

# HOSPITAL MANAGEMENT SYSTEM



A Major project report  
in partial fulfillment of requirements of

**Bachelor of Technology**  
in  
**Computer Science & Engineering**  
by

KOTHA RISHEEK REDDY [15K41A0592]

Under the guidance of

**MD.SALLAUDDIN**  
Asst. Professor

submitted to



2018-2019



**SR**  
**Engineering**  
**College**  
Innovation . Creativity . Entrepreneurship

## **DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

### **CERTIFICATE**

This is to certify that the Project Report entitled “**HOSPITAL MANAGEMENT SYSTEM**” is a record of bonafide work carried out by the students **KOTHA RISHEEK REDDY [15K41A0592]** during the academic year 2018-2019 in partial fulfillment of the award of the degree of **Bachelor of Technology in Computer Science & Engineering** by the **Jawaharlal Nehru Technological University, Hyderabad**.

**SUPERVISOR**  
**MD.SALLAUDDIN**  
Asst. Professor

**HEAD OF THE DEPARTMENT**  
**SRINIVAS ALUVALA**  
Asst.Professor

**EXTERNAL EXAMINER**

# ACKNOWLEDGEMENT

I would like to express our deep felt appreciation and gratitude to **Sri Dr. V.MAHESH**, SR ENGINEERING COLLEGE for making available facilities for carrying out our project.

I would like to express our heart-felt gratitude in deep humility to **Sri SRINIVAS ALUVALA**, Asst.Prof and Head, Department of Computer Science & Engineering, for his wholehearted support, Who has provided all the facilities to conduct our project work with immense cooperation and inspiration.

I would like to express our gratitude to our project guide, **Sri MD.SALLAUDDIN**, Asst.Professor in Computer Science & Engineering Engineering for his excellent advice and encouragement. Teacher stood as pillar of strength, right through our project, till the preparation of report and helped, by boosting our morale. So we could surmount the difficulties we faced in our project.

My special thanks to each and every faculty member and programmers in the department for their suggestions and taking care of all software needs.

I would like to express our gratitude towards our parents for their kind cooperation and encouragement which helped us in the completion of this project.

Last but not least, I express our thanks to our colleagues, who gave us their assistance, suggestions and support.

KOTHA RISHEEK REDDY [15K41A0592]

# Contents

<b>List of Figures</b>	<b>iii</b>
<b>List of Abbreviations</b>	<b>iv</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Existing System . . . . .	2
1.2.1 Disadvantage . . . . .	2
1.3 Proposed System . . . . .	2
1.3.1 Advantages . . . . .	2
<b>2 LITERATURE SURVEY</b>	<b>4</b>
2.1 Related Work . . . . .	4
2.2 System Analysis . . . . .	5
2.2.1 Functional and Non-Functional Requirements . . . . .	6
2.2.2 Feasibility Study . . . . .	7
2.2.3 Economic Feasibility . . . . .	7
2.2.4 Technical Feasibility . . . . .	7
2.2.5 Behavioral Feasibility . . . . .	8
<b>3 SYSTEM DESIGN</b>	<b>9</b>
3.1 System Specification . . . . .	9
3.1.1 Hardware Requirements . . . . .	9
3.1.2 Software Requirements . . . . .	9
3.2 UML diagrams . . . . .	9
3.2.1 UseCase Diagram . . . . .	10
3.2.2 Sequence Diagram . . . . .	11
3.2.3 Activity Diagram . . . . .	13
<b>4 IMPLEMENTATION</b>	<b>15</b>
4.1 MODULES . . . . .	15
4.1.1 Description Of Developed System Modules . . . . .	15
4.2 OVERVIEW TECHNOLOGY . . . . .	17
4.2.1 What is HTML ? . . . . .	17
4.2.2 What is CSS ? . . . . .	17
4.2.3 What is JavaScript ? . . . . .	17
4.2.4 What is SQL ? . . . . .	17
4.2.5 What is PHP ? . . . . .	18
4.2.6 What is Codeigniter? . . . . .	18
4.2.7 Model View Controller architecture (MVC) . . . . .	19
4.2.8 SAMPLE CODE . . . . .	20
<b>5 SYSTEM TESTING</b>	<b>25</b>
5.1 Test Cases . . . . .	25
5.2 Test Results . . . . .	26

<b>6</b>	<b>RESULTS</b>	<b>28</b>
6.1	Screenshots of Result . . . . .	28
<b>7</b>	<b>CONCLUSION</b>	<b>30</b>
<b>8</b>	<b>FUTURE SCOPE</b>	<b>31</b>
	<b>BIBLIOGRAPHY</b>	<b>32</b>

# List of Figures

3.1	Use Case Diagram . . . . .	10
3.2	Admin Sequence Diagram . . . . .	11
3.3	Doctor Sequence Diagram . . . . .	12
3.4	Admin Activity Diagram . . . . .	13
3.5	Member Activity Diagram . . . . .	14
4.1	MVC Architecture . . . . .	19
6.1	Dashboard . . . . .	28
6.2	Employee . . . . .	28
6.3	Department . . . . .	29
6.4	Attendance . . . . .	29

# List of Abbreviations

HTML	-	Hyper Text Markup Language
CSS	-	Cascading Style
JS	-	JavaScript
SQL	-	Structured Query Language
PHP	-	Hypertext Preprocessor
HTTP	-	Hypertext Transfer Protocol
XHTML	-	Extensible Hypertext Markup Language
CRUD	-	Create, Read, Update, and Delete
RDBMS	-	Relational Database Management System
IT	-	Information Technology
HMS	-	Hospital Management System
MVC	-	Model View Controller

## **ABSTRACT**

The purpose of the project entitled as “HOSPITAL MANAGEMENT SYSTEM” is to computerize the Front Office Management of Hospital to develop software which is user friendly simple, fast, and cost – effective. It deals with the collection of patient’s information, diagnosis details, etc. Traditionally, it was done manually. The main function of the system is register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully. System input contains patient details, diagnosis details, while system output is to get these details on to the screen. The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The data are well protected for personal use and makes the data processing very fast.



## Chapter 1

# INTRODUCTION

### 1.1 Introduction

Hospital Management System provides the benefits of streamlined operations, enhanced administration & control, superior patient care, strict cost control and improved profitability. HMS is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals. More importantly it is backed by reliable and dependable support.

The project Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. The software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. User can search availability of a doctor and the details of a patient using the id. The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast. Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals.

Hospital Management System is designed for multispeciality 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. Hospital Management System is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and activity-based costing. Hospital Management System 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 helps you manage your processes.

## **1.2 Existing System**

Hospitals currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread throughout the hospital management infrastructure. Often information is incomplete or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

### **1.2.1 Disadvantage**

- Time dependent.
- Data is stored locally.
- Data loss can happen.
- With the increase in database, it will become a massive job to maintain the database.
- Lack of security for the records, anyone disarrange the records of your system.
- To maintain the records of sale and service manually, is a Time-consuming job.
- transaction is done locally.
- Lot of paper works are needed for the safe keeping of the details of tablets.
- Cannot be accessed from anywhere, anytime.

## **1.3 Proposed System**

The Hospital Management System is designed for any hospital to replace their existing manual paper based system. The new system is to control the information of patients. Room availability, staff and operating room schedules and patient invoices. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks .

### **1.3.1 Advantages**

- User friendly interface.
- Fast access to database.
- Search facility.
- Notification facility.
- Separate logins for users.

- Can access from anywhere.
- Cloud based database.
- appointment booking system.
- Can be accessed from anywhere, anytime.
- Data security.

## **Chapter 2**

# **LITERATURE SURVEY**

This chapter summarizes the evaluation of the literature relevant to the Hospital Management System. It examines theories, concepts, approaches, methods and techniques relevant to the project. Similar existing technologies relating to the development the EMS are discussed.

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites.

### **2.1 Related Work**

Hospital Management system refers to the systems and processes at the intersection between EMS and information technology. It merges EMS as a discipline and in particular it's basic EM activities and processes with the information technology field whereas the programming of data processing systems evolved into standardized routines and packages of enterprise resource planning (ERP) software.

An organization or company with a very large number of staff manages a greater volume of data. This activity can be daunting without a more sophisticated tool to store and retrieve data. The various levels of sophistication can be examined by looking at the evolutionary aspects of HM technology. These aspects can be characterized into four stages of development: Paper-based systems, early personal computer technology, electronic databases, and Web-based technology.

The benefits of automation are becoming widely known to HM and other areas of the business. The focus has shifted to automating as many transactions as possible to achieve effectiveness and efficiencies.

The technology of the future will be about speedy access to accurate current information, and reliability to access this information via multiple systems will give organizations a strategic edge. EM

is expected to relinquish its role as sole owner of HM information, so that managers and employees can use this information to solve their own problems using Web-based systems. This new system will not necessarily mean reduction in HM staff.

The new system will enable HM professionals to focus on transforming information into knowledge that can be used by the organization for decision making; it will be about HR and IT working together to leverage this technology.

The two most popular Web-based HM applications used today are self-service for employees and self-service for managers. These applications have enabled companies to shift responsibility for viewing and updating records onto individual employees and have fundamentally changed the manner in which staff acquire information and relate to their departments.

## **2.2 System Analysis**

System Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Here the key question is what all problems exist in the present system? What must be done to solve the problem? Analysis begins when a user or manager begins a study of the program using existing system.

A good analysis model should provide not only the mechanisms of problem understanding but also the frame work of the solution. Thus, it should be studied thoroughly by collecting data about the system. Then the proposed system should be analyzed thoroughly in accordance with the needs.

## **2.2.1 Functional and Non-Functional Requirements**

### **Functional Requirements**

#### **Authentication**

- Login
- Logout

#### **Authorization**

- User role check

#### **Manage departments**

- Create a new department
- Edit any department

#### **Manage doctors**

- Create a new doctor
- Edit any doctor

#### **Manage patients**

- Manage all patients of the hospital
- Delete a patient

#### **Manage Appointments**

- View all approved appointments of the logged in patient.
- Apply for a new appointment

#### **Manage Beds**

- bed allotments

#### **Manage Medicines**

- Create a new medicines
- Edit any medicine

#### **Manage prescriptions**

- Create a new prescription
- Create diagnosis report for a prescription

#### **Manage Invoices**

- Create a new invoice
- View invoices list

### **Non-Functional Requirements**

- **Performance-** Quick Response must be provided by android information system tool.
- **Availability-** The application should be available any time.
- **Usability-** The GUI interface should be interactive. User can retrieve the information from the local database and to invoke web services.
- **Reliability-** Users expect the payroll system to be reliable, a common nonfunctional requirement for most systems. Information entered yesterday in the system should be there today.
- **Flexibility-** Invoice system must be nimble enough to allow for easy changes. Flexibility is an important nonfunctional requirement of a payroll program. As laws and regulations change, the system must be updated as well.

#### **2.2.2 Feasibility Study**

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. key considerations involved in the feasibility analysis are:

#### **2.2.3 Economic Feasibility**

This study is carried out to check the economic impact will have on the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customised products have to be purchased.

#### **2.2.4 Technical Feasibility**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes for the implementing this system.

### **2.2.5 Behavioral Feasibility**

It is a measure of how well a proposed system solves the problems, and takes an advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of the system development. Since the developed system is an interactive system, any user can use the system. The proposed system does not use any additional software during installation period.



## Chapter 3

# SYSTEM DESIGN

### 3.1 System Specification

#### 3.1.1 Hardware Requirements

Processor	: Intel core duo 2.0 or higher
RAM	: 512MB or above
Hard Disk	: 50GB or above
Keyboard	: Standard 102 Keys
Mouse	: 3 Buttons

#### 3.1.2 Software Requirements

Front End	: HTML,CSS,JavaScript,JQuery,BootStrap
Back End	: Php,Sql
Framework	: Codeigniter
OS	: Windows 7/Linux

### 3.2 UML diagrams

The Unified Modeling Language is fast becoming a required skill for virtually anybody involved in a software project. Requirements analysts, software developers, architects, UI designers, database professionals, testers and project managers are increasingly being asked to create and consume specifications written in UML.

### 3.2.1 UseCase Diagram

A use case defines a goal-oriented set of interactions between external users and the system under consideration or development. Thus a Use Case Scenario is a description that illustrates, step by step, how a user is intending to use a system, essentially capturing the system behavior from the user's point of view.

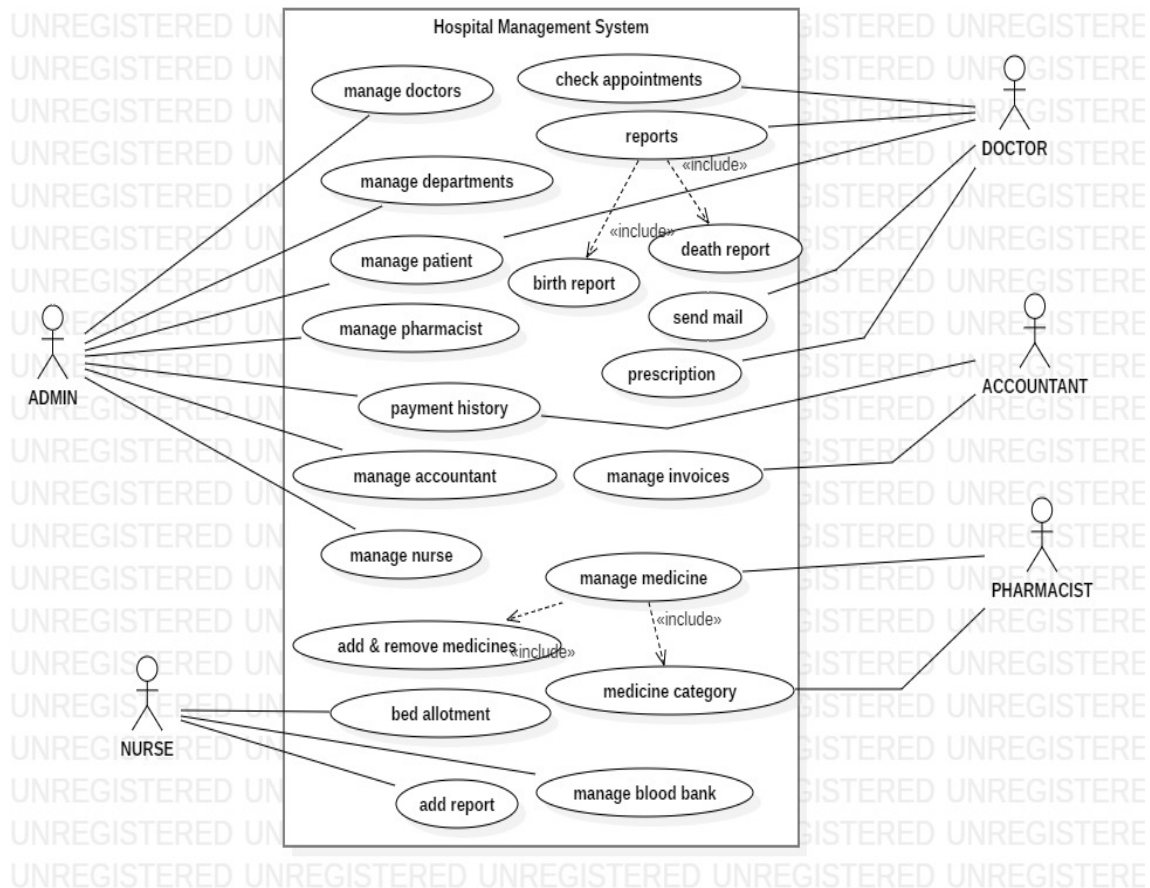


Figure 3.1: Use Case Diagram

**Description:** Use cases are used during the analysis phase of a project to identify and partition system functionality. They separate the system into *actors* and *use cases*.

### 3.2.2 Sequence Diagram

Sequence diagrams help in the identification of a detailed level of the operations required to implement the functionality depicted by a use case model.

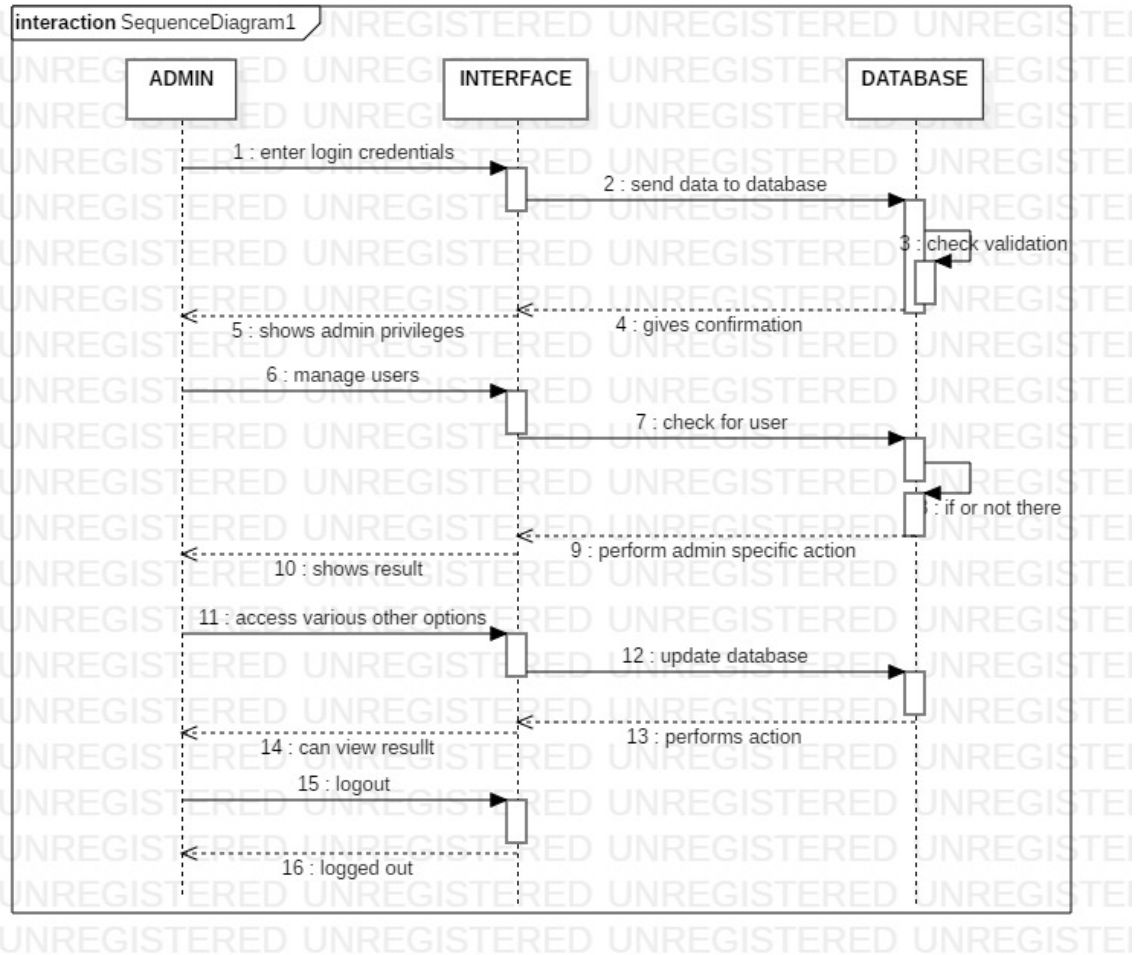


Figure 3.2: Admin Sequence Diagram

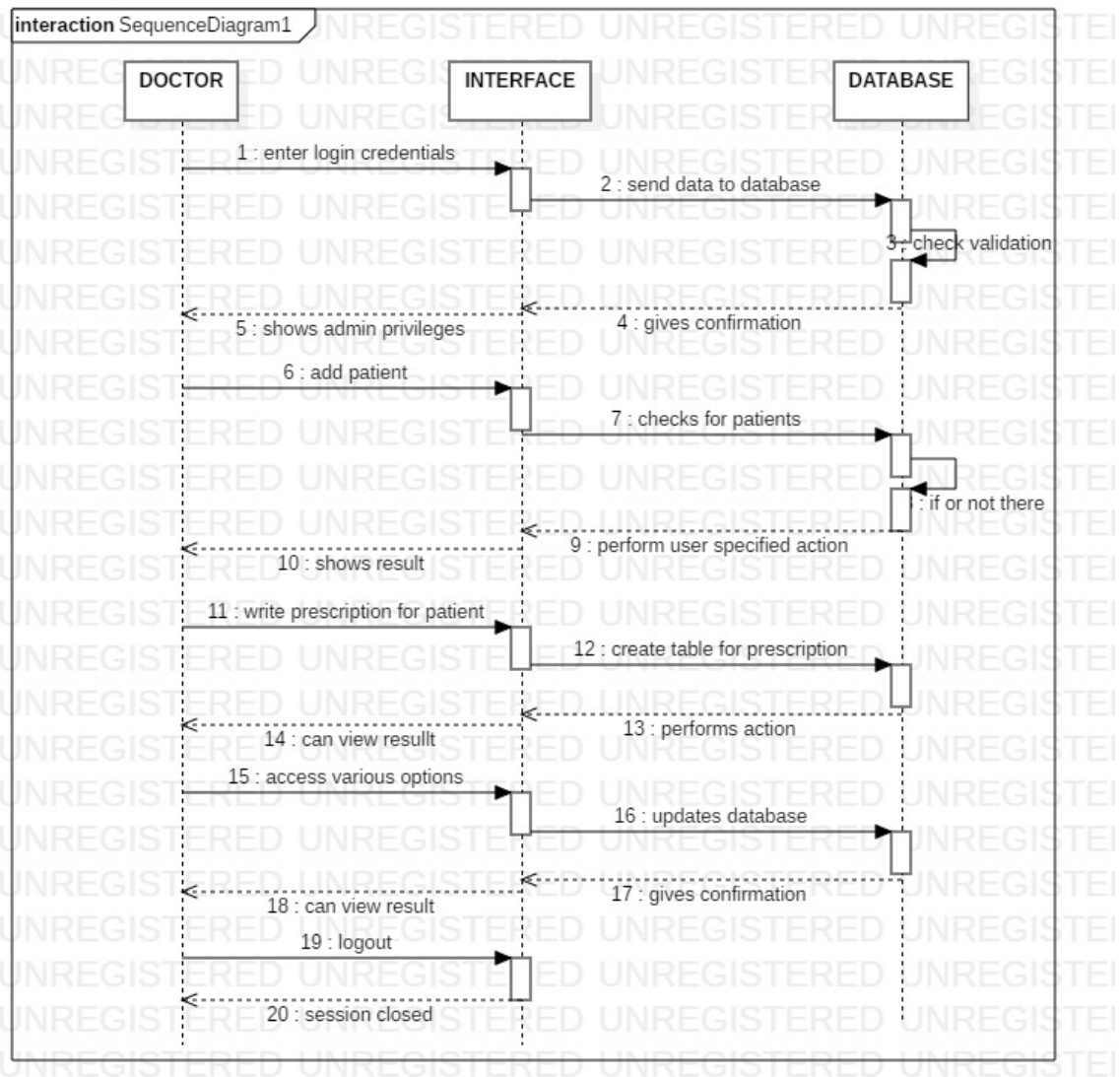


Figure 3.3: Doctor Sequence Diagram

**Description:** A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams typically are associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

### 3.2.3 Activity Diagram

Activity Diagrams are used to model different aspects of a system. The following activity diagram is used to model the leave application function.

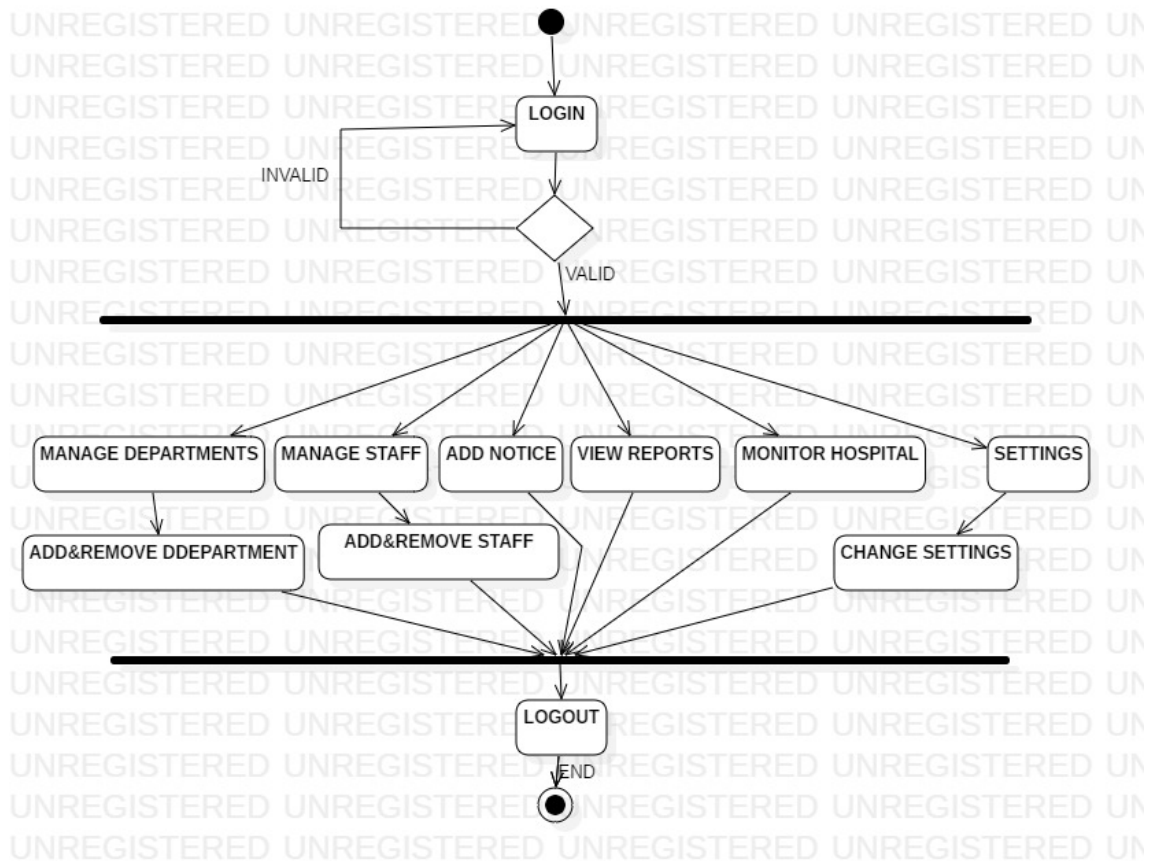


Figure 3.4: Admin Activity Diagram

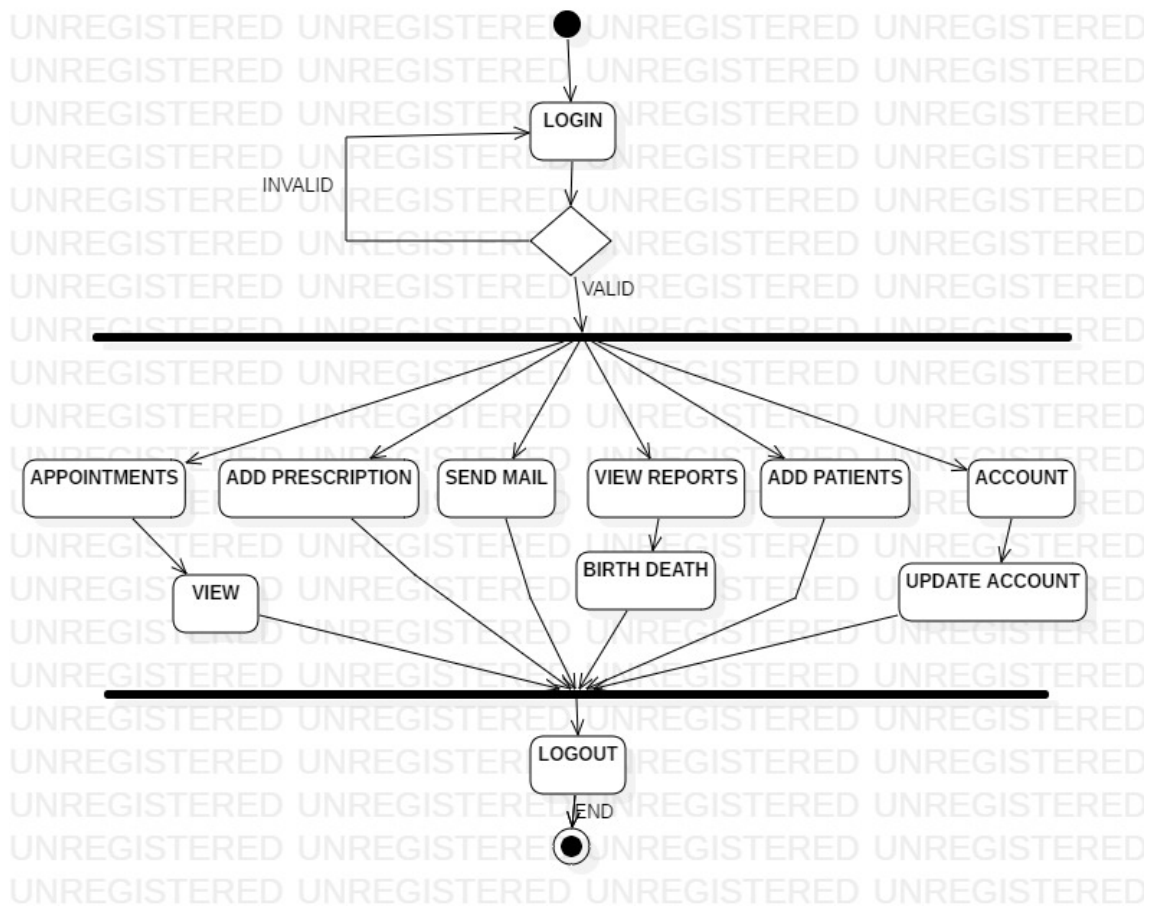


Figure 3.5: Member Activity Diagram

**Description:** Activity diagrams are graphical representations of workflows of step-wise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

## Chapter 4

# IMPLEMENTATION

This chapter will explore the different aspects concerned with the implementation of the developed system. This project was concerned with the development and implementation the employee management system. We began with analysis of the current and proposed systems, the design of the system to be developed, and in this chapter we shall deal with implementation of the developed system.

## 4.1 MODULES

The developed system encompasses various activities associated with managing employee information. The system has dual user mode namely Administrator and Employee.

- Administrator Module.
- Doctor Module.
- Patient Module.
- Accountant Module.
- Pharmacist Module

### 4.1.1 Description Of Developed System Modules

- Dashboard- The dashboard provides a full overview of the whole process.
- Doctor- New doctor can be added and doctor list with full description can be generated from doctor tab.
- Department- The department category adds the various divisions and departments in the organization with designated positions under the departments
- Patient- New patient can be added and patient list with full description can be generated from patient tab.
- Leave- The leave management portal is designed to approve or decline the leave request generated by the employee.
- Blood Bank- All the donors are listed in this tab of blood bank.
- Noticeboard- The app enables to create, edit, delete and send notices for the informed convenience in human resources management.
- Message- Doctors can directly contact with the patients view email.

- Account- The profile information can be edited by clicking at “Account” tab listed last at Dashboard. The information can be edited and saved using this “Update Profile” listed under “account”.
- Bed allotment- a patient can be allotted with a bed which has a number. A list of beds is shown in this tab. vacant beds as well as allotted.
- Invoice - Invoice can be prepared by accountant while discharging a patient.
- Medicines - Pharmacist takes care of management of medicines.
- Reports - Birth, death, operation reports can be generated and viewed;



## 4.2 OVERVIEW TECHNOLOGY

### 4.2.1 What is HTML ?

HTML stands for Hyper Text Markup Language. It is used to design web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages and the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. HTML is a markup language which is used by the browser to manipulate text, images and other content to display it in required format. HTML was created by Tim Berners-Lee in 1991.

### 4.2.2 What is CSS ?

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects. CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

### 4.2.3 What is JavaScript ?

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

### 4.2.4 What is SQL ?

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database. SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDBMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

#### **4.2.5 What is PHP ?**

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general purpose scripting language that is especially suited for web development and HTML can be embedded into it. The developer could use PHP and HTML to generate the homepage. Once a visitor accesses the index page, the server will execute the PHP command and send the outcomes of implementation to the visitor's browser, however, the difference is that PHP is open source and supports most of the popular platforms and it can be run on a Windows platform and multiple versions of UNIX. PHP consumes few resources, as a part of the IIS server, PHP does not need to call an external binary code and the server does not bear any additional burden. In addition to operating the page, PHP can cooperate with HTTP. There is no need to have a very special development environment with PHP, the block starts and ends with the tags `<?php` and `?>`. The PHP programming language is similar to Pascal. There is no need to define any variables before using them, and it is very simple to establish the array and the Hash. PHP also has some object-oriented features which are provided to support and to organizing and packaging the source code.

#### **4.2.6 What is Codeigniter?**

CodeIgniter is an application development framework, which can be used to develop websites, using PHP. It is an Open Source framework. It has a very rich set of functionality, which will increase the speed of website development work. If you know PHP well, then CodeIgniter will make your task easier. It has a very rich set of libraries and helpers. By using CodeIgniter, you will save a lot of time, if you are developing a website from scratch. Not only that, a website built in CodeIgniter is secure too, as it has the ability to prevent various attacks that take place through websites.

#### 4.2.7 Model View Controller architecture (MVC)

Model-View-Controller is a software architecture, or design pattern, that is used in software engineering, whose fundamental principle is based on the idea that the logic of an application should be separated from its presentation. MVC is a software approach that separates application logic from presentation. In practice, it permits your web pages to contain minimal scripting since the presentation is separate from the PHP scripting.

- The Model represents your data structures. Typically your model classes will contain functions that help you retrieve, insert, and update information in your database.
- The View is the information that is being presented to a user. A View will normally be a web page, a page fragment like a header or footer.
- The controller handles the model and view layers to work together. The controller receives a request from the client, invokes the model to perform the requested operations and sends the data to the View. The view formats the data to be presented to the user, in a web application as an html output.

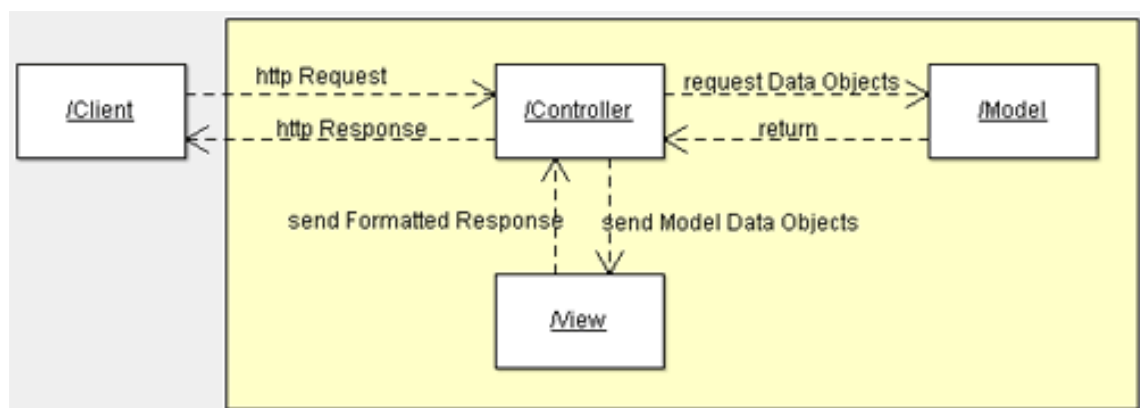


Figure 4.1: MVC Architecture

#### 4.2.8 SAMPLE CODE

*index.php*

```
<?php

if (!defined('BASEPATH'))
exit('No direct script access allowed');

/*
 * @author : Joyonto Roy
 * 30th July, 2014
 * Creative Item
 * www.creativeitem.com
 * http://codecanyon.net/user/joyontaroy
 */

class Login extends CI_Controller {

function __construct() {
parent::__construct();
$this->load->model('crud_model');
$this->load->database();
/* cache control */
$this->output->set_header('Last-Modified: ' .
gmdate("D, d M Y H:i:s") . ' GMT');
$this->output->set_header('Cache-Control: no-store,
no-cache, must-revalidate, post-check=0, pre-check=0');
$this->output->set_header('Pragma: no-cache');
$this->output->set_header("Expires: Mon, 26 Jul 2010 05:00:00 GMT");
}

//Default function, redirects to logged in user area
public function index() {

if ($this->session->userdata('admin_login') == 1)
redirect(base_url() . 'index.php?admin/dashboard', 'refresh');

else if ($this->session->userdata('doctor_login') == 1)
redirect(base_url() . 'index.php?doctor', 'refresh');

else if ($this->session->userdata('patient_login') == 1)
redirect(base_url() . 'index.php?patient', 'refresh');
```

```

else if ($this->session->userdata('nurse_login') == 1)
redirect(base_url() . 'index.php?nurse', 'refresh');

else if ($this->session->userdata('pharmacist_login') == 1)
redirect(base_url() . 'index.php?pharmacist', 'refresh');

else if ($this->session->userdata('laboratorist_login') == 1)
redirect(base_url() . 'index.php?laboratorist', 'refresh');

else if ($this->session->userdata('accountant_login') == 1)
redirect(base_url() . 'index.php?accountant', 'refresh');

else if ($this->session->userdata('receptionist_login') == 1)
redirect(base_url() . 'index.php?receptionist', 'refresh');

$this->load->view('backend/login');
}

//Ajax login function
function ajax_login() {
$response = array();

//Recieving post input of email, password from ajax request
$email = $_POST["email"];
$password = $_POST["password"];
$response['submitted_data'] = $_POST;

//Validating login
$login_status = $this->validate_login($email, $password);
$response['login_status'] = $login_status;
if ($login_status == 'success') {
$response['redirect_url'] = $this->session->userdata('last_page');
}

//Replying ajax request with validation response
echo json_encode($response);
}

//Validating login from ajax request
function validate_login($email = '', $password = '') {
$credential = array('email' => $email, 'password' => sha1($password));

```

```

// Checking login credential for admin
$query = $this->db->get_where('admin', $credential);
if ($query->num_rows() > 0) {
    $row = $query->row();
    $this->session->set_userdata('admin_login', '1');
    $this->session->set_userdata('login_user_id', $row->admin_id);
    $this->session->set_userdata('name', $row->name);
    $this->session->set_userdata('login_type', 'admin');
    return 'success';
}

$query = $this->db->get_where('doctor', $credential);
if ($query->num_rows() > 0) {
    $row = $query->row();
    $this->session->set_userdata('doctor_login', '1');
    $this->session->set_userdata('login_user_id', $row->doctor_id);
    $this->session->set_userdata('name', $row->name);
    $this->session->set_userdata('login_type', 'doctor');
    return 'success';
}

$query = $this->db->get_where('patient', $credential);
if ($query->num_rows() > 0) {
    $row = $query->row();
    $this->session->set_userdata('patient_login', '1');
    $this->session->set_userdata('login_user_id', $row->patient_id);
    $this->session->set_userdata('name', $row->name);
    $this->session->set_userdata('login_type', 'patient');
    return 'success';
}

$query = $this->db->get_where('nurse', $credential);
if ($query->num_rows() > 0) {
    $row = $query->row();
    $this->session->set_userdata('nurse_login', '1');
    $this->session->set_userdata('login_user_id', $row->nurse_id);
    $this->session->set_userdata('name', $row->name);
    $this->session->set_userdata('login_type', 'nurse');
    return 'success';
}

```

```

$query = $this->db->get_where('pharmacist', $credential);
if ($query->num_rows() > 0) {
    $row = $query->row();
    $this->session->set_userdata('pharmacist_login', '1');
    $this->session->set_userdata('login_user_id', $row->pharmacist_id);
    $this->session->set_userdata('name', $row->name);
    $this->session->set_userdata('login_type', 'pharmacist');
    return 'success';
}

$query = $this->db->get_where('laboratorist', $credential);
if ($query->num_rows() > 0) {
    $row = $query->row();
    $this->session->set_userdata('laboratorist_login', '1');
    $this->session->set_userdata('login_user_id', $row->laboratorist_id);
    $this->session->set_userdata('name', $row->name);
    $this->session->set_userdata('login_type', 'laboratorist');
    return 'success';
}

$query = $this->db->get_where('accountant', $credential);
if ($query->num_rows() > 0) {
    $row = $query->row();
    $this->session->set_userdata('accountant_login', '1');
    $this->session->set_userdata('login_user_id', $row->accountant_id);
    $this->session->set_userdata('name', $row->name);
    $this->session->set_userdata('login_type', 'accountant');
    return 'success';
}

$query = $this->db->get_where('receptionist', $credential);
if ($query->num_rows() > 0) {
    $row = $query->row();
    $this->session->set_userdata('receptionist_login', '1');
    $this->session->set_userdata('login_user_id', $row->receptionist_id);
    $this->session->set_userdata('name', $row->name);
    $this->session->set_userdata('login_type', 'receptionist');
    return 'success';
}

return 'invalid';
}

```

```

/*      * *DEFAULT NOR FOUND PAGE***** */

function four_zero_four() {
$this->load->view('four_zero_four');
}

/*      * *RESET AND SEND PASSWORD TO REQUESTED EMAIL*** */

function reset_password() {
$account_type = $this->input->post('account_type');
if ($account_type == "") {
redirect(base_url(), 'refresh');
}
$email = $this->input->post('email');
$result = $this->email_model->password_reset_email
($account_type, $email); //SEND EMAIL ACCOUNT OPENING EMAIL
if ($result == true) {
$this->session->set_flashdata('flash_message',
    get_phrase('password_sent'));
} else if ($result == false) {
$this->session->set_flashdata('flash_message',
    get_phrase('account_not_found'));
}

redirect(base_url(), 'refresh');
}

/*      * *****LOGOUT FUNCTION ***** */
function logout() {
$this->session->unset_userdata();
$this->session->sess_destroy();
$this->session->set_flashdata('logout_notification', 'logged_out');
redirect(base_url(), 'refresh');
}
}

```



## **Chapter 5**

# **SYSTEM TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### **5.1 Test Cases**

#### **Unit Testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

#### **Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

#### **Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional

testing is centered on the following items:

- Valid Input: identified classes of valid input must be accepted.
- Invalid Input: identified classes of invalid input must be rejected.
- Functions: identified functions must be exercised.
- Output: identified classes of application outputs must be exercised.
- Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

## **System Testing**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

## **5.2 Test Results**

### **Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software life cycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

### **Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

### **Test objectives:**

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

**Features to be tested**

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

**Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or one step up–software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

## Chapter 6

# RESULTS

### 6.1 Screenshots of Result

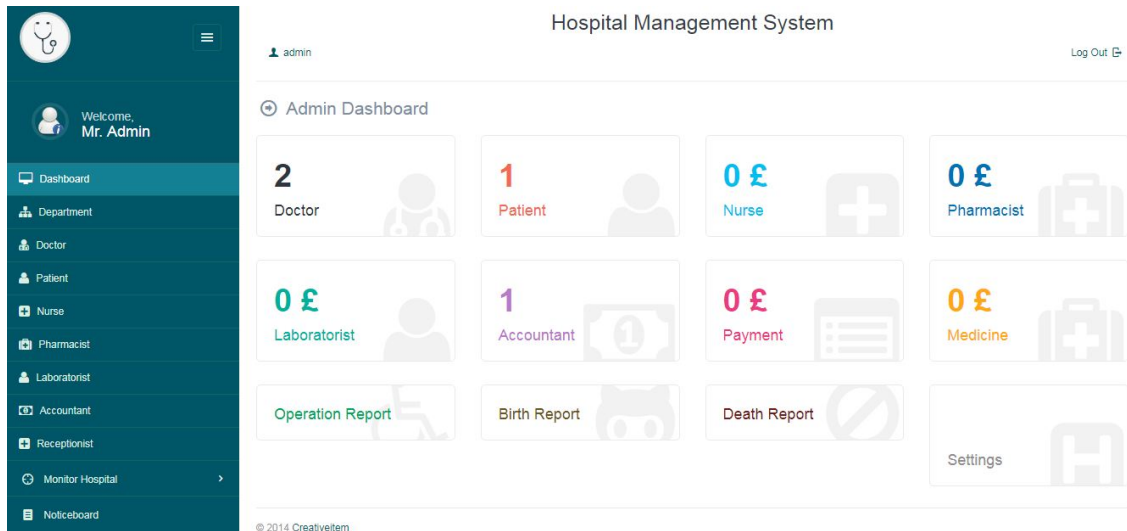


Figure 6.1: Dashboard  
Showing admin Dashboard after login

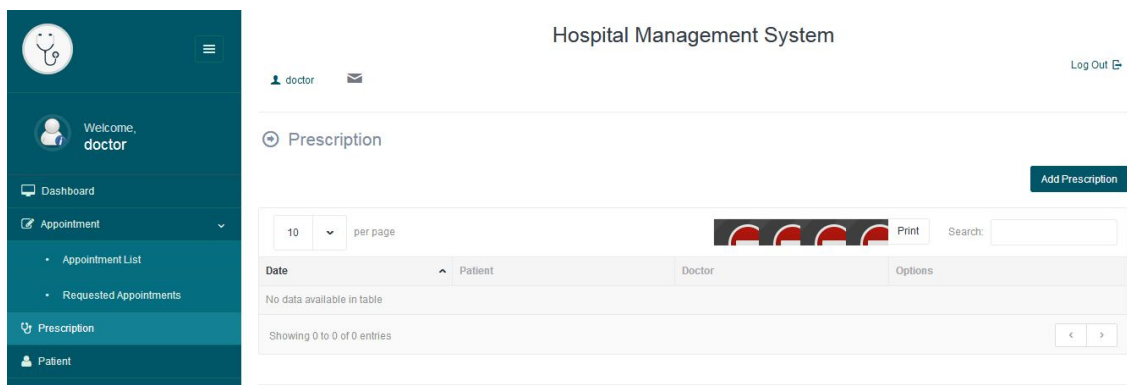


Figure 6.2: Employee  
Showing doctor module

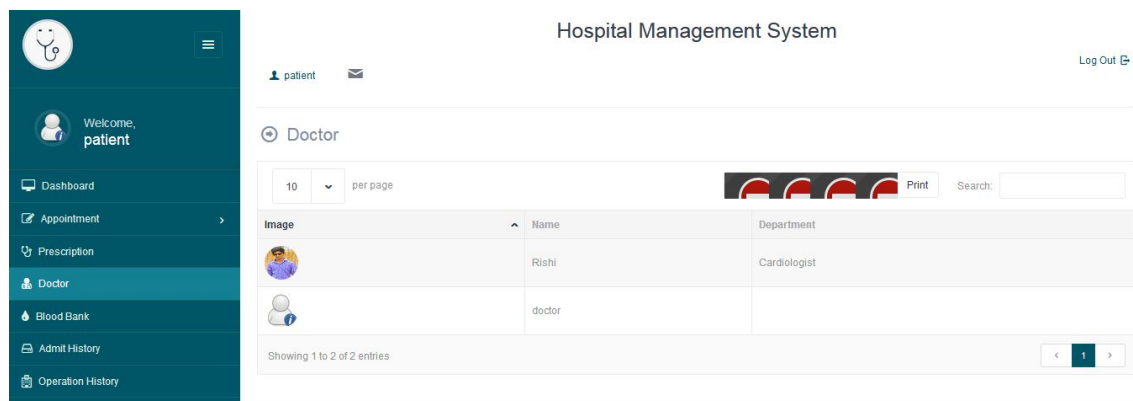


Figure 6.3: Department  
Showing patients dashboard

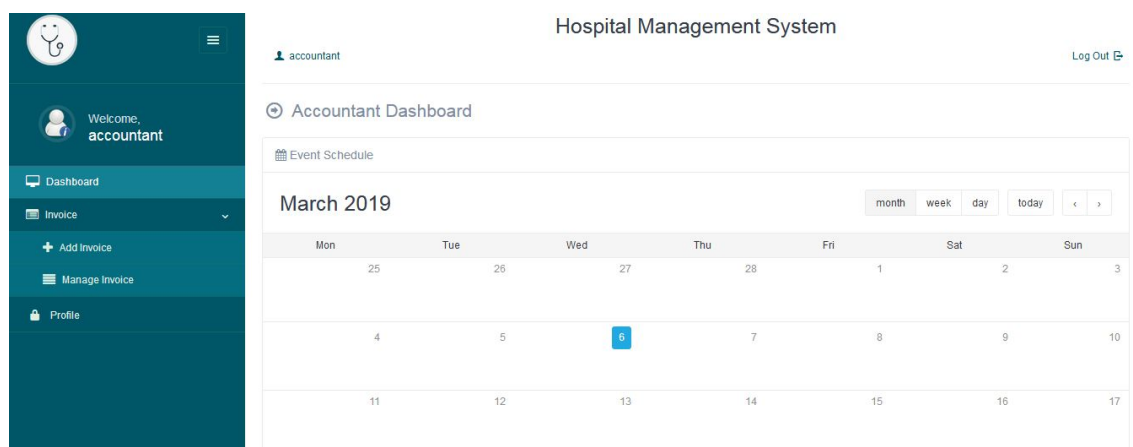


Figure 6.4: Attendance  
Showing accountant module

**Chapter 7****CONCLUSION**

The project Hospital Management System (HMS) is for computerizing the working in a hospital. The software takes care of all the requirements of an average hospital and is capable to provide easy and effective storage of information related to patients that come up to the hospital. It generates test reports; provide prescription details including various tests, diet advice, and medicines prescribed to patient and doctor. It also provides injection details and billing facility on the basis of patient's status whether it is an indoor or outdoor patient. The system also provides the facility of backup as per the requirement.

This project has been a rewarding experience in more than one way. The entire project work has enlightened us in the following areas.

- We have gained an insight into the working of the HOSPITAL. This represents a typical real world situation.
- Our understanding of database design has been strengthened this is because in order to generate the final reports of database designing has to be properly followed.
- Scheduling a project and adhering to that schedule creates a strong sense of time management.
- Sense of teamwork has developed and confidence of handling real life project has increased to a great extent.
- Initially, there were problem with the validation but with discussions, we were to implement validations.

## Chapter 8

### FUTURE SCOPE

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

- We can add printer in future.
- We can give more advance software for Hospital Management System including more facilities.
- We will host the platform on online servers to make it accessible worldwide.
- Integrate multiple load balancers to distribute the loads of the system.
- Create the master and slave database structure to reduce the overload of the database queries
- Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers.

The above mentioned points are the enhancements which can be done to increase the applicability and usage of this project. Here we can maintain the records of employee and salary. Also, as it can be seen that now-a-days the players are versatile, i.e. so there is a scope for introducing a method to maintain the Hospital Management System. Enhancements can be done to maintain all the employee, salary, leaves, attendance, registration.

We have left all the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them. In the last we would like to thank all the persons involved in the development of the system directly or indirectly. We hope that the project will serve its purpose for which it is developed there by underlining success of process.

# Bibliography

- [1] Ivan Bayross, SQL, PL/SQL programming language of Oracle, Second Edition, BPB Publication.
- [2] Ian Somerville, Software Engineering, Third Edition, Pearson Education.
- [3] ] Elmarsri and Navathe, Fundamentals of Database System (Third Edition), Addison Wesley.
- [4] HTML Elements. W3schools. Retrieved 3/16/15.
- [5] PHP Manual: [www.php.net](http://www.php.net)
- [6] PHP for Windows: [php.net](http://php.net). Retrieved 10/29/13. <https://wikipedia.org/wiki/PHP>
- [7] <https://wikipedia.org/wiki/MySQL>
- [8] [http://bw.businessworld.in/PDF\\_upload/Indian\\_Pharma.pdf](http://bw.businessworld.in/PDF_upload/Indian_Pharma.pdf)
- [9] <https://contribute.jquery.org/documentation/>