ANALYSIS AND DESIGN ONLINE VEHICLE REQUISITION SYSTEM FOR AUTOMOBILE SHOP

THIS PAPER IS SUBMITTED TO THE INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY (IICT), BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY (BUET) AS A PARTIAL FULFILLMENT FOR THE POST GRADUATE DIPLOMA IN INFORMATION TECHNOLOGY (PGDIT).

Submitted by –

Dhiraj Guha Biswas --- Student No. – DF0131011
Parvez Monon Ashraf --- Student No. – DF0131010
Md. Jainul Abedin Bashir --- Student No. – DF0131009

Under the Supervision of
Programmer Mr. Fazlur Rahman Khan

INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DHAKA
OCTOBER 2002
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
Dhaka – 1000

POST GRADUATE DIPLOMA IN INFORMATION TECHNOLOGY

DECLARATION

We do solemnly declare that this paper is submitted, in partial fulfillment of the requirements for the Post Graduate Diploma in Information Technology (PGDIT) of session 2001-2002 of Bangladesh University of Engineering and Technology (BUET), Dhaka, is the result of our own project work and written in our own language. That no part of this paper consists of materials, copied or plagiarized from published or unpublished work or other writers and that as all materials, borrowed or reproduced from other published or unpublished source have either been put under quotation or duly acknowledged with full reference in appropriate place(s).

Signature of Students

Dhruj Guha Biswas
(Student No. – DF0131011)

Parvez Monon Ashraf
(Student No. – DF0131010)

Md. Jainul Abedin Bashir
(Student No. – DF0131009)

Signature of Supervisor

Mr. Fazlur Rahman Khan.
Programmer
IICT, BUET
ACKNOWLEDGEMENT

It is a matter of immense pleasure to have the opportunity to present this report titled "Online Vehicle Requisition System for Automobile Shop (BUET)" for the partial fulfillment of the requirements of Post Graduate Diploma in Information Technology (PGDIT) under Institute of Information and Communication Technology (IICT), Bangladesh University of Engineering and Technology (BUET).

We express our profound gratitude and thanks to our supervisor, Md. Fazlur Rahman Khan, Programmer of IICT, BUET for his valuable advice, continuous guidance and inspiration during the entire length of this project.

We are grateful to all the teachers, officers and staffs of IICT for giving us their kind support and information to during the study.

We are also appreciative to all our friends and classmates for their valuable suggestions to develop this report.

We are also indebted to our parents whose continuous support all over the life has brought us this far in our career.

We, finally offer special thanks to the almighty Allah that we have been successful in our effort to accomplish the report.
ABSTRACT

As a mandatory requirement for the partial fulfillment of the Post Graduate Diploma in Information Technology under Institute of Information and Communication Technology, Bangladesh University of Engineering and Technology, the present report titled as “Analysis and Design On line Vehicle Requisition System for Automobile Shop (BUET)” was prepared.

Truthfully, the process of obtaining information at present is almost operated in a manual process and rest of the work is being performed with using the help of computer. Manual system takes more time and there is a scope of mistakes. A total computerized system under Internet (Web page) work can solve the problem.
Table of Content

Declaration ii
Acknowledgement iii
Abstract iv

Chapter One: Introduction

1.1 Objective of the Project 2
1.2 Background of BUET 2
1.2.1 Historical Background 2
1.2.2 The BUET Campus 3
1.2.3 Teaching staff of University 3
1.3 BUET Automobile shop 4
1.3.1 Background of Automobile shop 4
1.3.2 Vehicles 4
1.3.3 Staffs 4
1.3.4 Maintenance 5
1.4 Methodology 5
1.5 Present Requisition System 7
1.5.1 Present condition 7
1.5.2 Flow chart of information in present system 8
1.5.3 Limitations 8
1.6 Necessity of the proposed system 8
1.7 Objective of the proposed system 9
1.7.1 Proposed system 9
1.7.2 Limitations

v
Chapter Two: Project Inception

2.1 List of tasks
2.2 Resources
2.2.1 Hardware
2.2.2 Software
2.2.3 Manpower
2.3 Project Planning and Scheduling

Chapter Three: System Analysis

3.1 Use Case Diagram
3.2 Class Diagram
3.2.1 Class
3.2.2 Attribute
3.2.3 Operations
3.2.4 Classes in the System
3.3 Sequence Diagram
3.3.1 Sequence Diagram for Creating Users
3.3.2 Sequence Diagram for Booking
3.3.3 Sequence Diagram for Cancel Booking
3.3.4 Sequence Diagram for Bill Check
3.4 Code Generation
3.5 Data Requirements Analysis

Chapter Four: System Design

4.1 Create the Class Module of the System
4.2 Final Design of Database
4.3 Web page Basics and Design
List of Fig:

1. Present Requisition Form 7
2. Project Planning and Scheduling 12
3. Use Case Diagram 14
4. Class Diagram for the System 16
5. Sequence Diagram for Creating Users 20
6. Sequence Diagram for Booking 21
7. Sequence Diagram for Cancel Booking 22
8. Sequence Diagram for Check Bill 23
9. Database Relationship Diagram 26
10. Class Modules Develop In Visual basic 6 28
11. User Interface for LOGIN and Sign Up 31
12. User Interface for New User Registration form 32
13. User Interface for Cancel Requisition 33
14. User Interface for Information Details 34
15. User Interface for Car Show Room 35
16. Form for LOGIN to the System and Sign Up for the System 37
17. Requisition Form 38
18. New User Registration Form 39
19. Control Form 40
Chapter – 1

Introduction
Introduction

In order to conceptualize and understand the importance and the need of developing the online Requisition System for BUET (Bangladesh University of Engineering and Technology), it is essential to understand the requirement of the system in BUET. It is necessary for that to understand the organizational structure as being comprised of many subsystems or series of subsystems.

Here the system is designed for BUET. But it could be designed for other universities or other organizations. According to the organization and organizational approach, the system may design for them with a little change. So the system here is a model for BUET, can also used as model for other organizations too.

1.1 Objective of the Project:

Our main objective of this project is to analyze and Design the Online Vehicle Requisition System. We also try to find out the System requirement and the data requirements of the project.

1.2 Background of BUET:

Bangladesh University of Engineering and Technology, abbreviated as BUET, is situated at the centre of Dhaka city, Capital of Bangladesh.

1.2.1 Historical Background

BUET is the oldest institution for the study of Engineering and Architecture in Bangladesh. The history of this institution dates back to the days of Dhaka Survey School, which was established at Nalgola in 1876 to train surveyors for the then Government of Bengal of British India. As the years passed, the school became the
Ahsanullah School of Engineering offering three-year diploma courses in Civil, Electrical and Mechanical Engineering. In 1948, the school was upgraded to Ahsanullah Engineering College on its present premises as a Faculty of Engineering under the University of Dhaka, offering four-year bachelor's courses in Civil, Electrical and Mechanical Engineering with a view to meeting the increasing demand for engineers in the country and to expanding the facilities for advancement of engineering education. In order to create facilities for postgraduate studies and research, Ahsanullah Engineering College was upgraded to the status of a University under the name of East Pakistan University of Engineering and Technology in the year of 1962. After independence of Bangladesh in 1971, it was renamed as the Bangladesh University of Engineering and Technology.

Starting with two faculties, the university has now been enlarged into five faculties.

1.2.2 The BUET Campus

The BUET campus is the campus is compact with five main multistoried buildings housing sixteen departments. It also has several institutes, centre like Institute of Flood Control and Drainage Research (IFCDR), Energy Centers, Institute of Appropriate Technology and Institute of Information and Communication technology (IICT). Several large workshops like carpentry, foundry, sheet metal and machine shops etc. support and facilitate research and undergraduate project works. Students housing and teacher's residence are at walking distance. There are eight halls of residence for students including one for female students within the campus.

1.2.3 Teaching Staff of the University

The total number of filled up teaching posts is 474 out of which 358 teachers are in active service and 116 teachers are on leave pursuing higher studies abroad (as on 01 July 2001). The following is lists of numbers teachers in active service including those against leave vacancies.
1.3 BUET Automobile shop

BUET has an automobile shop its own. It consists of Vehicles, Manpower (Staff), Machine and equipments.

1.3.1 Background of Automobile shop:

Like the other universities, it has an automobile from where a staff of the university can requisite any vehicle for a period. He/she has to pay a fixed rate for using the vehicles (even for withdrawal of requisition as a charge).

1.3.2 Vehicles

It has a number of Cars, Microbuses and Buses. The cars and microbuses consist AC (Air condition) or non-AC. According to AC or Non Ac the rate will be difference.

1.3.3 Staffs

There are few staffs related to the shop and they are also related to the other faculty too. The number or staffs may change time to time.

<table>
<thead>
<tr>
<th>Serial</th>
<th>Designation</th>
<th>Active</th>
<th>Abroad</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Professor</td>
<td>110</td>
<td>18</td>
<td>128</td>
</tr>
<tr>
<td>02</td>
<td>Assoc. Prof.</td>
<td>60</td>
<td>03</td>
<td>63</td>
</tr>
<tr>
<td>03</td>
<td>Asst. Prof.</td>
<td>72</td>
<td>55</td>
<td>127</td>
</tr>
<tr>
<td>04</td>
<td>Lecturer</td>
<td>116</td>
<td>40</td>
<td>156</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>358</td>
<td>116</td>
<td>474</td>
</tr>
</tbody>
</table>
1.3.4 Maintenance
Like all mechanical machine/equipment, the vehicles must be maintained a regular repairing and supervising. Then the particular vehicle will not be available for requisition.

1.4 Methodology

Methodology is the working process. It is the process where the way or paths are described for the completion of the work. The following were taken for the project.

- **Collect Information for the design:**
  Information is very important for any job. Collection of information for a particular job is essential to develop the whole project. Various important information about the project was collected like collection of present requisition form and other particles.

- **Design the Project Planning with Project Management Software:**
  Project Management (PM) is a task that develops during the project for planning and scheduling the project. The system was designed in PM for planning and scheduling all the working days in a Gantt chart. Thus the time interval for a particular work could be estimated.

- **Analysis the System:**
  Analysis the system means the system should be analyzed thus it should be easy to design or future update.

- **Develop the Use case diagram:**
  The use case diagram was drawn. Rational Rose is the software that was used for the diagrams. Here the relationship among various class and attributes are shown with several diagrams.

- **Develop the Database in Microsoft Access (MS-Access):**
  The database was done in MS-Access in some tables with their relationship. The total data or information are normalized and then make tables.
- **Develop the project:**
  The total projects then develop in Visual basic programming with the connection of database in MS-access.

- **Develop the Web pages:**
  The web pages developed in Microsoft FrontPage. The several web pages are linked together by the programming But first the web pages are drawn manually.

- **Test the Project:**
  Testing is an important part of any project. It reduces the errors, problems and other mistakes in the programmed. The testing of the project was not completely done.

- **Documentation and User's Manual Creation:**
  Documentation is the process of making the whole work in a document (like a file in Microsoft Word). It helps the future system analyst or programmer to change or update the system. The documentation is done in Microsoft Word. The user manual is also created in Microsoft Word.

- **Implement the Project:**
  Finally after checking, testing, reviewing and changing the project must be implemented for the final work.
1.5 Present Requisition System

Present requisition system are followed as below

1.5.1 Present condition

In present system, one who wishes to requisite an automobile he/she has to fulfill the necessary information in a form and sent it to the administration for requisition officer. The administrations for requisition officer then check the form, registered the all information in data sheet and approve for the requisition.

Fig – Present requisition form
1.5.2 Flow chart of information in present system

Fig: Present requisition flow chart

1.5.3 Limitations

No one can see what is selected for him/her as a requisition. Even he/she cannot see the profile or driver or other else. Again sometimes they are in short of the form.

1.6 Necessity of the proposed system

The new system can solve the problems. Here one can requisite any vehicle for a time period and he/she do not need to fill up the form. Again he/she can see the profile of drivers or the vehicles. He/she can also see the requisition status thus he/she can know which one is requested which may help them to choose any other vehicles from them. He/she can also make a complain for the system.
1.7 Objective of the proposed system

Every project must have some goals that should be achieved. This system also has some objectives as follows.

1.7.1 Proposed system

According to the system one should be a registered staff of the university as may be a teacher or an officer or others and must get a login name and password. The password should be unique. The person or the authority may give the name but the authority must give the password.

1.7.1 Limitation

Any from other place or do not have any registered password cannot access the web pages. So staffs of BUET but not registered can not access the pages. So a staff without any recognized password cannot access the web page.
Chapter – 2

Project Inception
Project Inception

Project inception is the project tasks that are done or simple particles that used for beginning the project.

2.1 List of tasks

List of tasks means the tasks are done in the project with the help of hardware, software and manpower. They are as follows.

1. System analysis
2. System design
3. Develop design
   A. Web page
   B. Data base
   C. Programming in Visual basic
4. Project overview
5. Testing and implementation

2.2 Resources

Resource is the components or materials or manpower used for the project.

2.2.1 Hardware:

   Personal Computer with accessories

2.2.2 Software:

   Basically the software are:
   1. Rational Rose
   2. Microsoft Access
2.2.3 Manpower:

Three (3) students of IICT and the Project Supervisor do this Project.

2.3 Project Planning and Scheduling

Project planning and scheduling is the scheduling for the project thus the time can be used properly. The project scheduling is done by the Project management (PM). Here the schedule that made for the work but delayed as BUET was closed for several times.

![Gantt Chart](image_url)

Fig: Project Planning and Scheduling
Chapter - 3

System Analysis
System Analysis

To analyze the system we use UML as modeling language and Rational Rose 2000 as UML editor and CASE Tools. With the help of UML and Rational Rose 2000 we do the following tasks

- Use case Diagram for the System.
- Class Diagram of the different Activity of the system
- Sequence Diagram of the different Activity
- Automatic Code generation
- Documentation

3.1 Use Case Diagram

The Use Case Diagram describes the functionality that the system should deliver as perceived by the External "Actor" and the Requirements of the system. The Use Case is the one UML artifact that focuses on what the system will be contracted to do, not how it will do it. One Use Case can be responsible for satisfying many events. As a result, a Use Case may have more than one pathway through it.

We draw a Use Case Diagram for our System, which is given below

Fig: Use Case Diagram
From the above Figure of Use Case Diagram we can get following Information.
The Use Case “Sign Up or Register” is performed by the two Actors “New User” and
“Requisition Clerk”. The Actor “New user” is an external actor.
The Use Case “Makes Requisition” and “Cancel Requisition” is performed by two Actors
“Registered user” and “Booking clerk”.
Two Actors “Registered User” and “Office Record Keeping Clerk” perform the Use Case
“Complain” and “Get Information”.
Two actors the “Registered User” and the “Billing clerk” perform the Use Case “Bill
Preparation”.

3.2 Class Diagram

The Class Diagram contain the following component

- Classes of the different entities
- Attributes of each class
- Operations of each class

3.2.1 Classes

In the class diagram contain more than one class and each class contain its own attributes
and operations

3.2.2 Attributes

Attributes are usually member variables of the classes. Some times they are may be
public.

3.2.3 Operations

Every class may contain one or more operation. Which are usually either public
subroutines or public functions.
In this System we draw the following class diagram

Fig: Class Diagram for the System
3.2.4 Classes In the System

This System has the following major classes with their Attributes and Operations:

Class : SignUp
Attribute: Name
Address
Designation
Email
Password
Operations: CreateUser(Name,Address,Designation,Email,Password, ConfirmPassword)

Class : User
Attributes: User_ID
Name
Address
Designation
Email
Password
Operations: Get_Authentication(User_ID,Password)

Class : Booking
Attributes: User_ID
Requisition_ID
Booking_ID
Vehicle_ID
Driver_ID
Use_Date
Start_Time
Finish_Time
Operations: AddBooking(Vehicle_ID,Use_Date,Start_Time,Finish_Time)
              CancelBooking(Booking_ID)
              IsAvailable(Vehicle_ID,Use_Date,Start_Time,Finish_Time)

Class : Requisition
Attributes:
- User_ID
- Vehicle_ID
- Requisition_ID
- Use_Date
- Start_Time
- Finish_Time

Operations:
- AddNewReq(User_ID, Vehicle_ID, Use_Date, Start_Time, Finish_Time)

Class: Vehicle
Attributes:
- Vehicle_ID
- Type_ID
- SubCat_ID
- Model_No
- Registration_No
- Chassis_No

Operations:
- IsAvailable(Vehicle_ID, Use_Date, Start_Time, Finish_Time)
- CancelAssignment(Vehicle_ID, Use_Date, Start_Time, Finish_Time)

Class: VehicleSchedule
Attribute:
- Vehicle_ID
- Vehicle_Type
- Use_Date
- Start_Time
- Finish_Time

Operations:
- ShowStatus

Class: Driver
Attribute:
- Driver_ID
- Name
- Address
- Dtype_ID

Operations:
- IsAvailable(Driver_ID, Use_Date, Start_Time, Finish_Time)
- CancelAssignment(Driver_ID, Use_Date, Start_Time, Finish_Time)
3.3 Sequence Diagram

A sequence Diagram depicts the interactions among objects during a certain period of time. Because the pattern of interactions varies from one use case to another, each sequence diagram shows only the interaction pertinent to a specific use case. It shows the participating objects by their lifelines, and the interactions among those objects arranged in time sequence-by the messages they exchange with one another.

A sequence diagram may be presented either in a generic form or in an instance form. The generic form shows all the possible sequences of interactions, that is, the sequences corresponding to all the scenarios of a use case. The instance form, on the other hand, shows the sequence for only one scenario.

The vertical axis of the diagram represents the time and the horizontal axis represents the various participating objects. Time increases as we go down to the vertical axis.

To analyze and design the system we draw some Sequence Diagram which are given below with their details
3.3.1 **Sequence Diagram for Creating Users**

The new users of the System first need to register them on the system to use the system for booking vehicles. The sequence diagram below shows how to interact with the signup object. For signup or register the new users must give the information, which the system want to know.

Fig: Sequence Diagram For Creating User
3.3.2 Sequence Diagram for Booking

To give booking the users first authenticate themselves. Then they supply the requisition information to the requisition object. The information then passes through to the booking object. Booking object collects and checks the information from the vehicle object, vehicle Schedule object, driver object and diver schedule object. If the information is ok then the booking is complete otherwise prompt for new booking. The sequence diagram for booking is given below.

![Sequence Diagram for Booking](image)

Fig: Sequence Diagram for Booking
3.3.3 Sequence Diagram for Cancel Booking

To cancel booking the users must have a valid booking status, he must have an user authentication. Then he needs to supply the Booking information to the Booking object. Then the booking object cancels the assignment of vehicle and the driver. The sequence diagram for Cancel booking is given below.

Fig: Sequence Diagram for Cancel Booking
3.3.4 Sequence Diagram for Bill Check

To check the Bill after using the vehicles, the user must have a valid booking and authentication. The User then passes the User_ID and the booking_ID to the bill object to check the bill. The sequence diagram for Check Bill is given below.

![Sequence Diagram for Check Bill](image)

3.4 Code Generation

After drawing the Use case, Class and Sequence Diagram we generate code and design the different class modules in Visual Basic 6 with the help of Rational Rose 2000 and CASE Tools. Which are uses in the System Design Phase.
3.5 Data Requirement Analysis

In this phase we analyze the data, which are required for the proposed vehicle requisition system. We also normalize the data and create table in the MS Access 2000.

Data fields (attribute) that are required are put into the following normalized table:

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>User_ID</td>
</tr>
<tr>
<td></td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td>Address</td>
</tr>
<tr>
<td></td>
<td>Designation</td>
</tr>
<tr>
<td></td>
<td>Email</td>
</tr>
<tr>
<td></td>
<td>Password</td>
</tr>
<tr>
<td>Requisition</td>
<td>Requisition_ID</td>
</tr>
<tr>
<td></td>
<td>User_ID</td>
</tr>
<tr>
<td></td>
<td>Booking_ID</td>
</tr>
<tr>
<td></td>
<td>RequisitionDate</td>
</tr>
<tr>
<td>Booking</td>
<td>Booking_ID</td>
</tr>
<tr>
<td></td>
<td>Vehicle_ID</td>
</tr>
<tr>
<td></td>
<td>Requisition_ID</td>
</tr>
<tr>
<td></td>
<td>Use_Date</td>
</tr>
<tr>
<td></td>
<td>Start_Time</td>
</tr>
<tr>
<td></td>
<td>Finish_Time</td>
</tr>
<tr>
<td></td>
<td>Use_Type</td>
</tr>
<tr>
<td></td>
<td>Driver_ID</td>
</tr>
<tr>
<td>Bill</td>
<td>Bill_ID</td>
</tr>
<tr>
<td></td>
<td>User_ID</td>
</tr>
<tr>
<td></td>
<td>TotalBill</td>
</tr>
<tr>
<td>Driver</td>
<td>Driver_ID</td>
</tr>
<tr>
<td></td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td>Address</td>
</tr>
<tr>
<td></td>
<td>DTypeID</td>
</tr>
<tr>
<td>Driver_Booking_Info</td>
<td>Driver_ID</td>
</tr>
<tr>
<td></td>
<td>Use_Date</td>
</tr>
<tr>
<td></td>
<td>Start_Time</td>
</tr>
<tr>
<td></td>
<td>Finish_Time</td>
</tr>
<tr>
<td>Table Name</td>
<td>Attribute</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>Driver_Type</td>
<td>DtypeID</td>
</tr>
<tr>
<td></td>
<td>Driver_Type</td>
</tr>
<tr>
<td>Vehicle</td>
<td>Vehicle_ID</td>
</tr>
<tr>
<td></td>
<td>Registration_No</td>
</tr>
<tr>
<td></td>
<td>Model_No</td>
</tr>
<tr>
<td></td>
<td>Chassis_No</td>
</tr>
<tr>
<td></td>
<td>Subcat_ID</td>
</tr>
<tr>
<td></td>
<td>Driver_ID</td>
</tr>
<tr>
<td>Vehicle_Booking_Info</td>
<td>Vehicle_ID</td>
</tr>
<tr>
<td></td>
<td>Use_Date</td>
</tr>
<tr>
<td></td>
<td>Start_Time</td>
</tr>
<tr>
<td></td>
<td>Finish_Time</td>
</tr>
<tr>
<td></td>
<td>Vehicle_Type</td>
</tr>
<tr>
<td>Vehicle_SubCat</td>
<td>Subcat_ID</td>
</tr>
<tr>
<td></td>
<td>SubCategory</td>
</tr>
<tr>
<td>Vehicle_Type</td>
<td>Type_ID</td>
</tr>
<tr>
<td></td>
<td>Vehicle_Type</td>
</tr>
</tbody>
</table>
After creating the table, we also find out the relationship among them, and the following relationship will form between the attributes.

Fig: Database Relationship Diagram
Chapter – 4

System Design
System Design

After analysis the system component with UML we finally design the system. In this phase we perform the following things

- Create the class module of the System
- Final design of Database in MS Access
- Design the User Interface

4.1 Create the class module of the System

We draw the Use Case Diagram, Class Diagram and Sequence Diagram in UML in the system analysis phase. With the help of these diagrams and the Rational rose software we generate the class modules of different classes in Visual Basic 6. The class modules that were developed given below.

Fig: Class Modules Develop in Visual Basic 6
4.2 Final design of Database

In this phase we have done little change in the database and the relationship among the attribute that was developed in the Data requirement analysis phase.

4.3 Web page basics and Design

4.3.1 Internet and web page:

Internet is the information media of today’s world. It is linked together for communication and information.
World Wide Web (WWW) is for give information available on Internet.

4.3.2 Components of Internet and web page:

Some important terms are Web browser, Hyper Text Markup Language (HTML), Hyper Text Transfer Protocol (HTTP), Uniform Resource Locator (URL) etc that is important for the Web page.

4.3.3 Web page design:

1. Get organized for the total work.
2. Collect required information and materials.
3. Design the web pages manually
4. Design the web pages in Computer. The web pages are done in Microsoft FrontPage.
5. Build a prototype and test it.
6. Link all the pages.
7. Connect with the database.
8. Final testing

Implementation.
4.4 Design the user Interface

In preliminary design phase we design some Web Pages in MS Front Page.

4.4.1 User Interface (Who is not yet registered)

- The first-time user must register through a sign up form to use the online service.
- After signing up the user gets a login name & password.
- The user must login with his login name & password to regularly use all the services.

4.4.2 User Interface (Registered user)

The user can have the following basic services

- Ask for Requisition
- Make Requisition
- Get Requisition Information & Status
- Cancel Requisition
- See Vehicle Information
- See Drivers' Profile
- Make Complain
- Check the Bill
The User Interfaces, which were developed in the preliminary design phase, are given below.

![User Interface for LOGIN and Sign Up](image)

**Fig:** User Interface for LOGIN and Sign Up
Fig: User Interface for New User Registration Form
### Cancel your requisition

<table>
<thead>
<tr>
<th>Reference No.</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>Department No.</td>
<td></td>
</tr>
<tr>
<td>Date of Requisition</td>
<td></td>
</tr>
<tr>
<td>Time of Requisition</td>
<td></td>
</tr>
<tr>
<td>Car No.</td>
<td></td>
</tr>
<tr>
<td>User Name</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td></td>
</tr>
</tbody>
</table>

**Fig:** User Interface for Cancel Requisition
<table>
<thead>
<tr>
<th>Information Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
</tr>
<tr>
<td>Status:</td>
</tr>
<tr>
<td>Requisition Status</td>
</tr>
<tr>
<td>Requisition Date:</td>
</tr>
<tr>
<td>Requisition Time:</td>
</tr>
<tr>
<td>Car No:</td>
</tr>
<tr>
<td>Car Type:</td>
</tr>
<tr>
<td>Maximum Speed:</td>
</tr>
<tr>
<td>Cost:</td>
</tr>
<tr>
<td>Rent Charge:</td>
</tr>
<tr>
<td>Driver's Name:</td>
</tr>
<tr>
<td>Driver's Record:</td>
</tr>
<tr>
<td>Others</td>
</tr>
</tbody>
</table>

Fig: User Interface for Information Details
<table>
<thead>
<tr>
<th>Vehicle No</th>
<th>Model</th>
<th>Engine No</th>
<th>Chassis No</th>
<th>Year of Buying</th>
<th>Seat No</th>
<th>Type</th>
<th>Maximum Speed</th>
<th>Cost</th>
<th>Rent Charge</th>
<th>Booking Status</th>
<th>Others</th>
</tr>
</thead>
</table>

**Fig:** User Interface for Car Show Room
Chapter – 5

System Development
System Development

Though our main objective of this project is to analyze and design the system. We also done the system development though it is not fully completed. In this phase we design some User Interface and Code for that in Visual Basic.

5.1 User Interface

Some of the designed User Interfaces are given below.

Fig: Form for LOGIN to the System and Sign Up for the System
Fig: Requisition Form
Please fill the following fields correctly with necessary data.

- Name
- Address
- Designation
- E-mail address
- Password
- Confirm Password

Submit

Fig: New User Registration Form
Fig: Control Form
5.2 Programming Code

Some sample codes are given below. The rest of the codes and the Class Modules contents are given in the Appendix.

Private Sub Label3_Click()
CreateUser.Show
UserName.Hide
End Sub

Private Sub summit_Click()

UserName.Hide
Dim rst As New Recordset
Dim UserID As String
Dim Password As String

rst.CursorType = adOpenDynamic
rst.LockType = adLockOptimistic
rst.ActiveConnection = "PROVIDER=MSDASQL;DSN=PROJECT;USER=sa;"

sqlstr = "select User_ID,Password from User where User_ID = " & txtUserName & ""

rst.Open sqlstr

UserID = rst.Fields("User_ID").Value
Password = rst.Fields("Password").Value

If (Password = txtPassword.Text) And (UserID = txtUserName.Text) Then
    MainForm.Show

41
UserName.Hide

Else
    MsgBox "Wrong User ID Or Password "
    txtPassword.Text = ""
    UserName.Show

End If

rst.Close

Exit Sub

GetErr:
    Call RaiseError(MyUnhandledError, "Error getting user")

End Sub
Chapter 6

Implementation
Implementation

6.1 Programming Language

Programmer can implement this System using an Object Oriented Programming Language like

- Visual Basic 6
- Java
- Visual C++ or any Object Oriented Programming
- Any Web Programming Language with OOP

And

- RDBMS like
  - Oracle
  - MS Access

System can be implemented using Distributed Technology like CORBA (Common Object Request Broker Architecture), DCOM and MTS.
6.2 System Requirements

Computers:
- Processor: Pentium II or above
- Ram: 128 MB
- AGP: 16 MB
- HDD: 20GB
- Modem: Internal/External 56 kbps

Internet Connection
- Dial Up Connection or Broad Brand Connection

User
- User with Valid Account
Chapter – 7

Recommendation and Conclusion
Recommendation

To fulfill all the requirement of the web pages the officers and other staffs must be connected to the Internet.

Again they should be given a valid login name and password to access the system.

Power failure is now becomes a serious problem in country. It could be happened any time. So a back up power system for server should be generated for this purpose.
Conclusion

A standard system has been developed by taking BUET as a model but the System can be implemented any other organization or institutions.

Finally, it will introduce modern booking System for vehicle requisition that influence the present System of BUET. Not only that it will also be the use of latest technology (Internet). It will reduce the time and will bring the transparency of the booking System.

More ever, the will be secured so no one without authorization cannot access the web pages.
Bibliography
1. Reed, Paul R, Jr Developing Application With Visual Basic and UML
2. IICT Information Booklet For Post Graduate Student 1st Edition August, 2001
Appendix
Create User

Private Sub cmdSubmit_Click()
UserName.Hide
Dim rst1 As New Recordset
Dim UserID As String
Dim Password As String
Dim userid1 As String

rst1.CursorType = adOpenDynamic
rst1.LockType = adLockOptimistic
rst1.ActiveConnection = "PROVIDER=MSDASQL;DSN=PROJECT;USER=sa;"

sqlstr = "select * from User"

rst1.Open sqlstr
rst1.MoveLast
UserID = rst1.Fields("User_ID").Value
MsgBox "UserID is : " & UserID
userid1 = Val(Mid(UserID, 3))
userid2 = userid1 + 1
MsgBox "UserID : " & userid2
rst1.AddNew
rst1.Fields("User_ID").Value = "VU" + "0" + CStr(userid2)
rst1.Fields("Name").Value = txtName.Text
rst1.Fields("Address").Value = txtAddress.Text
rst1.Fields("Designation").Value = txtDesignation.Text
rst1.Fields("Email").Value = txtEmail.Text
rst1.Fields("Password").Value = txtPassword.Text
rst1.Update
rst1.Close
MainForm.Show
CreateUser.Hide
End Sub

Control Form

Private Sub cmdBill_Click()
End Sub

Private Sub cmdDriverInfo_Click()
End Sub
Private Sub cmdRequisition_Click()
Requisitionfrm.Show
End Sub

Private Sub cmdVehicleInfo_Click()
End Sub

Requisition Form

Public StrRequisiton As New Requisition
Public objRequisition As New Requisition

Private Sub Form_Load()

txtUserId.Text = UserName.txtUserName.Text
RequDate.Caption = Date
Dim rst As New Recordset
Dim rst1 As New Recordset
Dim UserID As String
Dim Password As String
Dim Requisition_ID As String
rst.CursorType = adOpenDynamic
rst.LockType = adLockOptimistic
rst.ActiveConnection = "PROVIDER=MSDASQL;DSN=PROJECT;USER=sa;"
sqlstr = "select *from Vehicle_Type"
rst.Open sqlstr
rst1.CursorType = adOpenDynamic
rst1.LockType = adLockOptimistic
rst1.ActiveConnection = "PROVIDER=MSDASQL;DSN=PROJECT;Requisition=sa;"

'sqlstr = "select Requisition_ID from Requisition where "

'rst1.Open sqlstr
'Requisition_ID = rst1.Fields("Requisition_ID").Value
'txtRequild.Text = Requisition_ID
End Sub

Private Sub vehicleSubcat_Change()
vehicleSubcat_KeyPress(KeyAscii)
End Sub
Private Sub vehicleSubcat_Click()
Dim rst As New Recordset
Dim rst1 As New Recordset
Dim rst2 As New Recordset
Dim UserlD As String
Dim Password As String
Dim Requisition_ID As String
Dim subcatid As Integer
Dim typeid As Integer
Dim vehicleid As String
rst1.CursorType = adOpenDynamic
rst1.LockType = adLockOptimistic
rst1.ActiveConnection = "PROVIDER = MSDASQL; DSN = PROJECT; Requisition = sa;"
sqlstr = "select Type_id from Vehicle_Type where Vehicle_Type = " & Vehicle.List(Vehicle.ListIndex) & ""
rst1.Open sqlstr
    typeid = rst1.Fields("Type_id").Value
rst.CursorType = adOpenDynamic
rst.LockType = adLockOptimistic
rst.ActiveConnection = "PROVIDER = MSDASQL; DSN = PROJECT; USER = sa;"
sqlstr = "select Subcat_ID from Vehicle_SubCat where SubCatagory = " & vehicleSubcat.List(vehicleSubcat.ListIndex) & ""
rst.Open sqlstr
    subcatid = rst.Fields("Subcat_ID").Value
MsgBox "subcatid: " & subcatid
rst2.CursorType = adOpenDynamic
rst2.LockType = adLockOptimistic
rst2.ActiveConnection = "PROVIDER = MSDASQL; DSN = PROJECT; USER = sa;"
sqlstr = "select *Vehicle ID from Vehicle where Subcat ID = " & subcatid & " and type_id = " & typeid & ""
rst2.Open sqlstr
    vehicleid = rst2.Fields("Vehicle_ID").Value
    txtVehicleId.Text = vehicleid
End Sub

User Name Form

Private Sub Label3_Click()
CreateUser.Show
UserName.Hide
End Sub

Private Sub summit_Click()
'On Error Resume Next ' GoTo GetErr

'## Your code goes here ...
UserName.Hide
Dim rst As New Recordset
Dim UserID As String
Dim Password As String

rst.CursorType = adOpenDynamic
rst.LockType = adLockOptimistic
rst.ActiveConnection = "PROVIDER=MSDASQL;DSN=PROJECT;USER=sa;"

sqlstr = "select User_ID,Password from User where User_ID = " & txtUserName & "]"

rst.Open sqlstr

UserID = rst.Fields("User_ID").Value
Password = rst.Fields("Password").Value

If (Password = txtPassword.Text) And (UserID = txtUserName.Text) Then
    UserName.Hide
    MainForm.Show
    UserName.Hide
    MsgBox "Password OK"
Else
    MsgBox "Wrong User ID Or Password 
    txtPassword.Text = ""
    UserName.Show
End If

rst.Close

Exit Sub

GetErr:
    Call RaiseError(MyUnhandledError, "Error getting user")

End Sub

Authentication class

Private Sub Label3_Click()
CreateUser.Show
UserName.Hide
End Sub

Private Sub summit_Click()

' On Error Resume Next ' GoTo GetErr

'## Your code goes here ...
UserName.Hide
Dim rst As New Recordset
Dim UserID As String
Dim Password As String

rst.CursorType = adOpenDynamic
rst.LockType = adLockOptimistic
rst.ActiveConnection = "PROVIDER=MSDASQL;DSN=PROJECT;USER=sa;"

sqlstr = "select User_ID,Password from User where User_ID = " & txtUserName & ""

rst.Open sqlstr

UserID = rst.Fields("User_ID").Value
Password = rst.Fields("Password").Value

If (Password = txtPassword.Text) And (UserID = txtUserName.Text) Then
    'UserName.Hide
    MainForm.Show
    UserName.Hide
    MessageBox "Password OK"
Else
    MessageBox "Wrong User ID Or Password "
txtPassword.Text = ""
    UserName.Show
End If

rst.Close

Exit Sub

GetErr:
    Call RaiseError(MyUnhandledError, "Error getting user")

End Sub
Bill Class

Option Explicit

'##ModelId=3D59EAE1003E
Private User_ID As String

'##ModelId=3D59EAE80271
Private Booking_ID As String

'##ModelId=3D59EAF8004E
Private Vehicle_ID As String

'##ModelId=3D59EB410138
Private Total_BILL As Integer

'##ModelId=3D59EDBC031C
Private Bill_ID As String

'##ModelId=3D61E00303C8
Private mlClassDebugID As Long

'##ModelId=3D61D9CB0213
Public Sub ShowBill(User_ID As String, Booking_ID As String)
    On Error GoTo ShowBillErr

    '## Your code goes here ...

    Exit Sub

ShowBillErr:
    Call RaiseError(MyUnhandledError, "ShowBill Sub")
End Sub

'##ModelId=3D61E00303A9
Public Property Get ClassDebugID() As Variant
    On Error GoTo ClassDebugIDErr

    ClassDebugID = mlClassDebugID

    Exit Property

ClassDebugIDErr:
    Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property
Bill Preparation Class

Option Explicit

'##ModelId=3D58EBF3008C
Private Booking_ID As String

'##ModelId=3D58EC16029F
Private User_ID As String

'##ModelId=3D58EC1D03C8
Private Vehicle_ID As String

'##ModelId=3D58EC3A002E
Private Duration As Date

'##ModelId=3D58EC45031C
Private Total_BILL As Integer

'##ModelId=3D61E00701A5
Private mlClassDebugID As Long

'##ModelId=3D58ECBF03B9
Public NewProperty As Clerk

'##ModelId=3D61E0070186
Public Property Get ClassDebugID() As Variant
    On Error GoTo ClassDebugIDErr
    ClassDebugID = mlClassDebugID
    Exit Property
ClassDebugIDErr:
    Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property

Booking Class

Option Explicit

'##ModelId=3D59D2B20399
Private User_ID As String

'##ModelId=3D59D3D302FD
Private Booking_ID As String

'##ModelId=3D59D3DD029F
Private Vehicle_ID As String

'##ModelId=3D59D3EA0196
Private Driver_ID As String

'##ModelId=3D59D3F10148
Private Start_Time As Date

'##ModelId=3D59D3F802DE
Private Finish_Time As Date

'##ModelId=3D59D40503C8
Private Use_Date As Date

'##ModelId=3D59D41D0261
Private Requisition_ID As String

'##ModelId=3D61E003004E
Private mlClassDebugID As Long

'##ModelId=3D61B83401C5
Public Sub AddBooking(Vehicle_ID As String, VehicleType As String, vehicleSubcat As String, UseDate As Date, StartTime As Date, FinishTime As Date)
    On Error GoTo AddBookingErr

    '' Your code goes here ...

    Exit Sub

AddBookingErr:
        Call RaiseError(MyUnhandledError, "AddBooking Sub")
End Sub

'##ModelId=3D61B83D01C5
Public Sub CancelBooking(BookingID As String)
    On Error GoTo CancelBookingErr

    '' Your code goes here ...

    Exit Sub

CancelBookingErr:
        Call RaiseError(MyUnhandledError, "CancelBooking Sub")
End Sub
Public Sub IsAvailable(VehicleType As String, vehicleSubcat As String, UseDate As Date, StartTime As Date, FinishTime As Date)
    On Error GoTo IsAvailableErr
    '## Your code goes here ...
    Exit Sub
IsAvailableErr:
    Call RaiseError(MyUnhandledError, "IsAvailable Sub")
End Sub

Public Sub FindBookingByUser(UsedID As String)
    On Error GoTo FindBookingByUserErr
    '## Your code goes here ...
    Exit Sub
FindBookingByUserErr:
    Call RaiseError(MyUnhandledError, "FindBookingByUser Sub")
End Sub

Public Property Get ClassDebugIDO As Variant
    On Error GoTo ClassDebugIDErr
    ClassDebugID = mlClassDebugID
    Exit Property
ClassDebugIDErr:
    Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property

Booking Clerk

Option Explicit

Private Booking_ID As String
Private User_ID As String
Private Vehicle_ID As String

'##ModelId=3D58E712007D
Private Driver_ID As String

'##ModelId=3D58E73F00CB
Private Booking_Date As Date

'##ModelId=3D58E7610148
Private Use_Date As Date

'##ModelId=3D58E7CC02CE
Private Finish_Time As Date

'##ModelId=3D59D1C20203
Private Start_Time As Date

'##ModelId=3D61E00600AB
Private mlClassDebugID As Long

'##ModelId=3D58E89702BF
Public NewProperty As Clerk

'##ModelId=3D61E006008C
Public Property Get ClassDebugID() As Variant
    On Error GoTo ClassDebugIDErr
    ClassDebugID = mlClassDebugID
    Exit Property
ClassDebugIDErr:
    Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property

**Cancel Booking Class**

Option Explicit

'##ModelId=3D59D56603B9
Private Booking_ID As String

'##ModelId=3D59D59300DA
Private Cancel_ID As String

'##ModelId=3D61E00800FA
Private mlClassDebugID As Long
Public Sub Add()
    On Error GoTo AddErr

    "## Your code goes here ..."

    Exit Sub
AddErr:
    Call RaiseError(MyUnhandledError, "Add Sub")
End Sub

Public Sub CancelBooking()
    On Error GoTo CancelBookingErr

    "## Your code goes here ..."

    Exit Sub
CancelBookingErr:
    Call RaiseError(MyUnhandledError, "CancelBooking Sub")
End Sub

Public Property Get ClassDebugID() As Variant
    On Error GoTo ClassDebugIDErr

    ClassDebugID = mlClassDebugID

    Exit Property
ClassDebugIDErr:
    Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property

Clerk Class

Option Explicit

Private Clerk_Name As String

Private Clerk_ID As String

Private Clerk_Address As String
Private E_mail As String

Private mlClassDebugID As Long

Public Property Get ClassDebugID() As Variant
    On Error GoTo ClassDebugIDErr

    ClassDebugID = mlClassDebugID

    Exit Property
End Property

ClassDebugIDErr:
    Call RaiseError(MyUnhandledError, "ClassDebugID Property")

End Sub

Driver Class

Option Explicit

Private Driver_Name As String

Private Driver_ID As String

Private DriverJoin_Date As Date

Private mlClassDebugID As Long

Public Sub IsAvailable(DriverType As String, StartTime As Date, FinishedTime As Date, UsedDate As Date)
    On Error GoTo IsAvailableErr

    ' Your code goes here ...

    Exit Sub
End Sub

IsAvailableErr:
    Call RaiseError(MyUnhandledError, "IsAvailable Sub")

End Sub
Public Sub CancelAssignment()
    On Error GoTo CancelAssignmentErr

    '## Your code goes here ...

    Exit Sub

CancelAssignmentErr:
    Call RaiseError(MyUnhandledError, "CancelAssignment Sub")
End Sub

Public Property Get ClassDebugID() As Variant
    On Error GoTo ClassDebugIDErr

    ClassDebugID = mlClassDebugID

    Exit Property

ClassDebugIDErr:
    Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property

Driver Schedule

Option Explicit

Public Sub ShowSchedule(Driver_ID As String, Start_Time As Date, Finished_Time As Date, Use_Date As Date)
    On Error GoTo ShowScheduleErr

    '## Your code goes here ...

    Exit Sub
ShowScheduleErr:
    Call RaiseError(MyUnhandledError, "ShowSchedule Sub")
End Sub

'##ModelId=3D6318D301D4
Public Property Get ClassDebugID() As Variant
    On Error GoTo ClassDebugIDErr
    ClassDebugID = mlClassDebugID
    Exit Property
ClassDebugIDErr:
    Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property

**Driver Type Class**

Option Explicit

'##ModelId=3D62FCFD032C
Private Driver_ID As String

'##ModelId=3D62FD0B0399
Private Driver_Name As String

'##ModelId=3D62FD1500EA
Private DriverJoin_Date As Date

'##ModelId=3D62FD2D0119
Private Driver_Type As String

'##ModelId=3D6318D6029F
Private mlClassDebugID As Long

'##ModelId=3D62FCF400FA
Public Sub ShowType(Driver_ID As String)
    On Error GoTo ShowTypeErr

    '## Your code goes here ...

    Exit Sub
ShowTypeErr:
    Call RaiseError(MyUnhandledError, "ShowType Sub")
End Sub

'##ModelId=3D6318D60271
Public Property Get ClassDebugID() As Variant
    On Error GoTo ClassDebugIDErr

    ClassDebugID = mlClassDebugID

    Exit Property
ClassDebugIDErr:
    Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property

Office Record Keeping Clerk

Option Explicit

'##ModelId=3D58E999003E
Private Booking_ID As String

'##ModelId=3D58E9DA009C
Private Duration_of_Use As Timer

'##ModelId=3D58EAA5000F
Private Complain_status As String

'##ModelId=3D58EAED0157
Private Complain_ID As String

'##ModelId=3D61E00501F4
Private mlClassDebugID As Long

'##ModelId=3D58EB8A01C5
Public NewProperty As Clerk

'##ModelId=3D61E00501C5
Public Property Get ClassDebugID() As Variant
    On Error GoTo ClassDebugIDErr

    ClassDebugID = mlClassDebugID

    Exit Property
ClassDebugIDErr:
    Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property

16
Registered User Class

Option Explicit

'##ModelId=3D58DFC701B5
Private mlClassDebugID As Long

'##ModelId=3D58D6A80167
Public NewProperty As User

'##ModelId=3D58CB3B01A5
Public Function Make_RequisitionO As String
    On Error GoTo Make_RequisitionErr

    '## Your code goes here ...

    Exit Function

Make_RequisitionErr:
    Call RaiseError(MyUnhandledError, "Make_Requisition Function")
End Function

'##ModelId=3D58CB520157
Public Function Cancle_RequisitionO As String
    On Error GoTo Cancle_RequisitionErr

    '## Your code goes here ...

    Exit Function

Cancle_RequisitionErr:
    Call RaiseError(MyUnhandledError, "Cancle_Requisition Function")
End Function

'##ModelId=3D58CB6802EE
Public Function ComplainO As String
    On Error GoTo ComplainErr

    '## Your code goes here ...

    Exit Function

ComplainErr:
    Call RaiseError(MyUnhandledError, "Complain Function")
End Function

'##ModelId=3D58CB750251
Public Function Get_InformationO As String
    On Error GoTo Get_InformationErr

    '## Your code goes here ...

    Exit Function

Get_InformationErr:
    Call RaiseError(MyUnhandledError, "Get_Information Function")
End Function
## Your code goes here ...

Exit Function
GetInformationErr:
    Call RaiseError(MyUnhandledError, "GetInformation Function")
End Function

##ModelId=3D58D158031C
Public Function ChangePWD() As String
    On Error GoTo ChangePWDErr

    # Your code goes here ...

Exit Function
ChangePWDErr:
    Call RaiseError(MyUnhandledError, "ChangePWD Function")
End Function

##ModelId=3D58D1A80399
Public Function Change_Address() As String
    On Error GoTo Change_AddressErr

    # Your code goes here ...

Exit Function
Change_AddressErr:
    Call RaiseError(MyUnhandledError, "Change_Address Function")
End Function

##ModelId=3D58DFC70196
Public Property Get ClassDebugID() As Variant
    On Error GoTo ClassDebugIDErr

    ClassDebugID = mlClassDebugID

Exit Property
ClassDebugIDErr:
    Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property

Requisition Clerk

Option Explicit

##ModelId=3D58E51102DE
Private User_ID As String
'##ModelId=3D58E60500BB
Private Registration_Date As Date
'##ModelId=3D61E005034B
Private mlClassDebugID As Long
'##ModelId=3D58E66B03A9
Public NewProperty As Clerk
'##ModelId=3D58E8BE008C
Public NewProperty2 As Clerk
'##ModelId=3D61E005032C
Public Property Get ClassDebugIDO As Variant
  On Error GoTo ClassDebugIDErr

  ClassDebugID = mlClassDebugID

  Exit Property
ClassDebugIDErr:
  Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property

**Requisition Class**

Option Explicit

'##ModelId=3D59D12103C8
Private Vehicle_ID As String
'##ModelId=3D59D1350148
Private User_ID As String
'##ModelId=3D59D15F00DA
Private Start_Time As Date
'##ModelId=3D59D19F036B
Private Finish_Time As Date
'##ModelId=3D59D1F2030D
Private Use_Date As Date
'##ModelId=3D59D35E0148
Private Requisition_ID As String
Private mlClassDebugID As Long

Public Function AddNewReq(UserID As String, VehicleType As String, vehicleSubcat As String, DateOfUse As Date, StartTime As Date, FinishTime As Date, ReqDateTime As Date) As Integer
    On Error GoTo AddNewReqErr
    Exit Function
AddNewReqErr:
    Call RaiseError(MyUnhandledError, "AddNewReq Function")
End Function

Public Sub IfUsed()
    On Error GoTo IfUsedErr
    Exit Sub
IfUsedErr:
    Call RaiseError(MyUnhandledError, "IfUsed Sub")
End Sub

Public Property Get ClassDebugID() As Variant
    On Error GoTo ClassDebugIDErr
    ClassDebugID = mlClassDebugID
    Exit Property
ClassDebugIDErr:
    Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property

Sign UP Class

Option Explicit

Private Name As String
'##ModelId=3D59F5C20119
Private Address As String

'##ModelId=3D59F5C80119
Private Designation As String

'##ModelId=3D59F5D40119
Private Email As String

'##ModelId=3D59F5E00232
Private Password As String

'##ModelId=3D61E005007D
Private mlClassDebugID As Long

'##ModelId=3D61E007B0109
Private User_ID As Variant

Public Sub CreateUser(User_ID As String, user_name As String, Designation As String,
Email As String, Password As String)
On Error GoTo CreateUserErr

'## Your code goes here ...

Exit Sub
CreateUserErr:
   Call RaiseError(MyUnhandledError, "CreateUser Sub")
End Sub

'##ModelId=3D61E005005D
Public Property Get ClassDebugID() As Variant
On Error GoTo ClassDebugIDErr

   ClassDebugID = mlClassDebugID

Exit Property
ClassDebugIDErr:
   Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property

User Class

Option Explicit
Private Name As String
Private UserID As String
Private Address As String
Private Designation As String
Private E_mail As String
Private Password As String
Private user_name As String
Private mlClassDebugID As Long
Public NewProperty As Registered_User

Public Function Get_Authentication() As String
    On Error GoTo Get_AuthenticationErr
    '## Your code goes here ...
    Exit Function
Get_AuthenticationErr:
    Call RaiseError(MyUnhandledError, "Get_Authentication Function")
End Function

Public Sub GetUser_information()
    On Error GoTo GetUser_informationErr
    '## Your code goes here ...
    Exit Sub
GetUser_informationErr:
Call RaiseError(MyUnhandledError, "Get_User_information Sub")
End Sub

'##ModelId=3D58DFC7038A
Public Property Get ClassDebugID() As Variant
    On Error GoTo ClassDebugIDErr
    ClassDebugID = mlClassDebugID
    Exit Property
ClassDebugIDErr:
    Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property

Vehicle Class

Option Explicit

'##ModelId=3D58DB5102BF
Private Registration_no As String

'##ModelId=3D58DB7701B5
Private Vehicle_Type As String

'##ModelId=3D58DB9000CB
Private Model_No As String

'##ModelId=3D58DBA000DA
Private Seat_Number As Integer

'##ModelId=3D58DC6D037A
Private AirConditioner As String

'##ModelId=3D58DC8B8001F
Private Mileage As Long

'##ModelId=3D58DD540290
Private Chassis_No As String

'##ModelId=3D58DFC80186
Private mlClassDebugID As Long

'##ModelId=3D58DEAB0290
Public Function Get_Vehicle_Info() As Object
    On Error GoTo Get_Vehicle_InfoErr
'## Your code goes here ...

Exit Function
Get_Vehicle_InfoErr:
   Call RaiseError(MyUnhandledError, "Get_Vehicle_Info Function")
End Function

'##ModelId=3D58DED002FD
Public Sub Set_Vehicle_Info()
   On Error GoTo Set_Vehicle_InfoErr

'## Your code goes here ...

Exit Sub
Set_Vehicle_InfoErr:
   Call RaiseError(MyUnhandledError, "Set_Vehicle_Info Sub")
End Sub

'##ModelId=3D58DFC80157
Public Property Get ClassDebugID() As Variant
   On Error GoTo ClassDebugIDErr

   ClassDebugID = mlClassDebugID

   Exit Property
ClassDebugIDErr:
   Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property

'##ModelId=3D61D4270203
Public Sub CancelAssignment()
   On Error GoTo CancelAssignmentErr

'## Your code goes here ...

Exit Sub
CancelAssignmentErr:
   Call RaiseError(MyUnhandledError, "CancelAssignment Sub")
End Sub

'##ModelId=3D62F8470280
Public Sub Show_Type()
   On Error GoTo Show_TypeErr

'## Your code goes here ...

}24
Exit Sub
Show_TypeErr:
  Call RaiseError(MyUnhandledError, "Show_Type Sub")
End Sub

'## Modelld=3D63132A0119
Public Sub GetVehicleList(VehicleType As String, vehicleSubcat As String, VehicleList As String)
  On Error GoTo GetVehicleListErr

    '## Your code goes here ...

Exit Sub
GetVehicleListErr:
  Call RaiseError(MyUnhandledError, "GetVehicleList Sub")
End Sub

Vehicle Schedule Class

Option Explicit

'## Modelld=3D621FC702AF
Private Vehicle_ID As Variant

'## Modelld=3D621FD500EA
Private Vehicle_Type As Variant

'## Modelld=3D621FE701C5
Private Vehicle_Status As Variant

'## Modelld=3D622012008C
Private Start_Time As Variant

'## Modelld=3D62202B0261
Private Finished_Time As Variant

'## Modelld=3D6318D500FA
Private mlClassDebugID As Long

'## Modelld=3D62203801D4
Public Sub ShowStatus(Vehicle_ID As String, Start_Time As Date, Finished_Time As Date, Use_Date As Date)
  On Error GoTo ShowStatusErr

    '## Modelld=3D63132A0119
    Call RaiseError(MyUnhandledError, "ShowStatus Sub")

Exit Sub
ShowStatusErr:
  Call RaiseError(MyUnhandledError, "ShowStatus Sub")
End Sub
'## Your code goes here ...

Exit Sub
ShowStatusErr:
    Call RaiseError(MyUnhandledError, "ShowStatus Sub")
End Sub

'##ModelId=3D6318D500BB
Public Property Get ClassDebugID() As Variant
    On Error GoTo ClassDebugIDErr

    ClassDebugID = mlClassDebugID

    Exit Property
ClassDebugIDErr:
    Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property

Vehicle Type Class

Option Explicit

'##ModelId=3D62F8B7035B
Private Vehicle_ID As Variant

'##ModelId=3D62F8FC032C
Private Registration_no As Variant

'##ModelId=3D62F90902BF
Private Model_No As Variant

'##ModelId=3D62F913004E
Private Seat_Number As Variant

'##ModelId=3D62F928036B
Private AirConditioner As Variant

'##ModelId=3D62F9370280
Private Mileage As Variant

'##ModelId=3D62F94400EA
Private Status As Variant

'##ModelId=3D6318D401E4
Private mlClassDebugID As Long

'##ModelId=3D62F9F9003E
Public Sub ShowType(Vehicle_ID As String)
    On Error GoTo ShowTypeErr

    '## Your code goes here ...

    Exit Sub
ShowTypeErr:
    Call RaiseError(MyUnhandledError, "ShowType Sub")
End Sub

'##ModelId=3D6318D401A5
Public Property Get ClassDebugID() As Variant
    On Error GoTo ClassDebugIDErr

    ClassDebugID = mlClassDebugID

    Exit Property
ClassDebugIDErr:
    Call RaiseError(MyUnhandledError, "ClassDebugID Property")
End Property