to login the account.
* Drive enters the username and password to login to the system.
* System tracks the current location of the driver.

**Alternate flow:**

* System provides interface to login the account.
* Drive enters the username and password to login to the system.
* If driver fails to login the system, his username and password is reset or system provides the interface to sign up the new account.
* System tracks the current location of the driver.

**Use Case UC3: notify\_status**

**Primary Actor:** Driver

**Stakeholders:**

* **System:** Driver notifies the system, if he is free or busy in the service.

**Preconditions:** System tracks the current location of the driver.

**Postconditions:** System notifies the driver is requested for the service by the nearby passenger.

**Basic flow:**

* Drive enters the username and password to login to the system.
* System tracks the current location of the driver.
* Driver notifies the status whether he is free or busy.

**Alternate flow:**

* Drive enters the username and password to login to the system.
* If driver fails to login the system, his username and password is reset or system provides the interface to sign up the new account.
* System tracks the current location of the driver.
* Driver notifies the status whether he is free or busy.

**Use Case UC4: notify driver for the request**

**Primary Actor:** System

**Stakeholders:**

* **System:** Wants to notify the driver regarding requested service.

**Preconditions:** Driver notifies the status if he is free or busy.

**Postconditions:** Driver acknowledges the requested service.

**Basic flow:**

* System tracks the current location of the driver.
* Driver notifies the status whether he is free or busy.
* System notifies driver for passenger’s request.

**Alternate flow:**

* Drive enters the username and password to login to the system.
* If driver fails to login the system, his username and password is reset or system provides the interface to sign up the new account.
* System tracks the current location of the driver.
* Driver notifies the status whether he is free or busy.
* If driver is busy, system finds for another driver and wait for status. If he is free then, it notifies for passenger’s request.

**Use Case UC5: acknowledge requested service**

**Primary Actor:** Driver

**Stakeholders:**

* **Driver:** Wants to acknowledge the notification of the requested service by passenger provided by the system

**Preconditions:** System notifies driver for requested service.

**Postconditions:** Systemprocesses passenger request with nearby driver.

**Basic flow:**

* System tracks the current location of the driver.
* Driver notifies the status whether he is free or busy.
* System notifies driver for passenger’s request.
* Driver acknowledges the system notification of requested service.

**Alternate flow:**

* Drive enters the username and password to login to the system.
* If driver fails to login the system, his username and password is reset or system provides the interface to sign up the new account.
* System tracks the current location of the driver.
* Driver notifies the status whether he is free or busy.
* Driver acknowledges the system notification of requested service.

**Use Case UC6: p\_login**

**Primary Actor:** Passenger

**Stakeholders:**

* **Passenger:** Wants to login to the system.

**Preconditions:** Passenger sign up for the new account in the system.

**Postconditions:** Passenger requests the service to the system.

**Basic flow:**

* Passenger login to the system by providing his/her username and password.

**Alternate flow:**

* Passenger login to the system by providing his/her username and password.
* If passenger fails to login the system, system reset the login details on the request of the passenger or passenger is requested to create new account if he is a new user.

**Use Case UC7: track p\_location**

**Primary Actor:** System

**Stakeholders:**

* **System:** Wants to track the current location of the passenger to process the passenger service with the driver nearby.

**Preconditions:** Passenger must login to the system.

**Postconditions:** Passenger requests service to the system.

**Basic flow:**

* Passenger sign up the new account.
* System provides interface to login the account.
* Passenger enters the username and password to login to the system.
* System tracks the current location of the passenger.

**Alternate flow:**

* System provides interface to login the account.
* Passenger enters the username and password to login to the system.
* If driver fails to login the system, his username and password is reset or system provides the interface to sign up the new account.
* System tracks the current location of the passenger.

**Use Case UC8: request service**

**Primary Actor:** Passenger

**Stakeholders:**

* **Passenger:** Wants to request the taxi service in the system.

**Preconditions:** Passenger must login to the system.

**Postconditions:** System processes the passenger request with the nearby driver.

**Basic flow:**

* Passenger enters the username and password to login to the system.
* System tracks the current location of the passenger.
* Passenger requests for the taxi service.

**Alternate flow:**

* Passenger enters the username and password to login to the system.
* If driver fails to login the system, his username and password is reset or system provides the interface to sign up the new account.
* System tracks the current location of the passenger.
* Passenger requests for the taxi service.

**Use Case UC9: process passenger request with nearby driver**

**Primary Actor:** System

**Stakeholders:**

* **System:** Wants to process the passenger service with the driver nearby.

**Preconditions:**

* System tracks the current location of the passenger.
* Passenger must request service to the system.

**Postconditions:** Passenger chooses the taxi.

**Basic flow:**

* Passenger enters the username and password to login to the system.
* System tracks the current location of the passenger.
* Passenger requests the service to the system.
* System processes the passenger service request with nearby driver.

**Alternate flow:**

* System provides interface to login the account.
* Passenger enters the username and password to login to the system.
* If driver fails to login the system, his username and password is reset or system provides the interface to sign up the new account.
* System tracks the current location of the passenger.
* Passenger requests the service to the system.
* System processes the passenger service request with nearby driver.

**Use Case UC10: choose taxi**

**Primary Actor:** Passenger

**Stakeholders:**

* **Passenger:** Wants to choose the taxi from the list of available taxi provided by the system.

**Preconditions:** Passenger process the passenger request with nearby driver.

**Postconditions:** System notifies the passenger the arrival of the taxi service.

**Basic flow:**

* Passenger enters the username and password to login to the system.
* System tracks the current location of the passenger.
* Passenger requests for the taxi service.
* System processes the requested service with the nearby driver.
* Passenger chooses the taxi from the result of the list of taxi.

**Alternate flow:**

* Passenger enters the username and password to login to the system.
* If driver fails to login the system, his username and password is reset or system provides the interface to sign up the new account.
* System tracks the current location of the passenger.
* Passenger requests for the taxi service.
* System processes the requested service with the nearby driver.
* Passenger chooses the taxi from the result of the list of taxi.

**Use Case UC11: notify passenger for the arrival of taxi service**

**Primary Actor:** Driver

**Stakeholders:**

* **Driver:** Wants to notify the passenger about the arrival of the taxi.

**Preconditions:** Passenger chooses the taxi.

**Postconditions:** Passenger makes the payment.

**Basic flow:**

* System tracks the current location of the driver and passenger.
* Passenger requests the service.
* System processes the passenger service request with nearby driver.
* Passenger chooses the taxi.
* Driver notifies the passenger about the arrival of the taxi

**Alternate flow:**

* System tracks the current location of the driver and passenger.
* Passenger requests the service.
* System processes the passenger service request with nearby driver.

**Use Case UC12: monitor**

**Primary Actor:** Admin

**Stakeholder:**

**-Admin:** control the system by monitoring the databases of driver and passenger

**Basic Flow:**

-Admin login into the web interface

-Admin control the databases of the driver, passenger, booking, etc.

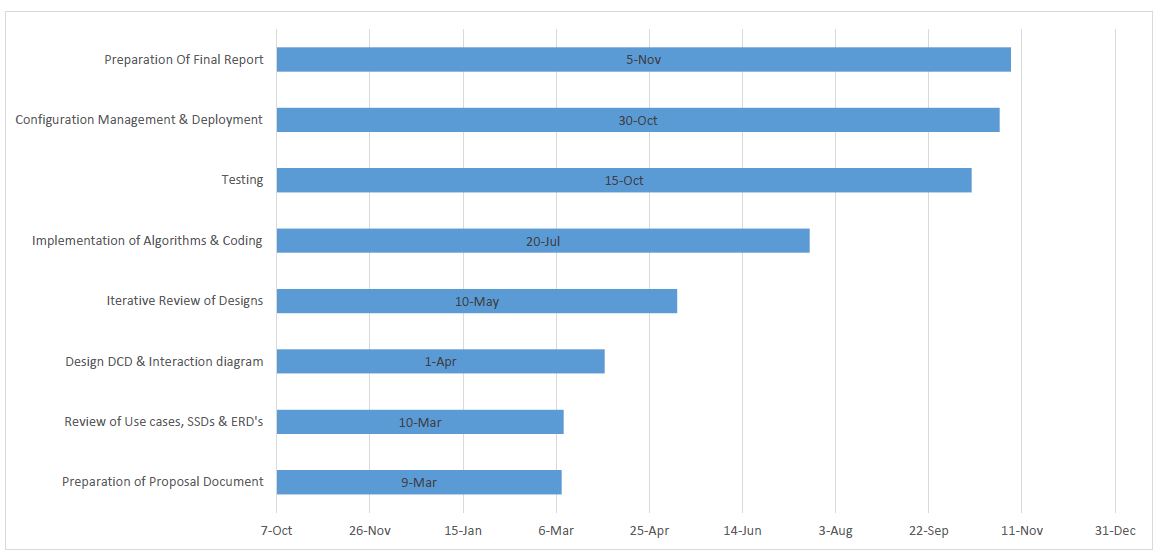
Chapter 4

**TIME LINE CHART**

The time line chart of the project is estimated from the start date 6th March, 2017 to the end date 6th, November, 2017. The time line chart is mentioned below with the bar diagram in Fig 1.5

|  |  |  |
| --- | --- | --- |
| **Task** | **Start Date** | **Days to Complete** |
| Inception of Idea | 24-Feb | 4 |
| Feasibility Testing | 28-Feb | 5 |
| Preparation of Concept Document | 3-Mar | 1 |
| Requirements & Analysis | 4-Mar | 3 |
| Designing Use cases, SSDs & ERDs | 7-Mar | 2 |
| Preparation of Proposal Document | 9-Mar | 1 |
| Review of Use cases, SSDs & ERD's | 10-Mar | 20 |
| Designing Domain diagrams, DCD & Interaction diagram | 1-April | 40 |
| Iterative Review of Designs | 10-May | 70 |
| Implementation of Algorithms & Coding | 20-July | 95 |
| Testing | 15-Oct | 15 |
| Configuration Management & Deployment | 30-Oct | 5 |
| Preparation Of Final Report | 5-Nov | 3 |

Fig 4.1. Timeline Chart



**BIBLIOGRAPHY**

*Gantt*: (November 16, 2015). In *Wikipedia.* Retrieved October 9, 2015, from [*https://en.wikipedia.org/wiki/Gantt\_chart*](https://en.wikipedia.org/wiki/Gantt_chart)

*Wikipedia:* The free encyclopedia. (2004, July 22). FL: Wikimedia Foundation, Inc. Retrieved October 9, 2015, from [*http://www.wikipedia.org*](http://www.wikipedia.org)

*Google:* The search engine. (2000). Google.com Inc. Retrieved October 9, 2015, from [*http://google.com.np*](http://google.com.np)

*Lifewire:* It teaches you how to best use your screens and gadgets, have fun with technology, fix what isn’t working, and buy the best gear for your life, from *https://www.lifewire.com/uber-app-review-3486481*

*Engadget:* It is the definitive guide to this connected life. From *https://www.engadget.com/2014/06/27/uber-lyft-explainer/*

*https://play.google.com/store/apps/details?id=me.lyft.android&hl=en*

*Insider Monkey:* Insider Monkey uses a hybrid evaluation system that exploits insider transactions and other market anomalies to reduce the number of insiders who are worthy enough to monkey. From *http://www.insidermonkey.com/blog/uber-lyft-gett-juno-via-best-nyc-taxi-alternatives-515032/?singlepage=1*

*Onver:* Smart Taxi For Smart People*.* From *https://www.onver.com.np/*

*Eddy Cab:* [*https://www.eddycab.com/np/*](https://www.eddycab.com/np/)