**CHAPTER-1**

**INTRODUCTION**

* 1. **OVERVIEW**

The “BusFriend” is an android based autonomous system which tracks the buses. This app is developed for a particular bus operators association. This application allows people to check the bus that they want to catch in real-time on Google map. Through this bus tracking application people can see the exact location of the bus by using his/her android phone. User can see the nearby bus stops and bus routes. They can send feedback and complaints to the bus operators association and give ratings to the bus. Using this application bus owners can also track their bus location and pass emergency notes to bus driver. The location of the bus is obtained in the Google map by the means of latitude and longitude with the help of GPS. This projection of bus on Google map would give a clear idea about the location of the bus the passenger wants to board.

* 1. **SCOPE OF THE PROJECT**

In today’s world of precision and timing, the public transportation has not been up to the mark. This project is developed targeting the increasing uncertainty of bus transport in major cities in Kerala. It aims to provide an application to the general public to use of public transport. The main goal of the proposed work is to improve the bus system by tracking the bus using GPS and show the location on map. GPS tracking system not only improve the efficiency of bus operation, but also encourage commuters to take advantage of bus system. This android app allows monitoring the location of the bus that increases the number of people using the buses for travelling.

This application is user friendly one that anyone can access for free of the cost. The basic idea for this project is to guide the travellers by display the exact location of the bus on map.

**CHAPTER-2**

**SYSTEM ANALYSIS**

**2. SYSTEM ANALYSIS**

System analysis is the process of collecting and interpreting facts, understanding problems and using the information to suggest improvements on the system. This will help to understand the existing system and determine how computers make its operation more effective. The aim of this analysis is to collect the detailed information on the system and the feasibility study of the proposed system. This analysis focuses on the flow of the system module by module and the efficiency of each. To design the proposed system, we need the exact processing logic as well as the extended features of the existing system such as reliability, consistency, storage capacity etc. This report will discuss the advantages and drawbacks/disadvantages of the existing system and the modifications and enhancements can be done. This analysis will concentrate on the information gathering for the efficient, user friendly and reliable system, which will carry forward the features of the existing system.

**2.1 REQUIREMENT ANALYSIS**

Requirements analysis results in the specification of software’s operational characteristics, indicates software’s interface with other system elements, and establishes constraints that software must meet. Requirements analysis allows you to elaborate on basic requirements established during the inception, elicitation, and negotiation tasks that are part of Requirements engineering.

**REQIUREMENT GATHERING**

The requirement gathering can be done by following ways.

* Interview
* Questionnaire
* Website visit

For requirement analysis I choose the questionnaire, website visit and interview.

There are some apps and website are available

* eTechTracker
* Track My Bus
* Bus Gps Tracking
* Pune Bus Guide
* Mumbai M-Indicator

I asked set of questions to the Passengers and Bus drivers.

The questions to the passengers are:

1. The bus always arrives at the destination on time

* Strongly agree
* Agree
* Disagree
* Strongly disagree

1. Is the bus service in your area frequent?

* Yes
* No

1. Have you experienced problem while travelling?

* Yes
* No

1. Bus provides timely and efficient service

* Strongly agree
* Agree
* Disagree
* Strongly disagree

1. Bus operating hours are convenient to all their passengers

* Strongly agree
* Agree
* Disagree
* Strongly disagree

1. The timetable in the bus stand is error free

* Strongly agree
* Agree
* Disagree
* Strongly disagree

1. Bus always inform people of change of timetable in advance

* Yes
* No

1. Have you seen a change in your bus service?

* Yes
* No

1. How reliable do you find the service?

* Very reliable
* Mostly reliable
* Unreliable

1. Communication of bus staff with passengers is clear and helpful?

* Strongly agree
* Agree
* Disagree
* Strongly disagree

1. It is easy to find and access the bus stand?

* Yes
* No

1. How easily can you find the time of your public transport service

* Very easy
* Okay
* Not very easy
* Very hard

1. Whether you are familiar with any kind of Vehicle tracking app?

* Yes
* No

1. Whether you depend on any particular bus?

* Yes
* No
* May be

1. Are you satisfied with the existing bus transport system?

* Yes
* No

The questions to the bus drivers are:

1. How often are you satisfied with your work as a bus driver?

* Always
* Often
* Sometimes
* Never

1. Are you stressed at work?

* Never
* Sometimes
* Often
* Always

1. It causes problems when the time schedule is too long

* Strongly Agree
* Agree
* Disagree
* Strongly Disagree

1. The management listen to your suggestions to changes in the time schedule

* Yes
* No

1. In order to comply with the time schedule, it is difficult to keep within the frames of traffic law

* Strongly Agree
* Agree
* Disagree
* Strongly Disagree

1. Whether the time schedule creates competition among buses?

* Yes
* No

1. Is the contact to the passengers important to you

* Always
* Often
* Sometimes
* Never

1. You only get feedback when something goes wrong

* Yes
* No

1. You have sufficient time to provide passengers good service?

* Always
* Often
* Sometimes
* Never

1. Do you feel that it is important that bus stops and schedules be regular and on time?

* Yes
* No

1. The management creates insecurity among the drivers

* Strongly Agree
* Agree
* Disagree
* Strongly Disagree

1. Do you feel motivated and committed to your work?

* Always
* Often
* Sometimes
* Never

1. The traffic service solves the tasks efficiently

* Strongly Agree
* Agree
* Disagree
* Strongly Disagree

1. Are the time schedules on your ordinary routes reasonable?

* Yes
* No

1. Your union takes well care of your interest

* Yes
* No

**Interview Questions**

1. What do you feel is the major problem in bus transportation is facing today?
2. Where do you get the bus schedules information?
3. What changes would you like to suggest for improving effectiveness in bus?
4. How often do bus services run in your area?
5. What is the main reason behind the competition among buses?
6. What improvements do you think could be made for the better interaction with the passengers?
7. How do you specify the current location of the bus?

**2.2 EXISTING SYSTEM**

There are some android apps are available for track the buses but there are many drawbacks with these apps. It is not user friendly and these apps are only used in urban areas. The existing system needs external hardware such as GPS device, RFID, GSM technology etc. So, this is not economically feasible.

The existing system has some of the drawbacks like

* The exact position of the bus cannot be retrieved.
* This application mainly used only by owners and administrator.
* The bus location cannot be retrieved from anywhere.
* The movement of the bus is also not visible in the Google map.

**2.3 PROPOSED SYSTEM**

The proposed system provides the user to find exact location of the bus from where they are. The position of the bus is displayed in the Google map. Depending on the information like longitude and latitude user can track the bus.

The advantages of proposed system are:

* This app is very user friendly
* Through this app user can track the exact location of bus.
* User can view bus timing
* Estimate the time of arrival.
* User can register the complaints.
* Passengers can send reviews to the bus operation association.
* Passengers can rate the buses.
* Through this app GPS, Google map technologies are merged into one device.

**2.4 FEASIBILITY STUDY**

Feasibility analysis is the procedure for identifying the candidate system, evaluating and electing the most feasible system. When conducted feasibility study understand the need for change or improvement in the current system, which is manually. This is done by investigating the existing system. A feasibility study is conducted to check whether it is

* Possible (to build with the given technology and resources)
* Affordable (given time and cost constraints of the organizations)
* Acceptable

Basically, feasibility study tries to find out whether it is worth developing a new system before actually proceeding to developing it. Certain key considerations are involved in the feasibility analysis are:

* Technical Feasibility
* Economic Feasibility
* Behavioural Feasibility
* Legal Feasibility

**2.4.1 Technical Feasibility**

In Technical feasibility, the main issue is to determine whether the current level of technology can support the proposed system. The proposed system can be developed using the latest technology Android and the Firebase real time database. In this proposed system the GPS technology and Google map are merged into one unit. Considering all advantages of the proposed system it will be technically feasible and we recommended to use the proposed system.

**2.4.2 Economic Feasibility**

In cost-benefit analysis various benefits and costs involved are considered, calculated and compared, if the benefits are more than the cost, the project is considered economically feasible. Economic feasibility is the most important and frequently used method for evaluating the effectiveness of the proposed system. It is very essential with increased efficiency of the cost versus the benefit and savings are expected from the proposed system

**2.4.3 Behavioral Feasibility**

In behavioral feasibility, the management considers that the proposed system will fulfill the requirements of the users, i.e. whether the proposed system covers all the jobs that were expected by the end user and whether it has considerable improvements. Understanding the advantages and efficiency of the proposed system, we decided to develop a new system. The proposed system helps to avoid the delay in bus schedule and it shows the exact location of the buses.

**2.4.4 Legal Feasibility**

Legal feasibility study determines whether the proposed system conflicts with legal requirements. It determines whether the proposed system violates any copyright act or any rules in the organization. Since the proposed system does not violate any copyright act and it does not break any rules in the organization the proposed system is legally feasible.

**2.5 SYSTEM REQUIREMENT SPECIFICATION**

System requirements are expressed in a software requirement document. The Software requirement specification (SRS) is the official statement of what is required of the system developers. This requirement document includes the requirements definition and the requirement specification. The software requirement document is not a design document. It should set out what the system should do without specifying how it should be done. The requirement set out in this document is complete and consistent. The software specification document satisfies the following: -

* It specifies the external system behaviour.
* It specifies constraints on the implementation.
* It is easy to change.
* It serves as reference tool for system maintainers.
* It records forethought about the life cycle of the system.
* It characterizes acceptable response to undesired events.

**2.5.1 ACTOR IDENTIFICATION**

An actor is someone or something that interacts with the system. An actor exchanges information with the system. Asking certain questions as detailed below can identify the actors of the application.

|  |  |  |
| --- | --- | --- |
| **1.** | Who will use the main functionality of the system? | Administrator, Owner, Bus driver, Passenger. |
| **2.** | Who will lead support from the system and do their tasks? | Administrator, Owner, Bus driver, Passenger. |
| **3.** | Who will maintain and administrate the system? | Administrator. |
| **4.** | With which other systems, does this system need to interact? | Database. |
| **5.** | Who was interest in the result produced by the system? | Administrator, Owner, Bus driver, Passenger. |

As per the above answers we can conclude the actors. They are:

* Admin
* Owner
* Bus driver
* Passenger

**2.5.2 USECASE IDENTIFICATION**

A use cases represents the functionality of an actor. It is defined as a set of actions performed by a system, which yields an observable result. An ellipse containing its name, it is placed inside the system boundary and connected to an actor with an association. This shows how the use cases and the actor interact.

To find out the use cases, ask the following questions to each of the actors.

* Which functions does the actor require from the system? What does the actor need to do?
* Does the actor need to read, create, destroy, modify or store some kind of information in the system?
* Could the actor’s daily work be simplified or made more efficient by adding new functions to the system?

**2.5.2.1 USE CASES**

**Use case for the actor Administrator**

|  |  |  |
| --- | --- | --- |
| **1** | Which functions does the Administrator require from the system? What does the Administrator need to do? | Administrator requires the following functionalities from the system such as bus registration, bus blocking and unblocking, Add bus stops, view bus owner and approval, view complaints, view ratings and reviews. |
| **2** | Does the Administrator need to read, create, destroy, modify or store some kind of information in the system? | Yes. Administrator need to create, view and edit the data if require. |
| **3** | Could the Administrator work be simplified by adding new functions to the system? | Yes, the system can reduce his/her work. |

Above questions give the following use cases for the actor Administrator.

* Login
* Driver registration
* View driver details
* View user
* Approve user
* Bus Blocking
* Bus Unblocking
* View bus details
* Add bus stops
* Add bus routes
* View bus stops
* View bus routes
* View complaints
* View Ratings
* View feedback and Reviews
* Logout

**Use case for the actor Owner**

|  |  |  |
| --- | --- | --- |
| **1** | Which functions does the Owner require from the system? What does the Owner need to do? | Owner requires the following functionalities from the system such as registration, view bus, set imei number, track bus, pass emergency notes to driver. |
| **2** | Does the Owner need to read, create, destroy, modify or store some kind of information in the system? | Yes. Owner need to create, view and edit the data if require. |
| **3** | Could the Owner work be simplified by adding new functions to the system? | Yes, the system can reduce his/her work. |

Above questions give the following use cases for the actor Owner

* Login
* Add bus
* View bus
* View routes
* View stops
* Allocate bus
* Set imei number
* Track bus
* Pass emergency notes to driver
* Logout

**Use case for the actor Bus driver**

|  |  |  |
| --- | --- | --- |
| **1** | Which functions does the Bus driver require from the system? What does the Owner need to do? | Bus driver requires the following functionalities from the system such as registration, location updating, generate voice notes, trip status updates. |
| **2** | Does the Bus driver need to read, create, destroy, modify or store some kind of information in the system? | Yes. Bus driver need to create, view and edit the data if require. |
| **3** | Could the Bus driver work be simplified by adding new functions to the system? | Yes, the system can reduce his/her work. |

Above questions give the following use cases for the actor Bus driver

* Login
* Update location
* Generate voice notes
* Update trip status
* Logout

**Use case for the actor Passenger**

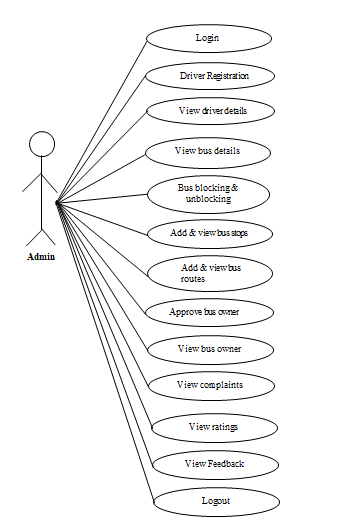
|  |  |  |
| --- | --- | --- |
| **1** | Which functions does the Passenger require from the system? What does the Owner need to do? | Passenger requires the following functionalities from the system such as bus search, track nearby buses, view bus timing, complaint registration, send reviews, rating bus. |
| **2** | Does the Passenger need to read, create, destroy, modify or store some kind of information in the system? | Yes. Passenger need to create, view and edit the data if require. |
| **3** | Could the Passenger work be simplified by adding new functions to the system? | Yes, the system can reduce his/her work. |

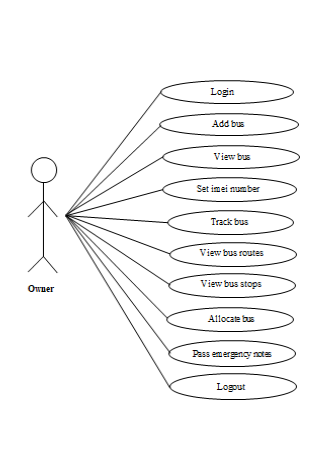
Above questions give the following use cases for the actor Passenger

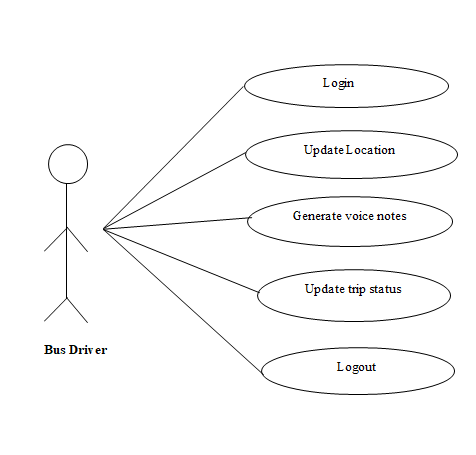
* Login
* Search bus
* Track bus
* View bus timing
* View bus stops
* View bus routes
* Register complaints
* Send feedback & reviews
* Rating bus
* Logout

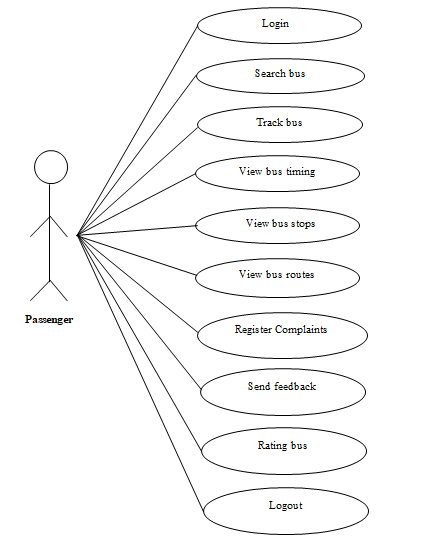
**2.5.2.2 USE CASE DIAGRAM**

**Use case diagrams** are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more **external users** of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system. An ellipse containing its name inside the ellipse or below it represents it. The use case diagram of this project has shown below. The actors of this project are admin, owner, bus driver, passenger.









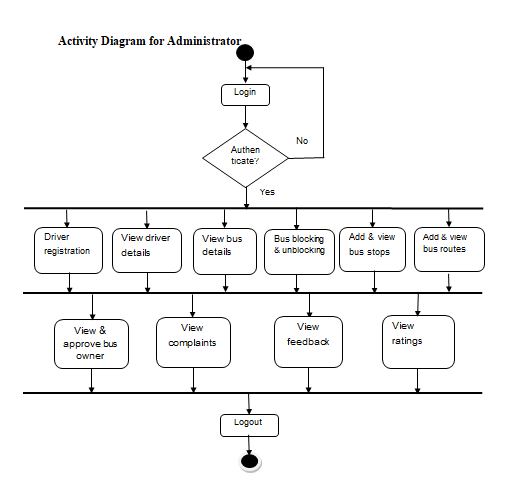
**2.5.3 ACTIVITY DIAGRAM**

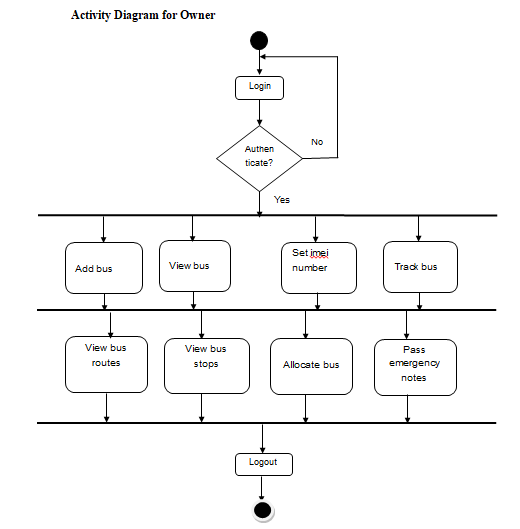
The activity diagram supplements the use case by providing a graphical representation of the flow of interaction within a specific scenario. It uses rounded rectangles to imply a specific system function, arrows to represent flow through the system, decision diamonds to depict a branching decision, and solid horizontal lines to indicate that parallel activities are occurring.

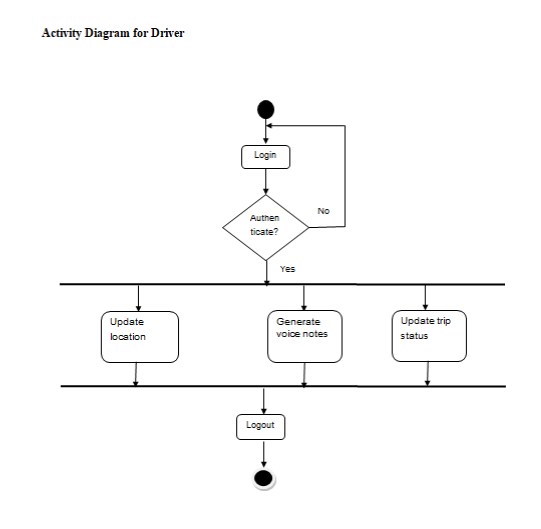
The basic purposes of activity diagrams are similar to other diagrams. It captures the dynamic behavior of the system. Other diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

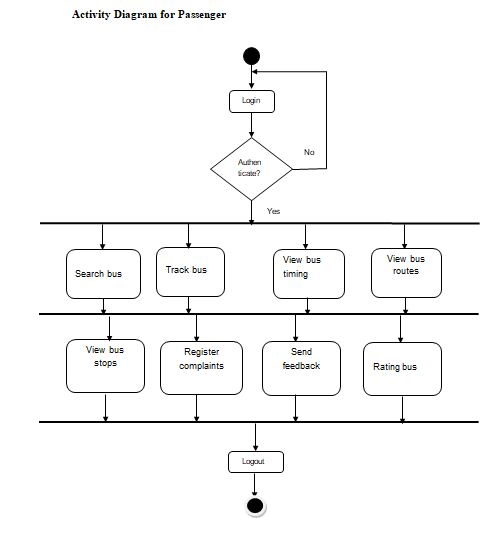
So, the purposes can be described as:

* Draw the activity flow of a system.
* Describe the sequence from one activity to another.
* Describe the parallel, branched and concurrent flow of the system.

****

****

****

****

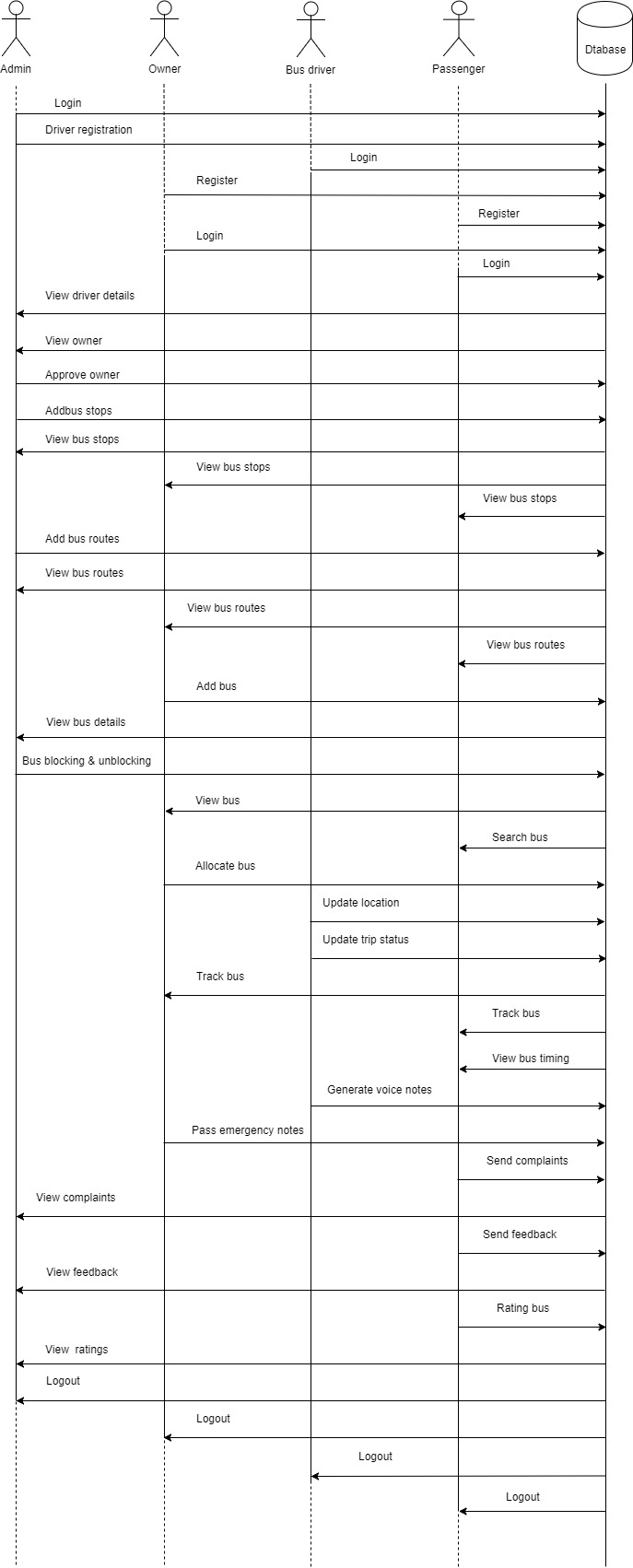
**2.5.4 SEQUENCE DIAGRAM**

Sequence diagrams are an easy and intuitive way of describing the behaviour of a system by viewing the interaction between the system and its environment. A sequence diagram shows an interaction arranged in a time sequence. It shows the objects participating in the interaction by their life lines and the messages they exchange, arranged in a time sequence.

A sequence diagram has two dimensions: a vertical dimension represents time, horizontal dimension represents different objects. The vertical line is called the object’s lifeline. The lifeline represents the object’s existence during the interaction. This form was first popularized by Jacobson. An object is shown as a box at top of a dashed vertical line. A role is slot for an object within a collaboration that describes the type of object that may play the role and its relationships to other roles. However, a sequence diagram does not show the relationships among the roles or the association among the objects. An object role is shown as a vertical dashed line, the life line.

Each message is represented by an arrow between the life lines of two objects. The order in which these messages occur shown top to bottom on the page. Each message is labeled with the message name. The label also can include the argument and some control information and show self-delegation, a message that an object sends to itself, by sending the message arrow back to the same lifeline. The horizontal ordering of the lifelines is arbitrary. Often, all arrows are arranged to proceed in one direction across the page, but this is not always possible and the order conveys no information.

The sequence diagram is very simple and has immediate visual appeal- this is its greatest strength. A sequence diagram is an alternative way to understand the overall flow of the control of a program. Instead of looking at the code and trying to find out the overall sequence of behavior, we can use the sequence diagram to quickly understand that sequence.

****

**2.6. SYSTEM REQUIREMENTS**

**2.6.1. HARDWARE AND SOFTWARE REQUIREMENTS**

Hardware and software requirements for the installation and smooth functioning of this product could be configured based on the requirements needed by the component of the operating environment that works as front-end system here we suggest minimum configuration for the both hardware and software components.

Working off with this application is requirements concrete on system environments. It includes two phases.

* Hardware Requirements
* Software Requirements

**External Interfaces Requirements**

This will use the standard input/output devices for a personal computer. This includes the following.

* Keyboard
* Mouse
* Monitor
* Mobile Phone/Tablet

**Hardware Interface**

SERVER:

1. Processor : AMD or higher
2. Memory : 2 GB or more
3. Hard disc : 10 GB or more
4. CLIENT:
5. Processor : Any processor
6. Memory : 1GB or more
7. Hard disc : 40 GB or more
8. An android Phone

**Software Interface**

* Operating System : Windows 10
* Front End : Android Java
* Back End : Firebase Real time database
* Software Used : Android Studio
* Web Browser :Chrome

**CHAPTER-3**

**SYSTEM DESIGN**

**3. SYSTEM DESIGN**

Design is a meaningful engineering representation of something that is to be built. It is an iterative process through which requirements are translated in to a blueprint for constructing the software. The goal of the design phase is to plan a solution of the problem specified by the requirements document.

Major activities during the design phase are:

* Data Base Design
* Architectural Design
* Interface Design
* Modular Design

**3.1. DATABASE DESIGN**

A database is collections of inter related data stored with minimum redundancy to serve many users quickly and efficiently. In database design data independence, accuracy, privacy, and security are given higher priority. Database design is an integrated approach to file design. This activity deals with the design of the physical database. All entries and attributes have been identified while creating the database. The database design deals with the grouping of data into number of tables so as to reduce the duplication of data, minimize storage space, and retrieve the data efficiently.

Guidelines for designing a database:

* Design a relational schema so that it is easy to explain its meaning. Do not combine attributed from multiple entity and relationships typeintosingle relation.
* Design the database schema so that no insertion, deletion or modification anomalies are present in the relation.
* As far as possible, avoid placing attributes in a base relation whose values may frequently be null.
* Design relation schemas so that they can be joined with equality conditions on attributes that are either primary keys or foreign keys in away that no spurious tuples are generated.

**Advantage**

* Ease of use
* Data independence
* Accuracy and integrity
* Avoiding inordinate delays
* Recovery from failure
* Privacy and security.

**3.1.1 E-R DIAGRAM**

An entity-relationship diagram is a data modeling technique that creates a graphical representation of the entities, and relationship between entities, within an information system.

**There are three basic elements in ER models:**

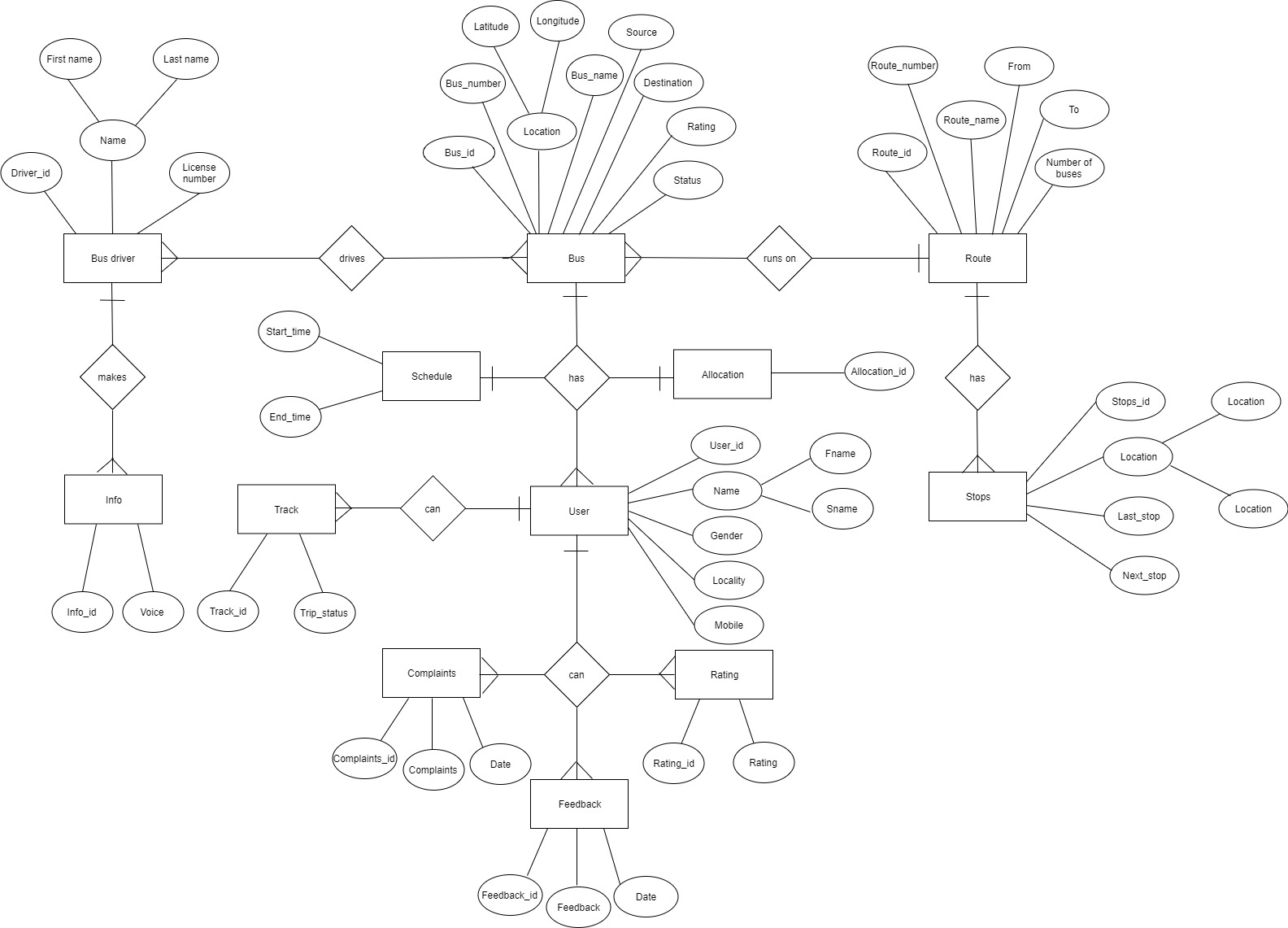
* **Entities** are the “things” about which we seek information
* **Attributes** are the data we collect about entities.
* **Relationships** provided the structure needed to draw information from multiple entities.

**E-R Diagram Symbols:**

Entity

Attributes

Relation

****

**3.1.2 TABLE DESIGN**

Table 1:Login\_tbl

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| Login\_id | Int | Primary key | Login id |
| Username | Char(10 | Not null | Name of user |
| Password | Char(10) | Not null | Password of user |
| User\_type | char(10) | Not null | Type of user |

Table 2:User\_tbl

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| User\_id | Int | Primary key | User id |
| Login\_id | Int | Foreign key | Login id |
| Fname | Varchar(10) | Not null | First name of user |
| Sname | Varchar(10) | Not null | Second name of user |
| Gender | Varchar(8) | Not null | Gender of user |
| DOB | Date | Not null | Date of birth of user |
| Age | Int | Not null | Age of user |
| Place | Varchar(10) | Not null | Place of user |
| Mobile\_no | Int | Not null | Mobile no of user |

Table 3:Bus\_tbl

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| Bus\_id | Int | Primary key | Bus id |
| User\_id | Int | Foreign key | User id |
| Bus\_number | Varchar(10) | Not null | Bus number |
| Bus\_name | Varchar(15) | Not null | Bus name |
| Image | Varchar(max) | Not null | Image of bus |
| Latitude | Int | Not null | Latitude value |
| Longitude | Int | Not null | Longitude value |
| Status | Varchar(10) | Not null | Status of bus |

Table 4:Driver\_tbl

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| Driver\_id | Int | Primary key | Driver id |
| Login\_id | Int | Foreign key | Login id |
| Bus\_id | Int | Foreign key | Bus id |
| Driver\_name | Varchar(20) | Not null | Driver name |
| License\_number | Int | Not null | License number |

Table 5:Route\_tbl

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| Route\_id | int | Primary key | Route id |
| Route\_number | int | Not null | Route number |
| Route\_name | Char(15) | Not null | Route name |
| From | Char(15) | Not null | Start place |
| To | Char(15) | Not null | End place |

Table 6:Stops\_tbl

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| Stops\_id | Int | Primary key | Stops id |
| Bus\_Type | Varchar(15) | Not null | Bus Type |
| Location | Char(10) | Not null | Location |
| Latitude | Int | Not null | Latitude value |
| Longitude | Int | Not null | Longitude value |

Table 7:RunningStatus\_tbl

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| Status\_id | Int | Primary key | Status id |
| Bus\_id | Int | Foreign key | Bus id |
| Latitude | Varchar(20) | Not null | Latitude |
| Longitude | Varchar(20) | Not null | Longitude |
| Last\_stop | Varchar(20) | Not null | Last stop |
| Next\_stop | Varchar(20) | Not null | Next Stop |
| Time | Time | Not null | Time |
| NextTime | Time | Not null | Time |

Table 8:Rating\_tbl

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| Rating\_id | Int | Primary key | Rating id |
| Bus\_id | Int | Foreign key | Bus id |
| User\_id | Int | Foreign key | User id |
| Ratings | Float | Not null | Ratings |

Table 9:Allocate\_tbl

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| Allocate\_id | Int | Primary key | Allocate id |
| Route\_id | Int | Foreign key | Route id |
| Stops\_id | Int | Foreign key | Stop id |

Table 10:Schedule\_tbl

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| Schedule\_id | Int | Primary key | Schedule id |
| Bus\_id | Int | Foreign key | Bus id |
| Route\_id | Int | Foreign key | Route id |
| Start\_time | Int | Not null | Start time |
| End\_time | Int | Not null | End time |

Table 11:Info\_tbl

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| Info\_id | Int | Primary key | Feedback id |
| Bus\_id | Int | Foreign key | Bus id |
| Voice | Varchar(20) | Not null | Voice |
| Date | Date | Not null | Date |
| Time | Time | Not null | Time |

Table 12:Feedback\_tbl

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| Feedback\_id | Int | Primary key | Feedback id |
| User\_id | Int | Foreign key | User id |
| Bus\_id | Int | Foreign key | Bus id |
| Reviews | Char(20) | Not null | Reviews |
| Feedback | Char(20) | Not null | Feedback |
| Date | Date | Not null | Date |

Table 13:Complaints\_tbl

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Constraints** | **Description** |
| Complaints\_id | Int | Primary key | Complaints id |
| Bus\_id | Int | Foreign key | Bus id |
| User\_id | Int | Foreign key | User id |
| Complaints | Varchar(20) | Not null | Complaints |
| Reply | Varchar(20) | Not null | Reply |
| Date | Date | Not null | Date |

**3.2ARCHITECTURAL DESIGN**

The architectural design develops a modular program structure and represents the control relationships between modules. It also defines interfaces that enable data to flow throughout the program.

**3.2.1 DATA FLOW DIAGRAM**

A data flow diagram is a graphical technique that depicts data flow and transforms that are applied as data move from input to output. The DFD is used to represent increasing information flow and functional details. A Level 0 DFD also called a fundamental system model or context model represents the entire software elements as a single bubble with input and output indicated by incoming and outgoing arrows respectively. Additional process and information flow parts are represented in next level i.e., Level 1 DFD. Each of the processes represented at level 1 are sub functions of overall system depicted in the context model.

**Data flow diagram symbol:**

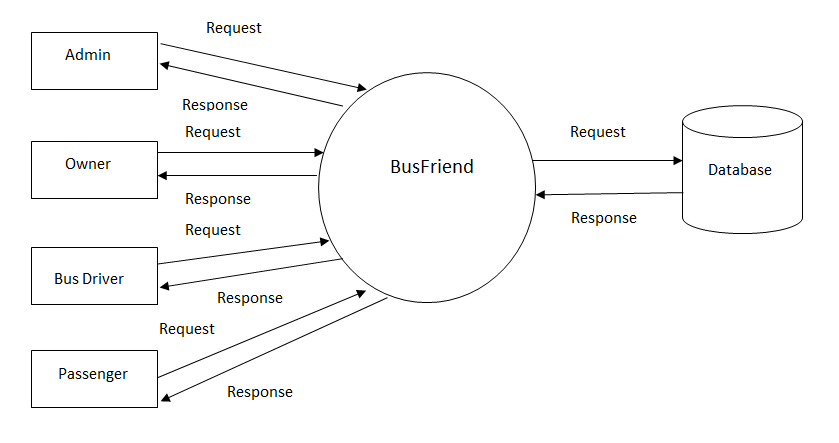
Source/Destination of Data

Data flow

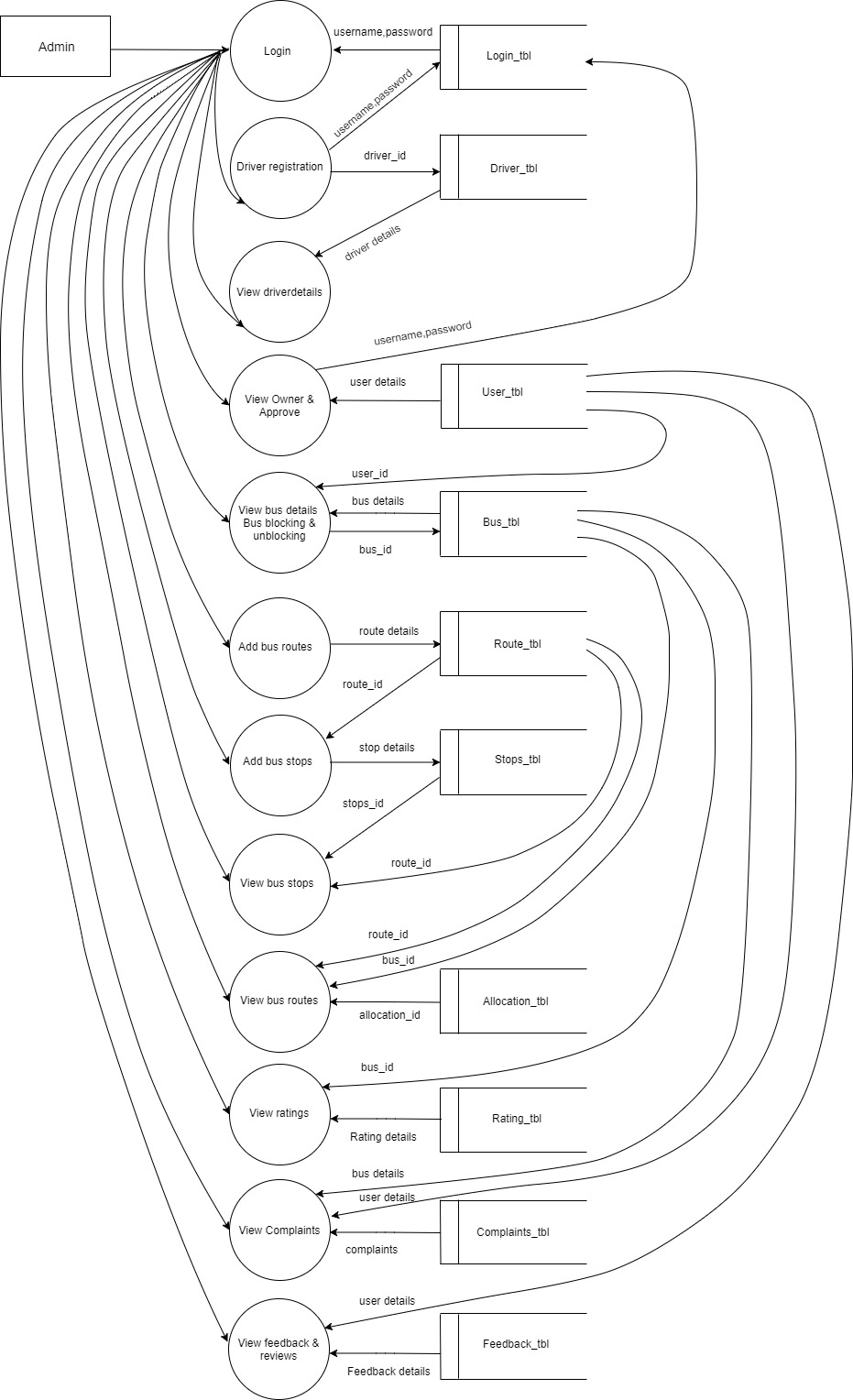
Process

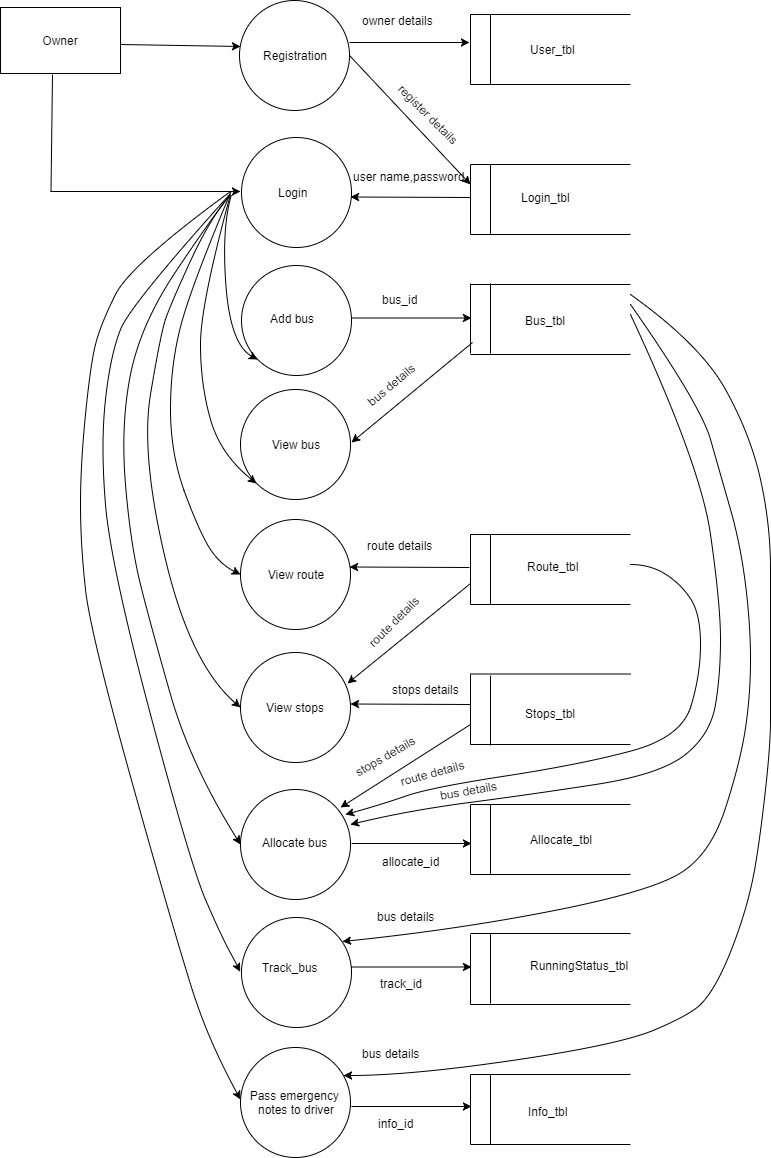
Storage

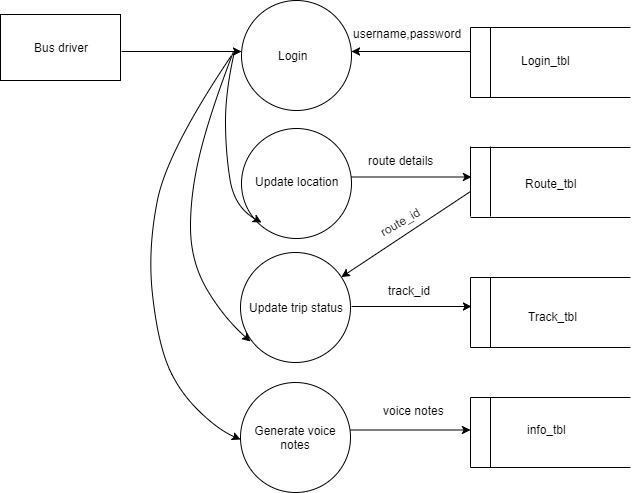
**Level 0: Context Level**

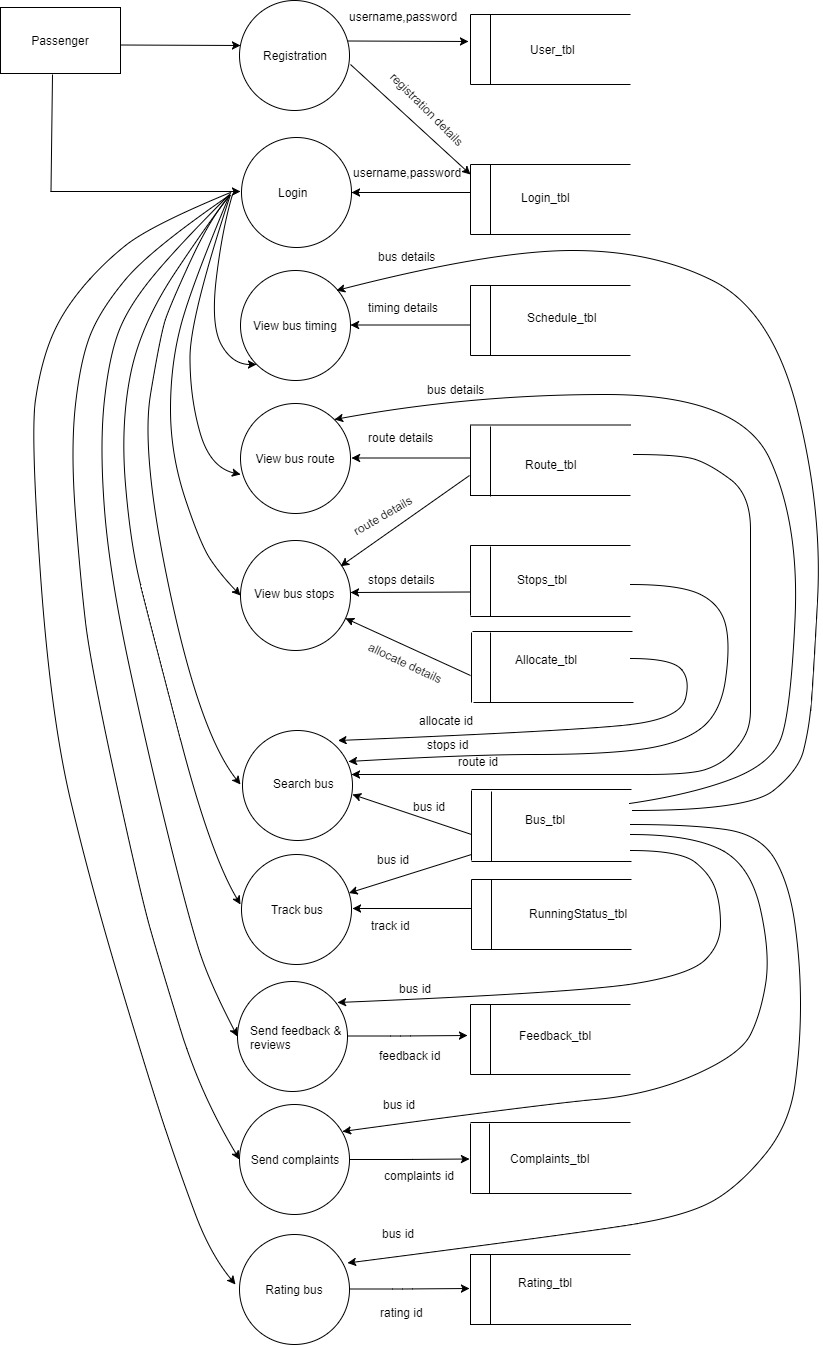


**Level 1**

****

****

****

****

**3.3. INTERFACE DESIGN**

An interface design elements for the software tell how information flows into and out of the system and how it is communicated among the components as part of the architecture.

**3.3.1 INPUT DESIGN**

Input design is the link between the information system and users and those steps that are necessary to put transaction data into a usable form for processing data entry. Instructing the computer to read data from a written printed document can active the activity of putting data into the computer for processing or it can occur by keying data directly into the system. The design of input focusing on controlling the errors, avoid delay, and keeping the process simple. System analyst decides the following input design details.

* What data to input?
* What medium to use?
* How the data is arranged and coded?

In my project named **“BusFriend”**, I tried to include the following design constrains provided in the software engineering.

**1: Avoid scattering of fields in the forms**

In all forms of the software the textboxes (which provided to input some data), label (which label the text boxes), spinner (list a set of values) etc. All are arranged in a neat and well format. It provides a simple look to the pages. The buttons are placed at the bottom of the page and easily accessible to the user. The menus are arranged below the heading and at a minimum level of menus are arranged with pages. Menu provides the continuity to the pages.

**2: User only needs to enter a minimum amount of data**

All forms contain a minimum amount data, but most essentials. No page provides or wanted bulky of data. It provides more easiness to the user. It creates more the software to the end user. Also the operation continues by single click.

**3: Avoid confusion in the forms**

All forms have a well-defined menus and each menu name indicate its purpose. So the user can easily access various forms without confusion. Each form and its sub forms are well labeled. So the user can easily identify the forms and work on that.

**The following are the input forms present in this project:**

* Login.
* Registration
* Add bus stops
* Add bus routes
* Approve owner
* Allocate bus
* Track bus
* Search bus
* Update location
* Update trip status
* Add ratings
* Register complaints
* Send feedback

**3.3.2 OUTPUT DESIGN**

Designing computer should proceed in well thought out manner. The term output means any information produced by the information system weather printed or displayed. Output design is a process that involves designing necessary output that have to be used by various users according to requirement. The efficient intelligent output design should remove the system relationship with the users and help in decision making.

When designing the output, system analyst must accomplish the following:

* Determine the information present
* Decide whether to print, display the information and select output medium
* Arrange information in acceptable format.

In my project, the outputs are well format and it provides the output in a correct and neat format.

**The following are the output forms present in this project:**

* View driver details
* View bus details
* View bus stops
* View bus routes
* View bus
* View owner
* View bus timing
* View ratings
* View feedback details
* View complaint details

**3.4 PROCEDURAL DESIGN**

The procedural design determines the modules included in the whole project which help us to identify the major functions.

**MODULE SPECIFICATIONS**

The following are the modules in this application.

* Admin
* Owner
* Bus driver
* Passenger

**1. Admin Module**

Admin is one who manages the entire working of a system. Following are the main functions that can be performed by Administrator:

* **Driver Registration:** Admin can register the bus drivers
* **View driver details:** Admin can view the driver details
* **View Owner:** Admin can view the owner details
* **Approve owner:** Admin can approve or reject the owner
* **View bus details:** Admin can view bus details
* **Bus blocking & unblocking:** Admin can block or unblock the bus
* **Add bus stops:** Admin can add bus stops
* **Add bus routes:** Admin can add bus routes
* **View bus stops:** Admin can view bus stops
* **View bus routes:** Admin can view bus routes
* **View complaints:** Admin can view the complaints that send by the passenger and give reply to the passenger
* **View feedback:** Admin can view the feedback about the buses
* **View ratings:** Admin can view the rating of buses

**2. Owner Module**

Following are the main functions that can be performed by Owner:

* **Add bus:** Owner can add their buses
* **View bus:** Owner can view their buses
* **View bus routes:** Owner can view the bus routes
* **View bus stops:** Owner can view the bus stops
* **Allocate bus:** Owner can allocate the bus
* **Set imei number:** Owner can set the imei number
* **Track bus:** Owner can track their buses on Google map
* **Pass emergency notes:** Owner can pass emergency notes to driver

**3. Bus driver Module**

Following are the main functions that can be performed by Bus driver:

* **Update location:** Bus driver can update the location of the bus
* **Generate voice notes:** Bus driver can generate voice notes and send to owner and passengers
* **Update trip status:** Bus driver can update the trip status of bus

**4. Passenger Module**

Following are the main functions that can be performed by Bus driver:

* **Search bus:** Passengers can search the buses for travelling
* **Track bus:** Passengers can track the location of buses
* **View bus timing:** Passengers can view the timing of buses
* **View bus stops:** Passengers can view the bus stops
* **View bus routes:** Passengers can view the bus routes
* **Register complaints:** Passengers can send complaints about the bus to the admin
* **Send feedback:** Passengers can send feedback about the bus
* **Rating bus:** Passengers can rate the buses

**CHAPTER-4**

**CODING**