# Project Log Book

**Group Members:**

| Entry Date | Work Done |
|---|---|
| September 7th, 2000 | Discussed the basic plan to build the prototype for CRM in class, noting down all constraints to be taken care of. Furthermore, we decided our next group meeting would be on September 15th, 2000 (Friday) at 5:30, meeting place: Zaida Morales' House. |
| September 15th, 2000 | Meeting at Zaida's Place: We discussed about the project objective. Using the Software Management Plan template printed from the web site, we stepped through each section and discussed what was required and what resources were available to us. We also discussed how this prototype should be flexible for other countries. There was constant reference to the "Chinese Railway Passenger Reservation System" and other related articles. |
| September 16th, 2000 | Finished a rough draft prototype and set it up on the online account. |
| September 19th, 2000 | Zaida M. Morales checked the document of the Software Project Management Plan, and she made some correction marking the corrections in red. |
| September 20th, 2000 | The mistakes were corrected on the web site, and email was sent to Zaida M. morales to check the document for any more mistakes |
| September 20th, 2000 | The document was checked by Zaida M. Morales and few more mistakes were found. These mistakes were corrected and put on the web. |
| September 22th, 2000 | Meeting at Zaida's Place: We discussed the Reservation System in more detail and added more information to the SPMP document. |
| September 25th, 2000 | Zaida M. Morales checked the document of the Software Project Management Plan, and she made some corrections. |
| September 27th, 2000 | The mistakes were corrected on the web site, and email was sent to Zaida M. morales to check the document for any more mistakes. |
| September 29th, 2000 | Meeting at Zaida's Place: We discussed parts 4 and 5 of the Software Project Management Plan in more detail and decided to update some information in the SPMP document. The different parts of the document were divided between the team for updates. |
| October 3th, 2000 | Finished updating the rough draft prototype and set it up on the online account. Sent all team members email with link to latest copy of the document. |
| October 4th, 2000 | Zaida M. Morales checked the document of the Software Project Management Plan. The mistakes were corrected on the web site. The latest version of the document is available online. |

***Last Updated on Novemeber 7th, 2000***

# Software Requirements Specification

*for*

## *Cab Management System*

***M.Hanif Hasan 9003***
***Ali Asghar Karani 8999***
***Muhammad Waseem 9039***
***Abdul Rafay 9353***
***Muhammad Ali***

***December 04, 2020***

| Version | Changes Made | Date |
|---|---|---|
| 1.0 | First Pass for Review | 10/24/2020 |
| 1.2 | Second Pass for Review | 11/07/2020 |
| 1.3 | Third Pass for Review | 11/28/2020 |
| 1.4 | CRM Review Version | 12/04/2020 |

# Table of Contents

# 1. Introduction

## 1.1 Purpose

This document describes the software requirements for the Automated Railroad Reservation System built for the Chinese Railway Ministry (CRM).

## 1.2 Scope In

The CRM is requesting proposals to build a prototype of an Automated Railroad Reservation System (ARRS) for their current system. This new ARRS needs to be scalable enough so that it can accommodate the increase in reservations caused by new railroad building in China.

The system will be designed to provide an electronic version of the railway passenger reservation system in China. The system will have a user-friendly graphical interface and will be more cost effective compared to the current non-electronic version of the reservation system.

The **objectives** of this development effort are:

1. To provide existing clerks with a new environment in which to make reservations for railroad travel.
2. To provide an avenue for customers to get their tickets in a more convenient way.
3. To regain control of the railway ticket sales to avoid scalping and overselling of tickets.
4. To implement a prototype of a scaled down version of the final system to test the solution and further develop requirements.
5. To collect statistics in a more efficient manner for future railroad development and construction.
6. To increase efficiency of railroads.

## 1.3 Scope Out

The following features will not be the part of this Project:

1.

## 1.3 Definitions, Acronyms, and Abbreviations.

APPM – AsiaPac Marketing Manager
ARRS – Automated Railroad Reservation System
CASE – Computer Aided Software Engineering
CITS – China International Travel Agency
CRM – Chinese Railroad Ministry
PP - Project Plan
SDD - Software Design Description

SRS - Software Requirement Specification
SDS – Software Design Specification
SPMP - Software Project Management Plan
GUI – Graphical User Interface
QAM – Quality Assurance Manager
PDM – Project Development Manager
PMP – Project Management Professional
TBD – To be determined
UML – Unified Modeling Language

### 1.4 References


- Situation Update – Chinese Railway Passenger Reservation System http://www.cs.swt.edu/~donshafer/Marketing Update(1).html
- China 2000 http://www.china2thou.com
- Pressman, Roger S., *Software Engineering: A Practitioner's Approach*, McGraw-Hill Companies, Inc., 1997.


### 1.5 Overview

Chapter 2 of the SRS is a brief description of the characteristics of the software to be built, its functions, its users, its constraints and its dependencies.

Chapter 3 is about specific requirements, such as functional requirements, external interface requirements, performance requirements, and also design constraints and quality characteristics.

Finally, chapter 4 includes all the supporting information, such as the Table of Contents, the Appendices, and the Index.

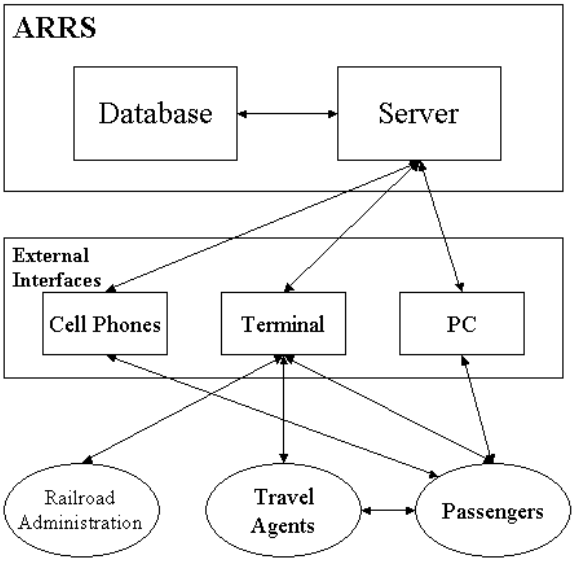
# 2. The General Description

This section describes the general factors that affect the product and its requirements. This section consists of five subsections that follow. This section does not state specific requirements. Each of the subsections makes those requirements easier to understand, it does not specify design or express specific requirements. Such detail is provided in section 3.

### 2.1 Product Perspective

The Automated Railway Reservation System diagram showing the overview of the system's modules and the relationship of the system to external interfaces is presented in Figure 2.1.

**Figure 2.1 Overview/Architecture Diagram of the ARRS**




**Functions of System Components:**

Database:

- Stores data
- Creates reports
- Provides access to data
- Updates information

Server:

- Provides access to the database
- Authenticates users
- Processes reservations
- Performs backups
- Produces reports

**External Interfaces:**

Terminal

- Users use terminals to access the server
- Passengers and travel agents use terminals to reserve the tickets and to get information about the available seats on particular trains.
- Railroad administration may use terminals to see the reports generated by the database software.

Personal Computers

- Users (passengers, travel agents, and railroad administration) may use personal computers to obtain a remote access to the server and the reservation database via the Internet.

Cell Phones

- Serve as a medium of accessing the server and the reservation database.
- Passengers may use cell phones and the latest telecommunication technologies to access the server and the reservation database via Internet, or they may use cell phones to call travel agents to inquire about railroad and ticket information.

Computer Hardware and Peripheral Equipment to be used:

- 30 workstations, which include CPUs, monitors, keyboards, and mice
- Printers
- Network
- Terminals
- Cell phones to test connection to the server via remote access

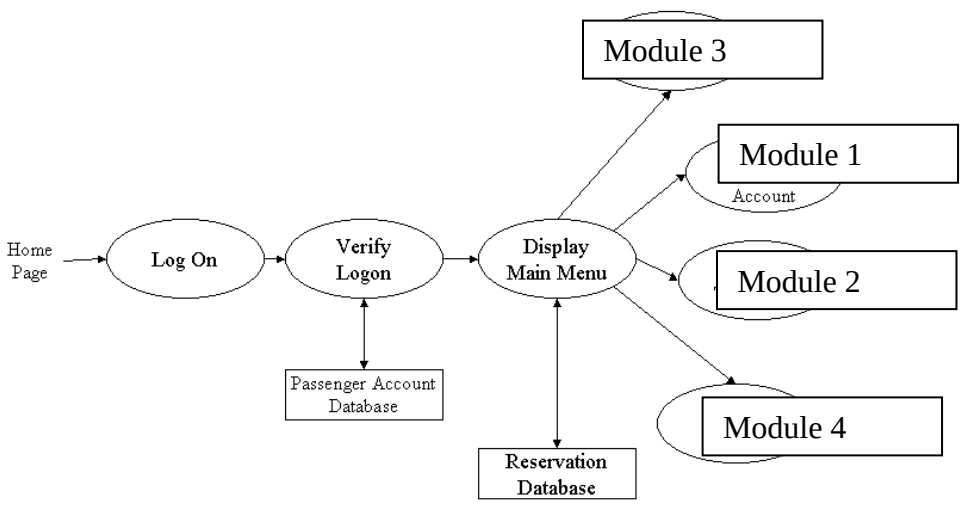
## 2.2 Product Functions

This section provides a summary of the functions that the software will perform.

### 2.2.1 Function Relationships

Figure 2.2 to 2.6 depict the relationships among the functions to be implemented by the system.

Figure 2.2 ARRS General Function Relationship/**Higher Level Usecase** Diagram



### 2.2.2 Function Descriptions (Functional Requirement Listings)

#### 2.2.2.1 Log In Function

***Description:*** This function ensures that only authorized users gain access to the Reservation databases. An authorized user is a user who has an account on the system. Users include passengers, train officials, and CRM ministry officials. The user must type a valid username and password to gain access.

#### 2.2.2 Module 1: ~~Make Reservation Function~~

***Description:*** This function allows the user to [Make | Drop | View | Update] a reservation for a particular train on a particular date for a certain number of tickets.

If the user does not already have a reservation, then all reservations are dropped. If the user already has a previous reservation, a chosen reservation is dropped from the list of current reservations, and the passenger account balance gets updated.

#### 2.2.3 Module 2:

***Description:*** This function

#### 2.2.4 Module 3:

***Description:*** This function

#### 2.2.5 Module 4:

***Description:*** This function

#### 2.2.6 Module 5:

***Description:*** This function

### 2.3 User Characteristics

The main users of the system will be the passengers buying train tickets, the travel agents that process reservations for passengers, and the CRM administration that access the reports generated by the system. The users are not required to have knowledge in the computer field. The graphical interface provides an easy way of using the ARRS system with minimum of training.

### 2.4 General Constraints

The constraints for the project are:

- The functional prototype should be available after 30 days upon the arrival of the management team to China. This may prove to be a serious time constraint on the development of a successful prototype.
- Communication with the Chinese team members may prove to be difficult since some Chinese developers do not speak English and the management team does not speak Chinese. Even with the presence of a translator, communication may be difficult. Absence of the translator may severely affect project development.
- Team members are restricted from bringing their own equipment, and insufficient equipment supply may hinder project development.
- Team members are restricted to bringing only the analysts of the team to China. This might affect the project development if more people are needed or the required skills are not available.
- The majority of the Chinese population does not have or have a limited access to the Internet.

**2.5 Assumptions and Dependencies or Business Logic**

The assumptions for the project are:

- Ten trains transport the passengers between three cities known as Guangzhou, Shanghai and Nanjing. These trains originate only in cities Guangzhou and Shanghai, and they make a stop at Nanjing before arriving to their destination.

- There are five classes of tickets as listed below
    - Sleeping (soft) - compartment style coaches - 4 passenger per compartment
    - Sleeping (hard) - compartment style coaches - 6 passenger per compartment

- Reservation can be made up to one month before a particular trip.
- Seats are assigned during reservation.
- Phone reservation involves tickets being purchased within 24 hours after making the reservation. Otherwise, the reservation will be cancelled.
- No reservations can be made 48 hours prior to the trip. Rather, it will be done on a first come first serve basis from that point on.
- Passenger lists will be provided for conductors at each stop.
- The expected reservations during test period may amount to approximately 25,000 per day. This volume varies by hour, day, and season.

- Chinese Ministry will provide us with information about identification process used in China, so that it can be applied to the reservation system and scalping of tickets is avoided.
- Network connection will always remain established.

# <ADD OOAD REPORT DIAGRAMS HERE>

## 3. Specific Requirements

This section of the SRS contains design requirements for the Cab Management System.

### 3.1 Functional Requirements

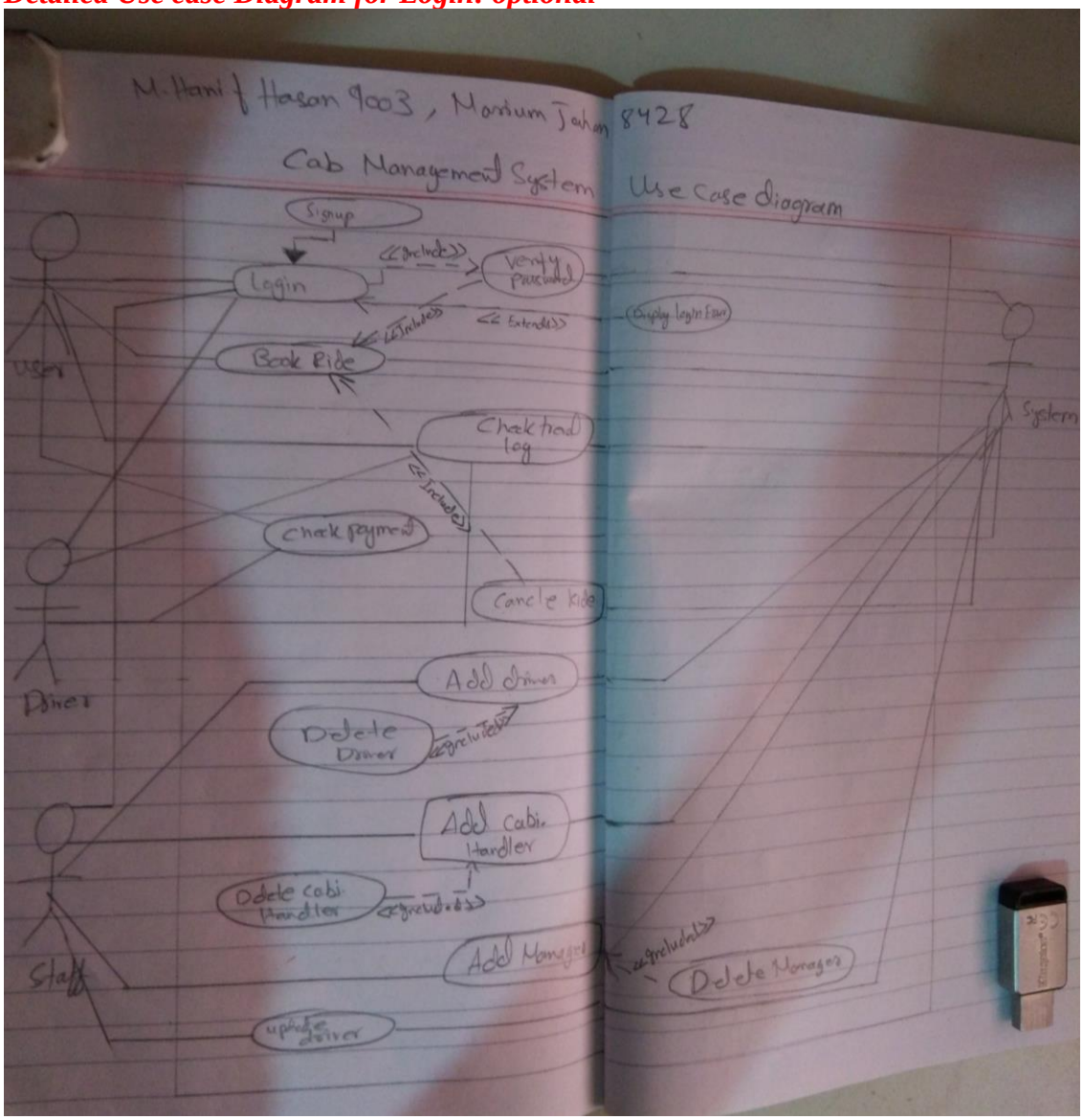
#### 3.1.1 Log In Function

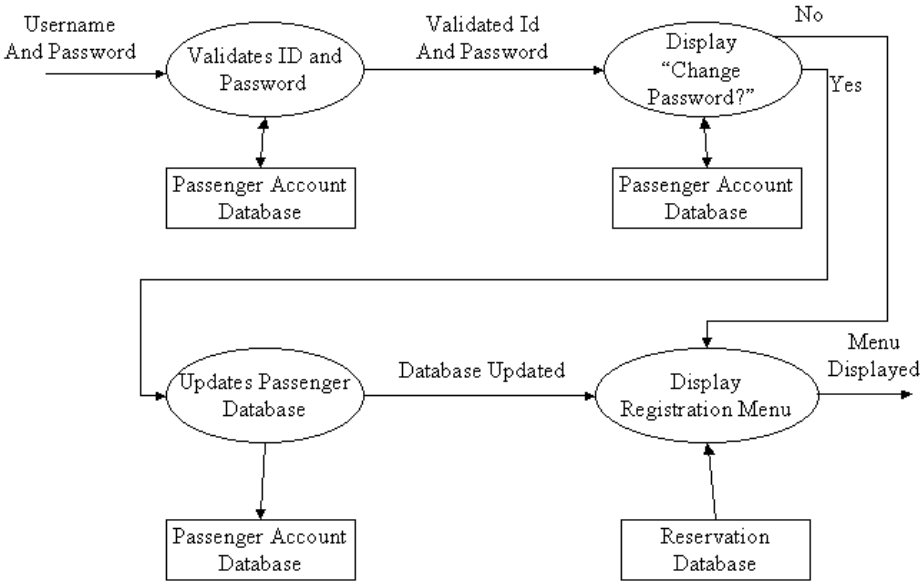
a) ***Description:*** This function ensures that only authorized users gain access to the Reservation databases. An authorized user is a user who has an account on the system. Users include Admins. The user must type a valid username and password to gain access.

b) ***Usage Scenario/ Use case Description/ Specification:***

| | |
|---|---|
| **Description** | Allows access to online CMS |
| **Inputs** | Username, password |
| **Source** | 1. User inputs username and password<br>2. Press Login Button |
| **Alternate case** | |
| **Outputs** | Successful login; unsuccessful login |
| **Destination** | None |
| **Precondition** | Authorized Admin |
| **Post Condition** | No change to Passenger Accounts Database |
| **Side Effects** | Failures and successful logins are sent to Reservation Database |

### c) *Detailed Use case Diagram for Login: optional*





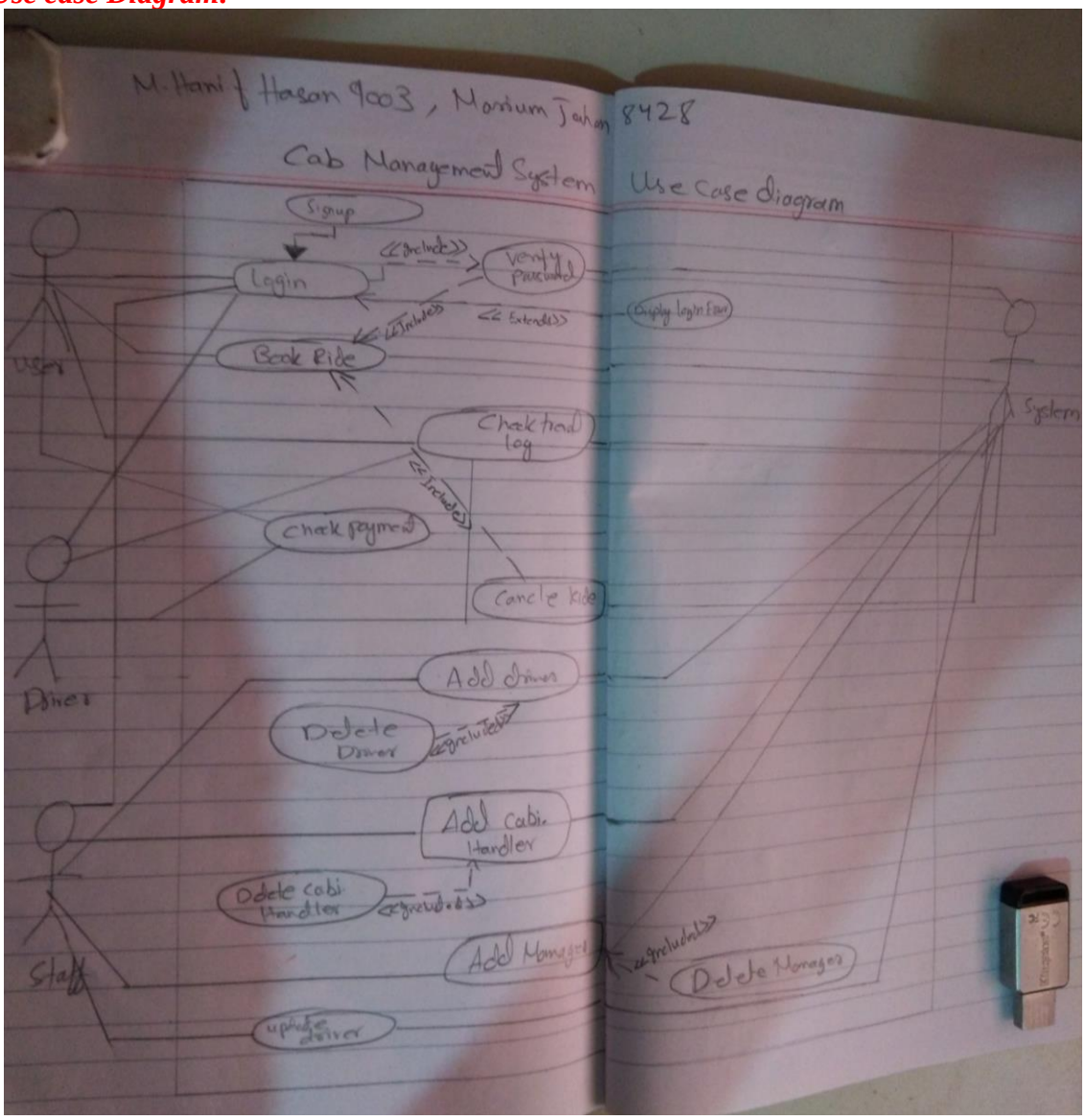

### 3.1.2 Module 1 complete CRUD on Registration and signin

a) ***Description:*** This function allows the admin or user to [make | drop | view | update] a Module Admin Accounts.

b) ***Usage Scenario/ Use case Description/ Specification:***

| | |
|---|---|
| **Description** | [ make \| drop \| view \| update] to the Admin accounts |
| **Inputs** | Username and passwords |
| **Source** | 1. User inputs from city, to city, seat type, travel date, return date and time<br>2. Press Button … |
| **Alternate Case** | |
| **Outputs** | Added \| Deleted \| Viewed \| Modified Accounts |
| **Precondition** | If they already have a account they can simply login , if they don't have a account they register. |
| **Post Condition** | On signing up added to Users account table |

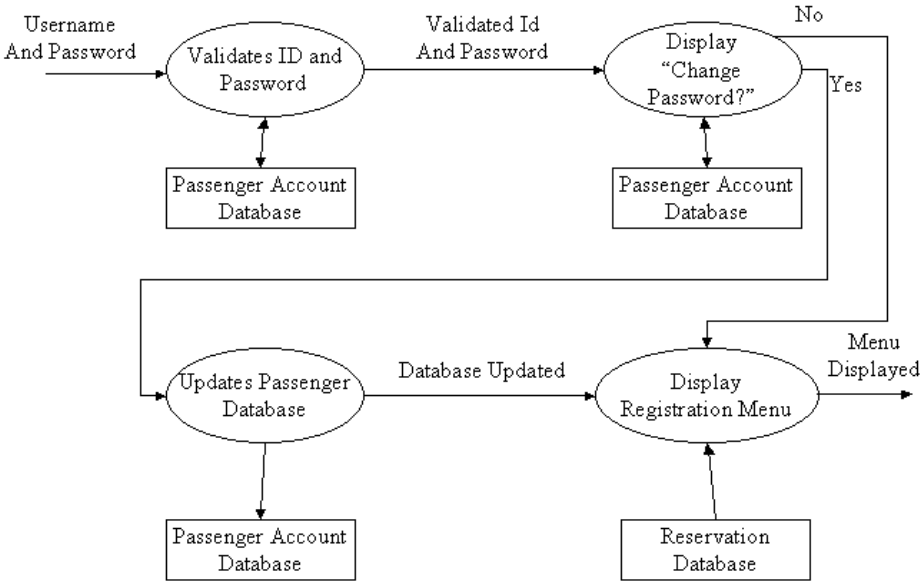
### *c) Use case Diagram:*



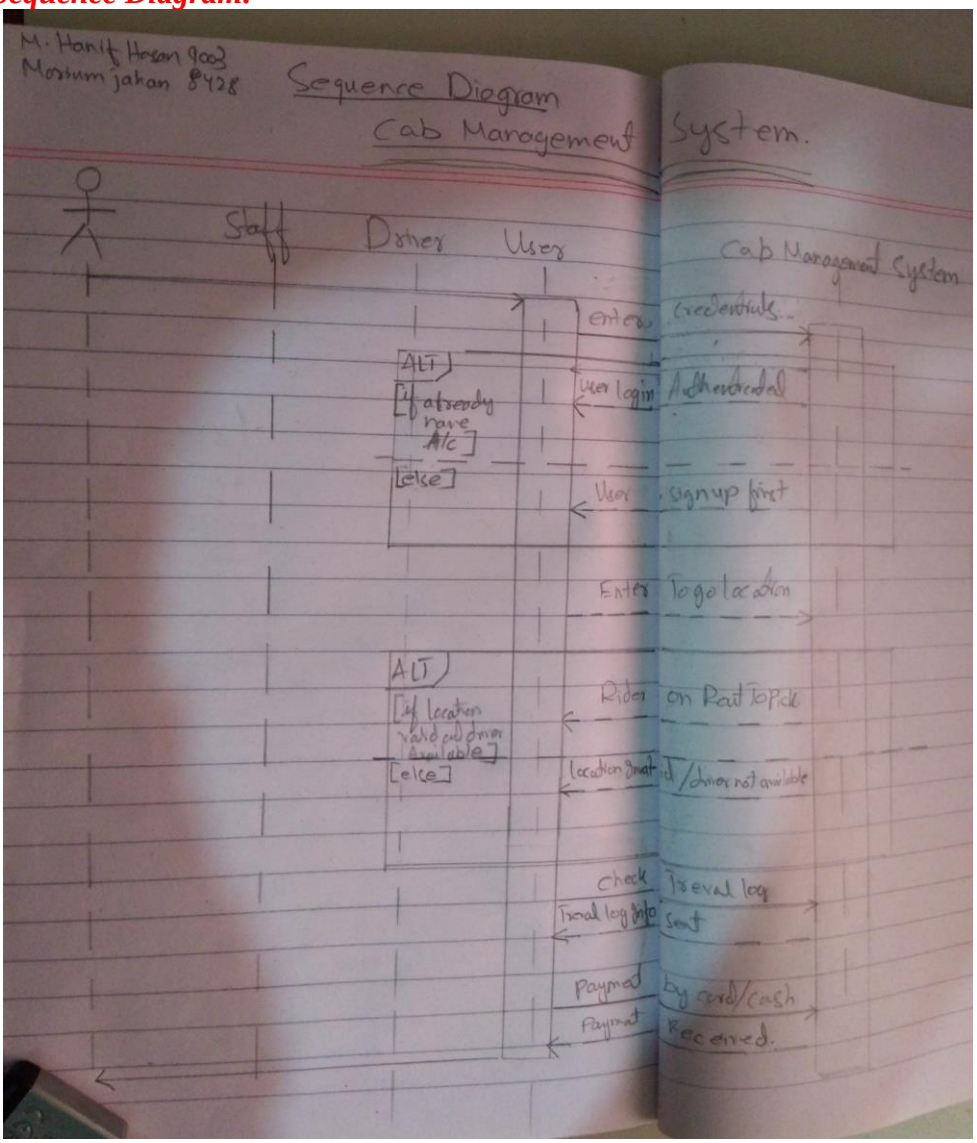

### *d) Use case Realization:*

The realization was not needed.

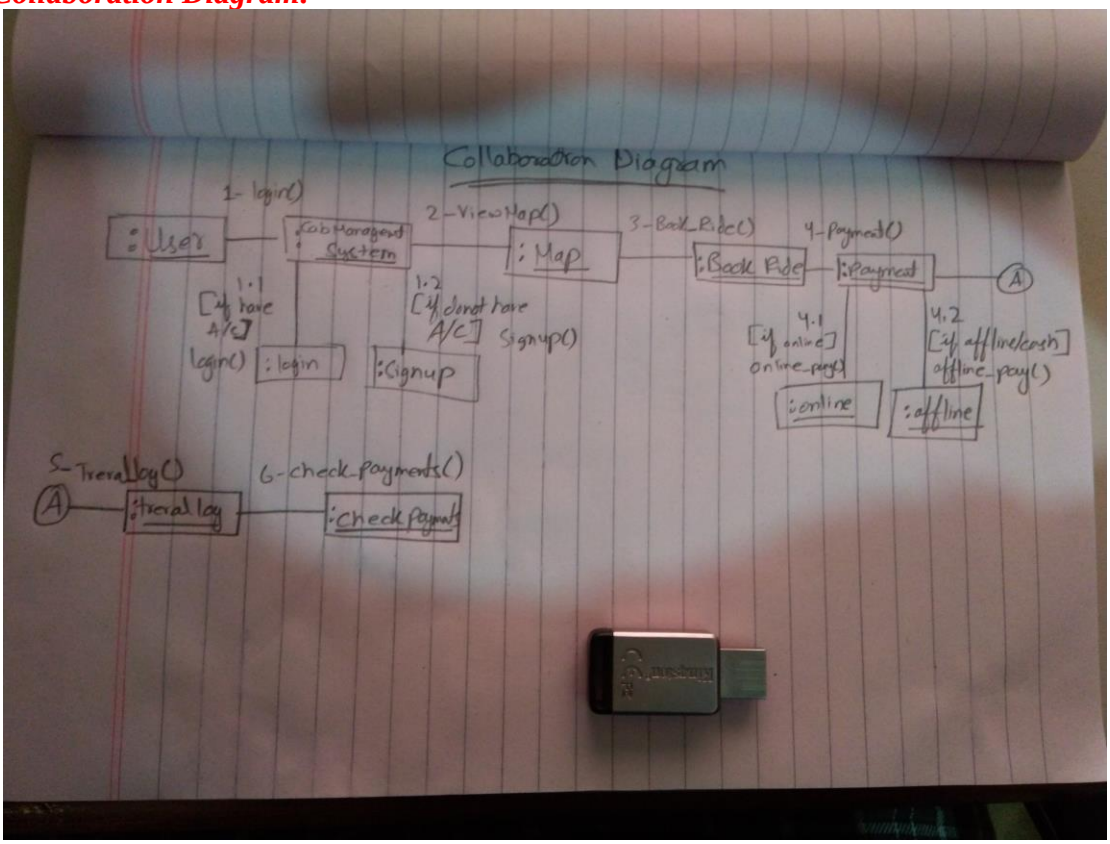
### *e) Flow of Event or Data Flow Diagram:*


Username
And Password
Validates ID and
Password
Validated Id
And Password
Display
"Change
Password?"
No
Yes
Passenger Account
Database
Passenger Account
Database
Updates Passenger
Database
Database Updated
Display
Registration Menu
Menu
Displayed
Passenger Account
Database
Reservation
Database

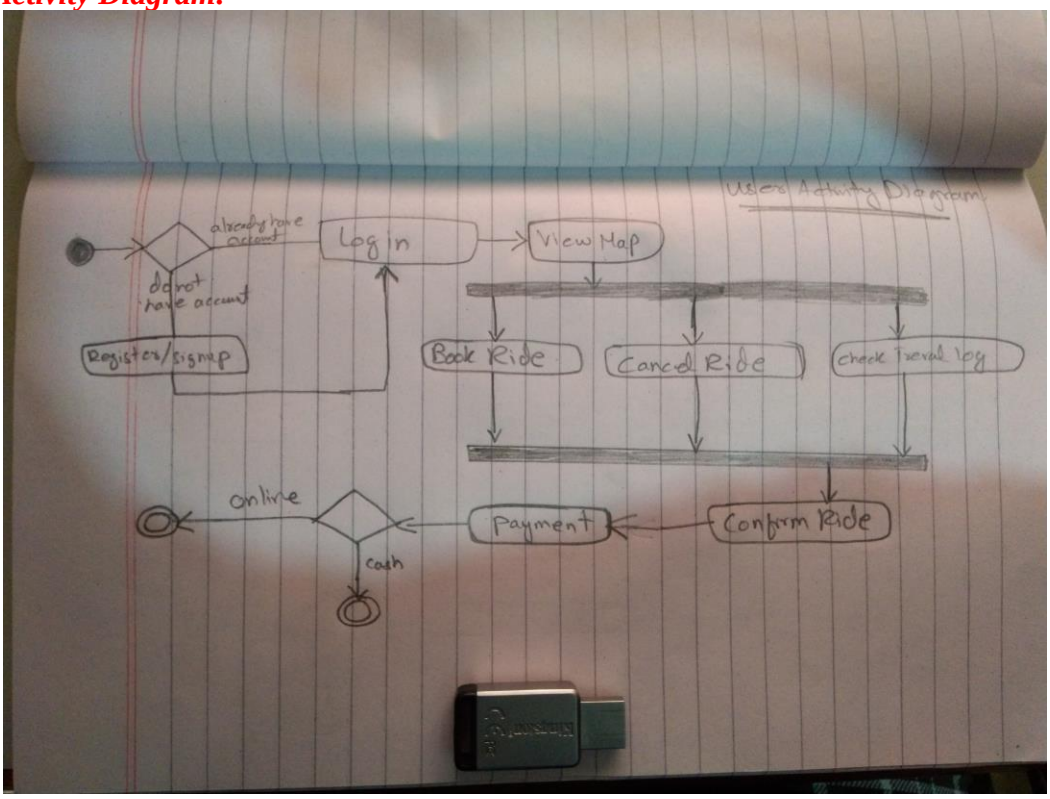
f) ***Sequence Diagram:***



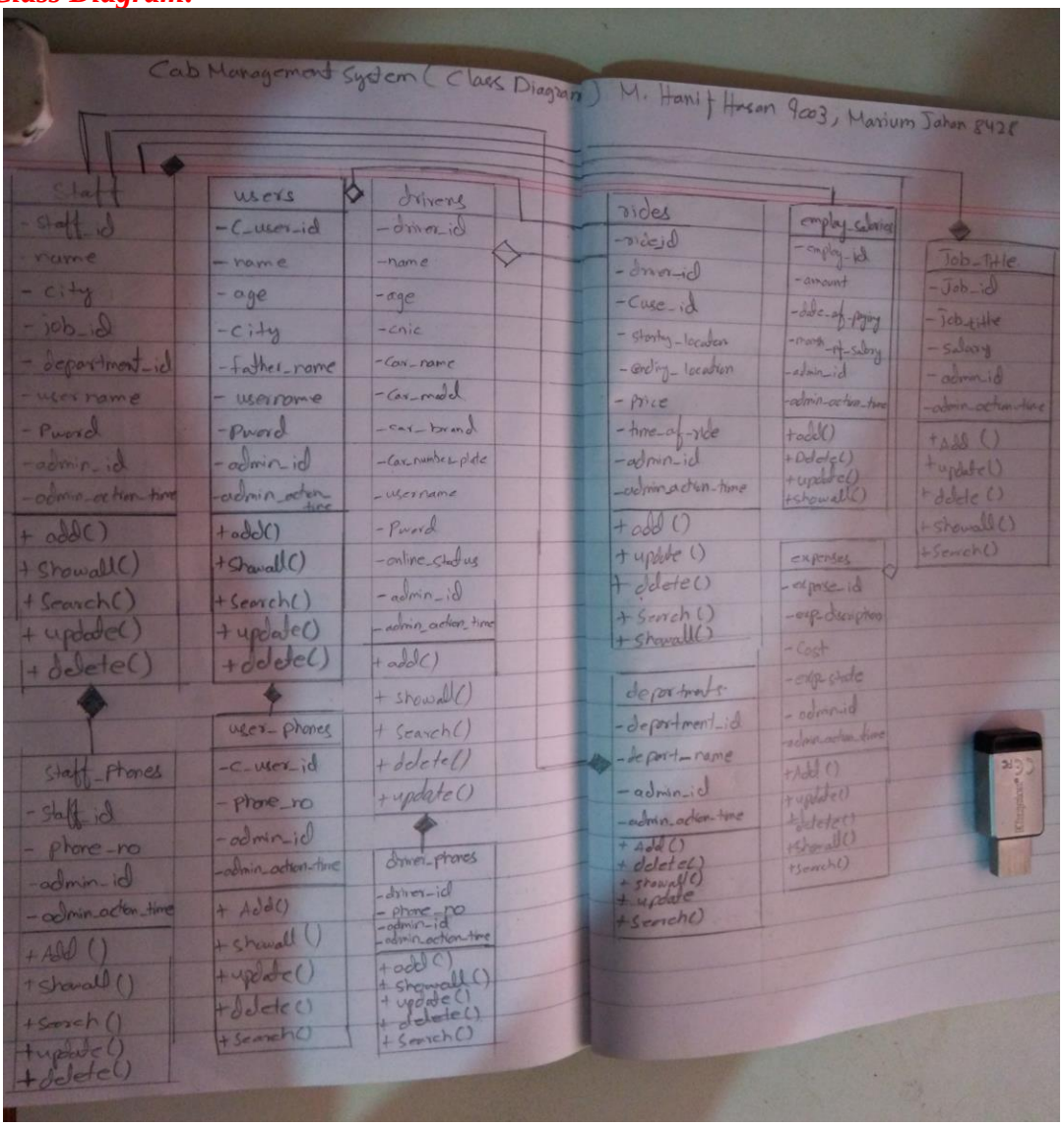
## *g) Collaboration Diagram:*



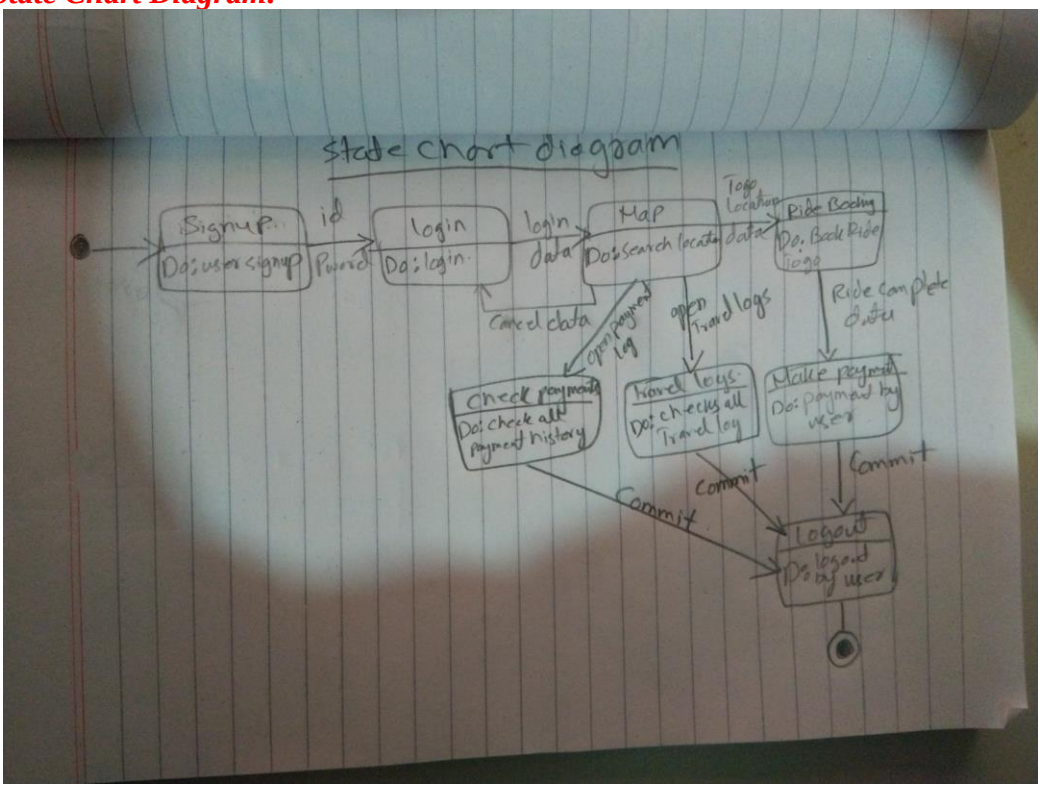

## *h) Activity Diagram:*



## *i) Class Diagram:*



*j) State Chart Diagram:*



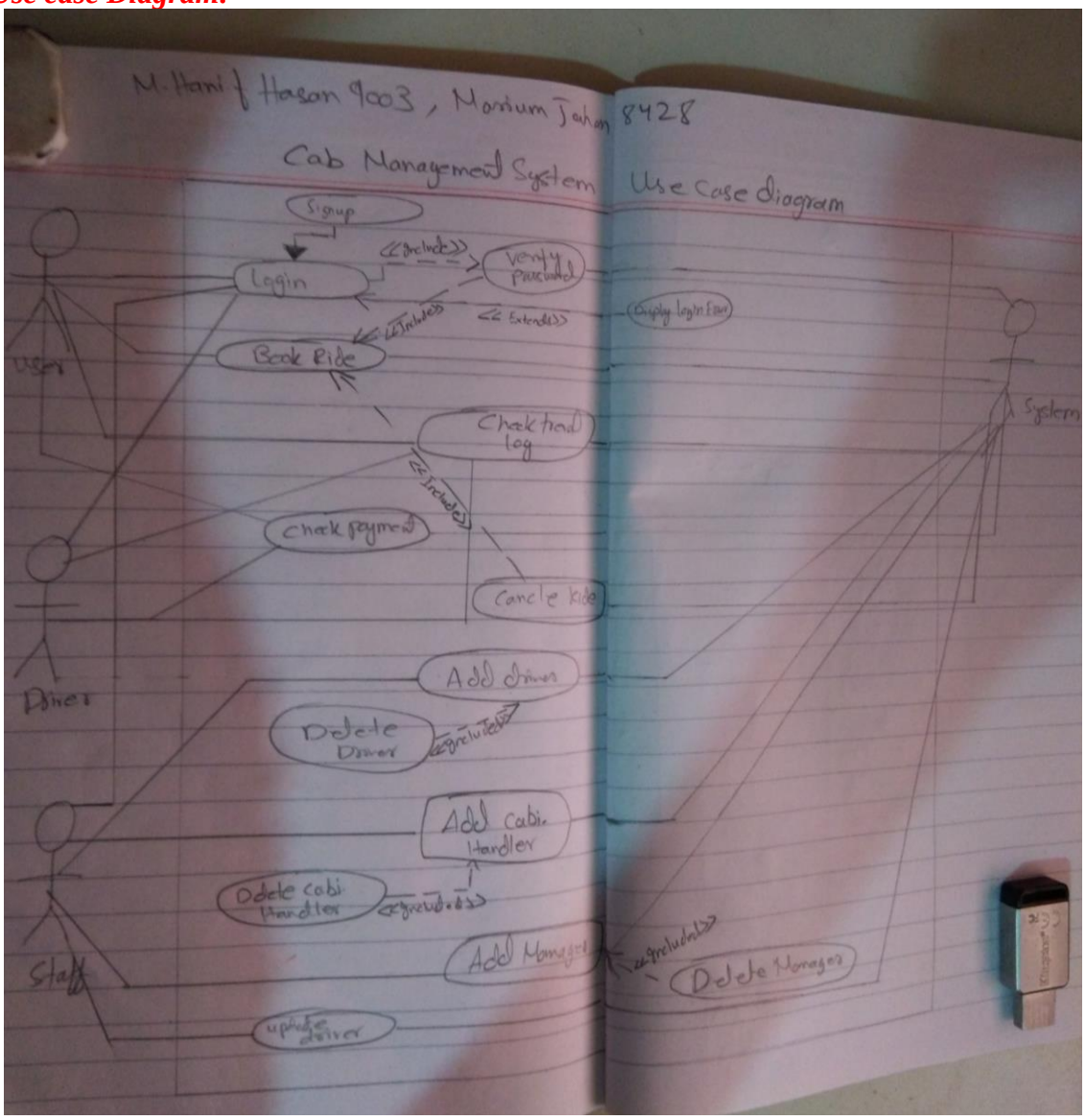

### 3.1.3 Module 2 complete CRUD Staff

a) ***Description:*** This modules Add,update,delete Staff Users.

k) ***Usage Scenario/ Use case Description/ Specification:***

| | |
|---|---|
| **Description** | [ make \| drop \| view \| update] to the user's accounts |
| **Inputs** | Username and passwords |
| **Source** | 3. User inputs from city, to city, seat type, travel date, return date and time<br>4. Press Button … |
| **Alternate Case** | |
| **Outputs** | Added \| Deleted \| Viewed \| Modified Accounts |
| **Precondition** | If they already have a account they can simply login , if they don't have a account they register. |
| **Post Condition** | On signing up added to Users account table |

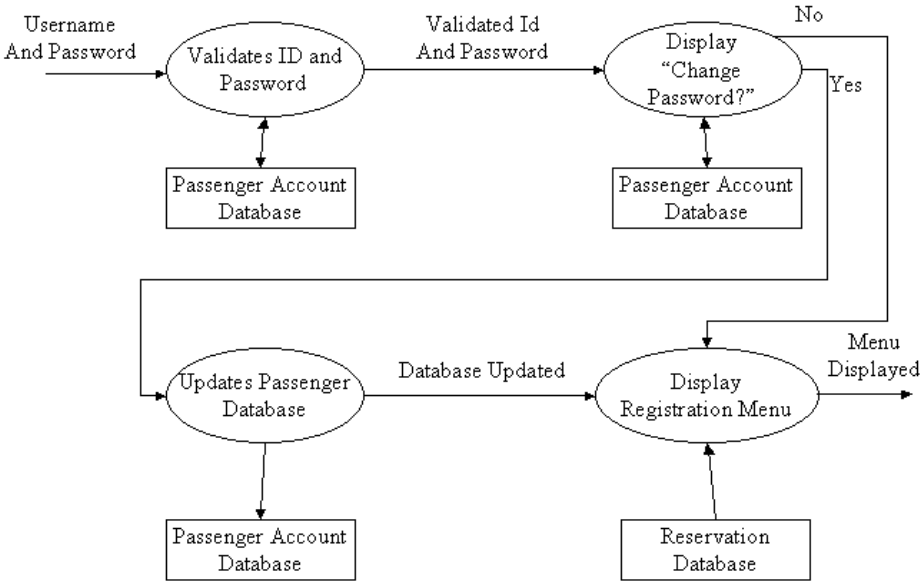
***l) Use case Diagram:***



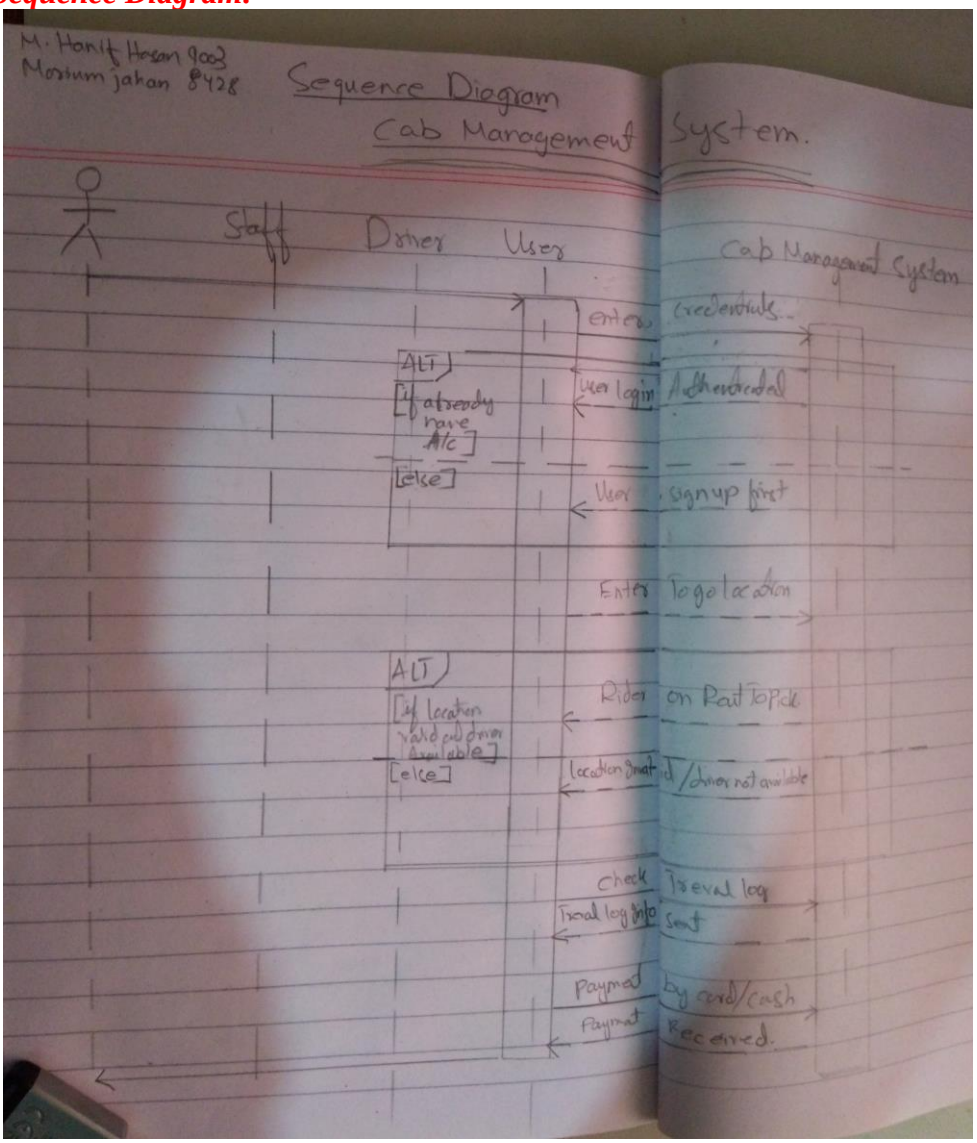

***m)Use case Realization:***

The realization was not needed.

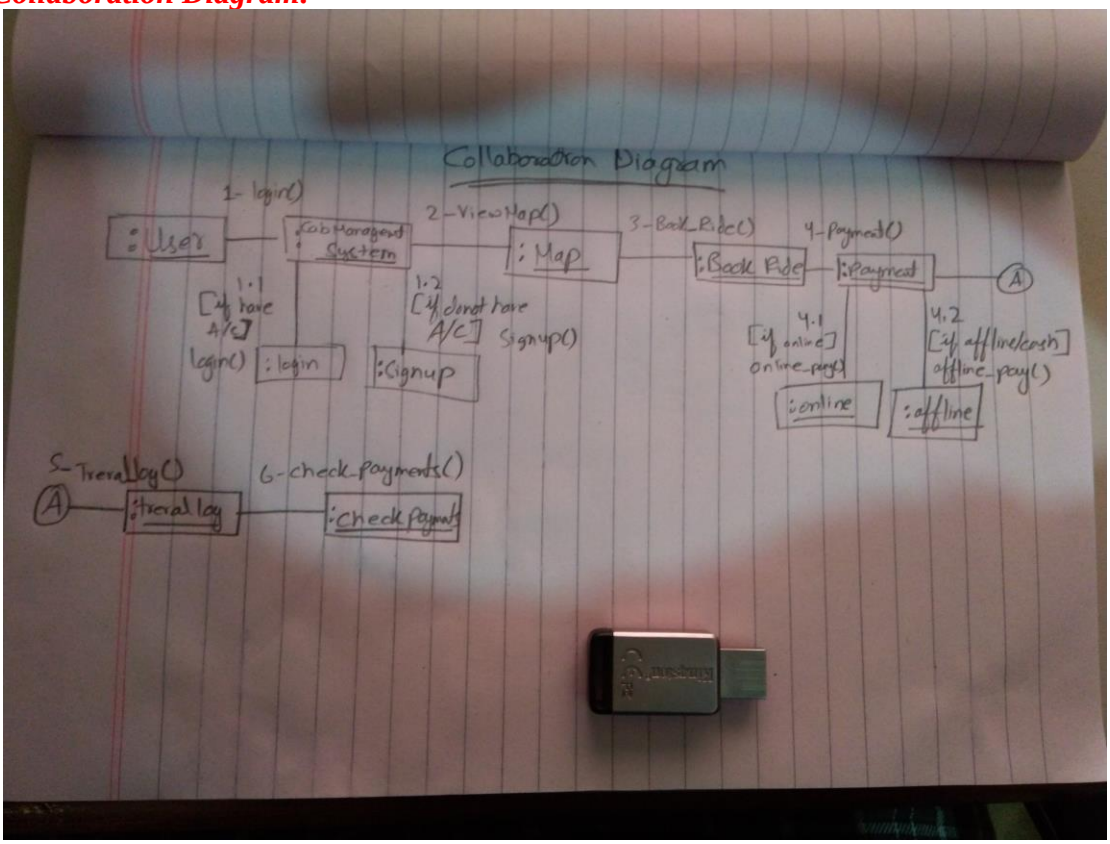
***n) Flow of Event or Data Flow Diagram:***


Username
And Password
Validates ID and
Password
Validated Id
And Password
Display
"Change
Password?"
No
Yes
Passenger Account
Database
Passenger Account
Database
Updates Passenger
Database
Database Updated
Display
Registration Menu
Menu
Displayed
Passenger Account
Database
Reservation
Database

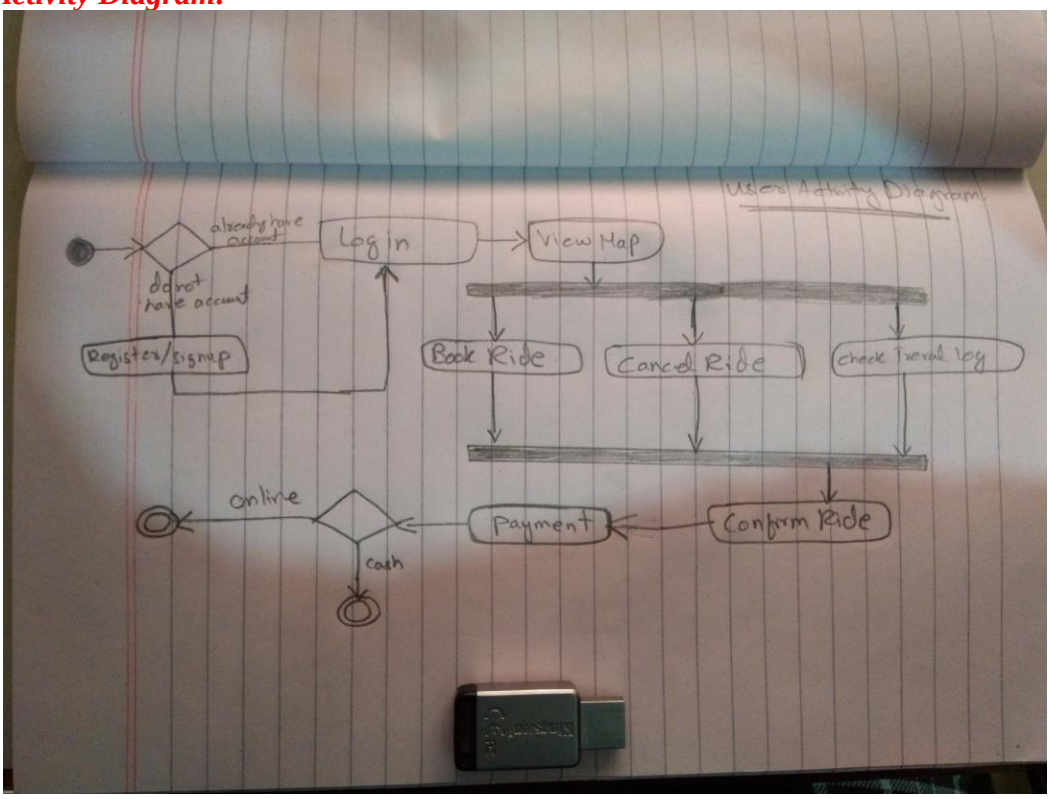
o) ***Sequence Diagram:***



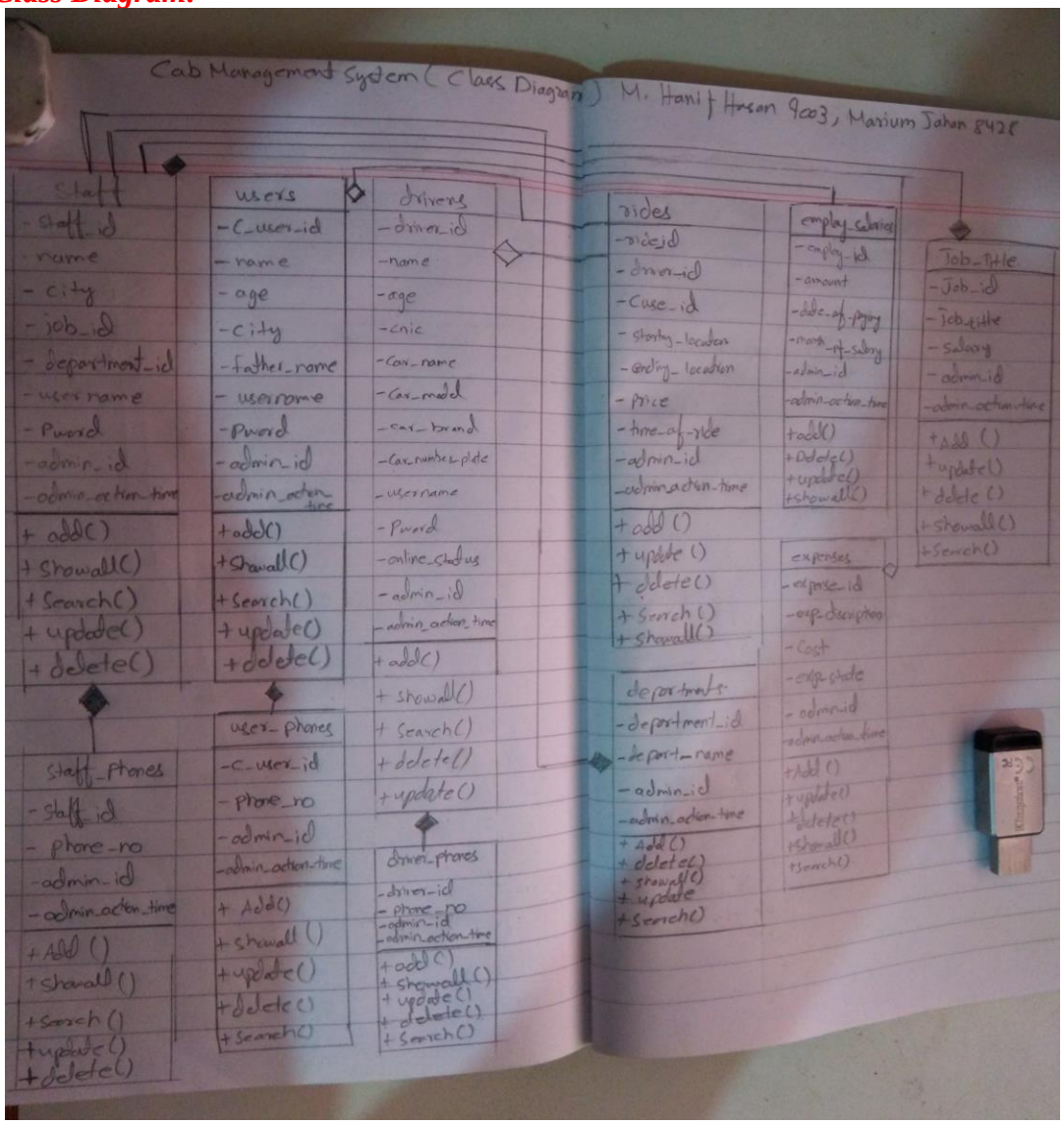
## *p) Collaboration Diagram:*



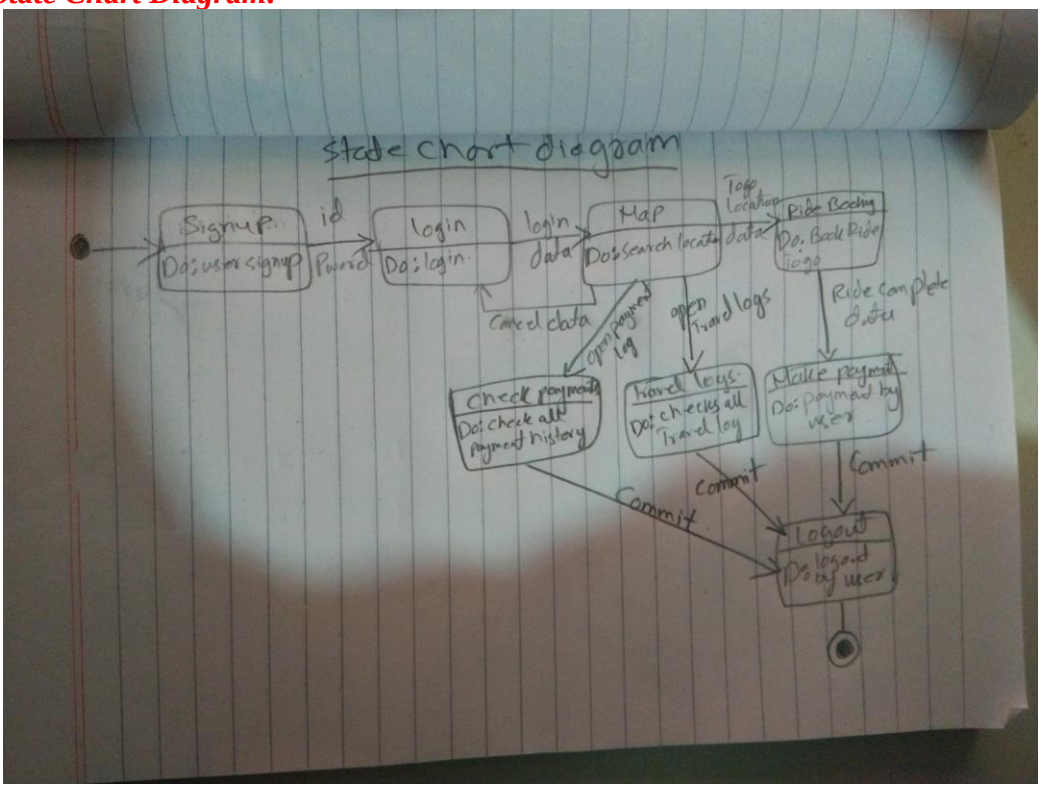

## *q) Activity Diagram:*



## r) Class Diagram:



***s) State Chart Diagram:***



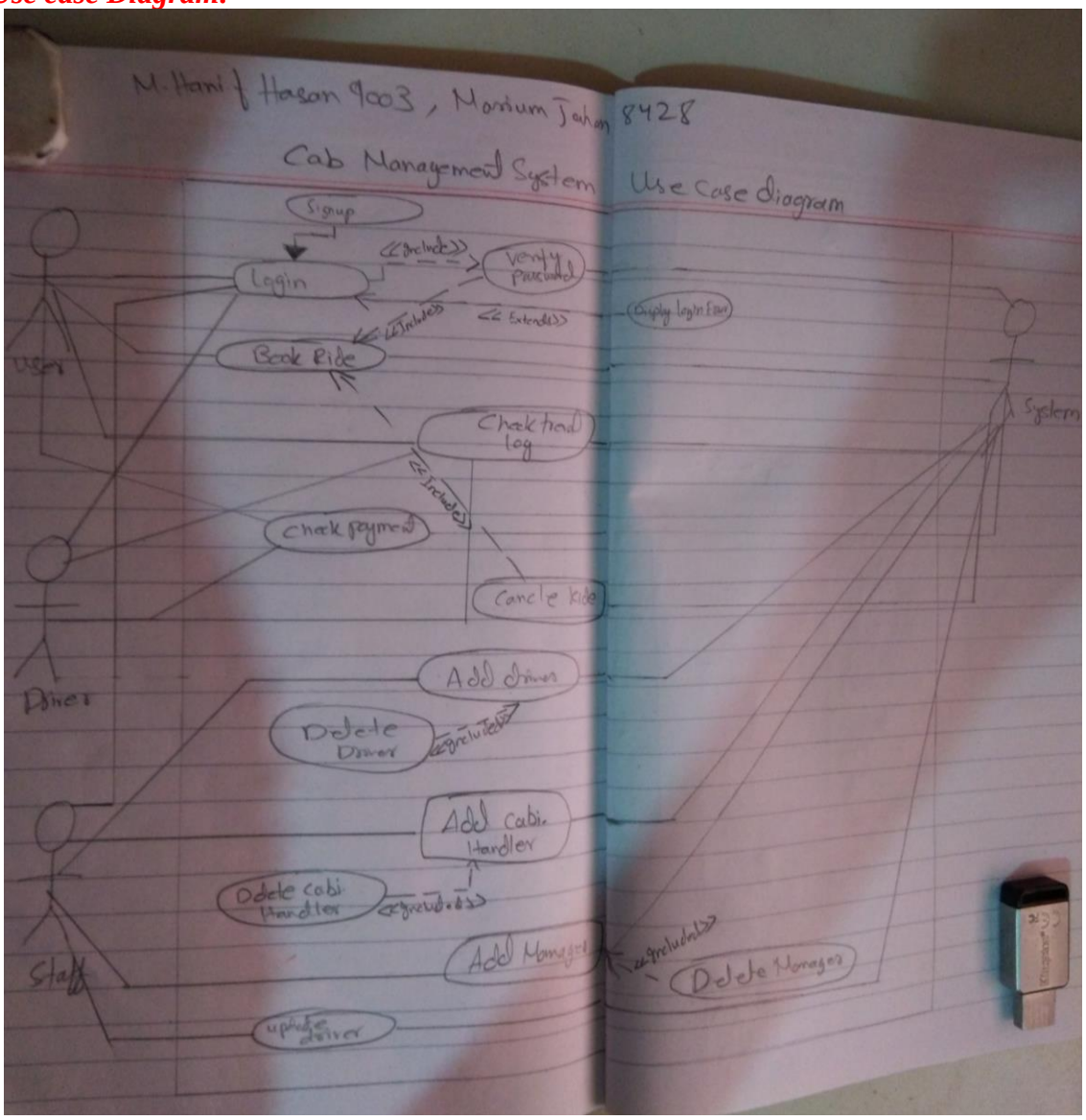

### 3.1.4 Module 3 complete CRUD Users

b) ***Description:*** This modules Add,update,delete Users.

t) ***Usage Scenario/ Use case Description/ Specification:***

| | |
|---|---|
| **Description** | [ make \| drop \| view \| update] to the user's accounts |
| **Inputs** | Username and passwords |
| **Source** | 5. Inputes user name .phone..etc<br>6. Press Button … |
| **Alternate Case** | |
| **Outputs** | Added \| Deleted \| Viewed \| Modified Users |
| **Precondition** | If login via module admin |

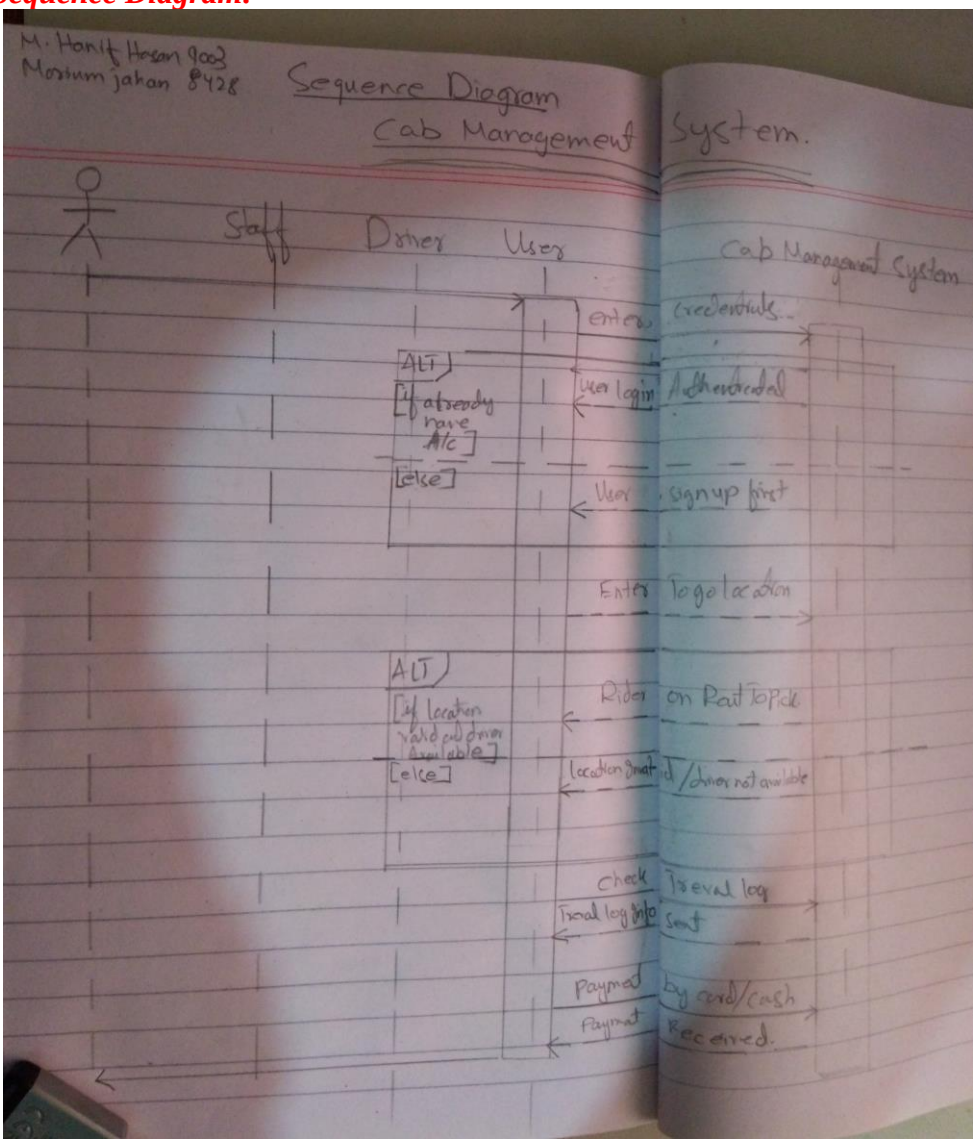
***u) Use case Diagram:***



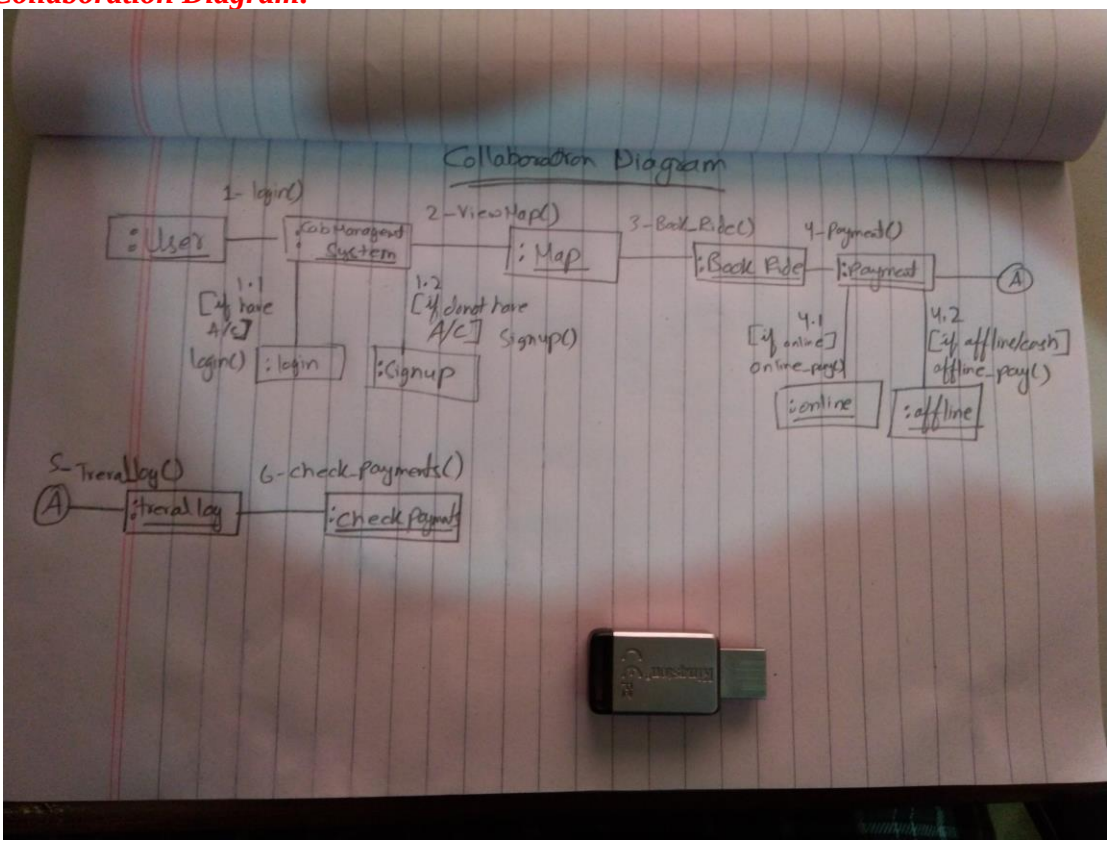

***v) Use case Realization:***

The realization was not needed.

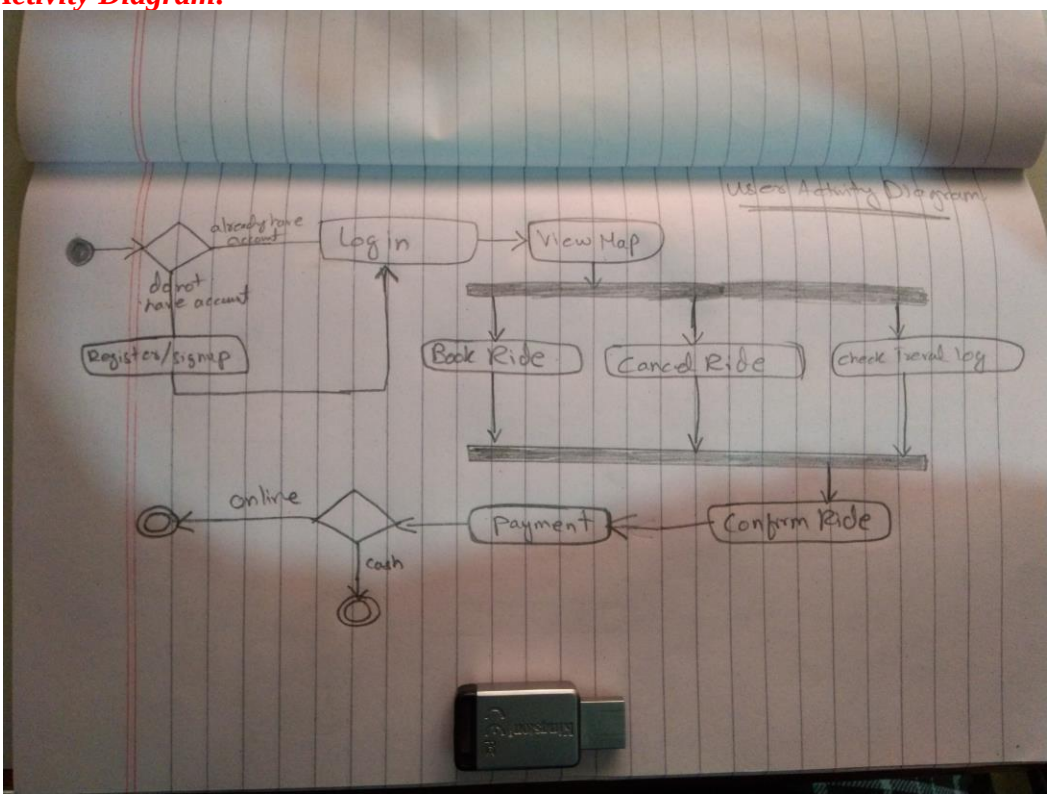
w) ***Sequence Diagram:***



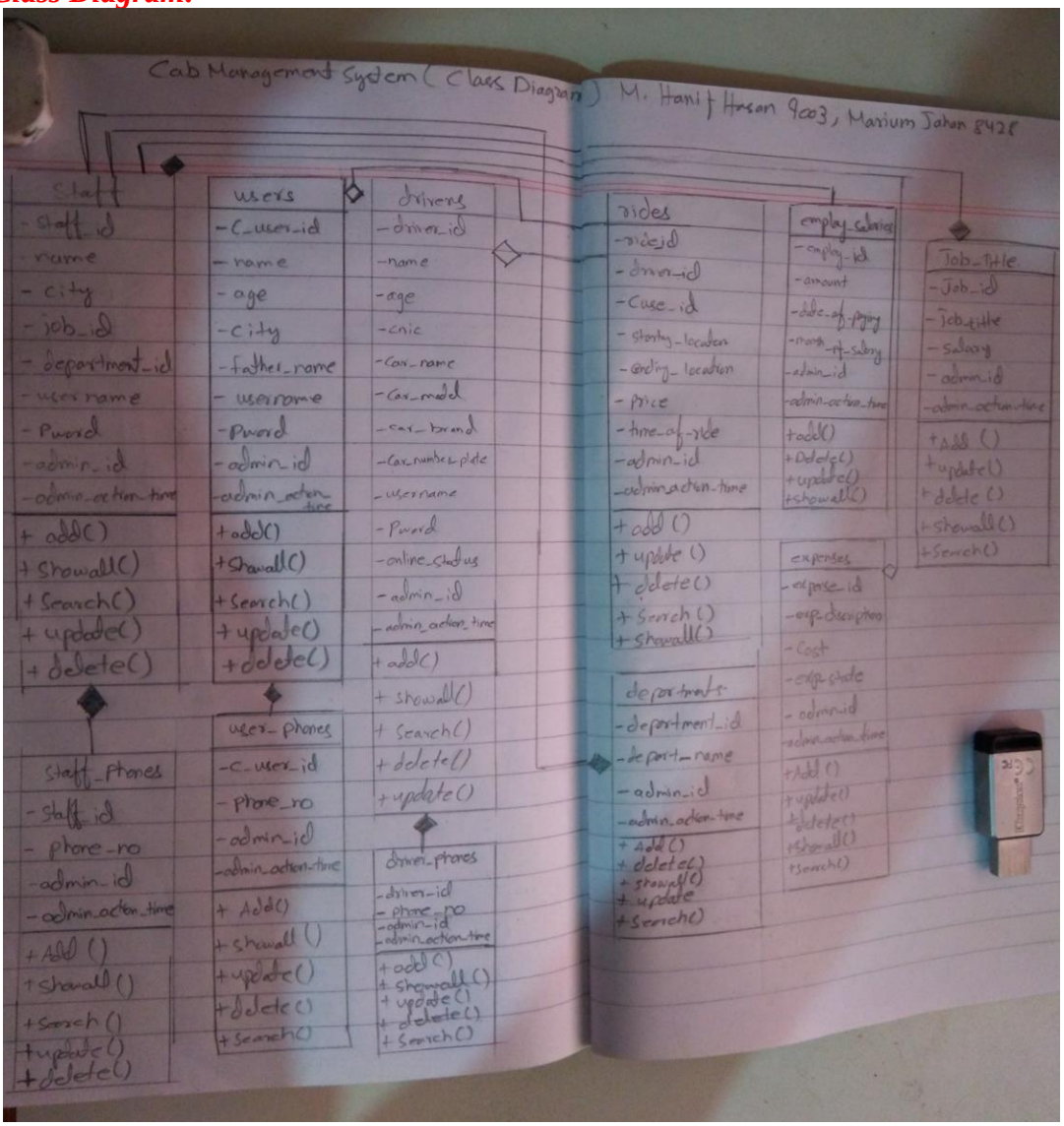
## *x) Collaboration Diagram:*



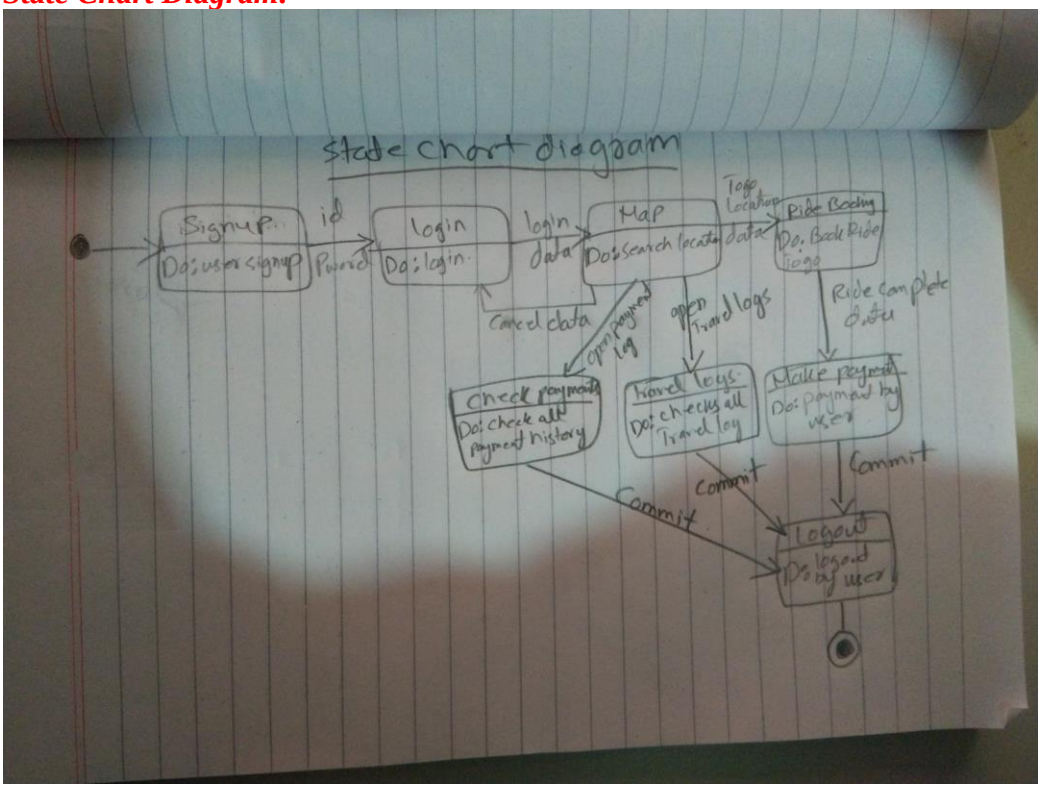

## *y) Activity Diagram:*



## *z) Class Diagram:*



***aa) State Chart Diagram:***



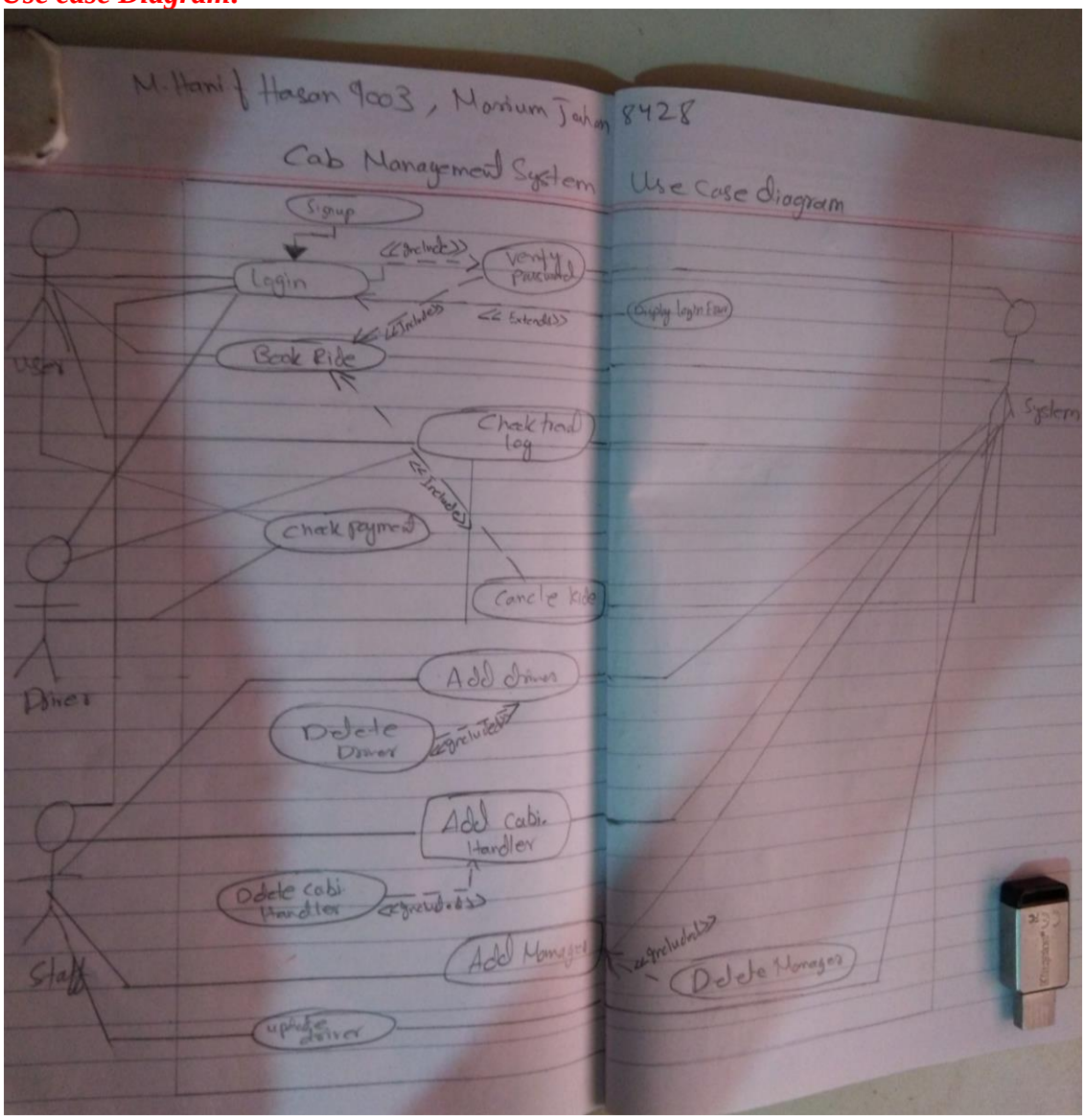

### 3.1.5 Module 4 complete CRUD Rides

c) ***Description:*** This modules Add,update,delete Rides.

bb) ***Usage Scenario/ Use case Description/ Specification:***

| | |
|---|---|
| **Description** | [ make \| drop \| view \| update] to the Rides Table |
| **Inputs** | Ride time,area,start time ,end time… |
| **Source** | 7. The Module admin sets a ride up<br>8. Press Button … |
| **Alternate Case** | |
| **Outputs** | Added \| Deleted \| Viewed \| Modified Rides |
| **Precondition** | If logedin via Module admin. |

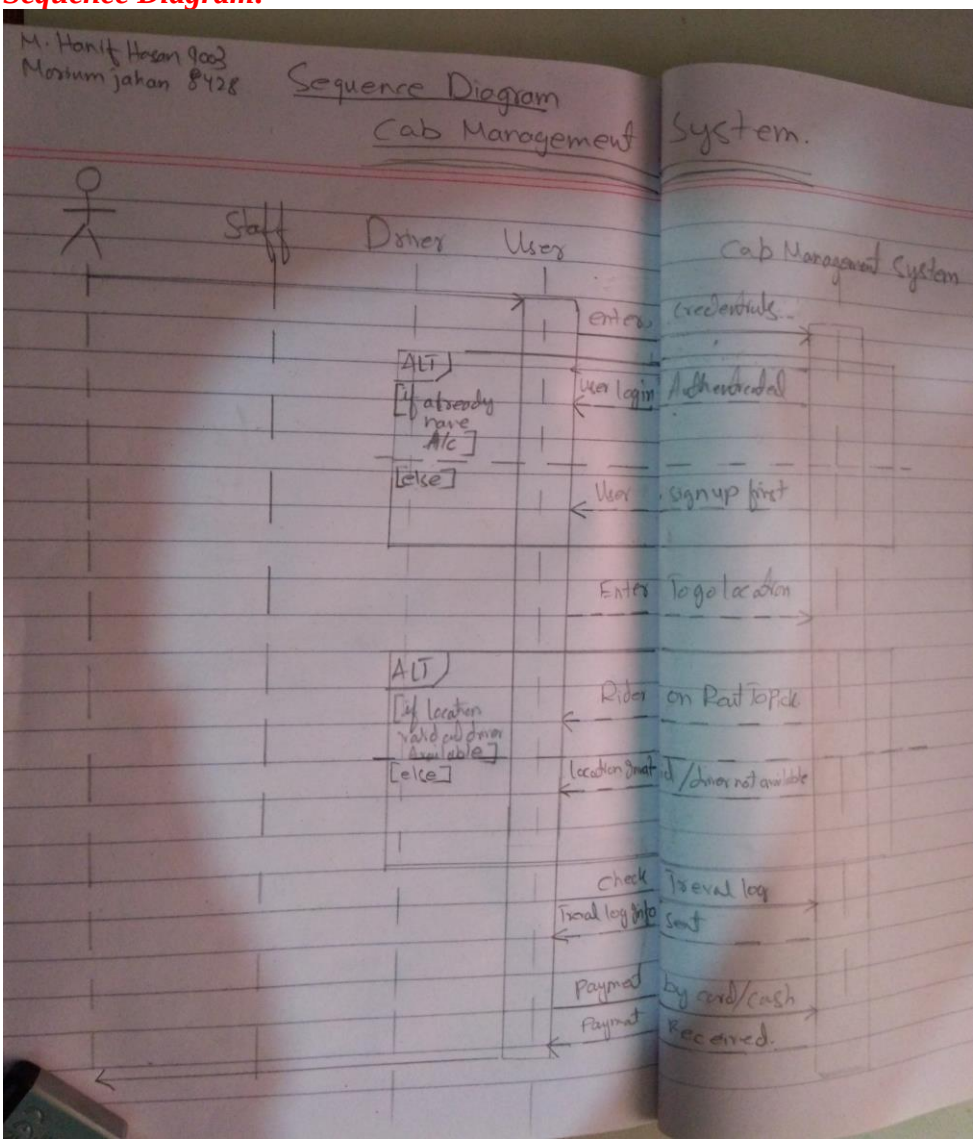
***cc) Use case Diagram:***



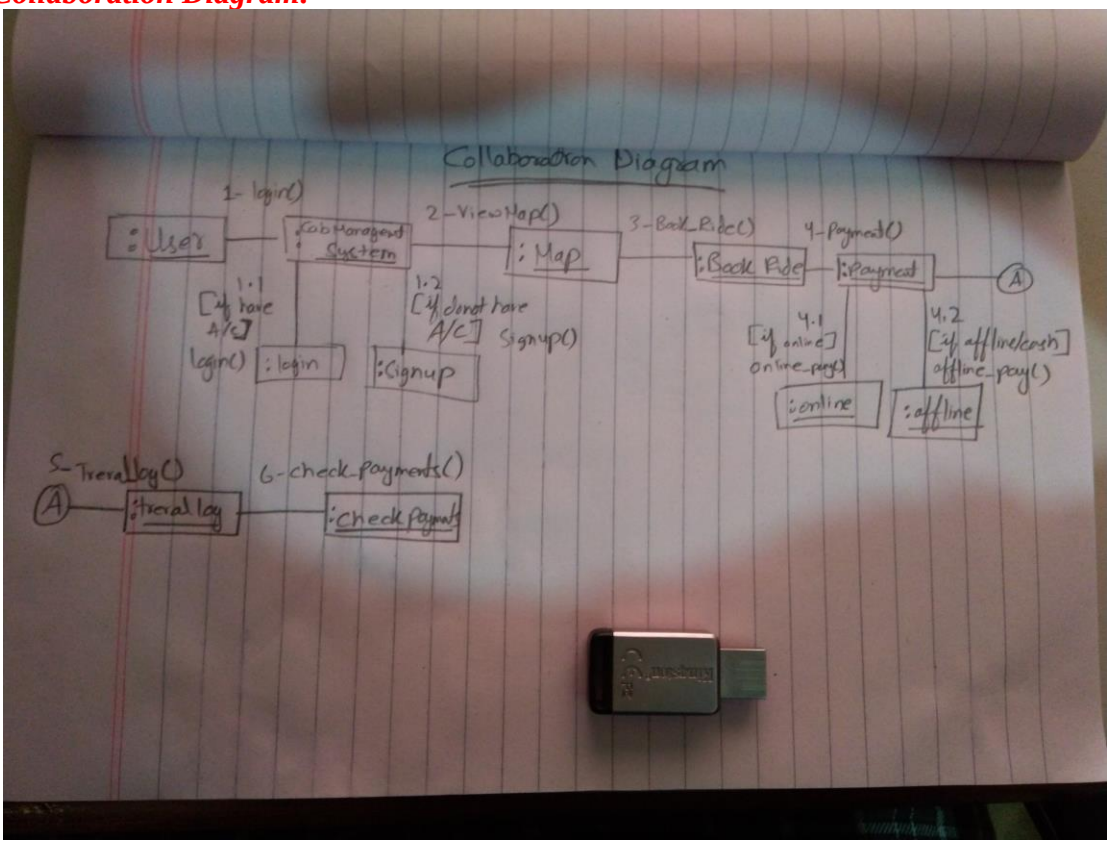

***dd) Use case Realization:***

The realization was not needed.

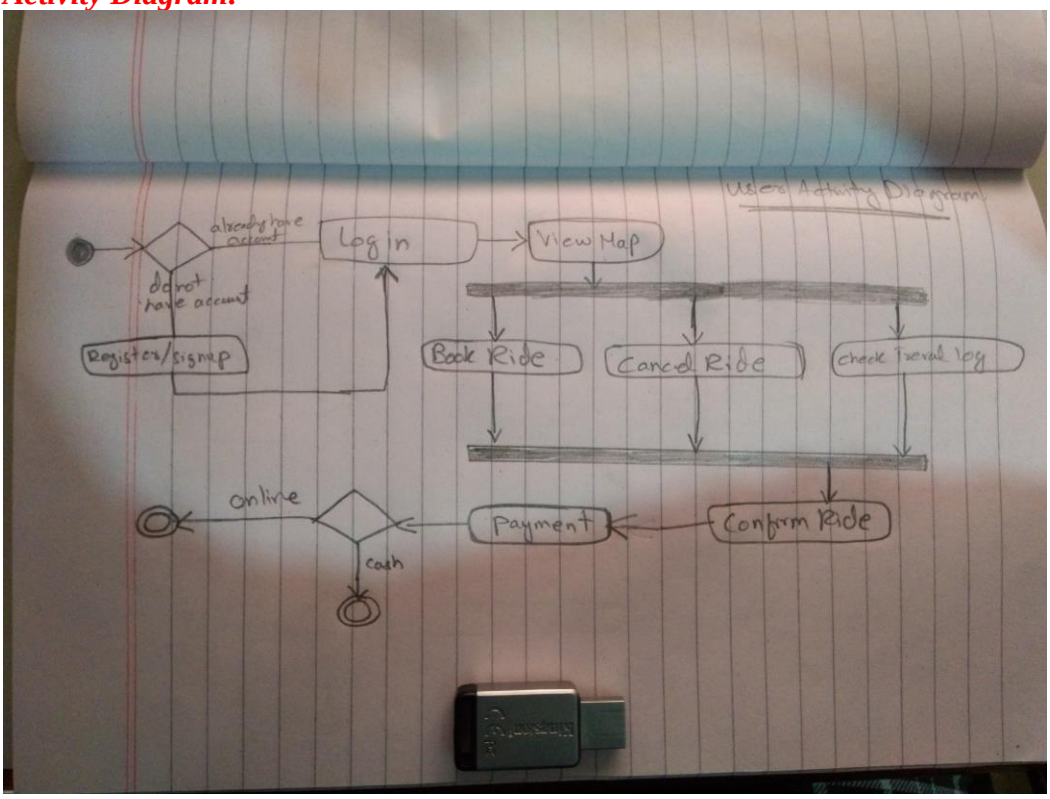
ee) ***Sequence Diagram:***



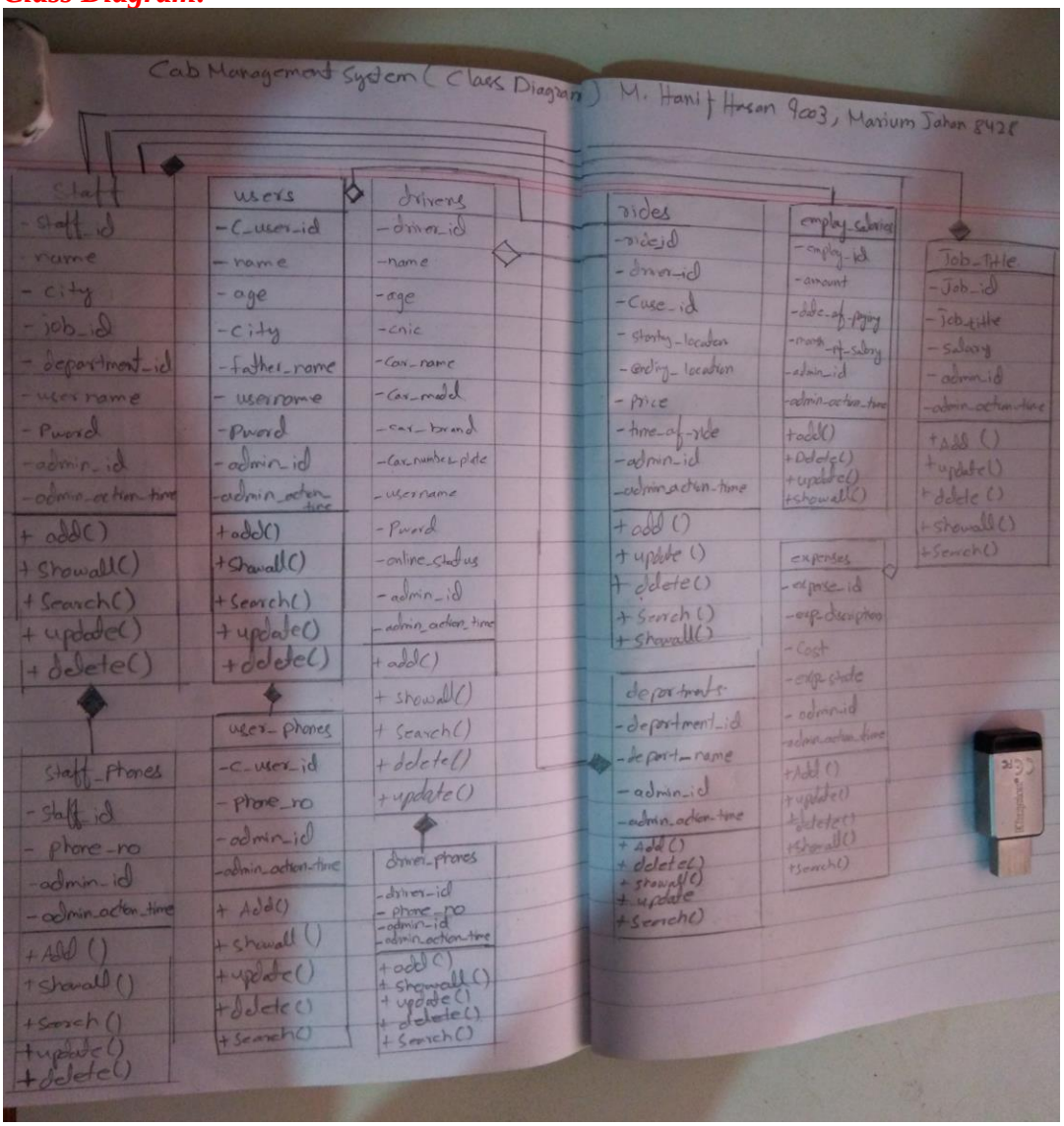
## *ff) Collaboration Diagram:*



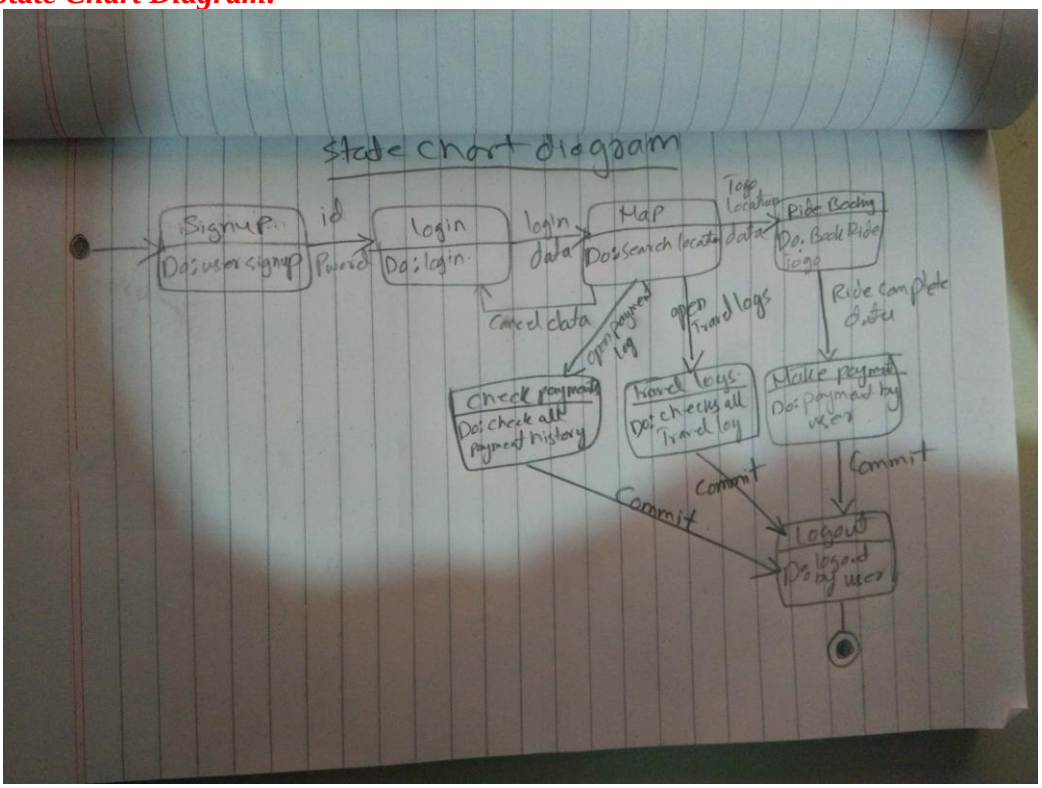

## *gg) Activity Diagram:*



## *hh)Class Diagram:*



***ii) State Chart Diagram:***



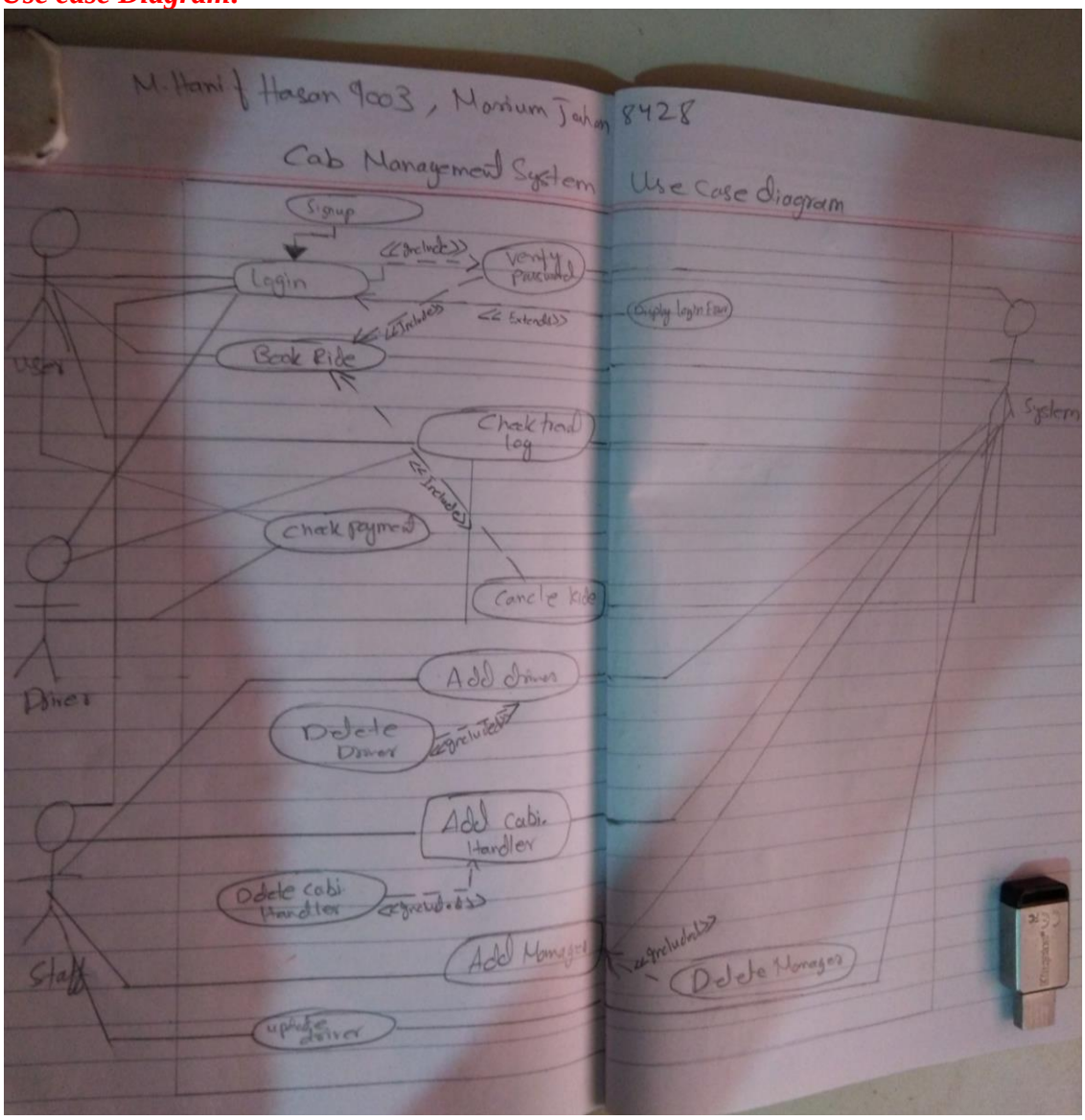

### 3.1.6 Module 5 complete CRUD Drivers

d) ***Description:*** This modules Add,update,delete Drivers.

jj) ***Usage Scenario/ Use case Description/ Specification:***

| | |
|---|---|
| **Description** | [ make \| drop \| view \| update] to the Drivers Table |
| **Inputs** | Driver Name,Adress,Car Name,Car Number,Phone,..etc |
| **Source** | 9. The Module admin sets a Driver up<br>10. Press Button … |
| **Alternate Case** | |
| **Outputs** | Added \| Deleted \| Viewed \| Modified Drivers |
| **Precondition** | If logedin via Module admin. |
| **Description** | [ make \| drop \| view \| update] to the Drivers Table |

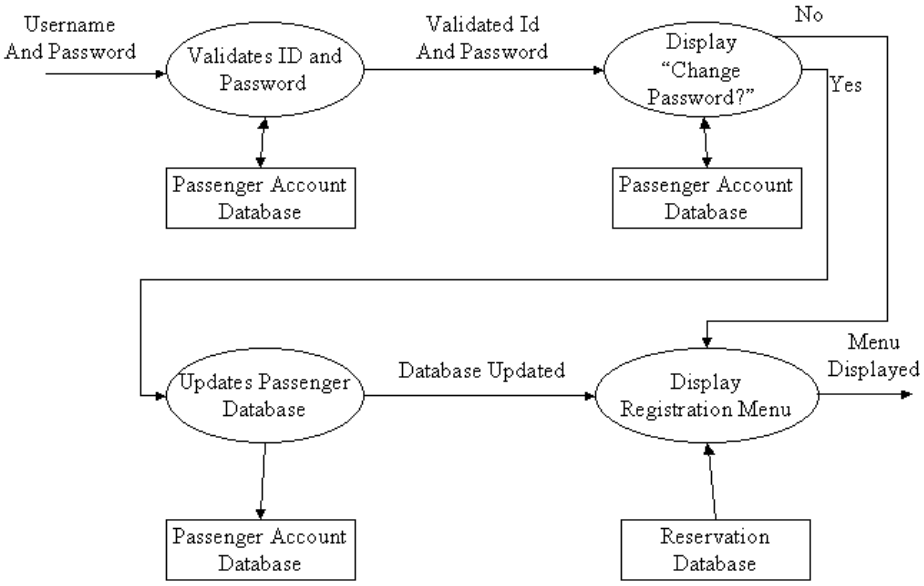
***kk) Use case Diagram:***



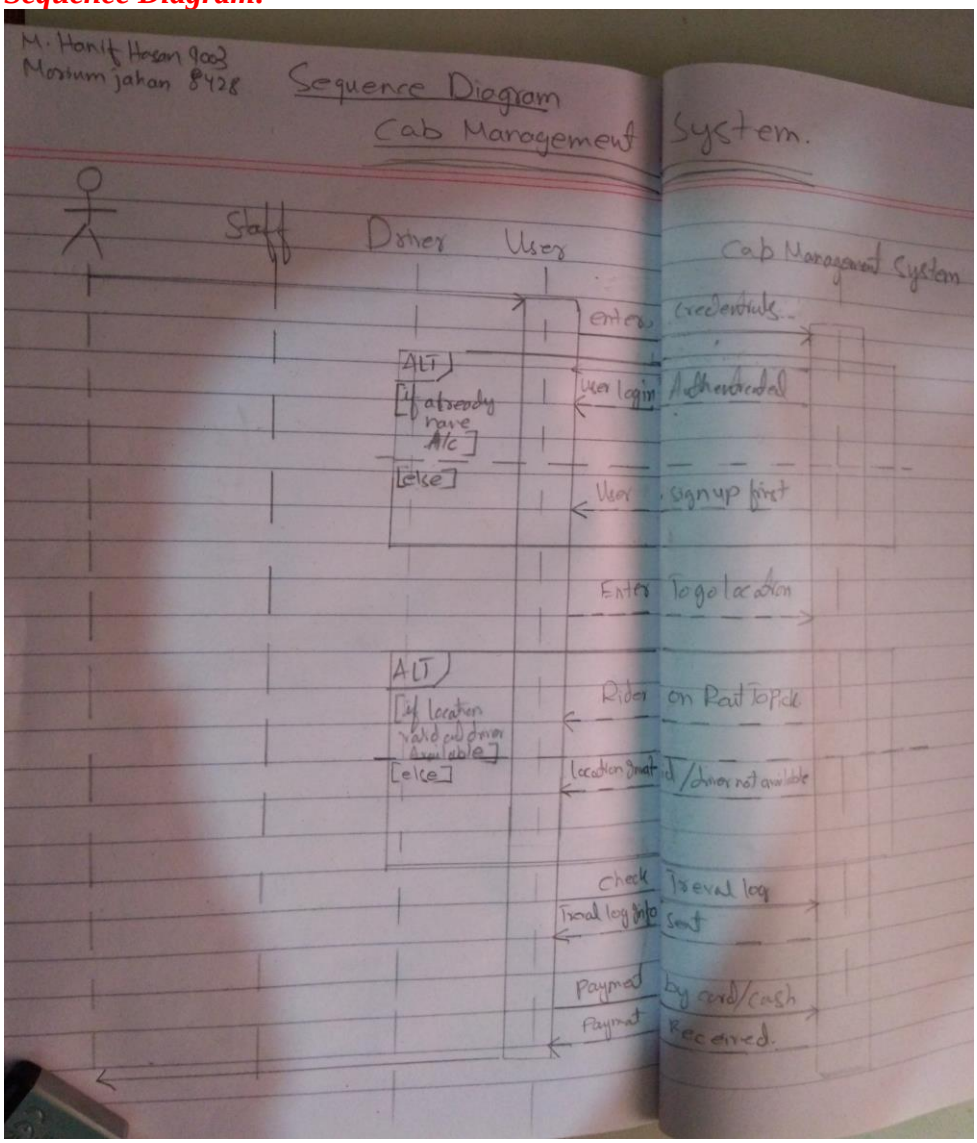

***ll) Use case Realization:***

The realization was not needed.

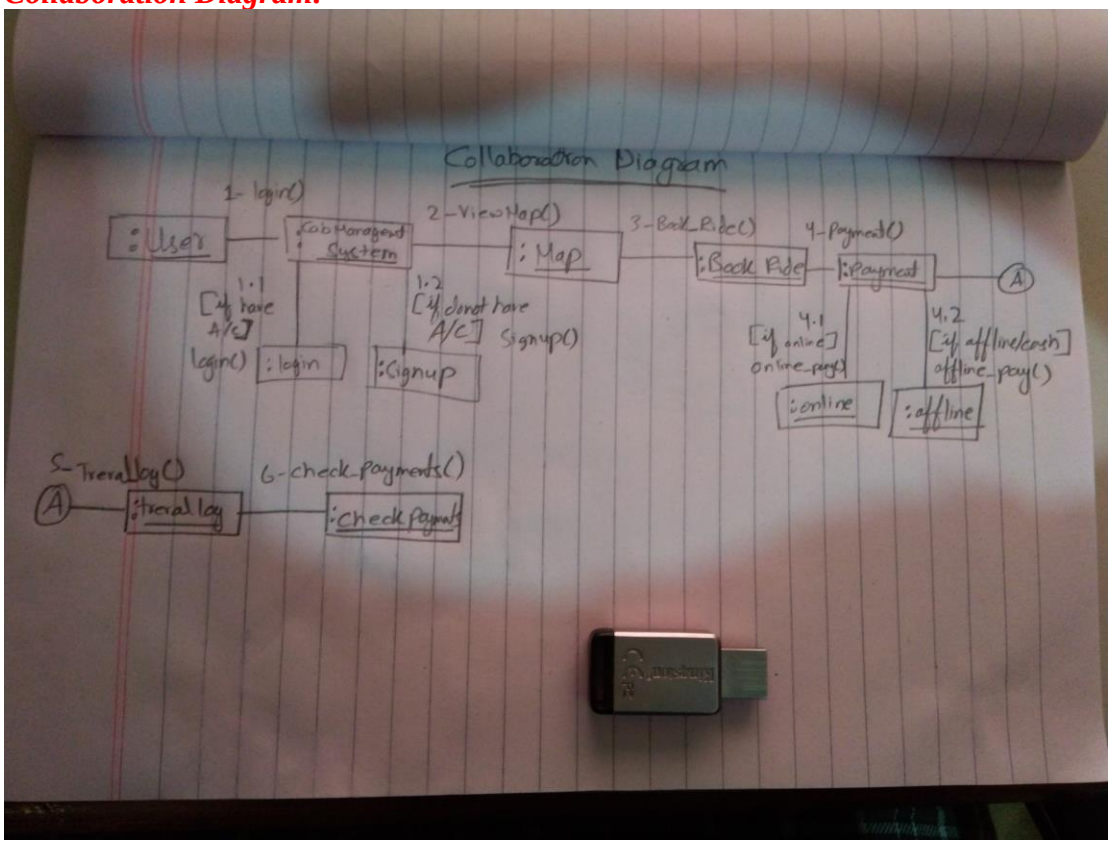
***mm) Flow of Event or Data Flow Diagram:***


Username
And Password
Validates ID and
Password
Validated Id
And Password
Display
"Change
Password?"
No
Yes
Passenger Account
Database
Passenger Account
Database
Updates Passenger
Database
Database Updated
Display
Registration Menu
Menu
Displayed
Passenger Account
Database
Reservation
Database

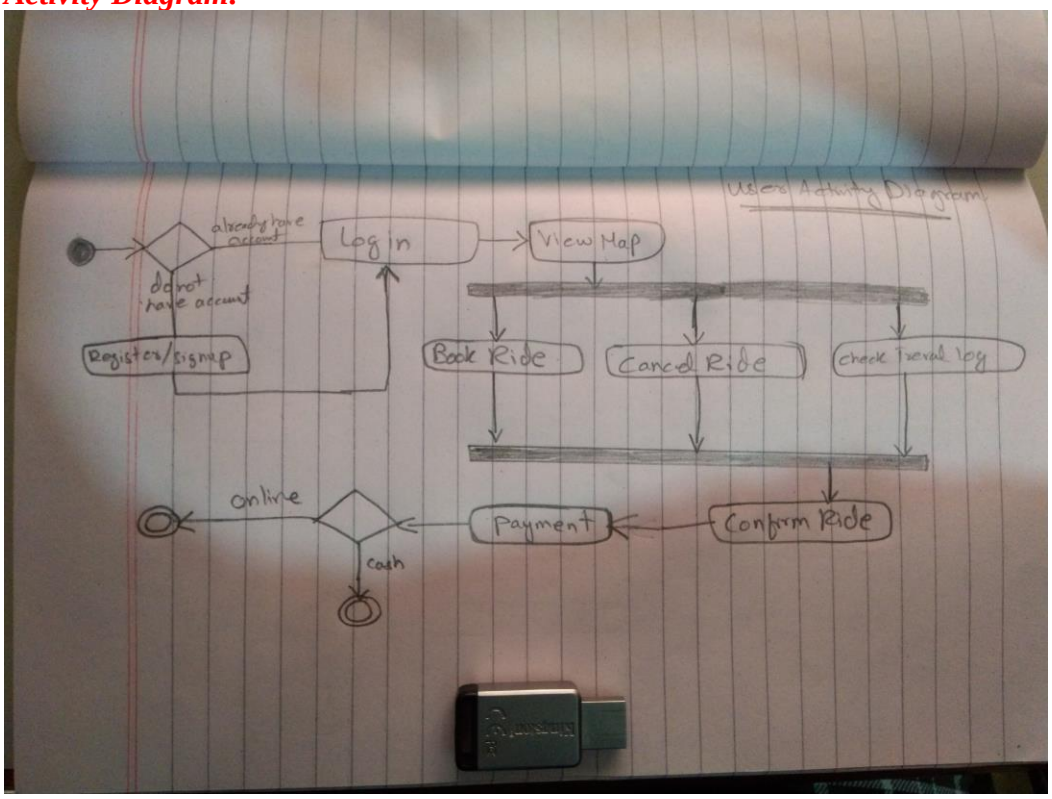
nn) ***Sequence Diagram:***



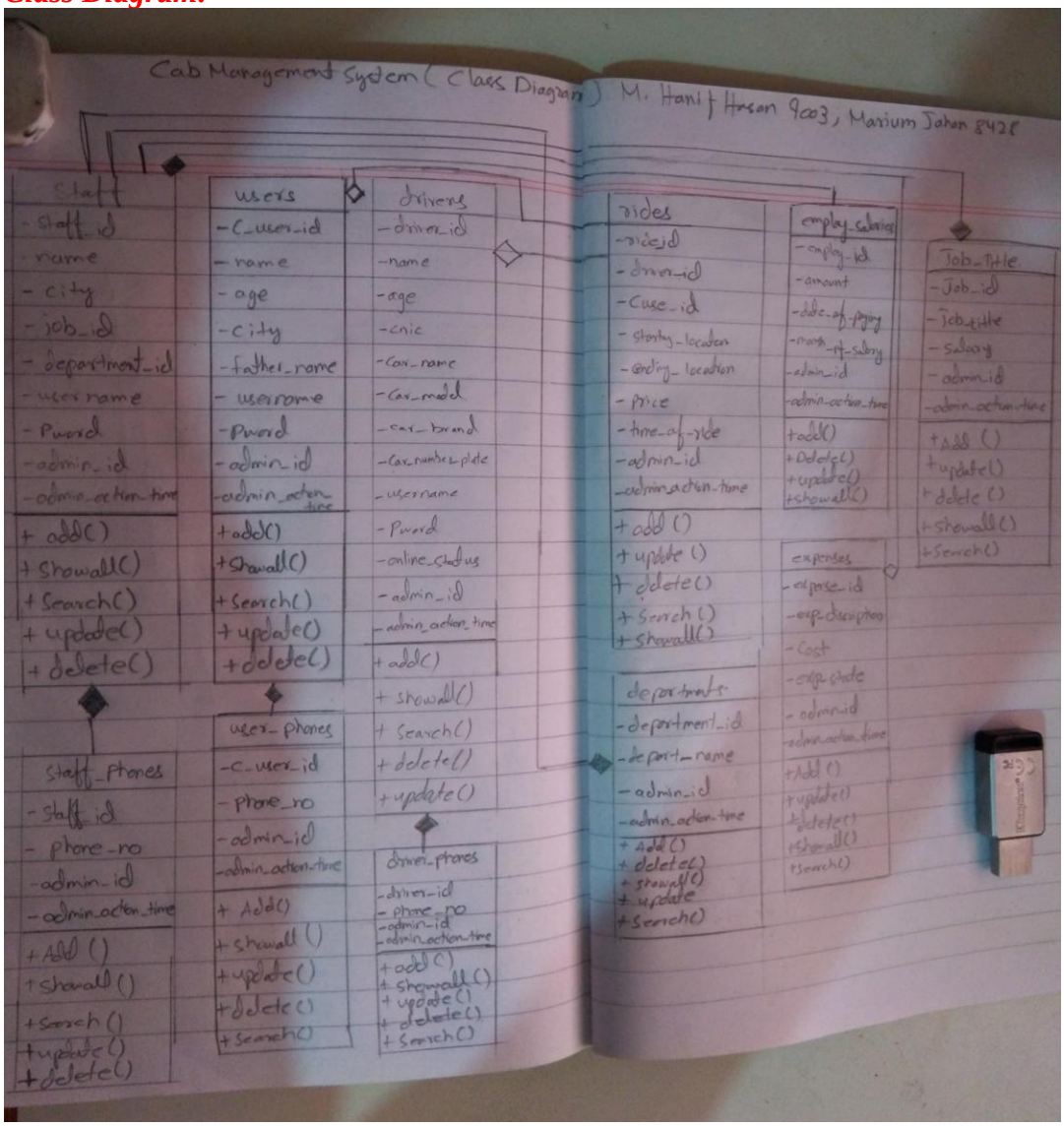
## *oo) Collaboration Diagram:*



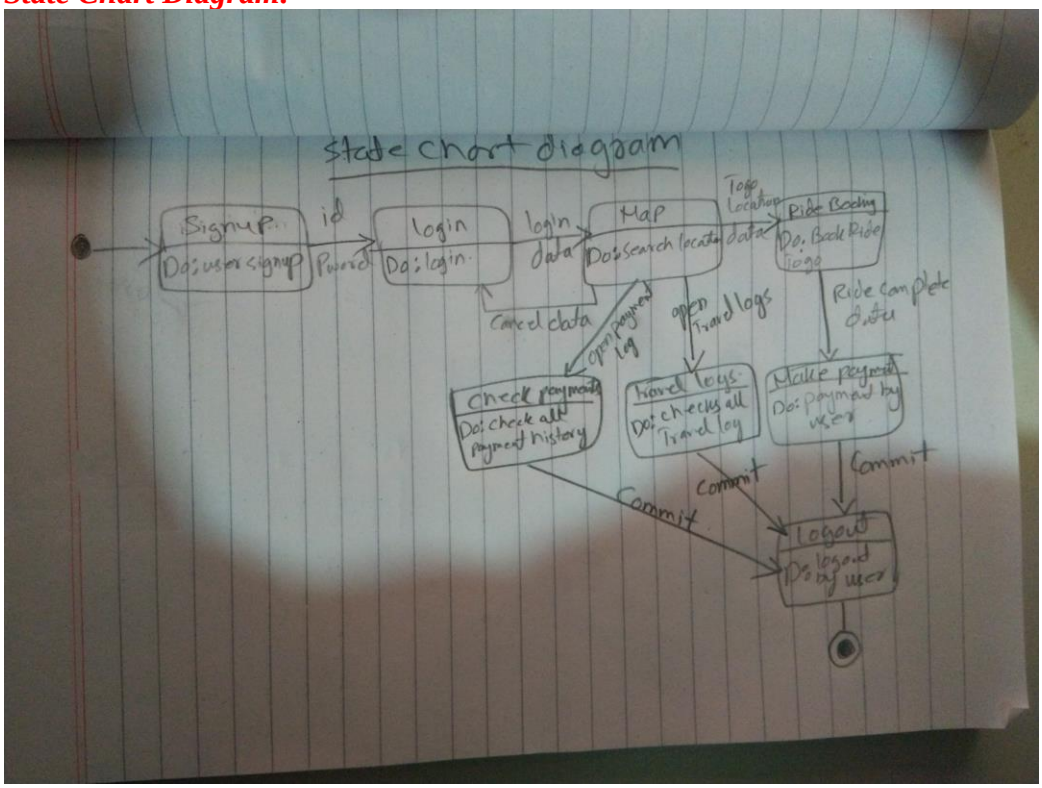

## *pp) Activity Diagram:*



***qq) Class Diagram:***



***rr) State Chart Diagram:***



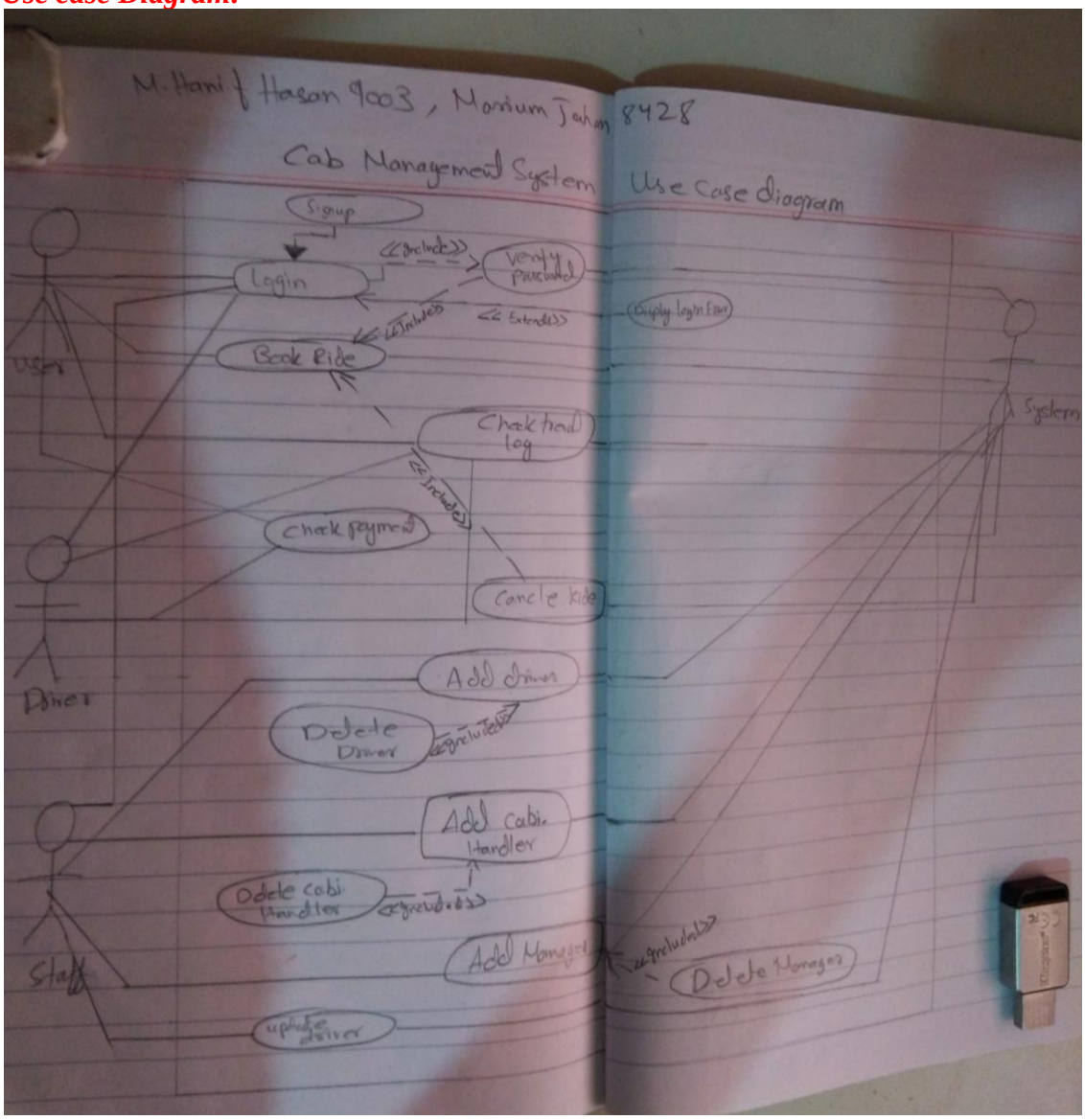

### 3.1.6 Module 6 complete CRUD Employ salary

e) ***Description:*** This modules Add,update,delete Employ Salary.

ss) ***Usage Scenario/ Use case Description/ Specification:***

| | |
|---|---|
| **Description** | [ make \| drop \| view \| update] to the Employ_Salarys Table |
| **Inputs** | Employ name,Salary, time stamp , Salary Ammount… |
| **Source** | 11. The Module admin sets a Employ_Salarys up<br>12. Press Button … |
| **Alternate Case** | |
| **Outputs** | Added \| Deleted \| Viewed \| Modified Employ_Salarys |
| **Precondition** | If logedin via Module admin. |
| **Description** | [ make \| drop \| view \| update] to the Employ_Salarys Table |

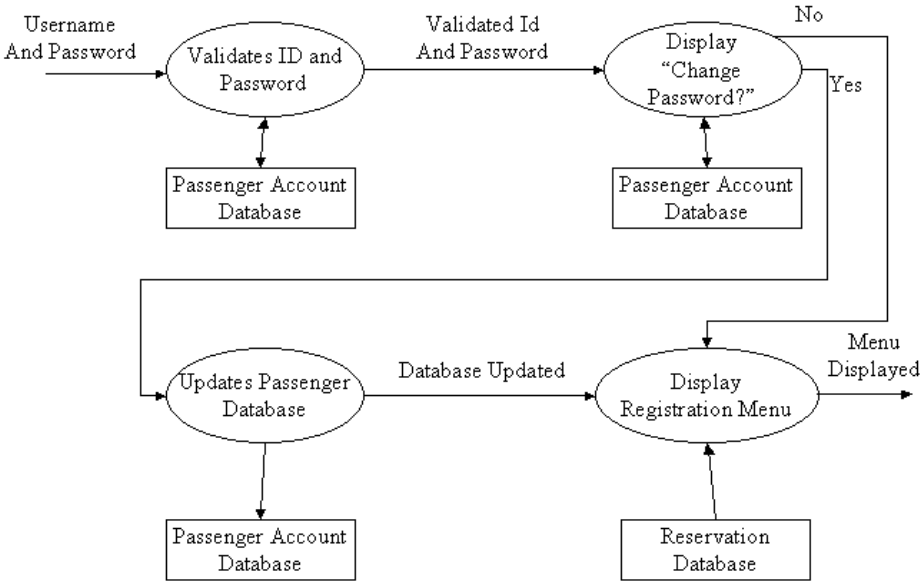
***tt) Use case Diagram:***



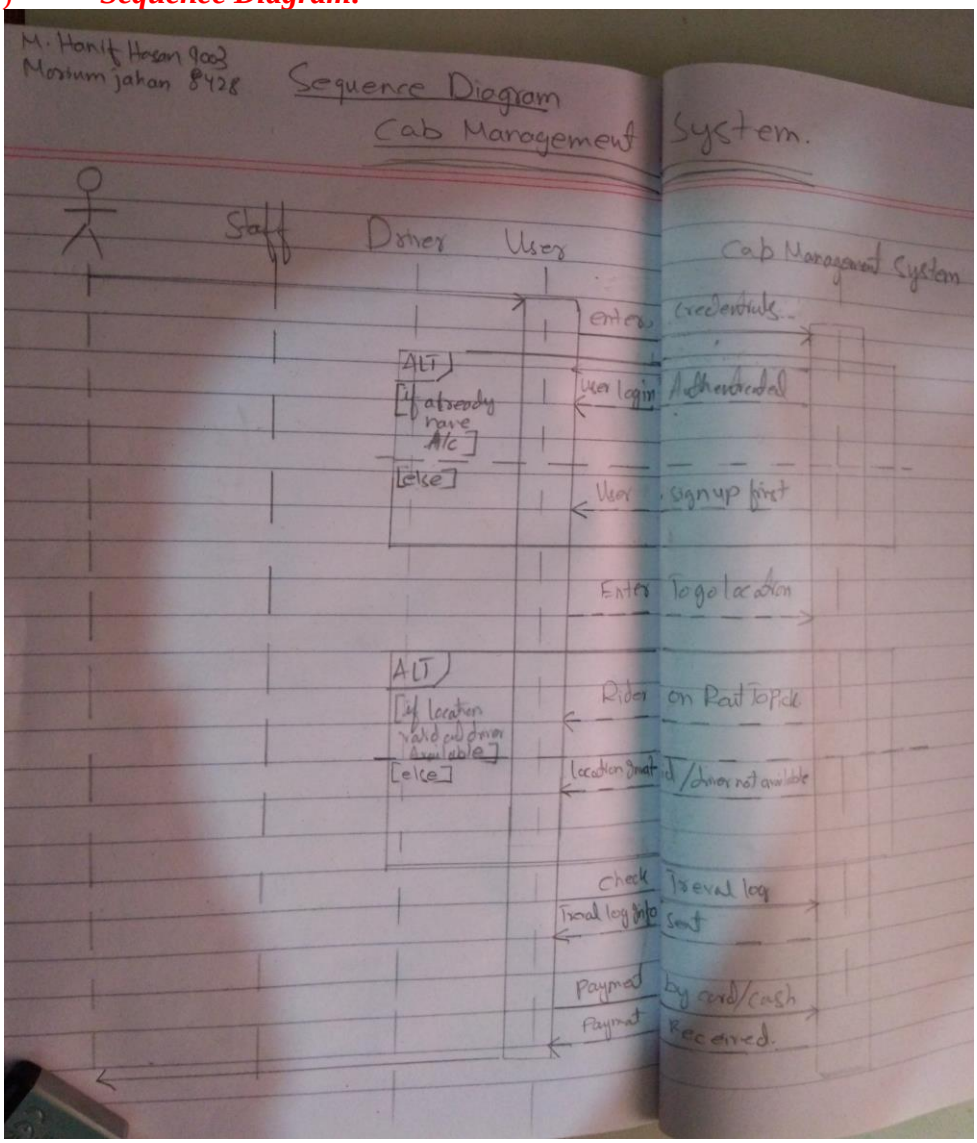

***uu) Use case Realization:***

The realization was not needed.

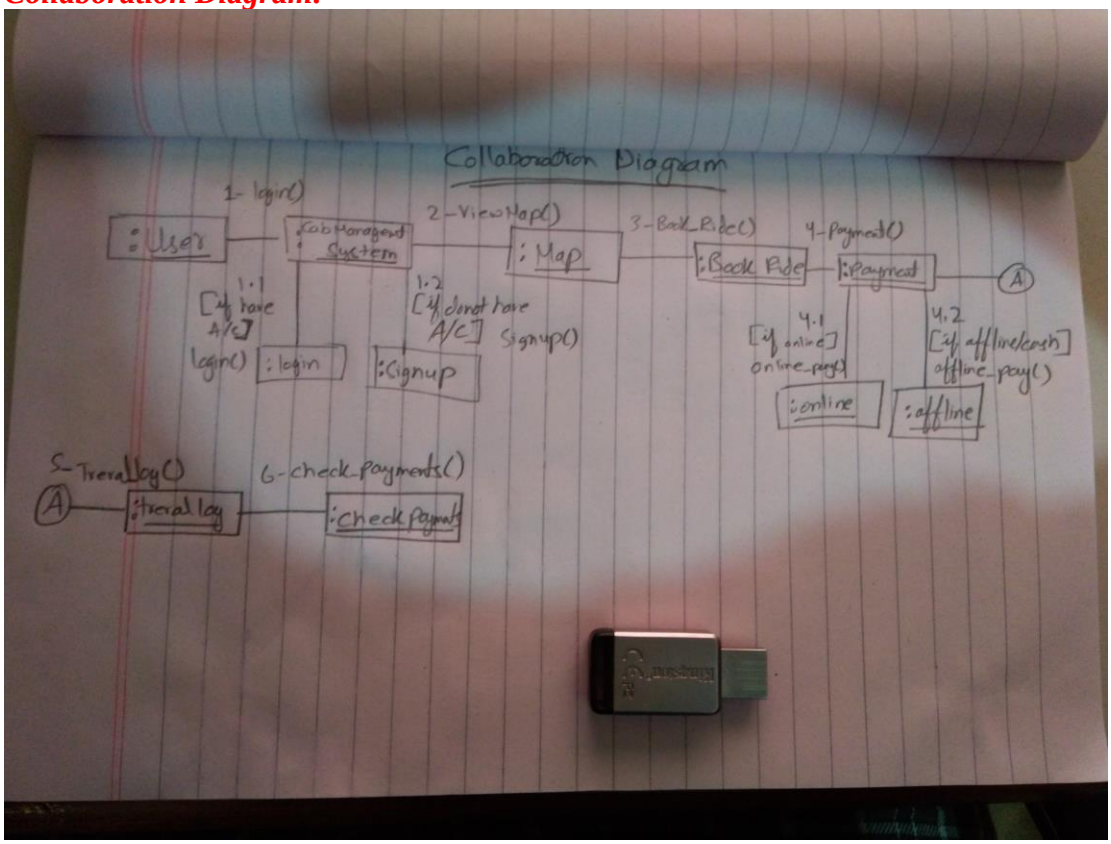
***vv) Flow of Event or Data Flow Diagram:***


Username
And Password
Validates ID and
Password
Validated Id
And Password
Display
"Change
Password?"
No
Yes
Passenger Account
Database
Passenger Account
Database
Updates Passenger
Database
Database Updated
Display
Registration Menu
Menu
Displayed
Passenger Account
Database
Reservation
Database

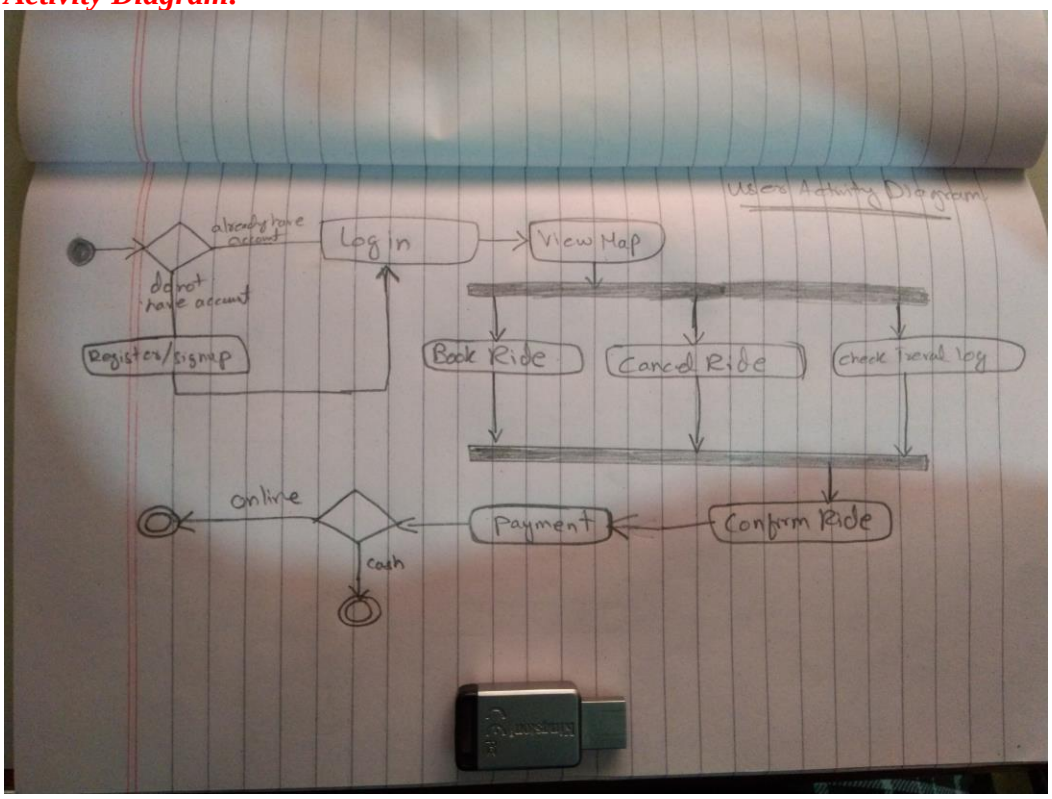
ww) ***Sequence Diagram:***



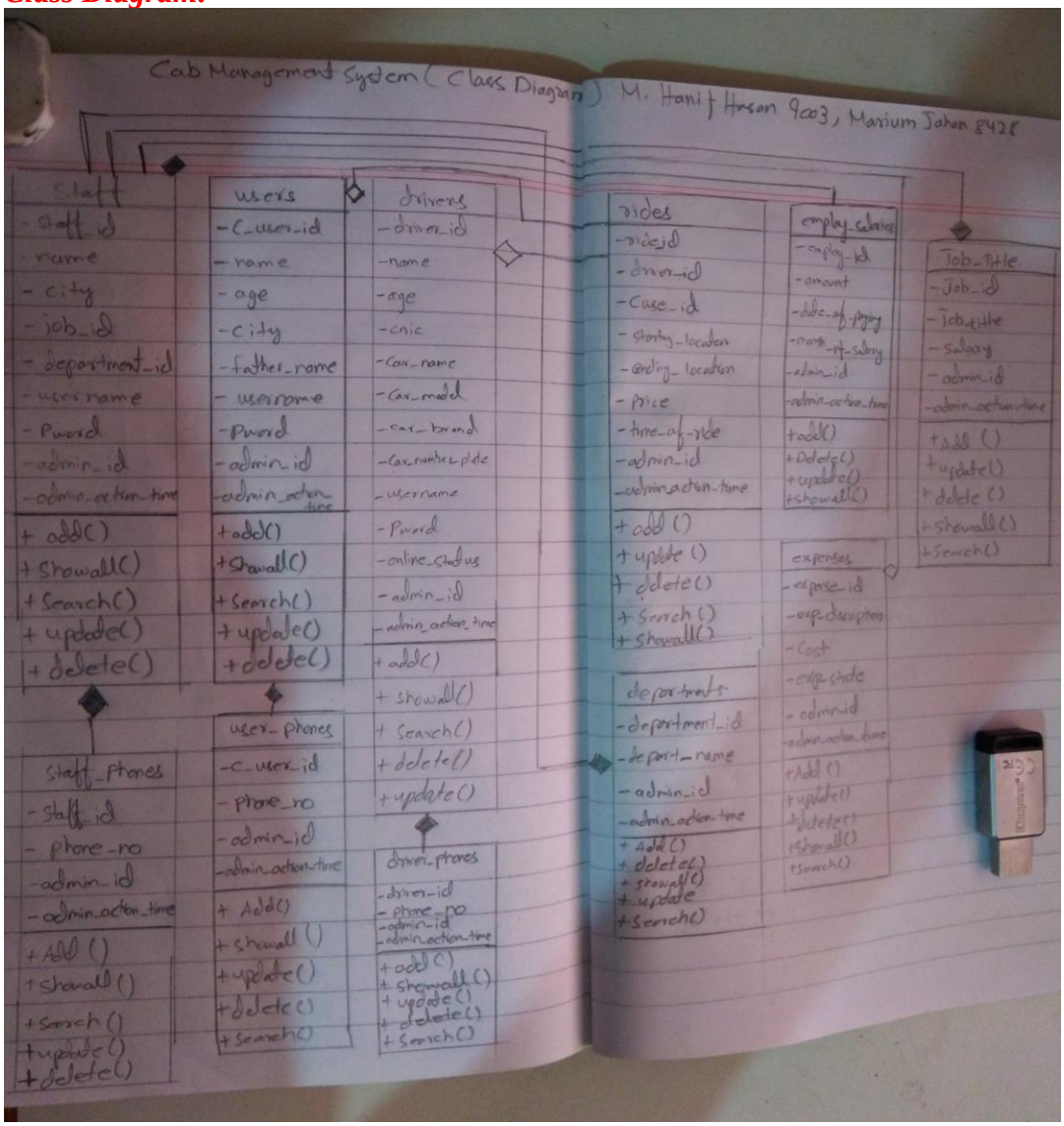
## *xx) Collaboration Diagram:*



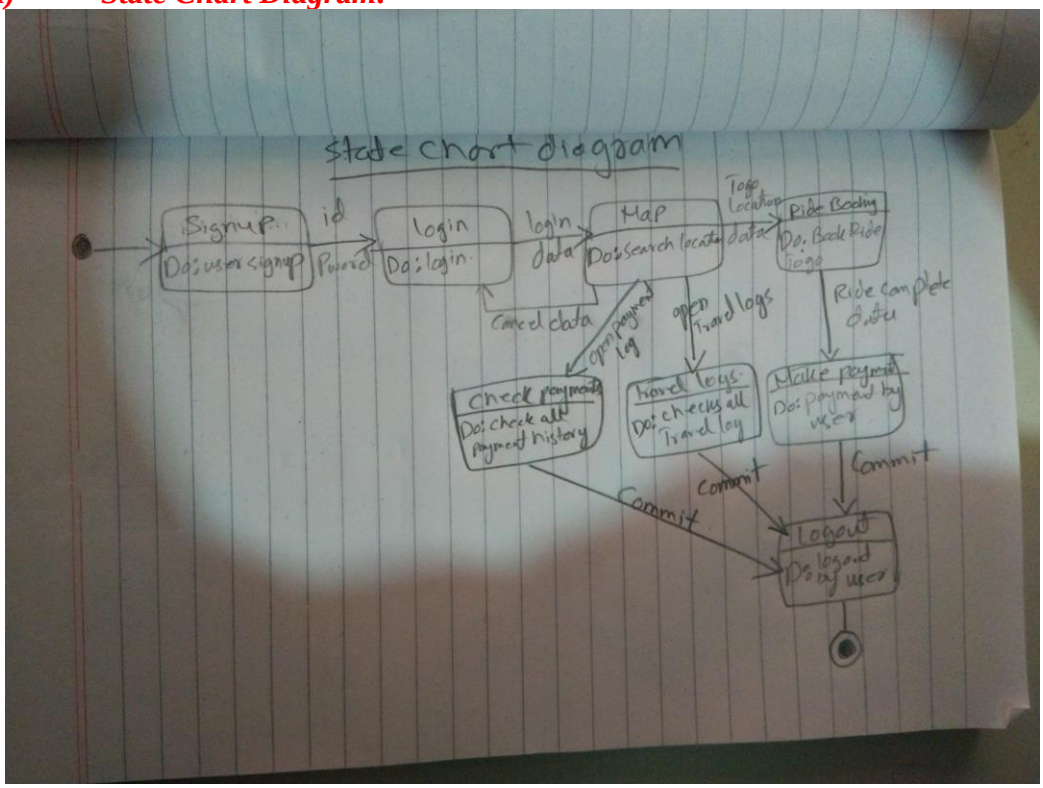

## *yy) Activity Diagram:*



## *zz) Class Diagram:*



***aaa) State Chart Diagram:***



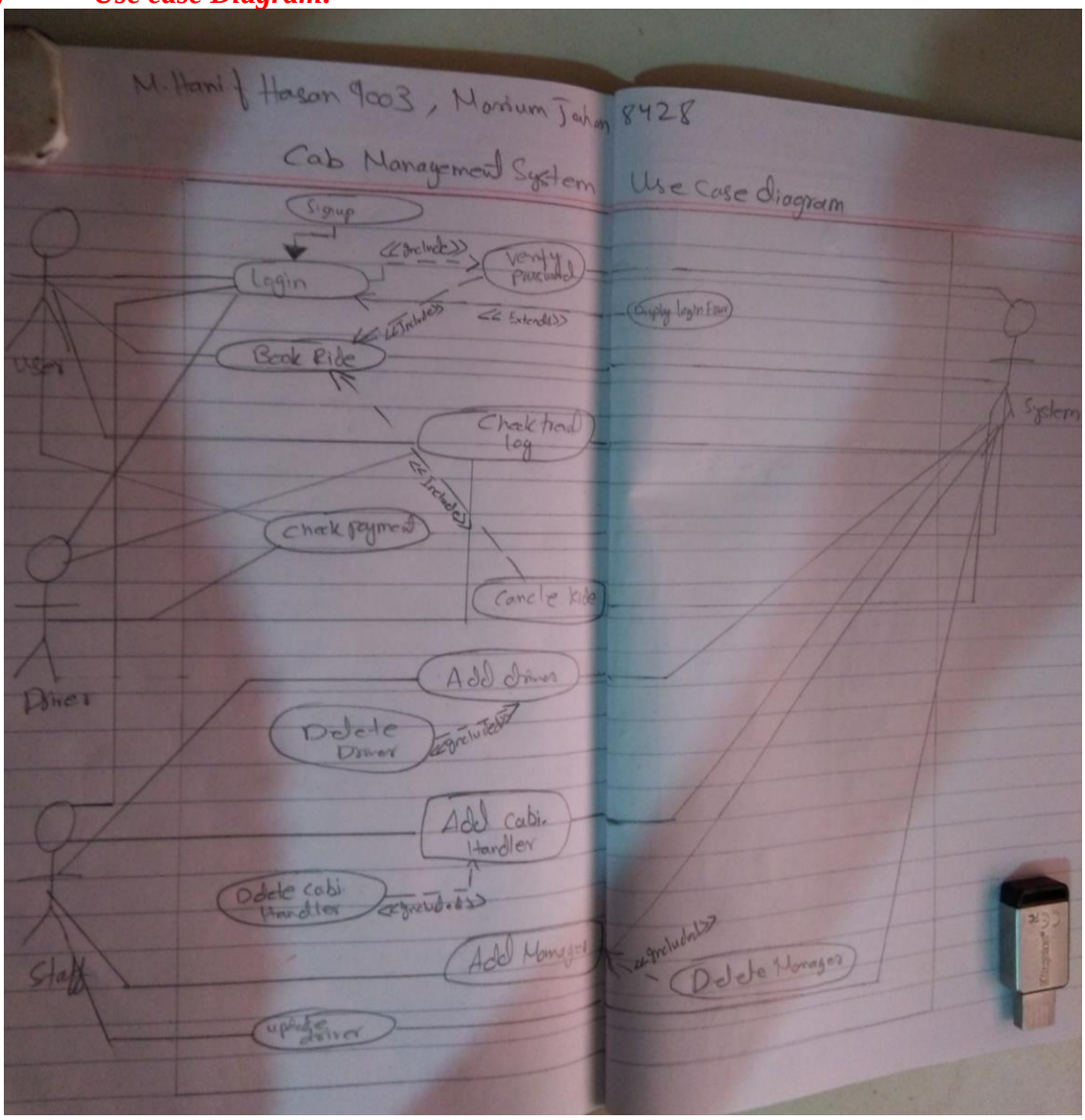

**3.1.6 Module 7 complete CRUD Job_titles**

f) ***Description:*** This modules Add,update,delete Job_titles.

bbb) ***Usage Scenario/ Use case Description/ Specification:***

| | |
|---|---|
| **Description** | [ make \| drop \| view \| update] to the Job_titles Table |
| **Inputs** | Job_TITLE,Job_Discription… |
| **Source** | 13. The Module admin sets a Job_titles up<br>14. Press Button … |
| **Alternate Case** | |
| **Outputs** | Added \| Deleted \| Viewed \| Modified Job_titles |
| **Precondition** | If logedin via Module admin. |
| **Description** | [ make \| drop \| view \| update] to the Job_titles Table |

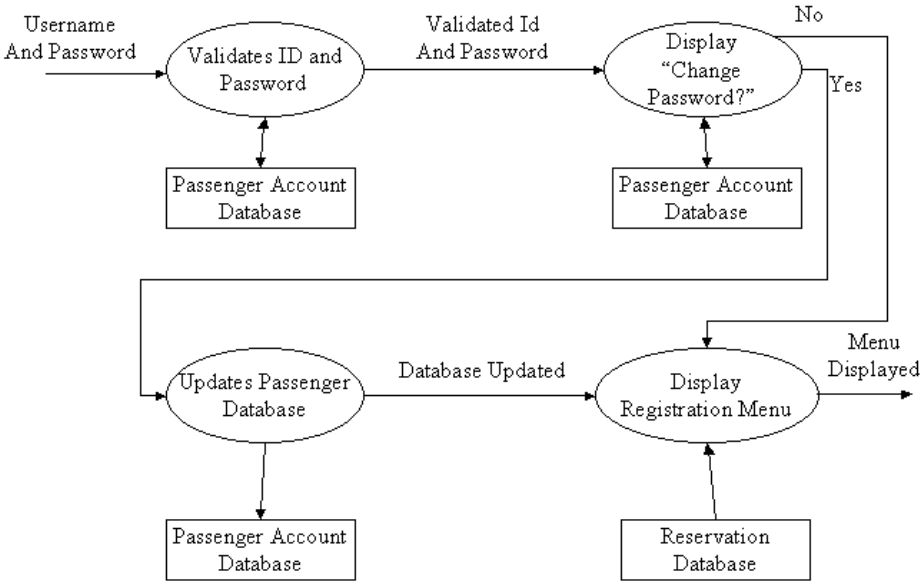
***ccc) Use case Diagram:***



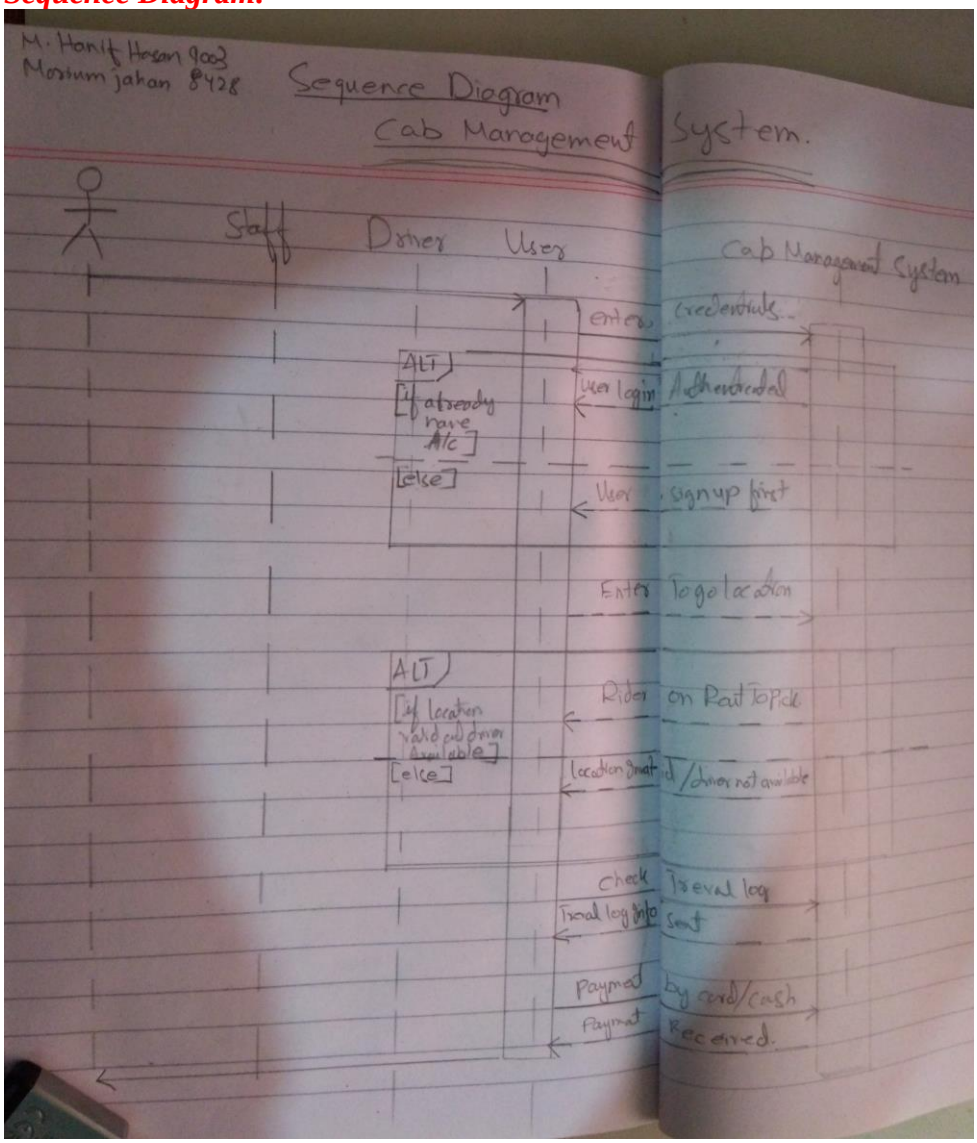

***ddd) Use case Realization:***

The realization was not needed.

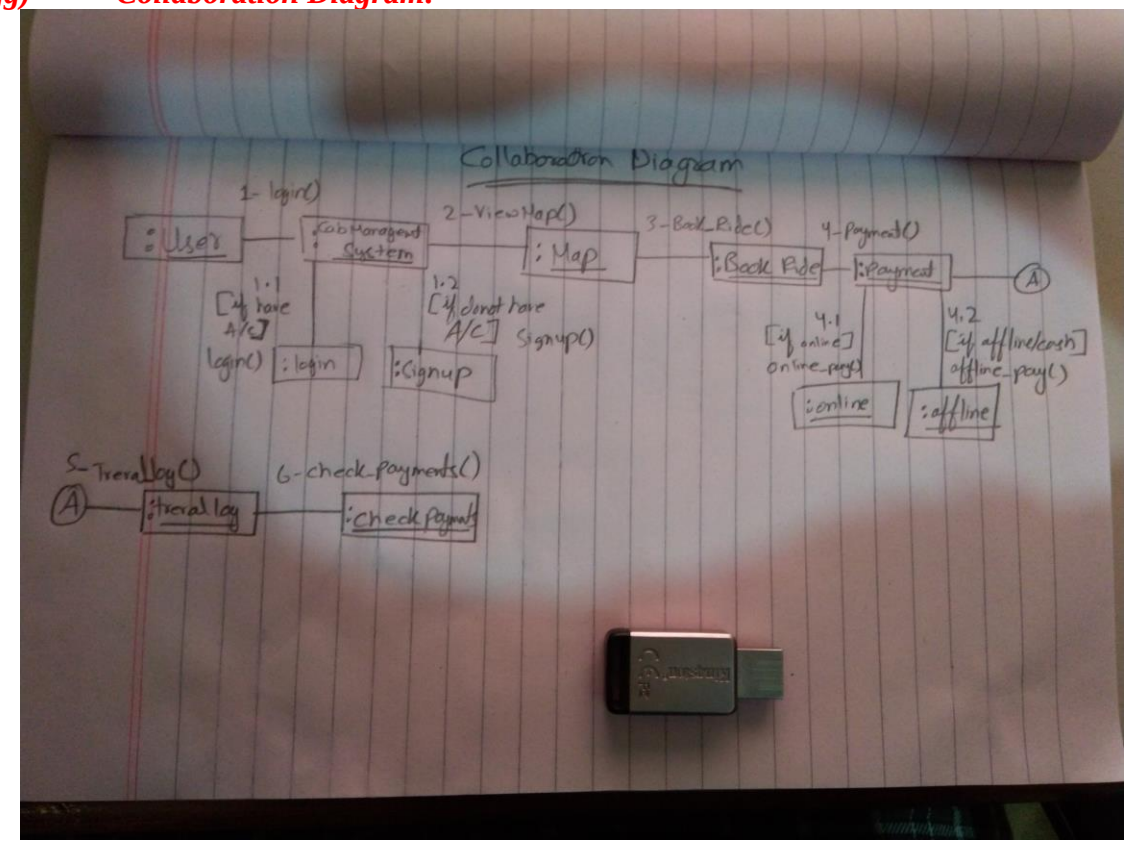
***eee) Flow of Event or Data Flow Diagram:***


Username
And Password
Validates ID and
Password
Validated Id
And Password
Display
"Change
Password?"
No
Yes
Passenger Account
Database
Passenger Account
Database
Updates Passenger
Database
Database Updated
Display
Registration Menu
Menu
Displayed
Passenger Account
Database
Reservation
Database

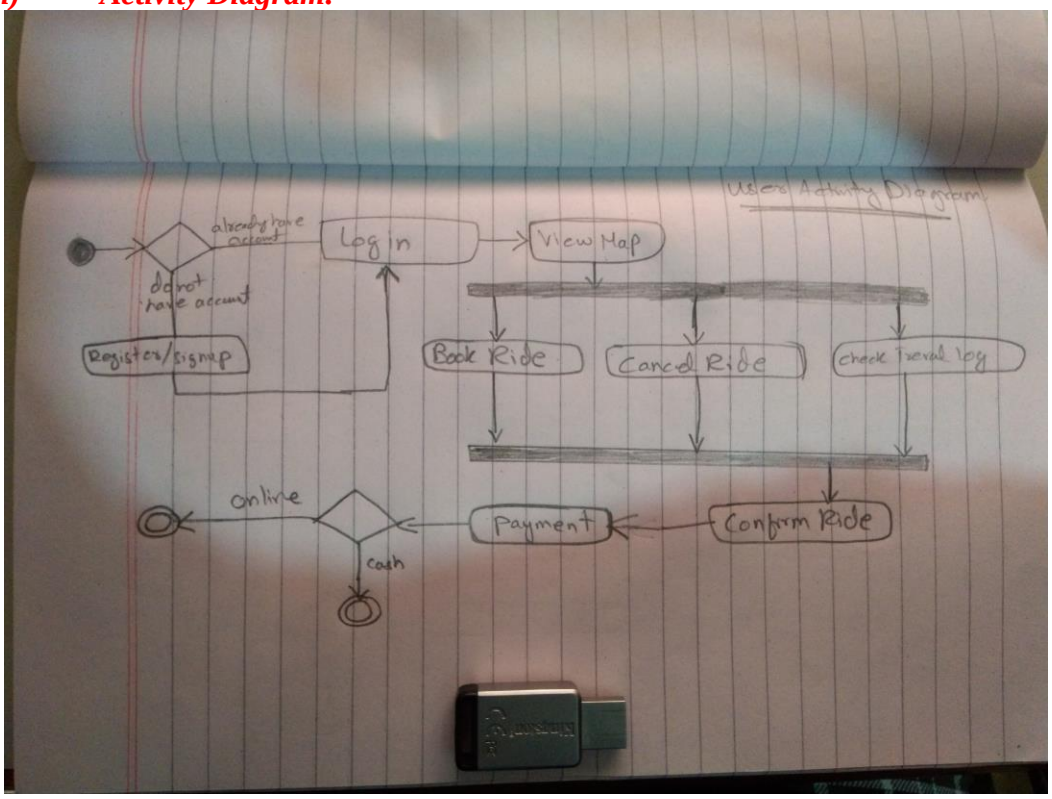
fff) ***Sequence Diagram:***



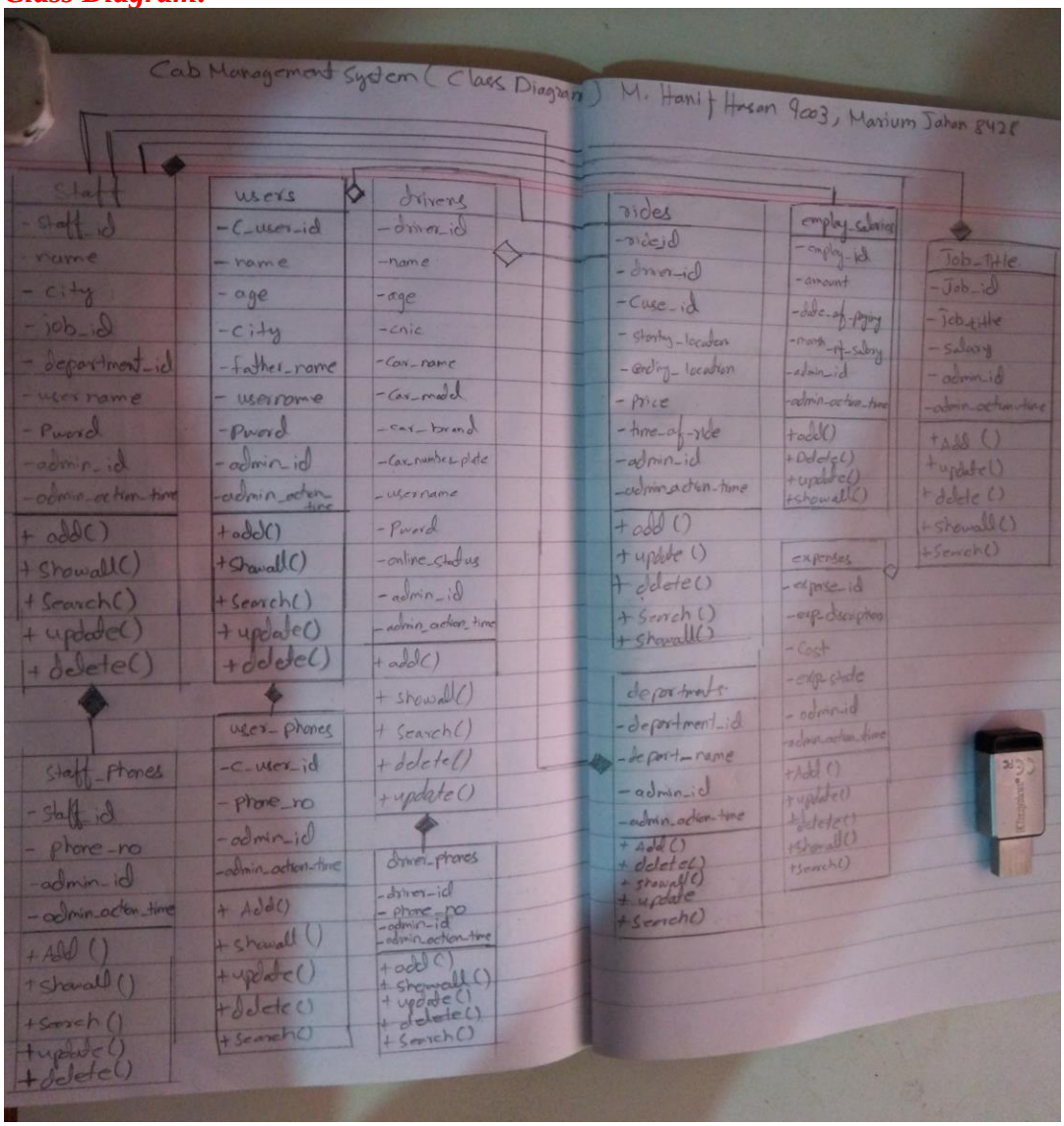
***ggg) Collaboration Diagram:***



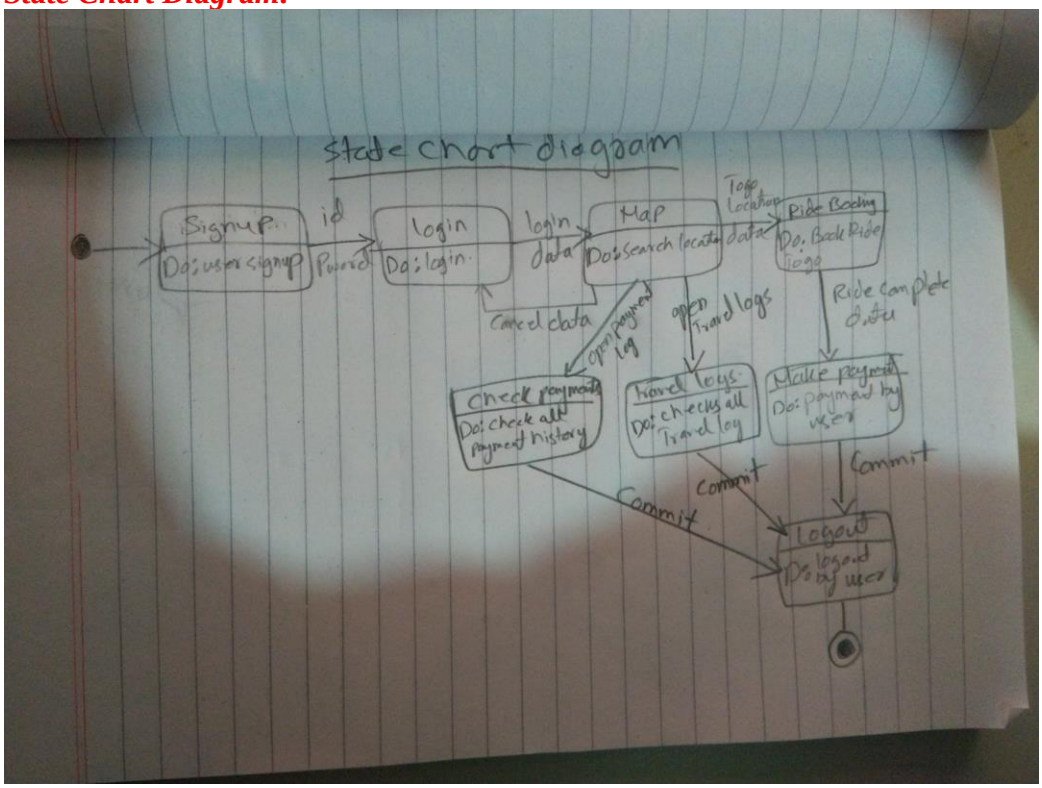

***hhh) Activity Diagram:***



### *iii) Class Diagram:*



***jjj) State Chart Diagram:***



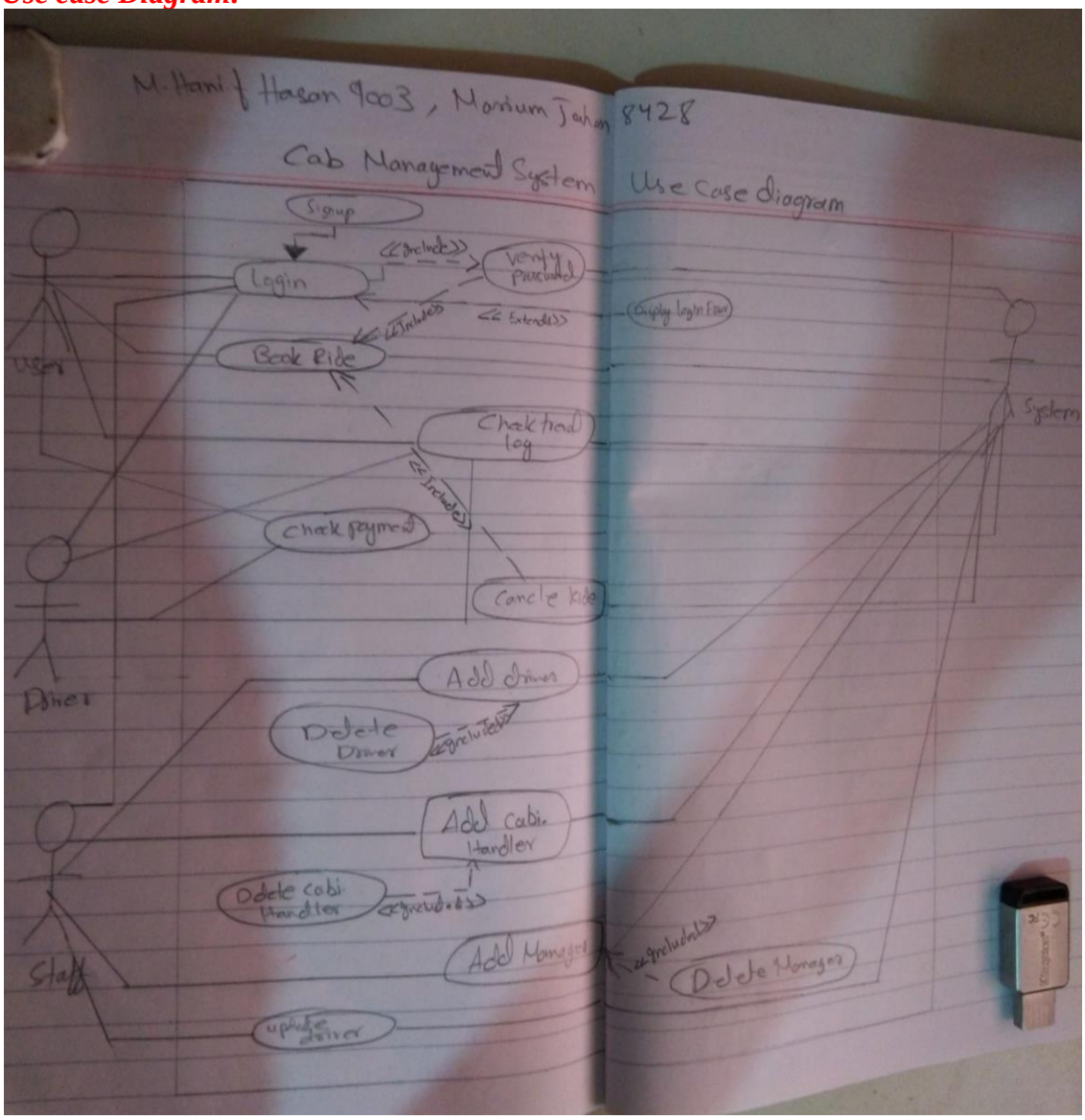

**3.1.6 Module 8 complete CRUD Staff_phones**

g) ***Description:*** This modules Add,update,delete Staff_Phones.

kkk) ***Usage Scenario/ Use case Description/ Specification:***

| | |
|---|---|
| **Description** | [ make \| drop \| view \| update] to the Staff_phones Table |
| **Inputs** | Staff_Name, Staff_Phone… |
| **Source** | 15. The Module admin sets a Staff_Phones up<br>16. Press Button … |
| **Alternate Case** | |
| **Outputs** | Added \| Deleted \| Viewed \| Modified Staff_Phones |
| **Precondition** | If logedin via Module admin. |
| **Description** | [ make \| drop \| view \| update] to the Staff_Phones Table |

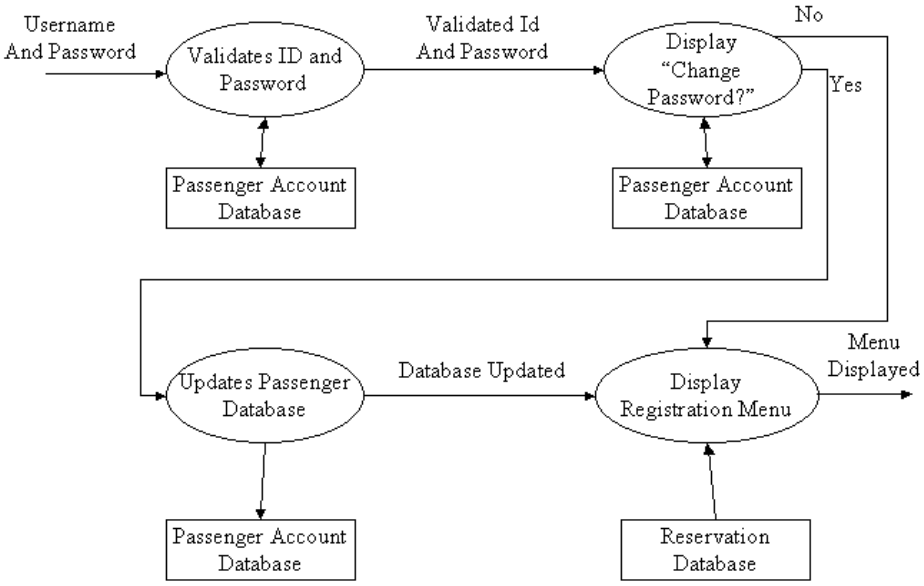
### *lll) Use case Diagram:*



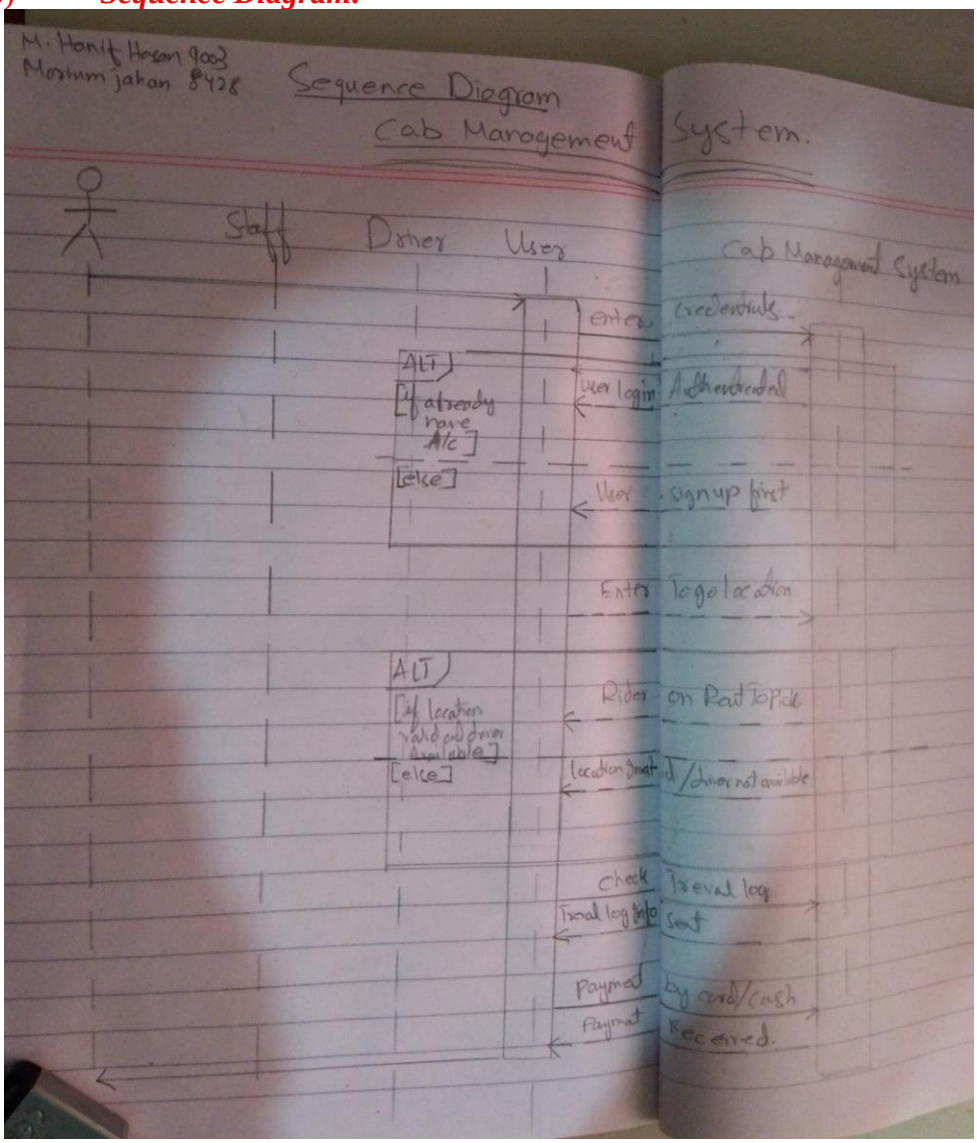

### *mmm) Use case Realization:*

The realization was not needed.

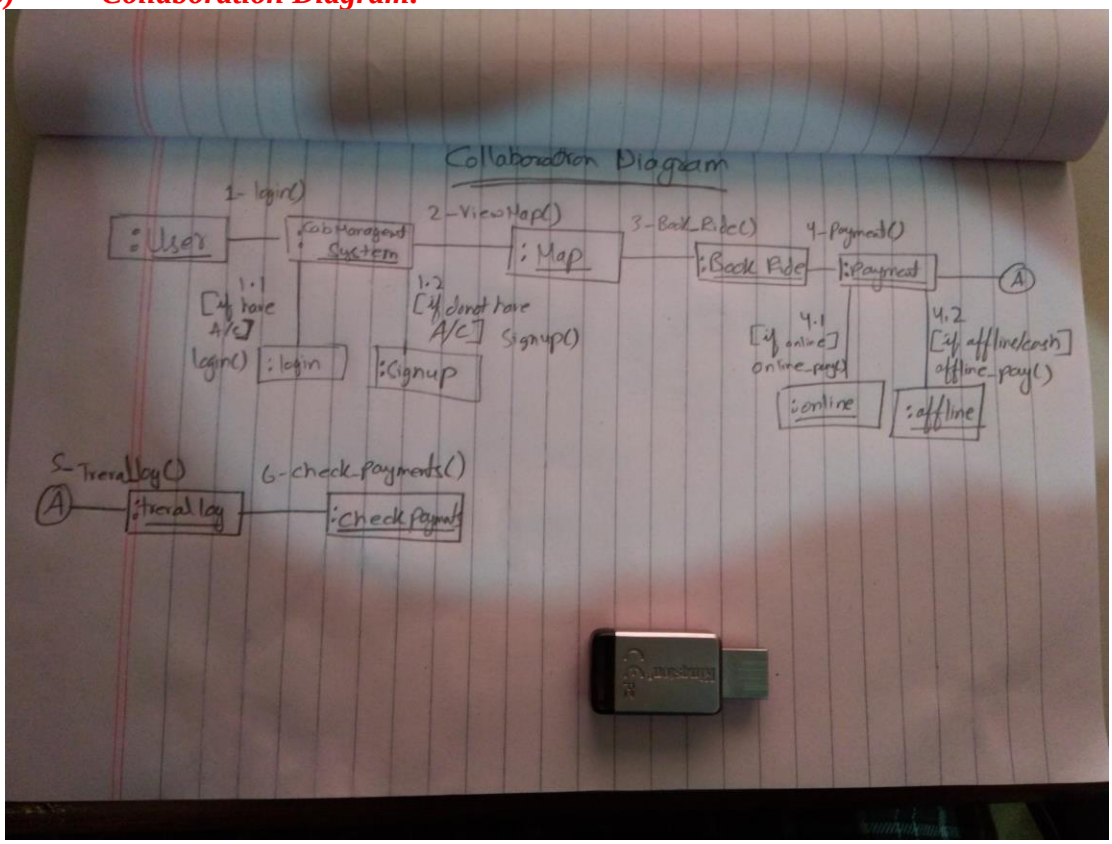
### *nnn) Flow of Event or Data Flow Diagram:*


Username
And Password
Validates ID and
Password
Validated Id
And Password
Display
"Change
Password?"
No
Yes
Passenger Account
Database
Passenger Account
Database
Updates Passenger
Database
Database Updated
Display
Registration Menu
Menu
Displayed
Passenger Account
Database
Reservation
Database

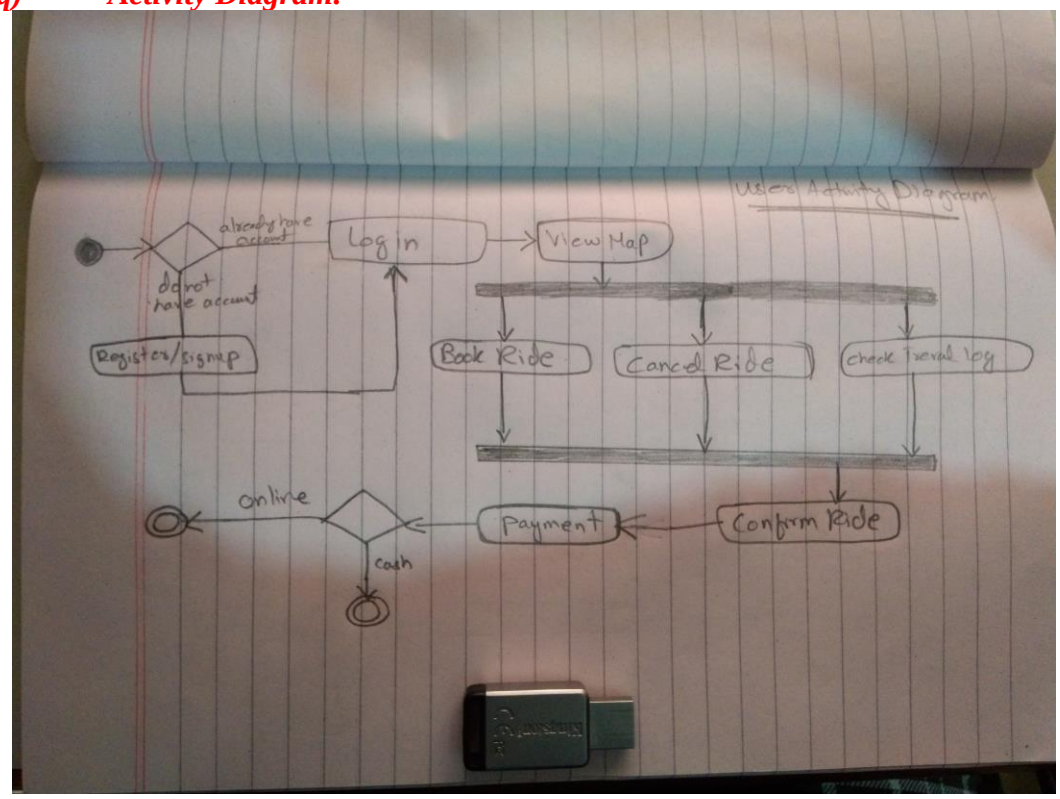
ooo) ***Sequence Diagram:***



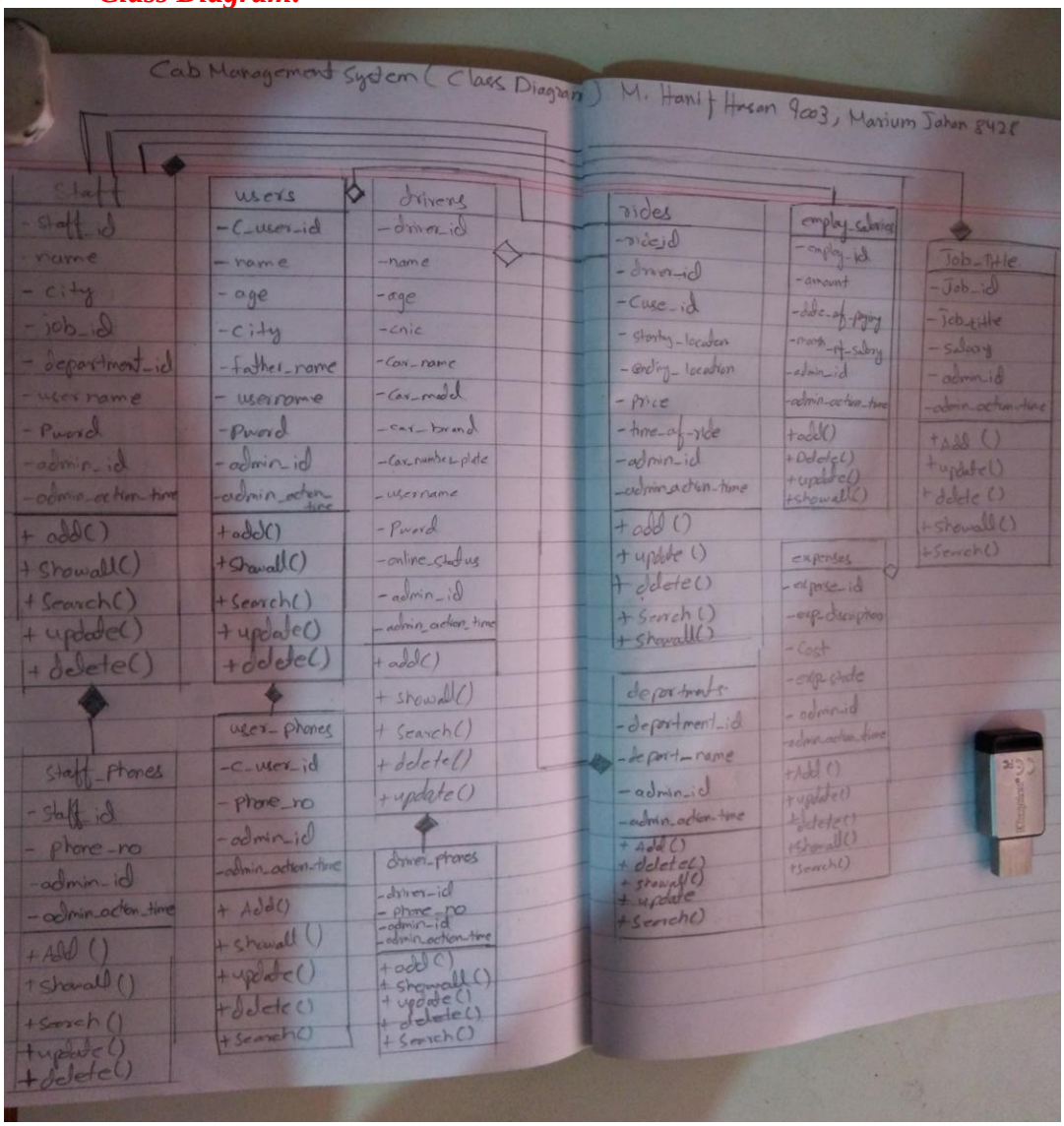
***ppp) Collaboration Diagram:***



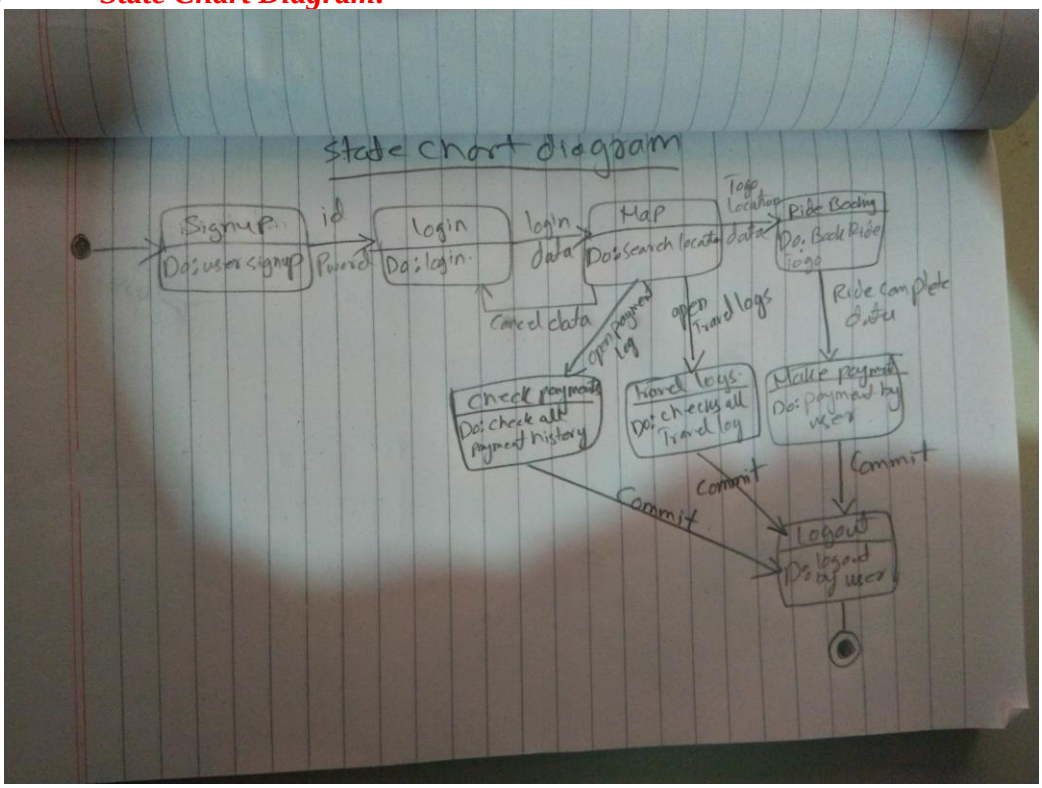

***qqq) Activity Diagram:***



***rrr) Class Diagram:***



***sss) State Chart Diagram:***



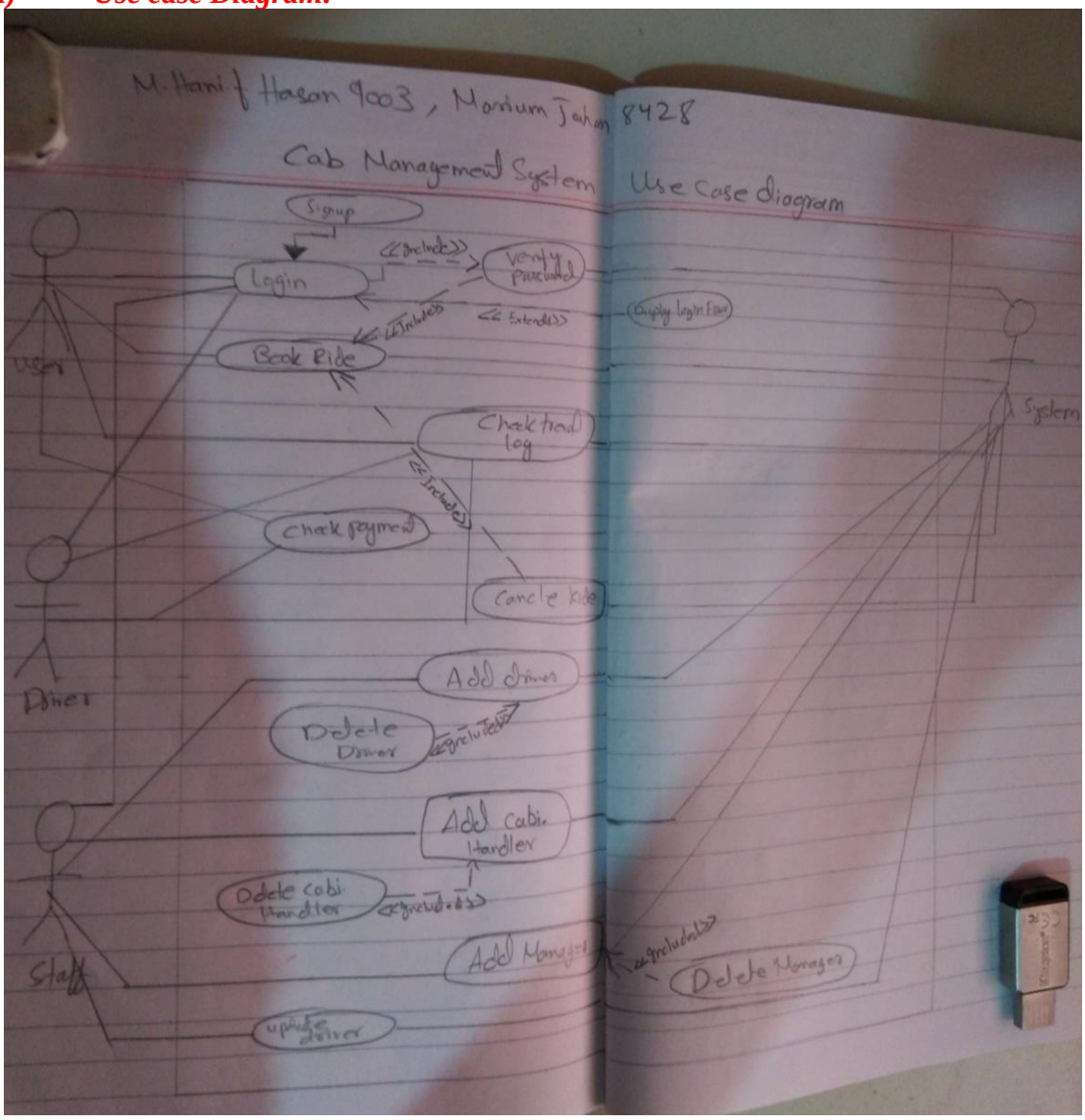

### 3.1.6 Module 9 complete CRUD User_phones

h) ***Description:*** This modules Add,update,delete User_Phones.

ttt) ***Usage Scenario/ Use case Description/ Specification:***

| | |
|---|---|
| **Description** | [ make \| drop \| view \| update] to the User_Phones Table |
| **Inputs** | User_Name,User_Phone… |
| **Source** | 17. The Module admin sets a UserPhones up<br>18. Press Button … |
| **Alternate Case** | |
| **Outputs** | Added \| Deleted \| Viewed \| Modified User_Prones |
| **Precondition** | If logedin via Module admin. |
| **Description** | [ make \| drop \| view \| update] to the User_Phones Table |

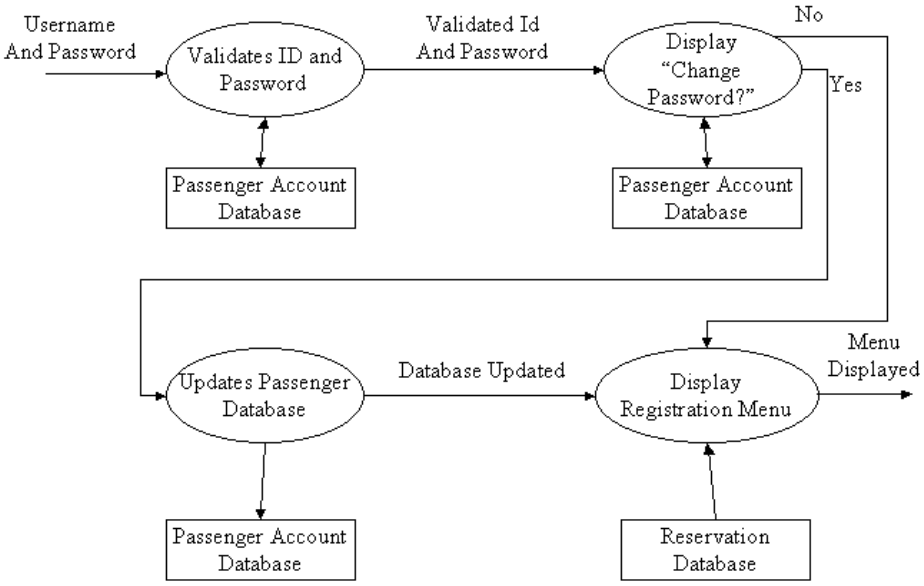
***uuu) Use case Diagram:***



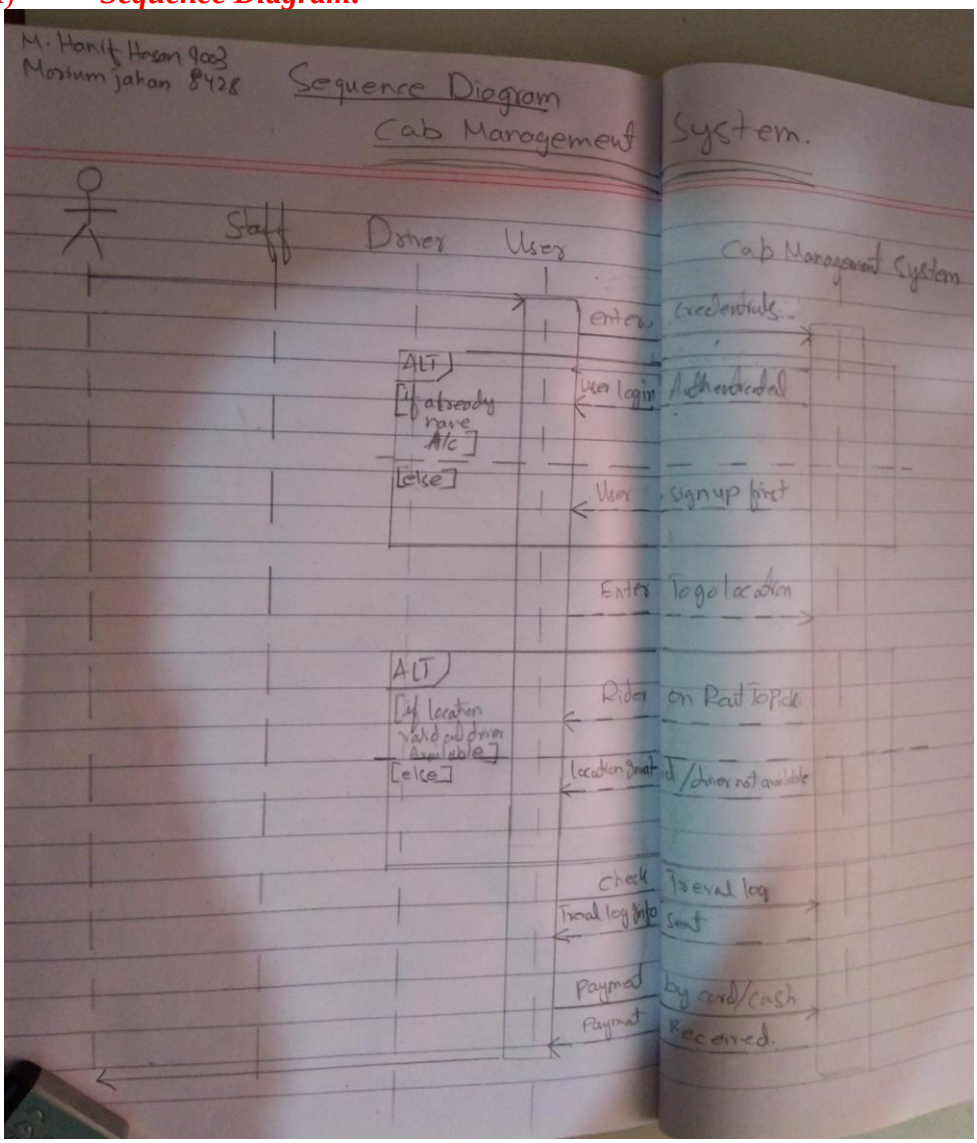

***vvv) Use case Realization:***

The realization was not needed.

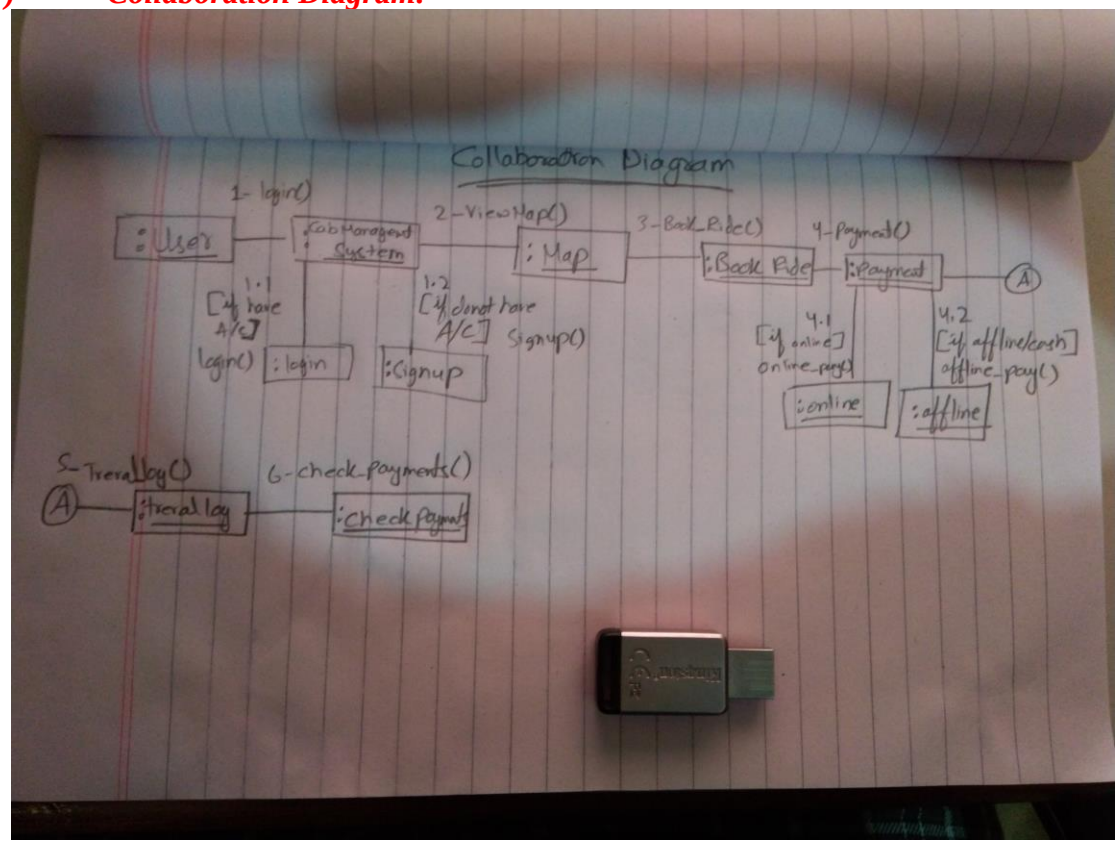
***www) Flow of Event or Data Flow Diagram:***


Username
And Password
Validates ID and
Password
Validated Id
And Password
Display
"Change
Password?"
No
Yes
Passenger Account
Database
Passenger Account
Database
Updates Passenger
Database
Database Updated
Display
Registration Menu
Menu
Displayed
Passenger Account
Database
Reservation
Database

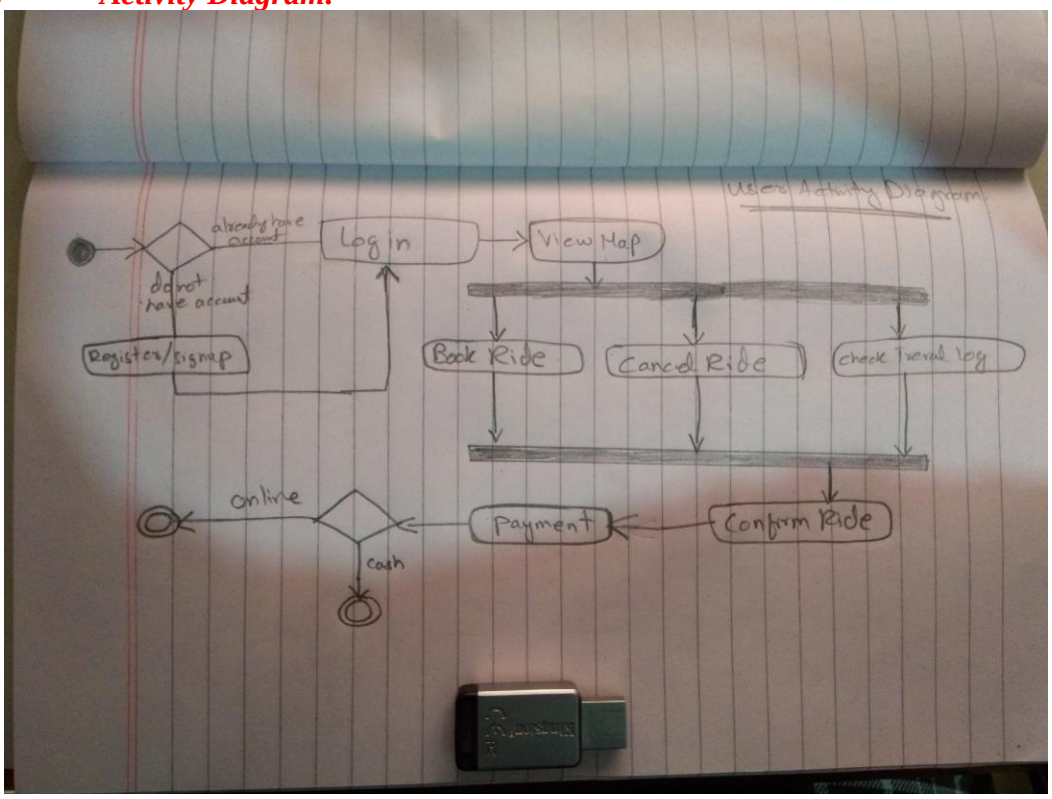
xxx) ***Sequence Diagram:***



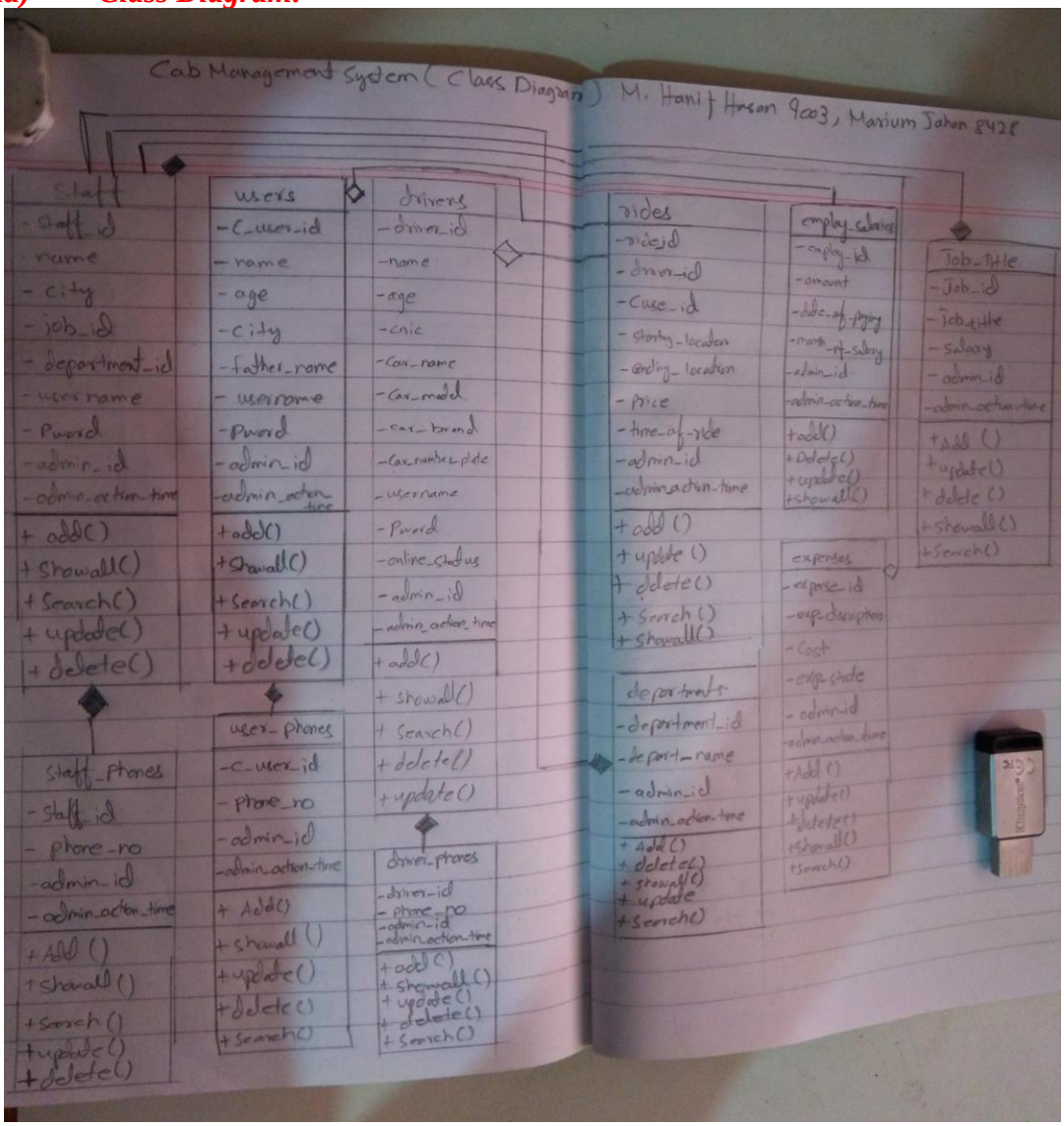
***yyy) Collaboration Diagram:***



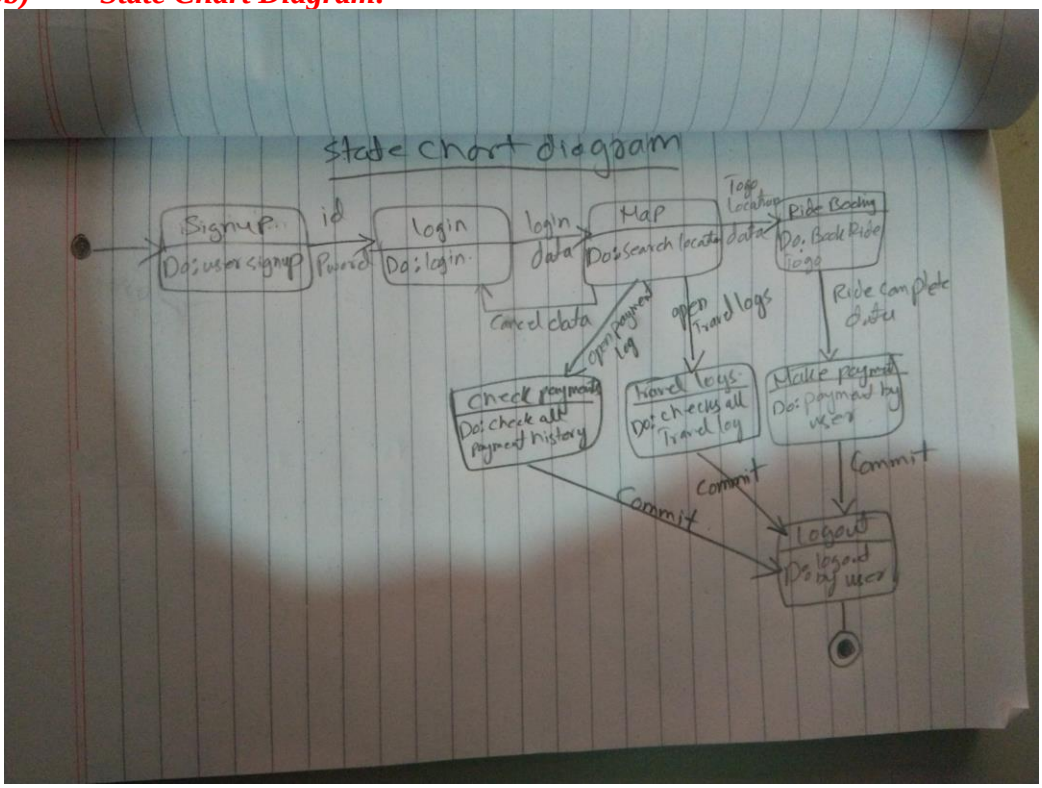

***zzz) Activity Diagram:***



***aaaa) Class Diagram:***



***bbbb) State Chart Diagram:***



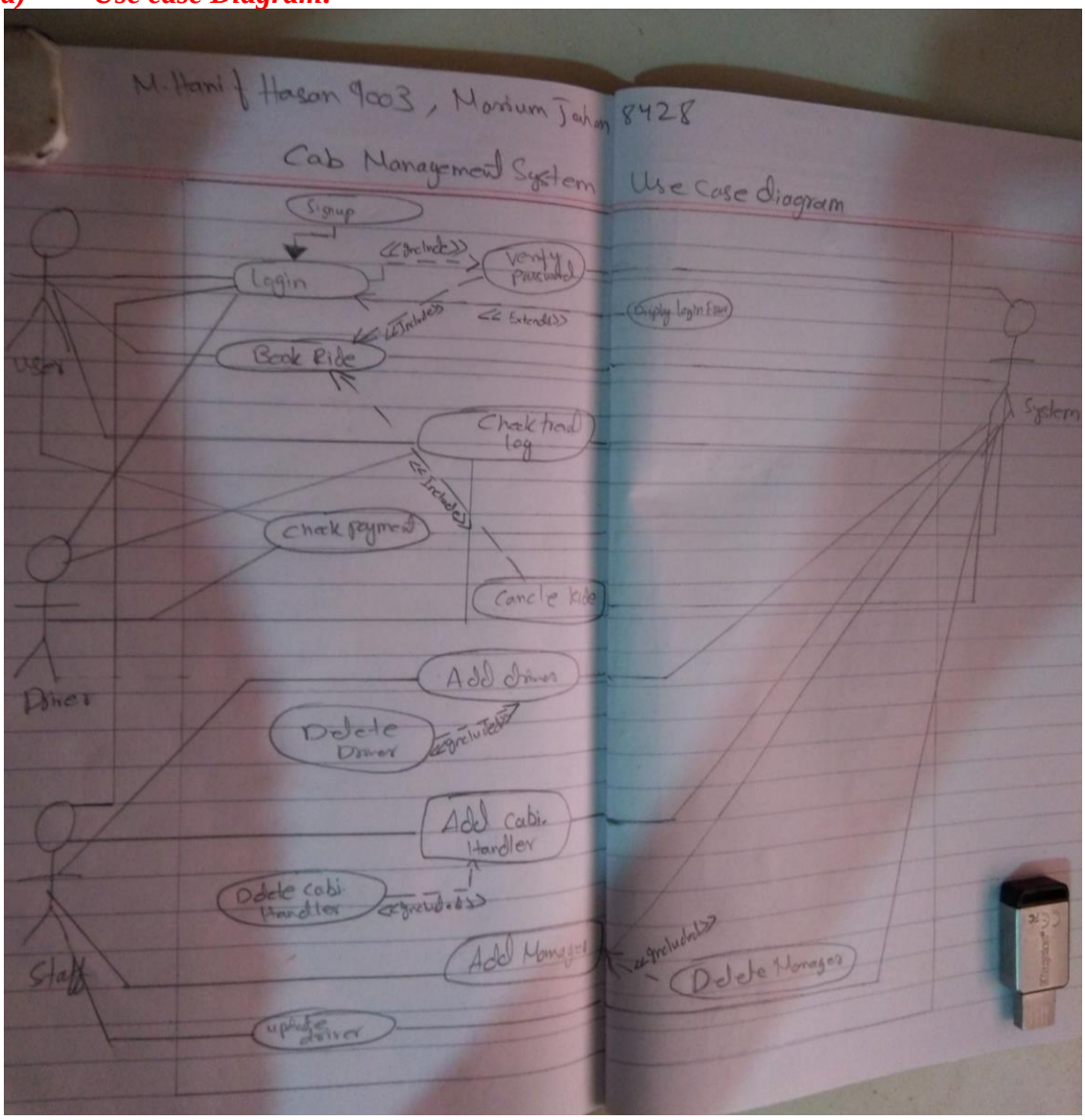

**3.1.6 Module 10 complete CRUD Driver_Phones**

i) ***Description:*** This modules Add,update,delete Driver_Phones.

cccc) ***Usage Scenario/ Use case Description/ Specification:***

| | |
|---|---|
| **Description** | [ make \| drop \| view \| update] to the Driver_Phnes Table |
| **Inputs** | Driver_Name,Driver_Phone… |
| **Source** | 19. The Module admin sets a Driver_Phones up<br>20. Press Button … |
| **Alternate Case** | |
| **Outputs** | Added \| Deleted \| Viewed \| Modified Driver_PHones |
| **Precondition** | If logedin via Module admin. |
| **Description** | [ make \| drop \| view \| update] to the Driver_Phones Table |

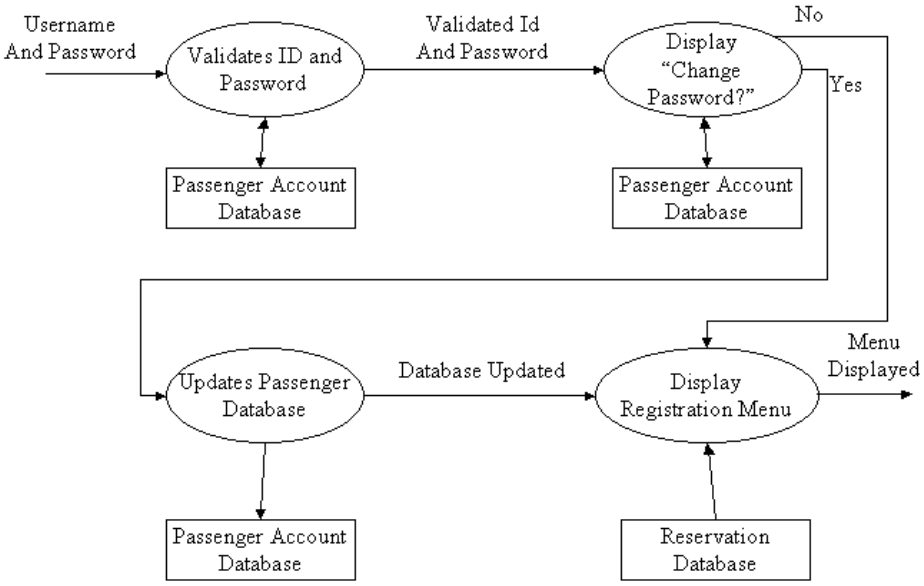
***dddd)*** ***Use case Diagram:***



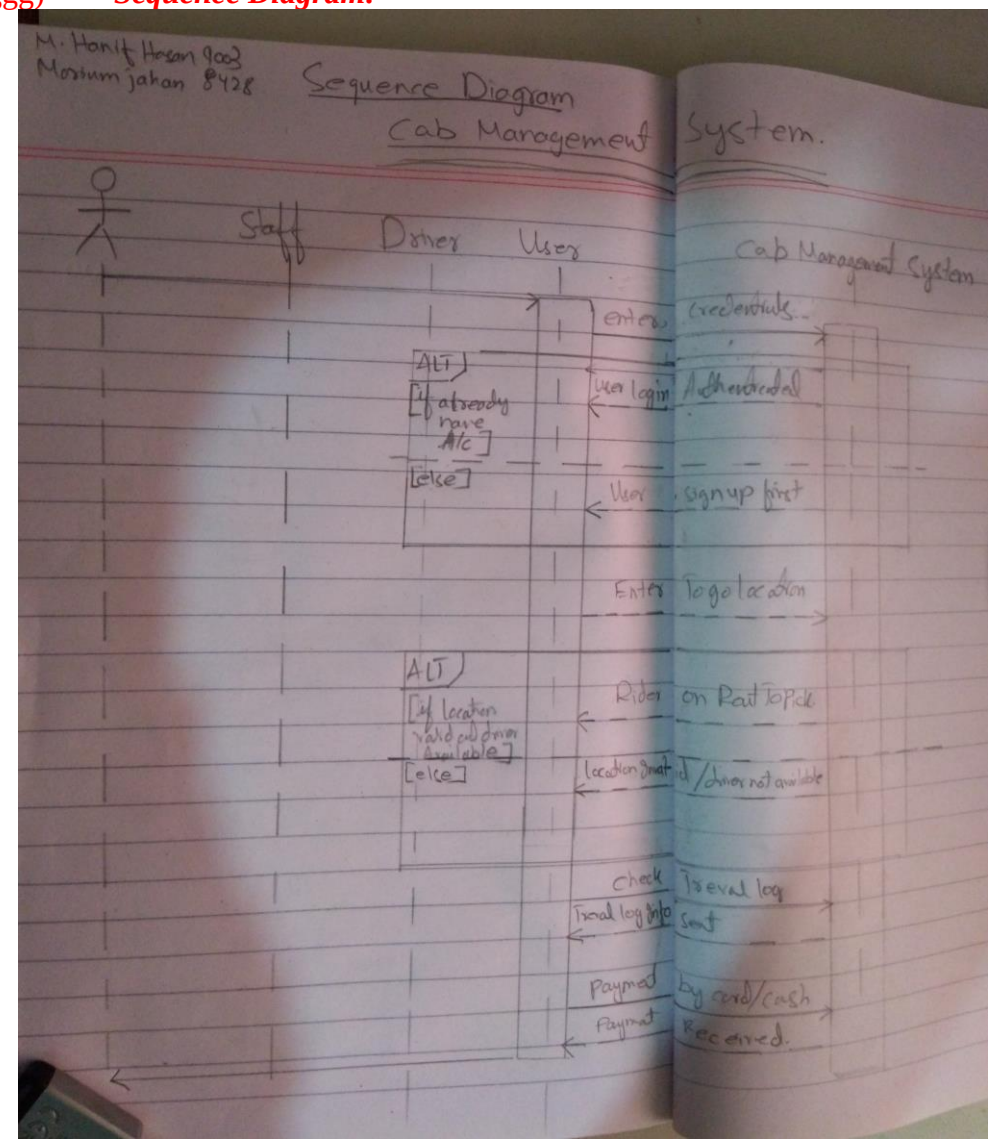

***eeee)*** ***Use case Realization:***

The realization was not needed.

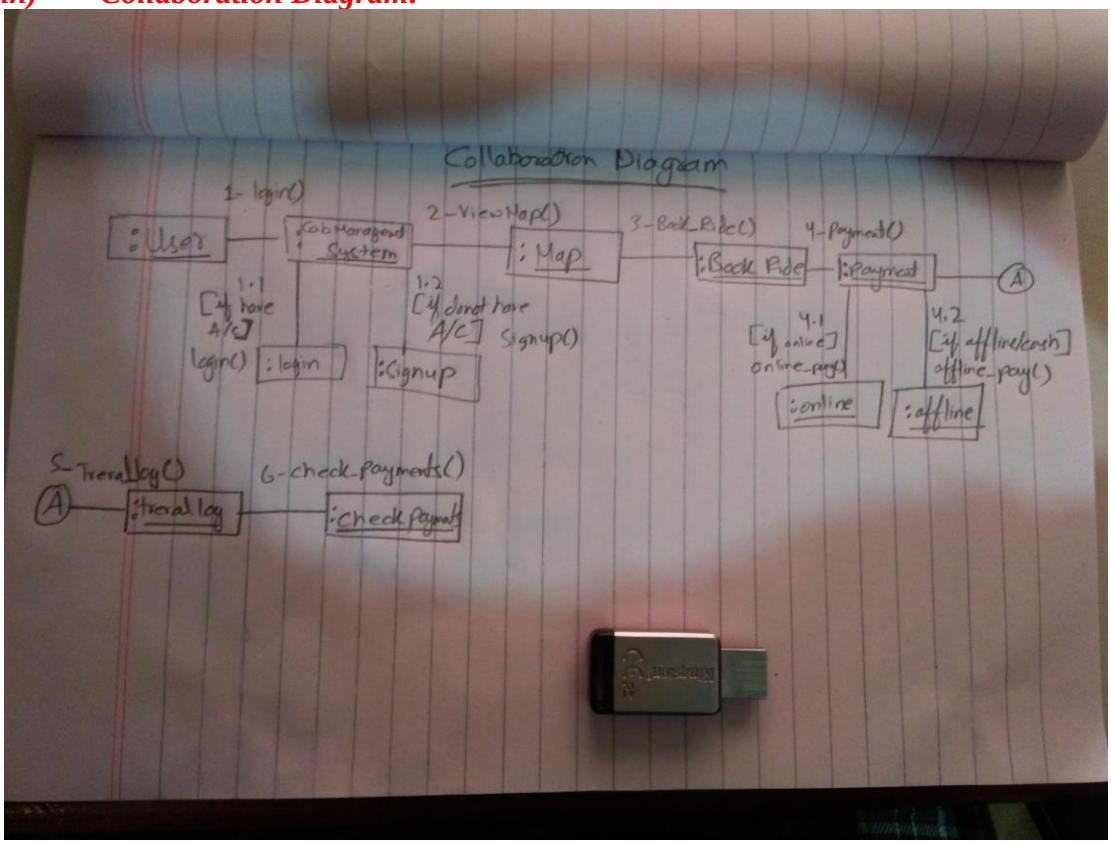
***ffff)*** ***Flow of Event or Data Flow Diagram:***


Username
And Password
Validates ID and
Password
Validated Id
And Password
Display
"Change
Password?"
No
Yes
Passenger Account
Database
Passenger Account
Database
Updates Passenger
Database
Database Updated
Display
Registration Menu
Menu
Displayed
Passenger Account
Database
Reservation
Database

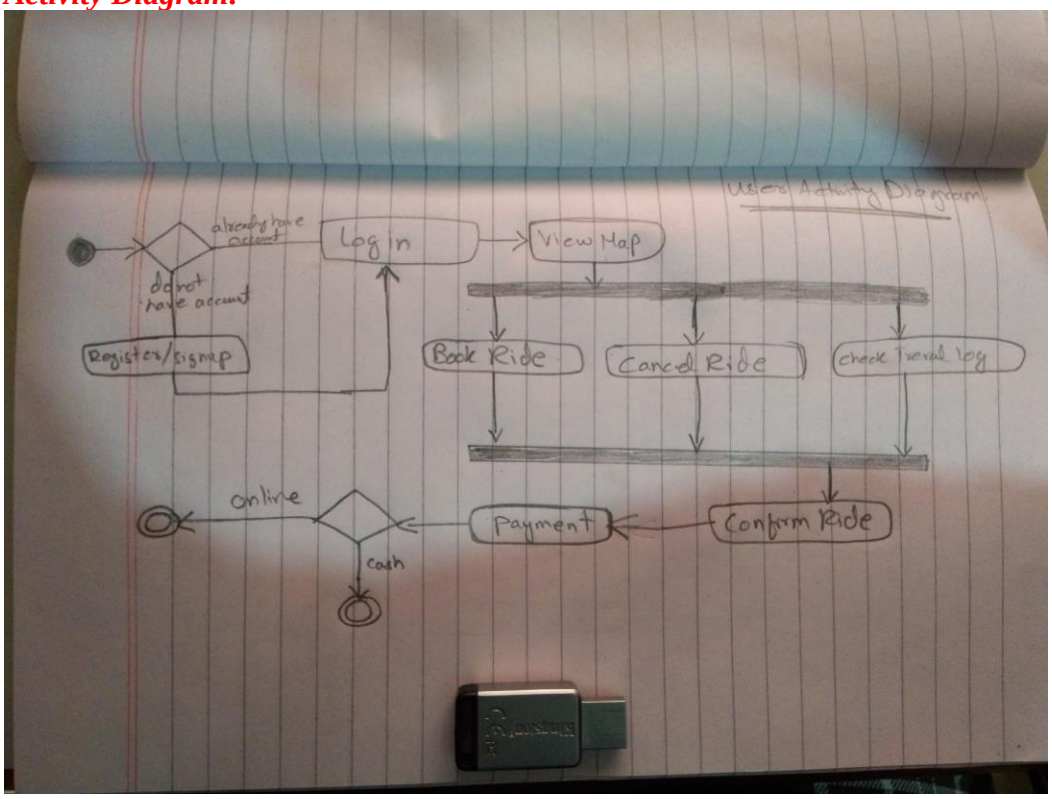
gggg) ***Sequence Diagram:***



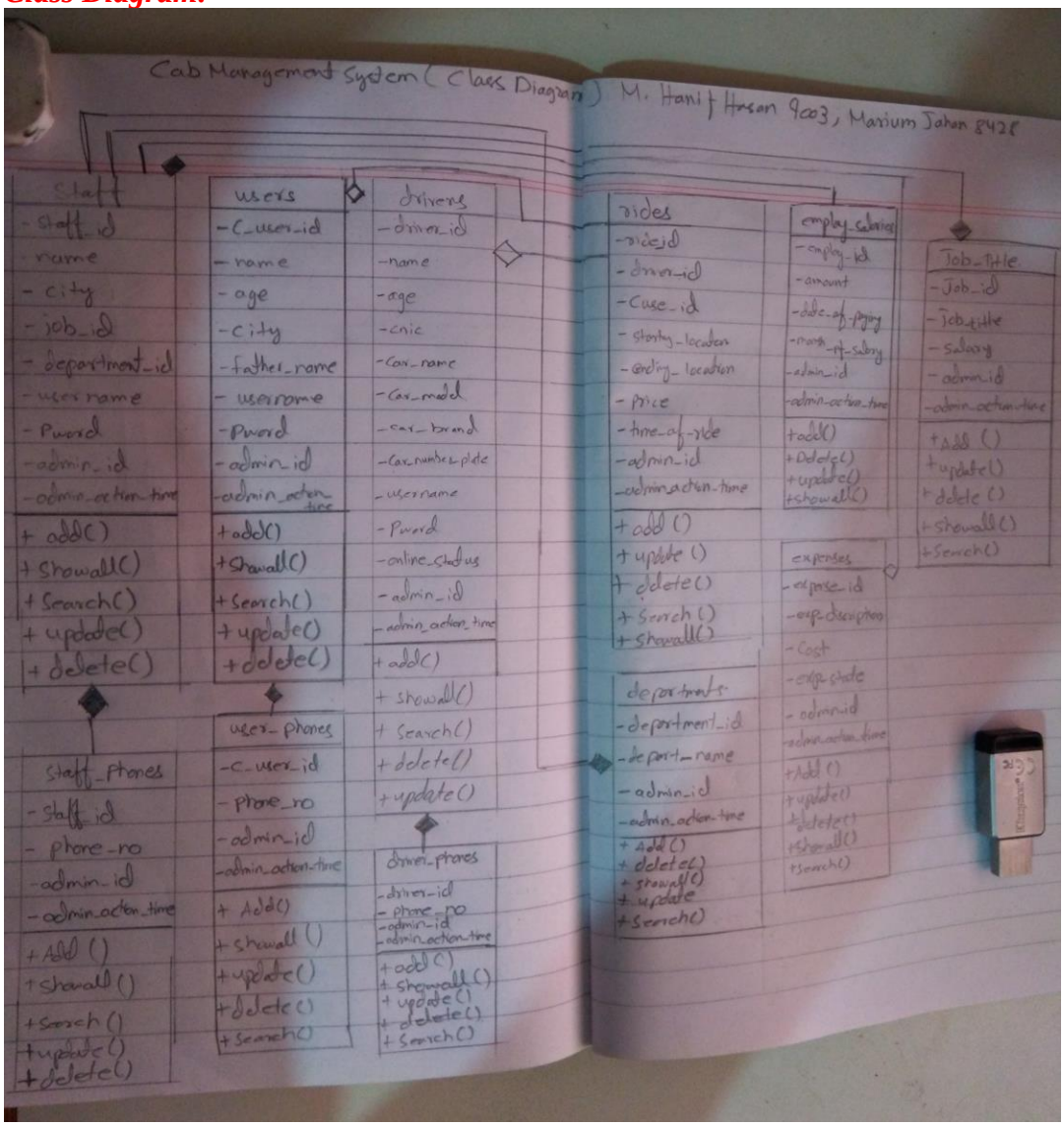
***hhhh) Collaboration Diagram:***



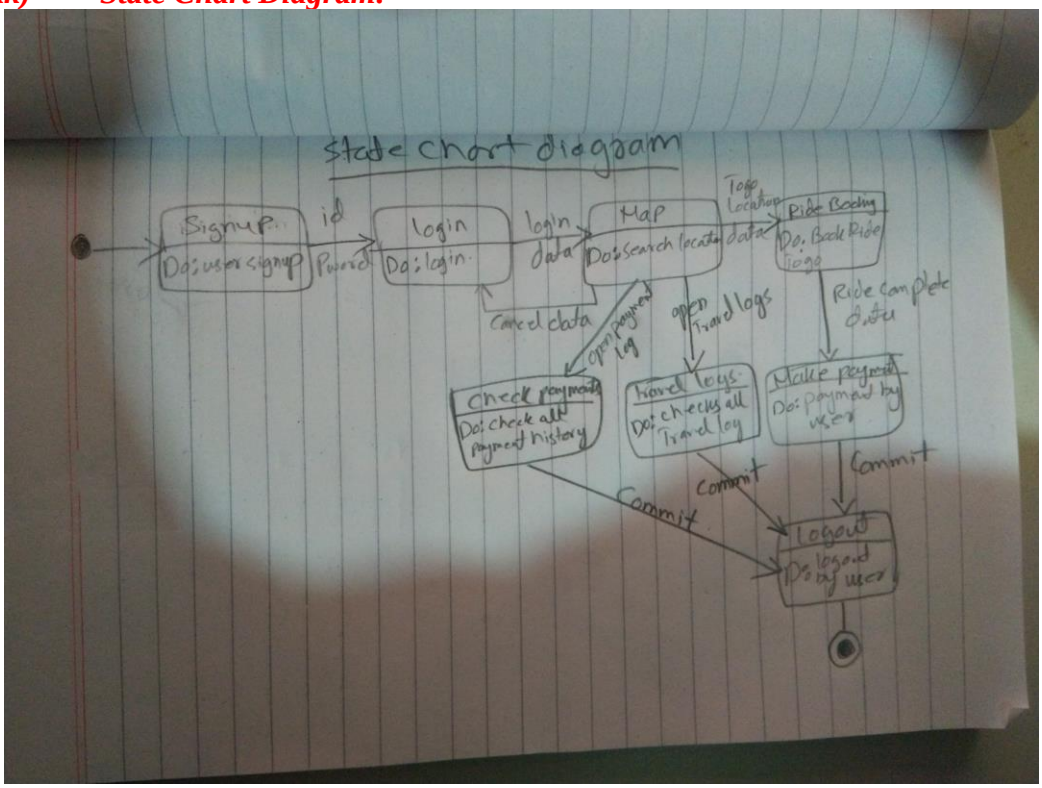

***iiii)Activity Diagram:***



***jjjj) Class Diagram:***



***kkkk) State Chart Diagram:***



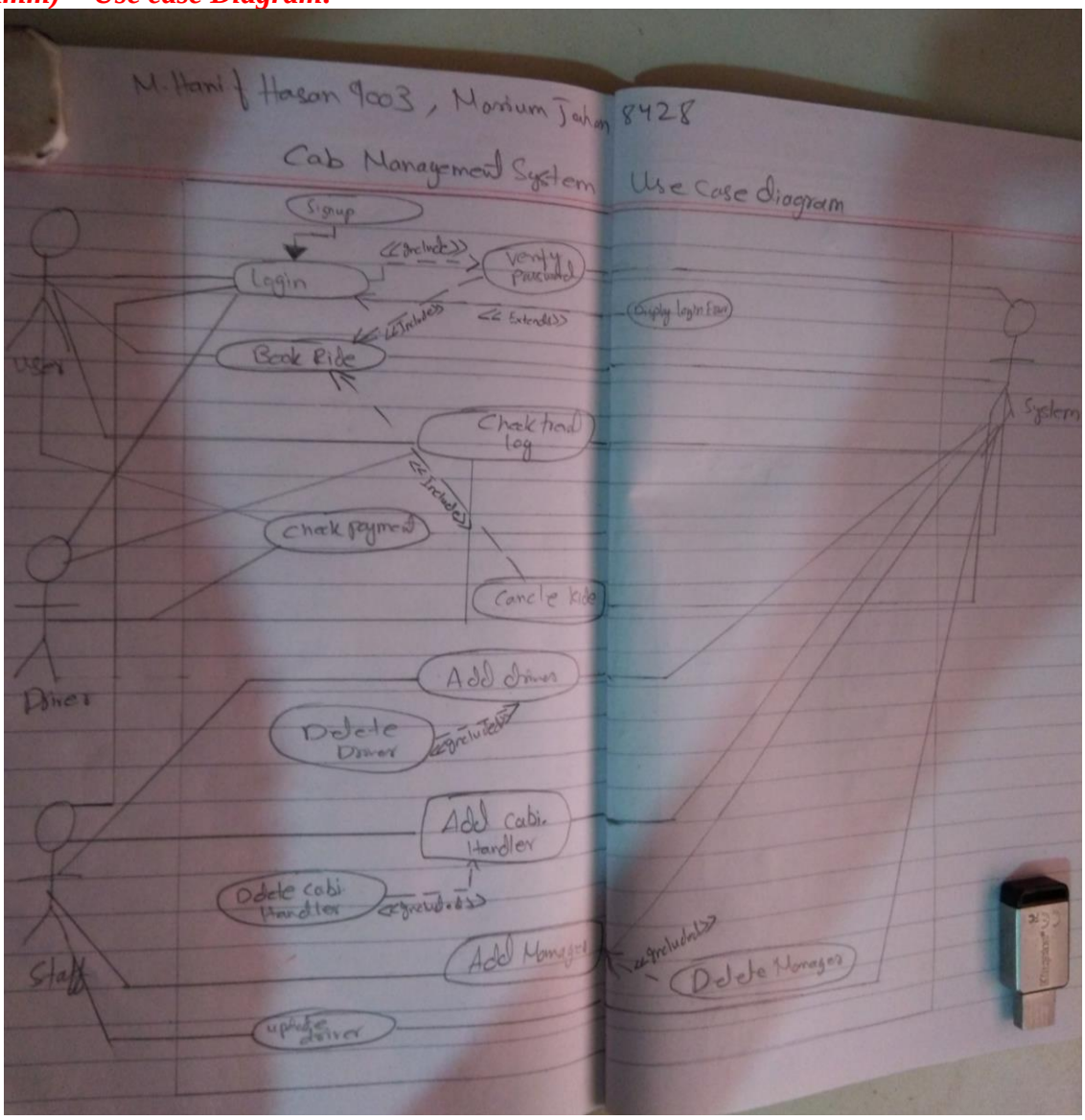

### 3.1.6 Module 11 complete CRUD Departments

j) ***Description:*** This modules Add,update,delete Departments.

llll) ***Usage Scenario/ Use case Description/ Specification:***

| | |
|---|---|
| **Description** | [ make \| drop \| view \| update] to the Departments Table |
| **Inputs** | Department_name,Department_discription… |
| **Source** | 21. The Module admin sets a Departments up<br>22. Press Button … |
| **Alternate Case** | |
| **Outputs** | Added \| Deleted \| Viewed \| Modified Departments |
| **Precondition** | If logedin via Module admin. |
| **Description** | [ make \| drop \| view \| update] to the Departments Table |

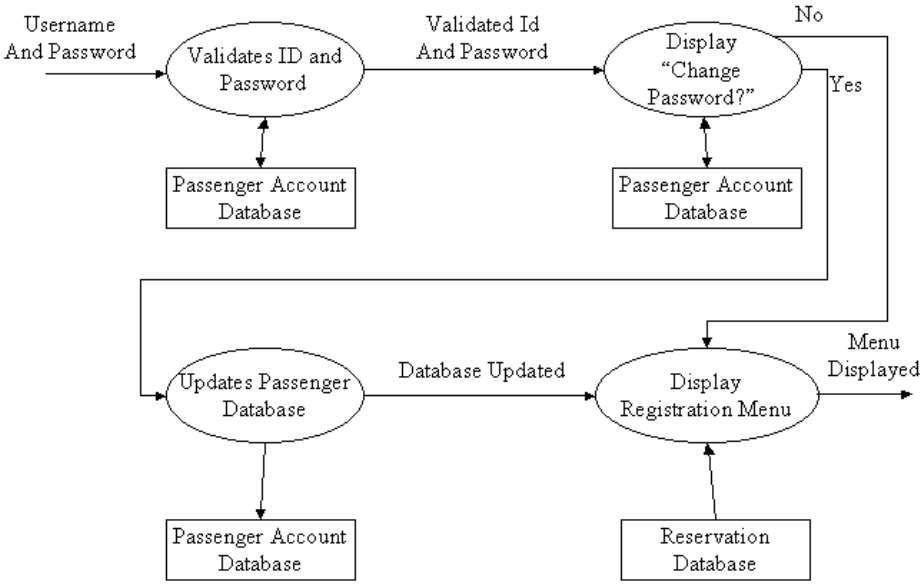
***mmmm) Use case Diagram:***



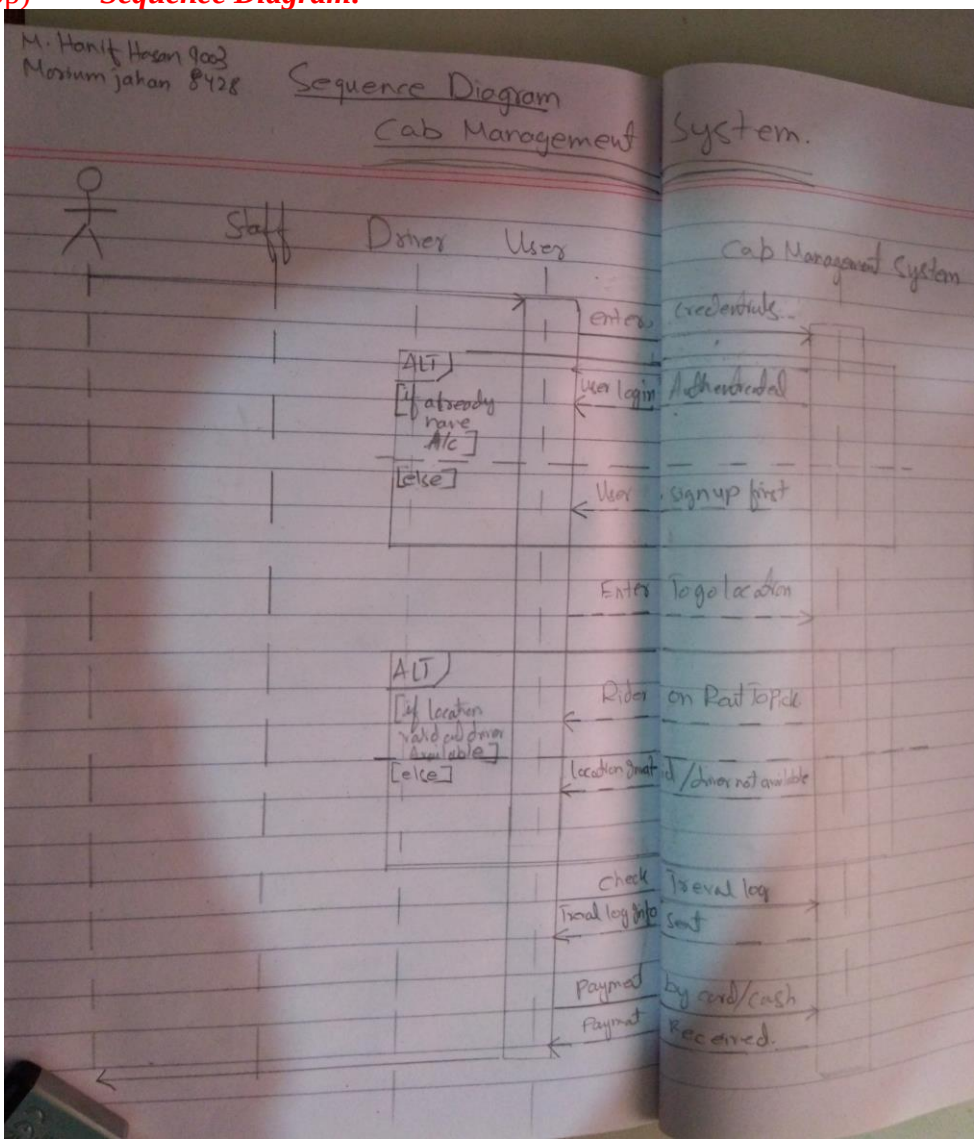

***nnnn) Use case Realization:***

The realization was not needed.

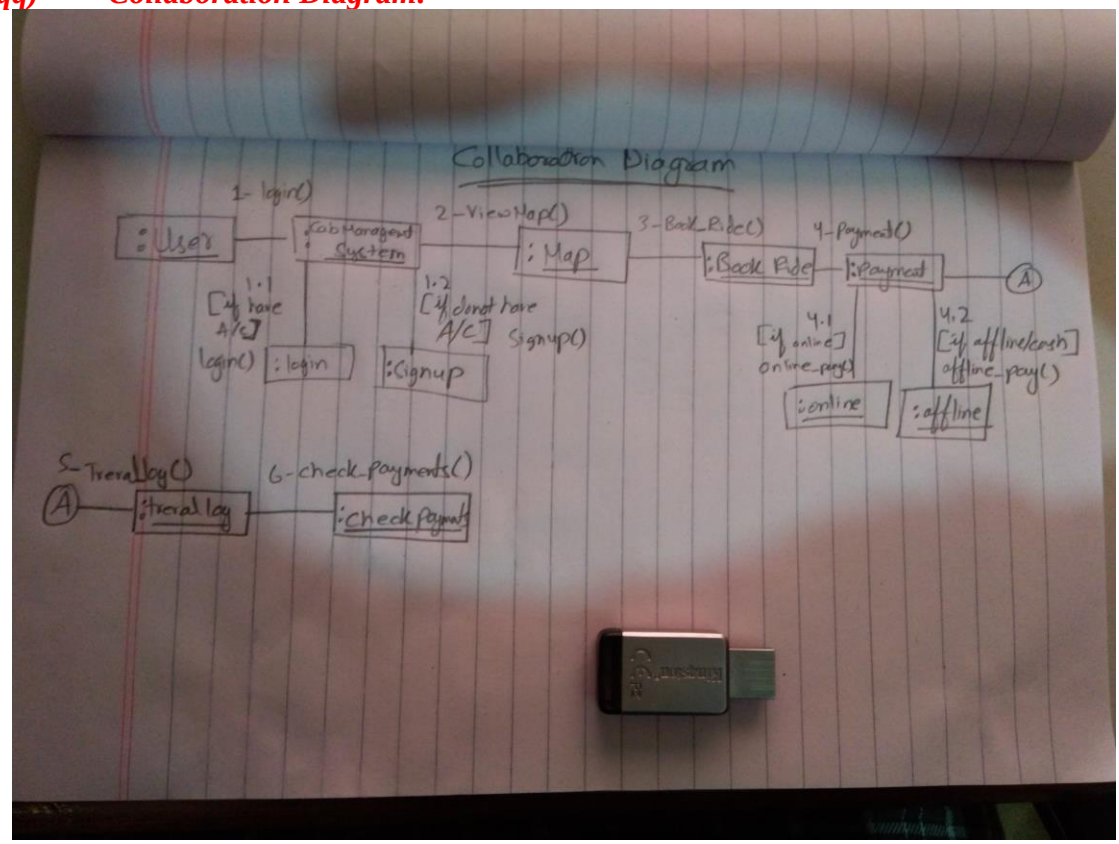
***oooo) Flow of Event or Data Flow Diagram:***


Username
And Password
Validates ID and
Password
Validated Id
And Password
Display
"Change
Password?"
No
Yes
Passenger Account
Database
Passenger Account
Database
Updates Passenger
Database
Database Updated
Display
Registration Menu
Menu
Displayed
Passenger Account
Database
Reservation
Database

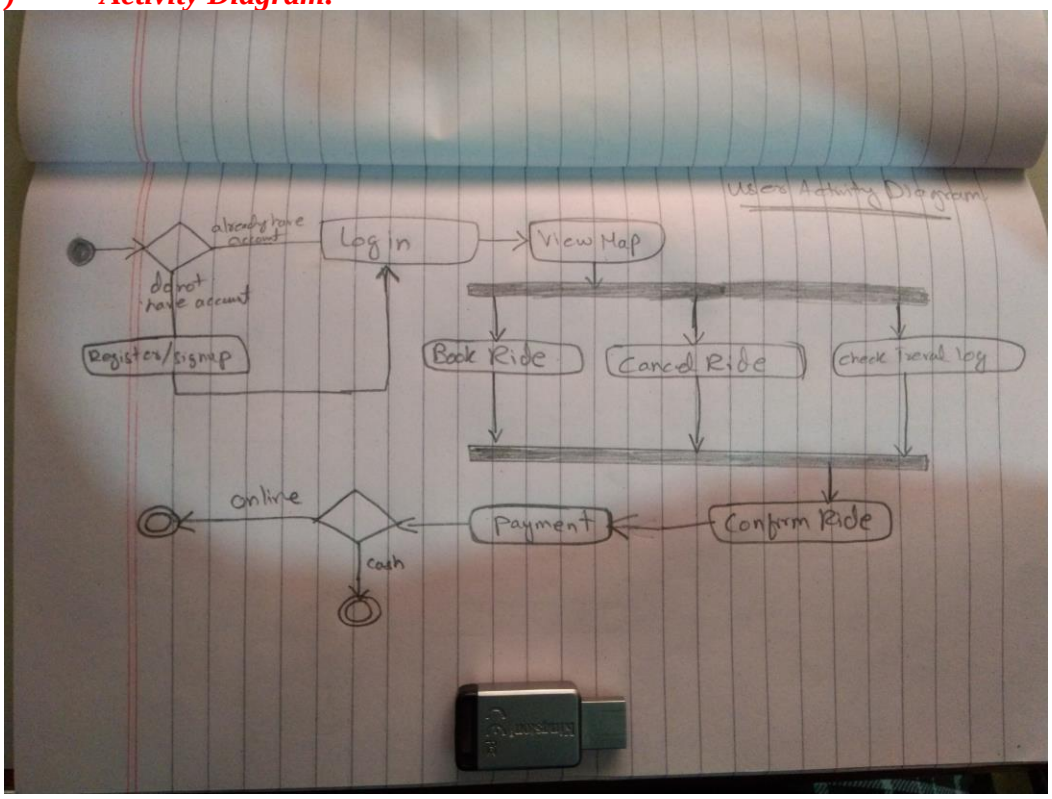
pppp) ***Sequence Diagram:***



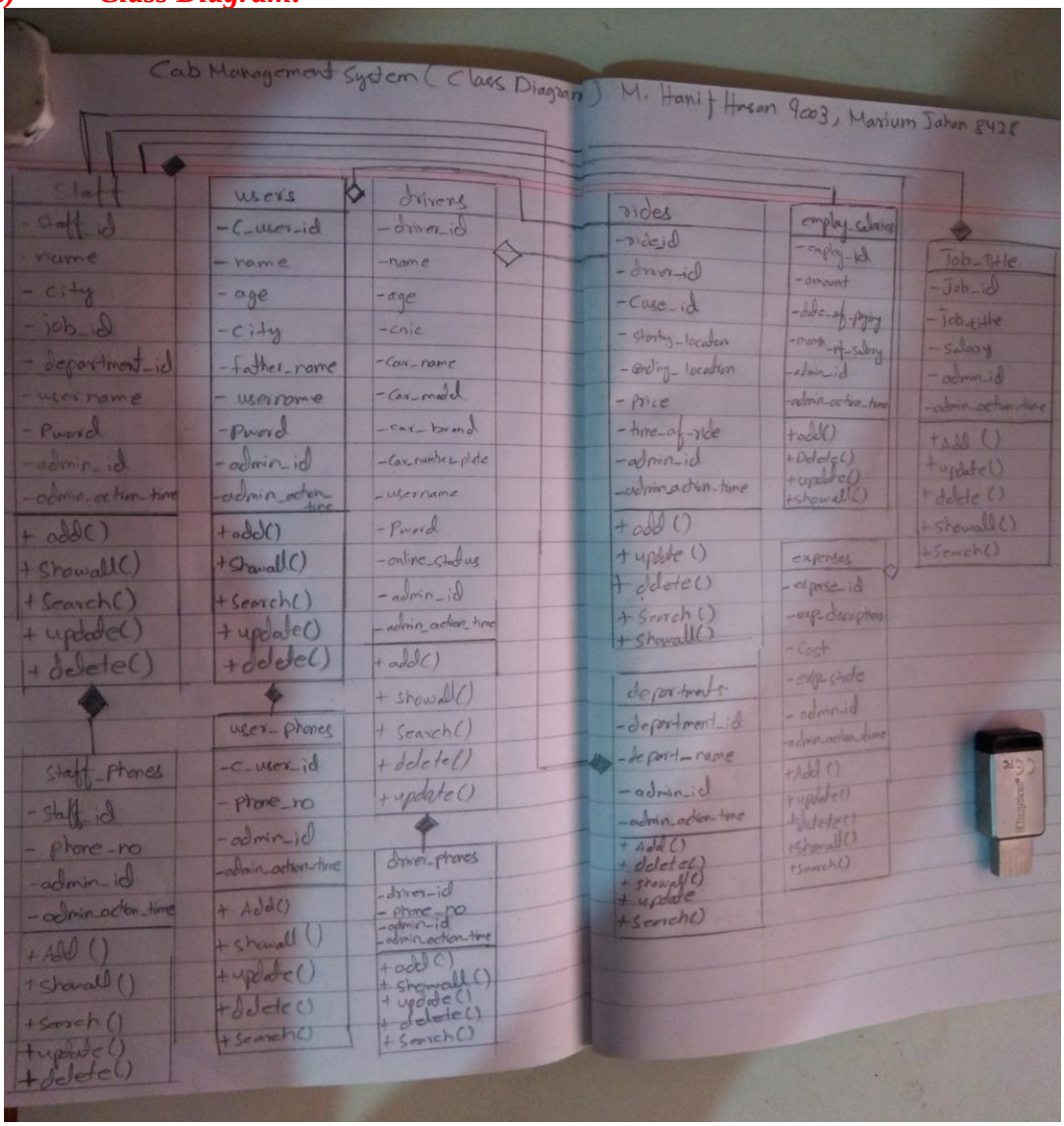
***qqqq) Collaboration Diagram:***



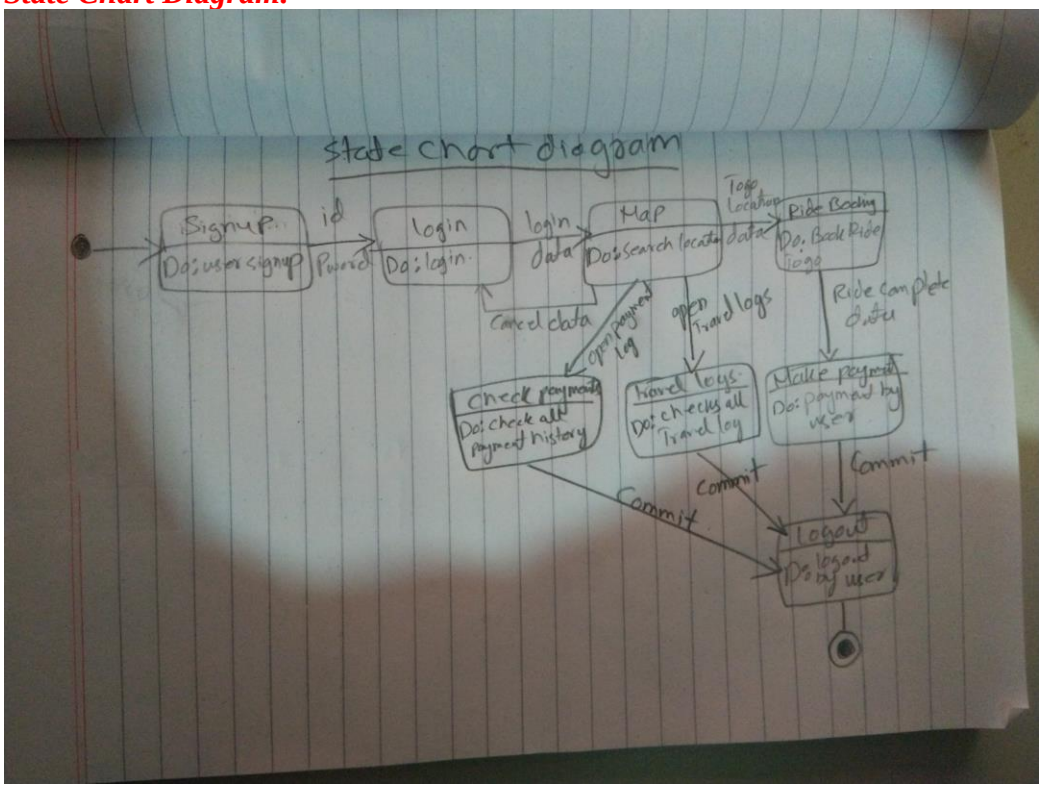

***rrrr) Activity Diagram:***



***ssss)*** ***Class Diagram:***



***tttt) State Chart Diagram:***



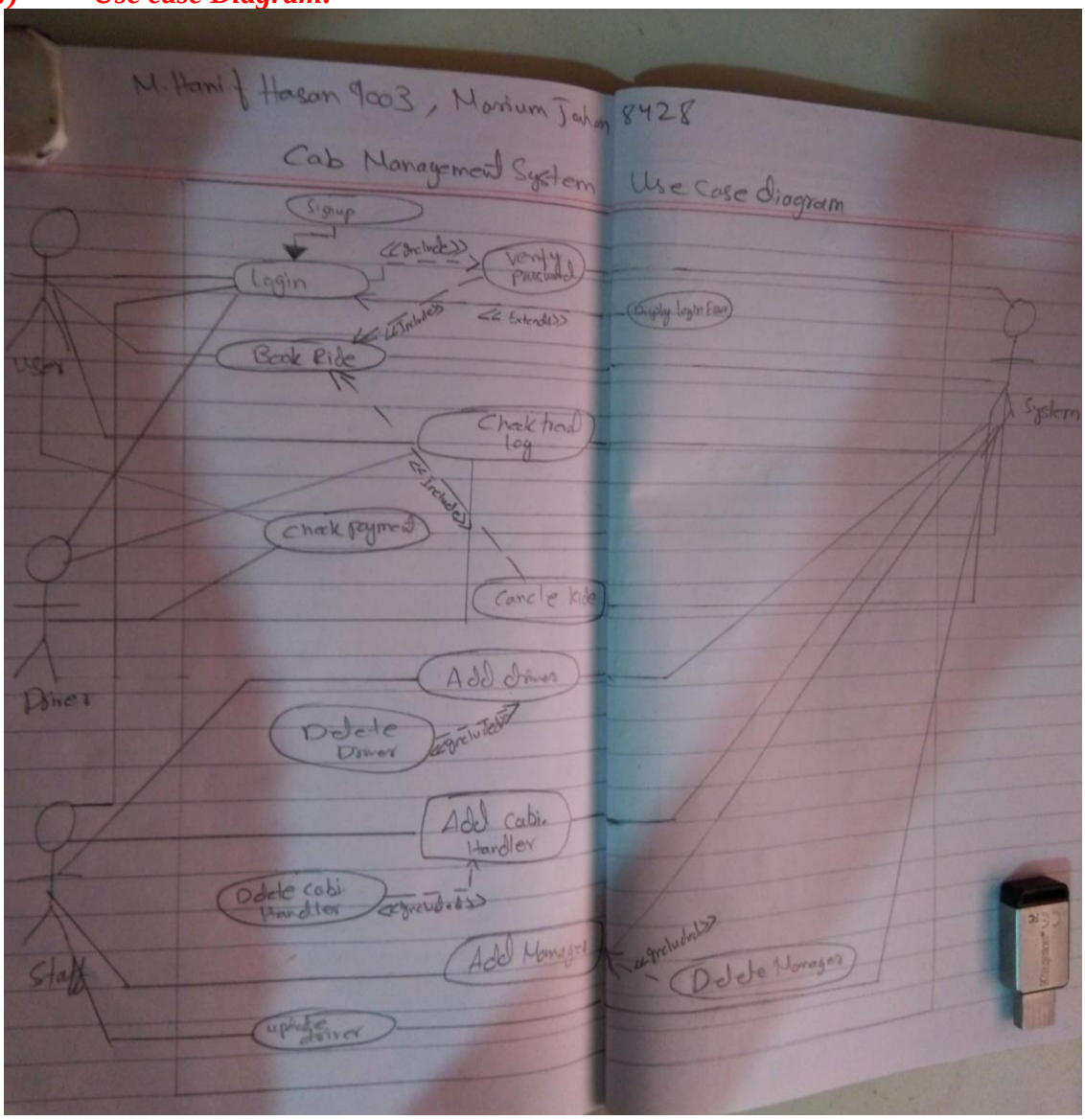

### 3.1.6 Module 12 complete CRUD Expenses

k) ***Description:*** This modules Add,update,delete Expenses.

uuuu) ***Usage Scenario/ Use case Description/ Specification:***

| | |
|---|---|
| **Description** | [ make \| drop \| view \| update] to the Expenses Table |
| **Inputs** | Expense_name, Expense_ammount… |
| **Source** | 23. The Module admin sets a Expenses up<br>24. Press Button … |
| **Alternate Case** | |
| **Outputs** | Added \| Deleted \| Viewed \| Modified Expenses |
| **Precondition** | If logedin via Module admin. |
| **Description** | [ make \| drop \| view \| update] to the Expenses Table |

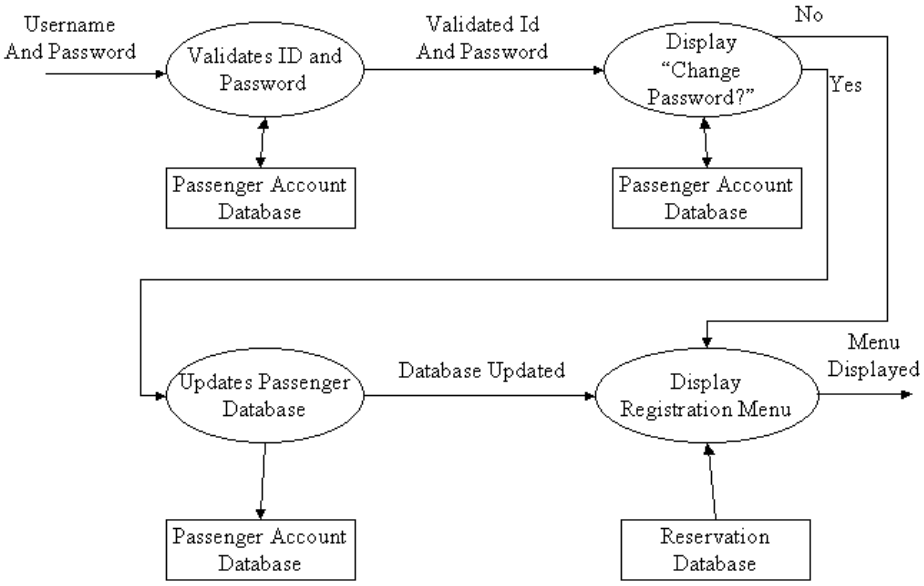
***vvv) Use case Diagram:***



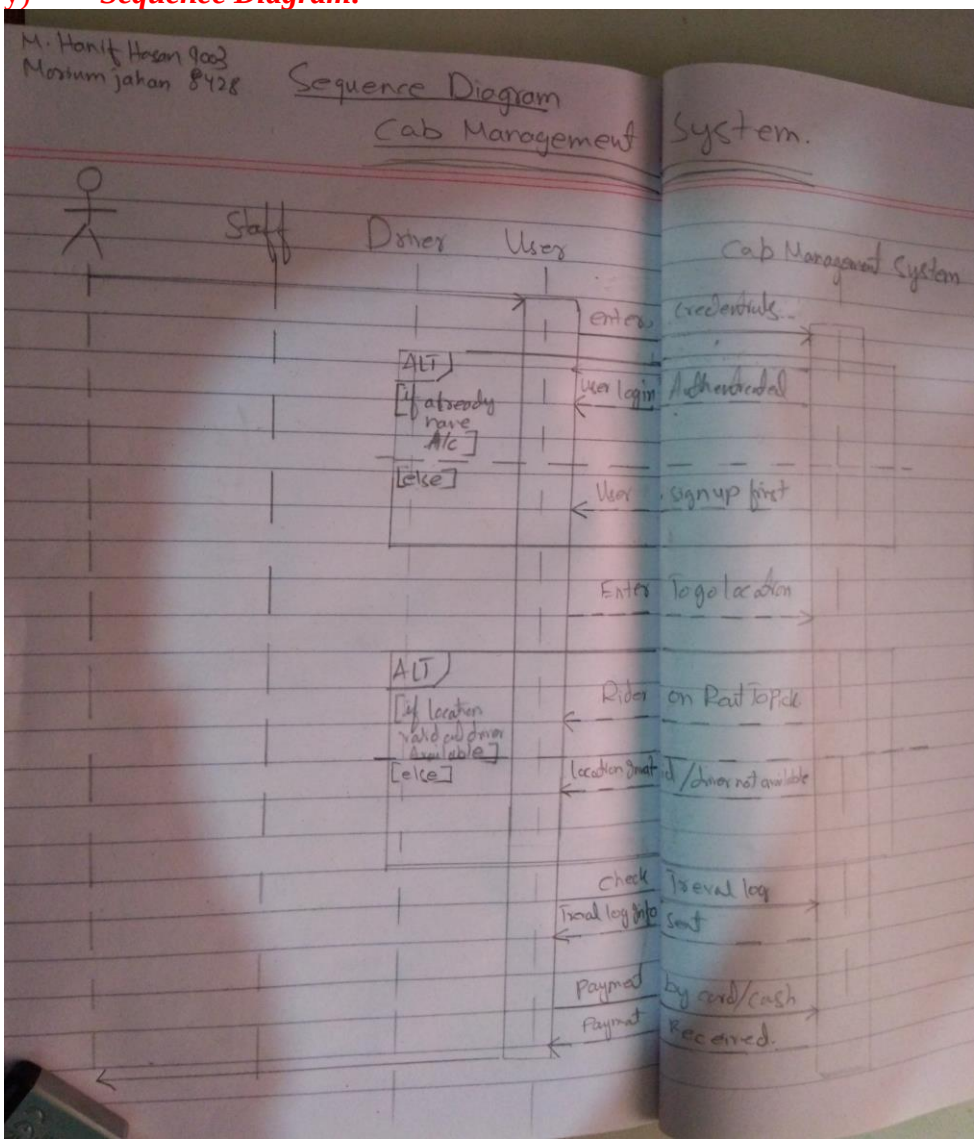

***wwww) Use case Realization:***

The realization was not needed.

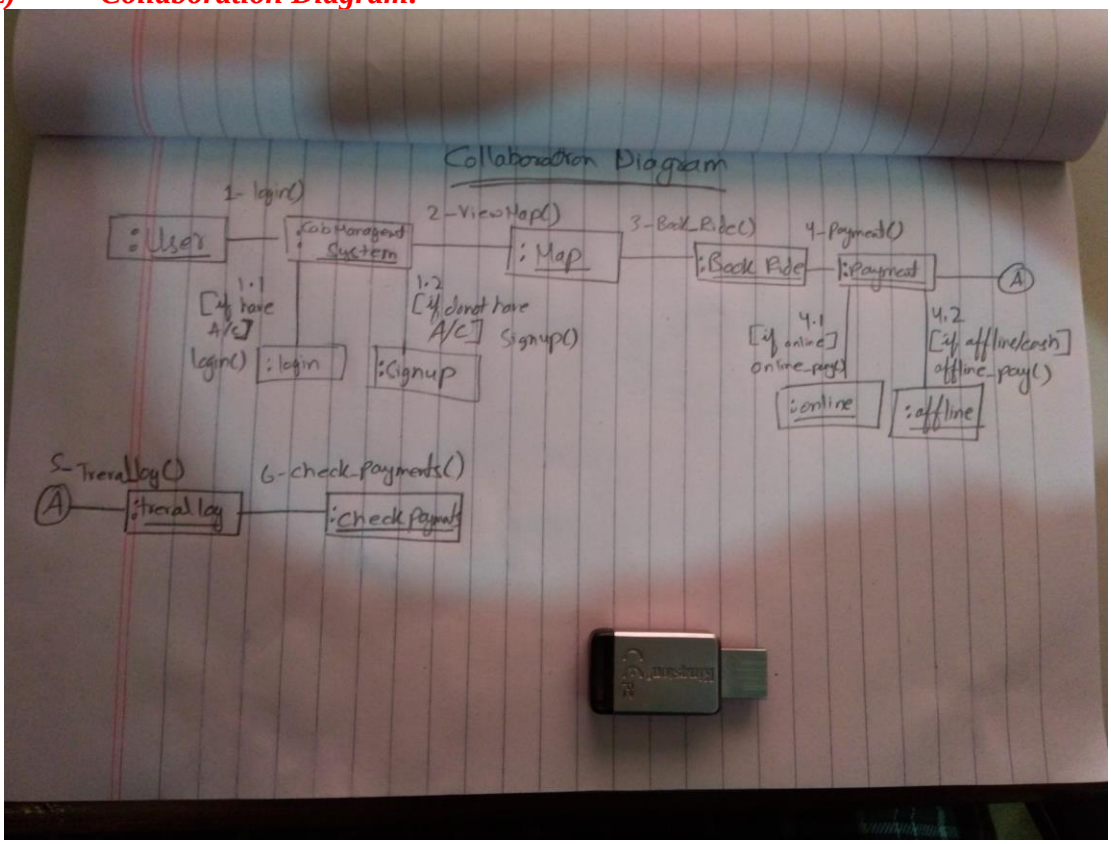
***xxxx) Flow of Event or Data Flow Diagram:***


Username
And Password
Validates ID and
Password
Validated Id
And Password
Display
"Change
Password?"
No
Yes
Passenger Account
Database
Passenger Account
Database
Updates Passenger
Database
Database Updated
Display
Registration Menu
Menu
Displayed
Passenger Account
Database
Reservation
Database

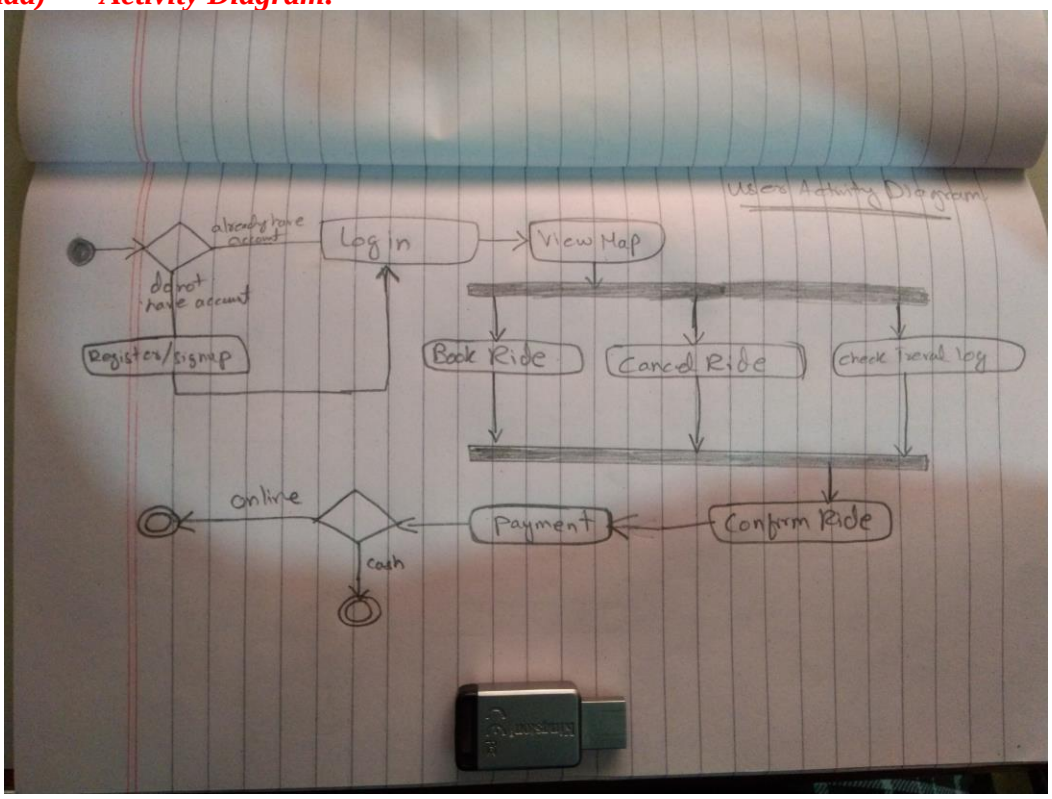
yyyy) ***Sequence Diagram:***



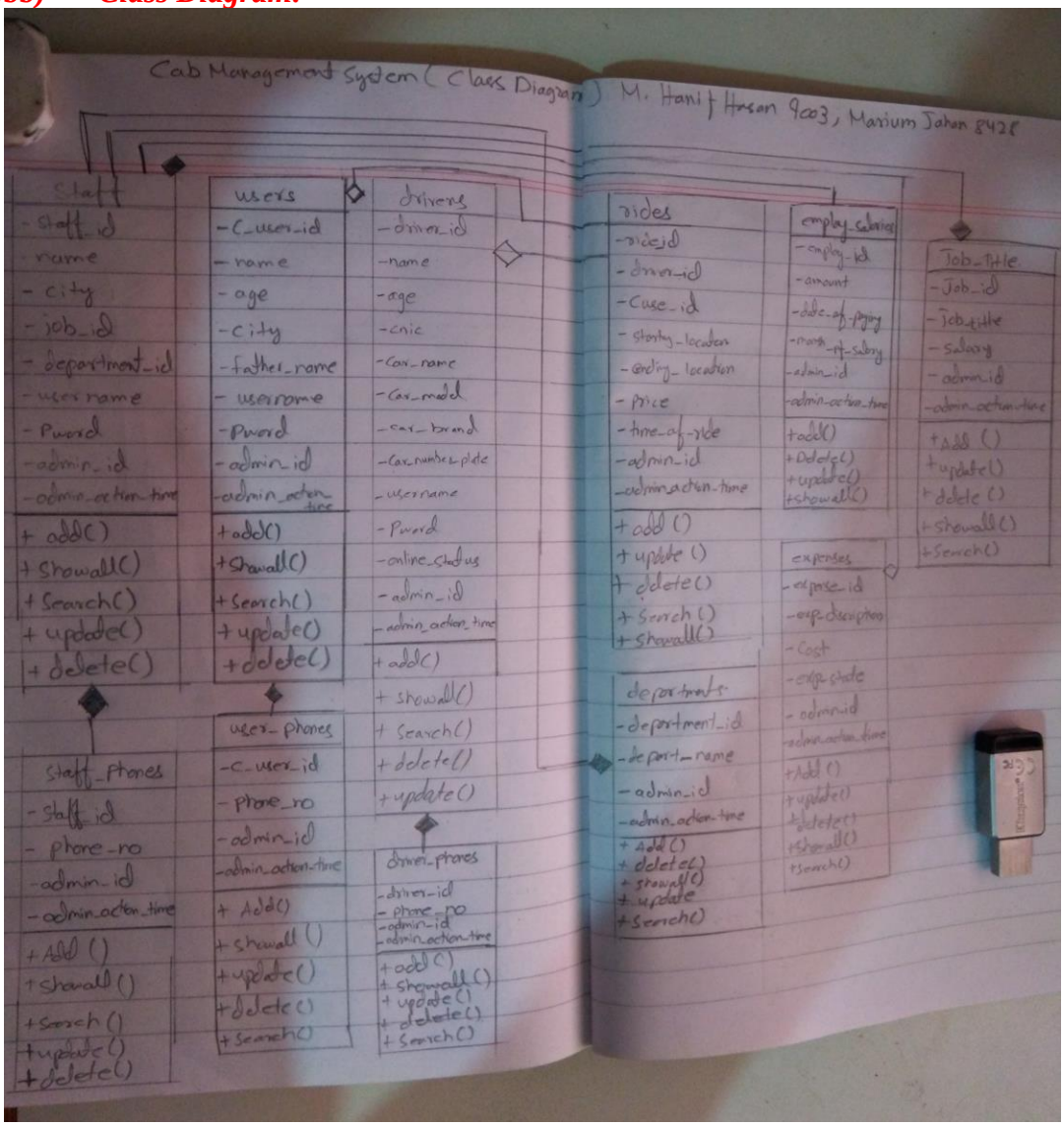
***zzzz) Collaboration Diagram:***



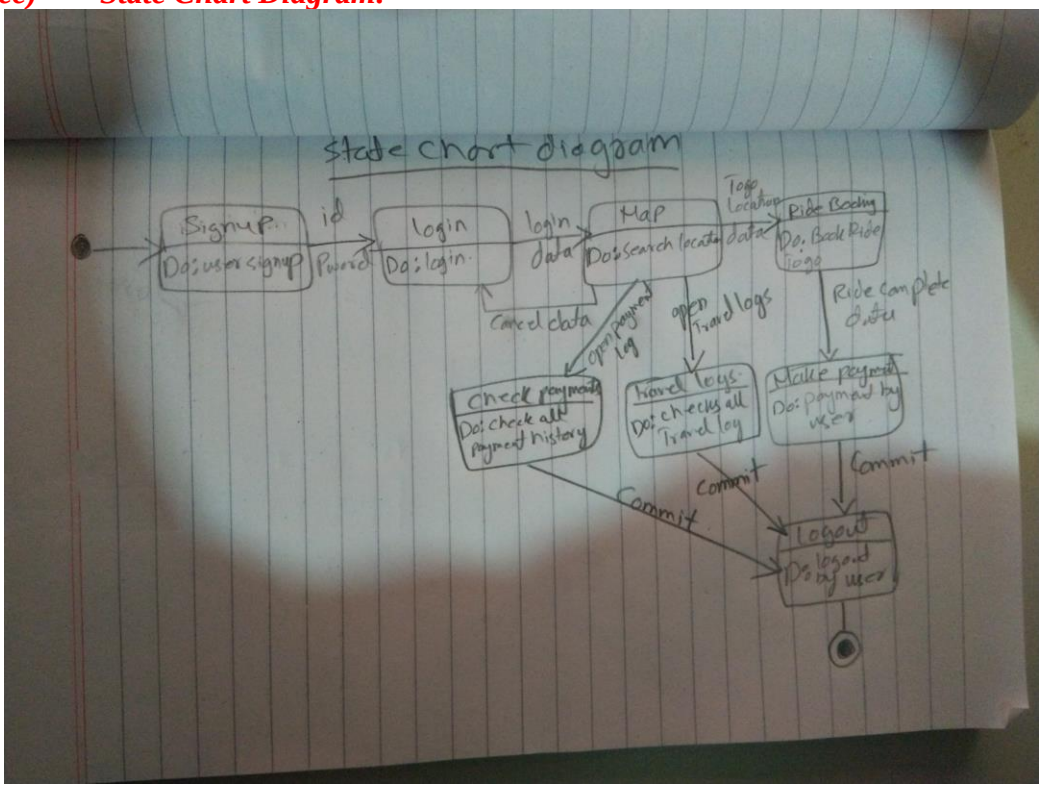

***aaaaa) Activity Diagram:***



***bbbbb) Class Diagram:***



***ccccc) State Chart Diagram:***

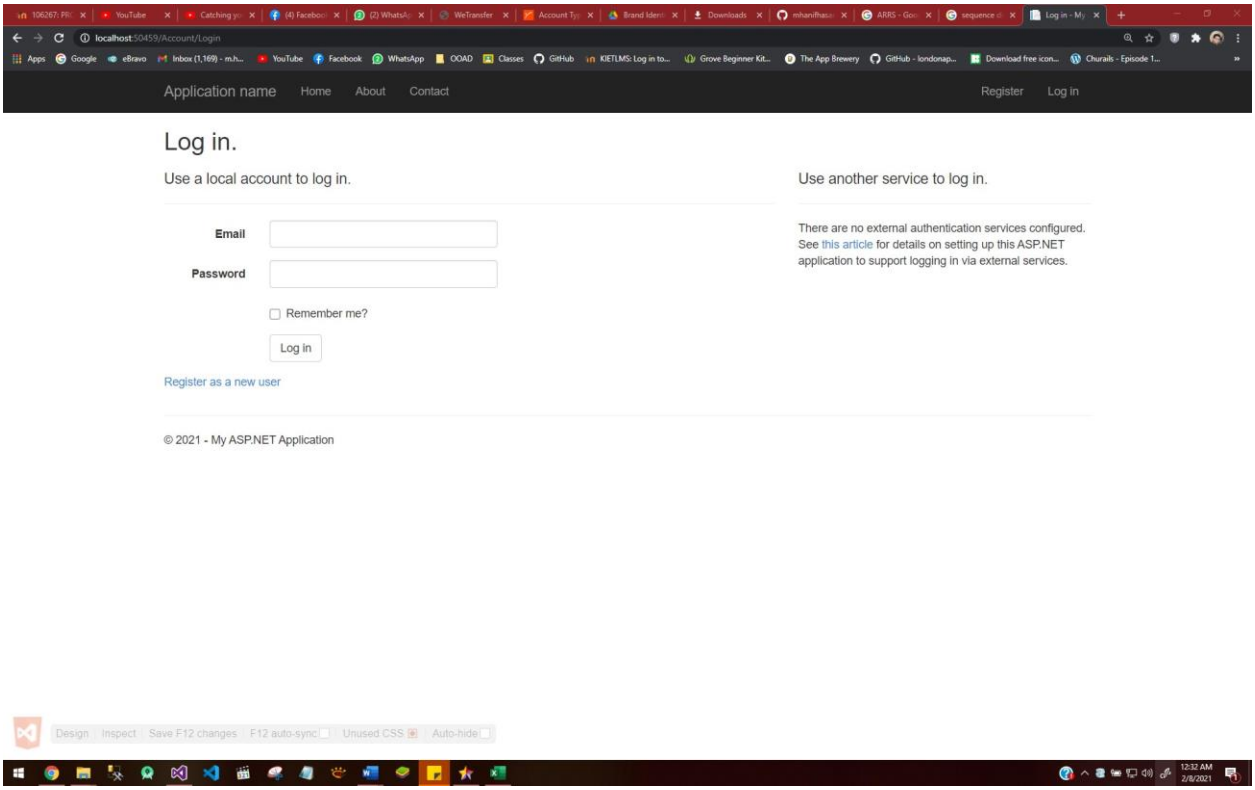


## 3.2. External Interface Requirements

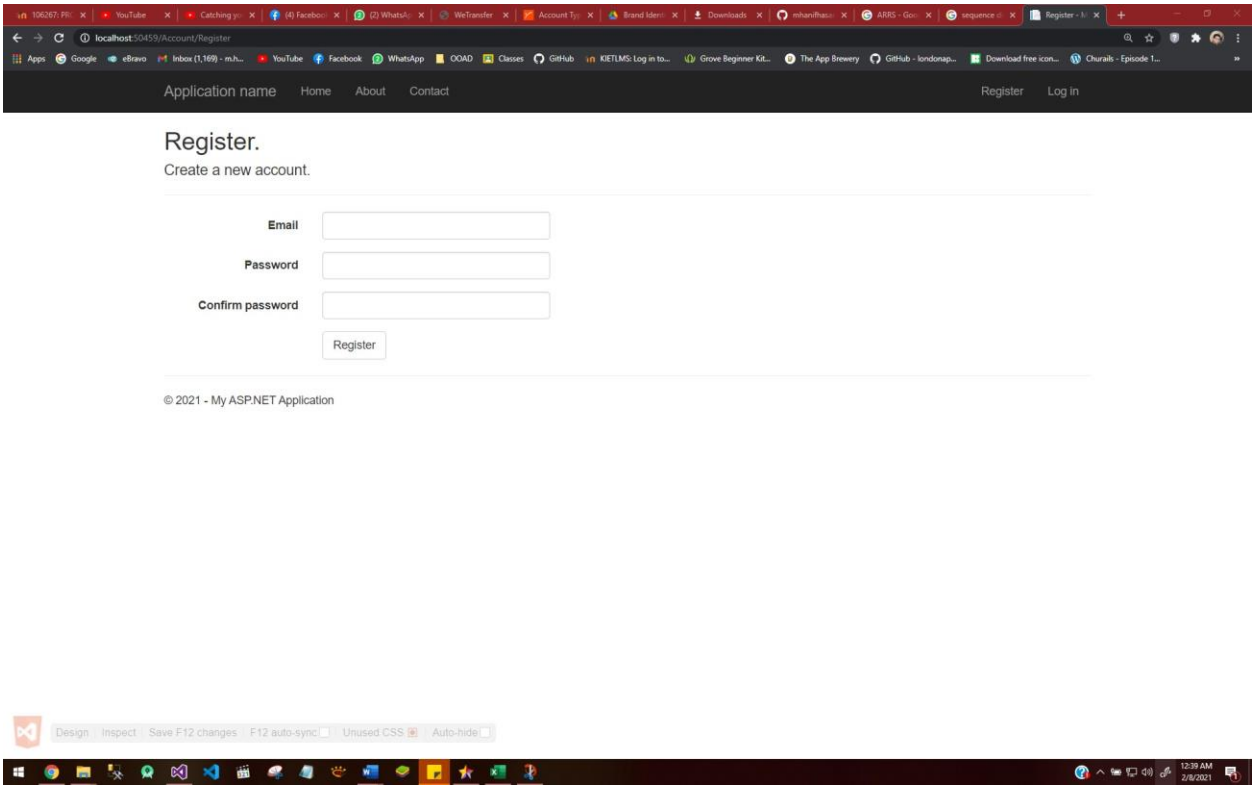
### 3.2.1 User Interfaces

Login Page:

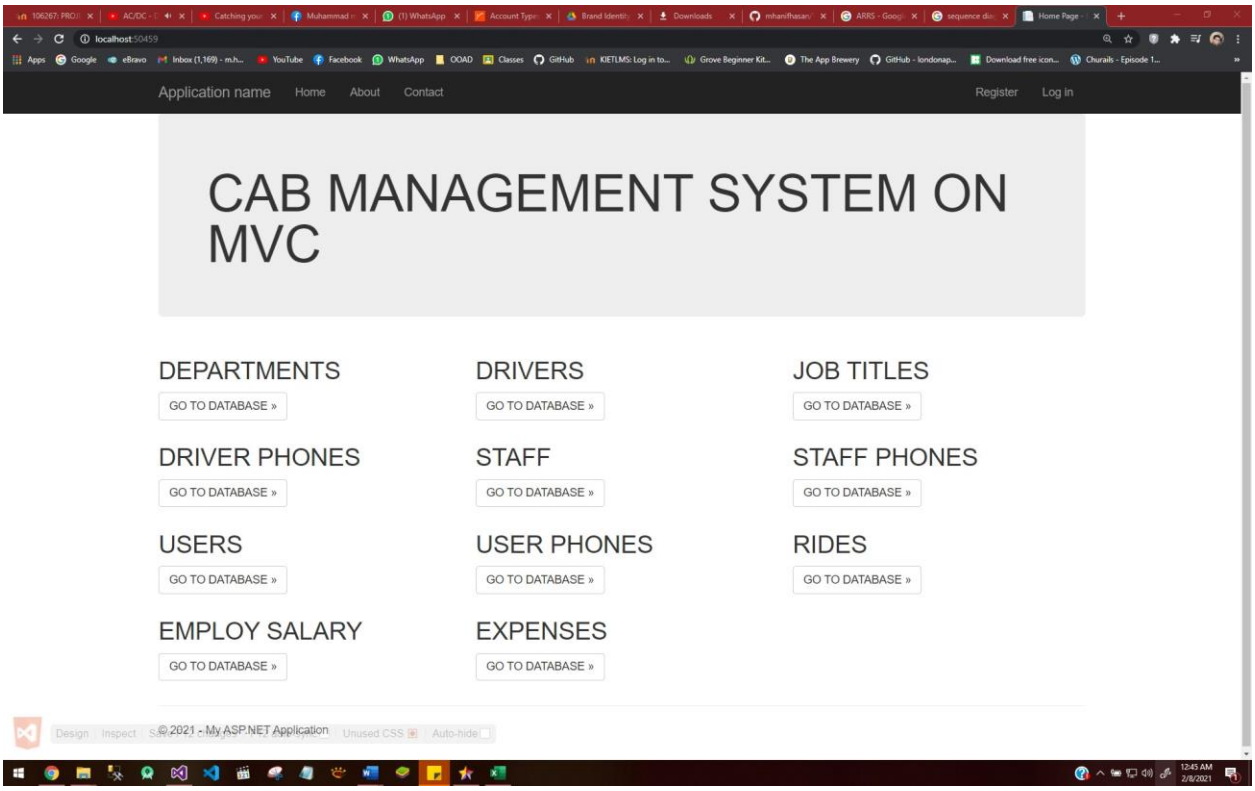
Signs in the appropriate admins for the different modules.



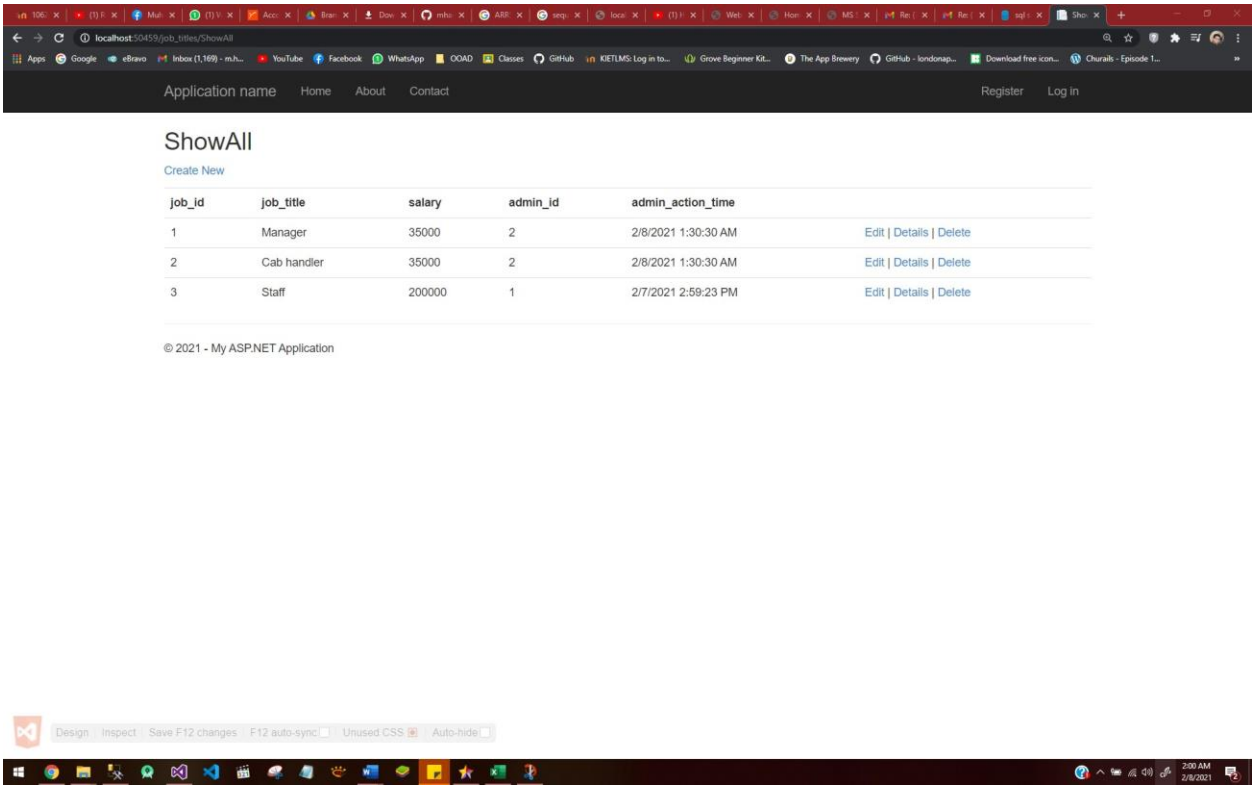
If you don’t have any Admin Account you can register here using Register page:



Once you login via a modile admin account you end up in the module show All page it changes according to the type of module admin :



The appropriate admin can perform CRUD functions on each module:




### 3.2.2 Hardware Interfaces

The ARRS includes two major hardware components: cellular phones and regular PC's. The cell phones require WAP (wireless application protocol) network protocol, which is already programmed in the latest phones.

The second component involves the regular PC's, which communicate with the server. The server then communicates with the database. The protocol involved between the PC's and the server is the HTTP protocol, which allows communication between the PC's and the Server. The remote PC's, such as someone accessing the ARRS from home using the Internet, are able access the information through the CGI. The requests come in through the HTTP protocol, and using an ODBC the database results are returned and processed using Perl to give an HTML web page. The format of the output is displayed as web pages.

### 3.2.3 Software Interfaces

An Sql Server DBMS will be used to manage the database and any changes made to it. Furthermore, the DBMS will make regular backups of the database and generate reports regularly so that they can be accessed by the CMS. the server will run on Somee.com. Furthermore, the HTML pages must be implemented such that they can be displayed on two common browsers: Netscape and Google Chrome.

Information about the products used for the CMS:

(1) Name: Sql Server
(2) Mnemonic: Sql Server
(3) Version Number: ?
(4) Source: Sql Server

(1) Name: Somee.com
(2) Mnemonic: Somee.com
(3) Version Number: ?
(4) Source: Online

(1) Name: Google Chrome
(2) Mnemonic: Chrome
(3) Version Number: ?
(4) Source: Google

### 3.3 Performance Requirements

The following sections list the performance requirements for the system.

#### 3.3.1 User Requirements

| **User Requirements** | **Description of Requirement For Design Environment** |
|---|---|
| Location(s) and Number(s) of Users | Guangzhou, Nanjing, Shanghai |
| Expected Growth in Number of Users | |
| After 1 Year | 50% |
| After 2 Years | TBD |
| After 3 Years | TBD |
| User Expectation | |
| Interactivity | User expect that it provides a very easy to use graphical user interface |
| Reliability | For some applications, reliability must be 100% during the application session |
| Adaptability | Network must adapt to user additions, deletions and changes |
| Security | Encryption software would be used for Credit Card transactions |
| Cost / Funding | Less than $250K |

#### 3.3.2 Application Requirements

Since no specified service is indicated, then we have listed the applications as best – efforts. This may change as we learn more about the application.

The communication package is determined to be bursty in nature, with small data sizes and frequent transmissions. We can consider this application to be interactive-burst, while the database transaction-processing application is described by the CRM as transferring large amounts of data (initial estimates are 1 MB/transaction), we have listed this application as interactive-bulk.

| Categorizing Applications | Best-Efforts | Application Locations |
|---|---|---|
| Communication | 100 Kb/s | Guangzhou and Nanjing |
| Database Access | 400 Kb/s | All Locations |
| Database Transaction processing | 1.5 Mb/s | All Locations |

### 3.3.3 Host Requirements

| | Type of Host or Equipment | Numbers and Locations |
|---|---|---|
| Host A | PC | Guangzhou (10), Nanjing(7), Shanghai(10) |
| Host B | Database Server | Shanghai |
| Host C | Application Server | Nanjing |

### 3.4.1 Standards Compliance

There are no design constraints that can be imposed by other standards limitations.

### 3.4.2 Software Limitations

- must be able to run Internet Explorer or Netscape Communicator web browsers to access the system.
- must have cell-phone web based capability to access the system from a mobile phone.

### 3.4.3 Hardware Limitations

- Input/Output: One or two-button mouse, keyboard, cell-phone, or touch screen required.
- Network card required at thin-client terminals to make communication with server possible.

## 3.5 Quality Characteristics

There are a number of quality characteristics that apply to the ARRS software system.

### 3.5.1 Portability

The ARRS system will be developed using HTML and Java so that it can be accessed from any type of system using just a regular web browser. It will also be

available to users that have web access on their cellular phones. The system will be tested on all types of hardware before being released to ensure that is it compliant with this requirement.

### 3.5.2 Reliability

The system should be capable of processing a given number of reservations within a give time frame with no errors and the system should be available and operational all the time. During the development of the prototype for the 3 cities, the system will be tested in its actual environment to ensure that it can handle the load of reservations that occur during a regular workday.

### 3.5.3 Usability

The ARRS system will be developed so that it is an easy to use system that requires the least amount of user input possible. Every input will be validated. The user should only have general computer use knowledge. Error messages will be displayed if the user enters an invalid value or tries to access a function without the required permissions. An easy and well-structured user manual will be provided to the CRM and the system will include descriptive help for all operations allowed.

### 3.5.4 Correctness

The ARRS system will be considered correct when the CRM approves the prototype presented and agrees that all the functions they require are implemented as stated in the Software Requirements Specification.

### 3.5.5 Flexibility

The ARRS system should be developed in such a way that it is easily customizable. If new functions are required by CRM, there will be little effort required to update the system to support new cities or new transactions.

### 3.5.6 Security

The ARRS system should not compromise the customer information at any time. The user information will never be sold to other parties and will be kept secure at all times. Users will be authenticated to ensure that no unauthorized users gain access to private information.

### 3.5.7 Maintainability

The ARRS source code will be kept well structure and documented so that it is easier to maintain and extend the system. All changes to the system shall be documented.

### 3.6 Other Requirements

Certain requirements may, due to the nature of the software, the user organization, etc., be placed in separate categories such as those below.

#### 3.6.1 Data Base

The Automate Railway Reservation System will have two main databases. One is the Reservation Database, and another is the Passenger Account Database. These database will be created with Oracle8i (Client/Server) version 8.1.6.0.0 Release 2. The following are the requirements for these databases that are to be developed as part of the product. They include:

**Reservation Database**

| | |
|---|---|
| Types of information | Schedule information for the trains, including date, time, departure city, destination city, ticket cost and ticket availability for a particular train |
| Frequency of use | Depends on the passenger demand, which may reach 25,000 per day during peak periods |
| Accessing capabilities | The database should allow access to at least 1,000 people at once; the users will have a general access to the information about the train schedule, and a secure access to the reports (available only to CRM officials) using a username and a password |
| Data element and file descriptions | To be determined |
| Relationship of data elements, records and files | To be determined |
| Static and dynamic organization | To be determined |
| Retention requirements for data | Train schedule information will be available as long as the train for a particular route is in use and at least one year after the train has been cancelled. The reports information will be available at least for 5 years |

**Passenger Account Database**

| | |
|---|---|
| Types of information | Passenger account information including their name, address, phone numbers, last reservations, balance owed, credit card number (if they paid by a credit card) |
| Frequency of use | Depends on the passenger demand, which may reach |

| | |
|---|---|
| | 25,000 per day during peak periods |
| Accessing capabilities | The database should allow access to at least 500 people at once; the users will have a secure access to the database using a username and a password |
| Data element and file descriptions | To be determined |
| Relationship of data elements, records and files | To be determined |
| Static and dynamic organization | To be determined |
| Retention requirements for data | Passenger account will be available for as long as a passenger is using the account, and at least for 6 month since the passenger logged on last time. |

### 3.6.2 Operations

The normal operations required by the user can be viewed as the following:

User-initiated Operations:

These operations include the login operation, which is initiated by the users. Also, the process of becoming a new user is in this category. Building, changing, and viewing itineraries, as well as paying for the itinerary are all initiated by the users. The user initiates the report generation activity, as well as changing train schedules.

Interactive Operations and Unattended Operations:

The users initiate all the operations mentioned above, and almost all of them are somehow interactive. Displaying the train schedule is non-interactive. The report display is a non-interactive operation, although selecting the desired reports will require user input.

Data Processing Support Functions:

The user account data is used to create new accounts, as well as to validate user id's during login functions. For building itineraries, user input, user account data, and train schedule data are used, and processed. User data along with final results of user interaction (whether the user purchased a trip, number of tickets bought, etc.) are collected, and used for report generation purposes. Administrative users' inputs are collected in order to modify and present schedules.

Backup and Recovery Operations:

Both databases used (passenger account database and reservations database) are production databases. The main operation used for the backup and recovery is Oracle's built-in cold backup, which is also known as the "archive mode". Depending on the customer's needs and budget, additional redundancy can be added using systems like RAID 5 and tape backup.

### 3.6.3 Site Adaptation Requirements

There are no site adaptation requirements for this project.

## 4. Supporting Information.

There is no supporting information required for this project.