Software Requirements Specification

Version 1.1

Hospital Management System

Team # 3

|  |  |  |
| --- | --- | --- |
| Member Name | Member Roll # | Primary Responsibility |
| Ali Awan | 15-4171 |  |
| Saif ul Islam | 15-4217 |  |
| Sharjeel Mansha | 15-4257 |  |
| Hassan Khan | 15-4150 |  |
| M. Kashan Siddiqui | 15-4111 |  |

# Table of Contents

1. Introduction 1

1.1 Scope 1

1.2 Business Goals 1

1.3 Document Conventions 2

1.4 References 2

2. Overall Description 3

2.1 Product Features 3

2.2 User Classes and Characteristics 3

2.3 Operating Environment 4

2.3.1 Software Requirements 4

2.3.2 Hardware Requirements 4

2.4 Design and Implementation Constraints 4

2.5 Assumptions and Dependencies 5

3. Functional Requirements 6

3.1 Sign up 6

3.2 Login 7

3.3 Request for Appointment 7

3.4 See history 8

3.5 Give Feedback 9

3.6 Add doctor and other Employee 10

3.7 Remove doctor and other Employee 10

3.8 Search Patient 11

3.9 Search doctor 12

3.10 View Upcoming appointments 13

3.11 View History of Patients 13

3.12 Get Notification of Appointment 14

3.13 Create Prescription 15

4. Nonfunctional Requirements 16

4.1 Performance Requirements 16

4.2 Security Requirements 16

5. Other Requirements 16

Appendix A: Glossary 17

Appendix B: Analysis Models 18

Appendix C: Design Models 24

Appendix D: Screenshots 25

Appendix E: Test Cases 26

Appendix F: IV & V Report 27

Appendix G: Risk Report 28

Appendix H: Activity Timesheet 29

Appendix I: Updated Project Plan 30

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name(s)** | **Date** | **Reason(s) For Change(s)** | **Version** |
| Section 2 |  |  |  |
|  |  |  |  |

# Introduction

Product We will be making a clinic management system that will help the patients to get in contact with the doctors and to see their clinical history. It will eliminate all the efforts that were required to get appointments manually and the patients will no longer have to keep their records with them. They will be available at one place. The patients will have their accounts on our website where all their clinical records will be kept safe. There will be a separate interface for Doctors and Patients along with their separate logins.The registration forms for the patients will be provided which will register the patients for the database. Also,the Admin of the Clinic will be able to search the patients through their names,Patient ID etc. The website will have the information regarding different departments of the clinic.It will also help the doctors to keep track of their currentappointments given to patients. The patient after knowing about the clinic would be able to make appointment. Moreover, the doctor will be able to see the clinical history of his patient. After all the treatment, the patient would also be able to give feedback about the doctor.

## Scope

**1.1.1 Project Purpose Statement:**

The project’s sole purpose is to provide ease to the patients by automating the basic formalities of the patients and the doctors. It will provide a medium between the patients and the doctors to share the prescriptions and other relevant documents. In short, this management system will help to get ridof all that old system of maintaining and carrying files.

**1.1.2 Technicalities:**

To make this happen, we need a broad knowledge of Database Management System and manipulating data using **SQL** queries. Therefore, we will be using clauses like join, set operators, unary operators etc. We will also use **HTML** and **JavaScript** to design our webpage and associate it with our database using suitable techniques. Moreover, some aggregate functions will also be utilized for stats purposes like computing total number of appointments or total patients treated by a doctor etc.

**1.1.3 Project Working:**

The website will provide an interface for our visitors which will provide details about the clinic. Then that visitor will first register as a patient and then he/she will be able to get an appointment with a doctor. The doctor could access the patient’s history and give him prescriptions etc. After treatment the patient will also be able to give feedback about the doctor.

## Business Goals

As now a days it’s the biggest challenge to go to the hospitals and to find the specialist to get yourself checked. Our system will help people to save their precious time they can get their appointments by just signing in t hospital’s website and go on appropriate time. In addition to that they don’t have any need to carry their prescription or medical history along with them. Our system will keep record of your past history.

## Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether italicized nouns represent external systems.>

## References

<List all books, conference papers, journal articles, websites, etc. used in preparing the content of this SRS. Provide enough information so that the reader could access a copy of each reference, including title, author, volume/edition number, page number(s), and publication year. Mention complete URLs for websites.>

# Overall Description

## Product Features

Our hospital management system is a complete management system that provides ease to administration, doctors and patients all at the same time. It is single data based system that will be keeping record at one place providing real time data access to every user. Major Features will be:

* **Patient management**– Our hospital management system will provide complete patient management feature that will help administration to keep record of patients in and patient out that will also help administration allocating resources like beds etc. Patient registrations could help managing their record and their clinical history. Scheduling an appointment, prescription, diagnosis and other such records can be maintained accurately.
* **Doctor records-** AHospital Management Systemcan be very helpful for doctors as well. The doctor would be able to accept or reject appointment requests generated by patients. Doctor will be able to access patients clinical record of patient and previous diagnosis. Furthermore, Doctor can add prescription of relevant patient and can refer him to other doctor as well. Doctor will also have access to all of the previous patient’s history including test reports and previous diagnosis.
* **Administration management-** In hospital management system the admin would be able to cover all the aspects of management regarding hospital like adding and removing working staff like nurses and lower working staff. Doctors can add or remove doctors from hospital. In addition to that admin can manage financial aspects of hospital as well. Furthermore, effective accounts management could be done using the automated system.

## User Classes and Characteristics

* **Patient:**

Patient will be using system frequently as he will be the end user of our system. His key features includes:

**Key Features:**

* Make request for appointment
* Check Doctor’s available slots
* Check his Prescription
* See doctors’ Profile
* **Doctor:**

Doctor will also be using system frequently he will be in communication and in interaction with patients. His key features include:

**Key Features:**

* Accept/Reject Patients’ Appointment request
* See patients past history
* Generate Bill
* Add prescription
* See patients Test Reports
* **Admin:**

Admin will be responsible for managing the hospital him key features include:

**Key Features:**

* Add/ Remove Doctors
* Add/Remove other staff
* Check Statistics
* Manage equipments

## Operating Environment

Our system is compatible with all the windows after windows 7. Some other hardware and Software requirements are as follows:

### Software Requirements

* Microsoft Sql Server 2014
* Microsoft Visual Studio Professional 2015

### Hardware Requirements

* Processor core i3 (2.3 – 2.4 GHz )
* RAM 4Gb
* 320 GB Hard drive.

## Design and Implementation Constraints

There are several constraints that need to be followed in order to implement this system and to use it without any delay or error. This system is a web based thus a browser is required.

1. **Hardware Limitations:**

* 4GB of RAM or above
* Minimum Dual Core processor or above
* Minimum of 4GB available hard drive space.
* No special graphic card required

1. **Software Limitations:**

* Sql Management Studio 2010 or above
* Visual studio 2012 or above
* Internet browser (chrome, Mozilla , etc)

1. **Database Required:**

* SQL server express2010 or above

1. **Programming Language:**

* C#
* JavaScript
* Asp.net
* SQL

## Assumptions and Dependencies

There are very less assumptions and dependencies in our system as it has

* There are fixed number of doctors for now
* There are fixed number of departments.
* User could give feedback with in 24 hours after his appointment.
* Not implementing all the hospital finance management.
* Only billing of patients and doctors are implemented.
* Everyone has valid username and password.
* Only admin can alter, add or remove record.

# Functional Requirements

<All functional requirements are expressed as use-cases. Fill out the following template for each use-case. Don’t really say “Use-Case 1.” State the use-case name in just a few words e.g. “Withdraw Cash from ATM”. A use-case may have multiple alternate courses of action.>

## Sign up

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | Sign Up | |
| **Purpose** | | Patients will sign up to do their respective actions | |
| **Priority** | | High | |
| **Actors** | | Patient | |
| **Pre-conditions** | | Patients must have | |
| **Post-conditions** | | Patient will be redirected to login page | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Patient will go to website main page | |  |
| **2** | Enter all the required fields | |  |
| **3** | Press the sign up button | | System will show success message and will redirect to Login page. |
| **…** |  | |  |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | If he enters any invalid data | | Invalid data |
| **2** | If user leaves something empty | | This field is required |
| **3** |  | |  |
| **…** |  | |  |

Table : UC-1

## Login

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | Login | |
| **Purpose** | | All the actors will login to do their respective tasks. | |
| **Priority** | | High | |
| **Actors** | | Patient, Doctor, Admin | |
| **Pre-conditions** | | Every one should have valid account | |
| **Post-conditions** | | All will be redirected to their respective homepage. | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | user will go to website login page | |  |
| **2** | Enter username and password | |  |
| **3** | Press the login button | | System will redirect to relevant homepage. |
| **…** |  | |  |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | If he enters any invalid username | | Invalid username or password |
| **2** | If he enters any invalid password | | Invalid username or password |
| **3** |  | |  |
| **…** |  | |  |

Table : UC-2

## Request for Appointment

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | Request appointment | |
| **Purpose** | | Patients will request to doctor for appointment. | |
| **Priority** | | High | |
| **Actors** | | Patient | |
| **Pre-conditions** | | Patients must have logged in | |
| **Post-conditions** | | Patient will be redirected to page showing all the departments | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Patient will go to get appointment tab from navigation bar | |  |
| **2** | Select department from the list | |  |
| **3** | Will select the doctor | |  |
| **4** | User will then opt the appropriate time of available slots | | System will show success message that request for appointment has been sent. |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | If he enters any invalid data | | Invalid data |
| **2** | If user leaves something empty | | This field is required |
| **3** |  | |  |
| **…** |  | |  |

Table : UC-3

## See history

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | See history | |
| **Purpose** | | Patients would be able to see their past history | |
| **Priority** | | Medium | |
| **Actors** | | Patient | |
| **Pre-conditions** | | Patient have logged in | |
| **Post-conditions** | | User will be redirected to history page | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | user will go to see history page | | System will redirect to history page. |
| **2** |  | |  |
| **3** |  | |  |
| **…** |  | |  |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | If the users session is not valid | | Please login to continue |
| **2** |  | |  |
| **3** |  | |  |
| **…** |  | |  |

Table : UC-4

## Give Feedback

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | Give Feedback | |
| **Purpose** | | Patients would be able to give feedback | |
| **Priority** | | Medium | |
| **Actors** | | Patient | |
| **Pre-conditions** | | Patient must have logged in | |
| **Post-conditions** | | User will be redirected to success page | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User will go to give feedback tab from navigation bar | | System will redirect to feedback page |
| **2** | Give feedback about doctor | |  |
| **3** | Press submit | | System will redirect to success page |
| **…** |  | |  |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | If the user enter invalid expression in given field | | You entered an invalid expression |

Table : UC-4

## Add doctor and other Employee

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | Add doctor and other employee | |
| **Purpose** | | Admin would be able to add doctors and other employee in to the hospital’s system | |
| **Priority** | | High | |
| **Actors** | | Admin | |
| **Pre-conditions** | | Admin must have logged in | |
| **Post-conditions** | | Admin will be redirected to success page | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User will go to add doctor/employee tab from nav bar | |  |
| **2** | Add details about doctor/employee | |  |
| **3** | Press “Add” button | | System will redirect to success page |
| **…** |  | |  |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | If the user enter invalid expression in given field | | You entered an invalid expression |
| **2** | If he leaves a mandatory field empty | | This field is required |

Table : UC-4

## Remove doctor and other Employee

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | Remove doctor and other employee | |
| **Purpose** | | Admin would be able to remove doctors and other employee in to the hospital’s system | |
| **Priority** | | High | |
| **Actors** | | Admin | |
| **Pre-conditions** | | Admin must have logged in | |
| **Post-conditions** | | Admin will be redirected to success page | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User will go to remove doctor/employee tab from nav bar | |  |
| **2** | Add details about doctor/employee | |  |
| **3** | Press “Remove” button | | System will redirect to success page |
| **…** |  | |  |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | If the user enter invalid expression in given field | | You entered an invalid expression |
| **2** | If he leaves a mandatory field empty | | This field is required |

Table : UC-5

## Search Patient

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | Search Patient | |
| **Purpose** | | Admin would be able to search any patient to get details | |
| **Priority** | | Normal | |
| **Actors** | | Admin | |
| **Pre-conditions** | | Admin must have logged in | |
| **Post-conditions** | | Admin will be redirected to patients details page | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User will go to search patient tab from nav bar | |  |
| **2** | Enters Patient Name in the search field | | System will show several patients with same name |
| **3** | Choose the correct name | |  |
| **…** | Get details | | System will redirect to patients details page |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | If the user enter invalid expression in given field | | You entered an invalid expression |
| **2** | If he leaves a mandatory field empty | | This field is required |
| **3** | If there isn’t any patient of such name exists | | Nothing found |

Table : UC-8

## Search doctor

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | Search doctor | |
| **Purpose** | | Admin would be able to search any doctor to get details | |
| **Priority** | | Normal | |
| **Actors** | | Admin | |
| **Pre-conditions** | | Admin must have logged in | |
| **Post-conditions** | | Admin will be redirected to doctor’s details page | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | User will go to search doctor tab from nav bar | |  |
| **2** | Enters doctor’s Name in the search field | | System will show several doctors with same name |
| **3** | Choose the correct name | |  |
| **…** | Get details | | System will redirect to doctor’s details page |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | If the user enter invalid expression in given field | | You entered an invalid expression |
| **2** | If he leaves a mandatory field empty | | This field is required |
| **3** | If there isn’t any doctor of such name exists | | Nothing found |

Table : UC-9

## View Upcoming appointments

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | View Upcoming Appointments | |
| **Purpose** | | Doctors would be able to see his upcoming appointments | |
| **Priority** | | Normal | |
| **Actors** | | Doctor | |
| **Pre-conditions** | | Doctor must have logged in | |
| **Post-conditions** | | Doctor will be redirected his appointment page | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Doctor will go to appointments tab from nav bar | |  |
| **2** |  | | System will show all appointments of this doctor |
| **3** |  | |  |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | If the user’s session is not valid | | Please login to continue |

Table : UC-10

## View History of Patients

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | View history of Patients | |
| **Purpose** | | Doctors would be able to see his Patient’s history | |
| **Priority** | | Normal | |
| **Actors** | | Doctor | |
| **Pre-conditions** | | Doctor must have logged in | |
| **Post-conditions** | | Doctor will be redirected to history page | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Doctor will go to patient’s history page | |  |
| **2** | Type the patients name | | System will show all the patients with this name |
| **3** | Choose the exact one out of the list | | System will show this patient’s complete history |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | If no such patient exist | | No record found |

Table : UC-11

## Get Notification of Appointment

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | Get notification of appointment | |
| **Purpose** | | Doctors would be able to see notifications of appointments | |
| **Priority** | | high | |
| **Actors** | | Doctor | |
| **Pre-conditions** | | Doctor must have logged in | |
| **Post-conditions** | | Doctor will be redirected to notification page | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Doctor will go to pending appointments tab from navigation bar | |  |
| **2** | Their will be the list of all pending appointments again each their will be the option of accept or reject | |  |
| **3** | User will keep on selecting accept or reject accordingly till all are done | | System will show there is no pending notifications now |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | If user’s session is invalid | | Please login to continue |

Table : UC-12

## Create Prescription

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | Create Prescription | |
| **Purpose** | | Doctors would be able create patients prescription after diagnosis and check up | |
| **Priority** | | high | |
| **Actors** | | Doctor | |
| **Pre-conditions** | | Doctor must have logged in, Patients appointment has arrived | |
| **Post-conditions** | | Doctor will be redirected to success page | |
| **Typical Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | Doctor will go to create prescription tab from navigation bar | |  |
| **2** | User will fill out all the details depending upon diagnosis, suggest medicines etc | |  |
| **3** | Press enter | | System will show success message |
| **Alternate Course of Action** | | | |
| **S#** | **Actor Action** | | **System Response** |
| **1** | If patients appointment hasn’t arrived yet | | This is not the time of this patient |
| **2** | If user enters invalid expression | | Please enter valid expression |

Table : UC-13

# Nonfunctional Requirements

## Performance Requirements

Performance requirements deal with the time it takes to respond and other such factors.

**4.1.1 Response Time**  
                        The response should be generated in like 1 second after checking the doctor’s information.   
**4.1.2**     **Capacity**   
                        System should be capable of dealing with almost 1000 request simultaneously. .   
**4.1.3**     **User-interface**

User interface should be very interactive and it should respond as quick as it can. Diversified type of people would be coming to the website

## Security Requirements

The biggest security threat that we will be dealing is to make sure that every one only perform his assigned tasks and doesn’t exceed from his limits. This means that a doctor can not add or remove other doctor this is a role that is only assigned to admin. So to make this sure session id will be used to ensure that relevant person has requested this page.   
In addition to that to prevent our website as we have many insertion fields so we have to implement cross site scripting to prevent sql injection attacks.

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. These might include database requirements, external (hardware, software, or communication) interface requirements, internationalization requirements, legal requirements, and reuse objectives for the project.>

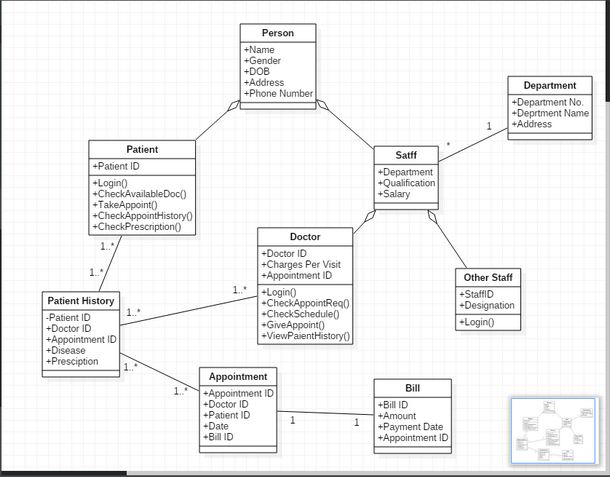
# Appendix A: Glossary

There are several sub headings that break down the topic into sub topics to elaborate in detail. DFD stands for Data Flow Diagram. HMS stands for Hospital Management System

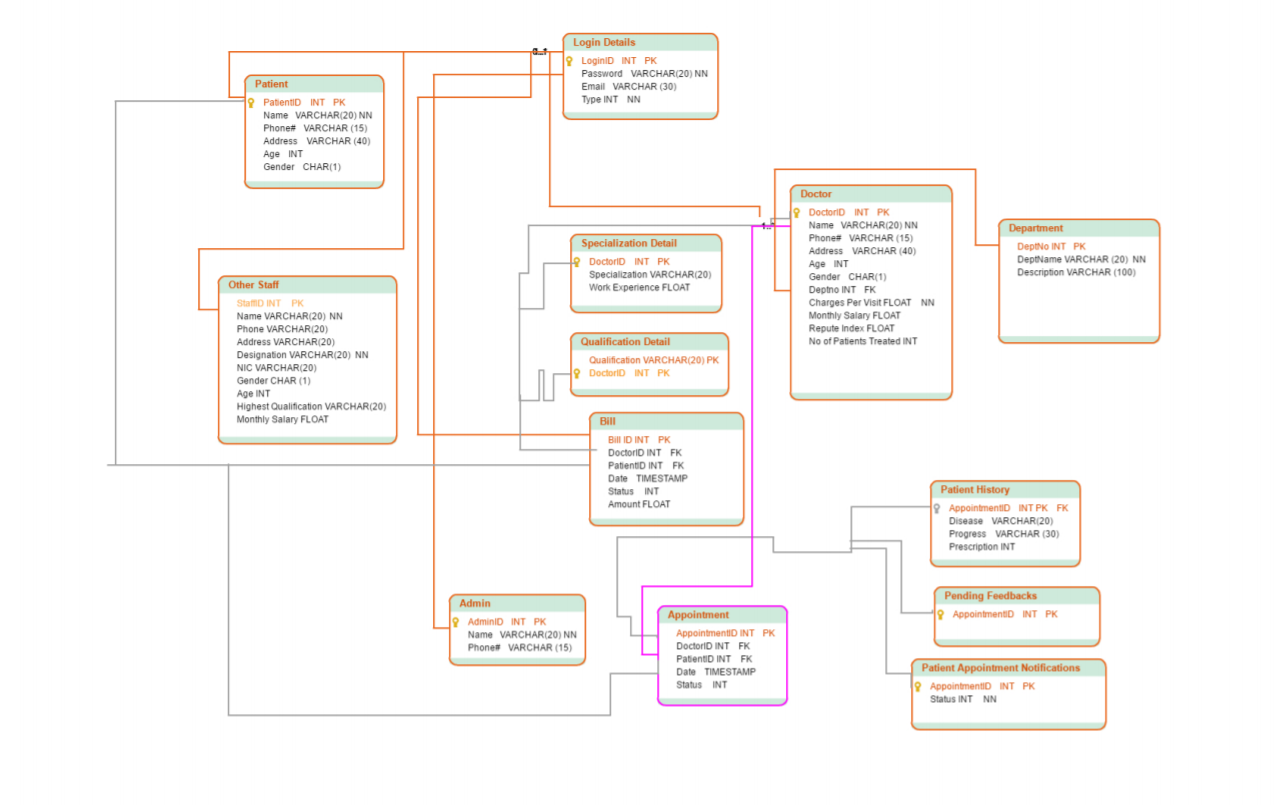
Appendix B: Analysis Models

<Include the following analysis models: use-case diagram, entity-relationship diagram, class diagram, data flow diagram.>

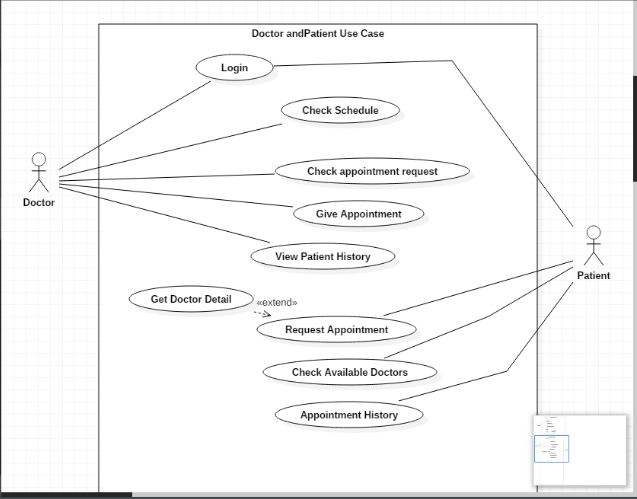
# ER Diagram:

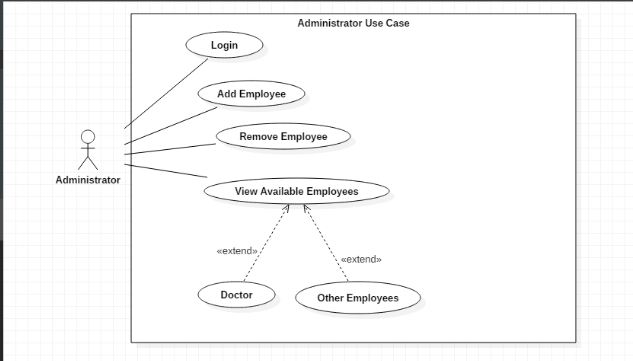


# CLASS DIAGRAM:



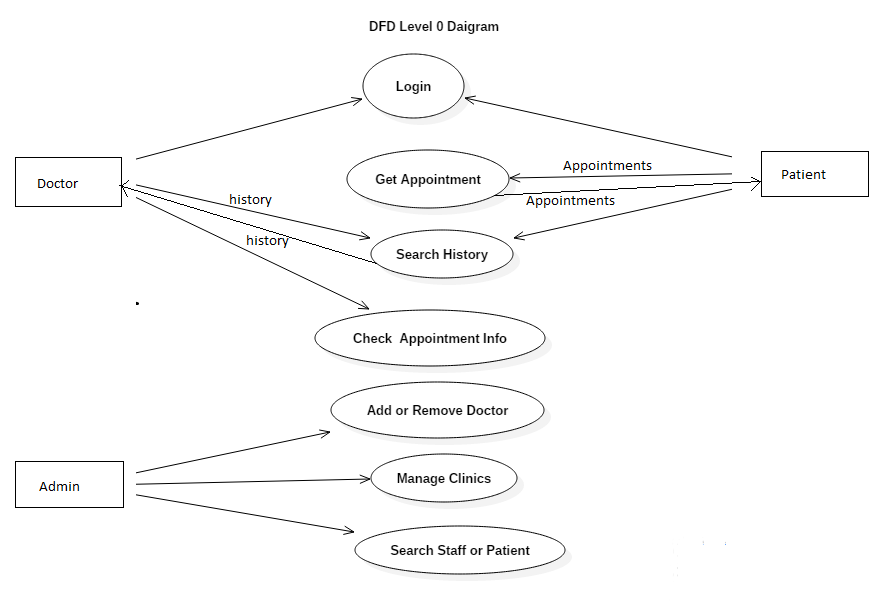
# USE CASE DIAGRAM

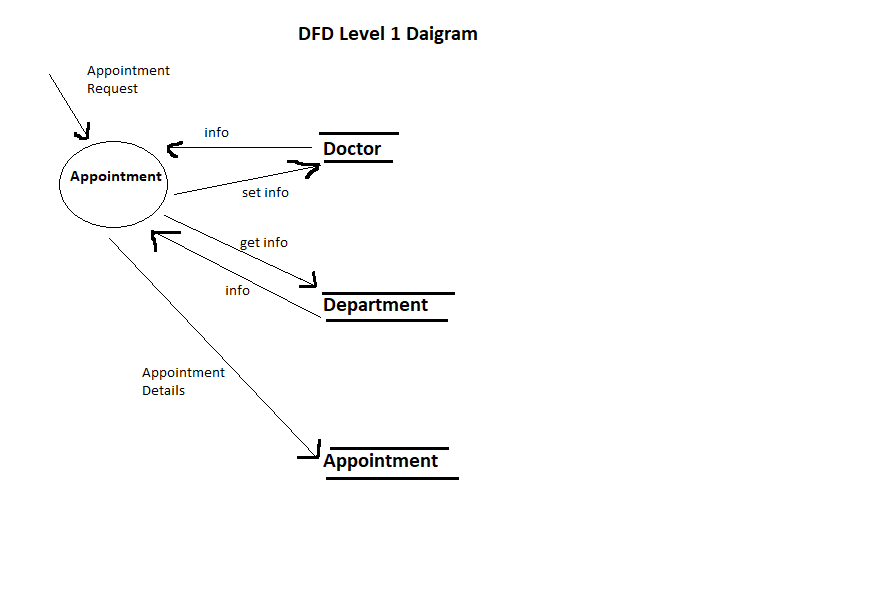


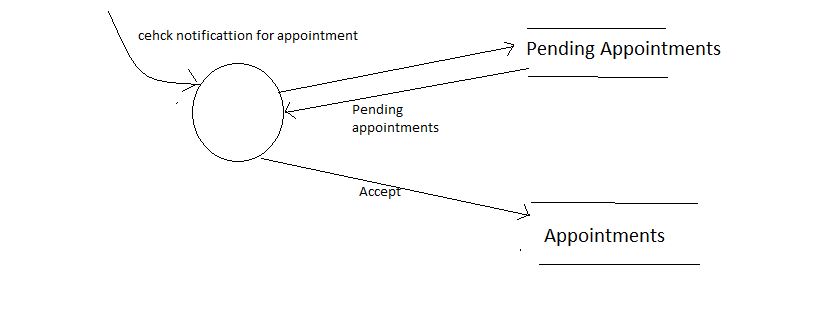


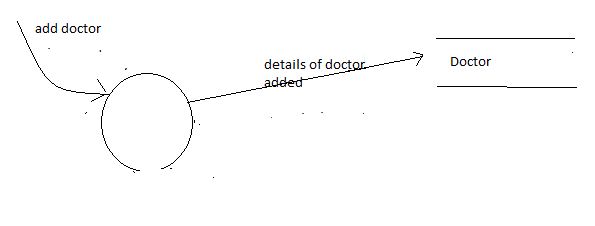
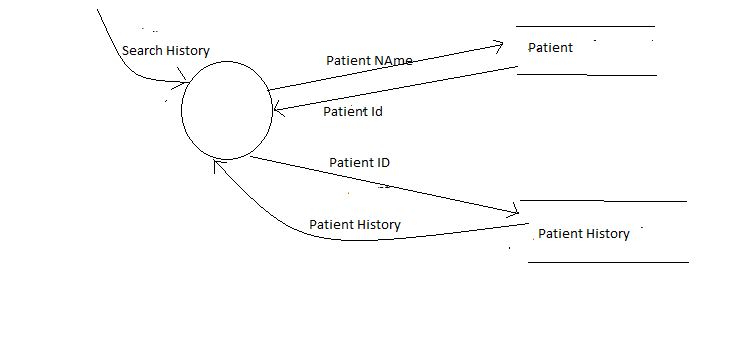
# DATA FLOW DIAGRAM:

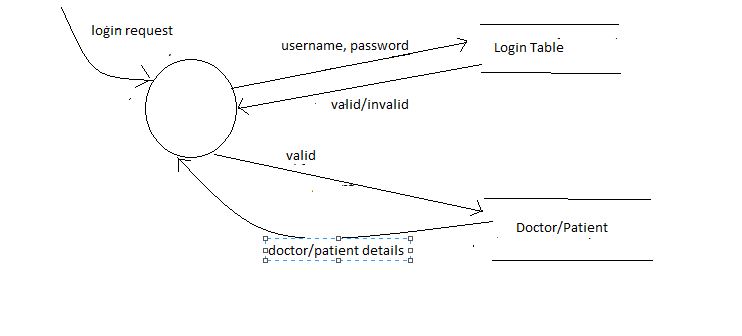
## Level 0 DFD



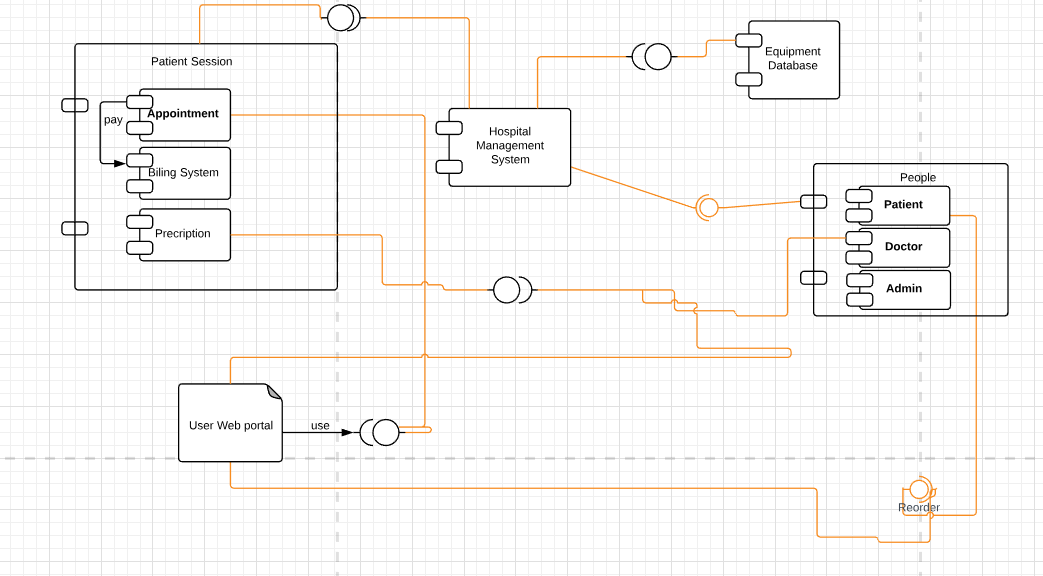






Appendix C: Design Models

*< Include the following design model: component diagram.>*

**Appendix D: Screenshots

*< Include all screenshots of your software application’s graphical user interface.>*

Appendix E: Test Cases

< Fill out the following template for each test case.>

|  |  |
| --- | --- |
| **Identifier** | TC-1 |
| **Priority** | <Choose one from {High, Medium, Low}> |
| **Related requirements(s)** | <Include use-case identifier(s) for functional requirement(s) and SRS section/sub-section number(s) for other requirement(s).> |
| **Short description** | … |
| **Pre-condition(s)** | … |
| **Input data** | … |
| **Detailed steps** | … |
| **Expected result(s)** | … |
| **Post-condition(s)** | … |

Table : TC-1

Appendix F: IV & V Report

**IV & V Resource**

Name Roll # Signature

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S#** | **Defect Description** | **Origin Stage** | **Status** | **Fix Time** | |
| **Hours** | **Minutes** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| … |  |  |  |  |  |

**Table 3: List of non-trivial defects**

Appendix G: Risk Report

**[[1]](#footnote-2)Project Risks**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk Description** | **Impact**  **(1 – 10)** | **Probability**  **(0 – 1)** | **[[2]](#footnote-3)Risk**  **Exposure** | **Weeks Active** | **Mitigation Strategy** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Appendix H: Activity Timesheet

|  |  |  |
| --- | --- | --- |
| **Activity** | **Time** | |
| **Hours** | **Minutes** |
| Requirements Engineering |  |  |
| Analysis and Design |  |  |
| Implementation |  |  |
| Testing |  |  |
| Deployment |  |  |
| Project Management |  |  |
| IV & V |  |  |

**Project Manager**

Name Roll # Signature

Appendix I: Updated Project Plan

< Include screenshots of your updated project plan prepared using MS Project.>

1. Risks should be sorted in descending order of risk exposure. [↑](#footnote-ref-2)
2. Risk Exposure = Risk Impact x Risk Probability [↑](#footnote-ref-3)