

**MicroLink Information Technology College**

**Department of Computer Science**

**Web Based E-Health System**

**Requirement Analysis Document (RAD)**

**For Addis Hiwot General Hospital**

**Prepared by:**

1. **Biruk Endalew**
2. **Dawit Tadesse**
3. **Eyosiyas Nigussie**
4. **Kirubel Maregn**
5. **Yohannes Melese**

**Advisor: Tensaye Desalegn**

19 June 2022

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 19/6/2022 | 1.0 | This is the first RAD document | All group members |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms and Abbreviations 2

1.4 References 2

1.5 Overview 2

2. Current system 2

3. Proposed system 3

3.1 Overview 3

3.2 Functional requirements 3

3.3 Nonfunctional requirements 4

3.3.1 User interface and human factors 4

3.3.2 Documentation 4

3.3.3 Hardware considerations 5

3.3.4 Performance characteristics 6

3.3.5 Error handling and extreme conditions 6

3.3.6 System interfacing 6

3.3.7 Quality issues 6

3.3.8 System modifications 6

3.3.9 Physical environment 7

3.3.10 Security issues 7

3.3.11 Resources and management issues 7

*3.3.12 Reliability……….…...………………………………………………………………………………...………7*

3.4 Constraints (“Pseudo requirements”) 8

3.5 System models 8

3.5.1 Scenarios 8

3.5.2 Use case model 11

3.5.3 Object model 16

3.5.4 Data dictionary 16

3.5.5 Class diagrams 18

3.5.6 Dynamic models 19

4. Glossary 24

5. Supporting Information 25

**System Requirements Specification**

# Introduction

This section gives an overview of the contents of the full document. the results of the

requirements elicitation effort reported in this Requirement Analysis Document (RAD). RAD uses a variety of textual, symbolic, and diagrammatic representations based on the subject's wishes. The primary functional activities, states, and item interactions within the design are shown using the system model. It acts as a contractual basis between the client and the developers, and it seeks to explain the system in terms of functional and non-functional requirements. This document quickly summarizes the system's functional and non-functional needs, as well as its security, hardware and software requirements, risk, cost, how the user interacts with the system, the type of training required, error handling, backup strategy, and power stability.

* 1. **Purpose**

The results of the requirements elicitation and the analysis activities are documented in the Requirements Analysis Document (RAD) For E Health System. This document completely describes the system in terms of functional and nonfunctional requirements and serves as a contractual basis between the customer and the developer. The RAD must be written in the language of the customer's domain of business/expertise. Under no circumstances should any "computerize" terminology creep into this document.

* 1. **Scope**

The proposed software product is the E Health System. In this project we are going to design and build a functional web-based E Health System.

The aim of this project is to create an E Health System that Patient can able to book doctor’s appointment and follow up with doctors (communicate after appointment). Doctors have access to view appointment and set available time for appointment thereby making it more convenient for them and follow up patient. The admin also has access to database to add or delete doctor on system.

The system will be web-based online application that is available on all major browsers. should be user friendly in all usual devise (PC, mobile, tablet).

* 1. **Definitions, Acronyms and Abbreviations**

|  |  |
| --- | --- |
| **RAD** | Requirement analysis document |
| **UI** | User interface |
| **User** | all system users, who can be patient, doctor |
| **Appointment** | a certain time for patient to visit doctor that is booked through system. |
| **System** | means E Health system |
| **Admin** | the system administration |
| **Interface** | interface between two or more systems describes how the system communicate |
| **Android** | Is mobile operating system. |
| **CSS** | Cascading Style Sheet is used to format the layout of web page. |
| **Database** | Is an organized collection of data |
| **JavaScript** | Is a programming language that is used to create application for android |
| **MySQL** | Is a relational database system |
| **PHP** | Is a scripting language and used to develop dynamic web application together with MySQL, HTML, CSS and Java Script. |

* 1. **References**
* Last year Rad Document
* UML Distilled: A brief guide to the standard object modeling language, Martin Flower Forewordsby CrisKobryn, Grady Booch,Ivar Jacobson, and Jim Rumbaugh, 3rdEdition.
* System Analysis and Design Methods: Professor Jeffery L Whitten & Professor
  1. **Overview**

The proposed system E Health System is organized to clearly define the proposed system using object-oriented software engineering components and techniques that briefly explain the proposed system definition, functional requirements, non-functional requirements, the software interfaces, what the software will do, the constraints under which it must operate, and how the software will react to external stimuli. This paper is designed for both end users and software developers.

1. **Current System**

The existing system of Addis Hiwot General Hospital

**Manual Appointment System**

Medical appointments have typically been made via telephone or in-person with schedulers. These methods are based on real-time verbal communication and allow for maximum flexibility in difficult situations. Because these old methods rely on schedulers, the ability to get an appointment on time is limited not only by the availability of appointment times but also by the availability of schedulers and phone lines. The ability to book at the proper time with the right health service providers has an impact on patient satisfaction with appointment scheduling.

**The Pervious way of Patient Doctor Follow Up**

Patients are unable to contact directly with doctors under the current system. It takes a long time to respond to the patient. There is very little dialogue between the patient and the doctor. Patients must physically travel to the doctor, which requires a significant amount of time and effort.

1. **Proposed system**
   1. **Overview**

The proposed system is to make web application based online Doctor or Physician appointments at any time and from any location with an internet connection and To ensures effective communication between the patient and the doctor.

In this subsection of the RAD, we will describe the proposed system Functional Requirements, Non-Functional Requirements and Constraints

* 1. **Functional Requirements**

In this sub-section we will explain about the functional requirements of the newly proposed system that define, the fundamental actions and description of activities that the system must perform and services a system must provide and accomplish. In other words, it describes the service provided for the users. The functional requirements categorized as the following:

* + 1. **Parients**
* The patient can access the E Health System.
* The patient inserts his/her email and password at log in form in the patient login page.
* If the patient inserts a valid email and password, he or she will see the main menu that contains the following items:
* Access His/her Profile.
* Personal information: his/her id given by system, name and email.
* See his/her previous appointment or follow up question and answers he/she made with doctor.

**Functional Requirements:**

* Patient can choose doctors.
* Patient can make appointment by choosing comfortable day and time.
* Patient can make contact to doctor to ask question and to get information or to give his/her status to doctor after they meet with doctor after appointment by Follow Up option.
* Patient can change his/her email, password, name or update his/her profile.
  + 1. **Doctor**
* The doctor can access the E Health System.
* The doctor insert/s his/her email and password at log in form in the doctor login page.
* If the doctor inserts a valid email and password, he or she will access the operation on:
* Access his/her Profile.
* Personal information: his/her id given by system, name and email.

**Functional Requirement**

* Doctor can response to the appointment made by patient if comfortable to him/her he/she will approve or decline.
* Doctor can response to the his/her patient question or will give information to patient.
* Doctor can update his/her profile like his/her profile image, name, email, password and contact.
  + 1. **Admin**
* The admin can access the E Health System.
* The admin insert/s his/her email and password at login form in the admin login page.
* After the admin insert valid email and password, he will access the operation on:
* Access all appointment made by patient weather approved or decline.
* Access all follow up or exchange of information by patient and doctor.

**Functional Requirement**

* Admin can register new doctor to the system
* Admin can delete doctor from the system if it is necessary
  1. **Nonfunctional Requirement**

Throughout this article, we will discuss the non-functional objectives of the newly suggested E Health System, as well as a description of features, constraints, and other qualities that constitute an acceptable system.

* + 1. **User interface and human factors**

The system will have three types of users that interact with the system. These users are Patients, Doctors and Admin.

The GUI will be intended to make a good first impression and appear user-friendly, so that people can easily utilize it. Error handling is done using massages which are popped up or alert to the user.

* + 1. **Documentation**

In the E Health System there will be four kinds of documentation

1. **Functional Description**: describes the general overview of the system.
2. **Install Manual**: explain how to set up the machine and setup the program.
3. **Introductory Manual**: describe how to get started with the program.
4. **Reference Manual**: describe all the program functions available for the user.
   * 1. **Hardware considerations**

The system will be executed and run in a computer and a smart phone with the following hardware considerations.

The hardware considerations listed in the below table were considered the system to provide the best of its performance.

**Table 1 Project Hardware Consideration**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Hardware Type | Minimum Requirement  Specification | Recommended Requirement  Specification |
| 1 | Desktop Computer | **Processor:** Dual Core, 2.10GHz  **RAM:** 2GB, DDR3  **HDD:** 500 GB, 12,000rpm, WD or Seagate | **Processor:**5 Core, 2.80GHz or above speed  **RAM:** 8GB, DDR3 or above capacity  **HDD:** 1TB, 12,000rpm, WD or Seagate |
| 2 | Server Computer | **Processor:**5 Core, 2.10GHz, 20MB Cache Memory per processor  **RAM:** 8GB, DDR4 SDRAM,1866MHz  **HDD:** 1TB, SATA 6GB/s, 12,000rpm, SCSI | **Processor:** 8 Core, 3.20GHz and above speed with 20MB Cache Memory per processor  **RAM:** 12GB and expandable to 16GB, DDR4 SDRAM, 1866MHz or above capacity  **HDD:** 2TB and expandable, SATA 6GB/s, 12,000rpm, SCSI |
| 3 | Smart phone | Android OS based smart Phone having version API 21 (Lollipop)  2 Gigabytes of RAM  Storage 8GB | Android OS based smart Phone having version API 27 (Oreo)  4 Gigabytes of RAM  Storage 32 Gb |
| 4 | Printer | HP LaserJet 2010 | HP LaserJet 2010 |
| 5 | Switch | 2 CISCO Small Business 200 Series, 10/100/1000 Base-TX | 5 CISCO Small Business 200 Series, 10/100/1000 Base-TX |
| 6 | Network Cable | CAT-5 | CAT-6 |
| 7 | Internet Connection | 500kbps bandwidth | 1mbps bandwidth |

* + 1. **Performance characteristics**

The system should respond to a user’s request for information in less than 0.1 sec. during peak time and 0.01 sec during normal time. The E Health System will allow users to access the system every 24/7. There are no size constraints on the data to be processed by the system.

* + 1. **Error handling and extreme conditions**
* System handles input errors and exceptions by pop-up or alert message to notifying the users.
* Admin should continuously maintain it.
* Power failures handled by Stabilizers.
* Hardware failure switch, Router, network cable, printer, Storage, Database problem, Network problems informing the user about the problem
  + 1. **System Interfacing**
  + The system interface has user-friendly features.
  + There will be a GUI that will be used to interact with the web application and a smart phone.
  + Focus on delivering messages. A void busy UI.
  + The website design should clearly explain the purpose of the service.
    1. **Quality issues**

The main things for the system to be reliable to run it have backup storage.

The system will trap all possible faults runtime and user invalid input error with its error handling method pops-up a message box or alert to the user what is happened, why is happened and what should be done and take care of about the fault.

The acceptable system downtime per 24-hour period will be the total of 45 minutes

* + The downtime when the system is unavailable, the time that a system fails the system will restart in less than 20 minutes all services must be activated.
    1. **System modifications**
  + As the number of users increases from time to time, technology changes and improvements will have an effect on the different hardware’s and software’s of the system, some features might require to be changed. So that the system will face performance problems, and the user also needs a user-friendly interface for the system to used, and then a change has to be made on some of the hardware, forms, user interface and additional features.
    1. **Physical environment**

The target equipment operates in a personal computer and server. On the first deployment of the system is in the organization single place and the system accessed from different sides of its user. The server will be located in a safe, appropriate room in which it may not be affected by dusts, unconditional room temperature, any person unless it is required maintenance.

* + 1. **Security issues**

Access to the system will be protected by user login screen that requires a user name and password.

Unverified users can’t access the system

The system’s back-end server shall only be accessible to authenticated person that is the admin.

The system will be secure and maintain physically.

* + 1. **Resources and management issues**

The system is backed by the sensitivity of the number of users that are using the system and information that is stored on the database initially for the first round the database is backed by monthly base later it can be modified.

System Administrator is responsible to:

System installation: managing, troubleshooting, licensing, and updating hardware and software assets.

Backup: implement backup database weekly, with at least daily transaction log backups.

Maintenance: Required to troubleshoot and fix issues that compromise system performance or access. By using this all method to manage the system.

* + 1. **Reliability**
* The software will not be able to connect to the centralized database in the event that the internet network fails or in the event of the server being down due to a hardware or software failure
* Components of the project code shall be tested alongside the implementation phase to ensure that they are functional.
* Final, integrated project Code shall be tested with any testing tool to ensure that greater than or equal to 80% of the integrated code is covered at run-time, and is functioning properly. The remaining 20% will be inspected through manual testing to ensure the highest chance of being quality code.
  1. **Constraints (“Pseudo requirements”)**

The pseudo requirement of the proposed system of E health System is defined and listed below:

* The front-end should be implement by using JavaScript for Web application
* The backend system should be implement using PHP for Web application
* The database should be implemented using MySQL database and also web application must be use MySQL, and PHP dynamic page for server-side scripting language
  1. **System models**
     1. **Scenarios**

**Table 2 Login/Logout Scenario**

|  |  |
| --- | --- |
| Scenario Name | User Log In/ Logout |
| Actors | Admin, Patients and Doctors  1. The user initiates the application.  2. The system redirects the user to the log in page.  3. The users inter the email and password.  4. The system checks the log in user and redirect to the home page.  5. The user will Logout after use |
| Flow of events |

**Table 3 Make Appointment**

|  |  |
| --- | --- |
| Scenario Name | Make Appointment |
| Actors  Flow of events | Patients   1. The user initiates the application. 2. The system redirects the user to the log in page. 3. The users inter the email and password. 4. The system checks the log in user and redirect to the home page. 5. The patient chooses doctors 6. Then Make appointment by choosing is comfortable day and time. 7. The System accepts the data 8. The user will Logout after use |

**Table 4 Follow up with doctor**

|  |  |
| --- | --- |
| Scenario Name | Follow up with doctor |
| Actors  Flow of events | * Patient  1. The user initiates the application. 2. The system redirects the user to the log in page. 3. The users inter the email and password. 4. The system checks the log in user and redirect to the home page. 5. Patient choose his/her doctor send questions or his/her status. 6. The System accepts the data. 7. The user will Logout after use |

**Table 5 Doctor Appointment Response**

|  |  |
| --- | --- |
| Scenario Name | Doctor Appointment Response |
| Actors  Flow of events | Doctors   1. The user initiates the application. 2. The system redirects the user to the log in page. 3. The users inter the email and password. 4. The system checks the log in user and redirect to the home page. 5. The doctor will approve or decline the appointment 6. The System accept the data. 7. The user will Logout after use |

**Table 6 Follow up Patient**

|  |  |
| --- | --- |
| Scenario Name | Follow up Patient |
| Actors  Flow of events | Doctors   1. The user initiates the application. 2. The system redirects the user to the log in page. 3. The users inter the email and password. 4. The system checks the log in user and redirect to the home page. 5. The doctor will answer the patient’s question or give information, 6. The System accept the data. 7. The user will Logout after use |

**Table 7 View Data**

|  |  |
| --- | --- |
| Scenario Name | View Data |
| Actors | Patient, Doctor, Admin |
| Flow of events | 1. The user initiates the application. 2. The system redirects the user to the log in page. 3. The users inter the name and password. 4. The system checks the log in user and redirect to the home page. 5. The user can view permissible data. 6. The user will Logout after use. |

**Table 8 Add Doctor**

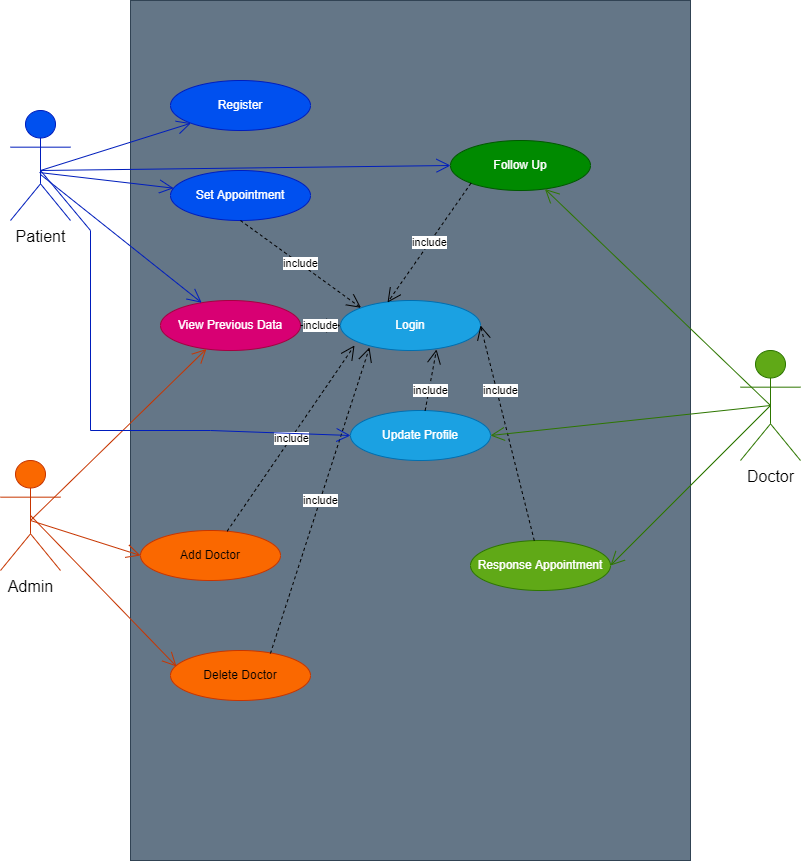
|  |  |
| --- | --- |
| Scenario Name | Add Doctor |
| Actors | Admin |
| Flow of events | 1. The user initiates the application. 2. The system redirects the user to the log in page. 3. The users inter the name and password. 4. The system checks the log in user and redirect to the home page. 5. The user will add new doctor to the system. 6. The System will save data 7. The user will Logout after use. |

**Table 9 Delete Doctor**

|  |  |
| --- | --- |
| Scenario Name | Delete Doctor |
| Actors | Admin |
| Flow of events | 1. The user initiates the application. 2. The system redirects the user to the log in page. 3. The users inter the name and password. 4. The system checks the log in user and redirect to the home page. 5. The user will delete doctor from the system. 6. The System will save data 7. The user will Logout after use. |

* + 1. **Use case model**

**Figure 1 Use case Model**



1. **Use case description for user registration**

|  |  |
| --- | --- |
| **Name** | User Registration |
| **Actor** | Patient |
| **Entry Conditions** | The user registration page should be loaded |
| **Flow of Events** | 1. 1. The user initiates the system. 2. 2. The system redirects to the Registration page. 3. 3. User click on signup button 4. 4. User to registration form page which include Full name,   Email, Emergence Name, Emergence Contact, Age, Contact, Password   1. 5. The user fills the registration form page and submits (Register). 2. 6. The systems prompt the home page. |
| **Exit Conditions** | User is registered with detail information |

1. **Use case Description for Login**

|  |  |
| --- | --- |
| Name | User Log In |
| Actors | Patient, Doctor, Admin |
| Entry Conditions | The Log in Page should be Loaded. |
| Flow of events | 1. The user initiates the application. 2. The system redirects the user to the log in page. 3. The users enter the email and password. 4. The system checks the log in user and redirect to the home page. |
| Exit Condition | email and Password are validated |

1. **Use case Set Appointment**

|  |  |
| --- | --- |
| Name | Set Appointment |
| Actors | Patient |
| Entry Conditions | The Log in Page should be Loaded. |
| Flow of events | 1.The user initiates the application.  2.The system redirects the user to the log in page.  3.The users enter the name and password.  4.The system checks the log in user and redirect to the home page.  5. Choose Doctor  6. Pick comfortable date and time set Appointment |
| Exit Condition | Set the appointment to doctor |

1. **Use case Follow up Doctor**

|  |  |
| --- | --- |
| Name | Follow up Doctor |
| Actors | Patient |
| Entry Conditions | The Log in Page should be Loaded. |
| Flow of events | 1.The user initiates the application.  2.The system redirects the user to the log in page.  3.The users enter the name and password.  4.The system checks the log in user and redirect to the home page.  5. Choose Doctor  6. Send question or his/her status to doctor. |
| Exit Condition | * Send to doctor |

1. **Use case View Data**

|  |  |
| --- | --- |
| Name | View Data |
| Actors | Patient |
| Entry Conditions | The Log in Page should be Loaded. |
| Flow of events | 1. The user initiates the application.  2. The system redirects the user to the log in page.  3. The users enter the name and password.  4. The system checks the log in user and redirect to the home page.  5. Choose Previous appointment or Previous follow up option |
| Exit Condition | * View selected data |

1. **Use case Response Appointment**

|  |  |
| --- | --- |
| Name | Response Appointment |
| Actors | Doctor |
| Entry Conditions | The Log in Page should be Loaded. |
| Flow of events | 1. The user initiates the application.  2. The system redirects the user to the log in page.  3. The users enter the name and password.  4. The system checks the log in user and redirect to the home page.  5. Approve or Decline the Appointment that made by patient |
| Exit Condition | * Response to the Appointment. |

1. **Use case Follow Up Patient**

|  |  |
| --- | --- |
| Name | Follow Up Patient |
| Actors | Doctor |
| Entry Conditions | The Log in Page should be Loaded. |
| Flow of events | 1. The user initiates the application.  2. The system redirects the user to the log in page.  3. The users enter the name and password.  4. The system checks the log in user and redirect to the home page.  5. Response the Patient’s Question |
| Exit Condition | * Response to the patient. |

1. **Use case Add Doctor**

|  |  |
| --- | --- |
| Name | Add Doctor |
| Actors | Admin |
| Entry Conditions | The Log in Page should be Loaded. |
| Flow of events | 1. The user initiates the application.  2. The system redirects the user to the log in page.  3. The users enter the name and password.  4. The system checks the log in user and redirect to the home page.  5. choose Add Doctor option.  6. Fill all required Field and submit. |
| Exit Condition | * Register new doctor |

1. **Use case Delete Doctor**

|  |  |
| --- | --- |
| Name | Delete Doctor |
| Actors | Admin |
| Entry Conditions | The Log in Page should be Loaded. |
| Flow of events | 1. The user initiates the application.  2. The system redirects the user to the log in page.  3. The users enter the name and password.  4. The system checks the log in user and redirect to the home page.  5. choose Edit Doctor option.  6. the System will list Register doctor then press delete option. |
| Exit Condition | * Delete doctor |

* + 1. **Object model**

List of the fundamental objects:

* **Entity Object**: Patient, Doctor, Admin
* **Boundary Object**: registration form, login form, appointment form, follow up form,
* **Control Object**: register User, User login, set appointment, follow up, add doctor, edit doctor.
  + 1. **Data dictionary**

**Table 10 Patient Register**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Patient Register** | | | | |
| Attributes | Data Type | Index | Null Value | Description |
| P\_id | Int(11) | Primary key | Not |  |
| Fname | varchar(100) |  | Not | First name |
| Lname | varchar(100) |  | Not | Last name |
| Email | varchar(100) |  | Not |  |
| Password | varchar(100) |  | Not |  |
| Contact | varchar(100) |  | Not |  |
| Age | Int(3) |  | Not |  |
| Emerg\_name | varchar(100) |  | Not | Emergence  name |
| Emerg\_contact | varchar(100) |  | Not | Emergence contact |

**Table 11 Doctor**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Doctor** | | | | |
| Attributes | Data Type | Index | Null Value | Description |
| D\_id | Int(11) | Primary key | Not |  |
| Fname | varchar(100) |  | Not | First name |
| Lname | varchar(100) |  | Not | Last name |
| Email | varchar(100) |  | Not |  |
| Password | varchar(100) |  | Not |  |
| Contact | varchar(100) |  | Not |  |
| Department | varchar(100) |  | Not |  |
| img | blob() |  | Not | Doctor profile image |

**Table 12 Appointment**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Appointment** | | | | |
| Attributes | Data Type | Index | Null Value | Description |
| app\_id | Int(11) | Primary key | Not |  |
| P\_id | int(100) | Foreign key | Not | Patient’s id |
| D\_id | int(100) | Foreign key | Not | Doctor’s id |
| pn | varchar(100) |  | Not | Patient’s name |
| Dn | varchar(100) |  | Not | Doctor’s name |
| Date | datetime(2) |  | Not | Schedule |
| permission | varchar(100) |  | Not | Status(approved or decline) |

**Table 13 Follow up**

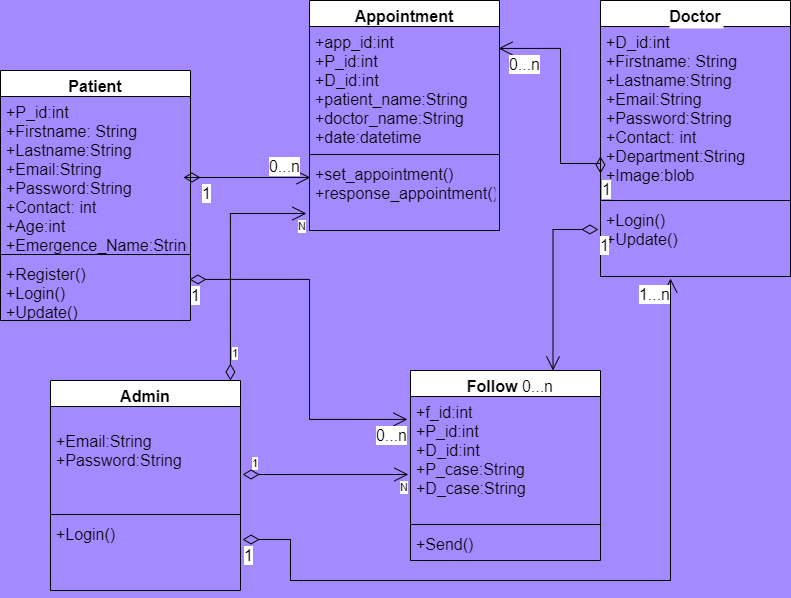
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Follow up** | | | | |
| Attributes | Data Type | Index | Null Value | Description |
| f\_id | Int(11) | Primary key | Not |  |
| P\_id | int(100) | Foreign key | Not | Patient’s id |
| D\_id | int(100) | Foreign key | Not | Doctor’s id |
| P\_case | varchar(100) |  | Not | Patient’s question |
| D\_ans | varchar(100) |  | Not | Doctor’s answer |

**Table 14 Admin**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Admin** | | | | |
| Attributes | Data Type | Index | Null Value | Description |
| id | Int(11) | Primary key | Not |  |
| Fname | Varchar(100) |  | Not |  |
| email | Varchar(100) |  | Not |  |
| password | Varchar(100) |  | Not |  |

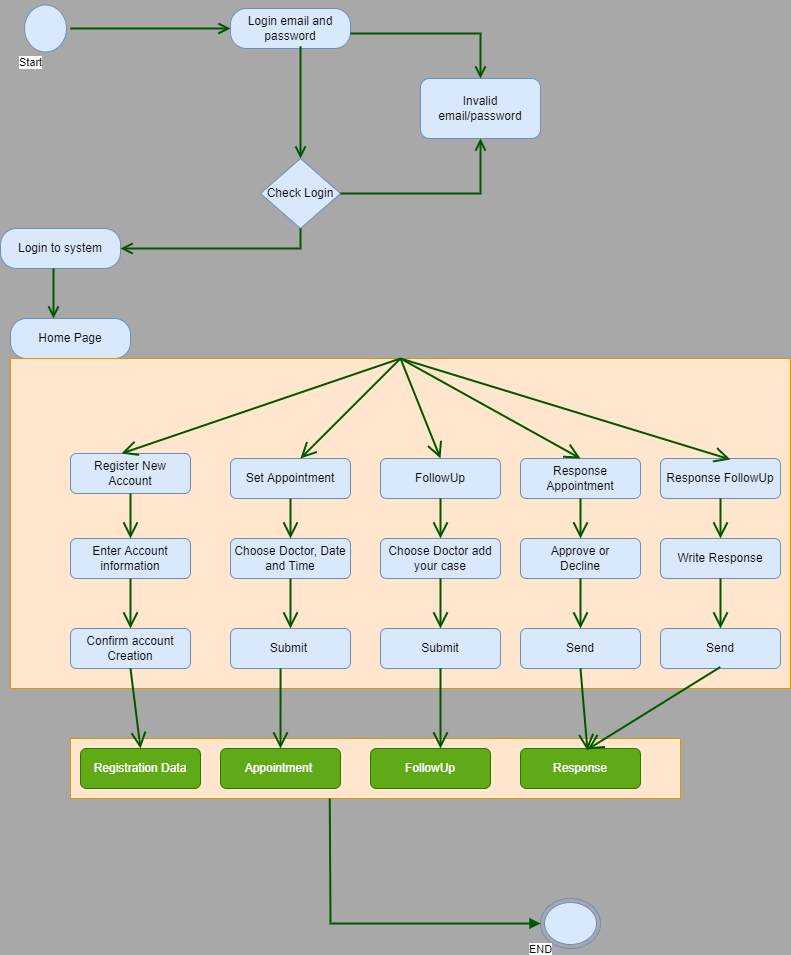
* + 1. **Class diagrams**

classes, associations, attributes and operations

Figure 2 Class Diagram

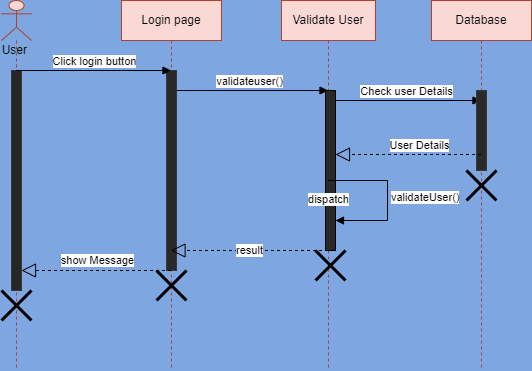
* + 1. **Dynamic models**
       1. **Sequence diagram**

**Figure 3 state diagram**

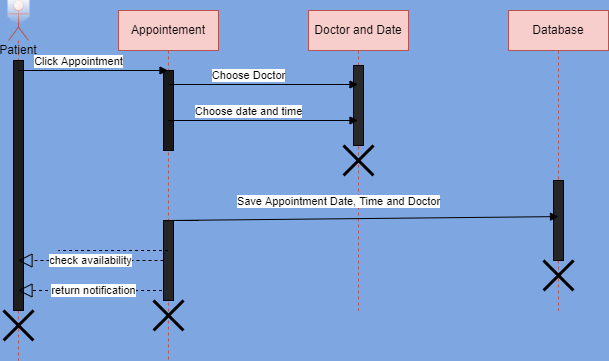
****

* + - 1. **Sequence diagram**

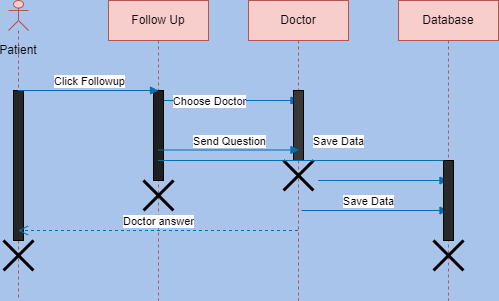
**Figure 4** **Sequence diagram for Login**



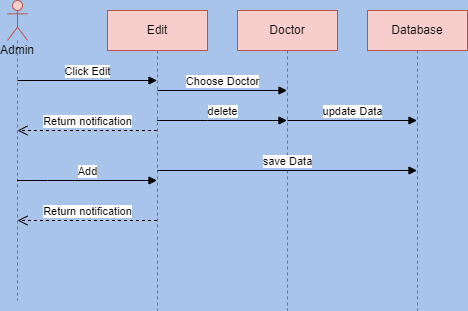
**Figure 5 Sequence diagram for Appointment**



**Figure 6 Sequence diagram for Patient Follow Up**

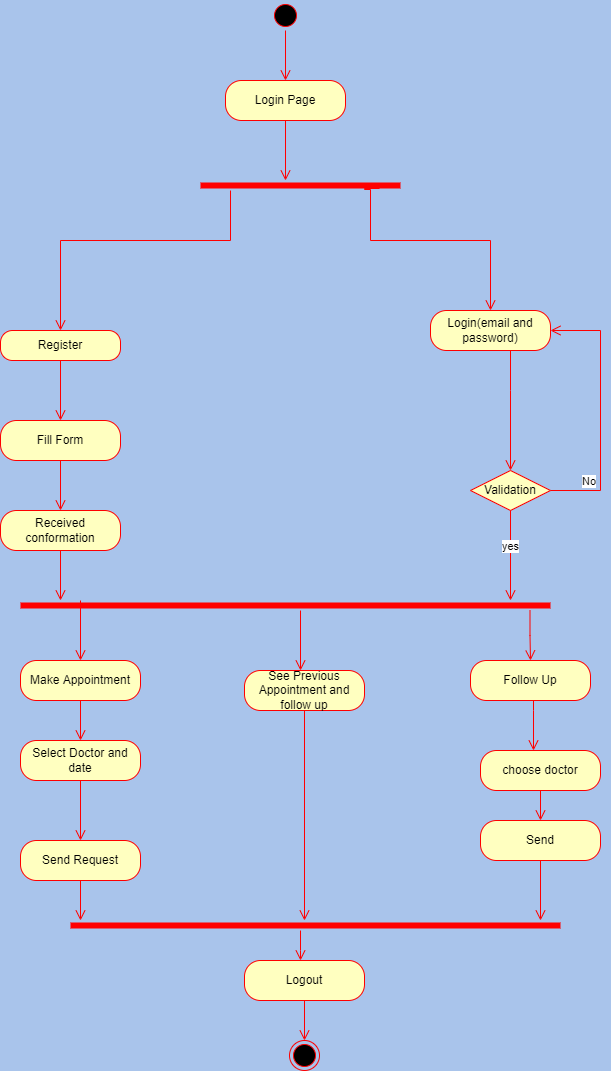


**Figure 7 Sequence diagram for admin**

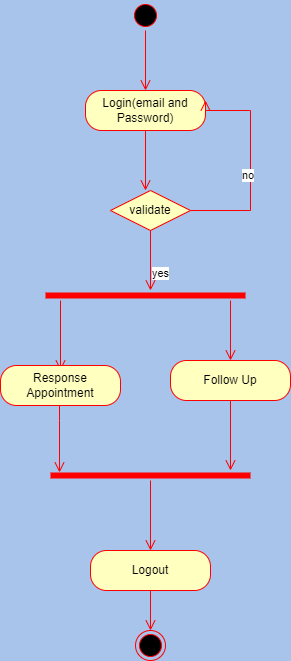


* + - 1. **Activity Diagram**

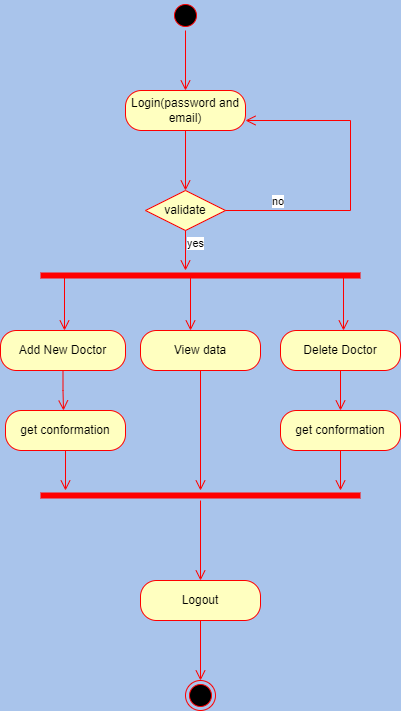
**Figure 8 Activity diagram for patient**

****

**Figure 9 Activity diagram for doctor**

****

**Figure 10 Activity diagram for admin**

****

1. **Glossary**
2. **System:** A set of inter-connected computers operating together in the same place at the same time.
3. **Admin:** A person in charge of the installation, management, configuration, information, and maintenance of software on a computer, network, and system, as well as system user access controls.
4. **Pseudo requirements:** The client or the environment in which the system is to operate imposes restrictions.
5. **Action:** A basic unit of behavior. Actions have the ability to accept a set of inputs, produce a set of outputs, and change the state of the system. Actions are basic in that they cannot be broken down any more.
6. **Scenario:** A short text description on the system function
7. **Use case:** An action carried out by a computer system.
8. **Diagram:** A basic diagram that depicts a machine, system, idea, or other object, and is frequently used to describe how it works.
9. **Class diagram:** Is a form of static structure diagram that depicts a system's structure by displaying the system's classes, attributes, processes, and object relationships.
10. **Sequence Diagram:** A diagram which represents sequence of action that human or system represents.
11. **Dynamic Model:** The object model's behavior in terms of state machine diagrams and sequence diagrams is documented in this paper. It describes the system's components that exhibit intriguing behavior. State machine diagrams, sequence diagrams, and activity diagrams are all included in Dynamic Model
12. **Activity diagram:** a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

# Supporting Information

* Object Oriented Software Engineering -Using UML, Patterns, and Java; Bernd Bruegge& Allen H. Dutoit; Published by Pearson Education Inc., 3rd Edition.
* Object Oriented Software Engineering –Practical Software Development Using UML and Java; Timothy C. Lethbridge& Robert Laganiere; Published by McGraw-Hill Education Limited. 2nd Edition.
* UML Distilled -A Brief Guide to The Standard Object Modeling Language; Martin Fowler -Forewords by CrisKobryn, Grady Booch, Ivar Jacobson, and Jim Rumbaugh, 3rd Edition.
* System Analysis and Design Methods; Professor Jeffrey L. Whitten & Professor Lonnie D. Bentley; Published by McGraw-Hill Education Limited. 7th Edition.
* System Analysis and Design; Alan Dennis & Barbara Haley Wixom &Roberta M. Roth; Published by John Wiley & Sons Inc., 5th Edition.