CERTIFICATE

It is certified that the contents and form of thesis entitled **“SaaS (Software as a Service) Based Practice Management System”** submitted by **Ajab Ali Abro**(2008-NUST-BIT-160),**Mansoor Ahmad Jamali** (2008-NUST-BIT-143) and **Muhammad Afzal** (2008-NUST-BIT-139) have been found satisfactory for the requirement of the degree.

Advisor: Dr. Ali Mustafa Qamar Signature: ____________________

Co-Advisor: SaadArshed Signature:______________________

DEDICATION

*My dearest dedications to my parents and family specially my eldest brother "**Rahmat Ali Abro**" who supported me in every aspect of life, my teachers specially "**Sir Basant Kumar**" and friends who always encourage me.*

Ajab Ali Abro

My dearest dedications to my parents, specially my father "Ghulam Hyder Jamali" who supported in every aspect of education, my teacher "Sir Mazhar Baloch" and specially our co-advisor, "Sir Saad Arshed".

Mansoor Ahmed Jamali

*My dearest dedications to my parents, specially my father "**Liaquat Ali**" who supported in every aspect of education, my teachers specially "**Sir Abdul Jabbar**" and friends who always motivates me.*

Muhammad Afzal

Abstract

Hospitals continue to face issue with maintaining and keeping patient records while managing daily operations of the hospital. On-premise Hospital management solutions excessive Total Cost of Ownership (TCO) requires investments beyond the scope of many hospitals.

Many hospitals end up using feature poor, non-compliant, difficult to use solutions, difficult to allocate resources (human as well as material resources) and sacrifices quality in the way they support, service and interact with their customer. Only on-demand hosted solution in the market that actually allows an agent to “plug and play,” setting up a hospital management connection in minutes with mature agent interfaces, administration which requires little to no end user training and facilitate administration to make decisions and use resources efficiently.

To handle these issues we plan to develop and host a practice management system for hospitals. The purpose of the solution will be to make advanced practice management functionality available, affordable, useable and easily deployable for hospitals. End Users to realize immediate resource use benefits with no equipment, no upfront deployment costs.

Our SaaS Based PMS will be an affordable, Feature-Rich Practice Management System that is fast, simple to deploy, easy to learn, scalable and geographically independent. The solution is made more attractive with the following features:

- Low Entry Costs
- Ease of Configuration and Management
- User friendly and robust user interface

The project covers the following scope:

- **EMR Module**
  - Electronic Medical Record will enable doctors to keep a complete record of a patient (history, reports) instead of maintaining paper based manual files.
- **Scheduling Module**
  - Scheduling of appointments with the doctors with respect to rooms, wards and resources availability.
- **Billing Module**
  - Billing of patient against each diagnosis and treatment and financial transactions.

ACKNOWLEDGEMENTS

We are deeply thankful to our Advisor and Co-Advisor, Dr. Ali Mustafa Qamar, and Mr. Saad Arshed for helping us throughout the course in accomplishing our final year project. Their guidance, support and motivation enabled us to achieve our objectives of the project.

Tables of Contents

Chapter 01

Table of Figures

Abbreviations

SaaS: Software as a Service

PMS: Practice Management System

EMR: Electronic Medical Record

HIPAA: Health Insurance Portability and Accountability Act

HL7: Health Level Seven International

TCO: Total Cost of Ownership

HTTP: Hyper Text Transfer Protocol

UI: User Interface

MS Azure: Microsoft Azure cloud

ERD: Entity Relationship Diagram

CSS: Cascading Style Sheets

SQL: Scripting Query Language

RE: Receptionist

INTRODUCTION *Chapter 01*

In this chapter we will discuss briefly about SaaS (Software as a Service) Based Practice Management System and its importance. We will define our purpose, scope and problem statement of the system with solution and challenges.

1.1 Purpose:

The main purpose of SaaS (cloud) based Practice Management System software is to provide services to hospitals on-demand to maintain the records of their patients (patients' history), dealing with scheduling (information of availability of their doctors).

1.2 Scope:

The software can provide services on-demand to any hospitals, clinics or dispensaries for maintaining their patient records, doctor schedules, hospital resources management, expenses and billing of encountered patients.

1.3 Problem:

Hospitals continue to face issue with maintaining and keeping patient records while managing daily operations of the hospital. On-premise Hospital management solutions excessive total cost of ownership (TCO) requires investments beyond the scope of many hospitals.

Many hospitals end up using feature poor, non-compliant, difficult to use solutions, difficult to allocate resources (human as well as material resources) and sacrifices quality in the way they support, service and interact with their customer. Only on-demand hosted solution in the market that actually allows an agent to "plug and play," setting up a hospital management connection in minutes with mature agent interfaces, administration which require little to no end user training and facilitate administration to make decisions and use resources efficiently.

1.4 Solution:

To handle these issues we plan to develop SaaS (cloud) based Practice Management System for hospitals. The purpose of this solution will be to provide services on-demand to hospitals and make advanced hospital management functionality available, affordable, useable and easily deployable/configurable for hospitals. End Users to realize immediate resource use benefits with no equipment, no upfront deployment costs.

1.5 Challenges:

- Develop and design EMR (Electronic Medical Record), which fulfills maximum hospital standards.
- Low Entry Costs
- Ease of Configuration and Management
- User friendly and robust user interface
- Scheduling of doctors
- Cloud Integration
- Services available on-demand

1.6 Benefits:

This product provides Electronic Medical Record (EMR) which fulfills maximum hospital standards like HL7 and HIPPA. In which low entry cost encountered, ease of configuration and management, user friendly and robust user interface (UI). In this there is capability to deal with resources of hospitals; resources may human as well as material. For that this system provides scheduler feature to handle these issues in which system deals with doctor’s scheduling with respect to rooms, wards, and emergency, it also has flavor to set it for whole year, week, month or selected days and time. Since there are millions of requests are expected so it has designed and deployed in deal with scalability and security to make data private. It also deals with financial management of the hospitals in which it deals with cash flow of the hospitals, invoices and payments against each invoice. This system has capacity to deals hundreds of hospital simultaneously and capability to provide services on demand when required by customer.

LITERATURE REVIEW *Chapter 02*

In this chapter we will discuss briefly about different existing Practice Management System with respect to their functionality, pros and cons.

2.1 Related Work and Literature Survey:

2.1.1 Hospital Management System (Fresh logics Solution):

The hospital management software is excellent software in the hospital management field. It manages all section of the hospital like reception, lab, indoor patient management, outdoor patient management, laboratory management, inventory and account etc. [2]

Key Features:

- Complete and automatic operating of reception.
- Laboratory Management.
- Indoor Management.
- OT Management.
- Complete history of patients.
- Up to-date account management with all accounting needs.
- Maintaining the inventory section of your hospital.
- Build in backup and restore facilities.
- LAN compatible
- Compatibles with any windows-9X or higher.
- The software is both menu and screen driven.
- Instant voice help available on all forms/options.

Pros:

- It has reception
- Laboratory management
- Indoor/outdoor patients
- Inventory and account management

- Open-Source

Cons:

- Limited access for hospital
- Low patients record entries
- Limited pharmacy storage
- Stand-alone system

2.1.2 Hospital Management Information System (NetSol Solutions):

The global healthcare industry is growing at a fast rate and is one of the areas that have the most urgent need of automation. NetSol Technologies understands this need and has developed a strategic collaboration with ShaukatKhanum Memorial Cancer Hospital as part of a long term commitment for IT development in the global health sector.

NetSol's Hospital Management Information System (HMIS) solution provides seamless integration with machines used in the hospital diagnostic practices. One of the widely used integrated machines is Picture Archiving and Communication System (PACS) used to store electronic images generated by machines during diagnoses and storing data against a particular patient's record for use and for reference in the future.

HMIS covers the day to day back office to front office operations of the hospital ranging from patients to the stakeholders of the hospital ensuring proper management of hospital administration ensuring a best possible Return on Investment (ROI). [4]

Specification:

Following is the list of modules present in the subject management information system known as Hospital Management Information system

Clinical

- Patient Registration
- Outpatient Management System
- Intensive Care Unit ITC/CCU Management System
- Echo Cardiology Management

- Cath Laboratory Management System
- Emergency/Casualty Management System
- Specialists & Consultant Clinics Management
- Diagnostic & Laboratories Management
- Pharmacy Management System
- Inpatient & Nursing Management
- Surgery / Operation Theater Management
- Centralized Sterilization Management

Administrative

- Appointment & Scheduling Management
- Medical Records Management
- Front Office Inquiry
- Human Resource Management System
- General Inventory Management

System Security & Administration

- Medical Stores Management
- Hospital Structure, Building Structure and Hospital Controlling Management
- Food Management System
- Asset Management System
- Maintenance Management
- Medical Equipment Management
- Transport Management System
- Purchase Management System

Financials

- General Ledger
- Operational Budgeting System

- Payroll Management System
- Provident Fund Accounting System
- Billing System
- Accounts Receivable System
- Financial Support & Assessment
- Donation Management System
- Procurement / Materials Management System
- Accounts Payable System

Pros:

- It has maximum features
- clinical management
- Administrative management
- System Security & Administration management
- Financials Management

Cons:

- Stand-alone system
- For specific organization
- Cannot assure availability, scalability and data storage

2.1.3 Hospital Management Information System (HMIS) (An electronic management)

An electronic management in a Hospital or a Nursing Home would require to very precise and must result into cost cutting and efficient management. The HMIS has developed this revolutionary product" Electra" is very accurate in its approach and suits all environments including large, medium or small sites. The crucial points that "Electra" emphasis on are listed in the following paragraphs which in turn justify your purchase. [5]

- You will require less number of Staff to cater more patients in same time or even less. You would have the choice to re-deploy them at other suitable locations.

- **Hospital Management System** not only provides an opportunity to the hospital to enhance their patient care but also can increase the profitability of the organization.
- Electra would enable hospitals or Nursing Homes to serve the rapidly growing number of health care consumers in a cost-effective manner.
- Electra can also save extra money on your current computer hardware shopping. Check up with our executive to more on this.
- Hospital administrators would be able to significantly improve the operational control and thus streamline operations.
- This would enable to improve the response time to the demands of patient care because it automates the process of collecting, collating and retrieving patient information.
- The senior Doctors would spend his precious time more in clinical activities than to put in clerical activities otherwise.
- This software interface would also save them a lot of time for special jobs only.
- Accounting sometimes becomes awfully pathetic and complex. This product will eliminate any such complexity, since the retrieval of information through its MIS will become virtually on the tip of your fingers.
- Very important for some, the reduced cost of the manpower would pay for the cost of this product within a short time after its implementation.

Pros:

- It has patients record management
- Laboratory management
- Senior doctors involvement in the system

Cons:

- Limited access for hospital
- Low patients record entries
- Stand-alone system

2.1.4 Hospital Management System (HMS) (An electronic Management):

ELECTRA is a software product suite designed to improve the quality and management of clinical care and hospital health care management in the areas of clinical process analysis and activity-based costing. ELECTRA enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital. ELECTRA helps you manage your processes. ELECTRA provides all process management tool elements: modeling, analysis, and simulation. Documentation though an important part of a Hospital, is a non-productive exercise for the intellectual human being, whose ability lies in core areas of excellence. Hence a systematic approach to the way documents are managed, can transform your Hospital resources to its highest utility and advantage.

Modules:

- Patient Registration
- Appointment Scheduling
- Admission Discharge Transfer
- Bed Management
- Wards Management Module
- Patient Relations
- Doctors Workbench
- Nursing Workbench
- Operation Theater
- Electronic Medical Record
- Clinic Specialties
- Laboratory Information System
- Radiology Information System
- Pharmacy
- Central Sterilized Supply Department
- Blood Bank
- Dietary (F&B)
- Housekeeping/Laundry

- Equipment Maintenance System (EMS)
- Healthcare Packages
- Patient Billing
- Insurance and Contracts Management
- Management Information System (MIS)
- Hospital Administration
- Roster Management
- HRMS
- Financial Accounting

Pros:

- It has patients record management
- Patient used resources management
- Laboratory management
- Financial management
- Administration Management
- Blood bank management
- Doctors management

Cons:

- Stand-alone system
- For specific organization
- Cannot assure availability, scalability and data storage

2.1.5 MediNous Hospital Management System:

MediNous HMS is powerful, flexible, easy to use and is designed and developed to deliver real conceivable benefits to hospitals and clinics. This HMS is designed for multispecialty hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow. [3]

Pros:

- It has patients record management
- Patient used resources management
- Laboratory management
- Financial management
- Administration Management
- Doctors management
- HL7 standards

Cons:

- Stand-alone system
- For specific organization on-demand software application
- Cannot assure availability, scalability and data storage from venders.

2.1.6 List of open source healthcare software (Wikipedia)

- Public Health and Bio surveillance
- Dental Management and Patient Record
- Electronic health or medical record
- Medical Practice Management Software
- Health System Management
- Imaging/Visualization
- Medical Information Systems
- Mobile / Handheld Devices

Pros:

- These have sufficient features
- Low Laboratory management

- Patients EMR
- Open-Source

Cons:

- Limited access for hospital
- Low patients record entries
- Limited pharmacy storage
- Stand-alone system
- Low standards
- Low availability, scalability and low data storage

2.1.7 eMDfix EMR (Electronic Medical Record):

eMDfix is a generic EMR application in which we have different modules to automate the medical record keeping system. We have registration module which deals with all the functions related to patient registration, EMR module for medical records, Scheduler for provider and patient appointment management etc. [6]

- **Registration**
  - **Login Management**
    - Management of different types of accounts like nurse and receptionist account are different and similarly admin provider and provider accounts are managed independently. So you can manage different types of accounts.
  - **Patient Record Management**
    - You can register a patient by providing the following information.
      - Personal Information
      - Contact Information
      - Insurance Information
      - Patient Picture
    - We have provided features for viewing and updating a patient record.
  - **Duplication Record Check**

        - We distinguish one patient from other on the basis of his SSN so for duplication check application will ask you the patient's SSN if that patient is already in your records, then application will inform you that this patient is already in your records.
    - **Zip, City, State Management**
        - After entering the patient's personal record you just have to enter patient's zip code and application will automatically select the patient's state and city information.
    - **Insurance Record Management**
        - Insurance Information can be saved during the registration process.
        - If a patient has multiple insurance companies or parties, you can add them simultaneously. And can view them when you are viewing the patient record.
    - **Contact Record Management**
        - Application has the facility to add all types of contact information
        - You can change the patient's contact information
- **EMR**

  EMR module manages the medical records of your patients.
    - **Medical Chart Management**
        - You can make medical chart for each patient, just input the required information and application will generate a chart for you based on standard templates.
        - You can view and update patient chart.
    - **Problem Management**
        - You can add problems to a patient's record directly.
        - You can view and update a patients problems
    - **Alerts Management**
        - You can add Alert information related to a particular patient.
        - You can view and update Alert information of a patient.
    - **Medicine Management**
        - You can add Medicine related information related to a particular patient.
        - You can view and update Medicine related information of a patient.

- **Direct Exam Management**
  - You can create Direct Exam Information. In it you can add physical information other types of information regarding the patient health.
- **Flags & Document Management**
  - You can create different type documents and send these to any other user within your domain.
  - User can update an existing document; append information in existing documents.
  - For short communication, we have provided flags in our application which are short messages.
  - You can create more users of the application and you can communicate with them. You can communicate in two ways.
  - You can communicate to the staff using flags (short messages)
  - You can communicate others by sharing documents with each other.
- **Flow Sheet Management**
  - Users can make flow sheets that contain date wise information about the patients
  - User can view and update a flow sheet.
- **Order Management**
  - User can save order information related to patient.
  - User can save pharmacy related information and later on you can update and view order information.
- **Graphs & Handouts**
  - User can create graphs. Graphs give you most appropriate idea about the patient health.
  - User can write different types of handouts against each patient and they can be used in future.

- **Scheduler**

  Scheduler is another powerful feature of the application. You can schedule appointments for patients and you can manage you staff using this module.

- **Appointment Management**
    - User can view Daily, Weekly and Monthly View of scheduled appointments of providers.
    - User can add appointment for the providers.
    - User can add and manage block time of the providers.
    - User can add appointments against each patient.
    - User can check the past and future appointment.
    - You can check, manage and reschedule manage patient’s appointments.

- **User Management**
    - You can add and manage different type of system users like receptionist, nurse and other medical staff.
    - You can manage multiple staff by making logins. If you have multiple providers in your hospital you can make logins too and using our scheduler, you can manage their appointments.
    - You can manage staff in shifts; you can check their availability to schedule their appointments.

METHODOLOGY *Chapter 03*

In this chapter we will discuss about the methodologies used throughout this project. We define system architecture and development life cycle of this project.

3.1 Software Development Cycle:

"In preparation for battle, I have always found that plans are useless, but planning is indispensable" (Dwight Eisenhower).

Since we know the most important factor is planning to determine success or failure of any software project. So we followed the Software Development Cycle which depends on analyzing, designing, implementing testing, maintaining. We describe one by one all these factors.

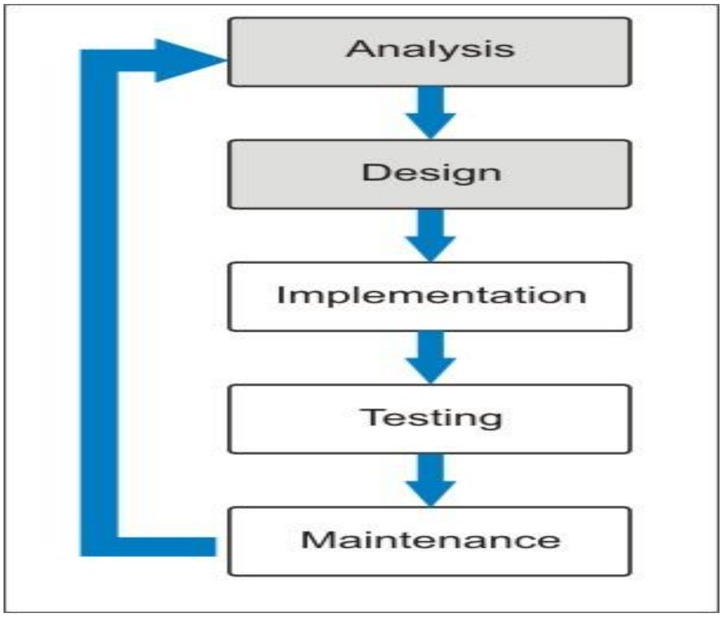


Figure 3.1: Software development cycle

3.1.1 Analyze:

a) **Background studies and approve:** From the first day we came to know about many projects offered by different faculty member and industrials. We selected to work on this project. We deeply studied the project description and follow the abstract with approval of advisory council.

b) **Related Literature Review:** For more understandings we went through different available systems as mentioned in Chapter 02.

c) **Technological Comparison:** For getting our objectives we compare all the technologies and choose ASP.NET Framework and MS Azure Cloud for this project.

3.1.2 Design:

a) **Use Cases and Mockup Design and Approve:** We started requirement analysis and get use cases. We design all use cases mockups Using Balsamiq Software to show prototype to approve. Balsamiq Software shows like real application navigation and demo.

b) **Entity Relation Diagram (ERD) Design and Approve:** After achieving above objective we started designing ERD and discuss with advisory council to approve.

3.1.3 Implementation:

In this we followed 3-tier architecture and describe below.

3.1.4 3-tierArchitecture:

Our system has 3-tier Architecture design and deployment on MS Azure Cloud. So we will discuss one by one tier with used methodology.

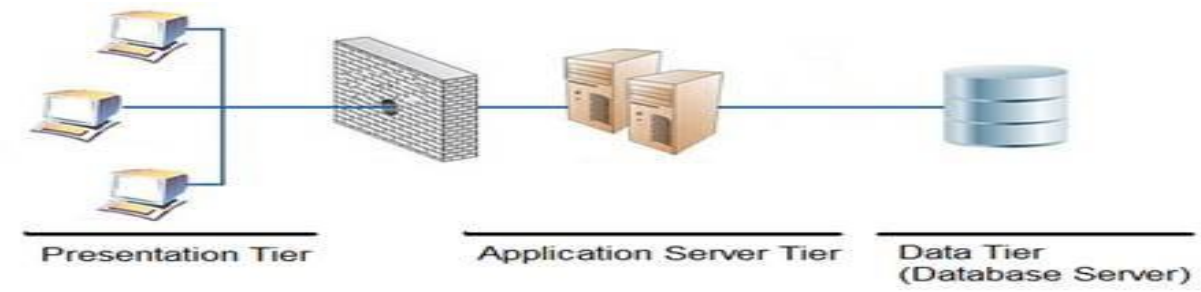


Figure: 3.2: 3-tier Architecture

3.1.4.1 Client Tier:

In this tier, system provides web based interfaces to the client/browser. So client tier is independent of other two tiers. Hence client sends and receives request and response through HTTP connection to Application tier.

To complete our presentation layer we have used:

1) **ASP.NET Master Pages:**

ASP.NET master pages allow you to create a consistent layout for the pages in your application. A single master page defines the look and feel and standard behavior that you want for all of the pages (or a group of pages) in your application. You can then create individual content pages that contain the content you want to display. When users request the content pages, they merge with the master page to produce output that combines the layout of the master page with the content from the content page.

We have used two master pages in our web application.

- **MainSite Master Page**: For Home page and other marketing pages of our web application
- **Site Master Page**: Layout for the users (Administrator, Practitioner, Receptionist) after they login.

2) **ASP.NET Standard Controls**:

- Button
- Checkbox
- Checkbox List
- Dropdown list
- File upload
- Hyperlink
- Image
- Label
- Link button
- RadioButtonlist
- Textbox

3) **ASP.NET Validation Controls**:

- RequiredFieldValidator

4) **ASP.NET Data Controls**:

- Grid view

5) **ASP.NET Ajax Extensions Controls**:

- Script Manager

- Update panel

6) **HTML Controls**:
    - Table
    - Div
    - Image

We have used Ajax to make our web application better, faster, interactive and more user-friendly.

7) **ASP.NET Ajax Control Toolkit**:
    - Calendar Extender
    - Tab Container

8) **TelerikRadSchedular:**
    - We customized TelerikRadSchedular according to our project requirements such as to display and set practitioner's daily, weekly, monthly schedule and also their timelines.

9) **JavaScript and JQuery:**
    - We have used a little bit JavaScript to make some client side manipulations and a little bit JQuery to make front page of Web application more interactive.

10) **Cascading Style sheets (CSS):**
    - We have used internal and external CSS to formatting pages.

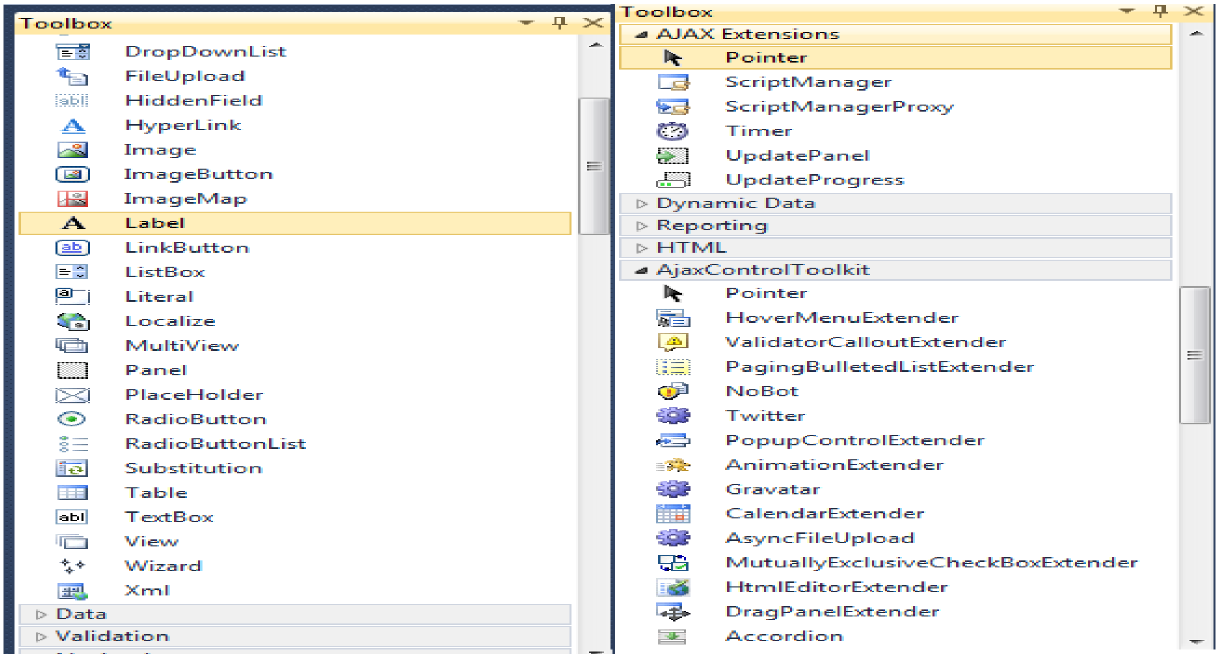


Figure 3.3: ASP.NET Server Controls

3.1.4.2 Application Tier:

In this tier system implements business logic so system gets different requests from many clients and responses all the requests by using HTTP connection with database tier and web services. Application tier sends request to database tier and gets response back and vice versa.

This layer includes the following types of files to cater the interface and the system logic implemented:

Aspx: For presentation layout as discuss above.

Aspx.cs: (The core business logic for the system)

CSS Files: For the easiness to set UI.

Web.config: For the configuration of all the files belong to system.

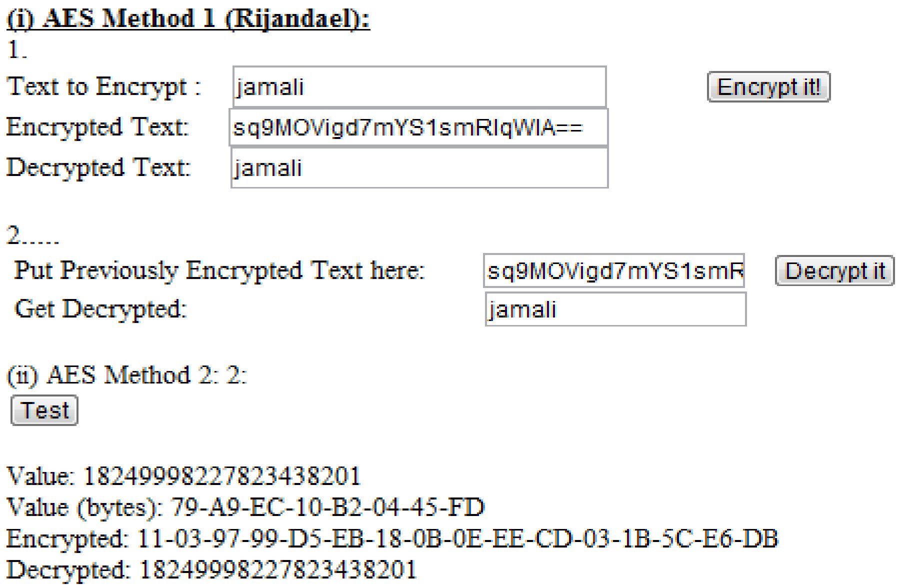
Site.Master: For the single design for all the pages.

Site.Master.cs: For the core business logic of the site master page.

Components used in the application tier:

AES Encryption and Decryption:

AES is a block of cipher algorithms originating from Rijndael algorithms. This algorithm has been chosen because of its very high security levels and performance combined with rapidity of calculation.



(i) AES Method 1 (Rijandael):

1.

Text to Encrypt : jamali

Encrypted Text: sq9MOVigd7mYS1smRIqWIA==

Decrypted Text: jamali

2.....

Put Previously Encrypted Text here: sq9MOVigd7mYS1smR

Get Decrypted: jamali

(ii) AES Method 2: 2:

Value: 18249998227823438201
Value (bytes): 79-A9-EC-10-B2-04-45-FD
Encrypted: 11-03-97-99-D5-EB-18-0B-0E-EE-CD-03-1B-5C-E6-DB
Decrypted: 18249998227823438201

SMTP (Simple Mail Transfer Protocol):

Simple Mail Transfer Protocol (SMTP) is an Internet standard for electronic mail (e-mail) transmission across Internet Protocol (IP) networks, uses TCP port 25. While electronic mail servers and other mail transfer agents use SMTP to send and receive mail messages, user-level client mail applications typically only use SMTP for sending messages to a mail server for relaying.The protocol is used to aware users their passwords by sending them an email.Further the connection of this layer with the database tier, it includes:

Functions:

Functions used to map database procedures.

Complex Type Objects:

Complex types are non-scalar properties of entity types that enable scalar properties to be organized within entities. Like entities, complex types consist of scalar properties or other complex type properties. Because complex types do not have keys, complex type objects cannot be managed by the Entity Framework apart from the parent object.

Libraries used:

System.IO

System.Data

System.Configuration

System.Web

System.Web.Security

System.Web.UI

System.Web.UI.WebControls

System.Web.UI.WebControls.WebParts

System.Web.UI.HtmlControls

System.Text

System.Web.SessionState

System.Drawing

System.Collections.Generic

System.Linq

System.Data

System.Data.Sql

System.Globalization

System.Data.SqlClient

System.ComponentModel

System.Security.Cryptography

System.Net.Mail

Sessions: It is used to maintain the state of the system and page navigation authentication.

Application Variables: It is used to save values for whole application navigation and operations.

3.1.4.3 Database Tier:

In this tier system provides database access so this tier receives requests from application tier and response back. For getting these objectives we have used ADO.NET Entity Framework.

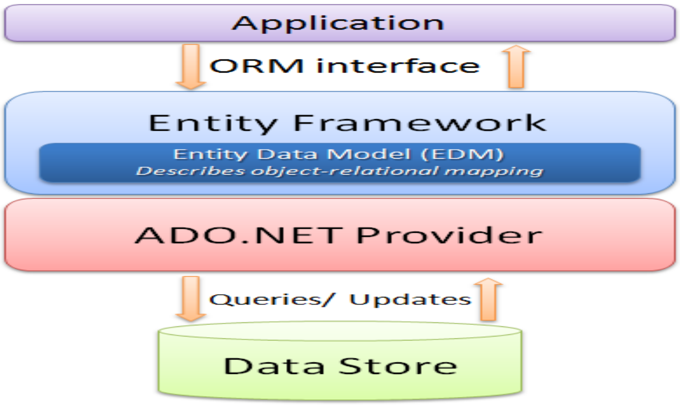


Figure 3.4: Entity Framework Architecture

3.1.5 ADO.NET Entity Framework:

The ADO.NET Entity Framework is a set of technologies that support the development of data-oriented software applications. The Entity Framework enables developers to work with data in the form of domain-specific objects and properties, such as customers and customer addresses, without having to concern themselves with the underlying database tables and columns where this data is stored. With the Entity Framework, developers can work at a higher level of abstraction when they deal with data, and can create and maintain data-oriented applications with less code than in traditional applications. [1] ADO.NET Entity Framework requires a model. A model is a combination of 3 files:

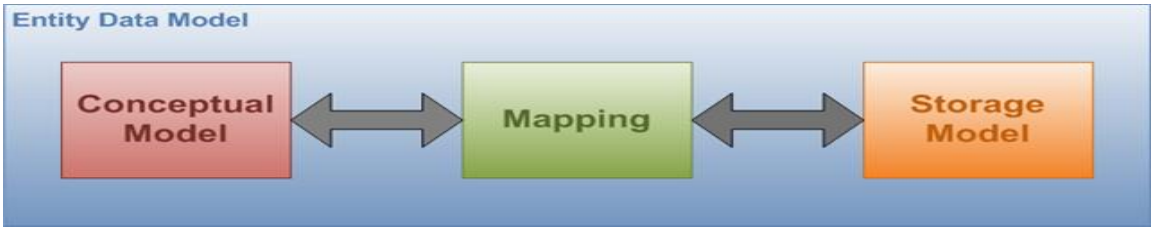


Figure 3.5: Entity Data Model

1. **Conceptual schema:** The conceptual schema describes your business entities.

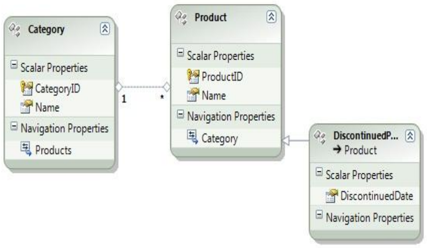
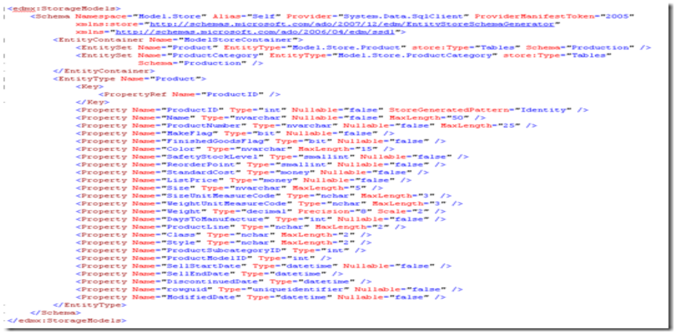


Figure 3.6: Conceptual Schema

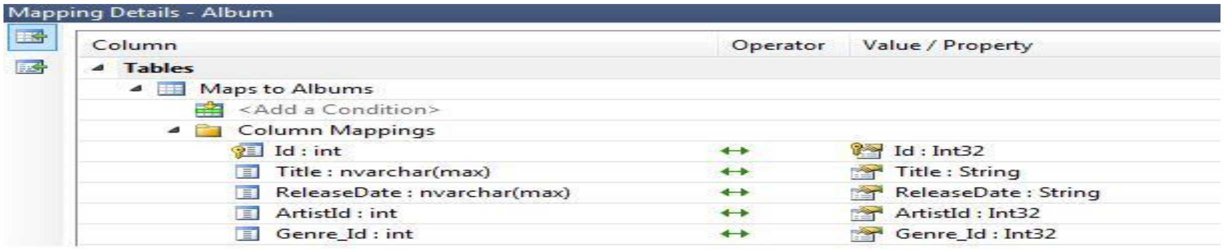
2. **Storage schema:** The storage schema describes your database schema.



```
<edmx:StorageModels>
    <Schema Namespace="Model.Store" Alias="Self" Provider="System.Data.SqlClient" ProviderManifestToken="2005"
            xmlns:store="http://schemas.microsoft.com/ado/2007/12/edm/EntityStoreSchemaGenerator"
            xmlns="http://schemas.microsoft.com/ado/2006/04/edm/ssdl">
        <EntityContainer Name="ModelStoreContainer">
            <EntitySet Name="Product" EntityType="Model.Store.Product" store:Type="Tables" Schema="Production" />
            <EntitySet Name="ProductCategory" EntityType="Model.Store.ProductCategory" store:Type="Tables"
                  Schema="Production" />
        </EntityContainer>
        <EntityType Name="Product">
            <Key>
                <PropertyRef Name="ProductID" />
            </Key>
            <Property Name="ProductID" Type="int" Nullable="false" StoreGeneratedPattern="Identity" />
            <Property Name="Name" Type="nvarchar" Nullable="false" MaxLength="50" />
            <Property Name="ProductNumber" Type="nvarchar" Nullable="false" MaxLength="25" />
            <Property Name="MakeFlag" Type="bit" Nullable="false" />
            <Property Name="FinishedGoodsFlag" Type="bit" Nullable="false" />
            <Property Name="Color" Type="nvarchar" MaxLength="15" />
            <Property Name="SafetyStockLevel" Type="smallint" Nullable="false" />
            <Property Name="ReorderPoint" Type="smallint" Nullable="false" />
            <Property Name="StandardCost" Type="money" Nullable="false" />
            <Property Name="ListPrice" Type="money" Nullable="false" />
            <Property Name="Size" Type="nvarchar" MaxLength="5" />
            <Property Name="SizeUnitMeasureCode" Type="nchar" MaxLength="3" />
            <Property Name="WeightUnitMeasureCode" Type="nchar" MaxLength="3" />
            <Property Name="Weight" Type="decimal" Precision="8" Scale="2" />
            <Property Name="DaysToManufacture" Type="int" Nullable="false" />
            <Property Name="ProductLine" Type="nchar" MaxLength="2" />
            <Property Name="Class" Type="nchar" MaxLength="2" />
            <Property Name="Style" Type="nchar" MaxLength="2" />
            <Property Name="ProductSubcategoryID" Type="int" />
            <Property Name="ProductModelID" Type="int" />
            <Property Name="SellStartDate" Type="datetime" Nullable="false" />
            <Property Name="SellEndDate" Type="datetime" />
            <Property Name="DiscontinuedDate" Type="datetime" />
            <Property Name="rowguid" Type="uniqueidentifier" Nullable="false" />
            <Property Name="ModifiedDate" Type="datetime" Nullable="false" />
        </EntityType>
    </Schema>
</edmx:StorageModels>
```

Figure 3.7: Storage schema

3. **Mapping schema:** The mapping schema is the glue that binds the conceptual schema to the storage schema by describing how the business entities translate to the database schema and vice versa.



Mapping Details - Album

Column	Operator	Value / Property
Tables		
Maps to Albums		
<Add a Condition>		
Column Mappings		
Id : int	↔	Id : Int32
Title : nvarchar(max)	↔	Title : String
ReleaseDate : nvarchar(max)	↔	ReleaseDate : String
ArtistId : int	↔	ArtistId : Int32
Genre_Id : int	↔	Genre_Id : Int32

Figure 3.8: Mapping Schema

3.1.6 Deployment on MS Azure Cloud:

As it is the era of moving towards cloud and those who want to move toward cloud based solutions will find financial savings in this approach as well as fault tolerance and a more responsive turn around in many cases.

Following is the demonstration of "How to deploy existing ASP.NET Web Application on MS Azure Cloud":

Pre-requisites:

- Backup of your web application before deployment on cloud.
- Download and install Windows Azure Cloud(https://www.windowsazure.com)
- Download and install Windows Azure tools for Visual Studio 2010.

After completing above requirements do the following steps:

1) Create Windows azure Account
2) Add Windows Azure deployment project to the web project
3) Deploy the web project to a new hosted service in the staging environment in your Windows Azure Account
4) Deploy and update Using Web Deploy
5) Create a SQL Azure database and configure the application to use that database when it runs in the cloud
6) Promote application from staging environment to production.

3.1.7 Testing:

Testing requires the satisfaction and assurance of user requirements so in this system we followed the following different test.

a. Unit testing
b. Integration testing
c. Usability testing
d. Performance testing
e. Scalability testing
f. Security testing
g. System testing

3.1.8 Maintenance:

It is the post project task in which service provides maintains the operations of the product.

DESIGN *Chapter 04*

In this chapter we will discuss about ERD, System Architecture, and System Features of SaaS (Software as a Service) Based Practice Management System.

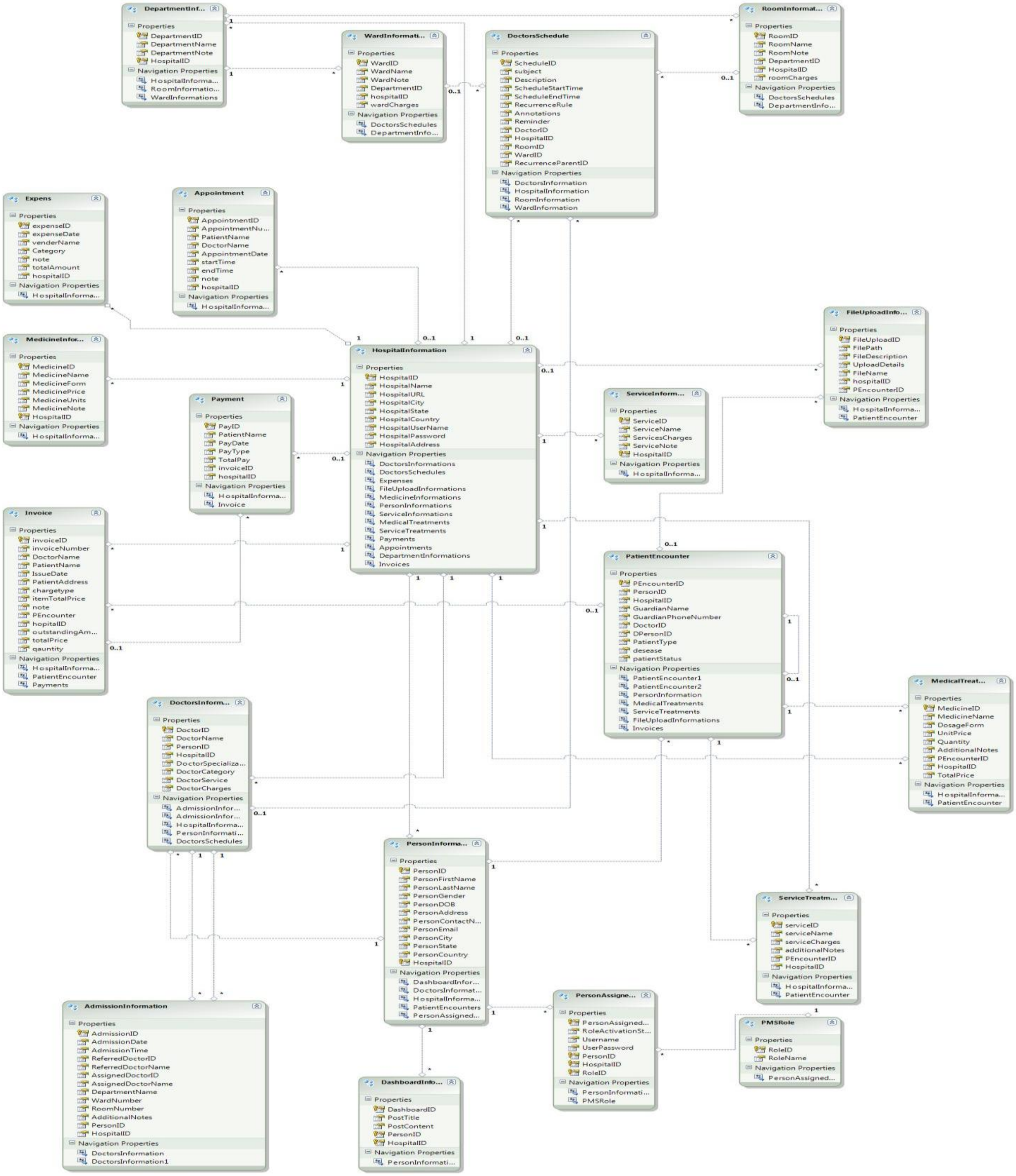


Figure 4.1: Entity Relationship Diagram

4.1 Brief Description Of Tables:

1. **Persons_Information:**

   In this table, system maintains the information of all the belonging persons.

   1. P_ID
   2. Hospital_ID
   3. Person_First_Name
   4. Person_Last_Name
   5. Person_Address
   6. Person_Gender
   7. Person_DOB
   8. Person_contact_number
   9. Guardian_contact_number
   10. PersonCity
   11. PersonState
   12. PersonCountry

2. **Patients_Vitals:**

   In this table, system maintains the information of all the belonging first aid checkup parameters of patient.

   1. **PV_ID**
   2. **PE_ID**
   3. PV_Weight
   4. PV_Height
   5. PV_BPSystolic
   6. PV_Pulse
   7. PV_Respiratory
   8. PV_Temperature

3. **Patients_Issue:**

   In this table, system maintains the information of the issues/diseases may appear.

   1. **P_Issue_ID**
   2. Disease_name

4. **Patients_Encountered:**

   In this table, system maintains the information of the entire belonging patient's issue/diseases who ia encountered in hospital.

   1. **PEI_ID**
   2. P_Issue_ID
   3. PE_ID
   4. Begin_Date
   5. Patient_Type
   6. Guardian_First_Name

    7. Guardian_Last_Name
    8. PersonID
    9. DoctorID

5. **Medicines_Information:**
   In this table, system maintains the information of the entire medicines.
    1. **MedicineID**
    2. Medicine_Name
    3. MedicineForm
    4. MedicinePrice
    5. medicineUnits
    6. MedicineNote
    7. HospitalID
6. **MedicineTreatment:**
   In this table, system maintains the information of the entire belonging patient's assigned medicines.
    1. **MedID**
    2. PE_ID
    3. Doctor_ID
    4. MedicineID
7. **Wards_Information:**
   In this table, system maintains the information of the wards of the hospital.
    1. **Ward_ID**
    2. WardName
    3. Number_Of_Bed
    4. WardCharges
    5. WardNote
    6. DepartmentID
    7. HospitalID
8. **Rooms_Information:**
   In this table, system maintains the information of the rooms of the hospital.
    1. **Room_ID**
    2. RoomName
    3. Room_Rent
    4. RoomNote
    5. DepartmentID
    6. **HopsitalID**
9. **Doctors_Information:**
   In this table, system maintains the information of all the belonging doctors of the hostel.
    1. **Doctor_ID**

2. P_ID
3. D_Type_ID
4. Doctor_Qualification
5. DoctorCategory
6. DoctorSpecialization
7. HospitalID
8. DoctorCharges

10. Doctors_Type:

In this table, system maintains the information of the entire types of doctors.

1. **D_Type_ID**
2. Doctor_Type
3. Dep_ID

11. Departments_Information:

In this table, system maintains the information of all the departments of hospital.

1. **Dep_ID**
2. Hospital_ID
3. Dep_Name

12. Patients_Appointment:

In this table, system maintains the information of the appointments of patients of particular doctor on his/her schedule.

1. **App_ID**
2. PE_ID
3. Schedule_ID
4. Appointment_Time
5. Appointment_Number
6. HospitalID

13. Doctors_Schedule:

In this table, system maintains the information of the schedule of the doctors.

1. **Schedule_ID**
2. Doctor_ID
3. Schedule_Date
4. Start_Time
5. End_Time
6. HospitalID

14. Services_Information:

In this table, system maintains the information of all the services served by hospital.

1. **Service_ID**
2. Service_Name
3. Service_charges

    4. HospitalID

15. **ServicesTreatment:**

    In this table, system maintains the information of all the offered services to encountered patient.

    1. **PES_ID**
    2. Service_ID
    3. PE_ID
    4. Service_Date
    5. hospitalID

16. **Hospitals_Information:**

    In this table, system maintains the information of the Hospitals/clients.

    1. **Hospital_ID**
    2. Hospital _Name
    3. Hospital _Location
    4. Hospital _City
    5. HospitalState
    6. HospitalCountry
    7. HospitalURL
    8. Hospital UserName
    9. Hospital _Password

17. **PMSRole:**

    In this table, system maintains the information of the different roles of the system.

    1. **RoleID**
    2. RoleName

18. **PersonAssignedRole:**

    In this table, system maintains the information of the different user with their roles.

    1. **PersonAssignedRoleID**
    2. RoleActivationStatus
    3. UserName
    4. UserPassword
    5. PersonID
    6. RoleID
    7. HospitalID

19. **Payments:**

    In this table, system maintains the information of the different payments of encountered patients.

    1. **PayID**
    2. PatientName
    3. PayDate

   4. PayType
   5. TotalPay
   6. invoiceID
   7. HospitalID

20. **Invoice:**

    In this table, system maintains the information of the different invoices occur during hospital operations.

    1. **InvoiceID**
    2. invoiceNumber
    3. PatientName
    4. DoctorName
    5. IssueDate
    6. PatientAddress
    7. ChargeType
    8. ItemTotalPrice
    9. Note
    10. OutstandingAmount
    11. InvoiceTotalPrice
    12. Qauntity
    13. totalPrice
    14. PEncounterID
    15. HospitalID

21. **PatientFiles:**

    In this table, system maintains the information of the different files of encountered patients.

    1. **FileID**
    2. PatientName
    3. FileName
    4. FilePath
    5. FileDescription
    6. UplaodDetails
    7. PEncounterID
    8. HospitalID

22. **Expenses:**

    In this table, system maintains the information of the different expenses occur in hospital operation.

    1. **ExpenseID**
    2. ExpenseDate
    3. VenderName

4. Category
5. Note
6. TotalAmount
7. HospitalID

23. DashBaord:

In this table, system maintains the information of the different posts of doctor for social awareness.

1. **DashID**
2. Posttitle
3. PostContent
4. PersonID
5. HospitalID

24. PatientAdmission:

In this table, system maintains the information of the different admission of encountered patients.

1. **AdmissionID**
2. AdmissionDate
3. AdmissionTime
4. ReferredDoctorID
5. ReferredDoctorName
6. AssignedDoctorName
7. AssignedDoctorID
8. DepartmentID
9. WardNumber
10. RoomNumber
11. AdditionalNotes
12. PersonID
13. HospitalID

4.2 System Architecture:

In this we followed 3-tier architecture and describe bellow.

4.2.1 3-tierArchitecture:

Our system has 3-tier Architecture design and deployment on MS Azure Cloud. So we will discuss one by one tier with used methodology.

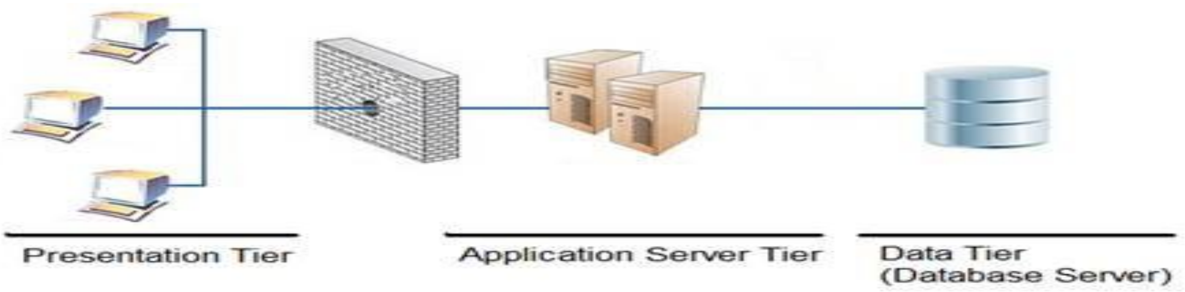


Figure 4.2: 3-tier architecture

4.2.1.1Client Tier:

In this tier, system provides web based interfaces to the client/browser. So client tier is independent of other two tiers. Hence client sends and receives request and response through HTTP connection to Application tier.

4.2.1.2 Application Tier:

In this tier system implements business logic so system gets different requests from many clients and responses all the requests by using HTTP connection with database tier and web services. Application tier sends request to database tier and gets response back and vice versa.

4.2.1.3 Database Tier:

In this tier system provides database access so this tier receives requests from application tier and response back. For getting these objectives we have used ADO.NET Entity Framework.

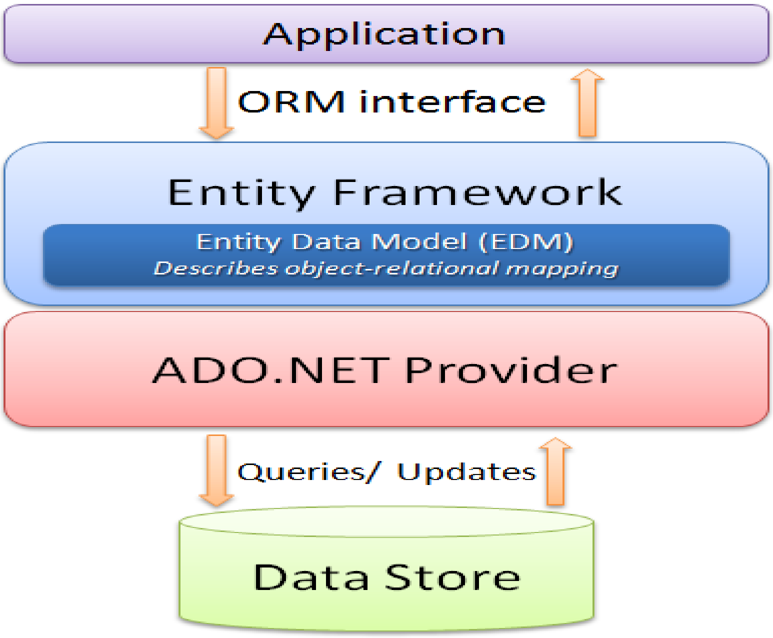


Figure 4.3: Entity Framework architecture

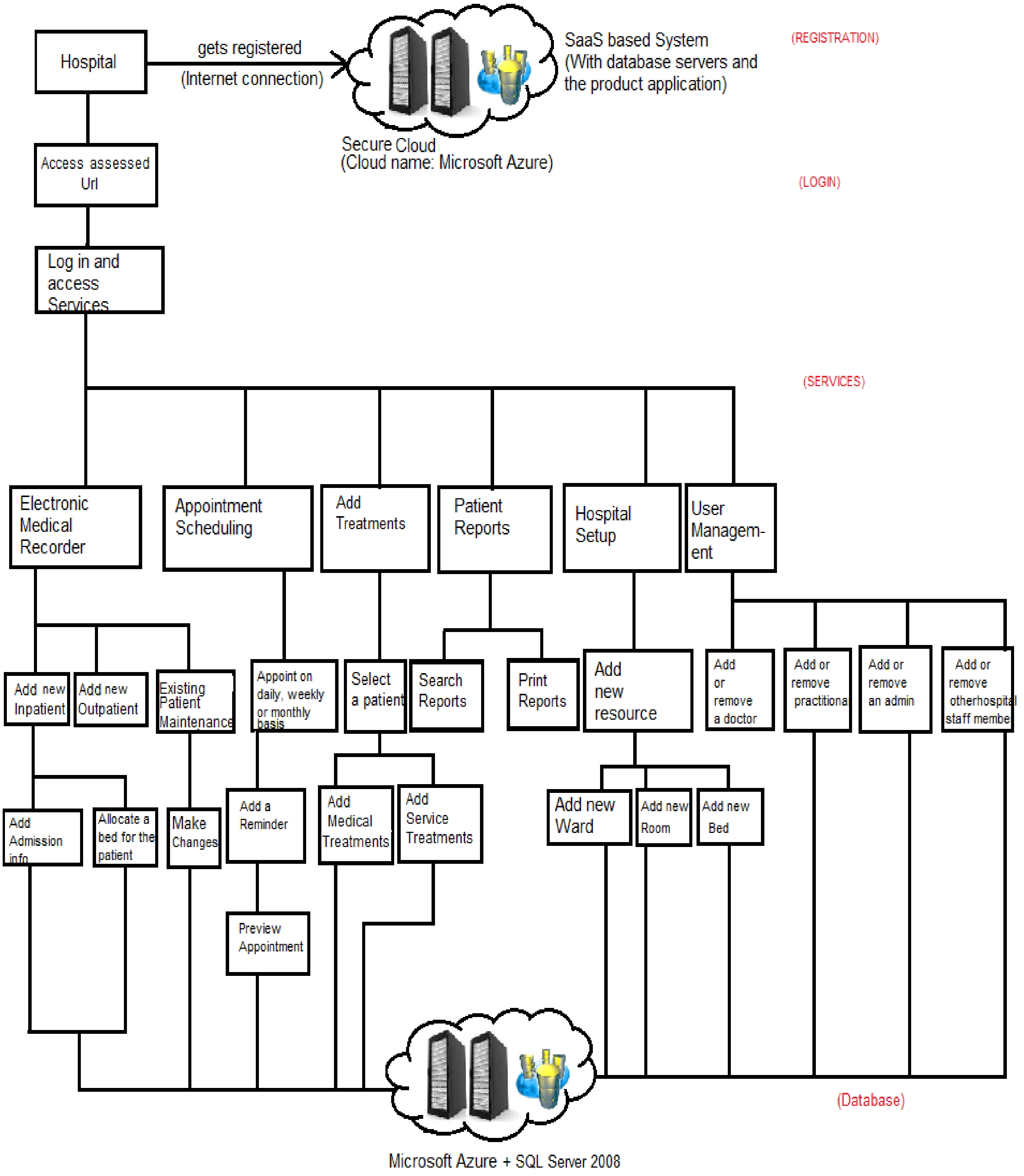


Figure 4.4: Sequence Diagram

4.3 System Features:

Use cases:

Use case ID: 1

Use case: **Log in**

Actor: RE / Doctor / Admin

Purpose: To access the system services.

Cross Reference:

Functional Requirements:

- The system takes the user name and password to the page where he can access services according to this role and privileges.

Pre-conditions:

1. The user should have a registered account.
2. User should have valid user name and password.

Course of event:

1) User clicks the login link on the homepage.
2) System will load the login page and displays before the user.
3) User will fill the following details:
    - Username
    - Password
    - Select the registered listed company name
4) User will click on the log in button.
5) System will validate the information entered. On true, takes him to the page where services are available to him.

Alternate course of event: On entering invalid username or password

1) System displays an error message on invalid username or password.
2) User re-enters/updates the invalid/wrong data. (And repetition from step 3 takes place).

Use case ID 2:

Use case: **Patient admission**

Actor: RE / Admin

Purpose: To store patient demographics in the system who has come to the hospital for healthcare.

Cross Reference:

Functional Requirements:

1. The system will create a unique patient entity for each patient.
2. The system will show the association of patient with particular visited hospital.
3. The system will show list of medications and doctors assigned for health care.
4. The system will show the services consumed by the patient.

Pre-conditions:

1. The user should be logged in.
2. The patient should not exist in database already.

Course of event:

1) User points the cursor to the patient maintenance tab.
2) User clicks the insert a new patient option from the menu bar.
3) User fills the personal details form with data like:
    - Patient ID
    - First Name
    - Last Name
    - Age
    - Occupation etc
4) User clicks on the next step button to proceed to the guardian information page.
5) System validates the entered information and displays the next page on correct validation.
6) User fills the guardian information form with data like
    - Guardian ID
    - First Name
    - Last Name
    - Relation with patient etc
7) User clicks the next step button to proceed to the patient information page.
8) System validates the entered information and displays the next page on correct validation.
9) User fills the patient information form with data like:
    - Admission ID
    - Admission Date
    - Admission Time
    - Selects the disease from a list by clicking browse button.

10) User clicks the save new patient button to save the new patient's information.
11) System validates the entered data and saves all the information successfully.

Alternate course of event: User leaves a required field empty or fills a field with invalid data.

1) System displays an error message on invalid or empty left data.
2) User re-enters/updates the invalid data.
3) User keeps on preceding the next page and repeats step no. 2 if step no.1 occurs due to invalid data.
4) On the last page patient information, user clicks on the save new patient button to save the new patient.
5) Step no. 11 of course of events repeats.

Use case ID: 3

Use case: **Updating existing patient**

Actor: RE / Admin

Purpose: To save the updated demographics changes of patient's record.

Cross Reference:

Functional Requirements:

5. The system will update a unique patient entity for each patient.
6. The system will show the association of patient with particular visited hospital.
7. The system will show list of medications and doctors assigned for health care.
8. The system will show the services consumed by the patient.

Pre-conditions:

1) The user should be login.
2) Patient's information should be already stored in the system.
3) The changes do not affect the confidentiality of the data.

Course of event:

1) User points the cursor to the patient maintenance tab.
2) User clicks the update existing patient option from the menu bar.
3) User updates the personal information if changed.
4) User clicks on the next step button to proceed to the guardian information page.
5) System validates the entered information and displays the next page on correct validation.
6) User updates patient guardians information

7) User clicks the next step button to proceed to the patient information page.
8) System validates the entered information and displays the next page on correct validation.
9) User fills the patient information form with data like:
    - Admission ID
    - Admission Date
    - Admission Time
    - Selects the disease from a list by clicking browse button.
10) User clicks the s patient button to save the existing patient's updated information.
11) System validates the entered data and saves all the information successfully.

Alternate course of event: User leaves a required field empty or fills a field with invalid data.

1) System displays an error message on invalid or empty left data.
2) User re-enters/updates the invalid data.
3) User keeps on proceeding to the next page and repeats step no. 2 if step no.1 occurs due to invalid data.
4) On the last page patient information, user clicks on the save patient button to save the updated information of existing patient.
5) Step no. 11 of course of events repeats.

Use case ID: 4
Use case: **Deleting a patient record**
Actor: RE / Admin
Purpose: Delete a patient's demographic history that lives no more but its record remains in database.
Cross Reference:
Functional Requirements:

1. The System will remove patient's demographic record(s).
2. The system will not select the patient more because it's disabled to select.
3. System will store patient's whole previous record in dead patients.

Pre-conditions:

1.The user should be logged in.
2.The patient should have no more.
3.The correct connection with dead patient's records.

Course of event:

1) User points towards the patient maintenance menu tab.
2) User selects the delete a patient menu item.

3) User selects criteria of searching like:
    - Patient ID
    - Patient’s Last Name
    - Disease etc
4) User fills the search by field.
5) User presses the search button.
6) System fetches the matching records in a tabular form.
7) User ticks/selects the desired record(s) to be deleted.
8) User clicks precede button.
9) System deletes the selected records and navigates to another page and displays a message the record(s) have been deleted.

Alternate course of event: User enters wrong data in the search by field.

Repetition of course of event step 1 to 3.
4) User enters wrong data in the search by field.
5) User presses the search button.
6) System displays an error message.
7) User updates the data in search by field.
8) Repetition of course of event step 5 to 7.

Use case ID: 5

Use case: **Doctor Registration**

Actor: Admin

Purpose: To create unique doctor entity with his/her qualifications and specialization in particular departments to access the assigned services and role.

Cross Reference:

Functional Requirements:

1. System will create a unique doctor with its personal information, qualification and unique departments.
2. System will show assigned department and wards and patients accordingly.
3. System will create the doctor with particular hospital and assign the roles.

Course of event:

1) User points the cursor to Doctor’s maintenance tab.
2) User clicks the Doctor’s maintenance option from the menu bar.

3) User fills the personal details form with data like:
    - Name
    - Address
    - Phone number etc
4) User also fills employment details form with data like:
    - Doctor’s specialization
    - Doctor category
    - Doctor Service etc
5) User clicks the save button to save the Doctor’s information.
6) System validates the entered data and saves all the information successfully.

Alternate course of event: User leaves a required field empty or fills a field with invalid data.

1) System displays an error message on invalid or empty left data.
2) User re-enters/updates the invalid data.
3) User clicks on the save button to save the doctor’s information.

Use case ID: 6

Use case: **Update Doctor Information**

Actor: Admin/doctor

Purpose: To keep updated information of qualification and specialization of doctor.

Cross Reference:

Functional Requirements:

1. System will update a unique doctor with its personal information, qualification and unique departments.
2. System will show assigned department and wards and patients accordingly.
3. System will create the doctor with particular hospital and assign the roles.
4. The system also maintains the previous information of doctors.
5. System will ensure the confidentiality of data.
6. System will inform doctor as well as admin for updates.

Course of event:

1) User points the cursor to Doctor’s maintenance tab.
2) User clicks the Doctor’s maintenance option from the menu bar.
3) And click on Doctors Search Wizard to search doctor whose data to be updated.

4) After searching doctor user fills the personal details if updated.
5) User also fills employment details if updated.
6) User clicks the save button to save the Doctor's updated information.
7) System validates the entered data and saves all the information successfully.

Alternate course of event: User leaves a required field empty or fills a field with invalid data.

1) System displays an error message on invalid or empty left data.
2) User re-enters/updates the invalid data.
3) User clicks on the save button to save the doctor's updated information.

Use case ID: 7

Use case: **Doctor's Schedule maintenance**

Actor: Admin/ Doctor

Purpose: To know the doctor's availability information.

Cross Reference:

Functional Requirements:

1. System will create the time line of doctor schedule in particular time and date.
2. System will show the availability of doctor in particular time and date.

Course of event:

1) User points the cursor to Doctor's maintenance tab.
2) User clicks the Doctor's Schedule maintenance option from the menu bar.
3) And click on Doctors Search Wizard to search doctor whose schedule to be updated.
4) After searching doctor ID user clicks on "Setup Doctor's channeling Times".
5) User selects the date from calendar and enters availability timings of doctor of that date.
6) User clicks the save button to save the Doctor's Schedule information.
7) System validates the entered data and saves all the information successfully.

Alternate course of event: User leaves a required field empty or fills a field with invalid data.

1) System displays an error message on invalid or empty left data.
2) User re-enters/updates the invalid data.
3) User clicks on the save button to save the doctor's schedule information.

Use case id: 8

Use case: **Add medicine**

Actor: Admin/RE

Purpose: To store the available medicine for prescription.

Cross Reference:

Functional Requirements:

1. The system will create unique medicine entity.
2. System shows it when doctor need to assign.

Course of event:

1) User clicks on to Medicine maintenance tab.
2) After that user clicks on add medicine button and fills the data like:
    - Medicine Name
    - Medicine Price
    - Medicine Quantity etc
3) User clicks the save button to save the medicine information.
4) System validates the entered data and saves all the information successfully.

Alternate course of event: User leaves a required field empty or fills a field with invalid data.

1) System displays an error message on invalid or empty left data.
2) User re-enters/updates the invalid data.
3) User clicks on the save button to save the medicine information.

Use case ID: 9

Use case: **Search and update medicine**

Actor: Admin/RE

Purpose: To facilitate user to search and to keep updated medicine information.

Cross Reference:

Functional Requirement:

1. The system will provide medicine information that is searched in order to update.
2. System will update the medicine stored statistics accordingly.

Course of event:

1) User clicks on to Medicine maintenance tab.
2) After that user clicks on search medicine button and search medicine by:
    - Medicine id
    - Medicine Name
3) User clicks on the medicine information.
4) User fills the updated information.
5) User clicks on the save button to save updated information about medicine.
6) System validates the entered data and saves all the information successfully.

Alternate course of event: User leaves a required field empty or fills a field with invalid data.

1) System displays an error message on invalid or empty left data.
2) User re-enters/updates the invalid data.
3) User clicks on the save button to save the medicine information.

Use case id: 10

Use case: **Add Department**

Actor: Admin

Purpose:

1. To deal patients for different types of diseases.
2. To deal with doctors for different specializations.
3. To categorize the wards and rooms in different departments.

Cross Reference:

Functional Requirements:

1. System will create unique department with specialization.

Course of event:

1) User clicks on to Departments maintenance tab.
2) After that user clicks on add department button and fills the data like:
    - Department ID
    - Department Name
3) User clicks the save button to save the Department information.
4) System validates the entered data and saves all the information successfully.

Alternate course of event: User leaves a required field empty or fills a field with invalid data.

1) System displays an error message on invalid or empty left data.
2) User re-enters/updates the invalid data.
3) User clicks on the save button to save the department information.

Use case ID: 11

Use case: **Search and update Department**

Actor: Admin

Purpose: To facilitate user to search and to keep updated department information.

Cross Reference:

Functional Requirements:

1. The system will provide department information that is searched in order to update.
2. The system will update particular department specialization.
3. System also maintains the previous data as back up.

Course of event:

1) User clicks on to department maintenance tab.
2) After that user clicks on department medicine button and search department by:
    - Department id
    - Department Name
3) User clicks on the department information.
4) User fills the updated information.
5) User clicks on the save button to save updated information about department.
6) System validates the entered data and saves all the information successfully.

Alternate course of event: User leaves a required field empty or fills a field with invalid data.

1) System displays an error message on invalid or empty left data.
2) User re-enters/updates the invalid data.
3) User clicks on the save button to save the department information.

Use case ID: 12

Use case: **Search and delete Department**

Actor: Admin

Purpose: To facilitate user to search and to remove information of department that is no longer available for providing services.

Cross Reference:

Functional Requirements:

1. The system will remove information of department that is searched.
2. System will remove the department which is closed by hospital management.
3. System will store departments all data as back up.

Course of event:

1) User clicks on to department maintenance tab.
2) After that user clicks on departments button and search department by:
    - Department id
    - Department Name
3) User clicks on the department information.
4) User clicks on the delete button to delete information about department.
5) System prompts a window to confirm and delete all the information successfully.

Alternate course of event:

User clicks on cancel button after selecting other department mistakenly.

Use case id: 13

Use case: **Add Ward**

Actor: Admin

Purpose: To deal with inpatient admission in a hospital.

Cross Reference:

Functional Requirements:

1. System will create a unique ward with department.
2. The system will store the number of beds in the ward.
3. System will maintain information about multiple Wards in a department.

Course of event:

1) User clicks on to Wards maintenance tab.

2) After that user clicks on add Wards button and fills the data like:
    - Ward ID
    - Ward Name
3) User clicks the save button to save the Ward information.
4) System validates the entered data and saves all the information successfully.

Alternate course of event: User leaves a required field empty or fills a field with invalid data.

4) System displays an error message on invalid or empty left data.
5) User re-enters/updates the invalid data.
6) User clicks on the save button to save the Ward information.

Use case ID: 14

Use case: **Search and update Ward**

Actor: Admin/RE/Doctor

Purpose: To facilitate user to search and to keep updated Ward information and store inpatient maintenance.

Cross Reference:

Functional Requirements:

1. System will provide the ward for searching.
2. System updates the ward information and shows it to other on search.

Course of event:

1) User clicks on to Ward maintenance tab.
2) After that user clicks on Ward button and search Ward by:
    - Ward id
    - Ward Name
3) User clicks on the Ward information.
4) User fills the updated information.
5) User clicks on the save button to save updated information about Ward.
6) System validates the entered data and saves all the information successfully.

Alternate course of event: User leaves a required field empty or fills a field with invalid data.

6) System displays an error message on invalid or empty left data.
7) User re-enters/updates the invalid data.
8) User clicks on the save button to save the Ward information.

Use case ID: 15

Use case: **Search and delete Ward**

Actor: Admin

Purpose: To facilitate user to search and to remove Ward that is no longer available for providing services.

Cross Reference:

Functional Requirements:

1. The system will remove Ward that is searched and is no linger for serving.
2. System will maintain its backup.

Course of event:

1. User clicks on to Ward maintenance tab.
2. After that user clicks on Ward maintenance button and search Ward by:
   - Ward id
   - Ward Name
3. User clicks on the Ward information.
4. User clicks on the delete button to delete information about Ward.
5. System prompts a window to confirm and delete all the information successfully.

Alternate course of event:

User clicks on cancel button after selecting other Ward mistakenly.

Use case id: 16

Use case: **Add Room**

Actor: Admin

Purpose: To deal with inpatient admission in a particular department if a hospital.

Cross Reference:

Functional Requirements:

1. System will create unique room in particular department.
2. The system can create multiple rooms.

Course of event:

1. User clicks on to rooms maintenance tab.
2. After that user clicks on add rooms button and fills the data like:
   - Room ID
   - Room Name
3. User clicks the save button to save the room information.
4. System validates the entered data and saves all the information successfully.

Alternate course of event: User leaves a required field empty or fills a field with invalid data.

1. System displays an error message on invalid or empty left data.
2. User re-enters/updates the invalid data.
3. User clicks on the save button to save the room information.

Use case ID: 17

Use case: **Search and Room**

Actor: Admin/RE/Doctor

Purpose: To facilitate user to search and to keep updated Room information.

Cross Reference:

Functional Requirements:

1. The system will provide Room information and status on search.
2. System will update the used rooms to others.

Course of event:

1. User clicks on to Room maintenance tab.
2. After that user clicks on Room button and search Room by:
   - Room id
   - Room Name
3. User clicks on the Room information.
4. User fills the updated information.
5. User clicks on the save button to save updated information about Room.
6. System validates the entered data and saves all the information successfully.

Alternate course of event: User leaves a required field empty or fills a field with invalid data.

1. System displays an error message on invalid or empty left data.
2. User re-enters/updates the invalid data.
3. User clicks on the save button to save the Room information.

Use case ID: 18

Use case: **Search and delete Room**

Actor: Admin

Purpose: To facilitate user to search and to remove Room that is no longer available for providing services.

Cross Reference:

Functional Requirements:

1. System will remove the room which will not provide services more.
2. System will maintain its back up of previous data.

Course of event:

1. User clicks on to Room maintenance tab.
2. After that user clicks on Room maintenance button and search Room by:
   - Room id
   - Room Name
3. User clicks on the Room information.
4. User clicks on the delete button to delete information about Room.
5. System prompts a window to confirm and delete all the information successfully.

Alternate course of event:

User clicks on cancel button after selecting other Room mistakenly.

Use case ID: 19
Use case: **Add medical treatments for a patient**
Actor: Doctor
Purpose: to assign medicine to a patient encountered.
Cross Reference:
Functional Requirements:

1. System will store medicine details for a patient.
2. System will also show previous assigned medicine.
3. System also stores all the prescription with time and date and doctor’s ID.

Pre-conditions:

1) The user should be logged in.
2) The user should have privileges/access to the medical treatments.

Course of events:

1) User points towards the treatments menu tab.
2) User points towards the medical treatments sub-menu tab.
3) User selects to add medical treatment menu item.
4) User fills the form with data like:
    - Medicine Name
    - Date of issue
    - Unit price
    - Quantity etc
5) User clicks the add button.
6) System validates the data and stores it.
7) System displays a message the data has successfully been stored.

Alternate course of events: User enters invalid data or leaves required fields empty.
Repetition course of event step 1 to 3.

4) User fills the form, entering invalid data or leaves required field(s) empty.

5) User clicks the add button.

6) System displays an error message.

7) User updates the fields.

Repetition course of event step 5 to 7.
Use case ID: 20
Use case: **Updating an existing medical treatment details.**
Actor: Doctor
Purpose: to update of an existing medical treatment record.
Cross Reference:
Functional Requirements:

1. System replaces the new information entered with an existing medical treatment.

Pre-conditions:
1) The user should be logged in.
2) The user should have privileges/access to the medical treatments.

Course of events:
1) User points towards the treatments menu tab.
2) User points towards the medical treatments sub-menu tab.
3) User selects the update a medical treatment menu item.
4) User selects criteria of searching like:
    - Treatment ID
    - Medicine ID

- Medicine Name etc

5) User fills the search by field.
6) User presses the search button.
7) System fetches the matching records in a tabular form.
8) User selects the desired record to be updated.
9) User clicks precede button.
10) System fills the form with the previously saved information and displays before the user.
11) User edits the field(s) and enters the new information.
12) User clicks the save button.
13) System saves the new information and displays a success message.

Alternate course of events: User enters wrong data in the search by field.

Repetition of course of event step 1 to 4.
5) User enters wrong data in the search by field.
6) User presses the search button.
7) System displays an error message.
8) User updates the data in search by field.
Repetition of course of event step 6 to 13.
Use case ID: 21
Use case: **Add a service treatment**
Actor: Doctor/RE/Admin
Purpose: to add a service for a patient.
Cross Reference:
Functional Requirements:

1. System will store the offered services.
2. System also shows the previous consumed services.
3. System will save service with time and date.
4. System records a new service treatment.

Pre-conditions:
1) The user should be logged in.
2) The user should have privileges/access to the service treatments.

Course of events:
1) User points towards the treatments menu tab.
2) User points towards the service treatments sub-menu tab.
3) User selects to add a service treatment menu item.
4) User fills the form with data like:
    - Service Name
    - Service Charges

5) User clicks the add button.
6) System validates the data and stores the information.
7) System displays a message the data has been stored successfully.

Alternate course of events: User leaves a required field blank or enters an invalid data.

1) System displays an error message.
2) User enters the valid data and fills the form.
3) User clicks the add button.
4) System validates the data and stores the information.
5) System displays a message the data has been stored successfully.

Use case ID: 22

Use case: **Update an existing service treatment**

Actor: RE/ Admin/Doctor

Purpose: to update of a service treatment for a patient previously stored takes place.

Cross Reference:

Functional Requirements:

1. System updates and stores the new service treatment information instead previous one.

Pre-conditions:

1) The user should be logged in.
2) The user should have privileges/access to the service treatments.

Course of events:

1) User points towards the treatments menu tab.
2) User points towards the service treatments sub-menu tab.
3) User selects the update a service treatment menu item.
4) User selects the search criteria like:
    - Service ID
    - Service Name
5) User fills the search for field with appropriate text.
6) User clicks the search button.
7) System fetches and shows matching data in a tabular form.
8) User selects the desired data.
9) System shows the data in the form.
10) User updates the desired field(s).
11) User clicks the save button
12) System validates and saves the updated information and displays a success message.

Alternate course of events: User enters a non-matching data in the search for field.

1) System displays an error message.
2) User updates the wrong data.

Repetition of course of event step 6 to 12.

Use case ID: 23

Use case: **Discharge a patient**

Actor: Doctor / RE / Admin

Purpose: to discharge a patient from the hospital.

Cross Reference:

Functional Requirements:

1. System updates the ward bed or room empty which is given to the patient.
2. System will save all the data of patient.
3. System shows the status of the patient and store respectively.

Pre-conditions:

1) The user must be logged in.
2) The user should have privileges/access to the discharge patients.
3) The patient's history must exist.

Course of events:

1) User selects to discharge a patient from the menu.
2) User fills the form with data like:
    - Discharge Date
    - Discharge Time
3) User clicks the discharge button.
4) System validates the data, updates the information and displays a success message.

Alternate course of events: User enters invalid data.

1) System displays an error message.
2) User updates the data.
3) User clicks the discharge button.
4) System validates the data, updates the information and displays a success message.

Use case ID: 24

Use case: **Schedule a patient's appointment**

Actor: Admin / RE/Doctor

Purpose: to record and maintain patient appointments schedule with the doctor.

Cross Reference:

Functional Requirements:

1. System will show appoint of patient with doctor in time line of doctor.
2. System will show it to doctor, admin and RE.

Pre-conditions:

1) The user must be logged in.
2) The user should have access to schedule appointments.

Course of events:

1) User points towards schedule appointments on menu.
2) User selects calendar menu item.
3) User selects schedule type like:
   - Daily
   - Weekly
   - Monthly
4) User clicks the new event button.
5) User fills the form.
6) User clicks the save button.
7) System validates the data and saves it.

Alternate course of events: User enters invalid data.

1) System displays an error message.
2) User updates the data.
3) User clicks the save button.
4) System validates the data and saves it.

Use case ID: 25
Use case: **View appointments**
Actor: Doctor/RE/Admin
Purpose: to allow user to view encounter appointments.
Cross Reference:
Functional Requirements:

1. System will show the given time lines of patient to doctor, admin and RE.

Pre-conditions:

1) User should be logged in.
2) User should have access to doctor’s scheduler.

3) Appointments must exist before they are viewed.

Course of events:

1) User points towards the schedule appointment menu.
2) User selects the view appointments menu item.
3) System displays a view of all the saved appointments before the user.

Use case ID: 26

Use case: **Add reminder for an appointment**

Actor: RE/Admin/Doctor

Purpose: To remind the scheduled appointment when the deadline/alarm is near or on late response of the patient.

Cross Reference:

Functional Requirements:

1. System will notify on time for the given appointment.
2. System will show all emergency reminders.

Course of events:

1) User points towards schedule appointments on menu.
2) User selects appointment reminders menu item.
3) User provides the information of the scheduled appointment for which he/she wants to add a reminder like Patient ID.
4) User previews the reminder before he saves it.
5) User clicks the save reminder button.
6) System matches for information with the already existing scheduled appointments and saves it.

Pre-condition:

1) User should be logged in.
2) User should have access to doctor's scheduler.
3) Appointment for which reminder is to be set must exist before.

Alternate course of events: User enters an invalid data that does not match the already existing scheduled appointment records.

1) System displays an error message.
2) User enters valid matching (existing scheduled) data.

Repetition of course of event step 4 to 6.

Use case ID: 27

Use case: **View and maintain the follow-up reminders**

Actor: RE/Admin/Doctor

Purpose: To view and edit/update all the saved reminders for the corresponding scheduled appointments.

Cross Reference:

Functional Requirements:

1. System will show the reminder.
2. System will allow editing reminder.
3. System will store the updated reminder.

Pre-condition:

1) User should be logged in.
2) User should have access to doctor’s scheduler.
3) Reminders to be updated must exist before.

Course of events:

1) User points towards the schedule appointments on menu.
2) User selects follow-up reminders menu item.
3) User edits a reminder and updates it.
4) User clicks the preview button in order to preview the updated appointment.
5) User clicks the save changes button.
6) System validates the information and saves it.

Use case ID: 28

Use case: **Add an out-of-office notice**

Actor: Doctor / RE / Admin

Purpose: to aware the stakeholders of the absence of the doctor/staff member(s) for a time period.

Cross Reference:

Functional Requirements:

1. System will disable the staff for any job/task.
2. System will maintain the schedule if he is doctor.
3. System shows it to admin

Pre-conditions:

1) The user should be logged in.

2) The user should have access to doctor’s schedule.

Course of events:

1) User points towards the schedule appointment menu.
2) User selects the out-of-notice menu item.
3) System displays a view of e-mail format containing data like:
    - E-mail subject field
    - E-mail body text area
    - Preview E-mail button
    - Send E-mail button
4) User fills the fields and provides a reason in text for out-of-notice from hospital.
5) User clicks the preview button.
6) System displays a view of data entered by the user in a traditional e-mail format.
7) User clicks the send to all patients button.
8) System e-mails the information to all the patients registered in the hospital.

Use case ID: 29

Use case: **user account**

Actor: Admin

Purpose: To give different privileges of services of system to users.

Cross Reference:

Functional Requirements:

1. System will create a unique user (doctor, recipient) entry.
2. System will assign roles and responsibilities to each entry.
3. System can create multiple entries.
4. System will create its password.

Course of event:

1) User points the cursor to Account maintenance tab.
2) User clicks the manage user accounts option from the menu bar.
3) User fills the user details form with data like:
    - Name
    - Email
    - Designation etc
4) User also fills user account details form with data like:

- User name
- User password

5) User clicks the save button to save the user’s account information.
6) System validates the entered data and saves all the information successfully.

Alternate course of event: User leaves a required field empty or fills a field with invalid data.

1. System displays an error message on invalid or empty left data.
2. User re-enters/updates the invalid data.
3. User clicks on the save button to save the user’s account information.

Use case ID: 30

Use case: **Change Password**

Actor: Admin/RE/Doctor

Purpose: Change password of user in case of forgotten or other security reason.

Cross Reference:

Functional Requirements:

1. System will update user password.

Course of event:

1) User points the cursor to Account maintenance tab.
2) User clicks the change password option from the menu bar.
3) User fills the user details form with data like:
   - User name
   - Old Password
   - New Password
4) User clicks the save button to save the user’s account information.
5) System validates the entered data and saves all the information successfully.

Alternate course of event: User leaves a required field empty or fills a field with invalid data.

6) System displays an error message on invalid or empty left data.
7) User re-enters/updates the invalid data.
8) User clicks on the save button to save the user’s account information.

Use case ID: 31

Use case: **View medication**

Actor: Admin/RE/Doctor

Purpose: view the treatments of patients.

Cross Reference:

Functional Requirements:

a. System will display patient treatments.

Course of event:

i. User points the cursor to Patient Treatment.

Use case ID: 32

Use case: **Add medication**

Actor: Doctor

Purpose: add he treatments of patients.

Cross Reference:

Functional Requirements:

i. System will allow doctors to add patient treatments.

Course of event:

1. User points the cursor to Patient Treatment.
2. User views all previous treatments.
3. User will assign medicines and services to patients.

Use case ID: 33

Use case: **Update medication**

Actor: Doctor

Purpose: Updatethe treatments of patients.

Cross Reference:

Functional Requirements:

1. System will allow doctors to update patient treatments.

Course of event:

2. User points the cursor to Patient Treatment.
3. User views all previous treatments.
4. User will assign medicines and services to patients.

Use case ID: 34

Use case: **Delete medication**

Actor: Doctor

Purpose: delete the treatments of patients.

Cross Reference:

Functional Requirements:

1. System will allow doctors to delete patient treatments.

Course of event:

2. User points the cursor to Patient Treatment.
3. User views all previous treatments.
4. User will delete medicines and services to patients.

Use case ID: 35

Use case: **Capture external clinical documents**

Actor: Doctor/Admin

Purpose: add the clinical documents of patients.

Cross Reference:

Functional Requirements:

1. System will allow users to add patient documents.

Course of event:

2. User points the cursor to Patient Reports.
3. User views all previous documents.

Use case ID: 36

Use case: **Report generation**

Actor: Doctor/Admin/DEO

Purpose: printout clinical reports generation of patients, hospital operations, doctor schedule.

Cross Reference:

Functional Requirements:

1. System will allow users to printout clinical reports generation of patients, hospital operations, doctor schedule.

Course of event:

2. User points the cursor to Reports.
3. User views all reports.
4. User clicks printout button for task.

TESTING AND RESULTS *Chapter 05*

In this chapter we will test and display results of all the use cases of SaaS (Software as a Service) Based Practice Management System.

5.1 Unit Testing:

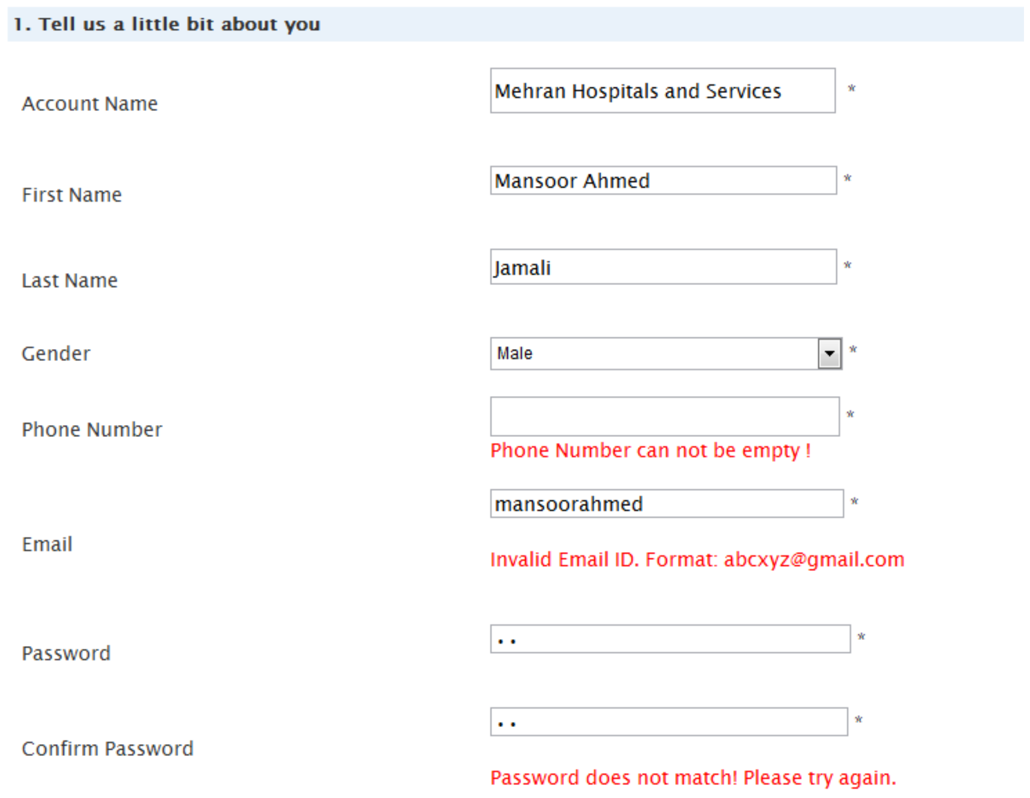
In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming a unit could be an entire module but is more commonly an individual function or procedure. In object-oriented programming a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are created by programmers.

Since, it's a SaaS Cloud based Practice Management System, it has so many units and modules/components to be tested on. The system's every unit has been tested. Following are the details of each unit and module with snapshots:

5.1.1 Practice Registration:

System generates a failure to practice registration on following attempts:

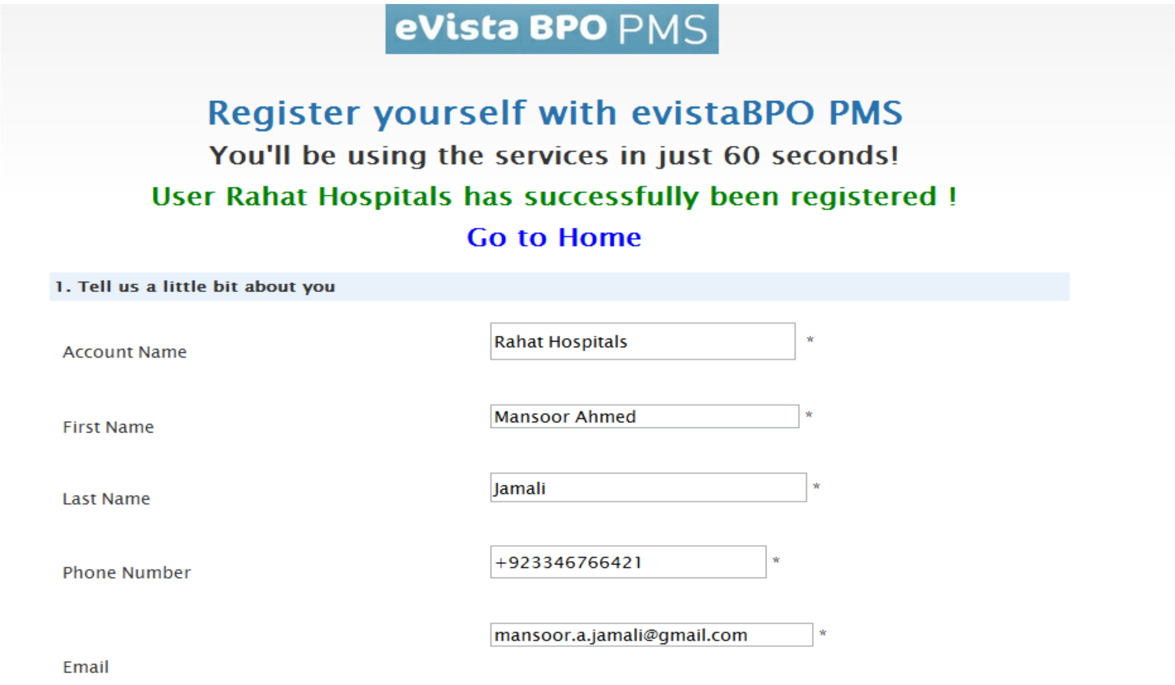
- User leaves the required fields blank and submits "Create Account" button.
- User enters the email address using an invalid format.
- User enters an email address already registered in the database.
- Password fields do not match.
- Invalid format of practice URL.
- Practice URL is already registered or is not available.
- Practice account name is already registered in the database or is not available.
- User does not select the checkbox for the company (eVista BPO) terms and policies to be agreed upon.



1. Tell us a little bit about you

Account Name	Mehran Hospitals and Services	*
First Name	Mansoor Ahmed	*
Last Name	Jamali	*
Gender	Male	*
Phone Number	 Phone Number can not be empty !	*
Email	mansoorahmed Invalid Email ID. Format: abcxyz@gmail.com	*
Password	••	*
Confirm Password	•• Password does not match! Please try again.	*

System registers a practice and stores the information entered in the database when all the attempts above are successful.



eVista BPO PMS

Register yourself with evistaBPO PMS

You'll be using the services in just 60 seconds!

User Rahat Hospitals has successfully been registered !

Go to Home

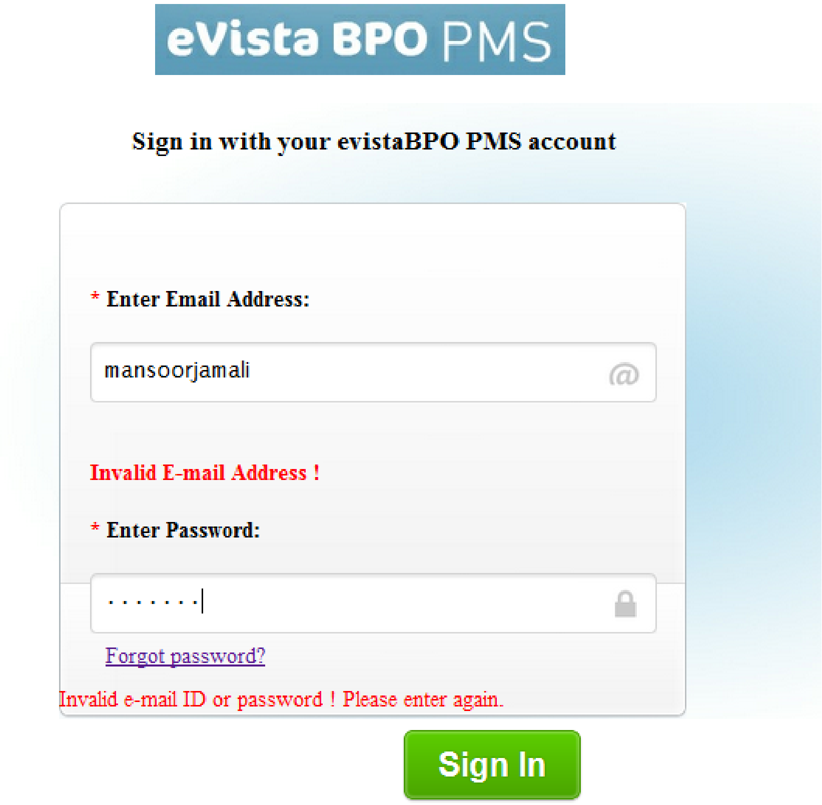
1. Tell us a little bit about you

Account Name	Rahat Hospitals	*
First Name	Mansoor Ahmed	*
Last Name	Jamali	*
Phone Number	+923346766421	*
Email	mansoor.a.jamali@gmail.com	*

5.1.2 User Login:

System does not allow user to log in and access services afterwards on the failure of following attempts:

- User enters an email of invalid format
- User enters wrong password.



* Enter Email Address: mansoorjamali

Invalid E-mail Address !

* Enter Password: ·······

Invalid e-mail ID or password ! Please enter again.

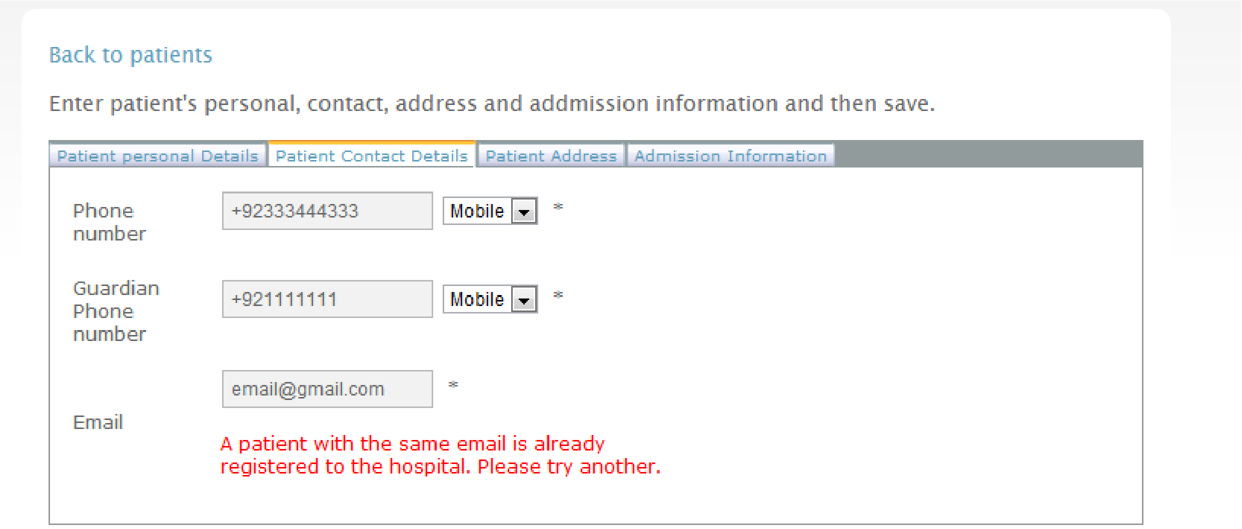
On successful attempt of entering the correct email address and password, system takes the user to the Dashboard page from where the user is enable to access practice services.

Adding a new patient (Inpatient as well as Outpatient) to the practice:

On the failure of following attempts, system does not allow user to add a new patient and pops up error message accordingly:

- User leaves a required field empty.
- User does not select patient gender from Radio Button List.
- User enters the patient email already stored/registered in the practice.
- User enters the referred doctor ID and name for the patient who is not already registered to the practice.
- User enters the assigned doctor ID and name for the patient who is not already registered to the practice.
- User enters the department name which is not already available in the practice.

- User allocates the bed and ward numbers for the patient (in case of Inpatient), which are not already defined in the practice.



Back to patients

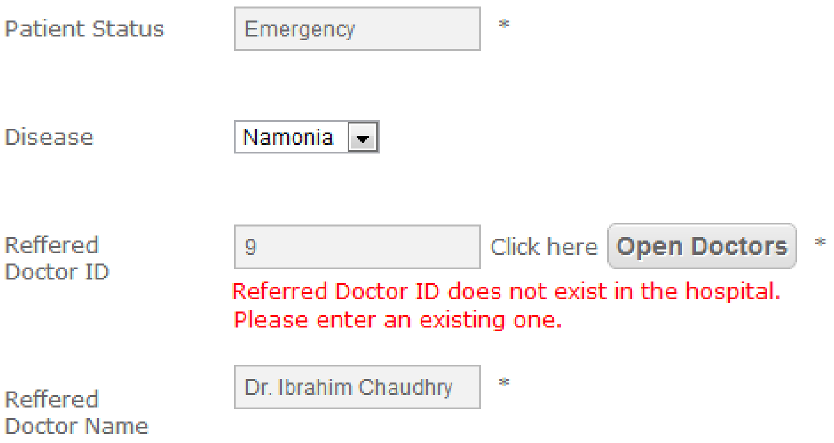
Enter patient's personal, contact, address and addmission information and then save.

Phone number: +92333444333 Mobile *

Guardian Phone number: +921111111 Mobile *

Email: email@gmail.com *

A patient with the same email is already registered to the hospital. Please try another.



Patient Status: Emergency *

Disease: Namonia

Reffered Doctor ID: 9 Click here Open Doctors *

Referred Doctor ID does not exist in the hospital. Please enter an existing one.

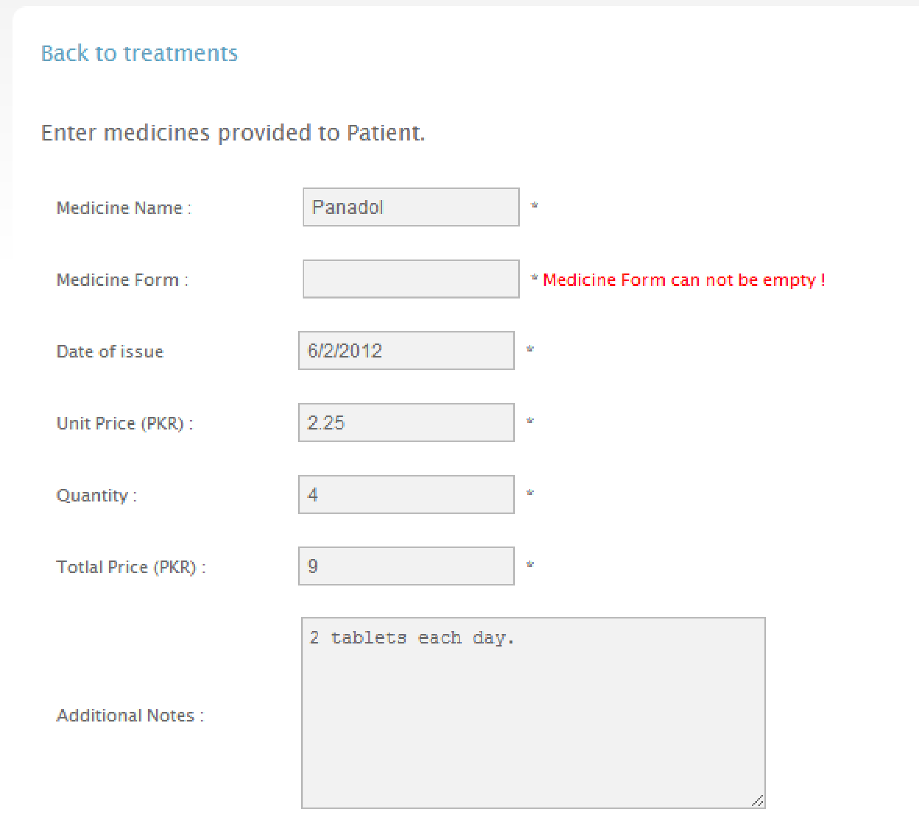
Reffered Doctor Name: Dr. Ibrahim Chaudhry *

System generates a success message on the successful attempts of the above and adds the patient to the practice.

Add medical treatments for the patients:

On failure of the following attempts, system does not allow user to add new medicine information for the particular patient:

- User leaves any required field empty.



Back to treatments

Enter medicines provided to Patient.

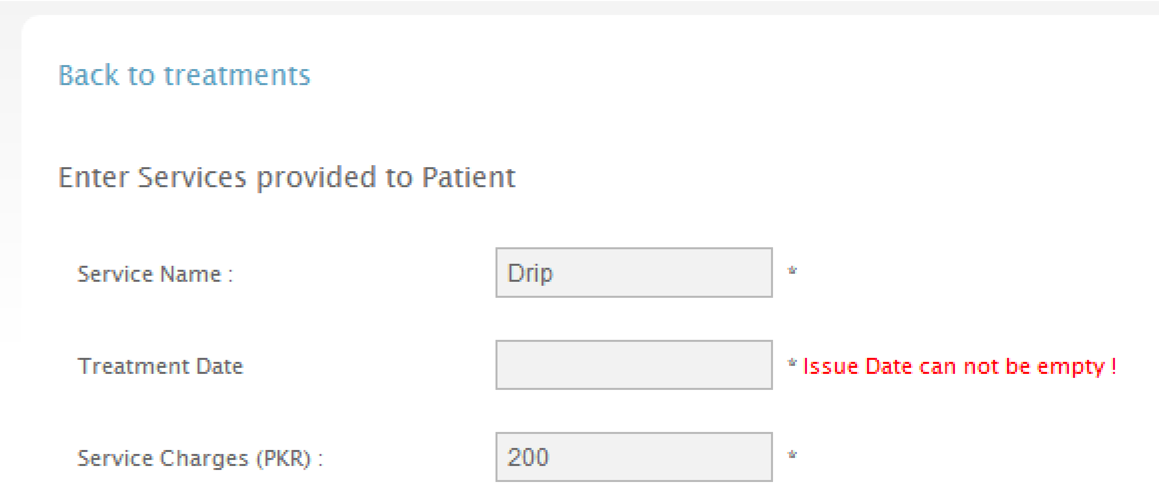
Medicine Name :	Panadol	*
Medicine Form :		* Medicine Form can not be empty !
Date of issue	6/2/2012	*
Unit Price (PKR) :	2.25	*
Quantity :	4	*
Totlal Price (PKR) :	9	*
Additional Notes :	2 tablets each day.	

On success of the above attempts, system adds new medicine for the patient in the practice.

Add service treatments for the patients:

On failure of the following attempts, system does not allow user to add new service treatment for the particular patient of the practice:

- User leaves any required field empty.



Back to treatments

Enter Services provided to Patient

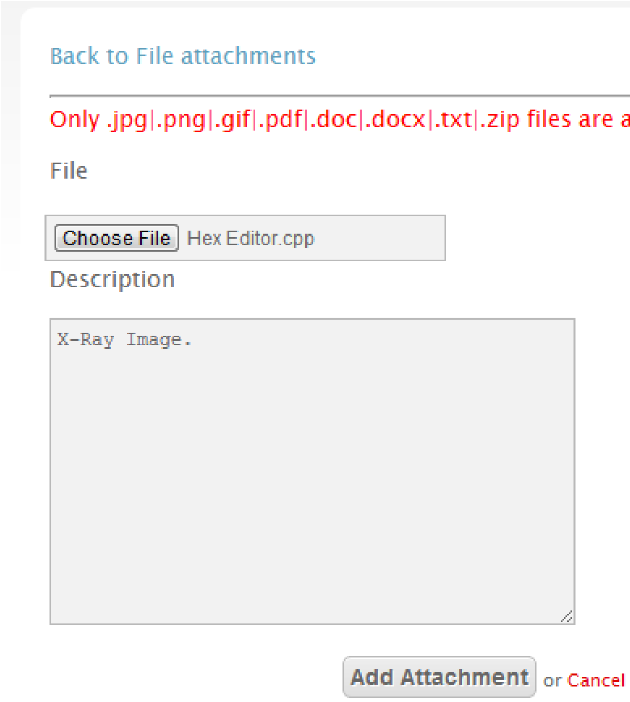
Service Name :	Drip	*
Treatment Date		* Issue Date can not be empty !
Service Charges (PKR) :	200	*

On success of the above attempts, system adds new service treatment for the patient in the practice.

Add a new file for the patient:

System pops an error message and does not allow adding a new file for the patient due to following failing attempts:

- User selects a file of format that is not allowed to add for the patient in the practice.



Only .jpg|.png|.gif|.pdf|.doc|.docx|.txt|.zip files are allowed!

File

Hex Editor.cpp

Description

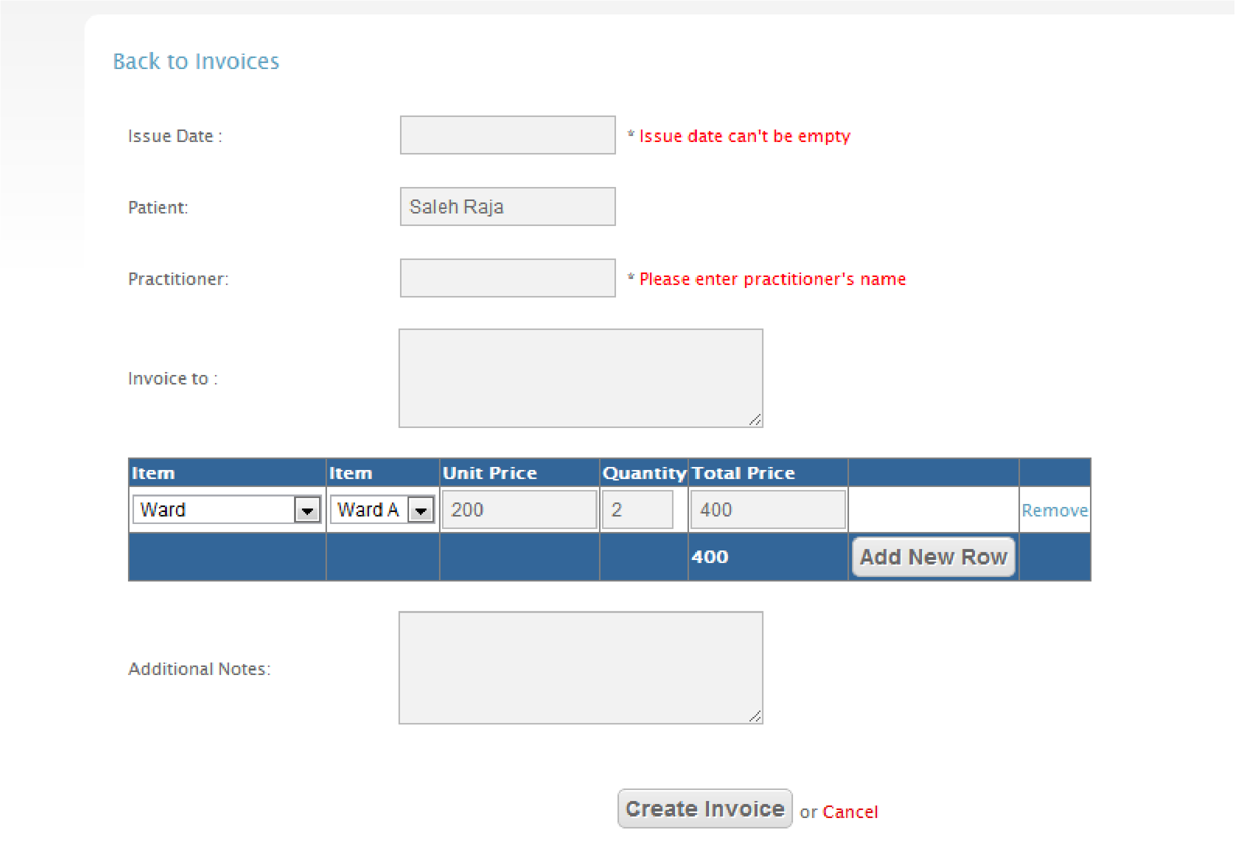
X-Ray Image.

Else system adds the selected file for the patient of defined and allowed formats.

Add patient appointments with doctors, add invoices and payments:

System does not allow adding a new appointment for the patient with the doctor, adding new invoice and payment for the patient due to the following attempts:

- User leaves the required fields empty.



Issue Date : * Issue date can't be empty

Patient: Saleh Raja

Practitioner: * Please enter practitioner's name

Invoice to :

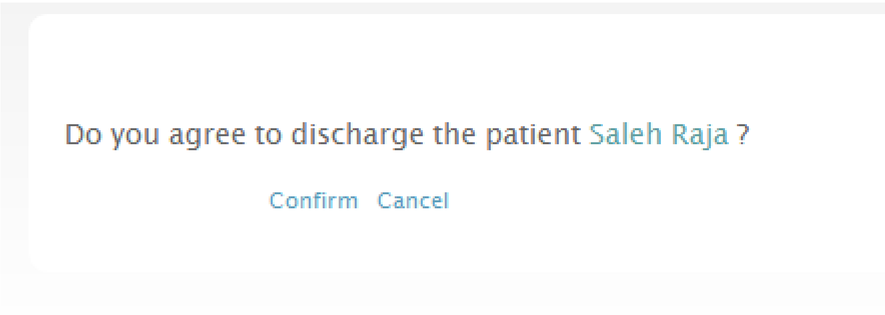
Item	Item	Unit Price	Quantity	Total Price		
Ward	Ward A	200	2	400		Remove
				400	Add New Row	

Additional Notes:

Discharge a patient:

Patient gets discharged when:

- User clicks the confirm link.

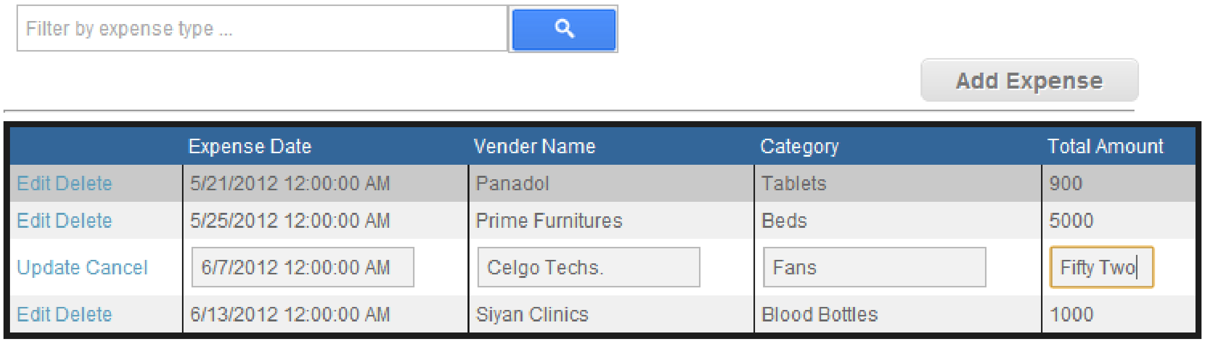


Do you agree to discharge the patient Saleh Raja ?

Add Practice Expenses:

System shows a failure message on the following attempts:

- User tries to update the numeric value of “Total Amount” of expense with alphabetic value.



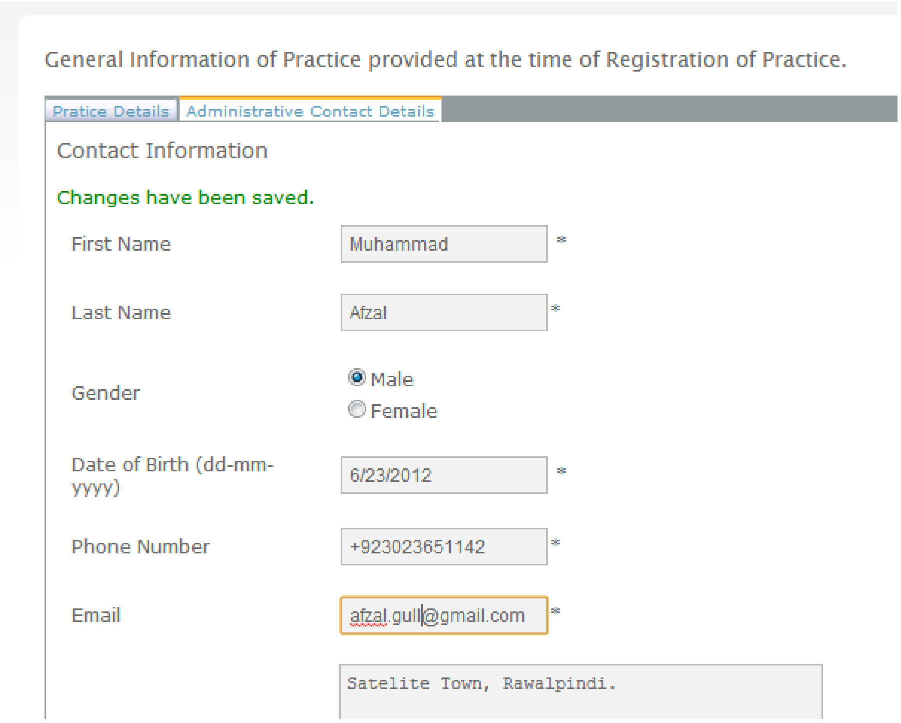
	Expense Date	Vender Name	Category	Total Amount
Edit Delete	5/21/2012 12:00:00 AM	Panadol	Tablets	900
Edit Delete	5/25/2012 12:00:00 AM	Prime Furnitures	Beds	5000
Update Cancel	6/7/2012 12:00:00 AM	Celgo Techs.	Fans	Fifty Two
Edit Delete	6/13/2012 12:00:00 AM	Siyan Clinics	Blood Bottles	1000

Updating practice's profile information:

System fails to save the practice's profile information if following attempts are to be made:

- Any required field is left blank.
- Email ID for the practice's admin is changed to one which is already registered in the database.

Else system saves the changed information of the practice in the database.



General Information of Practice provided at the time of Registration of Practice.

Contact Information

Changes have been saved.

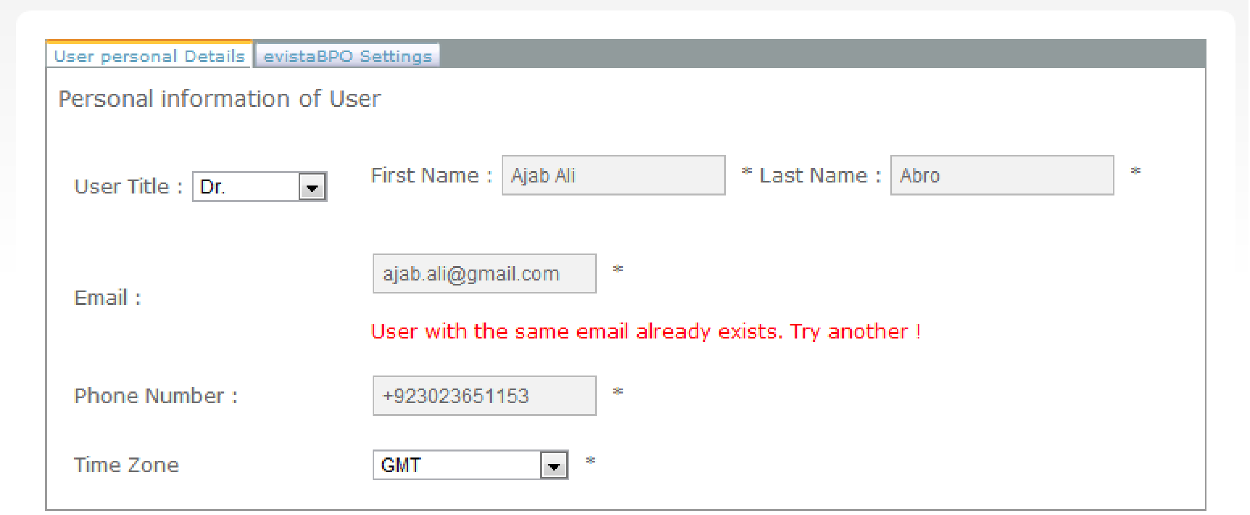
First Name	Muhammad *
Last Name	Afzal *
Gender	● Male ○ Female
Date of Birth (dd-mm-yyyy)	6/23/2012 *
Phone Number	+923023651142 *
Email	afzal.gull@gmail.com *
	Satelite Town, Rawalpindi.

Adding a new user (role) to the practice:

System fails to add a new user to the practice if following attempts are made:

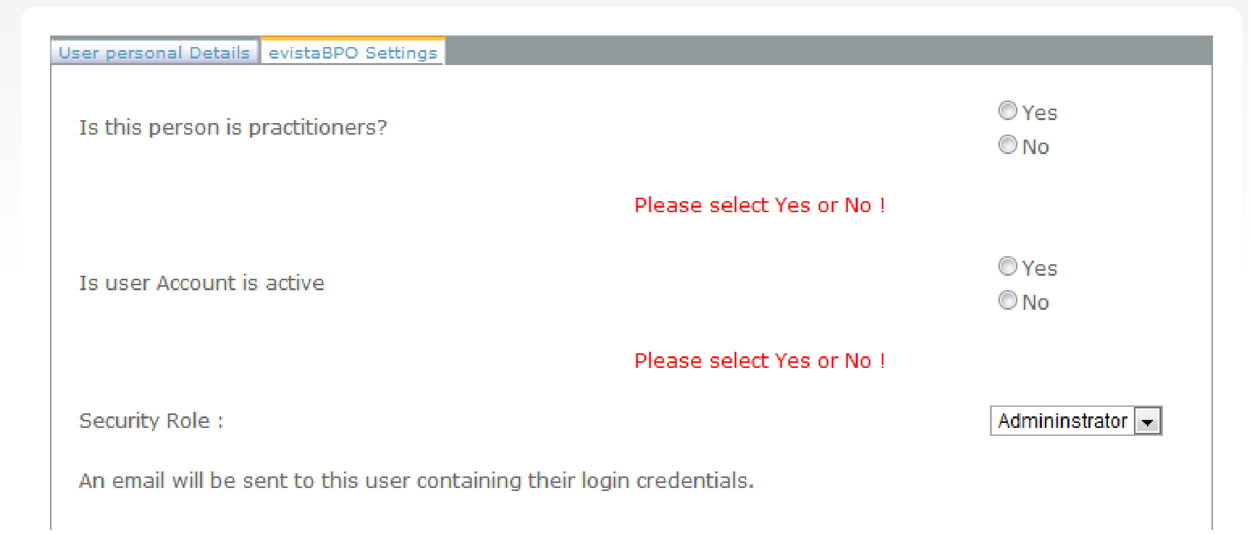
- User leaves any required field empty.

- User enters an email address which is already a user (or registered) to the practice.
- User does not select the Radio Button for whether if the role to be added is practitioner or not.
- User does not select the Radio Button for whether if the role to be added is active or not.



Personal information of User

User Title :	Dr.	First Name :	Ajab Ali	* Last Name :	Abro *
Email :	ajab.ali@gmail.com *				
	User with the same email already exists. Try another !				
Phone Number :	+923023651153 *				
Time Zone	GMT *				



Is this person is practitioners? ○ Yes ○ No

Please select Yes or No !

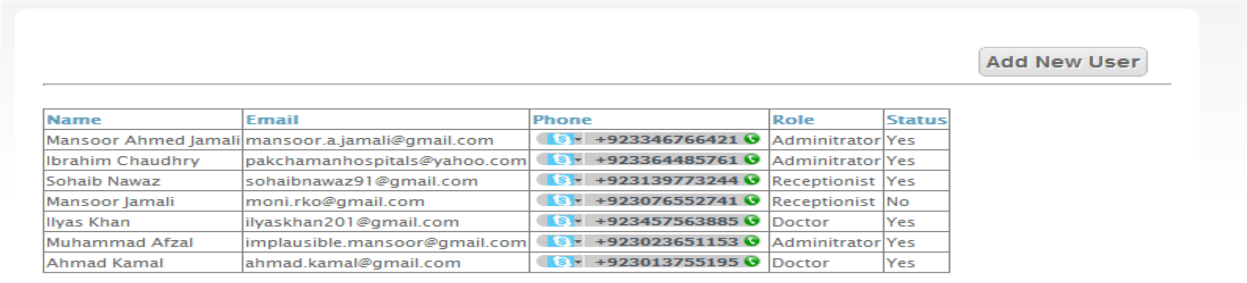
Is user Account is active ○ Yes ○ No

Please select Yes or No !

Security Role : Admininstrator

An email will be sent to this user containing their login credentials.

Else system adds the new user (role) to the practice.



Name	Email	Phone	Role	Status
Mansoor Ahmed Jamali	mansoor.a.jamali@gmail.com	+923346766421	Adminitrator	Yes
Ibrahim Chaudhry	pakchamanhospitals@yahoo.com	+923364485761	Adminitrator	Yes
Sohaib Nawaz	sohaibnawaz91@gmail.com	+923139773244	Receptionist	Yes
Mansoor Jamali	moni.rko@gmail.com	+923076552741	Receptionist	No
Ilyas Khan	ilyaskhan201@gmail.com	+923457563885	Doctor	Yes
Muhammad Afzal	implausible.mansoor@gmail.com	+923023651153	Adminitrator	Yes
Ahmad Kamal	ahmad.kamal@gmail.com	+923013755195	Doctor	Yes

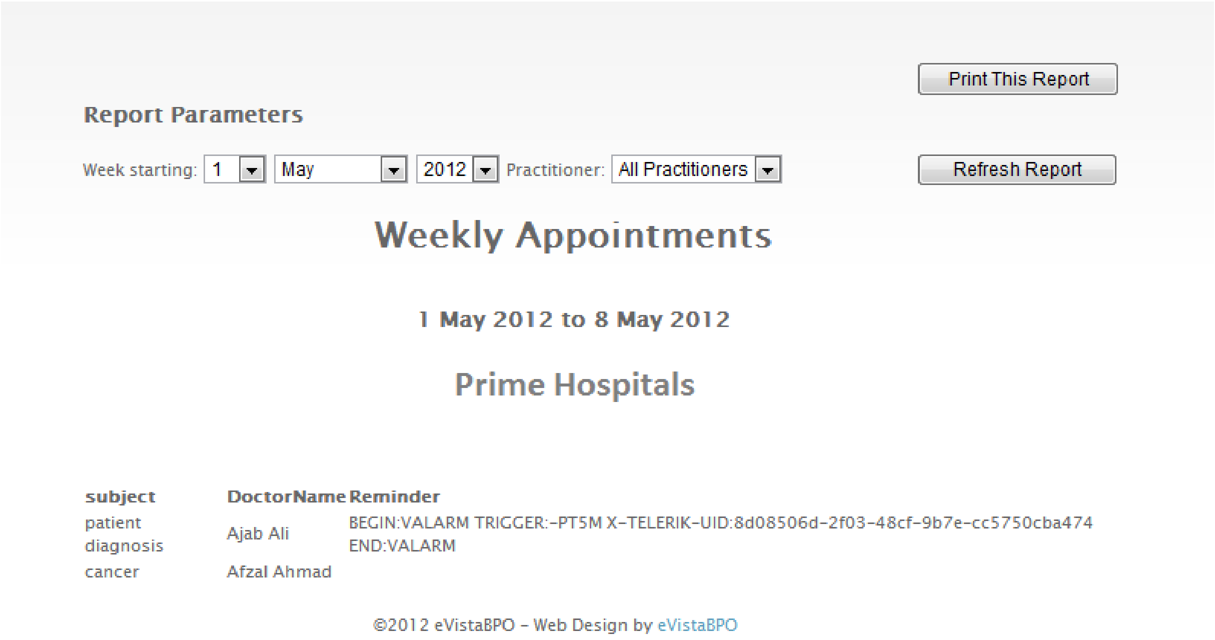
Reports:

System does not show any reports if:

- User selects a date in which no any schedules have been taken.

System lists the reports of the practice in case:

- User selects the correct date parameter in which any schedules have been taken.



Report Parameters

Week starting: 1 May 2012 Practitioner: All Practitioners

Weekly Appointments

1 May 2012 to 8 May 2012

Prime Hospitals

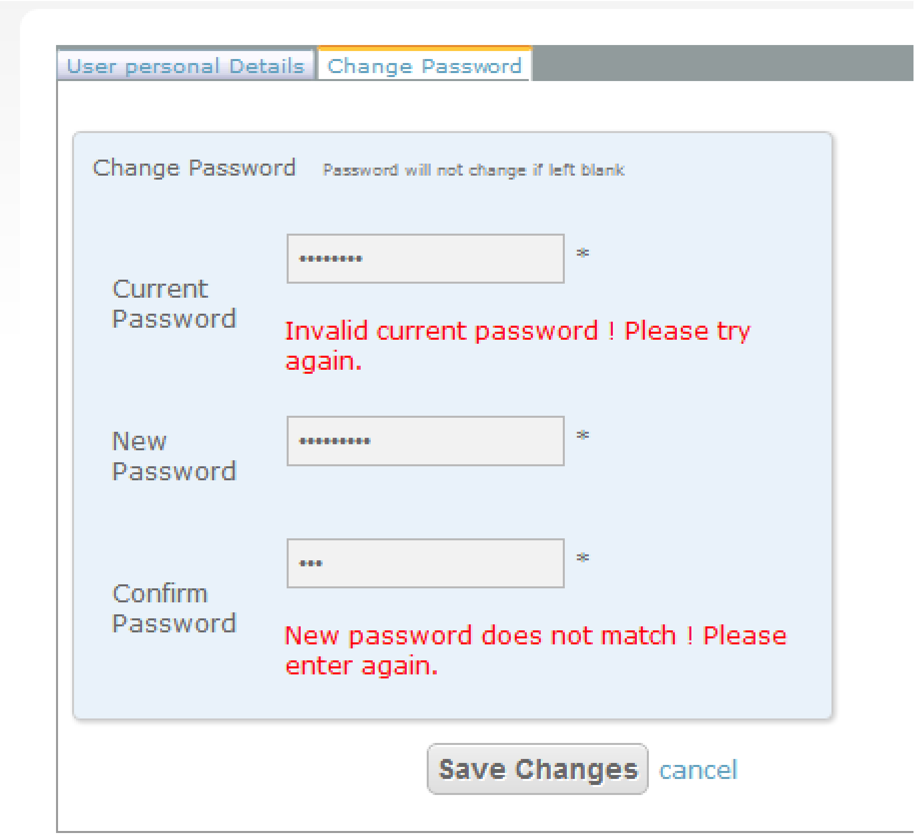
subject	DoctorName	Reminder
patient diagnosis	Ajab Ali	BEGIN:VALARM TRIGGER:-PT5M X-TELERIK-UID:8d08506d-2f03-48cf-9b7e-cc5750cba474 END:VALARM
cancer	Afzal Ahmad	

©2012 eVistaBPO - Web Design by eVistaBPO

Change current user’s password:

System will generate error message and will not modify user’s password if:

- Any field of either current password or the new password is left blank.
- User enters current password incorrectly.
- Value in both the fields of new password does not match.



Change Password Password will not change if left blank

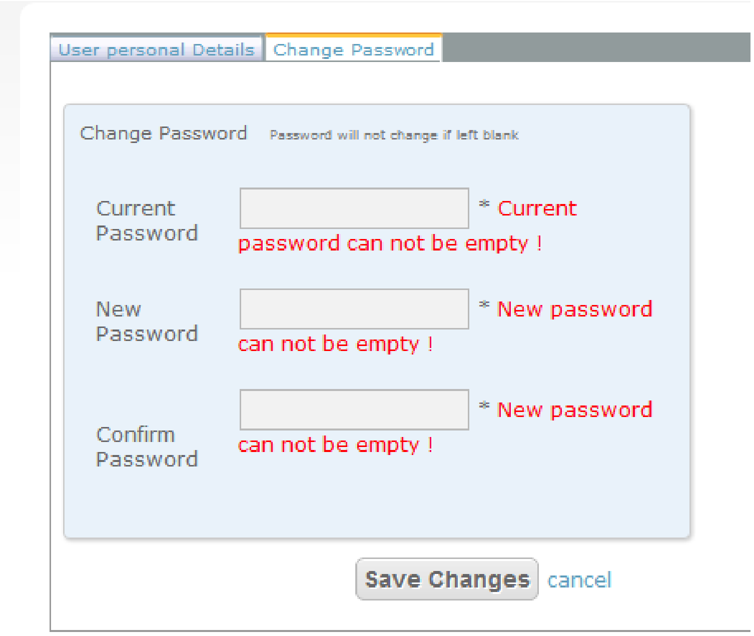
Current Password •••••••• *

Invalid current password ! Please try again.

New Password ••••••••• *

Confirm Password ••• *

New password does not match ! Please enter again.



Change Password Password will not change if left blank

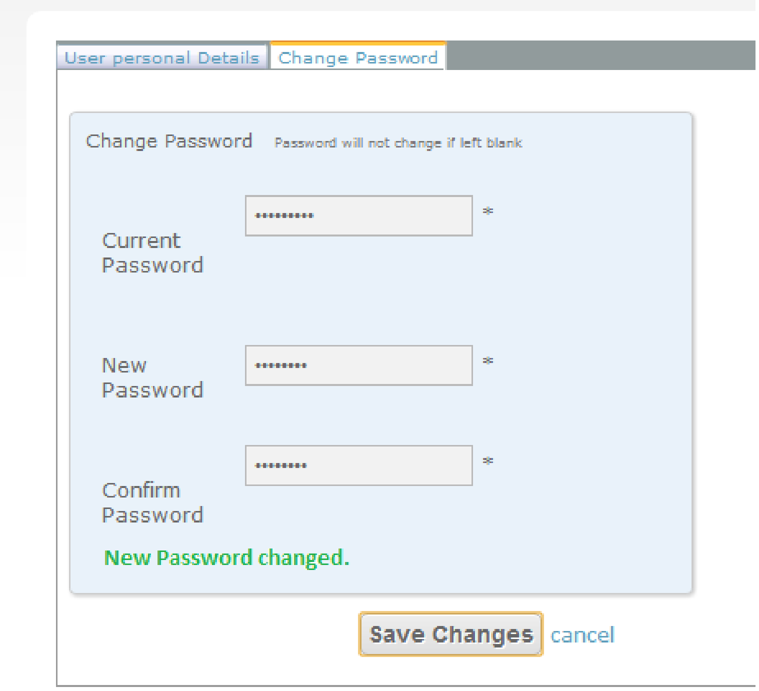
Current Password * Current password can not be empty !

New Password * New password can not be empty !

Confirm Password * New password can not be empty !

System will modify the current password to new password of the user logged in if:

- Current password is entered correctly.
- Both the fields of the new password match.



Change Password Password will not change if left blank

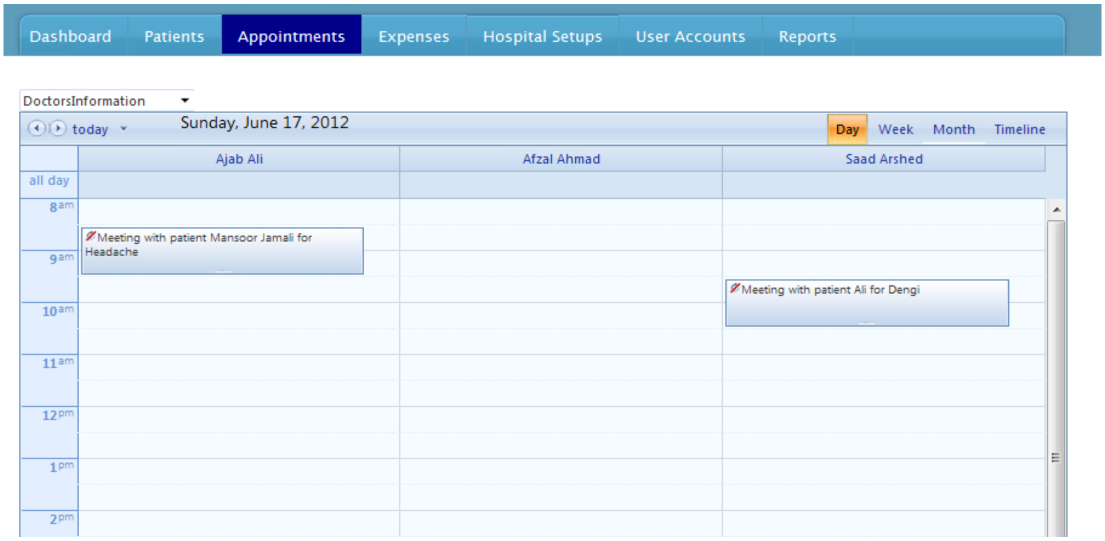
Current Password ********* *

New Password ******** *

Confirm Password ******** *

New Password changed.

- **Scheduler Module Test:** In this module test system is tested according to different scenarios as given bellow.

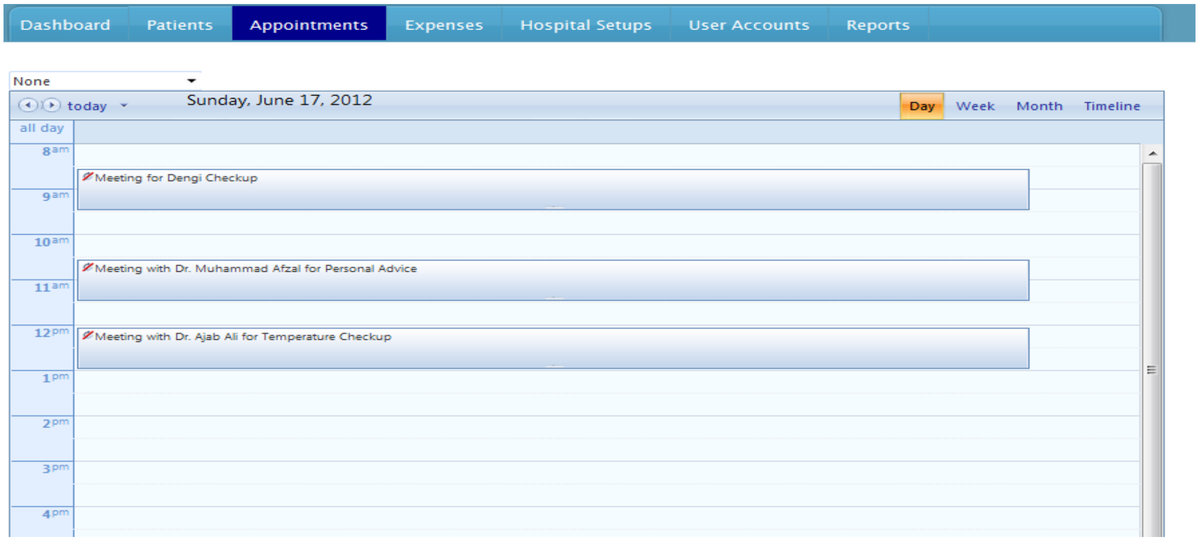


Sunday, June 17, 2012

Ajab Ali	Afzal Ahmad	Saad Arshed
Meeting with patient Mansoor Jamali for Headache		Meeting with patient Ali for Dengi

View of scheduler at first time which is according to doctor’s resource.

View of scheduler which is according to particular day.



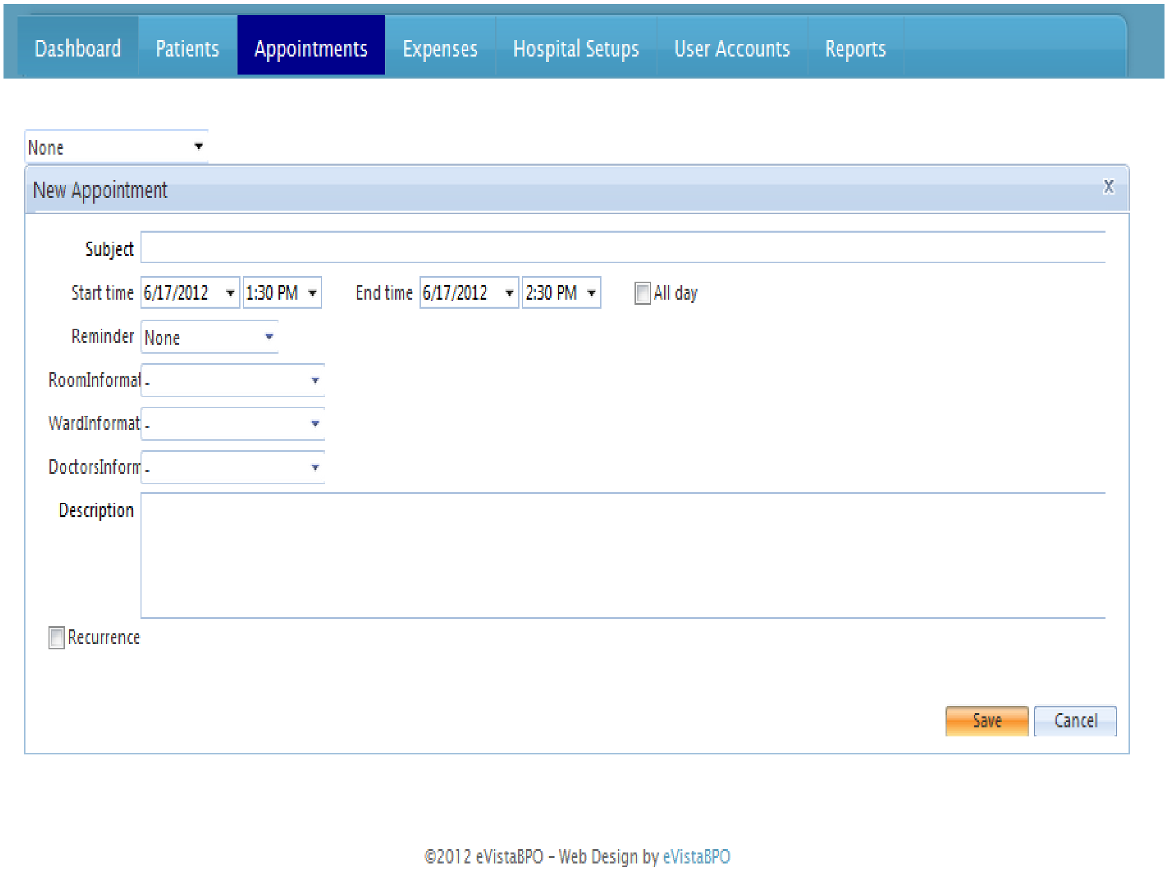
Sunday, June 17, 2012

Meeting for Dengi Checkup

Meeting with Dr. Muhammad Afzal for Personal Advice

Meeting with Dr. Ajab Ali for Temperature Checkup

View of scheduler assigning/setting scheduler with respect to rooms, wards, doctors, recurrence, day and time.



New Appointment

Subject

Start time 6/17/2012 1:30 PM End time 6/17/2012 2:30 PM □ All day

Reminder None

RoomInformat -

WardInformat -

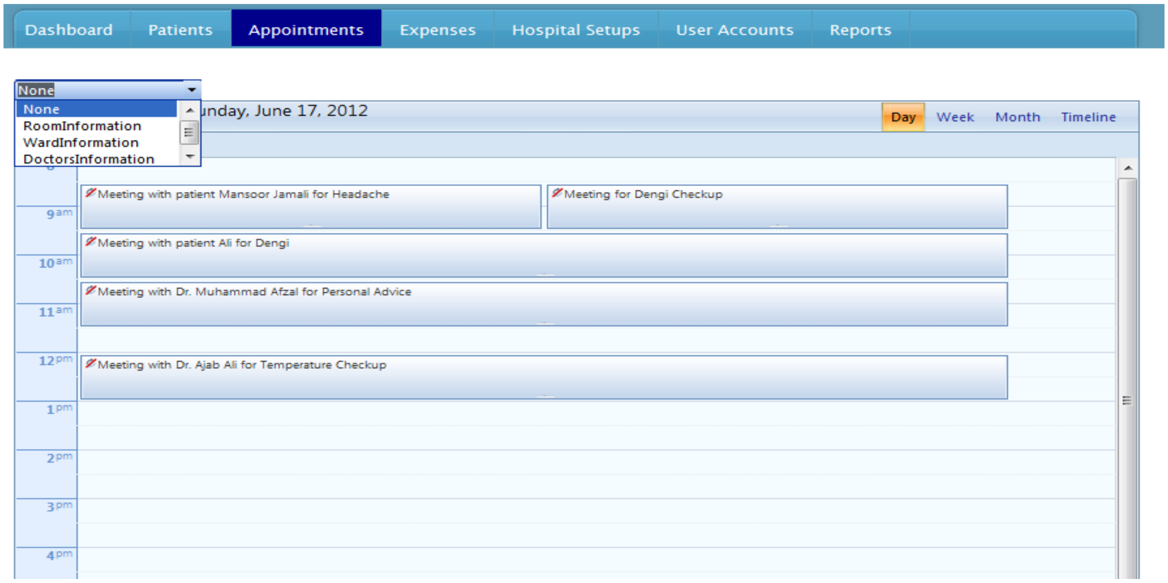
DoctorsInform -

Description

□ Recurrence

©2012 eVistaBPO - Web Design by eVistaBPO

Viewing scheduler with respect to doctor, room and ward.



unday, June 17, 2012

Meeting with patient Mansoor Jamali for Headache

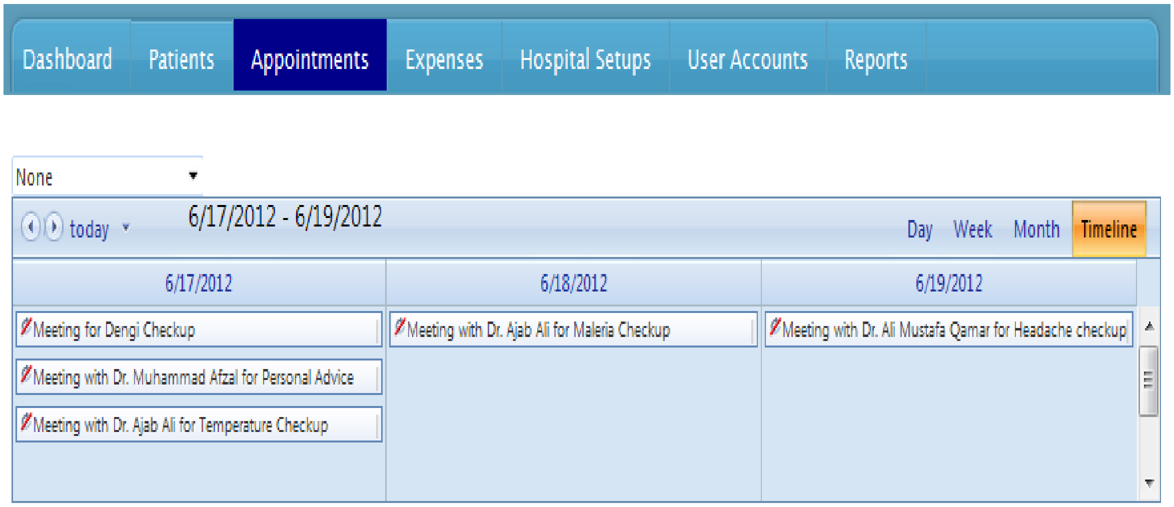
Meeting for Dengi Checkup

Meeting with patient Ali for Dengi

Meeting with Dr. Muhammad Afzal for Personal Advice

Meeting with Dr. Ajab Ali for Temperature Checkup

View of scheduler timeline which shows the whole scheduler in single row/line.



6/17/2012 - 6/19/2012

6/17/2012	6/18/2012	6/19/2012
Meeting for Dengi Checkup	Meeting with Dr. Ajab Ali for Maleria Checkup	Meeting with Dr. Ali Mustafa Qamar for Headache checkup
Meeting with Dr. Muhammad Afzal for Personal Advice		
Meeting with Dr. Ajab Ali for Temperature Checkup		

COCLUSION *Chapter 06*

In this chapter we will conclude the SaaS (Software as a Service) Based Practice Management System.

6.1 Conclusion:

Hospitals continue to face issue with maintaining and keeping patient records while managing daily operations of the hospital. On-premise Hospital management solutions excessive total cost of ownership (TCO) requires investments beyond the scope of many hospitals.

Many hospitals end up using feature poor, non-compliant, difficult to use solutions, difficult to allocate resources (human as well as material resources) and sacrifices quality in the way they support, service and interact with their customer. The purpose of our hosted solution to provide services on-demand to hospitals and make advanced hospital management functionality available, affordable, useable and easily deployable/configurable for hospitals.

Our cloud based solution fulfills maximum hospital standards like HL7 and HIPPA and provides all necessary operational features to run a practice such as maintaining medical records of indoor and outdoor patient's data, scheduling of patients' appointments with doctors, availability of doctors, maintaining resources (human as well as material) information as well as allocating resources efficiently such as assigning doctor to patients, admitting patients into wards or rooms, solution also covers billing module such as generating invoices against services or resources used by patient in a particular practice and payments are generated against each invoice, our solution has also capability to generate and print practice reports. Solution has also encryption module for information security and there are three roles (Administrator, practitioner, and receptionist) for accessing different features of solution according to their roles and responsibilities.

RECOMMENDATIONS *Chapter 07*

In this chapter we will discuss the objectives we have achieved and recommend the new features in this system.

Recommendation:

In this system proposed requirements are completed but yet there are many requirements which have in this system. Some of those are given bellow.

1. Forward Integration with Pharmacy: In this system will be capable to integrate it to the pharmacy provider to help them in prescription and statistics to sort out areas for medical operations.

2. Appointments through Smart Applications: Since this is the period of smart phones. So this system should have capability to integrate with smart applications to set appointments their selves when the slot is free with doctors.

3. Administration Module: In this module our system (clients/hospitals) should be monitored and administered through the service provider end to assure the security, storage, policies, and privacy for the system.

4. e-commerce integration: In this module system should have capability to make and accept transaction through smart credit cards and e-transactions.

There can be other recommendation but we point out these recommendation to make complete product to use which can save total cost of ownership.

References:

1. http://msdn.microsoft.com/en-us/library/gg696172%28v=vs.103%29.aspx (ADO.NET)
2. http://www.freshlogics.com/hospital_management_system.php (Freshlogics' Solution)
3. http://www.medinous.com/ (MediNous HMS)
4. http://www.netsoltech.com/products/hospital-management-system/ (NetSol's Solution)
5. http://acgil.com/products/hospital_management_system_001.htm (HMIS - An Electronic Management)
6. http://www.emdfix.com/product_guide.htm (eMDfix EMR)
7. http://www.revenuexl.com
8. https://windows.azure.com/ (Windows Azure official website)
9. Semantic Based EMR By NUST-SEECS Students (Thesis)
10. HMS Research Paper