

COS 301 - MINI PROJECT

FUNCTIONAL REQUIREMENTS AND
APPLICATION DESIGN

GROUP 1B

2015

Contents

1	Group Information	2
1.1	General Information	2
1.2	Group members	2
1.3	Contributions to project	2
1.3.1	Participation	2
1.3.2	Who did what	2
1.3.3	Github contribution	3
2	Functional requirements and application design	5
2.1	Use case prioritization	5
2.1.1	Critical	5
2.1.2	Important	5
2.1.3	Nice-To-Have	5
2.2	Services contracts	6
2.2.1	Analytics	6
2.2.2	Authentication	6
2.2.3	BuzzSpace	6
2.2.4	Communication	7
2.2.5	Plagiarism / Netetiquette	8
2.2.6	Tagging	8
2.2.7	Thread	9
2.2.8	Thread Posts	9
2.2.9	User	9
2.3	Required functionality	10
2.3.1	Analytics	10
2.3.2	Authentication	10
2.3.3	BuzzSpace	11
2.3.4	Communication	12
2.3.5	Plagiarism / Netetiquette	13
2.3.6	Tagging	14
2.3.7	Thread	15
2.3.8	Thread Posts	16
2.3.9	User	17
2.4	Process specifications	18
2.4.1	Authentication	18
2.4.2	BuzzSpace	19
2.4.3	Communication	21
2.4.4	Tagging	26
2.4.5	Thread	29
2.4.6	Thread posts	32
2.5	Domain Model	35

1 Group Information

1.1 General Information

We used Github as our version control service. The repository can be accessed at: <https://github.com/thinusn/Mini-Project-Requirements-Group-1B>

1.2 Group members

- 13033922 Elzahn Botha
- 12223426 Estian Rosslee
- 13025105 Jaco-Louis Kruger
- 13073878 Christopher Arajo
- 13093500 Paul Engelke
- 13028741 Frikkie Snyman
- 13019602 Thinus Naude

1.3 Contributions to project

1.3.1 Participation

Estian Rosslee (12223426) only attended the two 1-hour meetings we had, **he did not do any of the work presented in this document**. All other members of the group participated fully towards completing the project.

1.3.2 Who did what

- Planning: Everyone
- Use case prioritization: Everyone
- Services contracts: Jaco-Louis Kruger, Daniel Arajo, Thinus Naude, Paul Engelke and Frikkie Snyman
- Required functionality: Jaco-Louis Kruger and Thinus Naude
- Process specifications: Elzahn Botha, Daniel Arajo, Paul Engelke
- Domain Model: Frikkie Snyman
- LaTeX: Thinus Naude

1.3.3 Github contribution

IMAGE OF githubCommits

Figure 1: Github pushes for project.

2 Functional requirements and application design

2.1 Use case prioritization

2.1.1 Critical

A use case which is absolutely essential

- Create, Read/View/Get, Delete and Update of BuzzSpace
- Create, Read/View/Get, Delete and Update of Thread
- Create, Read/View/Get, Delete and Update of Thread Posts
- Authentication

2.1.2 Important

The system would still be useful without some of the important use cases, but the client would get quantifiably less value from the system.

- User (profiles and actions)
- User Communication (Email Templates)
- Tagging

2.1.3 Nice-To-Have

Its a requirement but the value to the client is insignificant.

- Plagiarism checker
- Netiquette checker

2.2 Services contracts

2.2.1 Analytics

IMAGE OF AnalyticsSC

Figure 2: Analytics services contracts.

2.2.2 Authentication

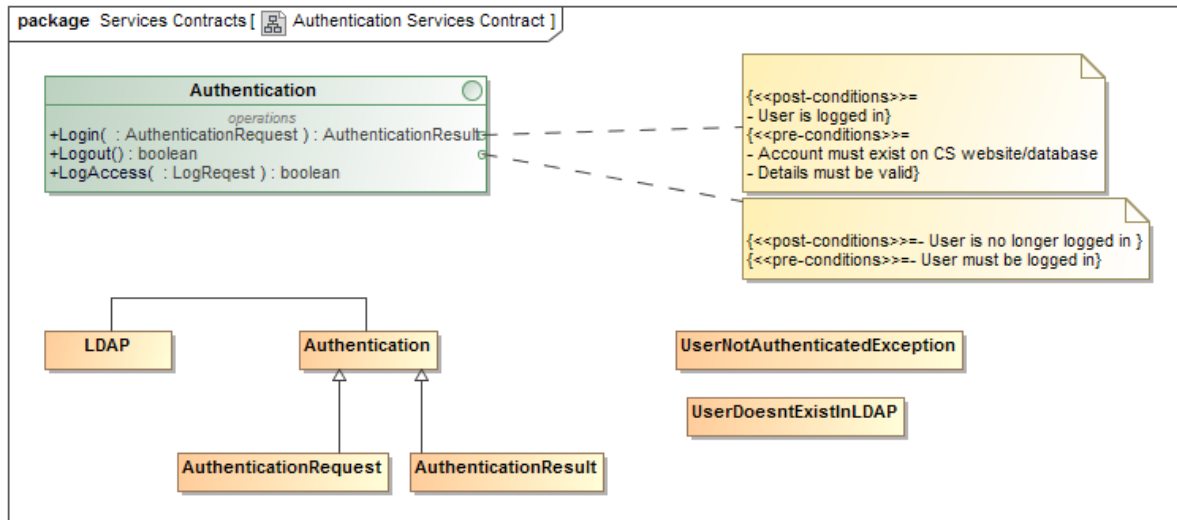


Figure 3: Authentication services contracts.

2.2.3 BuzzSpace

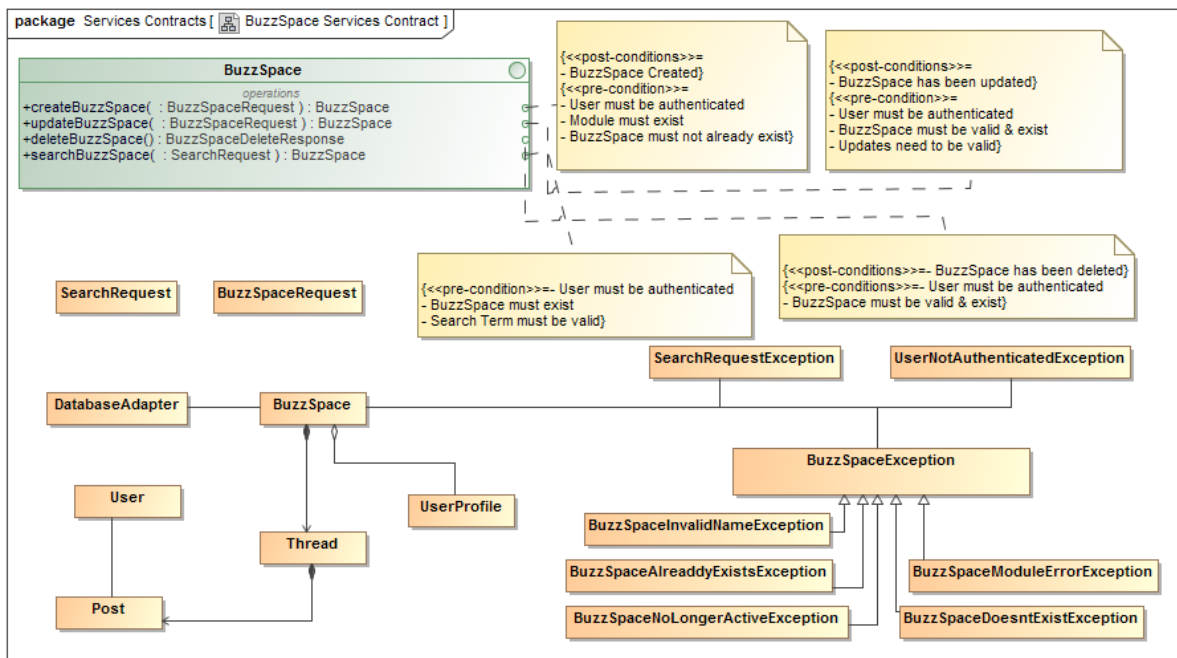


Figure 4: BuzzSpace services contracts.

2.2.4 Communication

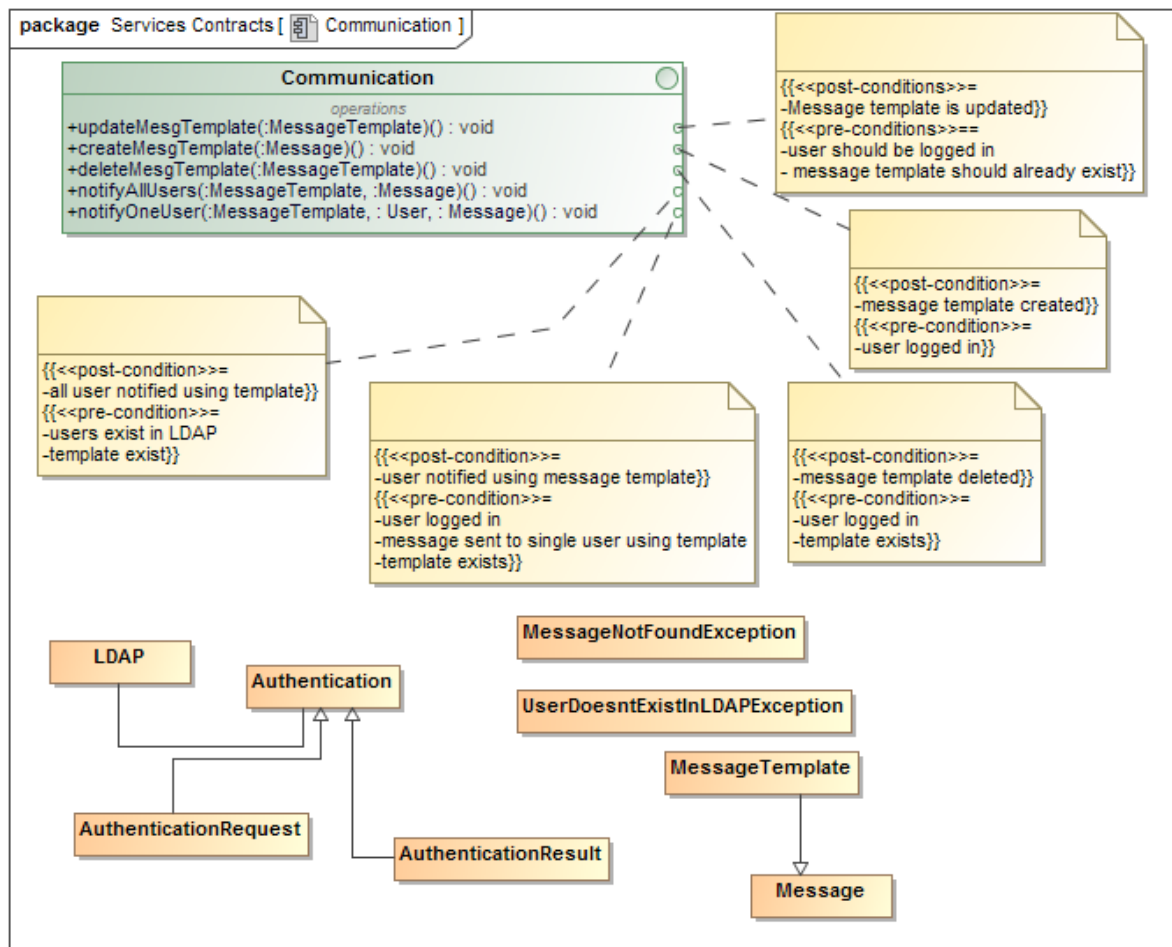


Figure 5: Communication services contracts.

2.2.5 Plagiarism / Netetiquette

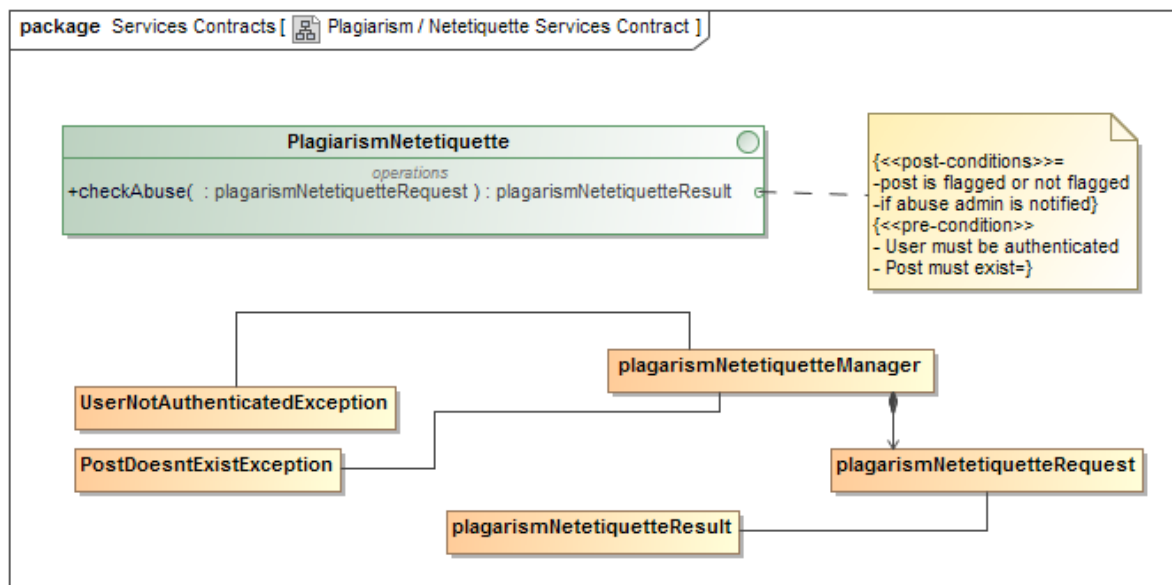


Figure 6: Plagiarism and Netetiquette services contracts.

2.2.6 Tagging

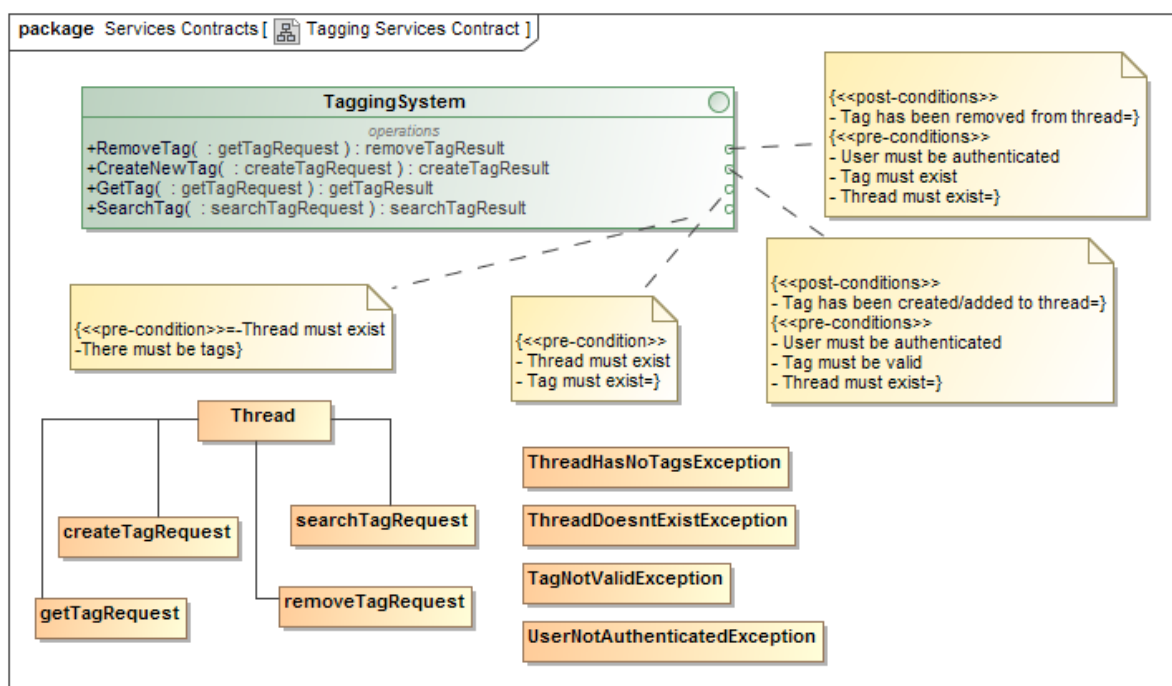


Figure 7: Tagging services contracts.

2.2.7 Thread

IMAGE OF ThreadSC

Figure 8: Tagging services contracts.

2.2.8 Thread Posts

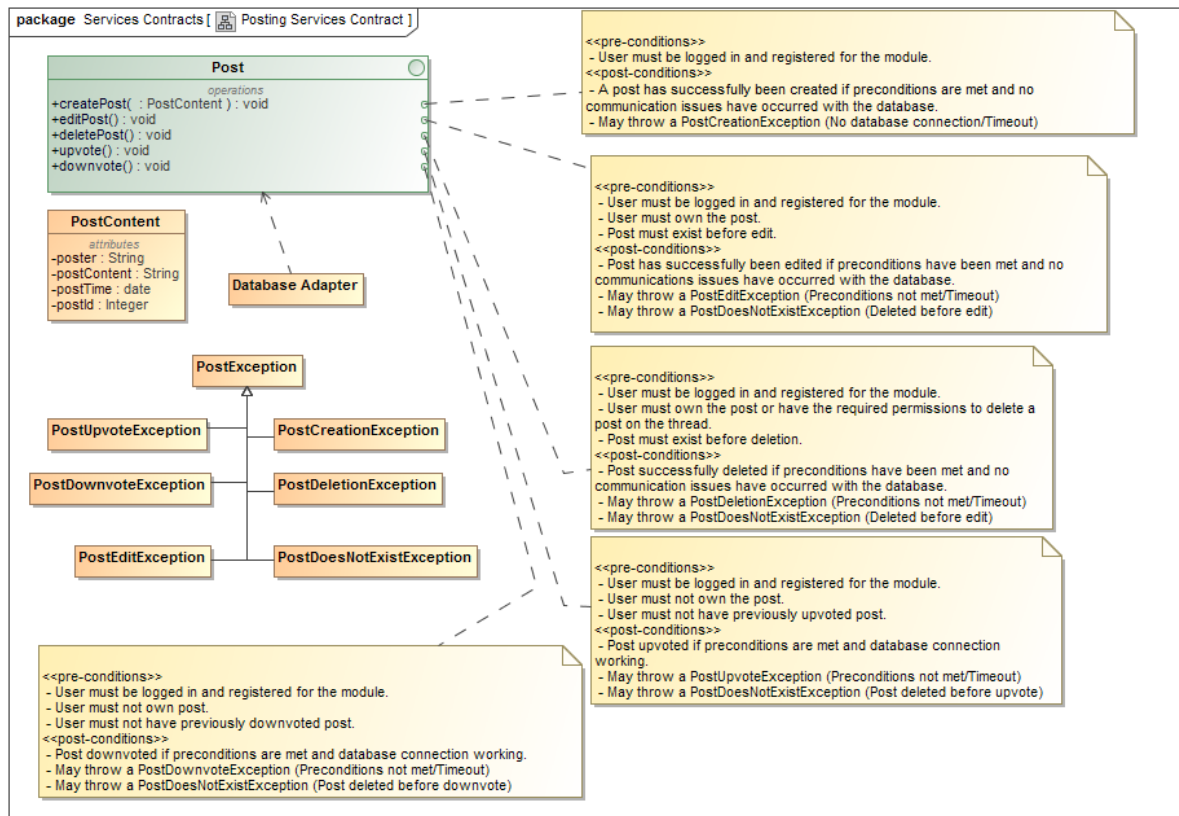


Figure 9: Thread Posts services contracts.

2.2.9 User

IMAGE OF UserSC

Figure 10: Tagging services contracts.

2.3 Required functionality

2.3.1 Analytics

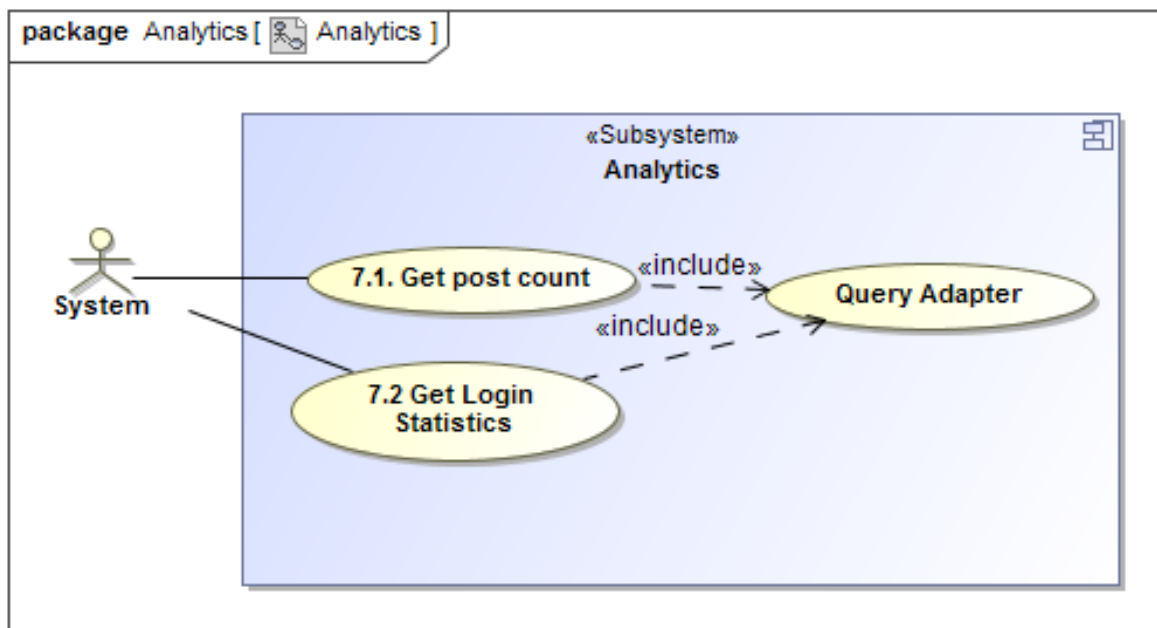


Figure 11: Analytics sub system use case diagram.

2.3.2 Authentication

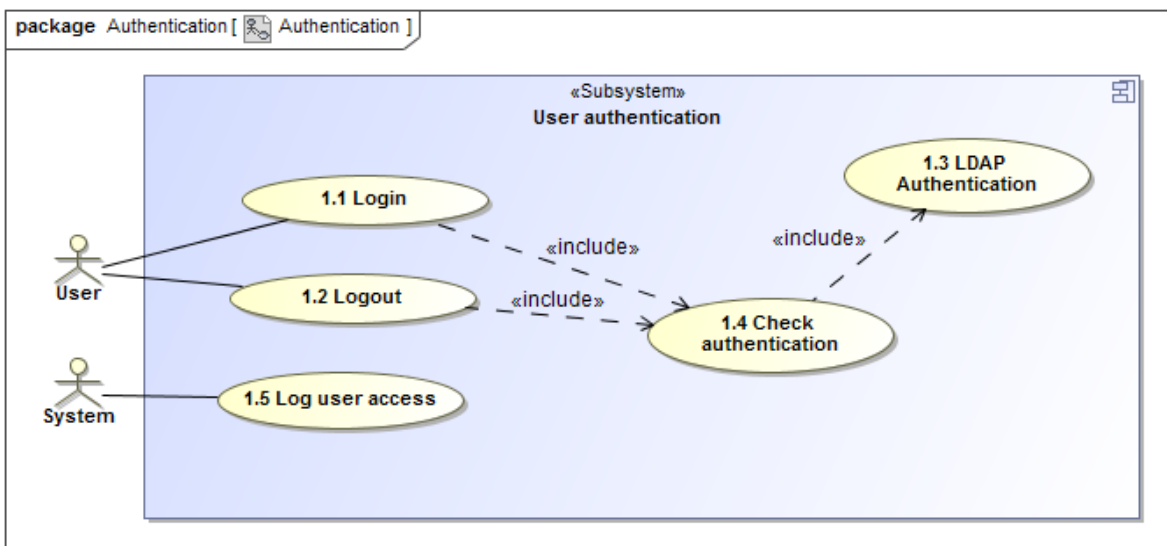


Figure 12: Authentication sub system use case diagram.

2.3.3 BuzzSpace

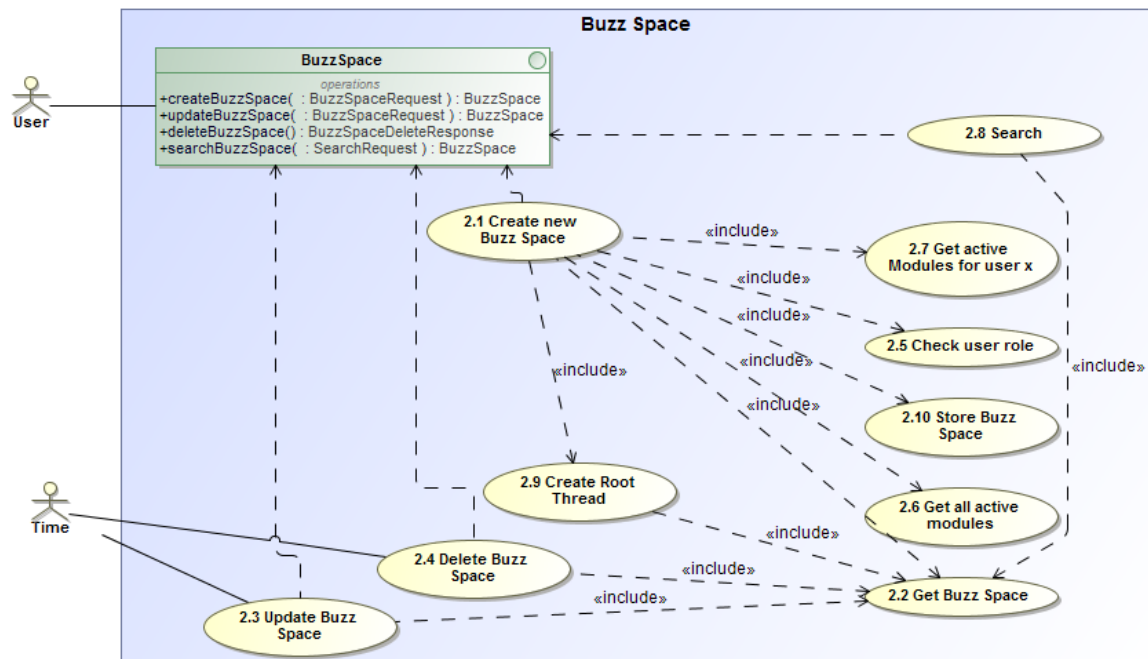


Figure 13: BuzzSpace sub system use case diagram.

2.3.4 Communication

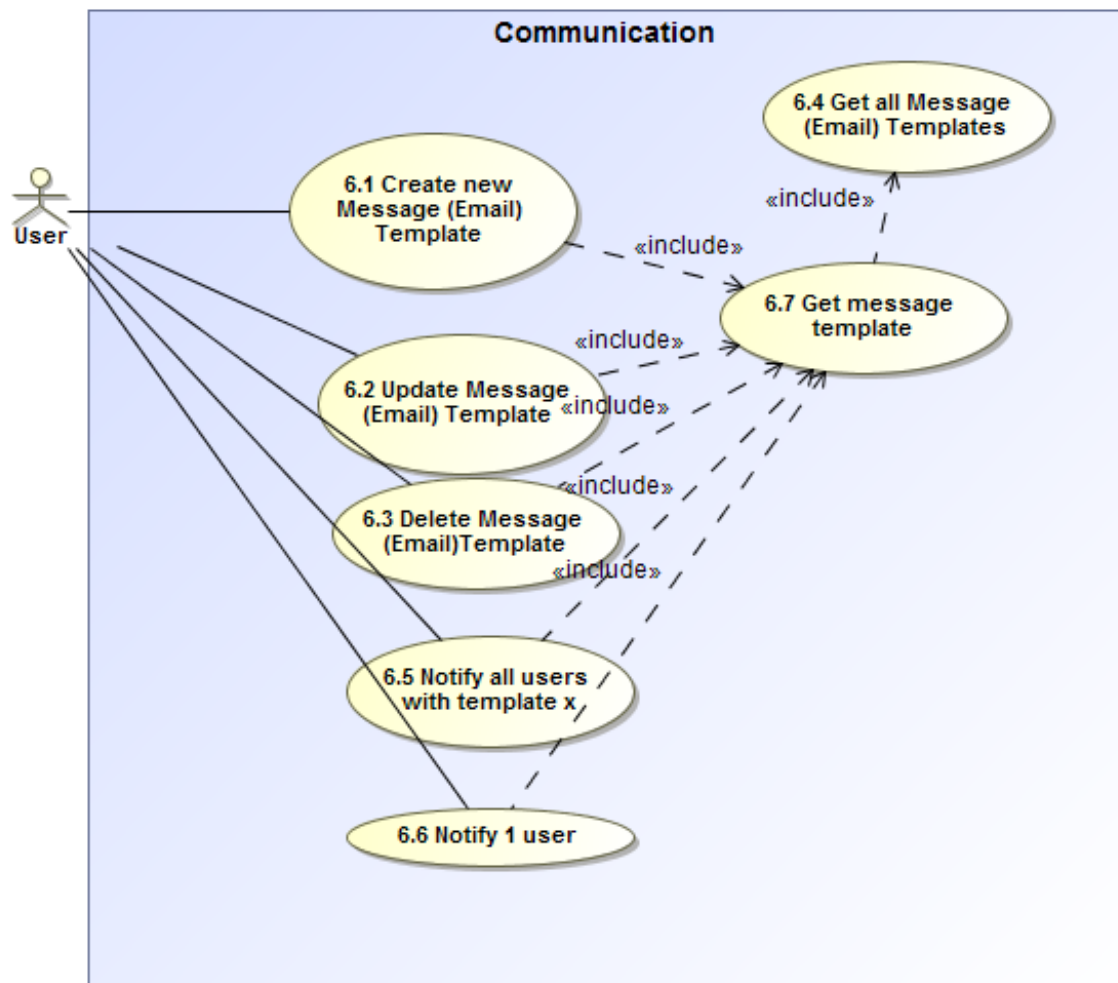


Figure 14: Communication sub system use case diagram.

2.3.5 Plagiarism / Netetiquette

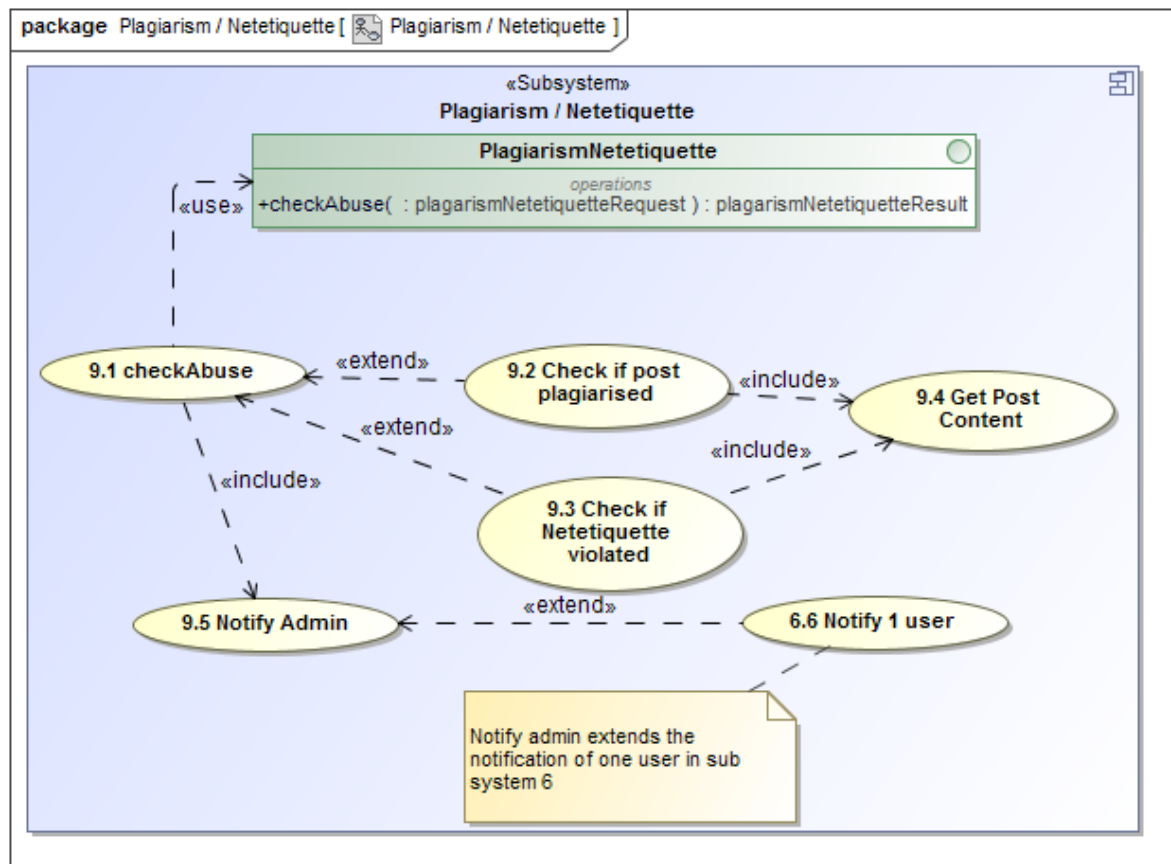


Figure 15: Plagiarism and Netetiquette sub system use case diagram.

2.3.6 Tagging

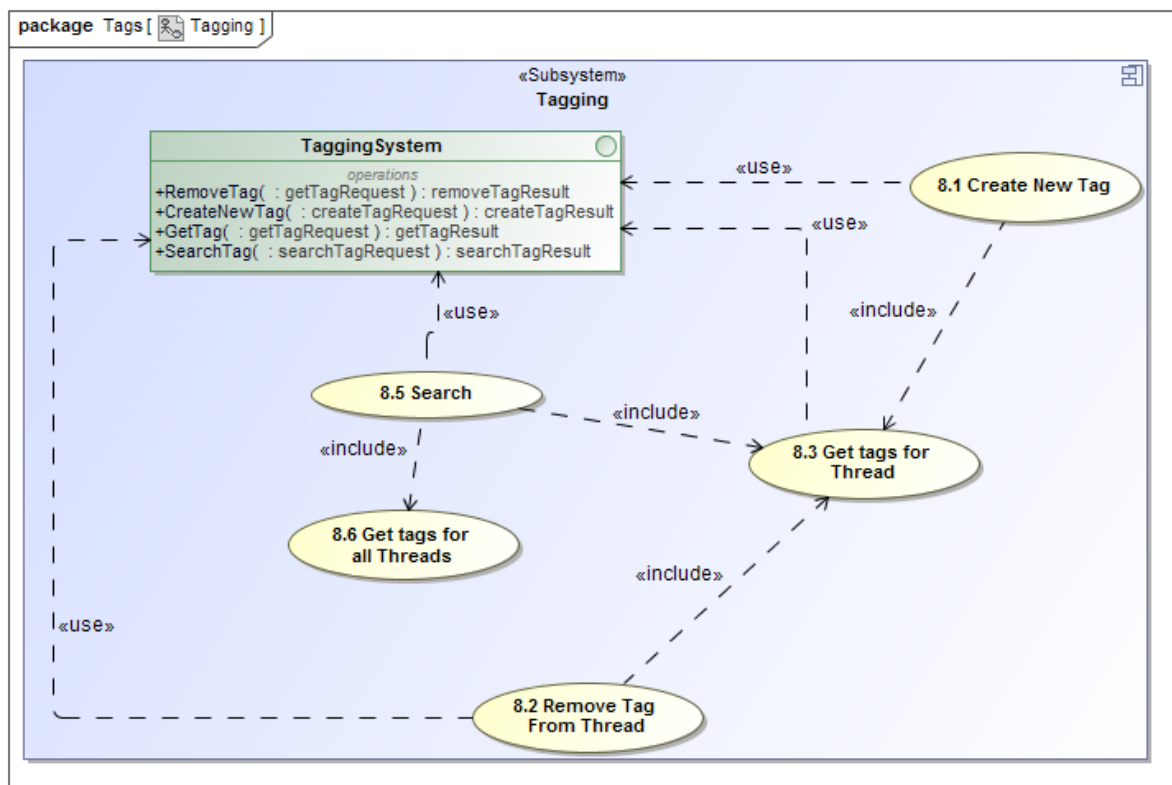


Figure 16: Tagging sub system use case diagram.

2.3.7 Thread

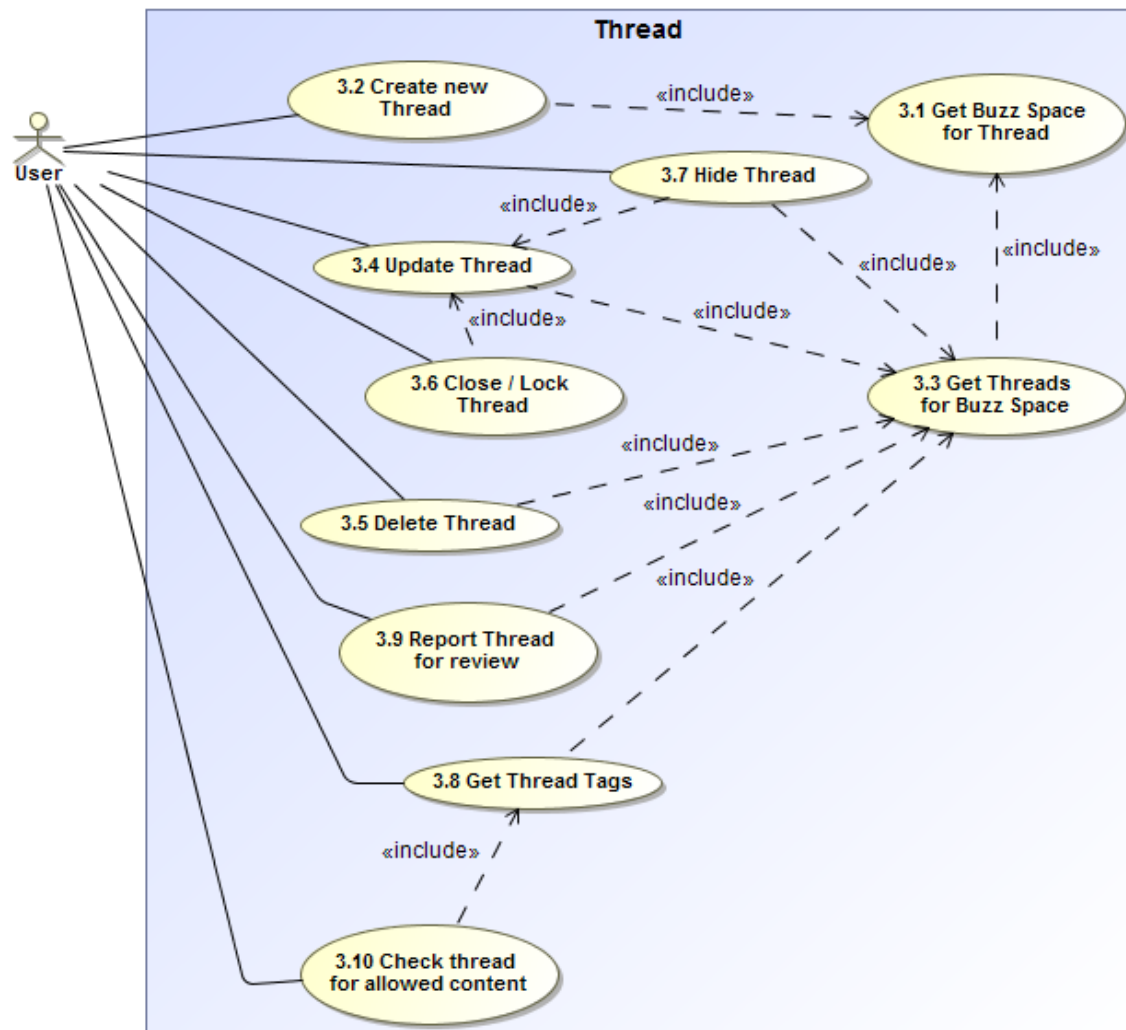


Figure 17: Tagging sub system use case diagram.

2.3.8 Thread Posts

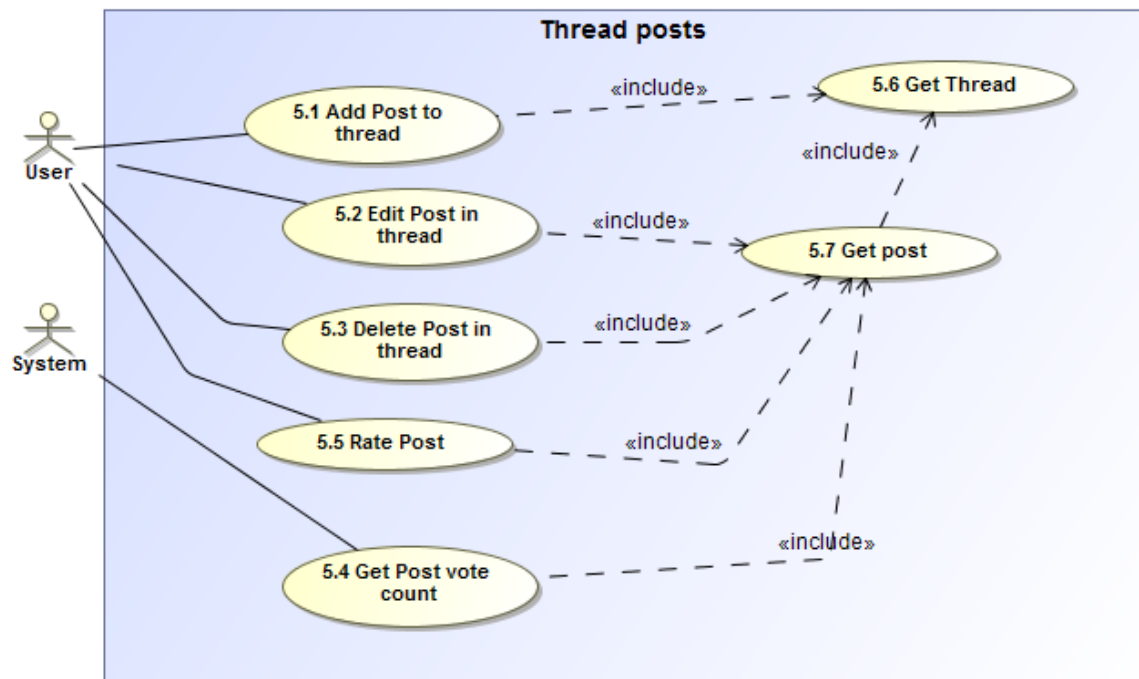


Figure 18: Thread Posts sub system use case diagram.

2.3.9 User

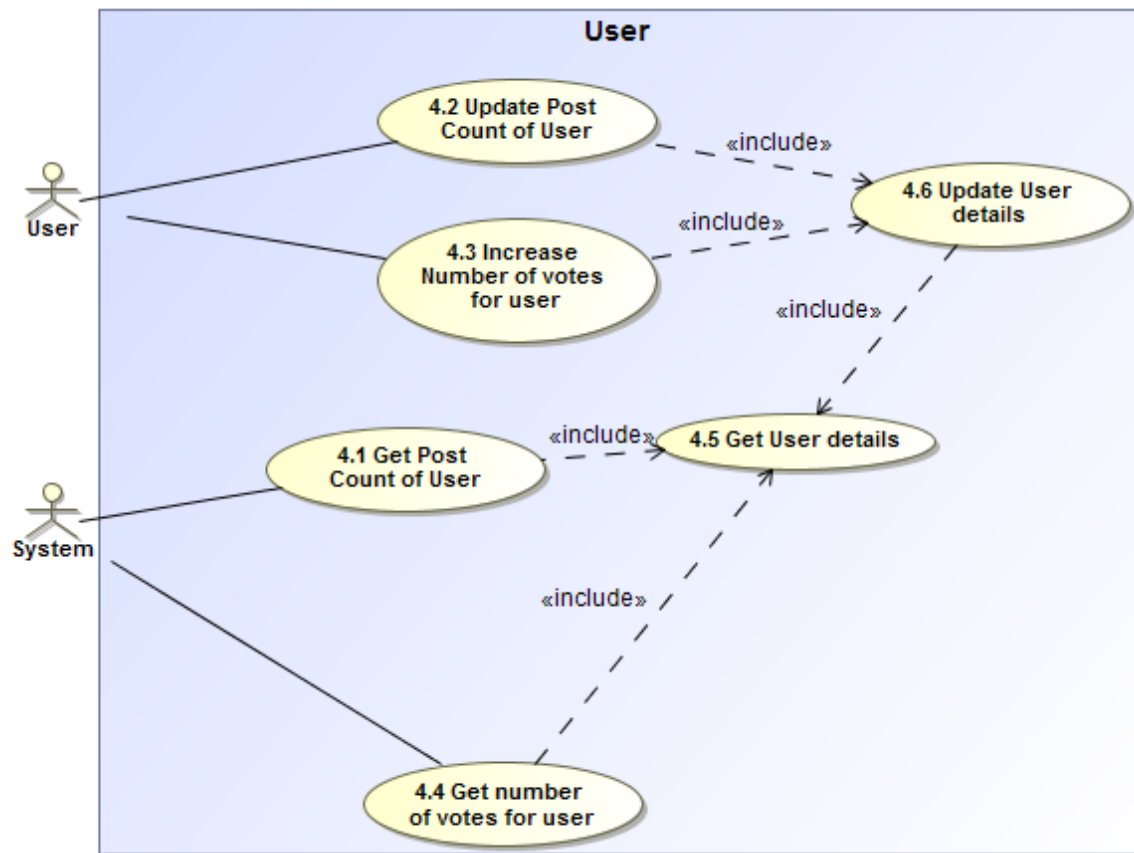


Figure 19: Tagging sub system use case diagram.

2.4 Process specifications

2.4.1 Authentication

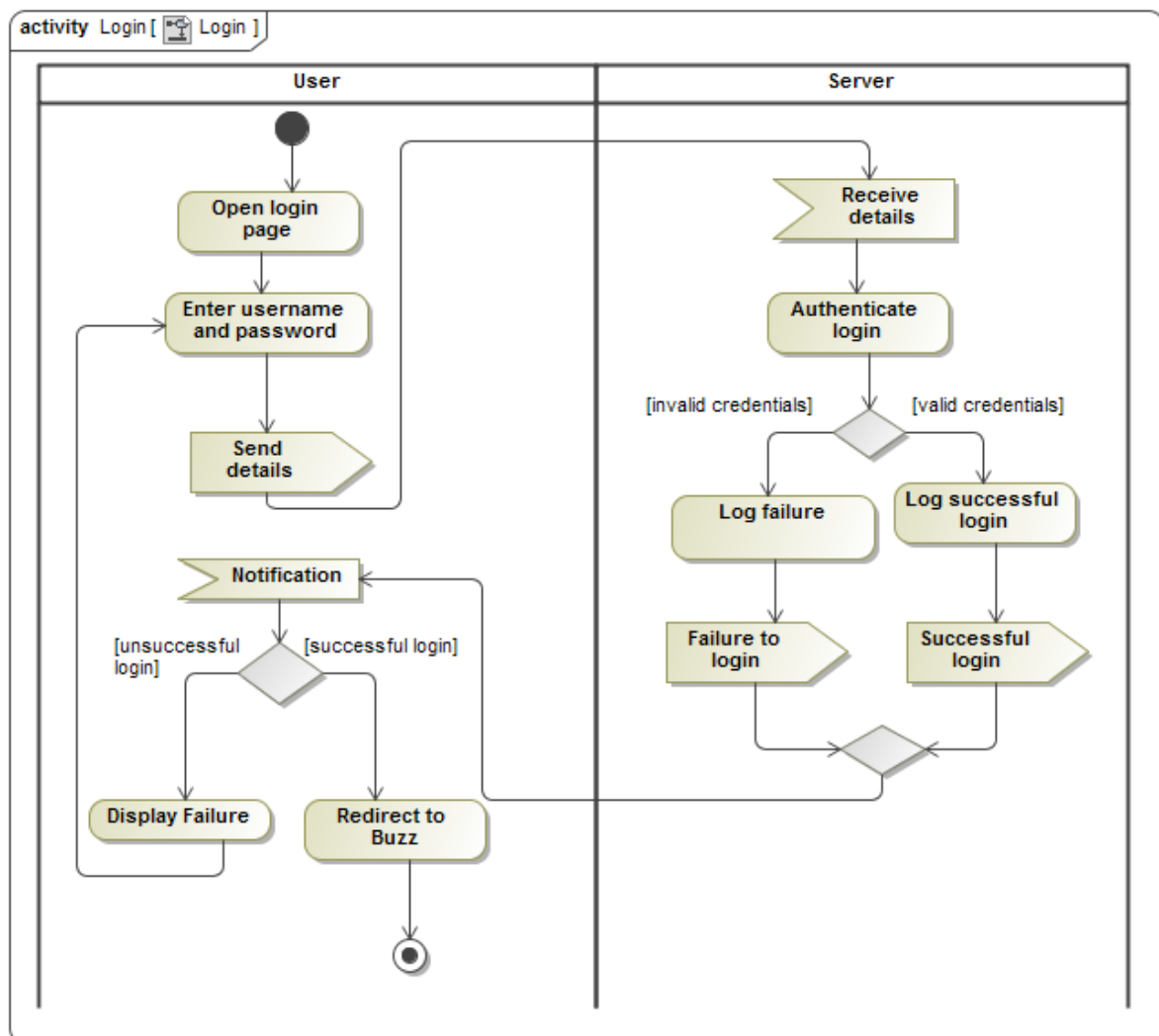


Figure 20: Login activity diagram.

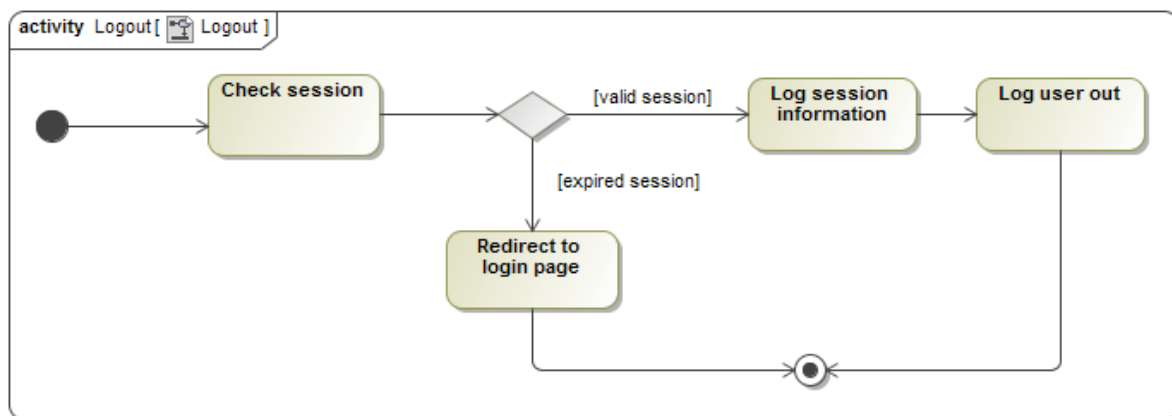


Figure 21: Logout activity diagram.

2.4.2 BuzzSpace

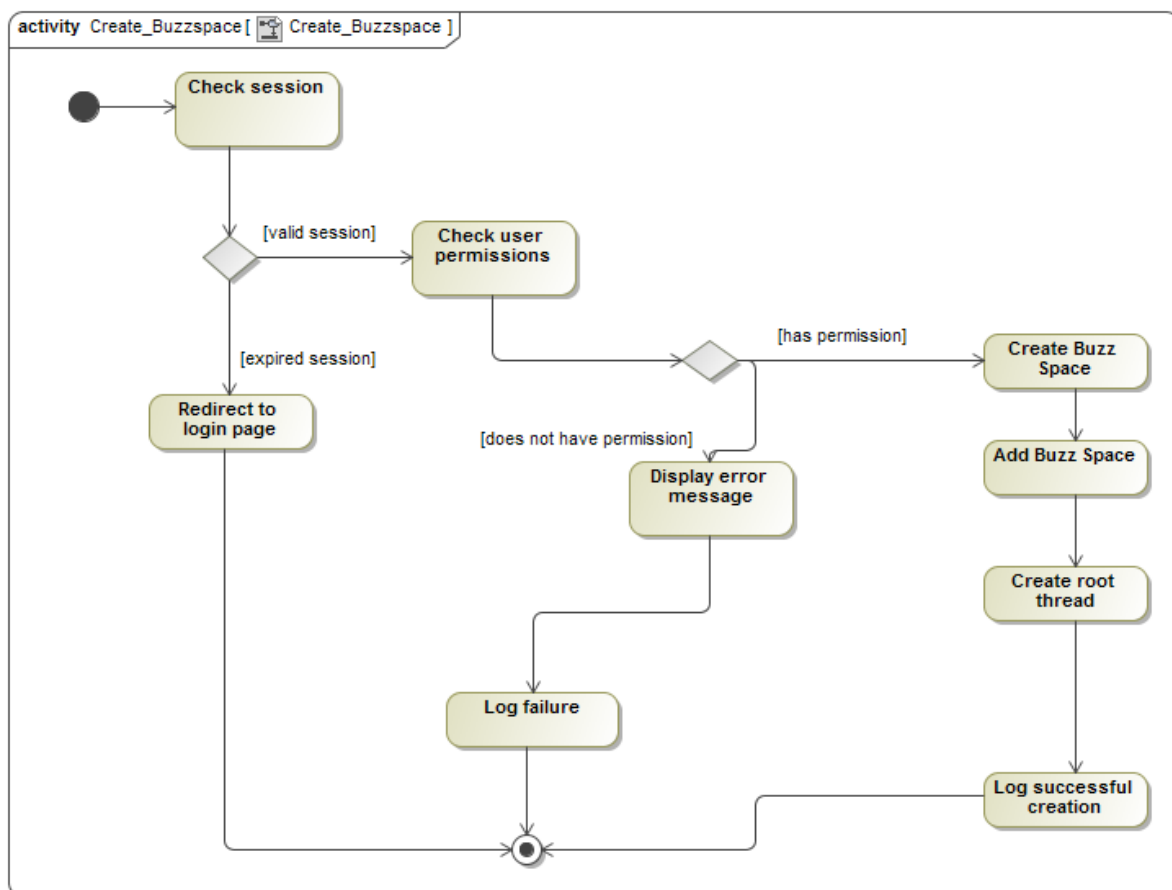


Figure 22: Create Buzzspace activity diagram.

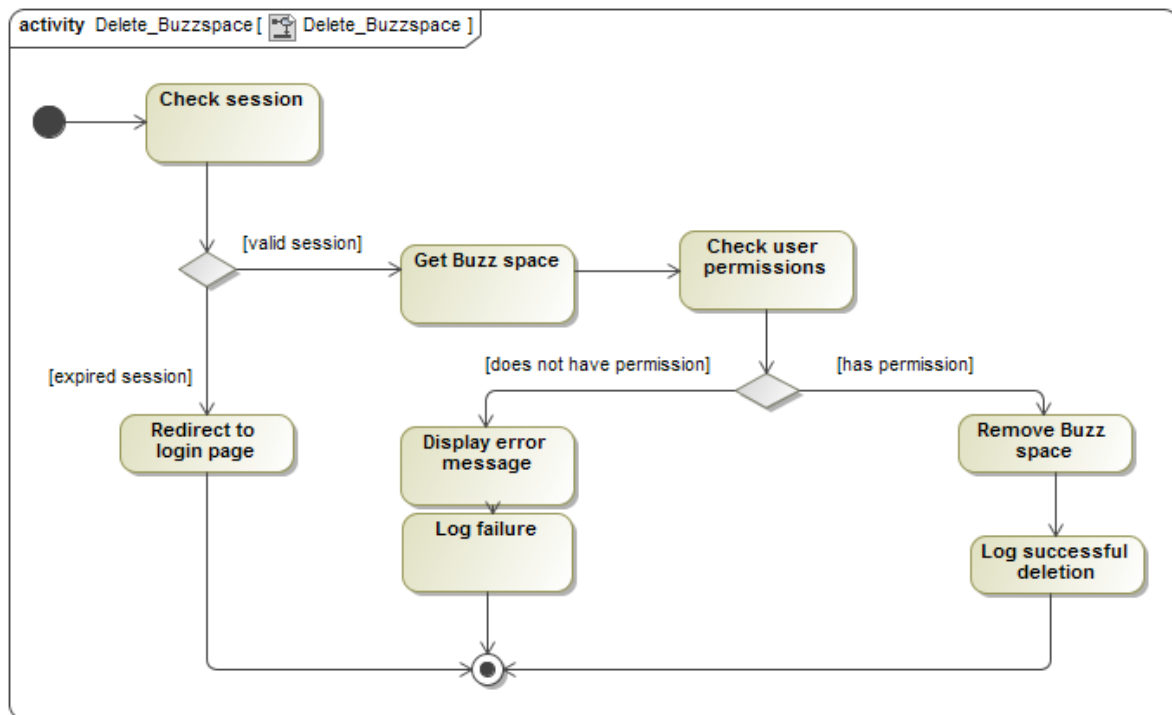


Figure 23: Delete Buzzspace activity diagram.

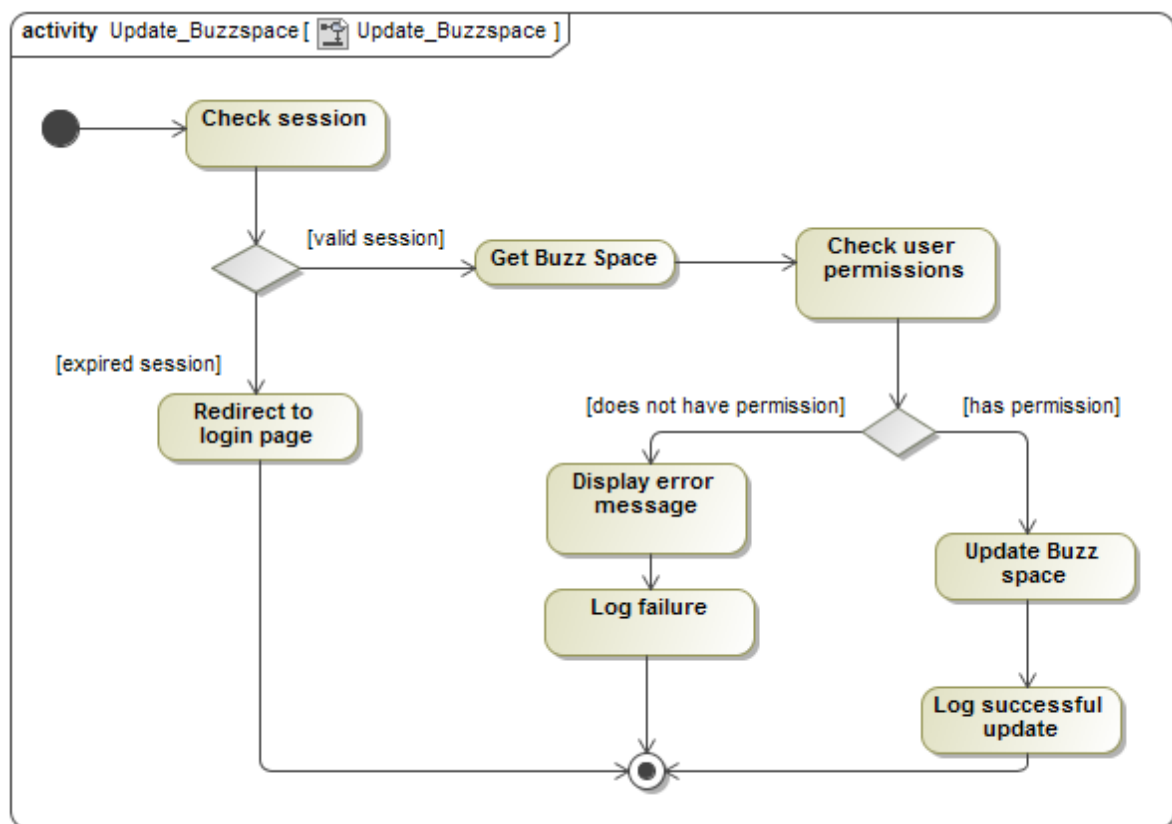


Figure 24: Update Buzzspace activity diagram.

2.4.3 Communication

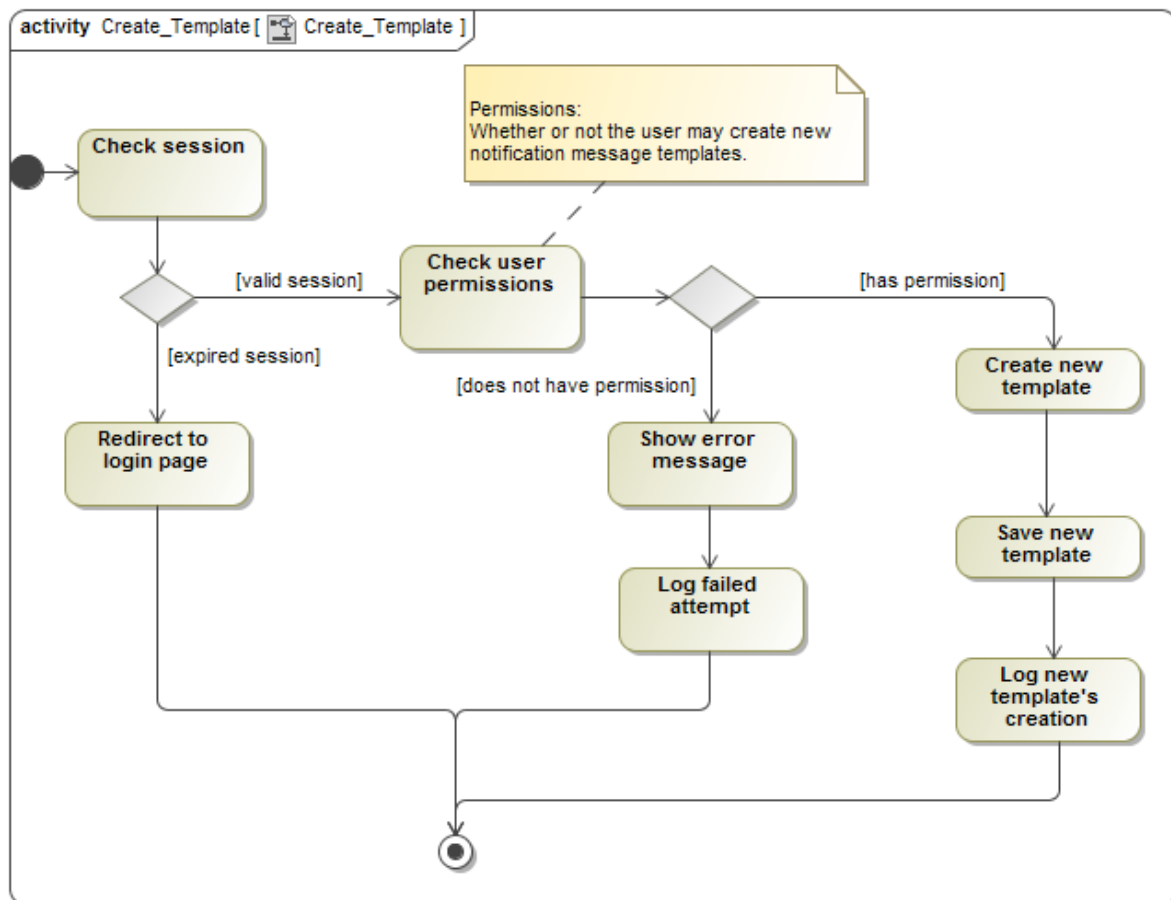


Figure 25: Create Template activity diagram.

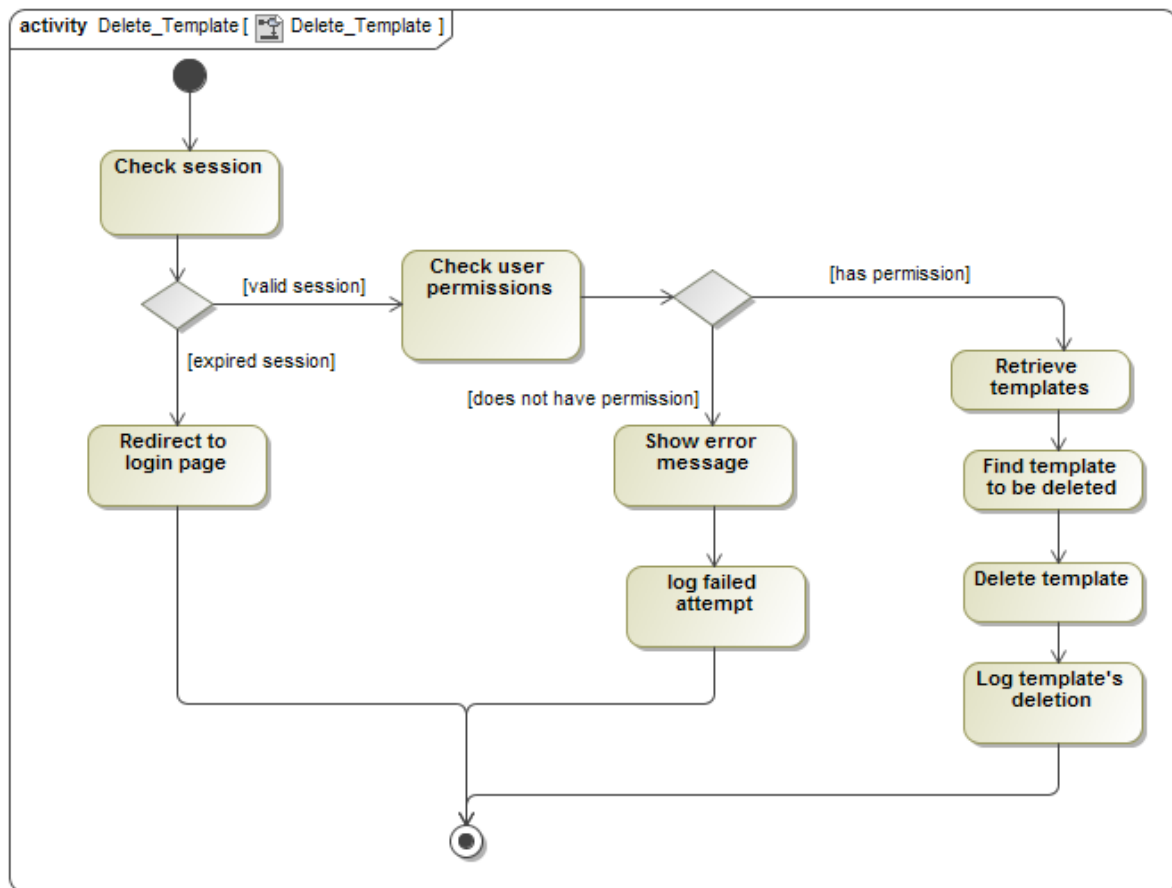


Figure 26: Delete Template activity diagram.

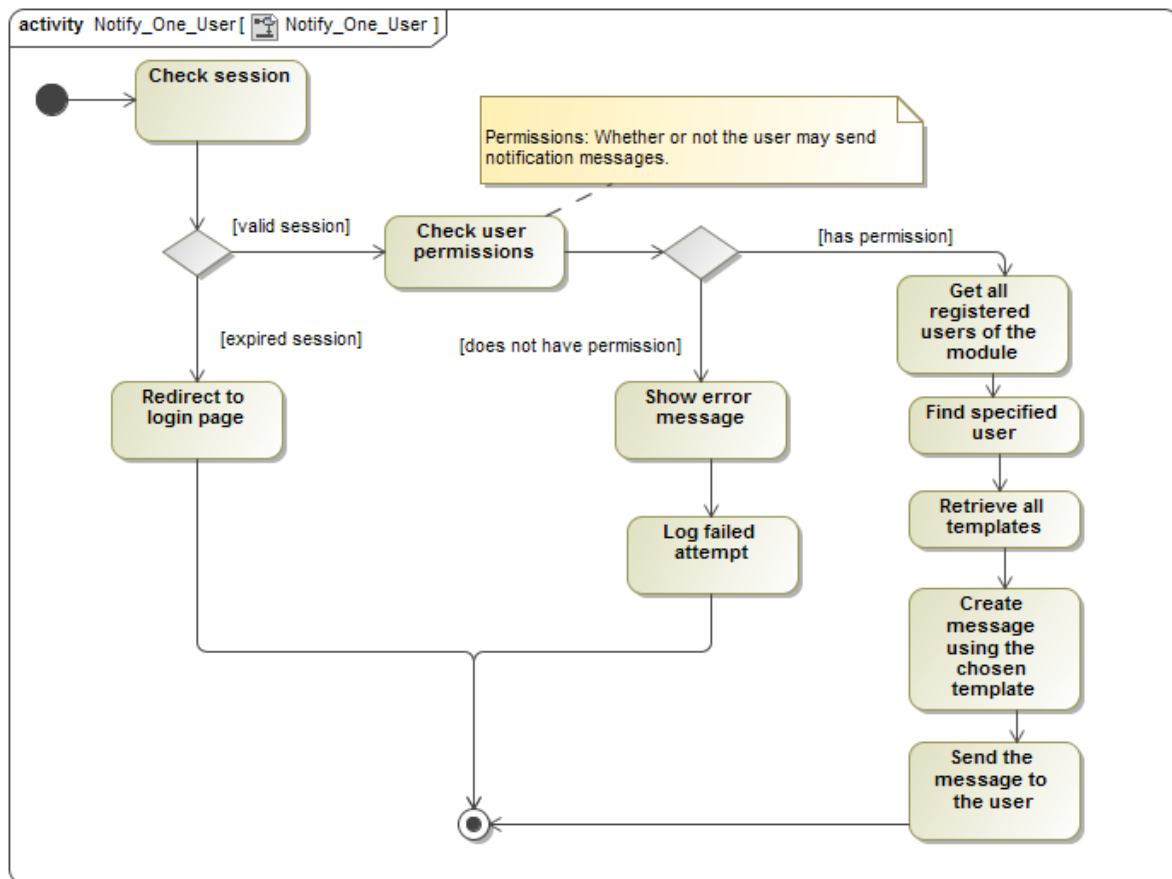


Figure 27: Notify One User activity diagram.

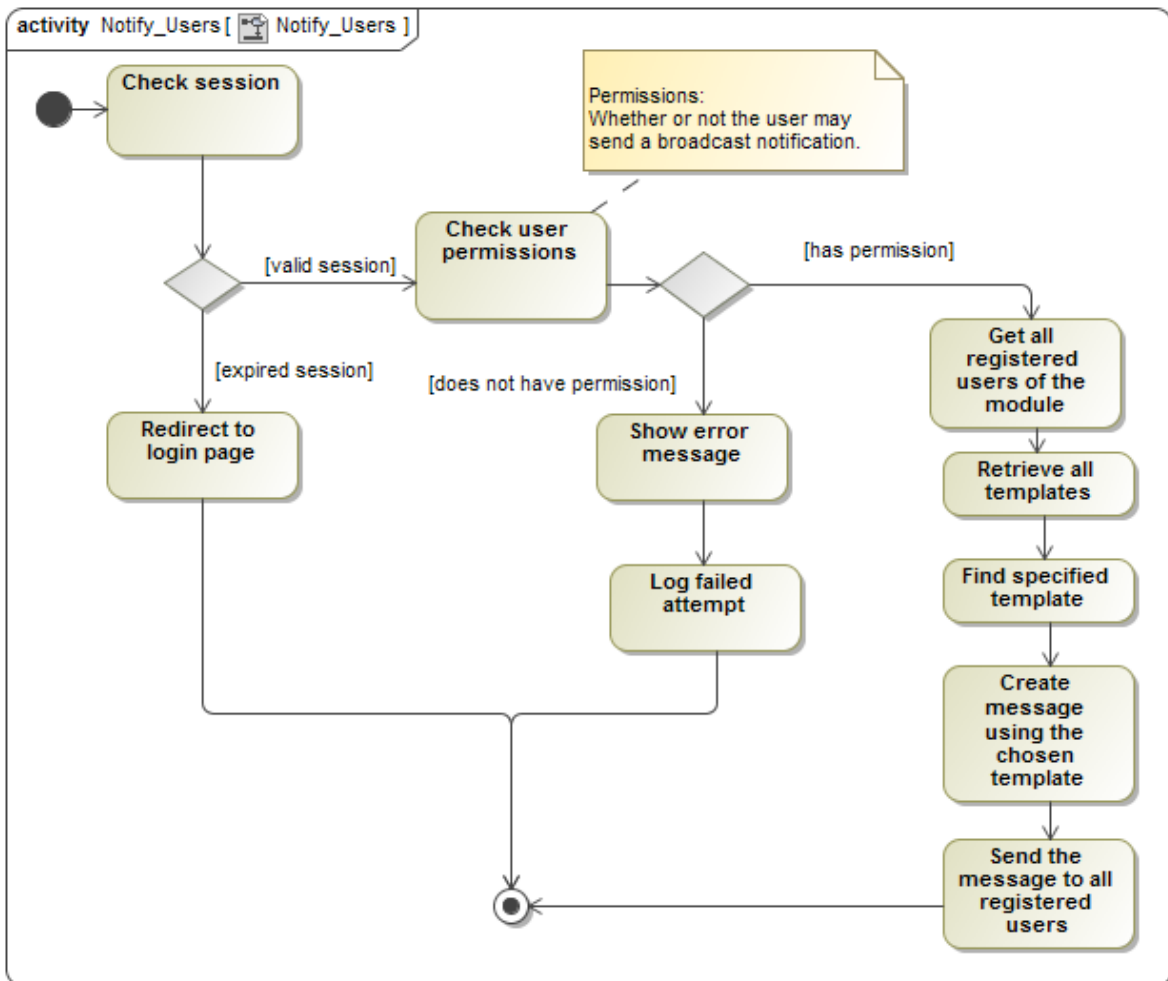


Figure 28: Notify Users activity diagram.

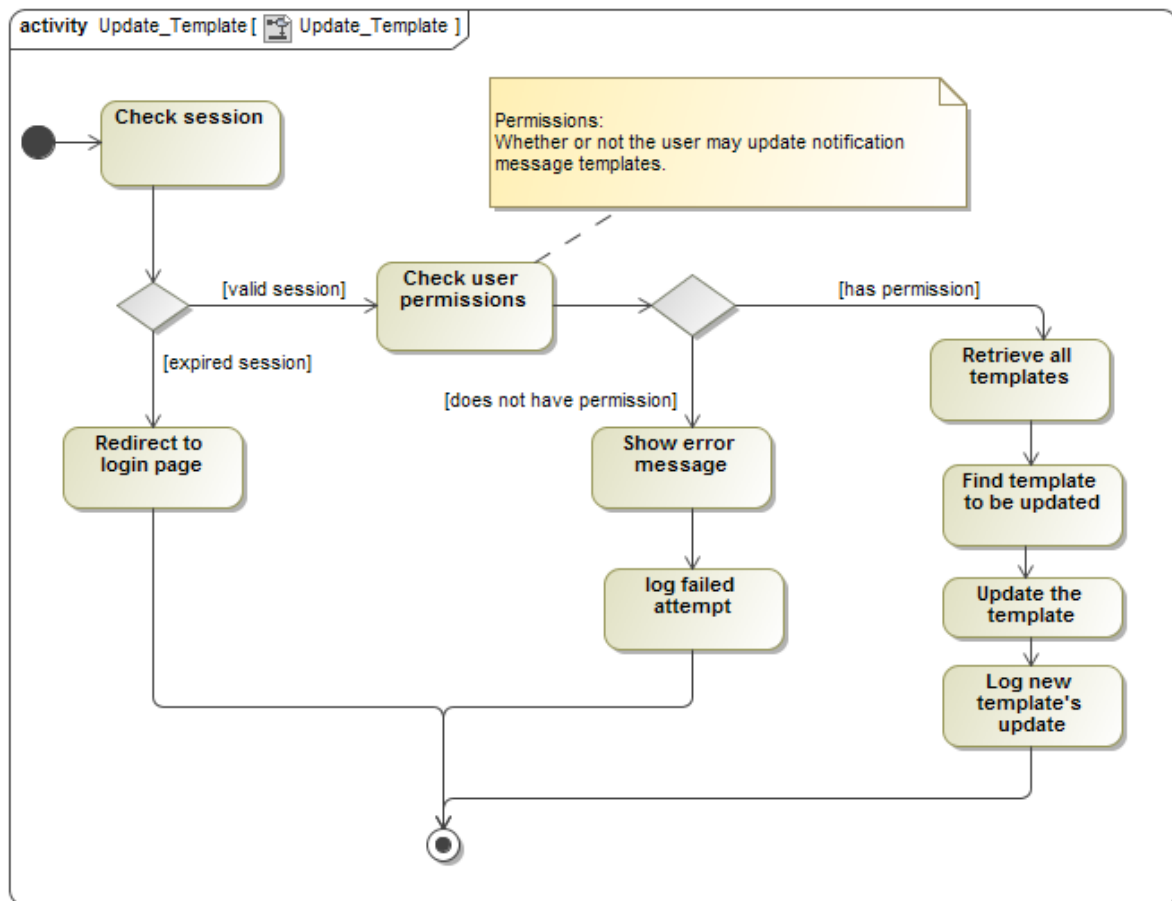


Figure 29: Update Template activity diagram.

2.4.4 Tagging

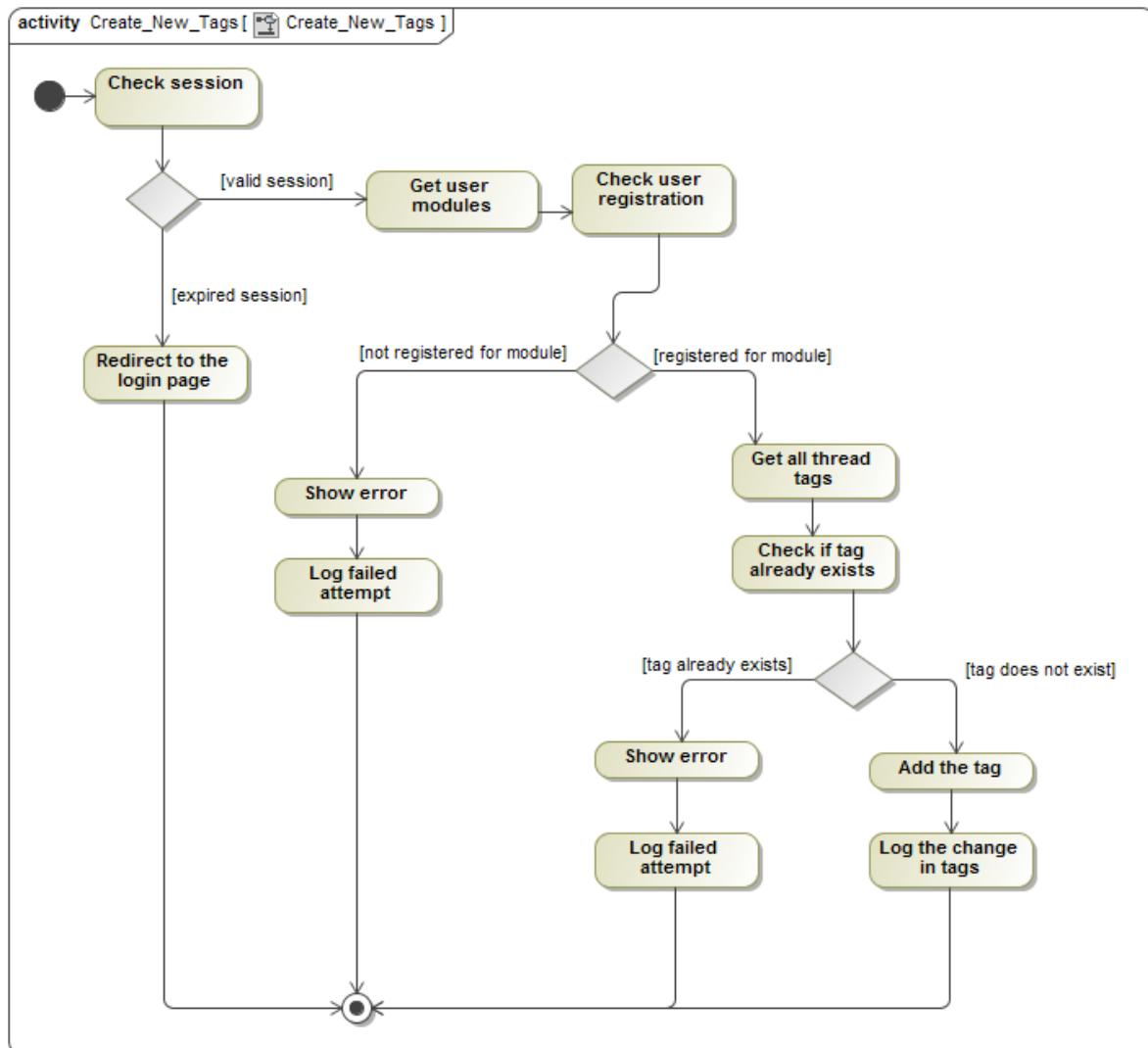


Figure 30: Create New Tags activity diagram.

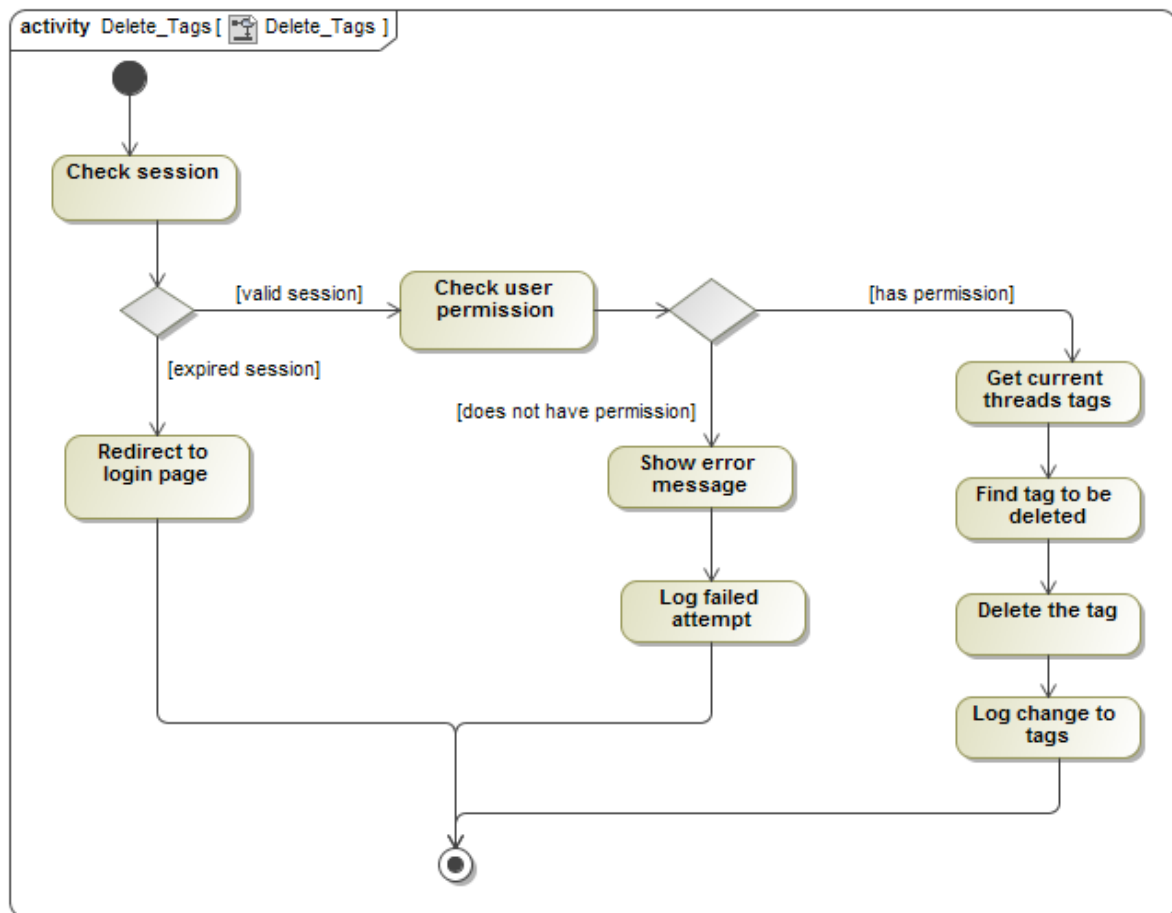


Figure 31: Delete Tags activity diagram.

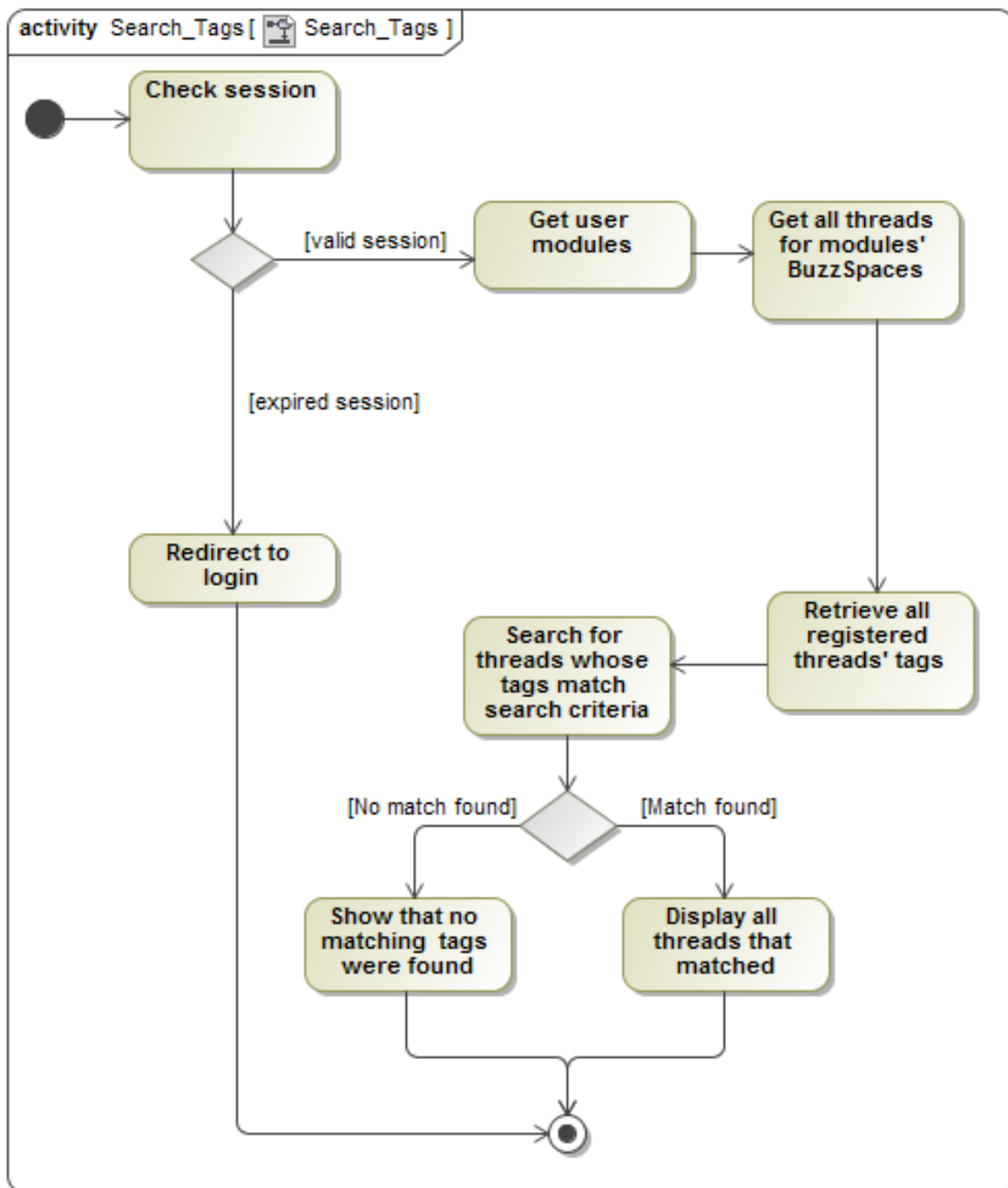


Figure 32: Search Tags activity diagram.

2.4.5 Thread

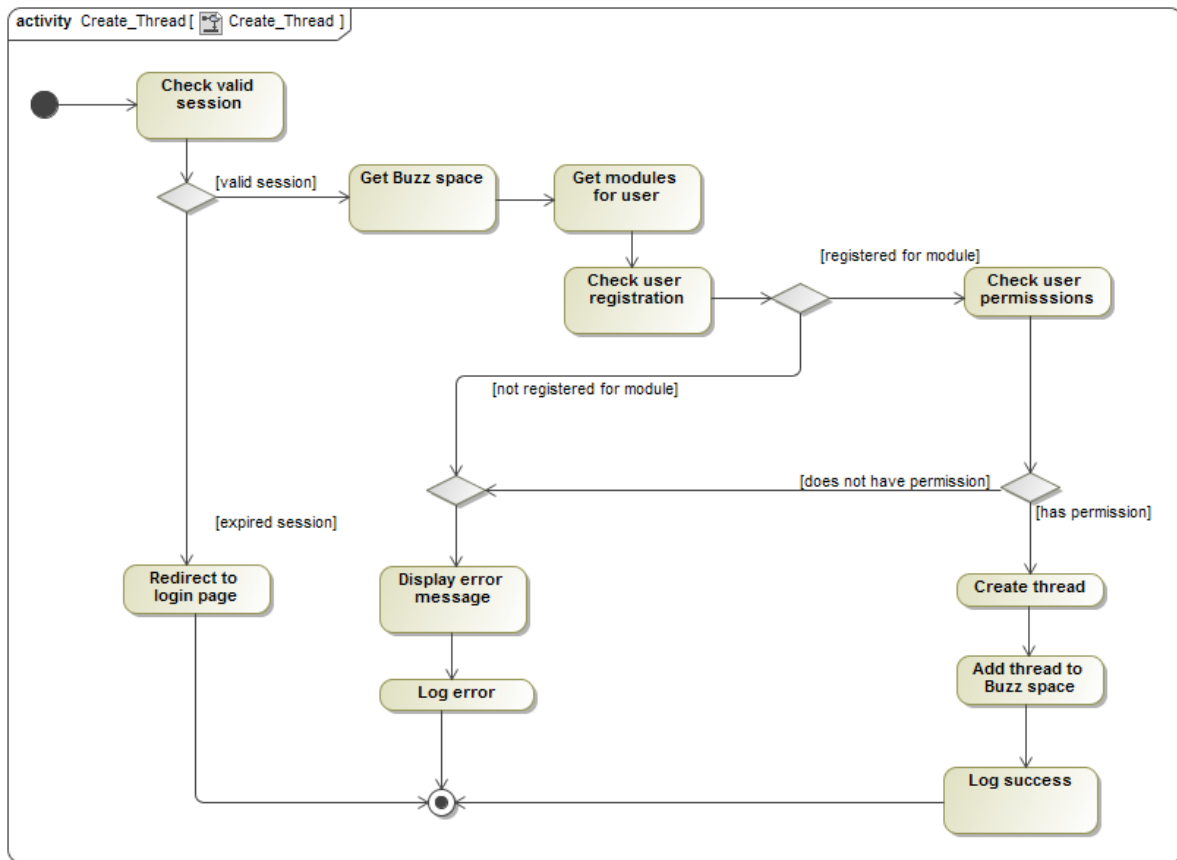


Figure 33: Create thread activity diagram.

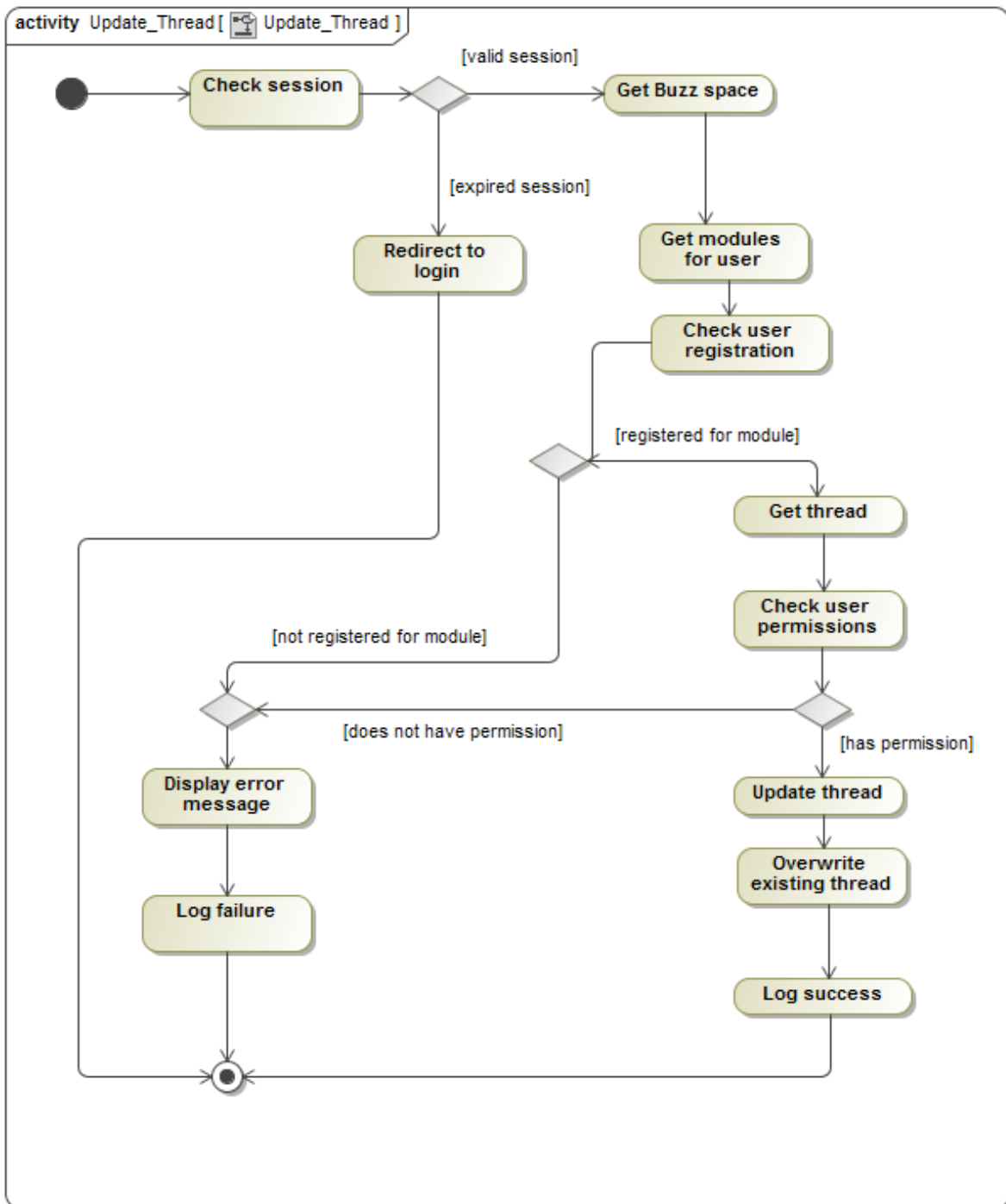


Figure 34: Update thread activity diagram.

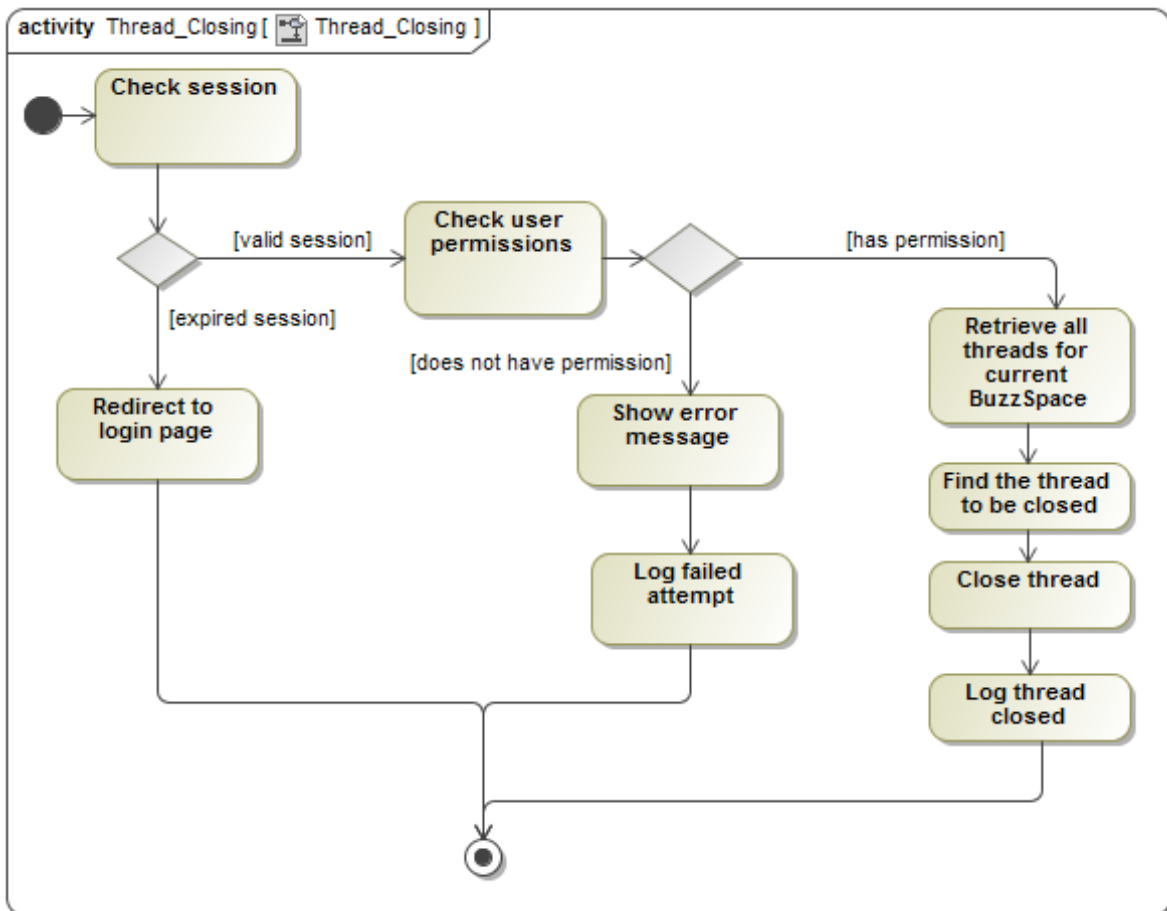


Figure 35: Close thread diagram.

2.4.6 Thread posts

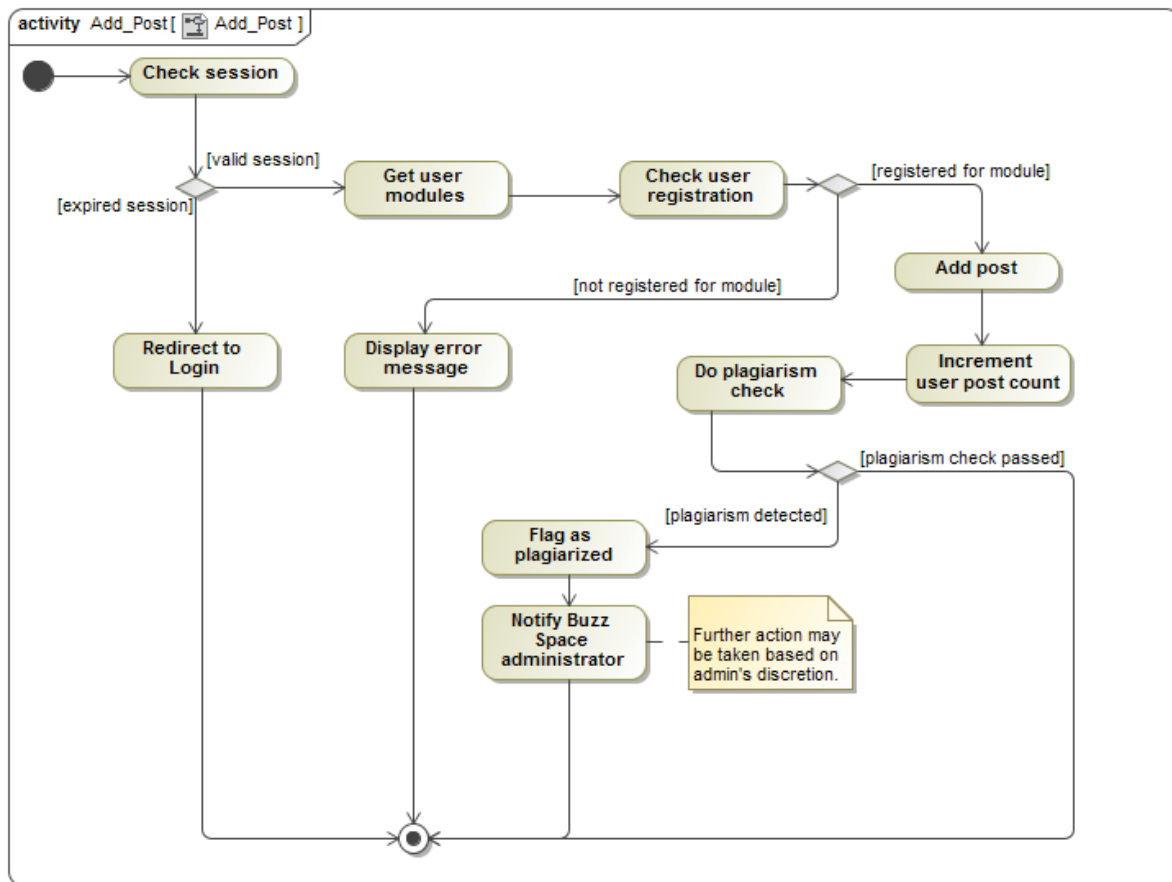


Figure 36: Add post to thread activity diagram.

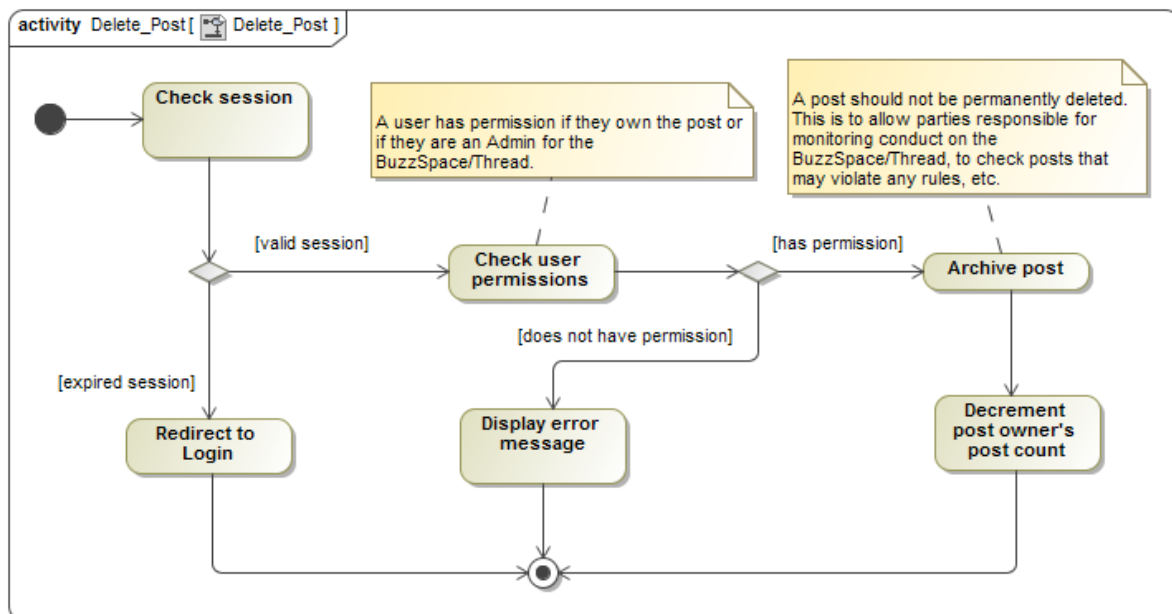


Figure 37: Delete thread post activity diagram.

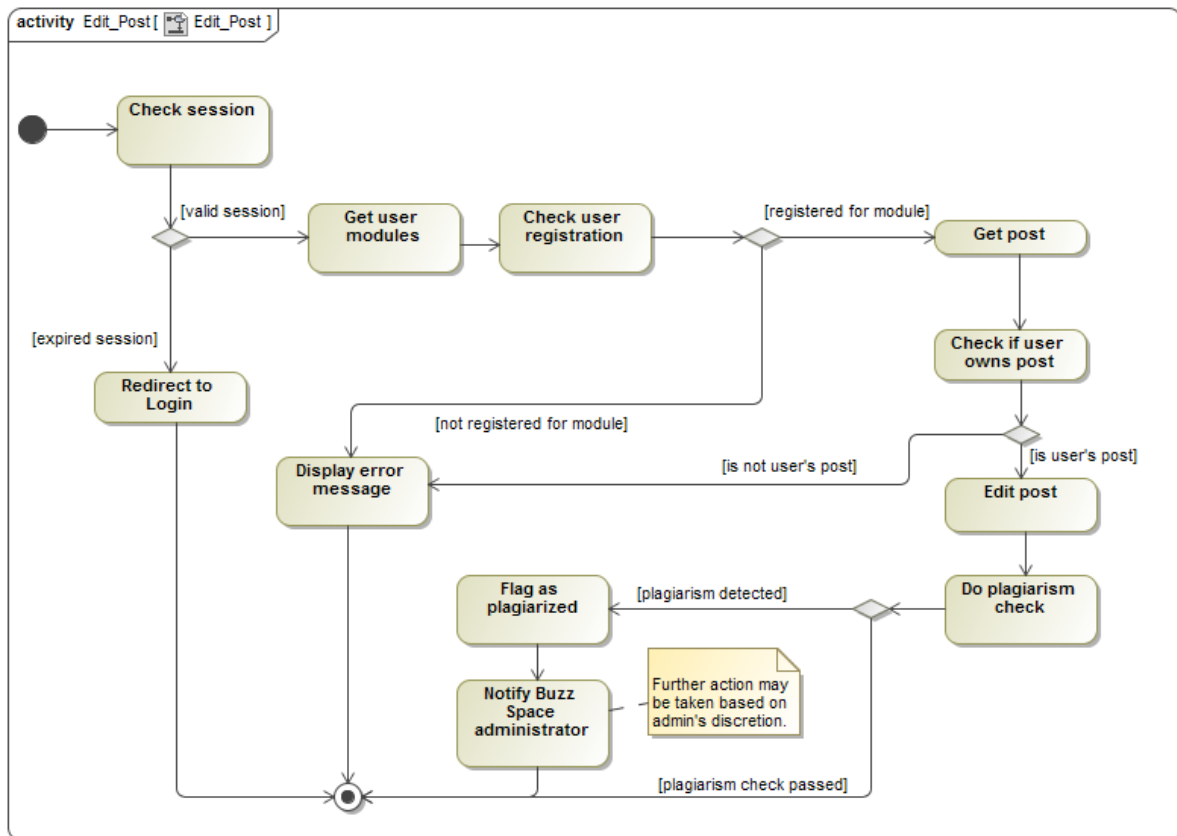


Figure 38: Edit thread post activity diagram.

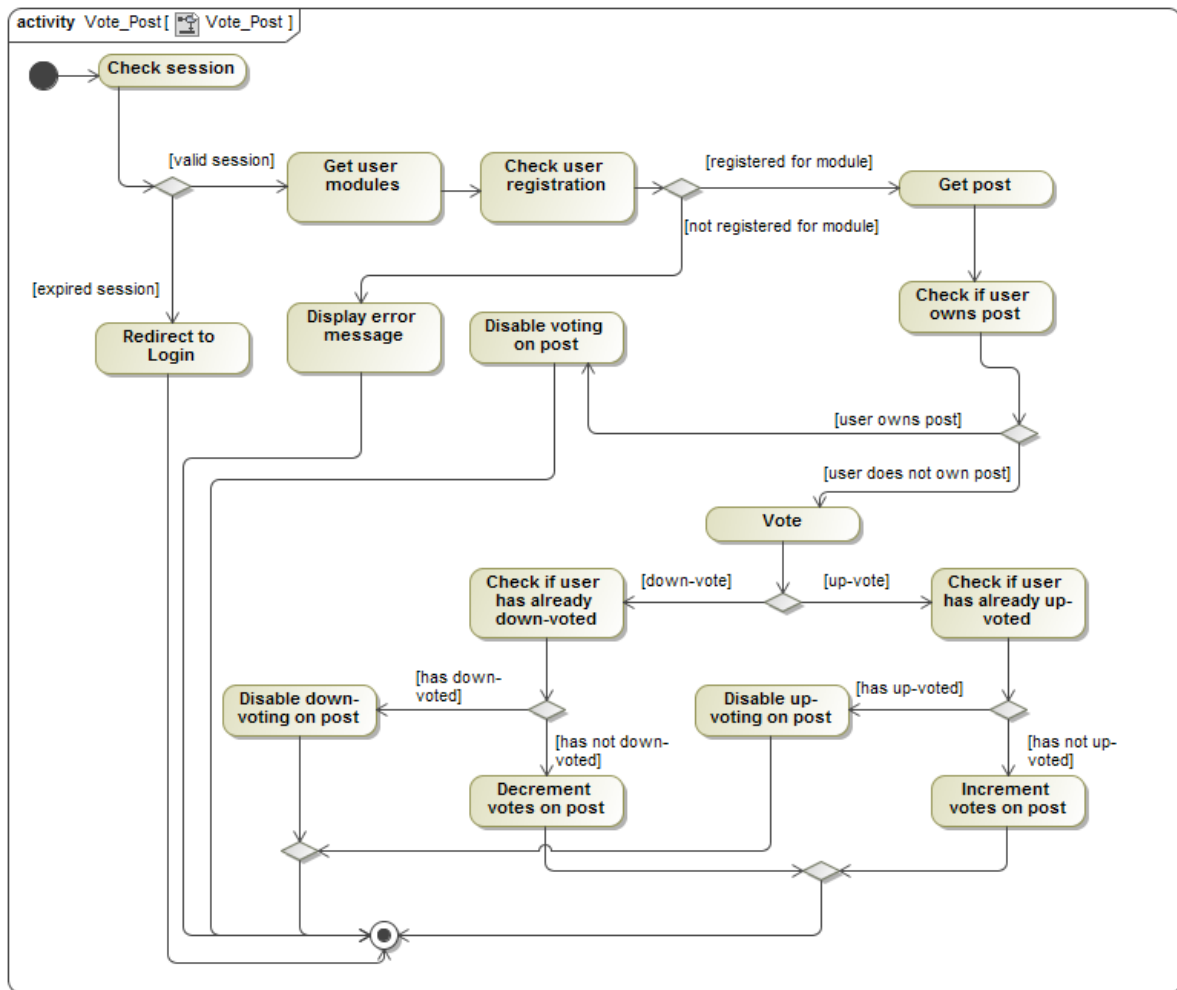


Figure 39: Vote for post activity diagram.

2.5 Domain Model

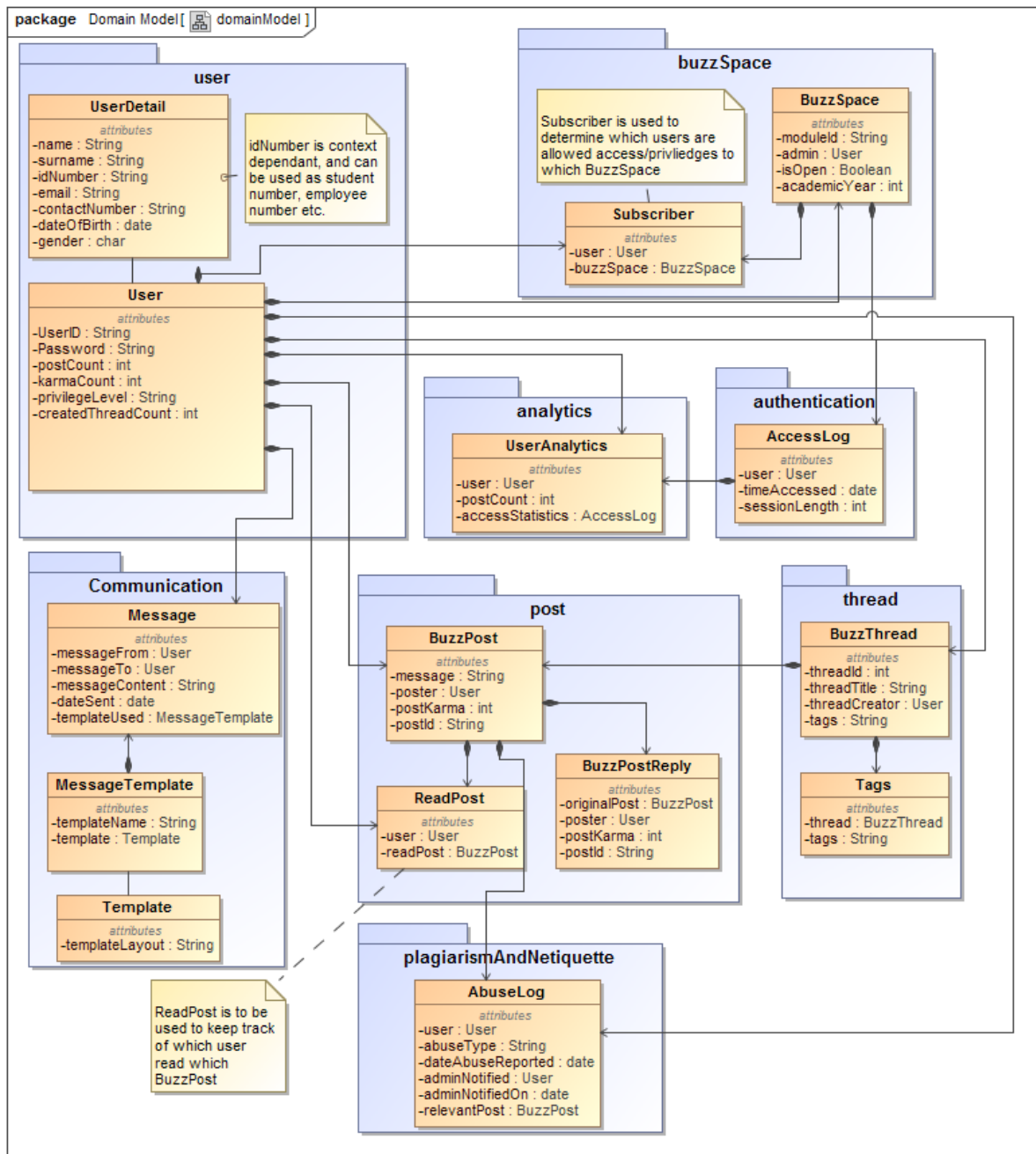


Figure 40: Domain model for the discussion board.