



Software Requirements Specification

for

NITC E-MAGAZINE

Version 1.0

Prepared by

Team Number: 4

Bhavana Ketavath
Abhinav P
Abhiram K P
Tom Saju

B190391CS
B190460CS
B190281CS
B191290CS

Course: CS3007D Object Oriented Systems

Date: 17/10/2021

1 Introduction	3
1.1 Document Purpose	3
1.2 Product Scope	3
1.3 Definitions, Acronyms and Abbreviations	4
1.5 Document Conventions	4
1.6 References and Acknowledgments	4
2 Overall Description	5
2.1 Product Overview	5
2.2 Product Functionality	6
3 Specific Requirements	7
3.1 Functional Requirements	7
3.2 Use Case Model	8
3.2.1 Use Case #1 (Sign up a user – U1)	9
3.2.2 Use Case #2 (Login a user - U2)	11
3.2.3 Use Case #3 (View Profile - U3)	13
3.2.4 Use Case #4 (Edit Profile - U4)	13
3.2.5 Use Case #5 (Comment on article - U5)	17
3.2.6 Use Case #6 (View Article - U6)	19
3.2.7 Use Case #7 (Submit Article - U7)	21
3.2.8 Use Case #8 (Database Administration - U8)	22
3.2.9 Use Case #9 (Server Upgradation and Maintenance - U9)	24
3.2.10 Use Case #10 (Article management - U10)	26
3.2.11 Use Case #11 (Edit Article - U11)	28
3.2.12 Use Case #12 (User Account management - U12)	29
3.2.13 Use Case #13 (Confirm Membership - U13)	32
3.2.14 Use Case #14 (Update Account Information - U14)	34
3.2.15 Use Case #15 (Review Moderator - U15)	37
Appendix A - Activity Log	

1 Introduction

1.1 Document Purpose

The purpose of this document is to provide a detailed description about the requirements for the NITC magazine. The application aims to provide services like creating user profiles and editing them, submitting their articles, other users can view and comment on all articles. The document contains a general description, functions, interface, performance requirements for the application, application usage scenarios and use-case diagrams.

1.2 Product Scope

The objective of NITC magazine application is to transform the laborious system of submitting articles manually to a hassle-free digitized version in the form of an application. Different users like students, faculties, guests, alumni and moderators have corresponding privileges.

A centralized database of all profiles and articles is created and only the admin has access to it. Students/faculties have their own profiles where they can submit articles, view and comment on them. All the Submitted article are put in a database where the moderator can review and edit those articles. Alumni have designated account, where he/she can create an account and view/comment on articles. Users can login as Guest to just view articles. The application seeks to save time and benefit every reader.

1.3 Definitions, Acronyms and Abbreviations

SNo.	Term/Abbreviation	Definition
1.	Admin	A person who manages all the database, maintains and upgrades the server and review the works by all moderators.
2.	Moderator	A person who reviews and edits all the articles, manages user accounts (confirms memberships and updates info).
3.	Registered member	The end users, who have registered and can submit/view/comment on an article.
4.	Alumni	Pass out students, who can view and comment on the articles.
5.	Guests	The users who don't have an account, but can only view the articles.

1.5 Document Conventions

This Document follows the IEE formatting requirements. Used Times New Roman font size 12, 14 throughout the document. Bold for sectional, sub sectional titles.

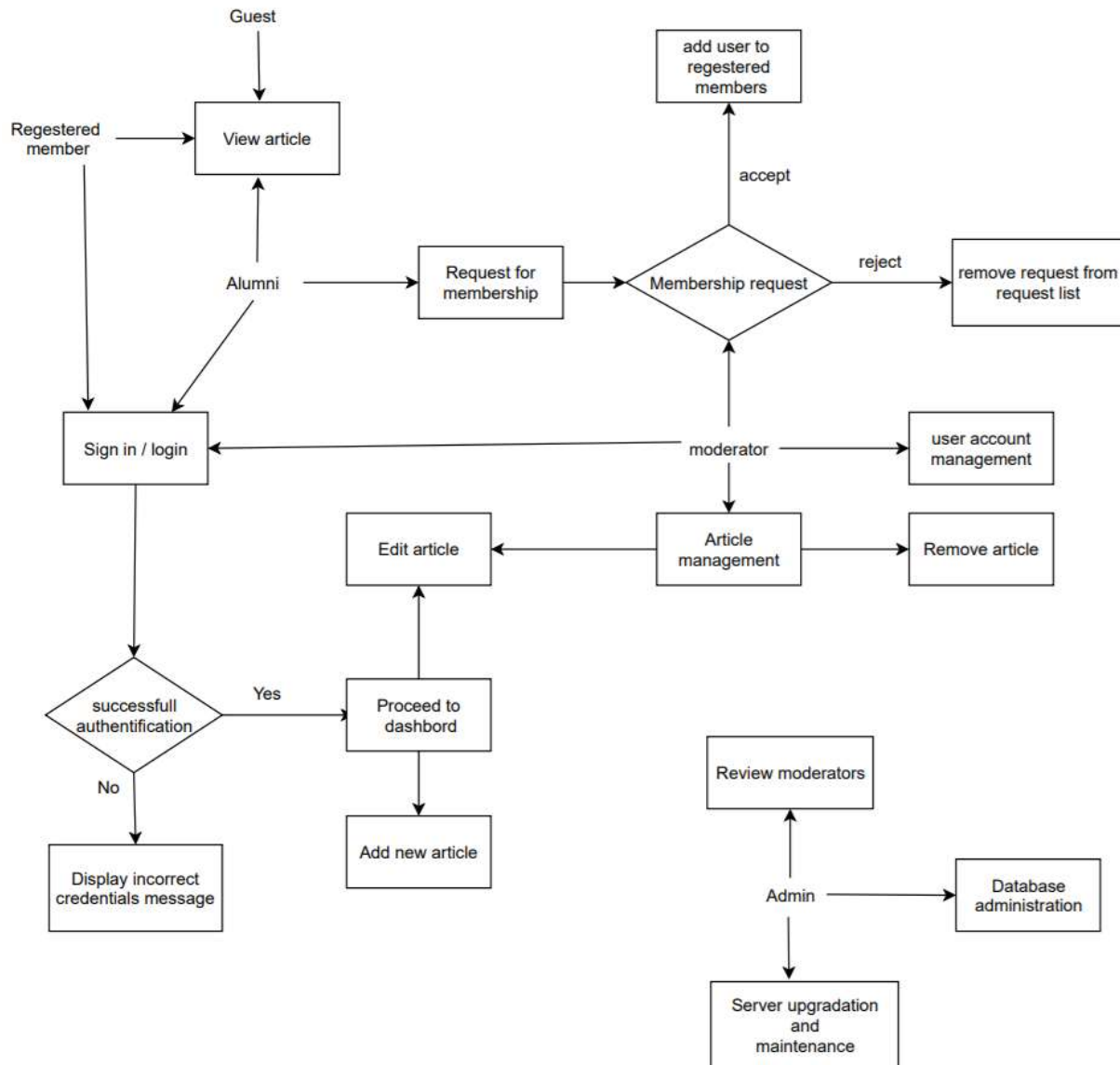
1.6 References and Acknowledgments

- IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998
- Use case analysis from eduserver course object-oriented systems NITC. [link](#)
- Object Oriented Modelling and Design with UML-Michael Blaha, James Rumbaugh.

2 Overall Description

2.1 Product Overview

All users can view articles and comments. When a registered user, alumni or moderator signs up/login, if successful, they are taken to the dashboard. Alumni and registered users can also view and comment on articles. If the authentication was not successful, a “incorrect credentials” message would be displayed. A person can log in as a guest to just view articles. Alumni can also request for registered membership. The moderator then accepts and adds the user to registered members. If rejected, they are removed from the request list. Moderator also manages the user accounts and articles. Moderators can review and edit/remove the articles if not appropriate. A centralized database of the registered users and articles are created and are only accessible to the admin. Admin also reviews the moderator’s works and manages the server up-gradation and maintenance.



2.2 Product Functionality

NITC magazine app allows:

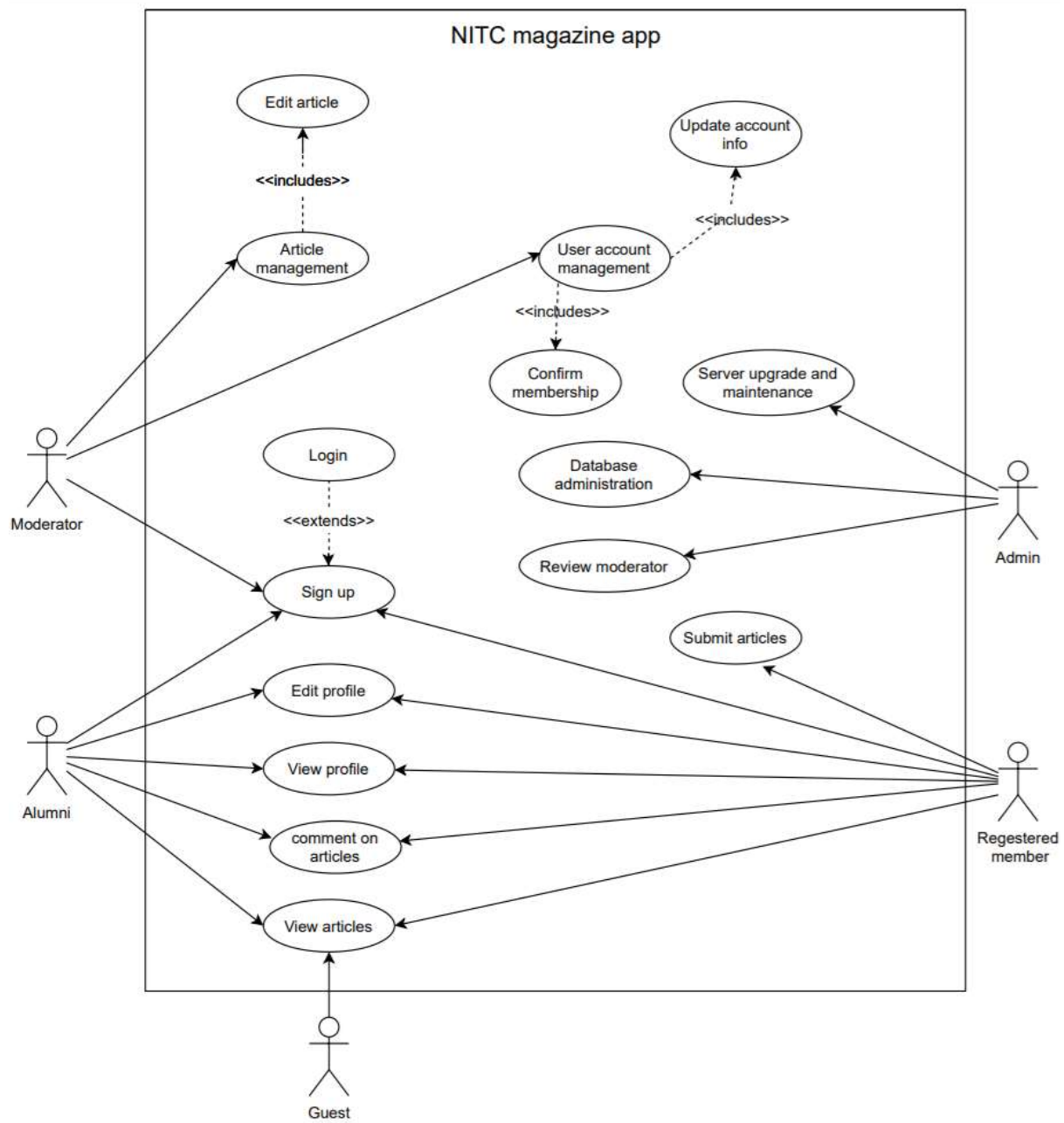
- NITC magazine app allows registered members to submit articles.
- The app allows the registered members, alumni and guests to view the posted articles.
- The app allows the registered members, alumni and guests to comment on the posted articles.
- The app allows the registered members and alumni to view the profile.
- The app allows the registered members and alumni to edit the profile details.
- The app allows alumni to send membership requests thus requesting to make them a registered member.
- The app allows the moderators to accept or reject membership requests.
- The app allows the moderators to edit the account information of registered members and alumni.
- The app allows the moderators to manage the posted articles , edit them, remove them etc.
- The app allows the admin to manage the database, manage the server .
- The app allows the admin to review the moderators. Add new moderators, remove them and monitor their activities.

3 Specific Requirements

3.1 Functional Requirements

- F1: The system shall add the user to the database after registration.
- F2: The system shall verify user details while they are logging in.
- F3: The system shall add the accepted membership request by the moderators to the database.
- F4: The system shall show all the details of the articles.
- F5: The system shall let the registered members add the articles to the database after submission.
- F6: The system shall let the moderator update the changes to the article in the database.
- F7: The system shall let the moderator update the user details in the database and manage the user database.
- F8: The system shall let the admin manage the database and make changes.
- F9: The system shall let the admin manage, add or remove moderators.
- F10: The system shall let the user view, edit their profile and make changes to the database.
- F11: The system shall add the comments by alumni & registered members to the database.
- F12: The system shall allow the admin to update/upgrade the server and manage it. This includes server shutdown activities.

3.2 Use Case Model



3.2.1 Use Case #1 (Sign up a user-U1)

Author – Abhinav P

Purpose - To sign in an institute alumni / registered member and proceed to dashboard

Requirements Traceability – F1

Priority - High

Preconditions - User must have NITC roll no, mail id and mobile number. Registered members should have a registered mobile number.

Post conditions –

1. The user successfully completes the sign in and arrives at the dashboard.
2. The user is able to access the various features of the application via the dashboard.
3. The user's details (email address, password and mobile number) obtained through the registration are stored in a database.

Actors –

1. Alumni:
 - a. Downloads the application and opens it / opens the website.
 - b. Opens the web application and fills the sign-up form.
2. Registered member:
 - a. Downloads the application and opens it /opens the website.
 - b. Opens the web application and fills the sign up form

Extends – None

Flow of Events -**1. Basic Flow:**

Actor's action	System's response
The user enters his/her valid NITC email address in the corresponding field.	The system validates the email address and displays a green tick.
The user enters his/her registered password in the corresponding field	
The user presses the 'Login' button	The system sends the password to verify and waits for the response from that module.

2. Alternative Flow:

Actor's action	System's response
The user enters his/her valid NITC email address in the corresponding field.	The system validates the email address and displays a green tick.
The user enters his/her registered password in the corresponding field	
The user presses the 'Login' button	The system states a message that the email address is not a registered user and to sign-in instead.
The user presses the 'Sign-in' text	The system displays the sign-in page.

3. Exceptions:

Actor's action	System's response
The user enters an email address in the corresponding field.	The system displays a message stating that the email address is not a valid NITC email address.
The user enters an email address in the corresponding field. The user enters a password in the corresponding field and presses the 'Login' button.	If the response from the verify module is "password is incorrect", the message will be displayed without redirection.

Includes - None

Notes/Issues – None

3.2.2 Use Case #2 (Login a user – U2)

Author – Abhiram K P

Purpose - To login a user through the portal and access the dashboard of the magazine

Requirements Traceability – F2

Priority - High

Preconditions - User must have login credentials such as username and password

Post conditions - The user is able to login and arrives at the dashboard. The user is able to access the various features of the application via the dashboard.

Actors –

1. Alumni:
 1. Opens the web application
 2. Fills the login form
2. Registered member:
 - a. Opens the web application
 - b. Fills the login form
3. Device:
 - a. A system for installing the application/ opening web browser.
 - b. Device runs the application

Extends – None

Flow of Events

1. Basic Flow:

Actor's action	System's response
The user enters his/her valid NITC email address in the corresponding field.	The system validates the email address and displays a green tick.
The user enters his/her registered password in the corresponding field	
The user presses the 'Login' button	The system sends the password to verify and waits for the response from that module.

2. Alternative Flow;

Actor's action	System's response
The user enters his/her valid NITC email address in the corresponding field.	The system validates the email address and displays a green tick.
The user enters his/her registered password in the corresponding field	
The user presses the 'Login' button	The system states a message that the email address is not a registered user and to sign-in instead.
The user presses the 'Sign-in' text	The system displays the sign-in page.

3. Exceptions:

Actor's action	System's response
The user enters an email address in the corresponding field.	The system displays a message stating that the email address is not a valid NITC email address.
The user enters an email address in the corresponding field. The user enters a password in the corresponding field and presses the 'Login' button.	If the response from the verify module is "password is incorrect", the message will be displayed without redirection.

Includes - None

Notes/Issues – None

3.2.3 Use Case #3 (View Profile – U3)

Author – Tom Saju

Purpose – To view the profile of alumni / registered members.

Requirements Traceability – F10

Priority – High

Preconditions – Alumni/ registered members must be registered and logged in to their accounts.

Post conditions –

1. The user is shown the profile of alumni/ registered users.

Actors –

1. Alumni
 - a. Downloads the application and opens it / opens the website.
 - b. Opens the application, clicks on View profile.
2. Registered Member:
 - a. Downloads the application and opens it / opens the website.
 - b. Opens the application, click on View profile.
3. Device:
 - a. A system for installing the application and opening the application.
 - b. Device runs the application.

Flow of Events –

1. Basic Flow:

Actor's action	System's response
----------------	-------------------

The user enters the credentials	The system checks and validates the key and proceeds the user to the dashboard page.
The user clicks on View Profile	The system responds with profile details.

2.Alternative Flow:

Actor's action	System's response
The user presses the Alumni/Registered member button	The system redirects to the Alumni / Registered member page.

3.Exceptions:

Actor's action	System's response
The user enters credentials.	The system displays a message that the credentials are not valid.

Extends – None

Includes – None

Notes/Issues – None

3.2.4 Use Case #4 (Edit Profile – U4)

Author – Tom Saju

Purpose – To edit and make changes to the details of alumni / registered members.

Requirements Traceability – F10

Priority – Medium

Preconditions – Alumni/ registered members must be registered and logged in to their accounts.

Post conditions –

1. The alumni/ registered member successfully completes the sign in and arrives at the profile dashboard.
2. The user is able to access the edit profile feature of the application via dashboard.
3. The user can edit the profile details.

Actors –

1. Alumni:
 - a. Downloads and opens the application.
 - b. Opens the web application, logs in, opens edit profile and makes changes to fields that can be changed in the application.
2. Registered Members:
 - a. Downloads and opens the application.
 - b. Opens the web application, logs in, opens edit profile and makes changes to fields that can be changed in the application.
3. Device:
 - a. A system for installing the application and opening application.
 - b. Device runs the application.

Flow of Events –

1.Basic Flow:

Actor's action	System's response
The user enters the credentials	The system checks and validates the key and proceeds the user to the dashboard page.

The user clicks on Edit Profile	The system shows all the options under edit profile menu.
The user edits the details and clicks the 'Run' button	The system updates the details in the database.

2. Alternative Flow:

Actor's action	System's response
The user presses the 'Run' button	The system responds with the message that no changes were made
The user presses the Alumni/Registered member button	The system redirects to the Alumni/Registered member page

3. Exceptions:

Actor's action	System's response
The user enters credentials.	The system displays a message that the credentials are not valid.
Tries to edit a field that cannot be changed	Displays the error message 'You are not authorized to edit this field'.

Extends – None

Includes – None

Notes/Issues – None

3.2.5 Use Case #5 (Comment on article – U5)

Author - Abhinav P

Purpose – To comment on a posted article by alumni / registered members.

Requirements Traceability – F11

Priority – High

Preconditions – Alumni/ registered members must be registered and logged in to their accounts.

Post conditions –

1. The user successfully completes their comment and a new comment is added to the post
2. The user's comment is stored in the database

Actors –

1. Alumni:
 - a. Downloads the application and opens it / opens the website.
 - b. Opens the web application, logs in, opens a post and writes a comment in the comment field.
2. Registered member:
 - a. Downloads the application and opens it /opens the website.
 - b. Opens the web application, logs in, opens a post and writes a comment in the comment field.
3. Device:
 - a. A system for installing the application / opening web browser
 - b. Device runs the application

Flow of Events –**1. Basic Flow:**

Actors' Action	System's Response
The user requests a specific post	The system responds with the post content and comments.
The user types a new comment and click `post comment`	The system checks if the user is logged in or not. if logged in the system adds comment to database

2. Alternative flow:

Actors' Action	System's Response
The user presses the 'Login' text.	The system redirects to the login page.

3. Exceptions:

Actors' Action	System's Response
The user requests a post	The system displays a message that post is not available
The user enters a comment and clicks `post comment`	The system displays a message that user is not logged in

Extends – None

Includes – None

Notes/Issues - None

3.2.6 Use Case #6 (View Article– U6)

Author - Abhinav P

Purpose – To view on a posted article by alumni / registered member/guest

Requirements Traceability – F4

Priority – High

Preconditions – None

Post conditions –

1. The user is shown the posted article along with all the comments.

Actors –

1. Alumni:
 - a. Downloads the application and opens it / opens the website.
 - b. Opens the application, clicks on an article
2. Registered member:
 - a. Downloads the application and opens it / opens the website.
 - b. Opens the application, clicks on an article
3. Guest:
 - a. Downloads the application and opens it / opens the website.
 - b. Opens the application, clicks on an article
4. Device:
 - a. A system for installing the application / opening web browser
 - b. Device runs the application

Flow of Events –**1. Basic Flow:**

Actors' Action	System's Response
The user requests a specific post	The system responds with the post content and comments.

2. Alternative flow:

Actors' Action	System's Response
The user presses the 'Login' text.	The system redirects to the login page.

3. Exceptions:

Actors' Action	System's Response
The user requests a post	The system displays a message that post is not available

Extends – None**Includes – None****Notes/Issues – None**

3.2.7 Use Case #7 (Submit Article– U7)

Author – Tom Saju

Purpose – To post an article by the registered member.

Requirements Traceability – F5

Priority – High

Preconditions – Registered members must be registered and logged in to their accounts.

Post conditions –

1. The posted article is made visible to all who uses the application

Actors –

1. Registered Member:
 - a. Downloads and opens the application.
 - b. Opens the web application, logs in, posts the article.
2. Device:
 - a. A system for installing the application and opening application.
 - b. Device runs the application.

Flow of Events –

1.Basic Flow:

Actor's action	System's response
The registered member enters the credentials.	The system checks and validates the key and proceeds the user to the dashboard page.
The user clicks on Submit article and hits 'post'.	The system adds the details of the article to its database and broadcasts it to the users.

2.Alternative Flow:

Actor's action	System's response
The user presses login.	The system redirects to login page
The user presses the Alumni/Registered member button.	The system redirects to the Alumni/Registered member page

3.Exceptions:

Actor's action	System's response
The user enters credentials.	The system displays a message that the credentials are not valid.
Tries to edit a field that cannot be changed	Displays the error message 'You are not authorized to edit this field'.

Extends – None

Includes – none

3.2.8 Use Case #8 (Database Administration– U8)

Author - Abhinav P

Purpose – To manage the database and format/clear the database by the admin.

Requirements Traceability – F8

Priority – High

Preconditions – User must be an admin and must have admin privileges to directly access the

database along with admin key.

Post conditions –

1. The admin successfully manages and modifies the database.
2. The admin is taken to the admin menu after the modification

Actors –

1. Admin:
 - a. Access the database server. types admin key and opens database section.
 - b. Makes necessary changes to the database.

Flow of Events –

1. Basic Flow:

Actors' Action	System's Response
The admin enters the admin key	The system checks the key and proceeds to the admin dashboard.
The admin clicks on database modification	The system gives all database modification options to the admin.
The admin presses the run button.	The system makes the necessary changes to the database.

2. Alternative flow:

Actors' Action	System's Response
----------------	-------------------

The user presses `run` button.	The system displays a message that no changes made.
The user presses the admin text.	The system redirects to the admin page.

3. Exceptions:

Actors' Action	System's Response
The user enters his/her admin key	The system displays a message that admin key is not valid

Extends – None

Includes – None

Notes/Issues - None

3.2.9 Use Case #9 (Server Upgradation and Maintenance – U9)

Author – Abhiram K P

Purpose - To upgrade properties of the server or confirm the proper working of it by fixing the errors as part of maintenance

Requirements Traceability – F12

Priority - High

Preconditions - The administrator must login to the admin panel using admin credentials

Post conditions -

1. New features are added or modifications are made to the server
2. Errors occurred to the server are resolved

Actors -

1. Admin:
 - a. Access the admin panel by logging in with admin credentials
 - b. Make necessary modifications/ fixes or add upgradations to the server

Flow of Events

1. Basic Flow:

Actor's action	System's response
The user enters the admin key	The system checks and validates the key and proceeds to the admin dashboard.
The user clicks on Server upgradation and maintenance	The system shows options for server upgradation to the admin
The user adds new features to the server or fixes the issues present	
The user presses the run button	The system makes the necessary changes to the server

2. Alternative Flow:

Actor's action	System's response
The user presses the 'run' button	The system responds with the message that no changes were made
The user presses the 'admin' button	The system redirects to the admin page

3. Exceptions:

Actor's action	System's response
The user enters admin credentials	The system displays a message that admin key is not valid
The user enters an email address in the corresponding field. The user enters a password in the corresponding field and presses the 'Login' button.	If the response from the verify module is “password is incorrect”, the message will be displayed without redirection.

Includes - None

Extends – None

Notes/Issues – None

3.2.10 Use Case #10 (Article Management– U10)

Author – Abhiram K P

Purpose - Manage the articles; modify or remove the posts from the server

Requirements Traceability – F6

Priority - High

Preconditions

- 1. The moderator must login to the moderator panel using the credentials

Post conditions -

- 1. Changes are made to the articles on the server
- 2. Articles are archived/hidden from the server

Actors -

- 1. Moderator:
 - 1. Access the moderator panel by logging in with credentials.

-
2. Make necessary changes to the article fixes or remove/ hide the article from the server

Extends – U11

Flow of Events -

1. Basic Flow:

Actor's action	System's response
The user enters the moderator credentials	The system checks and validates the key and proceeds the user to the moderator dashboard page
The user clicks on Article management	The system shows all the options under article management menu
The user edits the article or removes/ archives it	
The user presses the 'Run' button	The system makes the necessary changes to the server

2. Alternative Flow:

Actor's action	System's response
The user presses the 'Run' button	The system responds with the message that no changes were made
The user presses the 'Moderator' button	The system redirects to the moderator dashboard page

3. Exceptions:

Actor's action	System's response
The user enters moderator credentials	The system displays a message that the credentials are not valid

Includes – None.

Notes/Issues – None.

3.2.11 Use Case #11 (Edit Article – U11)

Author – Abhiram K P

Purpose - Edit and make changes to the articles published on magazine

Requirements Traceability – F6

Priority - High

Preconditions - The moderator must login to the moderator panel using the credentials

Post conditions -

1. Changes are made to the articles on the server

Actors –

1. Moderator:
 - a. Access the moderator panel by logging in with credentials.
 - b. Edit the contents/ add modifications to the articles.

Extends – None

Flow of Events

1. Basic Flow:

Actor's action	System's response
The user enters the moderator credentials	The system checks and validates the key and proceeds the user to the moderator dashboard page
The user clicks on Article management	The system shows all the options under article management menu
The user edits the article	
The user presses the 'Run' button	The system makes the necessary changes to the server

2. Alternative Flow:

Actor's action	System's response
The user presses the 'Run' button	The system responds with the message that no changes were made
The user presses the 'Moderator' button	The system redirects to the moderator dashboard page

3. Exceptions:

Actor's action	System's response
The user enters moderator credentials	The system displays a message that the credentials are not valid

Includes - None

Notes/Issues – None

3.2.12 Use Case #12 (User Account Management– U12)

Author - Abhinav P

Purpose – To manage the user accounts by changing account details, accept/reject membership requests.

Requirements Traceability – F7

Priority – High

Preconditions – Users must be logged into the system as a moderator. The user must have a moderator login id and password for successful logging into the system.

Post conditions –

1. The user successfully completes the sign in and arrives at the moderator dashboard.
2. The user is able to access the account management features of the application via the dashboard.

-
3. The user is able to accept/reject all membership requests.

Actors –

1. Moderator:
 - a. Downloads the application and opens it / opens the website.
 - b. Opens the application and login as a moderator. uses various features in the dashboard to manage accounts of users.
2. Device:
 - a. A system for installing the application / opening web browser.
 - b. Device runs the application

Flow of Events –

1. Basic Flow:

Actors' Action	System's Response
The user enters his/her moderator id and password in login	The system checks & validates the id and password and proceeds to moderator dashboard
The user clicks on user account management	The system shows all the options under user management menu
The user renews passwords, changes account data, accepts/ reject membership requests	The system generates the sequences to be executed
The user presses the `Run` button	The system makes the necessary changes to the server by executing the generated sequences

2. Alternative flow:

Actors' Action	System's Response
The user presses the moderator button	The system redirects the user to the moderator dashboard.
The user clicks the run button with system generated sequence empty	The system displays a message that no changes to be committed.
The user presses the 'Login' text.	The system redirects to the login page.

3. Exceptions:

Actors' Action	System's Response
The user enters his/her moderator id and password in login	The system displays a message that User is invalid and login failed
The user requests an account	The system displays a message that account is not found

Extends –None

Includes – U13, U14

Notes/Issues - None

3.2.13 Use Case #13 (Confirm Membership – U13)

Author - Abhinav P

Purpose – To enable moderators to accept / reject the membership request by different user accounts.

Requirements Traceability – F3

Priority – High

Preconditions – Users must be logged into the system as a moderator. The user must have a moderator login id and password for successful logging into the system.

Post conditions –

1. The user successfully completes the sign in and arrives at the moderator dashboard.
2. The user is able to access the account management features of the application via the dashboard.
3. The user is able to accept/reject all membership requests.

Actors –

1. moderator:
 - a. Downloads the application and opens it / opens the website.
 - b. Opens the application and login as a moderator.
 - c. Uses various features in the dashboard to manage accounts of users.
 - d. User clicks on `confirm membership` and accept/reject requests.
2. Device:
 - a. A system for installing the application / opening web browser.
 - b. Device runs the application

Flow of Events –

1. Basic Flow:

Actors' Action	System's Response
----------------	-------------------

The user enters his/her moderator id and password in login	The system checks & validates the id and password and proceeds to moderator dashboard
The user clicks on user account management	The system shows all the options under user management menu
The user clicks on `confirm memberships` button	the system shows all membership requests along with buttons to accept/reject these requests.
The user accepts/ reject membership requests by clicking on `accept` or `reject` button respectively	The system generates the sequences to be executed for accepting and rejecting the requests.
The user presses the `Run` button	The system makes the necessary changes to the server by executing the generated sequences.

2. Alternative flow:

Actors' Action	System's Response
The user presses the moderator button	The system redirects the user to the moderator dashboard.
The user clicks the run button with system generated sequence empty	The system displays a message that no changes to be committed.
The user presses the `Login` text.	The system redirects to the login page.

3. Exceptions:

Actors' Action	System's Response
The user enters his/her moderator id and password in login.	The system displays a message that User is invalid and login failed.

Extends – None.

Includes – None.

Notes/Issues – None.

3.2.14 Use Case #14 (Update Account Information – U14)

Author - Abhinav P

Purpose – To enable moderators to update the information of user accounts. view account data, cancel an account membership.

Requirements Traceability – F7

Priority – High

Preconditions – Users must be logged into the system as a moderator. The user must have a moderator login id and password for successful logging into the system.

Post conditions –

1. The user successfully completes the sign in and arrives at the moderator dashboard.
2. The user is able to access the account management features of the application via the dashboard.
3. The user is able to view/ modify account data, and cancel already existing memberships.

Actors –

-
1. Moderator:
 - a. Downloads the application and opens it / opens the website.
 - b. Opens the application and login as a moderator.
 - c. Uses various features in the dashboard to manage accounts of users.
 - d. User clicks on `update account info` to update various user account data.
 2. Device:
 - a. A system for installing the application / opening web browser.
 - b. Device runs the application

Flow of Events –

1. Basic Flow:

Actors' Action	System's Response
The user enters his/her moderator id and password in login	The system checks & validates the id and password and proceeds to moderator dashboard
The user clicks on user account management	The system shows all the options under user management menu
The user clicks on `update account info` button	The system shows all info of the user accounts excluding passwords along with buttons to edit, cancel memberships.
The user updates the data in the fields and cancel the required memberships using `cancel membership` button	The system generates the sequences to be executed for modifying account info.
The user presses the `Run` button	The system makes the necessary changes to the server by executing the generated sequences.

2. Alternative flow:

Actors' Action	System's Response
The user presses the moderator button	The system redirects the user to the moderator dashboard.
The user clicks the run button with system generated sequence empty	The system displays a message that no changes to be committed.
The user presses the 'Login' text.	The system redirects to the login page.

3. Exceptions:

Actors' Action	System's Response
The user enters his/her moderator id and password in login	The system displays a message that User is invalid and login failed

Extends – None

Includes – None

Notes/Issues - None

3.2.15 Use Case #15 (Review Moderator– U15)

Author – Bhavana Ketavath

Purpose – To manage, add and remove moderators by the admin.

Requirements Traceability – F9

Priority – High

Preconditions – User must be an admin and must have admin privileges to directly access the database along with admin key.

Post conditions –

1. The admin successfully manages the moderators.
2. The admin is taken to the admin menu.

Actors –

1. Admin:
 - a. Access the admin server. types admin key and opens manage moderator’s dashboard.
 - b. Makes necessary changes to the database of moderators or checks the moderators’ actions.

Flow of Events –

1. Basic Flow:

Actors’ Action	System’s Response
The admin enters the admin key	The system checks the key and proceeds to the admin dashboard.

The admin clicks on review moderators	The system gives options to manage, add or remove moderators. The system also lists the actions of each moderator.
The admin presses the run button.	The system makes the necessary changes to the database.

2. Alternative flow:

Actors' Action	System's Response
The user presses `run` button	The system displays a message that no changes made
The user presses the admin text.	The system redirects to the admin page.

3. Exceptions:

Actors' Action	System's Response
The user enters his/her admin key	The system displays a message that admin key is not valid

Extends – None

Includes – None

Notes/Issues – None

Appendix A - Activity Log

Group Meetings:

Sl. No	Date	Time
1	09-10-2021	4 PM - 5 PM
2	12-10-2021	7 PM - 8 PM
3	14-10-2021	10 AM - 11 AM
4	16-10-2021	4 PM - 5 PM
5	17-10-2021	10AM - 11AM
6	17-10-2021	3AM - 5AM
7	17-10-2021	7AM - 8AM

Contributions:

1. Introduction – Bhavana Ketavath
2. Overall description – Tom Saju, Abhiram KP and Bhavana K
3. Specific Requirements
 - Functional Requirement - Abhinav P, Tom Saju
 - Use case model diagram – All the team members.
 -

Use Case #	Team Member
1, 5, 6, 8, 12, 13, 14	Abhinav P
3, 4, 7	Tom Saju
2, 9, 10, 11	Abhiram KP
15	Bhavana Ketavath