**Online Ride Sharing**

**System Requirements Specification**

**Team Lead: Abdul Rehman Aamir**

**Members:**

Abdul Rehman Aamir

Musa Khan

Syed Qamar Abbas Hussnain Ali

Alina Zahid

Asad Javed

Muzammil Abbas

**University Name:** Capital University of Science and Technology.

**Department:** Computer Science

**REVISION HISTORY**

|  |  |  |
| --- | --- | --- |
| **Date** | **Description** | **Author** |
| 4/5/2018 | Online Ride Sharing | Abdul Rehman Aamir |

Contents

[1 Introduction 11](#_Toc516693500)

[**1.1** **Project Description:** 11](#_Toc516693501)

[**1.2** **Scope:** 12](#_Toc516693502)

[**1.3** **Modules:** 13](#_Toc516693503)

[**1.5 Definition, Acronyms and Abbreviation:** 13](#_Toc516693504)

[**1.5 Conclusion** 13](#_Toc516693505)

[2. Project Plane 13](#_Toc516693506)

[**2.1** **Overview** 13](#_Toc516693507)

[**2.2** **Organization** 15](#_Toc516693508)

[**2.2.1** **Project Organization** 15](#_Toc516693509)

[**2.3.2 Project-internal Functions** 15](#_Toc516693510)

[**2.3.3 Project Team** 15](#_Toc516693511)

[**2.4 Risk Management** 16](#_Toc516693512)

[**2.5 Description of Tasks** 16](#_Toc516693513)

[**2.5.1 Project Meeting:** 16](#_Toc516693514)

[**2.5.2 Project Start-up task:** 17](#_Toc516693515)

[**2.6** **Delivery Plan** 19](#_Toc516693516)

[**2.6.1 Deliverables and Receivers** 19](#_Toc516693517)

[3 Feasibility Study 19](#_Toc516693518)

[**3.1 Project history** 19](#_Toc516693519)

[**3.2 Project description** 20](#_Toc516693520)

[**3.3 Project Feature** 20](#_Toc516693521)

[**3.4 Conclusion** 26](#_Toc516693522)

[4 Functional and Non-Functional Requirements 26](#_Toc516693523)

[**4.1 Modules** 26](#_Toc516693524)

[**4.2 Definition, Acronyms and Abbreviation:** 27](#_Toc516693525)

[**4.3 Organizational Analysis** 27](#_Toc516693526)

[**4.4 Functional Requirements** 28](#_Toc516693527)

[**4.4.1 Student:** 28](#_Toc516693528)

[**4.4.2 Driver:** 28](#_Toc516693529)

[**4.4.3 Administrator:** 29](#_Toc516693530)

[5 System Model 31](#_Toc516693531)

[**5.1** **System Design Model** 31](#_Toc516693532)

[**5.2 User Requirements:** 31](#_Toc516693533)

[**5.3 Software Life Cycle Model:** 31](#_Toc516693534)

[6 Use case/Test Case 33](#_Toc516693535)

[**6.1** **Job Seeker** 33](#_Toc516693536)

[**6.1.1 New User** 33](#_Toc516693537)

[**6.1.2 Student update** 34](#_Toc516693538)

[**6.1.3 Delete** 35](#_Toc516693539)

[**6.1.4 Drop Location** 36](#_Toc516693540)

[**6.1.5 Pick Location** 36](#_Toc516693541)

[7 Prototyping 37](#_Toc516693542)

[7.**1 Login Page** 37](#_Toc516693543)

[**7.1.1 Admin Login page** 37](#_Toc516693544)

[**7.1.2 Student Login Page** 38](#_Toc516693545)

[**7.1.3 Driver Login page** 38](#_Toc516693546)

[**Chat Box** 39](#_Toc516693547)

[8 Prepare DFD 40](#_Toc516693548)

[**8.1** **Listing all Use Cases** 40](#_Toc516693549)

[8.1.1 Selecting Use Case 40](#_Toc516693550)

[Admin Login 40](#_Toc516693551)

[Login Admin. 40](#_Toc516693552)

[Main Steps: 40](#_Toc516693553)

[Defining Conditions According to Situation 40](#_Toc516693554)

[Identifying Entities 41](#_Toc516693555)

[Identifying Data Flow 41](#_Toc516693556)

[Identifying data store 41](#_Toc516693557)

[Identify Data 41](#_Toc516693558)

[Context Level Diagram 41](#_Toc516693559)

[3.6 Level 0 Diagram: 42](#_Toc516693560)

[Level 1 Diagram 42](#_Toc516693561)

[Selecting Use Case 43](#_Toc516693562)

[Assigning Name to Use Case 43](#_Toc516693563)

[Main Steps: 43](#_Toc516693564)

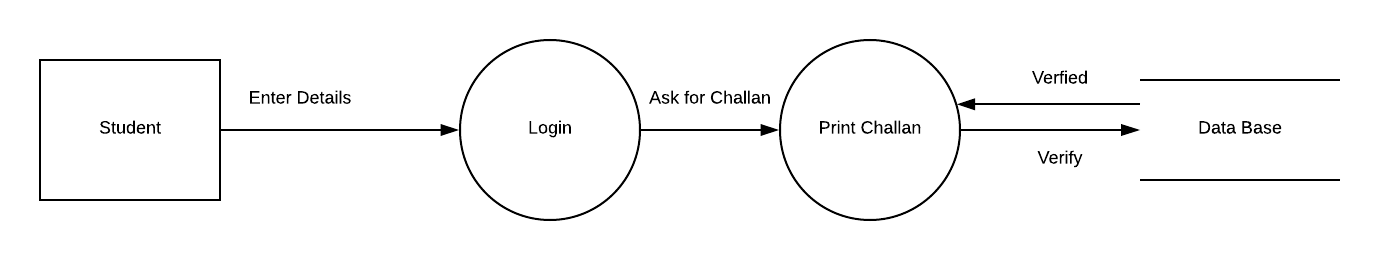
[1.1 Defining Conditions According to Situation 43](#_Toc516693565)

[Identifying Data Flow 44](#_Toc516693566)

[Identifying data store 44](#_Toc516693567)

[Identify Data 44](#_Toc516693568)

[Context Level Diagram 44](#_Toc516693569)

[ 44](#_Toc516693570)

[Level 0 Diagram 45](#_Toc516693571)

[Level 1 Diagram 45](#_Toc516693572)

[Selecting Use Case 45](#_Toc516693573)

[Student Can Generate Pass 45](#_Toc516693574)

[Assigning Name to Use Case 45](#_Toc516693575)

[Generating Pass 45](#_Toc516693576)

[Main Steps: 45](#_Toc516693577)

[Defining Conditions According to Situation 46](#_Toc516693578)

[Identifying Entities 46](#_Toc516693579)

[ Student 46](#_Toc516693580)

[Identifying Data Flow 46](#_Toc516693581)

[Login account to data base 46](#_Toc516693582)

[Identifying data store 46](#_Toc516693583)

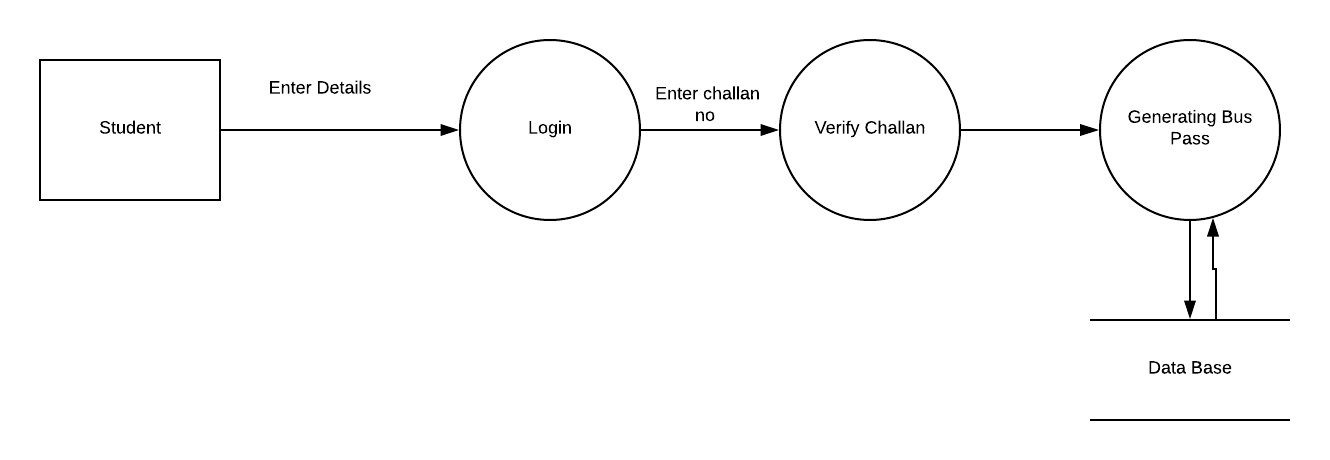
[Database 46](#_Toc516693584)

[Identify Data 46](#_Toc516693585)

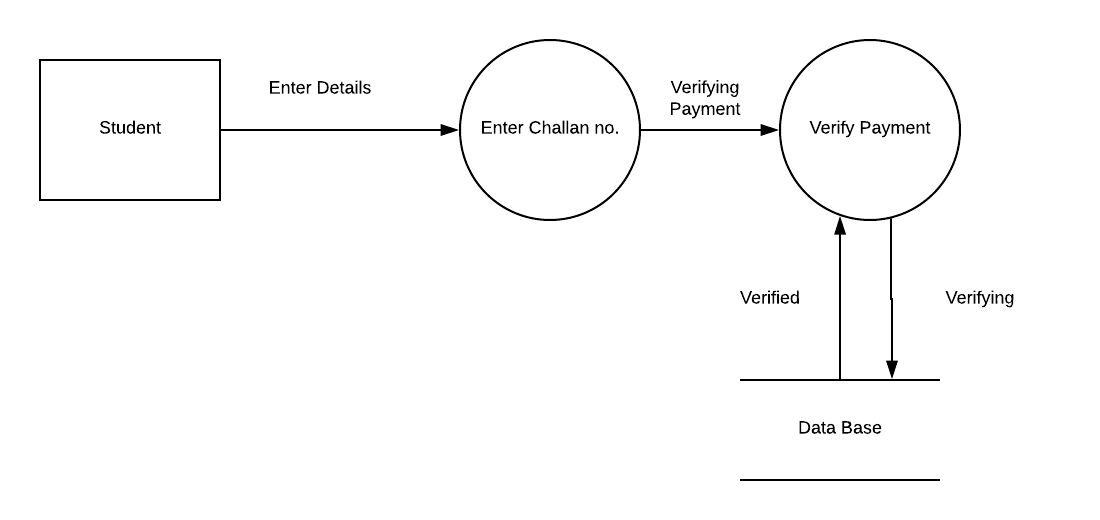
[Student Information 46](#_Toc516693586)

[Level 0 Diagram 46](#_Toc516693587)

[Level 0 Diagram 46](#_Toc516693588)

[ 47](#_Toc516693589)

[Level 1 Diagram 47](#_Toc516693590)

[ 47](#_Toc516693591)

[Level 1 Diagram 48](#_Toc516693592)

[Selecting Use Case 49](#_Toc516693593)

[Student Can login 49](#_Toc516693594)

[Assigning Name to Use Case 49](#_Toc516693595)

[Login student 49](#_Toc516693596)

[Main Steps: 49](#_Toc516693597)

[Defining Conditions According to Situation 49](#_Toc516693598)

[Identifying Entities 49](#_Toc516693599)

[ Student 49](#_Toc516693600)

[Identifying Data Flow 49](#_Toc516693601)

[Login account to data base 49](#_Toc516693602)

[Identifying data store 49](#_Toc516693603)

[Database 49](#_Toc516693604)

[Identify Data 49](#_Toc516693605)

[Admin Information 49](#_Toc516693606)

[Context Level Diagram: 50](#_Toc516693607)

[Level 0 Diagram: 50](#_Toc516693608)

[Level 1 Diagram 50](#_Toc516693609)

[Selecting Use Case 51](#_Toc516693610)

[Student Can Cancel Rides 51](#_Toc516693611)

[Assigning Name to Use Case 51](#_Toc516693612)

[Cancel Rides 51](#_Toc516693613)

[Main Steps: 51](#_Toc516693614)

[Defining Conditions According to Situation 51](#_Toc516693615)

[Identifying Entities 51](#_Toc516693616)

[ Student 51](#_Toc516693617)

[Identifying Data Flow 51](#_Toc516693618)

[Login account to data base 51](#_Toc516693619)

[Identifying data store 51](#_Toc516693620)

[Database 51](#_Toc516693621)

[Identify Data 51](#_Toc516693622)

[Admin Information 51](#_Toc516693623)

[Context Level Diagram: 52](#_Toc516693624)

[Level 0 Diagram 52](#_Toc516693625)

[Level 1 Diagram 53](#_Toc516693626)

[Selecting Use Case 54](#_Toc516693627)

[Driver Can Signup 54](#_Toc516693628)

[Assigning Name to Use Case 54](#_Toc516693629)

[Signup Driver 54](#_Toc516693630)

[Main Steps: 54](#_Toc516693631)

[1.2 Defining Conditions According to Situation 54](#_Toc516693632)

[Identifying Entities 54](#_Toc516693633)

[ Driver 54](#_Toc516693634)

[Identifying Data Flow 54](#_Toc516693635)

[Login account to data base 54](#_Toc516693636)

[Identifying data store 54](#_Toc516693637)

[Database 54](#_Toc516693638)

[Identify Data 54](#_Toc516693639)

[Admin Information 54](#_Toc516693640)

[Context Level Diagram 55](#_Toc516693641)

[Level 0 Diagram 55](#_Toc516693642)

[Level 1 Diagram 56](#_Toc516693643)

[9 Decision Tree 56](#_Toc516693644)

[**9.1** **Use Cases:** 56](#_Toc516693645)

[1. 9.13 Decision Tree: 57](#_Toc516693646)

[10 Decision Table 65](#_Toc516693647)

[**10.1** **Listing all Use Cases:** 65](#_Toc516693648)

[10. Use Case Description 65](#_Toc516693649)

[Passenger Logic 65](#_Toc516693650)

[Interpretation: 65](#_Toc516693651)

[Passenger sign up 65](#_Toc516693652)

[Interpretation: 65](#_Toc516693653)

[Driver Sign up 66](#_Toc516693654)

[Interpretation: 66](#_Toc516693655)

[Admin Block passenger 66](#_Toc516693656)

[Interpretation: 66](#_Toc516693657)

[Cancel ride 66](#_Toc516693658)

[Passenger Pick location 66](#_Toc516693659)

[Fare Estimate 66](#_Toc516693660)

[Drop location 67](#_Toc516693661)

[Bus Tracking 67](#_Toc516693662)

[Admin Delete Driver 67](#_Toc516693663)

[Use Case Decision Table 67](#_Toc516693664)

[Passenger Login 67](#_Toc516693665)

[Passenger Sign up 68](#_Toc516693666)

[Driver Sign up 69](#_Toc516693667)

[Admin Block Passenger 69](#_Toc516693668)

[Passenger cancel a ride 70](#_Toc516693669)

[Passenger pick location 71](#_Toc516693670)

[Fare estimate 71](#_Toc516693671)

[Drop location 72](#_Toc516693672)

[Bus tracking 73](#_Toc516693673)

[Admin Delete Driver 74](#_Toc516693674)

[**Process Descriptor** 75](#_Toc516693675)

[11 Use Cases 75](#_Toc516693676)

[Use case Description for Driver Signup: 75](#_Toc516693677)

[Use case for Student Login to system: 76](#_Toc516693678)

[Use case for Admin Login to system: 76](#_Toc516693679)

[Use case for Passenger Login to system: 76](#_Toc516693680)

[2.5 Use case for Drop location: 77](#_Toc516693681)

[Use case for pick location: 77](#_Toc516693682)

[Use case for Cancel Ride: 78](#_Toc516693683)

[Use case fair estimate: 78](#_Toc516693684)

[Use Case Diagram: 78](#_Toc516693685)

[Use Case Diagram for Admin Signup: 79](#_Toc516693686)

[Use Case Diagram for Student Login : 80](#_Toc516693687)

[Use Case Diagram for Passenger Login : 81](#_Toc516693688)

[Use Case Diagram for Driver Login : 82](#_Toc516693689)

[Use Case Diagram for Pick Location: 83](#_Toc516693690)

[Use Case Diagram for Drop Location: 84](#_Toc516693691)

[Use Case Diagram for Fair Estimate: 85](#_Toc516693692)

[Use Case Diagram for Cancel Ride: 86](#_Toc516693693)

[12 Prepare ER Diagram 87](#_Toc516693694)

[13 Architecture Design 87](#_Toc516693695)

[**13.1** **About:** 87](#_Toc516693696)

[**13.2** **Working:** 87](#_Toc516693697)

[**13.3** **Architectural Design Representation:** 87](#_Toc516693698)

[**13.4** **Architectural Styles:** 88](#_Toc516693699)

[**14.6 Object-Oriented Architecture:** 88](#_Toc516693700)

[**13.5** **Data-Centered Architecture:** 88](#_Toc516693701)

[14 Deployment Design 88](#_Toc516693702)

[**14.1** **Data-Centered Architecture:** 88](#_Toc516693703)

[**14.2** **Visual Representation:** 89](#_Toc516693704)

[15 Detailed Design 89](#_Toc516693705)

[**15.1** **Main Classes:** 89](#_Toc516693706)

[Class 1 “Admin” 90](#_Toc516693707)

[Data Members 90](#_Toc516693708)

[Methods 90](#_Toc516693709)

[Class 2 “Student” 90](#_Toc516693710)

[Data Members 90](#_Toc516693711)

[Methods 90](#_Toc516693712)

[Class 3 “Driver” 91](#_Toc516693713)

[Methods 91](#_Toc516693714)

[Class 4 “Registration” 91](#_Toc516693715)

[Data Members 91](#_Toc516693716)

[Methods 91](#_Toc516693717)

[Class 5 “Book Ride” 92](#_Toc516693718)

[Data Members 92](#_Toc516693719)

[Methods 92](#_Toc516693720)

[Class 6 “Cancel Ride” 92](#_Toc516693721)

[Data Members 92](#_Toc516693722)

[Methods 92](#_Toc516693723)

[Relationships 93](#_Toc516693724)

[Class Diagram 94](#_Toc516693725)

[16 Sequence Diagram 97](#_Toc516693726)

[**16.1** **User Registration:** 97](#_Toc516693727)

[**16.2** **User Login:** 97](#_Toc516693728)

[**16.3** **Delete User:** 99](#_Toc516693729)

[17 State Diagram 99](#_Toc516693730)

[**17.1** **Admin:** 100](#_Toc516693731)

[**17.2** **Applicant:** 101](#_Toc516693732)

[Appendix A: Glossary 102](#_Toc516693733)

[Appendix B: Analysis Models 102](#_Toc516693734)

[Appendix C: Issues List 102](#_Toc516693735)

[Bibliography 102](#_Toc516693736)

# **1 Introduction**

## **Project Description:**

Our proposed project is about online ride-share-app . In this project we will discuss the university bus system.The service is intended to provide transportation for students, faculty, and staff on an online application. In order to use this application we have two different policies.

The user have to submit its bus fee in the university bank. After submitting the fee user will get a user name and password for its bus service application will the help of that username and password user will sign-up in application.

Student is allowed to take several actions for example if student is not willing to take bus or he is on another location he can change the location or he can cancel the ride.

Admin have the permission to view everything going on and can update, delete or block the student and bus driver. He have the record of students, bus driver and buses and have the access to monitor busses on GPS.

Bus driver is a diver who drive bus he have the permission to sign up so in order to create account and can make his work easier and faster. The application will let him know that which student he have to pick first along with student location and student information. The application will also notify him that which student is not willing to take the bus, that particular will be dequeued for the current day.

GPS monitoring is the special feature of this application in which every actor depends. Admin will monitor busses through GPS, student will monitor the bus as well as the driver information he have to take through gps. Bus driver will figure out which student he have to pick first through gps.

Condition : only registered users are able to use this application

## **Scope:**

The scope for the system can be as follows:

* Maintain students record.
* Provide Online services.

## **Modules:**

The main users in the project can be categorized into three modules.

* Student.
* Driver.
* Administrator.

## **1.5 Definition, Acronyms and Abbreviation:**

**Student:** Student mean registered person who is able to register for bus services.

**Driver:** Driver means the person who has the authority to drive the bus and pick the students from their stations.

**Job Portal Administrator:** The person, who is responsible for the management, maintenance and administration of this portal.

## **1.5 Conclusion**

After Studying the whole feasibility report it is concluded that this application is useful and economical and will provide a good usability to the user. So this application is recommended by the organization.

# **Project Plane**

## **Overview**

In this project we will discuss the university bus system.The service is intended to provide transportation for students, faculty, and staff on an online application. In order to use this application we have two different policies.

The user have to submit its bus fee in the university bank. After submitting the fee user will get a user name and password for its bus service application will the help of that username and password user will sign-up in application.

Student is allowed to take several actions for example if student is not willing to take bus or he is on another location he can change the location or he can cancel the ride.

**Project Goals**

The project goal of this project is to satisfy the needs of University i.e. managing bus seats and avoid wasting time is the basic purpose of this project. Students don’t have to wait in a long queue and admins don’t have to deal ample of students at a time. All will be done online like students registration, generation of challan forms, passes all would be done online.

| **Project Goal** | **Priority** | **Comment/Description/Reference** |
| --- | --- | --- |
| **Functional Goals:** | 1 |  |
| Functionality 1 |  | Concurrency |
| Functionality 2 |  | Correctness |
| Functionality 3 |  | Deliverance |
| **Business Goals:** | 2 |  |
| Time to market |  | Will this project be that applicable when it is delivered to clients? |
| efficiency |  | Will this work efficient upon every operation system based system? |
| Cost |  | Will it exceed the budget limit or not? |
| Quality |  | With respect to corresponding software will this software project be good for business. |
| **Technological Goals:** | 3 |  |
| ­domain |  | Do we actually have the domain to save our data? |
| Technical staff |  | Our team has enough skill to rum the project smoothly. |
| **Quality Goals:** | 4 |  |
| Will it meet the standard |  | Has the software has all details that was required by customer. |
| functionality |  | The transection of each step and time dilation is standard or it slow. |
| **Constraints:** | 5 |  |
| environmental |  | User-friendly? |
| Application specific standards |  | Has enough operational function to |
| national standards |  | Won’t have any illegal actions or term and condition against government. |
| Cultural Relationship |  | Won’t have any text to hurt psychologically or mentally |
| **Other Goals:** | 6 |  |
| Usability |  | Will it be easy to use by any front end users |
| portability |  | Will it work on any systems like IOS, ANDROID, MAC and etc. |

## **Organization**

### **Project Organization**

The project has organized by the following team and their roles. Apart from their actual role some other task was given by Project Manager.

**Project Manager**

| **Role** | **Organization: Name** |
| --- | --- |
| Project Manager | Abdul Rehman |
| Technical Project Mgr. | Imran Mushtaq |

### **2.3.2 Project-internal Functions**

| **Function** | **Organization: Name** | **Comment** |
| --- | --- | --- |
| Quality Assurance | Abdul Rehman | Ensures requirements |
| System Test Lead | Musa Khan | Makes Test Cases + Questionnaire |
| Validation Lead | Alina Zahid | Verification. |
| Configuration Mgmt | Abdul Rehman | Manages and Configures tasks |
| Change Mgmt | Syed Qamar Abbas | Checks at every step to ensure change |
| Backing up data | Asad Javed | Backup all the done work. |

### **2.3.3 Project Team**

| Abdul Rehman | 24/7 | Project Planner & Project Manager |
| --- | --- | --- |
| Musa Khan | 9/5 (2 days off at end of week) | Document a Feasibility Report. |
| Syed Qamar Abbas | 9/5 (2 days off from week) | Prepares Functional Requirements document |
| Husnain Ali | 9/5 (2 days off from week) | Prepares Non-Functional Requirements document. |
| Alina Zahid | 9/3 (4 days off from week) | Proposing a System Model |
| Asad javed | 9/4 (3 days off from week) | Prepare Use Case / Test Case.(The UI) |
| Muzammil Abbas | 9/5 (2 days off from week) | Use prototype to validate requirements. |

## **2.4 Risk Management**

The risk management includes:

1. Has the delivery date is changed, before the delivery date.
2. Out of resources.
3. One of major member of technical staff is ill.
4. Machine or setup or system is broken or crashed.
5. Etc.

## **2.5 Description of Tasks**

### **2.5.1 Project Meeting:**

Projectmeeting is held in order to test the skills of the team as well as identifying the potential of team. Tasks are assigned to the individuals according to their skills and willingness to recognise it a meeting necessary. Project meeting is also very help to figure out the motivation of the team and check out the taken task can be completed by the time taken or not in order to avoid issues of time, budget and allocation of tasks project meeting is held. Further the tasks has been allocated to the team members along with the deadline in which they have to submit the task by all means.

### **2.5.2 Project Start-up task:**

Once the project has been given the official go ahead, it is imperative to establish the current project team and the project environment for this project phase. In this stage of the project, key individuals are assigned project roles. Initially, the number of people brought the team will be small. Groups represented on the team will probably underway. These initial leaders will work to ensure that staff, equipment’s and facility resources are in place when needed.

**2.5.3 Phase 1:**

It exclusively includes the project start up idea under which all the project will run. It includes the status meeting in which will be held time to time to ensure the progress of the team and their work.

**2.5.4 Phase 2:**

After the phase one that the project will be running the phase 2 comes in action in which the project scope in judge that rather the project is portable, profitable and reliable. Then 4 days are given to sponsorship that under which party the project will run that which one of the team will fund the project and help the project to be running. In this phase sources are the important part because this phase ensures that do we have the entire domain needed to run this project like preliminary resources that some common object and need are available to keep going this project. This phase last part includes cost and project plan update. Cost in terms that the project is feasible in terms of budget or not? Will the technical staffs that was hired will be paid off plus the expenses will meet the budget dead line or not?

**2.5.5 Phase 3:**

At first it is the delivery of first two phases result then combination of these two leads the third phase that is software requirements. The biggest question is understanding the requirements and requirement gathering that includes 20 days, a big part of clock is given to this particular part to really ensure that everything work good and real. The review of budget and feedback for the gathering of requirements also held in this phases and makes it a longest phase of the whole project.

**2.5.6 Phase 4:**

The phase 4 include the designing i.e. that by understanding the requirements the UI has been made like what the client wants. In this phase the 3 days are given to review the functional specification like which kinds of function this particular software will do. Understanding this specification is very necessary. Then comes prototyping 17 days: a very large time period is given to prototyping because as we know, we interact the client time to time that’s why a feedback what the staff is make is necessary. Then again 5 days are given to requirements specification to ensure that the work goes right way and then comes the approval days i.e. 4 days these 4days are very curial because if the client don’t like the UI then whole process will be in again running position that’s why requirement specification period comes two time between this phase. After approval the design is complete and handed over to the next phase this particular day gives the next phase staff to understand that what was needed and what was given has true scene then they will work their way next.

**2.5.7 Phase 5:**

The phase 5 includes the real development including the coding phase. This phase is given 12 days because at very first phase of this phase is requirements gathering like what was needed and what was understand. So the time duration needed to judge what was needed is a technical phase. In this phase 10 days are given to coding the testing 6 days and completion 3 days. The first 10 days are very crucial include the real work then making sure that what was need and given and understand stand here. The next 6 days for testing has a technical staff that test the code to judge that rather it has meet the level of standards or not. Meeting the standard is very important in this phase because by this period the project going to be gone to last completing phase.

**2.5.8 Phase 6:**

It includes the extra part of the project which is that if the client wants any change at the rare end.

## **Delivery Plan**

### **2.6.1 Deliverables and Receivers**

| **Ident.** | **Deliverable** | **Planned Date** | **Receiver** |
| --- | --- | --- | --- |
| D1 | Phase 1 | Wed 6/13/18 | Wed 6/13/18 |
| D2 | Phase 2 | Fri 3/23/18 | Fri 3/23/18 |
| D3 | Phase 3 | Wed 5/2/18 | Wed 5/2/18 |
| D4 | Phase 4 | Wed 6/6/18 | Wed 6/6/18 |
| D5 | Phase 5 | Mon 6/18/18 | Mon 6/18/18 |

# **3 Feasibility Study**

## **3.1 Project history**

The Project was proposed on 17th of March 2018 by Sir Imran Rao. My task is to prepare a feasibility report on this project to know whether this project actually needs to build or not.

|  |  |  |
| --- | --- | --- |
| **S.no** | **Names** | **Project Tasks** |
| 1. | Abdul Rehman Aamir | Team Lead: Prepare Project Plan & Presentation on Requirements |
| 2. | Musa Khan | Prepare a Feasibility Report and Document it |
| 3. | Alina Zahid | Prepare Functional Requirement |
| 4. | Muhammad Mudassir | Propose System Model / Justify it and Document it |
| 5. | Asad Javed | Prepare Use Case / Test Case and Document it |
| 6. | Muzammil Abbas | Use Prototypes to Validate Requirements |
| 7. | Hussnain Ali | Prepare non-Functional Requirement |

## **3.2 Project description**

Our proposed project is about online ride-share-app. In this project we will discuss the university bus system. The service is intended to provide transportation for students, faculty, and staff on an online application. In order to use this application, we have two different policies.

The user has to submit its bus fee in the university bank. After submitting the fee user will get a user name and password for its bus service application will the help of that username and password user will sign-up in application.

Student is allowed to take several actions for example if student is not willing to take bus or he is on another location he can change the location or he can cancel the ride.

Admin have the permission to view everything going on and can update, delete or block the student and bus driver. He has the record of students, bus driver and buses and have the access to monitor busses on GPS.

Bus driver is a diver who drive bus he has the permission to sign up so in order to create account and can make his work easier and faster. The application will let him know that which student he has to pick first along with student location and student information. The application will also notify him that which student is not willing to take the bus, that particular will be dequeued for the current day.

GPS monitoring is the special feature of this application in which every actor depends. Admin will monitor busses through GPS, student will monitor the bus as well as the driver information he has to take through gaps. Bus driver will figure out which student he has to pick first through gaps.

Condition: Only registered users are able to use this application

## **3.3 Project Feature**

|  |  |
| --- | --- |
| Element | Feature |
| Product feature | Admin Sign In |
| Project name | Ride sharing |
| Description | Allows user to create account |
| Key feature | Allows to create ID |
| Key feature supporting product | Permits to have access on app |
| Cost | 25$ |
| Timing | 6 days |
| Skills Required | Database control  Able to link database and application |

Pros:

* Admin will have the access of whole product
* He is allowed to delete, update and insert data
* He have the access to provide the alternate of problems

Cons

* The form is not highly secure robotic sign-in can take place
* Un-authorize account sign in.

Solution

* In order to resolve this issue/problem a Captcha is required which will stop the robotic signing
* In order to resolve the un-authorize account problem the admin have to be provided by a special key from the organization.

|  |  |
| --- | --- |
| **Element** | **Feature** |
| Product feature | Job seeker Sign In |
| Project name | University Job Seek portal |
| Description | Allows user to create account |
| Timing | 24days |
| Skills Required | Database control  Able to link database and application |

Pros:

* Job seeker will have the access on his/her data and can see driver info
* He can apply for any job
* He can view his profile

Cons

* The form is not highly secure robotic signing can take place.
* Un-authorize account sign in.

Solution

* In order to resolve this issue/problem a Captcha is required which will stop the robotic signings
* In order to resolve the un-authorize account problem the job seeker will be provided be special code by the admin.

|  |  |
| --- | --- |
| **Element** | **Feature** |
| Product feature | Student Sign In |
| Project name | Ride sharing |
| Description | Allows user to create account |
| Key feature | Allows to create ID |
| Key feature supporting product | Permits to have access on app |
| Cost | 30$ |
| Timing | 4 days |
| Skills Required | Database control  Able to link database and application |

Pros:

* Student will have the access on his/her data and can see driver info
* He can monitor driver and other busses location through Gps in his/her specific route
* He can cancel the ride if he wants

Cons

* The form is not highly secure robotic signin can take place.
* Un-authorize account sign in.

Solution

* In order to resolve this issue/problem a Capcha is required which will stop the robotic signings
* In order to resolve the un-authorize account problem the student will be provided ba special code from the student affairs.

|  |  |
| --- | --- |
| **Element** | **Feature** |
| Product feature | Driver Sign In |
| Project name | Ride sharing |
| Description | Allows user to create account |
| Key feature | Allows to create ID |
| Key feature supporting product | Permits to have access on app |
| Cost | 18 $ |
| Timing | 3 days |
| Skills Required | Database control  Able to link database and application |

Pros:

* Driver will have the access on his and student information
* He is allowed to monitor student location through gps

Cons

* The form is not highly secure robotic sign-in can take place
* Un-authorize account sign in.

Solution

* In order to resolve this issue/problem a Capcha is required which will stop the robotic signings
* In order to resolve the un-authorize account problem the driver have to be provided by a special pin from

|  |  |
| --- | --- |
| **Element** | **Feature** |
| Product feature | Admin log In |
| Project name | Ride sharing |
| Description | Allows user to use account |
| Key feature | Allows to access features |
| Key feature supporting product | Permits to have access on app |
| Cost | 18 $ |
| Timing | 3 days |
| Skills Required | Database control  Able to link database and application |

Pros:

* Driver will have the access on his and student information
* He is allowed to monitor student location through gps

Cons

* The form is not highly secure robotic sign-in can take place
* Un-authorize account sign in.

Solution

* In order to resolve this issue/problem a Capcha is required which will stop the robotic signings
* In order to resolve the un-authorize account problem the driver have to be provided by a special pin from

|  |  |
| --- | --- |
| **Element** | **Feature** |
| Product feature | Student log In |
| Project name | Ride sharing |
| Description | Allows user to use account |
| Key feature | Allows to access features |
| Key feature supporting product | Permits to have access on app |
| Cost | 15 $ |
| Timing | 5 days |
| Skills Required | Database control  Able to link database and application |

Pros:

* Driver will have the access on his and student information
* He is allowed to monitor student location through gps

Cons:

* The form is not highly secure robotic sign-in can take place
* Un-authorize account sign in.

Solution:

* In order to resolve this issue/problem a Capcha is required which will stop the robotic signings
* In order to resolve the un-authorize account problem the driver have to be provided by a special pin from

## **3.4 Conclusion**

After Studying the whole feasibility report it is concluded that this application is useful and economical and will provide a good usability to the user. So this application is recommended by the organization.

# 4 **Functional and Non-Functional Requirements**

## 

## **4.1 Modules**

The main users in the project can be categorized into three modules.

* Student.
* Driver.
* Administrator.

## **4.2 Definition, Acronyms and Abbreviation:**

**Student:** Student mean registered person who is able to register for bus services.

**Driver:** Driver means the person who has the authority to drive the bus and pick the students from their stations.

**Job Portal Administrator:** The person, who is responsible for the management, maintenance and administration of this portal.

## **4.3 Organizational Analysis**

Organizational analysis represents value creation. It stresses as relationship building and maintaining with broad range of stakeholders that includes customers, employees, government, and supplies and so on.

There are four types of stakeholders in this job portal system.

1 Primary Stakeholder:

Primary stakeholders are those who are directly interacted with the system. According to our system the primary stakeholders are administrator, jobseeker and employer.

2 Secondary STAKEHOLDERS:

Secondary stakeholders are those who are not directly interacted with the system but receive output. According to our system secondary stakeholders are Management who are not directly interacted with the system.

3 Tertiary Stakeholder:

The stakeholders who are not involved in the system. These are not primary as well as secondary stakeholder but they are affected by the success and failure of the system. These are the investors and competitors, developers whose profit demands on the success and failure of the system.

4 Facilitating Stakeholder:

This is the main stakeholder because they are responsible for developing, designing, maintenance the system.

## **4.4 Functional Requirements**

The functional requirements for job portal are explained below:

### **4.4.1 Student:**

**1.Registration:**

If he wants to register to the site, click on the registration button and then he must fill the appeared form after this once he clicked on submit button he will received via message or email.

1. **Login:**

It allows only authorized people to access the application. When the user logs in to the system, the user has to enter login details in the specific username and password fields. The user clicks on the login button and if id and password are validated, then the user is given access to the application and user profile is displayed.

1. **Forgot Password:**

If the user forgets the password so he can retrieve password by clicking on forgot password button. Then the system will ask you for security question after the verification a message with correct password will be provided to the user through email.

**Additional features:**

1. If there is any kind of query related to bus registration or form filling so he can send private message in the chat box.
2. An email notification will be send to job seeker when his form submission will be received successfully.

### **4.4.2 Driver:**

1. **Registration:**

For Driver registration siteIf he wants to register to the site, click on the registration button and then he must fill the appeared form after this once he clicked on submit button he will received via message or email.

1. **Login:**
2. Driver Name
3. Driver Address
4. Assigned bus number
5. Contact
6. Special Key
7. Image
8. Captcha
9. **Forgot password option:**

If the user forgets the password so he can retrieve password by clicking on forgot password button. Then the system will ask you for security question after the verification a message with correct password will be provided to the user through email

**Additional features:**

1. Monitor things
2. Able to edit, update or remove data.
3. Have the access to take different decisions like allocating tracks

### **4.4.3 Administrator:**

1. **Manage bus routes and student registrations:**

Admin has the responsibility to monitor busses through and GPS and to keep an eye on student registrations.

1. **Send Notification Message to Driver:**

Whenever any user registered to the site, the admin will have to send them notification message confirming their subscription.

.

**Ability to Change Website’s Interface Look:**

The option to make modifications in the look and appearance of the website interface relies in the hand of administrator. This includes changing of menu in the menu bar, display of contents and also the color if the interface.

**4.5 Non-Functional Requirements:**

The non-functional requirements necessary for the project are have better security and control over the system which the specific users would use it. The performance of the system to how quick the system works and the ease of use of the system by the user.

Nonfunctional requirements are:

**1.1 Reliability:**

The system should be reliable. The software should not crash.

**1.2** **Availability:**

The system should be available 24/7. The availability of the system is that the user is able to access the system anywhere and at any time. If some maintenance is required, the system would have down time when the system is not in use by any one for a short amount of time as much as an hour.

**1.3 Security:**

The security is major issue; the application software should be secure. Security is important because the application software web based. Security will be provided through Access Control Mechanism. The application software will have secure password authentication and prevent illegal access to Members accounts.

**1.4 Maintainability:**

The Job Portal will be designed in such a way that it can be maintained in Future. The day or time must be select for the maintainability of the system like Saturday or Sunday 2AM to 3Am at this time we can maintain the system.

**1.5 Usability:**

User interface is not much of concern because only the basic information is required to use the Portal. Email alerts will be sent to users who will subscribe to it so they can remain up to date with the system.

**1.6 Extendibility:**

The system is designed in such a way that if new features want to add in the future so this can be done easily.

# **System Model**

## **System Design Model**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Models** | **Fully Known Requirements** | **Time Efficient** | **Cost Efficient** | **User Involvement** | **System Complexity** |
| **Waterfall** | ✔ | 🗙 | 🗙 | 🗙 | 🗙 |
| **V Model** | ✔ | ✔ | ✔ | 🗙 | 🗙 |
| **Incremental** | 🗙 | ✔ | ✔ | 🗙 | ✔ |
| **Iterative** | 🗙 | 🗙 | 🗙 | ✔ | ✔ |
| **Prototyping** | 🗙 | 🗙 | 🗙 | ✔ | ✔ |
| **Spiral** | 🗙 | 🗙 | 🗙 | 🗙 | ✔ |
| **RAD** | ✔ | ✔ | ✔ | ✔ | ✔ |

## **5.2 User Requirements:**

|  |  |
| --- | --- |
| **Fully Known Requirements** | Yes |
| **Time Efficient** | No |
| **Cost Efficient** | No |
| **User Involvement** | No |
| **System Complexity** | No |

Conclusion: Waterfall model fully meets requirement of our client so we are using it. User requirements are well known, time and cost efficiency is not a problem, and user interaction is not required and doesn’t contain any system complexity in it.

## **5.3 Software Life Cycle Model:**

**Product requirements**

These requirement documents will include detailed requirements, user scenarios and potential layouts for the functionality.

Timeframe: 2 weeks

**Analysis**

Engineering team takes these requirements and analyzes them, asking questions as needed. Product manager updates documents as questions are resolved.

Timeframe: 1 week

**Design**

Engineering team creates a design for functionality, including database design, mock-ups and workflows.

Timeframe: 3 weeks

**Implementation**

Engineering team develops functionality and prepares it for testing.

Timeframe: 1 week

**Software product testing**

Product team tests entire functionality.

Timeframe: 2 weeks

**Release**

The product functionality is released.

TOTAL elapsed time: 9 weeks

References:

* <https://www.marsdd.com/mars-library/product-development-the-waterfall-methodology-model-in-software-development/>

# **Use case/Test Case**

## **Job Seeker**

### **6.1.1 New User**

|  |  |
| --- | --- |
| **Use Case Name** | **New user registration (sign up)** |
| **Use Case ID** | 1 |
| **Actor** | User |
| **Type** | Primary |
| **Scenario** | A web page which consists of a form having some text fields and buttons etc. |
| **Pre-Condition** | Signup form showing. |
| **Post Condition** | User is successfully registered. |
| **Description** | In this use case user can registered into app by adding the information.  This instruction must be implemented.   * Font size (16) for all fields * First Name (Length 15 character) * Last Name (Length 15 character) * Address (Length 255 character) * CNIC (Length 15 character) * Mobile Number (Length 11 digit) * Password (Minimum 8 character)   After adding the app check that weather the information entered by user is valid. once user is added then he/she is eligible for use the app. |
| **Normal Work Flow** |  |
|  | |  |  | | --- | --- | | **User** | **System** | | 1. User requests from system to add a new user in database. |  | |  | 1. System returns a form which consists of text fields or buttons etc. to gather user’s information. | | 1. After gathering information users forwarded the form to system. |  | |  | 1. System adds a new user to database and return a “Welcome Message”. | |  |  | |

### **6.1.2 Student update**

|  |  |
| --- | --- |
| **Use Case Name** | **User update** |
| **Use Case ID** | 2 |
| **Actor** | User |
| **Type** | Primary |
| **Scenario** | A web page which consists of a form having some text fields and buttons etc. |
| **Pre-Condition** | User must be registered before. |
| **Post Condition** | User information is updated. |
| **Description** | In this use case user can UPDATE his info into app by entering his id and password. After entering, the app check that weather the information entered by user is valid .if user is valid then he can update his information. |
| **Normal Work Flow** |  |
|  | |  |  | | --- | --- | | **User** | **System** | | 1. User requests from system to update user information in database. |  | |  | 1. System returns a form which consists of text fields or buttons etc. to gather user’s information. | | 1. After updating information users forwarded the form to system. |  | |  | 1. System updates a user to database and return a “SUCCESSFULLY UPDATED MSG”. | |  |  | |

### **6.1.3 Delete**

|  |  |
| --- | --- |
| **Use Case Name** | **User delete** |
| **Use Case ID** | 3 |
| **Actor** | User |
| **Type** | Primary |
| **Scenario** | A web page which consists of a form having some text fields and buttons etc. |
| **Pre-Condition** | User must be registered before. |
| **Post Condition** | User information is deleted. |
| **Description** | In this use case user can delete his account by entering his id and password. After entering, the app check that weather the information entered by user is valid .if user is valid then he can delete his account. |
| **Normal Work Flow** |  |
|  | |  |  | | --- | --- | | **User** | **System** | | 1. User requests from system to delete user information in database. |  | |  | 1. System returns a form which consists of text fields or buttons etc. to gather user’s information. | | 1. After entering information users forwarded the form to system. |  | |  | 1. System deletes a user from database and return a “SUCCESSFULLY DELETE MSG”. | |  |  | |

### **6.1.4 Drop Location**

|  |  |
| --- | --- |
| **Use Case Name** | Drop location |
| **Use Case ID** | 5 |
| **Actor** | User |
| **Type** | Primary |
| **Scenario** | A web page which shows different location |
| **Pre-Condition** | Select destination form showing. |
| **Post Condition** | Destination selected successfully. |
| **Description** | In this use case user can Select Destination .  After selecting the app check that weather the destination selected is valid. Once user selects the destination he/she redirects to Select Transportation page. |

### **6.1.5 Pick Location**

|  |  |
| --- | --- |
| **Use Case Name** | pick location |
| **Use Case ID** | 6 |
| **Actor** | User |
| **Type** | Primary |
| **Scenario** | A web page which shows different locations |
| **Pre-Condition** | Select pick location form showing. |
| **Post Condition** | location selected successfully. |
| **Description** | In this use case user can Select pick location.  After selecting the app check that weather the pick location selected is valid. Once user selects the pick location he/she redirects to Select Transportation page. |

# **Prototyping**

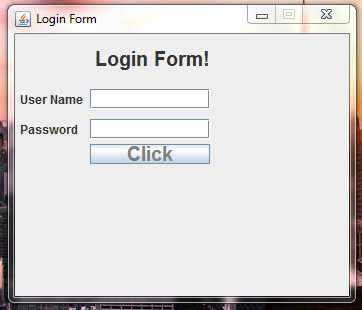
The first prototype developed by the team was to mitigate one of the most important risks. We developed a Smart App using Samsung Smart Things API.

## 7.**1 Login Page**

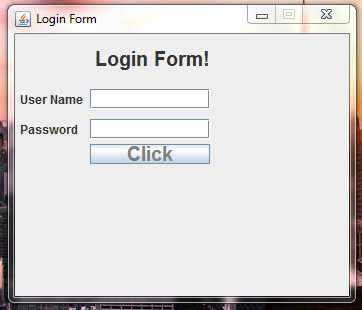
The features implemented in the prototype included the following significant capabilities:

### **7.1.1 Admin Login page**

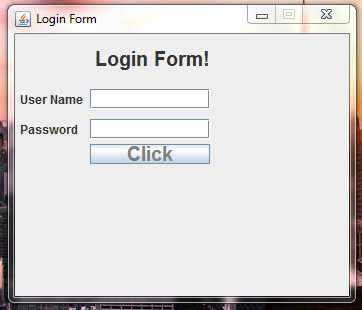
Login page implemented.



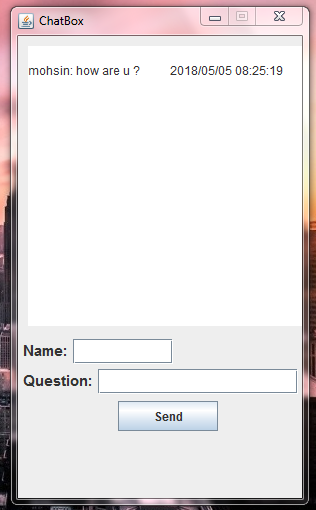
### **7.1.2 Student Login Page**



### **7.1.3 Driver Login page**



## **Chat Box**



# **Prepare DFD**

## **Listing all Use Cases**

* Admin Login
* Student Login
* Driver Login
* Student Signup
* Driver Signup
* Admin Signup
* Generate Pass
* Generate Challan Form
* Student Cancel Ride
* Bus Monitoring

### Selecting Use Case

### Admin Login

Assigning Name to Use Case

### Login Admin.

## Main Steps:

1. Enter User Name and Verify User Name
2. Enter Password and Verify Password
3. Enter Captcha and Verify Captcha

## Defining Conditions According to Situation

* Admin have to enter username correctly otherwise an error message will occur
* Admin have to enter password correctly otherwise an error message will occur
* Admin have to enter captcha correctly otherwise an error message will occur
* If he entered all the fields correctly then show the next page.

## Identifying Entities

* Admin

## Identifying Data Flow

* Admin to login account
* Login account to data base

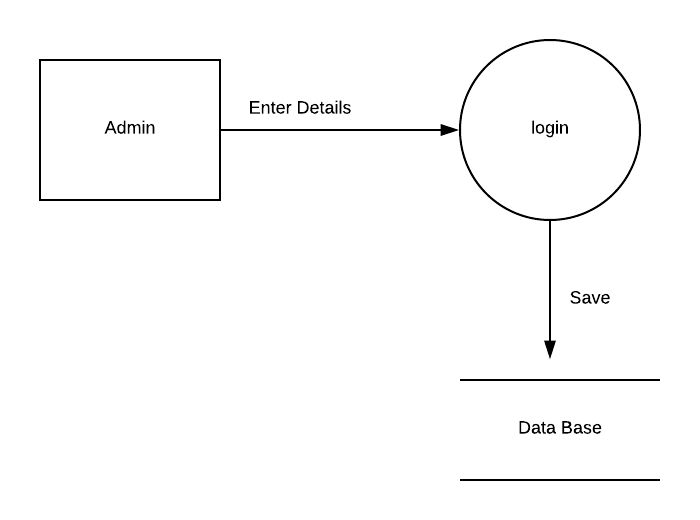
## Identifying data store

* Database

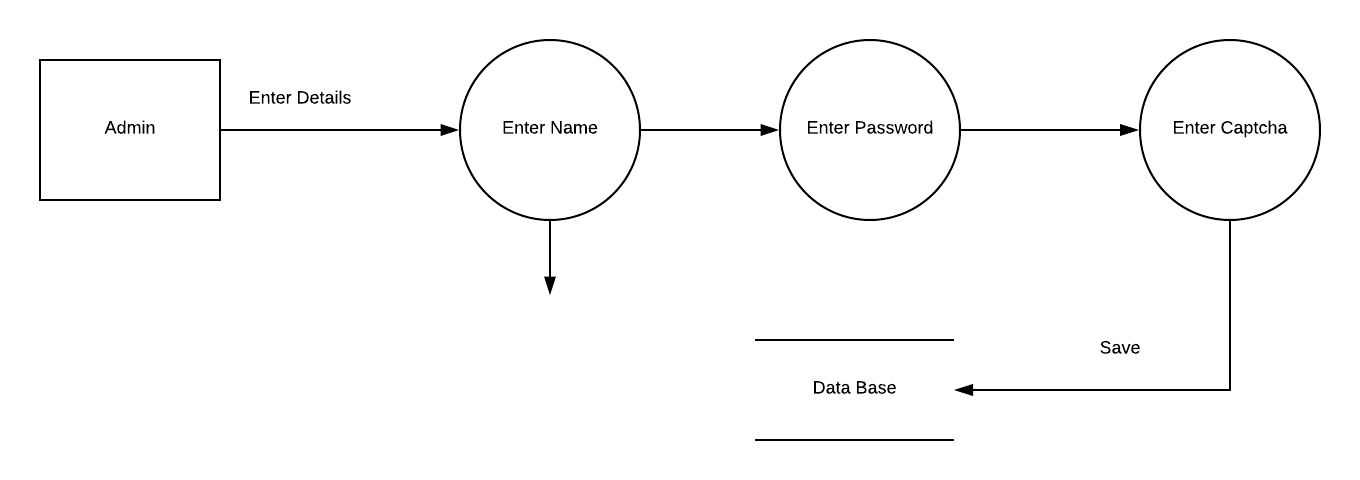
## Identify Data

Admin Information

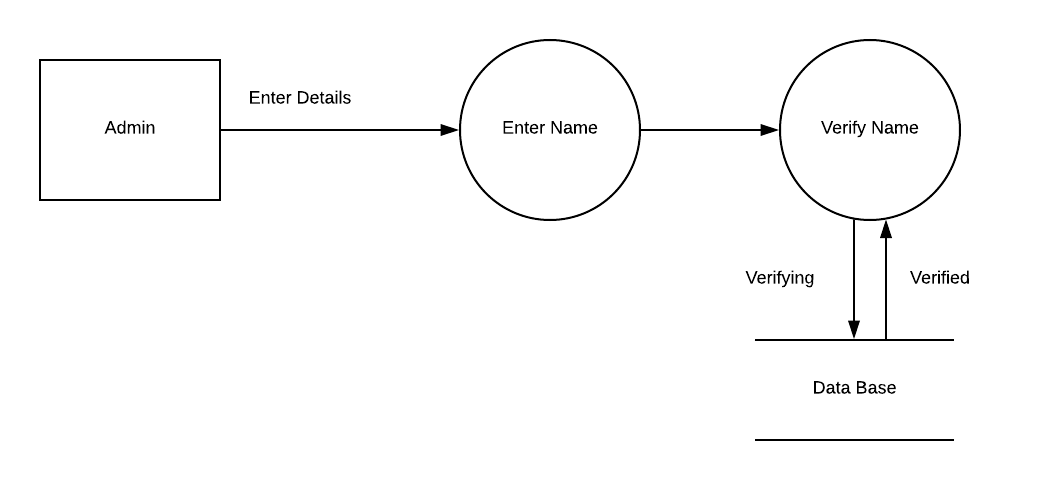
# Context Level Diagram

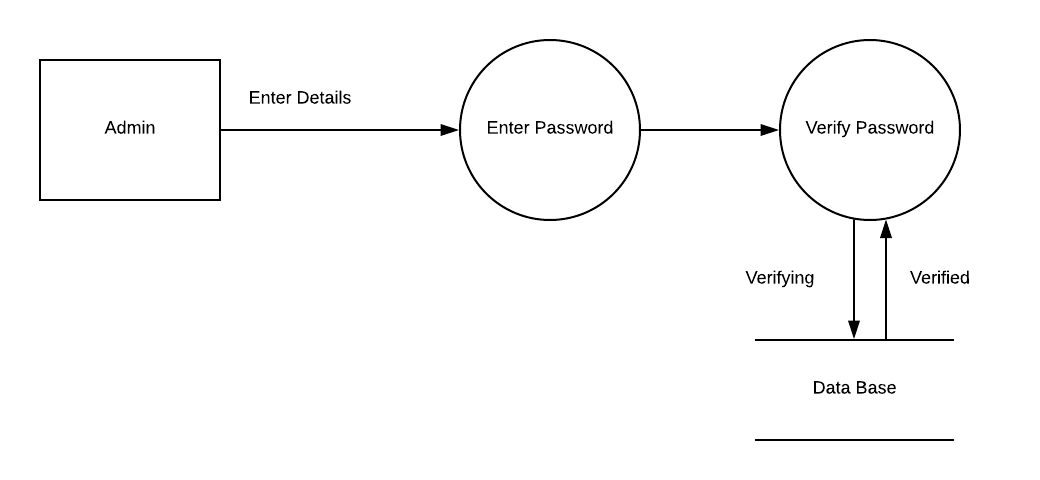


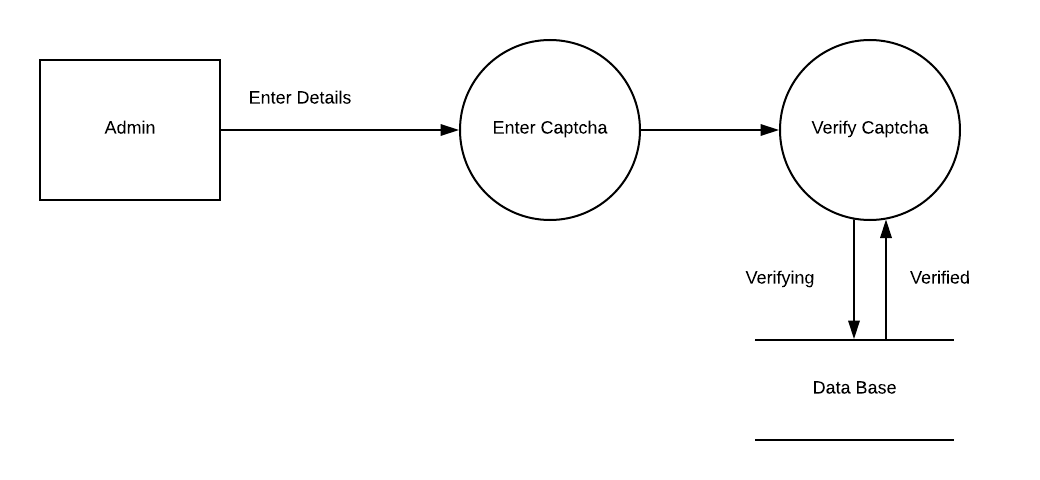
## Level 0 Diagram:



## Level 1 Diagram







## Selecting Use Case

Student Can Generate Challan

## Assigning Name to Use Case

Generating challan

## Main Steps:

1. Login to the Account
2. Ask for challan

## Defining Conditions According to Situation

* Student have to login to the valid account
* Student have to ask for challan
* Product will verify the details and prints it

Identifying Entities

Student

## Identifying Data Flow

Login account to data base

## Identifying data store

Database

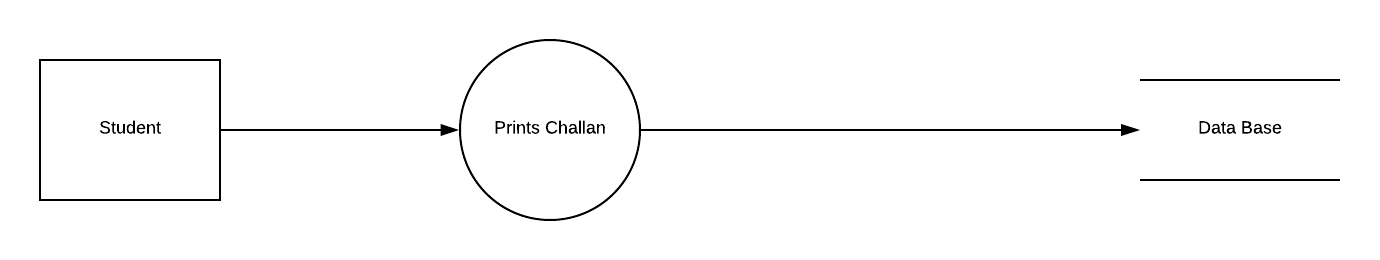
## Identify Data

Student Information

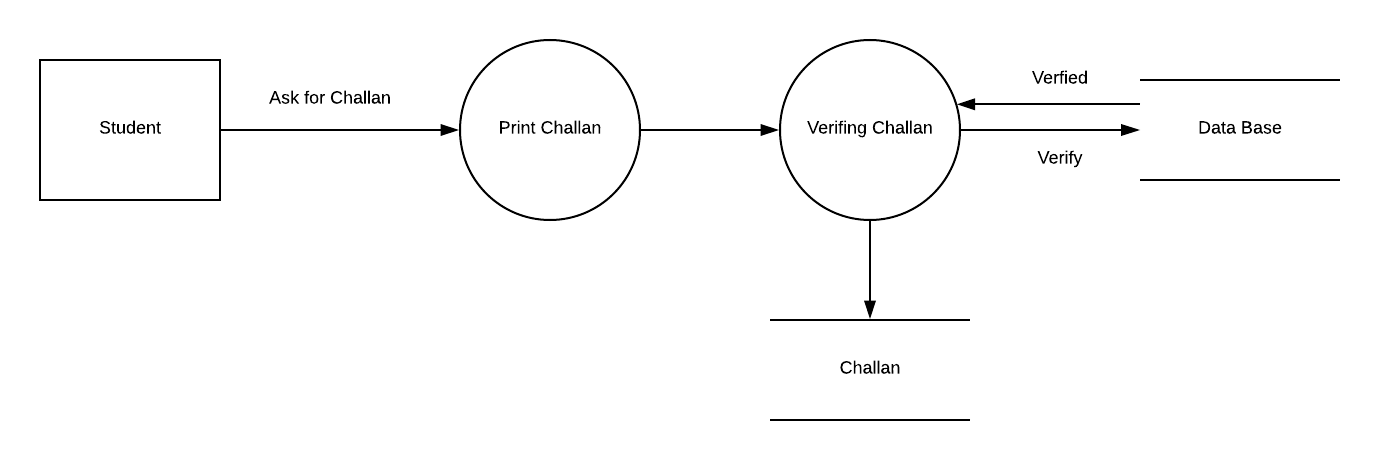
## Context Level Diagram

## 

## Level 0 Diagram



## Level 1 Diagram



### Selecting Use Case

### Student Can Generate Pass

## Assigning Name to Use Case

### Generating Pass

## Main Steps:

1. Login to the Account
2. Enter the valid number of paid challan form
3. Ask for pass

## Defining Conditions According to Situation

* Student have to enter username correctly otherwise an error message will occur
* Student have to enter password correctly otherwise an error message will occur
* Student have to enter captcha correctly otherwise an error message will occur
* If he entered all the fields correctly then show the next page.

## Identifying Entities

## Student

## Identifying Data Flow

## Login account to data base

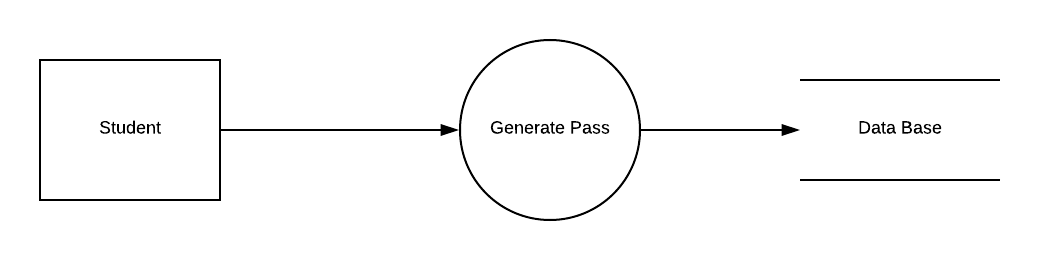
## Identifying data store

## Database

## Identify Data

## Student Information

## Level 0 Diagram



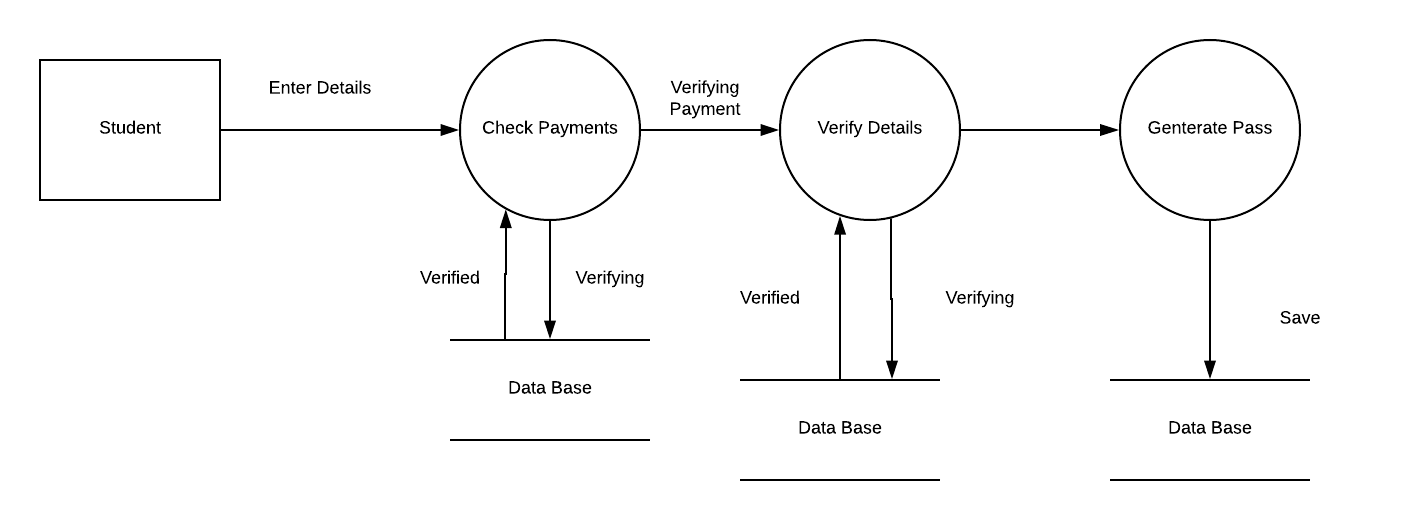
## Level 0 Diagram

## 

## Level 1 Diagram

## 

## Level 1 Diagram



### Selecting Use Case

### Student Can login

## Assigning Name to Use Case

### Login student

## Main Steps:

1. Enter User Name and Verify User Name
2. Enter Password and Verify Password
3. Enter Captcha and Verify Captcha

## Defining Conditions According to Situation

* Student have to enter username correctly otherwise an error message will occur
* Student have to enter password correctly otherwise an error message will occur
* Student have to enter captcha correctly otherwise an error message will occur
* If he entered all the fields correctly then show the next page.

## Identifying Entities

## Student

## Identifying Data Flow

## Login account to data base

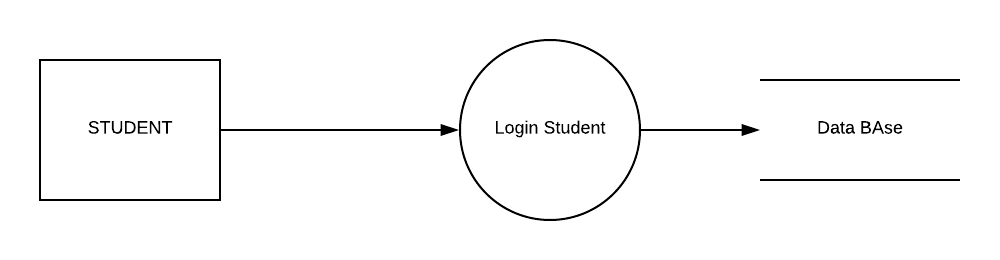
## Identifying data store

## Database

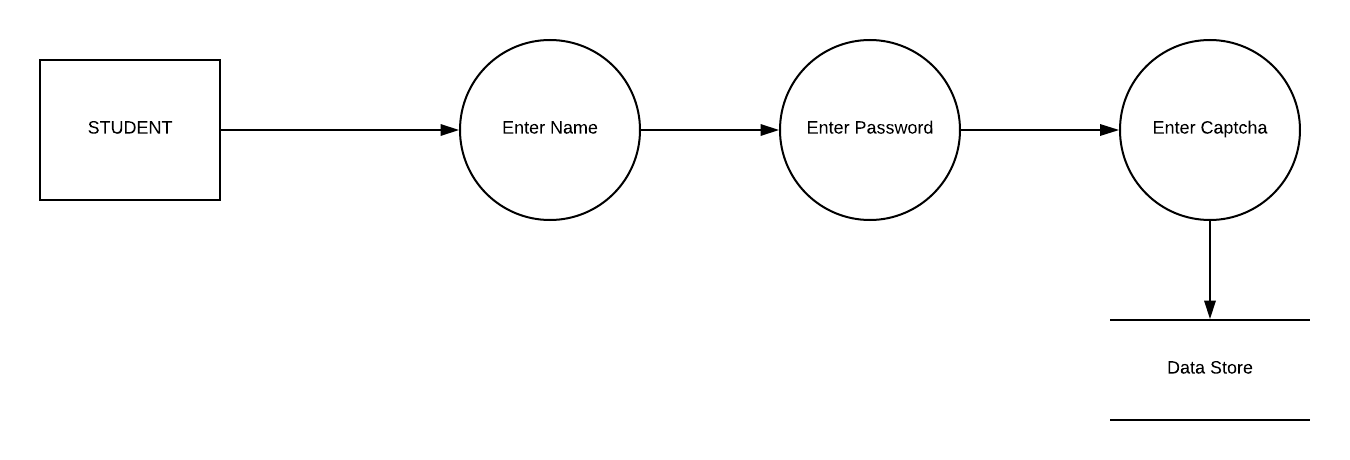
## Identify Data

## Admin Information

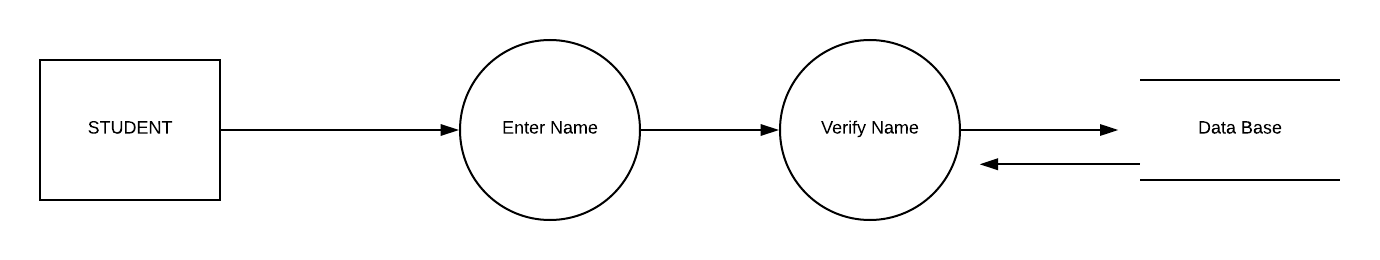
## Context Level Diagram:

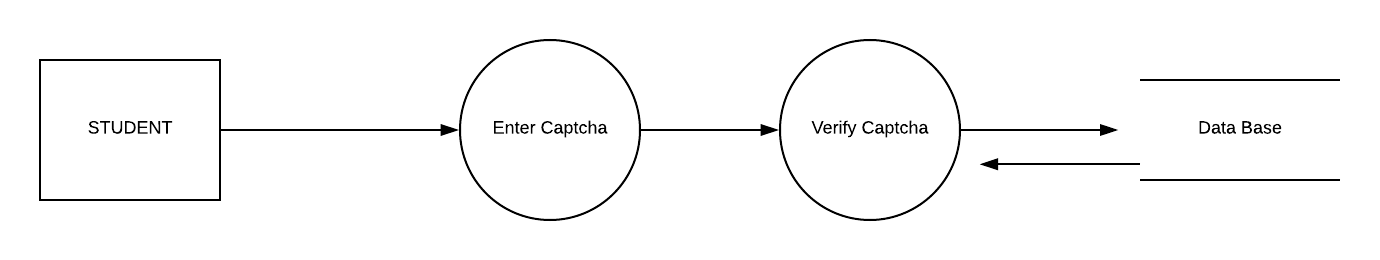


## Level 0 Diagram:



## Level 1 Diagram





### Selecting Use Case

### Student Can Cancel Rides

## Assigning Name to Use Case

### Cancel Rides

## Main Steps:

1. Login to the account
2. Enter the bus number
3. Ask for ride cancelling

## Defining Conditions According to Situation

* Student have to enter username correctly otherwise an error message will occur
* Student have to enter password correctly otherwise an error message will occur
* Student have to enter captcha correctly otherwise an error message will occur
* If he entered all the fields correctly then show the next page.

## Identifying Entities

## Student

## Identifying Data Flow

## Login account to data base

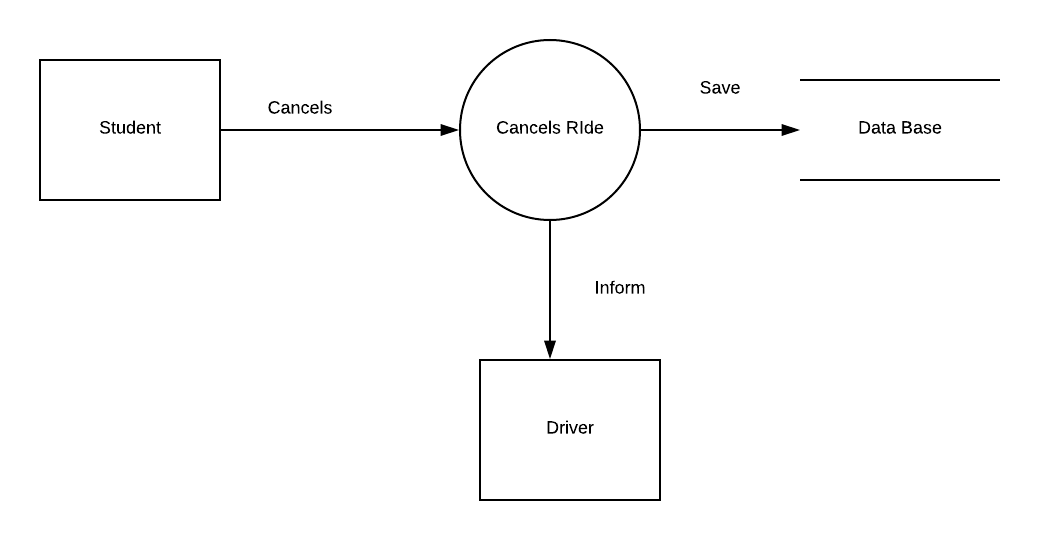
## Identifying data store

## Database

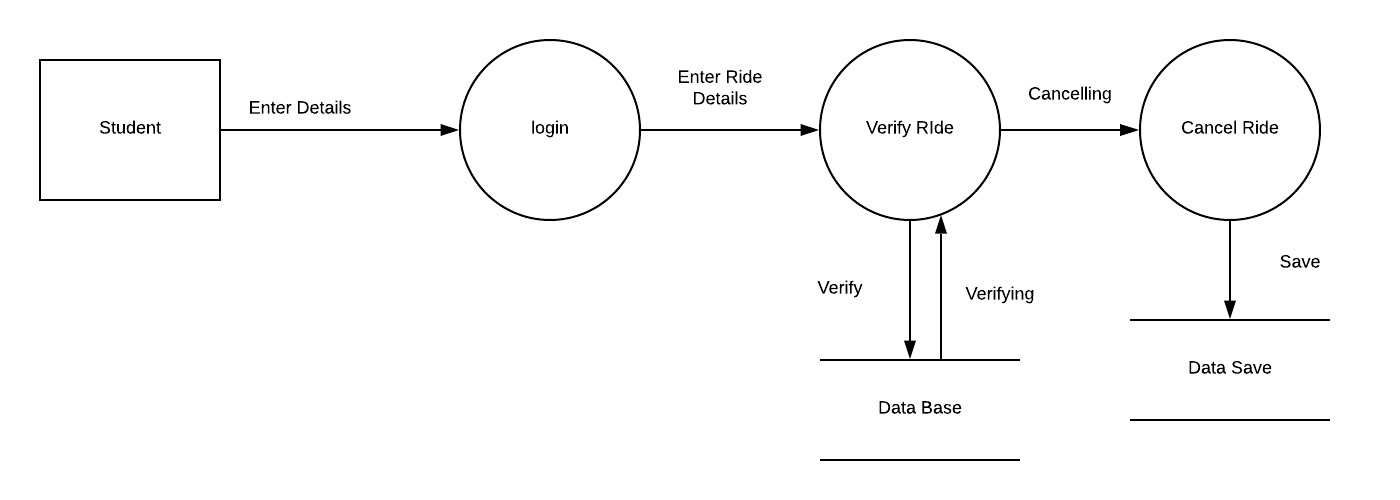
## Identify Data

## Admin Information

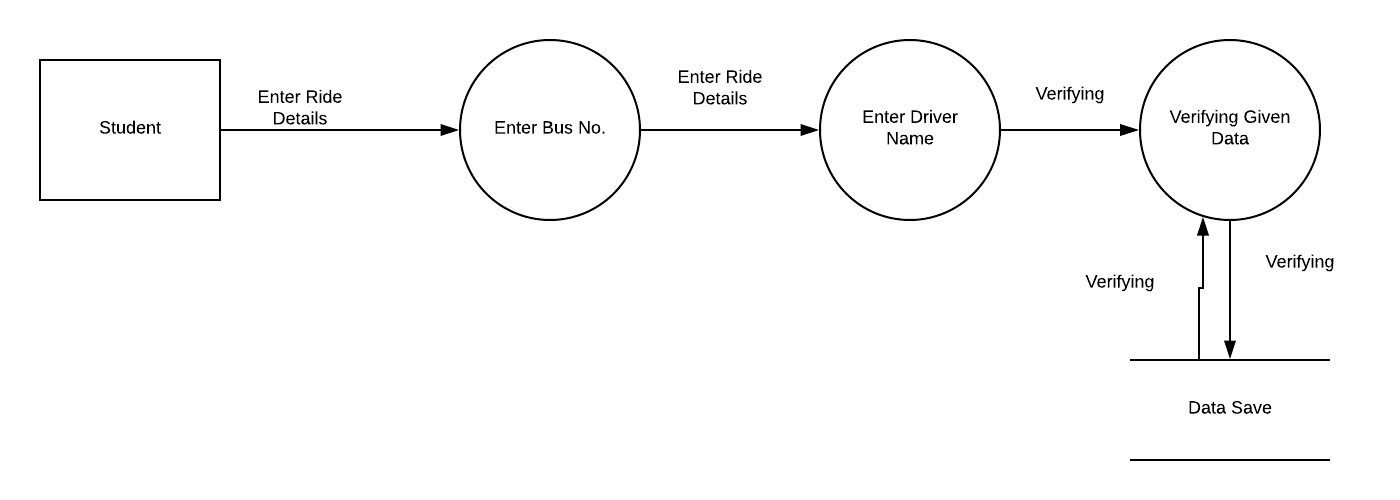
## Context Level Diagram:

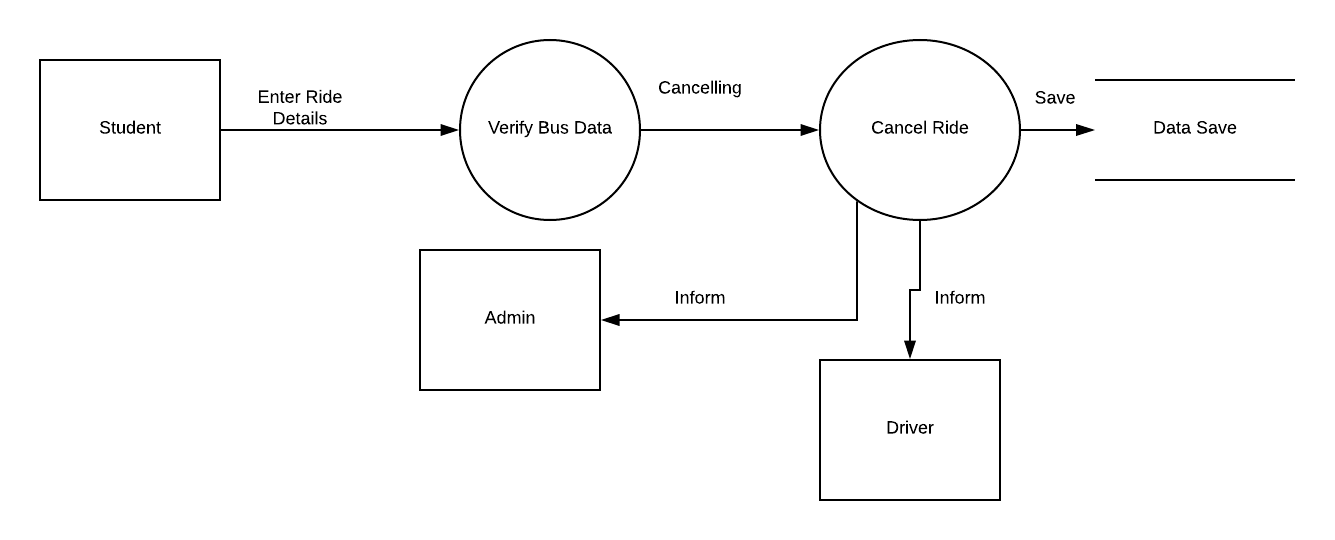


## Level 0 Diagram



## Level 1 Diagram





### Selecting Use Case

### Driver Can Signup

## Assigning Name to Use Case

### Signup Driver

## Main Steps:

1. Enter User Name and Verify User Name
2. Enter Password and Verify Password
3. Enter Captcha and Verify Captcha

## Defining Conditions According to Situation

* Driver have to enter username correctly otherwise an error message will occur
* Driver have to enter password correctly otherwise an error message will occur
* Driver have to enter captcha correctly otherwise an error message will occur
* If he entered all the fields correctly then show the next page.

## Identifying Entities

## Driver

## Identifying Data Flow

## Login account to data base

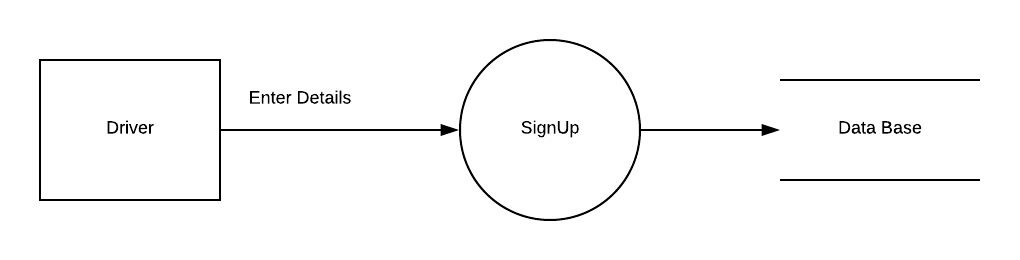
## Identifying data store

## Database

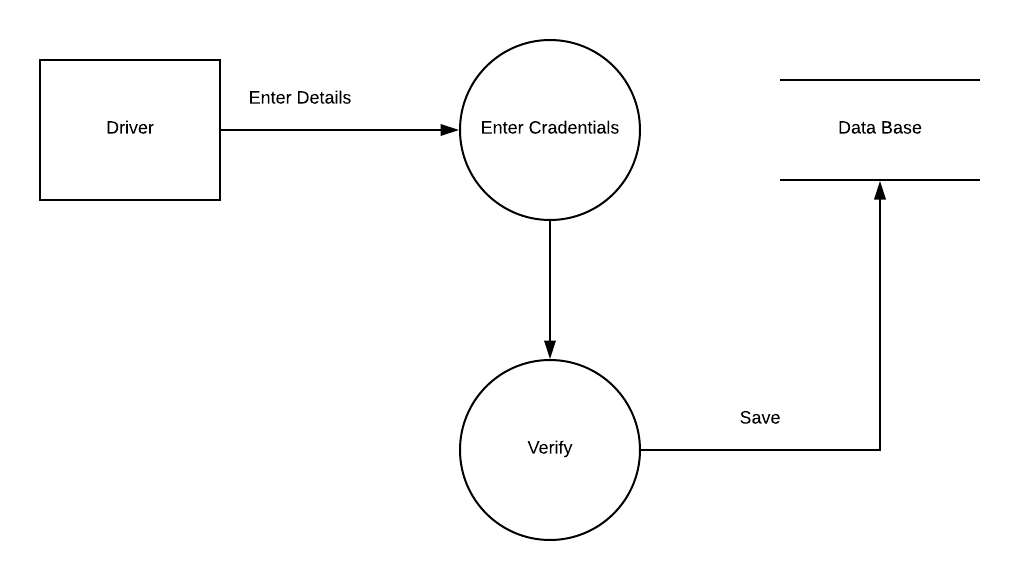
## Identify Data

## Admin Information

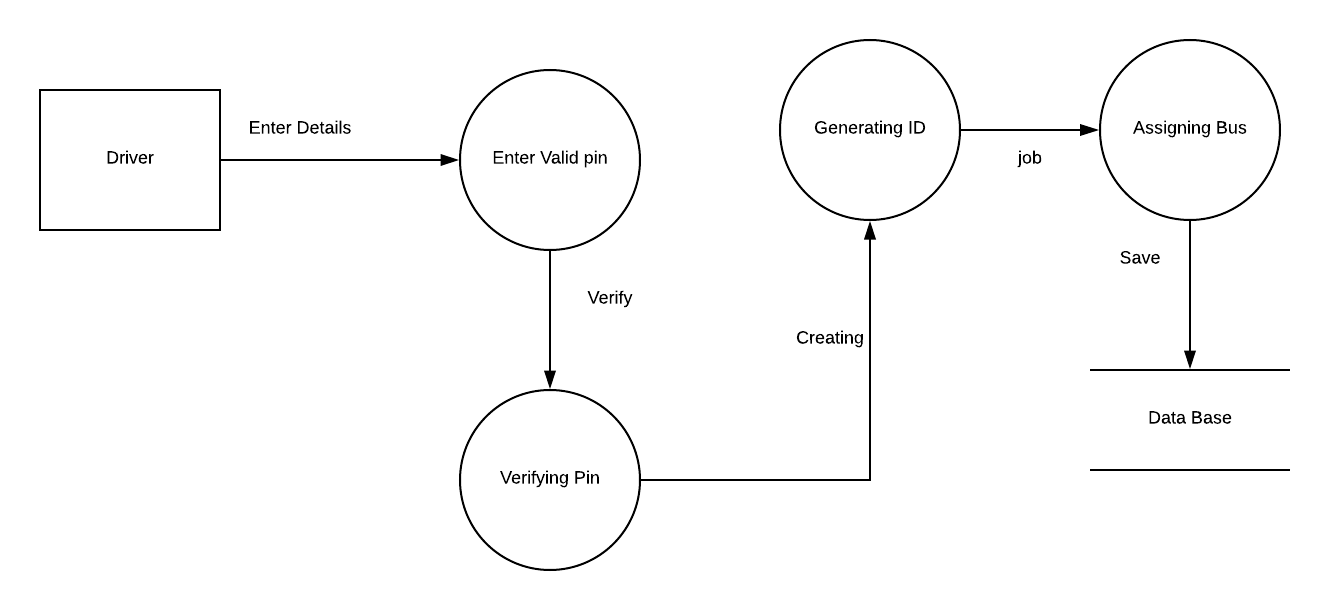
## Context Level Diagram



## Level 0 Diagram



## Level 1 Diagram



# **Decision Tree**

## **Use Cases:**

* Student Account sign in
* Driver/Admin Account sign in
* Student login
* Driver login/Admin login
* Admin update student info
* Mapping
* Canceling route
* Updating Route

# **9.13** Decision Tree:

* Student Account sign in

|  |
| --- |
| Allow user to create account |

Create account

App access

Yes

Able to have the access on his/her data and can see driver info

Valid Sign-in

signin

|  |
| --- |
| Student Sign-in |

No

Error message

Invalid E-mail address

Email option

Invalid option

* Driver Account sign in

|  |
| --- |
| Allow user to create account |

Create account

Able to have the access on his and student information

App access

Yes

Valid Sign-in

signin

|  |
| --- |
| Driver Sign-in |

No

Error message

Invalid Phone no

Invalid pin

Pin option

Phone no option

Invalid option

* Admin Account sign in

|  |
| --- |
| Allow user to create account |

Create account

Able to have the access of whole product

App access

Yes

Valid Sign-in

signin

|  |
| --- |
| Admin Sign-in |

No

Invalid option

Error message

Key option

Info

Invalid Key

|  |
| --- |
| Allow user to see his / Driver information |

* **Student Log-in**

Track bus location

Allow user to monitor driver and other busses location through gps in his/her specific route

Contacting

Yes

Cancel ride

Allow user to cancel the ride if he wants

Able to contact with driver in case of emergency

Valid log in

|  |
| --- |
| Student log in |

No

User name option

Password option

Capcha option

Invalid Capcha

Invalid password

Invalid User name

|  |
| --- |
| Allow user to have the access on his and student information |

* **Driver Log-in**

Tracking

Able to contact with the Student in case of emergency

Allow user to change route if needed

Change route

Contacting

Yes

Allow user monitor student location through gps

Valid log in

|  |
| --- |
| Driver log in |

No

Invalid User name

Invalid password

Invalid pin

Pin option

Password option

User name option

|  |
| --- |
| Allow user to have the access on his and student information |

Info

* **Admin Log-in**

Tracking

Allow user monitor student and driver location through gps

Updating route

Allow user to update route for driver for next location

Able to contact with the Student and driver in case of emergency

Allow user to Delete/update student and driver data

Contacting

Yes

Delete and update

Valid log in

|  |
| --- |
| Admin log in |

No

Invalid User name

Invalid password

Invalid key

Key option

Password option

User name option

* **Mapping**  Access

Allow user to use and access location on map

|  |
| --- |
| Driver is able to access student location through gps in map |

Student tracking

Student and Driver tracking

Admin is able to access student and driver location through gps in map

Driver tracking

Yes

Student is able to access driver location through gps in map

Valid Selection?

signin

|  |
| --- |
| Google  map |

No

Error message

Unable to access location

Weak signal

Invalid option

* **Contacting**  Access

Allow user to use and access contacts

|  |
| --- |
| Driver is able to contact student in case of emergency |

Contact Student

Contact Student and Driver

Admin is able to contact student and driver if necessary

Contact Driver

Yes

Student is able to Contact driver in case of emergency

Valid Selection?

signin

|  |
| --- |
| Contacts |

No

Error message

Unable to contact

Weak signal

Invalid option

# **Decision Table**

## **Listing all Use Cases:**

* Passenger login
* Passenger sign up
* Driver Sign up
* Admin Block passenger
* passenger Cancel a ride
* Pick Location
* Fare estimate
* Drop location
* Bus tracking
* Admin Delete driver

# **10.** Use Case Description

## Passenger Logic

Make sure that the website to be accessed is available and in running condition.

Make sure the data required to login is available. If the website is not access or not in running condition the Passenger will not be able to login account and will not be able to book a ride online.

If user name and password is invalid then the Passenger can get new password by having a click on the forget password and verify his account by getting a code on his number. If password and username is incorrect then try again. if Passenger has logic issue then he may register his self-first.

### Interpretation:

Case 1 – Username and password both were wrong. The user is shown an error message.

Case 2 – Username was correct, but the password was wrong. The user is shown an error message.

Case 3 – Username was wrong, but the password was correct. The user is shown an error message.

Case 4 – Username and password both were correct, and the user navigated to homepage

## Passenger sign up

For passenger sign up require email, name, password and phone number. Account will not be registered if one of these requirements not fulfill. The error message show on specific errors.

### Interpretation:

Case 1 – name and password were wrong. The user is shown a name format and password error.

Case 2 – email and number were wrong. The user is shown an invalid email and number error.

Case 3- all requirements were valid. the user is shown an account interface.

## Driver Sign up

Make sure that the website to be accessed is available and in running condition.

Make sure the data required to login is available. If the website is not access or not in running condition the Driver will not be able to login account and will not be able to book a ride online.

If user name and password is invalid then the driver can get new password by having a click on the forget password and verify his account by getting a code on his number. If password and username is incorrect then try again. if driver has logic issue then he may register his self-first.

### Interpretation:

Case 1 – phone number and email address both were wrong. The user is shown an error message.

Case 2 – email was correct, but the city was wrong. The user is shown an invalid city error message.

Case 3 – Username was valid and the password was valid. The user is shown an ID interface.

Case 4 – Username and password both were wrong, and the user shown an error message.

## Admin Block passenger

Admin can block the passenger by driver report on passenger due to some bad behavior of passenger with driver. If the driver waiting time exceed and passenger not arrived on pick point. Admin block the passenger is this condition.

### Interpretation:

Case 1-user wait time exceed then passenger is shown a block account message.

Case 2- driver report on passenger then passenger is shown a block account message.

## Cancel ride

Passenger can cancel a ride in some condition but if the passenger cancel a ride after 10 mints then the passenger charged the fare for book a ride.

## Passenger Pick location

Passenger set his pick location to book a ride. If the pick location of a passenger is out of range then the message shown the serve of online ride share is not available here. If the pick location is far from live location then passenger is shown a confirm msg for pick location.

## Fare Estimate

In Fare estimate not only include km on your ride but the traffic during ride and waiting charges also include in fare estimate. If the passenger gave extra charges then extra charges saved in passenger online ride share account. passenger can use this amount in next ride.

## Drop location

Passenger can set his drop location where he\she set destination. If the drop location is out of range then message shown to passenger your drop location is not in out of range. If the drop location and pick location same then the confirmation message show to passenger.

## Bus Tracking

Passenger can see the live location of bus on map. Bus location continuous update. If the passenger not book ride the bus live location not shown to the passenger. If the passenger books a ride and confirmed then passenger shown a live location of bus. Time estimate can show to passenger. in time estimate the traffic wait time also included.

## Admin Delete Driver

Admin can delete driver by few conditions. If the driver gave resign report to the Admin. Upon this condition the account of the driver will deleted. Another case is if the driver not using his account for a long time. In this condition admin gave a warning message first if the driver not response on warning. The driver account deletes on this condition.

# Use Case Decision Table

## Passenger Login

The condition is simple if the user provides correct username and password the user will be redirected to the homepage. If any of the input is wrong, an error message will be displayed

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **cONDItions** |  | | | |
| Username | T | T | F | F |
| Password | T | F | T | F |
| **ACTION** |  |  |  |  |
| Error Message Display |  | X | X | X |
| Passenger Home Screen Display | X |  |  |  |

## Passenger Sign up

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **cONDItions** |  | | | |
| Email | T | T | F | F |
| Full name | T | F | T | F |
| Strong Password | T | F | T | F |
| Valid Phone number | T | T | T | F |
| **ACTION** |  |  |  |  |
| Email error Message Display |  |  | X | X |
| Invalid name format error |  | X |  | X |
| Password Error |  | X |  | X |
| Invalid number Message display |  |  |  | X |
| Drive ID Interface Display | X |  |  |  |

## Driver Sign up

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **cONDItions** |  | | | |
| Select City | T | T | F | F |
| Phone Number | T | F | T | F |
| Email Address | T | T | F | F |
| **ACTION** |  |  |  |  |
| City not found message Display |  |  | X | X |
| Invalid phone number message |  | X |  | X |
| Invalid Email error Message Display |  |  | X | X |
| Passenger ID Interface Display | X |  |  |  |

## Admin Block Passenger

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **cONDItions** |  | | | |
| Wait time exceed | T | T | F | F |
| Bad behavior report from driver | T | F | T | F |
| **ACTION** |  |  |  |  |
| Block Passenger Account | X | X | X |  |

## Passenger cancel a ride

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **cONDItions** |  | | | |
| Passenger sudden plan change | T | T | F | F |
| Driver too far | T | F | T | F |
| **ACTION** |  |  |  |  |
| Ride cancel charges message display | X | X | X |  |
| Ride cancelled | X | X | X |  |

## Passenger pick location

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **cONDItions** |  | | | |
| Set pick location | T | T | F | F |
| Invalid range location | T | F | F | F |
| Search location | T | T | T | F |
| **ACTION** |  |  |  |  |
| Out of range error message |  | X | X | X |
| Location not set message |  |  | X | X |
| Confirmed Pick location | X |  |  |  |

## Fare estimate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **cONDItions** |  | | | |
| Ride KM | T | T | F | T |
| Traffic time period | T | F | F | F |
| Waiting time | T | T | T | F |
| **ACTION** |  |  |  |  |
| Km charges | X | X |  | X |
| Traffic time charges |  |  |  |  |
| Waiting time charges |  | X | X |  |
| Total charges | X | X | X | X |

## Drop location

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **cONDItions** |  | | | |
| Set drop location | T | T | F | F |
| Invalid range location | T | F | F | F |
| Search location | T | T | T | F |
| **ACTION** |  |  |  |  |
| Out of range error message |  | X | X | X |
| Destination not set message |  |  | X | X |
| Confirmed Drop location | X |  |  |  |

## Bus tracking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **cONDItions** |  | | | |
| Confirm Book ride | T | T | F | T |
| Internet access | T | F | T | F |
| Pick location | T | T | T | F |
| **ACTION** |  |  |  |  |
| Ride not book message |  |  | X |  |
| Check internet message |  | X |  | X |
| Bus live Gps tracker | X |  |  |  |

## Admin Delete Driver

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **cONDItions** |  | | | |
| Driver resign Report | T | T | F | F |
| Driver account not in use | T | F | T | F |
| **ACTION** |  |  |  |  |
| Account delete message | X | X | X |  |

## **Process Descriptor**

Didn’t performed by the member

# **Use Cases**

Use Case:

* Use case for Driver Signup
* Use case for Student Login
* Use case for Admin Login
* Use case for Passengers login
* Use case for pick location
* Use case drop location
* Use case for Cancel Ride
* Use case fair estimate

## Use case Description for Driver Signup:

## 

|  |  |
| --- | --- |
| **Use Case Name** | Signup to system |
| **Use Case ID** | 1 |
| **Actor** | Driver |
| **Type** | Primary |
| **Scenario** | A web page which consists of a form having some text fields and buttons etc. |
| **Pre-Condition** | Signup form showing. |
| **Post Condition** | Driver is successfully registered. |
| **Description** | In this use case Driver can registered into app by adding the information.  This instruction must be implemented.   * First Name (Length 15 character) * Last Name (Length 15 character) * Address (Length 255 character) * CNIC (Length 15 character) * Mobile Number (Length 11 digit) * Email address * Password (Minimum 8 character)   After adding the app check that weather the information entered by driver is valid. once user is added then he/she is eligible for use the app. |

## Use case for Student Login to system:

|  |  |
| --- | --- |
| **Use Case Name** | Student Login System |
| **Use Case ID** | 2 |
| **Actor** | User |
| **Type** | Primary |
| **Scenario** | A web page which consists of a form having Login and Password text field and login button. |
| **Pre-Condition** | Login form showing. |
| **Post Condition** | Student is successfully Logged In. |
| **Description** | In this use case user can login into app by adding the information.  This instruction must be implemented.   * Email address * Password (Minimum 8 character)   After adding the app check that weather the information entered by Student is valid. If the information is valid user can then use the app. |

## Use case for Admin Login to system:

|  |  |
| --- | --- |
| **Use Case Name** | Admin Login System |
| **Use Case ID** | 3 |
| **Actor** | User |
| **Type** | Primary |
| **Scenario** | A web page which consists of a form having Login and Password text field and login button. |
| **Pre-Condition** | Login form showing. |
| **Post Condition** | Student is successfully Logged In. |
| **Description** | In this use case admin can login into app by adding the information.  This instruction must be implemented.   * Email address * Password (Minimum 8 character)   After adding the app check that weather the information entered by admin is valid. If the information is valid admin can then use the app. |

## Use case for Passenger Login to system:

|  |  |
| --- | --- |
| **Use Case Name** | Passenger Login System |
| **Use Case ID** | 4 |
| **Actor** | User |
| **Type** | Primary |
| **Scenario** | A web page which consists of a form having Login and Password text field and login button. |
| **Pre-Condition** | Login form showing. |
| **Post Condition** | passenger is successfully Logged In. |
| **Description** | In this use case passenger can login into app by adding the information.  This instruction must be implemented.   * Email address * Password (Minimum 8 character)   After adding the app check that weather the information entered by user is valid. If the information is valid user can then use the app. |

## 2.5 Use case for Drop location:

|  |  |
| --- | --- |
| **Use Case Name** | Drop location |
| **Use Case ID** | 5 |
| **Actor** | User |
| **Type** | Primary |
| **Scenario** | A web page which shows different location |
| **Pre-Condition** | Select destination form showing. |
| **Post Condition** | Destination selected successfully. |
| **Description** | In this use case user can Select Destination .  After selecting the app check that weather the destination selected is valid. Once user selects the destination he/she redirects to Select Transportation page. |

## Use case for pick location:

|  |  |
| --- | --- |
| **Use Case Name** | pick location |
| **Use Case ID** | 6 |
| **Actor** | User |
| **Type** | Primary |
| **Scenario** | A web page which shows different locations |
| **Pre-Condition** | Select pick location form showing. |
| **Post Condition** | location selected successfully. |
| **Description** | In this use case user can Select pick location.  After selecting the app check that weather the pick location selected is valid. Once user selects the pick location he/she redirects to Select Transportation page. |

## Use case for Cancel Ride:

|  |  |
| --- | --- |
| **Use Case Name** | **Cancel Ride** |
| **Use Case ID** | 7 |
| **Actor** | User |
| **Type** | Primary |
| **Scenario** | A web page which consists of a form having text field asking user why to cancel trip. |
| **Pre-Condition** | A ride reservation has already been made. |
| **Post Condition** | Ride Canceled |
| **Description** | In this use case user gives the reason of cancelation.  After adding the reason, the trip reservation canceled and trip payment returned to his bank account after deduction of 30% cancellation charges. |

## 

## Use case fair estimate:

|  |  |
| --- | --- |
| **Use Case Name** | **Fair Estimate** |
| **Use Case ID** | 8 |
| **Actor** | User |
| **Type** | Primary |
| **Scenario** | A web page which consists of a form having some text fields and buttons etc. |
| **Pre-Condition** | Text fields to enter his fair estimates and details show. |
| **Post Condition** | Payment and fair estimate details added successful. |
| **Description** | In this use case user add his amount details (Credit Card details).  After adding the app redirects the user to check fair estimate. |

# Use Case Diagram:

* Use Case Diagram for Admin Signup
* Use Case Diagram for Student Login
* Use Case Diagram for Passenger Login
* Use Case Diagram for Driver Login
* Use Case Diagram for Drop Location
* Use Case Diagram for Pick Location
* Use Case Diagram for Fair Estimate
* Use Case Diagram for Cancel Ride

## Use Case Diagram for Admin Signup:

* Use Case : When the 'Sign Up' option is selected, the software asks details about the member like the member's name, address, phone number, e-mail address etc.
* Action: If proper information is entered then a Admin record is saved and created the account verification mail send to his provided e-mail address.

**Use Case Diagram:**

Admin

## Use Case Diagram for Student Login :

* Use Case: If the 'login' option is chosen, the software asks for the Student his e-mail address and his password to check whether he is a valid Student or not.
* Action: If the Email and Password is valid then the user is redirected to the homepage, otherwise an error message is displayed.

**Use Case Diagram:**

Stude  
nt

## Use Case Diagram for Passenger Login :

* Use Case: If the 'login' option is chosen, the software asks for the Passenger his e-mail address and his password to check whether he is a valid Passenger or not.
* Action: If the Email and Password is valid then the user is redirected to the homepage, otherwise an error message is displayed.

**Use Case Diagram:**

Passenger

## Use Case Diagram for Driver Login :

* Use Case: If the 'login' option is chosen, the software asks for the Driver his e-mail address and his password to check whether he is a valid Driver or not.
* Action: If the Email and Password is valid then the user is redirected to the homepage, otherwise an error message is displayed.

**Use Case Diagram:**

Driver

## Use Case Diagram for Pick Location:

* Use Case: When the Select Pick location' option is chosen, the software asks for the member to select the location from the given list.
* Action: If the member chooses the valid location the user is redirected to the places to visit page, otherwise an error message is displayed.

**Use Case Diagram:**

Admin

Student

## Use Case Diagram for Drop Location:

* Use Case: When the Select Drop location' option is chosen, the software asks for the member to select the location from the given list.
* Action: If the member chooses the valid location the user is redirected to the places to visit page, otherwise an error message is displayed.

**Use Case Diagram:**

Admin

Student

## Use Case Diagram for Fair Estimate:

* Use Case: In this case fair estimate is shown then user add ride if he/she easily pay .
* Action: If the price is not reasonable then his ride cancels.

**Use Case Diagram:**

Driver

GPS

## Use Case Diagram for Cancel Ride:

* Use Case: If the cancel ride option selected then app asks user to add reason to cancel ride.
* Action: If the adds the reason then his ride cancels and the ride payment returns to his bank account after the deduction of 30% cancelation fee.

**Use Case Diagram:**

Driver

Passenger

# **Prepare ER Diagram**

Inappropriate response from the member

# **Architecture Design**

## **About:**

The process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.

## **Working:**

It defines an abstraction level at which the designers can specify the functional and performance behavior of the system. It evaluates all top-level designs. It develops and documents top-level design for the external and internal interfaces.

## **Architectural Design Representation:**

* Architectural design can be represented using the following models.
* Structural model
* Dynamic model
* Process model
* Functional model
* Framework model

## **Architectural Styles:**

A set of *connectors* such as procedure call, events broadcast, database protocols and pipes to provide communication among the computational components. Constraints to define integration of components to form a system.

## **14.6 Object-Oriented Architecture:**

In object-oriented architectural style, components of a system encapsulate data and operations, which are applied to manipulate the data. This architectural style has important characteristics: Objects maintain the integrity of the system. An object is not aware of the representation of other objects. Some of the advantages associated with the object-oriented architecture are listed below.

## **Data-Centered Architecture:**

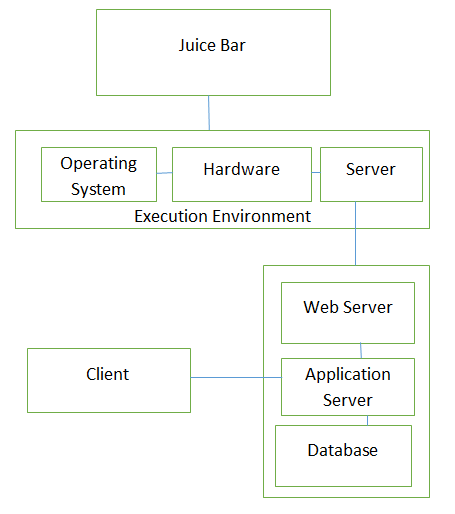
A data-centered architecture has two distinct components: a central data structure or data store (central repository) and a collection of client software. Some advantages of the data-centered architecture are listed below. Clients operate independently of one another. Data repository is independent of the clients. It adds scalability (that is, new clients can be added easily). It supports modifiability.

# **Deployment Design**

## **Data-Centered Architecture:**

Deployment design is a kind of structure diagram used in modeling the physical aspects of the system. It includes hardware required for the system, operating system requirements, plugins/libraries/servers/databases required for the system to operate, technologies required to develop the system.

## **Visual Representation:**



# **Detailed Design**

## **Main Classes:**

The main classes of University Job Seek Portal project are given below:

* Admin
* Student
* Driver
* Registration
* Book Ride
* Cancel Ride

# Class 1 “Admin”

## Data Members

* Name
* Address
* Contact No
* Login ID
* Password

## Methods

* setName(n:String)
* getName()
* setAddress(a:String)
* getAddress()
* setContactNo(n:int)
* getContactNo()
* setLoginID(id:String)
* getLoginID()
* setPassword(p:String)
* getPassword()

# Class 2 “Student”

## Data Members

* Name
* Pickup Address
* Contact No
* Login ID
* Password

## Methods

* setName(n:String)
* getName()
* setPAddress(a:String)
* getPAddress()
* setContactNo(n:int)
* getContactNo()
* setLoginID(id:String)
* getLoginID()
* setPassword(p:String)
* getPassword()

# Class 3 “Driver”

* **Data Members**
* Name
* Address
* Contact No
* Login ID
* Password
* Today Pickup Schedule

## Methods

* setName(n:String)
* getName()
* setAddress(a:String)
* getAddress()
* setContactNo(n:int)
* getContactNo()
* setLoginID(id:String)
* getLoginID()
* setPassword(p:String)
* getPassword()
* setSchedule(s:String)
* getSchedule()

# Class 4 “Registration”

## Data Members

* Type (e.g. student, driver etc.)
* Login ID
* Password
* Fee ( for student)

## Methods

* setType(t:String)
* getType()
* setLoginID(id:String)
* getLoginID()
* setPassword(p:String)
* getPassword()
* setFee(f:int)
* getFee()

# Class 5 “Book Ride”

## Data Members

* Source
* Destination
* Expected Time
* Price (Null for registered users)

## Methods

* setSource(s:String)
* getSource()
* setDestination(d:String)
* getDestination()
* setExpectedTime(et:String)
* setPrice(p:int)
* getPrice()

# **Class 6 “Cancel Ride”**

## Data Members

* Source
* Destination
* Reason (Cancellation Reason)
* Charges (Cancelation)

## Methods

* setSource(S:String)
* getSource()
* setDestionation(d:String)
* getDestination()
* setReason(r:String)
* getReason()
* setCharges(c:int)
* getCharges()

# Relationships

Admin

Registration

Student

Driver

CancelRide

BookRide

# Class Diagram

Admin

name: String

address: String

contactNo: Int

loginId: String

password: String

setName(n: String)

getName()

setAddress(a: String)

getAddress()

setContactNo(n: Int)

getContactNo()

setLoginID(id: String)

getLoginID()

setPassword(p: String)

getPaassword()

Student

name: String

pickupAddress: String

contactNo: Int

loginId: String

password: String

setName(n: String)

getName()

setpickupAddress(a: String)

getAddress()

setContactNo(n: Int)

getContactNo()

setLoginID(id: String)

getLoginID()

setPassword(p: String)

getPaassword()

Registration

type: String

fee: Int

loginId: String

password: String

setFee(n: Int)

getFee()

setLoginID(id: String)

getLoginID()

setPassword(p: String)

getPaassword()

Driver

name: String

address: String

contactNo: Int

loginId: String

password: String

tSchedule: String

setName(n: String)

getName()

setAddress(a: String)

getAddress()

setContactNo(n: Int)

getContactNo()

setLoginID(id: String)

getLoginID()

setPassword(p: String)

getPaassword()

setSchedule(s: String)

getSchedule()

CancelRide

source: String

destination: String

charges: Int

Reason: String

setSource(s: String)

getSource()

setDestination(d: String)

getDestination()

setCharges(c: Int)

getCharges()

setReason(r: String)

getReason()

BookRide

source: String

destination: String

price: Int

expectedTime: String

setSource(s: String)

getSource()

setDestination(d: String)

getDestination()

setPrice(p: Int)

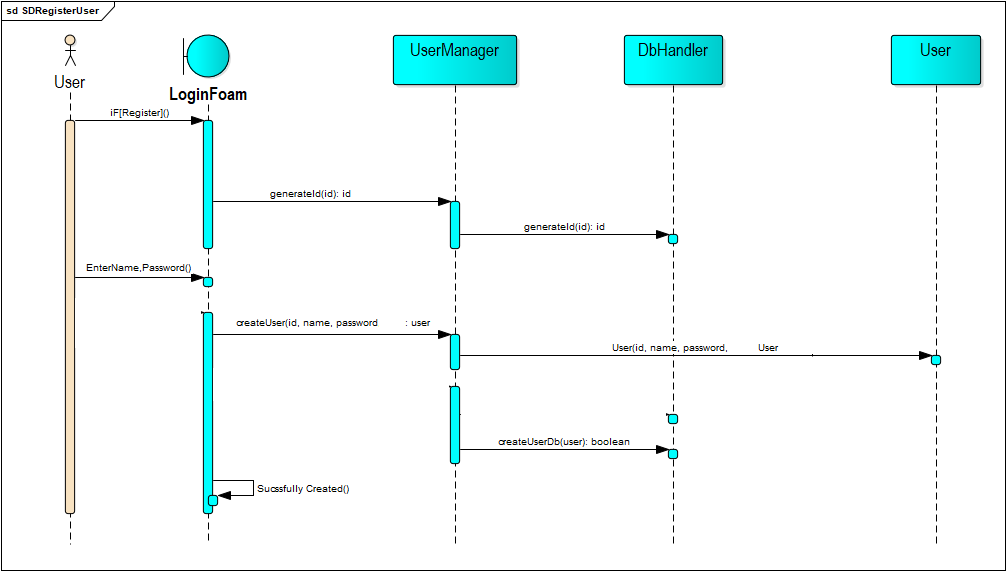
getPrice()

setExpectedTime(et: String)

getExpectedTime()

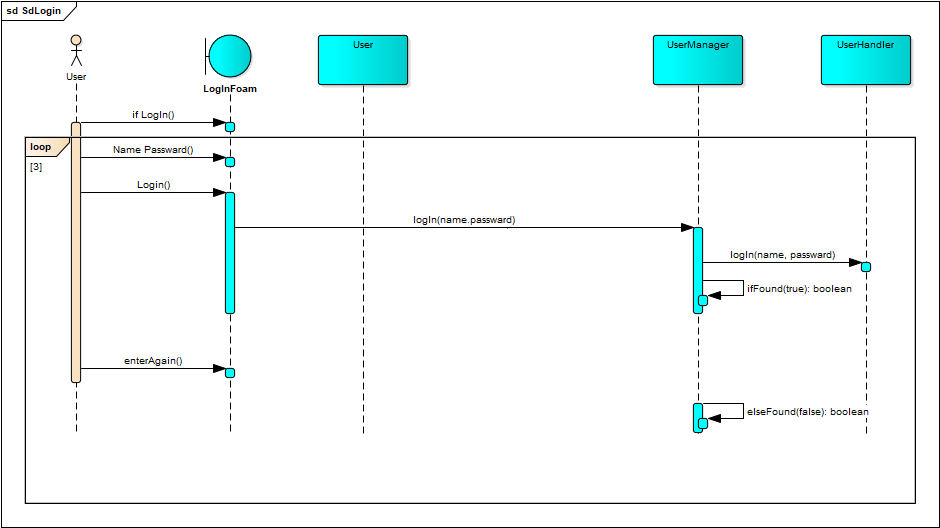
# **Sequence Diagram**

## **User Registration:**



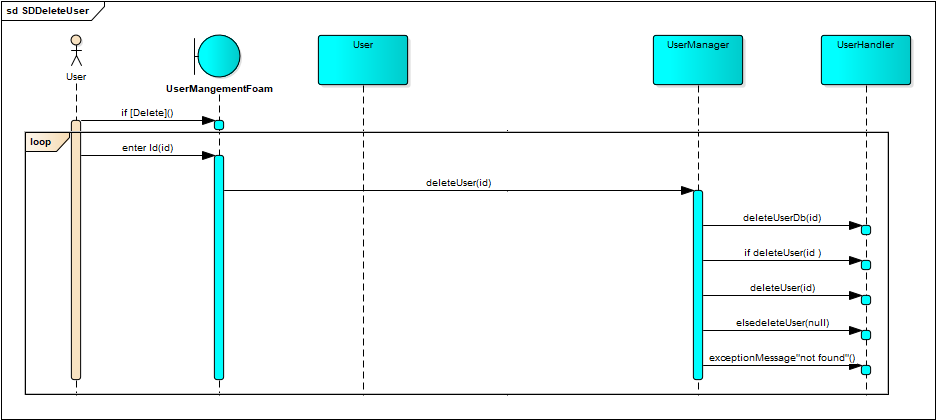
* This Diagram show the user registration sequence.
* We uses three class which are used in this case the first is user manager which is use for calling a function createuser().
* The second class is DbHandler which generate a query to save the valid data of user into a database.
* Third class is User class which have their own attributes like (name,id,password) this class is used for calling constructor.

## **User Login:**

****

* This Diagram show the user login sequence.
* After registration our next step is login in this scenario we can change classes.
* First is User class which have their own attributes like (name, id, password) this class is used for calling constructor.
* Second class is User Manager this class is basically check the user entered information through function named Found having Boolean data type.
* Third class is User Handler.

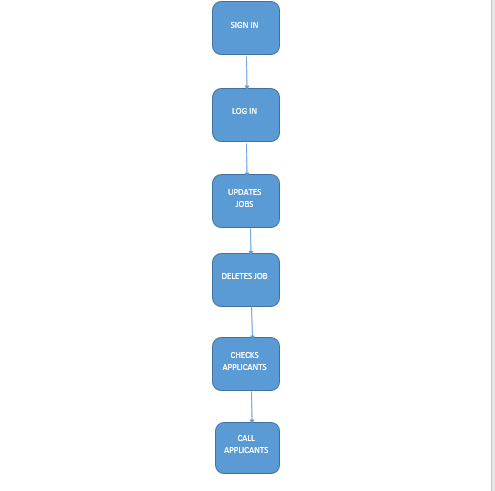
## **Delete User:**



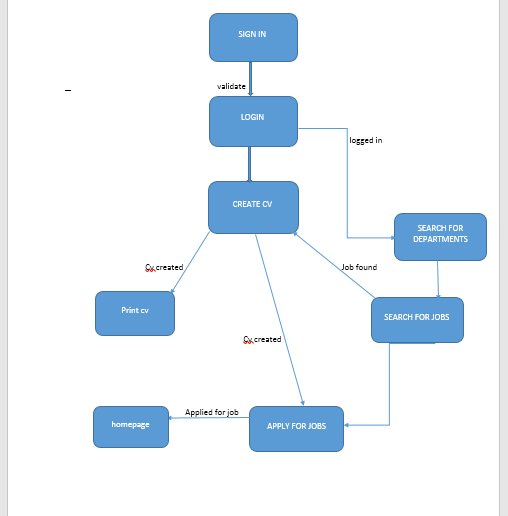
* This Diagram show the sequence of delete user information.
* Three classes are used first one User second is User manager and the third one is User handler.
* This sequence show’s that how user will delete his/her information.

# **State Diagram**

## **Admin:**



## **Applicant:**



# **Appendix A: Glossary**

The definitive source for learning about career, job-hunting, and employment terms — from your team at Quintessential Careers.This glossary of job, career, and employment terms is designed to give job-seekers a quick definition — and then provide links where you can find more details, samples, and much more information. If you’re looking for a job-hunting, employment-related, or career term that is not listed in the Job-Seeker’s Glossary, please.

# **Appendix B: Analysis Models**

Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams

# **Appendix C: Issues List**

* 1. Duplication of jobs - Sometimes job seekers could not reach the employers who originally posted the job.
  2. No proper feedback through portal after employer viewed the profile.
  3. Top job portals in India does not have jobs from start-ups.
  4. Not all the portal has cover letter in the job application.

# **Bibliography**

1. JOOMILA 25 A Beginners Guide by ERIC TIGGELER
2. PHP and MYSQL by W JASON GILMORE.
3. Fundamentals of software Engineering by RAJIB MALL.
4. Java Server Programming – Black Book