CABWAYS

Introduction

CABWAYS, The ideal solution for Fleet Management, is specially designed for Cab-Operators who operate a fleet of vehicles for hire. The vehicles maybe Cars, pickup vans, buses, etc. In an era where having a website of your own is a matter of prestige for some and necessity for others. Management of these websites is still an area which is ignored or done in traditional manner. In a changing world no corporate company or any website owner would like to host a static content over their site. We in a system called ‘ **CABWAYS** ’ provide a structural mechanism of building the content for your site. It also provides us a testing area where the created content is being hosted.

The structural mechanism provides in this system requires personal with specialization in different areas. The entire content is grilled through various phases before being made available for the website. The content being created or hosted can be placed into different categories available with the system. The crucial content can even be placed in a protected zone which can be accessed by only the registered users.

This application provides most of the features required to manage the projects developed in a software development firm.

This volume presents the manner in which the software was developed and how the various problems are tackled at the different levels to convince the user.

We hope that this package would prove to be an excellent environment for simpler for end user.

CABWAYS, The ideal solution for Fleet Management, is specially designed for Cab-Operators who operate a fleet of vehicles for hire. The vehicles maybe Cars, pickup vans, buses, etc.**CABWAYS** enables you to manage demand-supply position of vehicles, keep a tab on the mileage and reduce maintenance expenditure and operating costs. It automates your entire front-office operations like accepting customer bookings, allocation of vehicles & drivers prepare customer invoices. The system also takes care of all back-office operations like capturing of maintenance costs incurred, fuel purchase, accident records, service log etc. Admin Module handles the entire Master forms such as Company information, Employee information User allocation, change password, Vehicle type, vehicle tariff etc.

Objective

CABWAYS, The ideal solution for Fleet Management, is specially designed for Cab-Operators who operate a fleet of vehicles for hire. The vehicles maybe Cars, pickup vans, buses, etc. **CABWAYS** enables you to manage demand-supply position of vehicles, keep a tab on the mileage and reduce maintenance expenditure and operating costs. It automates your entire front-office operations like accepting customer bookings, allocation of vehicles & drivers prepare customer invoices. The system also takes care of all back-office operations like capturing of maintenance costs incurred, fuel purchase, accident records, service log etc. Admin Module handles the entire Master forms such as Company information, Employee information User allocation, change password, Vehicle type, vehicle tariff etc.,

.

**Existing System :**

Typically, the role of a Cab Operator is to Provide Vehicle on Hire from one place to another. He uses his own fleet of vehicles. Client is a Cab Operator to manage Demand – Supply position of vehicles on line, keep a tab on the Mileage, reduce maintenance expenditure and Operating costs. He maintains entire front office operations like customer details, Customer bookings, Vehicle Allocation, Vehicle maintenance Employee details etc., manually. At present all the business process is maintained by manually by using Bills, Vouchers, Ledgers, Books. So the total business process to be automated and computerized.

**Proposed Automation**

The main purpose of proposed system is essentially to automate entire operations and Information of Vehicles at a particular given point of time. So that the Cab Operator and Their Customer can get best and fast Information about Their Vehicles and drivers at any Point of Time. He needs to exercise tight control on the operations and input costs if he is to provide good service at reasonable rates and yet remain profitable.

# SCOPE

The scope of the product to manage the front End & Back Office functions of a CAB Operator consists of the following key modules. It is to be noted that all the modules may not be applicable to all transporters.

* Front-Office Management
* Vehicle Maintenance Management
* MIS Reports

Tools and Platform

**HARDWARE REQUIREMENTS :**

Processor **:** Standard processor with a speed of 2.4 GHz

RAM **: 2** GB RAM

Hard Disk **:** 80 GB

Monitor **:** Standard color monitor

Keyboard **:** Standard keyboard

Mouse **:** Standard mouse

**SOFTWARE REQUIREMENTS :**

Operating System **:** MS – Windows server 2000/XP/NT/2007/2008

Languages **:** Asp.Net with C# (Dotnet 2008)

Database System **:** MS-SQL Server 2008

Web Server **:** MS IIS 5.0

Documentation Tool **:** MS - Word 2007/08/XP

The “**CABWAYS**”is an online application which is based on Object Oriented System. Object modeling is useful for designing computer systems, whether those systems are to be implemented in object-oriented languages or not. Most designs are likely to need more than an object-oriented language, such as a database. Object modeling also has a use outside of the design of computer systems. It is an excellent analysis method, and it can be used in business process reengineering, in social science research, or any complex environment where it is important to capture the structure and functionality of some world. The Object Oriented Paradigm provides you with:

A simple, clear, analysis and design notation.

A method for construction of analyses and designs.

An object-oriented database management system (OODBMS) sometimes shortened to ODBMS for object database management system), is a database management system (DBMS) that supports the modeling and creation of data as objects. This includes some kind of support for classes of objects and the inheritance of class properties and methods by subclasses and their objects. There is currently no widely agreed-upon standard for what constitutes an OODBMS, and OODBMS products are considered to be still in their infancy. In the meantime, the object-relational database management system (ORDBMS), the idea that object-oriented database concepts can be superimposed on relational databases, is more commonly encountered in available products. An object-oriented database interface standard is being developed by an industry group, the Object Data Management Group (ODMG). The Object Management Group (OMG) has already standardized an object-oriented data brokering interface between systems in a network.

**Front-End (C# .Net):**

ASP.NET, also known as ASP+, is an enhanced version of ASP for the form. It supports executable programs compiled from Visual Basic, C#, C++ and other languages and is not backward compatible with regular ASP code. ASP.NET pages are always compiled rather than interpreted as are ASP pages.

C# is very popular for its friendly working (graphical) environment. And also C# is principal designer and lead architect at Microsoft is Anders Hejlsberg, who was previously involved with the design of Turbo Pascal Embarcadero Delphi, and Visual++. In interviews and technical papers he has stated that flaws] in most major programming languages are C++, Java Delphi, and Smalltalk drove the fundamentals of the Common Language Runtime , which, in turn, drove the design of the C# language itself. It is understand that "Java and C# are almost identical programming languages. Boring repetition that lacks innovation," "Hardly anybody will claim that Java or C# are revolutionary programming languages that changed the way we write programs," and "C# borrowed a lot from Java - and vice versa. Now that C# supports boxing and unblocking, we'll have a very similar feature in Java. Some notable distinguishing features of C# are:

* There are no global variables or functions. All methods and members must be declared within classes. Static members of public classes can substitute for global variables and functions.
* Local variables cannot shadow variables of the enclosing block, unlike C and C++. Variable shadowing is often considered confusing by C++ texts.
* C# supports a strict Boolean database
* In C#, memory address pointers can only be used within blocks specifically marked as unsafe, and programs with unsafe code need appropriate permissions to run.
* Managed memory cannot be explicitly freed; instead, it is automatically garbage collected.

***RDBMS/BACK END: -***

SQL Server 2005 was released in November 2005. The 2005 product is said to provide enhanced flexibility, scalability, reliability, and security to database applications, and to make them easier to create and deploy, thus reducing the complexity and tedium involved in database management. SQL Server 2005 also includes more administrative support. SQL Server 2005 added some extensions to the [T-SQL](http://en.wikipedia.org/wiki/T-SQL) language to allow embedding Query queries in T-SQL. In addition, it also defines a new extension to Query, called XML DML that allows query-based modifications to XML data. SQL Server 2005 also allows a database server to be exposed over [web services](http://en.wikipedia.org/wiki/Web_service) using TDS packets encapsulated within [SOAP (protocol)](http://en.wikipedia.org/wiki/SOAP_(protocol)) requests. When the data is accessed over web services, results are returned as XML.

For relational data, [T-SQL](http://en.wikipedia.org/wiki/T-SQL) has been augmented with error handling features and support for recursive queries. SQL Server 2005 has also been enhanced with new indexing algorithms and better error recovery systems. Data pages are [check summed](http://en.wikipedia.org/wiki/Checksum) for better error resiliency, and optimistic concurrency support has been added for better performance. Permissions and access control have been made more granular and the query processor handles concurrent execution of queries in a more efficient way. Partitions on tables and indexes are supported natively, so scaling out a database onto a [cluster](http://en.wikipedia.org/wiki/Cluster_(computing)) is easier. SQL CLR was introduced with SQL Server 2005 to let it integrate with the [.NET Framework](http://en.wikipedia.org/wiki/.NET_Framework). SQL Server 2005 introduced "MARS" (Multiple Active Results Sets), a method of allowing usage of database connections for multiple purposes.

**Operating System (MS Windows-XP Professional):**

With Windows XP Service Pack 2 (SP2), Microsoft is introducing a set of security technologies that will help improve Windows XP-based computers' ability to withstand malicious attacks from viruses and worms. These technologies include:

* Network protection and Memory protection
* Improved email security and Safer browsing

Together, these security technologies will help make it more difficult to attack Windows XP, even if the latest patches or updates aren't applied. These security technologies together are particularly useful mitigation against worms and viruses. To developers these technologies will have impacts on the applications that they create and the tools they use. This page contains resources to assist developers in dealing with these impacts. Windows XP Service Pack 2 includes an improved firewall that is enabled by default. This will provide increased security for your system but can also prevent normal operation of some software commonly used at Stanford such as PC-Leland and Big fix. This page will assist you in configuring the Windows Firewall to work with applications that require a port to be opened.

Analysis

System analysis will be performed to determine if it is feasible to design information based on policies and plans of the organization and on user requirements and to eliminate the weaknesses of the present system.

* The new system should be cost effective.
* To augment management, improve productivity and services.
* To enhance user / system interface.
* To improve information qualify and usability.
* To upgrade systems reliability, availability, flexibility and growth potential.

**Need for Developing**

The main purpose of proposed system is essentially to automate entire operations and Information of Vehicles at a particular given point of time. So that the Cab Operator and Their Customer can get best and fast Information about Their Vehicles and drivers at any Point of Time. He needs to exercise tight control on the operations and input costs if he is to provide good service at reasonable rates and yet remain profitable.

## Further Drawbacks of the Existing System:

The following are the drawbacks of the existing manual System.

**Time Delay:**

In the existing system, information related to all transactions is stored in different registers. Since all the transactions are stored in different registers it takes lot of time to prepare different reports.

**Redundancy:**

As the information passes through different registers, each register is consolidated and sent to next register. So the same information is being tabulated at each register, which involves lot of complication and duplication in work, thus it causes redundancy.

**Accuracy:**

Since the same data is compiled at different sections, the Possibility of tabulating data wrongly increases. Also if the data is more, validations become difficult. This may result in loss of accuracy of data.

**Information Retrieval:**

As the information is stored in the particular Format, it can only be retrieved in the same format. But if it is to be retrieve in different format, it is not possible.

**Storage Media:**

In the existing system, data transaction being stored on too long registers it is very difficult to refer after some time.

**Reports:**

At the various reports are tabulated manually. They are not such Attractive and require more time. They do not provide adequate help in maintaining the accounts.

**Enquiry:**

Enquiry for different level of information is much more difficult. On line enquiry of data is not possible.

System Requirement Specification (SRS)

## Introduction

### 1.1. Purpose

### Scope

### Definitions

### 1.4. References

### 1.5. Overview

## 2. Overall Description.

### 2.1. Project perspective

### Project functions

### 2.3. User characteristics

### 2.4. Constraints

### 2.5. Assumptions and Dependencies

## Specific Requirement

**3.1 External interface requirements**

**3.1.1. User interfaces**

**3.1.2. Hard ware interfaces**

* + 1. **Software interfaces**
    2. **Communication interfaces**

## *Introduction*:

### Purpose

**CABWAYS** has been specifically designed keeping in mind the requirements of a cab operator. A cab operator operates in a service industry, which requires high standards of efficiency to be successful. A cab operator provides vehicles on hire to customers and bills them according to the tariff card. The key success factors in this business are quality of service, condition of the vehicle, competitive pricing, customer-interface etc. A cab operator has to keep tight control over these factors to be competitive and remain profitable.

**1.2 Scope**

**The scope of work would include:**

The scope of the product to manage the front End & Back Office functions of a CAB Operator consists of the following key modules. It is to be noted that all the modules may not be applicable to all transporters.

* Front – Office Management
* Vehicle Maintenance Management
* Administration Module
* MIS Reports

## 

## 1.3. *Definitions*

**CABWAYS :** CP

**Customer** : Any Person / persons / institution / Organization which is interested to Hire the Vehicle .

* 1. **References**

1.IEEE Documents for SRS

## *Overview* :

CABWAYS, The ideal solution for cab Management , is specially designed for Cab-Operators who operate a cab of vehicles for hire. The vehicles maybe Cars, pickup vans, buses, etc. **CABWAYS** enables you to manage demand-supply position of vehicles, keep a tab on the mileage, reduce maintenance expenditure and operating costs. It automates your entire front-office operations like accepting customer bookings, allocation of vehicles & drivers prepare customer invoices. The system also takes care of all back-office operations like capturing of maintenance costs incurred, fuel purchase, accident records, service log etc. Admin Module handles the entire Master forms such as Company information, Employee information User allocation, change password, Vehicle type , vehicle tariff, This module also helps to configure such as Financial year configuration etc.,

You can now be better organized and efficiently equipped to manage a small or large cab of vehicles with up-to-date analytical information on Vehicle Bookings, Vehicle Allocation, Customer details, fuel issues, fuel consumption, mileage efficiency, maintenance expenditure, service log, etc

## Over all *description*:

### 2.1 Project perspective

a). **System Interfaces**:

The application enables users to login directly to Web site / Intranet from their desktop using a conventional Web browser.

CP- Cab system is an Internet / Intranet Application.

CP – Cab is an independent, totally self-contained application runs web server like IIS and on Windows Operating System. This web enabled application works with browser like Internet explorer 5.0 and above at client side.

CP – Cab will be based on Internet architecture and the layout of the application is explained in the following diagram.

User 1

Web

Server

IIS 6.0

Database

Server

(SQL Server 7)

B

User 2

B

User n

B

**Architecture of CABWAYS**

## A,User: Who ever uses the system

## B. Web Server: The Business Logic / Validations / Security of the CABWAYS will be implemented here.

## C. Database Server: The information of the CABWAYS will be stored in the Database Server. It can also be called as Data Services (SQL Server 2008).

**b). User interfaces:**

## *This section describes the application with the help of word document proto type screen flow design.*

## *The next few screens explain the navigation of each and every page of CABWAYS.*

**c). Hardware Interfaces**

This application is purely Internet Application, which runs through HTTP Protocol.

**d). Software Interfaces**

1. Application installed on Web server. Clients can access server through any client by using WEB browser like Internet explorer 5.0 and above.
2. Client network should support any network protocol like TCP/IP.

e). Communication interfaces:

For the establishment of Internet following Communication protocols are necessary.

1. TCP/IP protocol.
2. HTTP protocol.

f). Memory constraints:

There is no constraints on memory related issues. Normal server and client

Configuration is sufficient to run this application.

***2.2* ProjectFunctions** :

#### 2.2.1) Front Office Automation:

CABWAYS comes in a plug and play environment supporting two modules – Operations and Maintenance . The Operations module automates the front-office activities – customer bookings, allocation / Duty Slip of vehicles and customer billing. The system captures all-important details related to these activities for further MIS analysis.

* Maintaining the Party / Customer details
* Vehicle Booking Details
* Vehicle Allocation
* Vehicle Billing

## 2.2.2) *Maintenance Module:*

**Free Yourself from Maintenance Headaches**

The maintenance module allows you to capture the maintenance costs incurred on a vehicle on account of spares & labor. The maintenance can be classified against a trip or service or breakdown.. It gives you analytical information on the average maintenance cost, mean time & mean-mileage between failures. You can also get a detailed report on each maintenance activity, and it can also tell you when the next service is due and when a particular document is supposed to be renewed.

The Service Log is updated in two instances – once when the vehicle is being sent to the service station and next when it returns back to garage after service.

The Entry/ Edit tab is presented by default to enter the service particulars. The History tab enables you to view the service history details of the vehicles in the cab. The View tab shows the existing service records for editing pre-service details, update post-service details or cancellation.

* Maintaining The Accident Details Vehicle wise
* Managing the Services and breakdown of Vehicle
* Maintain the service Log of the vehicle
* Alerts for Maintenance

## *2.2.3) Administration Module:*

Admin Module handles the entire Master forms such as Company information, Employee information

User allocation, change password, Vehicle type , vehicle tariff, This module also helps to configure such as Financial year configuration etc.,

* Maintaining the Details of the Company
* Vehicle Type
* Vehicle Master
* Vehicle Tariff
* Employee Master
* User Master

**2.2.6). MIS Reports Module:**

The Reporting system is used for the management purpose, which easies the use of the application. For Management the total information will be available on finger tips. Reporting module consists of reporting by Vehicle availability, vehicle allocation, Vehicle Maintenance, Customer details, Customer bookings. Management reports (MIS) will be developed in crystal reports which can be displayed as reports given an option to convert that reports to Microsoft Office Word, Microsoft Office Excel, PDF formats.

This data can be used for management to take crucial decisions**.**

**2.3 User characteristics:**

**Intended Users:**

Only Registered or employees of the co., can browse the site. Users have to enter a userid and password to login. User should have a minimum awareness of web applications like how to navigate with browser and down loading.

## *2.4. Constraints:*

As it is a web application on Internet it won’t have any constraints, regarding Hardware bottlenecks, Network traffic, regulatory policies, safety and security of this application.

## 2.5 Assumptions and dependencies:

## *This application assumes the following:*

# 

# 3.Specific Requirements:

## 3.1) External interface requirements:

**Hard ware environment:**

Applicationis being implemented in the following Internet environment:

|  |  |
| --- | --- |
| Client systems | Any system that support Internet Browsing using any browser with a minimum 2 GB RAM and 200MHz clock speed  Preferable client system – Pentium or higher with 2 GB RAM running windows NT/2007/08/XP |
| Network | TCP/IP,HTTP |

### Development Environment

|  |  |
| --- | --- |
| Client Systems | PC’s running Windows XP/2000/NT/2007/08 |
| Server System | IIS4.0, Tomcat Server 8.1 |
| Network | TCP/IP,HTTP |
| Soft ware tools used | ASP.net , C# net , Sql server 2008 |
| Project Documentation | MS Word 2007 |

FEASIBILITY STUDY

Feasibility Studyis a high level capsule version of the entire process intended to answer a number of questions like: What is the problem? Is there any feasible solution to the given problem? Is the problem even worth solving? Feasibility study is conducted once the problem clearly understood. Feasibility study is necessary to determine that the proposed system is Feasible by considering the technical, Operational, and Economical factors. By having a detailed feasibility study the management will have a clear-cut view of the proposed system.

The following feasibilities are considered for the project in order to ensure that the project is variable and it does not have any major obstructions. Feasibility study encompasses the following things:

* Technical Feasibility
* Economical Feasibility
* Operational Feasibility

In this phase, we study the feasibility of all proposed systems, and pick the best feasible solution for the problem. The feasibility is studied based on three main factors as follows.

**Technical Feasibility:**

In this step, we verify whether the proposed systems are technically feasible or not. i.e., all the technologies required to develop the system are available readily or not.

Technical Feasibility determines whether the organization has the technology and skills necessary to carryout the project and how this should be obtained. The system can be feasible because of the following grounds.

* All necessary technology exists to develop the system.
* This system is too flexible and it can be expanded further.
* This system can give guarantees of accuracy, ease of use, reliability and the data security.
* This system can give instant response to inquire.

Our project is technically feasible because, all the technology needed for our project is readily available.

**Operating System :**  MS – Windows server 2003/XP/NT

**Languages :** Asp.Net with C# (Dotnet 2008)

**Database System :** MS-SQL Server 2000

**Web Server :**  MS IIS 5.0

**Documentation Tool :**  MS - Word 2003

**Economical Feasibility:**

In this step, we verify which proposal is more economical. We compare the financial benefits of the new system with the investment. The new system is economically feasible only when the financial benefits are more than the investments and expenditure. Economical Feasibility determines whether the project goal can be within the resource limits allocated to it or not. It must determine whether it is worthwhile to process with the entire project or whether the benefits obtained from the new system are not worth the costs. Financial benefits must be equal or exceed the costs. In this issue, we should consider:

* The cost to conduct a full system investigation.
* The cost of h/w and s/w for the class of application being considered.
* The development tool.
* The cost of maintenance etc.,

Our project is economically feasible because the cost of development is very minimal when compared to financial benefits of the application.

**Operational Feasibility:**

In this step, we verify different operational factors of the proposed systems like man-power, time etc., whichever solution uses less operational resources, is the best operationally feasible solution. The solution should also be operationally possible to implement. Operational Feasibilitydetermines if the proposed system satisfied user objectives could be fitted into the current system operation.

* The methods of processing and presentation are completely accepted by the clients since they can meet all user requirements.
* The clients have been involved in the planning and development of the system.
* The proposed system will not cause any problem under any circumstances.

Our project is operationally feasible because the time requirements and personnel requirements are satisfied. We are a team of four members and we worked on this project for three working months.

Entity Relationship Diagram

The Entity-Relationship model is a data model for high-level descriptions of conceptual data models, and it provides a graphical notation for representing such data models in the form of entity-relationship diagrams. Such data models are typically used in the first stage of information-system design; they are used, for example, to describe information needs and/or the type of information that is to be stored in the database during the requirements analysis.

*Basic elements in ER models:*

**Entities**

A data entity is anything real or abstract about which we want to store the data. Entity type falls into five classes: roles, events, location, tangible things or concepts.

**Attributes**

A data attribute is a characteristic common to all or most instances of a particular entity. An attribute or combination of attributes that uniquely identifies one & only one instance of an entity is called primary key or identifier.

**Relationships**

Relationship provides the structure needed to draw information from multiple entities. It is a natural association that exists between one or more entities.

**Cardinality**

Cardinality defines the number of occurrences of one entity for a single occurrence of related entity. Cardinality can be 1 to 1, 1 to many or vice versa and many to many.

**Symbols**

**Entity**

**Relation**

**Attribute**

**Key Attribute**

**Symbols used in ERD**

Passenger\_detail

Login\_detail

Employee\_detail

Agency\_detail

Starts

Owns

Has

Paying

Payment\_detail

Bank\_detail

deposited

Gps\_detail

Provide

Station\_detail

Works

Stores

Mechanics \_detail

Contain

Cab\_detail

Pay to

**E-R Diagram for CABWAYS**

1

1

1

1

1

M

M

N

1

M

M

N

1

M

1

1

1

M

Unified Modeling Language (UML)

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

* + User Model View
    1. This view represents the system from the users perspective.
    2. The analysis representation describes a usage scenario from the end-users perspective.
  + Structural model view
    1. In this model the data and functionality are arrived from inside the system.
    2. This model view models the static structures.
* Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

* Implementation Model View

In this the structural and behavioral as parts of the system are represented as they are to be built.

* Environmental Model View

In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are:

* UML Analysis modeling, this focuses on the user model and structural model views of the system.
* UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

Use case Diagrams represent the functionality of the system from a user’s point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.

Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer …etc., or another system like central database.

USECASE DIAGRAM









**Activity DIAGRAM:**



**COLLABRATION DIAGRAM:**

Collaboration Diagram for User Registration

Login

🚺 <- - - - - - - - - - - - - - - - - - - - - -

Login

Register User

Data Base

Administrator Invalid UserId

Logout

Logout

Success

Logout

Register

User Registration User

Successful

Collaboration Diagram for Change Password

Login

🚺 <- - - - - - - - - - - - - - - - - - - - - -

Login

Data Base

Log Out

Administrator Invalid UserId

Logout

Logout

Success

Change Password

Database

Password Update

Changed

**Sequence diagram for user registration:**

🚺

Logout

Login

Register

User database

Users

Login

Register

Invalid

Login

Update Details

Register

Failed

Registration Successful

Logout

Logout Successful

**Sequence diagram for Booking Vehicle:**

🚺

Booking Vehicle

Logout

Login

User database

Users

Login

Vehicle Booking

Invalid

Login

Vehicle Book

Failed

Vehicle Booking Successful

Logout

Logout Successful

Data Flow Diagram

A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an [information system](http://en.wikipedia.org/wiki/Information_system), modeling its *process* aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated.[[2]](http://en.wikipedia.org/wiki/Data_flow_diagram#cite_note-2) DFDs can also be used for the [visualization](http://en.wikipedia.org/wiki/Data_visualization) of [data processing](http://en.wikipedia.org/wiki/Data_processing) (structured design). A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a [flowchart](http://en.wikipedia.org/wiki/Flowchart))

*DFD Notation Used:*

External Entities/Terminator

These lie outside of the system being modelled. Terminators represent where the information comes from & where it goes. In designing a system, we have no idea about what these terminators do or how they do it. Processes modify the inputs in the process of generating the outputs.

Data Stores

Represent a place in the process where data comes to rest. A DFD does not say anything about the relative timing of the processes, so a data store might be a place to accumulate data over a year for the annual accounting process.

Data Flows

Represents how data moves between terminators, processes, & data stores(Those that cross the system boundary are known as IO or Input Output Description).

**Symbols**

Rectangle, which defines or destination.

Arrow, which shows dataflow.

Circle, which represent a process that transforms

Open rectangle, which shows a data store

**Symbols used in DFD**

Cabs Booking and Tracking

Associates

Employee

Passenger

Agency

Bank

GPS Device

**Context free Diagram for CABWAYS**

Registration

Reg\_Info

Info

1.0

Agency

Registration\_detail

Id & Pwd

Id & Pwd

Info

Login\_detail

Passenger

Id & Pwd

2.0

Login

Details

Id & Pwd

Agency

Login\_detail

Passenger

Id & Pwd

Verified

Route\_info

Employee

Cab\_detail

3.0

Search

Search\_Info

Gps\_detail

Cab\_infoo

Final

Data

Booking\_record

Passenger

Booking\_detail

4.0

Booking

Booking\_no

Cab\_detail

Info

Employee\_detail

Amount

Amount

5.0

Payment

Transaction\_no

Payment\_detail

Transaction\_no

**Zero level DFD for Cabs Booking & Tracking**

1.1

Info Verification

1.2

Login

Allotment

Passenger

Agency

Reg\_detail

Reg. Report

Login\_detail

Reg. Report

Reg\_no

Id & Pwd

Reg\_no

Info

Reg\_no

Id & Pwd

2.1

Check

Authorization

Passenger

Agency

Employee

Login\_detail

Id & Pwd

Id & Pwd

Id & Pwd

If

TRUE

Info

Return T / F

**Level -1 DFD for process 2.0**

**Level -1 DFD for process 1.0**

**Level -1 DFD for process 3.0**

Cabs\_detail

3.1

Search

Criteria

Passenger

Gps\_detail

3.2

Agency

Action

3.3

Document

Verify

3.4

Cab’s

Allotment

Right docs

Proceed

Cab info

Find

Location & details

Search info

Cab No & Driver info

4.1

Booking formultaion

Booking\_detail

4.2

Payment

Verification

4.3

Receipt

Generation

Cab info

Cab’s & Destination info

**Level -1 DFD for process 4.0**

**Level -1 DFD for process 5.0**

5.1

Payment

Mode

5.2

Tax

Calculation

5.3

Payment

Receipt

Payment\_detail

1.1.1

Online Verification

1.2

Offline Verification

Records

Documents

Reg\_detail

Online Verification

Report

Login\_detail

Offline Verification Report

Reg\_no

Id & Pwd

Reg\_no

Info

Reg\_no

Id & Pwd

5.1.1

Payment Gateway

Cash

Credit/Debit Card

Net Banking

Reserve\_Account

Amount

Amt +Id & Pwd

Cabs Allotment Process

Amount

Transaction No.

**Level -2 DFD for process 5.1**

**Level -2 DFD for process 1.1**

Amt +Id & Pwd

In the proposed system “**CABWAYS**” for Taxi Cabs has there are three modules. After combine these modules, we achieve a complete system. These are as follows:

**1. Registration**

Passenger

Agency

Check Cab’s Availability

Registration\_detail

Info

Id & Pwd

Verify

Id & Pwd

Agency info

Id & Pwd

**2. Cab Availability & Booking**

Passenger

Check Cab’s Availability

Gps\_detail

Check Cab’s Availability

Bank\_detail

Check

Info

Yes

Payment

Not Avail

**3. Payment**

Passenger

Agency

Payment

Process

Payment\_detail

Payment

Salary

Info

DataBase Design

**LOGIN\_DETAIL**

|  |  |  |
| --- | --- | --- |
| Field | Data type | Description |
| Login\_id | Integer | Primary key and specific User Login ID |
| Login \_Pwd | Character | Password for corresponding User |
| Login \_type | Number | Login Type (Either Admin or User) |

**EMPLOYEE\_DETAIL**

|  |  |  |
| --- | --- | --- |
| Field | Data type | Description |
| Emp \_id | Integer | Primary key and employee Identification |
| Emp\_name | Character | Employee name |
| Emp\_phone | Number | Employee Phone number |
| Emp\_Address | Character | Employee Address |
| Emp\_age | Number | Age of Employee |
| Emp\_A/cNo | Number | Bank Account No. of Employee |
| Emp\_designation | Character | Designation of Employee |

**AGENCY\_DETAIL**

|  |  |  |
| --- | --- | --- |
| Field | Data type | Description |
| Agency \_id | Integer | Primary key and Vehicle Agency Code |
| Agency\_name | Character | Vehicle Agency Name |
| Agency\_phone | Number | Vehicle Agency Phone no. |
| Agency\_Address | Character | Vehicle Agency Address |
| Agency\_A/c | Number | Bank Account no. of Vehicle Agency |

**PASSENGER\_DETAIL**

|  |  |  |
| --- | --- | --- |
| Field | Data type | Description |
| Passenger \_id | Integer | Primary key and Cab’s Passenger Id No. |
| Passenger\_name | Character | Name of Passenger |
| Passenger \_phone | Number | Passenger Phone Number |
| Passenger \_address | Character | Address of Passenger |
| Passenger \_age | Number | Age of Passenger |
| Passenger \_email | Number | Passenger E-mail ID |

**CAB\_DETAIL**

|  |  |  |
| --- | --- | --- |
| Field | Data type | Description |
| Cab\_id | Integer | Primary key and Cab’s Registration no. |
| Cab\_name | Character | Cab’s Alias Name |
| Cab\_no | Number | Cab’s Unit number |
| Cab\_fare(per km) | Character | Rate of Cab’s (A/C, Simple, Luxury etc) |
| Station\_id | Number | Foreign key |
| Agency\_id | Number | Foreign key |
| Emp\_id | Number | Foreign key |

**PAYMENT\_DETAIL**

|  |  |  |
| --- | --- | --- |
| Field | Data type | Description |
| Transaction\_no | Integer | Primary key and Payment Transaction Id |
| Transaction\_amt | Character | Transacted amount by Passenger |
| Transaction\_mode | Number | Mode of Transaction (Cash/Credit/Debit card/Net Banking/DD/Cheque etc) |
| Payment\_date | Character | Date of Transaction |
| Payment\_type | Number | Type of Payment (Partial/Complete) |
| Passenger\_id | Number | Foreign key |

**BANK\_DETAIL**

|  |  |  |
| --- | --- | --- |
| Field | Data type | Description |
| Bank\_name | Integer | Primary key and Bankers Name |
| Bank\_address | Character | Location of Bank and Branch |
| Bank\_branch | Number | Bank Branch Name |
| Bank\_code | Character | Bank Network Identification Number |

**STATION\_DETAIL**

|  |  |  |
| --- | --- | --- |
| Field | Data type | Description |
| Emp\_id | Integer | Foreign key |
| Station\_name | Character | Station Name/Stand Location of Cab’s |
| Station\_id | Number | Primary key and individual Identification |
| Station\_capacity | Number | Total Cab’s Standing Capacity |
| Station\_location | Number | Complete Address of Station with PIN |
| Station\_Admin-id | Number | Station Administrator Identification no. |

**GPS\_DETAIL**

|  |  |  |
| --- | --- | --- |
| Field | Data type | Description |
| Cab\_id | Integer | Foreign key |
| Gps\_id | Character | Primary key |
| Gps\_date | Number | Date of GPS Published Data |
| Gps\_time | Character | Time of GPS Published Data |

**MECHANIC\_DETAIL**

|  |  |  |
| --- | --- | --- |
| Field | Data type | Description |
| Mechanic\_id | Integer | Primary key & Mechanic Identification Id |
| Mechanic\_name | Character | Name of Mechanic |
| Mechanic\_location | Character | Address of Mechanic |
| Gps\_id | Number | Foreign key |

**ASSOCIATES\_DETAIL**

|  |  |  |
| --- | --- | --- |
| Field | Data type | Description |
| Associate\_id | Integer | Primary key and Association ID Number |
| Associate \_name | Character | Name of Associate |
| Associate \_address | Character | Address of Associate |
| Associate\_phone | Number | Phone number of Associate |
| Remarks | Character | Comments if any |

Module

**Modules**:

The Project is divided into three different modules. They are as follows.

#### Front Office Automation:

**CABWAYS** comes in a plug and play environment supporting two modules – Operations and Maintenance. The Operations module automates the front-office activities – customer bookings, allocation / Duty Slip of vehicles and customer billing. The system captures all-important details related to these activities for further MIS analysis.

* Maintaining the Party / Customer details
* Vehicle Booking Details
* Vehicle Allocation
* Vehicle Billing

## Maintenance *Module:*

**Free Yourself from Maintenance Headaches**

The maintenance module allows you to capture the maintenance costs incurred on a vehicle on account of spares & labor. The maintenance can be classified against a trip or service or breakdown.. It gives you analytical information on the average maintenance cost, mean time & mean-mileage between failures. You can also get a detailed report on each maintenance activity, and it can also tell you when the next service is due and when a particular document is supposed to be renewed.

The Service Log is updated in two instances – once when the vehicle is being sent to the service station and next when it returns back to garage after service.

The Entry/ Edit tab is presented by default to enter the service particulars. The History tab enables you to view the service history details of the vehicles in the fleet. The View tab shows the existing service records for editing pre-service details, update post-service details or cancellation.

* Maintaining The Accident Details Vehicle wise
* Managing the Services and breakdown of Vehicle
* Maintain the service Log of the vehicle
* Alerts for Maintenance

## AdministrationModule*:*

Admin Module handles the entire Master forms such as Company information, Employee information

User allocation, change password, Vehicle type , vehicle tariff, This module also helps to configure such as Financial year configuration etc.,

* Maintaining the Details of the Company
* Vehicle Type
* Vehicle Master
* Vehicle Tariff
* Employee Master
* User Master

**MIS Reports Module:**

The Reporting system is used for the management purpose, which easies the use of the application. For Management the total information will be available on finger tips. Reporting module consists of reporting by Vehicle availability, vehicle allocation, Vehicle Maintenance, Customer details, Customer bookings. Management reports (MIS) will be developed in crystal reports which can be displayed as reports given an option to convert that reports to Microsoft Office Word, Microsoft Office Excel, PDF formats.

This data can be used for management to take crucial decisions**.**

Program Code

Home

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" AutoEventWireup="true" CodeFile="Home.aspx.cs" Inherits="Home" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<table align="center" width="70%">

<tr>

<td valign="middle" align="center">

<table>

<tr>

<td align="left">

<asp:HyperLink ID="HyperLink1" runat="server" NavigateUrl="~/Employees.aspx">Add Employees</asp:HyperLink>&nbsp;

</td>

</tr>

<tr>

<td align="left">

<asp:HyperLink ID="HyperLink2" runat="server" NavigateUrl="~/vehecletype.aspx">Add Vehicle Type</asp:HyperLink>

</td>

</tr>

<tr>

<td align="left">

<asp:HyperLink ID="HyperLink3" runat="server" NavigateUrl="~/Vehiclemaster.aspx">Add Vehicle Details</asp:HyperLink>

</td>

</tr>

<tr>

<td align="left">

<asp:HyperLink id="hlVehicletariffs" runat="server" NavigateUrl="~/Vehicletariff.aspx">Add Vehicle Tariffs</asp:HyperLink>

</td>

</tr>

<tr>

<td align="left">

<asp:HyperLink id="HyperLink9" runat="server" NavigateUrl="~/customers.aspx">Add Customers</asp:HyperLink>

</td>

</tr>

<tr>

<td align="left">

<asp:HyperLink id="hlCustomerbooking" runat="server" NavigateUrl="~/Clientbooking.aspx">Customer Booking</asp:HyperLink>

</td>

</tr>

<tr>

<td align="left">

<asp:HyperLink id="hlVehicleallocation" runat="server" NavigateUrl="~/VehicleAllocation.aspx">Vehicle Allocation</asp:HyperLink>

</td>

</tr>

<tr>

<td align="left">

<asp:HyperLink ID="hlBillingdetails" runat="server" NavigateUrl="~/BillingDetails.aspx">Billing Details</asp:HyperLink>

</td>

</tr>

<tr>

<td align="left">

<asp:HyperLink ID="hlVehicleservice" runat="server" NavigateUrl="~/Service.aspx">Add Service Details</asp:HyperLink>&nbsp;

</td>

</tr>

<tr>

<td align="left">

<asp:HyperLink ID="hlAccident" runat="server" NavigateUrl="~/Accidentdetails.aspx">Add Accident Details</asp:HyperLink>&nbsp;

</td>

</tr>

</table>

</td>

<td width=70%>

<table>

<tr>

<td colspan="2" style="text-align: justify">

Fleet-Cab has been specifically designed keeping in mind the requirements of a cab operator. A cab operator operates in a service industry, which requires high standards of efficiency to be successful. A cab operator provides vehicles on hire to customers and bills them according to the tariff card. The key success factors in this business are quality of service, condition of the vehicle, competitive pricing, customer-interface etc. A cab operator has to keep tight control over these factors to be competitive and remain profitable.

<br />

</td>

</tr>

<tr>

<td>

<asp:HyperLink ID="HyperLink4" runat="server" NavigateUrl="~/Reports.aspx">View Reports</asp:HyperLink>

</td>

<td>

<asp:HyperLink ID="HyperLink6" runat="server" NavigateUrl="~/Employeedetails.aspx">View Employee Details</asp:HyperLink>

</td>

</tr>

<tr>

<td>

<asp:HyperLink ID="HyperLink5" runat="server" NavigateUrl="~/ViewVehilcemaster.aspx">View Vehicle Details</asp:HyperLink>

</td>

<td>

<asp:HyperLink ID="HyperLink7" runat="server" NavigateUrl="~/Viewclientbooking.aspx">View Customer Booking Details</asp:HyperLink>

</td>

</tr>

<tr>

<td>

<asp:HyperLink ID="HyperLink8" runat="server" NavigateUrl="~/Viewcustomers.aspx">View Customers</asp:HyperLink>

</td>

<td>

<%--<asp:HyperLink ID="HyperLink9" runat="server" NavigateUrl="~/Viewclientbooking.aspx">View Customer Booking Details</asp:HyperLink>--%>

</td>

</tr>

</table>

</td>

</tr>

</table>

</asp:Content>

Login page

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" Trace="true" AutoEventWireup="true" CodeFile="Login.aspx.cs" Inherits="Login" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

&nbsp;

<script type="text/javascript" language="javascript">

function validation(sender,args)

{

var check=document.getElementById("ctl00\_ContentPlaceHolder1\_txtName").value

//

if(check.length>0)

{

alert(check.length)

args.isValid=true

}

else

{

alert(check.length)

args.isValid=false

}

}

</script>

<br />

<table align="center">

<tr>

<td style="height: 26px"><asp:Label ID="lblName" runat="server" Text="Username"></asp:Label></td>

<td style="height: 26px"><asp:TextBox ID="txtName" runat="server"></asp:TextBox>

</td>

<td style="height: 26px">

<asp:CustomValidator ID="CustomValidator1" runat="server" ClientValidationFunction="validation" EnableClientScript="true" ErrorMessage="invalid" ControlToValidate="txtName" ValidateEmptyText="True" SetFocusOnError="True"></asp:CustomValidator> </td>

</tr>

<tr>

<td><asp:Label ID="lblPassword" runat="server" Text="Password"></asp:Label></td>

<td><asp:TextBox ID="txtPassword" runat="server"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ControlToValidate="txtPassword"

ErrorMessage="\*" ></asp:RequiredFieldValidator></td>

</tr>

<tr>

<td colspan="3">

&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp

<asp:Button ID="btnLogin" runat="server" Text="Login" OnClick="btnLogin\_Click" />

&nbsp

<asp:Label ID="Label1" runat="server" Visible="False"></asp:Label>

</td>

</tr>

</table>

</asp:Content>

Registration

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" AutoEventWireup="true" CodeFile="Register.aspx.cs" Inherits="Register" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<table align="center">

<tr>

<td><asp:Label ID="lblFirstname" runat="server" Text="First Name"></asp:Label></td>

<td><asp:TextBox ID="txtFirstName" runat="server"></asp:TextBox></td>

<td><%--<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ControlToValidate="txtName"

ErrorMessage="\*"/>--%> </td>

</tr>

<tr>

<td><asp:Label ID="lblLastname" runat="server" Text="lblLastName"></asp:Label></td>

<td><asp:TextBox ID="txtLastName" runat="server"></asp:TextBox></td>

<td><%--<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ControlToValidate="txtPassword"

ErrorMessage="\*" ></asp:RequiredFieldValidator>--%></td>

</tr>

<tr>

<td colspan="3">

<asp:RadioButtonList ID="RadioButtonList1" runat="server">

<asp:ListItem>Male</asp:ListItem>

<asp:ListItem>Female</asp:ListItem>

</asp:RadioButtonList>

</td>

</tr>

<tr>

<td><asp:Label ID="Label2" runat="server" Text="lblLastName"></asp:Label></td>

<td><asp:TextBox ID="TextBox1" runat="server"></asp:TextBox></td>

<td><%--<asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server" ControlToValidate="txtPassword"

ErrorMessage="\*" ></asp:RequiredFieldValidator>--%></td>

</tr>

<tr>

<td colspan="3">

&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp

<asp:Button ID="btnLogin" runat="server" Text="Login" />

&nbsp

<asp:Label ID="Label1" runat="server" Visible="False"></asp:Label>

</td>

</tr>

</table>

</asp:Content>

Reports

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" AutoEventWireup="true" CodeFile="Reports.aspx.cs" Inherits="Reports" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<table align=center width=50%>

<tr>

<td width=50%>

<asp:HyperLink ID="HyperLink1" runat="server" NavigateUrl="~/Employeerpt.aspx">Employees Details</asp:HyperLink>

</td>

<td width=50%>

<asp:HyperLink ID="HyperLink2" runat="server" NavigateUrl="~/Vehiclemasterrpt.aspx">Vehicle Details</asp:HyperLink>

</td>

</tr>

<tr>

<td>

<asp:HyperLink ID="HyperLink3" runat="server" NavigateUrl="~/Billingdetailsrpt.aspx">Billing Details</asp:HyperLink>

</td>

<td>

<asp:HyperLink ID="HyperLink4" runat="server" NavigateUrl="~/Accidentdetailrpt.aspx">Accident Details</asp:HyperLink>

</td>

</tr>

<tr>

<td>

<asp:HyperLink ID="HyperLink5" runat="server" NavigateUrl="~/Servicedetailsrpt.aspx">Service Details</asp:HyperLink>

</td>

<td>

<asp:HyperLink ID="HyperLink6" runat="server" NavigateUrl="~/Customersrpt.aspx">Customer Details</asp:HyperLink>

</td>

</tr>

<tr>

<td colspan="2" align="center">

<asp:Button ID="Button1" runat="server" Text="Back to home" PostBackUrl="~/Home.aspx" />

</td>

</tr>

</table>

</asp:Content>

Services

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" AutoEventWireup="true" CodeFile="Service.aspx.cs" Inherits="Service" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<table class="ds\_box" cellpadding="0" cellspacing="0" id="ds\_conclass" style="display: none;">

<tr><td id="ds\_calclass">

</td></tr>

</table>

<script type="text/javascript">

// <!-- <![CDATA[

// Project: Dynamic Date Selector (DtTvB) - 2006-03-16

// Script featured on JavaScript Kit- http://www.javascriptkit.com

// Code begin...

// Set the initial date.

var ds\_i\_date = new Date();

ds\_c\_month = ds\_i\_date.getMonth() + 1;

ds\_c\_year = ds\_i\_date.getFullYear();

// Get Element By Id

function ds\_getel(id) {

return document.getElementById(id);

}

// Get the left and the top of the element.

function ds\_getleft(el) {

var tmp = el.offsetLeft;

el = el.offsetParent

while(el) {

tmp += el.offsetLeft;

el = el.offsetParent;

}

return tmp;

}

function ds\_gettop(el) {

var tmp = el.offsetTop;

el = el.offsetParent

while(el) {

tmp += el.offsetTop;

el = el.offsetParent;

}

return tmp;

}

// Output Element

var ds\_oe = ds\_getel('ds\_calclass');

// Container

var ds\_ce = ds\_getel('ds\_conclass');

// Output Buffering

var ds\_ob = '';

function ds\_ob\_clean() {

ds\_ob = '';

}

function ds\_ob\_flush() {

ds\_oe.innerHTML = ds\_ob;

ds\_ob\_clean();

}

function ds\_echo(t) {

ds\_ob += t;

}

var ds\_element; // Text Element...

var ds\_monthnames = [

'January', 'February', 'March', 'April', 'May', 'June',

'July', 'August', 'September', 'October', 'November', 'December'

]; // You can translate it for your language.

var ds\_daynames = [

'Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'

]; // You can translate it for your language.

// Calendar template

function ds\_template\_main\_above(t) {

return '<table cellpadding="3" cellspacing="1" class="ds\_tbl">'

+ '<tr>'

+ '<td class="ds\_head" style="cursor: pointer" onclick="ds\_py();">&lt;&lt;</td>'

+ '<td class="ds\_head" style="cursor: pointer" onclick="ds\_pm();">&lt;</td>'

+ '<td class="ds\_head" style="cursor: pointer" onclick="ds\_hi();" colspan="3">[Close]</td>'

+ '<td class="ds\_head" style="cursor: pointer" onclick="ds\_nm();">&gt;</td>'

+ '<td class="ds\_head" style="cursor: pointer" onclick="ds\_ny();">&gt;&gt;</td>'

+ '</tr>'

+ '<tr>'

+ '<td colspan="7" class="ds\_head">' + t + '</td>'

+ '</tr>'

+ '<tr>';

}

function ds\_template\_day\_row(t) {

return '<td class="ds\_subhead">' + t + '</td>';

// Define width in CSS, XHTML 1.0 Strict doesn't have width property for it.

}

function ds\_template\_new\_week() {

return '</tr><tr>';

}

function ds\_template\_blank\_cell(colspan) {

return '<td colspan="' + colspan + '"></td>'

}

function ds\_template\_day(d, m, y) {

return '<td class="ds\_cell" onclick="ds\_onclick(' + d + ',' + m + ',' + y + ')">' + d + '</td>';

// Define width the day row.

}

function ds\_template\_main\_below() {

return '</tr>'

+ '</table>';

}

// This one draws calendar...

function ds\_draw\_calendar(m, y) {

// First clean the output buffer.

ds\_ob\_clean();

// Here we go, do the header

ds\_echo (ds\_template\_main\_above(ds\_monthnames[m - 1] + ' ' + y));

for (i = 0; i < 7; i ++) {

ds\_echo (ds\_template\_day\_row(ds\_daynames[i]));

}

// Make a date object.

var ds\_dc\_date = new Date();

ds\_dc\_date.setMonth(m - 1);

ds\_dc\_date.setFullYear(y);

ds\_dc\_date.setDate(1);

if (m == 1 || m == 3 || m == 5 || m == 7 || m == 8 || m == 10 || m == 12) {

days = 31;

} else if (m == 4 || m == 6 || m == 9 || m == 11) {

days = 30;

} else {

days = (y % 4 == 0) ? 29 : 28;

}

var first\_day = ds\_dc\_date.getDay();

var first\_loop = 1;

// Start the first week

ds\_echo (ds\_template\_new\_week());

// If sunday is not the first day of the month, make a blank cell...

if (first\_day != 0) {

ds\_echo (ds\_template\_blank\_cell(first\_day));

}

var j = first\_day;

for (i = 0; i < days; i ++) {

// Today is sunday, make a new week.

// If this sunday is the first day of the month,

// we've made a new row for you already.

if (j == 0 && !first\_loop) {

// New week!!

ds\_echo (ds\_template\_new\_week());

}

// Make a row of that day!

ds\_echo (ds\_template\_day(i + 1, m, y));

// This is not first loop anymore...

first\_loop = 0;

// What is the next day?

j ++;

j %= 7;

}

// Do the footer

ds\_echo (ds\_template\_main\_below());

// And let's display..

ds\_ob\_flush();

// Scroll it into view.

ds\_ce.scrollIntoView();

}

// A function to show the calendar.

// When user click on the date, it will set the content of t.

function ds\_sh(t) {

// Set the element to set...

ds\_element = t;

// Make a new date, and set the current month and year.

var ds\_sh\_date = new Date();

ds\_c\_month = ds\_sh\_date.getMonth() + 1;

ds\_c\_year = ds\_sh\_date.getFullYear();

// Draw the calendar

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

// To change the position properly, we must show it first.

ds\_ce.style.display = '';

// Move the calendar container!

the\_left = ds\_getleft(t);

the\_top = ds\_gettop(t) + t.offsetHeight;

ds\_ce.style.left = the\_left + 'px';

ds\_ce.style.top = the\_top + 'px';

// Scroll it into view.

ds\_ce.scrollIntoView();

}

// Hide the calendar.

function ds\_hi() {

ds\_ce.style.display = 'none';

}

// Moves to the next month...

function ds\_nm() {

// Increase the current month.

ds\_c\_month ++;

// We have passed December, let's go to the next year.

// Increase the current year, and set the current month to January.

if (ds\_c\_month > 12) {

ds\_c\_month = 1;

ds\_c\_year++;

}

// Redraw the calendar.

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

}

// Moves to the previous month...

function ds\_pm() {

ds\_c\_month = ds\_c\_month - 1; // Can't use dash-dash here, it will make the page invalid.

// We have passed January, let's go back to the previous year.

// Decrease the current year, and set the current month to December.

if (ds\_c\_month < 1) {

ds\_c\_month = 12;

ds\_c\_year = ds\_c\_year - 1; // Can't use dash-dash here, it will make the page invalid.

}

// Redraw the calendar.

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

}

// Moves to the next year...

function ds\_ny() {

// Increase the current year.

ds\_c\_year++;

// Redraw the calendar.

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

}

// Moves to the previous year...

function ds\_py() {

// Decrease the current year.

ds\_c\_year = ds\_c\_year - 1; // Can't use dash-dash here, it will make the page invalid.

// Redraw the calendar.

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

}

// Format the date to output.

function ds\_format\_date(d, m, y) {

// 2 digits month.

m2 = '00' + m;

m2 = m2.substr(m2.length - 2);

// 2 digits day.

d2 = '00' + d;

d2 = d2.substr(d2.length - 2);

// YYYY-MM-DD

// return y + '-' + m2 + '-' + d2;

return m2 + '/' + d2+ '/' + y;

}

// When the user clicks the day.

function ds\_onclick(d, m, y) {

// Hide the calendar.

ds\_hi();

// Set the value of it, if we can.

if (typeof(ds\_element.value) != 'undefined') {

ds\_element.value = ds\_format\_date(d, m, y);

// Maybe we want to set the HTML in it.

} else if (typeof(ds\_element.innerHTML) != 'undefined') {

ds\_element.innerHTML = ds\_format\_date(d, m, y);

// I don't know how should we display it, just alert it to user.

} else {

alert (ds\_format\_date(d, m, y));

}

}

// And here is the end.

// ]]> -->

</script>

<table align="center">

<tr>

<td colspan="3" align="center"><h3>

Service Log</h3>

</td>

</tr>

<tr>

<td colspan="3" align="center">

<asp:Label ID="lblMessage" runat="server" Visible="False" CssClass="lblMessage"></asp:Label>

</td>

</tr>

<tr>

<td>

<asp:Label ID="lblVehicleno" runat="server" Text="Vehicle Number" meta:resourcekey="lblFirstNameResource1"></asp:Label></td>

<td>

<asp:DropDownList ID="ddlVehicleno" runat="server">

</asp:DropDownList>

</td>

<td style="width: 114px">

<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ErrorMessage="\*" ControlToValidate="ddlVehicleno" meta:resourcekey="RequiredFieldValidator1Resource1" SetFocusOnError="True" InitialValue="Select">Select</asp:RequiredFieldValidator></td>

</tr>

<tr>

<td><asp:Label ID="lblDate" runat="server" Text="Service sent date" meta:resourcekey="lblLastnameResource1"></asp:Label></td>

<td>

<asp:TextBox ID="txtServicingdate" runat="server" onclick="ds\_sh(this)" onfocus="ds\_sh(this)"></asp:TextBox></td>

<td style="width: 114px"><asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ErrorMessage="\*" ControlToValidate="txtServicingdate" meta:resourcekey="RequiredFieldValidator2Resource1" SetFocusOnError="True">Date Required</asp:RequiredFieldValidator></td>

</tr>

<tr>

<td><asp:Label ID="lblMeterreading" runat="server" Text="Meter Reading"></asp:Label></td>

<td>

<asp:TextBox ID="txtMeterReading" runat="server" MaxLength="12"></asp:TextBox>

</td>

<td style="width: 114px"><asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server" ErrorMessage="\*" ControlToValidate="txtMeterReading" meta:resourcekey="RequiredFieldValidator3Resource1" SetFocusOnError="True"></asp:RequiredFieldValidator>

<asp:CompareValidator ID="CompareValidator3" runat="server" ErrorMessage="CompareValidator" ControlToValidate="txtMeterReading" Operator="DataTypeCheck" Type="Integer">Eg:1234567890</asp:CompareValidator>

</td>

</tr>

<tr>

<td style="height: 40px"><asp:Label ID="lblReasonforservice" runat="server" Text="Reason for Service" meta:resourcekey="lblDateofjoinResource1"></asp:Label></td>

<td style="height: 40px">

<asp:TextBox ID="txtReasonforservice" runat="server" TextMode="MultiLine"></asp:TextBox>

</td>

<td style="height: 40px; width: 114px;"><asp:RequiredFieldValidator ID="RequiredFieldValidator5" runat="server" ErrorMessage="\*" ControlToValidate="txtReasonforservice" meta:resourcekey="RequiredFieldValidator5Resource1">Reason Required</asp:RequiredFieldValidator></td>

</tr>

<tr>

<td><asp:Label ID="lblSaparesexpenditure" runat="server" Text="Expenditure on Spares" meta:resourcekey="lblQualificationResource1"></asp:Label></td>

<td>

<asp:TextBox ID="txtExpenditureonspares" runat="server" MaxLength="10"></asp:TextBox>

</td>

<td style="width: 114px"><asp:RequiredFieldValidator ID="RequiredFieldValidator6" runat="server" ErrorMessage="\*" ControlToValidate="txtExpenditureonspares" meta:resourcekey="RequiredFieldValidator6Resource1"></asp:RequiredFieldValidator>

<asp:CompareValidator ID="CompareValidator1" runat="server" ErrorMessage="Eg: 10000" ControlToValidate="txtExpenditureonspares" SetFocusOnError="True" Type="Integer" Operator="DataTypeCheck"></asp:CompareValidator>

</td>

</tr>

<tr>

<td><asp:Label ID="lblLabourexpenditure" runat="server" Text="Expenditure on Manpower" meta:resourcekey="lblDateofbirthResource1"></asp:Label></td>

<td>

<asp:TextBox ID="txtExpenditureonmanpower" runat="server" MaxLength="8"></asp:TextBox></td>

<td style="width: 114px"><asp:RequiredFieldValidator ID="RequiredFieldValidator7" runat="server" ErrorMessage="\*" ControlToValidate="txtExpenditureonmanpower"></asp:RequiredFieldValidator>

<asp:CompareValidator ID="CompareValidator2" runat="server" ErrorMessage="Eg: 10000" ControlToValidate="txtExpenditureonmanpower" Operator="DataTypeCheck" SetFocusOnError="True" Type="Integer"></asp:CompareValidator>

</td>

</tr>

<tr>

<td><asp:Label ID="lblDuedate" runat="server" Text="Next Service Date" ></asp:Label></td>

<td>

<asp:TextBox ID="txtNextservicedate" runat="server" onclick="ds\_sh(this)" onfocus="ds\_sh(this)" ></asp:TextBox></td>

<td style="width: 114px"><asp:RequiredFieldValidator ID="RequiredFieldValidator8" runat="server" ErrorMessage="\*" ControlToValidate="txtNextservicedate" meta:resourcekey="RequiredFieldValidator8Resource1">Date Required</asp:RequiredFieldValidator></td>

</tr>

<tr>

<%--<td><asp:Label ID="lblCountry" runat="server" Text="Country " meta:resourcekey="lblCountryResource1"></asp:Label></td>

<td>

<asp:DropDownList ID="ddlCountry" runat="server" meta:resourcekey="ddlCountryResource1">

<asp:ListItem Value="0" meta:resourcekey="ListItemResource15">Select</asp:ListItem>

<asp:ListItem Value="1" meta:resourcekey="ListItemResource16">India</asp:ListItem>

<asp:ListItem Value="2" meta:resourcekey="ListItemResource17">Others</asp:ListItem>

</asp:DropDownList></td>

<td><asp:RequiredFieldValidator ID="RequiredFieldValidator9" runat="server" ErrorMessage="\*" ControlToValidate="ddlCountry" meta:resourcekey="RequiredFieldValidator9Resource1">Select</asp:RequiredFieldValidator></td>

</tr>

<tr>

<td><asp:Label ID="lblState" runat="server" Text="State" meta:resourcekey="lblStateResource1"></asp:Label></td>

<td>

<asp:TextBox ID="txtState" runat="server" meta:resourcekey="txtStateResource1"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="RequiredFieldValidator10" runat="server" ErrorMessage="\*" ControlToValidate="txtState" meta:resourcekey="RequiredFieldValidator10Resource1">State Required</asp:RequiredFieldValidator></td>

</tr>

<tr>

<td><asp:Label ID="lblPhone" runat="server" Text="Contact No" meta:resourcekey="lblPhoneResource1"></asp:Label></td>

<td>

<asp:TextBox ID="txtContactno" runat="server" meta:resourcekey="txtContactnoResource1"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="RequiredFieldValidator11" runat="server" ErrorMessage="\*" ControlToValidate="txtContactno" meta:resourcekey="RequiredFieldValidator11Resource1"></asp:RequiredFieldValidator>

<asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server" ErrorMessage="\*" ControlToValidate="txtContactno" SetFocusOnError="True" ValidationExpression="^([0-9]( |-)?)?(\(?[0-9]{3}\)?|[0-9]{3})( |-)?([0-9]{3}( |-)?[0-9]{4}|[a-zA-Z0-9]{7})$"></asp:RegularExpressionValidator>

</td>

</tr>

<tr>

<td><asp:Label ID="lblEmailid" runat="server" Text="Email Id" meta:resourcekey="lblEmailidResource1"></asp:Label></td>

<td>

<asp:TextBox ID="txtEmailid" runat="server" meta:resourcekey="txtEmailidResource1"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="RequiredFieldValidator12" runat="server" ErrorMessage="\*" ControlToValidate="txtEmailid" meta:resourcekey="RequiredFieldValidator12Resource1"></asp:RequiredFieldValidator>

<asp:RegularExpressionValidator id="RegularExpressionValidator2" runat="server" SetFocusOnError="True" ControlToValidate="txtEmailid" ErrorMessage="\*" ValidationExpression="\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*">Eg: proj@g.com</asp:RegularExpressionValidator>

</td>

</tr>--%>

<tr>

<td colspan="3" align="center">

<asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit\_Click" meta:resourcekey="btnSubmitResource1" />&nbsp;

<asp:Button ID="btnBack" runat="server" Text="Back" PostBackUrl="~/Home.aspx" CausesValidation="False" meta:resourcekey="btnBackResource1" />

</td>

</tr>

</table>

</asp:Content>

ServiceDetailsrpt

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" AutoEventWireup="true" CodeFile="Servicedetailsrpt.aspx.cs" Inherits="Servicedetailsrpt" Title="Untitled Page" %>

<%@ Register Assembly="CrystalDecisions.Web, Version=10.2.3600.0, Culture=neutral, PublicKeyToken=692fbea5521e1304"

Namespace="CrystalDecisions.Web" TagPrefix="CR" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<CR:CrystalReportViewer ID="CrystalReportViewer1" runat="server" AutoDataBind="True" Height="1039px" ReportSourceID="CrystalReportSource1" Width="901px" />

<CR:CrystalReportSource ID="CrystalReportSource1" runat="server">

<Report FileName="Servicelog.rpt">

</Report>

</CR:CrystalReportSource>

</asp:Content>

Vehicletype

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" AutoEventWireup="true" CodeFile="vehecletype.aspx.cs" Inherits="vehecletype" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<table align="center">

<tr>

<td colspan="3" align="center"><h3>Add Vehicle Type</h3></td>

</tr>

<tr>

<td><asp:Label ID="lblName" runat="server" Text="VehicleName"></asp:Label></td>

<td><asp:TextBox ID="txtName" runat="server" AutoCompleteType="Disabled"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ControlToValidate="txtName"

ErrorMessage="Vehicle name Required" SetFocusOnError="True"/>&nbsp;

</td>

</tr>

<%--<tr>

<td colspan="3">

&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp

&nbsp

</td>

</tr>--%>

<tr>

<td colspan="3" align="center">

<asp:Button ID="btnInsert" runat="server" Text="Insert" OnClick="btnLogin\_Click" />&nbsp;

<asp:Button ID="btnBack" runat="server" Text="Back" PostBackUrl="~/Home.aspx" CausesValidation="False" />&nbsp;

<asp:Label ID="lblMessage" runat="server" Visible="False" CssClass="lblMessage">Inserted Succesfully</asp:Label>

</td>

</tr>

</table>

</asp:Content>

VehicleAllocation

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" AutoEventWireup="true" CodeFile="VehicleAllocation.aspx.cs" Inherits="VehicleAllocation" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<table class="ds\_box" cellpadding="0" cellspacing="0" id="ds\_conclass" style="display: none;">

<tr><td id="ds\_calclass">

</td></tr>

</table>

<script type="text/javascript">

// <!-- <![CDATA[

// Project: Dynamic Date Selector (DtTvB) - 2006-03-16

// Script featured on JavaScript Kit- http://www.javascriptkit.com

// Code begin...

// Set the initial date.

var ds\_i\_date = new Date();

ds\_c\_month = ds\_i\_date.getMonth() + 1;

ds\_c\_year = ds\_i\_date.getFullYear();

// Get Element By Id

function ds\_getel(id) {

return document.getElementById(id);

}

// Get the left and the top of the element.

function ds\_getleft(el) {

var tmp = el.offsetLeft;

el = el.offsetParent

while(el) {

tmp += el.offsetLeft;

el = el.offsetParent;

}

return tmp;

}

function ds\_gettop(el) {

var tmp = el.offsetTop;

el = el.offsetParent

while(el) {

tmp += el.offsetTop;

el = el.offsetParent;

}

return tmp;

}

// Output Element

var ds\_oe = ds\_getel('ds\_calclass');

// Container

var ds\_ce = ds\_getel('ds\_conclass');

// Output Buffering

var ds\_ob = '';

function ds\_ob\_clean() {

ds\_ob = '';

}

function ds\_ob\_flush() {

ds\_oe.innerHTML = ds\_ob;

ds\_ob\_clean();

}

function ds\_echo(t) {

ds\_ob += t;

}

var ds\_element; // Text Element...

var ds\_monthnames = [

'January', 'February', 'March', 'April', 'May', 'June',

'July', 'August', 'September', 'October', 'November', 'December'

]; // You can translate it for your language.

var ds\_daynames = [

'Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'

]; // You can translate it for your language.

// Calendar template

function ds\_template\_main\_above(t) {

return '<table cellpadding="3" cellspacing="1" class="ds\_tbl">'

+ '<tr>'

+ '<td class="ds\_head" style="cursor: pointer" onclick="ds\_py();">&lt;&lt;</td>'

+ '<td class="ds\_head" style="cursor: pointer" onclick="ds\_pm();">&lt;</td>'

+ '<td class="ds\_head" style="cursor: pointer" onclick="ds\_hi();" colspan="3">[Close]</td>'

+ '<td class="ds\_head" style="cursor: pointer" onclick="ds\_nm();">&gt;</td>'

+ '<td class="ds\_head" style="cursor: pointer" onclick="ds\_ny();">&gt;&gt;</td>'

+ '</tr>'

+ '<tr>'

+ '<td colspan="7" class="ds\_head">' + t + '</td>'

+ '</tr>'

+ '<tr>';

}

function ds\_template\_day\_row(t) {

return '<td class="ds\_subhead">' + t + '</td>';

// Define width in CSS, XHTML 1.0 Strict doesn't have width property for it.

}

function ds\_template\_new\_week() {

return '</tr><tr>';

}

function ds\_template\_blank\_cell(colspan) {

return '<td colspan="' + colspan + '"></td>'

}

function ds\_template\_day(d, m, y) {

return '<td class="ds\_cell" onclick="ds\_onclick(' + d + ',' + m + ',' + y + ')">' + d + '</td>';

// Define width the day row.

}

function ds\_template\_main\_below() {

return '</tr>'

+ '</table>';

}

// This one draws calendar...

function ds\_draw\_calendar(m, y) {

// First clean the output buffer.

ds\_ob\_clean();

// Here we go, do the header

ds\_echo (ds\_template\_main\_above(ds\_monthnames[m - 1] + ' ' + y));

for (i = 0; i < 7; i ++) {

ds\_echo (ds\_template\_day\_row(ds\_daynames[i]));

}

// Make a date object.

var ds\_dc\_date = new Date();

ds\_dc\_date.setMonth(m - 1);

ds\_dc\_date.setFullYear(y);

ds\_dc\_date.setDate(1);

if (m == 1 || m == 3 || m == 5 || m == 7 || m == 8 || m == 10 || m == 12) {

days = 31;

} else if (m == 4 || m == 6 || m == 9 || m == 11) {

days = 30;

} else {

days = (y % 4 == 0) ? 29 : 28;

}

var first\_day = ds\_dc\_date.getDay();

var first\_loop = 1;

// Start the first week

ds\_echo (ds\_template\_new\_week());

// If sunday is not the first day of the month, make a blank cell...

if (first\_day != 0) {

ds\_echo (ds\_template\_blank\_cell(first\_day));

}

var j = first\_day;

for (i = 0; i < days; i ++) {

// Today is sunday, make a new week.

// If this sunday is the first day of the month,

// we've made a new row for you already.

if (j == 0 && !first\_loop) {

// New week!!

ds\_echo (ds\_template\_new\_week());

}

// Make a row of that day!

ds\_echo (ds\_template\_day(i + 1, m, y));

// This is not first loop anymore...

first\_loop = 0;

// What is the next day?

j ++;

j %= 7;

}

// Do the footer

ds\_echo (ds\_template\_main\_below());

// And let's display..

ds\_ob\_flush();

// Scroll it into view.

ds\_ce.scrollIntoView();

}

// A function to show the calendar.

// When user click on the date, it will set the content of t.

function ds\_sh(t) {

// Set the element to set...

ds\_element = t;

// Make a new date, and set the current month and year.

var ds\_sh\_date = new Date();

ds\_c\_month = ds\_sh\_date.getMonth() + 1;

ds\_c\_year = ds\_sh\_date.getFullYear();

// Draw the calendar

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

// To change the position properly, we must show it first.

ds\_ce.style.display = '';

// Move the calendar container!

the\_left = ds\_getleft(t);

the\_top = ds\_gettop(t) + t.offsetHeight;

ds\_ce.style.left = the\_left + 'px';

ds\_ce.style.top = the\_top + 'px';

// Scroll it into view.

ds\_ce.scrollIntoView();

}

// Hide the calendar.

function ds\_hi() {

ds\_ce.style.display = 'none';

}

// Moves to the next month...

function ds\_nm() {

// Increase the current month.

ds\_c\_month ++;

// We have passed December, let's go to the next year.

// Increase the current year, and set the current month to January.

if (ds\_c\_month > 12) {

ds\_c\_month = 1;

ds\_c\_year++;

}

// Redraw the calendar.

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

}

// Moves to the previous month...

function ds\_pm() {

ds\_c\_month = ds\_c\_month - 1; // Can't use dash-dash here, it will make the page invalid.

// We have passed January, let's go back to the previous year.

// Decrease the current year, and set the current month to December.

if (ds\_c\_month < 1) {

ds\_c\_month = 12;

ds\_c\_year = ds\_c\_year - 1; // Can't use dash-dash here, it will make the page invalid.

}

// Redraw the calendar.

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

}

// Moves to the next year...

function ds\_ny() {

// Increase the current year.

ds\_c\_year++;

// Redraw the calendar.

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

}

// Moves to the previous year...

function ds\_py() {

// Decrease the current year.

ds\_c\_year = ds\_c\_year - 1; // Can't use dash-dash here, it will make the page invalid.

// Redraw the calendar.

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

}

// Format the date to output.

function ds\_format\_date(d, m, y) {

// 2 digits month.

m2 = '00' + m;

m2 = m2.substr(m2.length - 2);

// 2 digits day.

d2 = '00' + d;

d2 = d2.substr(d2.length - 2);

// YYYY-MM-DD

// return y + '-' + m2 + '-' + d2;

return m2 + '/' + d2+ '/' + y;

}

// When the user clicks the day.

function ds\_onclick(d, m, y) {

// Hide the calendar.

ds\_hi();

// Set the value of it, if we can.

if (typeof(ds\_element.value) != 'undefined') {

ds\_element.value = ds\_format\_date(d, m, y);

// Maybe we want to set the HTML in it.

} else if (typeof(ds\_element.innerHTML) != 'undefined') {

ds\_element.innerHTML = ds\_format\_date(d, m, y);

// I don't know how should we display it, just alert it to user.

} else {

alert (ds\_format\_date(d, m, y));

}

}

// And here is the end.

// ]]> -->

</script>

<table align="center">

<tr>

<td colspan="3" align="center"><h3>Vehicle Allocation</h3>

</td>

</tr>

<tr>

<td colspan="3" align="center">

<asp:Label ID="lblMessage" runat="server" Visible="False" CssClass="lblMessage"></asp:Label>

</td>

</tr>

<tr>

<td>

<asp:Label ID="lblBookingid" runat="server" Text="Booking Id"></asp:Label></td>

<td>

<asp:DropDownList ID="ddlBookingid" runat="server" AutoPostBack="True">

<asp:ListItem>Select</asp:ListItem>

</asp:DropDownList></td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ErrorMessage="Select ID" ControlToValidate="ddlBookingid" InitialValue="Select" SetFocusOnError="True"></asp:RequiredFieldValidator></td>

</tr>

<tr>

<td style="height: 26px">Customer Name</td>

<td style="height: 26px">

<asp:TextBox ID="txtCustomername" runat="server" ReadOnly="True"></asp:TextBox></td>

</tr>

<tr>

<td>

<asp:Label ID="lblVehicletype" runat="server" Text="Vehicle Type"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtVehicletype" runat="server" ReadOnly="True"></asp:TextBox>

</td>

</tr>

<tr>

<td>

<asp:Label ID="lblVehicleno" runat="server" Text="Vehicle No"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtVehicleno" runat="server"></asp:TextBox>

</td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ErrorMessage="\*" ControlToValidate="txtVehicleno" SetFocusOnError="True">Vehicle no required</asp:RequiredFieldValidator></td>

</tr>

<tr>

<td>

<asp:Label ID="lblDutyslipno" runat="server" Text="Duty Slip No"></asp:Label></td>

<td>

<asp:TextBox ID="txtDutyslipno" runat="server" ReadOnly="True"></asp:TextBox>

</td>

</tr>

<tr>

<td>

<asp:Label ID="lblDutyslipdate" runat="server" Text="Duty Slip date"></asp:Label></td>

<td>

<asp:TextBox ID="txtDutyslipdate" runat="server" ReadOnly="True"></asp:TextBox></td>

</tr>

<%--<tr>

<td>

<asp:Label ID="lblPassengers" runat="server" Text="Passengers"></asp:Label></td>

<td>

<asp:TextBox ID="txtPassengers" runat="server"></asp:TextBox></td>

</tr>--%>

<tr>

<td style="height: 26px">

<asp:Label ID="lblGuestname" runat="server" Text="Guest Name"></asp:Label></td>

<td style="height: 26px">

<asp:TextBox ID="txtGuestname" runat="server" ReadOnly="True"></asp:TextBox></td>

</tr>

<tr>

<td style="height: 26px">

<asp:Label ID="lblAddress" runat="server" Text="Pickup Address"></asp:Label></td>

<td style="height: 26px">

<asp:TextBox ID="txtAddress" runat="server" ReadOnly="True"></asp:TextBox></td>

</tr>

<tr>

<td>

<asp:Label ID="lblPhoneno" runat="server" Text="Phone No"></asp:Label></td>

<td>

<asp:TextBox ID="txtPhoneno" runat="server" ReadOnly="True"></asp:TextBox></td>

</tr>

<%--<tr>

<td>

<asp:Label ID="lblDriverallot" runat="server" Text="Driver Allocated"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtDriverallot" runat="server"></asp:TextBox>

</td>

</tr>--%>

<tr>

<td>

<asp:Label ID="lblDriver" runat="server" Text="Driver name"></asp:Label></td>

<td>

<asp:DropDownList ID="ddlDriver" runat="server">

</asp:DropDownList></td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server" ErrorMessage="\*" ControlToValidate="ddlDriver" InitialValue="Select">Select Driver</asp:RequiredFieldValidator></td>

</tr>

<tr>

<td colspan="2" align="center">

<asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit\_Click1" />&nbsp;

<asp:Button ID="btnBack" runat="server" Text="Back" PostBackUrl="~/Home.aspx" CausesValidation="False" />

</td>

</tr>

</table>

</asp:Content>

VehicleMantinance

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" AutoEventWireup="true" CodeFile="VehicleMaintenance.aspx.cs" Inherits="VehicleMaintenance" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<%--<script>

function th(t)

{

var elem=document.getElementById("<%= txtServiceforkms.ClientID%>");

alert(elem.innerHTML);

}

</script>--%>

<table width=95%>

<tr><td colspan="2" align="center"><h3>Vehicle Maintenance</h3></td></tr>

<tr>

<td colspan="2" align="center" style="height: 21px">

<asp:Label ID="lblMessage" runat="server" Visible="False" CssClass="lblMessage"></asp:Label>

</td>

</tr>

<tr>

<td align="center" >

<asp:Label ID="lblVehicletype" runat="server" Text="Vehicle type"></asp:Label>

&nbsp;

<asp:DropDownList ID="ddlVehicletype" runat="server" AutoPostBack="True" OnSelectedIndexChanged="ddlVehicletype\_SelectedIndexChanged">

</asp:DropDownList>

&nbsp;&nbsp;

<asp:Label ID="lblVehicleno" runat="server" Text="Vehicle no"></asp:Label>

&nbsp;

<asp:DropDownList ID="ddlVehicleno" runat="server" AutoPostBack="True" OnSelectedIndexChanged="ddlVehicleno\_SelectedIndexChanged">

</asp:DropDownList>

&nbsp;&nbsp;

<asp:Label ID="lblDate" runat="server" Text="Date"></asp:Label>

&nbsp;

<asp:TextBox ID="txtDate" runat="server"></asp:TextBox></td>

</tr>

<%--<tr>

<td colspan="2">

<fieldset>

<legend><i>Fuel Purchase Record</i></legend>

<table width=95%>

<tr>

<td>

<table>

<tr>

<td>

<asp:Label ID="lblThismeterreading" runat="server" Text="Meter Reading Now"></asp:Label>

</td>

<td>

<asp:TextBox ID="TextBox6" runat="server"></asp:TextBox>

</td>

</tr>

<tr>

<td>

<asp:Label ID="lblLastmeterreading" runat="server" Text="Last Meter Reading"></asp:Label>

</td>

<td>

<asp:TextBox ID="TextBox7" runat="server"></asp:TextBox>

</td>

</tr>

<tr>

<td>

<asp:Label ID="lblTotalKms" runat="server" Text="Km's Driven"></asp:Label>

</td>

<td>

<asp:TextBox ID="TextBox3" runat="server" ReadOnly="True"></asp:TextBox>

</td>

</tr>

</table>

</td>

<td>

<table>

<tr>

<td>

<asp:Label ID="lblFuelprice" runat="server" Text="Fuel Price/Lit"></asp:Label>

</td>

<td>

<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox></td>

</tr>

<tr>

<td>

<asp:Label ID="lblPurchase" runat="server" Text="Purchase"></asp:Label>

</td>

<td>

<asp:TextBox ID="TextBox4" runat="server"></asp:TextBox>

</td>

</tr>

<tr>

<td>

<asp:Label ID="Label2" runat="server" Text="Liters"></asp:Label></td>

<td>

<asp:TextBox ID="TextBox5" runat="server"></asp:TextBox></td>

</tr>

<tr>

<td colspan="2">

<b>Km's/Liter</b>

</td>

</tr>

<tr>

<td>

<asp:Label ID="lblThistank" runat="server" Text="This tank"></asp:Label></td>

<td>

<asp:TextBox ID="TextBox8" runat="server"></asp:TextBox></td>

</tr>

<tr>

<td>

<asp:Label ID="lblAccumulated" runat="server" Text="Accumulated"></asp:Label></td>

<td>

<asp:TextBox ID="TextBox9" runat="server"></asp:TextBox></td>

</tr>

</table>

</td>

</tr>

</table>

</fieldset>

</td>

</tr>--%>

<tr>

<td colspan="2">

<fieldset>

<legend>

<i>Service</i>

</legend>

<table width=95%>

<tr>

<td>

<table >

<tr>

<td>

<asp:Label ID="lblPerformed" runat="server" Text="Service has to be performed for every\_\_kms"></asp:Label></td>

<td>

<asp:TextBox ID="txtServiceforkms" runat="server" OnTextChanged="txtServiceforkms\_TextChanged" AutoPostBack="True"></asp:TextBox></td>

</tr>

<tr>

<td>

<asp:Label ID="lblLastperformed" runat="server" Text="Performed at"></asp:Label></td>

<td>

<asp:TextBox ID="txtLastservice" runat="server" ></asp:TextBox></td>

</tr>

<tr>

<td style="height: 26px">

<asp:Label ID="lblTobeperformed" runat="server" Text="Next service has to be performed at"></asp:Label></td>

<td style="height: 26px">

<asp:TextBox ID="txtServiceat" runat="server" AutoPostBack="True" onfocus=this(t);></asp:TextBox></td>

</tr>

</table>

</td>

<td>

<table>

<tr>

<td align=right>

<asp:Label ID="lblLastfuelpurchase" runat="server" Text="To be performed at every/Months"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtServiceformonths" runat="server"></asp:TextBox></td>

</tr>

<tr>

<td>

<asp:Label ID="lblSincekms" runat="server" Text="Last performed at which date"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtServiceddate" runat="server"></asp:TextBox></td>

</tr>

<tr>

<td>

<asp:Label ID="Label2" runat="server" Text="To be performed at"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtServicedate" runat="server"></asp:TextBox></td>

</tr>

</table>

</td>

</tr>

<tr>

<td colspan="2" align="center" style="height: 26px">

<asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit\_Click" />&nbsp;

<asp:Button ID="btnBack" runat="server" Text="Back" PostBackUrl="~/Home.aspx" />

</td>

</tr>

</table>

</fieldset>

</td>

</tr>

</table>

</asp:Content>

Vehicle Master

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" AutoEventWireup="true" CodeFile="Vehiclemaster.aspx.cs" Inherits="Vehiclemaster" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<%--For Canlender--%>

<table class="ds\_box" cellpadding="0" cellspacing="0" id="ds\_conclass" style="display: none;">

<tr><td id="ds\_calclass">

</td></tr>

</table>

<script type="text/javascript">

// <!-- <![CDATA[

// Project: Dynamic Date Selector (DtTvB) - 2006-03-16

// Script featured on JavaScript Kit- http://www.javascriptkit.com

// Code begin...

// Set the initial date.

var ds\_i\_date = new Date();

ds\_c\_month = ds\_i\_date.getMonth() + 1;

ds\_c\_year = ds\_i\_date.getFullYear();

// Get Element By Id

function ds\_getel(id) {

return document.getElementById(id);

}

// Get the left and the top of the element.

function ds\_getleft(el) {

var tmp = el.offsetLeft;

el = el.offsetParent

while(el) {

tmp += el.offsetLeft;

el = el.offsetParent;

}

return tmp;

}

function ds\_gettop(el) {

var tmp = el.offsetTop;

el = el.offsetParent

while(el) {

tmp += el.offsetTop;

el = el.offsetParent;

}

return tmp;

}

// Output Element

var ds\_oe = ds\_getel('ds\_calclass');

// Container

var ds\_ce = ds\_getel('ds\_conclass');

// Output Buffering

var ds\_ob = '';

function ds\_ob\_clean() {

ds\_ob = '';

}

function ds\_ob\_flush() {

ds\_oe.innerHTML = ds\_ob;

ds\_ob\_clean();

}

function ds\_echo(t) {

ds\_ob += t;

}

var ds\_element; // Text Element...

var ds\_monthnames = [

'January', 'February', 'March', 'April', 'May', 'June',

'July', 'August', 'September', 'October', 'November', 'December'

]; // You can translate it for your language.

var ds\_daynames = [

'Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'

]; // You can translate it for your language.

// Calendar template

function ds\_template\_main\_above(t) {

return '<table cellpadding="3" cellspacing="1" class="ds\_tbl">'

+ '<tr>'

+ '<td class="ds\_head" style="cursor: pointer" onclick="ds\_py();">&lt;&lt;</td>'

+ '<td class="ds\_head" style="cursor: pointer" onclick="ds\_pm();">&lt;</td>'

+ '<td class="ds\_head" style="cursor: pointer" onclick="ds\_hi();" colspan="3">[Close]</td>'

+ '<td class="ds\_head" style="cursor: pointer" onclick="ds\_nm();">&gt;</td>'

+ '<td class="ds\_head" style="cursor: pointer" onclick="ds\_ny();">&gt;&gt;</td>'

+ '</tr>'

+ '<tr>'

+ '<td colspan="7" class="ds\_head">' + t + '</td>'

+ '</tr>'

+ '<tr>';

}

function ds\_template\_day\_row(t) {

return '<td class="ds\_subhead">' + t + '</td>';

// Define width in CSS, XHTML 1.0 Strict doesn't have width property for it.

}

function ds\_template\_new\_week() {

return '</tr><tr>';

}

function ds\_template\_blank\_cell(colspan) {

return '<td colspan="' + colspan + '"></td>'

}

function ds\_template\_day(d, m, y) {

return '<td class="ds\_cell" onclick="ds\_onclick(' + d + ',' + m + ',' + y + ')">' + d + '</td>';

// Define width the day row.

}

function ds\_template\_main\_below() {

return '</tr>'

+ '</table>';

}

// This one draws calendar...

function ds\_draw\_calendar(m, y) {

// First clean the output buffer.

ds\_ob\_clean();

// Here we go, do the header

ds\_echo (ds\_template\_main\_above(ds\_monthnames[m - 1] + ' ' + y));

for (i = 0; i < 7; i ++) {

ds\_echo (ds\_template\_day\_row(ds\_daynames[i]));

}

// Make a date object.

var ds\_dc\_date = new Date();

ds\_dc\_date.setMonth(m - 1);

ds\_dc\_date.setFullYear(y);

ds\_dc\_date.setDate(1);

if (m == 1 || m == 3 || m == 5 || m == 7 || m == 8 || m == 10 || m == 12) {

days = 31;

} else if (m == 4 || m == 6 || m == 9 || m == 11) {

days = 30;

} else {

days = (y % 4 == 0) ? 29 : 28;

}

var first\_day = ds\_dc\_date.getDay();

var first\_loop = 1;

// Start the first week

ds\_echo (ds\_template\_new\_week());

// If sunday is not the first day of the month, make a blank cell...

if (first\_day != 0) {

ds\_echo (ds\_template\_blank\_cell(first\_day));

}

var j = first\_day;

for (i = 0; i < days; i ++) {

// Today is sunday, make a new week.

// If this sunday is the first day of the month,

// we've made a new row for you already.

if (j == 0 && !first\_loop) {

// New week!!

ds\_echo (ds\_template\_new\_week());

}

// Make a row of that day!

ds\_echo (ds\_template\_day(i + 1, m, y));

// This is not first loop anymore...

first\_loop = 0;

// What is the next day?

j ++;

j %= 7;

}

// Do the footer

ds\_echo (ds\_template\_main\_below());

// And let's display..

ds\_ob\_flush();

// Scroll it into view.

ds\_ce.scrollIntoView();

}

// A function to show the calendar.

// When user click on the date, it will set the content of t.

function ds\_sh(t) {

//hiding dropdownlists while calender popup

var elem=document.getElementById("<%=ddlFueltype.ClientID %>");

elem.style.display='none';

var elem=document.getElementById("<%=ddlAc.ClientID %>");

elem.style.display='none';

// Set the element to set...

ds\_element = t;

// Make a new date, and set the current month and year.

var ds\_sh\_date = new Date();

ds\_c\_month = ds\_sh\_date.getMonth() + 1;

ds\_c\_year = ds\_sh\_date.getFullYear();

// Draw the calendar

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

// To change the position properly, we must show it first.

ds\_ce.style.display = '';

// Move the calendar container!

the\_left = ds\_getleft(t);

the\_top = ds\_gettop(t) + t.offsetHeight;

ds\_ce.style.left = the\_left + 'px';

ds\_ce.style.top = the\_top + 'px';

// Scroll it into view.

ds\_ce.scrollIntoView();

}

function ds\_hh(t) {

// Set the element to set...

ds\_element = t;

// Make a new date, and set the current month and year.

var ds\_sh\_date = new Date();

ds\_c\_month = ds\_sh\_date.getMonth() + 1;

ds\_c\_year = ds\_sh\_date.getFullYear();

// Draw the calendar

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

// To change the position properly, we must show it first.

ds\_ce.style.display = '';

// Move the calendar container!

the\_left = ds\_getleft(t);

the\_top = ds\_gettop(t) + t.offsetHeight;

ds\_ce.style.left = the\_left + 'px';

ds\_ce.style.top = the\_top + 'px';

// Scroll it into view.

ds\_ce.scrollIntoView();

}

function ds\_hh(t) {

// Set the element to set...

ds\_element = t;

// Make a new date, and set the current month and year.

var ds\_sh\_date = new Date();

ds\_c\_month = ds\_sh\_date.getMonth() + 1;

ds\_c\_year = ds\_sh\_date.getFullYear();

// Draw the calendar

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

// To change the position properly, we must show it first.

ds\_ce.style.display = '';

// Move the calendar container!

the\_left = ds\_getleft(t);

the\_top = ds\_gettop(t) + t.offsetHeight;

ds\_ce.style.left = the\_left + 'px';

ds\_ce.style.top = the\_top + 'px';

// Scroll it into view.

ds\_ce.scrollIntoView();

}

// Hide the calendar.

function ds\_hi() {

//displaying dropdownlists

var elem=document.getElementById("<%=ddlFueltype.ClientID %>");

elem.style.display='block';

var elem=document.getElementById("<%=ddlAc.ClientID %>");

elem.style.display='block';

ds\_ce.style.display = 'none';

}

// Moves to the next month...

function ds\_nm() {

// Increase the current month.

ds\_c\_month ++;

// We have passed December, let's go to the next year.

// Increase the current year, and set the current month to January.

if (ds\_c\_month > 12) {

ds\_c\_month = 1;

ds\_c\_year++;

}

// Redraw the calendar.

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

}

// Moves to the previous month...

function ds\_pm() {

ds\_c\_month = ds\_c\_month - 1; // Can't use dash-dash here, it will make the page invalid.

// We have passed January, let's go back to the previous year.

// Decrease the current year, and set the current month to December.

if (ds\_c\_month < 1) {

ds\_c\_month = 12;

ds\_c\_year = ds\_c\_year - 1; // Can't use dash-dash here, it will make the page invalid.

}

// Redraw the calendar.

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

}

// Moves to the next year...

function ds\_ny() {

// Increase the current year.

ds\_c\_year++;

// Redraw the calendar.

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

}

// Moves to the previous year...

function ds\_py() {

// Decrease the current year.

ds\_c\_year = ds\_c\_year - 1; // Can't use dash-dash here, it will make the page invalid.

// Redraw the calendar.

ds\_draw\_calendar(ds\_c\_month, ds\_c\_year);

}

// Format the date to output.

function ds\_format\_date(d, m, y) {

// 2 digits month.

m2 = '00' + m;

m2 = m2.substr(m2.length - 2);

// 2 digits day.

d2 = '00' + d;

d2 = d2.substr(d2.length - 2);

// YYYY-MM-DD

// return y + '-' + m2 + '-' + d2;

return m2 + '/' + d2+ '/' + y;

}

// When the user clicks the day.

function ds\_onclick(d, m, y) {

// Hide the calendar.

ds\_hi();

// Set the value of it, if we can.

if (typeof(ds\_element.value) != 'undefined') {

ds\_element.value = ds\_format\_date(d, m, y);

// Maybe we want to set the HTML in it.

} else if (typeof(ds\_element.innerHTML) != 'undefined') {

ds\_element.innerHTML = ds\_format\_date(d, m, y);

// I don't know how should we display it, just alert it to user.

} else {

alert (ds\_format\_date(d, m, y));

}

}

// And here is the end.

// ]]> -->

</script>

<%--Main content--%>

<table align="center">

<tr>

<td colspan="3" align="center"><h3>Add Vehicle Details</h3></td>

</tr>

<tr>

<td colspan="3" align="center">

<asp:Label ID="lblMessage" runat="server" Visible="False" CssClass="lblMessage"></asp:Label>

</td>

</tr>

<tr>

<td>

<asp:Label ID="lblVehicletype" runat="server" Text="Vehicle Type"></asp:Label></td>

<td>

<asp:DropDownList ID="ddlVehicletype" runat="server">

</asp:DropDownList>

</td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ErrorMessage="\*" ControlToValidate="ddlVehicletype" InitialValue="Select" SetFocusOnError="True">Select Vehicle Type</asp:RequiredFieldValidator></td>

</tr>

<tr>

<td><asp:Label ID="lblRegno" runat="server" Text="Registration No"></asp:Label></td>

<td>

<asp:TextBox ID="txtRegno" runat="server"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ErrorMessage="\*" ControlToValidate="txtRegno" SetFocusOnError="True">Registration no Required</asp:RequiredFieldValidator></td>

</tr>

<tr>

<td><asp:Label ID="lblRegdate" runat="server" Text="Registration Date"></asp:Label></td>

<td>

<asp:TextBox ID="txtRegdate" onclick="ds\_sh(this)" onfocus="ds\_sh(this)" runat="server"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server" ErrorMessage="\*" ControlToValidate="txtRegdate">Date Required</asp:RequiredFieldValidator></td>

</tr>

<tr>

<td><asp:Label ID="lblEngineno" runat="server" Text="Engine Num"></asp:Label></td>

<td>

<asp:TextBox ID="txtEngno" runat="server"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server" ErrorMessage="\*" ControlToValidate="txtEngno" SetFocusOnError="True">Engine Num Required</asp:RequiredFieldValidator></td>

</tr>

<tr>

<td><asp:Label ID="lblChasisno" runat="server" Text="Chasis No"></asp:Label></td>

<td>

<asp:TextBox ID="txtChasisno" runat="server"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="RequiredFieldValidator5" runat="server" ErrorMessage="\*" ControlToValidate="txtChasisno" SetFocusOnError="True">Chasis Num required</asp:RequiredFieldValidator></td>

</tr>

<tr>

<td><asp:Label ID="lblFueltype" runat="server" Text="Fuel Type"></asp:Label></td>

<td>

<asp:DropDownList ID="ddlFueltype" runat="server">

<asp:ListItem Value="Select">Select</asp:ListItem>

<asp:ListItem>Petrol</asp:ListItem>

<asp:ListItem>Desiel</asp:ListItem>

</asp:DropDownList></td>

<td><asp:RequiredFieldValidator ID="RequiredFieldValidator6" runat="server" ErrorMessage="\*" ControlToValidate="ddlFueltype" InitialValue="Select" SetFocusOnError="True">Select Fuel</asp:RequiredFieldValidator></td>

</tr>

<tr>

<td><asp:Label ID="lblTankcapacity" runat="server" Text="Tank Capacity in lts"></asp:Label></td>

<td>

<asp:TextBox ID="txtTankcapacity" runat="server"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="RequiredFieldValidator7" runat="server" ErrorMessage="\*" ControlToValidate="txtTankcapacity">Fuel capacity required</asp:RequiredFieldValidator></td>

</tr>

<tr>

<td><asp:Label ID="lblSeating" runat="server" Text="No of Seating"></asp:Label></td>

<td>

<asp:TextBox ID="txtSeating" runat="server"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="RequiredFieldValidator8" runat="server" ErrorMessage="\*" ControlToValidate="txtSeating">Num of Seating required</asp:RequiredFieldValidator></td>

</tr>

<%--<tr>

<td><asp:Label ID="lblMeterreading" runat="server" Text="Meter Reading"></asp:Label></td>

<td>

<asp:TextBox ID="TextBox8" runat="server"></asp:TextBox></td>

<td></td>

</tr>--%>

<tr>

<td><asp:Label ID="lblAc" runat="server" Text="A/C "></asp:Label></td>

<td>

<asp:DropDownList ID="ddlAc" runat="server">

<asp:ListItem Value="Select">Select</asp:ListItem>

<asp:ListItem Value="Yes">Yes</asp:ListItem>

<asp:ListItem Value="No">No</asp:ListItem>

</asp:DropDownList></td>

<td><asp:RequiredFieldValidator ID="RequiredFieldValidator9" runat="server" ErrorMessage="\*" ControlToValidate="ddlAc" InitialValue="Select">Select</asp:RequiredFieldValidator></td>

</tr>

<tr>

<td>

<asp:Label ID="lblCost" runat="server" Text="Vehicle Cost" Height="20px" Width="84px"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtVehiclecost" runat="server"></asp:TextBox>

</td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator10" runat="server" ErrorMessage="\*" ControlToValidate="txtVehiclecost" SetFocusOnError="True">Cost Required</asp:RequiredFieldValidator></td>

</tr>

<tr>

<td>

<asp:Label ID="lblPurchasedate" runat="server" Text="Purchase Date" Height="20px" Width="95px"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtPurchasedate" onclick="ds\_hh(this)" onfocus="ds\_hh(this)" runat="server"></asp:TextBox>

</td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator11" runat="server" ErrorMessage="\*" ControlToValidate="txtPurchasedate" SetFocusOnError="True">Date Required</asp:RequiredFieldValidator></td>

</tr>

<%--<tr>

<td>

<asp:Label ID="lblPrice" runat="server" Text="Price" Height="20px" Width="84px"></asp:Label>

</td>

<td>

</td>

<td></td>

</tr>--%>

<tr>

<td>

<asp:Label ID="lblInsurence" runat="server" Text="Insurance Amt" Height="20px" Width="92px"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtInsuranceamt" runat="server"></asp:TextBox>

</td>

<td>

</td>

</tr>

<tr>

<td>

<asp:Label ID="lblExpiredate" runat="server" Text="Insurance Expire date" Height="20px" Width="96px"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtInsuranceexpdate" onclick=ds\_hh(this) runat="server"></asp:TextBox>

</td>

<td>

</td>

</tr>

<tr>

<td colspan="3" align="center">

<asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit\_Click" /> &nbsp;

<asp:Button ID="btnBack" runat="server" Text="Back" PostBackUrl="~/Home.aspx" CausesValidation="False" />

</td>

</tr>

</table>

</asp:Content>

Vehiclemasterrpt

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" AutoEventWireup="true" CodeFile="Vehiclemasterrpt.aspx.cs" Inherits="Vehiclemasterrpt" Title="Untitled Page" %>

<%@ Register Assembly="CrystalDecisions.Web, Version=10.2.3600.0, Culture=neutral, PublicKeyToken=692fbea5521e1304"

Namespace="CrystalDecisions.Web" TagPrefix="CR" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<CR:CrystalReportViewer ID="crvVehiclemaster" runat="server" AutoDataBind="True" Height="1039px" ReportSourceID="CrystalReportSource1" Width="901px" />

<CR:CrystalReportSource ID="CrystalReportSource1" runat="server">

<Report FileName="Vehiclemaster.rpt">

</Report>

</CR:CrystalReportSource>

</asp:Content>

Vehicletarff

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" AutoEventWireup="true" CodeFile="Vehicletariff.aspx.cs" Inherits="Vehicletariff" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<table align="center">

<tr>

<td colspan="3" align="center"><h3>Add Vehicle Tariff</h3></td>

</tr>

<tr>

<td colspan="3" align="center">

<asp:Label ID="lblMessage" runat="server" Visible="False" CssClass="lblMessage"></asp:Label>

</td>

</tr>

<tr>

<td>

<asp:Label ID="lblVehicletype" runat="server" Text="Vehicle Type"></asp:Label></td>

<td>

<asp:DropDownList ID="ddlVehicletype" runat="server">

</asp:DropDownList>

</td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ErrorMessage="\*" ControlToValidate="ddlVehicletype" InitialValue="Select"></asp:RequiredFieldValidator>

</td>

</tr>

<%--<tr>

<td><asp:Label ID="lblBaseprice" runat="server" Text="Base Price"></asp:Label></td>

<td>

<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ErrorMessage="\*"></asp:RequiredFieldValidator></td>

</tr>--%>

<tr>

<td><asp:Label ID="lblAc" runat="server" Text="A/C"></asp:Label></td>

<td>

<asp:DropDownList ID="ddlAc" runat="server">

<asp:ListItem>Select</asp:ListItem>

<asp:ListItem>Yes</asp:ListItem>

<asp:ListItem>No</asp:ListItem>

</asp:DropDownList></td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ErrorMessage="\*" ControlToValidate="ddlAc" InitialValue="Select"></asp:RequiredFieldValidator></td>

</tr>

<%--<tr>

<td><asp:Label ID="lblPrice" runat="server" Text="Price per Km"></asp:Label></td>

<td>

<asp:TextBox ID="TextBox3" runat="server"></asp:TextBox></td>

<td><asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server" ErrorMessage="\*"></asp:RequiredFieldValidator></td>

</tr>--%>

<tr>

<td>

<asp:Label ID="lblKm1" runat="server" Text="Base Price"></asp:Label></td>

<td>

<asp:TextBox ID="txtBaseprice" runat="server"></asp:TextBox></td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server" ErrorMessage="\*" ControlToValidate="txtBaseprice" SetFocusOnError="True"></asp:RequiredFieldValidator></td>

</tr>

<tr>

<td>

<asp:Label ID="lblKm2" runat="server" Text="KM's"></asp:Label></td>

<td>

<asp:TextBox ID="txtKms" runat="server"></asp:TextBox></td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server" ErrorMessage="\*" ControlToValidate="txtKms"></asp:RequiredFieldValidator></td>

</tr>

<tr>

<td>

<asp:Label ID="lblExtrakm" runat="server" Text="Extra Price/KM"></asp:Label></td>

<td>

<asp:TextBox ID="txtExtrakm" runat="server"></asp:TextBox></td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator5" runat="server" ErrorMessage="\*" ControlToValidate="txtExtrakm"></asp:RequiredFieldValidator></td>

</tr>

<tr>

<td>

<asp:Label ID="lblExtrahour" runat="server" Text="Extra Price/Hour"></asp:Label>

</td>

<td>

<asp:TextBox ID="txtExtrahour" runat="server"></asp:TextBox></td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator6" runat="server" ErrorMessage="\*" ControlToValidate="txtExtrahour"></asp:RequiredFieldValidator></td>

</tr>

<tr>

<td><asp:Label ID="lblStandcharges" runat="server" Text="Stand Charge per Day"></asp:Label></td>

<td>

<asp:TextBox ID="txtStandcharges" runat="server"></asp:TextBox></td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator7" runat="server" ErrorMessage="\*" ControlToValidate="txtStandcharges"></asp:RequiredFieldValidator></td>

</tr>

<tr>

<td colspan="2" align="center">

<asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit\_Click" />&nbsp;

<asp:Button ID="btnBack" runat="server" Text="Back" PostBackUrl="~/Home.aspx" CausesValidation="False" />

</td>

</tr>

</table>

</asp:Content>

VeiwClientbooking

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" AutoEventWireup="true" CodeFile="Viewclientbooking.aspx.cs" Inherits="Viewclientbooking" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<table align="center">

<tr>

<td>

<asp:GridView ID="grdClient" runat="server" AutoGenerateColumns="False" OnRowEditing="grdClient\_RowEditing" OnRowUpdating="grdClient\_RowUpdating" OnRowCancelingEdit="grdClient\_RowCancelingEdit" AllowPaging="True" OnPageIndexChanging="grdClient\_PageIndexChanging" PageSize="5" OnRowDeleting="grdClient\_RowDeleting" >

<Columns>

<asp:TemplateField>

<ItemTemplate>

<input id="Hidden1" type="hidden" value='<%# Eval("bookingid") %>' runat="server"/>

</ItemTemplate>

</asp:TemplateField>

<asp:BoundField DataField="customername" HeaderText="Name" SortExpression="customername" />

<asp:BoundField DataField="phoneno" HeaderText="Phone No" SortExpression="phoneno" />

<asp:BoundField DataField="email" HeaderText="Email Id" SortExpression="email" />

<asp:BoundField DataField="pickup" HeaderText="Pickup Point" SortExpression="pickup" >

</asp:BoundField>

<asp:BoundField DataField="droppoint" HeaderText="Drop Point" SortExpression="droppoint">

</asp:BoundField>

<asp:BoundField DataField="fromdate" HeaderText="Pickup Date" SortExpression="fromdate" />

<asp:BoundField DataField="driver" HeaderText="Driver" SortExpression="driver" ReadOnly="true"/>

<asp:CommandField HeaderText="Edit" ShowDeleteButton="True" ShowEditButton="True"

ShowHeader="True" />

</Columns>

</asp:GridView>

</td>

</tr>

<tr>

<td align="center">

<asp:Button ID="Button1" runat="server" Text="Back" PostBackUrl="~/Home.aspx" />

</td>

</tr></table></asp:Content>

ViewCustomer

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" AutoEventWireup="true" CodeFile="Viewcustomers.aspx.cs" Inherits="Viewcustomers" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<table align="center">

<tr>

<td>

<asp:GridView ID="grdCustomers" runat="server" AutoGenerateColumns="False" OnRowEditing="grdCustomers\_RowEditing" OnRowUpdating="grdCustomers\_RowUpdating" OnRowCancelingEdit="grdCustomers\_RowCancelingEdit" AllowPaging="True" OnPageIndexChanging="grdCustomers\_PageIndexChanging" PageSize="5" OnRowDeleting="grdCustomers\_RowDeleting" >

<Columns>

<asp:TemplateField>

<ItemTemplate>

<input id="Hidden1" type="hidden" value='<%# Eval("id") %>' runat="server"/>

</ItemTemplate>

</asp:TemplateField>

<asp:BoundField DataField="name" HeaderText="Name" SortExpression="name" />

<asp:BoundField DataField="address" HeaderText="Address" SortExpression="address" />

<asp:BoundField DataField="phone" HeaderText="Contact Number" SortExpression="phone">

</asp:BoundField>

<asp:BoundField DataField="email" HeaderText="Email Id" SortExpression="emailid" />

<asp:CommandField HeaderText="Edit" ShowDeleteButton="True" ShowEditButton="True"

ShowHeader="True" />

</Columns>

</asp:GridView>

</td>

</tr>

<tr>

<td align="center">

<asp:Button ID="Button1" runat="server" Text="Back" PostBackUrl="~/Home.aspx" />

</td>

</tr>

</table>

</asp:Content>

ViewServiceDetail

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" AutoEventWireup="true" CodeFile="Viewservicedetails.aspx.cs" Inherits="Viewservicedetails" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<table align="center">

<tr>

<td>

<%--<asp:GridView ID="grdSevice" runat="server" AutoGenerateColumns="False" OnRowEditing="grdSevice\_RowEditing" OnRowUpdating="grdSevice\_RowUpdating" OnRowCancelingEdit="grdSevice\_RowCancelingEdit" AllowPaging="True" OnPageIndexChanging="grdSevice\_PageIndexChanging" PageSize="5" OnRowDeleting="grdSevice\_RowDeleting" >

<Columns>

<asp:TemplateField>

<ItemTemplate>

<input id="Hidden1" type="hidden" value='<%# Eval("empId") %>' runat="server"/>

</ItemTemplate>

</asp:TemplateField>

<asp:BoundField DataField="firstname" HeaderText="Name" SortExpression="firstname" />

<asp:BoundField DataField="designation" HeaderText="Designation" SortExpression="designation" />

<asp:BoundField DataField="salery" HeaderText="Salery" SortExpression="salery" />

<asp:BoundField DataField="dateofjoin" HeaderText="Joiningdate" SortExpression="dateofjoin" >

</asp:BoundField>

<asp:BoundField DataField="phone" HeaderText="Contact Number" SortExpression="phone">

</asp:BoundField>

<asp:BoundField DataField="emailid" HeaderText="Email Id" SortExpression="emailid" />

<asp:CommandField HeaderText="Edit" ShowDeleteButton="True" ShowEditButton="True"

ShowHeader="True" />

</Columns>

</asp:GridView>--%>

</td>

</tr>

</table>

</asp:Content>

ViewVehicle

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" AutoEventWireup="true" CodeFile="Viewvehilce.aspx.cs" Inherits="Viewvehilce" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

</asp:Content>

ViewVehiclemaster

<%@ Page Language="C#" MasterPageFile="~/Fleetmanagement.master" AutoEventWireup="true" CodeFile="ViewVehilcemaster.aspx.cs" Inherits="ViewVehilcemaster" Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

<table align="center">

<tr>

<td>

<asp:GridView ID="grdVehiclemaster" runat="server" AutoGenerateColumns="False" OnRowEditing="grdVehiclemaster\_RowEditing" OnRowUpdating="grdVehiclemaster\_RowUpdating" OnRowCancelingEdit="grdVehiclemaster\_RowCancelingEdit" AllowPaging="True" OnPageIndexChanging="grdVehiclemaster\_PageIndexChanging" PageSize="5" OnRowDeleting="grdVehiclemaster\_RowDeleting" >

<Columns>

<asp:BoundField DataField="vehicletype" HeaderText="Vehicle" SortExpression="vehicletype" />

<asp:BoundField DataField="regno" HeaderText="Registration No" SortExpression="regno" ReadOnly="true"/>

<asp:BoundField DataField="fueltype" HeaderText="Fuel" SortExpression="fueltype" />

<asp:BoundField DataField="purchasedate" HeaderText="Purchased Date" SortExpression="purchasedate" />

<asp:BoundField DataField="noofseating" HeaderText="Seating No" SortExpression="noofseating" />

<asp:BoundField DataField="vehiclecost" HeaderText="Vehicle Cost" SortExpression="vehiclecost" />

<asp:CommandField HeaderText="Edit" ShowDeleteButton="True" ShowEditButton="True"

ShowHeader="True" />

</Columns>

</asp:GridView>

</td>

</tr>

<tr>

<td align="center">

<asp:Button ID="Button1" runat="server" Text="Back" PostBackUrl="~/Home.aspx" />

</td>

</tr>

</table>

</asp:Content>

Style.CSS

body {

}

.lblMessage

{

font-size:large;

font-style:normal;

font:15;

color:Green;

}

.ds\_box {

background-color: #FFF;

border: 1px solid #000;

position:absolute;

/\*z-index: 32767;\*/

z-index: 999

}

.ds\_tbl {

background-color: #FFF;

}

.ds\_head {

background-color: #333;

color: #FFF;

font-family: Arial, Helvetica, sans-serif;

font-size: 13px;

font-weight: bold;

text-align: center;

letter-spacing: 2px;

}

.ds\_subhead {

background-color: #CCC;

color: #000;

font-size: 12px;

font-weight: bold;

text-align: center;

font-family: Arial, Helvetica, sans-serif;

width: 32px;

}

.ds\_cell {

background-color: #EEE;

color: #000;

font-size: 13px;

text-align: center;

font-family: Arial, Helvetica, sans-serif;

padding: 5px;

cursor: pointer;

}

.ds\_cell:hover {

/\*background-color: #F3F3F3;\*/

background-color:Green;

} /\* This hover code won't work for IE \*/

Testing

Testing is a process rather than a single activity. This process starts from test

planning then designing test cases, preparing for execution and evaluating status

till the test closure. So, we can divide the activities within the fundamental test

process into the following basic steps:

Planning

Specification

Execution

Recording

Checking for Test Completion

**Planning:** This Test Planning activity produces a test plan specific to a level of

testing (e.g. system testing). These test level specific test plans should state how

the test strategy and project test plan apply to that level of testing and state any

exceptions to them. When producing a test plan, clearly define the scope of the

testing and state all the assumptions being made. Identify any other software

required before testing can commence (e.g. stubs & drivers, word processor,

spreadsheet package or other 3rd party software) and state the completion criteria

to be used to determine when this level of testing is complete.

**Planning for testing of OSMS**

1. Time taken for testing of **osms** should be around 3-5 weeks.

2. The level of testing should be taken to Acceptance Testing.

3. System requirement for testing environment, such as hardware – CPU,

Keyboard, Mouse, Printer, scanner etc., software - OS (windows xp, 7, 8),

MS-Office etc.

4. Completion criteria for testing:

100% of test cases have been run;

80% of high severity faults fixed;

80% of low & medium severity faults fixed;

Time has run out;

Testing budget is used up.

**Specification:** The fundamental test process describes this activity as designing

the test cases using the techniques selected during planning. For each test case,

specify its objective, the initial state of the software, the input sequence and the

expected outcome.

**Specification can be considered as three separate tasks:**

Identify test conditions

Design test cases – determine ‘how’ the identified test conditions are going

to be exercised;

Build test cases – implementation of the test cases (scripts, data, etc.).

Specification for testing

Test conditions for **osms**:

Login

User Management

Admin Management

Shopping Activity

These test conditions will be examined manually.

**Execution:** The purpose of this activity is to execute all of the test cases (though

not necessarily all in one go). This can be done either manually or with the use of

a test execution automation tool.

In this project the test condition will be examined manually.

**Recording:** In practice the Test Recording activity is done in parallel with Test

Execution. To start with we need to record the versions of the software under test

and the test specification being used. Then for each test case we should record the

actual outcome and the test coverage levels achieved for those measures specified

as test completion criteria in the test plan. In this way we will be marking off our

progress.

The output of test cases will be stored in a excel sheet for analysis and report

generation that will help in future.

**Checking for Test Completion:** This activity has the purpose of checking the

records against the completion criteria specified in the test plan. If these criteria

are not met, it will be necessary to go back to the specification stage to specify

more test cases to meet the completion criteria. There are many different types of

coverage measure and different coverage measures apply to different levels of

testing.

The checking for Test Completion is done by co-relating with the document of

planning of testing.

**LEVELS OF TESTING**

In order to uncover the errors present in different phases we have the concept of levels of testing. The basic levels of testing are

Client Needs Acceptance Testing

Requirements System Testing

Design Integration Testing

Code Unit Testing

A series of testing is done for the proposed system before the system is ready for the user acceptance testing.

The steps involved in Testing are:

**Unit Testing:**

Unit testing focuses verification efforts on the smallest unit of the software design**,** the module**.** This is also known as “Module Testing”**.** The modules are tested separately**.** This testing carried out during programming stage itself**.** In this testing each module is found to be working satisfactorily as regards to the expected output from the module**.**

**Integration Testing:**

Data can be grossed across an interface**;** one module can have adverse efforts on another**.** Integration testing is systematic testing for construction the program structure while at the same time conducting tests to uncover errors associated with in the interface. The objective is to take unit tested modules and build a program structure**.** All the modules are combined and tested as a whole**.** Here correction is difficult because the isolation of cause is complicate by the vast expense of the entire program. Thus in the integration testing stop**,** all the errors uncovered are corrected for the text testing steps**.**

**System testing:**

System testing is the stage of implementation that is aimed at ensuring that the system works accurately and efficiently for live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, then goal will be successfully achieved.

**Validation Testing:**

At the conclusion of integration testing software is completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of software tests begins**,** validation test begins**.** Validation test can be defined in many ways**.** But the simple definition is that validation succeeds when the software function in a manner that can reasonably expected by the customer. After validation test has been conducted one of two possible conditions exists.

One is the function or performance characteristics confirm to specifications and are accepted and the other is deviation from specification is uncovered and a deficiency list is created. Proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

**Output Testing:**

After performing validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated by the system under consideration. Here the output format is considered in two ways, one is on the screen and other is the printed format. The output format on the screen is found to be correct as the format was designed in the system designed phase according to the user needs. For the hard copy also the output comes as the specified requirements by the users. Hence output testing does not result any corrections in the system.

**User Acceptance Testing:**

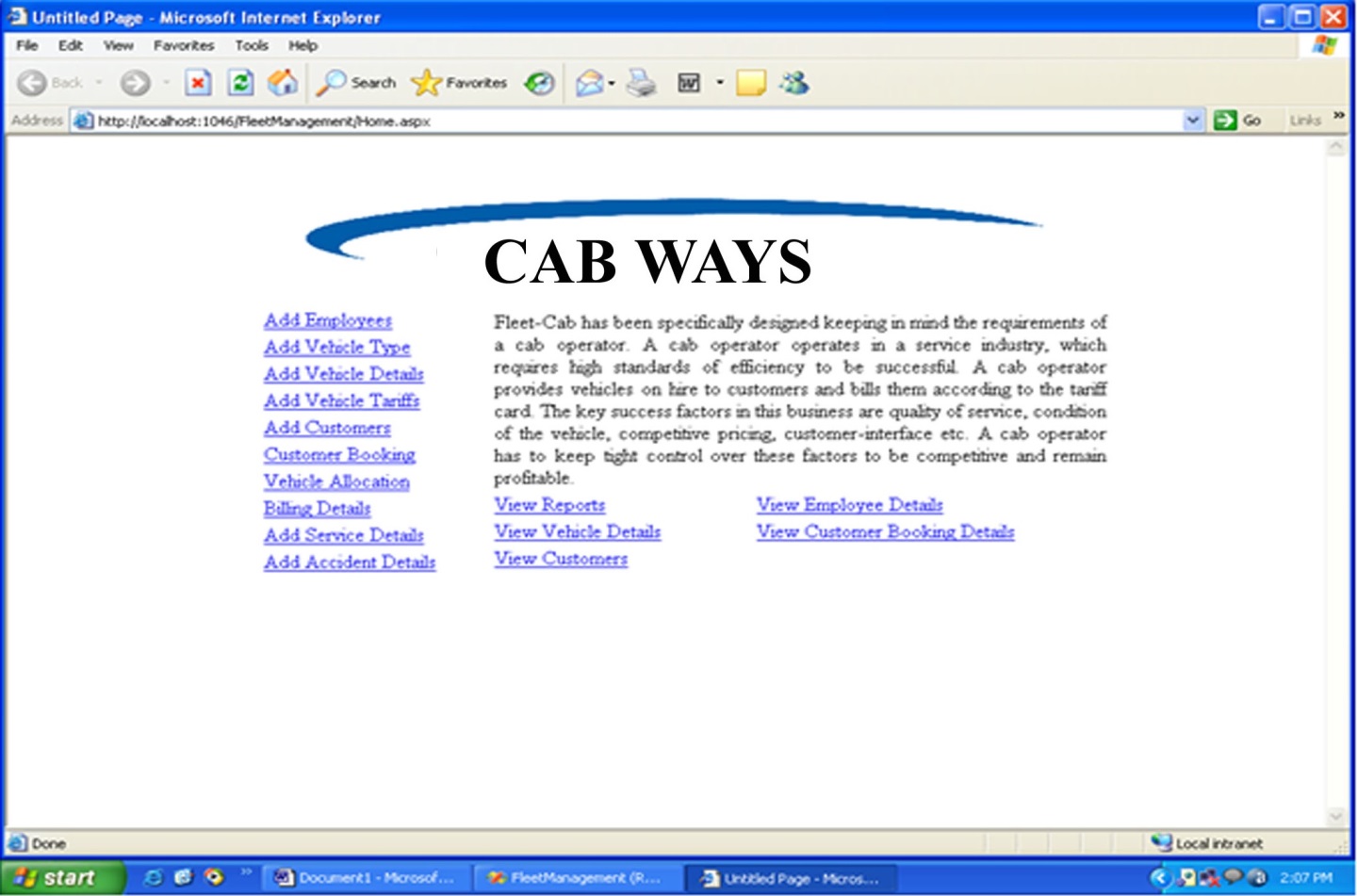
User acceptance of a system is the key factor of the success of any system. The system under study is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required.

**Test Data:**

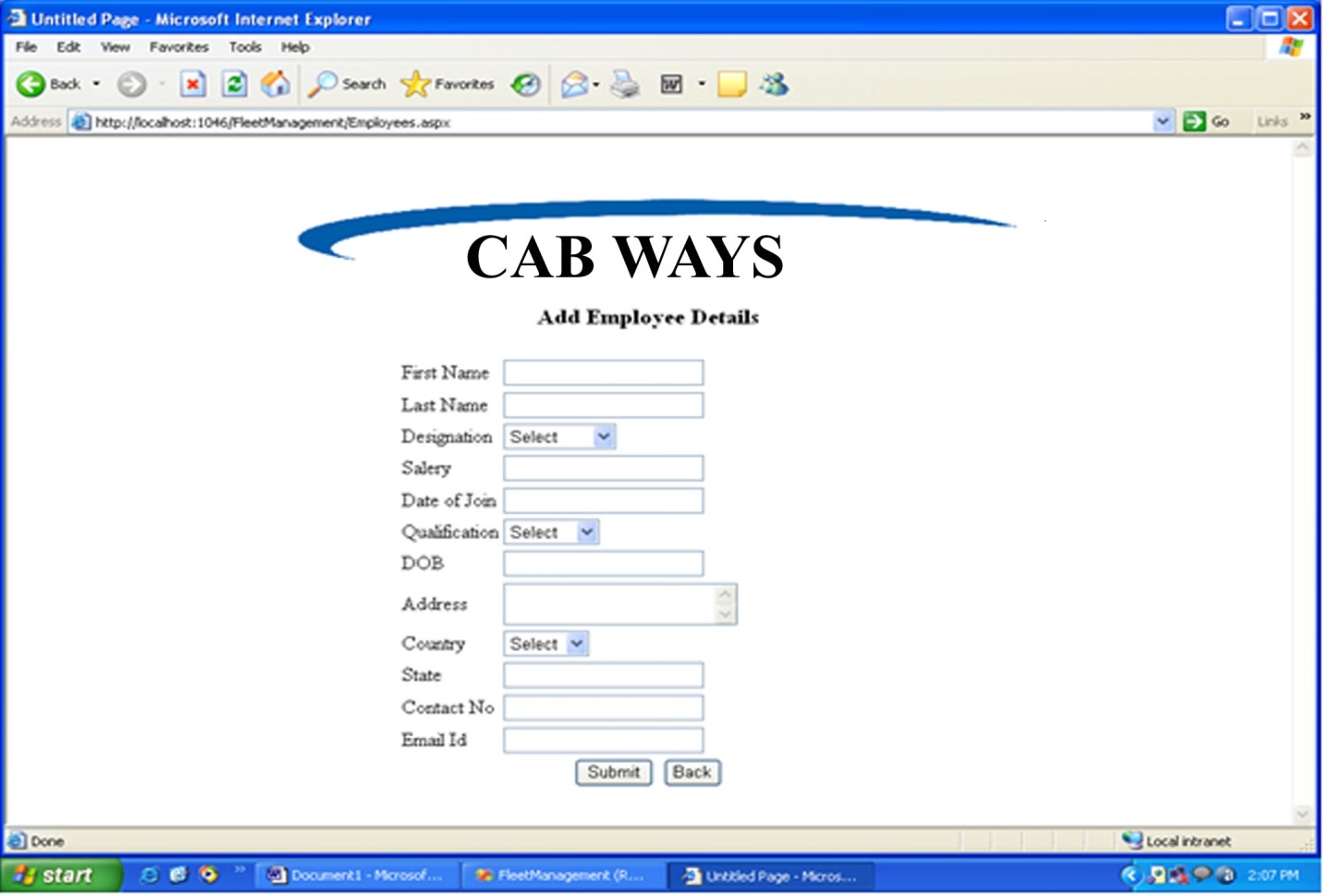
Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing after preparing the test data the system under study is tested using the test data. While testing the system by using the test data errors are again uncovered and corrected by using above testing steps and corrections are also noted from the future use.

Input And Output Screens

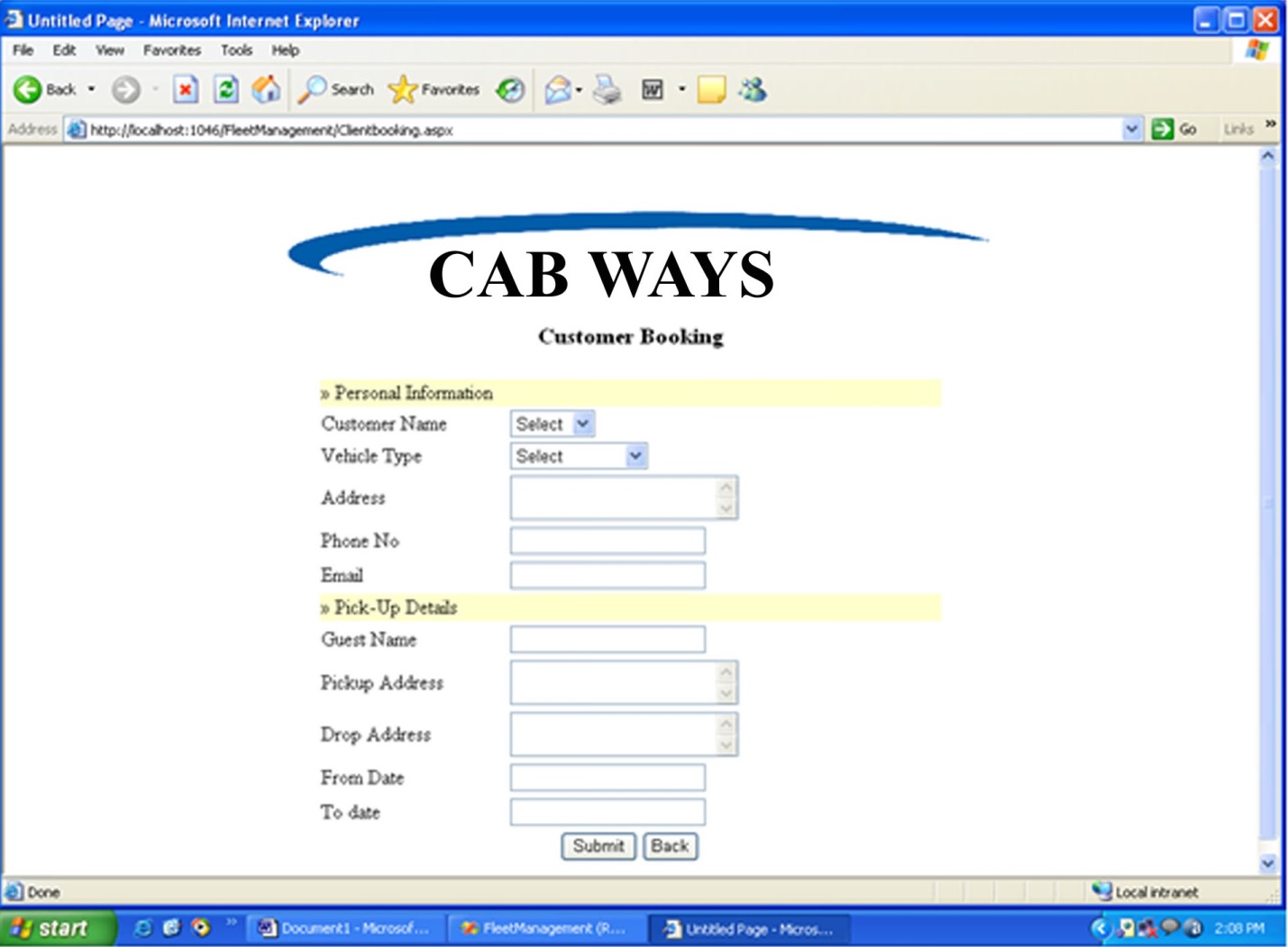
Homepage



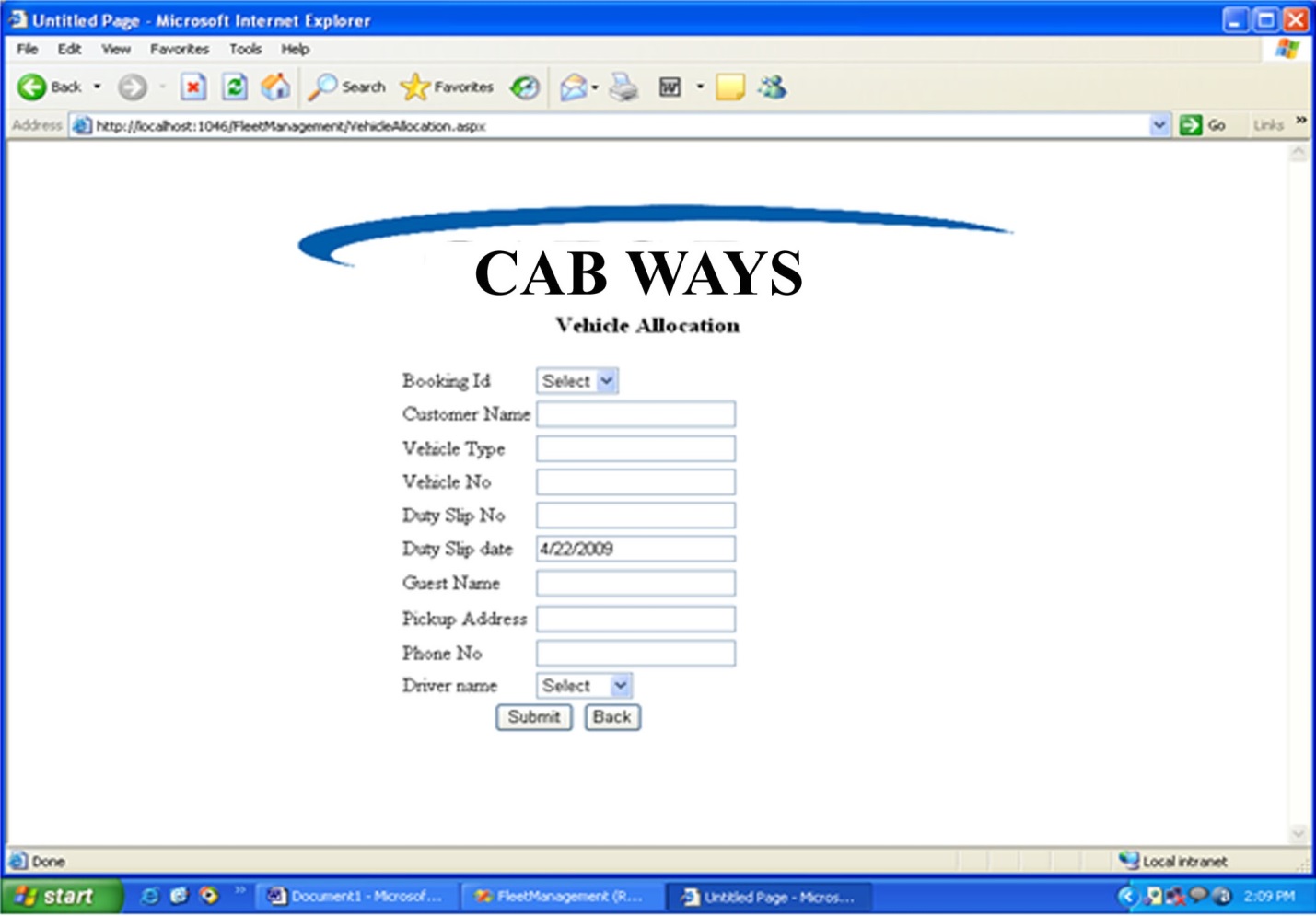
ADD Employee Details Page



Customer Booking Page



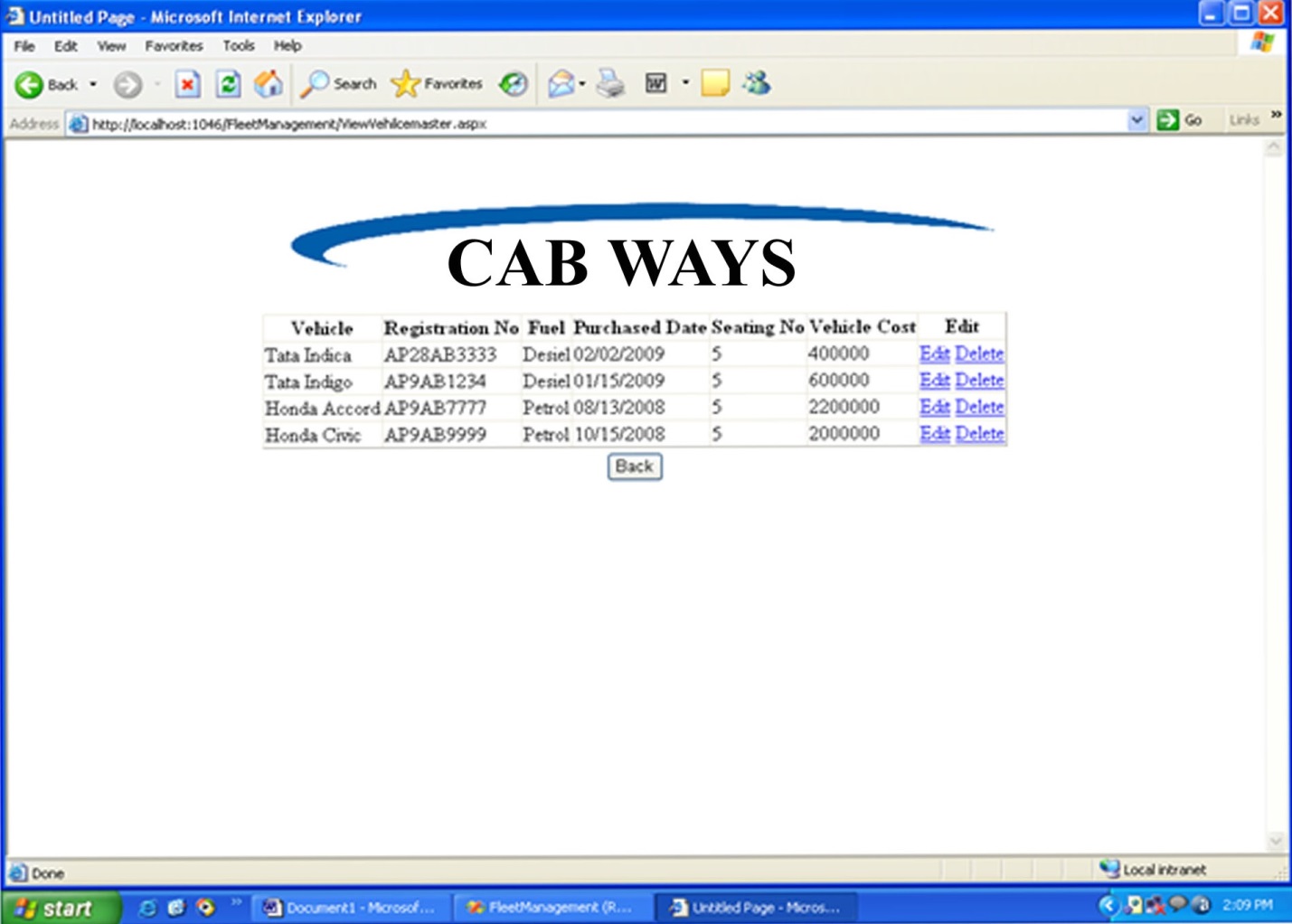
Vehicle allocation page



Allocation Details



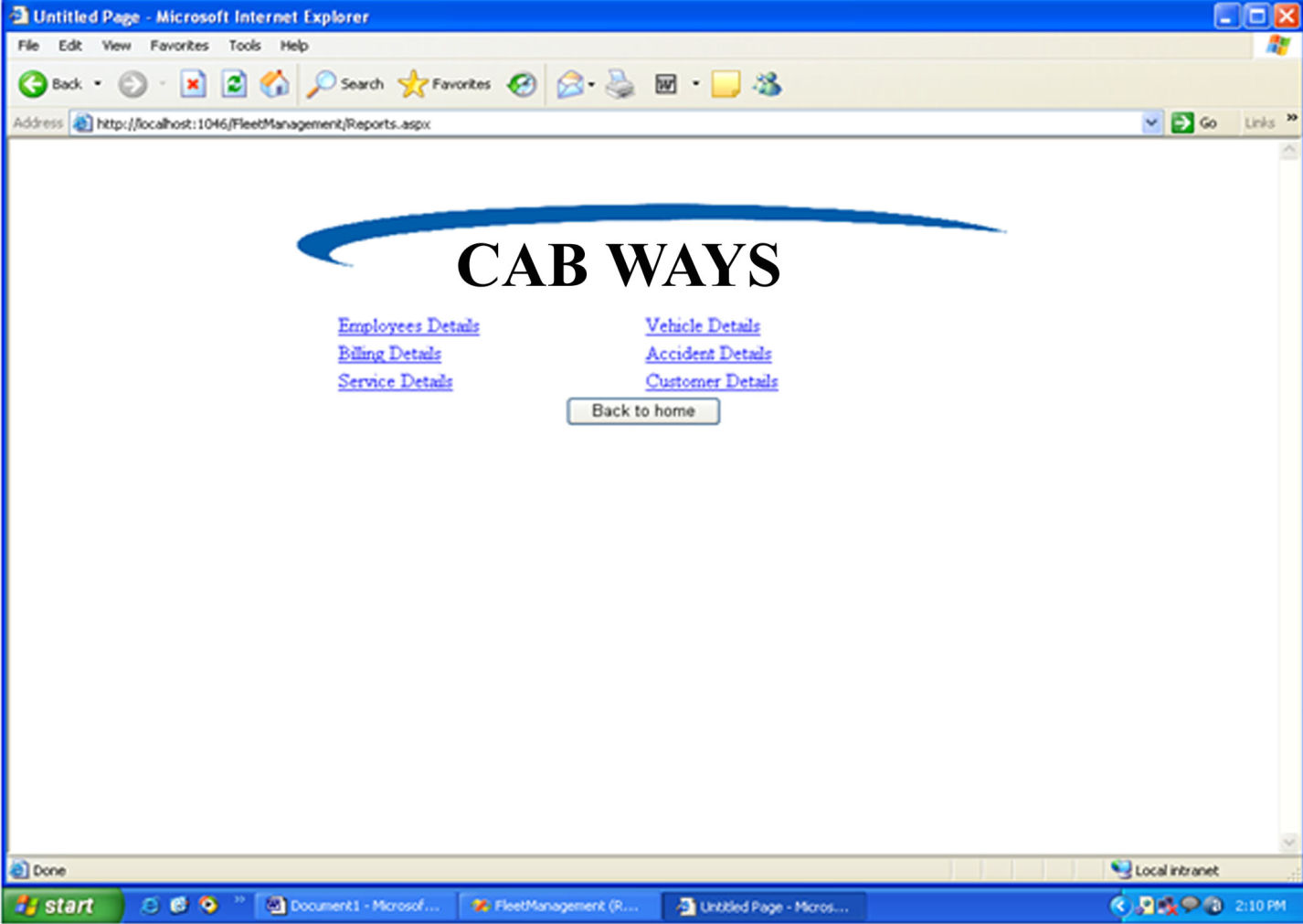
Report of Allocation of vehicle



Report of Employee



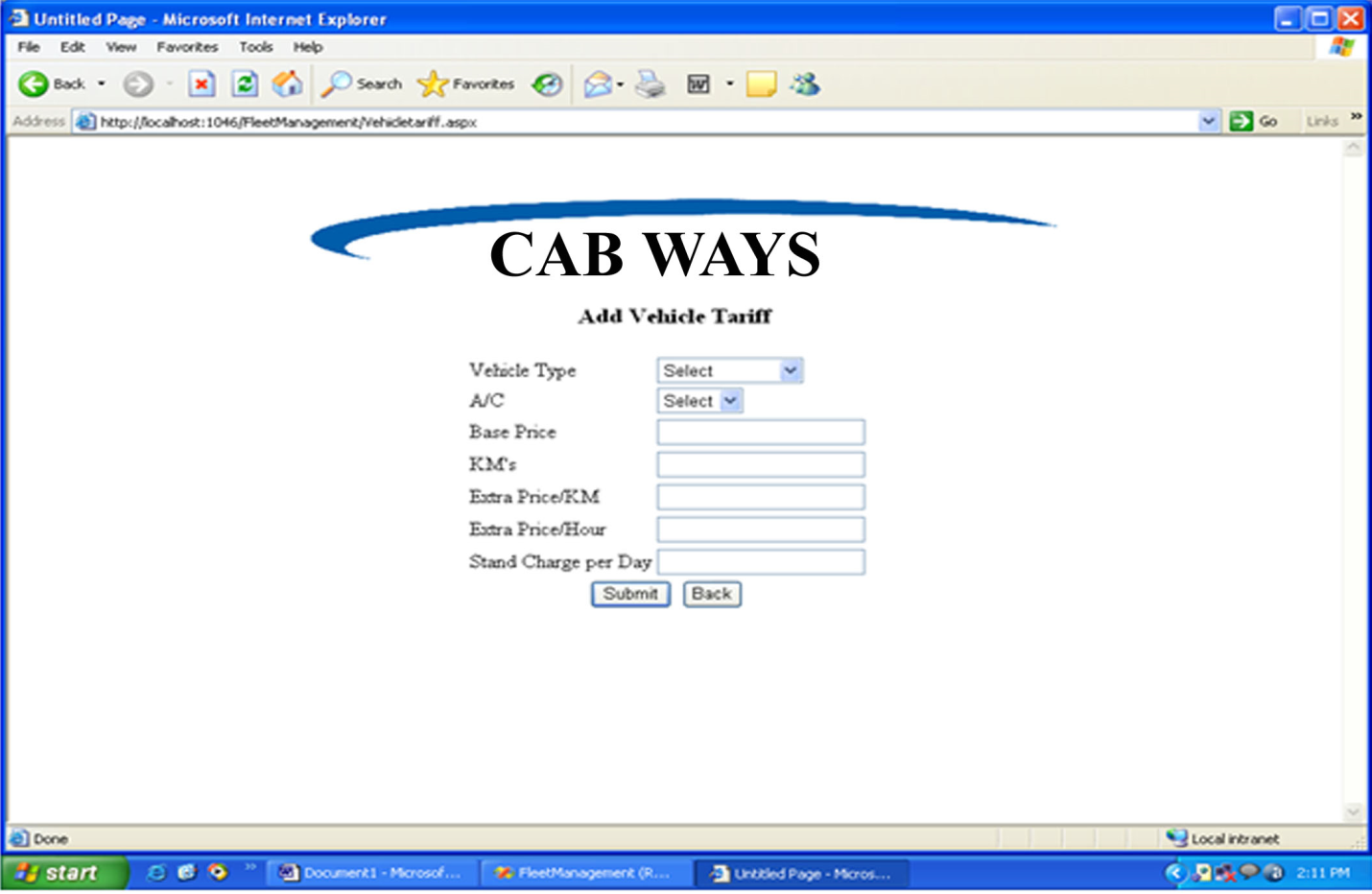
Details page



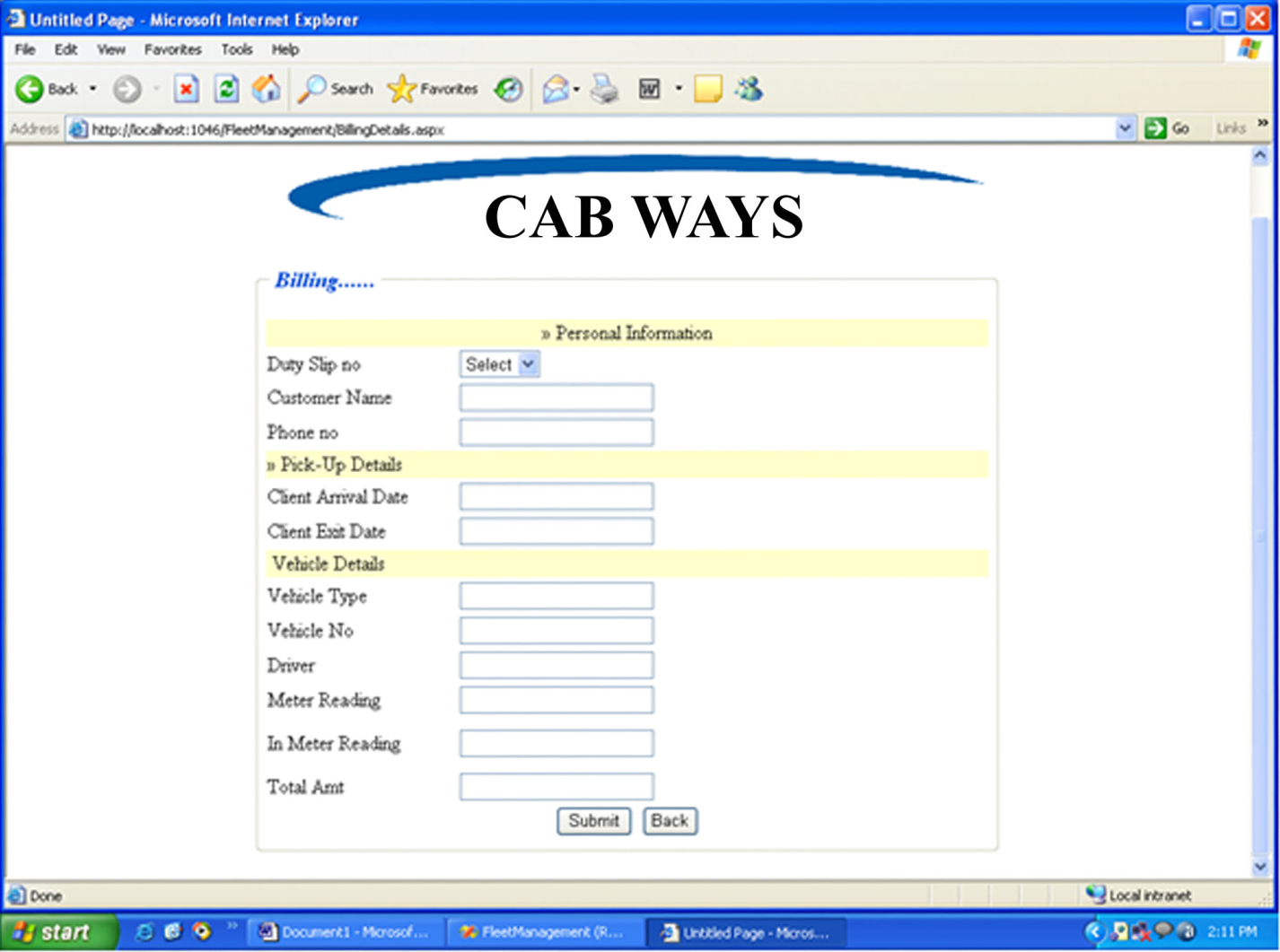
Service log page



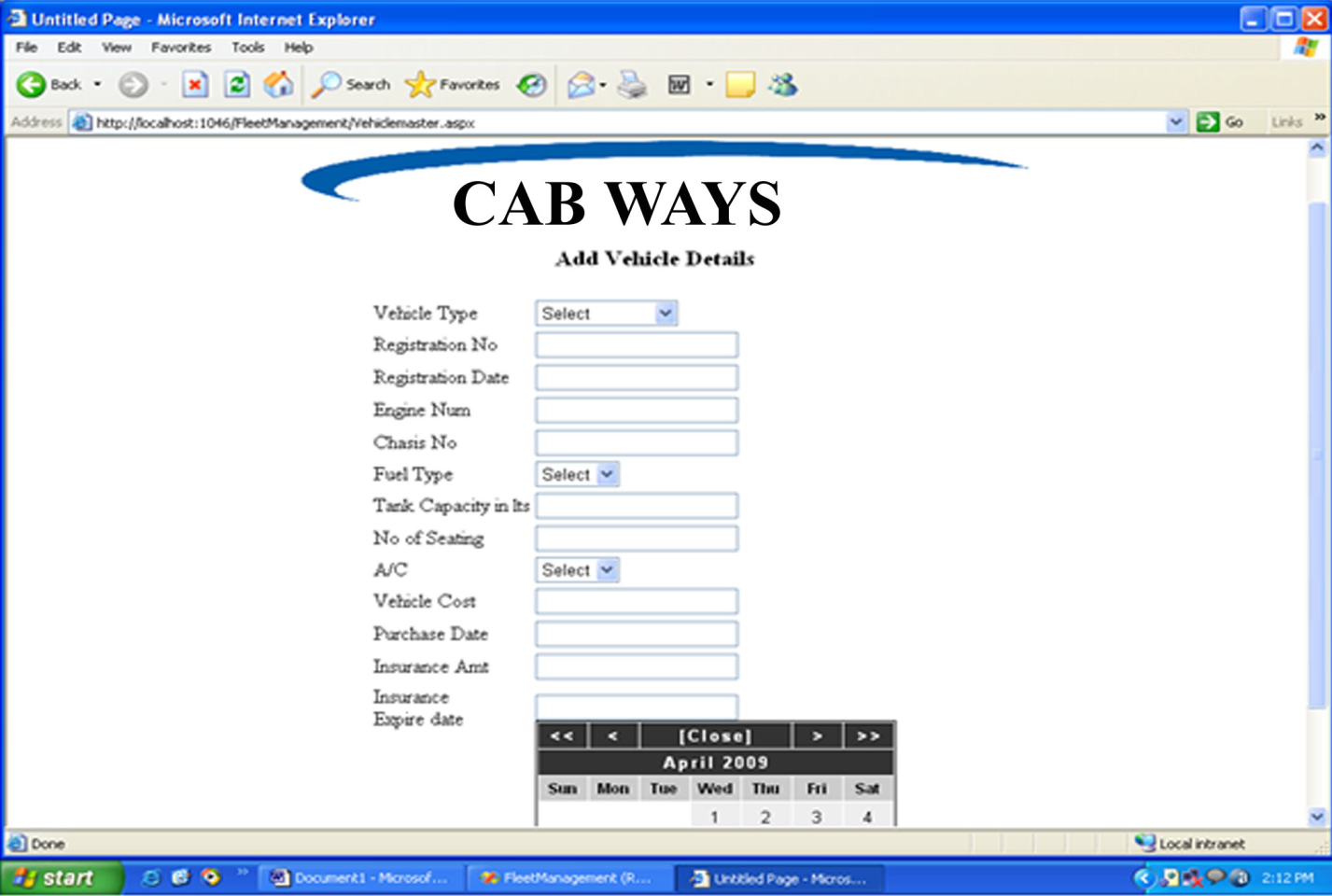
Vehicle Tariff page



Billing page



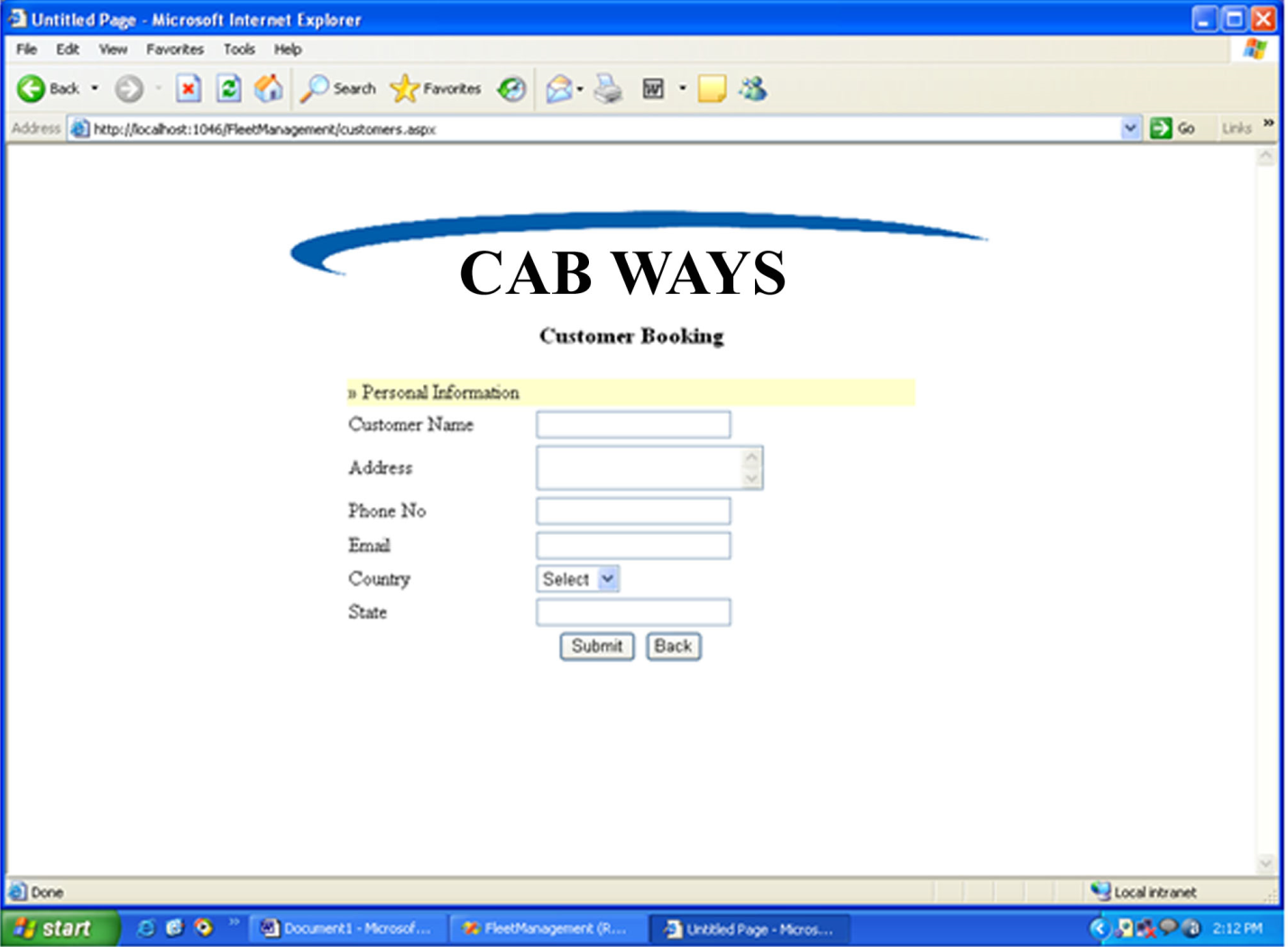
Vehicle Detail page



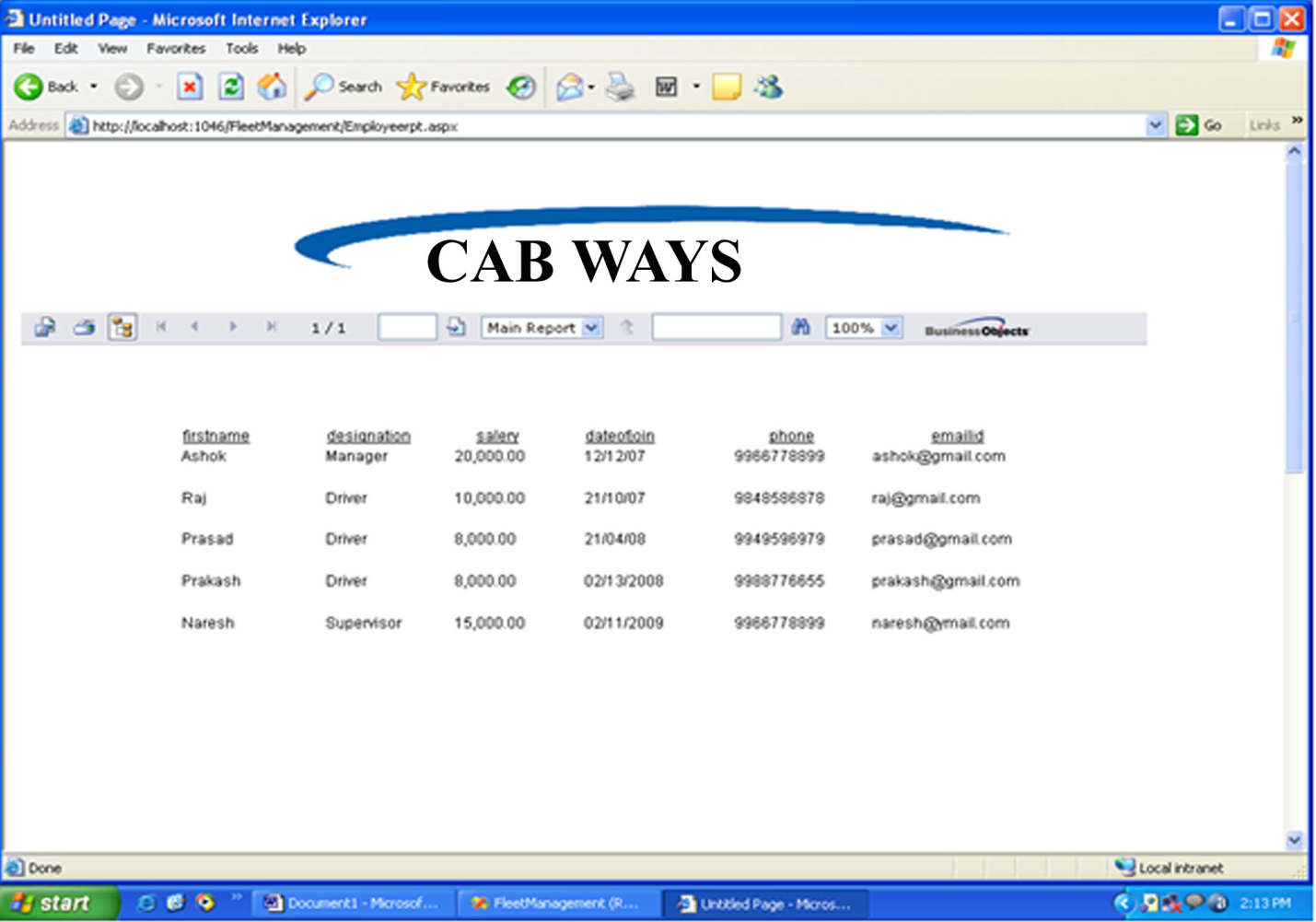
Detail of Drivers



Customer booking Page



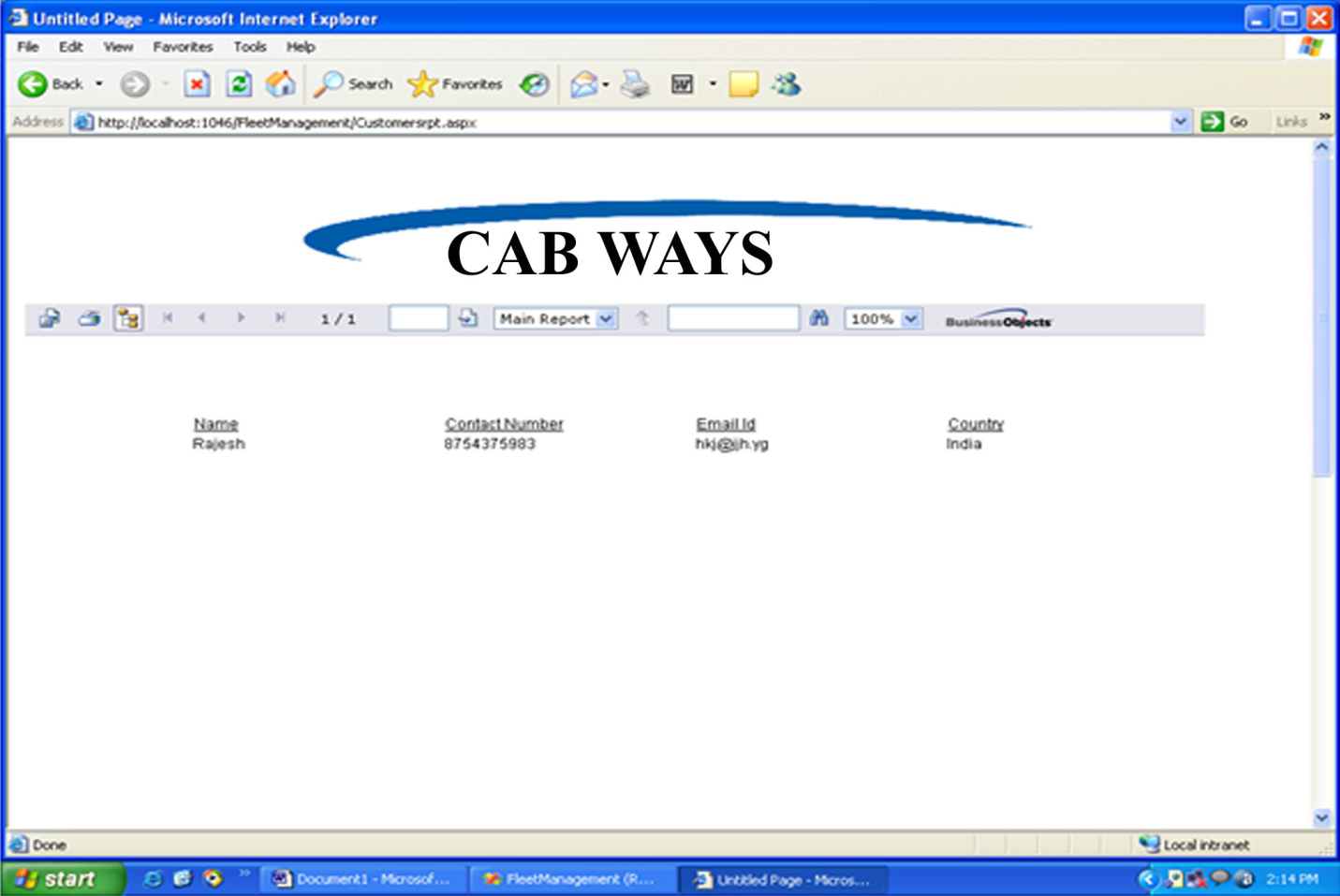
Customer report



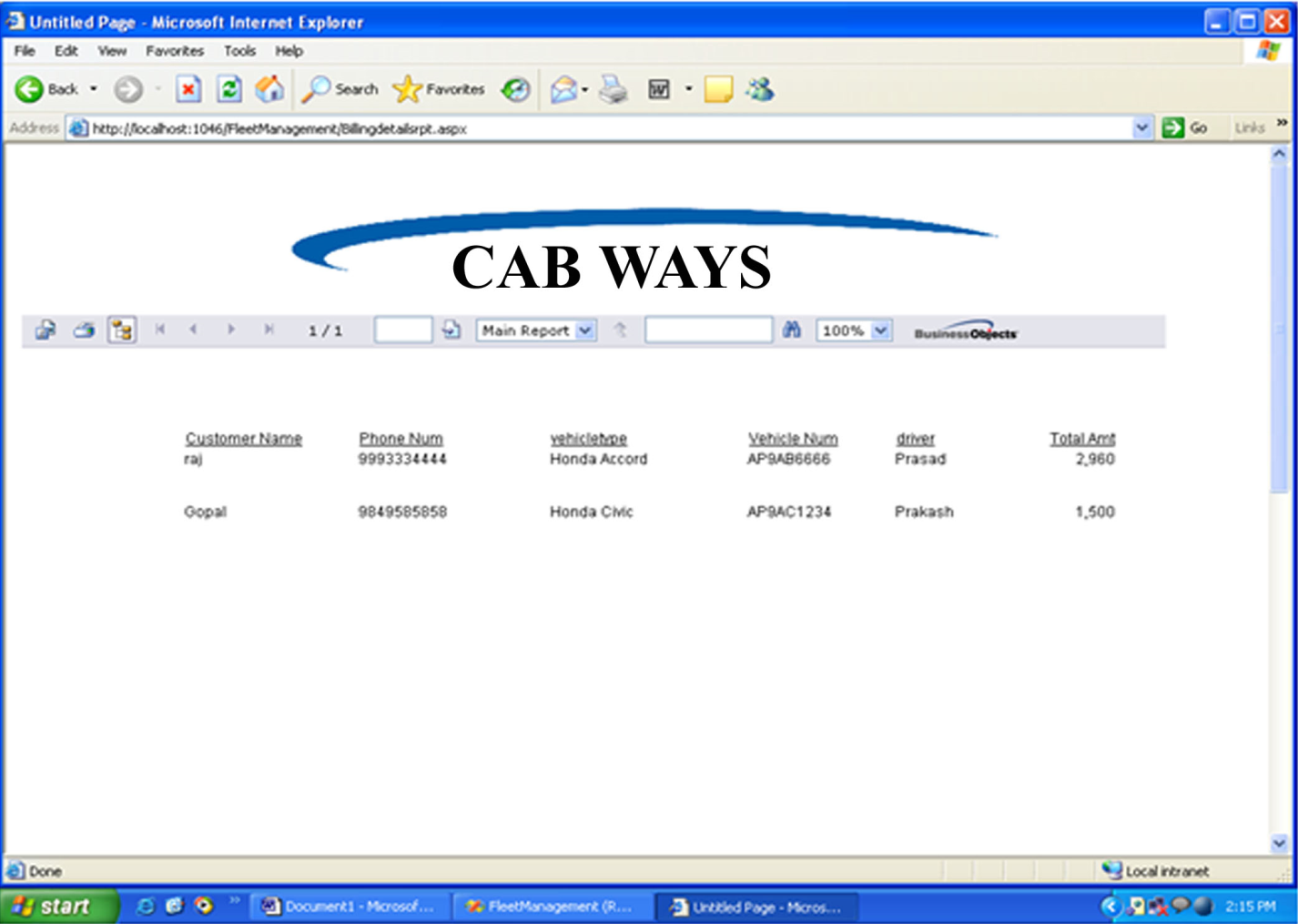
Customer Report



Customer Report









Security

The “**CABWAYS**” provides two levels of security. The 1st level of security, which is provided by ASP.NET named as FormBasedSecurity checks for authentication and authorization and prevent to enter anonymous users. The 2nd level of security provided by the database which is being used.

**- FRONT END**

We use rich set of validation controls provided by ASP.NET and where we not achieved this through validation control we implemented JavaScript for all the Client side validations. We usually use lot of client side validation not server side. The form is not submitted until user fills in correct data. It is extremely useful to restrict mistakes by user.

**- BACK END**

We have used SQL-SERVER as the Back end. SQL-Server provides efficient/effective solution for major database tech:

- Large database and space management.

- Many concurrent database users.

- High transaction processing requirement

- High Availability

- Industry accepted standards

- Manageable security

- Portability

**FRONT END SECURITY**

This security level is developed or designed by the s/w developer or designer. S/he provides her/his s/w the security for the system, by considering many factors regarding to the particular s/w. Following functionality are used to provide security in this Software.

**LOGIN CHECKING**

There are two login’s are provided in this software one for user/patient and another for admin. In order to avail the services of user have to provide a valid id and password and only after entering correct id and password user will get entry otherwise they will be blocked from accessing any services of this site. Similarly a login check is provided for admin that check credentials for admin which can do any admin job. In this way security have been provided for both user/patient section and admin section. There is also provision to change the password in case password is known by other.

**BACK END SECURITY**

In this particular software our back end used is SQL-Server. So all the inbuilt security aspects provided by the SQL-Server database is used as it is. It will provide the strong feature of security so that it will be difficult to change, modify any personal or university data. Information is vital to success, but when damaged or in the wrong hands, it can threaten success. SQL-Server provides extensive security features to safe guard your information from both unauthorized viewing and intentional or inadvertent damage. This security is provided by granting or revoking privileges on a person- by- person and privilege-by-privilege basis.

Future Scope

The future scope of “**CABWAYS**” is completely depends on the fulfillment of requirements and management of database in sense of not only storing & accessing but also searching, sorting and updating the database. I will develop my software using front end Visual Basic .Net - 2008 (Framework 3.5) and Database Microsoft SQL Server - 2008, which is best for a medium level Organization. Same vendor Microsoft develops both tools front-end and back-end, so there is no confliction between database connectivity. So, the database is properly managed & maintained. All types of reports and statements will easily generate and lots of data are handled in a systematic manner. I will try to implement back up, data recovery, and zipping mechanism also in my proposed system. The primary scope of this system will be:

* Protection from unauthorized access
* New Module can be easily added in existing system.
* Integration of system at the data & functions.
* Balance local & official information requirements
* Less paper driven systems and Streamline approval procedures
* Better decision support systems and Customized printed reports

Ultimately, it can be say that the future scope of this application is better and may be some further enhancement as per the requirements for “**CABWAYS”** for cab Booking and Tracking**.**

Bibliography

The following books were referred during the analysis and execution phase of the project

**SOFTWARE ENGINEERING**

By Roger.S. Pressman

**COMPLETE HTML**

Steven Holzner

**UNIFIED MODELING LANGUAGE**

By Grady Booch,Ranbaugh,Jacobson

**SQL FOR PROFESSIONALS**

By Jain

**C#.NET Black Book**

By Evangeleous Petereous

**ASP.NET**

Matthew mac Donald and Mario Szpuszta

**MSDN 2005**

By Microsoft

**Websites**

[www.w3schools.com](http://www.w3schools.com)

[www.programmersheaven.com](http://www.programmersheaven.com)

[www.codeproject.com](http://www.codeproject.com)