R.M.D. ENGINEERING COLLEGE

**(An Autonomous Institution)**

**R.S.M Nagar, Kavaraipettai, Gummidipoondi Taluk, Thiruvallur District, Tamil Nadu- 601206**

Affiliated to Anna University, Chennai / Approved by AICTE, New Delhi/Accredited by NAAC

An ISO 9001:2015 Certified Institution / All the Eligible UG Programs are accredited by NBA, New Delhi

**DEPARTMENT OF INFORMATION TECHNOLOGY**

s

**21IT413**

**INTERNSHIP**

**HOSPITAL MANAGEMENT SYSTEM**

**SURIYA V - 111521203049**

**PALLA.SUMANTH - 111521203037**

**MONISH KUMAR S - 111521203301**

### 21IT401 SOFTWARE ENGINEERING LABORATORY OBJECTIVES:

* To understand the software engineering methodologies for project development.
* To gain knowledge about open source tools for Computer Aided Software Engineering.
* To develop an efficient software using case tools.

### SOFTWARE REQUIRED:

Open source Tools: Star UML / UMLGraph / Topcased

Prepare the following documents for each experiment and develop the software using softwareengineering methodology.

1. **Problem Analysis and Project Planning -**Thorough study of the problem – Identify Projectscope, Objectives and Infrastructure.
2. **Software Requirement Analysis -** Describe the individual Phases/modules of the project andIdentify deliverables.
3. **Data Modelling -** Use work products – data dictionary, use case diagrams and activitydiagrams, build and test class diagrams, sequence diagrams and add interface to class diagrams.
4. **Software Development and Debugging** – implement the design by coding
5. **Software Testing** - Prepare test plan, perform validation testing, coverage analysis, memoryleaks, develop test case hierarchy, Site check and site monitor.

### SAMPLE EXPERIMENTS:

**Academic domain**

1. Course Registration System
2. Patient marks analyzing system

### Railway domain

1. Online ticket reservation system
2. Platform assignment system for the trains in a railway station

### Medicine domain

1. Expert system to prescribe the medicines for the given symptoms
2. Remote computer monitoring

### Finance domain

1. ATM system
2. Stock maintenance

### Human Resource management

1. Quiz System
2. E-mail Client system.

**INDEX**

|  |  |
| --- | --- |
| **SL.NO** | **NAME OF THE EXPERIMENT** |
|  | **HOSPITAL MANAGEMENT SYSTEM** |
| 1 | Problem Analysis  1(a)Problem Statement |
|  | 1(b)Project Planning |
| 2 | Software Requirement Analysis |
| 3 | Modeling  3(a)Design 3(b)Data Dictionary |
| 4 | Implementation |
| 5 | Testing - Test Cases |
| 6 | Documentation |

# Ex.No 1(a) PROBLEM ANALYSIS

**Problem Statement**

Computerized Hospital Management system is used to monitor and control the transactions in a hopital. It needs to maintain the record of new books and retrieve the details of patients available in the hospital . Should be able to perform basic operations in a hospital like adding new patients, new doctors, and updating new appointments, searching patient record and doctor recorsd and provide facility to print bill after discharge.

# Analysis

Hospital Management System should allow the user to add new patients and appointments. The system has three categories of users: Administrator, Patient and Doctor.

Administrator should be able to:

* Register new doctor
* View the doctor record
* Update the doctor record
* View patient record
* Update patient details
* Admit new patient
* Approve patient
* Discharge patient
* Generate bill
* View the available appointments
* Book new appointments
* Approve the awaiting appointments

Patient should be able to:

* Book and view his/her appointments
* View the doctors treating him/her
* View the bill

Doctor should be able to:

* View his patients (present and discharged)
* View and delete appointments

# Feasibility study

### Technical feasibility

The Hospital Management System (LMS) runs with a minimum system resources:

* Python
* Django
* Sqlite3
* Html

Above said system resources are available as open source. Hence it is feasible to develop HMS in this environment.

### Operational feasibility

As the system has HTML based GUI no special skill set is required for working with the system, hence it is operationally feasible.

### Economic feasibility

As the LMS requires minimum system resources, hence it is economically feasible.

**Ex.No 1(b) PROJECT PLANNING**

# Overview

Automation of Hospital Management System is to handle the entire activity of a . The system keeps track of all the information about the books and their complete details. The system contains a database where all the information will be stored. The system is user-friendly.

# Goals and Scope

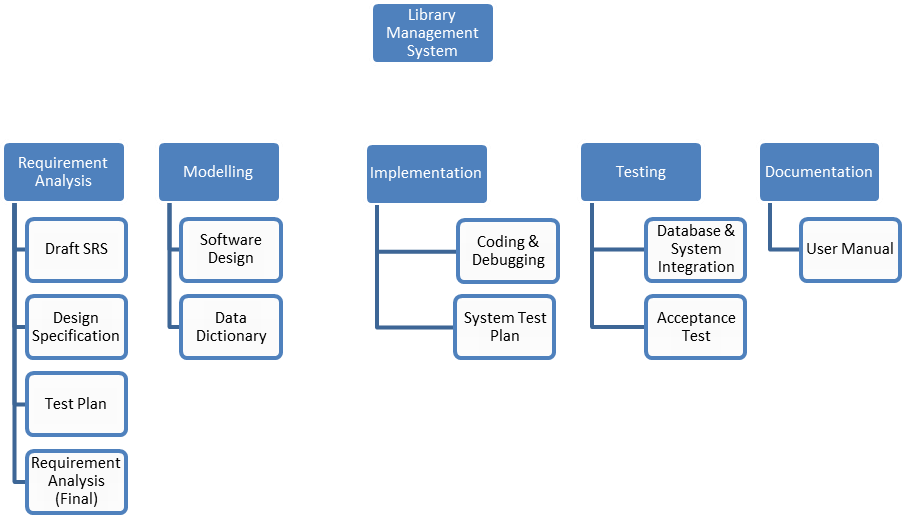
**Goal :** To automate the Hospital Information System with the following functional goals

* + 1. Login
    2. Adding Patient/Doctor Details.
    3. Updating Patient/Doctor Details
    4. Deleting Patient/Doctor Details
    5. Generating Bill
    6. Viewing/Searching for Details.

**Scope :** The system accepts the General Hospital Transactions of appointment issue, doctor record and for the members. Different areas where we can use these applications are:

1. Any hospital institutes can make use of it for providing information about doctors, appointments of the available patients.
2. The system would provide basic set of features to add/update patients, add/update doctors, and manage appointments and generate bills.

# Schedule and Budget Work Breakdown Structure



**Schedule and Milestones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestones** | **Description** | **Milestone Criteria** | **Planned week** |
| M0 | Problem Analysis |  | 1st week |
|  |  | Problem statement, Analysis, Feasibility Study |  |
| M1 | Project Planning |  | 2nd week |
|  |  | Scope and concept described |  |
| M2 | Requirement Analysis |  | 2nd and 3rd week |
|  |  | Draft SRS, Design Specification, Test Plan, Requirement Analysis (Final) |  |
| M3 | Study of UML Notations |  | 3rd week |
|  |  | Architecture reviewed and stable |  |
| M4 | Modeling |  | 4th week |
|  |  | Software Design, Data Dictionary |  |

**Problem Analysis**

**Create Data Dictionary**

**Problem Planning**

**Require- ment Analysis**

**Model- ing**

**Coding**

**UML**

**design**

**Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestones** | **Description** | **Milestone Criteria** | **Planned week** |
| M5 | Implementation |  | 5th week |
|  |  | Coding of functionality, Debugging, System Test Plan. |  |
| M6 | Testing |  | 6th week |
|  |  | Database & System Integration, Acceptance Testing |  |
| M7 | Documentation |  | 7th week |
|  |  | User Manual |  |

# Budget

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Category** | **Budget for Period in kUS$** | | | | | |
| **M0-M1** | **M1-M2** | **M2-M3** | **M3-M4** | **M4-M5** | **M5-M6** |
| Human Resources (internal) |  |  |  |  |  |  |
| Human Resources (external) |  |  |  |  |  |  |
| Purchases (COTS) |  |  |  |  |  |  |
| Equipment |  |  |  |  |  |  |
| Premises |  |  |  |  |  |  |
| Tools |  |  |  |  |  |  |
| Travel costs |  |  |  |  |  |  |
| Training |  |  |  |  |  |  |
| Review activities |  |  |  |  |  |  |
| Other |  |  |  |  |  |  |
| Total | 1 | 1 | 2 | 5 | 2 | 1 |
| **Total cumulated** | **1** | **2** | **4** | **9** | **11** | **12** |

For a detailed list of costs of all resources see <document> [x].

# Development Process

**Risk Management**

Unexpected Holidays, Non availability of computer resources, Absence of Human Resource are the identified risks for not meeting the deadlines. Additional efforts need to put in by the human resources to complete the work within the deadline by the way of working after working hours.

# Delivery Plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Ident.** | **Deliverable** | **Planned Date** | **Receiver** |
| D1 | Analysis and Feasibility Report | 1st week | Client |
| D2 | Project Plan | 2nd week | Client |
| D3 | SRS | 3rd week | Client |
| D4 | Design | 4th week | Client |
| D5 | Test Plan | 5th week | Client |
| D6 | Code | 6th week | Client |
| D7 | Test Report | 6th week | Client |

**Ex.No.2 SOFTWARE REQUIREMENT ANALYSIS**

# Software Requirement Specification (SRS)

1. **Introduction**

Hospital Management system is used to monitor and control the transactions in a hospital. It needs to maintain the record of new patients and retrieve the details of patients available in the hospital. Should be able to perform basic operations in a hospital like adding new doctors/ patients, updating doctor/patient information, searching for doctors/patients and provide facility to book appointments and view bill. The report generation facility should allow generating various reports.

Booking appointments or viewing the available appointments at the Hospital is currently done manually. This project is specifically designed for the use of receptionists and hospital users. It is especially useful for an corporate hospitals where modifications in the content can be done easily according to the requirements.

# Purpose of the requirements document

The purpose of this document is to analyze and elaborate on the high-level needs and features of the Hospital System**.** It focuses on the capabilities and facilities provided by a Hospital .The main Objective of this document is to illustrate the requirements of Hospital Management System. This document gives the detailed description of both functional and non-functional requirements proposed by the clients.

This document defines and describes the operations, interfaces, performance, and quality assurance requirements of the Hospital System. The document also describes the nonfunctional requirements such as the user interfaces. It also describes the design constraints that are to be considered when the system is to be designed, and other factors necessary to provide a complete and comprehensive description of the requirements for the software.

# Scope of the product

The Software Requirements Specification captures all the requirements in a single document. The Hospital Management System that is to be developed provides the Administrator, patient and doctors with appointment information, and many other facilities.

The system accepts the General Hospital Transactions of appointment booking, view and pay for the bill. The system would provide basic set of features to add/update doctors,

add/update patients, and manage bills in specifications for the systems based on the client's statement of need.

# Definitions, acronyms and abbreviations

PHP- Hypertext Preprocessor SQL- Structured Query Language GUI- Graphical User Interface

HMS – Hospital Management System

# References

1. Roger s Pressman ,”Software Engineering -a practitioner's approach” 7th edition
2. Ian sommerville , “software engineering “ 9th edition

# 1.5. Overview of the remainder of the document

The SRS will provide a detailed description of the Hospital Management System. This document will provide the outline of the requirements, overview of the characteristics and constraints of the system.

Section 2 of this document provides the General description such as Product perspective, Product functions and the characteristics of the user’s of this product. Section 3 describes the Specific requirements which cover the functional, non-functional and interface requirements. This is obviously the most substantial part of the document but because of the wide variability in organizational practice, it is not appropriate to define a standard structure for this section. The requirements may document external interfaces, describe system functionality and performance, specify logical database requirements, design constraints, emergent system properties and quality characteristics.

# General description Product perspective

The Hospital Management System is a package to be used by hospitals to improve the efficiency of Administrator and Users. The system provides doctors catalog and information to members and helps them decide on the doctors to treat from the hospital. The receptionist can keep the appointment catalog updated all the time so that the patients get the updated information all the time.

# Product functions

The Hospital Management System provides real time information about the appointmets available in the

Hospital and the user information. The Product functions are more or less the same as described

in the product perspective. The functions of the system include the system providing different type of services based on the type of users [Administrator/User].

1. Administrator should be able to add doctor ,book appointment and generate bill.
   * Add doctor
   * book appointments
   * Generate bill
2. Doctor should be able to access patient under the respective doctor and discharge summary

Of the patient

1. Patient should able to view the corresponding doctor for their respective problem and get the

Doctor information.

# User characteristics

The users of the system are administrators, Patient and Doctor members, the administrator who maintain the system. The members are assumed to have basic knowledge of the computers . The administrators of the system to have more knowledge of the internals of the system and is able to rectify the small problems that may arise due to disk crashes, power failures and other catastrophes to maintain the system. The proper user interface, user’s manual, online help and the guide to install and maintain the system must be sufficient to educate the users on how to use the system without any problems.

# General constraints

* + The information of all the users must be stored in a database that is accessible by the Hospital Management System.

.

* + The Hospital Management System is connected to the server computer and is running all 24 hours a day.

# Assumptions and dependencies

* + The users have sufficient knowledge of computers.
  + The users know the English language, as the user interface will be provided in English
  + The product can access the college patient database

# Specific requirements Functional Requirements

This section describes in detail all the functional requirements.

(It shows what the system can do)

* 1. Login
  2. Appoint patient
  3. Add doctors
  4. Get patient information
  5. Appoint patient to respective doctor
  6. Generating discharge summary
  7. Generate bill

# Non- Functional Requirements

### Usability

* The system is user friendly and self-explanatory.

### Reliability

The system has to be very reliable due to the importance of data and the damages incorrect or incomplete data can do.

### Availability

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

### Mean Time Between Failures (MTBF)

The system will be developed in such a way that it ***may*** fail once in a year.

### Mean Time to Repair (MTTR)

Even if the system fails, the system will be recovered back up within an hour or less.

### Accuracy

The accuracy of the system is limited by the accuracy of the speed at which the employees of the hospital and users of the hospital use the system.

### Maximum Bugs or Defect Rate

Not specified.

### Access Reliability

The system shall provide 100% access reliability.

### Performance Response Time

The system shall respond to the member in not less than two seconds from the time of the request

submittal. The system shall be allowed to take more time when doing large processing jobs.

The requirements may document external interfaces, describe system functionality and performance, specify logical database requirements, design constraints, emergent system properties and quality characteristics.

# Hardware and software requirements

## Hardware Interfaces

Processor: Pentium or Higher. RAM: 312MB or Higher.

## 3.3..2. Software Interfaces

Operating System: Unix, Linux, Mac, Windows etc.. Development tool: Django ,python

Data Base: Sqlite3

# 3.4 External Interfaces

### User Interfaces

The user-interface of the system shall be designed as shown in the user-interface prototypes.

# Appendices

1. **Index**

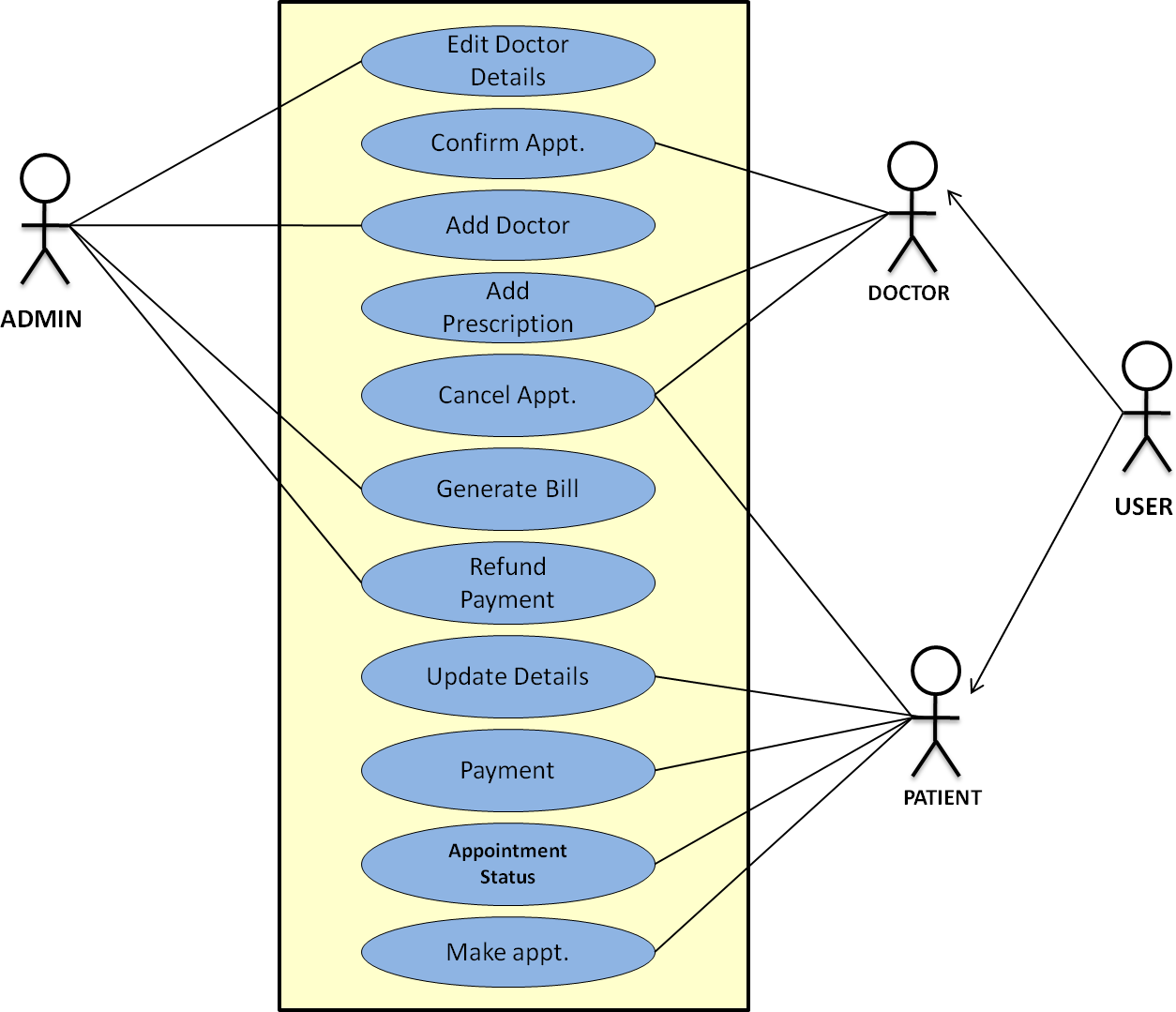
**Result:** Thus the Software Requirement Specification Document for Hospital Management System has been completed.

# Ex.No. 3 MODELING

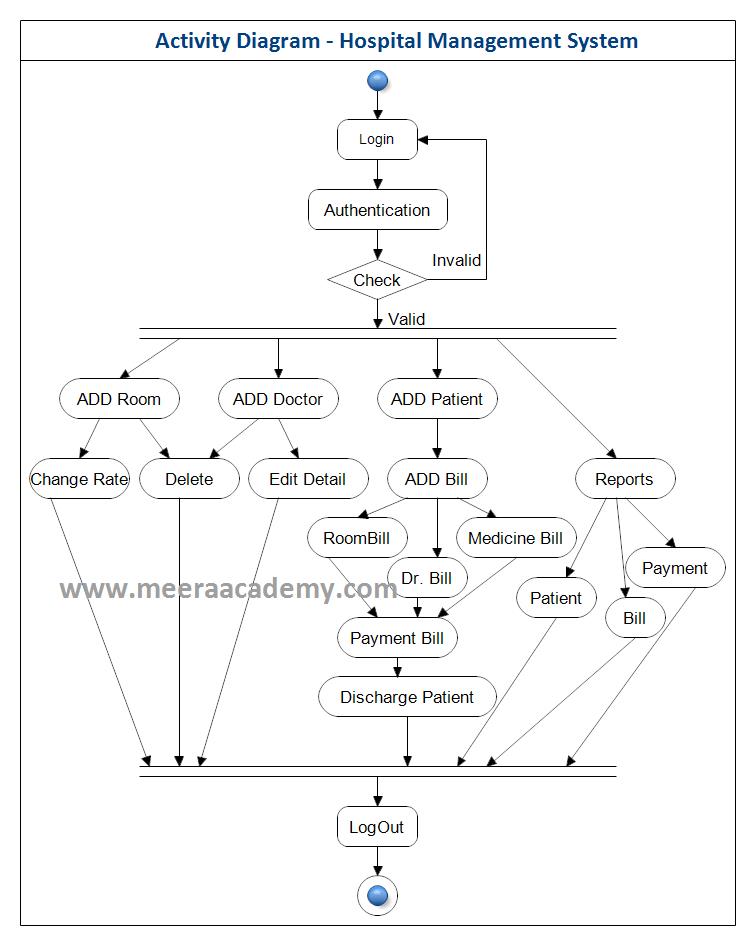
**(i) Design model –UML diagrams**

**Use case diagram**

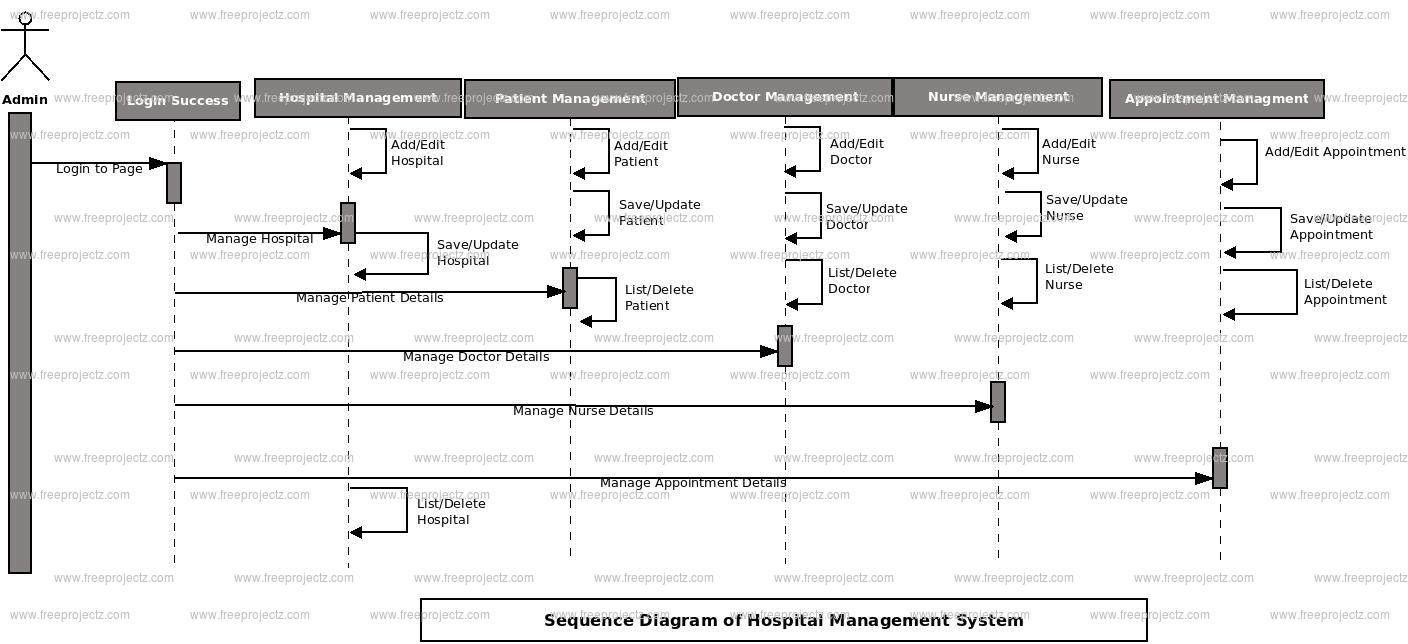
**Step 1:start ARGO UML->Create-> Use Case Diagram**



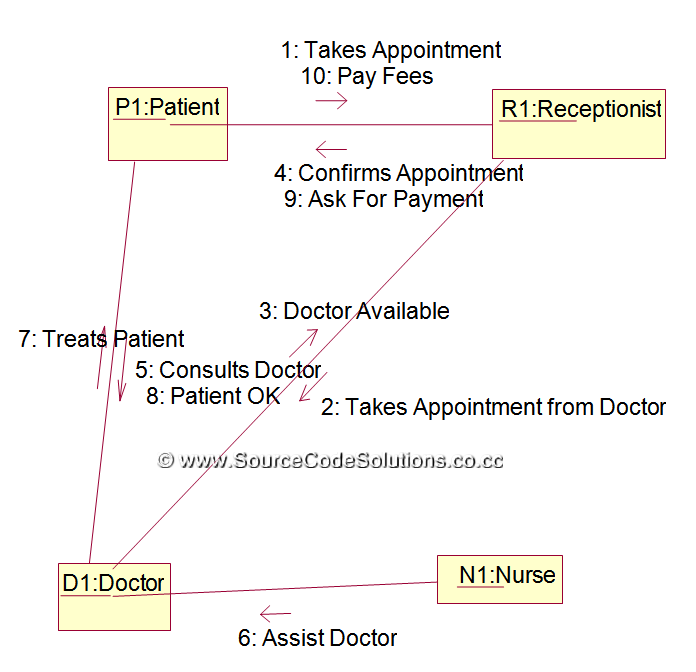
**Activity Diagram for Lending of book**



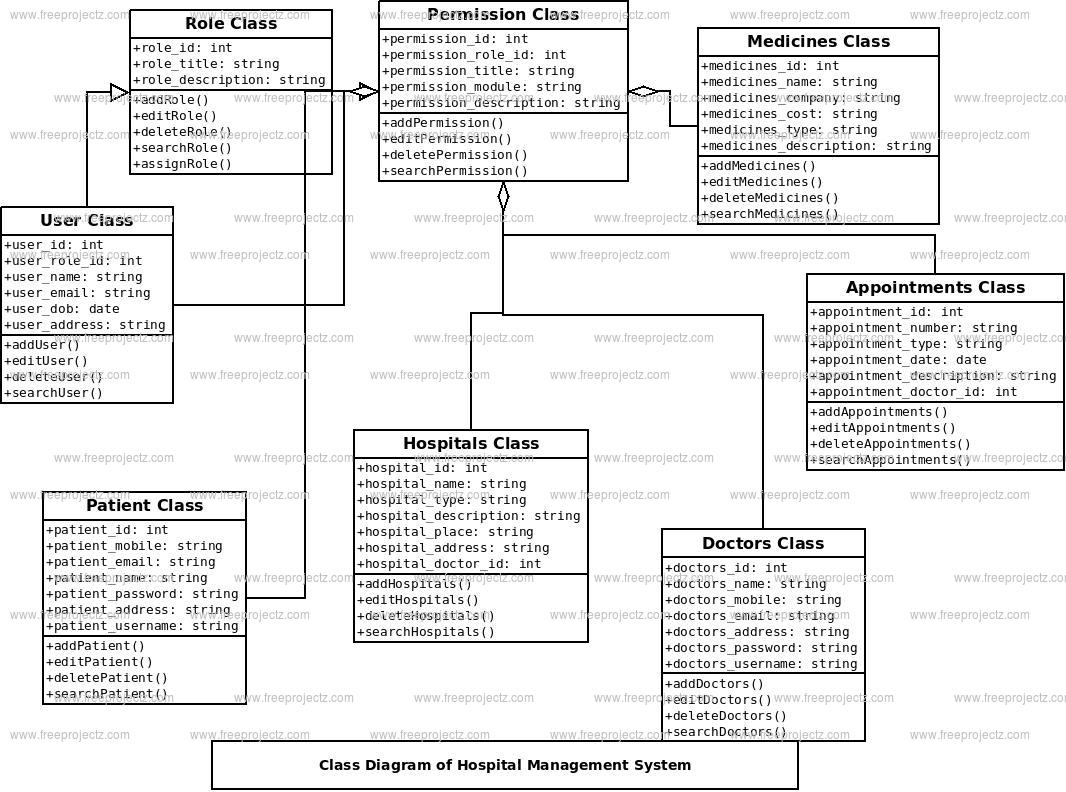
**Sequence diagram for Lending of book**



**Collaboration diagram for Lending of book**



**Class Diagram**



# Ex.No.3 (b) DATA DICTIONARY

**Patient Details**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Name** | **Alias Name** | **Where Used/How Used** | **Supplementary Data** | |
| **Data Type** | **Limitations** |
| 1 | Doctor name | Title | Borrow, Display | string | Up to 20 char |
| 2 | Symptoms | - | Return, Display | string | Up to 250 char |
| 3 | Address |  | Borrow, Display | string | Up to 200 char |
| 4 | Doctor Mobile |  | Borrow, Display | integer | Up to 10 digits |
| 5 | Admit date |  | Borrow, Display | integer | Up to 8 digit |

# Doctor Details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Name** | **Alias Name** | **Where Used/How Used** | **Supplementary Data** | |
| **Data Type** | **Limitations** |
| 1 | Doctor name | - | Membership, Borrow, Display | string | Up to 20 char |
| 2 | Specialization | - | Membership | string | Up to 20 char |
| 3 | Appointments |  | Membership | string | Up to 20 char |
| 4 | Patients |  | Membership | string | Up to 100 char |
| 5 | Patient Discharge |  | Membership | integer | Up to 8 digit |

**Ex.No.4 IMPLEMENTATION**

# Login use case:

<!DOCTYPE html>

{% load widget\_tweaks %}

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title>SURIYA HOSPITAL</title>

<style type="text/css">

body {

color: #aa082e;

background-color: #b6bde7;

font-family: 'Roboto', sans-serif;

}

a:link {

text-decoration: none;

}

.note {

text-align: center;

height: 80px;

background: -webkit-linear-gradient(left, #0072ff, #8811c5);

color: #fff;

font-weight: bold;

line-height: 80px;

}

.form-content {

padding: 5%;

border: 1px solid #ced4da;

margin-bottom: 2%;

}

.form-control {

border-radius: 1.5rem;

}

.btnSubmit {

border: none;

border-radius: 1.5rem;

padding: 1%;

width: 20%;

cursor: pointer;

background: #0062cc;

color: #fff;

}

</style>

</head>

<body>

{% include "hospital/navbar.html" %}

<br>

<br>

<br><br>

<!--- login page for admin by admin(sumit) ---------->

<form method="post">

{% csrf\_token %}

<div class="container register-form">

<div class="form">

<div class="note">

<p>Admin Login Page</p>

</div>

<div class="form-content">

<div class="row">

<div class="col-md-6">

<div class="form-group">

{% render\_field form.username class="form-control" placeholder="Username" %}

</div>

</div>

<div class="col-md-6">

<div class="form-group">

{% render\_field form.password class="form-control" placeholder="Password" %}

</div>

</div>

</div>

<button type="submit" class="btnSubmit">Login</button>

<div class="text-center">Do not have account? <a href="adminsignup">Signup here</a></div>

</div>

</div>

</div>

</form>

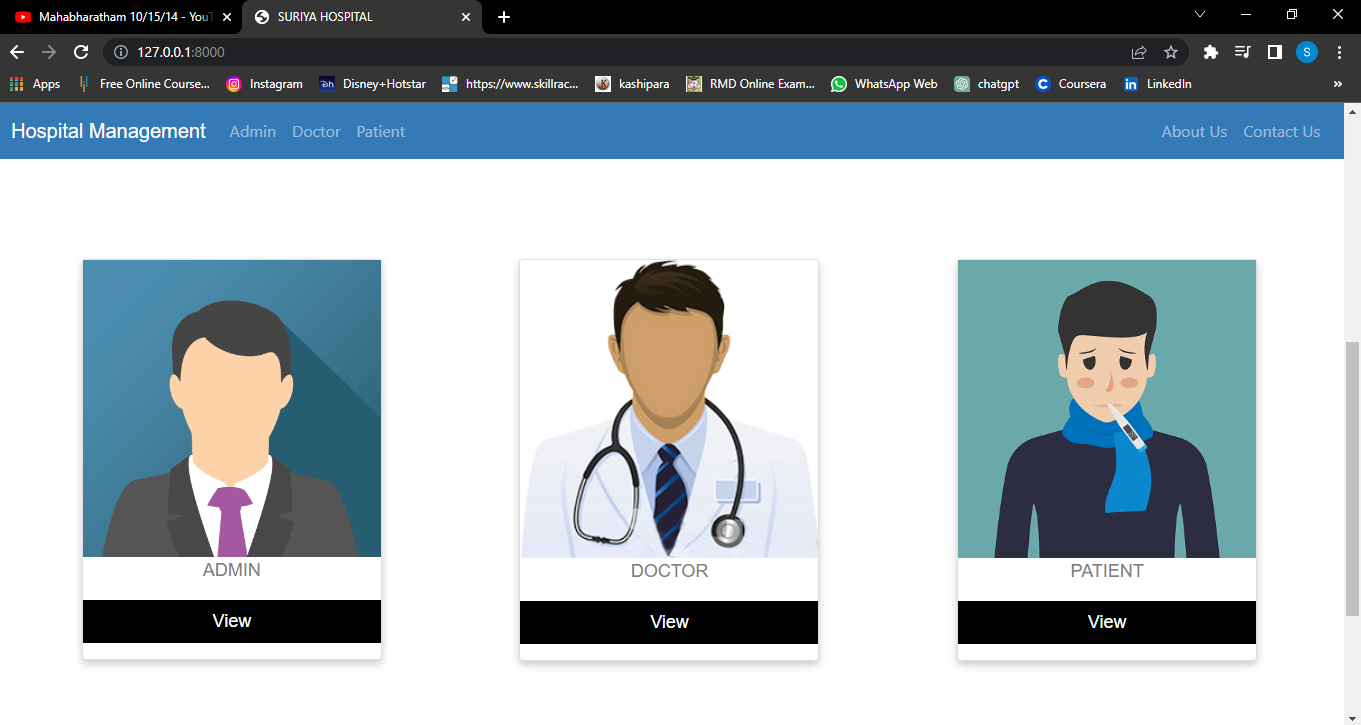
<br><br><br>

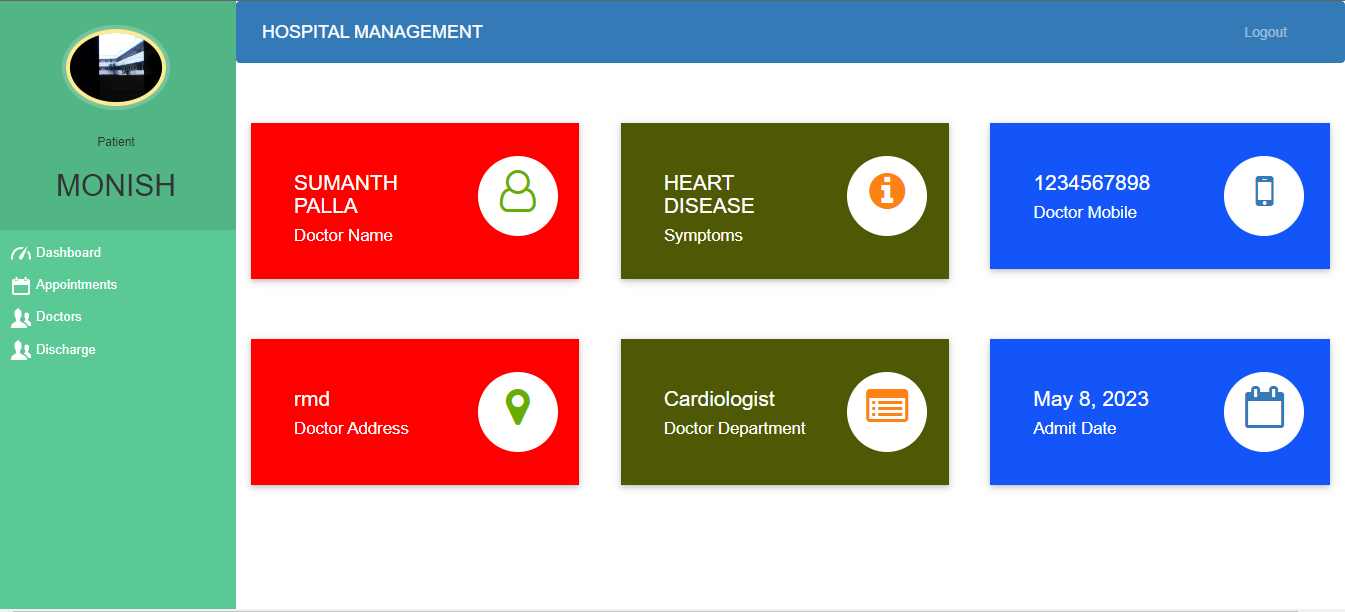
{% include "hospital/footer.html" %}

</body>

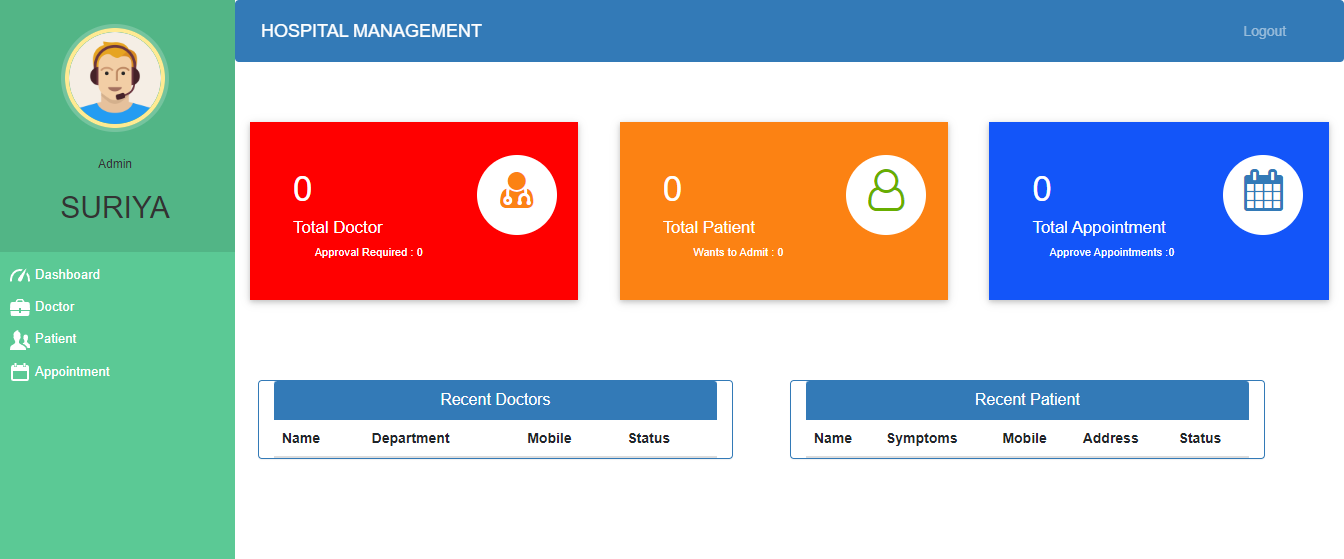
</html>

**HOME PAGE**

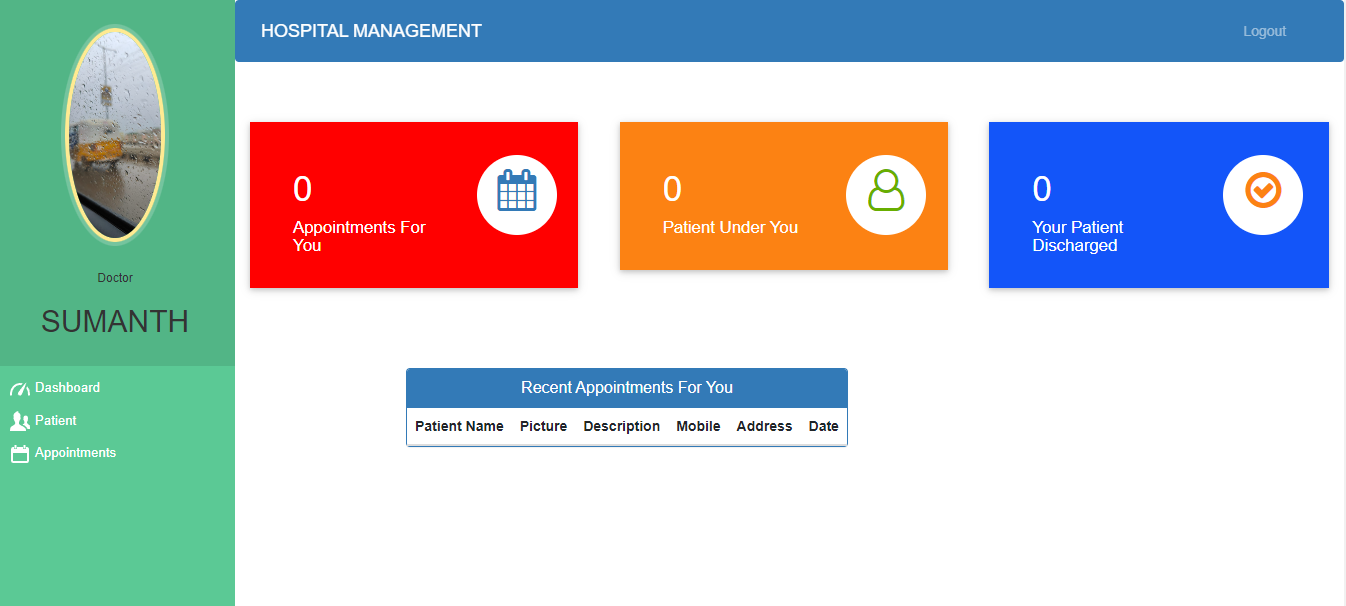




PATIENT PAGE



ADMIN PAGE



DOCTOR PAGE

# Ex.No 5 TESTING

### Test cases:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Requirement** | **Description** | **Input** | **Expeted o/p** | **Actual o/p** |
| LOGIN | Username & password should be entered | PATIENT: Can login using unique Id and Password after this system shall show his/her profile.  DOCTOR: Can login using unique Id and Password after this system shall show his/her profile.  ADMIN: Can login using unique Id and Password after this system shall show a  profile with links to maintain the website. | Username,password | Valid msg | Valid msg |
| REGISTRATION | Username & password should be entered | PATIENT: Can Register by filling all the required details, after this the system will verify the details and check if already  registered or not. | Username,password | Valid msg | Valid msg |
| MAKE APPT. | Patient should apply and admin should approve | PATIENT: Can Select doctor, date time and make an appointment request after this system shall show a confirmation for  appointment request. | Patient name , doctor name, date and time. | Valid msg | Valid msg |
| CANCL APPT. | Doctor has the facility to cancel appointment | PATIENT : Can Cancel appointment if want to by just one click after this system shall ask for re-schedule or refund of payment.  DOCTOR : Can Cancel appointment if want to by just one click after this system shall  send a message to the patient. | Appointment number | Valid msg | Valid msg |
| PAYMENT | Should be redirected to Payment portal | PATIENT : Enter payment details and make payment after this system shall show the  generated bill by the hospital. | Bill no ,select payment method | Valid msg | Valid msg |
| DOCTOR MODULE | All details of doctors | ADMIN : Can add a new doctor by filling all the details after this system shall show a confirmation message.  Can Remove a doctor by just one click after this system shall show confirmation  message. | Username,password | Valid msg | Valid msg |
| PATIENT MODULE | All details of patients | PATIENT : Can view payment history or can search for a particular bill also after this system shall show a bill or history.  Can also See or search for a doctor by entering dept. name or doctor id if known after this system will check for the doctor if found shall show doctor’s profile.  Can also update details after this system shall ask for re-enter password and after  verifying password shall update details. | Username,password | Valid msg | Valid msg |
| ADD PRESCRIPTION | Prescription generated by doctor | DOCTOR : Enter Patient Id and after this all the treatment details and medicine, remark and advice for the patient after this system  shall show a message for update. | patient id | Valid msg | Valid msg |