NANYANG TECHNOLOGICAL UNIVERSITY

SCHOOL OF COMPUTER ENGINEERING

WEB APPLICATION FOR MOBILE INPATIENT INFORMATION SYSTEM

Shee Bo Chun

School of Computer Engineering

2013

NANYANG TECHNOLOGICAL UNIVERSITY

SCHOOL OF COMPUTER ENGINEERING

SCE12-0193

WEB APPLICATION FOR MOBILE INPATIENT INFORMATION SYSTEM

Author: Shee Bo Chun (U1021351D)

Supervisor: Assoc Prof. Lin Feng (Dr.)

Submitted in Partial Fulfilment of the Requirements for the Degree of Bachelor of Computer Engineering of Nanyang Technological University

School of Computer Engineering

2013

ABSTRACT

With the advancement of IT technology in recent years, the concept of Electronic Medical Record (EMR) was introduced to the health care industry and incorporated to the health care services to provide better and efficient care to the patients. Referencing the concept of EMR, this project is modeled by the Software Development Life Cycle (SDLC) to design and develop the Inpatient Information System (IPIS). IPIS is a web application for health care professionals to access patient health information during ward registration, discharge planning, inpatient procedures and prescriptions while using hand-held devices or workstations to make better decisions and thus, increases efficiency in providing better health care services to the patients.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Professor Lin Feng for his patient guidance, encouragement, advice, and assistance to this project.

I also would like to thank Nanyang Technological University (NTU) for giving me a chance to explore more on this project. It has truly helped me gain a lot of knowledge.

Last but not least, this project would never have been completed without the support and encouragement of my family and friends.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

1.1 Background

Traditionally, health care professionals and organizations have been using paper written format to record the individual patient's health information medical records. The medical records are usually filed and maintained in folders and are kept in the hospital site for a period of time. However, with the advancement of information technology in recent years, it has played an important role in the increasing efficiency of recording and retrieval of data and thus, improving and influencing the practices of health care professionals and organizations such as the ease of documenting patient care through hand-held devices or workstations.

One of the most popular advancement in health care technology is the Electronic Medical Record (EMR). EMR is a software system that provides an electronic-based mean to record all the patient health information securely without relying on paper records. It helps integrate and streamline the health care processes, providing up-to-date and accurate information to be accessed by authorized health care professionals. This greatly increases the overall efficiency of documenting and retrieving of patient's information which in turn, saves time and increases the quality of care-giving to the patients. It also reduces costs attributed by physical storage of patient medical record files, and most importantly, protects the patient's safety as it reduces medical errors caused by illegible handwritings or negligence.

The trend of using EMR in context of Singapore is definitely on the rise in the recent years. From the year 1998 onwards, Singapore's corporatized hospitals have been at various stages of computerizing their medical records, but working independently of each other and using various platforms to manage their medical records. Exchanging patients' medical records from different hospitals was a hassle due to the different standards of computerized medical records practiced in different hospitals and there wasn't a common platform in place then to record and retrieve all the patients' medical records regardless of the different hospitals visited by the patients. On 1 April 2004, Ministry of Health has launched and implemented the Electronic Medical Record Exchange (EMRX), a computerized platform to allow the sharing of electronic medical records across all public hospitals and polyclinics in Singapore. [1]

Last year, four hospitals in Singapore were awarded the Stage 6 EMR Adoption Model (EMRAM) by US-based HIMSS (Healthcare Information and Management Systems Society) Analytics for their use of the EMR systems. [2] The highest attainable level is at Stage 7 where the hospital provides a complete paperless EMR environment.

This project introduces the Inpatient Information System (IPIS) which offers nurses and doctors a platform to record and retrieve patient's medical records using electronic means. By providing a simple and easy to use system, it aims to encourage more health care professionals to adopt the usage of EMR in their daily workflows. Chapter 3 will give an overview of the proposed system.

1.2 Objective

The main objective of having Inpatient Information System (IPIS) is to provide a simple-to-use and efficient system which allows quick and easy access to the patients' medical records to aid the nurses and doctors' daily workflows thus, enhancing the quality of care-giving to the warded patients.

1.3 Scope

To create a web application for Inpatient Information System (IPIS) to allow administrators, nurses and doctors to use IPIS on both handheld devices and workstations to record and retrieve the necessary patients' records during ward registration, discharge planning, inpatient procedures and prescriptions.

1.4 Project Plan and Schedule

There are two planned schedules that are conceived during the designing and developing phases of the project respectively. Both the initial planned and the actual planned schedules are then divided into 3 major segments namely the research phase, development phase and the report phase.

The initial planned schedule, as can be seen in Figure 1 below, describes the overall project development guideline in an ideal and straightforward manner while missing out on the impact of certain factors would bring to alter the schedule such as school assignments, examinations, searching for better methods or solutions, troubleshooting etc.

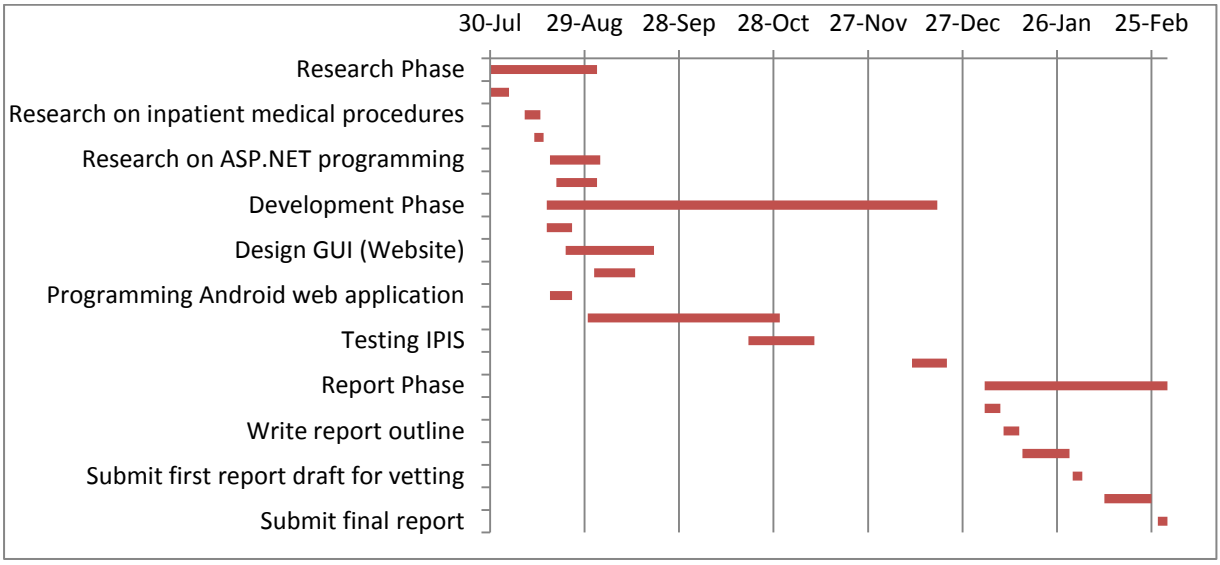

Figure 1: Initial Planned Schedule

On the other hand, the actual planned schedule, as shown in Figure 2 below, incorporated those elements that were not considered in the initial planned schedule.

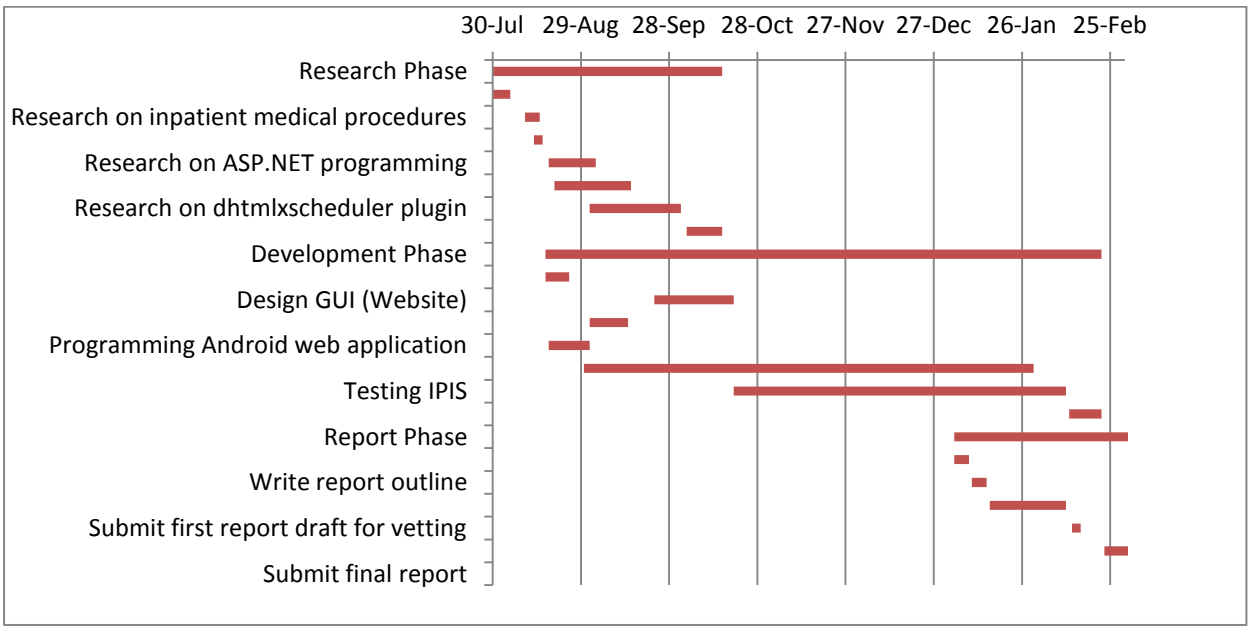

Figure 2: Actual Planned Schedule

1.5 Report Structure

This report is segmented into different chapters. There are a total of 8 chapters which comprise of the introduction, review on previous work and technology, an overview of the proposed system, system design, system implementation, results and discussion and last but not least, the conclusion.

Chapter 1 describes the background, objectives and the scope of the project and it also includes a planned and an actual timeline of the project schedule which display the durations of the different project stages.

Chapter 2 illustrates a review of previous works that are similar to the proposed system and also shows why these previous existing systems are implemented successfully or unsuccessfully.

Chapter 3 describes the overview of the features and functionalities of the project while providing insights on the interactions of the different user types with the system.

Chapter 4 specifies the model of the Software Development Life Cycle (SDLC) that the project has adopted and provides further elaboration on the activities that occur during the different stages of the SDLC.

Chapter 5 specifies the development tools that are used for the implementation of the project followed by the illustration of the modules in the project.

Chapter 6 describes the representation of the data structure through the Entity-Relationship (ER) diagram and the database system design of the project and illustrates how the data relates to one another.

Chapter 7 gives an analysis and evaluation of the proposed system after the development process. It shows how the system will operate. This chapter also includes problems encountered, lessons learnt, and future enhancement to the system.

Chapter 8 gives the overall conclusion of the project from planning to implementation.

CHAPTER 2 REVIEW OF PREVIOUS WORK & TECHNOLOGY

This chapter reviews the existing information systems currently used in the health care industry. From my research, I had come across many successful implementations of Electronic Medical Record (EMR) systems as well as unsuccessful EMR systems. This chapter discusses why some EMR systems are successful and why some EMR systems failed to implement successfully.

eClinicalWorks is a renowned medical software vendor who has developed an award winning and comprehensive EMR system. eClinicalWorks was ranked as the most popular EMR software solution with an established customer base of more than 220,000 health care organizations and 410,000 healthcare professionals. [3] Numerous health care organizations have been using the system and have positive responses about it. The following two paragraphs will describe the two health care organizations who had decided to deploy eClinicalWorks comprehensive EMR solution to all its hospitals and clinics and how it has helped change their operations.

Firstly, Unity Health Care is Washington, DC's largest non-profit health and social services organization who has partnered with eClinicalWorks to deploy an EMR system. After the implementation, Unity Health Care was able to restructure their process for handling patient appointment scheduling which had led to an increase in the number of patients in receiving treatment. The usage of ERM also helps Unity Health Care to better meet and serve the needs of the communities through improved patient medical reporting and retrieving and synchronization over 31 locations across the District of Columbia. [4] Since implementing eClinicalWorks, Unity Health Care has experienced a 21% increase in the overall productivity and realized US$12.2 million dollars in additional revenue which is a 106% in return of investment. Unity Health Care had also received a 2012 HIMSS (Healthcare Information and Management Systems Society) Community Health Organization Davies Award, which honors excellence in implementing and use of EMR for health care organizations. [5]

Secondly, Open Door Family Medical Center is a Federally Qualified Health Center that has deployed eClinicalWorks' ERM in April of 2007. After the implementation, Open Door Family Medical Center's reliance on written paper record were slowly phased out and replaced eventually. Open Door Family Medical Center's patient care services became more efficient through the use of the ERM system as crucial information which

identifies patients with high risk of chronic diseases who required additional medical attention, can be retrieved with ease and securely.
In addition, the result's progress of the laboratory and diagnostic imaging procedures are now traceable in the system. This traceable feature quickens the process for follow-up action when the result is ready for processing. [6] Open Door Family Medical Center also received the 2010 HIMSS Community Health Organization Davies Award of Excellence. [7]

Karim et al. [8] observed that some of the reasons that have led to successful implementation of the EMR systems are attributed as such: Firstly, the organization's sufficient budget in investing and maintaining the EMR system. Secondly, choosing the correct EMR software that fits the organization's needs and requirements which includes factors like user-friendliness, information integration. Thirdly, the adequate and continuous training classes conducted by the health care organizations for their health care professionals to equip and upgrade their skills in using the system, the support provided by organizations to improving the system to better serve the needs after gathering constructive feedbacks from the health care professionals.

Reda [9] explained some of the reasons that have led to unsuccessful implementation of the EMR systems. Firstly, lack of training. In order to reduce costs, training hours were reduced. This caused frustration to some health care professionals who may have had difficulties using the system to give up in the end. Secondly, lack of interoperability where diverse systems are unable to work together. Thirdly, not user-friendly systems also cause more inconvenience to the health care professionals than helping them.

Keeping the successful and unsuccessful factors in mind, the Inpatient Information System aims to adopt the successful ones and avoid the unsuccessful pitfalls.

CHAPTER 3 OVERVIEW OF PROPOSED SYSTEM

Inpatient Information System (IPIS) is a web application developed for nurses and doctors working in hospitals to enhance their care-giving abilities to their warded patients. IPIS takes into consideration the use of Electronic Medical Record (EMR) which allows efficient storage, retrieval, and modification of medical records. Its main objective is to help ease the process from the ward registration of patients to the treatment of patients and then to the patient's hospital discharge procedures. All patients' information such as personal particulars, medical record, and medical history are stored in the IPIS database. All nurses and doctors are able to retrieve the necessary and up-to-date information once they have logged into the system. IPIS is capable to track and reflect any update or new inputs of the patient's particulars or treatment procedures to the respective nurses and primary doctor. Conducting a search in IPIS is definitely easier and saves time than searching through patient's paper records manually. It also saves in terms of cost in storage, and altogether, IPIS allows nurse and doctors to provide a more efficient and effective health care services to the patients.

IPIS allows three types of users namely administrators, doctors and nurses. For administrators, the main functionality is to add new doctors and nurses for the eligibility to access the IPIS. For doctors, the main functionality is to prescribe inpatient treatments, medications and create/edit patient appointments, if necessary. For nurses, the main functionality is to add new patients into the system and add appointments for existing patients. Once they login into the system, administrators, doctors and nurses will see their respective portals. The three different portals have different access rights and different functionalities. This is to ensure patient's confidentiality in which only authorized users can access the patient's health information.

Once after an administrator has logged in to his/her admin portal, information such as the total number of staffs and patients are displayed in panels at the home page of the portal. Administrators have the capability to add new doctors and nurses into the system, thus giving them login access to IPIS. They are also capable to add and edit patients, inpatient treatments, prescriptions, wards, rooms etc.

When a doctor has logged in to his/her doctor's portal, a personalized appointment calendar which is unique to every doctor will be displayed at the home page of the portal. It shows the patient appointments he/she needs to attend to. Doctors can also

view a list of patients and their respective health information.
Doctors are able to add new appointment, inpatient treatment and prescription records to the patients. After the records had been added, doctors can then view and edit the details. In addition, doctors can view statistics of the number of patients diagnosed by year or month.

On the other hand, when a nurse logins to his/her nurse's portal, a to-do list will be displayed as the home page of the portal. The to-do list shows the medication schedule of the patients in the ward that the nurse is assigned to. In addition, the nurse can add follow-up appointment schedule for the patients to see their respective doctors upon their discharge from the hospital. Nurses have the functionality to add new patients and also perform search on existing patients.

IPIS creates a user-friendly, simple-to-use experience to nurses and doctors while providing all the necessary functionalities to aid their workflow. The next few chapters will discuss how IPIS is designed and implemented.

CHAPTER 4 SYSTEM DESIGN

This chapter illustrates the overview design process for IPIS and the interactions between the different type of users and the system. This project is modeled after the waterfall development model of the Software Development Life Cycle (SDLC) as waterfall model addresses all requirements to be specified before entering into the design phase to ensure minimal wastage of time and effort and reduces the risk of schedule slippage. It goes through the process from requirement specification, design, implementation, verification, and maintenance.

The first step in the waterfall model is the requirement specification gathering stage. In this stage, the functional and non-functional requirements of the system are specified. Functional requirements state what functionalities the system would provide and how the system would behave in different situations. Functional requirements implemented in the proposed system are summarized in Table 1 below.

Functional requirements
Create Doctor/Nurse accounts System must allow Administrator to create Doctor and Nurse accounts.
Update profile System must allow Administrator, Doctor and Nurse to modify and update their respective profiles.
Create/Edit Patient record System must allow Administrator and Nurse to create patient records and edit an existing patient record.
Create/Edit Patient medical record System must allow Administrator and Doctor to create the various patient medical records and edit an existing patient medical record.
Create/Edit Patient admission record System must allow Administrator and Nurse to create the patient admission records and edit an existing patient admission record.

Table 1: Functional Requirements

Non-functional requirements specify the criteria to define the operations of the system in terms of reliability, availability, and response time. Non-functional requirements implemented in the proposed system are summarized in Table 2 below.

Non-functional requirements
Each page must load in less than 15 seconds
System uptime must be at least 99%
System must run with no errors at least 24 hours from booting up
System must be able to handle the traffic during system uptime
System must be able to allow authorized medical personnel to create/view/update patients' records

Table 2: Non-Functional Requirements

The second step in the waterfall model is the design stage. In this stage, the process of deciding how the software will meet the requirements is specified. First, the proposed system is decomposed into major components such as 'Patient', 'Admission Record' etc. Next, the major components and their dependencies are identified. Lastly, the major components are further decomposed into small components if necessary. Figure 3 below shows a use case diagram which describes the overall design of the proposed system and the interactions between the users and the proposed system.

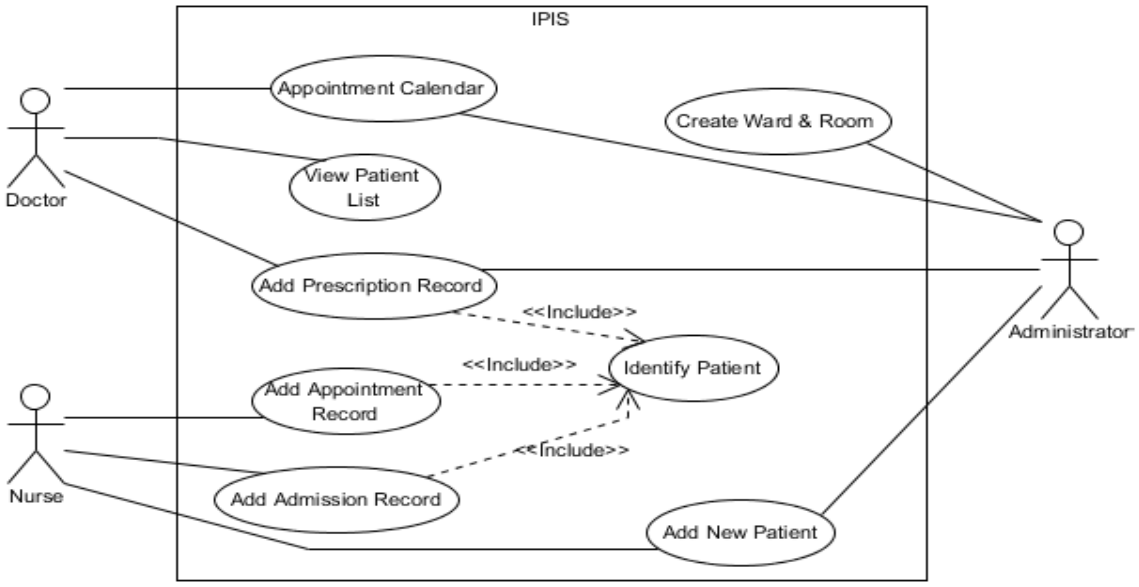

Figure 3: Use Case Diagram

The third step in the waterfall model is the implementation stage. In this stage, the data structures, data types, frameworks, methodologies and algorithms used in the proposed system are specified. More details on the implementation will be discussed in Chapter 5.

The last two steps in the waterfall model are validation and maintenance stages. Validation is an important process where the proposed system needs to be tested thoroughly. Errors are not tolerated especially in health care system as it involves the safety of a patient. More details on validation will also be discussed in Chapter 5. Moreover, maintenance of the system involves enhancing existing functions, provides new functions, and also troubleshooting to resolve the existing flaws. In order to improve the design and implementation of the system, maintenance of the system is a crucial step to keep the system in full working condition at all times.

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 Development Tools

Three software applications are used to develop the Inpatient Information System (IPIS). They are Microsoft Visual Studio 2010, Microsoft SQL Server 2008 and Eclipse IDE.

5.1.1. Microsoft Visual Studio 2010

Microsoft Visual Studio 2010 is an Integrated Development Environment (IDE) software used to develop the web application for IPIS. The main programming language used is C#, together with Cascading Style Sheets (CSS), jQuery, jQuery Mobile and Razor.

IPIS is structured using ASP.NET MVC 4 where it is a software architecture that separates the view presentation of the information (view), the modeling information state (model) and the actions based on the user input (controller) into three separate classes. Both the view and controller are dependent on the model but however, the model is not dependent on either the view or the controller resulting that the business logic is loosely coupled from the user interface logic. Therefore, ASP.NET MVC 4 encourages modularity and reusability purposes which can reduce development time. ASP.NET MVC 4 is also capable to test each and every component of IPIS right from the beginning, without initially connecting to any database.

CSS helps set the format and look of the overall IPIS in both desktop and mobile web version while both jQuery and jQuery Mobile are both open source software that support desktop and mobile web platforms and they are capable of creating a powerful and dynamic web application experience for the users through the usage of Ajax, simplified event-handling and animations. Razor is a view engine used by .NET programming languages to create dynamic web pages.

5.1.2. Microsoft SQL Server 2008

Microsoft SQL Server 2008 is a database server which is used to store and retrieve all the patient health information. Data is the most important asset in IPIS and maintaining these data is crucial. Using the SQL Server Management Studio, configuring and managing database tables became much easier. In addition, Microsoft SQL Server 2008 provides the security, availability and performance.

5.1.3. Eclipse IDE

Eclipse IDE is an Integrated Development Environment (IDE) software that comprises a base workspace and an extensible plug-in system which allows customization to the environment. Together with the aid of Android Software Development Kit (SDK), the Eclipse IDE was used to develop the Android mobile web view application for IPIS which will direct the user to the web application site upon launching.

5.2 Implementation of Main Modules

This section describes the implementation of the main modules in Inpatient Information System (IPIS).

5.2.1. Appointment Calendar

Appointment calendar is displayed in a month view as the default view at the 'Home' page of the doctor portal once the doctor has logged in to IPIS while an administrator's appointment calendar will be displayed also in a month view as the default view on the 'Appointment' page of the administrator portal. Doctors and administrators have the options to toggle among day, week or month view of their appointment calendars. The current date is highlighted for easy viewing.

Doctors and administrators can create a patient appointment by double-clicking on a blank space of that particular date in the calendar or edit the details of an existing patient appointment when double-clicked on that appointment of the calendar. A pop-up dialog will appear during the creating or editing of patient appointment.

When creating the patient appointment, doctors must fill in the appointment details, patient name from drop down list, appointment start time and appointment end time while administrators have an additional field of selecting doctor name from drop down list to fill in. The appointment calendar is implemented using dhtmlxschedule, a rich Javascript calendar UI component.

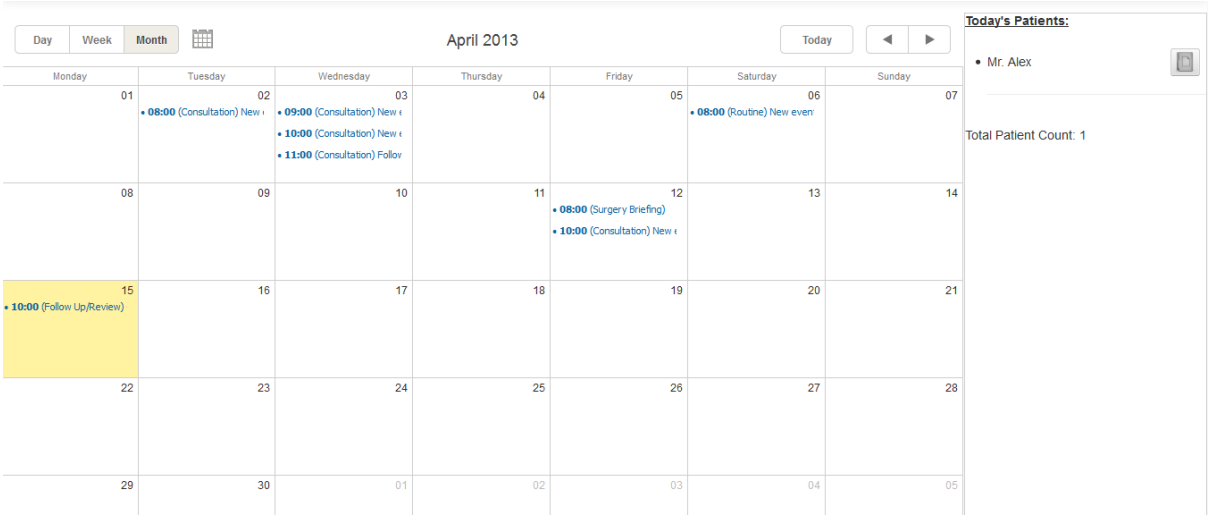

Figure 4: Doctor's Appointment Calendar

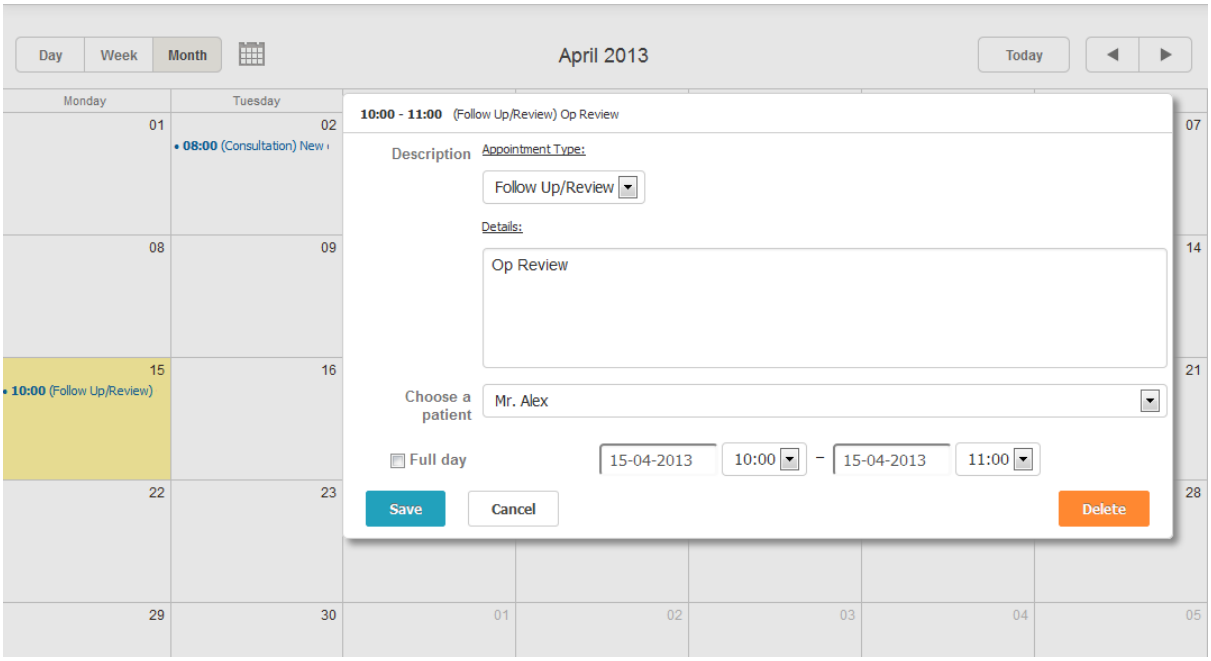

Figure 5: Viewing Appointment Details

5.2.2. Patient List

Doctors can browse through their patient lists to create, edit or remove prescription, inpatient treatment and appointment records. Any patient who has been attended by the doctor would be added to the patient list. The patient list is displayed in a table form in the ‘Patients’ page of the doctor portal. Below the patient list of the ‘Patients’ page is a pie chart that contains the proportion between warded patients and not warded patients. Selecting any patient from the patient list will display the respective patient’s ‘Patient Details’ page in the doctor portal.

‘Patient Details’ page is divided into five sections. First section shows the patient's particulars and his/her admission status. Second section shows the patient's prescription records. Doctors can create/edit the prescription records. Third section shows the patient's inpatient treatment records. Doctors can only create the inpatient treatment record if the patient is currently warded. Doctors have the ability to edit existing inpatient treatment record. Fourth section shows the patient's appointment records. Doctors can create/edit the prescription records. Fifth section shows the patient's admission history. Second to fifth sections are structured using jQuery Accordion. Accordion gives doctors the flexibility and freedom to expand and collapse the different sections.

5.2.3. View Statistics

Doctors are able to view monthly or yearly statistics on the number of patients they had attended to, number of patients they had performed inpatient treatment on and number of patients they had issued prescription to. Nurses are able to view monthly or yearly statistics on the occupancy rate of the ward assigned to the nurses. Administrators are able to view monthly or yearly statistics on the total occupancy rate of all the wards, total number of patients attended to, total number of patients with treatment performed and total number of patients with prescription issued. The graphs displayed are implemented using Highcharts, a Javascript chart UI component.

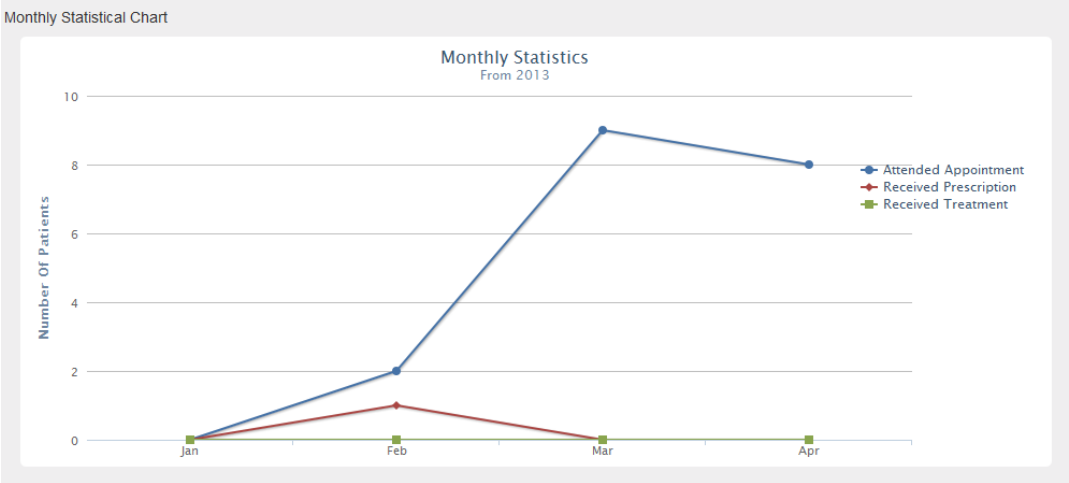

Figure 6: Monthly Statistics of Viewing Statistics

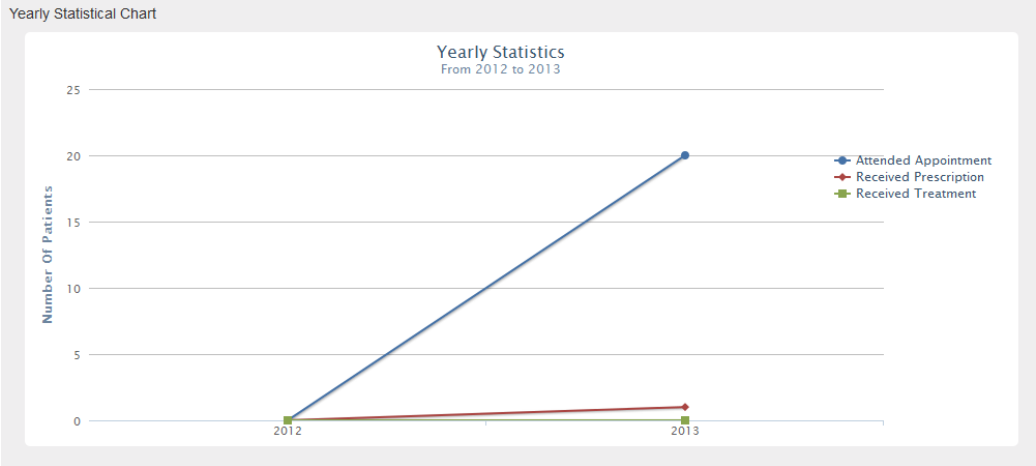

Figure 7: Yearly Statistics of Viewing Statistics

5.2.4. Add Prescription Record

Administrators and doctors are able to create prescription records. When doctors create prescription record for a particular patient, doctors must fill in the drug name from the drop down list, dosage amount of that particular drug and prescription instructions.

However when administrators create prescription record, administrators must fill in the patient name from the drop down list before filling in any content. After the patient name has been filled in, administrators must then fill in the doctor's name from a list of doctors whom that particular patient has visited in the drop down list populated through Ajax. After the doctor name has been filled in, administrators must fill in the drug name from

the drop down list, dosage amount of that particular drug and prescription instructions. If the prescribed drug is one of the drug allergies of the patient, the prescription record cannot be created and the appropriate error message will be reflected on the page. After creating the prescription records, administrators and doctors can edit and update the details accordingly.

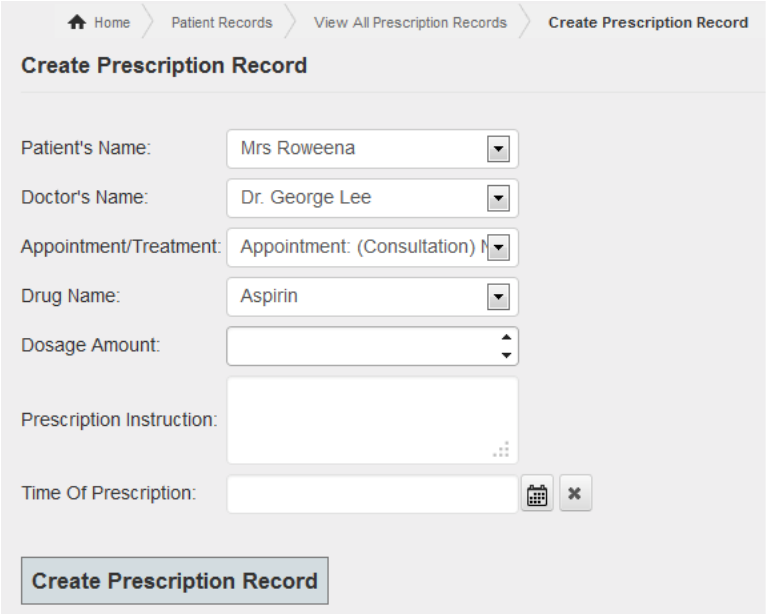

Figure 8: Administrator's Create Prescription Page

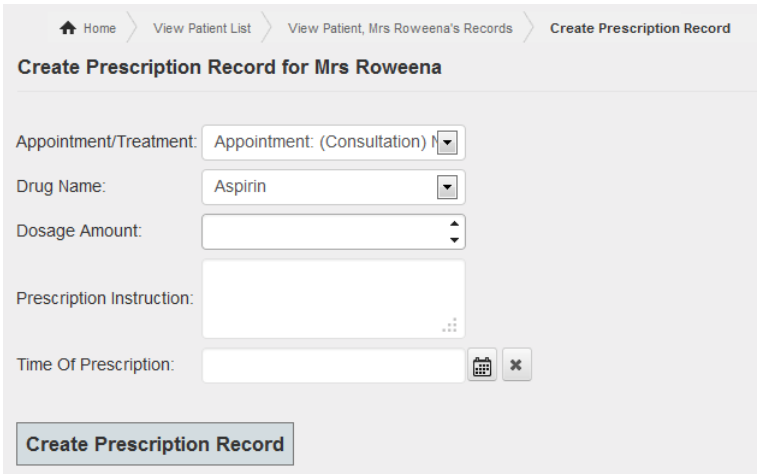

Figure 9: Doctor's Create Prescription Page

5.2.5. Add Inpatient Treatment Record

Administrators and doctors can only create inpatient treatment records to patients who are currently warded. When doctors create inpatient treatment record for a particular patient, doctors must fill in the inpatient treatment name from the drop down list, duration in minutes.

However when administrators create inpatient treatment record, administrators must fill in the patient name from the drop down list before filling in any content. After the patient name has been filled in, administrators must then fill in the doctor name from a list of doctors whom that particular patient has visited in the drop down list populated through Ajax. After the doctor name has been filled in, administrators must fill in the inpatient treatment name from the drop down list, duration in minutes. If the any of the drugs administered during the inpatient treatment is one of the drug allergies of the patient, the inpatient treatment record cannot be created and the appropriate error message will be reflected on the page. After creating the inpatient treatment records, administrators and doctors can edit and update the details accordingly.

Figure 10: Doctor's Create Treatment Page

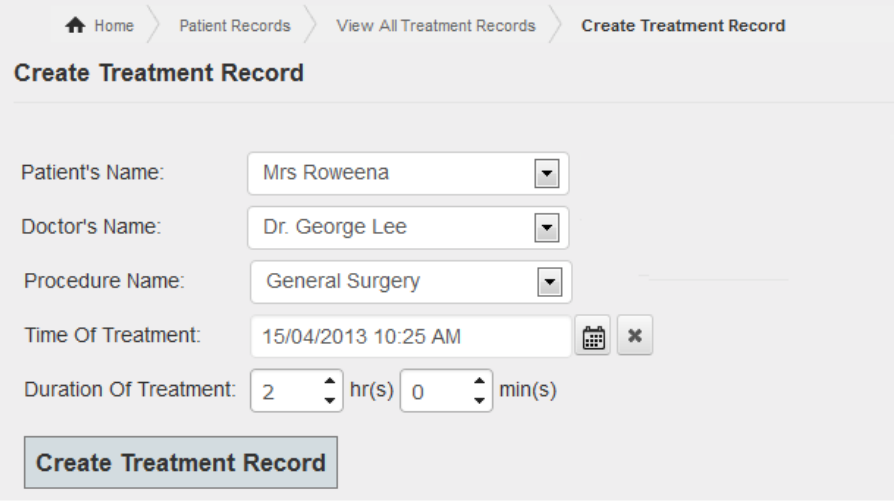

Figure 11: Administrator's Create Treatment Page

5.2.6. Add Admission Record

Administrators and nurses can only create admission records to patients who are currently not warded. When administrators and nurses create admission record, they must fill in the patient name from the drop down list, room number from a list of rooms with available bed capacity in the drop down list, admission date. Discharge date is not compulsory to be filled when creating the admission records. After creating the admission records, administrators and nurses can edit and update the details accordingly.

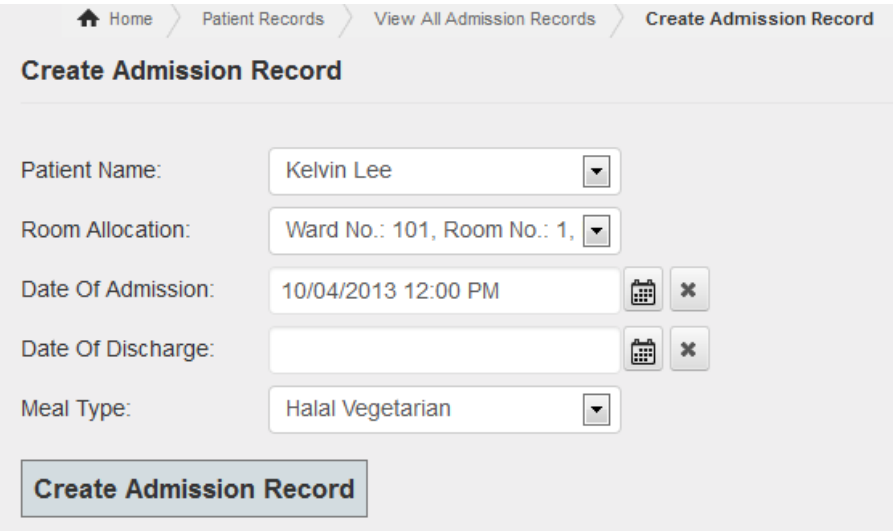

Figure 12: Administrator And Nurse's Create Admission Page

5.2.7. Add New Patients

Administrators and nurses can add new patients by entering some basic personal particulars like NRIC/FIN number, name, gender, nationality, date of birth, address, contact number, and email address. A photo of the patient can also be uploaded. After adding patients, administrators and nurses can also edit and update these details accordingly.

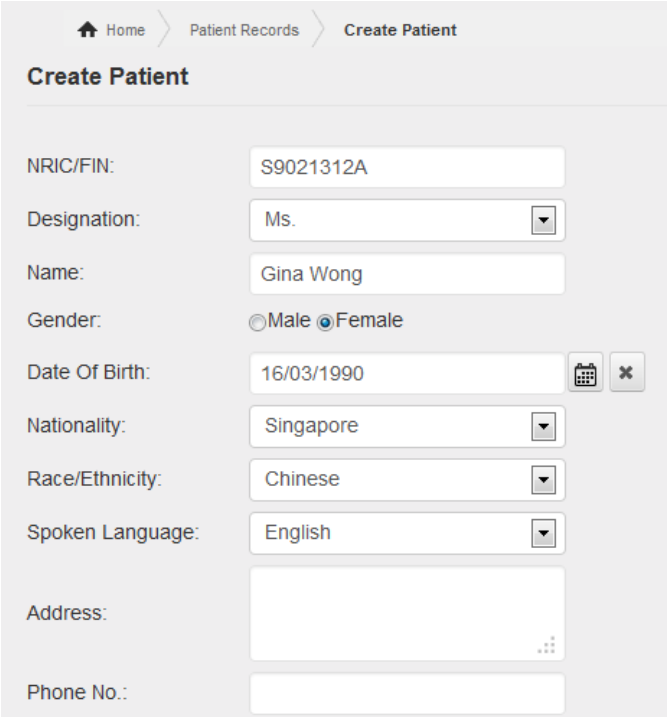

Figure 13: Administrator And Nurse's Create Patient Page

5.2.8. Add Follow Up Appointment

One important task of a nurse is to add patient follow up appointment record before the patient is discharged from the ward. Nurses can browse through the list of patients warded in the same ward assigned to the nurses. Once a patient has been selected from the list of patients, nurses can then proceed to add appointment record for that particular patient. When creating the appointment records, nurses must fill in the doctor name from a drop down list, appointment details, appointment start time, and appointment end time. After adding follow up appointment record, nurses can also edit and update these details accordingly.

5.3 Validation and Testing

To ensure that Inpatient Information System (IPIS) functions correctly and performs the desired functionalities specified accurately, it is crucial to have validation checks and tests frequently. As IPIS is structured using ASP.NET MVC 4's architecture, IPIS is eligible to apply ASP.NET MVC 4's validation logic to the numerous models such as login, patient, prescription record, admission record and many other models in IPIS. With the application of validation rules, validation checks are activated to perform during every submission of user request during login, create/edit patient and create/edit prescription records and many other processes. In the event where the submission of user request has failed the validation check, an appropriate error message will be displayed on the page to help the user to identify which input to amend before the next submission.

In addition, the SQL transactional feature was implemented in IPIS's database which ensures all the transactions are executed correctly before committing into the database. If there is any error in the execution, rollback to the previous saved state would be executed and data in the database would remain unaffected. This rollback property is crucial especially when the execution involves multiple cascading actions. For example, when a nurse adds an admission record for a patient, the admission status of the respective patient record will be modified. Therefore, if committing the admission record into the database fails unexpectedly, no modification would be made to the respective patient record. Likewise, if the modification of the admission status in patient record is unsuccessful, the added admission record will not be committed into the database but instead the process of rollback to the state before the admission record was added would be executed.

CHAPTER 6 DATABASE SYSTEM DESIGN

Patient health data is an asset to any health care information system. It must ensure accuracy and security. Thus, designing how data are stored and retrieved is a crucial step in creating the web application for Inpatient Information System (IPIS). In this chapter, a simplified Entity Relationship (ER) Model and the various database tables are shown to illustrate the structure of the data and also how they relate to each other.

6.1 ER Diagram

To simplify the ER Diagram, the attributes of each table are not shown. Only the tables and their relationships with the other tables are shown in Figure 14 below. From the diagram, we can see that there are three types of users – administrators, doctors and nurses. The main workflow of an administrator is to manage ward, room, drug, side effect, inpatient procedure, prescription, treatment, appointment, admission and patient records. The main workflow of a doctor is to manage prescription, treatment and appointment records for patient while the main workflow of a nurse is to manage admission, appointment and patient records. After finalizing the ER diagram, database tables were then created. Database tables created are shown in the next section

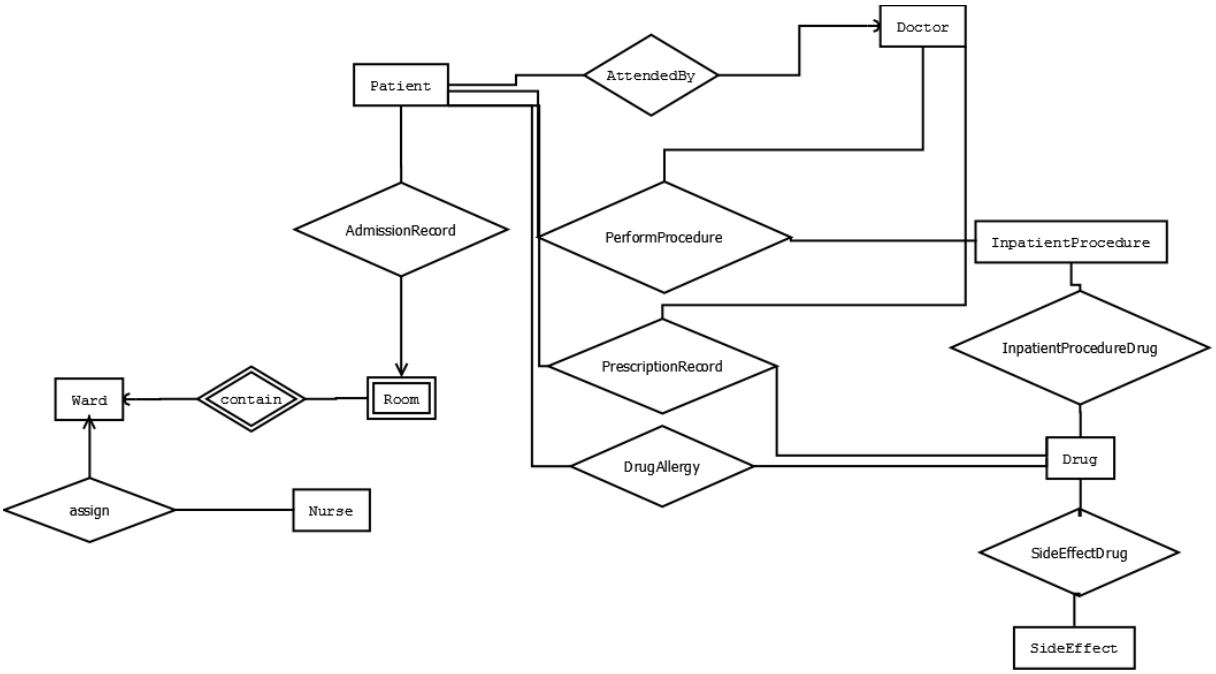

Figure 14: ER Diagram

6.2 Database Tables

Membership	
	CREATE TABLE Membership (UserId int NOT NULL, CreateDate datetime NULL, ConfirmationToken nvarchar(128) NULL, IsConfirmed bit NULL, Password nvarchar(128) NOT NULL, PasswordChangedDate datetime NULL PRIMARY KEY(UserId));
Description	• This table stores all the users' login information of the Inpatient Information System whom the administrators had registered.
Constraints	• UserId is auto generated. • UserId is the primary key.

Table 3: Database Table - Membership

Roles	
	CREATE TABLE Roles (RoleId int IDENTITY NOT NULL, RoleName nvarchar(256) NOT NULL, PRIMARY KEY (RoleId));
Description	• This table stores the static roles used in the Inpatient Information System namely 'Administrator', 'Doctor' and 'Nurse'.
Constraints	• RoleId is auto generated. • RoleId is the primary key.

Table 4: Database Table - Roles

UsersInRoles	
	CREATE TABLE UsersInRoles (UserId int NOT NULL, RoleId int NOT NULL, PRIMARY KEY (UserId, RoleId), FOREIGN KEY (UserId) REFERENCES Membership, FOREIGN KEY (RoleId) REFERENCES Role);
Description	• This table stores the records of the roles assigned to the users in the Inpatient Information System.
Constraints	• UserId and RoleId are composite primary keys. • UserId and RoleId must exist in both Membership and Role tables respectively.

Table 5: Database Table – UsersInRoles

Administrator	
	CREATE TABLE Administrator (AdminID nchar(9) NOT NULL, Name varchar(50) NOT NULL, Gender varchar(50) NOT NULL, DateOfBirth datetime NOT NULL, Age int NOT NULL, Nationality varchar(50) NOT NULL, Address varchar(100) NOT NULL, Email varchar(50) NOT NULL, Phone int NOT NULL, LastLogin datetime NULL, LastUpdate datetime NULL, ProfilePicture image NULL, UserId int NOT NULL, PRIMARY KEY (AdminID), FOREIGN KEY (UserId) REFERENCES Membership);
Description	• This table stores all administrators' account in the Inpatient Information System.
Constraints	• AdminID is primary key. • UserId must exist in Membership table.

Table 6: Database Table – Administrator

Nurse	
	CREATE TABLE Nurse (NurseID nchar(9) NOT NULL, Designation varchar(50) NOT NULL, Position varchar(50) NOT NULL, Name varchar(50) NOT NULL, Gender varchar(50) NOT NULL, DateOfBirth datetime NOT NULL, Age int NOT NULL, Nationality varchar(50) NOT NULL, Address varchar(100) NOT NULL, Email varchar(50) NOT NULL, Phone int NOT NULL, LastLogin datetime NULL, LastUpdate datetime NULL, ProfilePicture image NULL, UserId int NOT NULL, WardID int NULL, PRIMARY KEY (NurseID), FOREIGN KEY (UserId) REFERENCES Membership, FOREIGN KEY (WardID) REFERENCES Ward);
Description	• This table stores all nurses' account in the Inpatient Information System.
Constraints	• NurseID is primary key. • UserId must exist in Membership table. • WardID must exist in Ward table.

Table 7: Database Table – Nurse

Doctor	
	CREATE TABLE Doctor (DoctorID nchar(9) NOT NULL, Designation varchar(50) NOT NULL, Speciality varchar(50) NOT NULL, Position varchar(50) NOT NULL, Name varchar(50) NOT NULL, Gender varchar(50) NOT NULL, DateOfBirth datetime NOT NULL, Age int NOT NULL, Nationality varchar(50) NOT NULL, Address varchar(100) NOT NULL, Email varchar(50) NOT NULL, Phone int NOT NULL, LastLogin datetime NULL, LastUpdate datetime NULL, ProfilePicture image NULL, UserId int NOT NULL, PRIMARY KEY (DoctorID), FOREIGN KEY (UserId) REFERENCES Membership);
Description	• This table stores all doctors' account in the Inpatient Information System.
Constraints	• DoctorID is primary key. • UserId must exist in Membership table.

Table 8: Database Table – Doctor

Patient	
	CREATE TABLE Patient (PatientID nchar(9) NOT NULL, Designation varchar(50) NOT NULL, Name varchar(50) NOT NULL, Gender varchar(50) NOT NULL, DateOfBirth datetime NOT NULL, Age int NOT NULL, Nationality varchar(50) NOT NULL, Address varchar(100) NOT NULL, Email varchar(50) NOT NULL, Phone int NOT NULL, Height float NOT NULL, Weight float NOT NULL, BloodType varchar(50) NOT NULL, Admitted varchar(50) NOT NULL, ProfilePicture image NULL, UserId int NOT NULL, DoctorID char(9) NOT NULL, PRIMARY KEY (PatientID), FOREIGN KEY (UserId) REFERENCES Membership, FOREIGN KEY (DoctorID) REFERENCES Doctor);
Description	• This table stores all patients' records with each patient being assigned to a primary doctor in the Inpatient Information System.
Constraints	• PatientID is primary key. • UserId must exist in Membership table. • DoctorID must exist in Doctor table.

Table 9: Database Table – Patient

Drug	
	CREATE TABLE Drug (DrugName varchar(50) NOT NULL, DrugType varchar(50) NOT NULL, DosageForm varchar(50) NOT NULL, DrugDescription varchar(100) NOT NULL, PRIMARY KEY (DrugName));
Description	• This table stores all drug records in the Inpatient Information System.
Constraints	• DrugName is primary key. • No value can be null.

Table 10: Database Table – Drug

SideEffect	
	CREATE TABLE SideEffect (SideEffectID int IDENTITY NOT NULL, SideEffectDescription varchar(50) NOT NULL, PRIMARY KEY (SideEffectID));
Description	• This table stores all records of side effects that can be used to couple with created drugs in the Inpatient Information System.
Constraints	• SideEffectID is auto generated • SideEffectID is primary key. • No value can be null.

Table 11: Database Table - SideEffect

DrugAllergy	
	CREATE TABLE DrugAllergy (PatientID char(9) NOT NULL, DrugName varchar(50) NOT NULL, PRIMARY KEY (PatientID, DrugName), FOREIGN KEY (PatientID) REFERENCES Patient, FOREIGN KEY (DrugName) REFERENCES Drug);
Description	• This table stores the records of patient's drug allergies in the Inpatient Information System.
Constraints	• PatientID and DrugName are composite primary keys • PatientID and DrugName must exist in Patient and Drug tables respectively • No value can be null.

Table 12: Database Table – DrugAllergy

SideEffectDrug	
	CREATE TABLE SideEffectDrug (SideEffect_SideEffectID int NOT NULL, Drug_DrugName varchar(50) NOT NULL, PRIMARY KEY (SideEffect_SideEffectID, Drug_DrugName), FOREIGN KEY (SideEffect_SideEffectID) REFERENCES SideEffect, FOREIGN KEY (Drug_DrugName) REFERENCES Drug);
Description	• This table stores the many-to-many relationship of drugs having side effects in the Inpatient Information System.
Constraints	• SideEffect_SideEffectID and Drug_DrugName are composite primary keys • SideEffect_SideEffectID and Drug_DrugName must exist in SideEffect and Drug tables respectively • No value can be null.

Table 13: Database Table – SideEffectDrug

InpatientProcdeure	
	CREATE TABLE InpatientProcedure (InpatientProcedureID int IDENTITY NOT NULL, InpatientProcedureName varchar(50) NOT NULL, InpatientProcedureDescription varchar(100) NOT NULL, PRIMARY KEY (InpatientProcedureID));
Description	• This table stores all records of inpatient procedures in the Inpatient Information System.
Constraints	• InpatientProcedureID is auto generated • InpatientProcedureID is primary key. • No value can be null.

Table 14: Database Table – InpatientProcedure

InpatientProcedureDrug	
	CREATE TABLE InpatientProcedureDrug (InpatientProcedure_InpatientProcedureID int NOT NULL, Drug_DrugName varchar(50) NOT NULL, PRIMARY KEY (InpatientProcedure_InpatientProcedureID, Drug_DrugName), FOREIGN KEY (InpatientProcedure_InpatientProcedureID) REFERENCES InpatientProcedure, FOREIGN KEY (Drug_DrugName) REFERENCES Drug);
Description	• This table stores the many-to-many relationship of inpatient procedures using drugs in the Inpatient Information System.
Constraints	• InpatientProcedure_InpatientProcedureID and Drug_DrugName are composite primary keys • InpatientProcedure_InpatientProcedureID and Drug_DrugName must exist in InpatientProcedure and Drug tables respectively • No value can be null.

Table 15: Database Table – InpatientProcedureDrug

Ward	
	CREATE TABLE Ward (WardID int NOT NULL, WardType varchar(50) NOT NULL, BuildingName varchar(50) NOT NULL, FloorLevel int NOT NULL, PRIMARY KEY (WardID));
Description	• This table stores all ward records in the Inpatient Information System.
Constraints	• WardID is primary key. • No value can be null.

Table 16: Database Table – Ward

Room	
	CREATE TABLE Room (WardID int NOT NULL, RoomID int NOT NULL, RoomCode varchar(50) NOT NULL, RoomClass varchar(50) NOT NULL, BedCapacity int NOT NULL, PRIMARY KEY (WardID, RoomID), FOREIGN KEY (WardID) REFERENCES Ward);
Description	• This table stores all room records and shows which room belongs to which ward in the Inpatient Information System.
Constraints	• WardID and RoomID are composite primary keys. • No value can be null.

Table 17: Database Table – Room

AttendedBy	
	CREATE TABLE AttendedBy (PatientID char(9) NOT NULL, DoctorID char(9) NOT NULL, PRIMARY KEY (PatientID, DoctorID), FOREIGN KEY (PatientID) REFERENCES Patient, FOREIGN KEY (DoctorID) REFERENCES Doctor);
Description	• This table stores records of patients whom the doctors had attended to in the Inpatient Information System.
Constraints	• PatientID and DoctorID are composite primary keys. • PatientID and DoctorID must exist in Patient and Doctor tables respectively. • No value can be null.

Table 18: Database Table – AttendedBy

PerformProcedure	
	CREATE TABLE PerformProcedure (PatientID char(9) NOT NULL, DoctorID char(9) NOT NULL, InpatientProcedureID int NOT NULL, ProcedureTime datetime NOT NULL, ProcedureDuration float NOT NULL, PRIMARY KEY (PatientID, DoctorID, InpatientProcedureID, ProcedureTime), FOREIGN KEY (PatientID) REFERENCES Patient, FOREIGN KEY (DoctorID) REFERENCES Doctor, FOREIGN KEY (InpatientProcedureID) REFERENCES InpatientProcedure);
Description	• This table stores the records of doctors performing procedure to the patients in the Inpatient Information System.
Constraints	• PatientID, DoctorID and InpatientProcedureID are composite primary keys. • PatientID, DoctorID and InpatientProcedure must exist in Patient, Doctor and InpatientProcedure tables respectively

	• No value can be null.

Table 19: Database Table – PerformProcedure

AdmissionRecord	
	CREATE TABLE AdmissionRecord (PatientID char(9) NOT NULL, WardID int NOT NULL, RoomID int NOT NULL, AdmissionDate datetime NOT NULL, DischargeDate datetime NULL, RoomCode varchar(50) NOT NULL, PRIMARY KEY (PatientID, WardID, RoomID, AdmissionDate), FOREIGN KEY (PatientID) REFERENCES Patient, FOREIGN KEY (WardID, RoomID) REFERENCES Room);
Description	• This table stores all admission records in the Inpatient Information System.
Constraints	• PatientID, WardID, RoomID and AdmissionDate are composite primary keys. • PatientID, WardID and RoomID must exist in Patient and Room tables respectively.

Table 20: Database Table – AdmissionRecord

PrescriptionRecord	
	CREATE TABLE PrescriptionRecord (PatientID char(9) NOT NULL, DoctorID char(9) NOT NULL, DrugName varchar(50) NOT NULL, TimeOfPrescription datetime NOT NULL, DosageAmount int NOT NULL, PrescriptionInstruction varchar(100) NULL, PRIMARY KEY (PatientID, DoctorID, DrugName, TimeOfPrescription), FOREIGN KEY (PatientID) REFERENCES Patient,

	FOREIGN KEY (DoctorID) REFERENCES Doctor, FOREIGN KEY (DrugName) REFERENCES Drug);
Description	• This table stores all prescription records in the Inpatient Information System.
Constraints	• PatientID, DoctorID, DrugName, TimeOfPrescription are composite primary keys. • PatientID, DoctorID, DrugName must exist in Patient, Doctor and Drug tables respectively.

Table 21: Database Table – PrescriptionRecord

AppointmentRecord	
	CREATE TABLE AppointmentRecord (AppointmentID int IDENTITY NOT NULL, PatientID char(9) NOT NULL, DoctorID char(9) NOT NULL, AppointmentText varchar(50) NOT NULL, StartTime datetime NOT NULL, EndTime datetime NULL, PRIMARY KEY (AppointmentID), FOREIGN KEY (PatientID) REFERENCES Patient, FOREIGN KEY (DoctorID) REFERENCES Doctor);
Description	• This table stores all appointment records in the Inpatient Information System.
Constraints	• AppointmentID is auto generated. • AppointmentID is the primary key. • PatientID, DoctorID must exist in Patient and Doctor tables respectively.

Table 22: Database Table – AppointmentRecord

CHAPTER 7 RESULTS AND DISCUSSIONS

7.1 Analysis and Evaluation

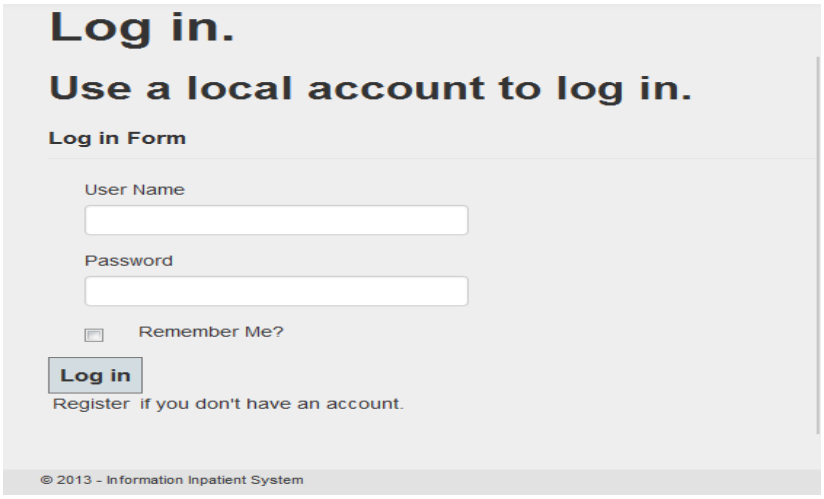

Figure 15: Login Page

Figure 15 shows the login page for Inpatient Information System (IPIS). Only administrators, doctors or nurses with authorized access to the system can login into the IPIS.

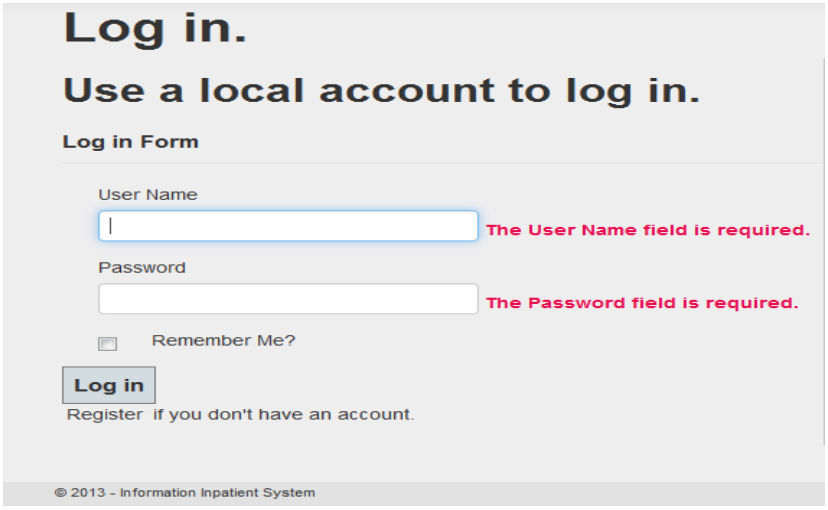

Figure 16: Validation on Login Page

Figure 16 shows the required field validation check at the login page.

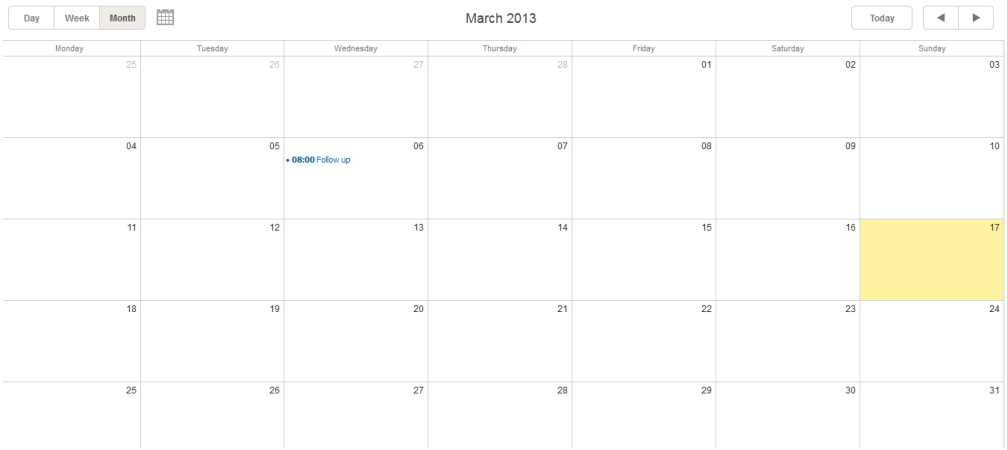

Figure 17: Appointment Calendar

Figure 17 shows the ‘Home’ page of the doctor's portal. The purpose of this calendar is to clearly show the doctor in the form of a calendar the number of patient appointments. Clicking on the bulleted appointment will open a popup window that shows the details of appointment record.

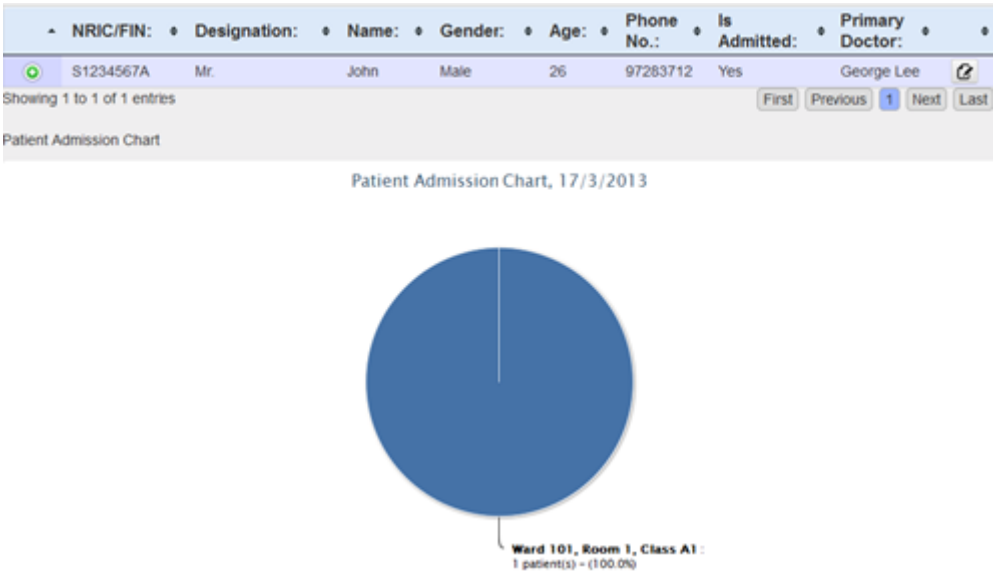

Figure 18: View Patient List

Figure 18 shows ‘View Patient List’ page of the doctor's portal. In this page, it will show the records of all the patients attended by the doctor in a table form followed by a patient admission chart to display the percentage of warded patients under the doctor’s care.

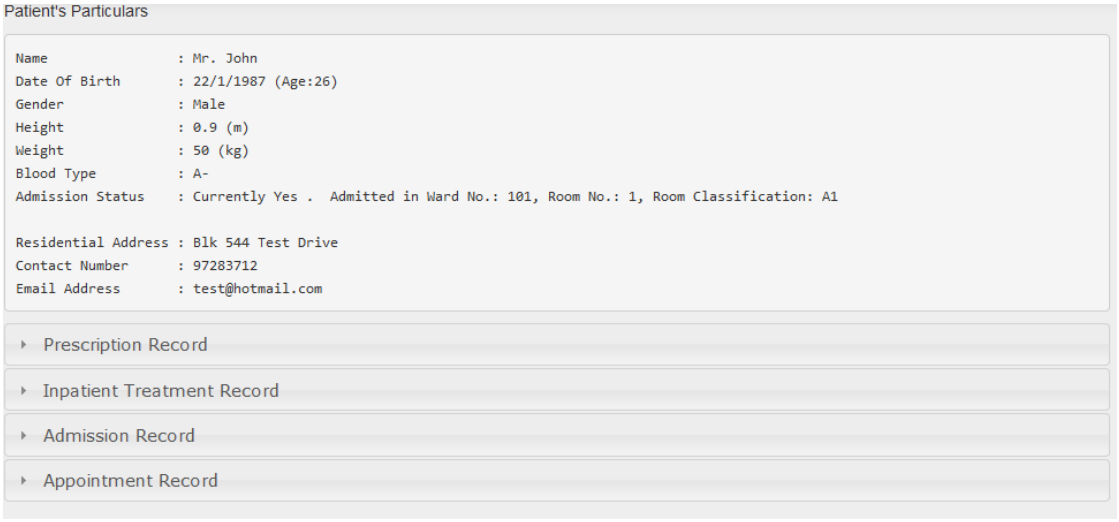

Figure 19: View Patient Record

Figure 19 shows the ‘Patient Record’ page of the doctor's portal. Doctors are able to create treatment, prescription and appointment records for a particular patient and edit the existing records.

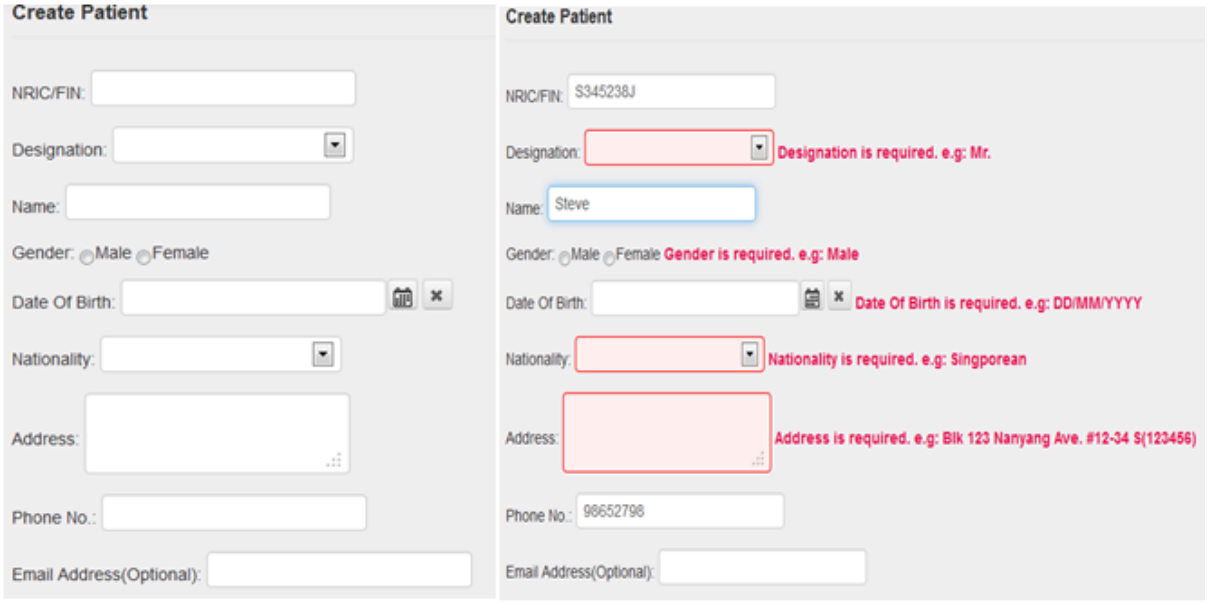

Figure 20: Create Patient Record

Figure 20 shows the ‘Create Patient’ page. Only administrators and nurses are capable of creating new patient records. When creating the patient record, validation check will be performed whenever the user focuses on any element of the page. IPIS will reflect to the user with the appropriate error messages whenever the validation rules had been violated.

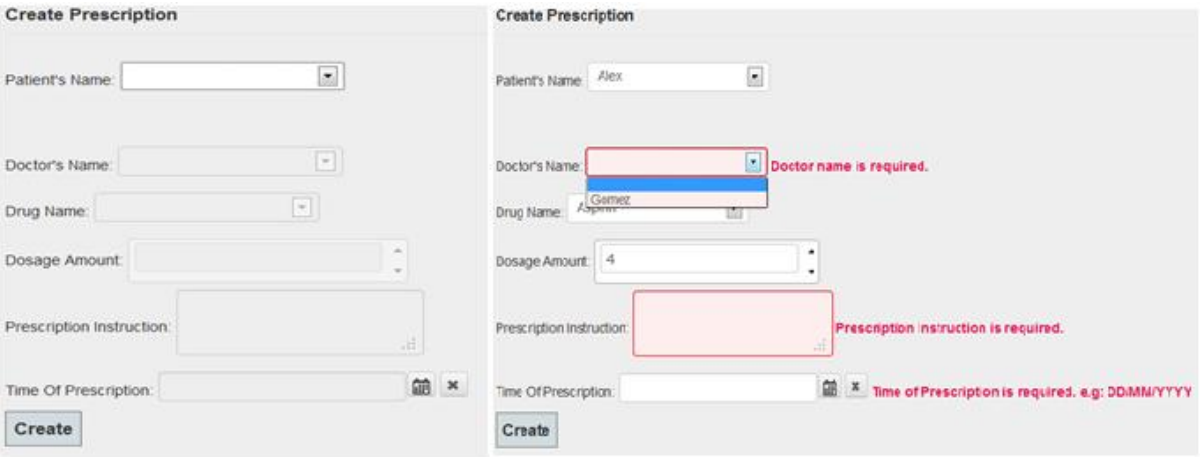

Figure 21: Create Prescription Record

Figure 21 shows the 'Create Prescription page. Only administrators and doctors are capable of creating prescription records. The patient's name has to be input first and through Ajax submission, those doctors who has attended by that patient will be populated in the drop down list of 'Doctor's Name'. When creating the prescription record, validation check will be performed whenever the user focuses on any element of the page. IPIS will reflect to the user with the appropriate error messages whenever the validation rules had been violated.

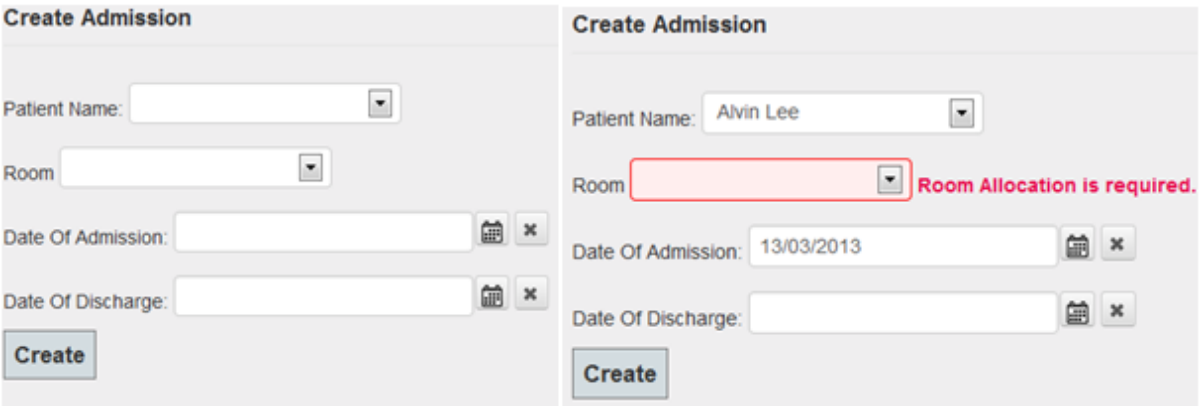

Figure 22: Create Admission Record

Figure 22 shows the 'Create Admission page. Only administrators and nurses are capable of creating admission records of patients who are currently not warded. Rooms with available bed capacity will be populated in the drop down list of 'Room'. When creating the admission record, validation check will be performed whenever the user focuses on any element of the page. IPIS will reflect to the user with the appropriate error messages whenever the validation rules had been violated.

7.2 Problems Encountered

This section describes in detail obstacles and problems I had encountered while doing research for this project and developing the Inpatient Information System (IPIS).

Firstly, one of the problems was to juggle between school work and project. Having to attend lectures and tutorials, with the addition of coping with assignments, tests, and exams, the remaining amount of time to spend on developing the web application and researching for information was very limited. In order to overcome the time limitation and to better manage time, a schedule was created to help me identify the available time to be allocated for this project each week.

Secondly, another problem I had encountered was the lack of knowledge related to the areas of health care industry and health care services. Reading journal articles, searching information through the Internet and interviewing health care professionals with a wealth of experience has helped me gather more knowledge about the health care industry and helped me understand the current trend of using Electronic Medical Record (EMR) and its importance to the patient's well-being.

Thirdly, another problem was my unfamiliarity of ASP.NET MVC 4 architecture. Through spending time reading about the architecture on the Internet and learning through online examples of ASP.NET MVC 4 implementation, knowledge and skills in programming with ASP.NET MVC 4 increased and thus, development of IPIS became less error-prone.

Lastly, storing data in the database is an important factor that requires careful planning. Initially, I encountered some difficulties in deciding which data are to be stored and how they are to be stored. Through specifying the different data requirement based on the different components, the ER diagram had gone through numerous changes over time to arrive at its present state.

Though problems and obstacles may arise during development, it is important to face the problems and obstacles directly and then search for solutions rather than shunning away from the problems and obstacles. If there is no known direct solution available, alternative solutions should be considered and implemented.

7.3 Lessons Learnt

This section describes in details the lessons learnt while doing research for this project and developing the Inpatient Information System (IPIS).

One of the lessons learnt is that I understand more about the health care system and how IT technology can be integrated into the health care services to improve efficiency and effectiveness. Some health care organizations have already started using the Electronic Medical Record (EMR) to manage all their patient health information. Therefore, this project aims to help health care professionals improve the work of their daily routines by providing a simple, easy to use and user-friendly web application.

In terms of programming, implementing ASP.NET MVC 4 architecture requires a lot of learning efforts as I had no experience in programming using ASP.NET MVC 4. Through determination and a commitment to learn the architecture, programming in ASP.NET MVC 4 became much easier as time progressed. During the development of IPIS, I had also learnt how to use jQuery and some Javascript libraries namely dhtmlxschedule and Highcharts and implement those components into IPIS to enhance user experience.

In summary, not all lessons learnt can be easily implemented. Continuous improvement is needed to keep up with the advancement in technology. The next section gives more details on what future enhancements can be implemented in the proposed Inpatient Information System.

7.4 Future Work

As IT technology keeps progressing over time, software systems need to be upgraded periodically to keep up with the latest trend. Thus, Inpatient Information System (IPIS) too has to undergo the phase of upgrading and improvements. However, due to the limitation of time, the enhancements will be plan for future work.

Security is always an important factor to consider, especially in the health care industry where protecting patient privacy takes the highest priority. Therefore information should be encrypted when storing and retrieving to prevent unauthorized personnel from intercepting or sniffing the information. As the number of data increases greatly over time, it is important to ensure these data are securely stored in the database. Therefore, some security measures need to be implemented at both the system and the database.

The mode of retrieving and storing data is another one of the important factors to consider. How data is stored and retrieved will determine the system performance. Query performance can be measured by three metrics namely query cost, page reads and query execution time. [10] Query cost assesses the CPU and IO resources used during the execution of the query. Page reads refer to the number of pages read off the hard disk during query execution. Query execution time measures the time taken for query execution in milliseconds. There are several ways to enhance the query performance. One way is to rewrite queries to reduce the scope of retrieving the data set. Another way is to normalize the tables to produce efficient querying. When normalized tables produce adverse effect in querying, the option to de-normalize the tables should be considered. Another way is to index the columns of a table which are frequently accessed as indexing can shorten the query execution time.

CHAPTER 8 CONCLUSION

As indicated in the Chapter 1, the objectives of this project is to provide warded patients with efficient health care services and at the same time streamlines the nurses and doctors' daily workflow. The Inpatient Information System (IPIS) has achieved these objectives by providing a simple, easy to use, and user-friendly system that allows nurses and doctors to access complete patient health information during ward registration, discharge planning, inpatient procedures and prescriptions.

IPIS is modeled after the waterfall development model which ensures minimal wastage of time and effort and reduces the risk of schedule slippage. IPIS is structured after ASP.NET MVC 4 software architecture which separates the presentation from the business and user interaction logic and encourages modularity and reusability purposes which can reduce development time. On top of the ASP.NET MVC 4 architecture, jQuery and other Javascript libraries such as dhtmlxschdule and Highcharts were also implemented to enhance the user experience when using IPIS. The overall design and implementation of IPIS were generally successful.

To continue improving the IPIS, the current research could also be extended to cover the integration across the different information systems like Outpatient Information System, Pharmacy Information System, Radiology Information System, and Laboratory Information System. This will gives the IPIS a more complete coordinated health care service.

REFERENCES

[1] Meng-Kin Lim. "Shared Electronic Medical Records". *Health Policy Monitor*, Available: http://www.hpm.org/survey/sg/a4/3, 2004.

[2] *Four public hospitals lauded for their electronic medical records systems*. Available: http://www.channelnewsasia.com/stories/singaporelocalnews/view/1135066/1/.html

[3] *The top 20 solutions most popular EMR software solutions* Available: http://www.capterra.com/infographic-top-20-emr-software-solutions

[4] *Health Center* Available: http://www.eclinicalworks.com/healthcenters.htm

[5] *eClinicalWorks Customer Unity Health Care Receives HIMSS Davies Award* Available: http://www.marketwatch.com/story/eclinicalworks-customer-unity-health-care-receives-himss-davies-award-2013-03-04

[6] *Open Door Family Medical Centers* Available: http://www.eclinicalworks.com/Collateral/Documents/English-US/OpenDoor_CaseStudy_2012_May09.pdf

[7] *eClinicalWorks Customer, Open Door, Receives HIMSS Davies Award* Available: http://www.medicexchange.com/news-article/eclinicalworks-customer-open-door-receives-himss-davies-award.html

[8] Karim Keshavjee et al. “Best Practices in EMR Implementation: A Systematic Review” Available:http://www.competestudy.com/documents/Best_Practices_in_EMR_Implementation.pdf

[9] Reda Chouffani “Top 10 EHR Failure Contributing Factors” Available: http://www.healthcareitinsider.com/ehr/top-10-ehr-failure-contributing-factors/

[10] *SQL Tutorial – Improving Query Performance* Available: http://www.sqlsteps.com/sql-tutorial-improving-query-performance