**HOSPITAL MANAGEMENT SYSTEM**

**Application Design**

**V1.0**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.no** | **Prepared By** | **ROLE** | **SIGNATURE** | **Date** | **Reviewed by** | **Approved By** |
| **1.** | Y.Prakash | Team Leader | Y.Prakash | 01-04-2015 |  |  |
| **2.** | M.Pradeep Raj | Team member | M.Pradeep Raj | 01-04-2015 |  |  |
| **3.** | M.Gopala Krishna | Team member | M.Gopal | 01-04-2015 |  |  |
| **4.** | L.Bhargavi Krishna | Team member | Bhargavi | 01-04-2015 |  |  |
| **5.** | Asma Sultana | Team member | Asma | 01-04-2015 |  |  |

# Introduction

## Purpose of this document

The detailed design contains the design of every module / sub-system. This covers the following:

* Design of common / generic components
* Module / Sub-system design
* Unit / Class / Program Specifications

From a content consideration, the above three have been combined in this document. However depending on the scale of design or projects, from manageability and configurability point of view, the above three may be split and maintained

## Scope

The scope of the system is explained through its modules as follows

* + Enroll Patient – Hospital Administrators to register a Patient into the system. The administrator should have the details of the patient to be entered into the system.
  + Add Physician - Hospital Administrators to save the Physician details into the system.
  + Physician Search - Hospital Administrators to search for Physician based on specified search criteria
  + Patient Diagnosis Details - by Hospital Administrators to save the patients diagnosis details into the system.
  + View Patient History - Administrator wishes to view the patient case history details

## Intended Audience

The intended audience for this document is

* Developers, testers, other associates in the project
* Client
* Delivery Assurance Group, Process Engineering group

## Definition & Acronyms

|  |  |
| --- | --- |
| **Acronyms** | **Definitions** |
| RL Document | Regulation/Legislation document. |

# Conventions and Standards Followed



# Assumptions, Dependencies and Risks

**Assumptions**:

The HMS Admin must possess transaction data along with their patient’s inform and status for the entire enrolled patient in the hospital.

**Dependencies**:

If there is any similar Hospital management process required for other applications of the same client, this system should be able to accommodate such a reusability dependency.

**Risks**:

If there is any failure in Hospital management process, the system should be able to handle such a financial risk.

# Object Descriptions

### Login Page

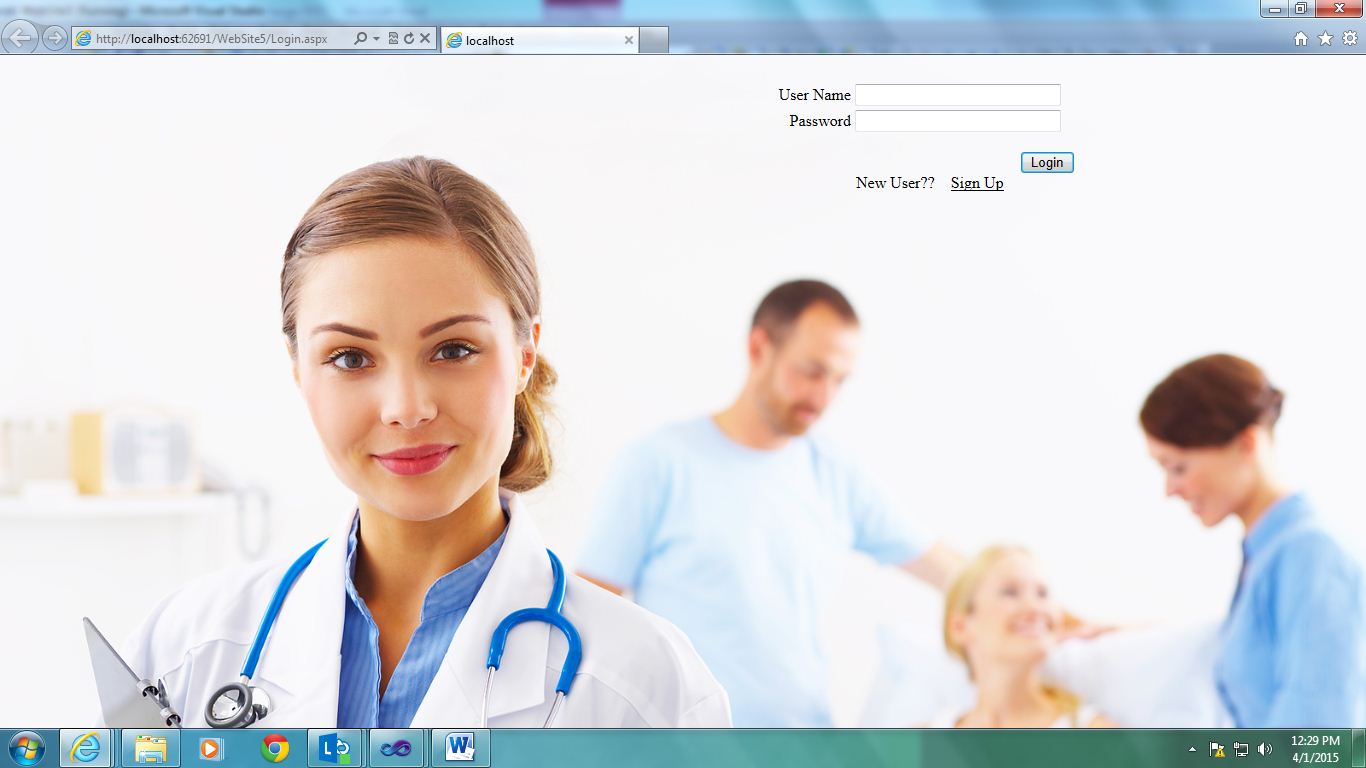
### Functional Requirement

Login Page checks the authentication of the Administrator and redirects to home page on successful login.

### UI Methods Definition

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| Page\_Load | None | Loads the Login page |
| LoginControl\_Authenticate | object sender, AuthenticateEventArgs  e | Allows user to redirect to the home page |

### UI Design



### 4.1.1.5 UI Controls

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **UI Component** | **Type** | **Name** | **Details** |
| 1 | Login Control | LoginControl | LoginControl | Allows the Admin login to the Hospital Management system |
| 2 | UserName | Label | AdminIdLabel |  |
| 3 | Username | Textbox | AdminTextBox | Get the user id of administrator |
| 4 | PASSWORD | Label | PswdLabel |  |
| 5 | PASSWORD | Textbox | PswrdTextBox | Gets the password of admin |
| 6 | LOG IN | Button | LoginButton | To log in |
| 7. | Newuser?? | Label | newuserlabel |  |
| 8. | SignUp | HyperLink | SignUpLink | To register |

### Panel Design

#### 4.1.2.1 Admin Panel



#### 4.1.2.2 UI Controls

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **UI Component** | **Type** | **Name** | **Details** |
| 1 | HOME | Menu Strip | Menu\_Strip\_Home | Reverts to home page |
| 2 | ENROLL PATIENT | Menu Strip | Menu\_Strip\_Enroll\_Patient | Reverts to enroll patient |
| 3 | ADD PHYSICIAN | Menu Strip | Menu\_Strip\_Add\_Physician | Reverts to add physician |
| 4 | PHYSICIAN SEARCH | Menu Strip | Menu\_Strip\_Physician\_Search | Reverts back to physician search |
| 5 | PATIENT DIAGNOSIS DETAILS | Menu Strip | Menu\_Strip\_Patient\_Diagnosis\_Details | Reverts to Patient diagnosis details |
| 6 | PATIENT HISTORY | Menu Strip | Menu\_Strip\_Patient\_History | Reverts to Patient History |
| 7 | LOGOUT | Menu Strip | Menu\_Strip\_Logout | Reverts back to main page |

#### 4.1.2.3 Enroll PatientC:\Users\486124\Desktop\c#\hms\AP1.JPG

#### 4.1.2.4 UI Controls

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **UI Component** | **Type** | **Name** | **Details** |
| 1 | First Name | Label | FirstNameLabel |  |
| 2 | First Name | Textbox | FirstNameTextBox | Get the first name details of patient from admin |
| 3 | First Name Validator | Required Field Validator | FirstNameRequiredFieldValidator | Checks if the first name has been entered. |
| 4 | Last Name | Label | LastNameLabel |  |
| 5 | Last Name | Textbox | LastNameTextBox | Gets the last name of patient from admin |
| 6 | Last Name Validator | RequiredFieldValidator | LastnameRequiredFieldValidator | Checks if last name has been entered. |
| 7 | Department | Label | DepartmentLabel |  |
| 9 | Date of Birth | Label | DobLabel |  |
| 10 | Date of Birth | Textbox | DateOfBirthTextBox | Gets the date of birth of patient |
| 11 | Email | Label | EmailLabel |  |
| 12 | Email | Textbox | EmailTextBox | Get the email of patient from admin |
| 13 | Email Validator | RegularExpressionValidator | EmailRegularExpressionValidator | Validates if email address is entered in correct format. |
| 14 | Contact No | Label | ContactLabel |  |
| 15 | Contact No | Textbox | ContactTextBox | Gets the phone no of employee. |
| 16 | Contact Validator | RegularExpressionValidator | phoneRegularExpressionValidator | Validates if phone no is entered in correct format. |
| 17 | State | Label | StateLabel |  |
| 18 | State | DoropDownList | StateDropDownList | Shows states |
| 19 | Insurance Plan | Label | InsLabel |  |
| 20 | Insurance Plan | DropDownList | InsDropDownList | Shows insurance plans |
| 21 | Submit | Button | SubmitPatButton | Add patient details to the database |
| 22 | Add Insurance Plan Label | Label | AddInsuLabel | Displays to add a new insurance plan into database |
| 23 | Add Insurance Plan | TextBox | InsuTextBox | Takes new insurance plan name |
| 24 | Add | Button | InsAddButton | Add new insurance plan in database |

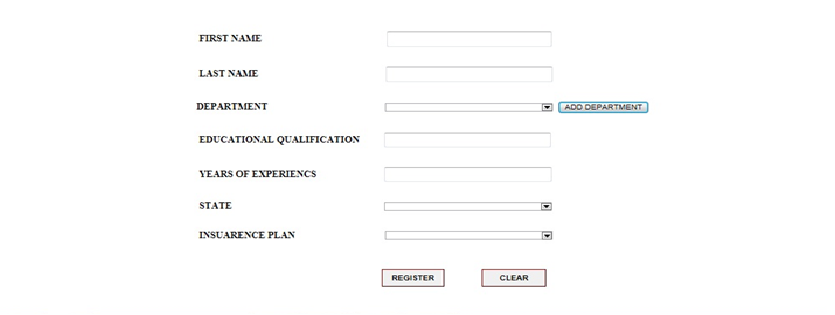
#### 4.1.2.5 UI Events

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| Page\_Load | None | Load the insurance plan and state details from database. |
| SubmitButton\_Click | object sender, EventArgs e | Allows the user to insert the patient details |
| AddInsButton\_Click | object sender, EventArgs e | Allows to add a new insurance plan to database |

#### 4.1.2.6 Ui Method Definition

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| BindPlanDetails | None | Binds the plan details to plan DropdownList. |

#### 4.1.2.7 Add Physician



#### 4.1.2.8 UI Controls

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **UI Component** | **Type** | **Name** | **Details** |
| 1 | First Name | Label | firstNameLabel |  |
| 2 | First Name | Textbox | firstNameTextBox | Get the first name details of physician from admin |
| 3 | First Name Validator | Required Field Validator | firstNameRequiredFieldValidator | Checks if the first name has been entered. |
| 4 | Last Name | Label | lastNameLabel |  |
| 5 | Last Name | Textbox | lastNameTextBox | Gets the last name of physician from admin |
| 6 | Last Name Validator | RequiredFieldValidator | lastNameRequiredFieldValidator | Checks if last name has been entered. |
| 7 | Department | Label | departmentLabel |  |
| 8 | Department | Dropdownlist | departmentDropdownList | Gets the department details of physician from admin |
| 9 | Department validator | RequiredFieldValidator | departmentRequiredFieldValidator | Checks if department has been entered. |
| 10 | Educational  Qualification | Label | educationLabel |  |
| 11 | Educational  Qualification | Textbox | educationTextBox | Gets the date of birth of employee |
| 12 | Educational qualification Validator | RequiredFieldValidator | educationRequiredFieldValidator |  |
| 13 | Year of Experience | Label | experienceLabel |  |
| 14 | Year of Experience | Textbox | experienceTextBox | Get the year of experience |
| 15 | Year of Experience Validator | RequiredFieldValidator | experienceRequiredFieldValidator |  |
| 16 | State | Label | stateLabel |  |
| 17 | State | Dropdown list | stateDropdownList | Gets the state of patient |
| 18 | State Validator | RequiredFieldValidator | stateRequiredFieldValidator |  |
| 19 | Insurance | Label | insuranceLabel |  |
| 20 | Insurance | Dropdown list | insuranceTextBox | Gets the insurance type of physician |
| 21 | Insurance Validator | RequiredFieldValidator | insuranceRequiredFieldValidator |  |
| 22 | Register | Button | addPhysicianButton | Add Physician details to the database |
| 23 | Reset | Button | resetPhysicianButton | Resets all the fields |

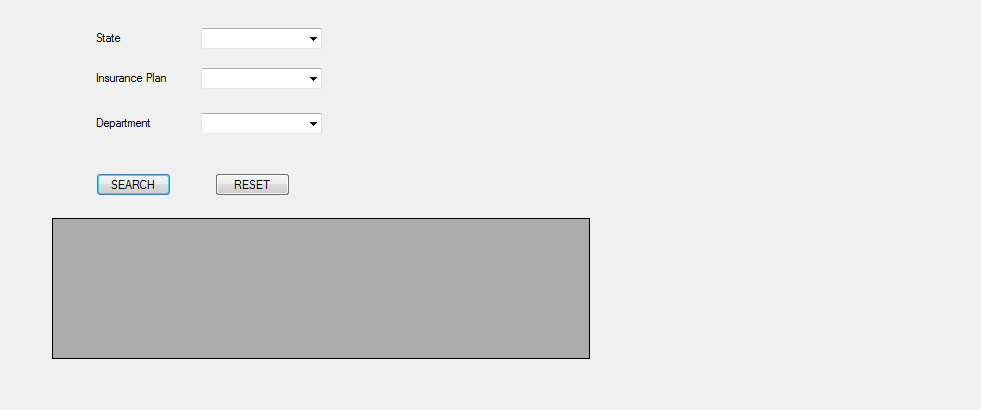
#### 4.1.2.9 UI Events

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| Page\_Load | None | Load the department, insurance plan and state details from database. |
| RegisterButton\_Click | object sender, EventArgs e | Allows the user to insert the physician details |
| AddDepartmentButton\_Click | object sender, EventArgs e | Allows to add a new department. A text box and button appears |
| GoDepartmentButton\_Click | object sender, EventArgs e | Inserts a new department in database. |

#### 4.1.2.10 UI Method Definitions

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| BindDepartmentDetails | None | Binds the department details to the department DropdownList. |
| BindPlanDetails | None | Binds the plan details to plan DropdownList. |

#### 4.1.2.11 Physician search



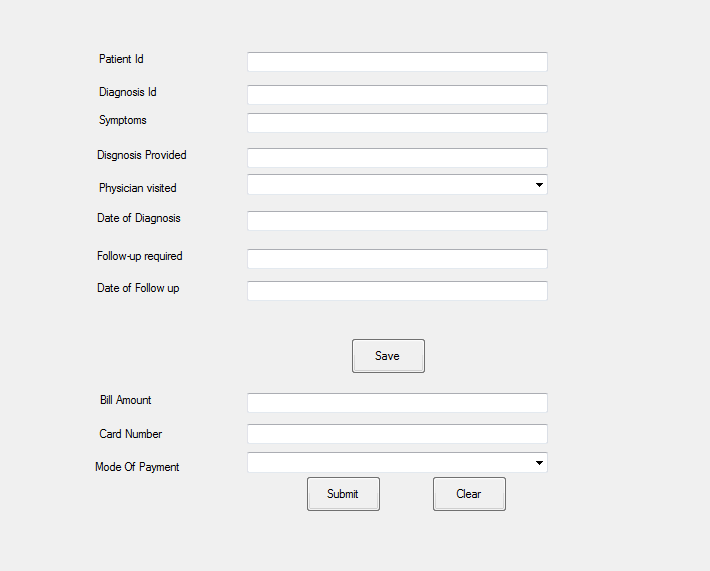
#### 4.1.2.12 UI Controls

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **UI Component** | **Type** | **Name** | **Details** |
| 1 | State | Label | StateLabel |  |
| 2 | State | DropDown List | StateDropDown | Displays State of Physician |
| 3 | Insurance Plan | Label | InsurancePlanLabel |  |
| 4 | Insurance Plan | DropDown List | InsuranceDropDownList | Displays Insurance Plans |
| 5 | Department | Label | DepartmentLabel |  |
| 6 | Department | DropDown List | DepartmentDropDownList | Displays Departments |
| 7 | Search | Button | SearchButton | Details will be Submitted |
| 8 | Reset | Button | ResetButton | Data will Reset |
| 10 | Physicain Details | Data Grid View | PhysicaianDetailsDataGridView | Displays Physicain Details |

#### 4.1.2.13 UI Events

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| Page\_Load | None | Load the department, insurance plan and state details from database. |
| SearchButton\_Click | object sender, EventArgs e | Allows the user to Search Physician Details |
| ResetButton\_Click | object sender, EventArgs e | Allows the user to reset the Entered Values |

#### 4.1.2.14 Patient diagnosis Details



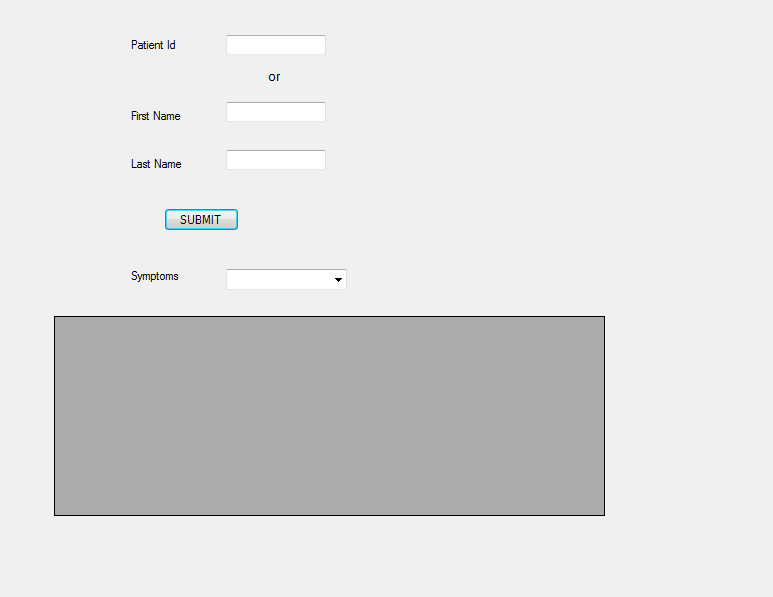
#### 4.1.2.15 UI Controls

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **UI Component** | **Type** | **Name** | **Details** |
|  | Patient Id | Label | PatientIdLabel |  |
|  | Patient Id | Textbox | PatientIdTextbox | Enter Patient Id |
|  | Patient Id Validator | Required Field Validator | PatientIdRequiredFieldValidator | Checks if Patient Id is entered |
|  | Diagnosis Id | Label | DiagnosisIdLabel |  |
|  | Diagnosis Id | Textbox | DiagnosisIdTextBox | Enter Diagnosis Id |
|  | Diagnosis Id Validator | Required Field Validator | DiagnosisIdRequiredFieldValidator | Checks if Diagnosis Id is entered |
|  | Symptoms | Label | SymptomsLabel |  |
|  | Symptoms | Textbox | SymptomsTextbox | Enter Symptoms |
|  | Symptoms Validator | Required Field Validator | SymptomsRequiredFieldValidator | Checks if Diagnosis Id is entered |
|  | Diagnosis Provided | Label | DiagnosisProvidedLabel |  |
|  | Diagnosis Provided | Textbox | DiagnosisProvidedTextbox | Enter Diagnosis Provided |
|  | Diagnosis Provided Validator | Required Field Validator | DiagnosisProvidedRequiredFieldValidator | Checks if Diagnosis Provided is entered |
|  | Physician visited | Label | PhysicianVisitedLabel |  |
|  | Physician Visited | DropDownList | PhysicianVisitedDropDownList | Displays Physician Visited |
|  | Date Of Diagnosis | Label | DateOfDiagnosisLabel |  |
|  | Date Of Diagnosis | Textbox | DateofDiagnosisTextbox | Enter Date of Diagnosis |
|  | Date Of Diagnosis Validator | Required Field Validator | DateofDiagnosisValidator |  |
|  | Follow-Up Required | Label | Follow-upRequiredLabel |  |
|  | Follow-Up Required | Textbox | Follow-upRequiredTextbox | Enter Follow-up required |
|  | Follow-up required validator | Required Field Validator | Follow-upRequiredRequiredFieldValidator | Checks if Follow-up required is entered |
|  | Date of follow up | Label | DateOfFollowupLabel |  |
|  | Date of follow up | Textbox | DateOfFollowupTextbox | Enter Date of Follow up |
|  | Date of follow up validator | Required Field Validator | DateOfFollowupRequiredFieldValidator | Checks if Date of Follow up is entered |
|  | Save | Button | SaveButton | Saves data in the database |
|  | Bill amount | Label | BillAmountLabel |  |
|  | Bill amount | Textbox | BillAmountTextbox | Enter Bill amount |
|  | Bill amount validator | Required Field Validator | BillAmountRequiredFieldValidator | Checks if Bill Amount is entered |
|  | Card number | label | CardNumberLabel |  |
|  | Card number | Textbox | CardNumberTextbox | Enter Card number |
|  | Card number validator | Required Field Validator | CardNumberRequiredFieldValidator | Checks if Card number is entered |
|  | Mode of payment | Label | ModeOfPaymentLabel |  |
|  | Mode of payment | Drop Down List | ModeOfPaymentDropDownList | Displays Mode Of Payment |
|  | Submit | Button | SubmitButton | Saves data in the database |
|  | Clear | Button | ClearButton | Clears entered filled |

#### 4.1.2.16 UI Events

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| Page\_Load | None | Load the Physician visited details from database. |
| SaveButton\_Click | object sender, EventArgs e | Allows the user to insert the Patient Diagnosis details |
| ClearButton\_Click | object sender, EventArgs e | Resets all fields |

#### 4.1.2.17 Patient History



#### 4.1.2.18 UI Controls

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **UI Component** | **Type** | **Name** | **Details** |
| 1 | Patient Id | Label | PatientIdLabel |  |
| 2 | Patient Id | Textbox | PatientIdTextBox | Enter Patient\_Id |
| 4 | First Name | Textbox | FirstNameTextBox | Enter First Name |
| 5 | Last Name | Label | LastNameLabel |  |
| 6 | Last Name | Textbox | LastNameTextBox | Enter Last Name |
| 7 | Submit | Button | SubmitButton | Details will be Submitted |
| 8 | Symptoms | Label | SymptomsLabel |  |
| 9 | Symptoms | DropDownList | SymptomsDropDownList | Displays list of Symptoms |
| 10 | Patient\_History | Data Grid View | PatientHistoryDataGridView | Displays Patient History |

#### 4.1.2.19 UI Events

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| Page\_Load | None | Load the Symptoms details from database. |
| SubmitButton\_Click | object sender, EventArgs e | Allows the user to submit Patient Details |

### BusinessPatientEnroll.cs

### Functional Requirement

This class is used to handle business rules regarding patient for the Hospital management system application. Also it passes the patient details to the database layer to perform database add and read.

### Method Definition

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| AddPatientDetails | Int Pat\_Id  string FirstName, string LastName, string DateOfBirth, string Email,  string PhoneNo,  string State,  String Insurance | Send the patient details to database layer to perform database addition operation. |
| GetInsuranceDetails | None | Get the insurance details from database |
| UpdateInsurancePlans | String InsName | Adds new insurance plan to database |

### BusinessAddPhysician.cs

### Functional Requirement

This class is used to handle business rules regarding physician for the HMS portal application. Also it passes the Physician details to database layer to perform database add, update operation

### Method Definition

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| GetDepDetails | None | Get the department details from database layer |
| AddPhysicianDetails | string PhyID,  string PhyFirstName, string PhyLastName,int DeptNo,  string Phyqualification,  int Phyexperience,int PhyState,int PhyPlan | Send the physician details to database layer to perform database addition operation |
| UpdateDepartmentDetails | string Deptname | Send the new department details to database layer to perform database update operation |
| GetPlanDetails | None | Get the plan details from database layer |

### BusinessPhysicianSearch.cs

### Functional Requirement

This class is used to handle business rules regarding physician search for the Hospital Management system application and it passes the search criteria details to database to retrieve physician details

### Method Definition

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| GetInsuranceDetails | None | Get the insurance details from database layer. |
| GetDepartmentDetails | None | Get the Department details from database layer. |
| SearchDoctor | String State,  String Insurance,  String DocDept | Passes three parameters to search doctor from database |

### BusinessPatientDiagnosisDetails.cs

### Functional Requirement

This class is used to handle business rules regarding patient diagnosis for the Hospital management system application. It is used to add the patient diagnosis details to database.

### Method Definition

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| AddDiagnosisDetails | string Patient\_Id,  string Diagnosis\_Id  string Symptoms  string DiagnosisProvided  string DateOfDiagnosis  string Follow-Up Required  string Date of follow up  float Bill amount  string Card number  string Mode of payment | Sends the diagnosis details to database layer. |
| GetPhysicianVisitedDetails | None | Get the physician details from database layer. |

### BusinessViewPatientHistory.cs

### Functional Requirement

This class is used to handle business rules regarding view patient history for the Hospital management system application. It retrieves the patient history details.

### Method Definition

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| GetSymptoms | None | Get the Symptoms details from database layer. |
| GetPatientHistory | None | Get the patient history details from database |

### DALEnrollPatient.cs

#### 4.1.8.1 Functional requirement

This class makes a call to common class CommonFunction to perform database operation such as add,update database.

#### 4.1.8.2 Method Definition

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| AddPatientDetails | Int Pat\_Id  string FirstName, string LastName, string DateOfBirth, string Email,  string PhoneNo,  string State,  String Insurance | Passes the procedure name and input parameters to Common Function. ExecuteNonQuery  to insert the patient details into database. |
| GetInsuranceDetails | None | Passes the procedure name as input to CommonFunction. ExecuteDataSet  to get the insurance details from database. |
| UpdateInsurancePlans | String InsName | Passes the procedure name and input parameters to CommonFunction. ExecuteNonQuery  to insert a new insurance details into database. |

### DALPhysicianEnroll.cs

#### 4.1.9.1 Functional requirement

This class makes a call to common class CommonFunction to perform database operation such as add,update database.

#### 4.1.9.2 Method Definition

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| GetDepDetails | None | Passes the procedure name as input to CommonFunction. ExecuteDataSet  to get the employee details from database. |
| AddPhysicianDetails | string PhyID,  string PhyFirstName, string PhyLastName,int DeptNo,  string Phyqualification,  int Phyexperience,int PhyState,int PhyPlan | Passes the procedure name and input parameters to CommonFunction. ExecuteNonQuery  to insert the physician details into database. |
| UpdateDepartmentDetails | string Deptname | Passes the procedure name and input parameters to CommonFunction. ExecuteNonQuery  to insert a new department details into database. |
| GetPlanDetails | None | Passes the procedure name as input to CommonFunction. ExecuteDataSet  to get the plan from database. |

### DALPhysicianSearch.cs

#### 4.1.10.1 Functional Requirement

This class makes a call to common class CommonFunction to perform database operation such as retrieve data from database.

#### 4.1.10.2 Method Definition

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| GetInsuranceDetails | None | Passes the procedure name as input to CommonFunction. ExecuteDataSet  to get the insurance details from database. |
| GetDepartmentDetails | None | Passes the procedure name as input to CommonFunction. ExecuteDataSet  to get the Department details from database. |
| SearchDoctor | String State,  String Insurance,  String DocDept | Passes three parameters to search doctor from database |

### DALPatientDiagnosis.cs

#### 4.1.11.1 Functional Requirement

This class makes a call to common class CommonFunction to perform database operation such as add,update database.

#### 4.1.11.2 Method Definition

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| AddDiagnosisDetails | string Patient\_Id,  string Diagnosis\_Id  string Symptoms  string DiagnosisProvided  string DateOfDiagnosis  string Follow-Up Required  string Date of follow up  float Bill amount  string Card number  string Mode of payment | Passes the procedure name and input parameters to Common Function. Execute Non Query  to insert the Patient Diagnosis details into database. |
| GetPhysicianVisitedDetails | None | Passes the procedure name as input to Common Function. Execute DataSet  to get the Physician Visited details from database. |

### DALPatientHistory.cs

#### 4.1.12.1 Functional Requirements

This class makes a call to common class CommonFunction to perform database operation such as retrieves data from database.

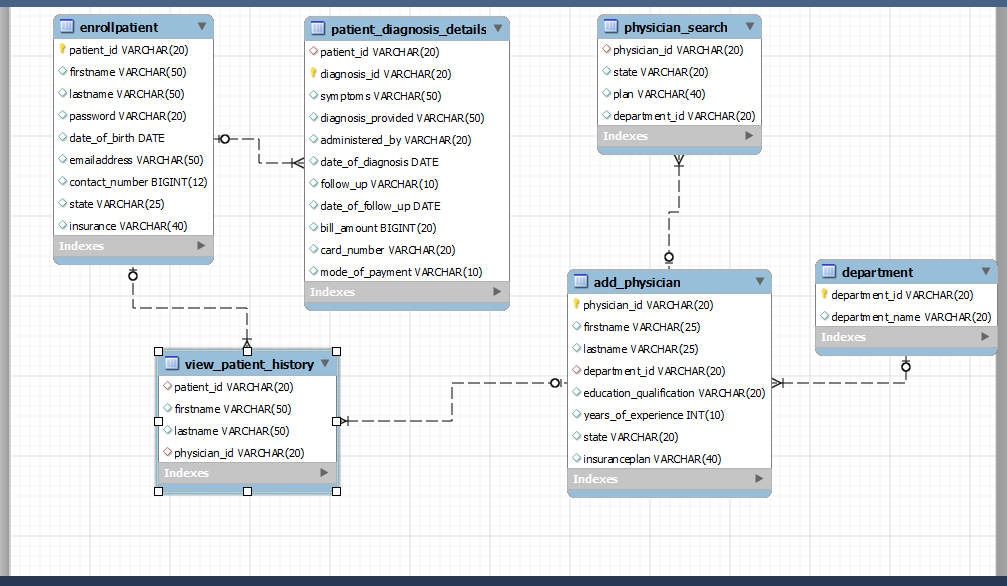
**4.1.12.2 Method Definition**

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Parameter** | **Details** |
| GetSymptoms | None | Passes the procedure name as input to Common Function. ExecuteDataSet  to get the Symptom details from database. |
| GetPatientHistory | None | Passes the procedure name as input to CommonFunction. ExecuteDataSet  to get the patient history details from database. |

# Database Design

## Data Model

## Identification of Tables



## Tables Structure

### Patient\_Enroll

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| Patient Id | Int | 11 | N |
| Patient\_FName | Varchar | 30 | N |
| Patient\_LName | varchar | 30 | N |
| Patient\_Dob | Date |  | N |
| Patient\_Email | Varchr | 50 | N |
| Patient\_Contact | Varchar | 10 | N |
| Patient\_State | varchar | 10 | N |
| Ins\_Id | Int | 11 | N |

### Physician\_Enroll

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| Phy\_Id | varchar | 30 | N |
| Phy\_Fname | varchar | 30 | N |
| Phy\_Lname | varchar | 30 | N |
| Phy\_Dept\_no | int | 11 | N |
| Phy\_Exp | float | 2 | N |
| Phy\_State | varchar | 20 | N |
| Phy\_Ins\_id | varchar | 20 | N |

### Patient\_Diagnosis

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| Patient\_Id | int |  | N |
| Diagnosis\_Id | Int |  | N |
| Symptoms | varchar | 30 | N |
| Diagnosis\_Provided | varchar | 500 | N |
| Physician\_Visited | Varchar | 30 | N |
| Date\_of\_Diag | DATE |  | N |
| Followup\_Req | varchar | 10 | N |
| Follow\_Date | DATE |  | N |
| Bill\_Amt | Float |  | N |
| Card\_No | Varchar | 20 | N |
| Mode\_of\_Pay | Varchar | 10 | N |

### Department

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Nulls** |
| Dept\_No | int | 11 | N |
| Dept\_Name | varchar | 30 | N |

## Objects Details

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Object Type** | **Object Description** |
| AddPatientDetails | Stored Procedure | Add the patient Details to the database |
| GetInsuranceDetails | Stored Procedure | Gets the insurance details from the database |
| UpdateInsurancePlans | Stored Procedure | updates new insurance details to the database using transcations |
| AddPhysicianDetails | Stored Procedure | Add the physician Details to the database |
| GetDepartmentDetails | Stored Procedure | Gets the department details from the database |
| UpdateEmployeeDetails | Stored Procedure | Add new department details to the database using transcations |
| GetPlanDetails | Stored Procedure | Gets the Insurance details from the database |
| AddPhysicianDetails | Stored Procedure | Add the physician Details to the database |
| GetDepartmentDetails | Stored Procedure | Gets the department details from the database |
| UpdateEmployeeDetails | Stored Procedure | Add new department details to the database using transcations |
| GetPlanDetails | Stored Procedure | Gets the Insurance details from the database |
| GetDepartmentDetails | Stored Procedure | Gets the department details from the database |
| GetPlanDetails | Stored Procedure | Gets the Insurance details from the database |
| GetPhysicianVisitedDetails | Stored Procedure | Call fn\_PhysicianVisitedDetails |
| AddDiagnosisDetails | Stored Procedure | Add the diagnosis Details to the database |
| GetSymptoms | Stored Procedure | Call fn\_SymptomsDetails |

## Scripts

# Change Log

*Please note that this table needs to be maintained even if a Configuration Management tool is used.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version Number | Changes made | | | | |
| V1.0 | *Document Creation* | | | | |
| V1.1 |  | | | | |
| Page no | Changed by | Effective date | | Changes effected |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
| V1.2 | *<If the change details are not explicitly documented in the table below, reference should be provided here>* | | | | |
| Page no | Changed by | Effective date | Changes effected | |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |