Software Requirements Specification

for

College Social Media System

Version 1.0

Prepared by

Group Name: G02

Amit Kumar	B190343CS	amit_b190343cs@nitc.ac.in
K Gopal Choudhary	B190501CS	gopal_b190501c@nitc.ac.in
Navnit Anand	B190404CS	navnit_b190404cs@nitc.ac.in
Mihir Madhav Gokhale	B190370CS	mihir_b190370cs@nitc.ac.in
Nishant Kumar Bhardwaj	B190427CS	nishant_b190427cs@nitc.ac.in

Instructor : Abdul Nazeer
Course : DataBase Management System
Date : 27-10-2021

Index

1 Introduction

College Social Media System

1.1 Document Purpose

The purpose of this Software Requirement Specification (SRS) document is to give an overview of the functional and non-functional requirements of the College Social Media Management System.

Based on this Document Students and the Clubs can get to know what services our application shall provide.

1.2 Product Scope

The aim of this project is to develop a College Social Media Management system that would enable all the students across the college campus to connect with each other. This Product can help Juniors to connect with the Seniors, and Clubs to have some common platform to make announcements regarding their events and run a smooth hustle free college life for the students.

Features:

- Provides Students and Clubs to have their own Social Media Profile
- Provides Students and Clubs to Post anything and Comment feature for the same
- Students/Clubs can discuss in threads or create their own too.
- Users can Upvote or Downvote a Post
- Users can know about all the upcoming events
- Users can DM each other
- Users can report inappropriate posts/behavior.
- Allows the administrators to monitor the legitimacy of the posts, and suspend a user if necessary.

1.3 Intended Audience and Document Overview

Intended Audience: Professor ,Client and Developers
Overview : The rest of the document contains an overall description of the product including its functionalities , Design , the Use Case Model , The functional and nonfunctional requirements for the product.
The sequence for reading the document for better understanding would be to start with the Overall Description of the Product which would result in understanding the Product Features and its dependencies (more important for Client and Professors) and then go through the hardware and software requirements to get an all overview of the required things to make sure the application runs smooth (more important for Developers and Professors) and then checking the Use Case Model and its analysis (Important for Developers) and atlast going through the other non functional requirements

1.4 Definitions, Acronyms and Abbreviations

Acronyms	Abbreviations
DM	Direct Message
IEEE	Institute of Electrical and Electronics Engineers
OS	Operating System
SQL	Structured Query Language

1.5 Document Conventions

This document follows the IEEE formatting requirements. Arial font size 11 has been used throughout the document for text. Use of italics for comments. Use of bold for headings. Document text is single spaced and 1” margins are maintained in this document.

1.6 References and Acknowledgments

- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
- Lucidchart to create the use case diagram

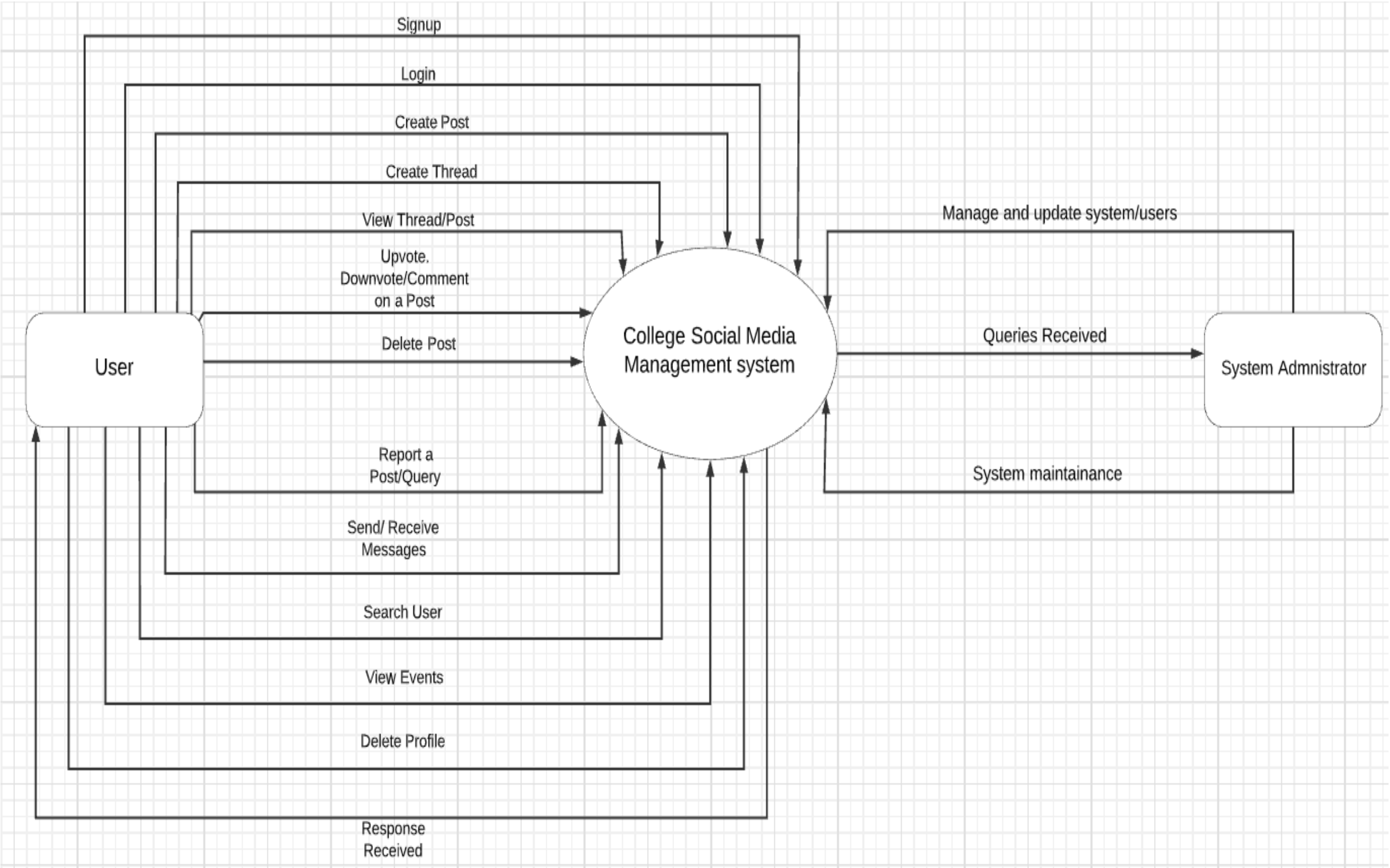
We gratefully acknowledge the support from the Database Management system course teachers for the lectures and for the opportunity to collaborate with fellow peers on this project.

2 Overall Description

2.1 Product Overview

This application will serve as a platform where students and clubs of NITC can interact, discuss and conveniently share information regarding events or any other things going in and around the campus. The user (student/faculty) can view their profile on their dashboard, view relevant posts, create posts, join threads, ask as well as answer queries, or DM other users.
The system administrators manage and update the system. They maintain the authenticity of the queries and have the authority to suspend a user.

DATA FLOW DIAGRAM



2.2 Product Functionality

College Social Media System has the following main functions:

- A user can have his own account similar to the other social media
- A user will have a feature to post
- Other users can upvote/Downvote/Comment on the post.
- Users can Report Inappropriate posts/queries.
- Clubs can announce/Share any ongoing/upcoming activities through their pages
- The administrator can manage all the posts on the platform.
- Users can create/delete the threads and discuss doubts among themselves.
- Users can use the DM feature for messaging
- Users can maintain their own profile on the application.

2.3 Design and Implementation Constraints

- **Development Environment**: The development environment used for the application is Windows OS, Visual Studio code editor.

- **Database**: The application relies on the database which contains data of all the users. Any changes made in the details of users in the database should reflect in the application as well. However, the application should not be allowed to make any changes to the users' data directly.

- **Security**: Since the application makes use of vital information about the institute, all communications should be encrypted using Transport Layer Security(TLS).

- **Design conventions and standards**: This application development process must follow the principles specified by the Software Development Lifecycle (SDLC).

2.4 Assumptions and Dependencies

- Since we will develop our application with Agile methods, user interfaces and functionalities may change in the future according to the need of the hour.
- Due to the huge rush, there may be a server-down situation that can be restored after some cautious inspection.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Users will interact using a webapp with the system that provides a simple and intuitive user interface. Based on the User role Functionalities will be extended.

3.1.2 Hardware Interfaces

Processor : i3 or greater
Ram : 512MB
Memory : Depends upon the Database size

3.1.3 Software Interfaces

- The Front-end connects with Backend through functional API's.
- Backend request data to Database using sequelize.
- We use mailgun API to send mail.

Tech Stack:
FrontEnd: ReactJS,Bootstrap,Tailwind
Backend: NodeJS, ExpressJS, MySQL

3.2 Functional Requirements

F1: The system shall have a Signup page for the users
F2: The system shall do Email Verification which will be done while registration for users
F3: The system shall have a Login page for the users
F4: Login Credentials will be verified and check if its a valid entry or not
F5: The system shall have Forgot password feature for Users
F6: The system shall have a homepage for Users
F7: The system shall have a Profile page for Users
F8: The system shall allow Users to view Posts
F9: The system shall have a create and share Post feature for Users
F10: The system shall have a Comment feature for Users
F11: The system shall allow users to upvote or downvote on a Post
F12: The system shall allow User to report any inappropriate post.
F13: The system shall restrict the visibility of the post only to the same year's Users
F14: The system shall allow the users to see the post creator.
F15: The system shall allow users to search for other User's Profile using username/email ID

F16: The system shall allow users to view upcoming events (Event Section)

F17: The system shall allow users to add events to the Events Section

F18: The system shall allow Users to delete Posts

F19: The system shall allow Users to delete Account

F20: The system shall allow users to create threads for discussion

F21: The system shall allow users to answer the question in threads

F22: The system shall allow Users to close the thread

F23: The system shall close the thread if inactive for 7 days

F24: The system shall allow users to send/receive messages

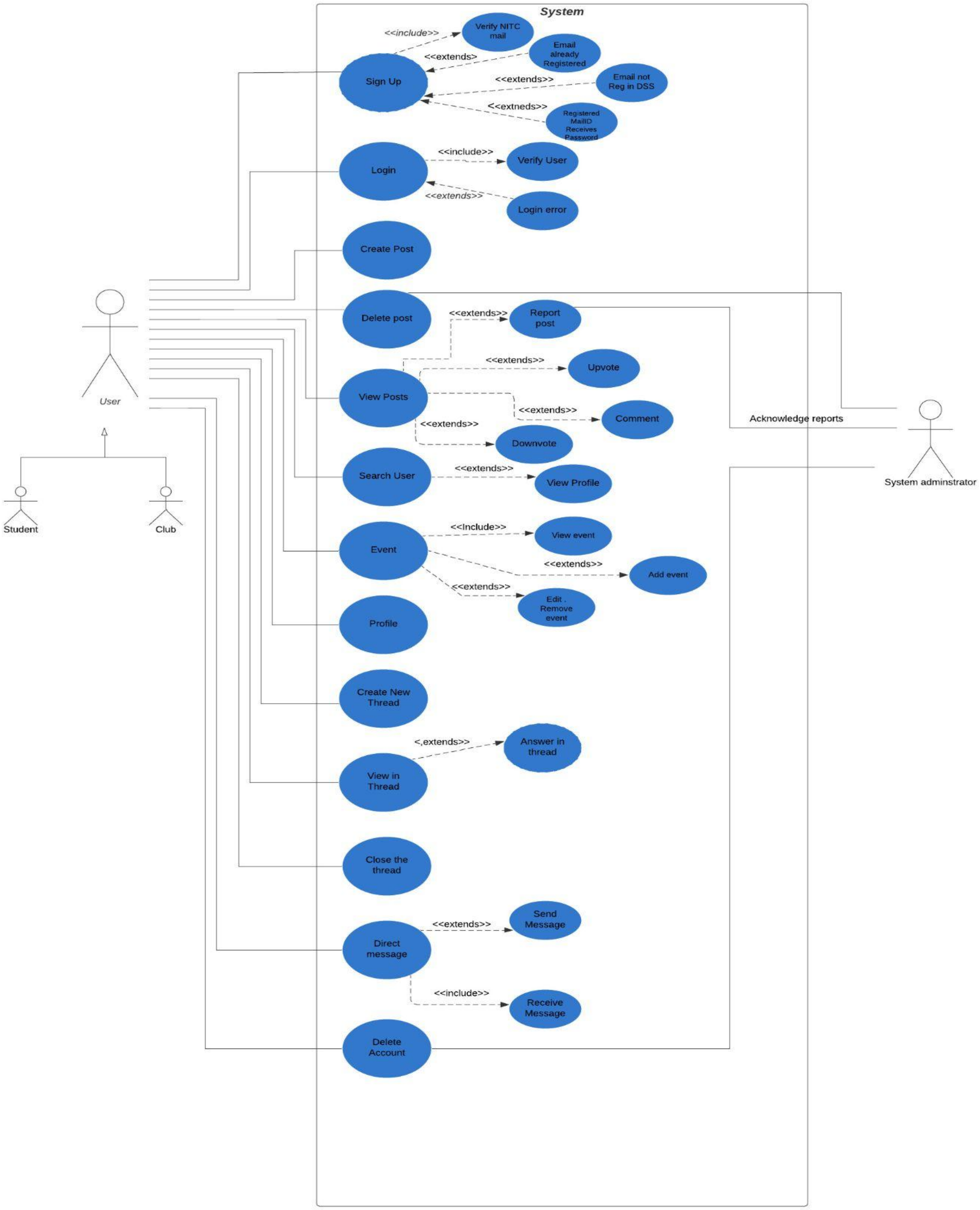
F25: The system shall allow users to view the chat

F26: The system administrator will be able to see and answer user queries

F27: The system administrator will be able to take actions such as delete a post/query or suspend an user based

F28: The system administrator will be able to release system updates

3.3 Use Case Model



3.2.1 Use Case #1 (Sign Up)- U1

Author – K Gopal Choudhary

Purpose - Allows a User to register in the application

Requirements Traceability – F1, F2

Priority - High

Preconditions - The User must not have registered before and should have a valid College mail ID

Postconditions - The user will be registered successfully and all the details entered will be stored successfully in the Database

Actors – User (Students and Clubs)

Extends –

- Email Already registered
- Email ID not in DSS

Flow of Events

1. Basic Flow -
   - The user opens the signup Page and enters the credentials required
   - The system checks whether it is a Valid College Email ID or not
   - A Randomly generated password is sent to the registered email ID
2. Alternative Flow -
   - If the Email ID entered is invalid the system will ask to re-enter the email ID
   - If trying to register through another email ID then User won't be able to get access to the password sent to the Registered email ID
3. Exceptions -
   - EmailID already registered
   - Email ID not in DSS

Includes - Email ID verification

3.2.2 Use Case #2 (Login) - U2

Author – Nishant Kumar Bhardwaj

Purpose - Allows a User to login in application after registering

Requirements Traceability – F3, F4, F5

Priority - High

Preconditions - The user must have registered before and should have the generated password

Post conditions - The User is directed to his/her homepage

Actors – Users, System Administrator

Extends – Login error

Flow of Events

1. Basic Flow -
    - User login with the given credentials.
    - The Admin verifies the details from the database.
    - User is directed to his/her homepage
2. Alternative Flow -
    - The system asks the user to again enter the correct credentials.
3. Exceptions -
    - Login error due to wrong entry of credentials.

Includes - Verification of user details by admin

3.2.3 Use Case #3 (Create Post)- U3

Author – K Gopal Choudhary

Purpose - Allows a User to create a post in the application

Requirements Traceability –F9

Priority - Medium

Preconditions - The user must be logged in

Postconditions - The User has created and shared the post successfully

Actors – User (Students and Clubs)

Extends – None

Flow of Events

1. Basic Flow -
    - The user opens the create post page
    - The user adds image/text to it and shares it on the profile
2. Alternative Flow -
    - User chooses to not share the created post
3. Exceptions - None

Includes - None

3.2.4 Use Case #4 (Delete Post)- U4

Author – Amit Kumar

Purpose - Allows a User to delete a post in the application

Requirements Traceability –F18, F26

Priority - Medium

Preconditions -

- For User: The user must be logged in and should have a post to delete
- For System Administrator: The post should have been reported by users/ or has violated any rules

Postconditions - The post has been deleted successfully

Actors – User (Students and Clubs), System Administrator

Extends – None

Flow of Events

1. Basic Flow -
    - The user views the post and chooses to delete the post
    - System Administrator deletes a post after getting many flags on any post after verifying
2. Alternative Flow - None
3. Exceptions - None

Includes - None

3.2.5 Use Case #5 (Create Thread)- U5

Author – Navnit Anand

Purpose - Allows a User to create a discussion thread in the application

Requirements Traceability –F20

Priority - Medium

Preconditions - The user must be logged in

Postconditions - The User has created a thread successfully

Actors – User (Students and Clubs)

Extends – None

Flow of Events

1. Basic Flow -
   - The user opens the create thread feature
   - The user enters the required details like title, content and initializes the thread on the application
2. Alternative Flow -
   - The user chooses to not share the created thread
3. Exceptions - None

Includes - None

3.2.6 Use Case #6 (Close Thread)- U6

Author – K Gopal Choudhary

Purpose - Allows a User to close the thread in the application or the server to close the thread

Requirements Traceability – F22, F23

Priority - low

Preconditions - The User must be logged in

Postconditions - The Thread is closed

Actors – Users, System Administrator

Extends – None

Flow of Events

1. Basic Flow -
    - The user goes to the thread he wants to delete
    - User Choose to delete the thread
    - The thread is deleted
2. Alternative Flow -
    - The server deletes the thread if it is found inactive for more than 7 days
3. Exceptions - None

Includes - None

3.2.7 Use Case #7 (Search User)- U7

Author – Navnit Anand

Purpose - Allows a User to search a profile of other users.

Requirements Traceability – F15

Priority - Medium

Preconditions - The User must be logged in

Postconditions - View the profile

Actors – Users

Extends – None

Flow of Events

1. Basic Flow -
    - The user goes to the search bar
    - User types the email ID/Username of the user to be searched
    - User Profile is Displayed
2. Alternative Flow - No Such User exists.
3. Exceptions - Username/email ID entered wrong.

Includes - None

3.2.8 Use Case #8 (Events)- U8

Author – Nishant Kumar Bhardwaj

Purpose - Allows a User to view events and edit the events section

Requirements Traceability – F16, F17

Priority - Medium

Preconditions - The User must be logged in

Postconditions - The User can join the events or edit the events section.

Actors – Users

Extends – Add/Edit the events.

Flow of Events

1. Basic Flow -
    - Users can see the ongoing/upcoming events on Campus.
    - Users can add their own events.
2. Alternative Flow - None
3. Exceptions - None

Includes - Some events Related to clubs will be already present.

3.2.9 Use Case #9 (Direct Message)- U9

Author – Amit Kumar

Purpose - Allows a User to chat directly with other users

Requirements Traceability – F23, F24

Priority - Medium

Preconditions - The User must be logged in

Postconditions - The User will be able to send and receive messages

Actors – Users

Extends – Sending the message to other users.

Flow of Events

1. Basic Flow -
    - Users can send messages to other users
    - Users can receive messages from other users

- Chats will be stored in the database

2. Alternative Flow -
   - One of the User deleted Account , The chat gets deleted from the deleted User side
3. Exceptions -
   - One of the users deletes the account.

Includes - Receiving messages from other users.

3.2.10 Use Case #10 (Delete Account)- U10

Author – Nishant Kumar Bhardwaj

Purpose - Allows a User to delete his/her account

Requirements Traceability – F19

Priority - Low

Preconditions - The User must be logged in

Post conditions - User account is deleted

Actors – Users, System Administrator

Extends – Login error

Flow of Events

1. Basic Flow -
   - User requests to delete the account
   - The Admin erases all the data of the user from the database.
   - The user account is deleted.
2. Alternative Flow -
   - Account gets reported and System administrator deleted the account
3. Exceptions -
   - Deleting the uncreated account.

Includes - None

4 Other Non-functional Requirements

4.1 Performance Requirements

Performance requirements for this product are:

- It should support various types of browser like Chrome , Edge, Firefox, Safari , Brave etc.,
- The Software should be able to support multiple users at same time.
- The database should be able to store large amounts of data.

4.2 Safety and Security Requirements.

Safety and security requirements are listed as below:

- The Data must be encrypted like text messages ,passwords and personal information such as phone numbers.
- There is a chance that the webapp may get crashed due to some OS failure , virus etc., so it is required to have a backup database always .

4.3 Software Quality Attributes

Reliability:

- As the program is made using different modules and error in one doesn't affect other
- It can tolerate the user making mistakes or using the software in unexpected ways
- The system prevents any unauthorized access and abuse as the SignUp details are verified by System Administrator

Portability:

- Easy to access using various browsers like Chrome , Edge, Firefox, Safari , Brave etc.,
- Easy to install in various OS (Windows, Linux, MacOS)

Flexibility:

- Users can Update details without any special administration rights.
- System Admin can Manage posts and make necessary action as required.
- Developers can update the Web application .

Appendix A – Data Dictionary

Database	A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a Database Management system.
SQL	SQL is a programming language developed for storing and retrieving large amounts of data. Learn more about its history, importance, and career prospects.
MySQL	It is an open source relational database management system. It is based on the structure query language (SQL).
Data Flow Diagram	It is a way of representing a flow of data through a process or a system.
Encryption	In cryptography, encryption is the process of encoding information. This process converts the original representation of the information, known as plaintext, into an alternative form known as ciphertext.
ReactJS	React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript.library for building user interfaces or UI components.
NodeJS	NodeJS is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Appendix B - Group Log

We had the following meets to complete our SRS Document:

- Meet on 18 October-2021 from 5:30 PM -7:30 PM:
  Started figuring out the basic outline for the project idea and started with Section 1: Discussing Document Purpose,features of the product .
- Meet on 19 October-2021 from 5:30 PM- 8:00 PM:
  Started Working on Section -2 ,discussed briefly about product functionalities and the requirements for it were discussed . Flowchart for the product was made, The Introduction and overall Description part was done.
- Meet on 20 October-2021 from 3:00 PM- 6:00 PM:
  Discussion Functional requirement section and then we worked on Use Case Model and each one of us was assigned to work on Use Cases.
- Meet on 22 October-2021 from 4:00 PM - 6:30 PM:
  Worked on the Non- functional Requirements part and checked each other's use cases and made few changes wherever required.