**INTERNSHIP PROJECT REPORT**

**(Term Jan-May, 2019)**

**REMEDY ACKNOWLEDGEMENT**

**Submitted By**

**Deepak Kumar**

**Registration No:11505320**

**Course Code: CSE 441**

Under the guidance of

**Shilpa Mahajani**

School of computer science and engineering

****

**DECLARATION**

I hereby declare that the project work entitled (“Remedy Acknowledgement”) is an authentic record of my work carried out as requirements of Internship (INDUSTRY CO-OP PROJECT-I) for the award of B. Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Manoj Kulkarni, during January to May 2019. All the information furnished in this Internship project report is based on my own intensive work and is genuine.

Name of Student: Deepak Kumar

Registration Number:11505320

Signature:

Date:

**CONTINUOUS ASSESSMENT (CA) FOR INTERNSHIP**

(By external internship in-charge from organization)

Name of the Student: **\_\_\_\_\_\_\_\_\_\_\_\_** Registration Number **\_\_\_\_\_\_\_\_\_\_\_\_**

Internship Project Title (if any): **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Name of Organization & Address: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Name of External Internship in-charge (with mobile number): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Criteria | Marks Obtained | Maximum Marks |
| 1 | Student conduct during internship |  | 10 |
| 2 | Punctuality and Enthusiasm |  | 20 |
| 3 | Technical Skill & Knowledge |  | 20 |
| 4 | Internship Project Marks |  | 50 |
|  | TOTAL |  | 100 |

Date **\_\_\_\_\_\_\_\_\_\_\_** Authorized Signatory \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of External Internship in-charge: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Designation:

Company Seal

Note: The external internship in-charge will mark the continuous assessment at the end of semester / internship. Students must ensure that evaluation marks are provided by the organization as per above parameters in the given format during ETP.

**CERTIFICATE**

This is to certify that the declaration statement made by student is correct to the best of my knowledge and belief. He has completed this Internship Project under my guidance and supervision. The present work is the result of his original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Internship Project is fit for the submission and partial fulfillment of the conditions for the award of B. Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara.

**Signature and Name of the Mentor**

**Designation**

**School of Computer Science and Engineering,**

Lovely Professional University,

Phagwara, Punjab

Date:

**ACKNOWLEDGEMENT**

I take this opportunity to present our votes of thanks to all those guideposts who really acted as lightening pillars to enlighten my way throughout this project that has led to successful and satisfactory completion of this study.

I am really grateful to Mr. Manoj Kulkarni for providing me with an opportunity to undertake this Internship project and providing us with all the facilities. I am highly thankful to sir for his active support, valuable time and advice, whole-hearted guidance, sincere cooperation and pains-taking involvement during the study and in completing the project within the time stipulated.

Lastly, I am thankful to all those, particularly the various colleagues, who have been instrumental in creating proper, healthy and conductive environment and including new and fresh innovative ideas for us during the project, without their help, it would have been extremely difficult for us to prepare the project in a time bound framework**.**

**TABLE OF CONTENTS**

[1.INTRODUCTION 8](#_Toc6321614)

[1.1Technologies Description 8](#_Toc6321616)

[1.1.1 Java: - 8](#_Toc6321617)

[1.1.2JSP: - 10](#_Toc6321618)

[vs. Active Server Pages (ASP) 10](#_Toc6321619)

[vs. Pure Servlets 10](#_Toc6321620)

[vs. Server-Side Includes (SSI) 10](#_Toc6321621)

[vs. JavaScript 11](#_Toc6321622)

[vs. Static HTML 11](#_Toc6321623)

[1.1.3Eclipse: - 11](#_Toc6321624)

[1.1.4RDBMS: - 11](#_Toc6321625)

[1.1.5MYSQL DATABASE: - 13](#_Toc6321626)

[1.1.6APACHE TOMCAT: - 13](#_Toc6321627)

[2.PROFILE OF PROBLEM 14](#_Toc6321628)

[3.Existing System. 14](#_Toc6321629)

[3.1Introduction 14](#_Toc6321630)

[3.2DFD for Present System 15](#_Toc6321631)

[Context Level DFD (Level 0) 15](#_Toc6321632)

[ADMIN 15](#_Toc6321633)

[Patient 16](#_Toc6321634)

[4.PROBLEM ANALYSIS 16](#_Toc6321635)

[4.1 PRODUCT DEFINITION 16](#_Toc6321636)

[Back End: MySQL 16](#_Toc6321637)

[4.2 FEASIBILITY ANALYSIS 16](#_Toc6321638)

[4.2.1TECHNICAL FEASIBILITY 17](#_Toc6321639)

[Servlet Advantage 17](#_Toc6321640)

[5.Software Requirement Analysis 17](#_Toc6321641)

[5.1Introduction 17](#_Toc6321642)

[5.1.1 Purpose 17](#_Toc6321643)

[5.1.2 Scope 18](#_Toc6321644)

[5.2General Description 18](#_Toc6321645)

[5.2.1 Product Description. 18](#_Toc6321646)

[5.2.2 USERS Characteristics 19](#_Toc6321647)

[5.3 Functional Requirements. 19](#_Toc6321648)

[5.4 Software and hardware Requirement 22](#_Toc6321649)

[5.5 Technologies Recommended 22](#_Toc6321650)

[6.Design 23](#_Toc6321651)

[6.1 Use Case Diagram 23](#_Toc6321652)

[6.2 Flow Chart Diagram 24](#_Toc6321653)

[6.3 Data Dictionary 24](#_Toc6321654)

[StatusDetail Table 24](#_Toc6321655)

[Acknowledgement Table 25](#_Toc6321656)

[7.Testing 25](#_Toc6321657)

[7.1 Testing Strategies 26](#_Toc6321658)

[7.1.1 Unit Testing 26](#_Toc6321659)

[7.1.2 Integration Testing 26](#_Toc6321660)

[7.1.2.1 Top Down Integration testing 26](#_Toc6321661)

[7.1.2.2 Bottom-up Integration testing 26](#_Toc6321662)

[7.1.3 System Testing 26](#_Toc6321663)

[7.1.4 Acceptance Testing 27](#_Toc6321664)

[7.1.4.1 Alpha Testing 27](#_Toc6321665)

[7.1.4.2 Beta Testing 27](#_Toc6321666)

[7.2 Testing Methods 27](#_Toc6321667)

[7.2.1 White Box Testing 27](#_Toc6321668)

[7.2.2 Black Box Testing 27](#_Toc6321669)

[7.3 Validation 28](#_Toc6321670)

[8.4 Limitations 28](#_Toc6321671)

[8.5 Test Results 28](#_Toc6321672)

[TEST CASES 29](#_Toc6321673)

[TEST CASES 1: Admin Login 29](#_Toc6321674)

[TEST CASES 2: Book Appointment Page 30](#_Toc6321675)

[TEST CASES 3: Mailbox Access 31](#_Toc6321676)

[TEST CASES 4: Check status 31](#_Toc6321677)

[8. IMPLEMENTATION 31](#_Toc6321678)

[8.1PROCESS MODEL USED BY US 32](#_Toc6321679)

[8.1.1Spiral model 32](#_Toc6321680)

[8.1.2Agile Methodology 32](#_Toc6321681)

[Scrum 33](#_Toc6321682)

[8.2CONVERSION PLAN 35](#_Toc6321683)

[8.3 POST IMPLEMENTATION OF PROJECT AND MAINTAINANCE 36](#_Toc6321684)

[9. PROJECT LEGACY 36](#_Toc6321685)

[9.1 Current Status of the project 36](#_Toc6321686)

[9.2 Technical and Managerial lessons learnt 37](#_Toc6321687)

[9.2.1 Technical lessons learnt 37](#_Toc6321688)

[9.2.2 Managerial lessons learnt 37](#_Toc6321689)

[10.Snapshots 38](#_Toc6321690)

[11.Bibliography 43](#_Toc6321691)

# 1.INTRODUCTION

Remedy Acknowledgement is basically like a customer care service. In our remedy acknowledgement we have taken into consideration doctor-patient interaction. A patient who is suffering from a particular ailment and is willing to consult doctor on a particular day and particular time wants to get his appointment fixed with the doctor.

In our project, in between the doctor and the patient the receptionist also plays a very important role as he is the one who acts as an intermediator in our project.

# The receptionist will book the appointment accordingly so that the best doctor is provided according to the ailment.

## 1.1Technologies Description

Technologies and tools that are used to build this project are: -

### 1.1.1 Java: -

Java is a high-level programming language originally developed by Sun Microsystems and released in 1995. Java runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX. Java is guaranteed to be **Write Once, Run Anywhere.**

Java is −

* **Object Oriented** − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
* **Platform Independent** − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.
* **Simple** − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
* **Secure** − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
* **Architecture-neutral** − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
* **Portable** − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
* **Robust** − Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.
* **Multithreaded** − With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.
* **Interpreted** − Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.
* **High Performance** − With the use of Just-In-Time compilers, Java enables high performance.
* **Distributed** − Java is designed for the distributed environment of the internet.
* **Dynamic** − Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

### 1.1.2JSP: -

Java Server Pages (JSP) is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications. JSP have access to the entire family of Java APIs, including the JDBC API to access enterprise databases.

Java Server Pages (JSP) is a technology for developing Webpages that supports dynamic content. This helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.

A Java Server Pages component is a type of Java servlet that is designed to fulfill the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands.

Using JSP, you can collect input from users through Webpage forms, present records from a database or another source, and create Webpages dynamically.

JSP tags can be used for a variety of purposes, such as retrieving information from a database or registering user preferences, accessing JavaBeans components, passing control between pages, and sharing information between requests, pages etc.

Advantages of JSP

Following table lists out the other advantages of using JSP over other technologies −

### vs. Active Server Pages (ASP)

The advantages of JSP are twofold. First, the dynamic part is written in Java, not Visual Basic or other MS specific language, so it is more powerful and easier to use. Second, it is portable to other operating systems and non-Microsoft Web servers.

### vs. Pure Servlets

It is more convenient to write (and to modify!) regular HTML than to have plenty of println statements that generate the HTML.

### vs. Server-Side Includes (SSI)

SSI is really only intended for simple inclusions, not for "real" programs that use form data, make database connections, and the like.

### vs. JavaScript

JavaScript can generate HTML dynamically on the client but can hardly interact with the web server to perform complex tasks like database access and image processing etc.

### vs. Static HTML

Regular HTML, of course, cannot contain dynamic information.

### 1.1.3Eclipse: -

Eclipse is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) used in [computer programming](https://en.wikipedia.org/wiki/Computer_programming), and is the most widely used Java IDE. It contains a base [workspace](https://en.wikipedia.org/wiki/Workspace) and an extensible [plug-in](https://en.wikipedia.org/wiki/Plug-in_(computing)) system for customizing the environment. Eclipse is written mostly in [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) and its primary use is for developing Java applications

### 1.1.4RDBMS: -

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems.

So nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

**Relational Database Management System (RDBMS)** is a software that:

* Enables you to implement a database with tables, columns and indexes.
* Guarantees the Referential Integrity between rows of various tables.
* Updates the indexes automatically.
* Interprets an SQL query and combines information from various tables.

**RDBMS TERMINOLOGY: -**

Before I proceed to explain MySQL database system, let's revise few definitions related to database.

* **Database:** A database is a collection of tables, with related data.
* **Table:** A table is a matrix with data. A table in a database looks like a simple spreadsheet.
* **Column:** One column (data element) contains data of one and the same kind, for example the column postcode.
* **Row:** A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
* **Redundancy:** Storing data twice, redundantly to make the system faster.
* **Primary Key:** A primary key is unique. A key value cannot occur twice in one table. With a key, you can find at most one row.
* **Foreign Key:** A foreign key is the linking pin between two tables.
* **Compound Key:** A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
* **Index:** An index in a database resembles an index at the back of a book.
* **Referential Integrity:** Referential Integrity makes sure that a foreign key value always points to an existing row.

### 1.1.5MYSQL DATABASE: -

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

### 1.1.6APACHE TOMCAT: -

**Apache Tomcat**, often referred to as **Tomcat Server**, is an open-source [Java Servlet Container](https://en.wikipedia.org/wiki/Servlet_container) developed by the [Apache Software Foundation](https://en.wikipedia.org/wiki/Apache_Software_Foundation) (ASF). Tomcat implements several [Java EE](https://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition) specifications including Java Servlet, [Java Server Pages](https://en.wikipedia.org/wiki/JavaServer_Pages) (JSP), and [Web Socket](https://en.wikipedia.org/wiki/WebSocket), and provides a "pure [Java](https://en.wikipedia.org/wiki/Java_(programming_language))" [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) [web server](https://en.wikipedia.org/wiki/Web_server) environment in which [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) code can run.

Tomcat is developed and maintained by an open community of developers under the auspices of the [Apache Software Foundation](https://en.wikipedia.org/wiki/Apache_Software_Foundation), released under the [Apache License](https://en.wikipedia.org/wiki/Apache_License) 2.0 license, and is [open-source software](https://en.wikipedia.org/wiki/Open-source_software).

## 2.PROFILE OF PROBLEM

Remedy Acknowledgement provides smooth working of the system as we have an intermediator who will be there continuously to help the user to connect with the remedy. In our project this intermediator is the receptionist who will help the system to work effectively without any hassle.

There will be no confusion regarding the fact that who is the one who will be visiting the doctor first. It is first come first served basis that is the one who will book the appointment first will be given the priority and appointment will be given on that basis.

Moreover, the receptionist will also be able to check who all are the patients whose appointment is still pending and the doctor needs to be assigned to them based on their ailment.

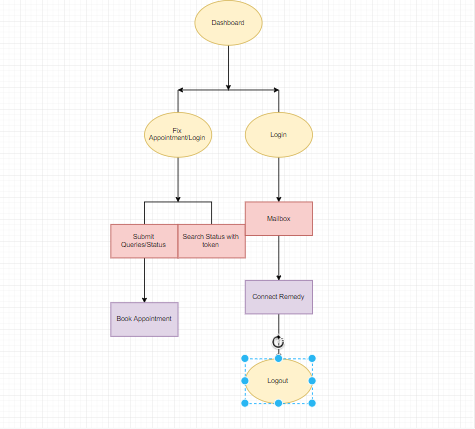
So basically, to put it in nutshell Remedy Acknowledgement, is basically a project that aims in providing services based on the kind of problem they are facing.

# 3.Existing System.

## 3.1Introduction

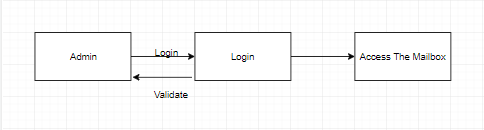
Remedy Acknowledgement aims at providing services to the people according to the resources that are suitable for allotment. In our case we took the example of doctor -patient case. We have a dashboard that contains a book appointment page and admin page. The book appointment page is responsible for submitting the queries in a form and the admin page is responsible for authenticating the id password and checking the mailbox and finally booking the appointment accordingly.

## 3.2DFD for Present System

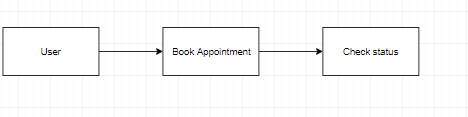


## Context Level DFD (Level 0)

### ADMIN



### Patient



# 4**.PROBLEM ANALYSIS**

## 4.1 PRODUCT DEFINITION

**Product Name**: Remedy Acknowledgement System

**Product Objective:** Aims at providing services as an intermediator and booking the appointment on the basis of the best available doctor available.

**Language Used:**

**Front End**: JAVA

## Back End: MySQL

## 4.2 FEASIBILITY ANALYSIS

The feasibility for the project is done on the basis of the parameters like time required,

workers needed, hardware and software requirements, cost involved in the building of

this application. We check whether developing the project is beneficial.

Resources that are required to build up this project and make the testing system live

on the internet are:

Time – There are about 1-1.5 months to build up this project, starting from mid- March

2019 to end of April 2019.

Workers – The team must consist of the members skilled in coding of Java and must

be familiar with the concepts of Software Development.

This project was found to be completely feasible after the analysis done on the basis

of the above mention parameters.

### 4.2.1TECHNICAL FEASIBILITY

### **Servlet Advantage**

1. Servlets provide a way to generate dynamic documents that is both easier to write and faster to run.
2. provide all the powerful features of JAVA, such as Exception handling and garbage collection.
3. Servlet enables easy portability across Web Servers.
4. Servlet can communicate with different servlet and servers.
5. Since all web applications are stateless protocol, servlet uses its own API to maintain session.

MySQL has been chosen over the other possibilities with the following reasoning:

* MySQL is license free for registered charities.
* MySQL can handle 50,000,000 + records.
* MySQL is an ideal database type to create a prototype of an e-commerce web site.
* The MySQL database can be easily upgraded to MS SQL at later date.

# 5.Software Requirement Analysis

## 5.1Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS.

### 5.1.1 Purpose

The purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

### 5.1.2 Scope

This SRS is aimed at specifying requirements of software to be developed but it can also be applied to get opinions of people and their views about it. The standard can be used to create software requirements specifications directly or can be used as a model for defining an organization or project specific standard. It does not identify any specific method, nomenclature

or tool for preparing an SRS.

## 5.2General Description

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product. General description of the project is discussed in section 2 of this document. Section 3 gives the functional requirements, data requirements and constraints and assumptions made. It also gives the user viewpoint of product. Section 3 also gives the specific requirements of the product. Section 3 also discusses the external interface requirements and gives detailed description of functional requirements. Section 4 is for supporting information.

### 5.2.1 Product Description.

This document contains the problem statement that the current system is facing which is not happening. It further contains the method and tools to efficiently stored the data and utilizes it in a convenient way. It also illustrates the way the data can be utilized to give people the opinions about a particular topic. It further lists and briefly describes the major features and a brief description of each of the proposed system.

The following SRS contains the detail product perspective from different users. It provides the detail product functions of sentimental analysis with user characteristics permitted constraints, assumptions and dependencies and requirements subsets.

### 5.2.2 USERS Characteristics

**Admin: -**

1.Admin can authenticate his id and password and check his mailbox.

2.Admin can update or modify the status according to the availability of the doctor

**Users: -**

1.User here is the patient. He can book appointment by filling the form.

2. He can check the status of the appointment by using search status by token.

## 5.3 Functional Requirements.

The functional requirements are charted for each of the high level requirements called out in the earlier section:

Additionally, the following elements are captured for each business requirement in the table provided below: -

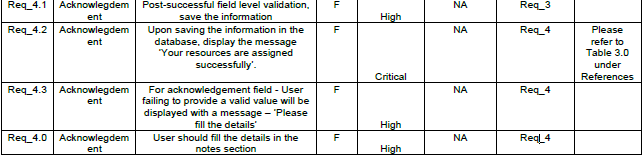
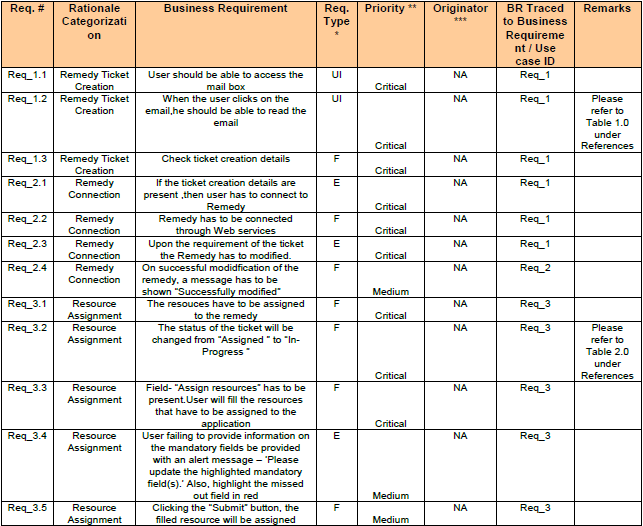
\* Req. Type = (F Core Functionality, E Exception, UI User Interface, R Reporting)

\*\* Priority of Requirement = (1=Base Functionality, 2=Advanced Functionality,

3=Additional Opportunities)

\*\* Originator = (Name of the business process of the system/ department or function

|  |  |  |
| --- | --- | --- |
| **Value** | **Rating** | **Description** |
| **1** | Critical | This requirement is critical to the success of the project. The project will not be possible without this requirement. |
| **2** | High | This requirement is high priority, but the project can be implemented at a bare minimum without this requirement. |
| **3** | Medium | This requirement is somewhat important, as it provides some value but the project can proceed without it. |
| **4** | Low | This is a low priority requirement, or a “nice to have” feature, if time and cost allow it. |

****

## 5.4 Software and hardware Requirement

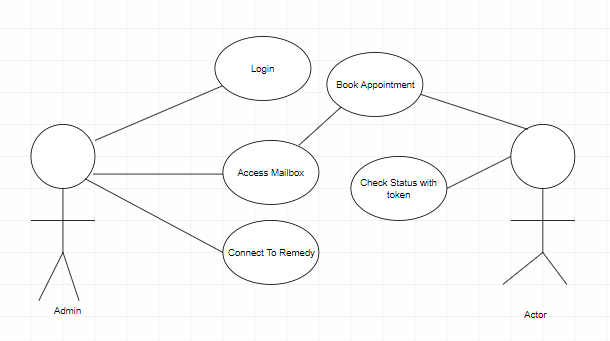
|  |  |  |
| --- | --- | --- |
| Technology | Hardware | Software |
| Java | Desktop PC with 8GB RAM | 1. Eclipse IDE for Java EE Developers (Oxygen) 2. Maven 3.6.0 3. Tomcat 9 4. MySQL Community Server 8.0 5. MySQL Workbench 8.0.14 |

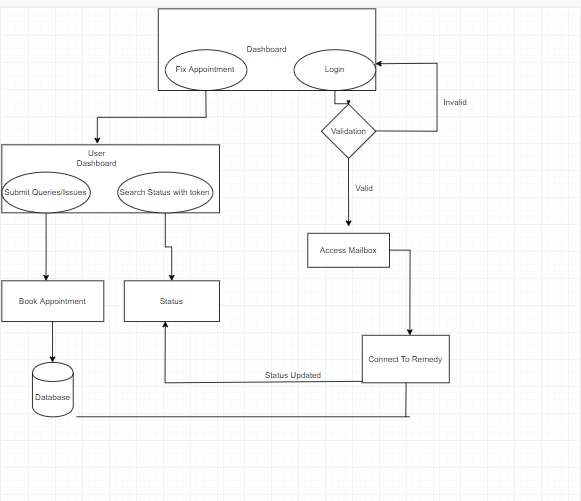
## 5.5 Technologies Recommended

|  |  |
| --- | --- |
| Front End | Java (HTML5, CSS3, JavaScript, Angular) |
| Middleware | Java (Spring, Spring MVC, Hibernate MVC, WEB API2) |
| Backend | Oracle/SQL Server |

# 6.Design

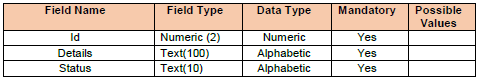
## 6.1 Use Case Diagram

****

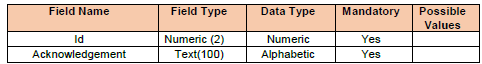
6.2 Flow Chart Diagram ****

## 6.3 Data Dictionary

### StatusDetail Table

****

### Acknowledgement Table

****

# 7.Testing

Testing is the process of evaluating a system or its components with the intent to find that whether it satisfies the specified requirements or not. This activity results in the actual, expected and difference between their results i.e. testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements.



## 7.1 Testing Strategies

In order to make sure that system does not have any errors, the different levels of testing strategies that are applied at different phases of software development are

Figure 11: Phases of Software Development 29

### 7.1.1 Unit Testing

The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

### 7.1.2 Integration Testing

The testing of combined parts of an application to determine if they function correctly together is Integration testing. This testing can be done by using two different methods

### 7.1.2.1 Top Down Integration testing

In Top-Down integration testing, the highest-level modules are tested first and then progressively lower-level modules are tested.

### 7.1.2.2 Bottom-up Integration testing

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules. In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing.

### 7.1.3 System Testing

This is the next level in the testing and tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets Quality Standards. 30

### 7.1.4 Acceptance Testing

The main purpose of this Testing is to find whether application meets the intended specifications and satisfies the client’s requirements. We will follow two different methods in this testing.

### 7.1.4.1 Alpha Testing

This test is the first stage of testing and will be performed amongst the teams. Unit testing, integration testing and system testing when combined are known as alpha testing. During this phase, the following will be tested in the application:

* Broken Links.
* The Application will be tested on machines with the lowest specification to test loading times and any latency problems.

### 7.1.4.2 Beta Testing

In beta testing, a sample of the intended audience tests the application and send their feedback to the project team. Getting the feedback, the project team can fix the problems before releasing the software to the actual users. 31

## 7.2 Testing Methods

### 7.2.1 White Box Testing

White box testing is the detailed investigation of internal logic and structure of the Code. To perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code. The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

### 7.2.2 Black Box Testing

The technique of testing without having any knowledge of the interior workings of the application is Black Box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, when performing a black box test, a tester will interact with the system’s user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

## 7.3 Validation

All the levels in the testing (unit integration, system) and methods (black box, white box) are implemented on our application successfully and the results obtained as expected.

### 8.4 Limitations

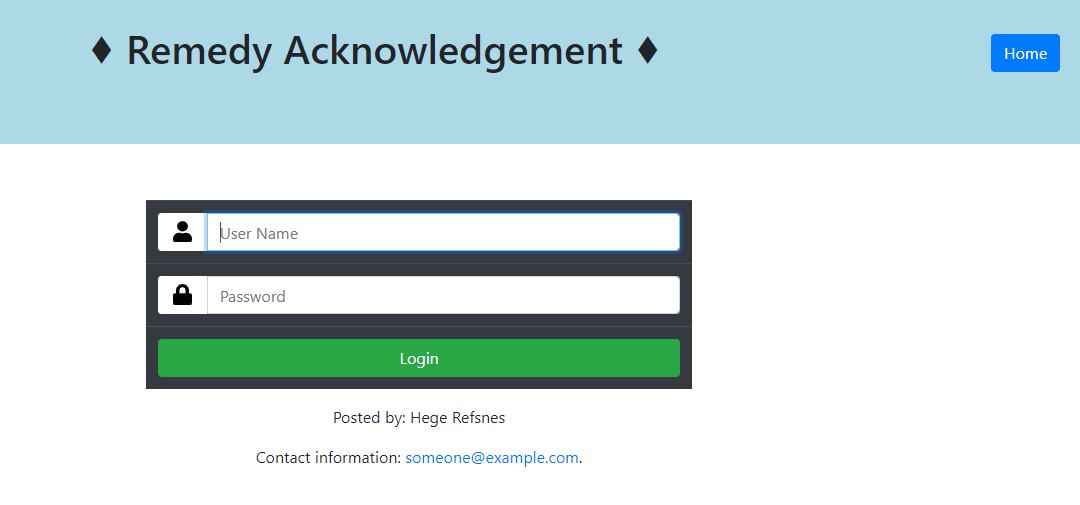
The execution time for support vector machine is more so that the user may not receive the result fast.

## 8.5 Test Results

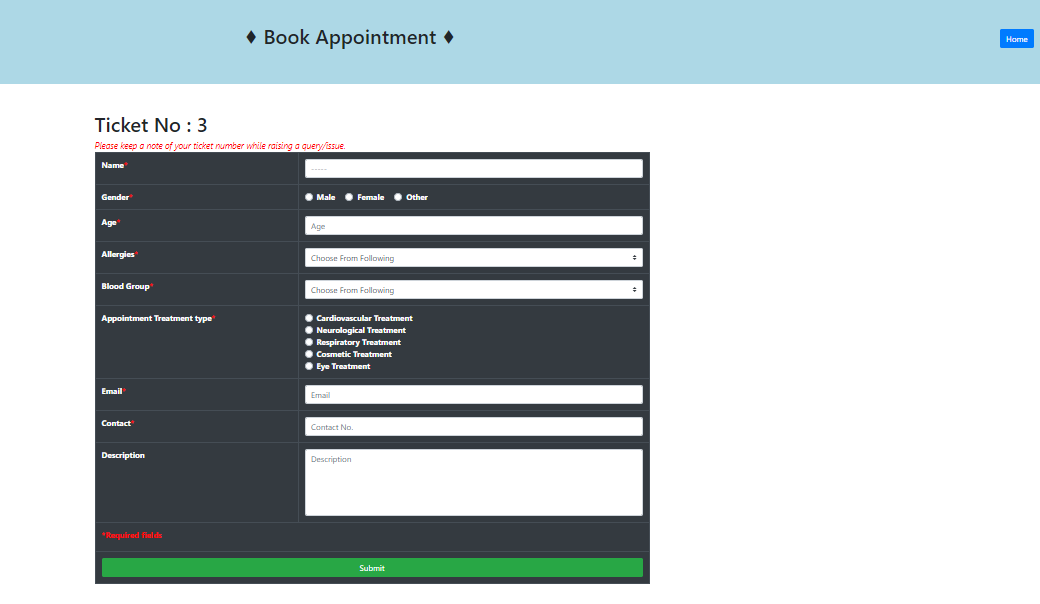
The testing is done among the team members and by the end users. It satisfies the specified requirements and finally we obtained the results as expected.

## TEST CASES

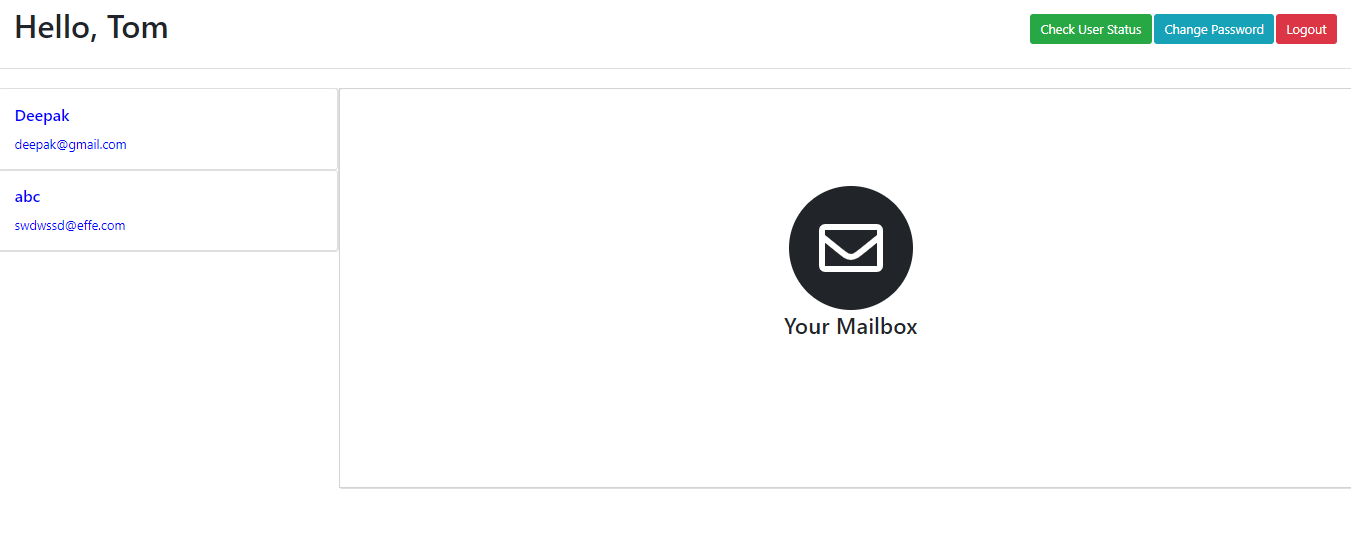
### **TEST CASES 1**: Admin Login

****

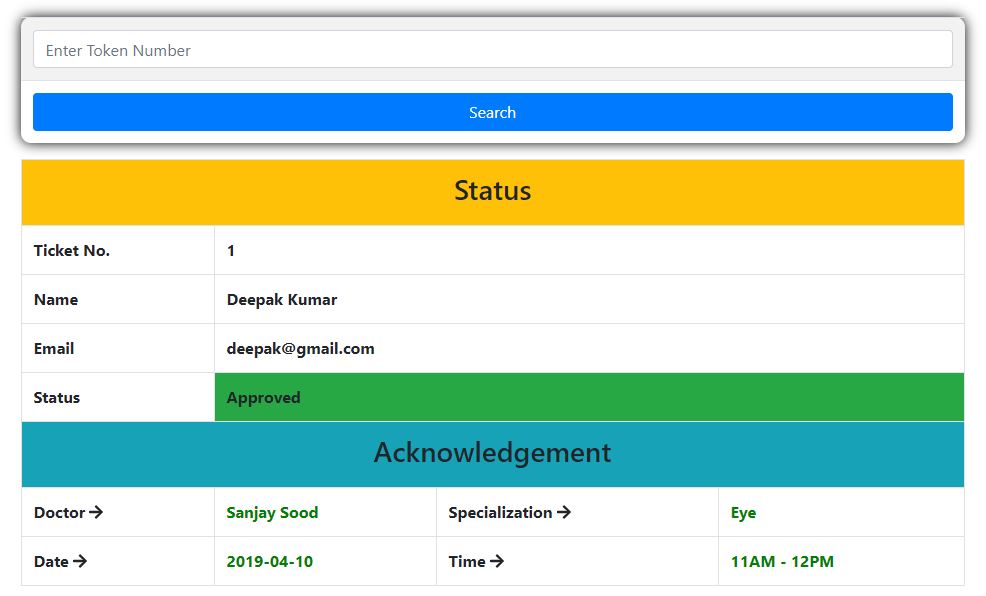
### **TEST CASES 2**: Book Appointment Page

****

### **TEST CASES 3**: Mailbox Access

****

### **TEST CASES 4**: Check status

****

# 8. IMPLEMENTATION

## 8.1PROCESS MODEL USED BY US

## 8.1.1Spiral model

The Spiral model originally proposed by Boehm, is an evolutionary software process

model that couples the iterative nature of prototyping with the controlled & systematic

aspects of the linear sequential model. It provides the potential for rapid development

of incremental version of the software.

Using the spiral model software is developed in a series of incremental releases. A

spiral model is divided into a number of framework activities also called task regions.

A spiral model contains six task regions:

**1. Customer Communication:** Tasks required to establish effective communication

between developer & customer.

**2. Planning:** Tasks required to define resources, timeline & other project related

information.

**3. Risk analysis:** Task required to access.

**4. Engineering:** Tasks required to build one or more representation of the application.

**5. Construction & release:** Task required to construct, test, install & provide user

support (e.g., documentation & Training)

**6. Customer evaluation:** Tasks required to obtain customer feedback based on

evolution of the software representation created during the engineering stage &

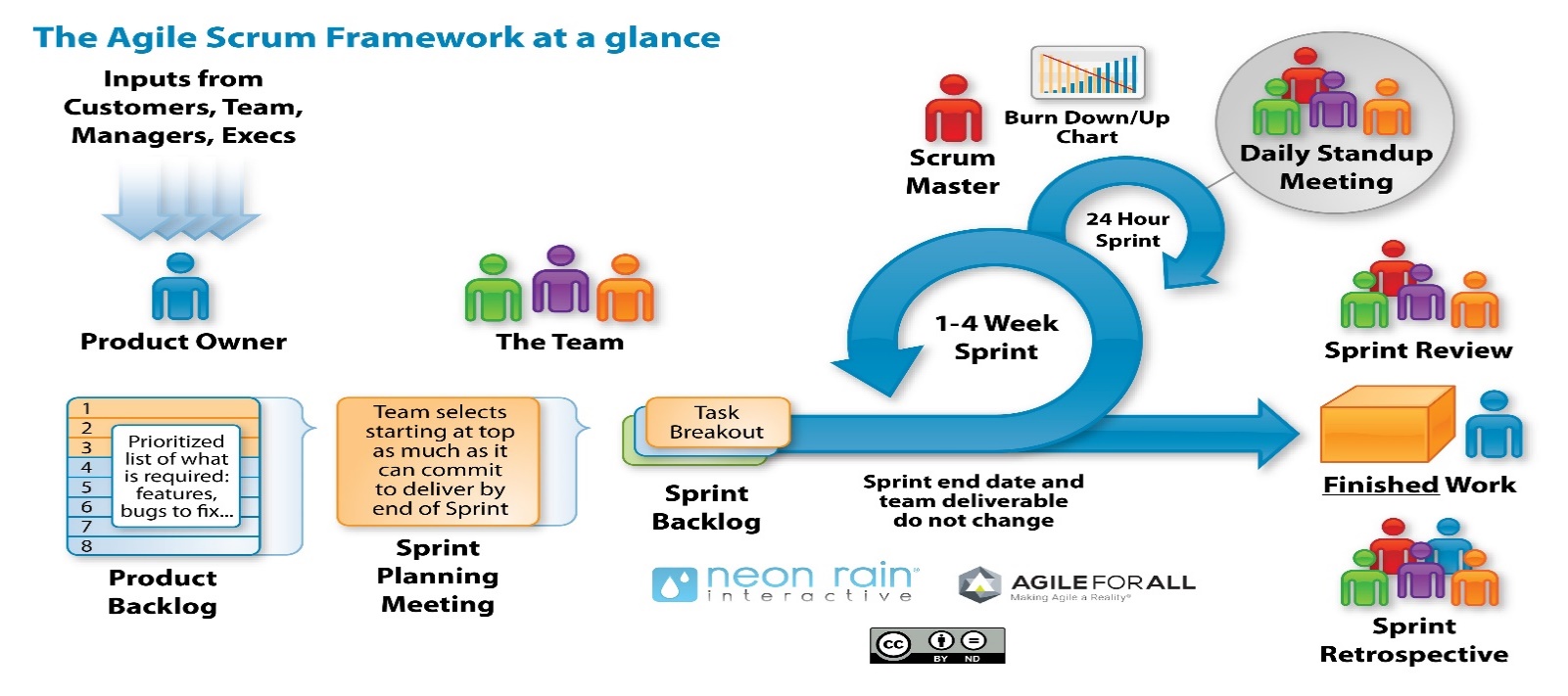
implemented during the installation stage.

## 8.1.2Agile Methodology

Agile is a software development approach where a self-sufficient and cross-functional team works on making continuous deliveries through iterations and evolves throughout the process by gathering feedback from the end users.

AGILE methodology is a practice that promotes **continuous iteration** of development and testing throughout the software development lifecycle of the project.

The agile software development emphasizes on four core values.

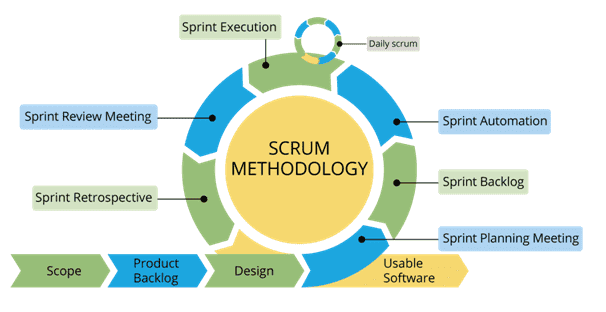
1. Individual and team interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

In our project we are using Scrum Agile Methodology

### Scrum

SCRUM is an agile development method which concentrates specifically on how to manage tasks within a team-based development environment. Basically, Scrum is derived from activity that occurs during a rugby match. Scrum believes in empowering the development team and advocates working in small teams (say- 7 to 9 members). It consists of four roles, and their responsibilities are explained as follows:

* Product owner
  + The Product owner provides specification and requirement of the project and also provide the time stamp in which the project must be completed. The review of project is taken care by product owner end when the project is ready to serve and brought out for demo.
* Proxy product owner
  + The proxy product owner understands the project requirement from the product owner and assign them to a particular development team and scrum master associated with it. The frequent reviews of project are taken care by the proxy product owner. In our project proxy product owner are cognizant Business Analysts (BA).
* Scrum Master
  + Master is responsible for setting up the team, sprint meeting and removes obstacles to progress and take the status of the task assigned to each individual of development team on daily basis.
* Development Team
  + Team manages its own work and organizes the work to complete the sprint or cycle. Development team consists of UI/UX Developer, service layer developer, database management team, tester, architecture.



## 8.2CONVERSION PLAN

To make this project live, i.e., to build the project followed:

1. Firstly, senior architect and business analyst analyzes the project requirement in terms of hardware and software needed in the project.
2. Senior architecture asks infrastructure team to provide resources for client specifications.
3. Infrastructure team provide the software and hardware requirements for the specification asked by senior architecture. For our project the hardware and software requirements provided by infrastructure team is as follow:
4. Installation of Node and Visual Studio Code.
5. Installation of angular cli, Express and PostgreSQL.
6. Once the angular cli set up successfully then setup PostgreSQL.
7. Then access the software via remote server.

5.The application is now live and open for target organization.

## 8.3 POST IMPLEMENTATION OF PROJECT AND MAINTAINANCE

The Post Implementation Review (PIR) is conducted after a project has been

completed. The purpose of the PIR is to evaluate how successfully the project

objectives have been met and how effective the project management practices were in

keeping the project on track.

In our project the all objectives met to the requirements and it is more affective as

user wants. According to the user requirements the project functionality and

objectives are made according to his.

It is generally found that systems that are easy to use, require less manpower, saves

the data entry and well received by people. But still the following points have to

consider.

1. How have systems changed the way in which operations were performed?

2. How have systems changed the timeliness of information and reports user

Received?

# 9. PROJECT LEGACY

## 9.1 Current Status of the project

All the modules of project are in complete working state. The patient can book the appointment of filling the book appointment form and he is also capable of checking his status of his appointment with the help of search status by token.

Admin can authenticate his id and password and he can access the mailbox and by seeing the ailment of the patient he can book the appointment with the appropriate doctor and he is also capable of checking the status as to who all the doctor is assigned and to whom it is not assigned.

## 9.2 Technical and Managerial lessons learnt

### 9.2.1 Technical lessons learnt

1. Servlet for making the application
2. MySQL for creating the database and managing it.
3. Bootstrap, CSS, Java Script for making the UI.

### 9.2.2 Managerial lessons learnt

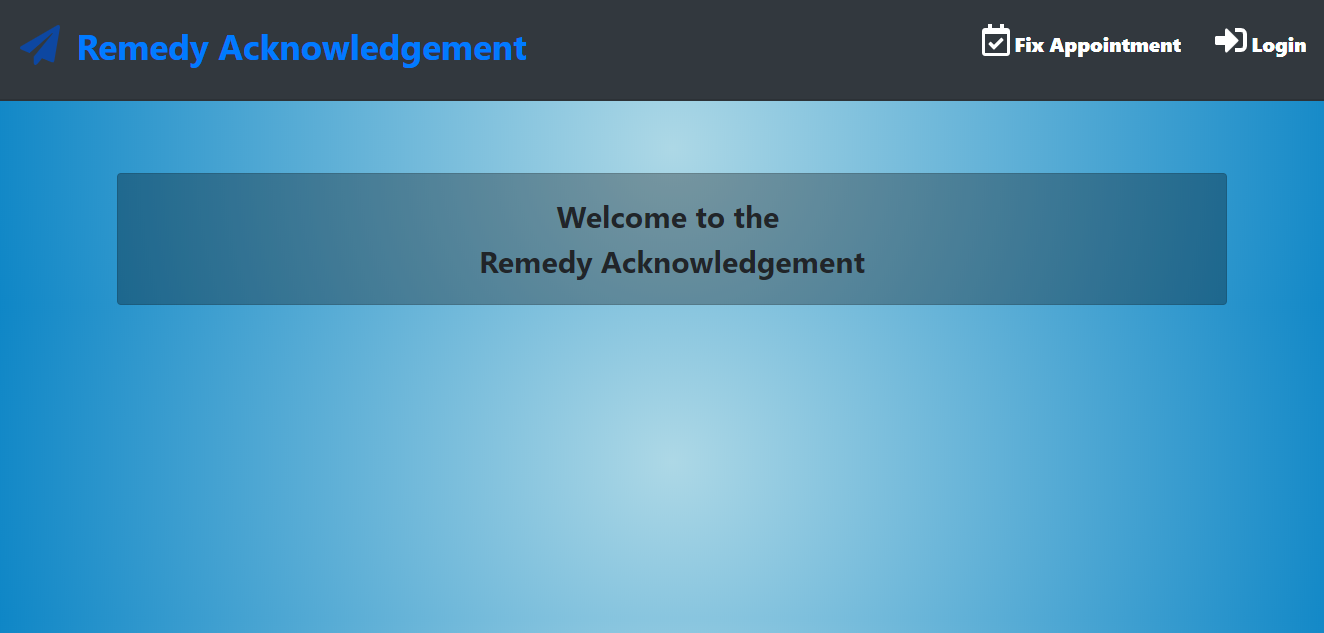
i. Working in a team

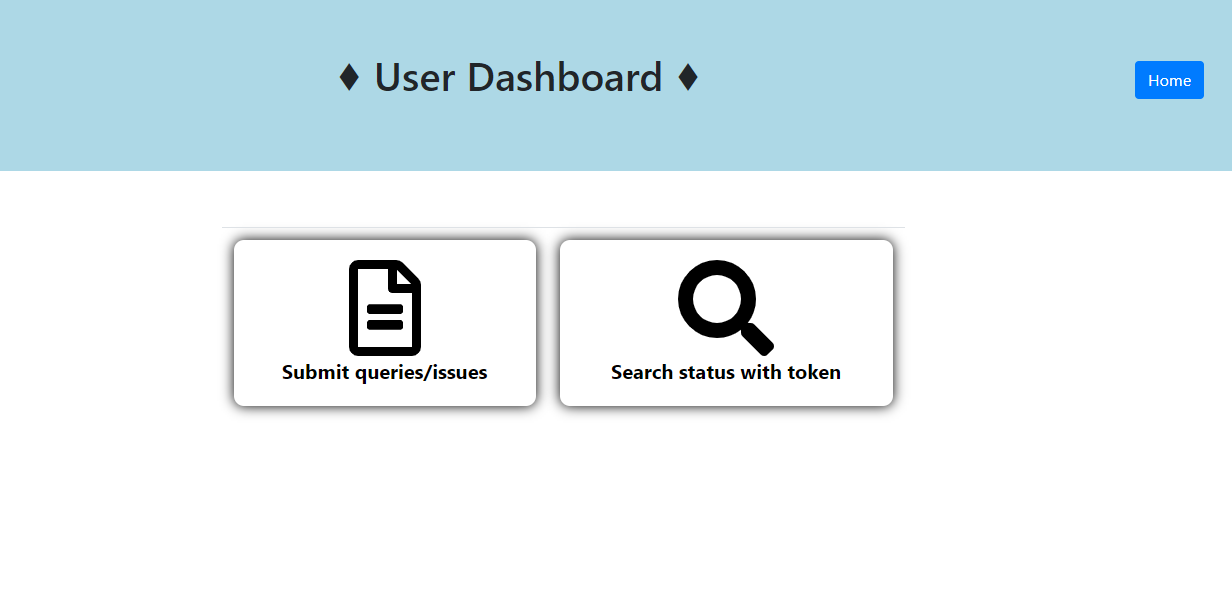
ii. Project Planning

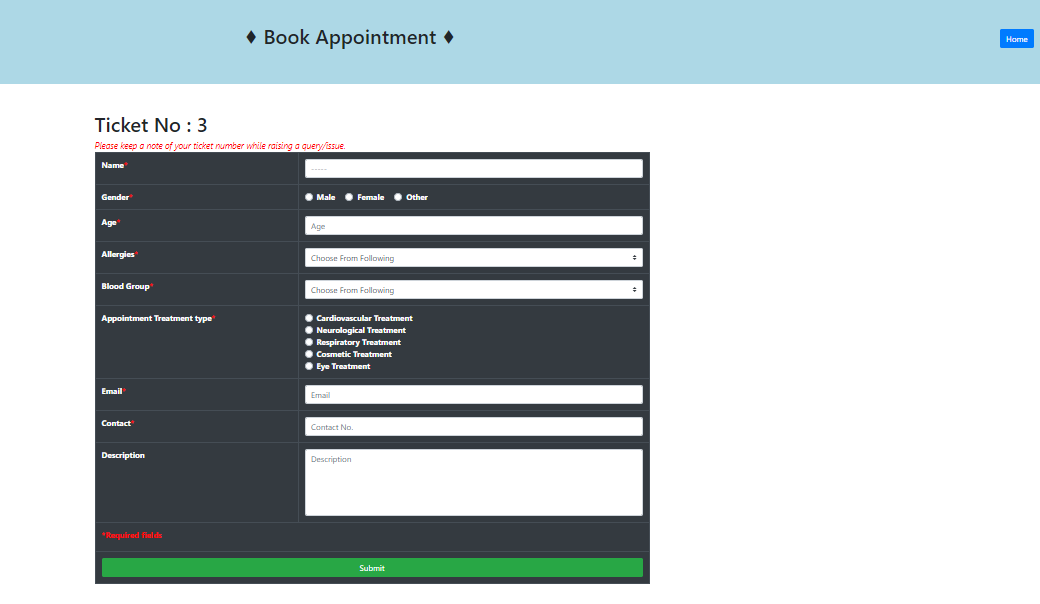
iii. The whole project is discussed with mentor and team members

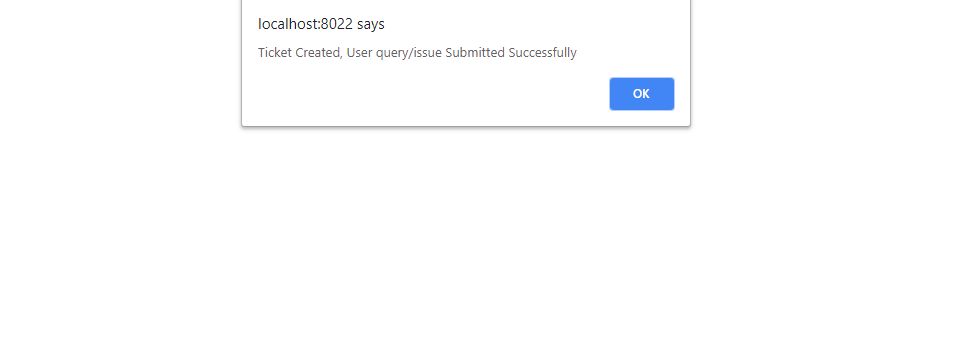
iv. Time management

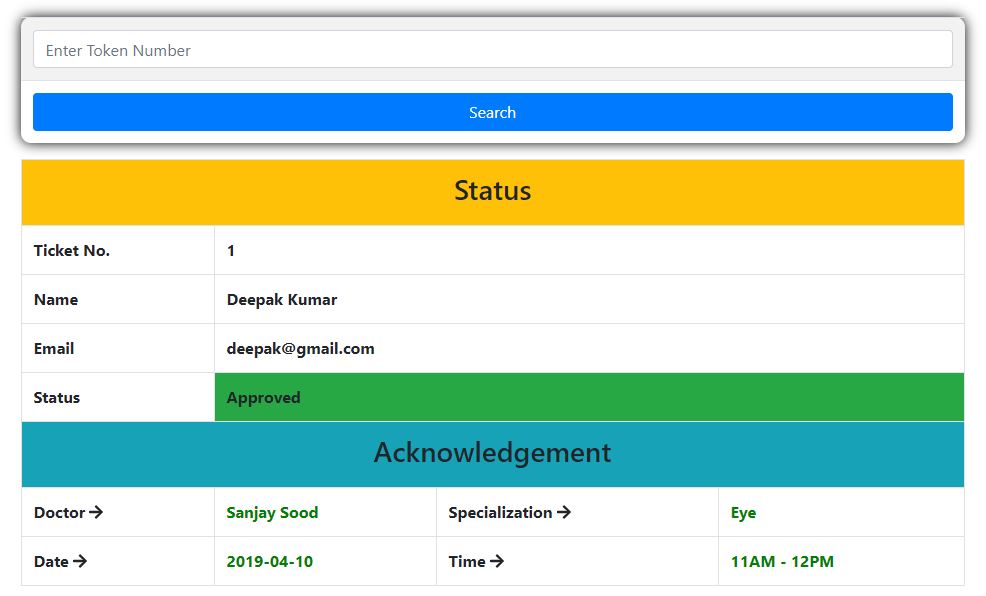
# 10.Snapshots

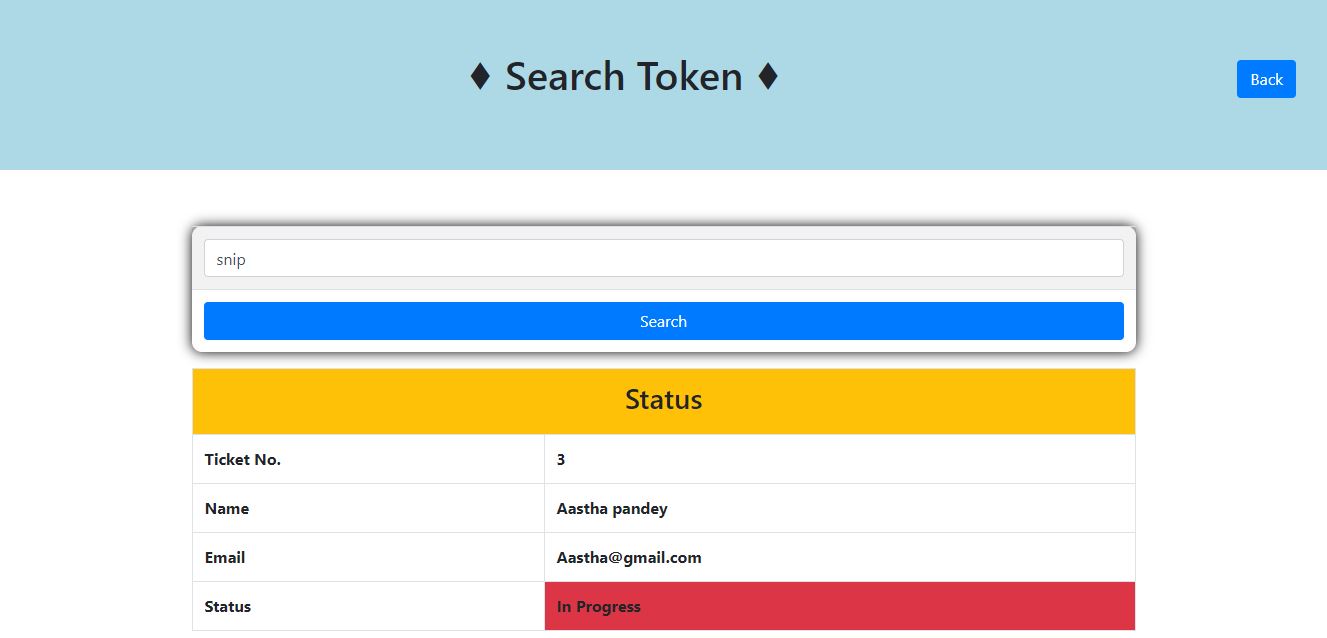


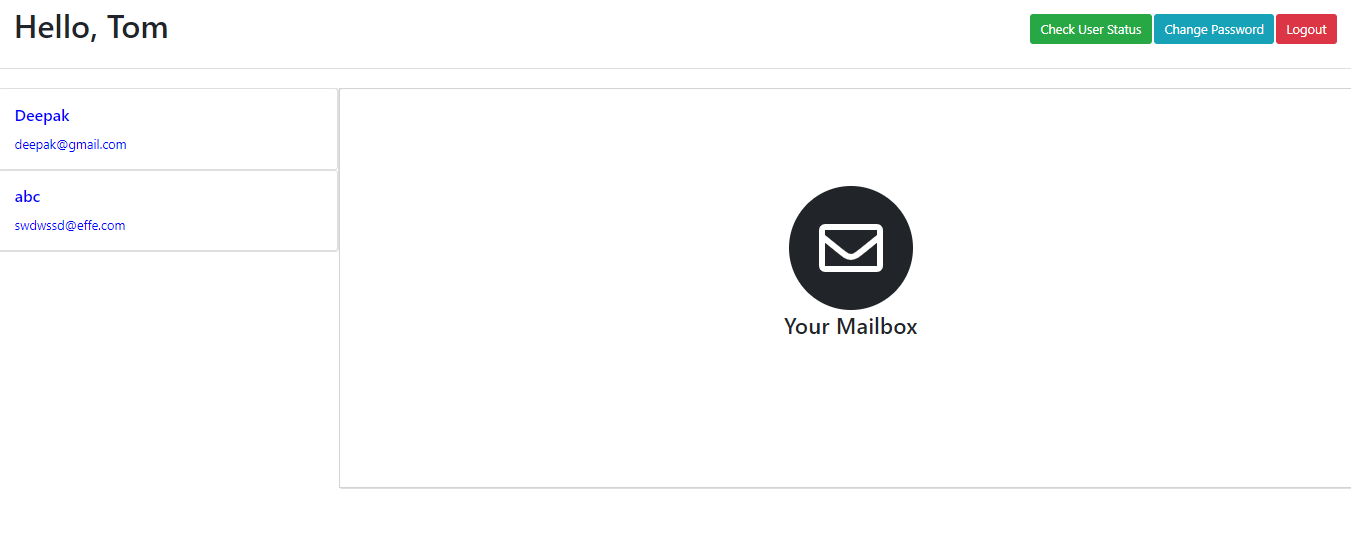


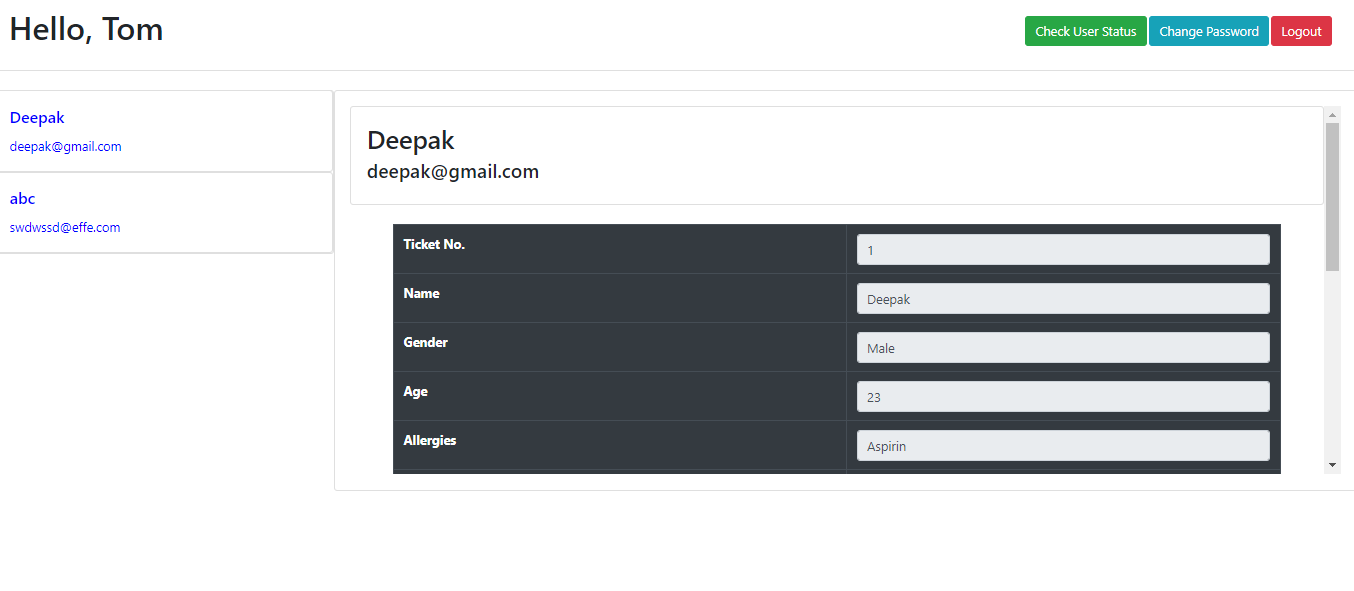


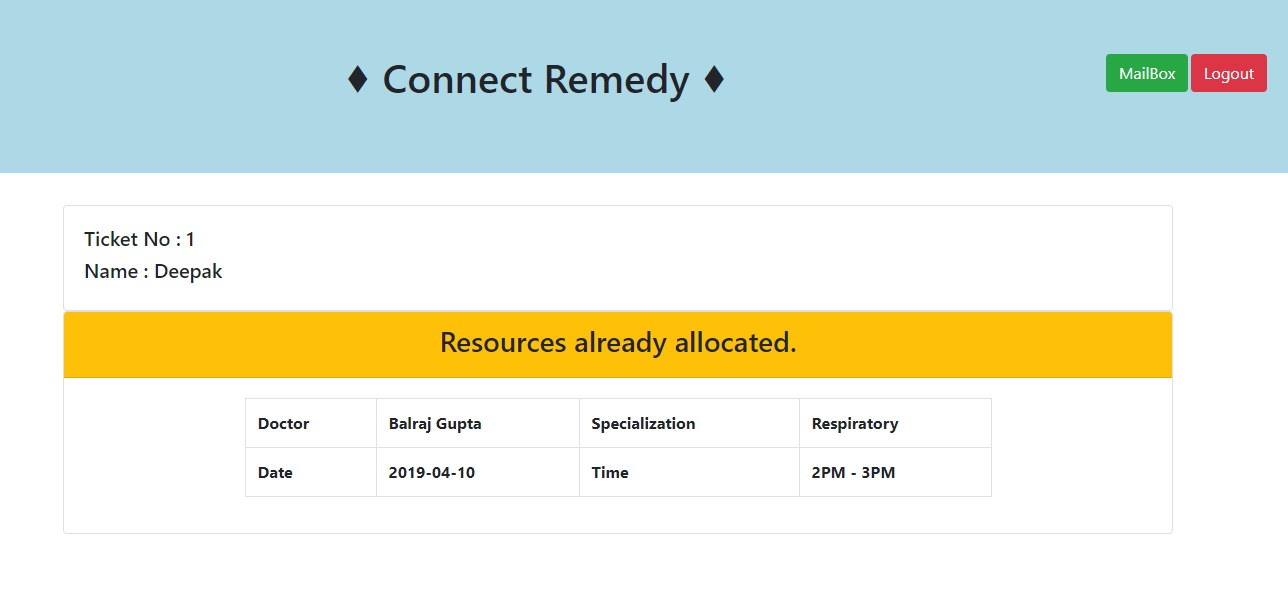
****





****

****

****

# 11.Bibliography

[1] https://www.tutorialspoint.com

[2] https://getbootstrap.com/docs/4.1/getting-started/introduction/

[3] https://www.javatpoint.com/

[4] https://www.w3schools.com/bootstrap4/