ignou
THE PEOPLE'S
UNIVERSITY

Synopsis
on
Bus Management System

Submitted by

Nishant
Enrollment No.: 176011833
BCA

Under guidance of

Mrs. Shuruti Singhal (MCA)

INDIRA GANDHI NATIONAL OPEN UNIVERSITY
(2019-2020)

Index

Introduction

Busses are the most frequently available and most used vehicle for going from small distance trip to very long trip. There’re a no. of types of buses available in the market according to the private or public.

In our case we’re about to start a new bus company (Shivaji Roadways) this company don’t want to go traditionally for management of the system manually.

So that we’re here to make a project which will be manage all the works related to database management and other operation such as E-ticket booking, Fair checking, estimated time for reaching location, Selecting best route etc.

Bus Management System is a web based management system. This system will ensure that the management process of this company are smoothly done. This system will also help in decreasing error mostly cause by human mistake. This system will develop as one the solution to transfer from current manual practices to a more systematic computerization system. This system will be developed by using HTML, CSS, JSP, JavaScript, and MySQL and will about to be run on Apache Tomcat 9.0 server on Windows 10.

Briefly there are three main modules for this system. The first module will be The Passenger Management System, the second module will be The Staff Management System and last one is for Company.

Objective

The main objective of the Bus Management System is to manage the details of the following:

1. Details for Passengers

- **Distance** to inform Passengers how far are they for their destination (Distance between current location to destination).
- **Fair** to inform them about cost for ticket for both non A/C busses **(₹5/ ₹10/ ₹15)** and A/C busses **(₹10/ ₹15/ ₹20/ ₹25).**
- **Time** to inform them about estimated time for reaching destination from where he/she is.
- **Route** to suggest passengers best route for reaching a particular location with having less traffic jam and less taken time to complete out the journey.
- **E-Ticket** to provide facility for buy a ticket online.

2. Details for Staff

- **Salary** information regarding the salary of the employee such as salary package amount, and whole table about previews received salary.
- **Leave** Staff such as Driver, Conductor, Ticket checker and other peoples can online apply for a leave.
- **E-Report** they can report online if bus is being faulty.
- **Request for Transfer / Promotion** they can request for a transfer if their family is shifting somewhere else and also they can request for promotion if they think the sum of salary given by company is not negligible in respect of workload.
- **Fair** they can too know fairs (In such cases in arguing with passengers).

3. Details for Company(Admin)

- **Updating Fairs** bus company have right to increase or decrees fairs so that we’ll prove facility for do that.
- **Updating Routes** company will be able to maintain flow of busses to maximize profit by avoiding broken roads and heavy traffic jams and other issues.
- **Issuing more busses** if company purchases some new buses or get them on rent for temporary or permanent bases, we’ll provide a portal for updating information regarding new buses for maintain manageability.
- **Introducing New Route** company can introduce new routes, let’s take a real world scenario to know the importance of this. When the Signature bridge was under construction and the alter one (bridge of Wazirabad) was also under maintenance so that was the point when Dehli Transport Corporation (DTC) was introduce new route for 971 via Kashmiri gate. To implement it efficiently we’ll provide this module.
- **Staff Payment** this will store all the details of payments for staff this will also visible in for staff.
- **Policies of Company** This section will provide all the details about company policies to avoid misconception of staff duty and liabilities and other things.

Project Category

RDBMS

RDBMS stands for "Relational Database Management System." An RDBMS is a DBMS designed specifically for relational databases. Therefore, RDBMSes are subset of DBMSes.

A relational database refers to a database that stores data in a structured format, using rows and columns. This makes it easy to locate and access specific values within the database. It is "relational" because the values within each table are related to each other. Tables may also be related to other tables. The relational structure makes it possible to run queries across multiple tables at once.

While a relational database describes the type of database a RDBMS manages, the RDBMS refers to the database program itself. It is the software that executes queries on the data, including adding, updating, and searching for values. A RDBMS may also provide a visual representation of the data. For example, it may display data in a tables like a spreadsheet, allowing you to view and even edit individual values in the table. Some RDBMS programs allow you to create forms that can streamline entering, editing and deleting data.

Most well-known DBMS applications fall into the RDBMS category. Examples include Oracle Database, MySQL, Microsoft SQL Server, and IBM DB2. Some of these programs are primarily used for relational database management.

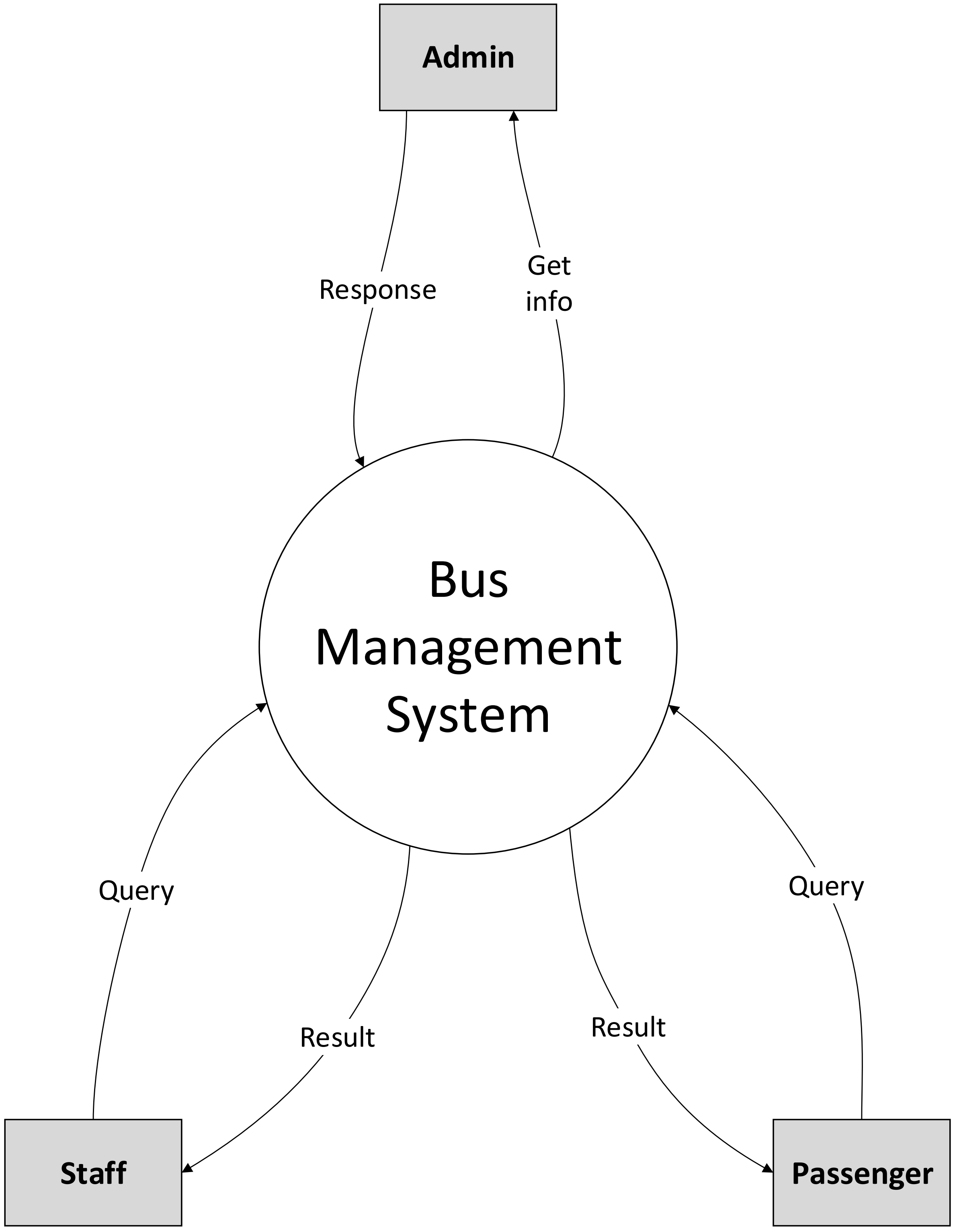
Example of non-relational databases include Apache Hbase, IBM Domino, and Oracle NoSQL Database.

These type of database are managed by other DBMS programs that support NoSQL, which do not fall into the RDMBS category.

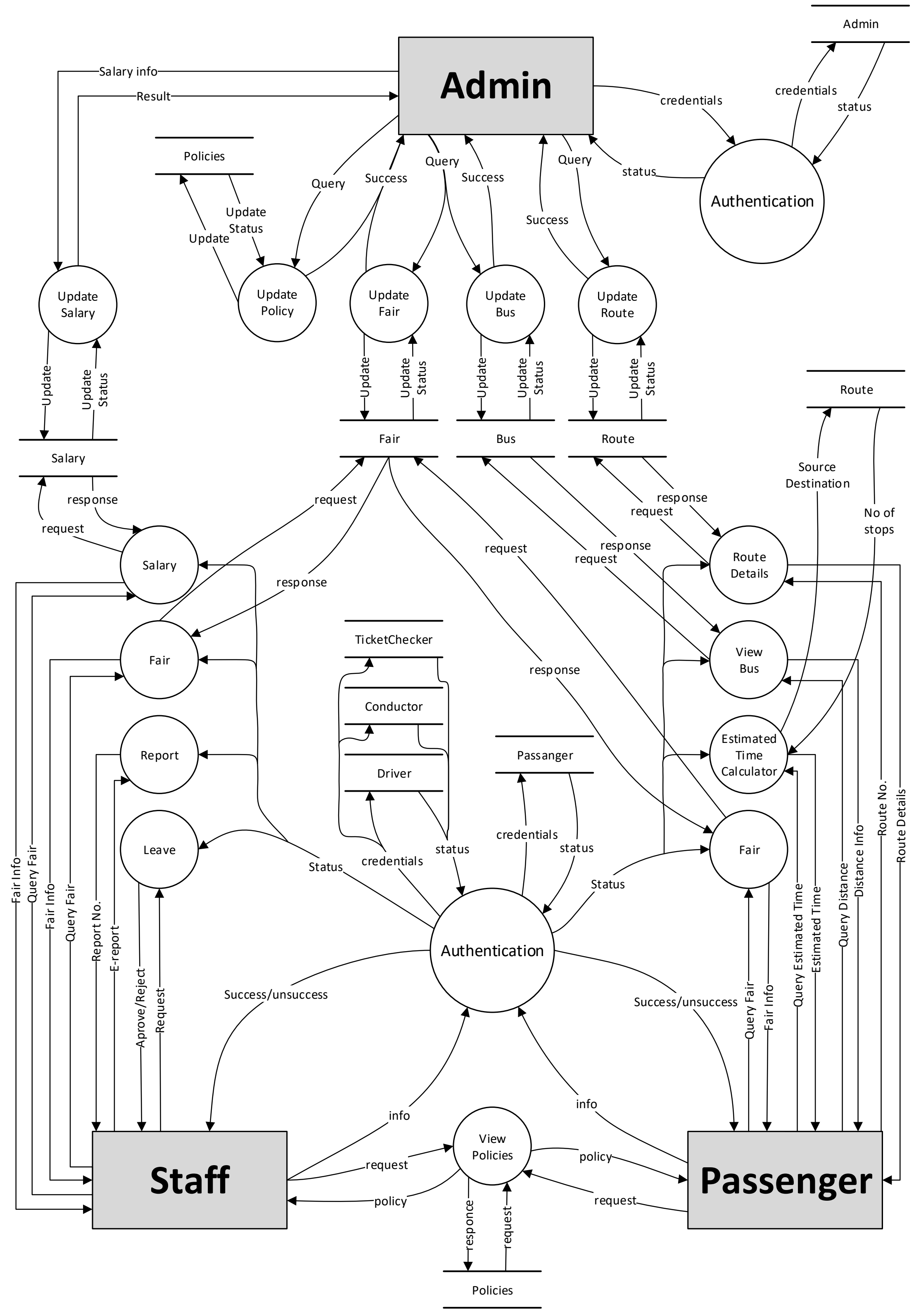
In our case we're about to use MySQL for RDBMS.

Data Flow Diagrams

Level 0 DFD

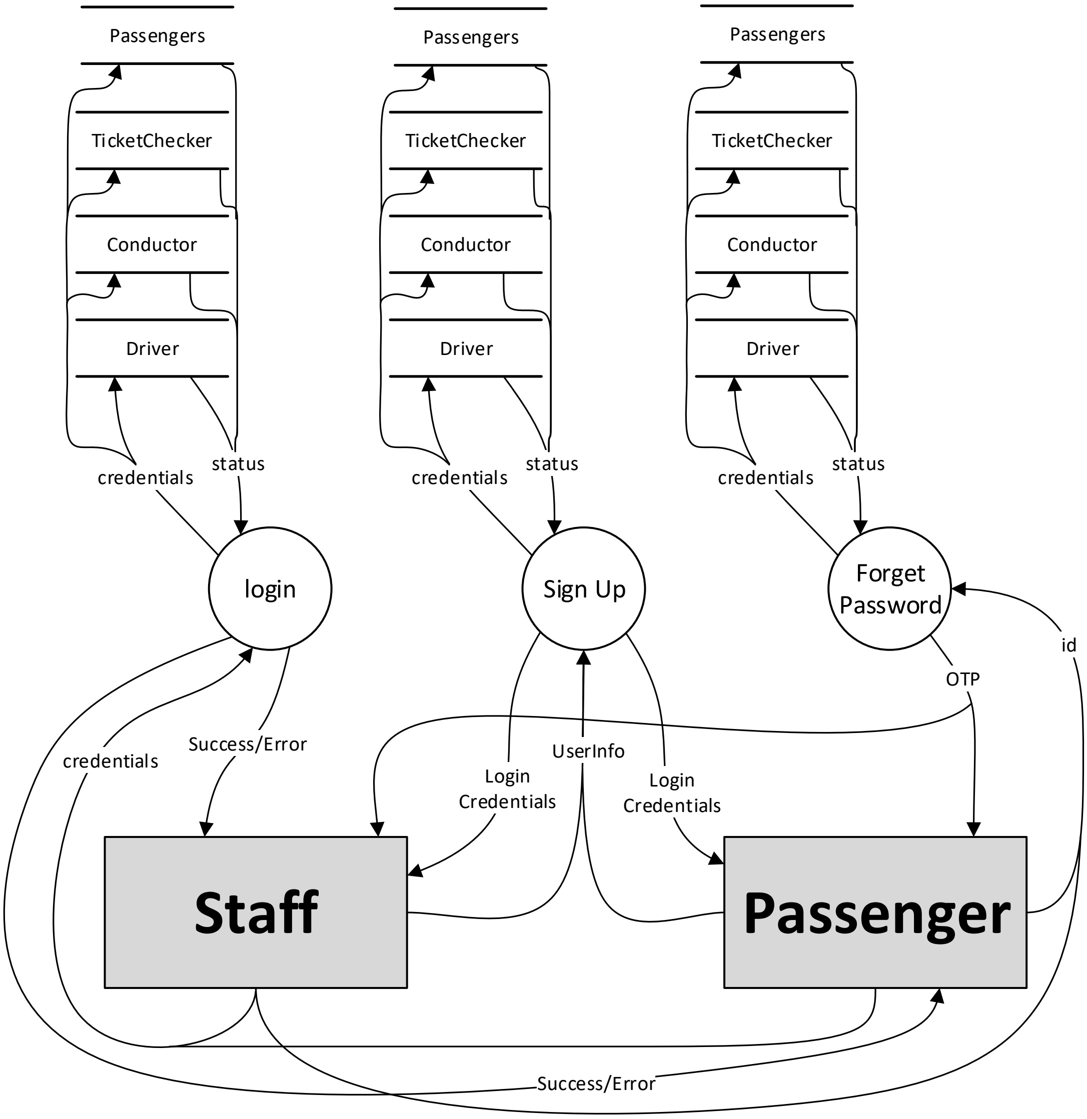


Level 1 DFD

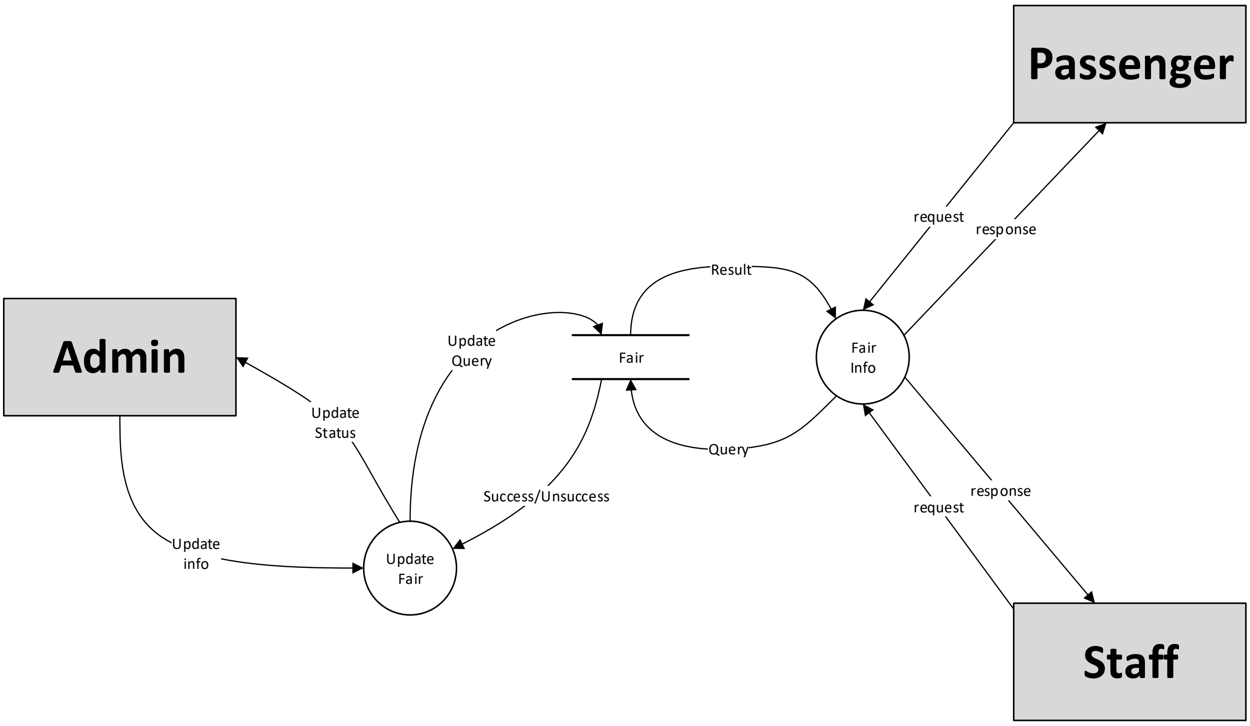

Admin
Admin
credentials
credentials
status
status
Authentication
Salary info
Result
Policies
Query
Success
Query
Success
Query
Success
Update
Update Status
Update Salary
Update Policy
Update Fair
Update Bus
Update Route
Update
Update Status
Update
Update Status
Update
Update Status
Update
Update Status
Salary
Fair
Bus
Route
Route
Source Destination
No of stops
response
request
request
response
request
response
request
response
request
response
request
response
Salary
Fair
Report
Leave
Route Details
View Bus
Estimated Time Calculator
Fair
TicketChecker
Conductor
Driver
Passanger
credentials
status
credentials
status
Status
Status
Authentication
Success/unsuccess
Success/unsuccess
info
info
Fair Info
Query Fair
Fair Info
Query Fair
Report No.
E-report
Aprove/Reject
Request
Query Fair
Fair Info
Query Estimated Time
Estimated Time
Query Distance
Distance Info
Route No.
Route Details
Staff
Passenger
View Policies
request
policy
policy
request
responce
request
Policies

Level 2 DFD

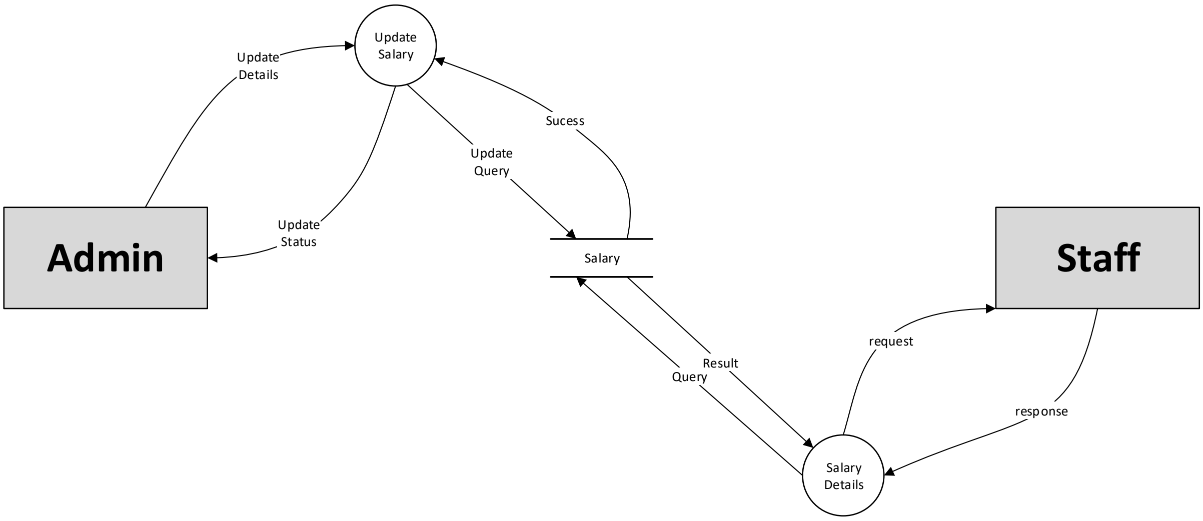
2.1 Authentication



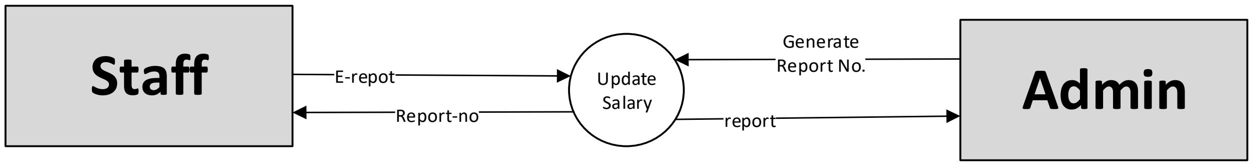
2.2 Fair



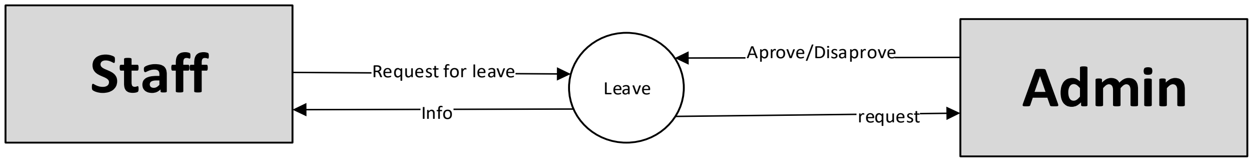
2.3 Salary



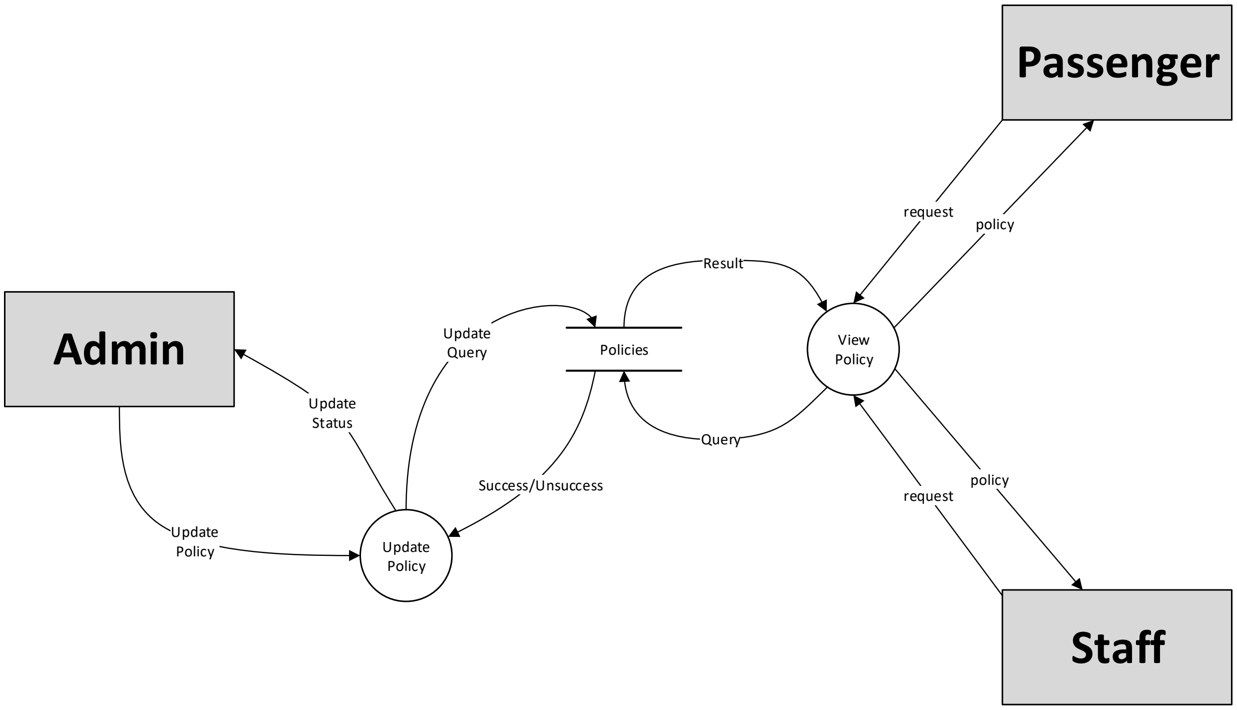
2.4 Report



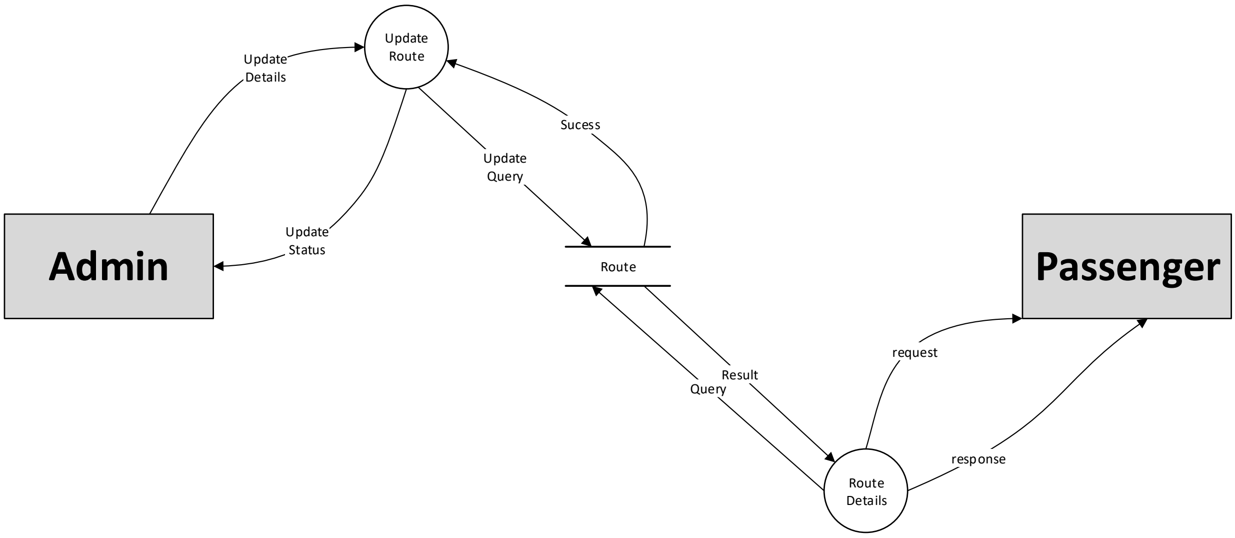
2.5 Leave



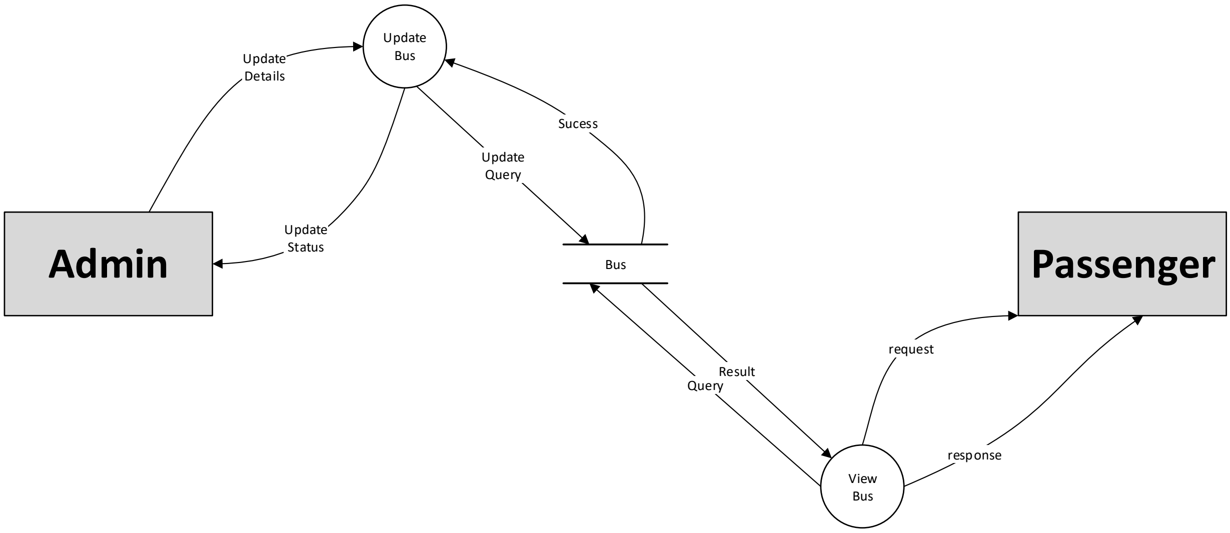
2.6 Policies



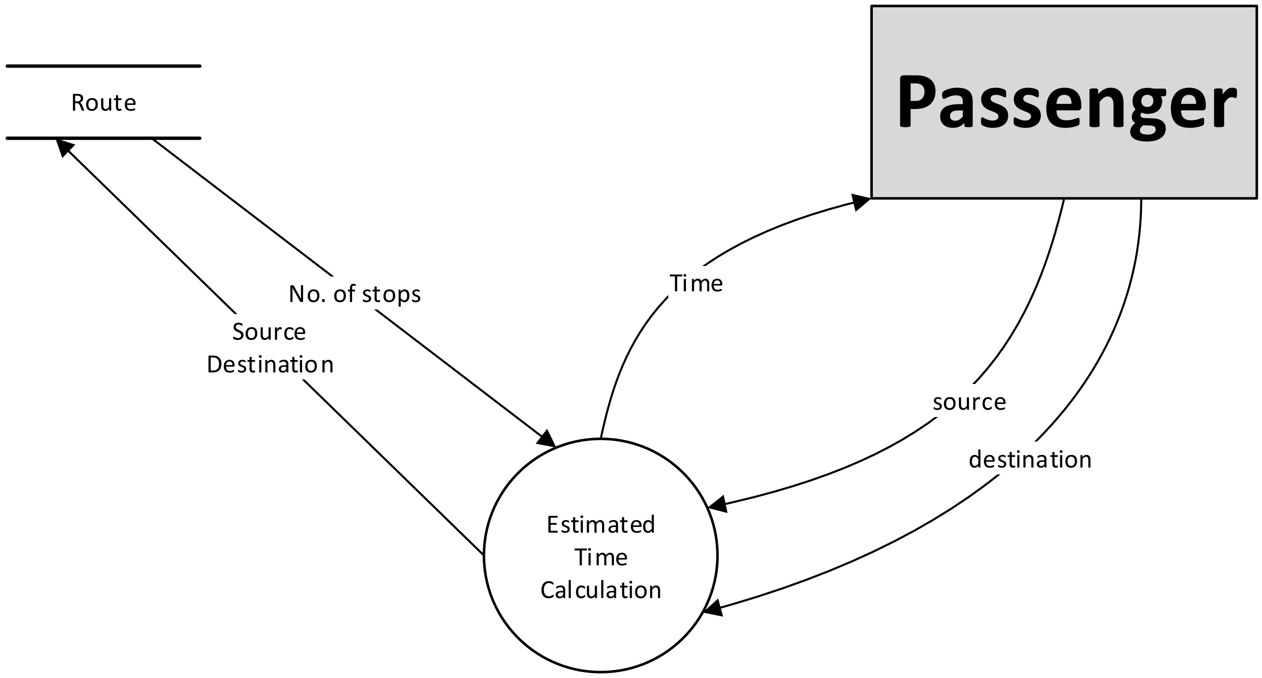
2.7 Route



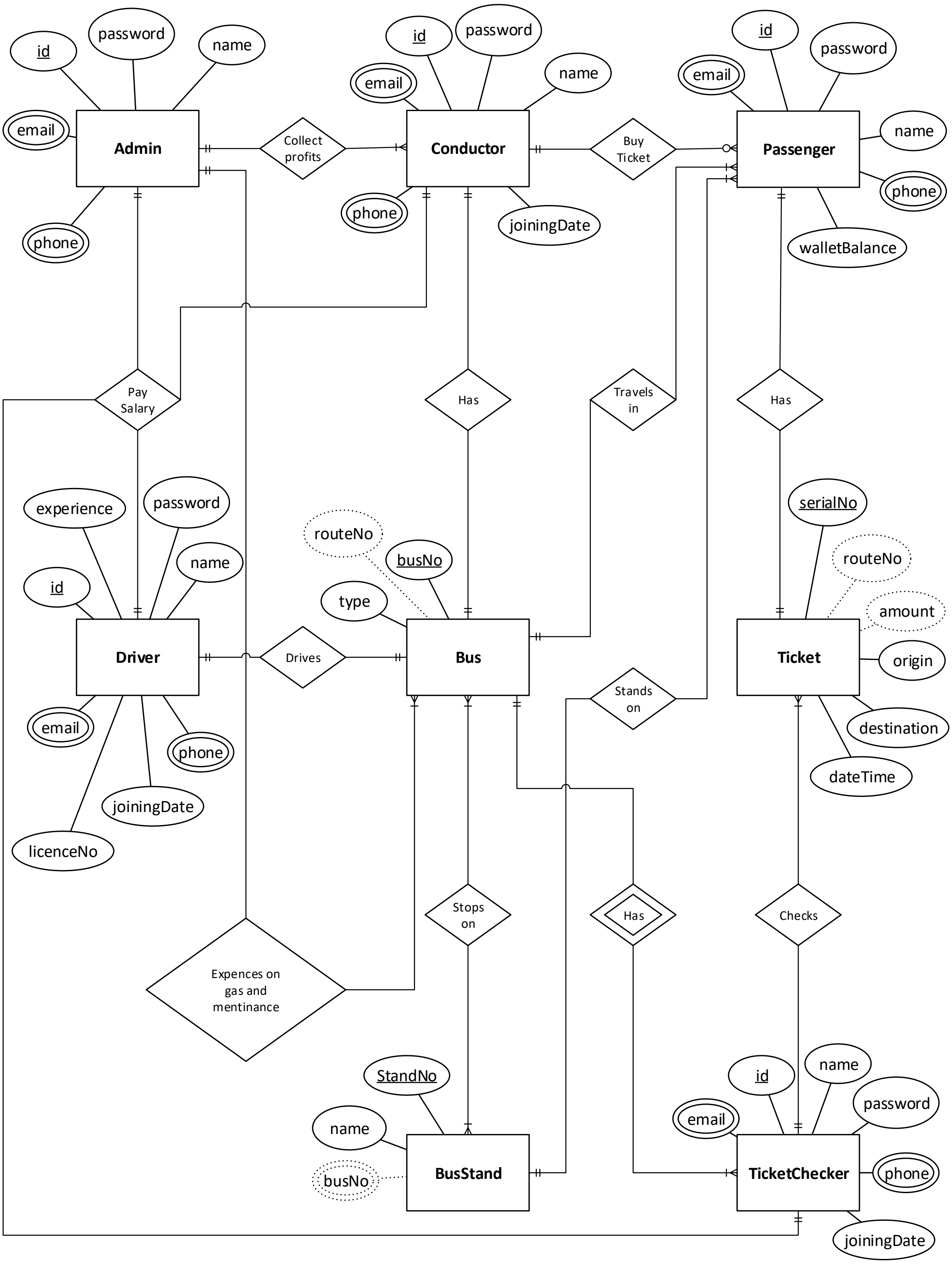
2.8 Bus Details



2.9 Estimated Time Calculation



Entity Relationship Diagram



Class Diagram

Admin
-id: int
-password: String
-name: String
-phone: int[1..3]
-email: String[1..3]
+setId(int): void
+getId(): int
+setPassword(String): void
+getPassword(): String
+setName(String): void
+getName(): String
+addPhone(int): void
+removePhone(int): void
+getPhoneNumbers(): int[]
+addEmail(String): void
+removeEmail(String): void
+getEmails(): String[]

Ticket
-serialNo: int
-routeNo: String
-amount: int
-dateTime: String
-origin: String
-destination: String
+setSerialNo(int): void
+getSerialNo(): int
+setRouteNo(String): void
+getRouteNo(): String
+setDate(): void
+getDate(): String
+setTime(): void
+getTime: void
+setOrigin(String): void
+getOrigin(): String
+setDestination(String): void
+getDestination(): String

BusStand
-standNo: int
-name: String
-busNo: String [1..*]
+setStandNo(int): void
+getStandNo(): int
+setName(String): void
+getName(): String
+addBus(String): void
+removeBus(String): void
+getBusNumbers(): String[]

Bus
-busNo: String
-routeNo: String
-type: Boolean
+setBusNo(String): void
+getBusNo(): String
+setRouteNo(String): void
+getRouteNo(): String
+setType(Boolean): void
+getType(): Boolean

Driver
-id: int
-password: String
-name: String -joiningDate: String -experience: int
-phone: int[1..3]
-email: String[1..3]
-licenceNo: String
+setId(int): void
+getId(): int
+setPassword(String): void
+getPassword(): String
+setName(String): void
+setJoiningDate(String): void
+getJoiningDate(): String
+setExperience(int): void
+getExperience(): int
+getName(): String
+addPhone(int): void
+removePhone(int): void
+getPhoneNumbers(): int[]
+addEmail(String): void
+removeEmail(String): void
+getEmails(): String[]
+setLicenceNo(String):void
+getLicenceNo():String

Conductor
-id: int
-password: String
-name: String
-joiningDate: String
-phone: int[1..3]
-email: String[1..3]
+setId(int): void
+getId(): int
+setPassword(String): void
+getPassword(): String
+setName(String): void
+setJoiningDate(String): void
+getJoiningDate(): String
+getName(): String
+addPhone(int): void
+removePhone(int): void
+getPhoneNumbers(): int[]
+addEmail(String): void
+removeEmail(String): void
+getEmails(): String[]

TicketChecker
-id: int
-password: String
-name: String -joiningDate: String
-phone: int[1..3]
-email: String[1..3]
+setId(int): void
+getId(): int
+setPassword(String): void
+getPassword(): String
+setName(String): void
+getName(): String
+setJoiningDate(String): void
+getJoiningDate(): String
+addPhone(int): void
+removePhone(int): void
+getPhoneNumbers(): int[]
+addEmail(String): void
+removeEmail(String): void
+getEmails(): String[]

Passenger
-id: int
-password: String
-name: String
-walletBalance: float
-phone: int[1..3]
-email: String[1..3]
+setId(int): void
+getId(): int
+setPassword(String): void
+getPassword(): String
+setName(String): void
+getName(): String
+updateBal(float): void
+getBal(): float
+addPhone(int): void
+removePhone(int): void
+getPhoneNumbers(): int[]
+addEmail(String): void
+removeEmail(String): void
+getEmails(): String[]

Database Design

Admin

Field	Type	Null	Key	Default	Extra
id	int(4)	NO	PRI	NULL	auto_incriment
password	varchar(20)	NO		NULL	
name	varchar(20)	NO		NULL	
email	varchar(30)	NO	UNI	NULL	
phone	varchar(10)	YES		NULL	

Conductor

Field	Type	Null	Key	Default	Extra
id	int(4)	NO	PRI	NULL	auto_incriment
password	varchar(20)	NO		NULL	
name	varchar(20)	NO		NULL	
email	varchar(30)	NO	UNI	NULL	
phone	varchar(10)	YES		NULL	
joiningDate	date	NO		NULL	

Passenger

Field	Type	Null	Key	Default	Extra
id	int(4)	NO	PRI	NULL	auto_incriment
password	varchar(20)	NO		NULL	
name	varchar(20)	NO		NULL	
email	varchar(30)	NO	UNI	NULL	
phone	varchar(10)	YES		NULL	
walletBalance	float(7,2)	NO		0.00	

Driver

Field	Type	Null	Key	Default	Extra
id	int(4)	NO	PRI	NULL	auto_incriment
password	varchar(20)	NO		NULL	
name	varchar(20)	NO		NULL	
email	varchar(30)	NO	UNI	NULL	
phone	varchar(10)	YES		NULL	
joiningDate	date	NO		NULL	
experience	int(2)	YES		NULL	
licenceNo	varchar(10)	NO	UNI	NULL	

Bus

Field	Type	Null	Key	Default	Extra
busNo	varchar(6)	NO	PRI	NULL	
routeNo	int(5)	NO	MUL	NULL	
type	varchar(6)	NO		NULL	

Ticket

Field	Type	Null	Key	Default	Extra
serialNo	int(7)	NO	PRI	NULL	auto_incriment
routeNo	int(5)	NO	MUL	NULL	
amount	int(2)	NO	MUL	NULL	
dateTime	datetime	NO		CURRENT_ TIMESTAMP	DEFAULT_ GENERATED
destination	varchar(20)	NO		NULL	
origin	varchar(20)	NO		NULL	

BusStand

Field	Type	Null	Key	Default	Extra
standNo	int(4)	NO	PRI	NULL	auto_increment
name	varchar(20)	NO		NULL	
busNo	varchar(6)	NO		NULL	

TicketChecker

Field	Type	Null	Key	Default	Extra
id	int(4)	NO	PRI	NULL	auto_incriment
password	varchar(20)	NO		NULL	
name	varchar(20)	NO		NULL	
email	varchar(30)	NO	UNI	NULL	
phone	varchar(10)	YES		NULL	
joiningDate	date	NO		NULL	

Route

Field	Type	Null	Key	Default	Extra
routeNo	varchar(6)	NO	PRI	NULL	auto_incriment
source	int(5)	NO	MUL	NULL	
destination	int(5)	NO		NULL	

Fair

Field	Type	Null	Key	Default	Extra
amount	int(2)	NO	PRI	NULL	
distance	int(2)	NO		NULL	

Salary

Field	Type	Null	Key	Default	Extra
tranId	int(4)	NO	PRI	NULL	auto_incriment
userId	int(4)	NO	MUL	NULL	
date	date	NO		NULL	

Policy

Field	Type	Null	Key	Default	Extra
pNo	int(4)	NO	PRI	NULL	auto_incriment
type	varchar(10)	NO		NULL	
discription	varchar(999)	NO		NULL	

Module Description

Authentication

Registration:

User information have to be compelled to be registered within the system thus on establish every of them unambiguously and do the required group acts as the real potential.

On the far side, this plenty of things require measure there wherever we will reference him. Without registration, there are few options and pages one user can see which are landing on the home page and taking the features read but he won't be allowed to use those.

For use, he will have to register. One person needs to put his all the details correctly and precisely as it will be helpful in identifying them and believing that he is the real person who has booked for the same.

It also includes driver license for them who is driving and parameter too.

Log in:

After registration one will register within the system because of the operator of the system either on behalf of the user. When this he has the different helpful interfaces accessible for any action.

Here either bride or groom both have to log in with their unique identity and passwords. After this, they will be directed to the primary user interface from where they have further options.

Forgot password:

This is quite often that people tend to forget the password they keep for login. So this could be very tedious and hectic to recover the password manually in case if one needs to log in in the emergency.

So to overcome this problem we have this module named as forgot password and using this module user can recover their password in seconds. So here we need only to put our registered email Id and hit the enter.

Then one confirmation email will go to the email where he may reset the password. In seconds one can use this module and get rid of forgetting password problem.

Fair

This module is for mange all the functions related to fair such as Update and view.

Update Fair:

As all of us knows increasing in public spending, hoarding, tax reductions, price rise in international markets are the **cause of inflation.** These factors leads to rising prices. Also, increase demands cause higher prices which leads to inflation. So may be in future we have to increase the Fairs of busses or may be reduce for implanting this functionality we have fair update module.

View Fair:

This sub-module is for showing the information related to Fairs for passengers and sometimes this is for Staff to in condition of arguing with the passengers.

Salary

A **Salary** is a form of payment from an employer to an employee, which may be specified in an employment contract. It is contracted with piece wages, where each job, hour or other unit is paid separately, rather than on a periodic basis. From the point of view of running a business, salary can also be viewed as the cost of acquiring and retaining human resources for running operations, and is then termed personnel expense or salary expense. In our module staff can see the info regarding salary and salary can be updated through Admin.

Report

A report is a document that presents information in an organized format a specific audience and purpose. Although summaries of reports may be delivered orally, complete reports are almost always in the form of written documents.

Staff can report in case of **bus is being faulty** and can get report no. which is generated through Admin

Leave

A leave of absence in an extended period of time off from your job. Depending on the organization, you may simply be able to ask for time off from work. Or, there may be a formal process you need to follow approved for a leave of absence. In our Bus Management System Staff can request for a leave online and can get approval.

Policies

Policies are needed because they set a general plan of action used to guide desired outcomes and is a fundamental guideline to help make decisions. The purpose of healthcare procedures it to communicate to employees the desired outcomes of the organization. We implement both sub modules for update and view policy.

Route

A route is a way for travel or movement, the path from source location to the Destination location. The route module will give all the information to the passengers and routes can be modified through admin.

Bus Details

This Module is to provide users information about bus from where this information maintained by the admin of the System

Estimated Time Calculation

This module as per name suggests is for calculation the time for being journey complete which internally works on distance between locations and the flow of traffic

Hardware Requirements

Operating System:	Windows 7/8/8.1/10
Hard Disks:	40GB
RAM:	256GB
GPS:	Needed
Graphic Card:	Decent
NIC:	Needed

Software Requirements

- **Java 1.8** (JDK+JRE)
- **HTML, CSS and JavaScript**
- **MySQL 8.0**
- **Apache Tomcat 9.0**
- **GitBash 2.24**
- **JDBC Connectors**
- **Google Chrome (Browser)**
- **Visual Studios Code (IDE)**

Is this project for Client/Industry?

No

This Project is not for any Industry or Client. This project is solely made for **Final Year Project** of BCA IGNOU**.** I'm assuming here the Company **(Assumed Shivaji Roadways)** for which I'm making this **Bus Management System** is a new company in industry so that they do not tell me about how the Software will work and they also don't know which are the modules those they need for managing their company efficiently.

I'm responsible for making this web project successful, and if in future any company will exist like Shivaji Roadways and they want's a web application like my project so absolutely I would like to sale them my project.

Future scope and enhancement

The Project has covered almost all the requirements. Further requirements and improvements can easily be done since the code mainly structured and modular in nature.

The project has a very vast scope in future. The project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion.

The following are the future scope for the project:

- **Maps:** We can add module for maps to know traffic, bus stand location and availability of busses on that particular location where he/she is.
- **Bar code Reader:** We can add module for reading barcode for instantly validate an issued ticket or pass is valid or not this will very helpful for ticket checker and another reason behind this module implementation may be ticket checking need at least 5 minutes which may be very useful for someone.
- **GPS:** We can add a module by some hardware implementation in this module we need to insert a GPS on each and every bus to inform passenger how far is the nearest bus, so that is case of urgency passengers can go for Metro service or any other such as auto rickshaw, gramin sewa etc.
- **Maintenance on bus:**
  We can add this module to store information regarding maintenance on buses this will store, Bus No., Date of Service, Type of repairing, cost on repairing and status of bus.

Conclusion

Finally, we have a system where a user who books the bus ticket giving the details and doing the necessary payment option. He gets a token with all detail he boards on the bus on given time.

The conductor who is staff will check the ticket and punch the conformation owner check the bills in the end. The user drops the destination. Hence completes the task of the system