Software Engineering

Civil Work Organization Report

Team 34

Team Members

Nardin Philip Boles
20201448030

Yara Hassan Mohsen
20201447189

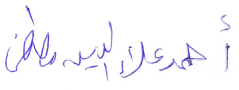
Mennatullah Magdy Mahmoud
20201378920

Table of Contents

Customer Proof

The Customer is a Civil Work Organization.
For Example: FCDS Alex ICPC Community.
The President of the community is Ahmed Alaa.
Phone Number: 01093738141
E-mail: Ahmedamahran5501@gmail.com

Signature

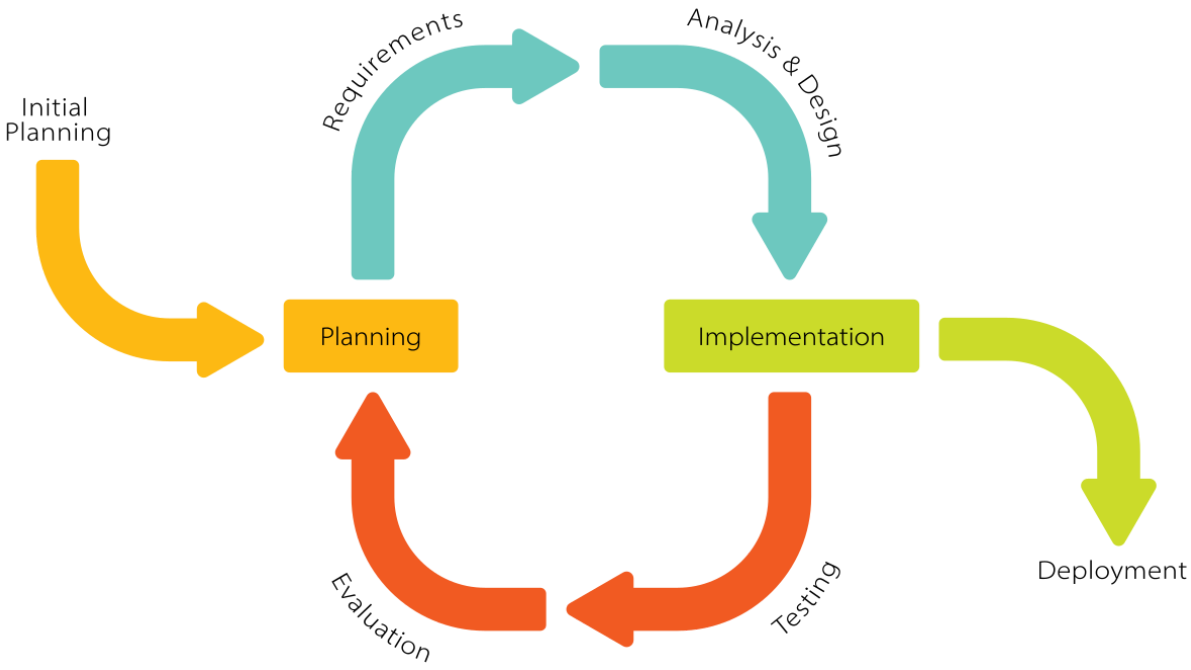


Project Idea

To create a software application that manages any civil work organization, controls its data and shows its services.

Software Model

Incremental Model



The software requirements are clearly defined and understood.
Hence, It can be arranged and prioritized by the developing team as well as the customer.

Each version of the software adds more functionality to the software.

Requirements

Functional Requirements

1) Registration

- Every participant/user is required to Register by his real name, e-mail address, phone number, his/her committee (If participant), and finally set the account's username, password, and confirms password.
- An Automated Verification for accounts' creation.
- An Automated function sends a unique code to the staff members via emails to be able to access the 'Staff Members' Page.

2) Log in

- Insertion of username and password by participants and end-users.
- In case of forgetting the account's password, an automated 'Forgot Password' Function sends a code to the account's owner to set a new password.

3) Profile Set-up and Editing

- Every participant/user can Set-up or edit his/her profile by changing profile picture, name, bio, and committee name (If participant).

4) Request

- Participants (Members) can request to join their committees' channels through the 'Staff Members' Page.

5) Contact Us

- 'Contact Us' Function receives questions from end-users.

6) Submit Tasks

- Staff Members can submit their tasks.

7) Meetings

- Staff Members can join the meetings started by the Board Members.

8) Messages

- Messaging between staff members and the board of the organization.

Requirements

Functional Requirements

9) Log Out

- The user logs out from his account.

10) Admin Panel

- The Head of each committee will be able to approve the join request of the members of that specific committee only, and decline/ ignore the join request of members of another committees (if present).
- General Announcements of the organization are created by the President or Vice-President of the organization, where organization's activities, events, ranking, and any urgent announcements are shared.
- The Board of the organization can reply to the questions of the end users sent through 'Contact Us' Function.
- Board members can create tasks for the Staff members.
- Board members will access the submitted tasks for evaluation and creating reports.
- The created reports are posted by the Board Members where each report will be posted on the member's profile.
- Specific Committees' announcements are created by the Head of each committee.
- Meetings are started by the board and the staff members can join.

Requirements

Non-Functional Requirements

1) Profiles

- Profile picture, name, username, e-mail address, and committee name of participants are displayed on the profile of the user.
- A ‘Show Report’ icon is displayed on the profile of the user to find his report (It is only visible to the profile’s owner).
- An ‘Edit your Profile’ icon is displayed to let the user edit his profile information and change his picture.
- A ‘Log Out’ icon is displayed.

2) Home Page

- General announcements of the organization, ‘About us’ , and ‘Contact us’ are displayed.
- 'About us' text will be given to the Front-end developer by the President.

3) Staff Members Page

- General meetings and events announcements are displayed.
- Icons of Committees Channels’ names are shown in order to be accessed by staff members (A staff member of a certain committee can only access the channel of their committee).
- Committees Channels include committee’s announcements, the tasks created by the head, and a ‘Submit Task’ button.
- The ‘Admin Panel’ can only be accessed by the Board members.
- The panel includes ‘Approve Join Request’, ‘Delete Join Request’, ’Create Announcement’, ‘Reply to Received Messages’, ‘Create Task’, ‘Evaluate Task and Create Report’, ‘Post Report’, and ‘Start Meeting’ icons.

Note:

There is a link between our software and the database of the organization in order to keep track of the staff members to guarantee a secure access behavior.

Tasks Management

- The team members will describe the Outline for each of the initial version, intermediate versions, and final version according to the sorted requirements by priority of the customer and developers(**T1**).
- Each member will start the tasks they are responsible for once they are done with the task they are working on.
- Versions will be worked on and done in parallel.

1) The Initial Version

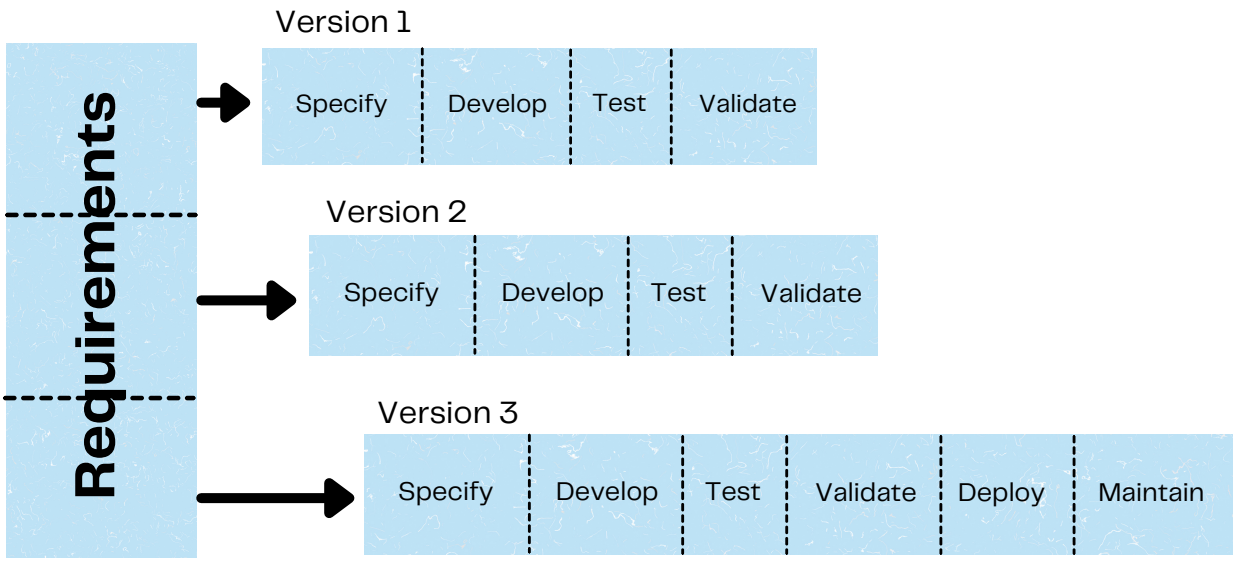
- Member one will be responsible for putting the Specifications for the initial version(**T2**).
- Member two will be responsible for Development(**T3**).
- Member three will be responsible for Testing and Validation(**T4**).

2) The Intermediate Version(s)

- Member one will be responsible for putting the Specifications for the second version (**T5**).
- Member two will be responsible for the development of the version(**T6**).
- Member three will be responsible for testing and validation(**T7**).

3) The Final Version

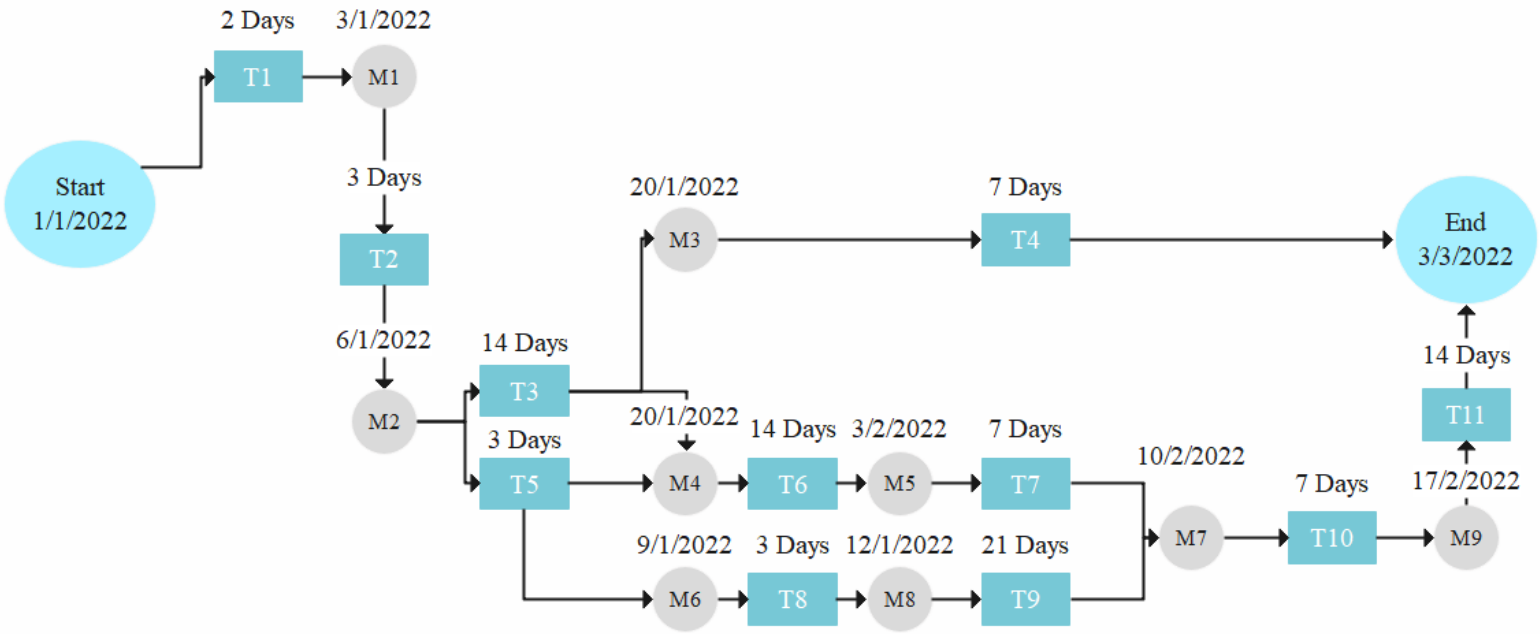
- Member one will be responsible for putting the Specifications for the final version, taking into consideration the customer's feedback, and the lessons learnt from the previous versions(**T8**).
- Member two will be responsible for the development of the version(**T9**).
- Member three will be responsible for testing and validation(**T10**).
- The whole team will be responsible for the deployment of the software, and any required maintenance(**T11**).



Time

Tasks	Duration	Dependencies
T1	2 Days	–
T2	3 Days	T1 (M1)
T3	14 Days	T2 (M2)
T4	7 Days	T3 (M3)
T5	3 Days	T2 (M2)
T6	14 Days	T3,T5 (M4)
T7	7 Days	T3,T6 (M5)
T8	3 Days	T5 (M6)
T9	21 Days	T8 (M8)
T10	7 Days	T7,T9 (M7)
T11	14 Days	T10 (M9)

Time Diagram



Cost

Basic COCOMO model for Semi-detached system

Software Projects	a	b	c	d
Semi-detached	**3.0**	**1.12**	**2.5**	**0.35**

Total Cost	
Lines Of Code	**5 KLOC**
Salary	**LE 500**
Effort= a(KLOC)^b = 3.0 (5) ^1.12	**18,196 PM**
Schedule (Time)= c (Effort)^d= 2.5 (18.196) ^0.35	**6.898 Months**
Team members (Person Required)= Effort/Time= 18.196/ 6.898= 2.64	**3 Persons**
Productivity= Lines of Code (LOC)/Time= 5000/6.898	**724.848 LOC/Month**
Total Cost= Person*Time*Salary= 3x6.898x500	**LE 10,347**

Risks

1) Shortage of employees.

2) Insufficient tools.

3) It is hard to predict the time for the project because of using the incremental model.

4) Solving a problem in one unit requires correction in all units and consumes a lot of time.

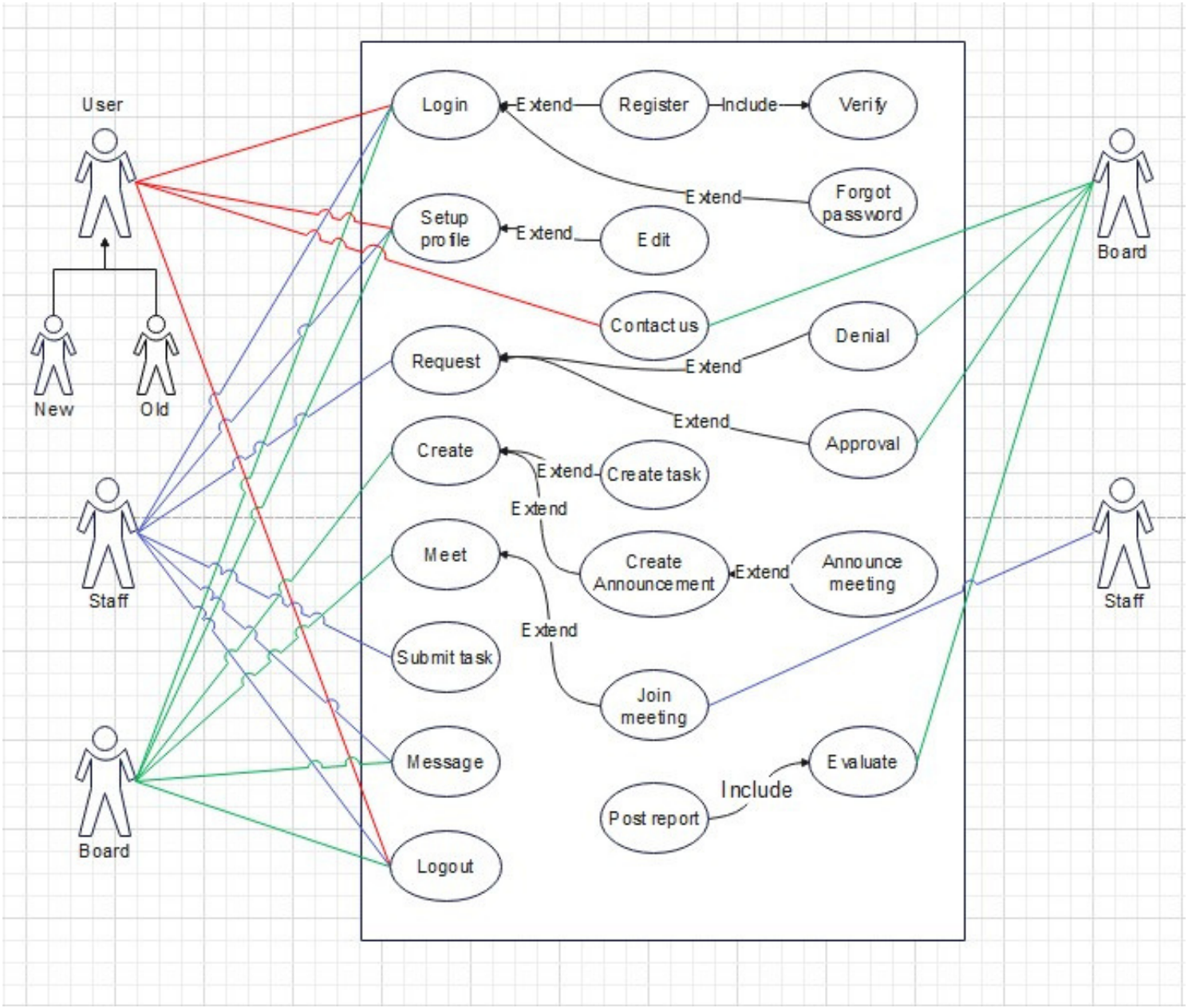
5) Civil work organizations do not have a budget for websites.

6) Getting feedback from a customer might be difficult in the first versions because the interface will not be complete.

7) Inexperienced Staff.

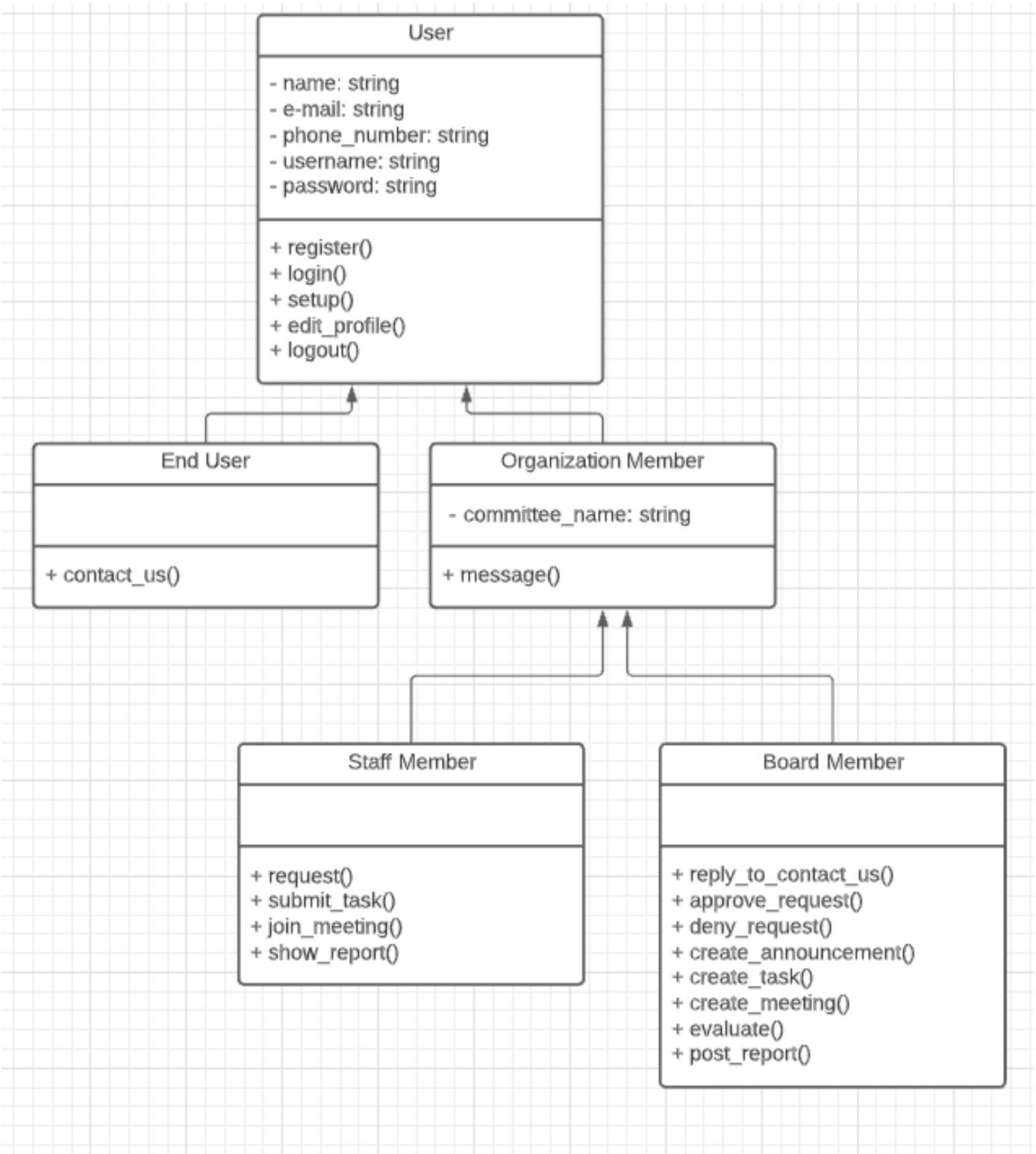
UML

Use Case Diagram



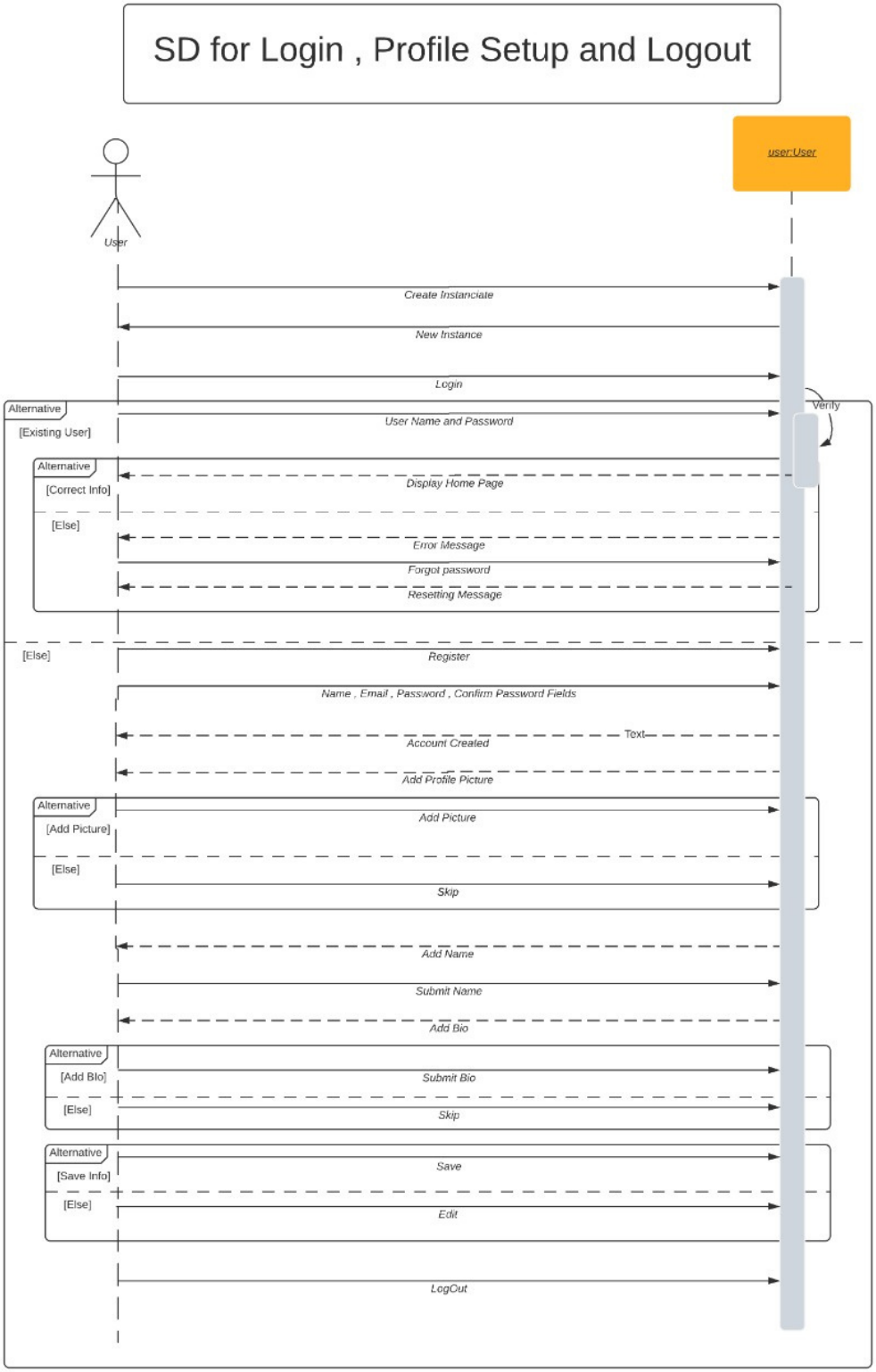
UML

Class Diagram



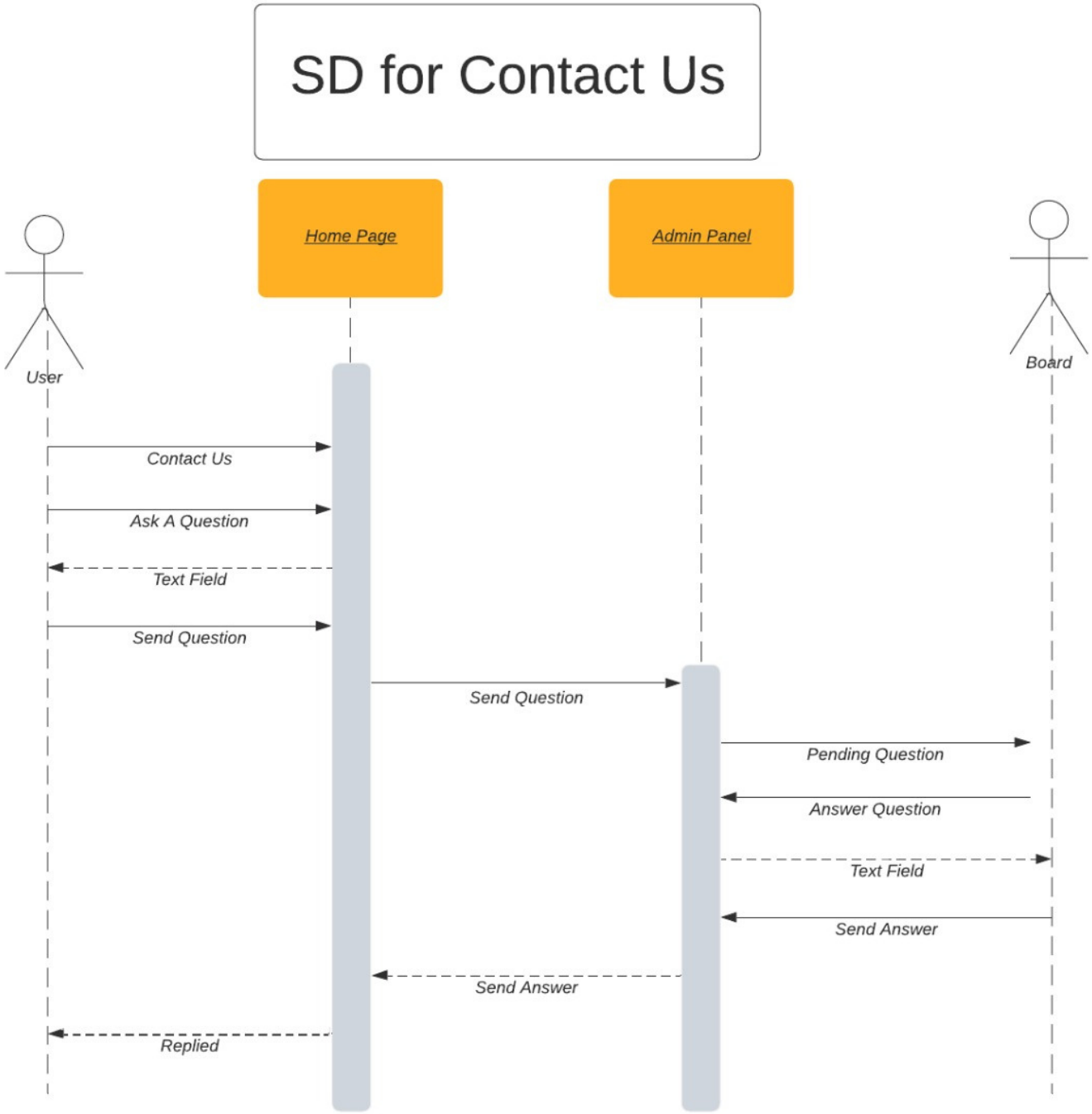
UML

Sequence Diagram



UML

Sequence Diagram

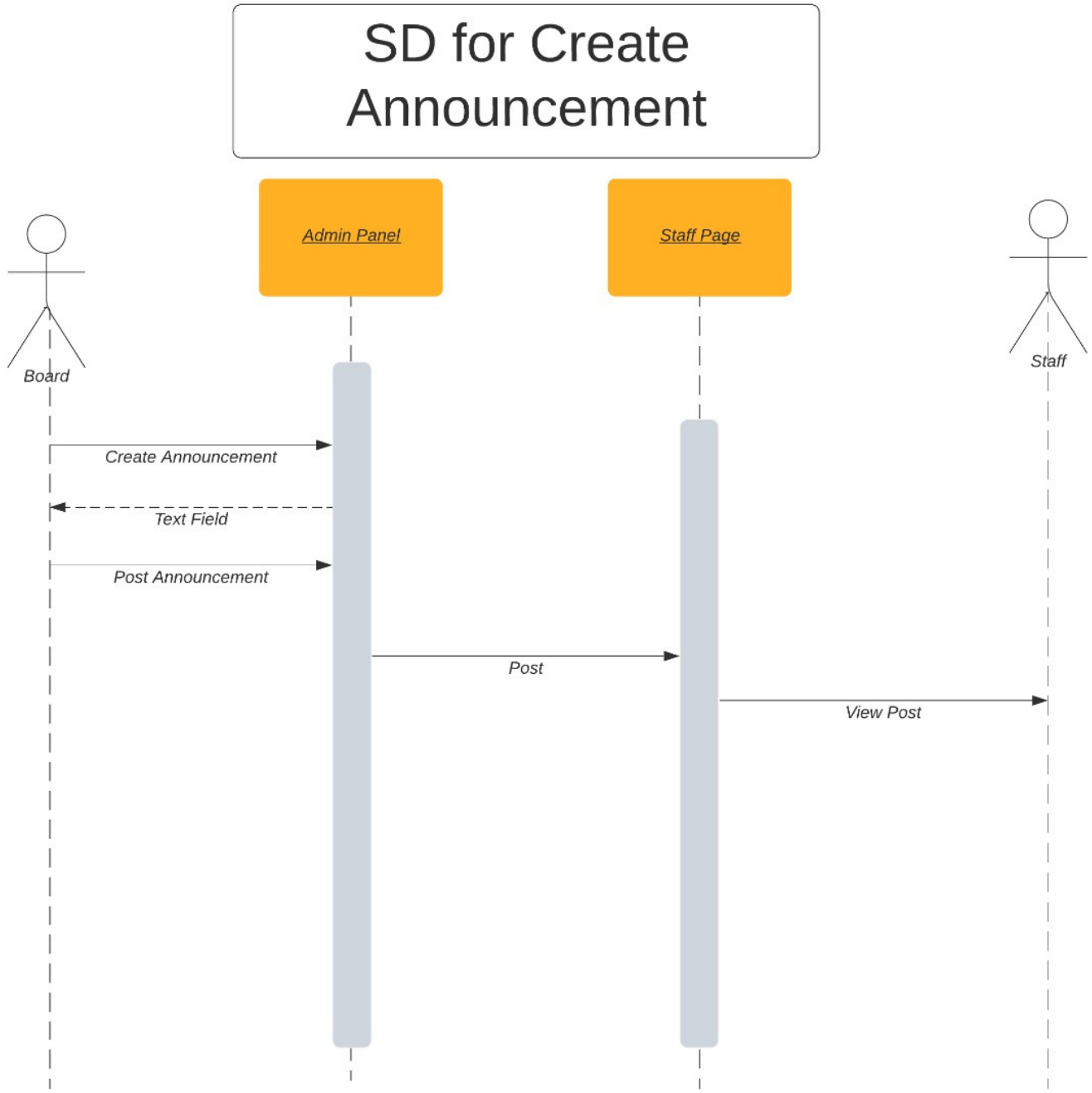


UML

Sequence Diagram

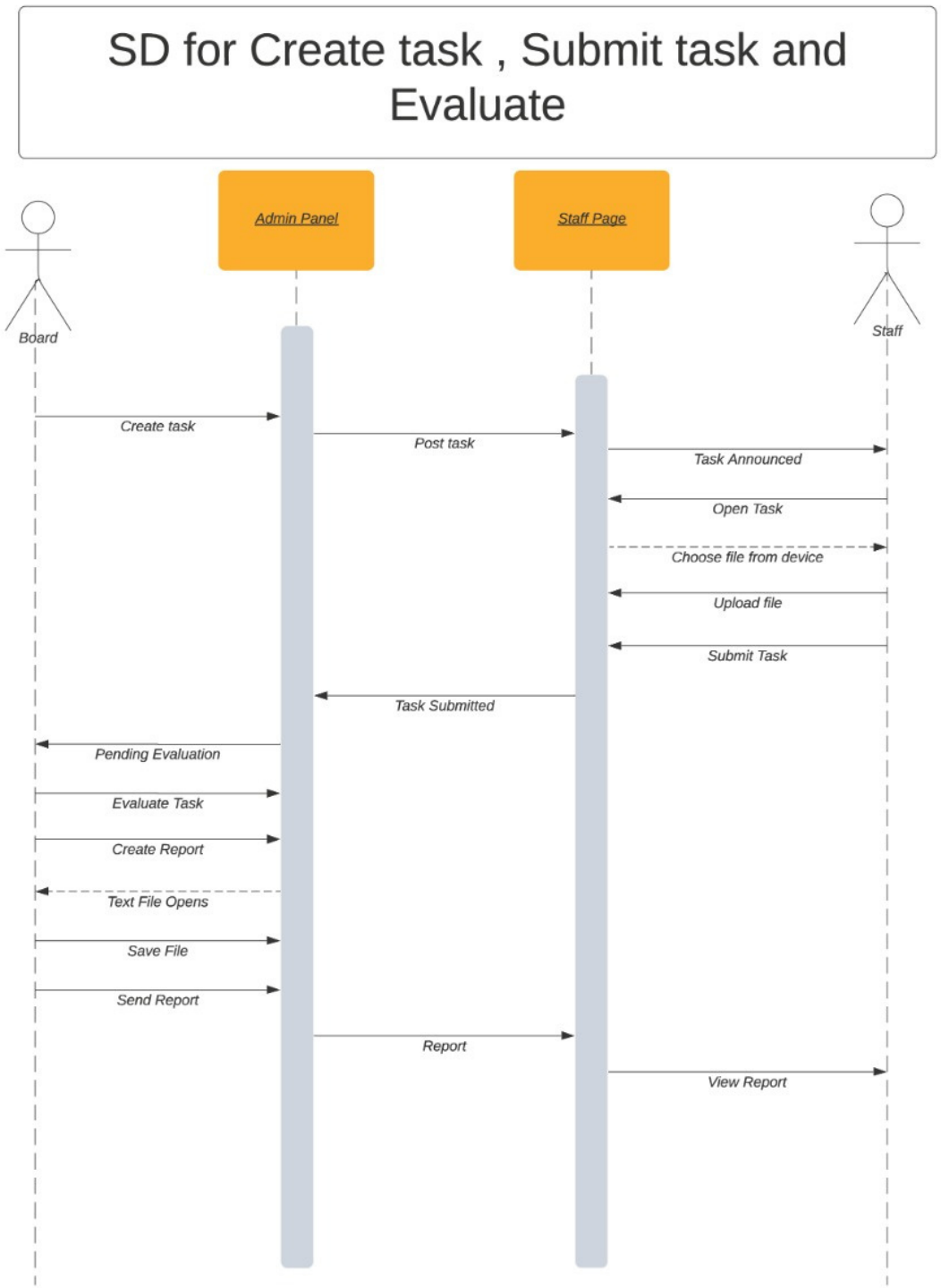
UML

Sequence Diagram



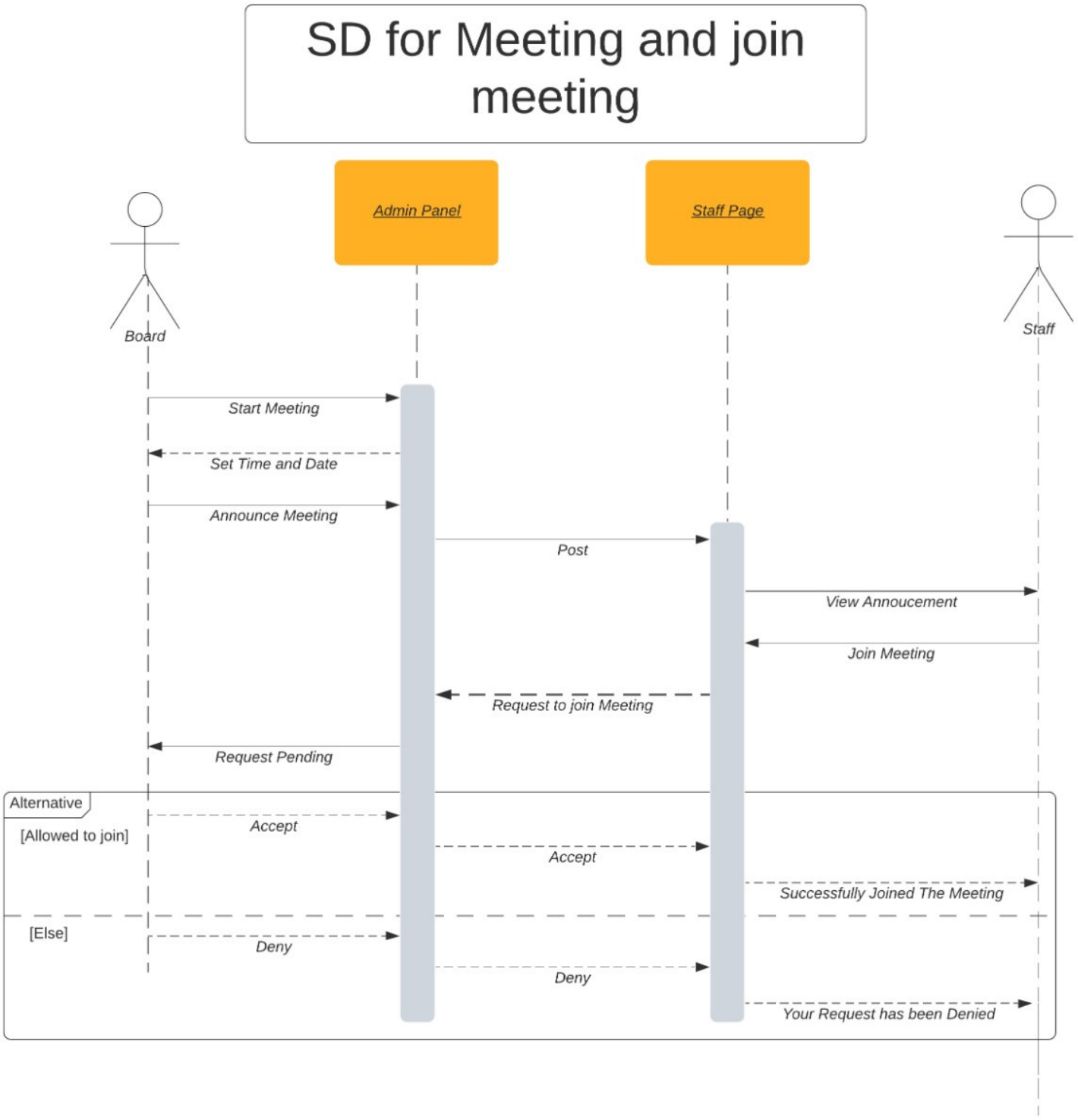
UML

Sequence Diagram



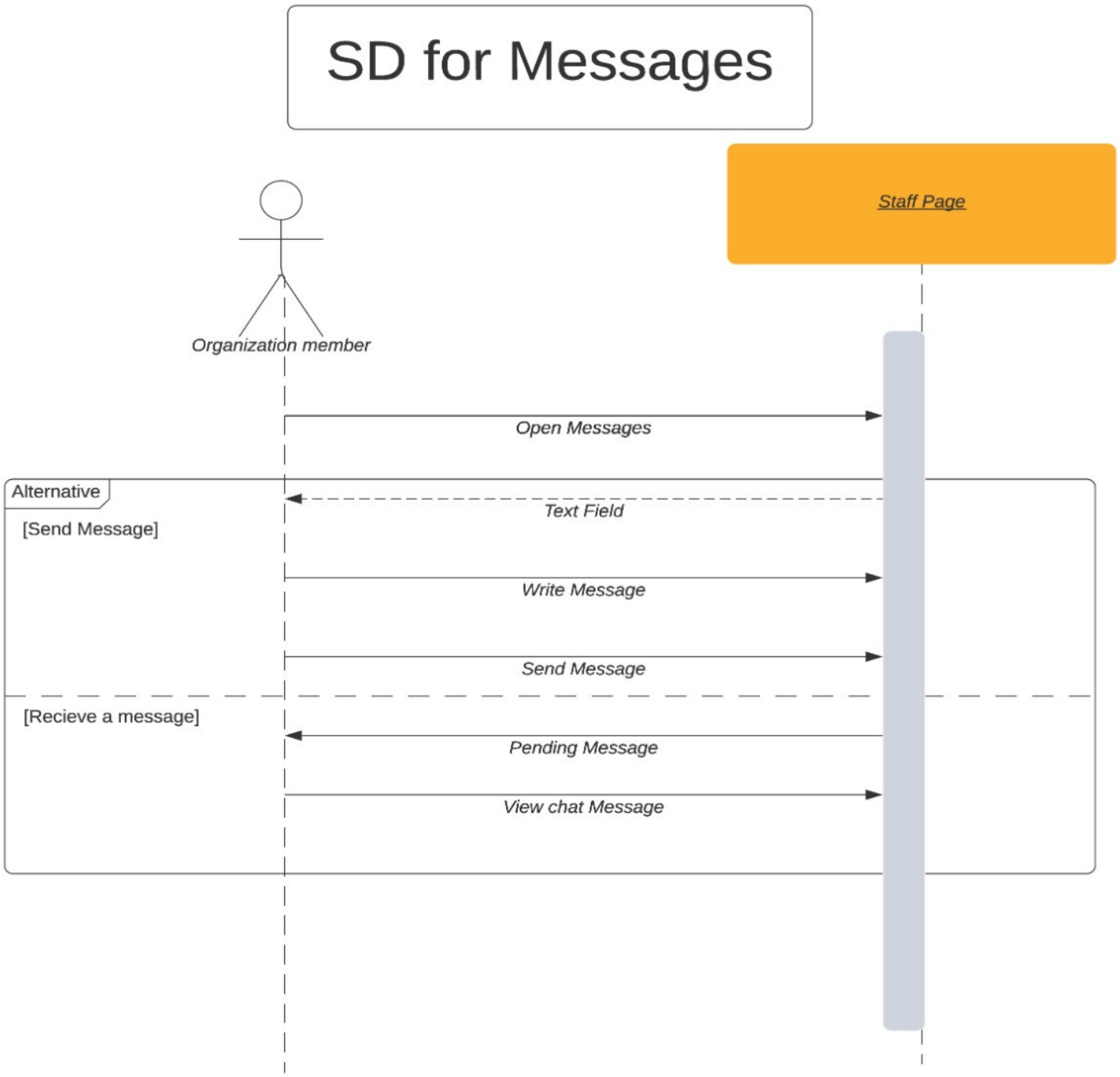
UML

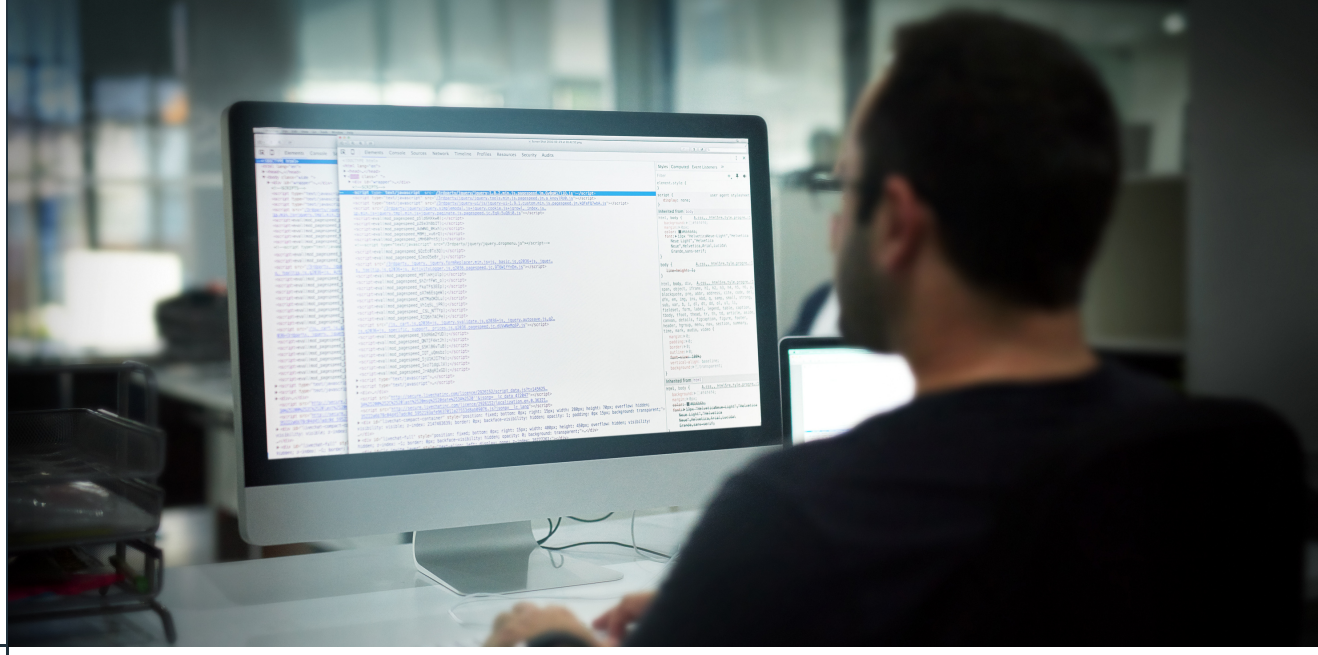
Sequence Diagram



UML

Sequence Diagram





Test Cases

Examples for some scenarios

1) A User tried to register while trying to enter the required data for account creation it was found out that he entered the wrong formatted text (Input Validation Test).

2) A User is not able to log in with a Valid Username and invalid Password (Input Validation Test).

3) While trying to set up a user profile, the user cannot add his profile picture (Unit Test).

4) A Staff Member can request to join his committee's channel, and the Board Member can accept the join request (Integration Test).

5)A Staff Member in a certain committee requested to join another committee's channel, and the Board can deny his request (Integration Test).

6)An end-user can send his question through the 'Contact Us' Function, and the Board can receive his question (Integration Test).

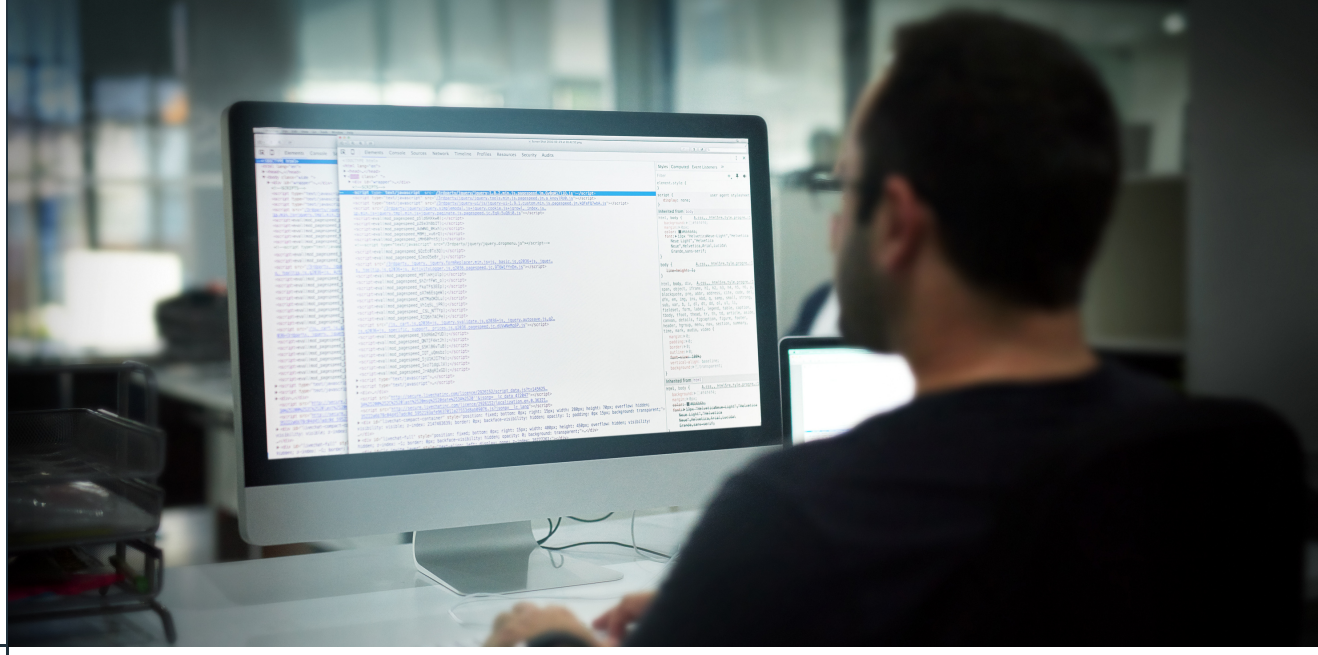
7)A Board Member can respond to the received question, but it was found out that the end-user cannot receive the answer (Integration Test).

8) A Board Member can start a meeting through the admin panel, and a Staff Member can join that meeting successfully (Integration Test).

9) A Board Member was able to announce a general announcement successfully, but that announcement wasn't displayed on the 'Home Page' of the website (Integration Test).

10) All Organization Members can message each other through the 'Messages' Function (Unit Test).

11) A User can log out from his account, and when trying to log into his account again, he must enter his username and password (Unit Test).



Test Case for Task Submission

- In the second version, the board member will create the task and this will be shown in the announcements section. The staff members will be able to submit that task.
- In the third version, the board member will access this task submitted and start evaluating it then they will post a report.
- We will use a back-to-back test after each of the second and the third versions to make sure that each function and the access of the members are working.
- After the third version is done. We might face some problems. So instead of creating version four to solve them we will update version three and use the regression test to check that everything is working as we want.

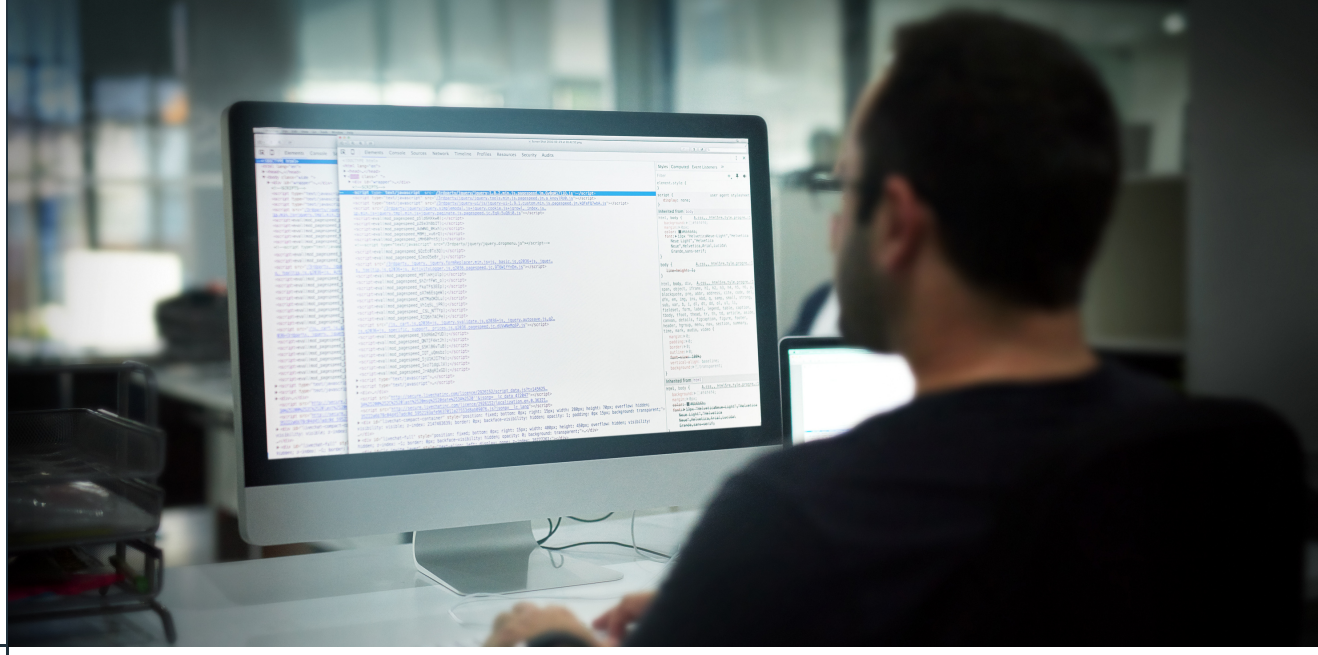
Examples for some scenarios

1) A board member created a task but it wasn't announced for the staff members. This problem will be tested by a (Back-to-Back test) after version 2 and then solved in version 3.

2) A staff member couldn't submit the task. This problem will be tested using (Back-to-Back Test) after version 2 and then solved in version 3.

3) A board member couldn't access the submitted task. We will use (Regression Test) on version 3.

4) A board member couldn't post the report. We will use (Regression Test) on version 3.



Test Techniques

- **Input Validation Test**

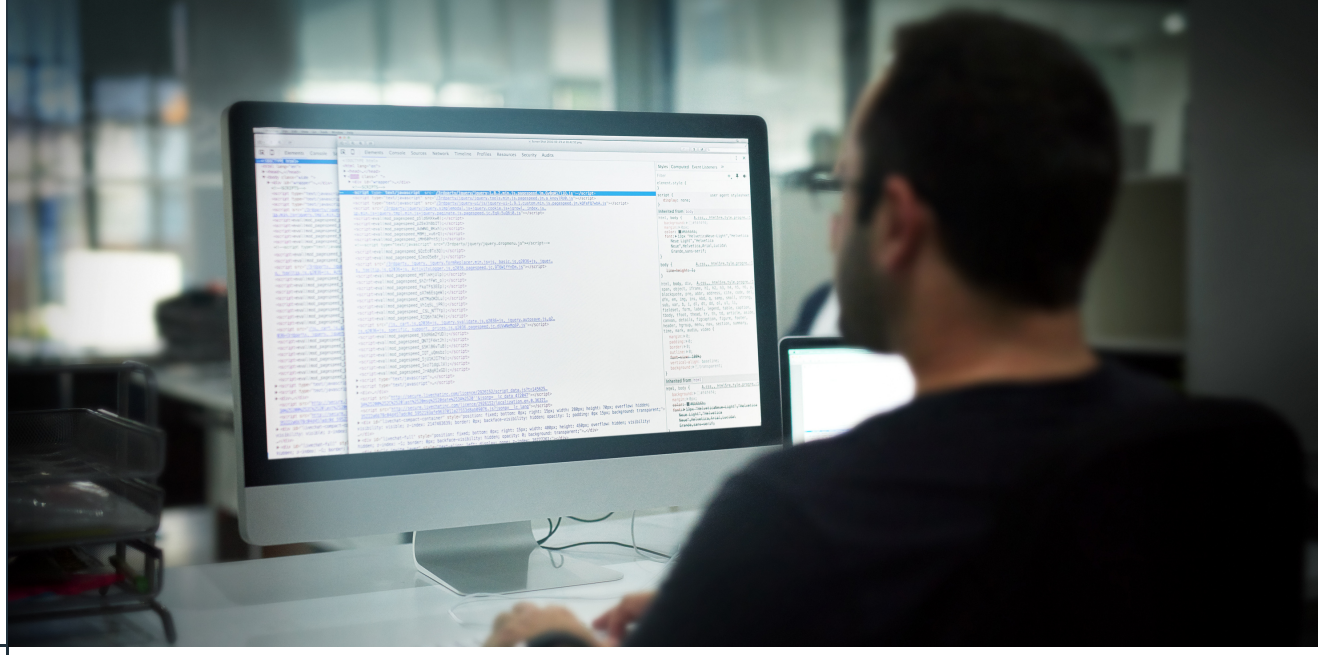
Validation testing is confirmation that a product meets its intended use and the needs of its users. Following successful verification, development teams should employ validation testing with the initial production product and in the actual (or simulated) use environment.

- **Back-to-Back Test**

Back-to-back testing is a type of testing which is conducted if there are two or more variants of components with similar functionality. Back-to-back testing aims to compare the results to check if there are any divergences in the work.

- **Regression Test**

Test cases are re-executed to check the previous functionality of the application is working fine, and the new changes have not produced any bugs. Regression testing can be performed on a new build when there is a significant change in the original functionality.



Test Techniques

- **Unit Test**

Unit testing is a type of testing in which individual units or functions of software testing. Its primary purpose is to test each unit or function. A unit is the smallest testable part of an application. It mainly has one or a few inputs and produces a single output.

- **Integration Test**

Integration Testing focuses on checking data communication amongst software modules which are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated.

References

1) Lecture Materials

2) Lucid chart

- Types of UML Diagrams: https://www.lucidchart.com/blog/types-of-UML-diagrams
- UML Use Case Diagram: https://youtu.be/zid-MVo7M-E
- UML Class Diagram: https://youtu.be/UI6lqHOVHic
- UML Sequence Diagram: https://youtu.be/pCK6prSq8aw
- Tools used: https://www.lucidchart.com/pages/examples/uml_diagram_tool

3) GeeksforGeeks

- Incremental Process Model: https://www.geeksforgeeks.org/software-engineering-incremental-process-model/
- Introduction about UML: https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/
- https://www.geeksforgeeks.org/software-engineering-cocomo-model/