|  |  |
| --- | --- |
|  |  |

**eProject Document**

Vehicle Showroom Management System

|  |  |
| --- | --- |
| **Group 2** | |
| **Members** | Tran Quoc Dat - Student1183197  Vo Nhu Nhat - Student1183211  Phung Nguyen Anh Tu – Student1183163 |
| **Instructor** | Lai Duc Chung |
| **Batch** | C1808J1 |
| **Semester** | III |

**🙠 Date of issue 2020 🙢**

**Table of Content**

1. **Problem definition**
   1. ***Introduction***
   2. ***Existing Scenario***
   3. ***Proposed scenario***
2. **Customer Requirement Specification**
   1. ***Customer Requirement***
   2. ***Hardware – Software Requirements***
3. **Architecture and Design of the System**
4. **Flow Chart**
5. **Data Flow Diagram**
6. **Entity Relationship Diagram**
   1. ***Entity relationship diagram***
   2. ***Entity - Attributes***
7. **Database Design/Structure**
   1. ***Database relationship diagram***
   2. ***Table design***
8. **Some typical interface**
9. **Task Sheet**
10. **Check List of Validation**
11. **Submission Checklist**
12. **Problem Definition**
    1. **Introduction**

* This software is to be prepared for a car show room, to maintain details of the vehicle, quotation printing, invoice printing etc. Our client is a car dealer cars.
* The software will store the information of all the employees, sales data and will also generate the monthly reports with the help of which owner can analyze the sales and take necessary steps in order to enhance the business further.
* The software hereafter would be called as “Vehicle Showroom Management System”.
  1. **Existing Scenario**
* There are many executives are working in the show room. Lot of attrition keeps on happening every now and then. And new people are recruited. For them time is required to understand all vehicles details as well as keep track of all the processes taking place in the showroom. To over ride the gap the owner thought of this software, where all the information is pre loaded, and at any moment, any car information is available at just a click.
  1. **Proposed Scenario**
* Vehicle showroom management system describes the complete process of selling a vehicle to a customer from the dealer’s showroom. Before selling, the vehicle belongs to the fixed asset of the dealer’s showroom. So the main point of this scenario is posting the vehicle from a fixed asset to a current asset. While executing the process, the dealer can manually maintain the vehicle business transaction type, which means, the vehicle can be set as a new or used vehicle to sell to the customer. This scenario shows the process of new vehicle sales.
* Vehicle showroom management system is ideal for dealers or resellers of any size. The features include:

+ Easy stock inventory updates

+ Full user management

+ Photograph uploads

+ Marketing reports

+ Easy installation

+ etc…

1. **Customer Requirement Specification**
   1. **Customer Requirement**
      1. **Users of the system**

* Admin
* Employee
* Customer
  + 1. **System functions**

|  |  |
| --- | --- |
| **User** | **Functions** |
| **Admin** | * Login * Change profile * Employee management * Brand management * Customer management * Image management * Model car management * Car management   - Order management |
| **Employee** | * Login * Change profile * Brand management * Customer management * Image management * Model car management * Car management * Order management |
| **Customer** | * View website * Contact to showroom |

* 1. **Admin functions**

|  |  |
| --- | --- |
| **Login** | |
| Input | Login to system, user must be enter username and password |
| Process | Access information in database |
| Output | Result of login |
| **Change Profile** | |
| Input | Information details of user such as Name, Address, Phone, Email … |
| Process | Maintain user’s information details |
| Output | Result of Change profile |
| **Employee management** | |
| Input | Information details of employee such as ID, Name, Birthday, Email, Address, Phone, … |
| Process | Access information in database to modify, add new employee to database, delete from database. |
| Output | Display or maintain Information detail of Employee. |
| **Brand management** | |
| Input | Information details of Brand such as Name, Image |
| Process | Access information in database to modify, add new brand to database, delete from database. |
| Output | Display or maintain Information detail of Brand. |
| **Customer management** | |
| Input | Information details of Customer such as Name, Address, Phone, Email, … |
| Process | Access information in database to modify, add new customer to database, delete from database. |
| Output | Display or maintain Information detail of Customer. |
| **Image management** | |
| Input | Information details of Car Image |
| Process | Access information in database to modify, add new image to database, delete from database |
| Output | Display image in Car details. |
| **Model car management** | |
| Input | Information details of Model car such as Name, Brand, Price, … |
| Process | Access information in database to modify, add new Model car to database, delete from database. |
| Output | Display or maintain Information detail of Model car. |
| **Car management** | |
| Input | Information details of Car such as Model number, Name, Price, Color, … |
| Process | Access information in database to modify, add new car to database, delete from database. |
| Output | Display or maintain Information detail of car. |
| **Orders management** | |
| Input | Vehicle information to be entered (or sold) such as Model car, Price, Date(entered or sold) |
| Process | Access information in database to modify, add new Order to database, change status, delete from database |
| Output | Display the list Orders. |

* 1. **Employee functions**

|  |  |
| --- | --- |
| **Login** | |
| Input | Login to system, user must be enter username and password |
| Process | Access information in database |
| Output | Result of login |
| **Change Profile** | |
| Input | Information details of user such as Name, Address, Phone, Email … |
| Process | Maintain user’s information details |
| Output | Result of Change profile |
| **Brand management** | |
| Input | Information details of Brand such as Name, Image |
| Process | Access information in database to modify, add new brand to database, delete from database. |
| Output | Display or maintain Information detail of Brand. |
| **Customer management** | |
| Input | Information details of Customer such as Name, Address, Phone, Email, … |
| Process | Access information in database to modify, add new customer to database, delete from database. |
| Output | Display or maintain Information detail of Customer. |
| **Image management** | |
| Input | Information details of Car Image |
| Process | Access information in database to modify, add new image to database, delete from database |
| Output | Display image in Car details. |
| **Model car management** | |
| Input | Information details of Model car such as Name, Brand, Price, … |
| Process | Access information in database to modify, add new Model car to database, delete from database. |
| Output | Display or maintain Information detail of Model car. |
| **Car management** | |
| Input | Information details of Car such as Model number, Name, Price, Color, … |
| Process | Access information in database to modify, add new car to database, delete from database. |
| Output | Display or maintain Information detail of car. |
| **Orders management** | |
| Input | Vehicle information to be entered (or sold) such as Model car, Price, Date(entered or sold) |
| Process | Access information in database to modify, add new Order to database, change status, delete from database |
| Output | Display the list Orders. |

* 1. **Customer functions**

|  |  |
| --- | --- |
| **View all model cars** | |
| Input | Customer click Cars button. |
| Process | Find all model cars in database. |
| Output | Display the result. |
| **Search order** | |
| Input | Fill appropriate information to search |
| Process | Find the model cars in database |
| Output | List of model cars |

* 1. **Hardware – Software requirement**

### Minimum Hardware Requirements

* CPU: Intel Core i3 2.00 GHz.
* RAM: 4GB.
* Hard Disk: 50 GB.
* OS: Window 7.

### Recommended Hardware Requirements

* CPU: Intel Core i5 2.70 GHz or higher.
* RAM: 8 GB or higher.
* Hard Disk: 100 GB or higher.
* OS: Window 10.

### Software Requirements

* Environment: Microsoft .NET Framework 4.0 or later
* Database Management System: Microsoft SQL Server 2014 or later
* IDE: Microsoft Visual Studio 2017 or later
* Web Server: Internet Information Services (IIS 10.0) or later

1. **Architecture and design of the system**

The Application will be made of a Web-based distributed three-tier architecture.

User Interface with HTML

Web - Tier

And ASPX pages

Middle Tier

C# (Code - behind) file containing

business logic

Database - Tier

SQL Server 2014

**Web – based distributed 3 - Tier Architecture of the Project**

1. **Flow chart**
   1. **Login**

Start

Start

Display confirm exit

Input data

Check Data

Display Error

**false**

Logout

**true**

Stop

Successful login

b. System Logout

Stop

1. System Login
   1. **Add / Update Data**

Start

Start

Load data

Input Data

**false**

Display error

Input Data

Display error

Check Data

**true**

Check Data

Save database

**false**

**true**

Save database

Display messenger

Display messenger

Stop

1. Add data

Stop

b. Update data

* 1. **Search data**

Start

Input keyword

Display error

Check data

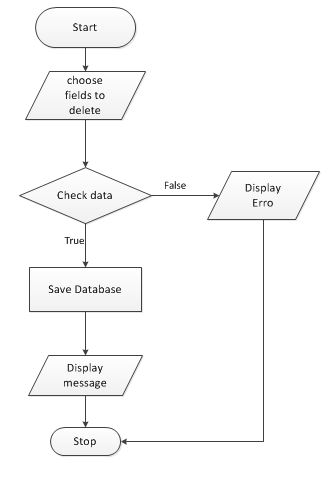
**false**

**true**

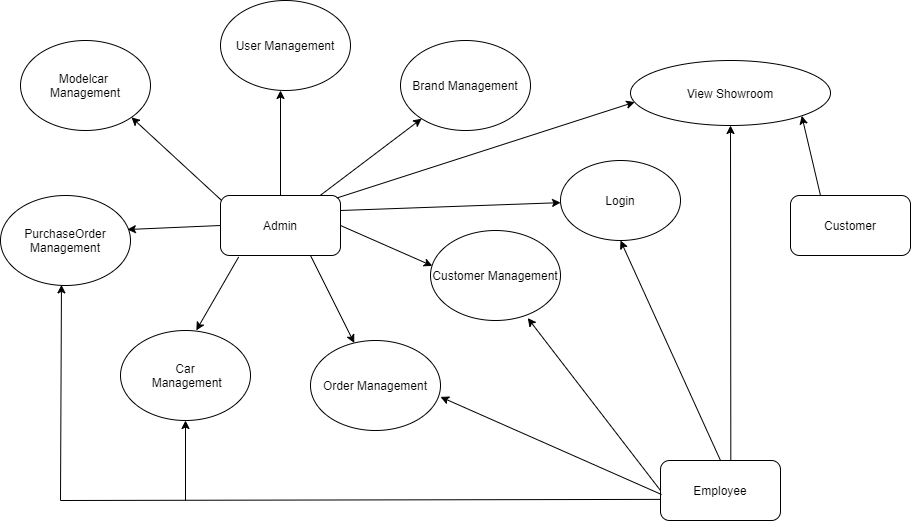
Display result

Stop

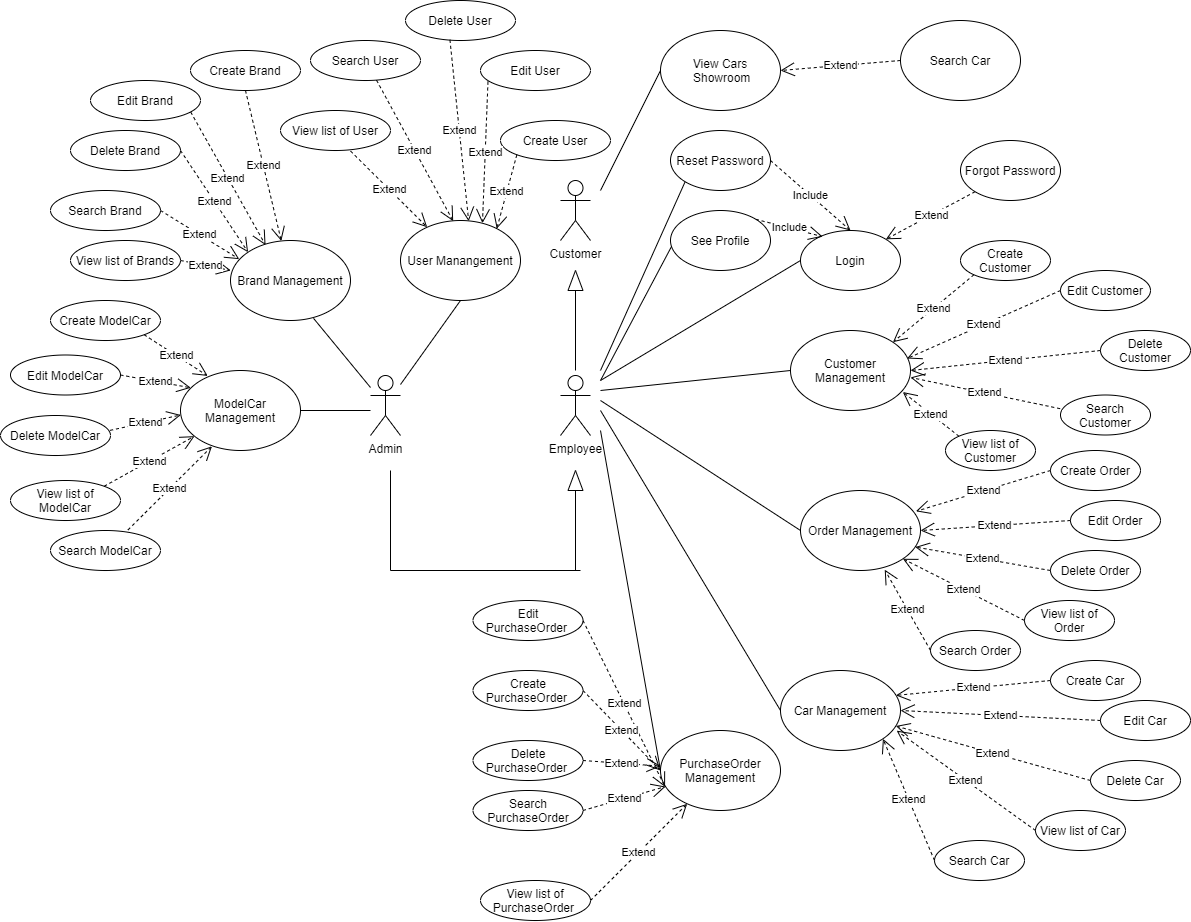
* 1. **Delete Data**

****

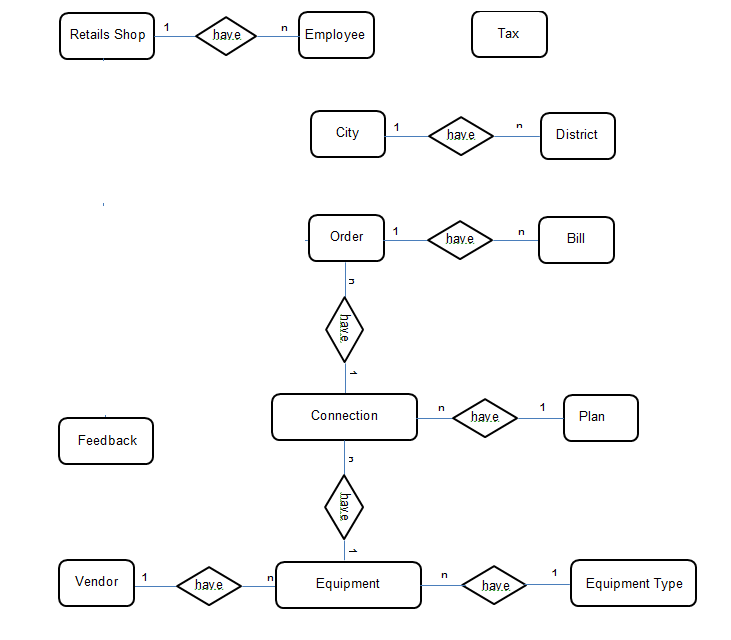
1. **Data flow diagram**
   1. **General Diagram**

****

* 1. **Use case**

****

1. **Entity relationship diagram**
   1. **Entity relationship diagram**

****

* 1. **Entity - Attributes**

**Bill**











**District**







**FeedBack**



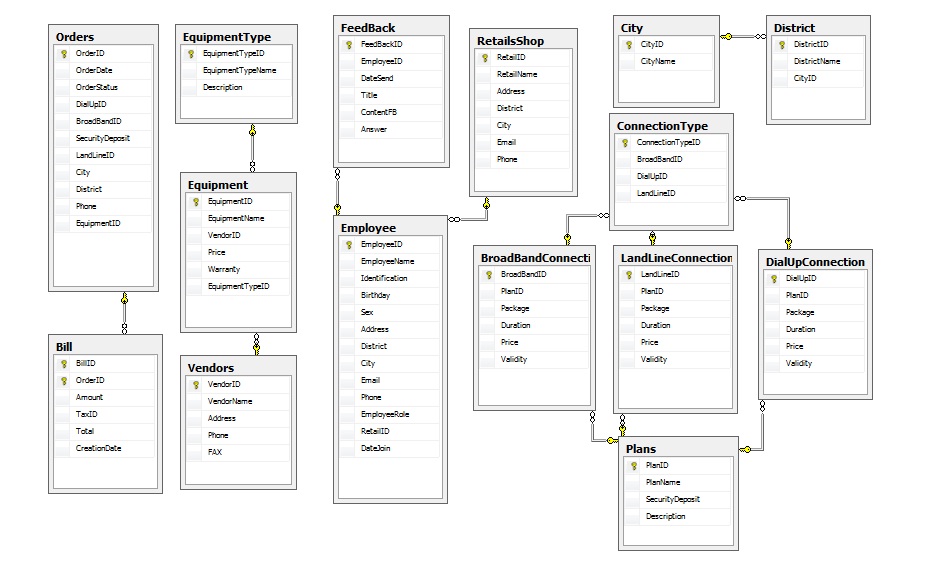
**Orders**







1. **Database Design/Structure**
   1. **Database relationship diagram**

****

* 1. **Table Design**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table:** **UserAccount** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| UserId | int | 🞎 | PK | ID of UserAccount |
| FullName | varchar(256) | 🗹 |  | Name of User |
| UserName | varchar(256) | 🗹 |  | UserName of UserAccount |
| Password | varchar(256) | 🗹 |  | Password of UserAccount |
| Address | text | 🗹 |  | Address of User |
| Email | varchar(100) | 🗹 |  | Email of User |
| PhoneNumber | varchar(50) | 🗹 |  | PhoneNumber of User |
| Birthday | Date | 🞎 |  | Birthday of User |
| CreatedDate | DateTime | 🗹 |  | CreatedDate of UserAccount |
| UpdatedDate | DateTime | 🗹 |  | UpdatedDate of UserAccount |
| Status | int | 🗹 |  | Status of UserAccount |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table:** **Brands** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| BrandId | int | 🞎 | PK | ID of Brands |
| BrandName | varchar(256) | 🗹 |  | Name of Brands |
| image | text | 🗹 |  | Image of Brands |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table:** **ModelCars** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| ModelCarId | int | 🞎 | PK | ID of ModelCars |
| ModelCarName | varchar(256) | 🗹 |  | Name of Plan |
| BrandId | int | 🞎 | FK | ID of Brands |
| PriceOutput | float | 🗹 |  | Price of ModelCars |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table:** **Images** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| ImageId | int | 🞎 | PK | ID of Images |
| ModelCarId | int | 🞎 | FK | ID of ModelCars |
| name | text | 🗹 |  | Name of Images |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: PurchaseOrders** | | | | |
| **Column** | **Data Type** | **Allow Nulls** | **Key** | **Description** |
| PurchaseOrderId | int | 🞎 | PK | ID of PurchaseOrders |
| ModelCarId | int | 🞎 | FK | ID of ModelCars |
| TotalPriceOutput | float | 🗹 |  | Total price of order |
| QuantityInput | int | 🗹 |  | Quantity of car in order |
| PurchaseDate | Date | 🗹 |  | PurchaseDate of Orders |
| CreatedDate | DateTime | 🗹 |  | CreatedDate of Orders |
| UpdatedDate | DateTime | 🗹 |  | UpdatedDate of Orders |
| Status | int | 🗹 |  | Status of Orders |

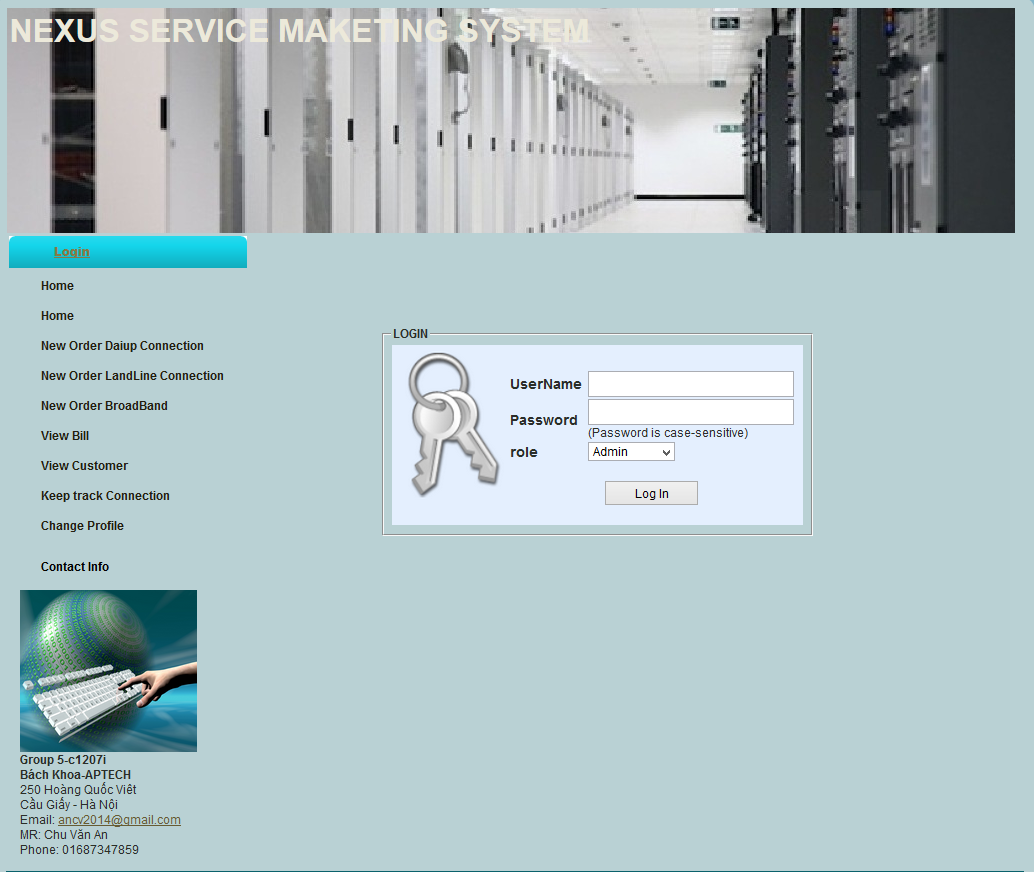
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Cars** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| ModelNumber | varchar(100) | 🞎 | PK | ModelNumber of Cars |
| PurchaseOrderId | int | 🞎 | FK | ID of PurchaseOrders |
| CarName | varchar(256) | 🗹 |  | Name of Cars |
| PriceInput | float | 🗹 |  | PriceInput of Cars |
| SeatQuantity | Int | 🗹 |  | SeatQuantity of Cars |
| Color | varchar(50) | 🗹 |  | Color of Cars |
| Gearbox | varchar(256) | 🗹 |  | Gearbox of Cars |
| Engine | varchar(256) | 🗹 |  | Engine of Cars |
| FuelConsumption | float | 🗹 |  | FuelConsumption of Cars |
| KilometerGone | float | 🗹 |  | KilometerGone of Cars |
| Status | int | 🗹 |  | Status of Cars (new, old) |
| Checking | int | 🗹 |  | Checking of Cars (available, unavailable) |
| CreatedDate | DateTime | 🗹 |  | CreatedDate of car information |
| UpdatedDate | DateTime | 🗹 |  | UpdatedDate of car information |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table:** **Customers** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| CustomerId | Int | 🞎 | PK | ID of Customers |
| UserId | int | 🞎 | FK | ID of UserAccount |
| FullName | varchar(256) | 🗹 |  | Name of Customers |
| Address | text | 🗹 |  | Address of Customers |
| Email | varchar(256) | 🗹 |  | Email of Customers |
| Phone | varchar(256) | 🗹 |  | Phone of Customers |
| Birthday | Date | 🗹 |  | Birthday of Customers |
| CreatedDate | DateTime | 🗹 |  | CreatedDate of customer information |
| UpdatedDate | DateTime | 🗹 |  | UpdatedDate of customer information |

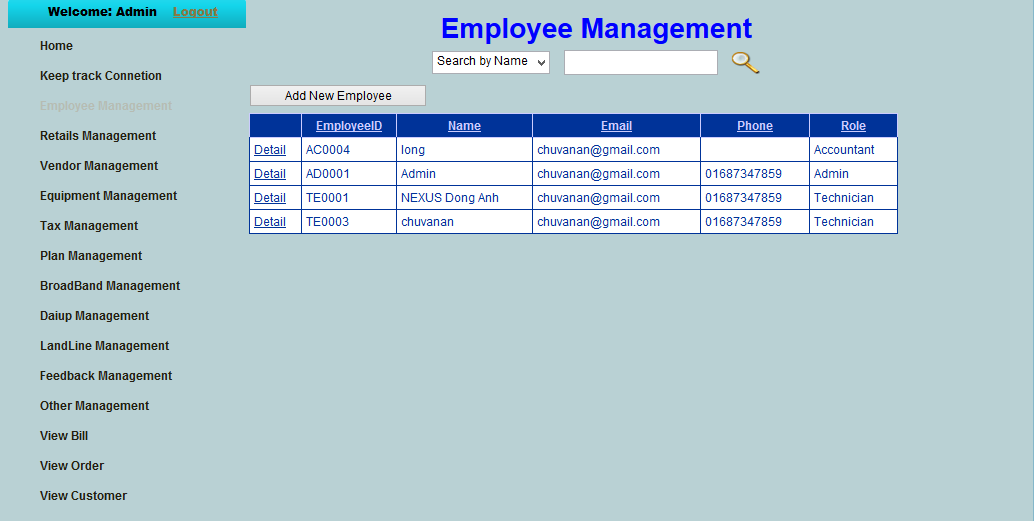
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Orders** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| OrderId | int | 🞎 | PK | ID of Orders |
| ModelNumber | varchar(100) | 🞎 | FK | ModelNumber of Cars |
| CustomerId | int | 🞎 | FK | ID of Customers |
| TotalMoney | float | 🗹 |  | TotalMoney of Orders |
| CreatedDate | DateTime | 🗹 |  | CreatedDate of Orders |
| UpdateDate | DateTime | 🗹 |  | UpdateDate of Orders |
| Status | int | 🗹 |  | Status of Orders |

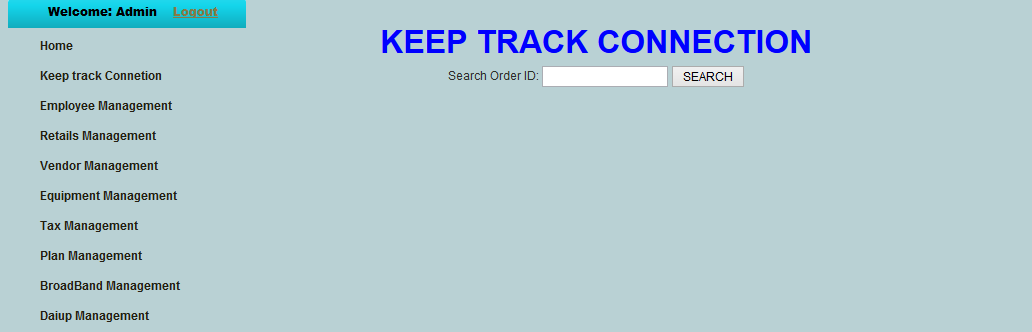
1. **Some typical interface**

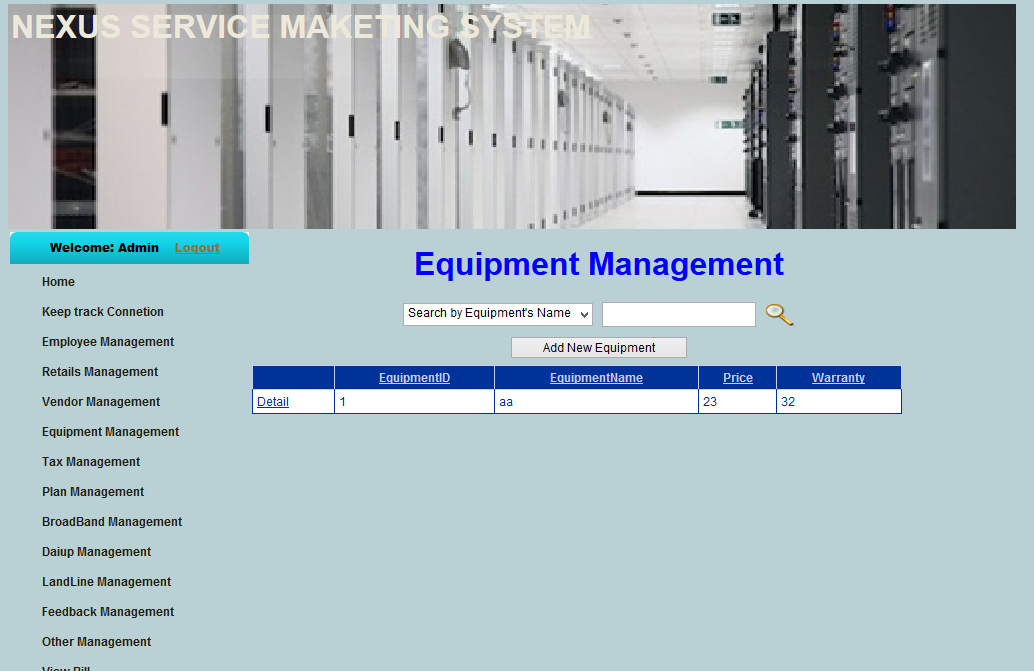
****

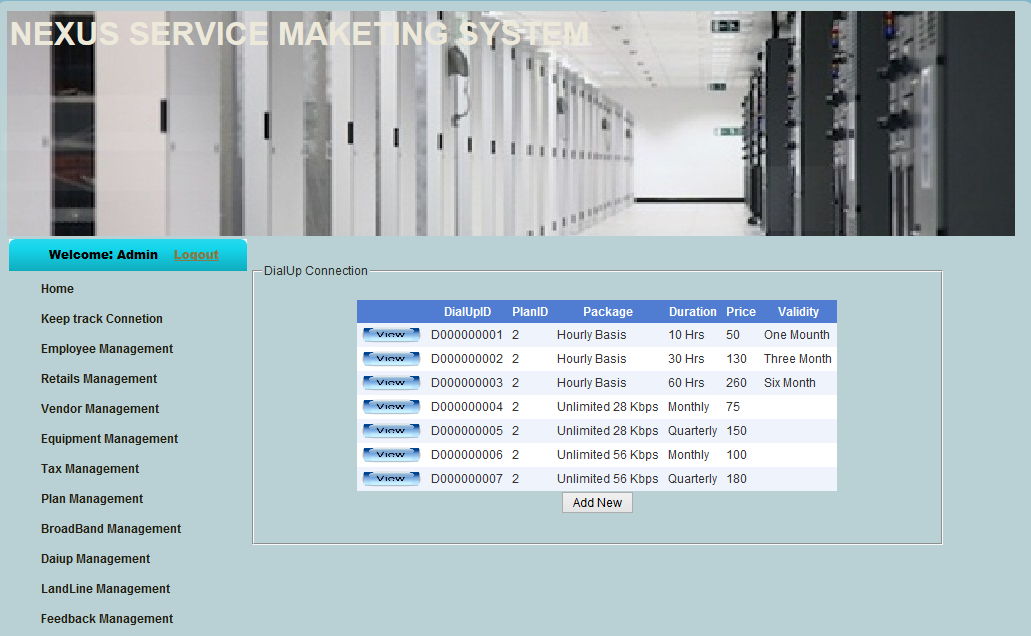
****

****

****

****

****

****

1. **Task Sheet**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Ref. No: 1** | | Project Title: Nexus Service Marketing System | **Date of Preparation of Activity Plan** | | | |
| **Sr. No.** | **Task** | | **Actual Start Date** | **Actual Days** | **Team Member Names** | **Status** |
| 1 | Problem definition | |  | 7 | Tran Quoc Long | complete |
| 2 | User of system | | Tran Quoc Long | complete |
| 3 | System Functions | | Tran Quoc Long | complete |
| 5 | Admin functions | | Chu Van An | complete |
| 6 | Customer functions | | Chu Van An | complete |
| 7 | Retail shop functions | | Chu Van An | complete |
| 8 | Accountant functions | | Nguyen Ba Son | complete |
| 9 | Technician functions | | Tran Quoc Long | complete |
| 10 | Hardware and Software Requirements | | Le Thanh Hop | complete |
| 11 | Login.aspx | |  | 13 | Chu Van An | complete |
| 12 | AddNewEmployee.aspx | | Chu Van An | complete |
| 13 | AddNewEquipment.aspx | | Chu Van An | complete |
| 14 | AddNewRetailShop.aspx | | Chu Van An | complete |
| 15 | AddNewTax.aspx | | Chu Van An | complete |
| 16 | AddNewVendor.aspx | | Chu Van An | complete |
| 17 | BillManagement.aspx | | Le Thanh Hop | Not complete |
| 18 | ChangeProfile.aspx | | Chu Van An | complete |
| 19 | MainPage.aspx | | Nguyen Ba Son | complete |
| 20 | PayBill.aspx | | Le Thanh Hop | Not Complete |
| 21 | Some other…..etc.. | |  |  | Chu Van An | Not complete |

1. **Check list of validation**

|  |  |
| --- | --- |
| Option | Validate |
| Can a new user who get registered, enter the application after logging in ? | Yes |
| Do all the option present in the application display the correct result | Yes |
| Does the application’s functionality resolve the user problem and satisfy their needs ? | Yes |
| Has the hardware and software been correctly chosen ? | Yes |

1. **Submission checklist**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.No | Particular | Yes | No | Comments |
| 1 | Are the users able to login to application after validations is performed on the on the user name and password ? | Yes |  |  |
| 2 | Are the user able to modify their detail after getting registered | Yes |  |  |
| 3 | Are all the screen contents devoid spelling mistakes | Yes |  |  |
| 4 | Is the application user-friendly | Yes |  |  |

**THE END**