**Security Requirements**

*Authorization*

The user needs to login before being able to post a comment in an existing thread.

*Authorization*

The user needs to verify that they are not a robot by CAPTCHA before being able to register, post, create thread, change password or edit user information.

*Integrity*

The application needs to ensure proper validation in all input fields throughout the application.

*Accountability*

The database needs to log creation, edit and deletion in user tables.

**Misuse Cases**

*Misuse Case: 1*

Name of Misuse Case: Cross-Site Scripting (XSS)

Security Threat:

The attacker tries to to use the application to store XSS attacks.

Basic Path:

The attacker registers as a user and creates a thread with dangerous script as the post information. The data will get stored in the database and when the users click on the thread. The data will get fetched and will get sent to the browser as an javascript and that will get executed to gather information of the victims. The attacker later take the acquired login information and uses it to access other victims profiles.

Capture Points:

Implement proper input validation to strip tags and not allow scripts to be written in fields.

*Misuse Case: 2*

Name of Misuse Case: Spam

Security Threat:

The attacker tries to overload database by registers new users, post comments or creates new threads.

Basic Path:

The attacker uses software that automatically registers users, post comments or creates threats until the database is full.

Capture Points:

Implement CAPTCHA

*Misuse Case: 3*

Name of Misuse Case: SQL Injections

Security Threat:

The attacker tries to to attack application by SQL Injections.

Basic Path:

The attacker injects malicious SQL code in a input field. The data will be intepreted as a part of the query and the database will execute it as a command.

Capture Points:

Implement proper validation and use parameterized stored procedures.