**NEUROLOGIST’S DAILY APPOINTMENT SCHEDULER DESIGN**

# For the Degree of

# Bachelor of Engineering in

# Computer Science and Engineering

# By

**Swayambhu Kurdukar**

**Mandar Kulkarni**

**Pratik Nage**

**Yogesh Kanhegawankar**

Under the Guidance of

**Prof.S.D.Ghode**



Department of Computer Science and Engineering

**Marathwada Institute of Technology, Aurangabad**

**Maharashtra State, India**

**2018-2019**

**A**

**Project Report**

**on**

**NEUROLOGIST’S DAILY APPOINTMENT SCHEDULER DESIGN**

Submitted by

**Swayambhu Kurdukar**

**Mandar Kulkarni**

**Pratik Nage**

**Yogesh Kanhegawankar**

In partial fulfillment for the award of

# Bachelor of Engineering in

# Computer Science and Engineering

Guided by

**Prof.S.D.Ghode**

Department of Computer Science and Engineering

**Marathwada Institute of Technology, Aurangabad**

**Maharashtra State, India**

**2018-2019**

**CERTIFICATE**

This is to certify that, the project entitled **“Neurologist’s Daily Appointment Scheduler Design”,** which has been submitted herewith in the partial fulfillment for the award of the **‘Bachelor of Engineering’** in **‘Computer Science and Engineering’** of Dr. BabasahebAmbedkarMarathwada University, Aurangabad (M.S.). This is the result of the original work and contribution by **Swayambhu Kurdukar, Mandar Kulkarni, Pratik Nage, Yogesh Kahnegawankar** under my supervision and guidance.

Place: Aurangabad

Date:

**Prof.S.D.Ghode Dr. Radhakrishna Naik**

**Guide**  **Head**

### Department of Computer

### Science and Engineering

Dr.N. G. Patil

**Principal**

**Marathwada Institute of Technology**

Aurangabad (M.S.) - 431 005

Acknowledgement

We take this opportunity to express our sincere gratitude to all those who helped us in various capacities in undertaking this project and devising the report.

We firstly express our gratitude to respected **Dr. Nilesh Patil**,Principal,Marathwada Institute of Technology for their encouragement.

We are privileged to express our gratitude to our respected teacher, **Prof. S.Ghode,** whose unparalleled knowledge, moral fiber and judgment along with her know-how was an immense support in completing the project.

We are also grateful to **Dr. Radhakrishna Naik**, the Head of the Dept., Computer Science and Engineering, for brainwave and encouragement given.

We take this opportunity to thank our friends and contemporaries for their cooperation and compliance.

Mr. Swayambhu Kurdukar

Mr. Mandar Kulkarni

Mr. Pratik Nage

Mr. Yogesh Kanhnegawankar

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

FRONTPAGE

CERTIFICATE

ACKNOWLEDGEMENT

ABSTRACTII

LIST OF TABLES III

CHAPTER 1 INTRODUCTION I

REQUIREMENT ANALYSIS

SYSTEM ANALYSIS AND DESIGN

TESTING

FUNCTIONAL REQUIREMENT

NON FUNCTIONAL REQUIREMENT

CONCLUSION

REFERENCE

**ABSTRACT**

Neurologist daily scheduler software  for Hospital provides a systematic, priority-based approach to planning, scheduling, allocating, tracking and analyzing patients effectively. It is a software to plan and run their patient visits effectively and always on schedule. It can help your organization to follow best practices of managing time. Particularly, this software will let you plan time, develop schedules, and keep track of appointments.

This Appointment management in particular field like Neurology becomes exceptionally important as people are entrusting their lives in these doctor’s hands; so slacking behind is not an option. The objective of this project is to manage these doctors time and to give latest possible updates about patient as fast as possible without the need of any special staff to manage the entries of patients who are admitted at the hospital.

II

**LIST OF TABLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Table Number** | **Name of Table** | **Page Number** |
| 1 | 2.1 | Hardware Requirements |  |
| 2 | 2.2 | Software Requirements |  |
| 3 | 2.3 | Functional Requirements |  |
| 4 | 4.1 | Test cases |  |

III

**CHAPTER 1**

**Introduction**

Times becomes most precious commodity a neurologist has; we are developing a Appointment scheduler so as to let a person schedule Appointment without his presence around. Software requirement specification describes all the details describing the various attributes of a Appointment scheduler.

It gives description of the purpose and scope of a Appointment scheduler, about the Appointment that would be performed and the basic framework as to how the Appointment are intended to be scheduled with respect to the stimulated time and order. This SRS delineates the aspects such as reference and also compares the existing system with the proposed system.

System environment has been described and comprises of description of the hardware interface, software interface, and communication interface. It gives us insight as to what the application interface and input output interface would look like and the way in which they would function. The change management process has been taken into consideration and plans have been formulated regarding what steps would be taken at the time when changes would be demanded after the completion of project.

Neurologist are medical doctors who specialize in diagnosing and treating conditions related to central and peripheral nervous system. They mainly sees patients who have been referred by primary care provider, but occasionally they come from emergency room, physician referrals or referrals from other specialists.

Effective scheduling lets decision makers allocate available human resources to Appointments in a way that ensures on-time completion of related business goals. **Appointment scheduler** software helps you assign an appropriate number of workers to each Appointment, set activity time-frames and durations, create job schedules, prioritize todo lists, receive automatic notifications.

**SOFTWARE REQUIREMENT**

The purpose of Software Requirement Specification (SRS) document is to describe the external behavior of the Appointment scheduler. Requirement specification defines and describe the operations, interfaces; performance of the Appointment scheduler. The document also describes the other requirements such as performance requirements. The Software Requirements Specifications (SRS) captures the complete software requirement for the system, or a portion of system.

Also, this would help us to gain tremendous knowledge and experience doing a demanding project like this by practically implementing the theories we learned in our classes. This document has the development team to review the requirements so as to design and implement the system effectively.

A Appointment scheduler needs to execute Appointments according to the schedule decided beforehand in a synchronized manner in order to save the precious time of user.

This SRS includes brief description of software and its features. The scheduler would have option for scheduling, cancelling and updating various Appointments. It automates the routine Appointments thereby making the system user friendly.

The Appointment scheduler can automate many of user’s routine Appointment. Automatic popup reminders let you not forget important things that you were planning to get done.

**1.2 The Benefits of Effective Appointment Scheduling**

If there were no detailed schedule, it would be difficult to link Appointment assignments to working hours. You would have just a todo list with multiple Appointments assigned to you, but you couldn't realize when each one should be started and what deadlines should be met. Then lack of scheduling would make you inefficient at the workplace, and most likely you would just waste your working hours, rather than deliver expected value to your company. Besides, you couldn't collaborate with your colleagues and management.

So the greatest advantage of having a detailed job schedule in place is predictability. Your daily schedule tells you what Appointments are to be done, what deadlines are to be met, and what priorities are to be followed. This document allows you to predict your possible steps and plan for effective use of resources, tools and equipment. Other possible benefits of Appointment scheduling are as follows:

* Daily work is planned according to strategic priorities
* Decreased turnaround time
* Improved communications and collaboration with personnel
* Avoidance  of overloaded and underused resources
* Job delays and schedule deviations are more readily apparent
* An ability to review and forecast personnel effectiveness

**1.3 Why Use Appointment Scheduler**

All of these benefits can be achieved with use of effective Appointment scheduler software. For example, in Appointment Manager you can use the Calendar view to create and manage job schedules. The program lets display Appointments on each user's calendar, so you can get multiple calendars displayed on a single screen. This feature makes it easier to review and manage several schedules at once. In addition, you can switch between daily, weekly and monthly modes to easily schedule short-term, mid-term and long-term Appointments. In addition, you can use the timeline mode to get your Appointments displayed on a time scale view.

**1.4 Priorities, Time-frames & Due Dates to Schedule Todo Lists**

Priorities, time-frames and due dates help make job scheduling more accurate and aligned with business objectives. Prioritization lets you organize and rank your Appointments by their urgency and importance, so that you will know what items in your todo list should come up first. Time-frames tell you expected start and finish time or duration for each todo item - you can use this data to plan your actions [more effectively](http://www.mindtools.com/pages/article/newHTE_07.htm). Due dates can be used to link your todo items to business goals and projects. In Appointment Manager you can set different priority levels (from Urgent and Highest to Normal and Lowest), due dates and start & finish time for your Appointments. The software lets make your todo list easier to read and understand - you can group all your Appointments by the Priority column in the Appointment List view.

**1.5 Reminders and Notifications to Track Events and Changes**

Poor productivity and missed deadlines can be caused by forgetfulness. Even if you have a detailed schedule and a prioritized todo list you can't remember about each item that must be done. Certainly you can look at your daily Appointments and due dates to know what assignments you need to accomplish today. However, when you are overloaded or stressed you can just miss something important in your todo list or forget about today's meeting or urgent phone call.

Automatic reminders and notifications will be the right solution. For example, when you schedule something important in Appointment Manager, you can set a reminding alert that will come up before your Appointment's Start time/Finish time/Due date or at exact date and time. The [Appointment scheduler](http://en.wikipedia.org/wiki/Windows_Task_Scheduler) sends reminders right to your inbox and desktop. In case you want to track changes made to your schedule, you can set the program to send notifications of updates.

**Chapter 2**

**Requirement Analysis**

This section lists the software and hardware functional requirements.

2.1 Hardware Requirements

2.1.1 The Laptop

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Requirement** | **Description of the requirement** | **Implementation** |
| **Number** | **Category** |  | **Phase** |
|  |  |  |  |
| 2.1.1.1 | Laptop Speed | The laptop may have a processing | Phase 1 |
|  |  | speed of at least 2.4 Hz. |  |
|  |  |  |  |
| 2.1.1.2 | Laptop RAM | The mobile should have RAM, | Phase 1 |
|  |  | which is a form of data storage that |  |
|  |  | stores data and machine code |  |
|  |  | currently being used. |  |
|  |  | The laptop may have a RAM of at |  |
|  |  | least 4 GB. |  |
|  |  |  |  |
| 2.1.1.3 | Laptop OS | The laptop may have Windows OS | Phase 1 |
|  |  | as its operating system. |  |
|  |  |  |  |
| 2.1.1.4 | Laptop | The laptop may have a processor of | Phase 1 |
|  | Processor | any company like intel, AMD,etc. |  |
|  |  |  |  |
| 2.1.1.5 | Laptop System | The laptop may have system type as | Phase 1 |
|  | Type | 64-bit OS, x64-based processor. |  |
|  |  |  |  |

2.1.2 The Mobile

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Requirement** | **Description of the requirement** | **Implementation** |
| **Number** | **Category** |  | **Phase** |
|  |  |  |  |
| 2.1.2.1 | Mobile Speed | The mobile may have a processing | Phase 1 |
|  |  | speed of at least 2.4 Hz. |  |
|  |  |  |  |
| 2.1.2.2 | Mobile RAM | The mobile should have RAM, | Phase 1 |
|  |  | which is a form of data storage that |  |
|  |  | stores data and machine code |  |
|  |  | currently being used. |  |
|  |  | The mobile may have a RAM of at |  |
|  |  | least 2 GB. |  |
|  |  |  |  |
| 2.1.2.3 | Mobile ROM | ROM is a type of non-volatile | Phase 1 |
|  |  | memory used in computer and all |  |
|  |  | electronic devices. |  |
|  |  | The mobile may have a ROM of at |  |
|  |  | least 16 GB. |  |
|  |  |  |  |
| 2.1.2.4 | Mobile OS | The mobile may have Android OS | Phase 1 |
|  |  | or as its operating system. |  |
|  |  |  |  |

2.2 Software Requirements

2.2.1 The Software

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Requirement** | **Description of the requirement** | **Implementation** |
| **Number** | **Category** |  | **Phase** |
|  |  |  |  |
| 2.2.2.1 | Database | Appointment scheduling system | Phase 1 |
|  |  | Shall be installed in the |  |
|  |  | Machine. |  |
| 2.2.2.2 | LAMP | Linux ,Apache,Mysql,PHP server | Phase 1 |

2.3 Functional requirements

2.3.1 The Web App

2.3.1.1 Doctor’s Login Module

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Requirement** | **Description of the requirement** | **Implementation** |
| **Number** | **Category** |  | **Phase** |
|  |  |  |  |
| 2.3.1.1.1 | Login Page | The Login page shall be displayed | Phase 1 |
|  |  | on the load of the web app. |  |
|  |  |  |  |
| 2.3.1.1.2 | Username | The Page shall contain a username | Phase 1 |
|  |  | Text box to type in the Username. |  |
|  |  |  |  |
| 2.3.1.1.3 | Password | The Page shall contain a Password | Phase 1 |
|  |  | Text box to type in the Password. |  |
|  |  |  |  |
| 2.3.1.1.4 | Login Button | The page shall contain a Login | Phase 1 |
|  |  | button below the boxes to enter into |  |
|  |  | the main pages. |  |
|  |  |  |  |
|  |  |  |  |

2.3.1.2 The Appointment Module

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Requirement** | **Description of the requirement** | **Implementation** |
| **Number** | **Category** |  | **Phase** |
|  |  |  |  |
| 2.3.1.2.1 | Appointment Page | The appointment page shall be | Phase 1 |
|  |  | Displayed on the load of the web app. |  |
|  |  |  |  |
| 2.3.1.2.2 | First name | The Page shall contain a firstname | Phase 1 |
|  |  | Text box to type in the First name of the user. |  |
|  |  |  |  |
| 2.3.1.2.3 | Last name | The Page shall contain a lastname | Phase 1 |
|  |  | Text box to type in the Last name of the user. |  |
|  |  |  |  |
| 2.3.1.2.4 | Email id | The page shall contain a email id Text box to type in the Email id of | Phase 1 |
|  |  | the user. |  |
|  |  |  |  |
| 2.3.1.2.5 | Mobile Number | The page shall contain a mobile number Text box to type in the | Phase 1 |
|  |  | Mobile number of the user. |  |
|  |  |  |  |
| 2.3.1.2.6 | Date Of Appointment | The page shall contain a date Text box to type in the date of appointment of user | Phase 1 |
|  |  |  |  |
| 2.3.1.2.7 | Time Of Appointment | The page shall contain a time of appointment which will set time of appointment of the user. | Phase 1 |
| 2.3.1.2.8 | Gender | The page shall contain a gender of user. | Phase 1 |
| 2.3.1.2.9 | DOB | The page shall contain a DOB of user. | Phase 1 |
| 2.3.1.2.10 | Address | The page shall contain a address of user. | Phase 1 |
| 2.3.1.2.11 | Doctor | The page shall contain a doctor’s name and their respective hospital. | Phase 1 |
| 2.3.1.2.12 | Symptoms | The page shall contain a various symptoms that can be experienced by user. | Phase 1 |

2.3.1.3 The Dashboard Module

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Requirement** | **Description of the requirement** | **Implementation** |
| **Number** | **Category** |  | **Phase** |
|  |  |  |  |
| 2.3.1.3.1 | Dashboard | The user shall be | Phase 1 |
|  | Page | directed to the Dashboard page to |  |
|  |  | Select the option |  |
|  |  |  |  |
| 2.3.1.3.2 | Book Appointment | Here user can book appointment | Phase 1 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 2.3.1.3.3 | View Appointment | Here user can view booked appointment | Phase 1 |
|  |  |  |  |
| 2.3.1.3.4 | Cancel Appointment | Here user can cancel booked appointment | Phase 1 |
| 2.3.1.3.5 | Doctor’s Login | Here Doctor can log in | Phase 1 |

2.4 Non – Functional requirements

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Requirement** | **Requirement** |  | **Description of the requirement** | **Implementation** |  |
|  |  |  |  |  |  |  |
|  | **Number** | **Category** |  |  | **Phase** |  |
|  |  |  |  |  |  |  |
|  | 2.4.1 | Performance |  | The system must be interactive and | Phase 1 |  |
|  |  |  |  | the delays involved must be less. So |  |  |
|  |  |  |  | in every action – response of the |  |  |
|  |  |  |  | system, there are no immediate |  |  |
|  |  |  |  | delays. |  |  |
|  |  |  |  |  |  |  |
|  | 2.4.2 | Safety |  | Information transmission should be | Phase 1 |  |
|  |  |  |  | securely transmitted to the server |  |  |
|  |  |  |  | without any changes in information. |  |  |
|  |  |  |  |  |  |  |
|  | 2.4.3 | Reliability |  | As the system provides the right | Phase 1 |  |
|  |  |  |  | tools for discussion, problem |  |  |
|  |  |  |  | solving it must be made sure that |  |  |
|  |  |  |  | the system is reliable in its |  |  |
|  |  |  |  | operation and for securing the |  |  |
|  |  |  |  | sensitive details. |  |  |
|  |  |  |  |  |  |  |
|  | 2.4.4 | Usability |  | As the system is easy to handle and | Phase 1 |  |
|  |  |  |  | navigates in the most expected way |  |  |
|  |  |  |  | with no delays. In that case the |  |  |
|  |  |  |  | system program reacts accordingly |  |  |
|  |  |  |  | and transverses quickly between its |  |  |
|  |  |  |  | states. |  |  |
|  |  |  |  |  |  |  |
|  | 2.4.5 | Availability |  | If the internet service gets disrupted | Phase 1 |  |
|  |  |  |  | while sending information to the |  |  |
|  |  |  |  | server, the information can be send |  |  |
|  |  |  |  | again for the verification. |  |  |
|  |  |  |  |  |  |  |
|  | 2.4.6 | Security |  | The main security concerns is for | Phase 1 |  |
|  |  |  |  | users account hence proper login |  |  |
|  |  |  |  | mechanism should be use to avoid |  |  |
|  |  |  |  | hacking. Hence, security will be |  |  |
|  |  |  |  | provided by password |  |  |
|  |  |  |  | authentication. |  |  |
|  |  |  |  |  |  |  |

**COST ANALYSIS**

The Constructive cost model (COCOMO) was developed by Boehm. This model also estimates the total effort in terms of person-months of the technical project staff. The effort estimate includes development, management, and support Appointments but does not include the cost of the secretarial and other staff that might be needed in an organization.

The initial estimate is determined by an equation of the form used in the static single variable models, using KDLOC as the measures of size. To determine the initial effort Ei in person-months the equation used is of the type

Ei=a\*(KDLOC)b.  
The value of the constants a and b depend on the project type. In COCOMO, projects are categorized into three types – organic, semidetached, and embedded.

Ourproject is a Semidetached type.

The basic COCOMO equations for Semidetached type take the form of:

* Effort(E)=ab(KLOC)bb
* Deployment Time(D)=cb(E)db
* Staff Size(SS)=E/D persons
* Productivity(P)=KLOC/E

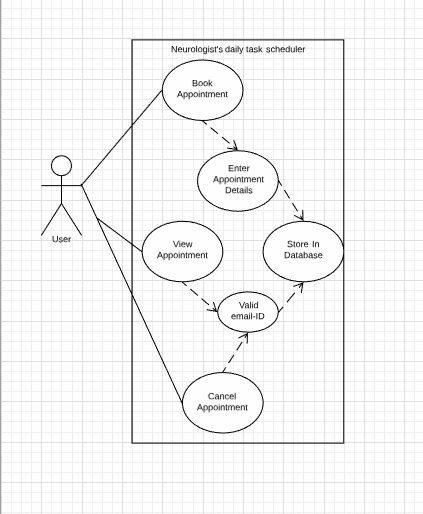
ab, bb , cb ,db are the coefficients here, with their values for semidetached mode as follows

1. Ab=3.0
2. Bb=1.12
3. Cb=2.5
4. Db=0.35

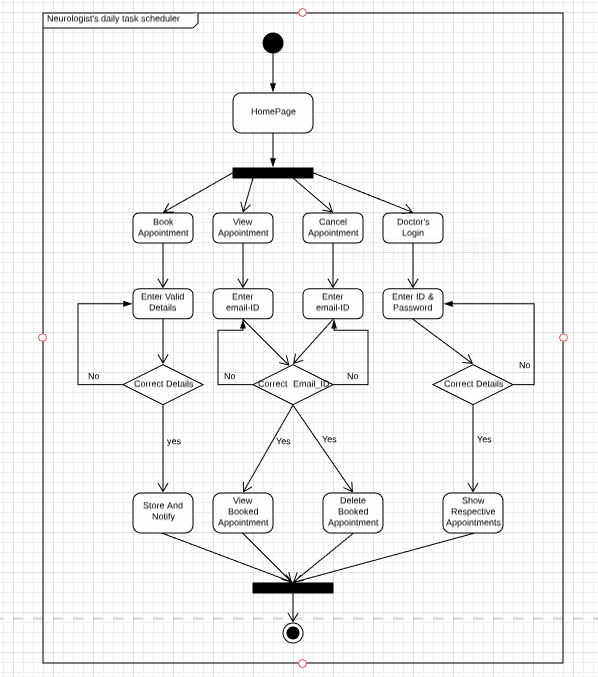
**Chapter 3**

**System Analysis & Design**

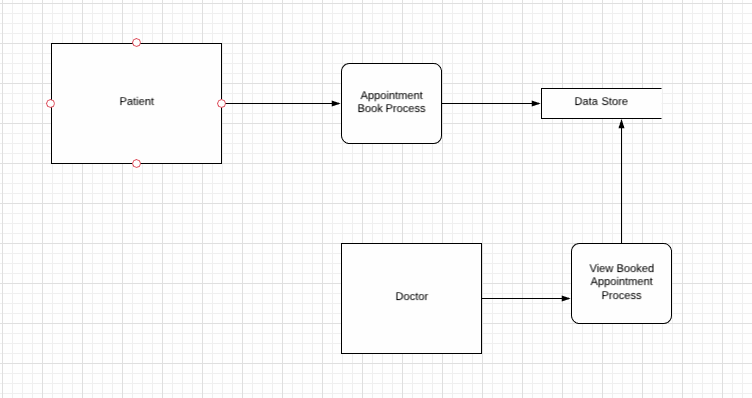
**USE CASE DIAGRAM:**

****

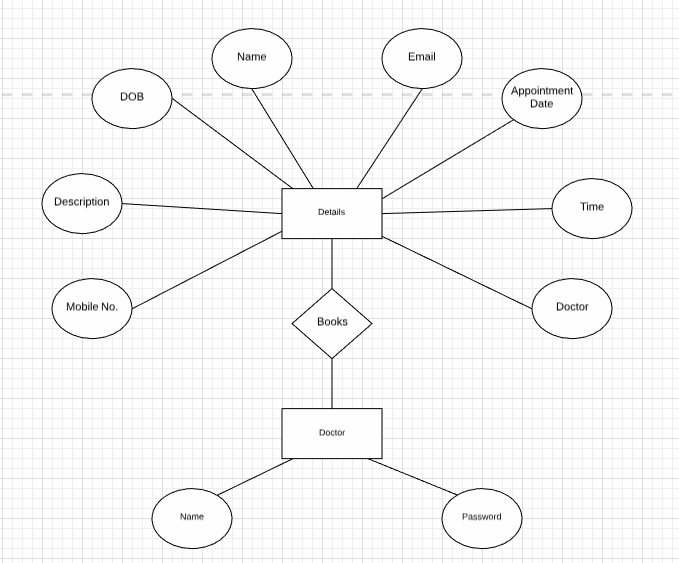
**ACTIVITY DIAGRAM:**

****

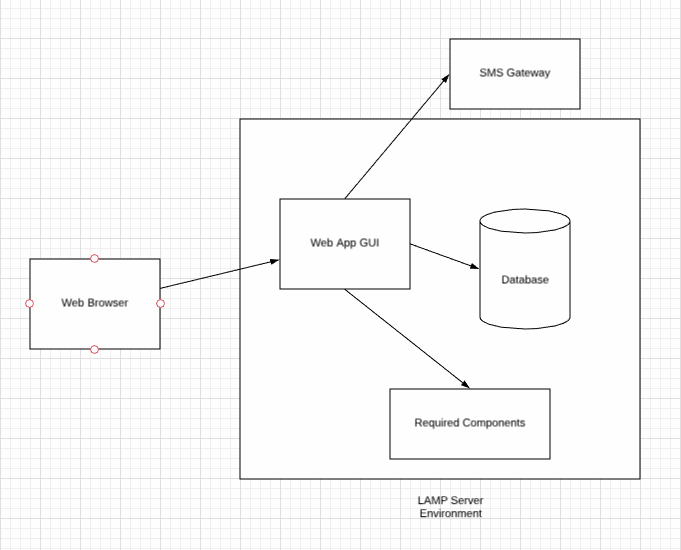
**DFD 0 DIAGRAM:**

****

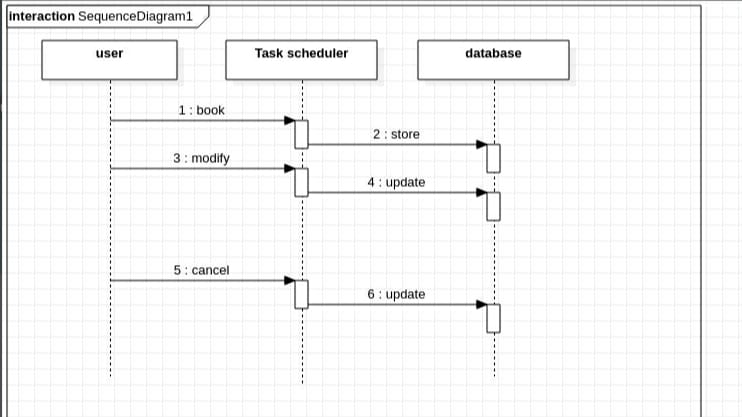
**E-R DIAGRAM:**

****

**ARCHITECTURAL DIAGRAM:**

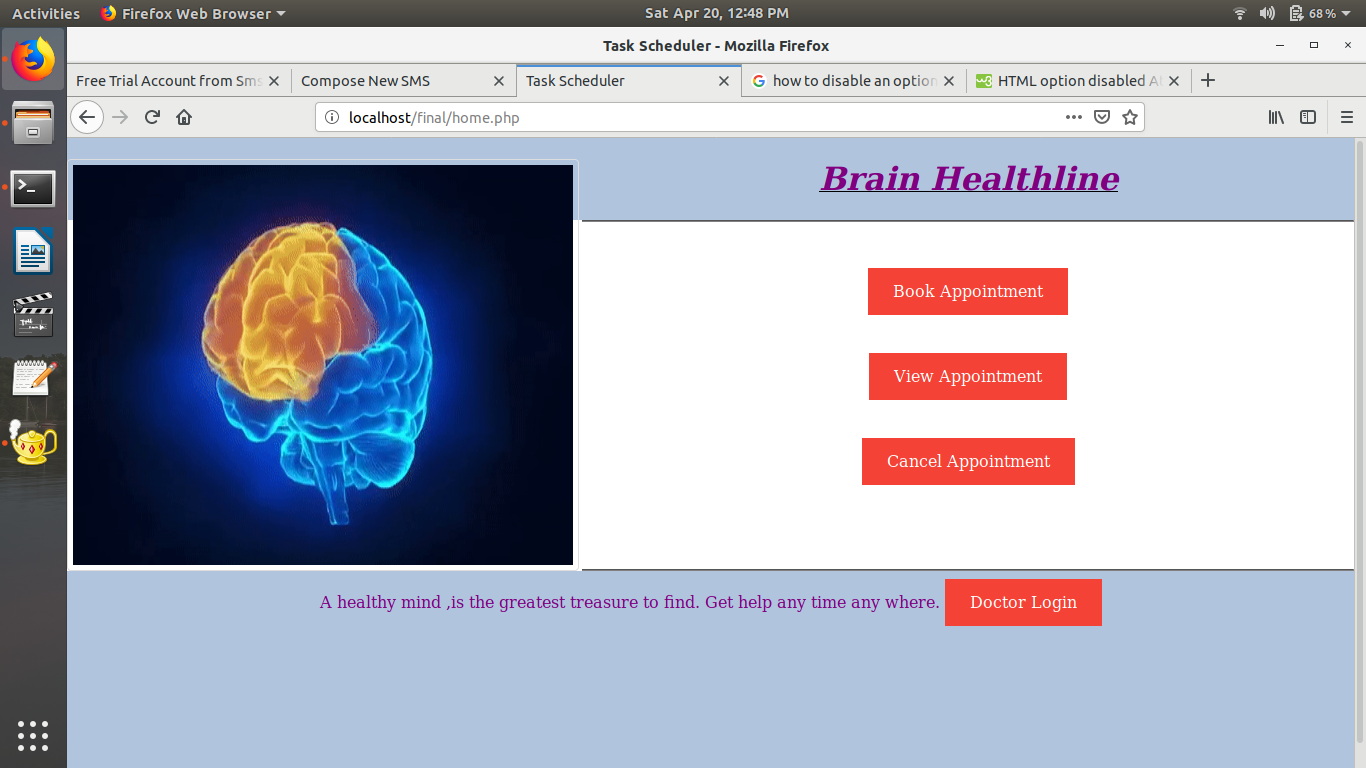
****

**SEQUENCE DIAGRAM:**

****

**GUI Design**

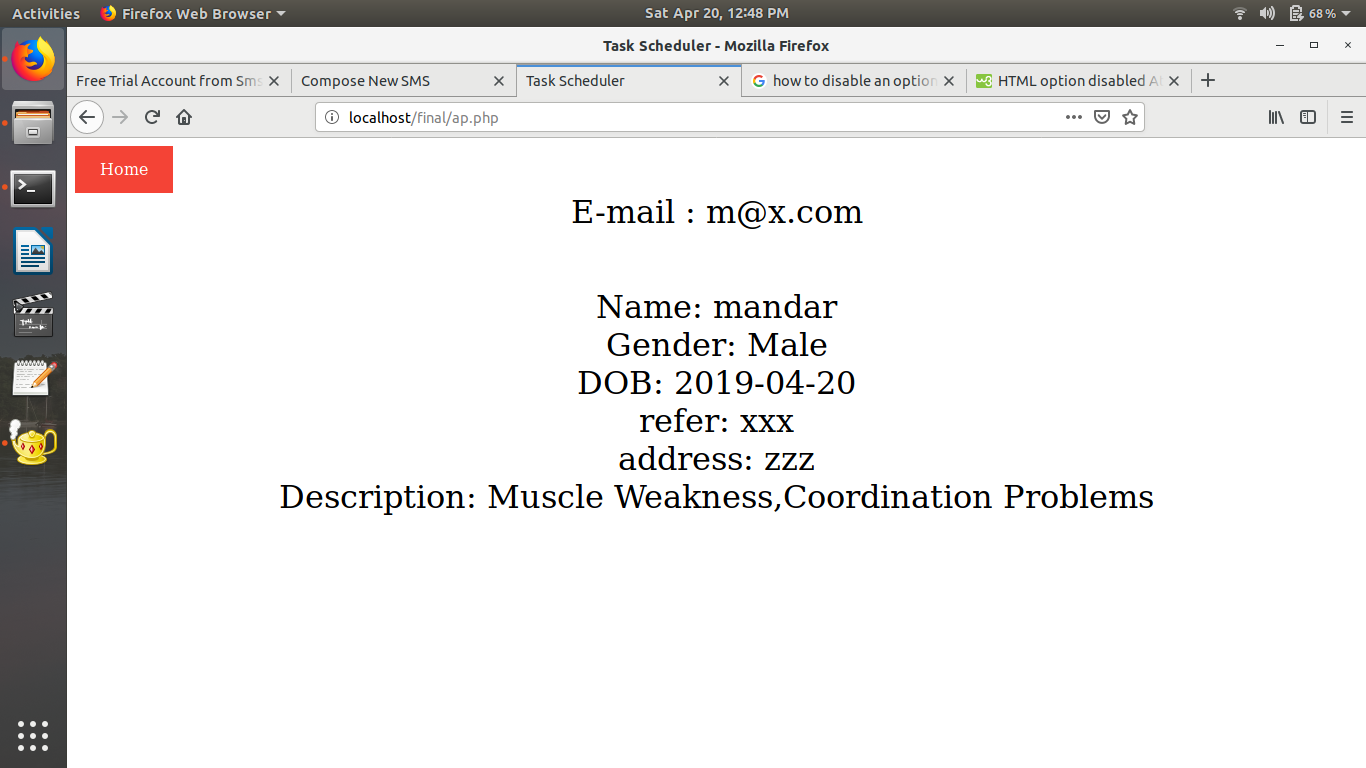
**HOMEPAGE:**

****

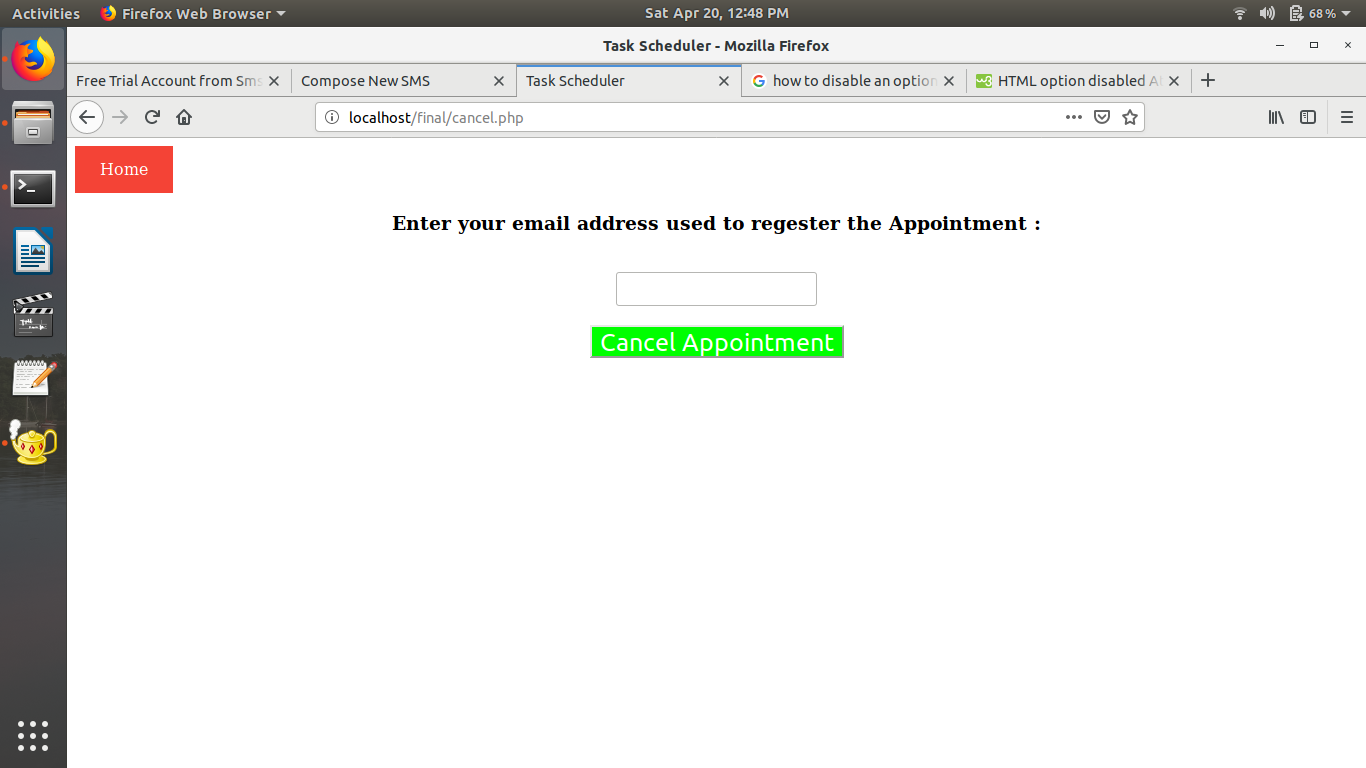
**APPOINTMENT PAGE:**

****

**VIEW PAGE:**

****

**CANCEL PAGE:**

****

**Chapter 4**

**Testing**

**Introduction to Testing:**

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test.[1] Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

Software testing can also be stated as the process of validating and verifying that a software program/application/product:

1. meets the business and technical requirements that guided its design and development;

2. Works as expected; and

3. Can be implemented with the same characteristics.

**Verification:**

Verification is the process of evaluating work-products of a development phase to determine whether they meet the specified requirements.

Verification ensures that the product is built according to the requirements and design specifications. It also answers to the question, Are we building the product right?

**Validation:**

Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfills its intended use when deployed on appropriate environment.

It answers to the question, Are we building the right product?

**4.3.1 Testing Objectives**

1. Testing is a process of executing a program with the intent of finding an error.

2. A good test case is one that has a probability of finding an as yet undiscovered error.

3. A successful test is one that uncovers an undiscovered error.

**4.3.2 Testing Principles**

1. All tests should be traceable to end user requirements

2. Tests should be planned long before testing begins

3. Testing should begin on a small scale and progress towards testing in large

4. Exhaustive testing is not possible

**4.3.3 Testing Strategies**

A strategy for software testing must accommodate low-level tests that are necessary to verify that all small source code segments has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.

* **Types of Testing**
* Unit Testing
* Integration Testing
* **Unit Testing**

Unit testing is essential for the verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules.

Using the detailed design description as a guide, important paths are tested to uncover errors within the boundary of the modules. These tests were carried out during the programming stage itself.

* **Integration Testing**

Integration testing is the logical extension of unit testing. In its simplest form, two units that have already been tested are combined into a components and the interface between them is tested. A component, in this sense, refers to an integrated aggregate of more than one unit.

**4.3.5 Test Cases**

The test case specification for system testing has to be submitted for review before system testing commences.

Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

Using Unit testing methods, the software engineer can drive test cases that;

* Guarantee that logical decisions on their true and false sides.
* Exercise all logical decisions on their true and false sides.
* Execute all loops at their boundaries and within their operational bounds.
* Exercise internal data structure to assure their validity.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Project Name | Neurologist’s Daily Appointment Scheduler Design |  |  |  |  |  |
| Module Name | Login |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Test Scenario ID | Test Scenario Description | Test Case Description | Test Steps | Expected Result | Actual Result | Status |
| TS\_1 | Verify the Login Functionality of Doctor Login Page | Enter a valid username and password | 1.Enter Valid Username 2.Enter Valid Password 3.Click on Login Button | Successful Login | Login successfully | Pass |
| TS\_2 | Verify the Login Functionality of Doctor Login Page | Enter a valid username and invalid password | 1.Enter Valid Username 2.Enter Invalid Password 3.Click on Login Button | A pop up message box to show an error"invalid username/password" | Login unsuccessfully | Fail |
| TS\_3 | Verify the Login Functionality of Doctor Login Page | Enter a invalid username and password | 1.Enter Invalid Username 2.Enter Valid Password 3.Click on Login Button | A pop up message box to show an error"invalid username/password" | Login unsuccessfully | Fail |
| TS\_4 | Verify the Login Functionality of Doctor Login Page | Enter a invalid username and invalid password | 1.Enter Invalid Username 2.Enter Invalid Password 3.Click on Login Button | A pop up message box to show an error"invalid username/password" | Login unsuccessfully | Fail |
| Project Name | Neurologist’s Daily Appointment Scheduler Design |  |  |  |  |  |
| Module Name | Home Page |  |  |  |  |  |
| TS\_5 | Verify the Home Page | Click on "Book Appointment" | Step-1 Click on "Book Appointment"Button | "Patients all Details Form" should be validate. | "Patients all Details form" is validated. | Pass |
| TS\_6 | Verify the Home Page | Click on "Book Appointment" | Step-1 Fill all Details. Step-2Click on "Submit"Button | "Patients all Details Form" should be validate. | "Patients all Details form" is not validated. | Fail |
| TS\_7 | Verify "View Appointment"page | Click on "View Appointment" | Step-1 Click on "View Appointment" Button | Appointment Details should open. | Appointment Details is Displayed. | Pass |
| TS\_8 | Verify "View Appointment"page | Click on "View Appointment" | Step-1 Click on "View Appointment" Button | Appointment Details should open. | Appointment Details is not Displayed. | Fail |
| TS\_9 | Verify "Cancel Appointment"page | Click on "Cancel Appointment" | Step-1 Click on "Cancel Appointment" Button Step-2 Enter correct Email ID which is entered at the time of Booking the Appointment. | Appointment should be cancelled and again the user should not possible to view. | Appointment is Cancelled. | Pass |
| TS\_10 | Verify "Cancel Appointment"page | Click on "Cancel Appointment" | Step-1 Click on "Cancel Appointment" Button Step-2 Enter wrong Email ID which is entered at the time of Booking the Appointment. | Appointment should be cancelled and again the user should not possible to view. | Appointment is not cancelled and display popup message i.e.Enter correct Email ID. | Fail |

**Table 4.1:Test Cases**

**Conclusion**

By all requirement gathering and the investigation of the applicability, conclusion of this SRS is the Appointment scheduler enables you to automatically perform routine Appointments of doctors on chosen computer. The Appointment scheduler does this by monitoring whatever criteria you choose to initiate the Appointments and executing the Appointments when the criteria is met

The Appointment Scheduler can be used to execute Appointments such as starting an application, sending an email message, or showing a message box. Appointments can be scheduled to execute:

* When a specific system event occurs.
* At a specific time.
* At a specific time on a daily schedule.
* At a specific time on a weekly schedule.
* At a specific time on a monthly schedule.
* At a specific time on a monthly day-of-week schedule.
* When the system enters an idle state.
* When the Appointment is registered.
* When the system is booted.
* When a user logs on.

**REFERENCES**

[1] Fatma Poni Mardiah, Mursyid Hasan Basri, “The Analysis of Appointment System to Reduce Outpatient Waiting Time at Indonesia’s Public Hospital”, Human Resource

Management Research, 3(1) (2013), 27-33

[2] Cayirli, Tugba, Emre Veral,” Outpatient Scheduling in Healthcare”, Production and Operation Management, vol 12(4)

[3] Danielle Hayday,”Online Appointment Scheduling”, 2014

[4] Wenjun Cao et.al, “A web-based appointment system to reduce waiting for outpatients: A retrospective study”, BMC Health Services Research, vol 11, 2011

[5] Lucille Keir, Barbara A. Wise, Connie Krebs, Medical Assisting: Essentials of Administrative and Clinical Competencies, Cenage Learning pp.131-133

[6] Arthur Hylton III, Suresh Sankaranarayanan, “Application of Intelligent Agents in Hospital Appointment Scheduling System”, International Journal of Computer Theory and Engineering, Vol. 4(4) (2012)

[7] <http://www.weberp4.com/industry-solutions/hospital-management/doctors-appointment-management.htm>

[8] Xiuju Zhan, Xiufeng Liu, “Design and Implementation of Clinic Appointment Registration System”, Engineering, (5) (2013), pp.527- 529

[9] Karen Davis, Stephen C. Schoenbaum, and Anne-Marie J. Audet,” A 2020 Vision of Patient-Centered Primary Care," Journal of General Internal Medicine, vol. 20, pp. 953-957, 2005.

[10] Xiaojun Zhang “Developing an Online Patient Appointment Scheduling System Based on Web Services Architecture”, APAMI Conference Proceedings, 2012

[11] Ding, J. 2008. MSc Project in A web 2.0 approach to a fashionista website. Heriot-Watt University.

[12] Jazayeri, M. 2007 Some trends in web application development. Future of Software Engineering.

[13] Lu, K. M. and Ab Hamid, S.H.2007, Conceptual Design of Web-Based Appointment Management System using Object

WebML.

[14] McCracken, H. The state of Windows 7 satisfaction. PCWorld.com. February 2010.

[15] Nations,D. 2010, What is a Web Application. Available at:

http://webtrends.about.com/od/webapplications/a/web\_application.htm. Last accessed 26 Mar 2011.

[16] O'Reilly, T. 2005, What Is Web 2.0? Available at:

http://oreilly.com/lpt/a/6228. Last Accessed 13 May 2010.

[17] Stern, Z. 2010 Set up a multiplatform network in Windows 7. PCWorld.com.

[18] Unutzer, J., Choi, Y., Cook, I. A. and Oishi, S. 2002. A Web-Based Data Management System. Psychiatric Services.

* www.microsoft.com
* www.Appointmentmanagementsoft.com
* [www.windowsreport.com](http://www.windowsreport.com).
  + [**^**](https://en.wikipedia.org/wiki/Appointment_scheduling_software#cite_ref-:1_1-0) [*"Cox Tracks Field Techs With TOA's Software"*](http://www.multichannel.com/article/133756-Cox_Tracks_Field_Techs_With_TOA_s_Software.php)*.*[*Multichannel News*](https://en.wikipedia.org/wiki/Multichannel_News)*. 2008-06-24. Retrieved 2012-02-17.*
  + [**^**](https://en.wikipedia.org/wiki/Appointment_scheduling_software#cite_ref-2) [*"ONO and TOA Technologies Join Forces to Bring the Power of Customer Appointment Management to Spain's Largest Broadband Communications Provider"*](https://web.archive.org/web/20090313092120/http:/toatech.com/news/Press_Releases.html?top=9)*(Press release).*[*TOA Technologies*](https://en.wikipedia.org/wiki/TOA_Technologies)*. 2008-05-08. Archived from*[*the original*](http://toatech.com/news/Press_Releases.html?top=9)*on 2009-03-13.*
* <https://en.wikipedia.org/wiki/Appointment_scheduling_software>

**1. Instructions about Page Numbering/Figure Numbering etc..**

First page of first chapter should not have a printed page no.

Further page nos. should occur at the bottom centre, 1cm. from the bottom.

The title of the **table** should be at the **top** …

Table 2.1 Timing Analysis

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

The title of **figure/photograph/graph** should be at the **bottom.**

The titles should start at top/bottom with no additional line spacing.

1. **About Size of the Report –**

Normally the Project Report would be approximately 40 pages. It may be in the range of 30 to 50 pages (including appendix, data sheets etc.). This may change in exceptional cases.

1. **No. of Copies to be prepared –**

1 copy for Department per Project group (Spiral Binding)