# **Requirements List**

Requirements in this color are meant for the full application, and will not be considered for the prototype

1. **Functionality**
   1. **User Authentication**
      1. **Users can authenticate with Login [MUST]**User must have a login username and password, with which they can access the platform, through a login form.

*Validation: User Login Use Case*

*Verification: Log into the system.*

* + 1. **Users can register an account**
       1. **Users can register an account [MUST]**User can fill in a form, containing username, email and password as fields, in order to become a registered user.  
          *Validation:Use Registration Use Case*

*Verification: Register an account into the system.*

* + - 1. **No accounts can have a duplicate username/email [MUST]**If a user enters into the registration form an existent username/email, they should not be able to register their account.  
         *Validation:Use Registration Use Case*

*Verification:Request to create a new account with existent username/email is rejected.*

* + - 1. **User passwords must meet requirements [COULD]**

User can not register, unless their password is at least 8 characters long.

*Validation: Use Registration Use Case*

*Verification: Request to create an account with a password less than 8 characters long, should be rejected.*

* 1. **Create and Edit Posts/Topics by Authenticated Users**
     1. **Authenticated Users “New Post” button**Authenticated users see a “New Post” button in the main top navigation bar, which takes them to the post editor.  
        *Validation:* This is shown by the *Create Post Use Case  
        Verification:* As an authenticated user, you can see a “New Post” button in the main top navigation, which when clicked, will take you to the Post Editor.
     2. **Posts Editor Page (Creation/Editing)**
        1. **Text Editor [MUST]**

The post editor page has a text editor, including formatting tools (font size, font color, bold, italics, underline) and blocks (links, code snippets).  
*Validation:* This is referenced in the *Posts Editor* *Use Case*  
*Verification:* As an authenticated user, on the post editor page, you can see a text editor with the aforementioned formatting and blocking tools.

* + - 1. **Discussion Group Selection [MUST]**

The post editor page should allow the author to select a discussion group to target the post at  
*Validation:* This is referenced in the *Posts Editor* *Use Case*  
*Verification:* As an authenticated user, on the post editor page, you can select a discussion group

* + - 1. **Submission functionality [MUST]**

The post editor page should allow the user to submit their post form, in order to publish their post on the platform

*Validation:* This is mentioned in the *Posts Editor Use Case  
Verification:* As an authenticated user, I can publish my post from the post editor page

* + - * 1. **Empty posts can not be submitted [MUST]**If a user attempts to publish an empty post to the platform through the post editor page, their request should be rejected  
           *Validation:* This is seen from the *Posts Editor Use Case  
           Verification:* As an authenticated user on the post editor page, if I try to publish an empty post, my request does not go through.
        2. **User is taken to the published post upon successful submission [MUST]**Upon successful submission of the post, the user should be taken to the published post on the platform  
           *Validation:* This is included in the *Posts Editor Use Case*  
           *Verification:* After submitting a post (that meets submission requirements) I am taken to the published post
    1. **Post Viewing**
       1. **Author of post sees an edit button [MUST]**On the post viewing page, the author of the post can access the functionality to edit the post, which will take the author to the Post Editor page  
          *Validation:* Refer to the *Post Viewing Use Case  
          Verification:* Open up a post that you published, and you should be able to edit the post
       2. **Rating [COULD]**Non-authors can give the post an integer rating out of 5 points  
          *Validation:* Refer to the *Post Viewing Use Case  
          Verification:* View a post that you did not write up yourself, and you should be able to give the post an integer rating out of 5 points

**c. User Subscription to topics and subsequent new notifications**

**1. Users can subscribe to threads and posts [MUST]**

Users are able to subscribe to a thread or post by clicking the “subscribe” button.

*Validation:* This is seen from the User Subscription Use Case

*Verification:* As an authenticated user, you can subscribe to threads and posts.

**2. User can unsubscribe [MUST]**

Users can unsubscribe to a thread or post should they decide that they are not interested or decide that it’s not of use anymore.

*Validation:* This is seen from the User Subscription Use case.

*Verification:* As an authenticated user, you can unsubscribe a thread or a post.

**3. Users are notified if posts are made on subscribed threads [MUST]**

Users already subscribed to a subscription thread is automatically notified of a new post

*Validation:* This is shown from the User Subscription Use Case.

*Verification:* As an authenticated user, you will receive a notification inform you that the post is successfully posted on the subscription thread.

**4. Users can’t subscribe to private threads [SHOULD]**

Users are not allowed to be subscribed to private threads since are meant to be private.

*Validation:* This is shown from the User Subscription Use case.

*Verification:* If you have created your own thread or post, you are not allow to unsubscribe your own thread or post.

**4.1. Group Members can subscribe**

Users who have access have the right to subscribe

*Validation:* This is seen from the User Subscription Use

Case

*Verification:* As an authenticated user and as a group

member of the team, you can subscribe the thread or

the post.

**4.2. Group Members can unsubscribe**

Users who are already subscribed to the private threads can always unsubscribe to them as well.

*Validation:* This is seen from the User Subscription Use Case.

*Verification:* An an authenticated user and as a group member, you can unsubscribe the the thread or the post.

**d. Creation of Discussion groups and optional restrictions**

**1. Create groups based on interest/tags [COULD]**

User is matched with another user based on tags used and can then decide to form a group with those users by adding them.

*Validation:* This is shown from the Discussion groups Use Case.

*Verification:* As an authenticated user, you can create groups based on interest/tags.

**2. Drop-down menu of all users [WONT]**

User gets a drop-down menu with all users registered to the website

*Validation:* This is shown from the Discussion groups Use Case.

*Verification:* User attempts to add another user via the drop-down menu

**3. Search to add users [SHOULD]**

User can use the search bar to find a particular user and add them into a group

Validation: This is shown from the Discussion groups Use Case.

Verification: User attempts to add another user to a group using the search function

**4. Users can leave the group [COULD]**

Should a user decide to leave a group for a specific reason can do so by clicking the “Leave Group” button.

*Validation:* This is shown from the Discussion groups Use Case.

*Verification:* User attempts to leave a group and is no longer a group member

**5. Users can request access to groups [SHOULD]**

Users who wish to be part of a group can request access to the group admin of the post

*Validation:* This is shown from the Discussion groups Use Case.

*Verification:* User attempts to request access to a group

**6. Group admin can add users after creation [SHOULD]**

Group admins of a post are able to add users even after the creation of a discussion group

Validation: This is shown from the Discussion groups Use Case.

Verification: group admin attempts to add another user.

**7. Suggested users to add [COULD]**

Group admins are given a suggested user drop-down menu of those who should be added to the group.

*Validation:* This is shown from the Discussion groups Use Case.

*Verification:* User attempts to create a thread and drop-down menu is shown.

**8. Group members are subscribed by default [COULD]**

Users who are in a group are subscribed to a thread by default

*Validation:* Discussion Use Case

*Verification:* User checks subscription settings and sees the group as subscribed.

**9. Group Admin can make members Group Admins[COULD]**

Group Admin can decide on whether a user can be a group admin, gaining right such as adding or removing members

*Validation:* Discussion Use Case

*Verification:* User attempts to add another user to the group after being added and is successful in doing so.

**10. User can create group[MUST]**

User can create group.

Validation: Discussion Use Case

Verification: As an authenticated user, user can create a group.

**e. Creating a User Profile and Direct Messaging**

**1. Clicking a user’s profile picture/username shows profile [MUST]**

A User can click on a targeted user’s picture/user name

through one of their post and be redirected to the targeted user’s

profile.

*Validation*: User Profiles Use Case.

*Verification*: Click on the target user's name or profile picture,

available on one of the target user's posts and you should see

their profile.

**2. User send messages to another user .[SHOULD]**

Previewing user can fills the textbox with their desired

Message.Previewing user can click on a direct message button to send a message to the target user.

*Validation*: Direct Messaging Use case.

*Verification*: Send a direct message and check if the target user

can view it.

**3. The target user see the message if he is online[SHOULD]**

Recipient user can see a message if he is online.

*Validation*: Direct Messaging Use Case.

*Verification*: User check if he has received a message.

**4. The target user receive a notification if he is offline.[SHOULD]**

Recipient user can receive a notification if he is not online.

*Validation*: Direct Messaging Use Case

*Verification*: User check the notification if he has received a notification.

1. **Usability**
   1. **User Authentication**
      1. **Users see a notification when logging in/registering an account, and their attempt is not successful [MUST]**

Users see a prompt in the login/registration form window, letting them know what issues resulted in the attempt not being successful

*Validation: User Registration Use Case, and User Login Use Case*

*Verification: All failure extensions in either the User Registration Use Case or the User Login Use Case, should result in a prompt*

* 1. **Create and Edit Posts/Topics by Authenticated Users**
     1. **Post Editor Page (Creation/Editing)**
        1. **Discussion group dropdown menu [MUST]**On the post editor page, I can see a discussion group dropdown menu, under the text editor, defaulting to “None”.  
           *Validation:* Refer to the *Post Editor Use Case  
           Verification:* As an authenticated user, if you access the post editor page (either by editing an existing post you wrote, or creating a new one) you can see the discussion group dropdown menu under the text editor.
     2. **Post Viewing**
        1. **Author of post sees an edit button [MUST]**Author can see an edit button at the top right corner of their post when they are viewing the post  
           *Validation:* Refer to the *Post Viewing Use Case  
           Verification:* View a post that you published, and you should see an edit button in the top right corner of the post

**c. Creating a User Profile and Direct Messaging**

1. **The User profile shows user details [MUST]**

Previewing user can see the profile picture, username, email and

registration date of the target user

Validation: User Profiles Use case.

Verification: Open up a user profile and ensure that the

aforementioned details are present.

1. **Reliability**
2. **Performance**
3. **Supportability**
4. **Design constraints**
5. **The Flask framework is used.**

Use of the flask web development framework imposes design restrictions regarding how web services are structured and how mobile devices are supported. Much of the project architecture is pre-determined by the flask framework.

*Validation:* Imposed by the COMP2005 context of the project

*Verification:* Examine the implementation code.

**b. Python will be used as an implementation language.**

Use of the python language imposes design restrictions

regarding how the whole project are implemented. The

implementation includes algorithm design, database design, user

interface design.

*Validation*: Imposed by the Group-L context of the project.

Verification: Examine the implementation code.

1. **Implementation requirements**
2. **Interface requirements**
3. **Physical requirements**