

Chapter 1

Introduction

1. Project Overview

We aim to become a pioneer in the vehicle rental industry by completely focusing on customers, our employees, growth, innovation and efficiency. All of these elements will drive us towards success and show us as one company that can perform and give value for money.

2. Product Description

When it comes to cab rental services, Cool Service is the most trusted and reliable name in the travel business. The most advanced travel agents offering cab rental and car hire in India, making full use of information technology to improve the level of our efficiency. However, this is only one aspect of services. And this project continually strive to offer the best of services - both in terms of man and machine, to our clients

Moreover, this project has a fleet of cars ranging from luxury to budget cabs. While, it offers online cab hire service for corporate houses. And this project claim to offer the best of rates, which are tailor-made depending upon the facilities, availed and offer both intercity and intra-city cab facilities. All cabs have proper permits and documentation so that the clients couldn't be hassled for the lack of documents. However, this project has strategic backup system for any eventuality. Cab drivers are educated, polite, and reliable and are trained to handle acute breakdowns. The cab service includes all categories of cars from luxury to budget.

Further, this project's utmost priority is quality. To achieve this, vehicles are well maintained and tested for delivering optimum and uninterrupted performance. Team of professionals in the travel business enables this system to design trips that suits to all budgets and preferences of the travelers. In addition, workforce including drivers and administrative staff are well trained to discharge their duties with a lot of efficiency.

Modules of the Projects:

- ☐ Admin module
- ☐ HR module
- ☐ Maintenance module
- ☐ Movement module
- ☐ Finance module
- ☐ Quality Assurance module

Chapter 2

PROBLEMS AND SOLUTIONS OF THE PROJECT

2.1 Existing System

Cool cab Service is an innovative thought to simplify the Transportation problems of Employees of an organization. In the present System, Organization do maintain a person for the allocating and proper functioning of transportation .The Person appointed needs to look after the assigning and movement of cabs.Authorised person maintains the transportation details in papers, which is a tedious task if any updations or changes need to be done.

- ☐ Details are stored in Papers.
- ☐ Maintenance is a huge problem.
- ☐ Updation, changes in details is a tedious task.
- ☐ Performance is not achieved up to the requirements.

2.2 Proposed System

In the Previous System,Details are Stored Manually in papers,to share the details between employees was a Financial drawback. Updations in the details is a tedious task.

But a new system was proposed to overcome the above drawbacks.

Functionalities and advantages of proposed system are:

- ☐ Data is Centralized which has overcome the Sharing problem in previous system.
- ☐ As data is Maintained electronically, it's easy for a person to update the details, which has overcome the tedious updation in previous system.
- ☐ Maintenance is easy and performance is good.
- ☐ Mainly the system has automated the Transportation Process.

Chapter 3

Feasibility Report

3. Feasibility Report

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- ✓ Technical Feasibility
- ✓ Operation Feasibility
- ✓ Economical Feasibility

3.1. Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- ✓ Does the necessary technology exist to do what is suggested?
- ✓ Do the proposed equipments have the technical capacity to hold the data required to use the new system?
- ✓ Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- ✓ Can the system be upgraded if developed?
- ✓ Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web

based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified.

3.2. Operational Feasibility

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

- ✓ Is there sufficient support for the management from the users?
- ✓ Will the system be used and work properly if it is being developed and implemented?
- ✓ Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

3.3. Economic Feasibility

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

Chapter 4

System Analysis

Software Requirement Specification

Overview

We aim to become a pioneer in the vehicle rental industry by completely focusing on customers, our employees, growth, innovation and efficiency. All of these elements will drive us towards success and show us as one company that can perform and give value for money. This Service will make A manager to think of whether the transportation is expensive and maintaining quality or not.

1. Admin Module

Admin is the Super user of the system, he is responsible for the creation and maintenance of the accounts to the system. Admin is responsible for the creation of different kind of managers. Admin looks after the maintenance of these accounts. He has a feature of getting the password of a username ..

Tbl_AdminLogin

- AdminID
- UserName
- Password
- EmailID
- Department

Tbl_Manger

- MangerID
- EmpName
- Address
- Qualification
- DOB
- Gender

- PhoneNo
- EmailID
- Designation
- Department
- DOJ
- Age

Functionalities

- ✓ Associaton between Admin and Manager tables
- ✓ Admin Creates the Accounts for different types of manager.
- ✓ Admin Logins into the System
- ✓ Admin Can view the login details and maintain these Details
- ✓ Admin can get the password to the Username.

Queries

- ✓ What is the Password of Username?
- ✓ What is the username of logged in User?
- ✓ What is the number of Users in the System?
- ✓ What is the Username of the Admin?

Alerts

- ✓ Username Already Exists
- ✓ New User account is successfully created
- ✓ Invalid username or password
- ✓ All fields Are Mandatory

2. HR Manager

In Real world, Hr manager is responsible for the human resources of employees in an organization. As Transportation is also an facilities provided to an employee, HR is responsible for providing the transportation by cab .Here HR is Mainly used to register the employees for the cab facility ,after registering employees,he is the person to make shifts and Batches .Totally the functionalities of HR manager is to Maintain the Employee, Shift details and Batch scheduling.

Tbl_BatchDetails

- BID
- BatchID
- TotalNoOfEmployees
- ShiftID

Tbl_ShiftTimeing

- SID
- ShiftID
- ShiftName
- StartingTime
- DispatchTime
- NoBatches

Tbl_EmployeeDetails

- EID
- EmpID
- EmpName
- PAddress
- CAddress
- Qualification
- DOB

- VehicleRequire
- Gender
- PhoneNo
- Designation
- Department
- DOJ
- Status
- Age
- Time Span
- Image Path

Tbl_ShiftSchedule

- SSID
- ShiftScheduleID
- EmpID
- Department
- BatchID
- EmpName
- ShiftID
- Routed

Functionalities

- ✓ Association Employee and shift schedule
- ✓ Association between shiftschedule and batch
- ✓ Association between batch and Shift timing
- ✓ Association between shiftschedule and shift timings
- ✓ Adding employees and providing ids to the employees
- ✓ Assigning shifts to the Employees.
- ✓ Assigning batches for the Shifts and employees.

Queries

- ✓ What is the Employee id of an Employee?
- ✓ What is the total number of Employees?
- ✓ What is the Batch id of an empid?
- ✓ What is the total number of employees in a batch.?

Alerts

- ☐ Username already exists
- ☐ Invalid username and password.
- ☐ Re-enter password not matching
- ☐ All fields are mandatory

3. Maintenance Manager

Maintenance manager is one of the user in the system, main functionalities of this manager is to add vehicles, drivers and vendors of the vehicles. Maintenance manager is responsible for maintenance of these details and making the spare parts billing to the vehicles. The Maintenance manager is responsible for the Spare parts billing, registering the vehicles, drivers to the vehicles, mostly the vendor details and date of purchase of vehicles. These details provide Manager whether the cabs are providing profit or loss.

Tbl_DriverDetails

- DVID
- DriverID
- Name
- Address
- PhoneNo
- DOB

- DOJ
- Experience
- LicenceNo
- ImagePath
- NoOfAccident

Tbl_VehicleDetails

- VHID
- VehicleID
- Name
- VenderID
- DriverID
- VehicleType
- RegistorNo
- RateKm
- Capacity
- Routed
- ImagePath

Tbl_VenderDetails

- VID
- VenderID
- VenderName
- Address
- PhoneNo
- EmailID
- Remarks
- ImagePath

Tbl_SparePartBilling

- BillNo
- VehicleID
- SpareType
- Quantity
- BillDate
- SparePart
- Price
- TotalAmount

Tbl_SparePartsDetails

- SPID
- SparerPartID
- DealerName
- SparePartType
- Quantity
- SparePart
- DateOfPurchase
- Price
- AmountPaid

Functions

- ✓ Association of drive details between vehicle details
- ✓ Association between vehicle details and vendor details

- ✓ Association between Sparepart billing and vehicle details
- ✓ Adding driver to a vehicle
- ✓ Maintaining the vendor details with vehicle details

Queries

- ✓ What is the name of a driver to a vehicle
- ✓ What is the number of drivers registered with company.
- ✓ What is the vendor name of an vehicle
- ✓ How many vehicles are from an vendor

Alerts

- ☐ DriverName Should be Character
- ☐ MobileNo Digits Only
- ☐ No Of Accident Should be Numbers
- ☐ Licence No Should be Character or Numbers
- ☐ FirstName Should be Character

Reports

- ☐ What are the Vehicles present in company
- ☐ What are the details of drivers

4. Movement Manager

Movement Manager is one kind of user in the system who is responsible for the creation of Driver shift details, route details. He is responsible for the Vehicle allocation and maintenance of the tripsheets of the Cabs. Here Movement Manager has facility to search shift details and

the Route details.Movement manager is not only to create but also responsible for the maintenance of the driver shift details,route details ,trip sheets and Vehicle allocated details.

Tbl_DriverShiftDetails

- DSID
- DriverShiftID
- Name
- DriverID
- ShiftID
- ShiftDate
- Shifting

Tbl_RouteDetails

- RTID
- Routed
- RouteDescription
- Source
- Destination

Tbl_TripSheet

- TID
- TripSheetID
- AllocationID
- VehicleID

- RateKM
- KM
- TotalAmount
- Remark

Tbl_VehicleAllocationDetails

- VAID
- VehicleAllocationID
- VehicleID
- EmployeeID
- DriverID
- PickupDrop
- Routed
- VDate

Functionalities

- ✓ Association between DriverShiftDetails and driver details
- ✓ Association between DriverShiftDetails and shift timing
- ✓ Association between Route details and Vehicle allocation table
- ✓ Association between Tripsheet and Vehicle allocation table
- ✓ Association between tripsheet and vehicle details
- ✓ Maintaining the Drivers,shifts,routes and trip sheet details

Queries

- ✓ What is the trip id of a vehicle
- ✓ What is the speed of an vehicle
- ✓ What is the route of an vehicle
- ✓ What is the name of an driver id

Alerts

- ☐ DriverName Should be Character
- ☐ RouteDescription Should be Character
- ☐ Source Should be Character
- ☐ No Data Found
- ☐ All fields are mandatory

Reports

- ☐ What are the details of VehicleAllocated to a person
- ☐ What are the ShiftDetails of an driver
- ☐ What are the RouteDetails of a trip
- ☐ What are the vehicles in a TripSheet
- ☐ What are the details of VehicleAllocated

5. Finance Manager

Finance Manager is a type of user in the system, he is responsible for the cost estimation of the vendor, vehicle billing and viewing the feedbacks posted by the employees on the events as accidents and the driving nature of a particular cab. The Feedback posted by employers give the quality and performance of the cabs, by this result managers get a chance to improve the performance of the transportation to get profits.

Tbl_VehicleBillingTransction

- BID
- BillNo
- VehicleID

- Amount
- DateOfBilling
- VenderID
- Deduction
- NetAmount

Tbl_FeedBackFrom

- FBID
- FeedBackID
- EmpID
- VehicleID
- DriverID
- Remarks

Functionalities:

- ✓ Association between Vehicle billing and vehicle details
- ✓ Association between Vehicle billing and Vendor details
- ✓ Association between feedback and employee details
- ✓ Association between feedback and vehicles
- ✓ Maintaining the feedback and vehicle details

Queries

- ✓ What is the amount of an vehicle id?
- ✓ What is the vendor id of an vehicle?
- ✓ What is the feedback an vehicle?

Alerts

- ☐ Vendor required
- ☐ Deduction Digits Only
- ☐ All fields are mandatory

Reports

- ☐ What are the feedbacks to an vehicle
- ☐ What are the billing details of an vehicle.

6. Quality Assurance Manager

Quality Assurance Manager is a user in the System, He is responsible for the Maintaining the Quality in the transportation, to provide Quality he checks the performance by maintaining the feedbacks from employees and the accident details of the Cab. Here Quality Assurance Manager is responsible for inserting and maintaining the Accident details, by this details he requests for the finance manager to provide the Amount to the damage. Thus by maintaining all these details, Quality Assurance manager can give the Quality transportation facility to its employees.

Tbl_AccidentDetails

- ADID
- AccidentID
- VehicleID
- ADate
- ATime
- Remarks

Functionalities

- ✓ Association between vehicles and accident
- ✓ Maintaining the accident details

Queries

- ✓ What is the vehicle id of an accident had
- ✓ What is the vendor id of an vehicle

Alerts

- ☐ All fields are mandatory
- ☐ Reentered password doesn't match
- ☐ Invalid user and password

4.2) Hardware Requirements

- ✓ P4 2.8GB processor and above.
- ✓ Ram 512 MB and above.
- ✓ HDD 20 GB Hard Disk and above.

4.3) Software Requirements

- o Microsoft .Net framework 2.0

- o Microsoft ASP.Net, HTML
- o AJAX Tool kit.
- o Microsoft C#.Net language
- o Microsoft SqlServer 2000 and above.

Chapter 5

System Design

5.1. Module design:

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word “Quality”. Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer’s view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

5.2. DATA FLOW DIAGRAMS

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show

the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD'S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design.

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

DFD SYMBOLS:

In the DFD, there are four symbols

1. A square defines a source(originator) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows

3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data



Process that transforms data flow.



Source or Destination of data



Data flow



Data Store

CONSTRUCTING A DFD:

Several rules of thumb are used in drawing DFD'S:

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.

2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each word capitalized

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

SAILENT FEATURES OF DFD'S

1. The DFD shows flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
2. The DFD does not indicate the time factor involved in any process whether the dataflow take place daily, weekly, monthly or yearly.
3. The sequence of events is not brought out on the DFD.

TYPES OF DATA FLOW DIAGRAMS

1. Current Physical
2. Current Logical
3. New Logical
4. New Physical

CURRENT PHYSICAL:

In Current Physical DFD process label include the name of people or their positions or the names of computer systems that might provide some of the overall system-processing label includes an identification of the technology used to process the data. Similarly data flows and data stores are often labels with the names of the actual physical media on which data are stored such as file folders, computer files, business forms or computer tapes.

CURRENT LOGICAL:

The physical aspects at the system are removed as much as possible so that the current system is reduced to its essence to the data and the processors that transform them regardless of actual physical form.

NEW LOGICAL:

This is exactly like a current logical model if the user were completely happy with the user were completely happy with the functionality of the current system but had problems with how it was implemented typically through the new logical model will differ from current logical model while having additional functions, absolute function removal and inefficient flows recognized.

NEW PHYSICAL:

The new physical represents only the physical implementation of the new system.

RULES GOVERNING THE DFD'S

PROCESS

- 1) No process can have only outputs.
- 2) No process can have only inputs. If an object has only inputs than it must be a sink.
- 3) A process has a verb phrase label.

DATA STORE

- 1) Data cannot move directly from one data store to another data store, a process must move data.
- 2) Data cannot move directly from an outside source to a data store, a process, which receives, must move data from the source and place the data into data store
- 3) A data store has a noun phrase label.

SOURCE OR SINK

The origin and /or destination of data.

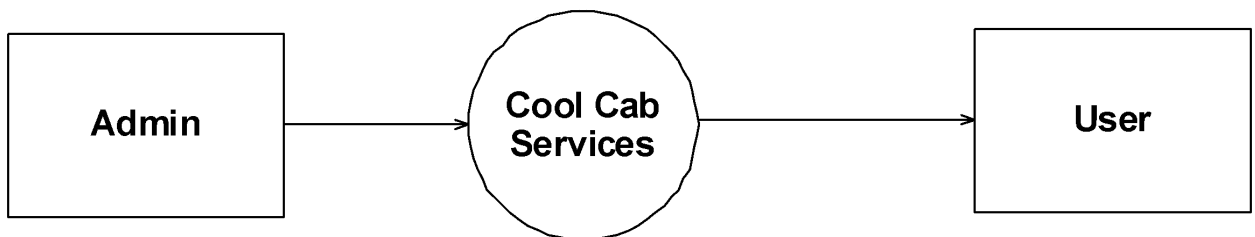
- 1) Data cannot move directly from a source to sink it must be moved by a process
- 2) A source and /or sink has a noun phrase label

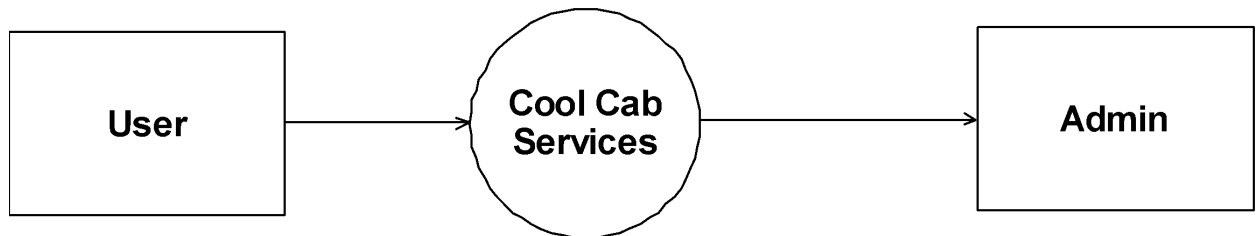
DATA FLOW

- 1) A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The later is usually indicated however by two separate arrows since these happen at different type.
- 2) A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.
- 3) A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow produce some other data flow returns the original data into the beginning process.
- 4) A Data flow to a data store means update (delete or change).
- 5) A data Flow from a data store means retrieve or use.

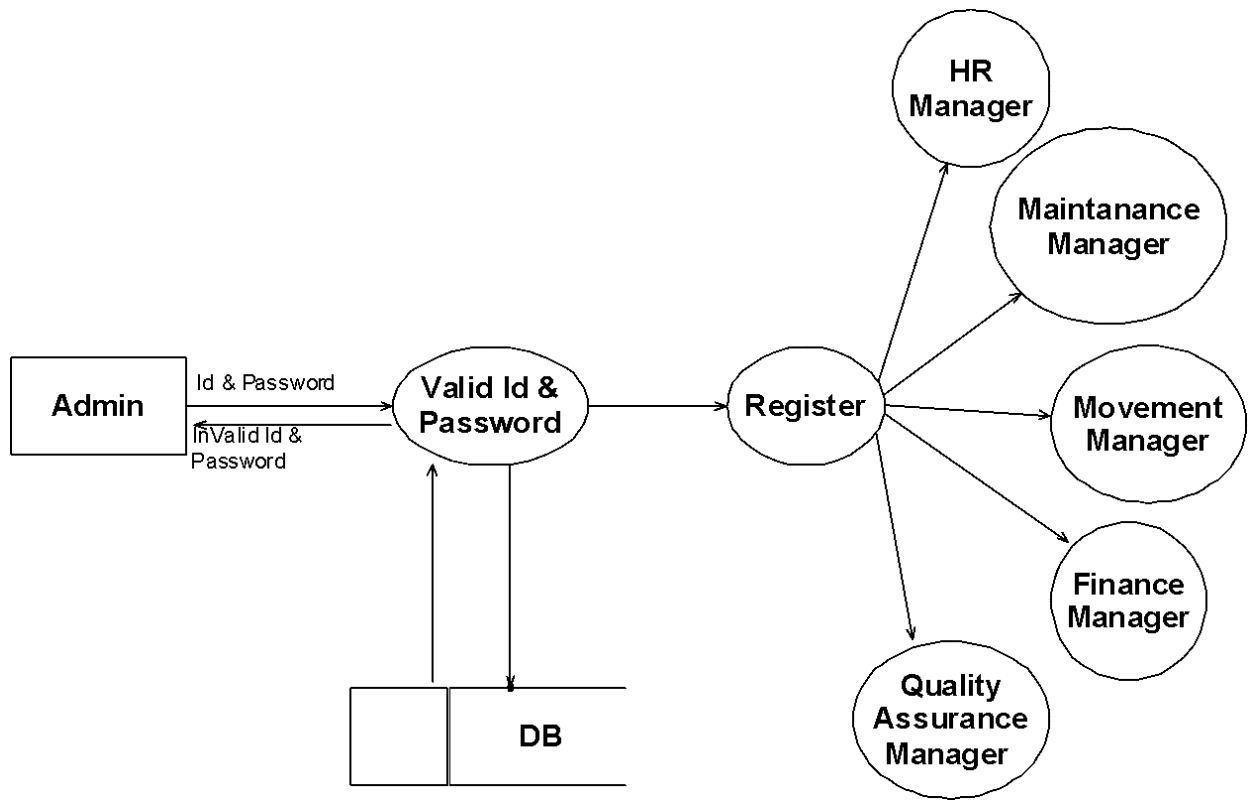
A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

Level 0 Diagram:

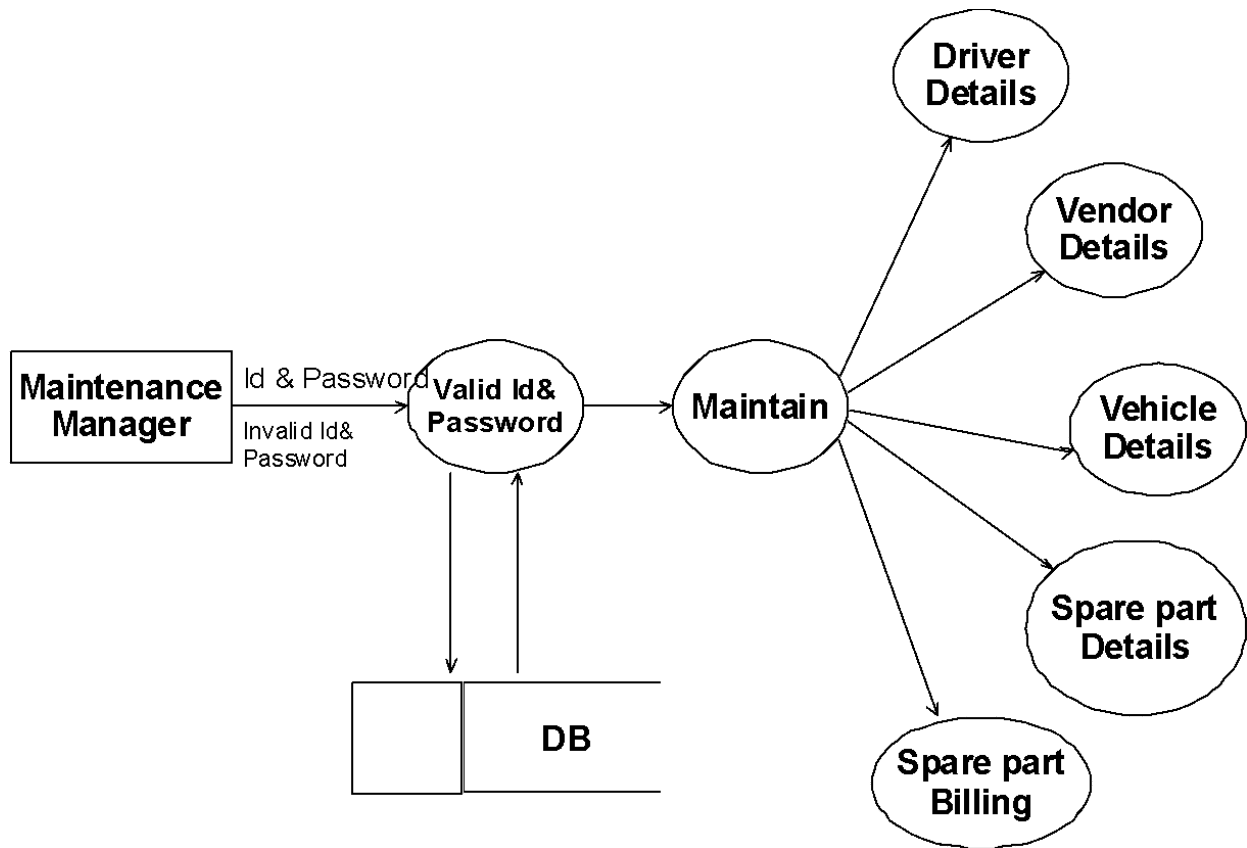




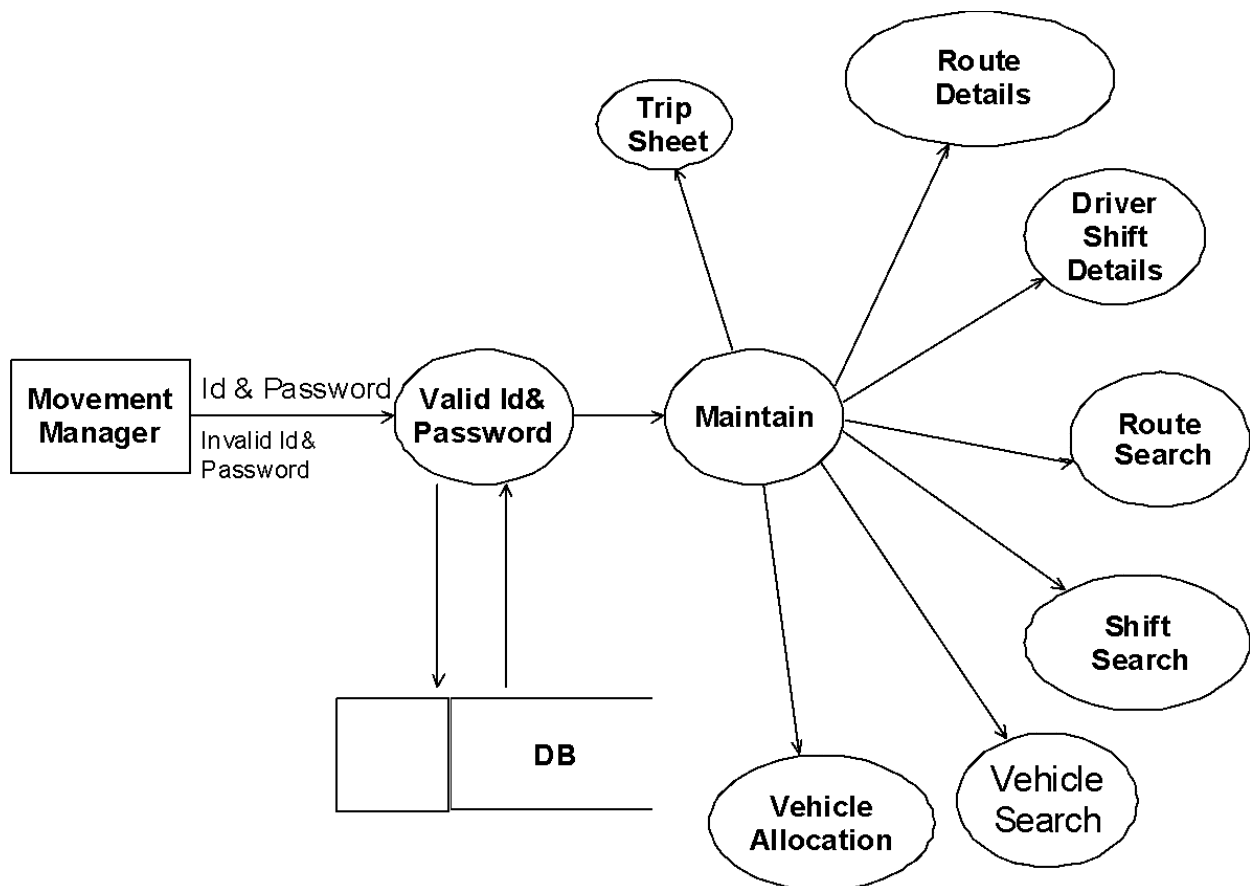
Level 1 Diagram Admin:



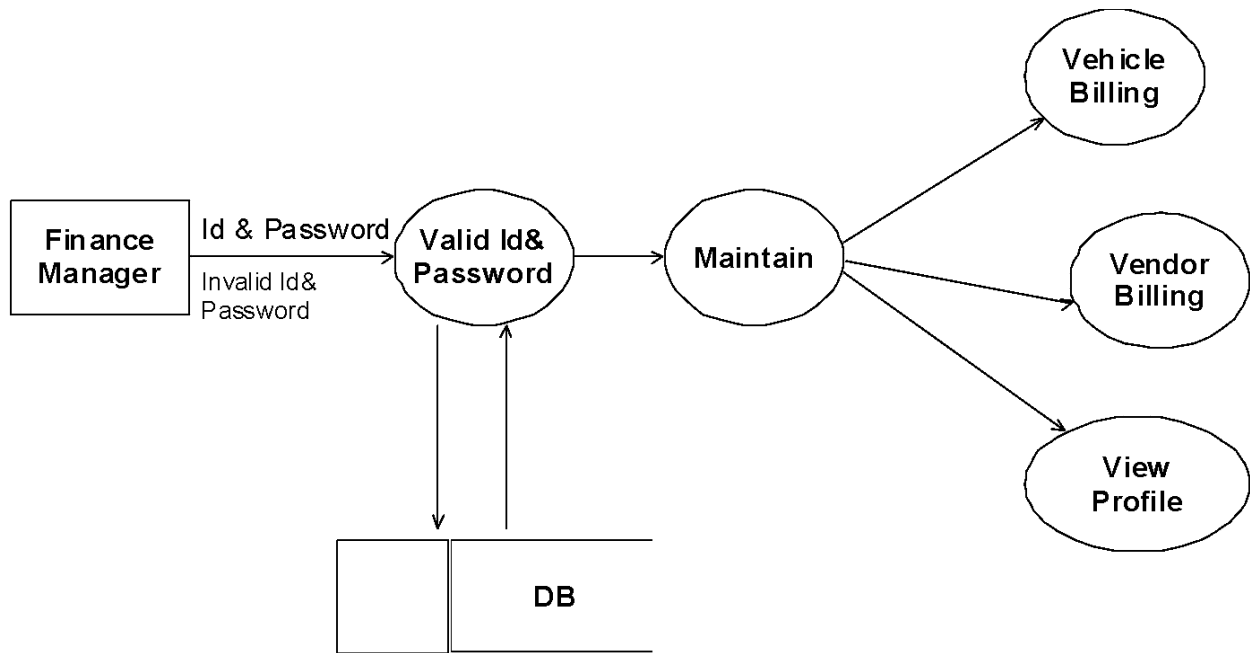
Level 2 Diagram Maintenance manager:



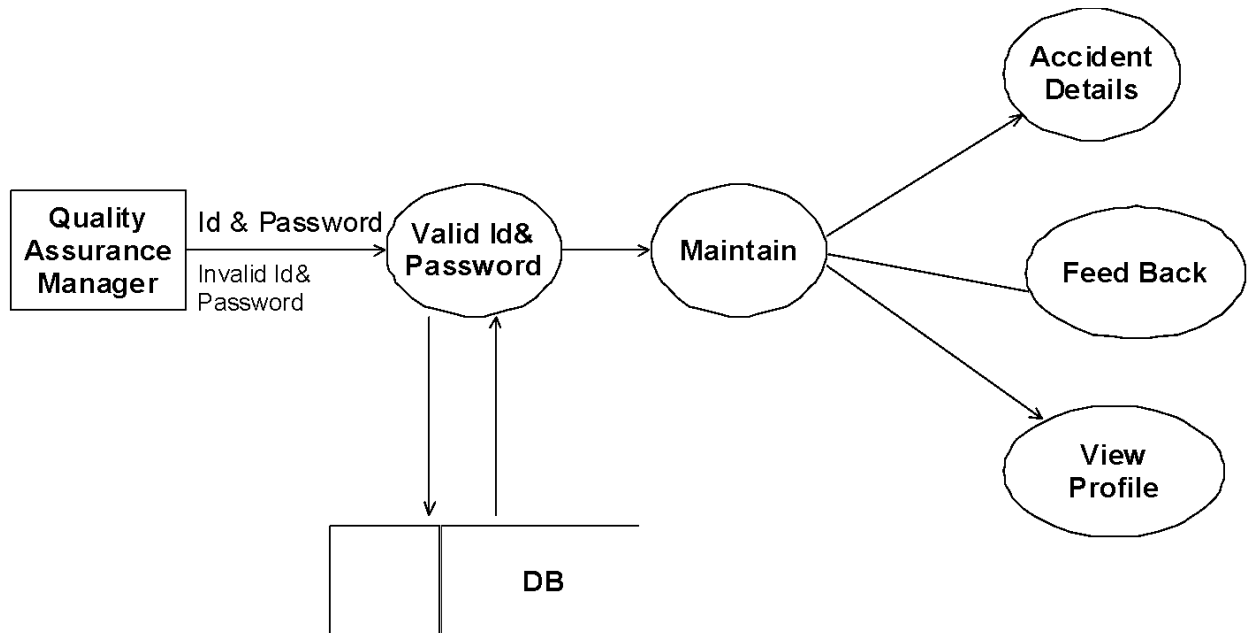
Level 2 Diagram Movement manager:



Level 2 Diagram Finance manager:



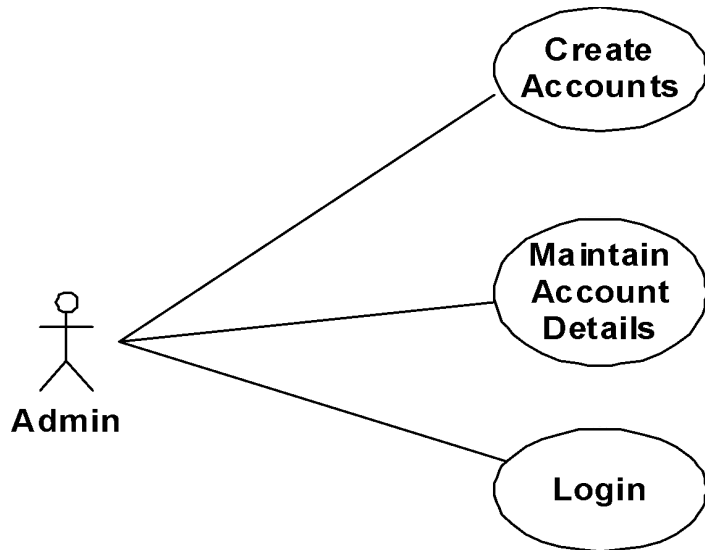
Level 2 Diagram Quality Assurance manager:



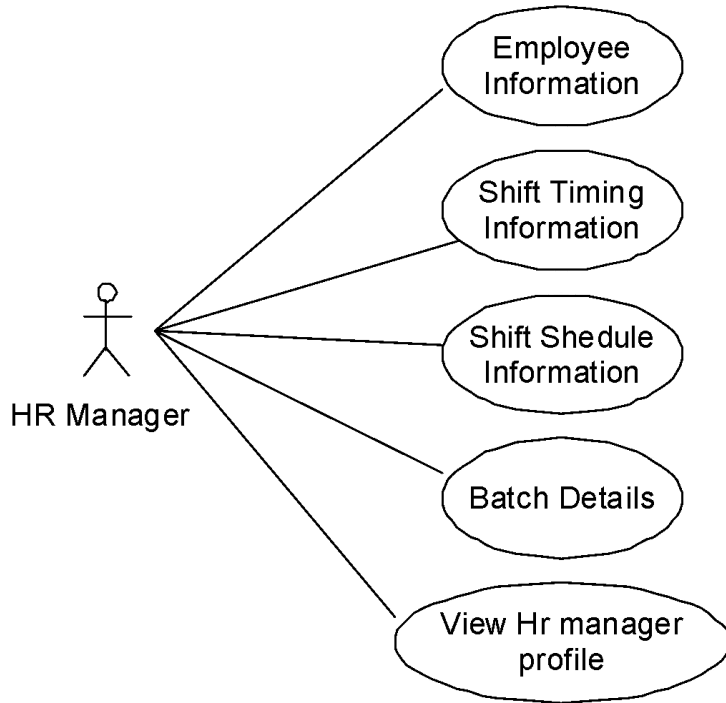
5.3UML Diagrams

Use Case:

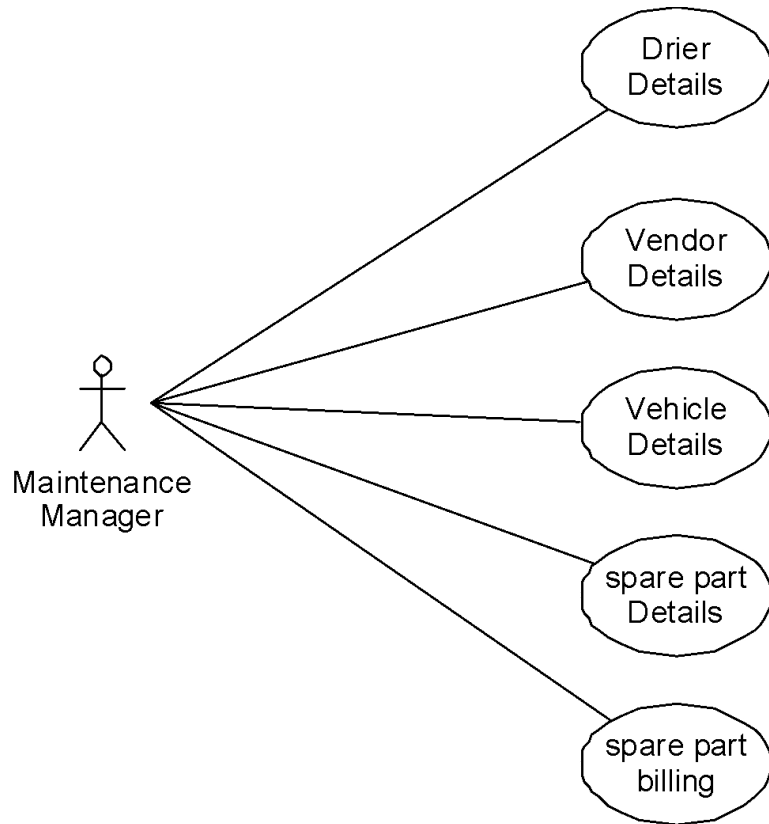
Admin:



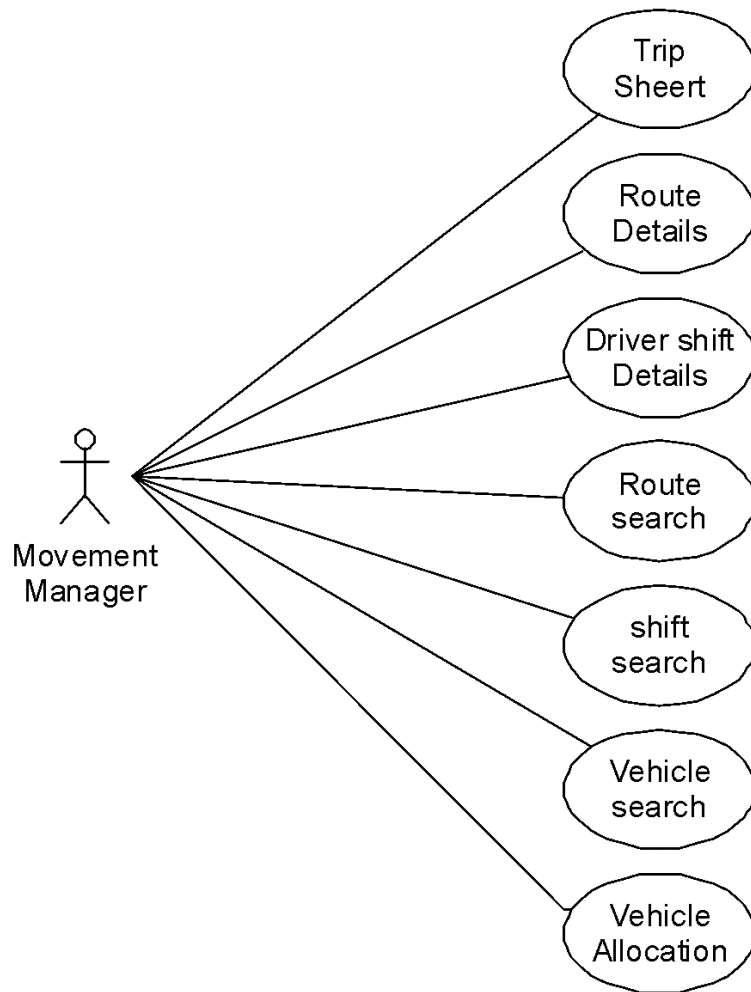
HR Manager:



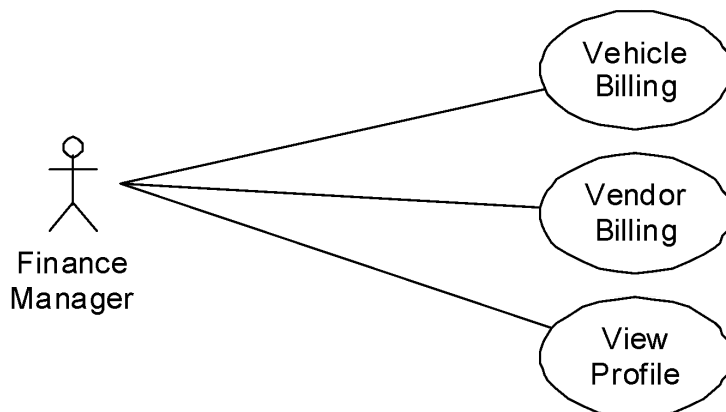
Maintenance Manager:



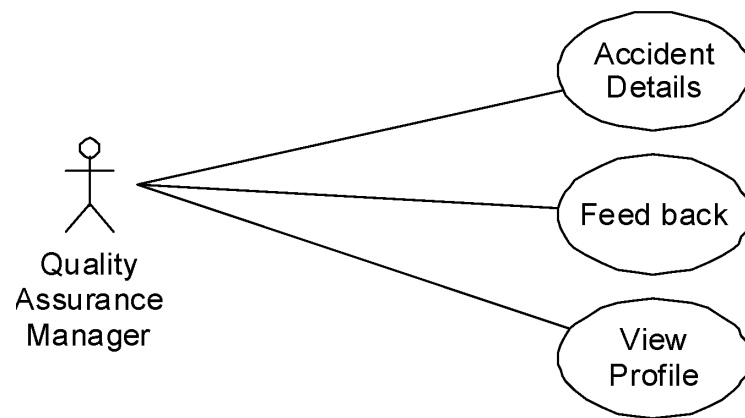
Movement Manager:



Finance Manager:

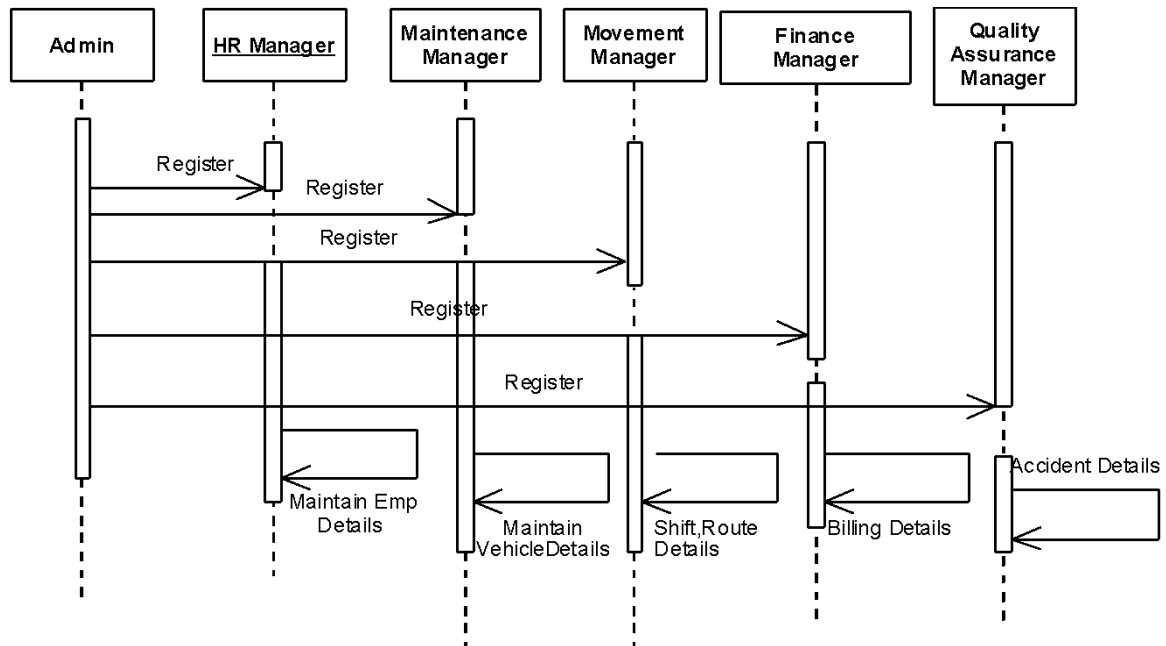


Quality Assurance Manager:



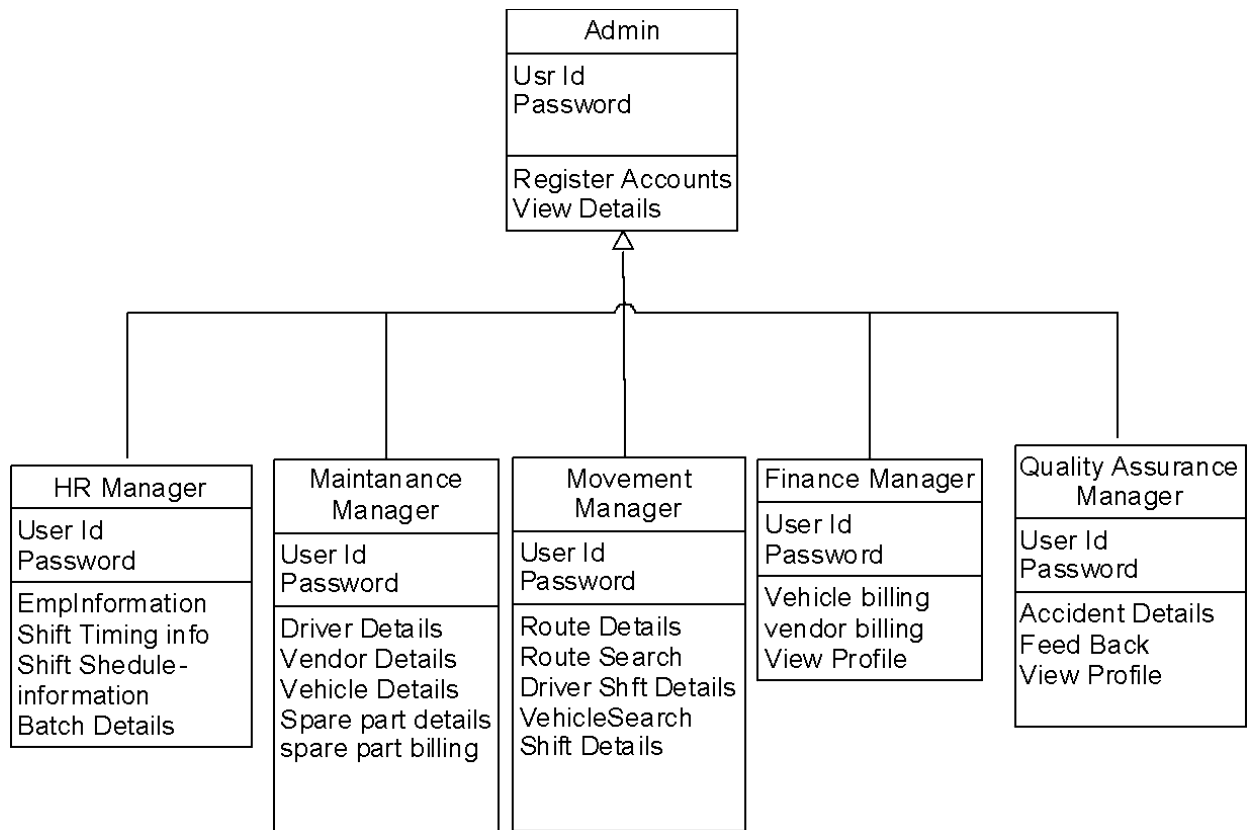
Sequence Diagram:

COOL CAB SERVICES

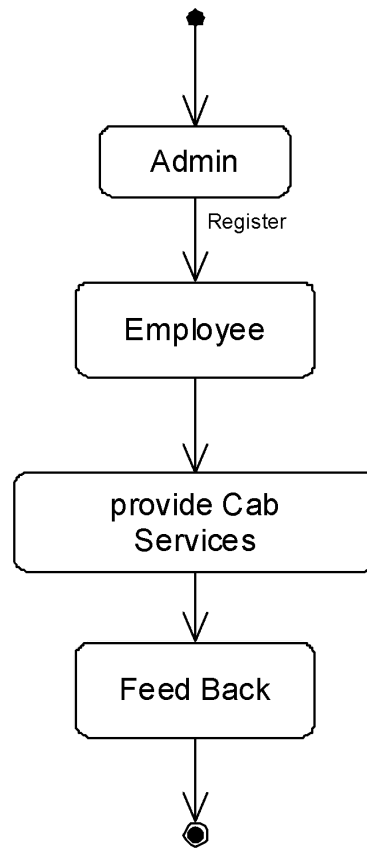


Class Diagram:

COOL CAB SERVICES



State Chart Diagram:

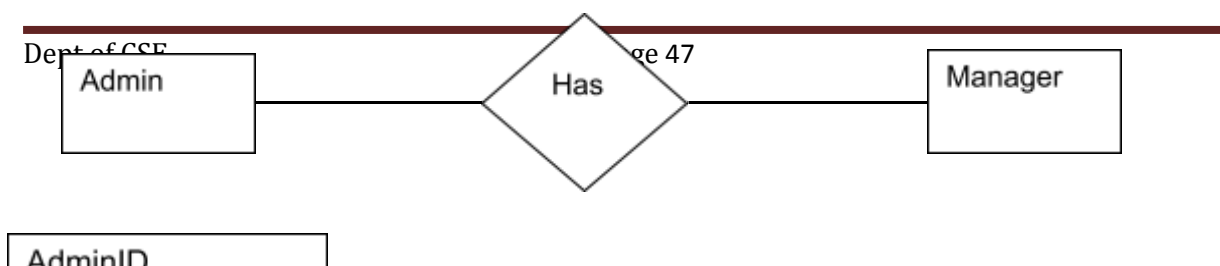


5.4 DB DESIGN

Entities

- Tbl_AccidentDetails
- Tbl_AdminLogin
- Tbl_BatchDetails
- Tbl_DriverDetails
- Tbl_DriverShiftDetails
- Tbl_EmployeeDetails
- Tbl_FeedBackFrom
- Tbl_Manger
- Tbl_RouteDetails
- Tbl_ShiftSchedule
- Tbl_ShiftTiming
- Tbl_SparePartBiiling
- Tbl_SparePartsDetails
- Tbl_TripSheet
- Tbl_VehicleAllocationDetails
- Tbl_VehicleBillingTransction
- Tbl_VehicleDetails
- Tbl_VenderDetails
- tblAdminNewLogin

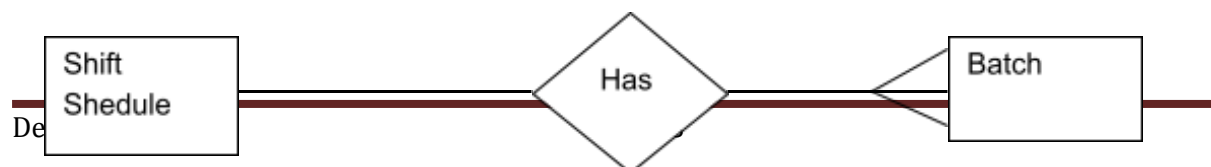
ER Diagrams



COOL CAB SERVICES

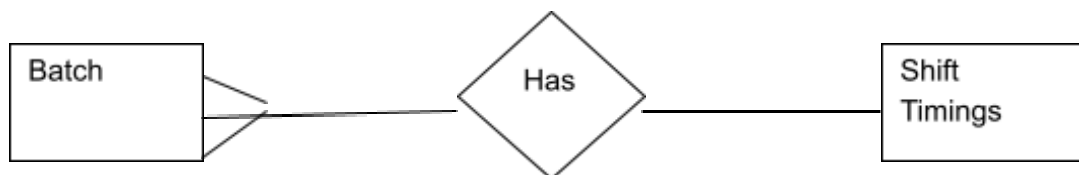


MangerID
EmpName
Address
Qulification
DOB
Gender
PhoneNo
EmailID
Designation
Department
DOJ
Age



SID
SShiftScheduleID
EmpID
Department
BatchID
EmpName
ShiftID
RouteID

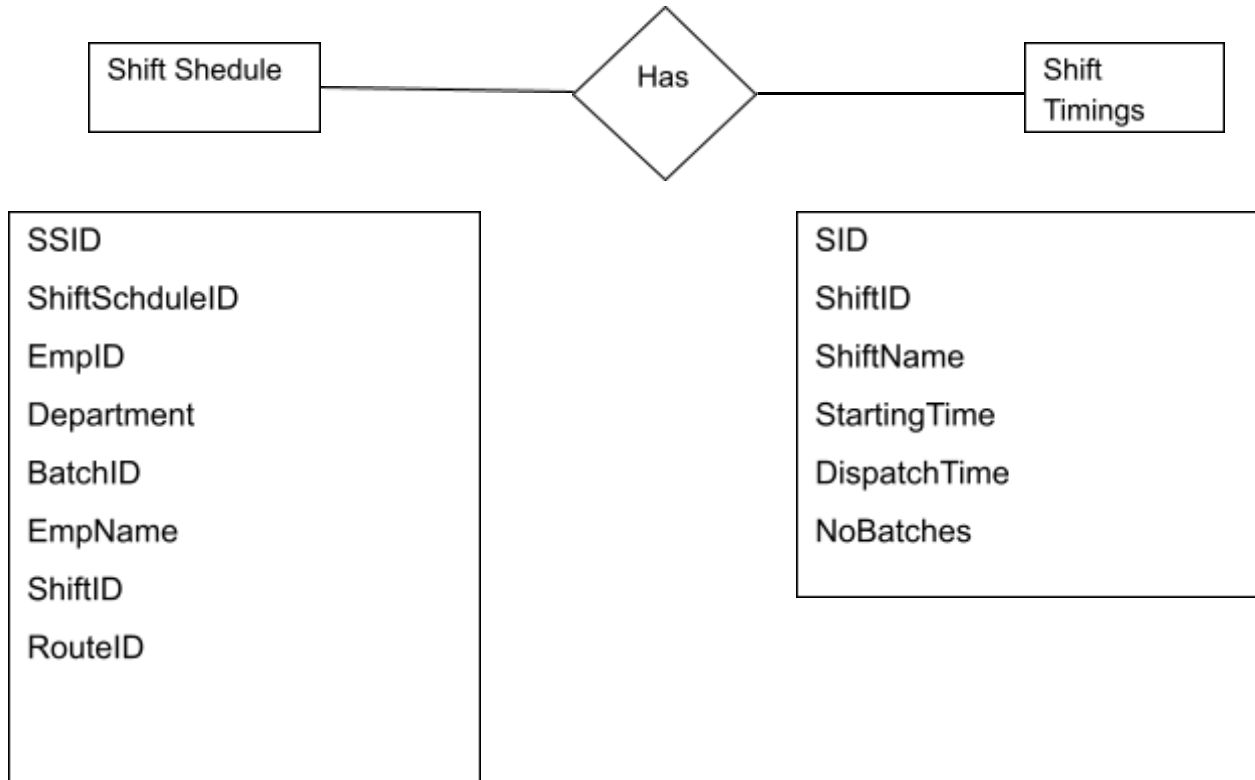
BID
BatchID
TotalNoOfEmployees
ShiftID

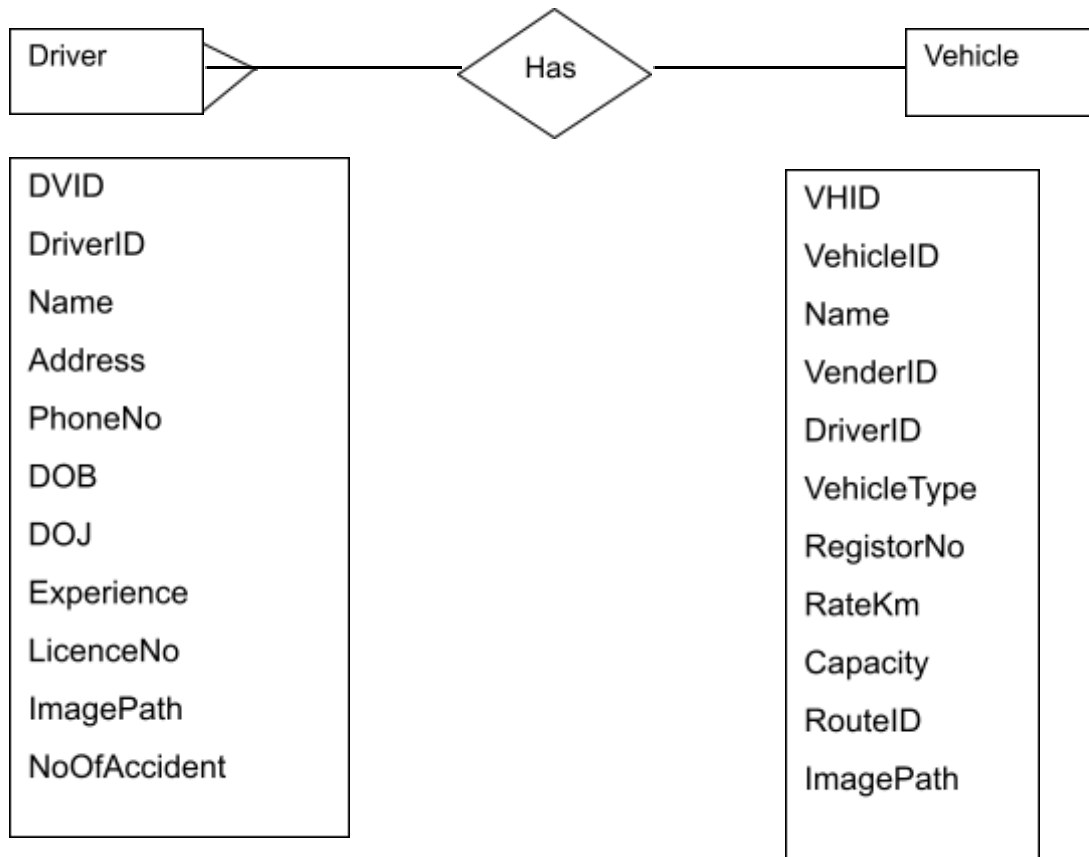


BID
BatchID
TotalNoOfEmployees
ShiftID

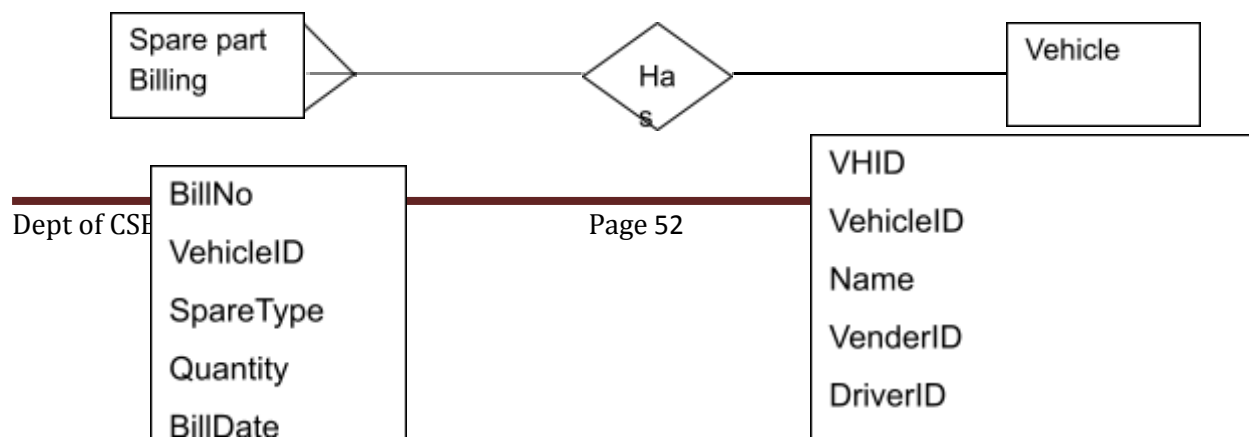
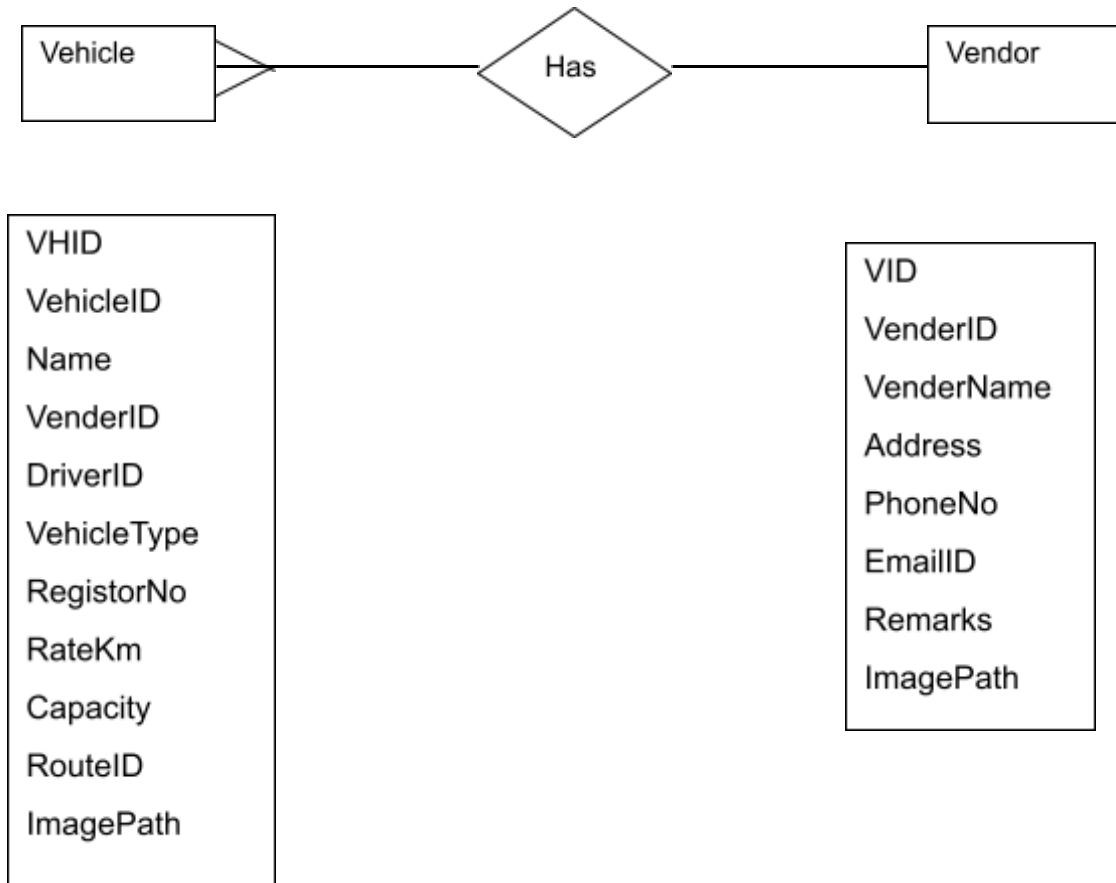
SID
ShiftID
ShiftName
StartingTime
DispatchTime
NoBatches

COOL CAB SERVICES

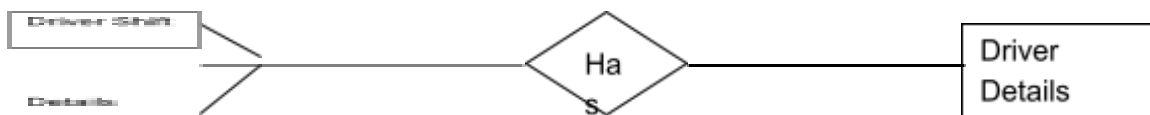




COOL CAB SERVICES

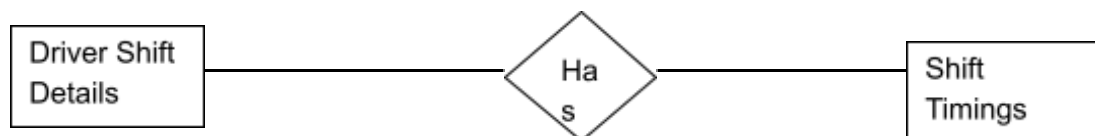


COOL CAB SERVICES



DSID
DriverShiftID
Name
DriverID
ShiftID
ShiftDate
Shifting

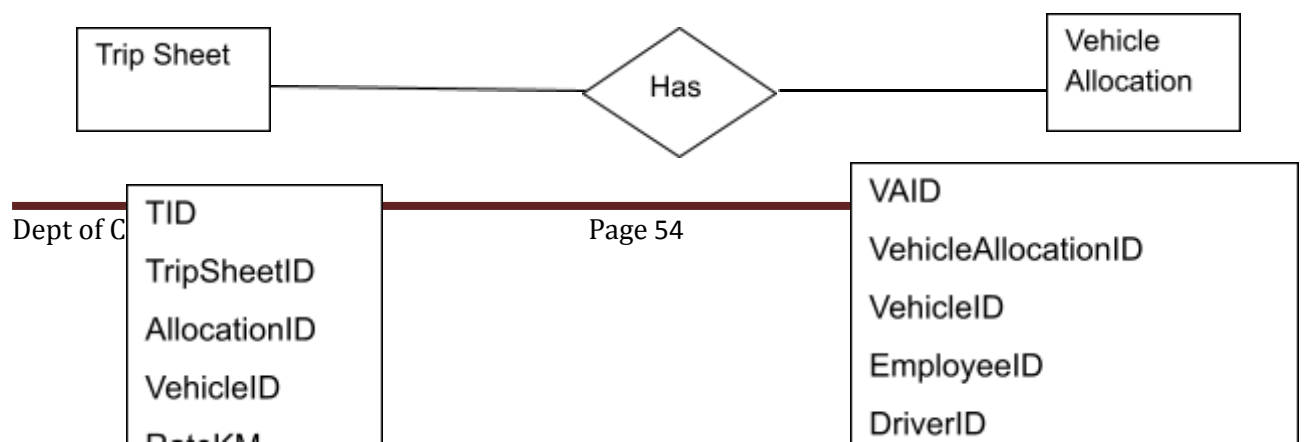
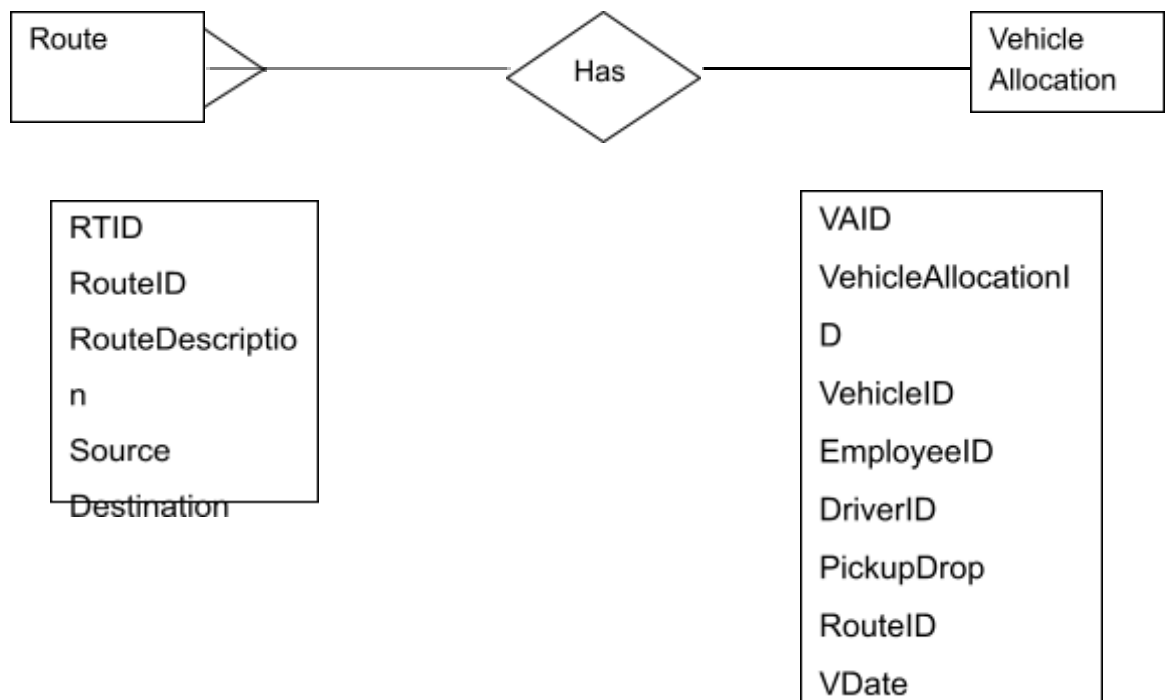
DVID
DriverID
Name
Address
PhoneNo
DOB
DOJ
Experience



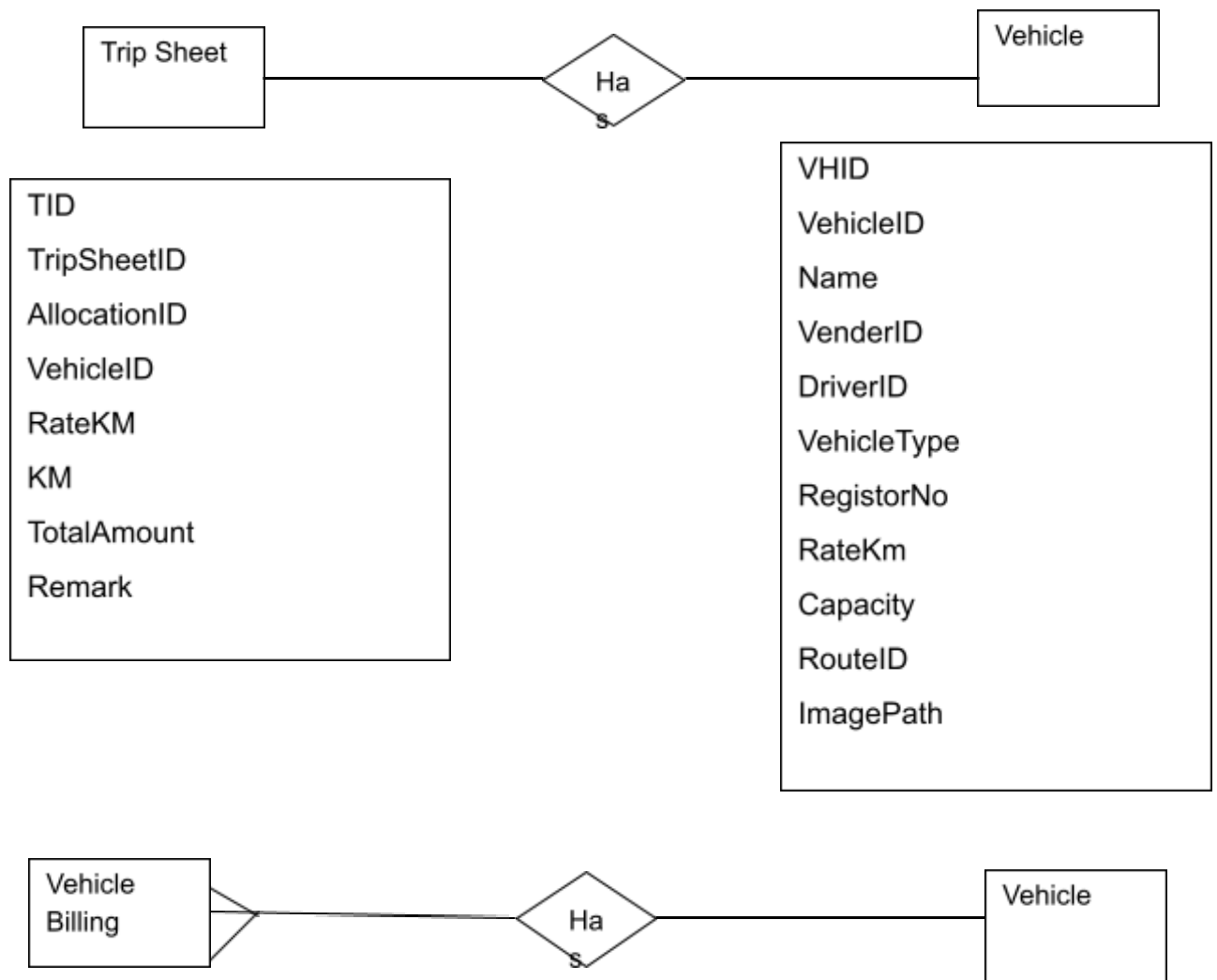
Dept o

DSID
DriverShiftID
Name
DriverID
ShiftID

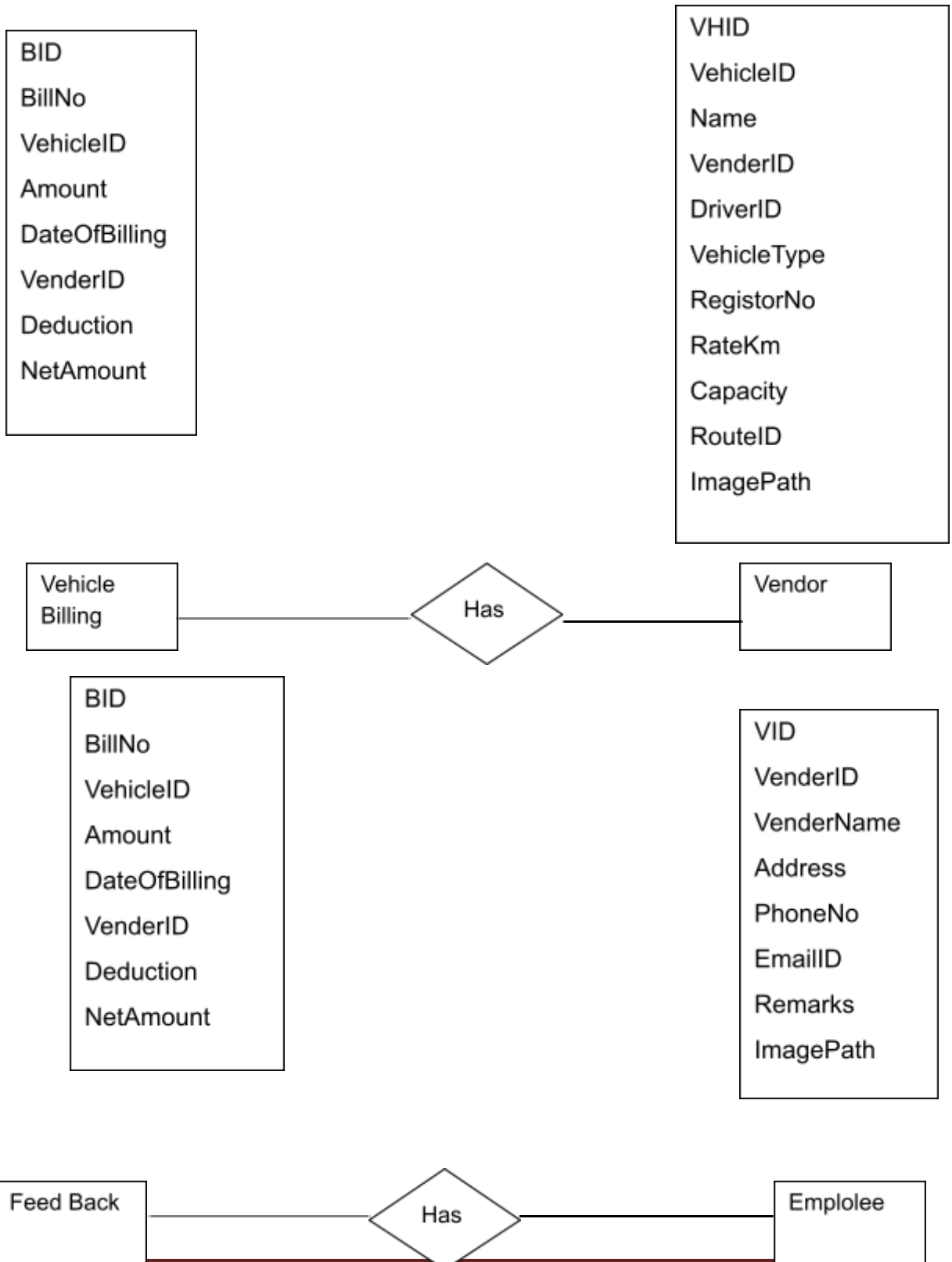
SID
ShiftID
ShiftName
StartingTime
DispatchTime



COOL CAB SERVICES



COOL CAB SERVICES

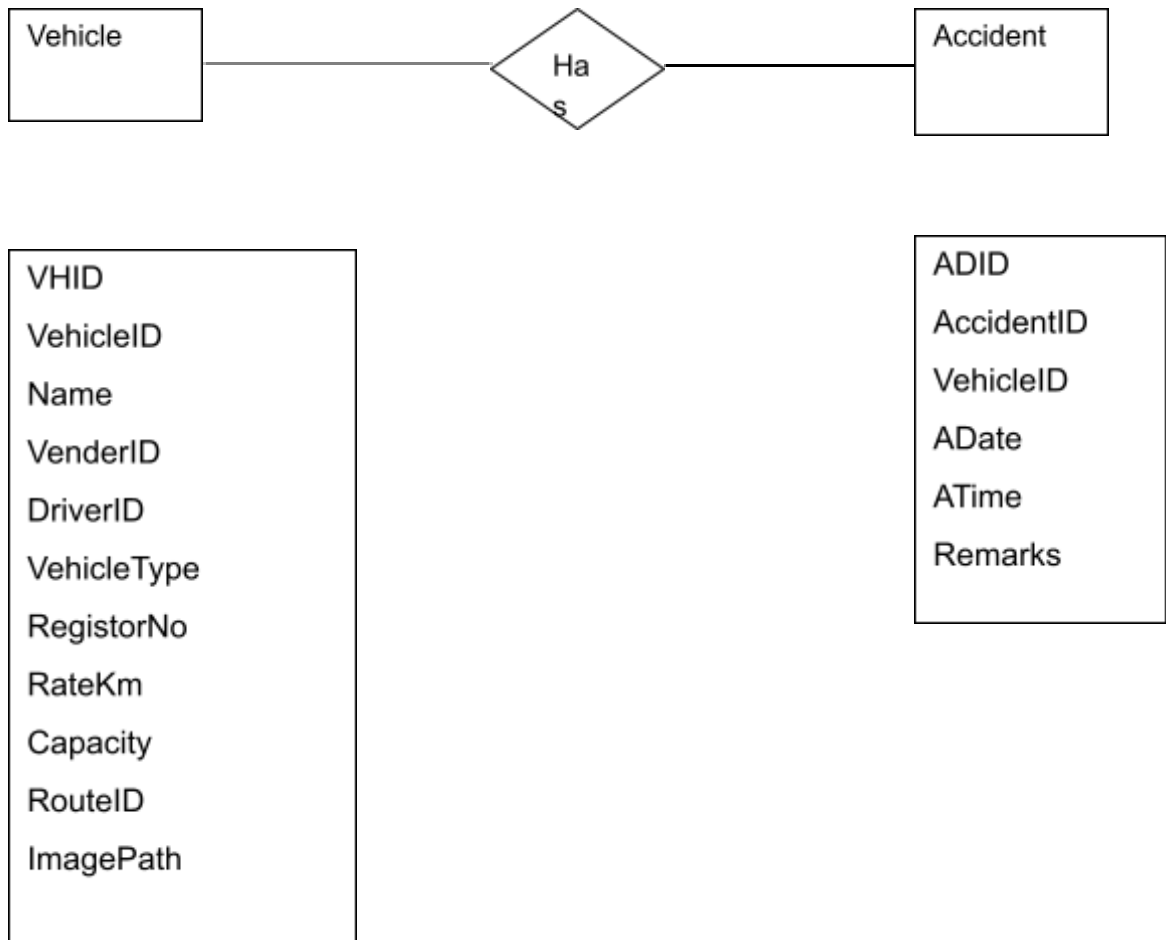


COOL CAB SERVICES

FBID
FeedBackID
EmpID
VehicleID
DriverID
Remarks

EID
EmpID
EmpName
PAddress
CAddress
Qulification
DOB
VehicleRequire
Gender
PhoneNo
Designation
Department
DOJ
Status
Age
TimeSpan
ImagePath

COOL CAB SERVICES



DATA DICTIONARY

Tbl_AccidentDetails

Sno	Columnname	Datatype	Constraint	reference
1	ADID	int	Primary key	
2	AccidentID	varchar(50)	FK	Tbl_accident details
3	VehicleID	varchar(50)	FK	Vehicle details
4	ADate	varchar(50)	Allow Null	
5	ATime	varchar(50)	Allow Null	
6	Remarks	varchar(50)	Allow Null	

Tbl_AdminLogin

Sno	Columnname	Datatype	Constraint	reference
1	AdminID	varchar(50)	Primary key	
2	UserName	varchar(50)	Not null	
3	Password	varchar(50)	Notnull	
4	EmailID	varchar(50)	Allow null	
5	Department	varchar(50)	Not null	

Tbl_BatchDetails

Sno	Columnname	Datatype	Constraint	reference
1	BID	int	Primary key	
2	BatchID	varchar(50)	Not null	
3	TotalNoOfEmployees	varchar(50)	Allownull	
4	ShiftID	varchar(50)	Foreignkey	Tbl_shift details

Tbl_DriverDetails

COOL CAB SERVICES

Sno	Columnname	Datatype	Constraint	reference
1	DVID	int	Primary key	
2	DriverID	varchar(50)	Allow null	
3	Name	varchar(50)	Allow null	
4	Address	varchar(50)	Allow null	
5	PhoneNo	varchar(50)	Allow null	
6	DOB	Datetime	Allow null	
7	DOJ	datetime	Allow null	
8	Experience	varchar(50)	Allow null	
9	LicenceNo	varchar(50)	Allow null	
10	ImagePath	varchar(50)	Allow null	
11	NoOfAccident	varchar(50)	Allow null	

Tbl_DriverShiftDetails

Sno	Columnname	Datatype	Constraint	reference
1	DSID	int	PK	
2	DriverShiftID	varchar(50)	Not null	
3	Name	varchar(50)	Not null	
4	DriverID	varchar(50)	Not null	
5	ShiftID	varchar(50)	Not null	
6	ShiftDate	datetime	Not null	
7	Shifting	varchar(50)	Not null	

Tbl_EmployeeDetails

COOL CAB SERVICES

Sno	Columnname	Datatype	Constraint	reference
1	EID	int	PK	
2	EmpID	varchar(50)	Notnull	
3	EmpName	varchar(50)	Notnull	
4	PAddress	varchar(50)	Notnull	
5	CAddress	varchar(50)	Allownull	
6	Qualification	varchar(50)	Notnull	
7	DOB	datetime	Allownull	
8	VehicleRequire	varchar(50)	Notnull	
9	Gender	varchar(50)	Allownull	
10	PhoneNo	varchar(50)	Notnull	
11	Designation	varchar(50)	Notnull	
12	Department	varchar(50)	Notnull	
13	DOJ	datetime	Allownull	
14	Status	varchar(50)	Notnull	
15	Age	varchar(50)	Allownull	
16	TimeSpan	varchar(50)	Notnull	
17	ImagePath	varchar(50)	Allownull	

Tbl_FeedBackFrom

COOL CAB SERVICES

Sno	Columnname	Datatype	Constraint	reference
1	FBID	int	PK	
2	FeedBackID	varchar(50)	Not NULL	
3	EmpID	varchar(50)	FK	Tbl_Employee
4	VehicleID	varchar(50)	FK	TBL Vehicle
5	DriverID	varchar(50)	FK	Tbl_Driver details
6	Remarks	varchar(50)	Not NULL	

Tbl_Manger

Sno	Columnname	Datatype	Constraint	Reference
1	MangerID	varchar(50)	PK	
2	EmpName	varchar(50)	notnull	
3	Address	varchar(50)	Notnull	
4	Qualification	varchar(50)	Notnull	
5	DOB	Datetime	Notnull	
6	Gender	varchar(50)	notnull	
7	PhoneNo	varchar(50)	Notnull	
8	EmailID	varchar(50)	Notnull	
9	Designation	varchar(50)	Notnull	
10	Department	varchar(50)	Notnull	
11	DOJ	Datetime	Notnull	
12	Age	int	Notnull	

Tbl_RouteDetails

COOL CAB SERVICES

Sno	Columnname	Datatype	Constraint	reference
1	RTID	Int	PK	
2	Routed	varchar(50)	NotNULL	
3	RouteDescription	varchar(50)	NotNULL	
4	Source	varchar(50)	NotNULL	
5	Destination	varchar(50)	NotNULL	

Tbl_ShiftSchedule

Sno	Columnname	Datatype	constraint	reference
1	SSID	int	PK	
2	ShiftSchduleID	varchar(50)	FK	Tbl_Shifttimings
3	EmpID	varchar(50)	FK	Tbl_Employee
4	Department	varchar(50)	Notnull	
5	BatchID	varchar(50)	FK	Tbl_batchdetails
6	EmpName	varchar(50)	Notnull	
7	ShiftID	varchar(50)	FK	Tbl_Shiftdetails
8	Routed	varchar(50)	FK	Tbl_Routedetails

Tbl_ShiftTimeing

COOL CAB SERVICES

Sno	Columnname	Datatype	Constraint	reference
1	SID	int	PK	
2	ShiftID	varchar(50)	NotNull	
3	ShiftName	varchar(50)	NotNull	
4	StartingTime	varchar(50)	NotNull	
5	DispatchTime	varchar(50)	NotNull	
6	NoBatches	varchar(50)	NotNull	

Tbl_SparePartBilling

Sno	Columnname	Datatype	Constraint	Reference
1	BillNo	varchar(50)	PK	
2	VehicleID	varchar(50)	FK	Tbl_Vehicledetails
3	SpareType	varchar(50)	Notnull	
4	Quantity	Int	Notnull	
5	BillDate	varchar(50)	Notnull	
6	SparePart	varchar(50)	Notnull	
7	Price	Int	Notnull	
8	TotalAmount	int	Notnull	

Tbl_SparePartsDetails

COOL CAB SERVICES

Sno	Columnname	Datatype	Constraint	Reference
1	SPID	Int	PK	
2	SparerPartID	varchar(50)	NotNull	
3	DealerName	varchar(50)	NotNull	
4	SparePartType	varchar(50)	NotNull	
5	Quantity	varchar(50)	NotNull	
6	SparePart	varchar(50)	NotNull	
7	DateOfPurchase	datetime	NotNull	
8	Price	varchar(50)	NotNull	
9	AmountPaid	varchar(50)	NotNull	

COOL CAB SERVICES

Tbl_TripSheet

Sno	Columnname	Datatype	Constraint	Datatype
1	TID	int	PK	
2	TripSheetID	varchar(50)	Not NULL	
3	AllocationID	varchar(50)	FK	Tbl_VehicleAllocation
4	VehicleID	varchar(50)	FK	Tbl_Vehicle details
5	RateKM	varchar(50)	Not NULL	
6	KM	varchar(50)	Not Null	
7	TotalAmount	int	Not Null	
8	Remark	varchar(50)	Not Null	

Tbl_VehicleAllocationDetails

Sno	Columnname	Datatype	Constraint	Reference
1	VAID	int	PK	
2	VehicleAllocationID	varchar(50)	Not NULL	
3	VehicleID	varchar(50)	Fk	Tbl_Vehicle details
4	EmployeeID	varchar(50)	FK	Tbl_Employee details
5	DriverID	varchar(50)	FK	Tbl_DriverDetails
6	PickupDrop	varchar(50)	Not NULL	
7	Routed	int	Not NULL	
8	VDate	varchar(50)	Not NULL	

Tbl_VehicleDetails

COOL CAB SERVICES

Sno	Columnname	Datatype	Constraint	Reference
1	VHID	int	PK	
2	VehicleID	varchar(50)	Not Null	
3	Name	varchar(50)	Not Null	
4	VenderID	varchar(50)	Not Null	
5	DriverID	varchar(50)	Not Null	
6	VehicleType	varchar(50)	Not Null	
7	RegistorNo	varchar(50)	Not Null	
8	RateKm	varchar(50)	Not Null	
9	Capacity	varchar(50)	Not Null	
10	Routed	int	Not Null	

Tbl_VenderDetails

Sno	Columnname	Datatype	Constraint	reference
1	VID	Int	PK	
2	VenderID	varchar(50)	NotNull	
3	VenderName	varchar(50)	NotNull	
4	Address	varchar(50)	NotNull	
5	PhoneNo	varchar(50)	NotNull	

Chapter 6

Implementation of Project

Description of Technology Used in Project.

6.1. INTRODUCTION TO .NET Framework

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts..

FEATURES OF THE COMMON LANGUAGE RUNTIME

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive functions, even if it is being used in the same active application.

The runtime enforces code access security. For example, users can trust that an executable embedded in a Web page can play an animation on screen or sing a song, but cannot access their personal data, file system, or network. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally featuring rich.

The runtime also enforces code robustness by implementing a strict type- and code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The various Microsoft and third-party language compilers

Generate managed code that conforms to the CTS. This means that managed code can consume other managed types and instances, while strictly enforcing type fidelity and type safety.

In addition, the managed environment of the runtime eliminates many common software issues. For example, the runtime automatically handles object layout and manages references to objects, releasing them when they are no longer being used. This automatic memory management resolves the two most common application errors, memory leaks and invalid memory references.

The runtime also accelerates developer productivity. For example, programmers can write applications in their development language of choice, yet take full advantage of the runtime, the class library, and components written in other languages by other developers.

.NET FRAMEWORK CLASS LIBRARY

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

For example, the .NET Framework collection classes implement a set of interfaces that you can use to develop your own collection classes. Your collection classes will blend seamlessly with the classes in the .NET Framework.

As you would expect from an object-oriented class library, the .NET Framework types enable you to accomplish a range of common programming tasks, including tasks such as string management, data collection, database connectivity, and file access. In addition to these common tasks, the class library includes types that support a variety of specialized development scenarios. For example, you can use the .NET Framework to develop the mentioned types of applications and services:

- i. Console applications.
- ii. Scripted or hosted applications.
- iii. Windows GUI applications (Windows Forms).
- iv. ASP.NET applications.
- v. XML Web services.
- vi. Windows services.

For example, the Windows Forms classes are a comprehensive set of reusable types that vastly simplify Windows GUI development. If you write an ASP.NET Web Form application, you can use the Web Forms classes.

CLIENT APPLICATION DEVELOPMENT

Client applications are the closest to a traditional style of application in Windows-based programming. These are the types of applications that display windows or forms on the desktop, enabling a user to perform a task. Client applications include applications such as word processors and spreadsheets, as well as custom business applications such as data-entry tools, reporting tools, and so on. Client applications usually employ windows, menus, buttons, and other GUI elements, and they likely access local resources such as the file system and peripherals such as printers.

Another kind of client application is the traditional ActiveX control (now replaced by the managed Windows Forms control) deployed over the Internet as a Web page. This application is much like other client applications: it is executed natively, has access to local resources, and includes graphical elements.

In the past, developers created such applications using C/C++ in conjunction with the Microsoft Foundation Classes (MFC) or with a rapid application development (RAD) environment such as Microsoft® Visual Basic®. The .NET Framework incorporates aspects of these existing products into a single, consistent development environment that drastically simplifies the development of client applications.

The Windows Forms classes contained in the .NET Framework are designed to be used for GUI development. You can easily create command windows, buttons, menus, toolbars, and other screen elements with the flexibility necessary to accommodate shifting business needs.

For example, the .NET Framework provides simple properties to adjust visual attributes associated with forms. In some cases the underlying operating system does not support changing these attributes directly, and in these cases the .NET Framework automatically recreates the forms. This is one of many ways in which the .NET Framework integrates the developer interface, making coding simpler and more consistent.

6.2 ASP.NET

Server Application Development

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows your custom managed code to control the behavior of the server. This model provides you with all the features of the common language runtime and class library while gaining the performance and scalability of the host server.

The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and SQL Server can perform standard operations while your application logic executes through the managed code.

SERVER-SIDE MANAGED CODE

ASP.NET is the hosting environment that enables developers to use the .NET Framework to target Web-based applications. However, ASP.NET is more than just a runtime host; it is a complete architecture for developing Web sites and Internet-distributed objects using managed code. Both Web Forms and XML Web services use IIS and ASP.NET as the publishing mechanism for applications, and both have a collection of supporting classes in the .NET Framework.

XML Web services, an important evolution in Web-based technology, are distributed, server-side application components similar to common Web sites. However, unlike Web-based applications, XML Web services components have no UI and are not targeted for browsers such as Internet Explorer and Netscape Navigator. Instead, XML Web services consist of reusable software components designed to be consumed by other applications, such as traditional client applications, Web-based applications, or even other XML Web services. As a result, XML Web services technology is rapidly moving application development and deployment into the highly distributed environment of the Internet.

ACTIVE SERVER PAGES.NET

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET offers several important advantages over previous Web development models:

- **Enhanced Performance.** ASP.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box. This amounts to dramatically better performance before you ever write a line of code.
- **World-Class Tool Support.** The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides.
- **Power and Flexibility.** Because ASP.NET is based on the common language runtime, the power and flexibility of that entire platform is available to Web application developers. The .NET Framework class library, Messaging, and Data Access solutions are all seamlessly accessible from the Web. ASP.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages. Further, common language runtime interoperability guarantees that your existing investment in COM-based development is preserved when migrating to ASP.NET.
- **Simplicity.** ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration. For example, the ASP.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.
- **Manageability.** ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and Web applications.

Because configuration information is stored as plain text, new settings may be applied without the aid of local administration tools. This "zero local administration" philosophy extends to deploying ASP.NET Framework applications as well. An ASP.NET Framework application is deployed to a server simply by copying the necessary files to the server. No server restart is required, even to deploy or replace running compiled code.

- **Scalability and Availability.** ASP.NET has been designed with scalability in mind, with features specifically tailored to improve performance in clustered and multiprocessor environments. Further, processes are closely monitored and managed by the ASP.NET runtime, so that if one misbehaves (leaks, deadlocks), a new process can be created in its place, which helps keep your application constantly available to handle requests.
- **Customizability and Extensibility.** ASP.NET delivers a well-factored architecture that allows developers to "plug-in" their code at the appropriate level. In fact, it is possible to extend or replace any subcomponent of the ASP.NET runtime with your own custom-written component. Implementing custom authentication or state services has never been easier.
- **Security.** With built in Windows authentication and per-application configuration, you can be assured that your applications are secure.

LANGUAGE SUPPORT

The Microsoft .NET Platform currently offers built-in support for three languages: C#, Visual Basic, and JScript.

WHAT IS ASP.NET WEB FORMS?

The ASP.NET Web Forms page framework is a scalable common language runtime programming model that can be used on the server to dynamically generate Web pages.

CODE-BEHIND WEB FORMS

ASP.NET supports two methods of authoring dynamic pages. The first is the method shown in the preceding samples, where the page code is physically declared within the originating .aspx file. An alternative approach--known as the code-behind method--enables the page code to be more cleanly separated from the HTML content into an entirely separate file.

INTRODUCTION TO ASP.NET SERVER CONTROLS

In addition to (or instead of) using `<% %>` code blocks to program dynamic content, ASP.NET page developers can use ASP.NET server controls to program Web pages. Server controls are declared within an .aspx file using custom tags or intrinsic HTML tags that contain a **runat="server"** attributes value. Intrinsic HTML tags are handled by one of the controls in the **System.Web.UI.HtmlControls** namespace. Any tag that doesn't explicitly map to one of the controls is assigned the type of **System.Web.UI.HtmlControls.HtmlGenericControl**.

Server controls automatically maintain any client-entered values between round trips to the server. This control state is not stored on the server (it is instead stored within an **<input type="hidden">** form field that is round-tripped between requests). Note also that no client-side script is required.

In addition to supporting standard HTML input controls, ASP.NET enables developers to utilize richer custom controls on their pages. For example, the following sample demonstrates how the **<asp:adrotator>** control can be used to dynamically display rotating ads on a page.

1. ASP.NET Web Forms provide an easy and powerful way to build dynamic Web UI.
2. ASP.NET Web Forms pages can target any browser client (there are no script library or cookie requirements).
3. ASP.NET Web Forms pages provide syntax compatibility with existing ASP pages.
4. ASP.NET server controls provide an easy way to encapsulate common functionality.

6.3 C#.NET

ADO.NET OVERVIEW

ADO.NET is an evolution of the ADO data access model that directly addresses user requirements for developing scalable applications. It was designed specifically for the web with scalability, statelessness, and XML in mind.

ADO.NET uses some ADO objects, such as the **Connection** and **Command** objects, and also introduces new objects. Key new ADO.NET objects include the **DataSet**, **DataReader**, and **DataAdapter**.

The important distinction between this evolved stage of ADO.NET and previous data architectures is that there exists an object -- the **DataSet** -- that is separate and distinct from any data stores. Because of that, the **DataSet** functions as a standalone entity. You can think of the **DataSet** as an always disconnected recordset that knows nothing about the source or destination of the data it contains. Inside a **DataSet**, much like in a database, there are tables, columns, relationships, constraints, views, and so forth.

A **DataAdapter** is the object that connects to the database to fill the **DataSet**. Then, it connects back to the database to update the data there, based on operations performed while the **DataSet** held the data. In the past, data processing has been primarily connection-based. Now, in an effort to make multi-tiered apps more efficient, data processing is turning to a message-based approach that revolves around chunks of information. At the center of this approach is the **DataAdapter**, which provides a bridge to retrieve and save data between a **DataSet** and its source data store. It accomplishes this by means of requests to the appropriate SQL commands made against the data store.

The XML-based **DataSet** object provides a consistent programming model that works with all models of data storage: flat, relational, and hierarchical. It does this by having no 'knowledge' of the source of its data, and by representing the data that it holds as

- **Connections.** For connection to and managing transactions against a database.
- **Commands.** For issuing SQL commands against a database.

- **DataReaders.** For reading a forward-only stream of data records from a SQL Server data source.
- **DataSets.** For storing, Remoting and programming against flat data, XML data and relational data.
- **DataAdapters.** For pushing data into a **DataSet**, and reconciling data against a database.

When dealing with connections to a database, there are two different options: SQL Server .NET Data Provider (System.Data.SqlClient) and OLE DB .NET Data Provider (System.Data.OleDb). In these samples we will use the SQL Server .NET Data Provider. These are written to talk directly to Microsoft SQL Server. The OLE DB .NET Data Provider is used to talk to any OLE DB provider (as it uses OLE DB underneath).

Connections:

Connections are used to 'talk to' databases, and are represented by provider-specific classes such as **SqlConnection**. Commands travel over connections and resultsets are returned in the form of streams which can be read by a **DataReader** object, or pushed into a **DataSet** object.

Commands:

Commands contain the information that is submitted to a database, and are represented by provider-specific classes such as **SqlCommand**. A command can be a stored procedure call, an UPDATE statement, or a statement that returns results. You can also use input and output parameters, and return values as part of your command syntax. The example below shows how to issue an INSERT statement against the **Northwind** database.

DataReaders:

The **DataReader** object is somewhat synonymous with a read-only/forward-only cursor over data. The **DataReader** API supports flat as well as hierarchical data. A **DataReader** object is returned after executing a command against a database. The format of the returned **DataReader** object is different from a recordset. For example, you might use the **DataReader** to show the results of a search list in a web page.

DATASETS AND DATAADAPTERS:

DataSets

The **DataSet** object is similar to the ADO **Recordset** object, but more powerful, and with one other important distinction: the **DataSet** is always disconnected. The **DataSet** object represents a cache of data, with database-like structures such as tables, columns, relationships, and constraints. However, though a **DataSet** can and does behave much like a database, it is important to remember that **DataSet** objects do not interact directly with databases, or other source data. This allows the developer to work with a programming model that is always consistent, regardless of where the source data resides. Data coming from a database, an XML file, from code, or user input can all be placed into **DataSet** objects. Then, as changes are made to the **DataSet** they can be tracked and verified before updating the source data. The **GetChanges** method of the **DataSet** object actually creates a second **DatSet** that contains only the changes to the data. This **DataSet** is then used by a **DataAdapter** (or other objects) to update the original data source.

The **DataSet** has many XML characteristics, including the ability to produce and consume XML data and XML schemas. XML schemas can be used to describe schemas interchanged via WebServices. In fact, a **DataSet** with a schema can actually be compiled for type safety and statement completion.

DATAADAPTERS (OLEDB/SQL)

The **DataAdapter** object works as a bridge between the **DataSet** and the source data. Using the provider-specific **SqlDataAdapter** (along with its associated **SqlCommand** and **SqlConnection**) can increase overall performance when working with a Microsoft SQL Server databases. For other OLE DB-supported databases, you would use the **OleDbDataAdapter** object and its associated **OleDbCommand** and **OleDbConnection** objects.

The **DataAdapter** object uses commands to update the data source after changes have been made to the **DataSet**. Using the **Fill** method of the **DataAdapter** calls the SELECT command; using the **Update** method calls the INSERT, UPDATE or DELETE command for each changed row. You can explicitly set these commands in order to control the statements used at runtime to resolve changes, including the use of stored procedures. For ad-hoc scenarios, a **CommandBuilder** object can generate these at run-time based upon a select statement. However, this run-time generation requires an extra round-trip to the server in order to gather required metadata, so explicitly providing the INSERT, UPDATE, and DELETE commands at design time will result in better run-time performance.

1. ADO.NET is the next evolution of ADO for the .Net Framework.
2. ADO.NET was created with n-Tier, statelessness and XML in the forefront. Two new objects, the **DataSet** and **DataAdapter**, are provided for these scenarios.
3. ADO.NET can be used to get data from a stream, or to store data in a cache for updates.
4. There is a lot more information about ADO.NET in the documentation.
5. Remember, you can execute a command directly against the database in order to do inserts, updates, and deletes. You don't need to first put data into a **DataSet** in order to insert, update, or delete it.
6. Also, you can use a **DataSet** to bind to the data, move through the data, and navigate data relationships

6.4 SQL SERVER

A database management, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, paradox, IMS, SQL Server and SQL Server. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record (it can also be referred to as raw or an occurrence). Each record is made up of a number of fields. No two fields in a record can have the same field name.

During an SQL Server Database design project, the analysis of your business needs identifies all the fields or attributes of interest. If your business needs change over time, you define any additional fields or change the definition of existing fields.

SQL SERVER TABLES

SQL Server stores records relating to each other in a table. Different tables are created for the various groups of information. Related tables are grouped together to form a database.

PRIMARY KEY

Every table in SQL Server has a field or a combination of fields that uniquely identifies each record in the table. The Unique identifier is called the Primary Key, or simply the Key. The primary key provides the means to distinguish one record from all other in a table. It allows the user and the database system to identify, locate and refer to one particular record in the database.

RELATIONAL DATABASE

Sometimes all the information of interest to a business operation can be stored in one table. SQL Server makes it very easy to link the data in multiple tables. Matching an employee to the department in which they work is one example. This is what makes SQL Server a relational database management system, or RDBMS. It stores data in two or more tables and enables you to define relationships between the table and enables you to define relationships between the tables.

FOREIGN KEY

When a field in one table matches the primary key of another field is referred to as a foreign key. A foreign key is a field or a group of fields in one table whose values match those of the primary key of another table.

REFERENTIAL INTEGRITY

Not only does SQL Server allow you to link multiple tables, it also maintains consistency between them. Ensuring that the data among related tables is correctly matched is referred to as maintaining referential integrity.

DATA ABSTRACTION

A major purpose of a database system is to provide users with an abstract view of the data. This system hides certain details of how the data is stored and maintained. Data abstraction is divided into three levels.

Physical level: This is the lowest level of abstraction at which one describes how the data are actually stored.

Conceptual Level: At this level of database abstraction all the attributes and what data are actually stored is described and entries and relationship among them.

View level: This is the highest level of abstraction at which one describes only part of the database.

ADVANTAGES OF RDBMS

- Redundancy can be avoided
- Inconsistency can be eliminated
- Data can be Shared
- Standards can be enforced
- Security restrictions can be applied
- Integrity can be maintained
- Conflicting requirements can be balanced
- Data independence can be achieved.

DISADVANTAGES OF DBMS

A significant disadvantage of the DBMS system is cost. In addition to the cost of purchasing of developing the software, the hardware has to be upgraded to allow for the extensive programs and the workspace required for their execution and storage.

FEATURES OF SQL SERVER (RDBMS)

SQL SERVER is one of the leading database management systems (DBMS) because it is the only Database that meets the uncompromising requirements of today's most demanding information systems. From complex decision support systems (DSS) to the most rigorous online transaction processing (OLTP) application, even application that require simultaneous DSS and OLTP access to the same critical data, SQL Server leads the industry in both performance and capability

ENTERPRISE WIDE DATA SHARING

The unrivaled portability and connectivity of the SQL SERVER DBMS enables all the systems in the organization to be linked into a singular, integrated computing resource.

PORTABILITY

SQL SERVER is fully portable to more than 80 distinct hardware and operating systems platforms, including UNIX, MSDOS, OS/2, Macintosh and dozens of proprietary platforms. This portability gives complete freedom to choose the database server platform that meets the system requirements.

OPEN SYSTEMS

SQL SERVER offers a leading implementation of industry –standard SQL. SQL Server's open architecture integrates SQL SERVER and non –SQL SERVER DBMS with industries most comprehensive collection of tools, application, and third party software products SQL Server's Open architecture provides transparent access to data from other relational database and even non-relational database.

DISTRIBUTED DATA SHARING

SQL Server's networking and distributed database capabilities to access data stored on remote server with the same ease as if the information was stored on a single local computer. A single SQL statement can access data at multiple sites. You can store data where system requirements such as performance, security or availability dictate.

UNMATCHED PERFORMANCE

The most advanced architecture in the industry allows the SQL SERVER DBMS to deliver unmatched performance.

SOPHISTICATED CONCURRENCY CONTROL

Real World applications demand access to critical data. With most database Systems application becomes “contention bound” – which performance is limited not by the CPU power or by disk I/O, but user waiting on one another for data access . SQL Server employs full, unrestricted row-level locking and contention free queries to minimize and in many cases entirely eliminates contention wait times.

NO I/O BOTTLENECKS

SQL Server’s fast commit groups commit and deferred write technologies dramatically reduce disk I/O bottlenecks. While some database write whole data block to disk at commit time, SQL Server commits transactions with at most sequential log file on disk at commit time, On high throughput systems, one sequential writes typically group commit multiple transactions. Data read by the transaction remains as shared memory so that other transactions may access that data without reading it again from disk. Since fast commits write all data necessary to the recovery to the log file, modified blocks are written back to the database independently of the transaction commit, when written from memory to disk.

6.5. SAMPLE CODE

```
using System;

using System.Data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Web.UI.HtmlControls;

public partial class Admin_AdminLogin : System.Web.UI.Page

{

    protected void Page_Load(object sender, EventArgs e)

    {

    }

    AdminBo Data = new AdminBo();

    protected void But_Login_Click(object sender, EventArgs e)
```

```
{  
  
    string Dept = string.Empty;  
  
    Data.UserId = Txt_UserName.Text;  
  
    Data.Password = Txt_Password.Text;  
  
    DataSet ds = new DataSet();  
  
    ds = Data.AdminLogin();  
  
    Dept = ds.Tables[0].Rows[0]["Department"].ToString().Trim();  
  
    if (Dept != "User NotExists")  
    {  
  
        Session["UserId"] = Txt_UserName.Text;  
  
        if (Dept == "Admin")  
        {  
  
            Response.Redirect("~/Admin/WelComeMainPage.aspx");  
  
        }  
  
        else if (Dept == "HRManager")  
        {  
  
            Response.Redirect("~/HRManger/HRManger.aspx");  
  
        }  
  
        else if (Dept == "MaintainenceManager")  
        {  

```

```
Response.Redirect("~/maintenanceManger/WelComeMaintenancePage.aspx");

    }

    else if (Dept == "Movementmanager")

    {

        Response.Redirect("~/MovementHomePage/MovementHomePage.aspx");

    }

    else if (Dept == "FinanceManager")

    {

        Response.Redirect("~/FinancialHomePage/WelcomePage.aspx");

    }

    else if (Dept == "QualityManager")

    {

        Response.Redirect("~/QulityAssranceManger/WelComepage.aspx");

    }

}

else

{

    Page.RegisterStartupScript("SS", "<script> alert('" + Dept +

    "");</script>");

}
```

Chapter 7

SYSTEM TESTING AND IMPLEMENTATION

7.1. INTRODUCTION

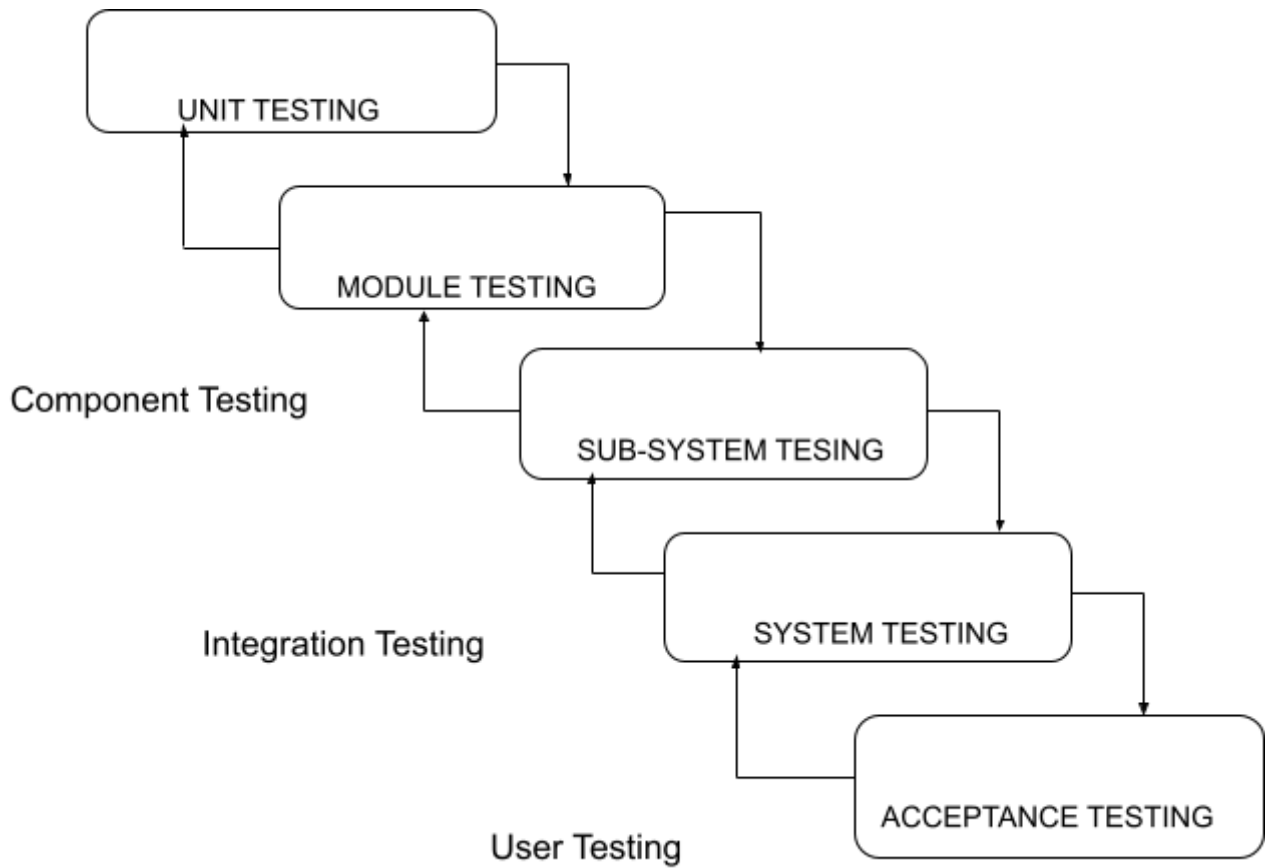
Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

7.2. SOFTWARE TESTING

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.



7.3. Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

1. WHITE BOX TESTING

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

2. BASIC PATH TESTING

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula:

$$V(G)=E-N+2 \text{ or}$$

$$V(G)=P+1 \text{ or}$$

$$V(G)=\text{Number Of Regions}$$

Where $V(G)$ is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

3. CONDITIONAL TESTING

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

4. DATA FLOW TESTING

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements.

5. LOOP TESTING

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

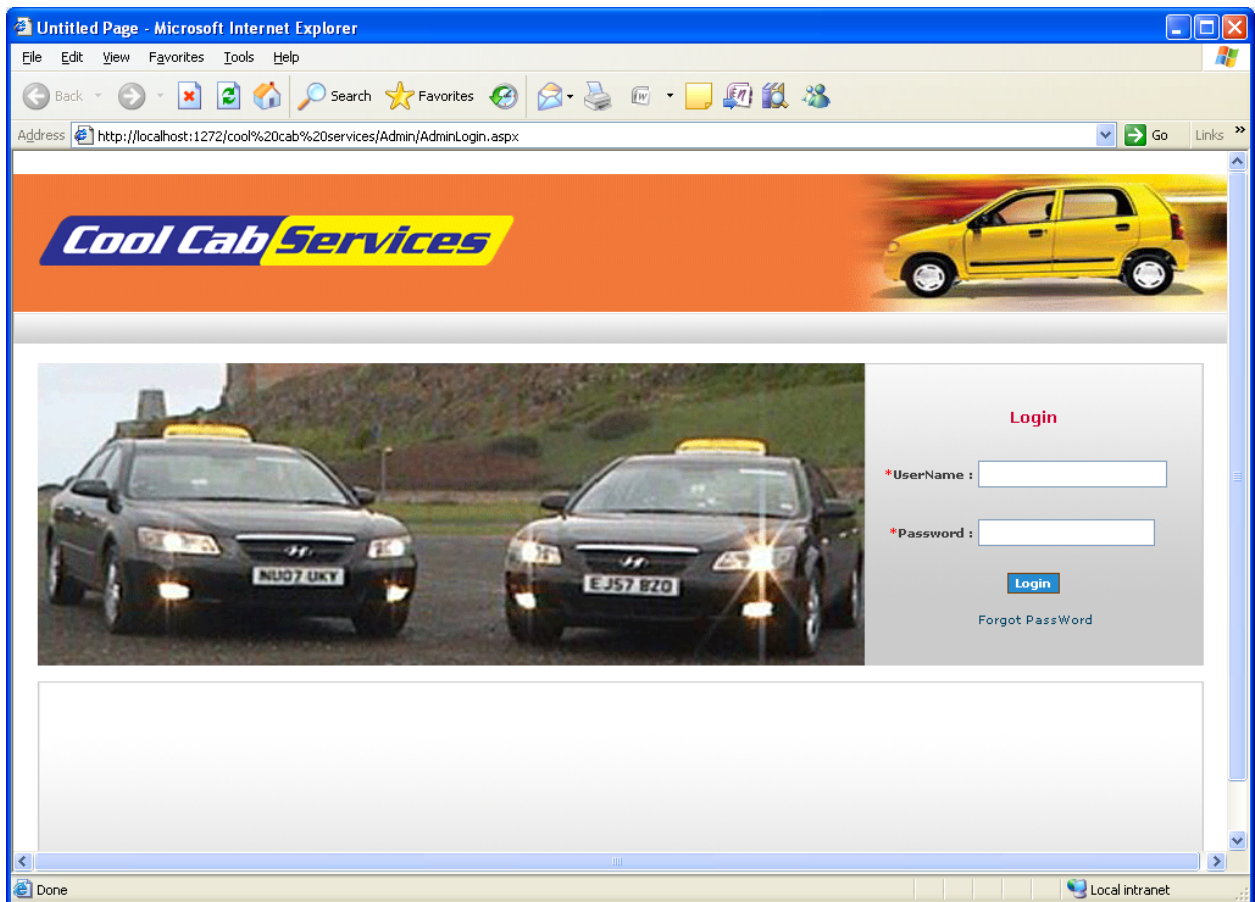
- All the loops were tested at their limits, just above them and just below them.
- All the loops were skipped at least once.
- For nested loops test the inner most loop first and then work outwards.
- For concatenated loops the values of dependent loops were set with the help of connected loop.
- Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Each unit has been separately tested by the development team itself and all the input have been validated.

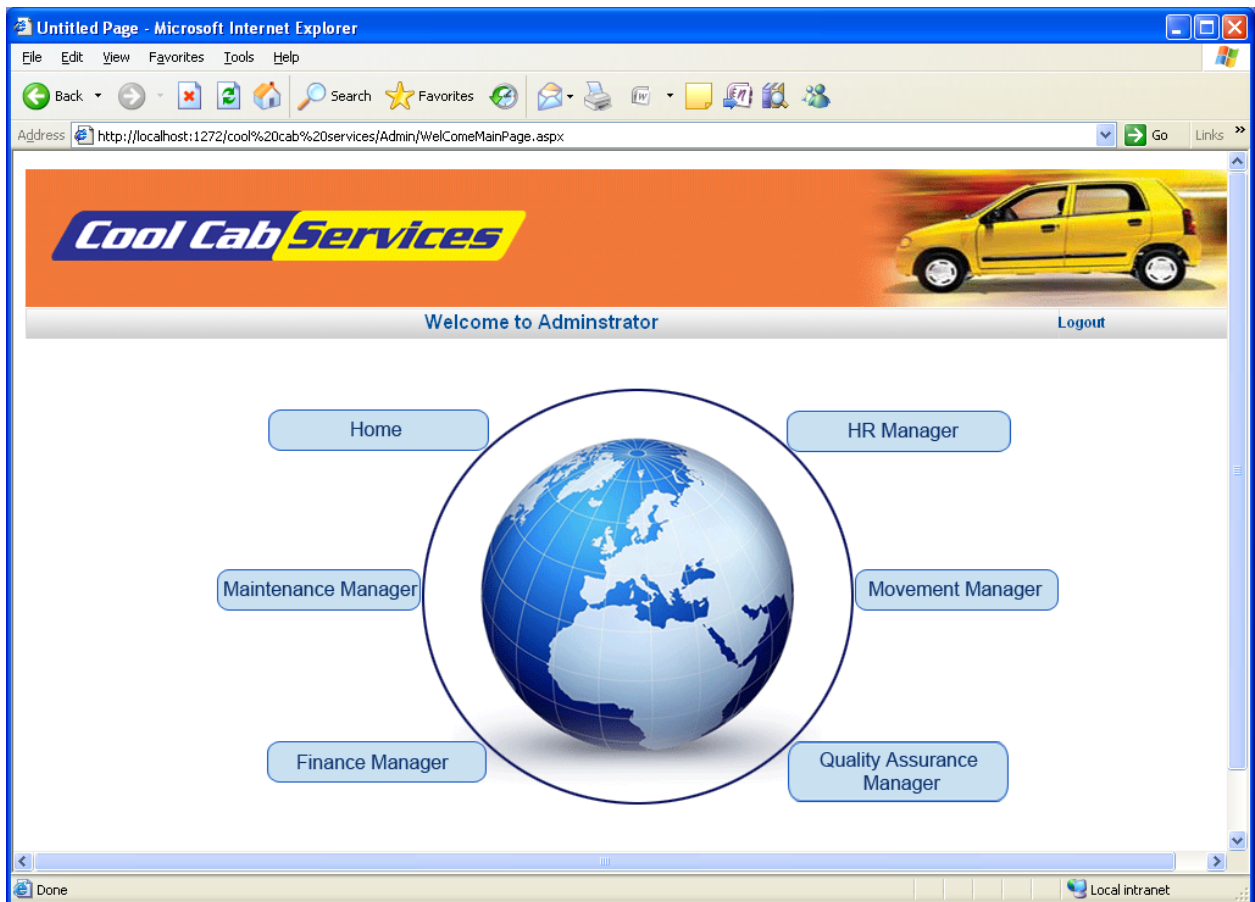
Chapter 8

Output Screens

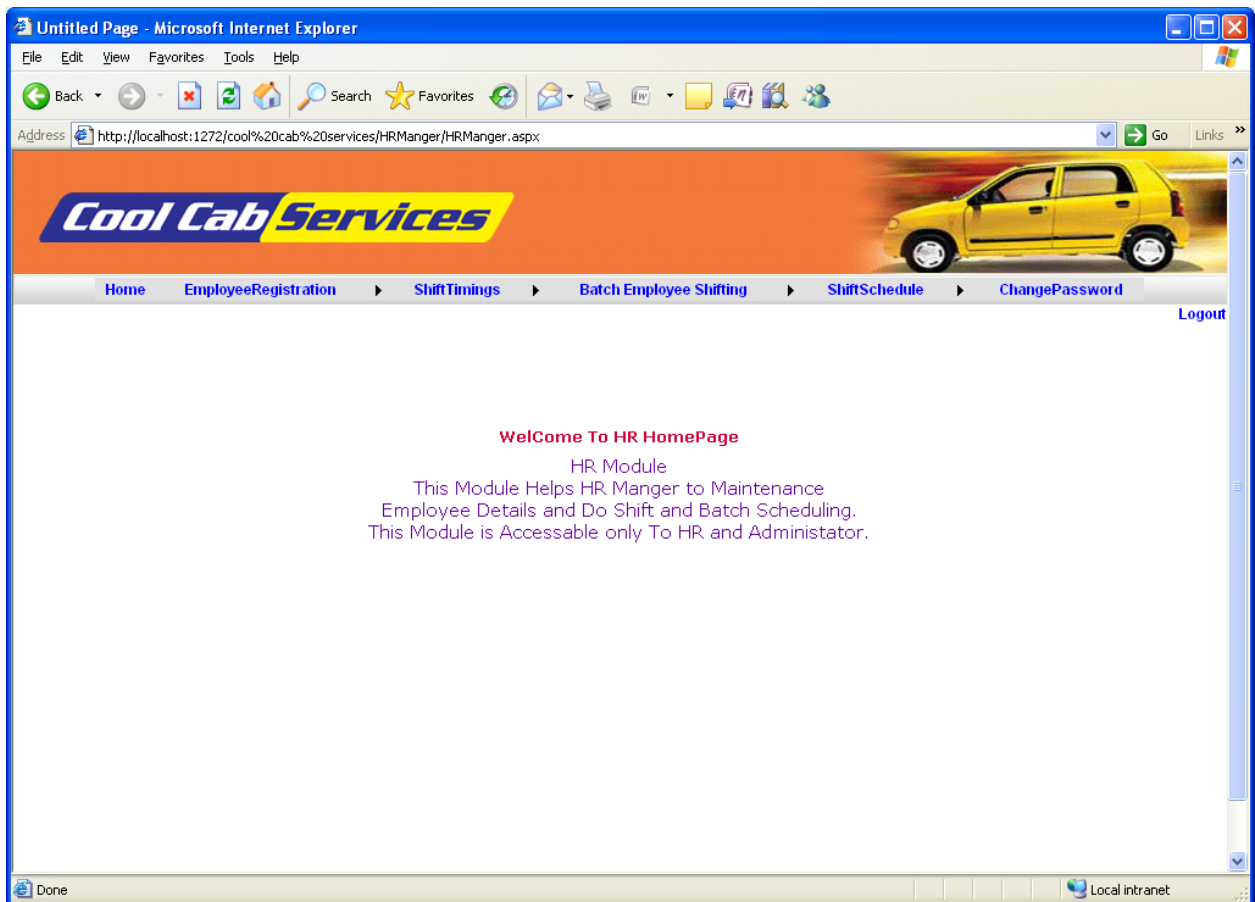
COOL CAB SERVICES



COOL CAB SERVICES



COOL CAB SERVICES



COOL CAB SERVICES

Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print Mail W PDF Word Excel PowerPoint People

Address <http://localhost:1272/cool%20cab%20services/HRManger/AddEmployeeDetails.aspx> Go Links

EmployeeDetails

*EmpName :

*PermanentAddress :

*CommunicationAddress :

*Qulification :

*DOB : 
mm/dd/yyyy

*Vehicle Require : Yes/No

*Gender : ☒ Male ☐ Female

* Phone NO :

*Designation :

*Department :

*DOJ : 
mm/dd/yyyy

*Status :

*Age :

*TimeSpan :

Select Emp Image :

Local intranet

COOL CAB SERVICES

Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Reload Home Search Favorites RSS Print Mail News Groups Feeds

Address http://localhost:1272/cool%20cab%20services/HRManger/ViewEmployeeDetails.aspx Go Links

Cool Cab Services

Home EmployeeRegistration ShiftTimings Batch Employee Shifting ShiftSchedule ChangePassword Logout

View EmployeeDetails

	EmpID	Name	Address	VehicleRequire	Gender	PhoneNo	Department	Designation	Status
Delete View Edit	EMP2	fgdgdg	fgdgd	dfg	dfgdg	53453	ertert	gdgdf	dfgd
Delete View Edit	EMP3	ertert	trtertert	ert	ertert	4545	dfgdgfd	terttert	ertert
Delete View Edit	EMP4	ertertertert	trtertert	ert	ertert	ertert	dfgdgfd	terttert	ertert
Delete View Edit	EMP5	ertertgthf	trtertert	ert	ertert	ertert	dfgdgfd	terttert	ertert
Delete View Edit	EMP6	asdads	trtertert	ert	ertert	ertert	dfgdgfd	terttert	ertert
Delete View Edit	EMP7	asdghjasds	trtertert	ert	ertert	ertert	dfgdgfd	terttert	ertert
Delete View Edit	EMP8	dfg	trtertert	ert	ertert	ertert	dfgdgfd	terttert	ertert
Delete View Edit	EMP9	ghfthf	ghf	fgh	fthghf	5345	terttert	hthghf	fgh
Delete View Edit	EMP12	Testdsdg	Test	Test	Male	4564	dfgdgfd	Test	Test
Delete View Edit	EMP16	gdgfd	dfgdgfd	gdgfdgfd	Male	345345	dgdfgd	dfgd	dfgd

1 2 3 4 5 6 7

Done Local intranet

COOL CAB SERVICES

Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print View Source View Images View All Links

Address http://localhost:1272/cool%20cab%20services/HRManger/ShiftTiming.aspx Go Links

Cool Cab Services

Home EmployeeRegistration ShiftTimings Batch Employee Shifting ShiftSchedule ChangePassword Logout

Shift Timing

*ShiftName:

*StartIng Time:

*Dispatch

*Total No Of Batches

Done Local intranet

COOL CAB SERVICES

Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Print Mail News RSS Feeds

Address http://localhost:1272/cool%20cab%20services/HRManger/ViewShiftDetails.aspx Go Links



Home EmployeeRegistration ShiftTimings Batch Employee Shifting ShiftSchedule ChangePassword Logout

View ShiftDetails

	ShiftID	ShiftName	StartingTime	DispatchTime	NoBatches
Delete Edit	Shift-ID1	rtert	ertetr	erte	tetretr
Delete Edit	Shift-ID10	dfghrter	10:30am		
Delete Edit	Shift-ID11	gf	10:45am		
Delete Edit	Shift-ID12	gfgd	10:45am		
Delete Edit	Shift-ID13	etert	ertert		
Delete Edit	Shift-ID14	cvbc	dfgdf	vbcvb	fgtdgd
Delete Edit	Shift-ID2	ertert	sdfs	erterte	tetretr
Delete Edit	Shift-ID3	gfrdf	sdfs	erterte	tetretr
Delete Edit	Shift-ID4	ertegtdfgwerwer	sdfs	erterte	434
Delete Edit	Shift-ID8	tdhgfh	10Am	3	546

1 2 3

Done Local intranet

COOL CAB SERVICES

Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print Mail W Word PDF Help People

Address <http://localhost:1272/cool%20cab%20services/HRManger/BatchDetails.aspx> Go Links



[Home](#) [EmployeeRegistration](#) [ShiftTimings](#) [Batch Employee Shifting](#) [ShiftSchedule](#) [ChangePassword](#) [Logout](#)

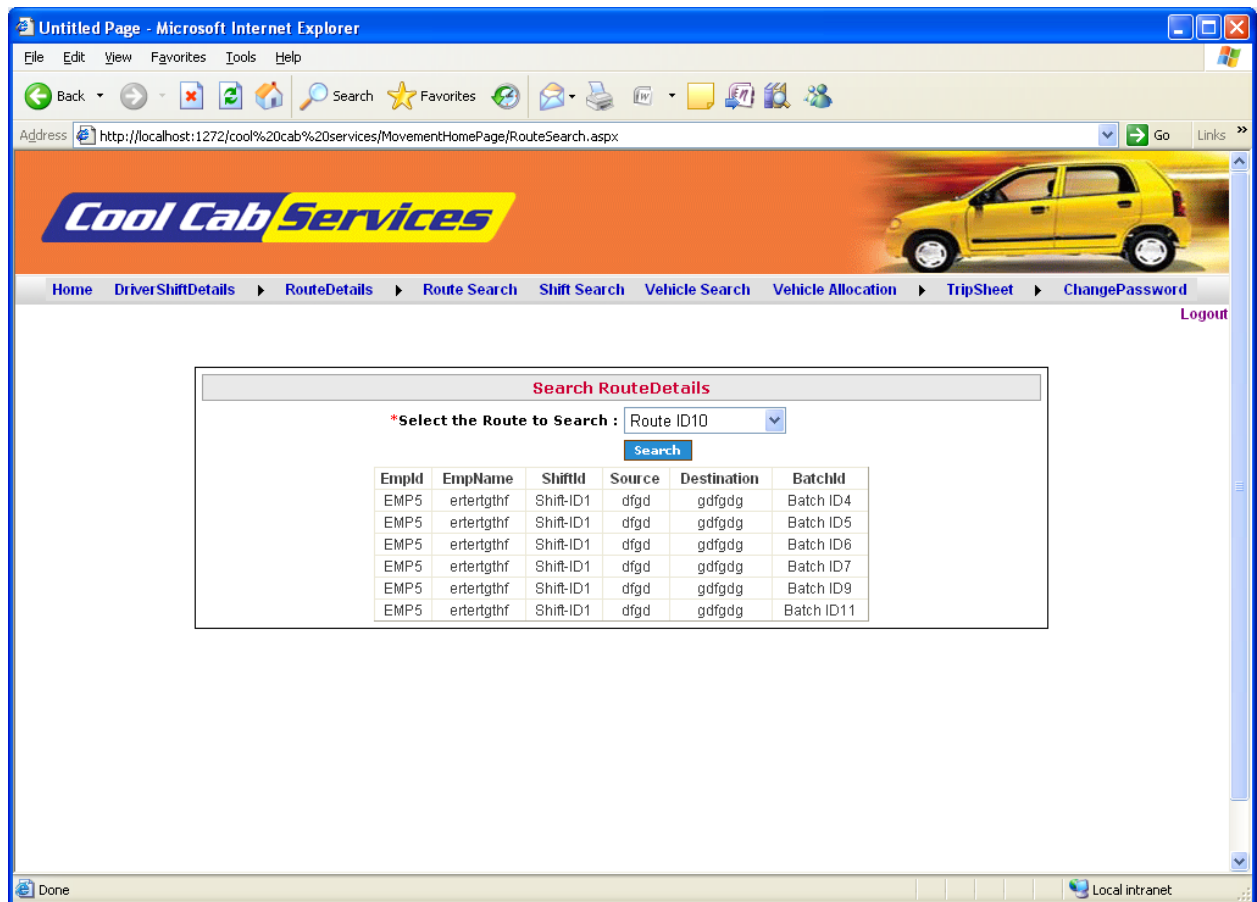
BatchDetails

Total No Of Employee

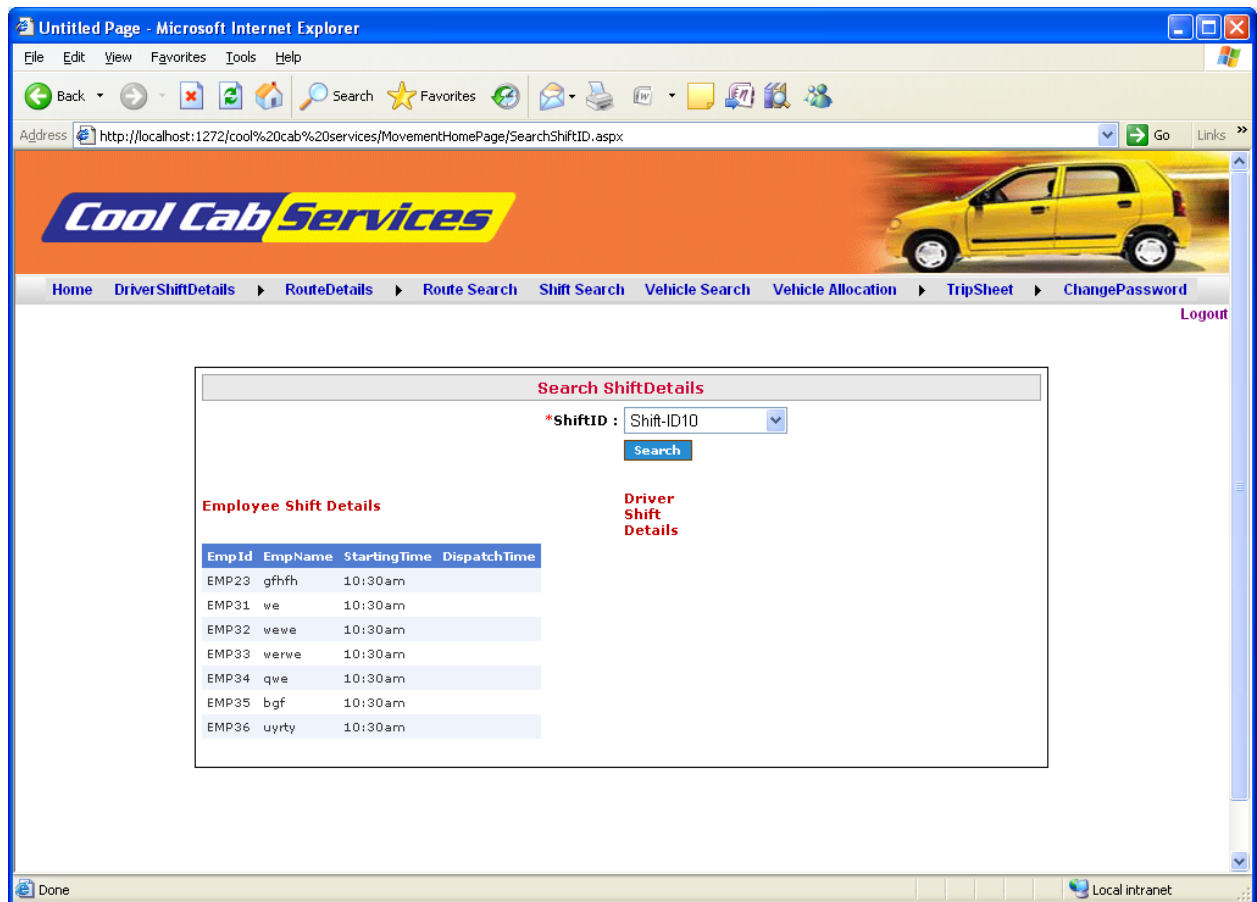
Shift ID

Done Local intranet

COOL CAB SERVICES



COOL CAB SERVICES



COOL CAB SERVICES

Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Reload Home Search Favorites Print Mail News RSS Feeds

Address http://localhost:1272/cool%20cab%20services/MovementHomePage/ViewVehicleAllocation.aspx Go Links

Cool Cab Services

Home DriverShiftDetails RouteDetails Route Search Shift Search Vehicle Search Vehicle Allocation TripSheet ChangePassword Logout

View VehicleAllocationDetails

	VehicleAllocationID	VehicleID	EmployeeID	DriverID	PickupDrop	RouteID	VDate
Delete Edit	VA ID1	Vehicle ID1	EMP1	DVRID1	Pickup	Route ID1	9/30/2008 12:00:00 AM
Delete Edit	VA ID10	Vehicle ID11	EMP3	DVR ID10	Pickup	Route ID10	9/23/2008 12:00:00 AM
Delete Edit	VA ID11	Vehicle ID15	EMP8	DVR ID11	Pickup	Route ID14	9/30/2008 12:00:00 AM
Delete Edit	VA ID2	Vehicle ID2	EMP1	DVRID1	Pickup	Route ID2	9/30/2008 12:00:00 AM
Delete Edit	VA ID3	Vehicle ID1	EMP2	DVRID2	Pickup	Route ID1	9/30/2008 12:00:00 AM
Delete Edit	VA ID4	Vehicle ID1	EMP2	DVR ID7	Pickup	Route ID4	9/30/2008 12:00:00 AM
Delete Edit	VA ID5	Vehicle ID1	EMP6	DVR ID9	Pickup	Route ID1	9/30/2008 12:00:00 AM
Delete Edit	VA ID6	Vehicle ID1	EMP1	DVR ID7	Pickup	Route ID1	9/30/2008 12:00:00 AM
Delete Edit	VA ID7	Vehicle ID5	EMP5	DVRID1	Pickup	Route ID3	9/24/2008 12:00:00 AM
Delete Edit	VA ID12	Vehicle ID7	EMP73	DVR ID22	Pickup	Route ID15	10/28/2008 12:00:00 AM

1 2

[Print](#)

Done Local intranet

COOL CAB SERVICES

Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print Mail Internet Options

Address http://localhost:1272/cool%20cab%20services/MovementHomePage/ViewTripSheet.aspx Go Links

Cool Cab Services

Home DriverShiftDetails RouteDetails Route Search Shift Search Vehicle Search Vehicle Allocation TripSheet ChangePassword Logout

View TripSheet Details

			Label				
	TripSheetID	AllocationID	VehicleID	RateKM	KM	TotalAmount	Remark
Delete Edit	TripSheet ID4	VA ID1	Vehicle ID1	232	4	928	gdfgdgdf
Delete Edit	TripSheet ID5	VA ID1	Vehicle ID1	4	4	16	gyhgfh
Delete Edit	TripSheet ID6	VA ID1	Vehicle ID1	454	3	1362	efgdf
Delete Edit	TripSheet ID7	VA ID2	Vehicle ID3	343	3	1029	fdg

Done Local intranet

COOL CAB SERVICES

Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Reload Home Search Favorites Print Mail News RSS Feeds

Address http://localhost:1272/cool%20cab%20services/FinancialHomePage/ViewVehicleBillingTransaction.aspx Go Links

Cool Cab Services

Home VehicleBilling VendorBilling ChangePassword Logout

View Vehicle Billing Transaction

	BID	BillNo	VehicleID	Amount	DateOfBilling	VenderID	Deduction	NetAmount
Delete Edit	1	BillNo1	VEhicle ID1	3453	9/1/2008 12:00:00 AM	VNDRID1	23	3430
Delete Edit	2	BillNo2	VEhicle ID1	2342	9/17/2008 12:00:00 AM	VNDRID1	232	2110
Delete Edit	3	BillNo3	VEhicle ID2	34234535	9/13/2008 12:00:00 AM	VNDR ID6	4321	34230225
Delete Edit	4	BillNo4	VEhicle ID1	232	9/3/2008 12:00:00 AM	VNDR ID10	323	23423
Delete Edit	5	BillNo5	VEhicle ID1	123424	9/2/2008 12:00:00 AM	VNDR ID10	123	424
Delete Edit	6	BillNo6	VEhicle ID4	2306	12/30/2008 12:00:00 AM	VNDR ID26	14756	12450

Done Local intranet

COOL CAB SERVICES

Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print Mail W PDF RSS Help

Address <http://localhost:1272/cool%20cab%20services/FinancialHomePage/Changepassword.aspx> Go Links

Cool Cab Services

[Home](#) [VehicleBilling](#) [VendorBilling](#) [ChangePassword](#) [Logout](#)

ChangePassword

*Oldpassword :

*NewPassword :

*Retype NewPassword :

Done Local intranet

COOL CAB SERVICES

Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Reload Home Search Favorites RSS Print Mail News Groups Feeds

Address http://localhost:1272/cool%20cab%20services/maintenanceManger/Vehicledetails.aspx Go Links

Cool Cab Services

Home DriverDetails VendorDetails **VehicleDetails** SparepartDetails SparePartBilling Changepassword Logout

VehicleDetails

VehicleName

Vendor ID :

Driver ID :

Vehicle Type :

Register No :

Rate/KM :

Capacity :

Route ID :

Done Local intranet

COOL CAB SERVICES

Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Reload Home Search Favorites RSS Print Mail News Groups Feeds

Address http://localhost:1272/cool%20cab%20services/maintenanceManger/ViewSparePartDetails.aspx Go Links

Cool Cab Services

Home DriverDetails VendorDetails VehicleDetails SparepartDetails SparePartBilling Changepassword Logout

SparePartDetails

	SparerPartID	DealerName	SparePartType	Quantity	SparePart	DateOfPurchase	Price	AmountPaid
Delete Edit	SRPT ID1	dfggh	dfgghjg	5	dghfjg	9/16/2008 12:00:00 AM	34534	172670
Delete Edit	SRPT ID2	ssdfs	fsdfsdf	34	dfsfs	9/15/2008 12:00:00 AM	322	10948
Delete Edit	SRPT ID5	Test	Test	3	Test	10/22/2008 12:00:00 AM	45	135
Delete Edit	SRPT ID6	Testi	Testdfid	3	Testifg	10/22/2008 12:00:00 AM	45	135
Delete Edit	SRPT ID7	Testi	xc	3	xc	10/22/2008 12:00:00 AM	45	135

Done Local intranet

COOL CAB SERVICES

Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print View Source

Address http://localhost:1272/cool%20cab%20services/maintenanceManger/SparePartsBilling.aspx Go Links

Cool Cab Services

Home DriverDetails VendorDetails VehicleDetails SparepartDetails SparePartBilling Changepassword Logout

SpareParts Billing

*VehicleID : Select...
*SpareType : Select...
*Quantity :
*BillDate : mm/dd/yyyy
*SparePart : Select...
*Price :
*TotalAmount :

Submit Clear

Done Local intranet

COOL CAB SERVICES

Untitled Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Reload Home Search Favorites RSS Feeds Print Mail News Groups

Address http://localhost:1272/cool%20cab%20services/maintenanceManger/ViewSparePartBilling.aspx Go Links

Cool Cab Services

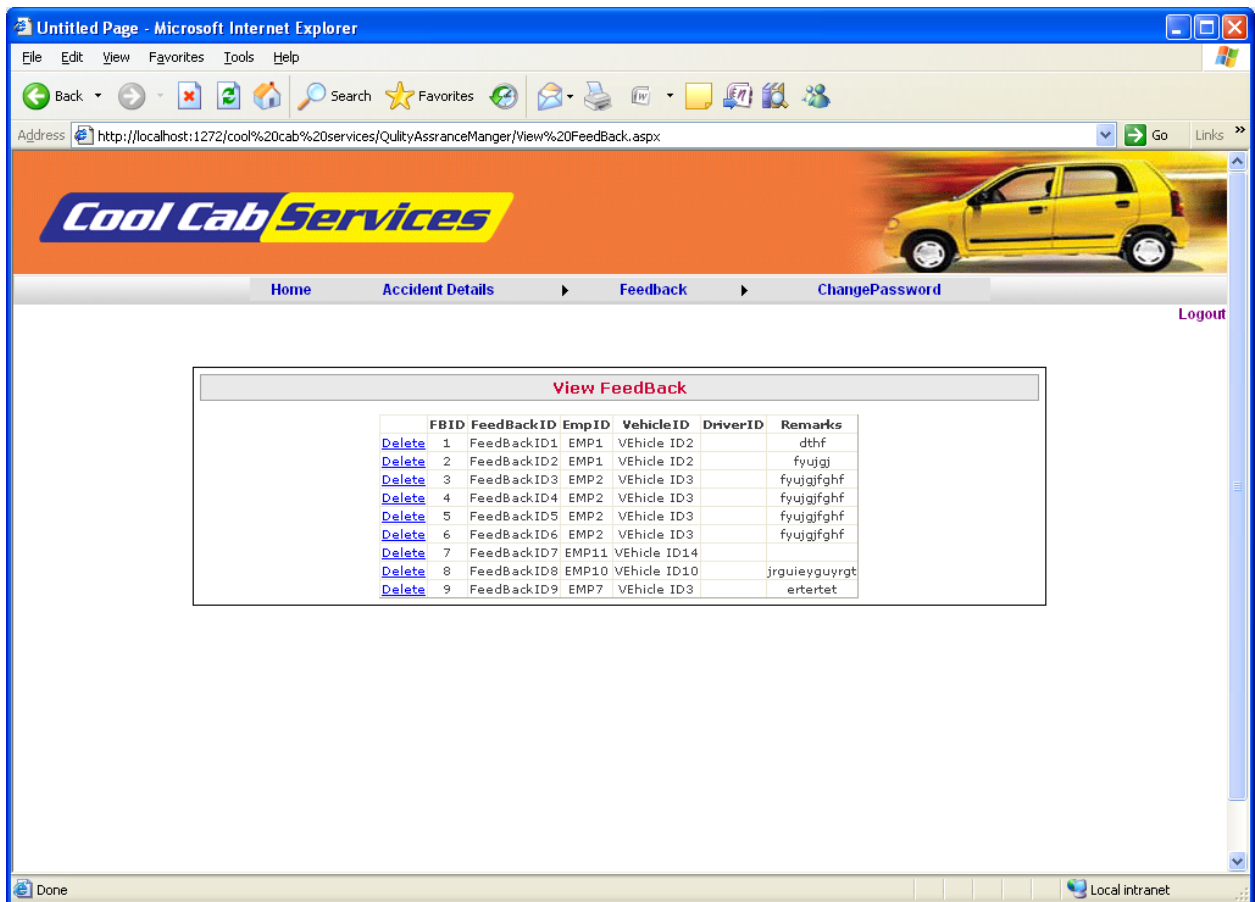
Home DriverDetails VendorDetails VehicleDetails SparepartDetails SparePartBilling Changepassword Logout

View SparePartsDetails

	BillNo	VehicleID	SpareType	Quantity	BillDate	SparePart	Price	TotalAmount
Delete Edit	Biilli ID1	VEhicle ID1	dfgghjg	23	10/28/2008 12:00:00 AM	dghhjg	23	529
Delete Edit	Biilli ID2	VEhicle ID2	fssdfsd	232	10/28/2008 12:00:00 AM	dfsfs	234	54288
Delete Edit	Biilli ID3	VEhicle ID4	fssdfsd	434	10/27/2008 12:00:00 AM	dghhjg	34	14756
Delete Edit	Biilli ID4	VEhicle ID1	dfgghjg	33	10/28/2008 12:00:00 AM	dfsfs	34534	1139622
Delete Edit	Biilli ID5	VEhicle ID3	Testdfd	434	10/29/2008 12:00:00 AM	dfsfs	45	54

Done Local intranet

COOL CAB SERVICES



Chapter 9

Conclusion

9.1 Limitations

Cool Cab Services is a Web application and it is restricted to only limited type of users. In this application ,Different types of managers have been given access rights and they are restricted up to their functionalities, so that the data is maintained securely and redundant data is prevented. As the Data is stored electronically, it is necessary to have a Computer and Network connection to access the Application. Here The Details of Employees and Drivers, cabs are maintained but accounts to these people are not created. using this application mangers do assign or update the batch,shift of cabs to drivers and employees. But employees are unable to view their details .

9.2 Future Enhancements

Every Edition of an book comes with new topics and modifications if any errors are present. In the similar way, in near future ,our application will overcome the flaws if occurred, and attains new features offered to employees for the Flexible and easy Transportation. Following are the Enhancements to the application.

- ☐ Providing Good User Interface.
- ☐ Providing access permissions to the employees
- ☐ Try to Implement the GPS system in the Cabs.

9.3 BIBLIOGRAPHY

- **FOR .NET INSTALLATION**

www.support.microsoft.com

- **FOR DEPLOYMENT AND PACKING ON SERVER**

www.developer.com

www.15seconds.com

- **FOR SQL**

www.msdn.microsoft.com

- **FOR ASP.NET**

www.msdn.microsoft.com/net/quickstart/aspplus/default.com

www.asp.net

www.fmexpense.com/quickstart/aspplus/default.com

www.asptoday.com

www.aspfree.com

www.4guysfromrolla.com/index.aspx