IT314 - Software Engineering

Lab 3 - Specifying Functional and Non - Functional Requirements

Educational Networking Tool for College Students

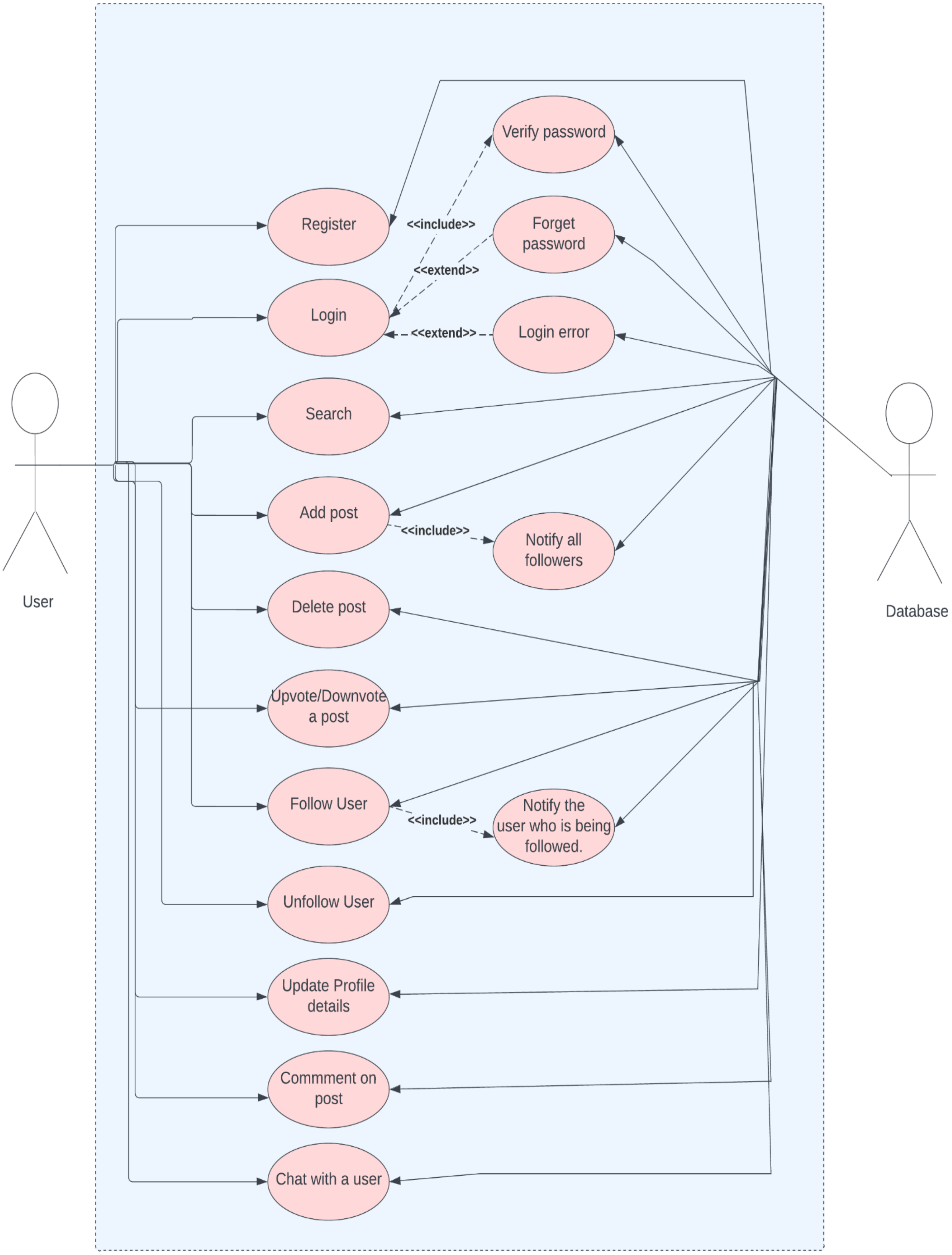
Group - 29

Group Members:

- 202001150 - Devdeep Shetranjiwala
- 202001173 - Harsh Patel
- 202001270 - Naisarg Shah
- 202001412 - Zeel Bhanderi
- 202001424 - Kaushal Patel
- 202001431 - Shrey Shah
- 202001441 - Priyanshi Parmar
- 202001449 - Achyut Shah
- 202001452 - Krish Patel
- 202001461 - Harsh Agheda
- 202001462 - Om Patel

Contents:

Use Case Diagram



Use Case Description

In the use case diagram, the user interacts with the system by performing various actions like logging in, searching content/people, adding posts, deleting posts, chatting with a user, commenting on posts, etc. To these actions, the database responds and gives appropriate results back to the user.

Textual Description of Use Cases

Use Case	**Register**
Description	Register user into the Database
Actors	**Primary Actor:** User **Secondary Actor:** Database
Pre Conditions	1. Input interface should be available to the user.
Post Conditions	1. Users should be able to log in using the given credentials during registration.
Flow	1. The user inserts the credentials. 2. The system will check the credentials and verify them. 3. The user's credentials will be stored in the Database if verified.
Alternate Flow	In step 3, if credentials are not verified, 3.a The system will ask the user to re-enter the credentials.

Use case	**Log in**
Description	Verify user credentials of user and give access to services
Actors	**Primary Actor:** User **Secondary Actor:** Database
Pre Conditions	1. The user must be connected to the internet and has already created an account. 2. The user accesses the input interface of the system.
Post Conditions	The user receives access to the system after logging in successfully.
Flow	1. User enters login credentials. 2. The credentials get verified from the database. 3. The user gets access to the system.
Alternate Flow	2.a If the credentials don’t get verified, then the user will get an error and will be shown an option “forgot password” by which he will receive a link to change his/her password on registered e-mail.

Use Case	**Search**
Description	Search a topic or a user on a website.
Actors	**Primary Actor:** User **Secondary Actor:** Database
Pre Conditions	Website should be accessible and the home page should be open.
Post Conditions	User should get the details of the person or the content he searched for.
Flow	1. User searches the name of the topic or the person he wants to see. 2. The system finds the name in the database. 3. Once found,the system brings the response from the database and displays the information about the person or the content he was looking for.
Alternate Flow	2.a If the person or the topic name is not found, the screen shows “result not found”.

Use case	**Add Post**
Description	This use case describes the process by which a user can create and publish a new post onto the website.
Actors	**Primary Actor:** User **Secondary Actor:** Database
Pre Conditions	1. The user must be registered and logged into their account on the system. 2. The user must have the necessary permissions to create and publish a post.
Post Conditions	A new post is created and published on the system.
Flow	1. The user navigates to the "Add Post" button on the system. 2. The user selects the appropriate course or subject for which they want to create the post. 3. The user enters the title of the post. 4. The user writes the content of the post. 5. The user can add attachments. 6. The user submits the post for publishing. 7. The post is published onto the system for other college students to view and comment on. 8. Notification is sent to all followers of the user about the post.
Alternate Flow	2.a. If the appropriate course or subject is not available for selection: 2.a.1 The user can contact the system's support team to request the addition of the course or subject. 5.a User gets an error while adding attachments. 5.a.1 System provides guidance on the acceptable format, size, and number of attachments.

Use Case	**Delete Post**
Description	The user who have posted the post want to delete the post
Primary Actor	**Primary Actor:** User who created the post **Secondary Actor** : Data Base
Pre Conditions	1.The Actor must have posted that post 2.The system is in operation
Post Conditions	1.The Post is deleted
Flow	1.The Actor logs in to the system. 2.The System Authenticate the actor and starts a session 3.The Actor chooses to delete the post 4.The System will ask user whether he/she is sure to delete the post 5.The Actor leaves the System
Alternate Flow	1.a The System fails to authenticate the actor. ● System will inform the user and doesn’t allow it to proceed.further
Special Requirement	1. Actors are connected to the Internet.

Use Case	**Upvote/ Downvote any post**
Description	Allow users to express their approval or disapproval of a particular post by giving it an upvote or downvote.
Actors	**Primary Actor:** User **Secondary Actor:** Database
Pre Conditions	1. The user must have an account in the system. 2. The user must be logged in the system. 3. The user must have the necessary permissions to upvote/downvote posts.
Post Conditions	1. The post score gets increased by 1 if user upvotes and gets decreased by 1 if user downvotes. 2. The visibility of the post is affected.
Flow	1. To read the content in their feed, the user goes to the home page. 2. Users can choose to upvote or downvote a post. 3. When a user votes, the system records that vote and adjusts the post's score. 4. The visibility of the post may be influenced if its score has significantly changed, as such by being moved up or down in a list.
Alternate Flow	3a. The user may click the alternative button to alter their vote if they have previously upvoted or downvoted the post and the system responds accordingly. 3b. The user can click the upvote/downvote button again to reverse their action if they accidentally voted one way or the other and the system responds accordingly. 3c. The user may be prompted to log in or get the necessary permissions before their vote is registered if they attempt to upvote or downvote a post while not signed in or have insufficient permissions.
Special Requirement	1. The user must be connected to the internet.

Use case	**Follow user**
Description	Follow a particular user to connect to a specific user and get notified if the following user adds a post.
Actors	**Primary Actor:** User **Secondary Actor:** Database
Pre Conditions	1. Users should be logged in, and the following should have an account/profile on this application.
Post conditions	1. Followers should get notified when the following adds a post. 2. The following should get a notification that some user is following him/her when the follower follows the user for the first time.
Flow	1. Search for the user you want to follow. 2. Go to the profile of the user you want to follow. 3. Click on the follow button to follow the user. 4. The following user will get a notification.

Use case	**Unfollow user**
Description	Unfollow a particular user who was being followed previously by user
Actors	**Primary Actor:** User **Secondary Actor:** Database
Pre Conditions	1. Users should be logged in, and the follower should follow the user.
Post conditions	1. Follower should not be notified of the post the previously followed user added.
Flow	1. Search for the user you want to unfollow. 2. Go to the profile of the user you want to unfollow. 3. Click on the unfollow button to unfollow the user.

Use Case	**Update Profile Details**
Description	The user will be shown a form containing various updatable fields to update their profile details.
Actors	**Primary Actor:** User **Secondary Actor:** Database
Pre Conditions	1. The user is connected to the internet. 2. The user has a registered account. 3. The user has logged into their account.
Post Conditions	1. The user's profile details are updated in the database. 2. The updated details are displayed on the user's profile.
Flow	1. The user navigates to the profile section 2. The user clicks on "Update Profile" button 3. A form containing various fields, that can be updated, is displayed 4. The user updates the required fields 5. The user clicks on the “Submit” button 6. The user’s profile is updated if the new data is valid. 7. The user is redirected to their updated profile page and can view their changes
Alternate Flow 1	5a. If the user does not want to commit the changes, they can click on the “Cancel” button 6a. If the new data is invalid or incomplete, the system displays an error message and asks the user to correct the errors

Use Case	**Comment on Post**
Description	To provide feedback or engage in a discussion related to a post on a platform.
Actors	**Primary Actor:** User **Secondary Actor:** Database
Pre-Condition	1. User must have account in the system 2. User should be logged in to the system 3. The user has access to a post that they wish to comment on.
Post-Condition	1. The comment posted by the user should be displayed below the post. 2. The website may send notifications to the user who has created the post.
Flow	1. The user is logged in to the system and navigates to the post that they wish to comment on. 2. The user clicks on the comment icon or the comment box below the post. 3. The user types in their comment in the comment box. 4. The user may choose to include multimedia elements such as emojis in their comment. 5. The user clicks on the submit button to post their comment. 6. The social media platform or website displays the user's comment below the post.
Alternate Flow	1a.) If the user is not logged in the user is redirected to the login page. 6a.) If the user wishes to edit or delete their comment, they can do so by clicking on the edit or delete button provided next to their comment.
Special Requirement	1. User must be connected to Internet

Use Case	**Chat with a user**
Description	This use case explains the procedure when a user wishes to communicate with another system user.
Actors	**Primary Actor:** User **Secondary Actor:** Database
Pre Conditions	1. The sender and receiver must both have confirmed accounts on the server as prerequisites. 2. The user signed into the platform. 3. The sender is aware of the receiver's identity.
Post Conditions	1. Throughout the entire procedure, the database is consistent. 2. The receiver/sender has been informed by the system.
Flow	1. The sender searches for the receiver's username in the search bar. 2. The sender opens the receiver's profile. 3. The sender clicks on the chat icon. 4. The sender types the message and sends it to the receiver. 5. The receiver receives the notification when he opens the account. 6. The receiver opens the chat box with the sender by searching for his name. 7. The receiver types his/her reply to the sender. 8. The sender receives the notification.
Alternate Flow	N/A

Non-Functional Requirements:

- **Security**: The tool should have robust security measures to protect the privacy and security of its users.

  Justification: Because it tackles the vital need to protect a system or application from possible dangers, assaults, and illegal access, security is a crucial non-functional prerequisite. A system or programme may be susceptible to various cyber threats without sufficient security measures in place, including data leaks, theft, hacking, and malware assaults.

- **User-friendly interface**: The interface should be intuitive, easy to navigate, and accessible to all users.

  Justification: Because it addresses the crucial need to make a system or programme simple to use and comprehend for its intended users, user-friendliness, also known as usability, is a crucial non-functional prerequisite. Users may find it difficult to engage with the system or programme without sufficient usability, which can cause frustration, mistakes, and a bad user experience.

- **Scalability**: The tool should be able to accommodate a large number of users and handle increasing amounts of data.

  Justification: Because it addresses the crucial need to guarantee that a system or application can manage rising workloads and user demands over time without noticeably degrading performance, scalability is a crucial non-functional prerequisite. Without sufficient scaling, a system or application may become overburdened by rising user demand or traffic, which could result in decreased performance, sluggish reaction times, and even outages.

- **Performance**: The platform should have fast load times and responsive navigation to ensure a seamless user experience.

  Justification : Because it tackles the crucial need to guarantee that a system or programme can execute its intended function within reasonable timeframes and resource usage, performance is a crucial non-functional prerequisite. A system or programme that lacks sufficient performance may be slow, unresponsive, or unable to manage user demands, which could result in lost productivity or even user harm.

- **Compatibility:** The tool should be compatible with a range of devices and operating systems.

Justification : Because it addresses the crucial need to guarantee that a system or application can work properly and efficiently with other systems, software, hardware, and devices, compatibility is a crucial non-functional prerequisite. A system or programme may experience mistakes, failures, or speed problems when dealing with other systems or devices if there is insufficient compatibility.

- **Reliability**: The platform should have high availability and be able to handle unexpected spikes in traffic.

  Justification : Because it tackles the crucial need to guarantee that a system or programme executes its intended function consistently and correctly over time, without unanticipated failures or errors, reliability is a crucial non-functional prerequisite. Without sufficient dependability, a system or application may be unpredictable, which could result in decreased output or even user damage.

- **Maintenance:** The tool should have a comprehensive maintenance plan to ensure it remains up-to-date and functional.

  Justification :Because it ensures that a system or application can be upgraded, improved, and maintained over time to satisfy shifting business requirements, repair defects, and handle security risks, maintenance is a crucial non-functional prerequisite. A system or programme can become outdated, unsupported, and susceptible to security risks and efficiency problems if proper upkeep is not performed on it.