

E-APPOINTMENT SYSTEM

**(Guided By Prof. Roxana Rohde)**

Prepared By

Sangeeta Joshi (90093)

Ekta Khiani (90109)

Shravani Edem (90613)

Nilam Patil (90200)

Shweta Sahu ()

Abdul Sattar (86306)

**Contents**

Document Revision History 4

Acknowledgement 5

1 Project Overview 7

1.1 Project Concept 8

1.1.1 Purpose 8

1.1.2 Scope 9

1.1.3 Product Perspective 9

1.2 Team and Roles 9

1.3 Development Model 9

1.3.1 Feature-Driven Development (FDD): 9

1.4 Problem Solved 10

1.5 Expected Audience 10

2 Project Requirement 13

2.1 Functional Requirements 13

2.2 Non Functional Requirements 14

2.3 Organizational and Derived Requirements 15

2.4 E-Appointment Web App 16

2.4.1 Admin User Functionality 16

2.4.2 Prospect Student Functionality 17

2.4.3 Current Student Functionality 17

2.4.4 Staff Functionality 17

2.5 System Functionality 18

2.6 Use Case Diagrams 19

2.6.1 Admin Use Case 20

2.6.1.1 Admin Use Case Description 20

2.6.2 Prospect Student Use Case 21

2.6.2.1 Prospect Student Use Case Description 23

2.6.3 Current Student/Staff Use Case 24

2.6.3.1 Current Student Use Case Descriptions 25

2.6.3.2 Staff Use Case Description 25

2.7 Activity Diagram 26

2.8 Sequence Diagram 28

2.8.1 Current Student Login Sequence Diagram 28

2.8.2 Staff Sequence Diagram 29

2.8.3 Admin Sequence Diagram 30

3 Project Planning 32

3.1 WBS 33

3.2 Effort Estimation v/s Actual Effort 33

3.3 Timeline (#sprints, iterations, milestones) 33

4 System Design Overview 35

4.1 System Architecture 36

4.2 Software Architecture 37

4.2.1 Event Driven Architecture 37

4.3 Tool & Technologies 38

5 Implementation 40

5.1 GIT Links 40

6 Testing 43

6.1 Test Plan Objective 43

6.2 Test Strategy 44

6.2.1 Testing types 44

6.2.1.1 Unit Testing 44

6.2.1.2 Integration Testing 45

6.2.1.3 Regression Testing 45

6.2.1.4 Acceptance Testing 45

6.2.1.5 System Testing 46

6.3 Suspension Criteria and Resumption Requirements 46

6.3.1 Suspension Criteria 46

6.3.2 Resumption Requirement 46

6.4 Pass or Fail Criteria 46

6.5 Test cases and test results 46

6.6 #Bugs detected and how tracking of bugs was done to completion 46

6.7 Coverage of requirements and design with tests (%) 46

6.8 Regression tests 47

7 Challenges 48

# Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPTION OF CHANGE** | **DATE** | **VERSION** | **MODIFIED BY** |
| Initial Draft | 2/15/16 | v0.1 | Sangeeta Joshi |
| Review | 4/20/16 | v0.2 | Sangeeta Joshi |
| Final review and formatting done | 4/24/16 | v0.3 | Sangeeta Joshi |
| Baseline version | 4/25/16 | v0.4 | Sangeeta Joshi |

# Acknowledgement

We express our profound sense of gratitude in all its humbleness to our beloved Professor Roxana Rohde, Faculty, International Technological University, USA for her gracious guidance, meticulous care and unstinted co-operation throughout our work in delivering this project.

We take this opportunity to thank her for the valuable support and guidance extended to us, which helped in the successful completion of the project.

We would like to express our heartfelt thanks to members of our Project for their involvement and contributions throughout the project in accomplishing the tasks assigned.

PROJECT OVERVIEW

# Project Overview

Our online E-Appointment system will help the students to overcome the problem faced by using the manual process and thereby making the appointment easier to make.

All the manual activities can be done using the computerized E-Appointment.

On the other hand the students, who prefer manual process, can still use existing manual system. The requirements of the E-Appointment system build a system for students, Staff wishing to automate appointments between prospect student, Current student and staff.

This system will streamline and automate many of the tasks (such as seeking time from ISO and faculty, scheduling an appointment) currently consuming front desk staff’s day—saving them precious hours they can devote towards other more productive activities. In addition, student and staff’s online appointment features not only provide convenience, but save more time as well. We hope the document can accurately portray a sample of what such a software suite would require and how it could be coded to become a reality.

## Project Concept

### Purpose

The purpose of this document is to serve as a guide to designers, developers and testers who are responsible for the engineering of this project. It should give the engineers all the information necessary to design, develop and test the software. This document contains software requirements and design outline for online E-Appointment System, which will facilitate ease of making appointments. Some prospect students are not local residents so it is not feasible to go physically to the university and schedule an appointment. This online system will provide simple, secure, quick way for prospect students to fill out and complete their appointment online – 24/7.

### Scope

This document contains a complete description of the functionality of online E-Appointment system project. It consists of use cases, functional requirements and nonfunctional requirements, which, when combined together form a complete description of the software.

### Product Perspective

E-Appointment will work with most common web browsers like Google Chrome, Internet Explorer, Firefox, etc.

## Team and Roles

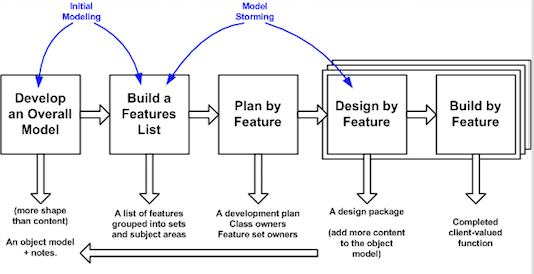
|  |  |  |
| --- | --- | --- |
| **NAME** | **ID** | **ROLE** |
| Sangeeta Joshi | 90093 | Project Manager |
| Ekta Khiani | 90109 | Technical Architect/Developer |
| Shravani Edem | 90613 | Software Developer |
| Abdul Sattar | 86306 | Software Developer |
| Shweta Sahu |  | QA |
| Nilam Patil | 90200 | QA |

## Development Model

### Feature-Driven Development (FDD):

1. Agile iterative development methodology

2. Based on breaking down the requirements into small client-valued pieces of functionality



## Problem Solved

The system is developed to 'upgrade' the manual system into computerized system minimizing the manual effort and resources. It provides ease of making appointment online. The authentication process of online system will allow only right people to access right data and change it accordingly.

## Expected Audience

|  |  |  |  |
| --- | --- | --- | --- |
| **NAME** | **DESCRIPTION** | **RESPONSIBILITY** | **STAKEHOLDER** |
| Admin | User of E-Appointment web app. | Admin needs to create the profile for every user (current students and staff) in the app and assign them a role. Once the user is created, admin will send the credentials to the users on ITU's email.  Admin can also update the information of existing user and delete the users form the system. | Self |
| Staff/Faculty | Authorized user of E-Appointment web app. | Review, approve and reject appointments received through E-Appointment Web app.  Also, manage their appointment time by setting up their available days and time.  Can also update their user profile and change the password. | Self |
| Current Student | Authorized user of E-Appointment web app. | Can schedule an appointment online using E-Appointment web app.  Current student can schedule appointment with ISO, faculties, Academic advisors and admissions.  An appointment with ISO should be made 24 hours in advance.  Can also update their user profile and change the password. | Self |
| Prospect Student | Guest user of E-Appointment web app. | Any prospect student can make appointment using the guest appointment feature, but they are limited to have appointments only with admissions office. | Self |

PROJECT REQUIREMENT

# Project Requirement

## Functional Requirements

In [software engineering](http://en.wikipedia.org/wiki/Software_engineering) (or [system engineering](http://en.wikipedia.org/wiki/System_Engineering)), a functional requirement defines a function of a [system](http://en.wikipedia.org/wiki/System) or its component.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **DESCRIPTION** | **TYPE** | **PRIORITY** |
| FR1 | The prospect student should be able to continue as a guest and be able to make an appointment online with admissions office. | User Interface | M |
| FR2 | The prospect should also get an email notification of the appointment. | User Interface | M |
| FR3 | ITU’s students, staff and admin must be able to log into the system using their authorized credentials | User Interface | M |
| FR4 | Admin should be able to login to system and create user into the app and assign them a role. | User Interface | H |
| FR5 | Admin should be able to send the E-Appointment credentials to the users registered ITU email id |  | M |
| FR6 | Admin should be able to update the user information | User Interface | M |
| FR7 | Admin should be able to delete the user if they are no longer associated with ITU. | User Interface | M |
| FR8 | The staff should be able to login to the system after authentication and take action (approve, reject) on the appointment | User Interface | M |
| FR9 | The staff should be able to view the new appointments made by the students to them | User Interface | M |
| FR10 | The staff should provide the reason when rejecting any appointment. | User Interface | H |
| FR11 | Email notification must be sent to the student upon approval of rejection of the appointment |  | M |
| FR12 | The staff should be able to set up their appointment availability. | User Interface | H |
| FR13 | The staff should be able to edit their profile like mobile number and should be able to change password. | User Interface | M |
| FR14 | The student should be able to login to the system after authentication and make appointment. | User Interface | M |
| FR15 | The student should be able to view all the appointment made by them. | User Interface | M |
| FR16 | The student should be able to schedule an appointment after selecting the respective department or faculty. | User Interface | M |
| FR17 | The student should be able to edit their profile like mobile number and should be able to change password. | User Interface |  |

Table 2.: Functional Requirements

## Non Functional Requirements

In [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering) and [requirements engineering](http://en.wikipedia.org/wiki/Requirements_engineering), a non-functional requirement is a [requirement](http://en.wikipedia.org/wiki/Requirement) that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **DETAILS** | **TYPE** | **PRIORITY** |
| NF1 | The Online E-Appointment should provide database modification permission only to the admin and staff members after complete authentication and authorization procedures. The student should be able to view and update his profile and make appointment only after complete authentication and authorization. | Safety &  Security | M |
| NF2 | The system should provide an online web application that provides an interface, which is user friendly, simple and elegant. The interface should have a structured architecture, which should enable the user to easily navigate the system and quickly get the required information. | Usability | M |
| NF3 | The system should be available to all the users 24X7. The screening of the application should be done at a very fast pace of about 15-30 seconds. The response time of the E-Appointment should be very quick and fast enough to avoid data update collisions. | Performance | M |
| NF4 | The system should be able to provide real time information about the faculty availability. The system should be highly reliable to ensure first come first appointment of time, in the case of receiving multiple appointment for the same faculty. | Reliability | M |
| NF5 | The E-Appointment is a web based application and should be platform independent and support multiple devices. | Software and Hardware Constraints | M |

Table 2.: Non Functional Requirements

## Organizational and Derived Requirements

In system engineering process, derived requirements are inferred or derived from user requirements. These requirements are developed through Requirement analysis.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **DETAILS** | **TYPE** | **PRIORITY** |
| D1 | Appointment availability should be updated. | Usability | M |
| D2 | Appointment with the ISO should be made 24 hours in advance. | Business | M |
| D3 | The system should be able to operate in extreme temperatures and climatic conditions. | Performance | M |
| D4 | The system should allow the staff to accept or reject the appointment. | Usability | M |
| D5 | The web interface should adapt to screens of all sizes. | User Interface | M |

Table 2.: Organizational and derived requirements

## E-Appointment Web App

* E-Appointment provides web interface to the students looking for the appointment online.
* Through this web interface, a prospect student can browse through the app and book an appointment online
* The E-Appointment web interface is a user interactive application through which, a user can schedule an appointment with ease.

### Admin User Functionality

* Login/Sign out – Admin needs to first log into the system in order to Add, Update and Delete users. After logging in, admin can update database. After making necessary changes, the admin needs to successfully log out of system in order to avoid misuse/theft of data.
* Add User – Admin can create the profile for every user such as student, faculty, staff and assign them a role. Admin also send the initial login credentials to users email id.
* Update User – Admin can search the user by name or id and update their information.
* Delete User – Admin can also delete the user by searching then with name or id, if they are no longer associated with ITU.

### Prospect Student Functionality

* Make Appointment – Prospect student are considered as the guest and they could make an appointment without logging into the system. These appointments can only be made with admissions office. Email notification is sent to the prospect student when they submit the appointment online.

### Current Student Functionality

* Login/Logout – Current students needs to first log into the system in order to make appointment. After logging in, student can make appointment. After making necessary appointments, the student needs to successfully log out of system in order to avoid misuse/theft of data.
* My Appointment – Students can view all of their appointments.
* Make Appointment – Student can make appointment online by choosing the respective department faculty. Email notification will be sent on appointment scheduling.
* Edit Profile – Student can also update their profile like updating mobile number, changing password.

### Staff Functionality

* Login/Logout – Staff needs to first log into the system in order to approve reject and set up appointment. After making necessary changes, the staff needs to successfully log out of system in order to avoid misuse/theft of data.
* My Appointment – Staff can all the appointments that they have received, they can also approve or reject appointment based on their availability. Email notification must be sent on approval of rejection of the email. In case of rejection staff must provide a reason.
* Set Up Appointment – Staff can set their appointment availability using this feature.
* Edit Profile – Staff can also update their profile like updating mobile number, changing password.

## System Functionality

* Create/Update/Delete User – E-Appointment system will be used by Admin to create different types of users and assigning them a role such as student, staff. User information can also be updated. Admin could also delete a user.
* Make Appointment – E-Appointment system can make the online appointments for the Prospect students and current students.
* Manage Appointment - The staff user of E-Appointment system can manage their appointment by accepting or rejecting it. They can also set their appointment availability.
* Email – Email notifications will be sent on any action taken on the appointment.
* Maintains Database – E-Appointment stores all the appointments, students and staff information in database and maintains data correctly.
* Sign out – E-Appointment provides a facility to sign out from the app.

## Use Case Diagrams

Use case diagrams are behavior diagrams, which are used to describe a set of actions also referred to as use cases, which some system or systems (subject) will be able to perform in collaboration with many external users of the system actors. All the system actors or the stakeholders are provided with some observable and valuable result through every use case.

### Admin Use Case

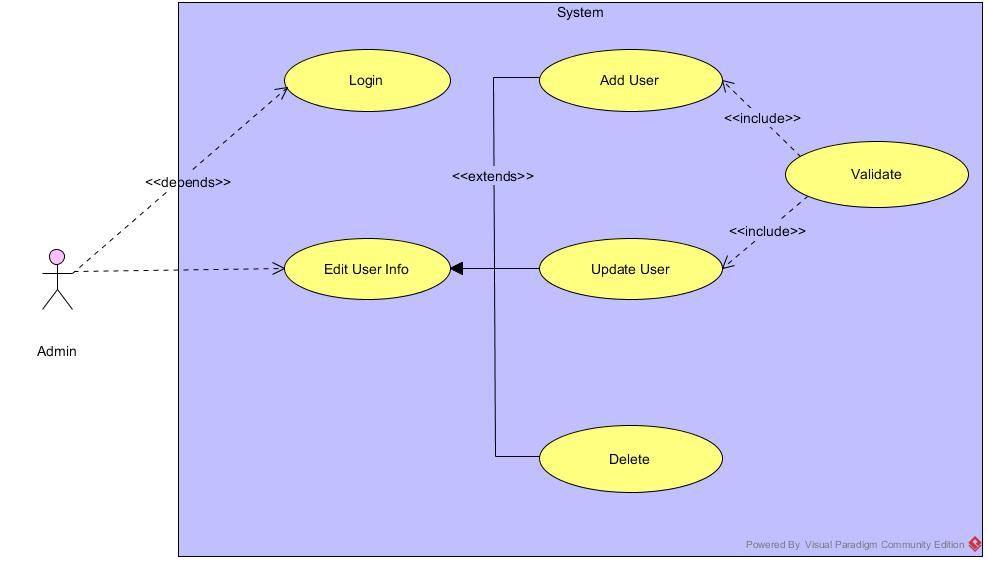


Figure 2.6.1: Admin Use Case

#### Admin Use Case Description

|  |  |
| --- | --- |
| **Use Case ID** | UC\_1 |
| **Use Case Name** | Admin Module |
| **Actor(s)** | Admin |
| **Goal in context** | The main purpose of this use case is to allow admin to add, update and delete user. |
| **Pre-conditions** | 1 .The system is running properly.  2. Admin logs into the system successfully |
| **Post-conditions** | 1. Different types id users will be created with a specific role such as student, staff  2. User credentials should be sent to the users ITU authorized email id. |
| **Basic Flow of Actions** | 1. The actor will open the E-Appointment user interface.  2. Then admin will login to the system user admin credentials.  3. Admin will have 3 sub tabs in the UI to add, update and delete user.  4. Admin can create users using the add user interface and assign them a role.  5. Admin can search a user by id or name in order to update the user info  6. Admin can delete the user by searching them by id or name. If they are no longer associated with ITU.  7. Can generate the report of the prospect student showing interest in the university programs. (Nice to have) |

### Prospect Student Use Case

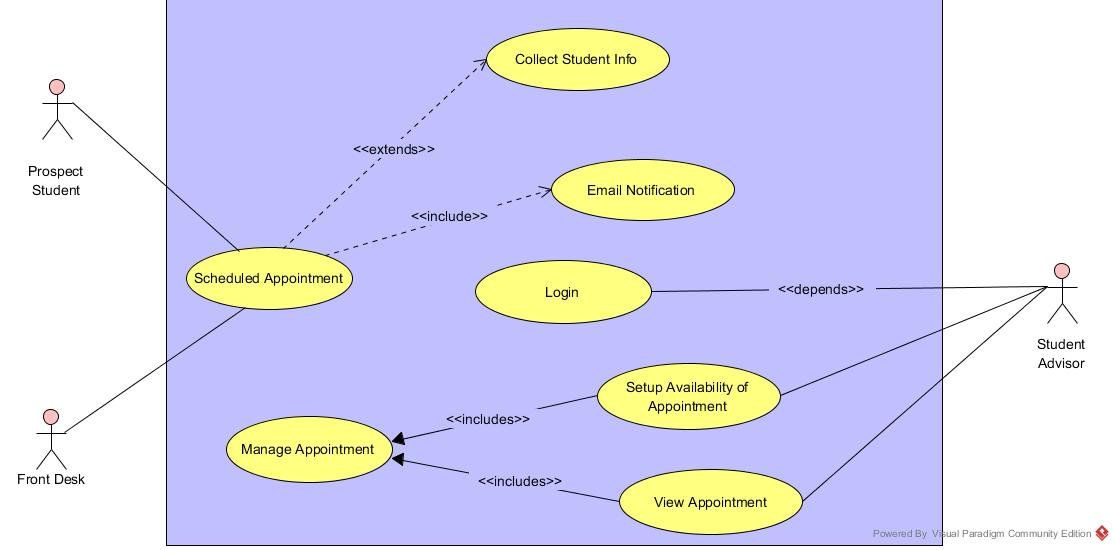


Figure 2.6.2: Prospect Student Use Case

#### Prospect Student Use Case Description

|  |  |
| --- | --- |
| **Use Case ID** | UC\_2 |
| **Use Case Name** | Prospect Student Use Case |
| **Actor(s):** | Prospect Student, Admission advisors, Front Desk |
| **Goal in context:** | The main purpose is to enable the prospect student make appointment online for their admission related queries. |
| **Pre-conditions:** | 1. The system is running properly.  2. The prospect student should be able to access the system as guest to make appointment. |
| **Post-conditions:** | 1. Appointment will be scheduled with the admission advisor.  2. An email notification will be sent to the prospect student for appointment confirmation. |
| **Basic Flow of Actions:** | 1.  The actor will open the E-Appointment system.  2. Prospect student can access the required part of the app by continuing as guest.  3. Then prospect student can schedule an appointment with admission advisors.  4. Prospect student can also call the front desk to make appointment on their behalf.  5. Front desk then uses the same system to schedule an appointment for prospect student.  6. Email notification is sent to the student for their appointment confirmation.  7. Admission advisors can login to the system using their authorized credentials.  8. Admission advisors can set their appointment availability by accessing the set up appointment page.  9. Admission advisors can view all the appointment scheduled. |

### Current Student/Staff Use Case

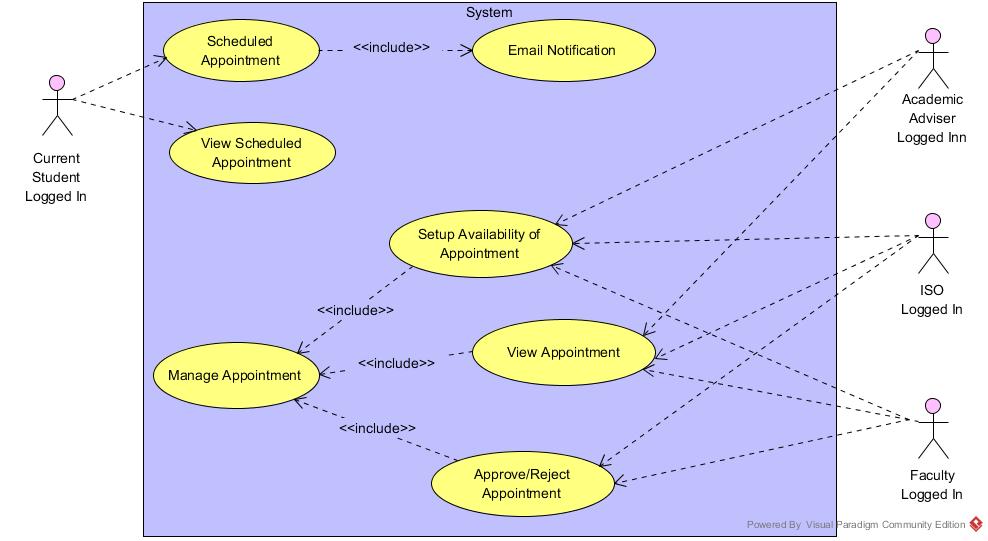


Figure 2.6.3: Current Student/Staff Use Case

#### Current Student Use Case Descriptions

|  |  |
| --- | --- |
| **Use Case ID** | UC\_3 |
| **Use Case Name:** | Current Student Use Case |
| **Actor(s):** | Current Student |
| **Goal in context:** | The main purpose of this use case is to allow student to make appointment online with their faculty, ISO and academic advisors. |
| **Pre-conditions:** | 1.The student should be logged into the system  2. The staff should be able to access the staff data from the database. |
| **Post-conditions:** | 1. Appointment should be scheduled with the respective department or team.  2. Email notification must be sent. |
| **Basic Flow of Actions:** | 1. The actor will log in to the E-Appointment system.  2. Then student can view all the appointment already scheduled.  3. Student can search for the respective department or faculty in order to schedule appointment.  4. Student can choose the staff and schedule an appointment.  5. Email notification should be sent on appointment scheduling.  6. Student can edit their info using Edit profile. |

#### Staff Use Case Description

|  |  |
| --- | --- |
| **Use Case ID** | UC\_3 |
| **Use Case Name:** | Staff Use Case |
| **Actor(s):** | Staff (ISO, Faculty, Academic advisors) |
| **Goal in context:** | The main purpose of this use case is to allow resident to view or update personal details and view payment history. Resident can also pay rent, reserve amenities or choose lease option through their profile. |
| **Pre-conditions:** | 1. The Staff should be logged into the system. |
| **Post-conditions:** | 1. Student will be notified by email if their appointment is accepted or rejected. 2. Student can see the current availability of the staff for appointment. |
| **Basic Flow of Actions:** | 1. The actor will log in to the E-Appointment system.  2. Then staff will view appointments and screen appointments.  3. Staff can accept or reject the appointment.  4. In case of the rejection, staff must provide a reason.  5. Email notification must be sent on approval or rejection of the appointment.  6. Staff can set their appointment availability by accessing the set up appointment page.  7. Staff can edit their info using Edit profile. |

### 

## Activity Diagram

An activity diagram is a graphical representation of business process or software algorithm as a series of actions. Activity diagram is basically a flow chart representing the flow form one activity to another activity. The activity can be described as an operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but using forward and reverse engineering techniques also uses them to construct the executable system.

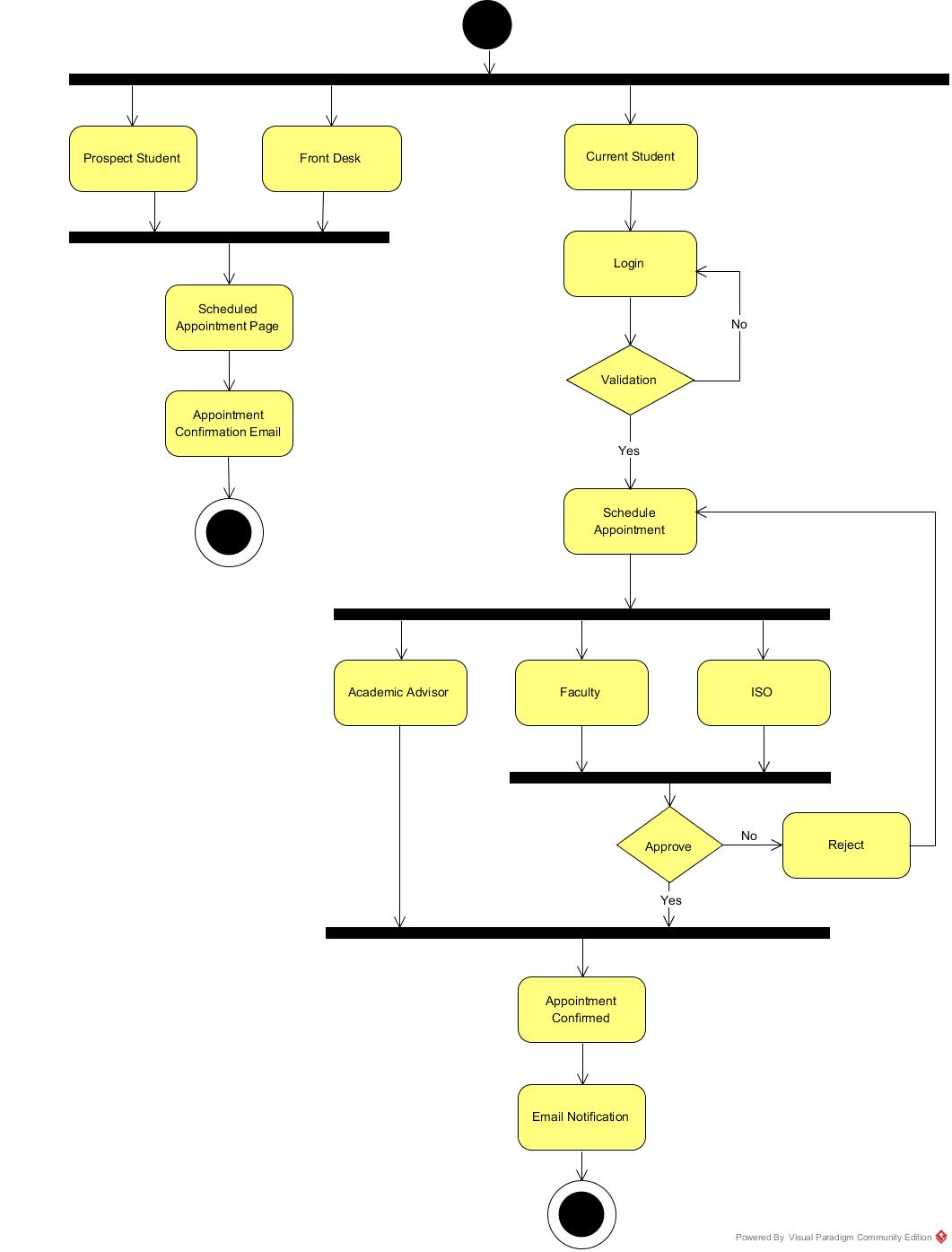
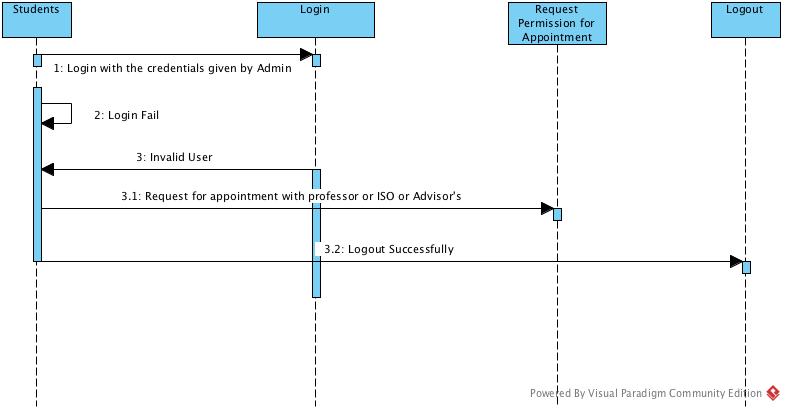


Figure 2.7: Activity Diagram

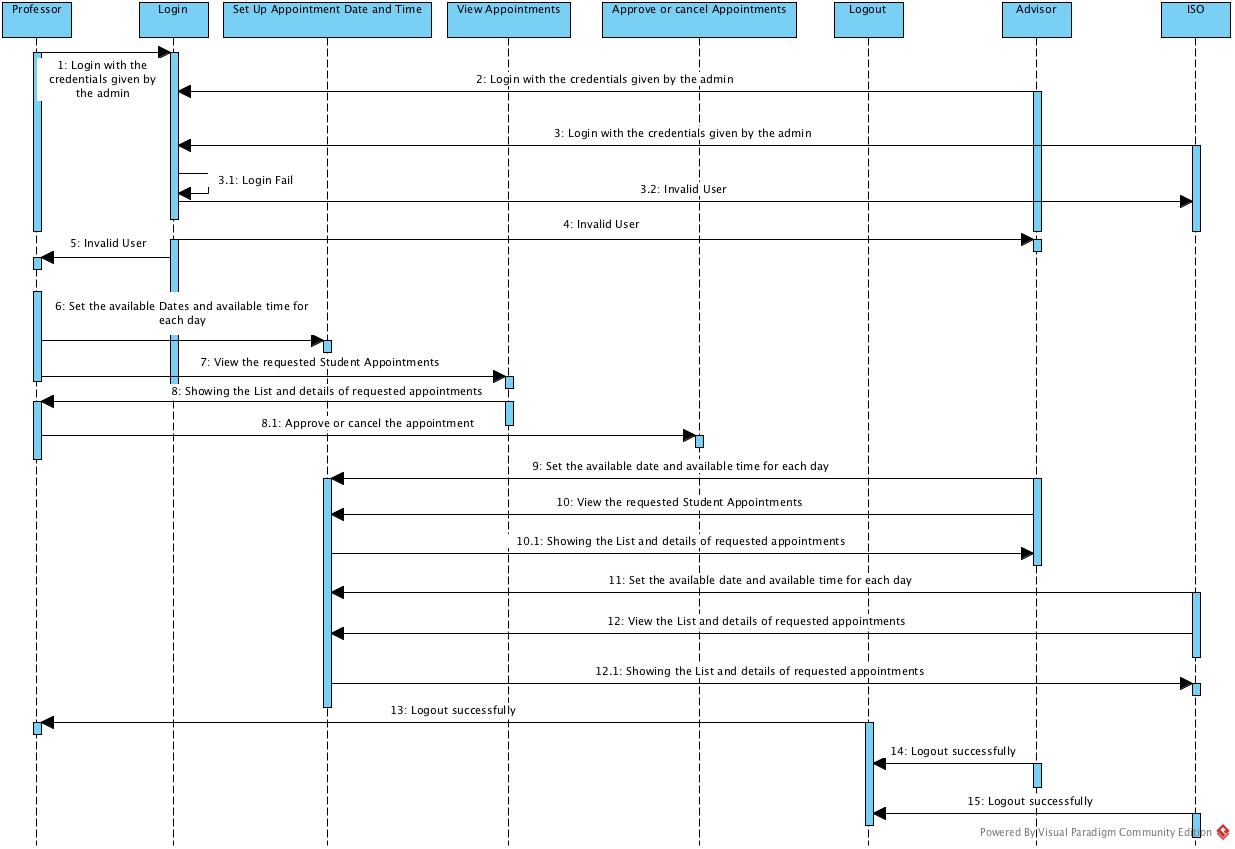
## Sequence Diagram

The primary use of the sequence diagram is to show the interactions between the objects in the order of sequence in which they occur. Sequence diagram also facilitates the transition from requirements depicted through use cases to the next and more formal level of refinement. They are very useful in the process of designing new system. In addition to this, sequence diagrams prove to be very helpful in documenting the interactions of the objects in an existing legacy system. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the activity and specifies the sequence of messages being exchanged between the objects in order to achieve the functionality of the scenario.

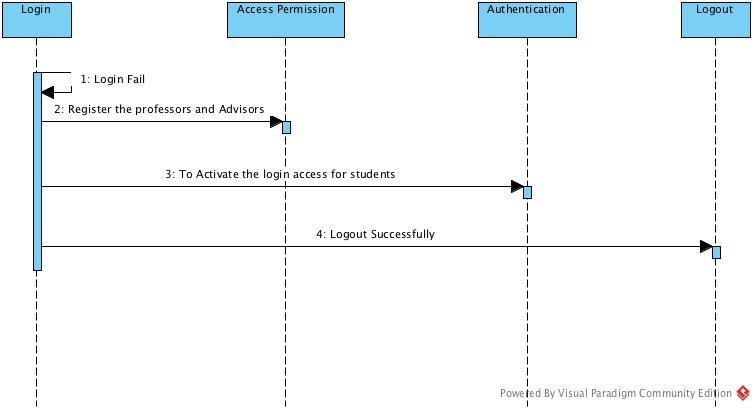
### Current Student Login Sequence Diagram



### Staff Sequence Diagram



### Admin Sequence Diagram



PROJECT PLANNING

# Project Planning

Software life cycle models describe phases of the software cycle and the order in which those phases are executed. Each phase produces deliverables required by the next phase in the life cycle.

Requirements are translated into design. Code is produced according to the design, which is called development phase. After coding and development the testing verifies the deliverable of the implementation phase against requirements.

There are following six phases in every Software development life cycle model:

Requirement gathering and analysis

Design

Implementation or coding

Testing

Deployment

Maintenance

1. Requirement gathering and analysis:  Business requirements are gathered in this     phase. This phase is the main focus of the project managers and stakeholders. Meetings with managers, stakeholders and users are held in order to determine the requirements like; who is going to use the system? How will they use the system?  What data should be input into the system?  What data should be output by the system?  These are general questions that get answered during a requirements gathering phase. After requirement gathering these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied. Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.
2. Design:  In this phase the system and software design is prepared from the requirement specifications that were studied in the first phase. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.
3. Implementation / Coding:  On receiving system design documents, the work is divided in modules/units and actual coding is started. Since, in this phase the code is produced so it is the main focus for the developer. This is the longest phase of the software development life cycle.
4. [Testing](http://istqbexamcertification.com/what-is-a-software-testing/):  After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. During this phase unit testing, integration testing, system testing, acceptance testing are done.
5. Deployment: After successful testing the product is delivered / deployed to the customer for their use.
6. Maintenance: Once when the customers start using the developed system then the actual problems comes up and needs to be solved from time to time. This process where the care is taken for the developed product is known as maintenance.

## WBS

GIT link in GIT Link Section

## Effort Estimation v/s Actual Effort

WBS GIT Link in GIT Link Section

Adhered to estimated timeline and completed the Project.

## Timeline (#sprints, iterations, milestones)

|  |  |  |
| --- | --- | --- |
| **SPRINT** | **TIMELINE** | **MILESTONES** |
| SPRINT 1 | Week 3 - Week 5 | Requirement analysis, Wireframes and design |
| SPRINT 2 | Week 6 – Week 7 | Development (Login, Prospect Student make appointment) |
| SPRINT 3 | Week 8 – Week 9 | Development (Current Student Schedule appointment, Staff - Set up appointment) |
| SPRINT 4 | Week 10 – Week 11 | Development (Current Student-My appointment, Staff- My Appointment, with Accept and Reject and Mail, Edit Profile- Student/Staff) |
| SPRINT 5 | Week 12 – Week 13 | Development (Admin-Add/Update/Delete User, Sign out) |

ARCHITECTURE & DESIGN

# System Design Overview

The E-Appointment system is a web-based widespread user-friendly system. The user can use web browser to access the system for activities like scheduling and viewing appointment. Another advantage is that E-Appointment system is platform independent; it works on all kinds of popular operating systems such as Unix, Linux, Windows or Mac. Front end UI will be developed using JSF and Java script and the server side application will be developed using JSP and JAVA. Platform independent feature increases the flexibility of the system, so that there is no need of new hardware that uses the E-Appointment system. The source code reusability will reduce the cost of application.

The UI data related to appointment availability is automatically refreshed and loaded from database.

## System Architecture

The architecture of the E-Appointment System is based on the three-tier architecture. This three-tier architecture mainly consists of three layers namely:

1. Presentation Tier
2. Business Tier
3. Data Access Tier

The Presentation Tier converts and displays information into a human legible form. This tier displays information related to services such as scheduling and viewing appointment. It communicates with the other tiers by outputting results to the browser/client tier and all the other tiers. The Business Logic tier is mainly responsible for information exchange between the user interface and the database of the project. The final layer of the three-tier architecture is the Data Access tier, which mainly consists of the Database servers. The information related to the users and appointment is stored and retrieved from here.

A simple representation of the three-tier architecture would be as follows:

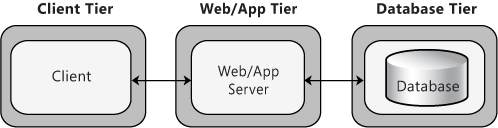


Figure 3.: Three Tier Architecture

The three-tier architecture would be discussed in detail in the following sections:

**1. Presentation Tier**

Occupies the top level and displays information related to services available on a website. This tier communicates with other tiers by sending results to the browser and other tiers in the network. The presentation tier is mainly responsible for the user interface of the application, which deals with the presentation of data to the user.

**2. Business Logic Tier**

Also called the middle tier, logic tier, business logic or logic tier, this tier is pulled from the presentation tier. It controls application functionality by performing detailed processing. The business logic for the E-Appointment System would be present here. In this project, the Java classes would be performing the duty of the business logic. This is the layer, which is responsible for the information exchange between the user interface and the database. The E-Appointment system mainly consists of the Users, which can be further classified into the Admin, staff, Prospect Student and Current student of the System website.

**3. Database Tier**

The database tier is the final and last tier of the three-tier architecture. All the data related to the E-Appointment System project is stored and retrieved from here. For this project we have used the MySQL Enterprise Edition Server to create the database. It is very easy to work with and makes creation and maintaining of tables very easy.

## Software Architecture

### Event Driven Architecture

* Event-driven architecture (EDA, also known  
  as Message-driven architecture) is a software architecture pattern promoting the production, detection, consumption of, and reaction to events.
* An event can be defined as "a significant change in state“
* Sequence diagram is the most common kind of interaction diagram, which focuses on the message interchange between a number of lifelines.
* Sequence diagram describes an interaction by focusing on the sequence of messages that are exchanged, along with their corresponding occurrence specifications on the lifelines.

## Tool & Technologies

|  |  |  |  |
| --- | --- | --- | --- |
| **TOOL/TECHNOLOGY NAME** | **VERSION** | **USED IN** | **WHY USED?** |
| Trello |  | Project Management | Easy Task management and Free for use |
| Java, Javascript, JQuery, HTML, Bootstrap |  | Development | Technology for web app development |
| GIT |  | Versioning | Great versioning tool and free to use |
| Source Tree |  | Versioning | Versioning desktop app for GIT |
| Visual Paradigm |  | Use Case Diagrams | Easy to use and free |
| Visual Paradigm |  | Sequence Diagrams | Easy to use and free |
| Selenium | 2.9.1 | Automation Testing | Automation, easy to use |
| Net Beans | 8.1 | Development IDE | IDE for development |
| My SQL Server | 5.7.11 | Database | Database |
| Oracle Web logic Server | 12c | Server | Server connection with IDE |
| Microsoft Excel |  | WBS, Project Status, Manual Test Cases | Inbuilt in System and easy to use |
| Microsoft Word |  | Requirement, Final Report | Inbuilt in System and easy to use |
| Microsoft Power point |  | Demo PPT | Inbuilt in System and easy to use |
| Keynote |  | Team Presentation Slides | Inbuilt in System and easy to use |

IMPLEMENTATION

# Implementation

## GIT Links

|  |  |
| --- | --- |
| **DOCUMENT** | **REPO LINK** |
| Team Presentation PPT | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Project%20Management/Team%20Presentation> |
| WBS | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Project%20Management/WBS> |
| Requirement Doc (SRS) | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Documentation/SRS> |
| Project Status Report | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Project%20Management/Project%20Status> |
| MOM | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Project%20Management/MOM> |
| Test Cases (Manual) | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Testing/Test%20Cases/Manual> |
| Automation Test Suite (Selenium) | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Testing/Test%20Cases/Automation> |
| Testing reports and BUGS | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Testing/Bug%20Report> |
| RTM | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Documentation/RTM> |
| UI Wireframes | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Documentation/Screens/New> |
| Release Plan | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Project%20Management/Release/Release%20Plan> |
| Release Notes | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Project%20Management/Release/Release%20Notes> |
| Source Code location | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Source%20Code/EAppointment> |
| Use Case, Activity, Sequence Diagrams | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Documentation/Sequence%20Diagrams>  <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Documentation/Use%20Case%20and%20Activity> |
| Code Review | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Documentation/Code%20review%20and%20Best%20Practices> |
| Coding Best Practices | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Documentation/Code%20review%20and%20Best%20Practices> |
| Unit Test | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Testing/Test%20Cases/Automation> |
| Project Demo Video | <https://drive.google.com/a/students.itu.edu/file/d/0B5vMzjETCODKSkx1YnZGTzg1QjA/view?ts=571d7db9> |
| Demo PPT | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Documentation/Demo/Demo%20Presentation> |
| Final Report | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Documentation/Final%20Report> |
| Proto type Screens | <https://github.com/sangeetajoshi/E-Appointment_Final/tree/master/Documentation/Proto%20Type%20Screens> |
| Selenium Demo Video | <https://drive.google.com/drive/u/1/folders/0B0SMrVb1LnqyYnltOHZPanlYNHM> |

TESTING

# Testing

## Test Plan Objective

The primary objective of testing is to assure that the system meets the full requirements, including non-functional requirements and fit metrics for each quality requirement and satisfies the use case scenarios and maintain the quality of the product.

The secondary objective of testing application systems will be to: identify and expose all issues and associated risks, communicate all known issues to the project team, and ensure that all issues are addressed in an appropriate matter before release.

The quality objectives of testing the E-Appointment application are to ensure complete validation of the business and software requirements. Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. It also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. The tests described in this document are designed to validate system level functionality of the entire system.

The objectives of the test plan include:

* + Verify software requirements are complete and accurate
  + Perform detailed test planning
  + Identify testing standards and procedures that will be used on the project
  + Prepare and document test scenarios and test cases
  + Regression testing to validate that unchanged functionality has not been affected by changes
  + Manage defect tracking process
  + Provide test metrics/testing summary reports
  + To identify and define all the activities necessary to prepare for and conduct the testing process on E-Appointment
  + To define the pass/fail criteria for each item that will be tested
  + To identify the deliverables of the testing phase
  + To define any suspension criteria and resumption techniques

## Test Strategy

The classical strategy for testing E-Appointment System begins with “testing in the small” and works outward toward “testing in the large.” In object oriented software testing, we begin with unit testing, then progress toward integration testing, and culminate with validation and system testing. Once each of the units has been tested individually, it is integrated into a program structure while a series of regression tests are run to uncover errors due to interfacing between the modules and side effects caused by the addition of new units. Finally, the system as a whole is tested to ensure that errors in requirements are uncovered. The project is using an agile approach, with bi-weekly iterations. At the end of each sprint the requirements identified for that iteration will be included and will be tested. The testing would be carried out on the E-Appointment while logging into the system as a customer or a normal user of the system.

### Testing types

#### Unit Testing

The Unit Testing is a test that is driven by class operations and state behavior, not algorithmic detail and data flow across module interface. This is mainly done to discover errors in the code of the E-Appointment System. In object-oriented software, encapsulation drives the definition of classes and objects. This means that each class and each instance of a class, attributes and the operations that manipulate these data. Because a class can contain a number of different operations and a particular operation may exist as part of a number of different classes, the meaning of unit testing changes dramatically. We can no longer test a single operation in isolation (the conventional view of unit testing) but rather as part of a class.

In the case of E-Appointment System, all the web user interfaces and their functionalities is tested. At the same time, unit testing has some disadvantages such as; it might not identify each and every error in the system.

#### Integration Testing

Integration Testing focuses on groups of classes that collaborate or communicate in some manner and the individual software modules are combined and tested as a whole unit. The main purpose of the integration testing is to test the functional and performance requirements on the major items of the project. All the modules of the E-Appointment project developed individually is combined together and tested as a whole system in the integration testing using bottom-up integration testing.

The integration test procedure irrespective of the test strategies:

1. Prepare the Integration Test Plan
2. Design the Test Scenarios, Cases, and Scripts
3. Executing the test Cases followed by reporting the defects
4. Tracking & re-testing the defects
5. Steps 3 and 4 are repeated until the completion of Integration is successfully.

#### Regression Testing

The Regression Testing is generally done whenever modifications are made to the source code of a project. The Regression Testing can also be defined as the process of testing changes made to the computer program and also makes sure that the older programming still works with the new changes. So, before any new version of a software product is released, the old test cases for the project will be run against the software with the changes made, to make sure that the old functionalities of the project still work.

#### Acceptance Testing

This testing is generally performed when the project is nearing its end. This test mainly qualifies the project and decides if the users of the system will accept it. The users or the customers of the project are responsible for the test.

#### System Testing

The system testing is mainly done on the whole integrated system to make sure that the project that has been developed meets all the functional requirement specification and non-functional requirements. The test cases for the system testing will be the combination of unit and integration tests.

## Suspension Criteria and Resumption Requirements

### Suspension Criteria

Testing for all the dependent features will be suspended if a test case fails. The failed test case will be logged onto the test log, which contains the description for the error.

### Resumption Requirement

The test cases that are not dependent on the case where the bug is reported will be executed in parallel with the bug fixing. Once the failed test case has been taken note of and has been identified and fixed then the testing for the failed test case will resume.

## Pass or Fail Criteria

The test cases executed on the E-Appointment System will pass if they meet the specific requirements mentioned. A test case is said to fail, if the desired functionality is not satisfied by the system.

## Test cases and test results

GIT link

## #Bugs detected and how tracking of bugs was done to completion

GIT link

## Coverage of requirements and design with tests (%)

RTM GIT link is given above

## Regression tests

Regression testing is performed using the selenium test suites.

GIT link of Selenium test suites is given above.

CHALLANGES

# Challenges

|  |  |
| --- | --- |
| **CHALLENGES** | **MITIGATION** |
| The project team took some unplanned leaves. | Had to put more effort and speed up the development for 1 sprint. |
| Faced challenges in installing the required software and server for development and testing in teammates machines. | By meeting online and on campus and spent extensive hours. |
| Web app development and Selenium automation testing was new, so had to spend more time to learn and do | By spending extra hours in learning and doing. |