

COS 301 - MINI PROJECT

FUNCTIONAL REQUIREMENTS AND
APPLICATION DESIGN

GROUP 1B

2015

Contents

1	Preface	2
1.1	General Information	2
1.2	Group members	2
1.3	Contributions to project	2
1.3.1	Who did what	2
2	Functional requirements and application design	3
2.1	Use case prioritization	3
2.1.1	Critical	3
2.1.2	Important	3
2.1.3	Nice-To-Have	3
2.2	Services contracts	4
2.2.1	Analytics	4
2.2.2	Authentication	4
2.2.3	BuzzSpace	5
2.2.4	Communication	6
2.2.5	Plagiarism / Netetiquette	7
2.2.6	Tagging	7
2.2.7	Thread	8
2.2.8	Thread Posts	9
2.2.9	User	10
2.3	Required functionality	11
2.3.1	Analytics	11
2.3.2	Authentication	11
2.3.3	BuzzSpace	12
2.3.4	Communication	13
2.3.5	Plagiarism / Netetiquette	14
2.3.6	Tagging	15
2.3.7	Thread	16
2.3.8	Thread Posts	17
2.3.9	User	18
2.4	Process specifications	19
2.4.1	Authentication	19
2.4.2	BuzzSpace	20
2.4.3	Communication	22
2.4.4	Tagging	27
2.4.5	Thread	30
2.4.6	Thread posts	33
2.5	Domain Model	36

1 Preface

1.1 General Information

We used Github as our version control system. 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

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

1.3.1 Who did what

- Planning: Group effort
- Use case prioritization: Group effort
- Services contracts: Jaco-Louis Kruger(1), Daniel Arajo (2), Thinus Naude (4), Paul Engelke(1) and Elzahn Botha(1)
- Required functionality: Jaco-Louis Kruger(6) and Thinus Naude(3)
- Process specifications: Elzahn Botha(10), Daniel Arajo(6) and Paul Engelke(4)
- Domain Model: Frikkie Snyman(everything)
- LaTeX: Thinus Naude(everything)
- Modification and 'Error' Checks: Jaco-Louis Kruger, Daniel Arajo, Thinus Naude, Paul Engelke and Elzahn Botha

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

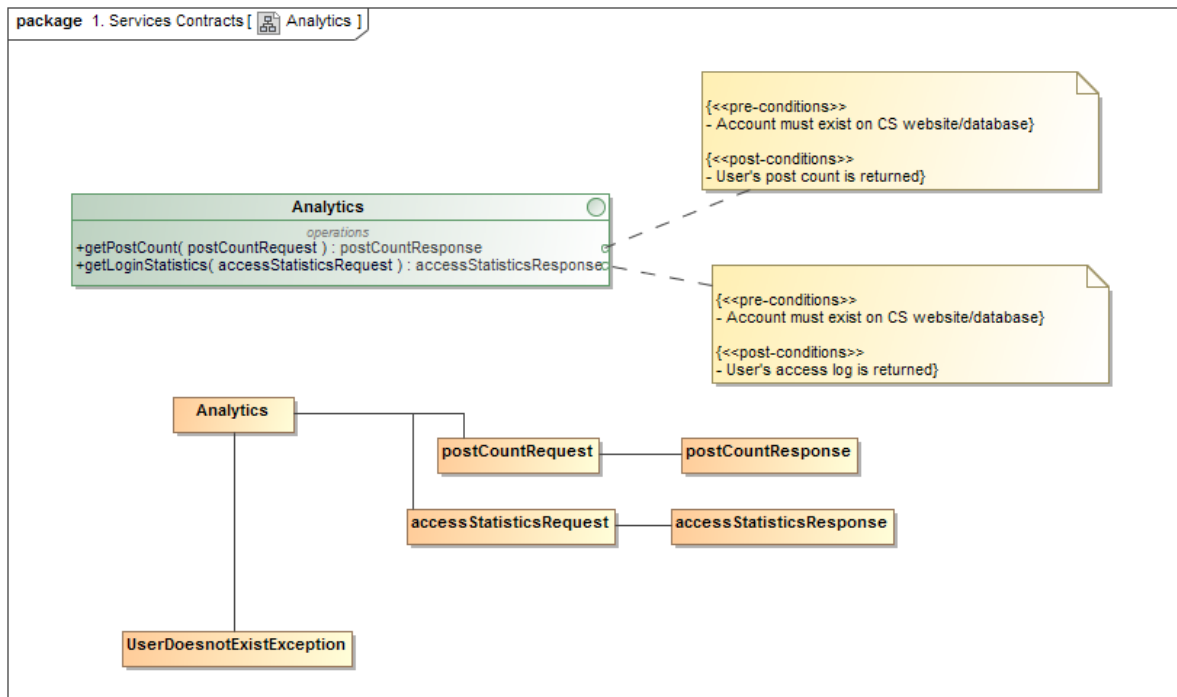


Figure 1: Analytics services contracts.

2.2.2 Authentication

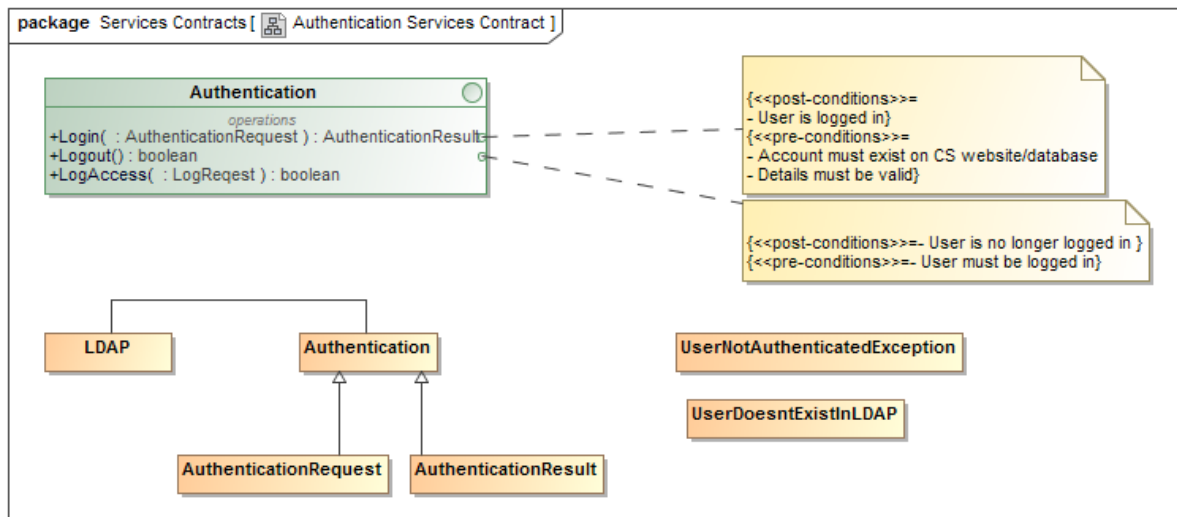


Figure 2: Authentication services contracts.

2.2.3 BuzzSpace

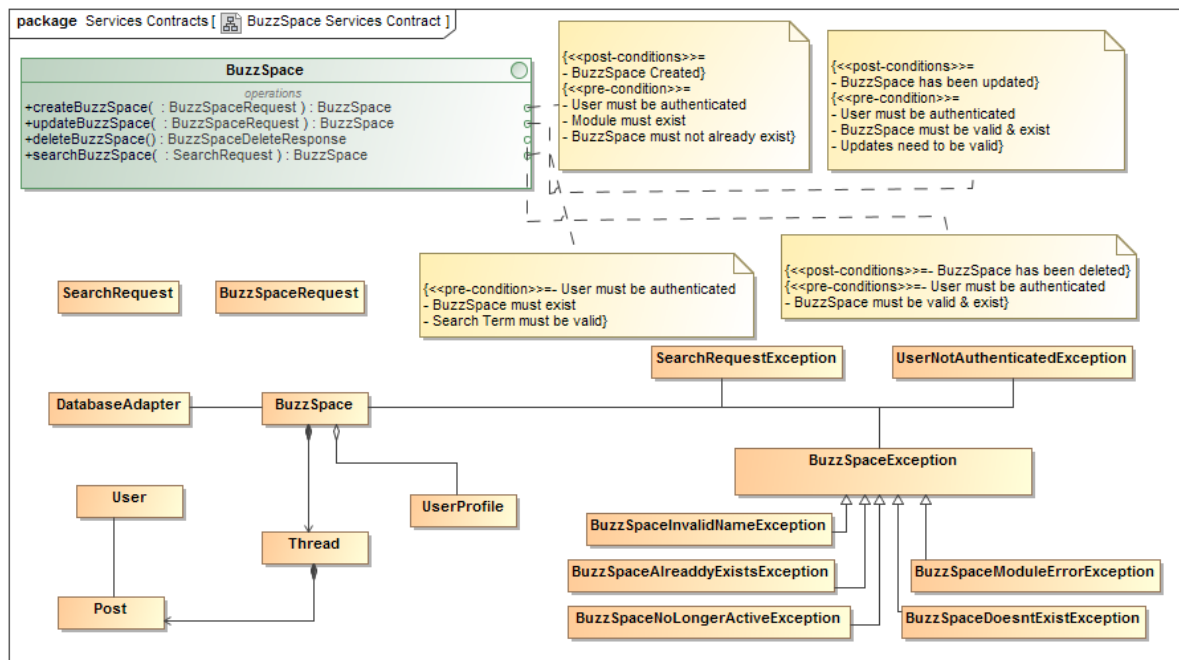


Figure 3: BuzzSpace services contracts.

2.2.4 Communication

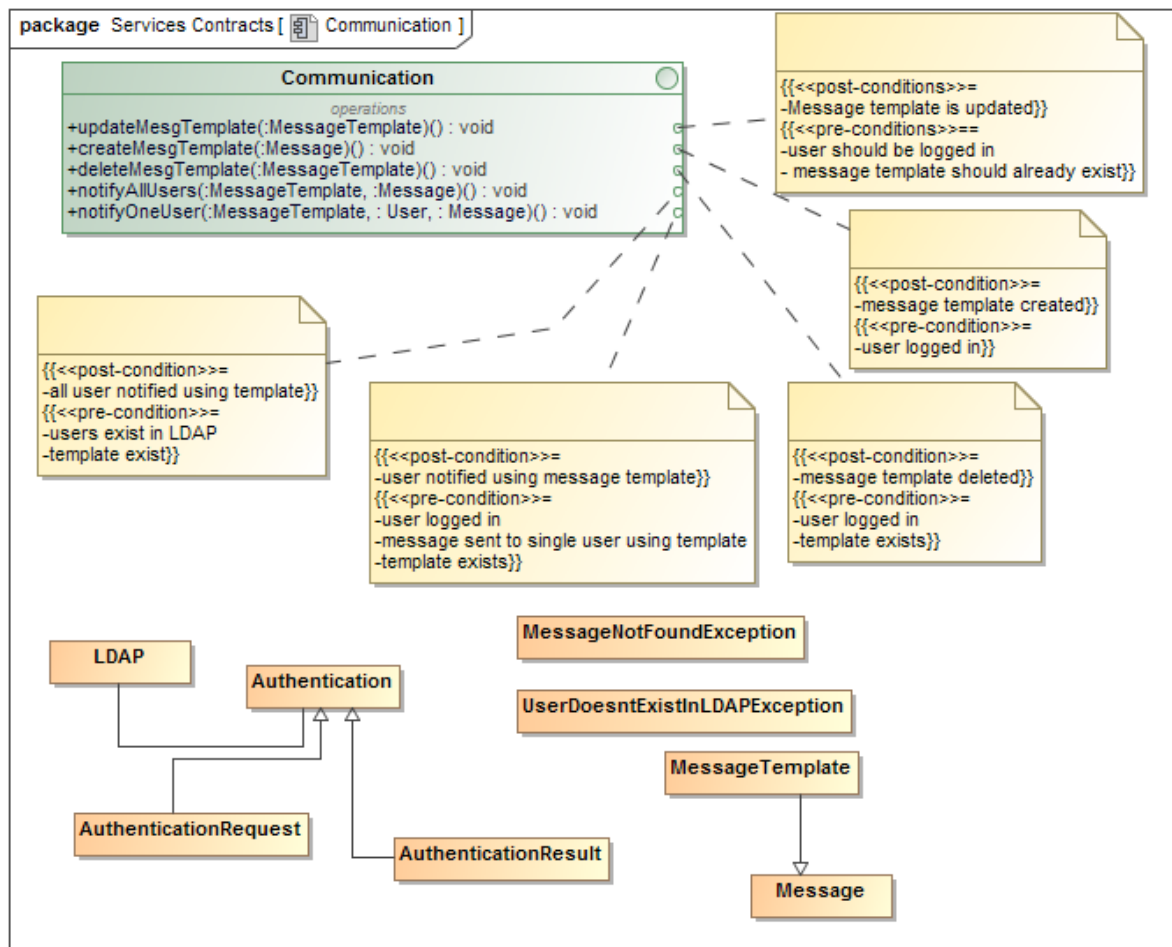


Figure 4: Communication services contracts.

2.2.5 Plagiarism / Netetiquette

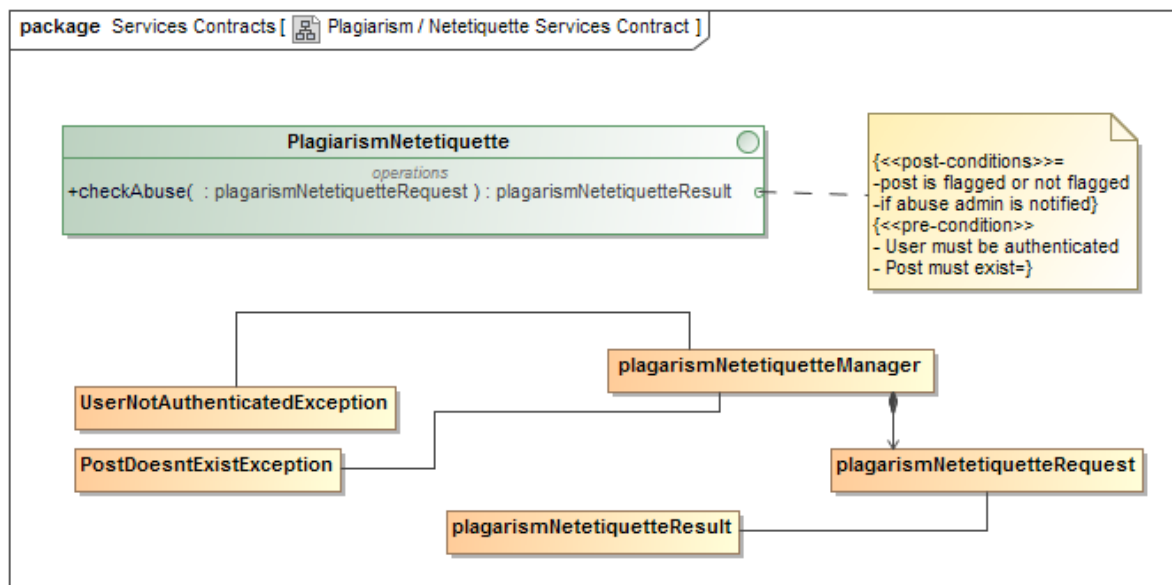


Figure 5: Plagiarism and Netetiquette services contracts.

2.2.6 Tagging

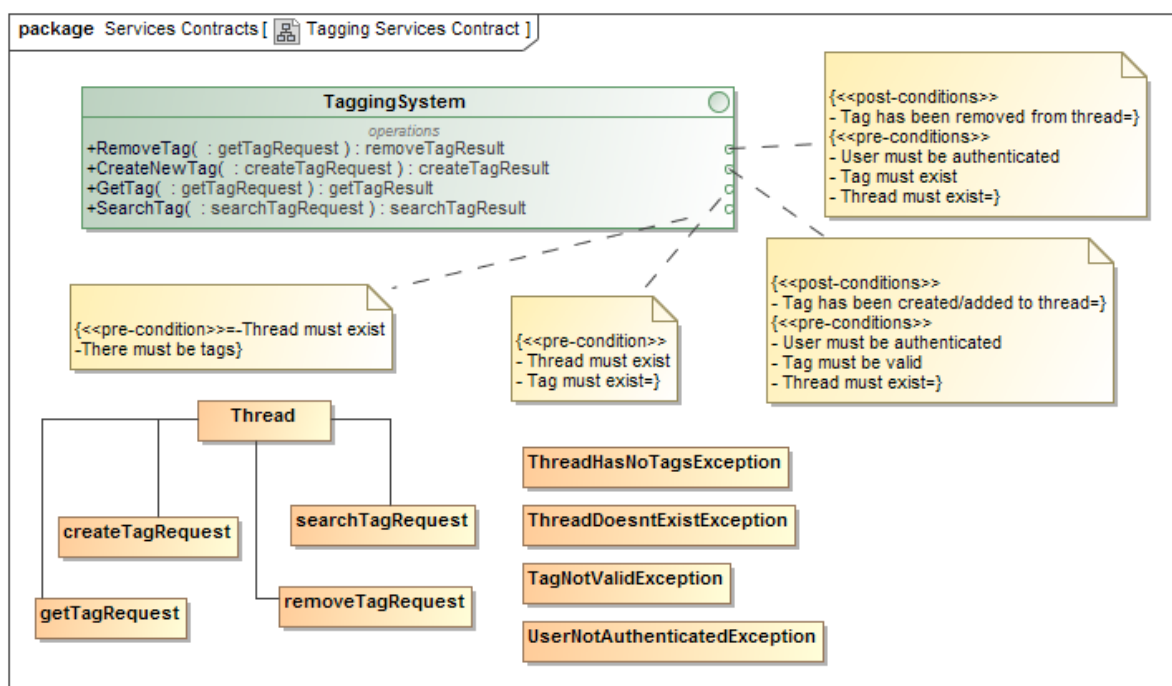


Figure 6: Tagging services contracts.

2.2.7 Thread

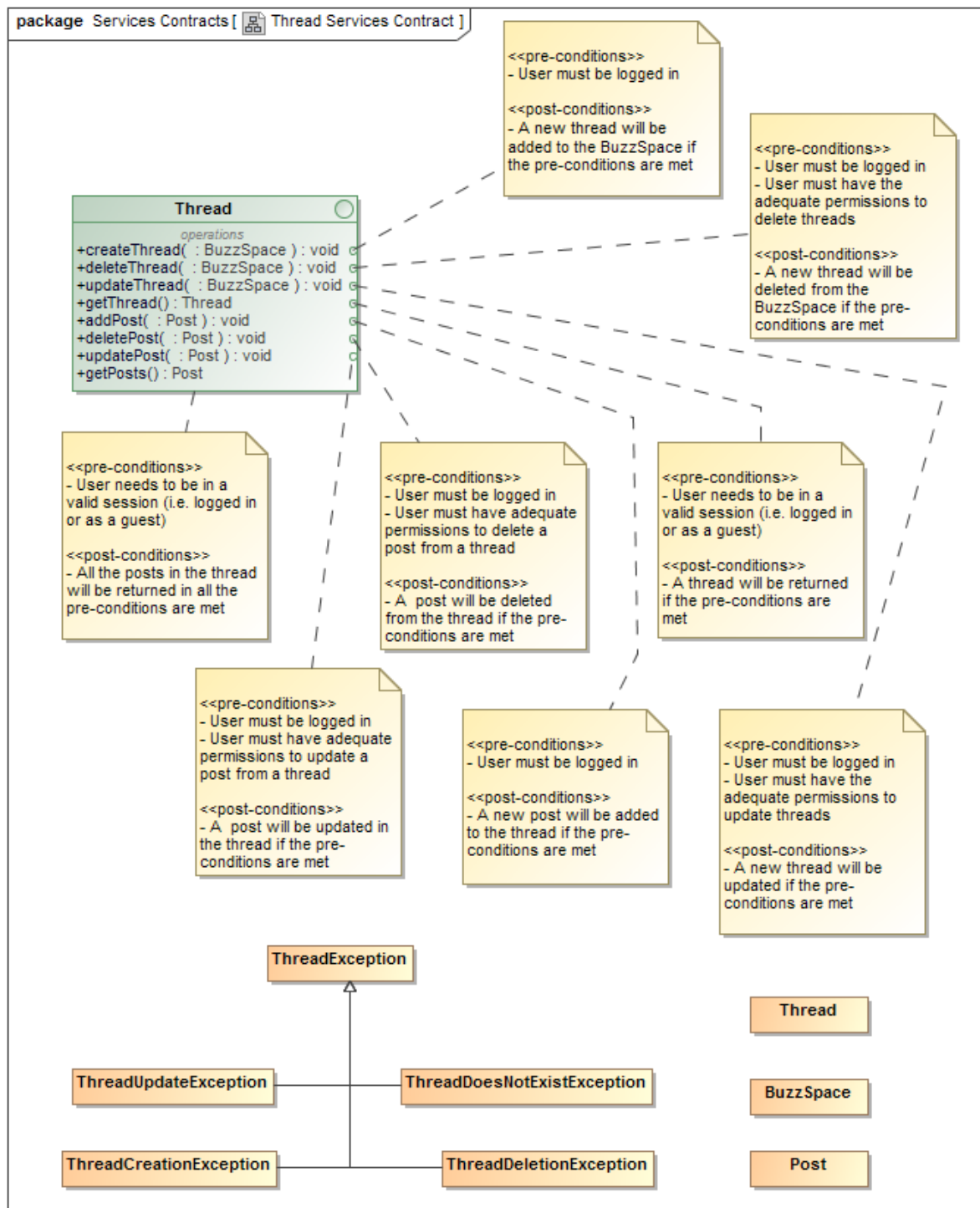


Figure 7: Thread services contracts.

2.2.8 Thread Posts

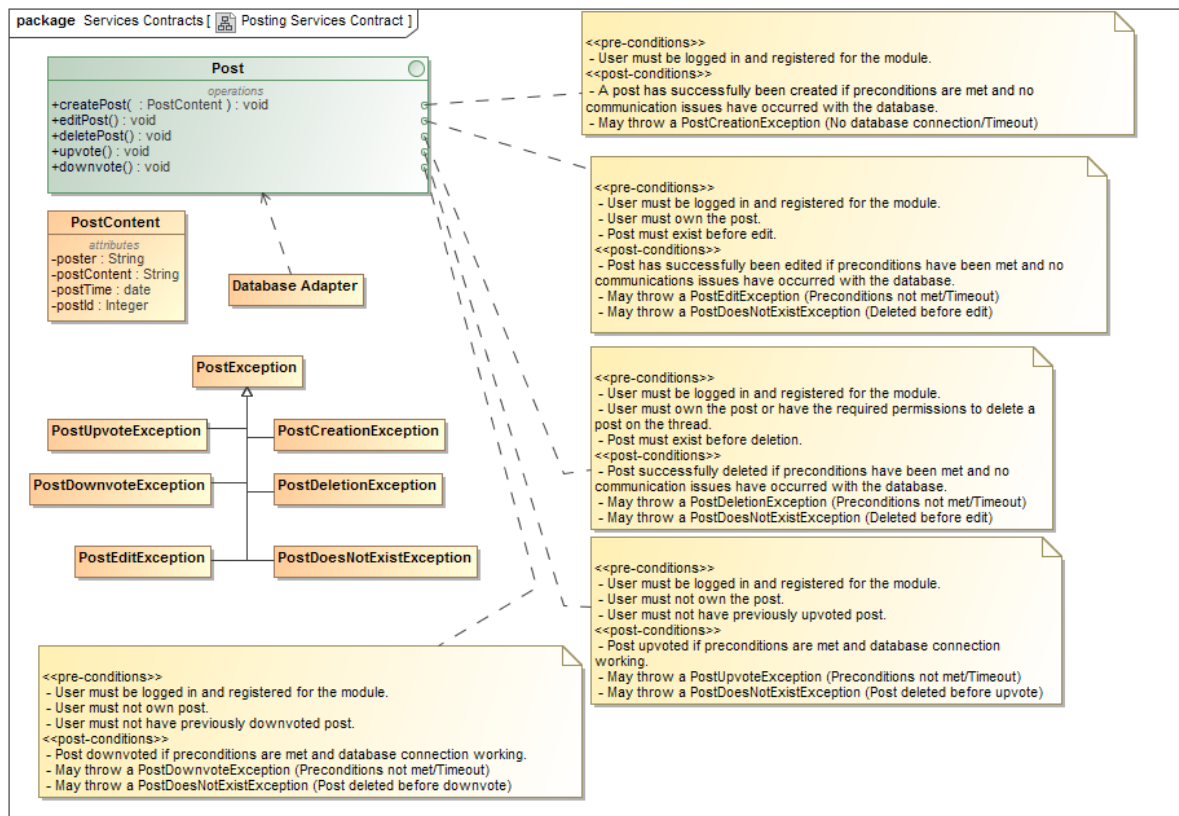


Figure 8: Thread Posts services contracts.

2.2.9 User

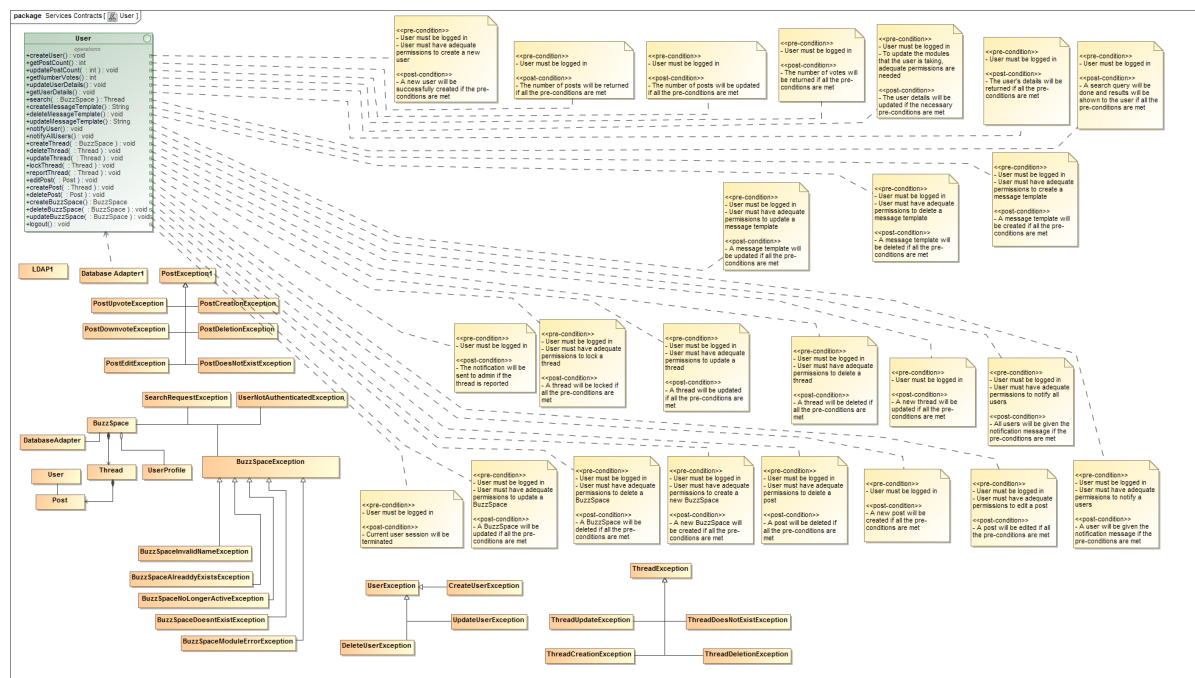


Figure 9: User actions services contracts.

2.3 Required functionality

2.3.1 Analytics

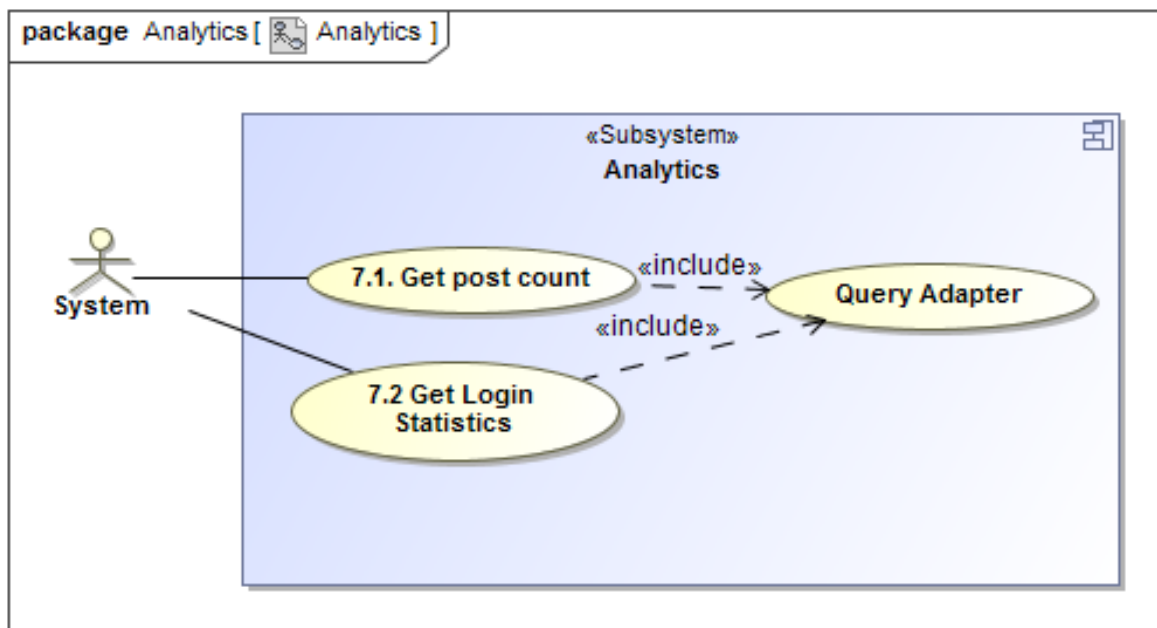


Figure 10: Analytics sub system use case diagram.

2.3.2 Authentication

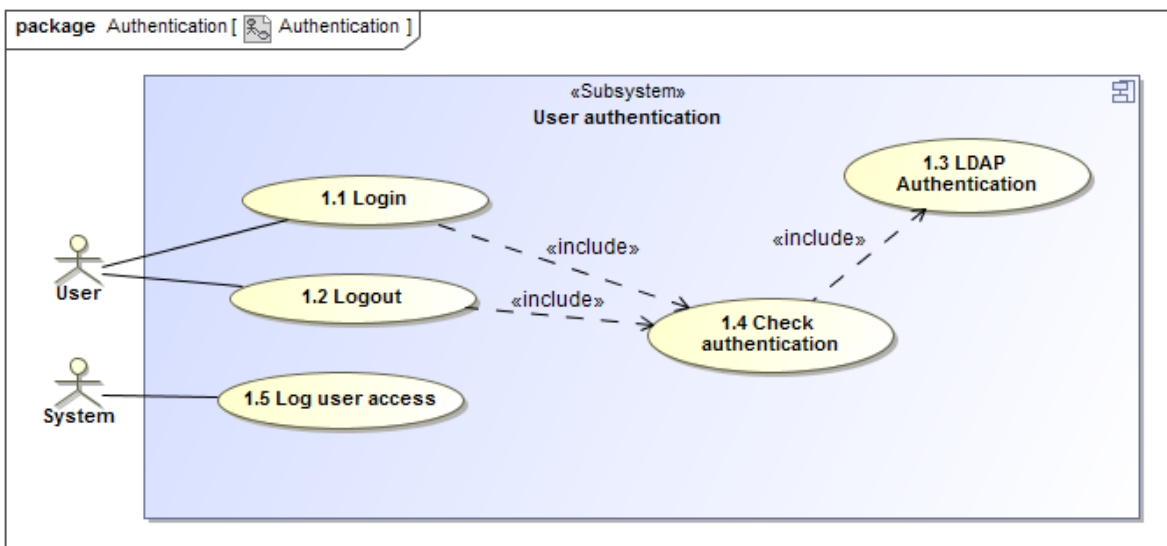


Figure 11: Authentication sub system use case diagram.

2.3.3 BuzzSpace

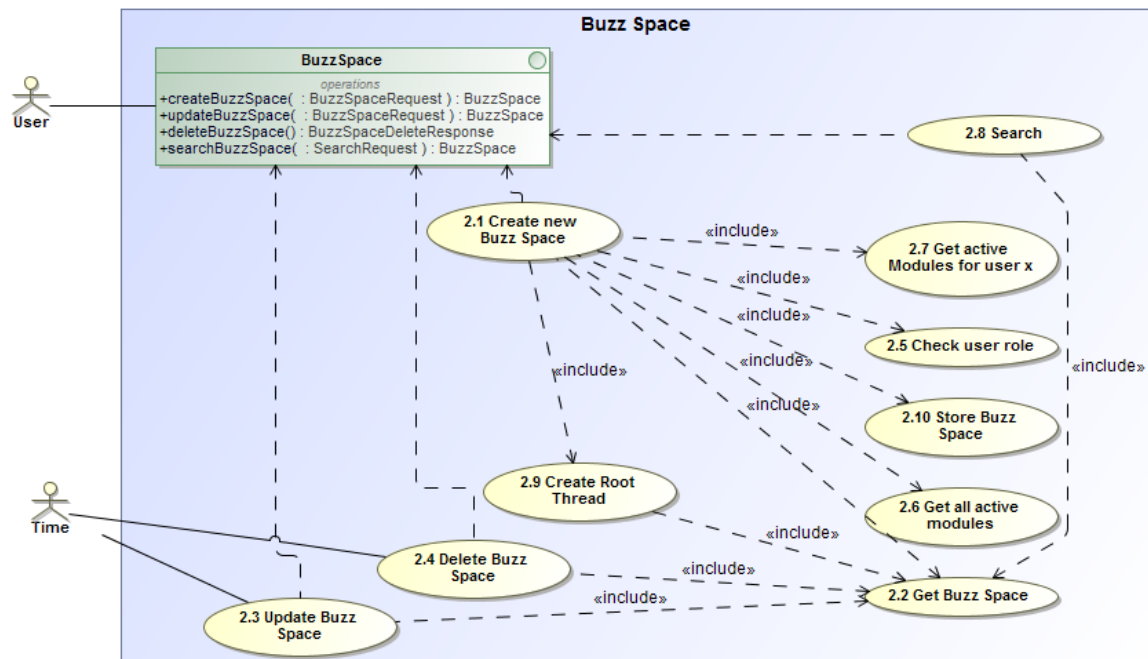


Figure 12: BuzzSpace sub system use case diagram.

2.3.4 Communication

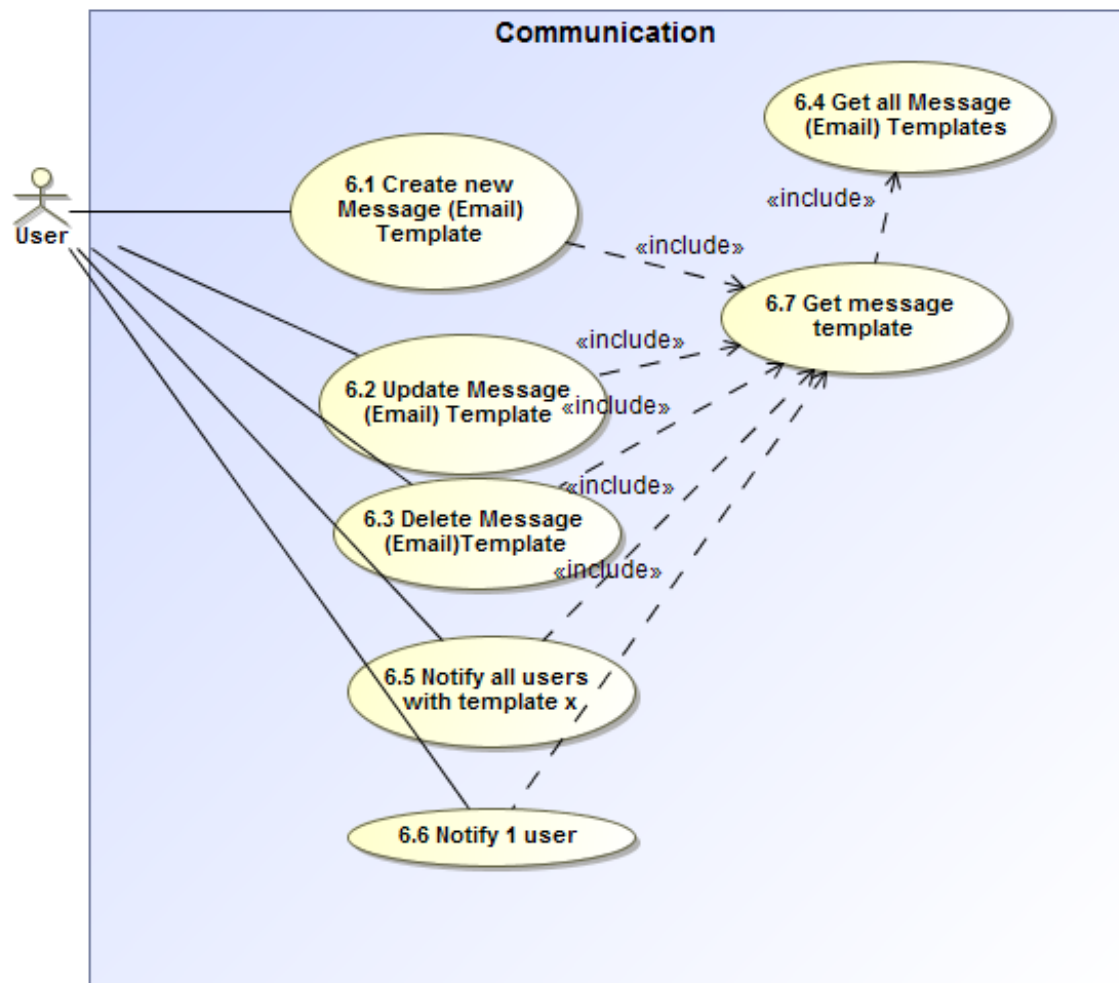


Figure 13: Communication sub system use case diagram.

2.3.5 Plagiarism / Netetiquette

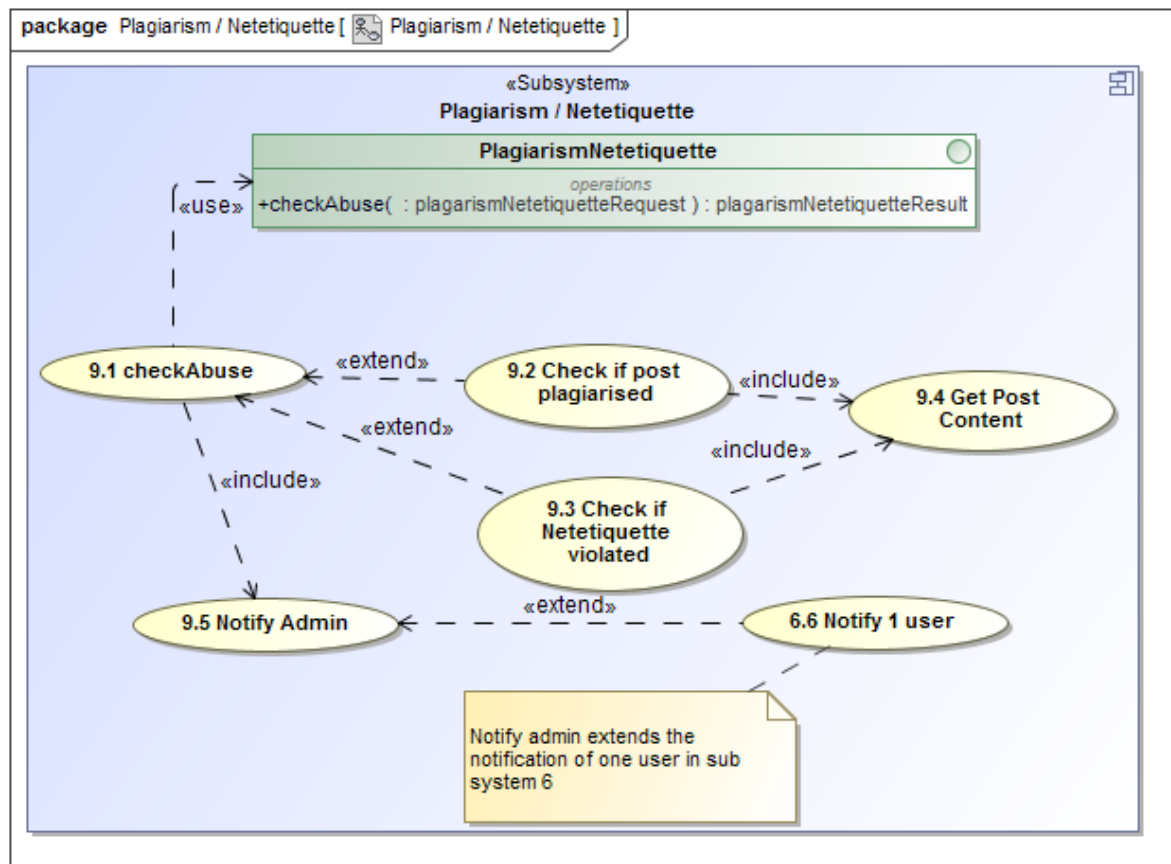


Figure 14: Plagiarism and Netetiquette sub system use case diagram.

2.3.6 Tagging

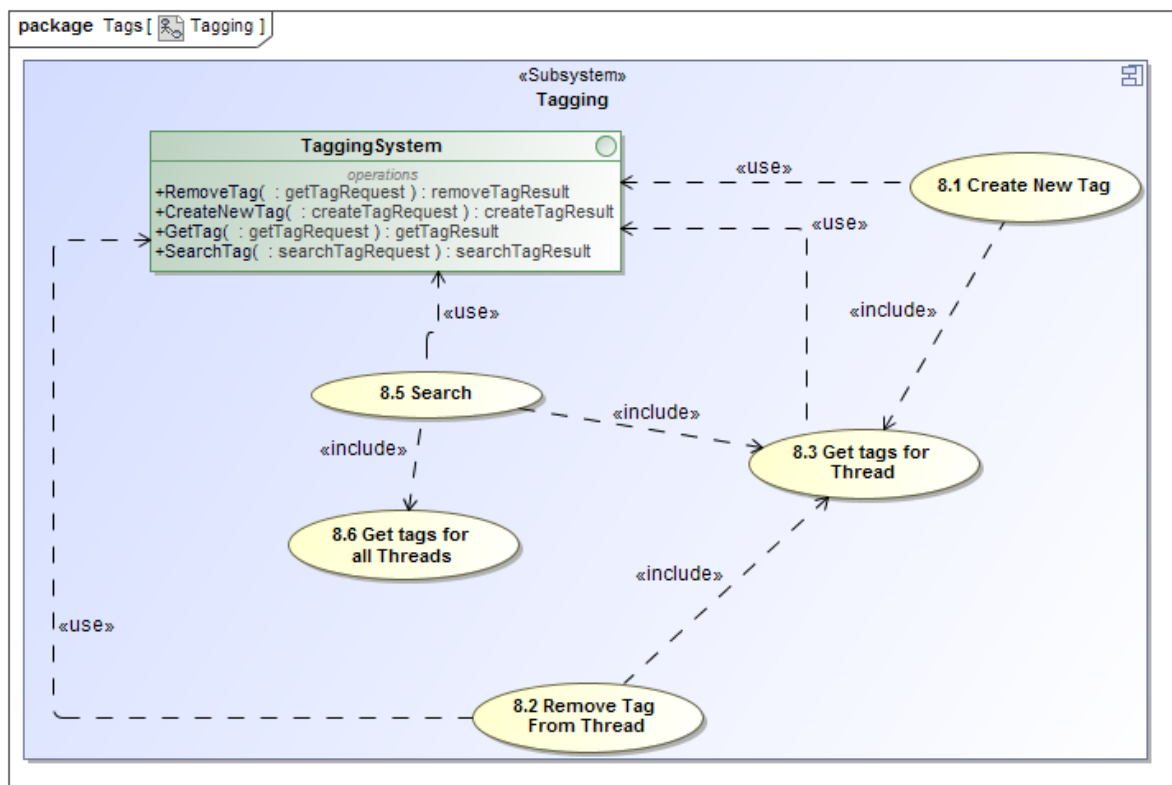


Figure 15: Tagging sub system use case diagram.

2.3.7 Thread

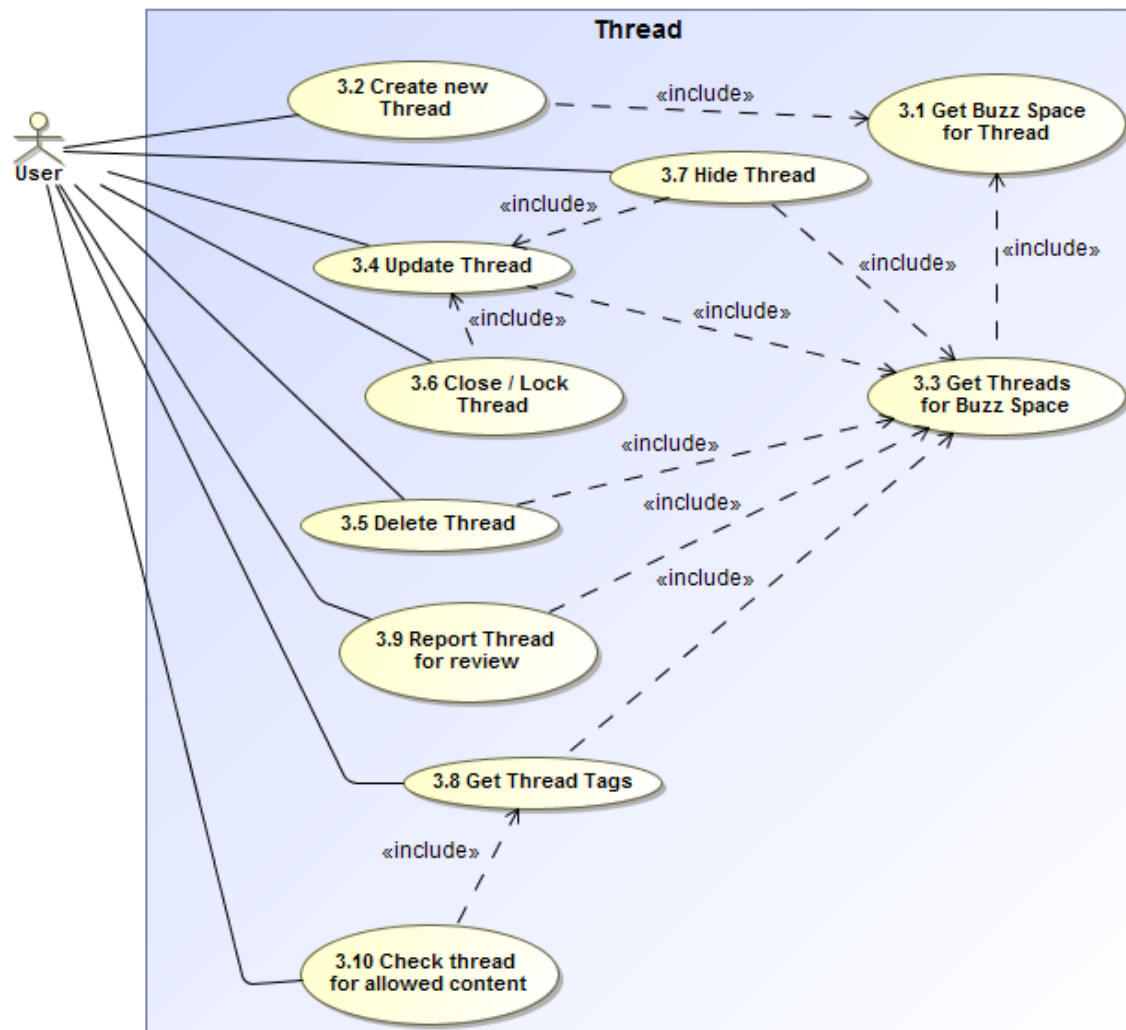


Figure 16: Tagging sub system use case diagram.

2.3.8 Thread Posts

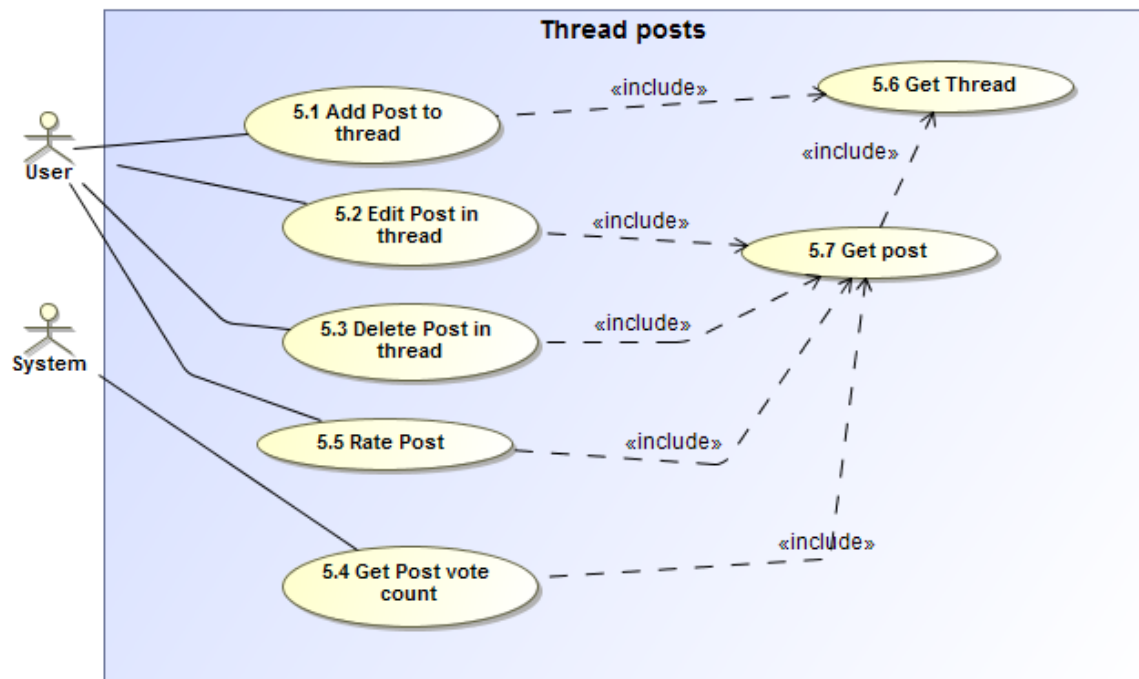


Figure 17: Thread Posts sub system use case diagram.

2.3.9 User

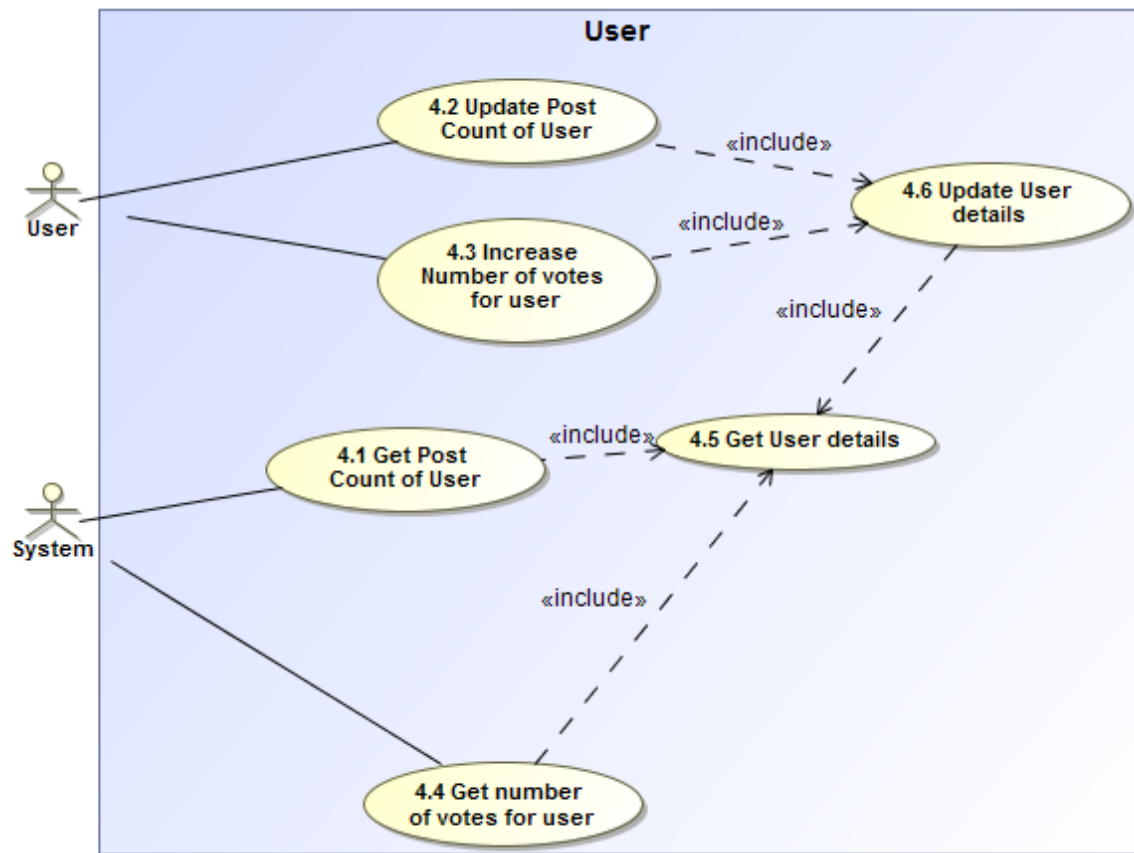


Figure 18: Tagging sub system use case diagram.

2.4 Process specifications

2.4.1 Authentication

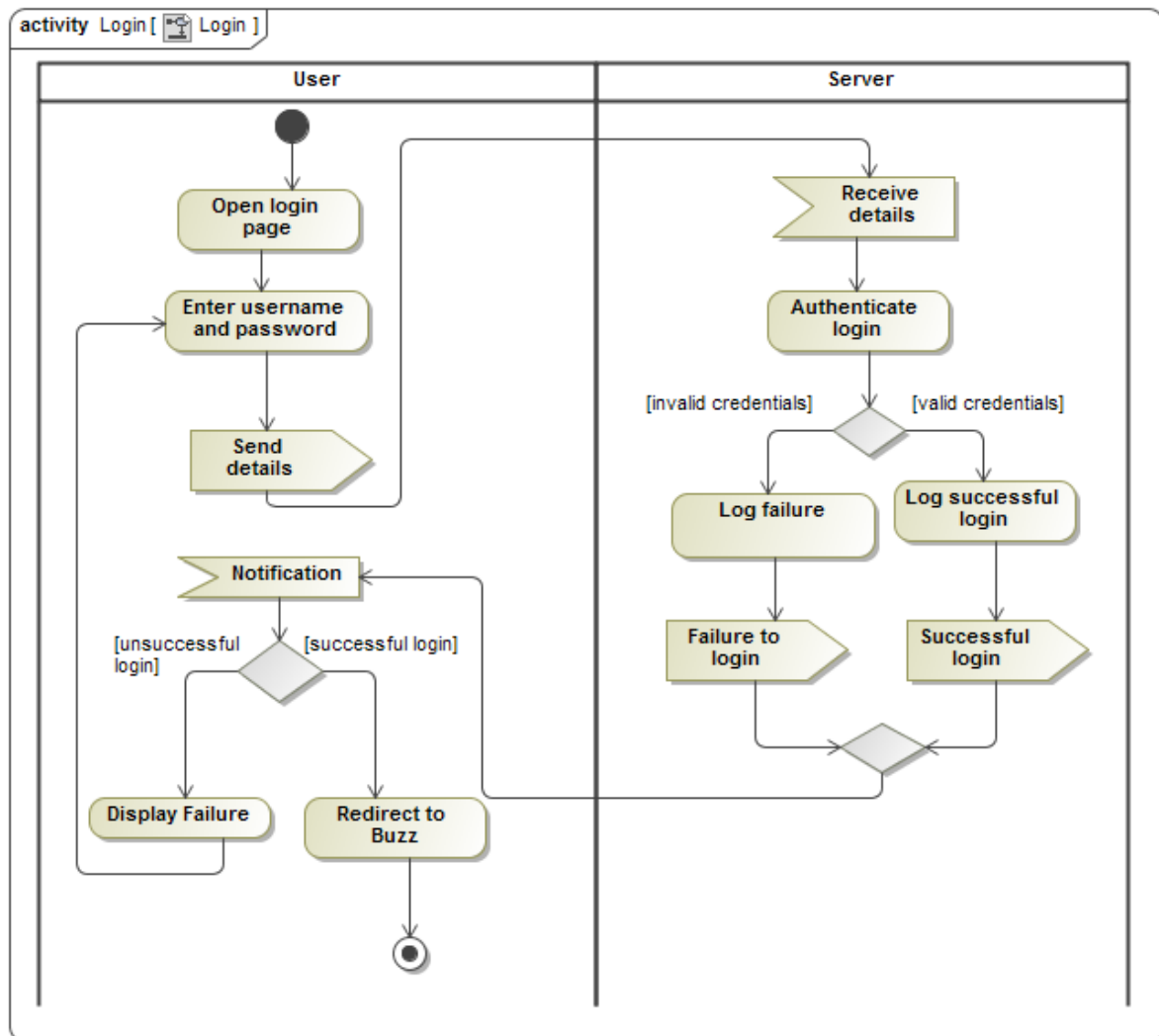


Figure 19: Login activity diagram.

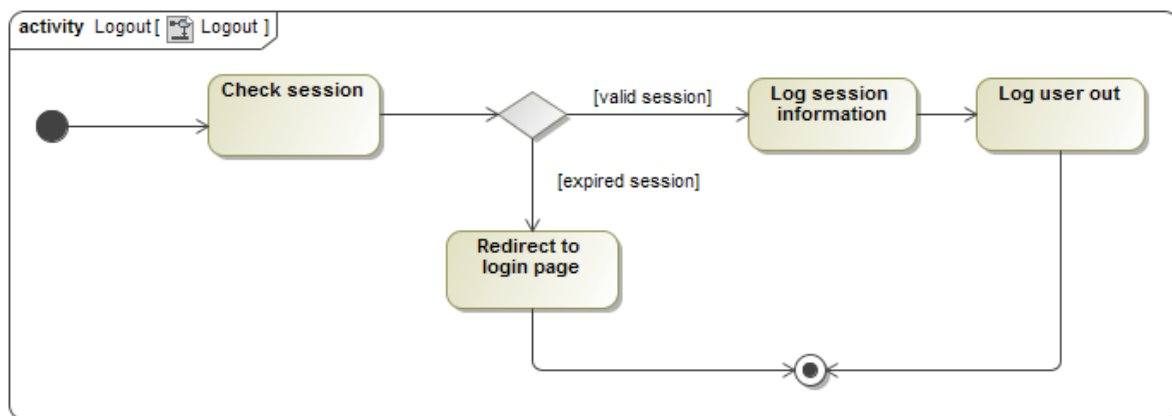


Figure 20: Logout activity diagram.

2.4.2 BuzzSpace

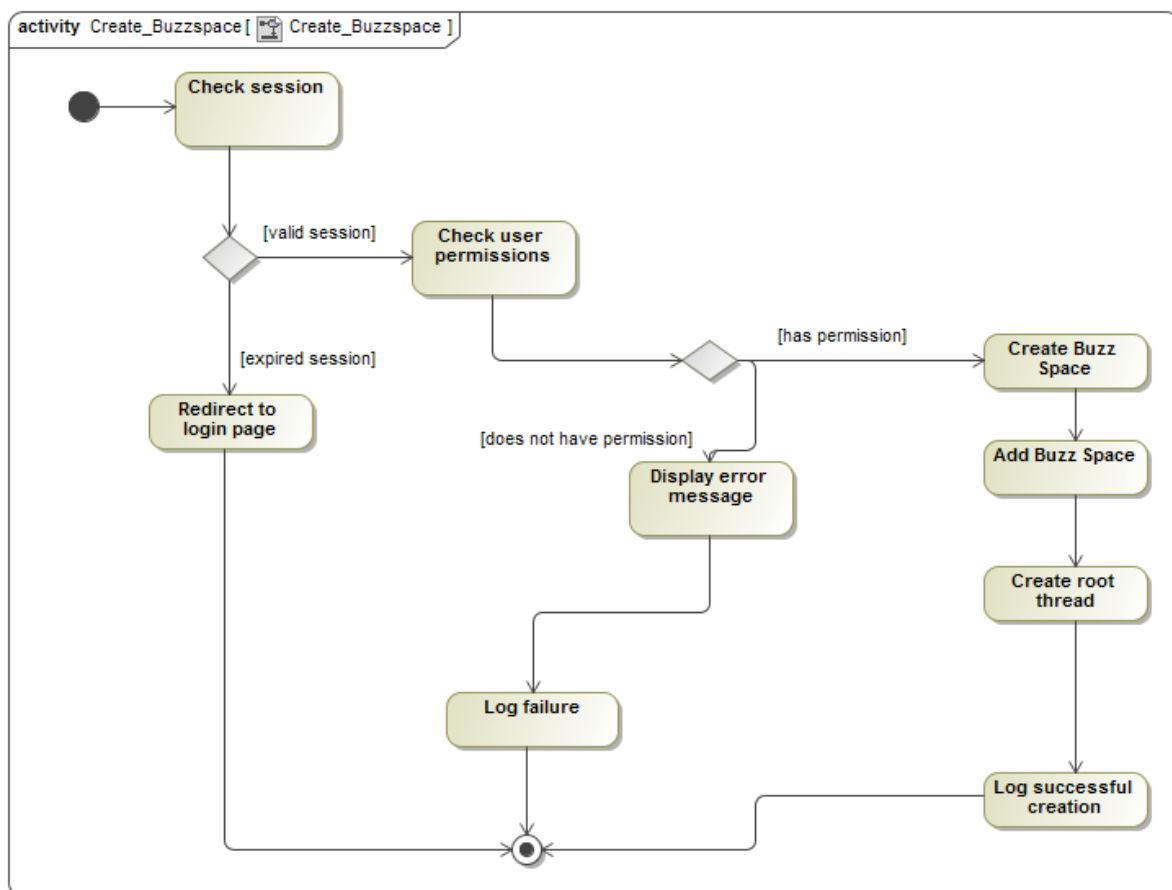


Figure 21: Create Buzzspace activity diagram.

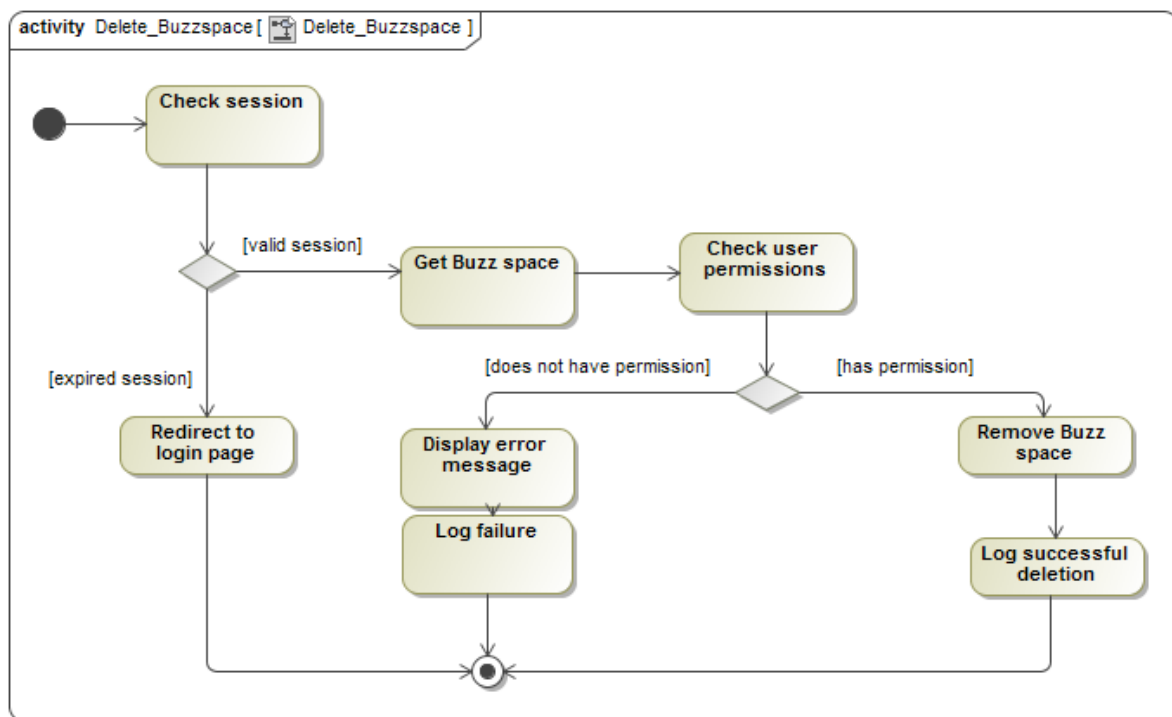


Figure 22: Delete Buzzspace activity diagram.

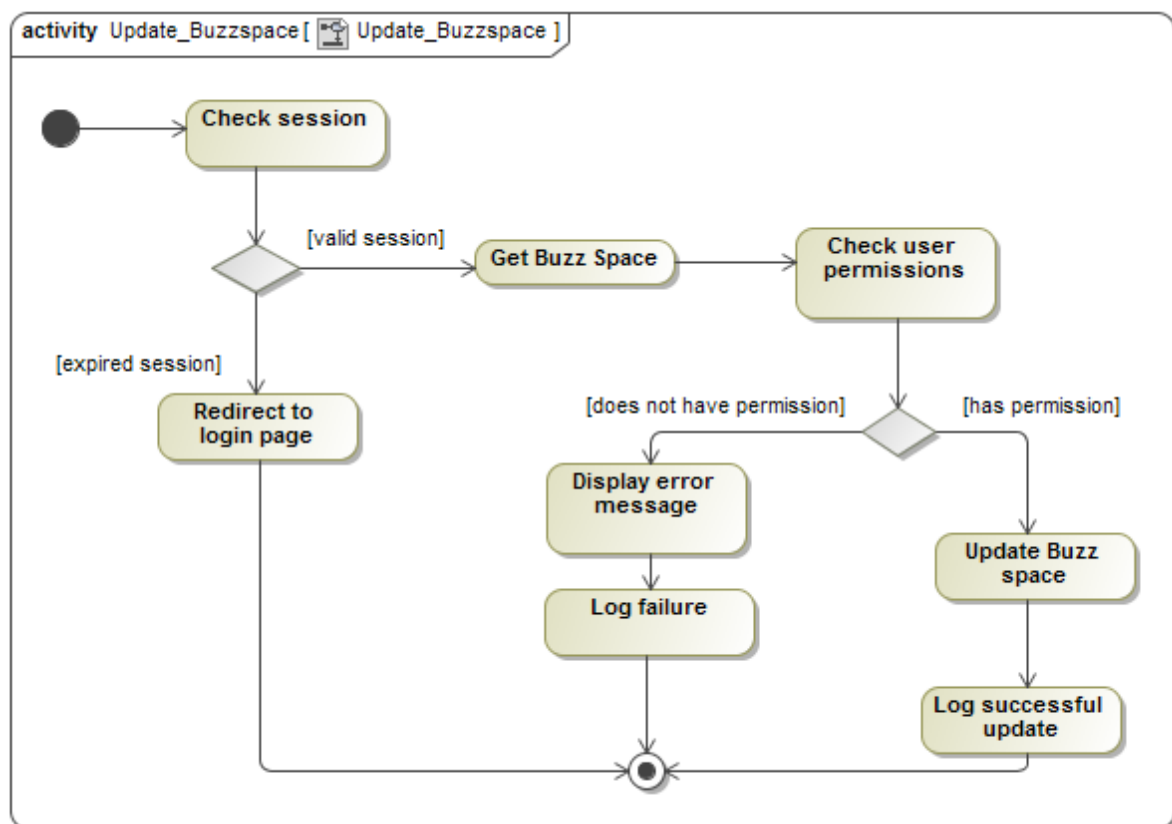


Figure 23: Update Buzzspace activity diagram.

2.4.3 Communication

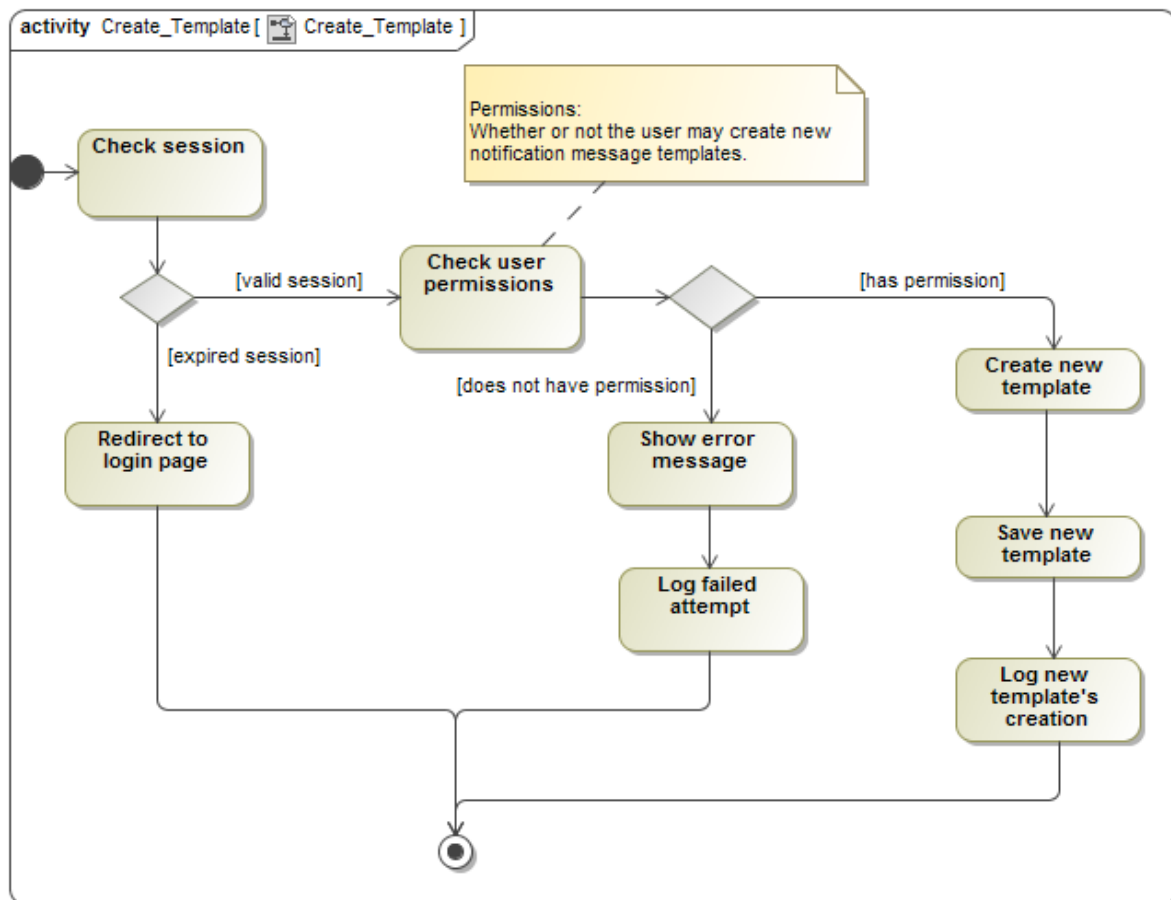


Figure 24: Create Template activity diagram.

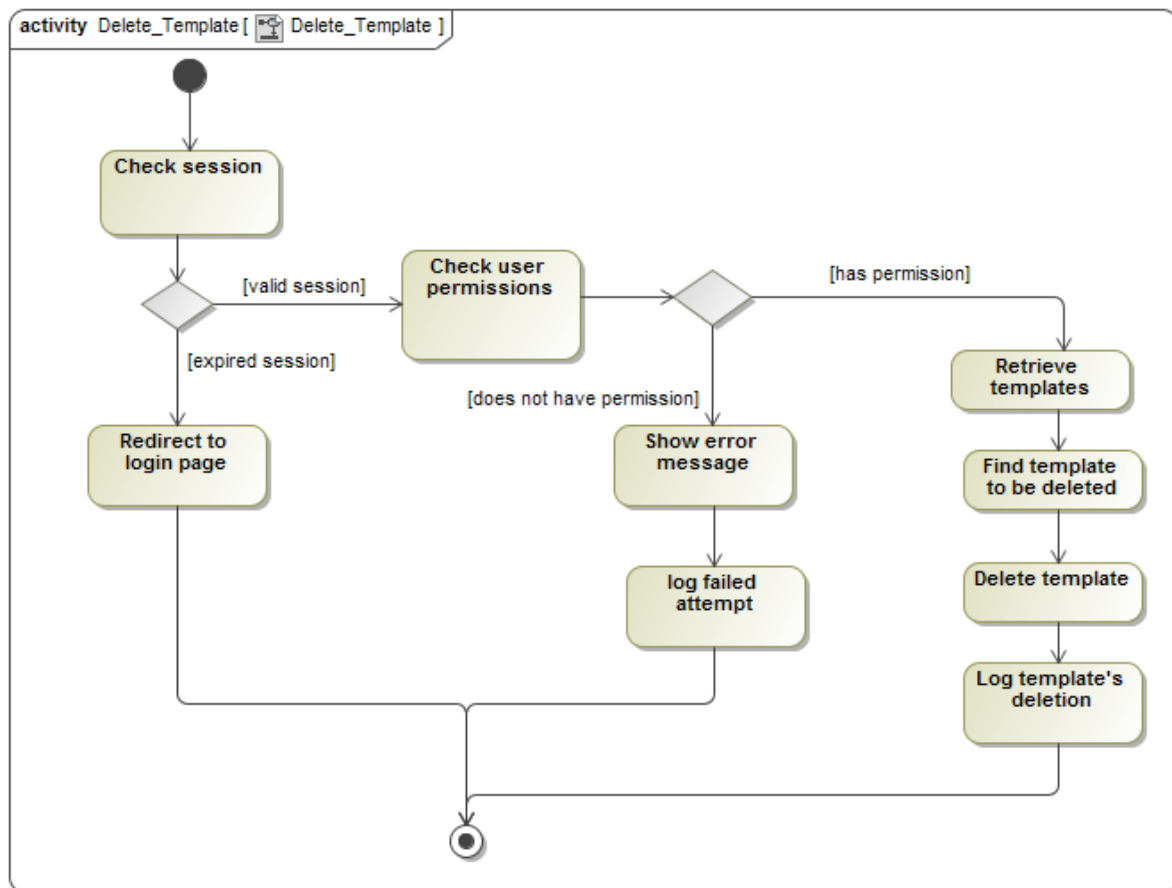


Figure 25: Delete Template activity diagram.

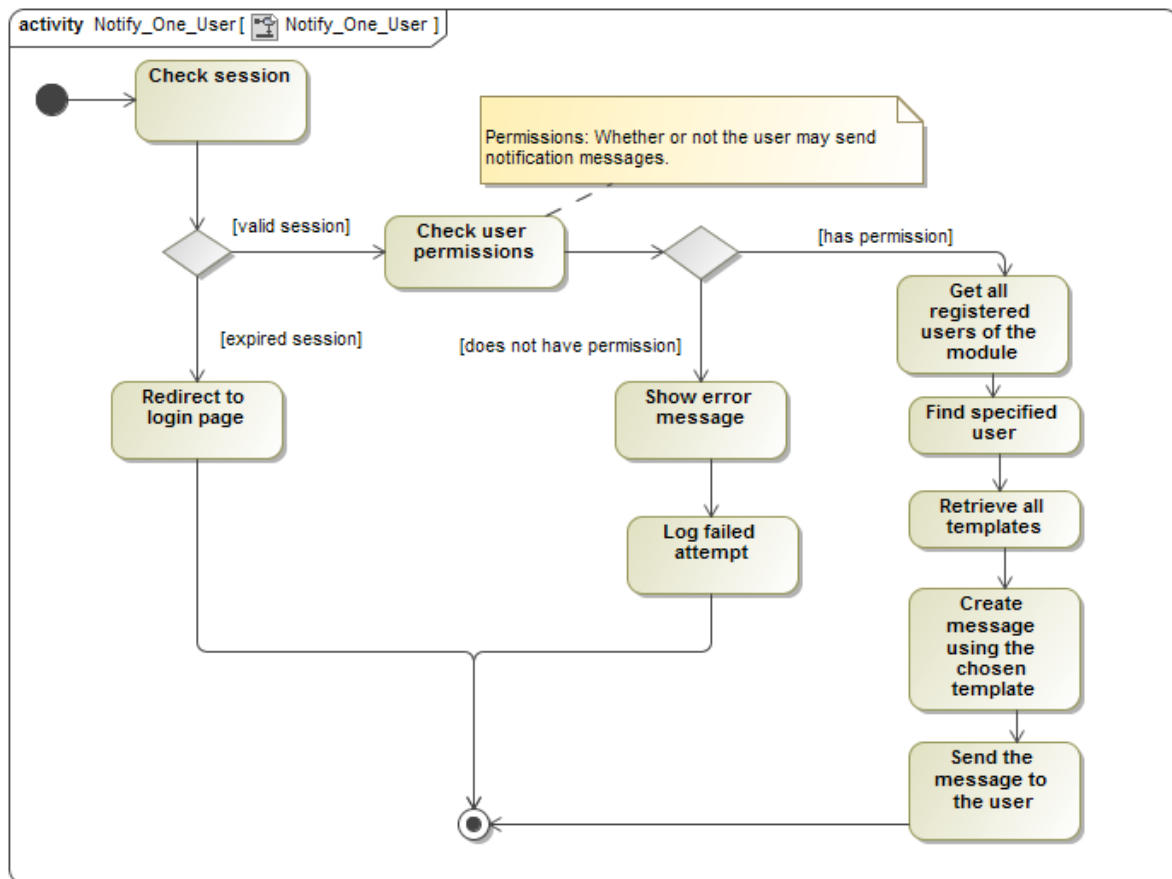


Figure 26: Notify One User activity diagram.

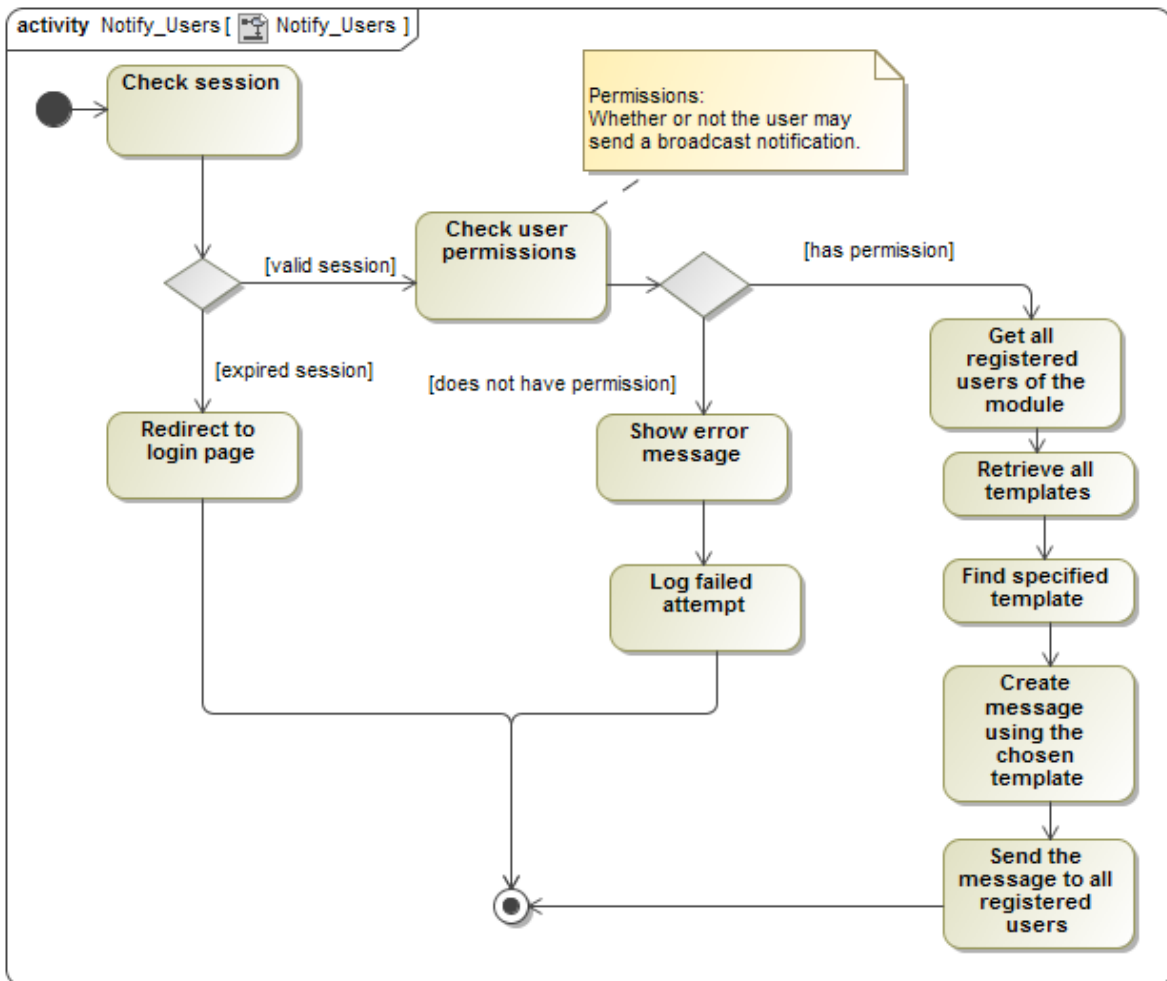


Figure 27: Notify Users activity diagram.

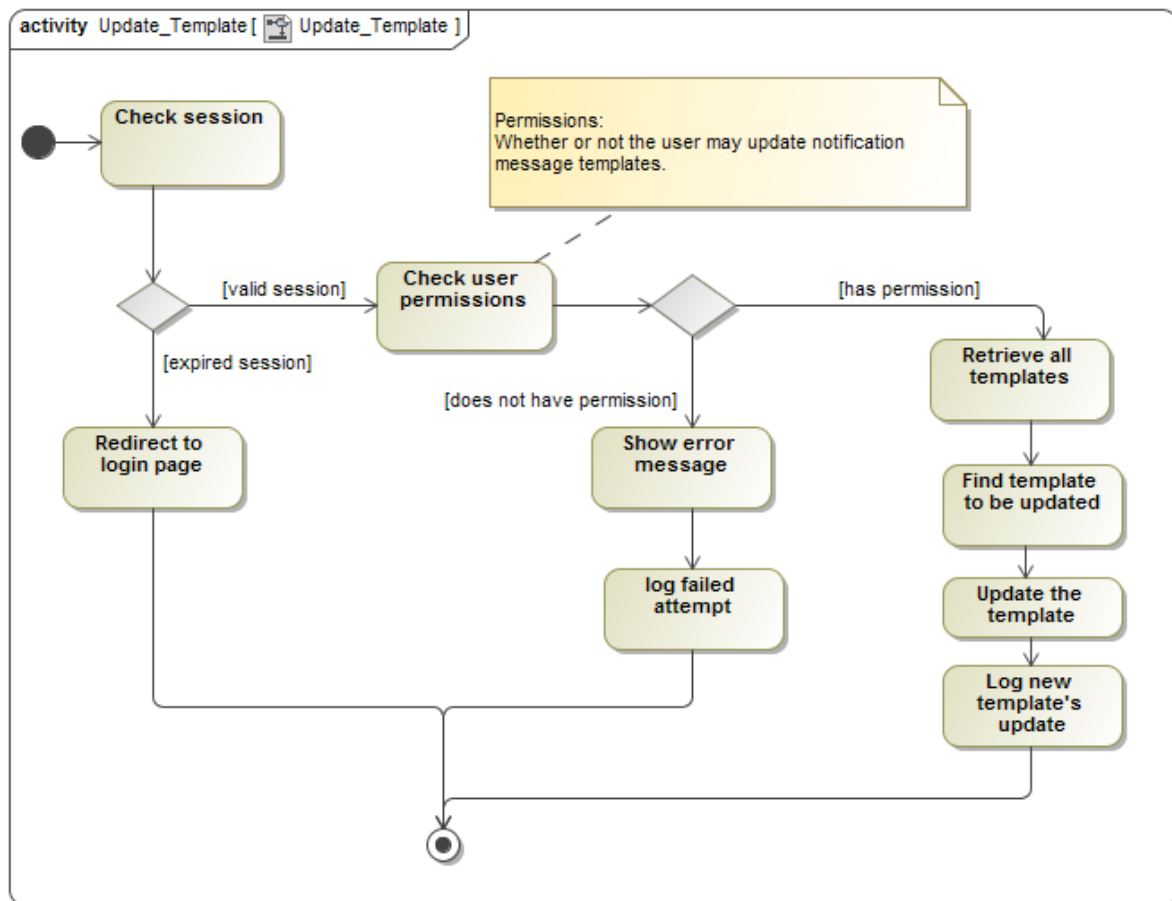


Figure 28: Update Template activity diagram.

2.4.4 Tagging

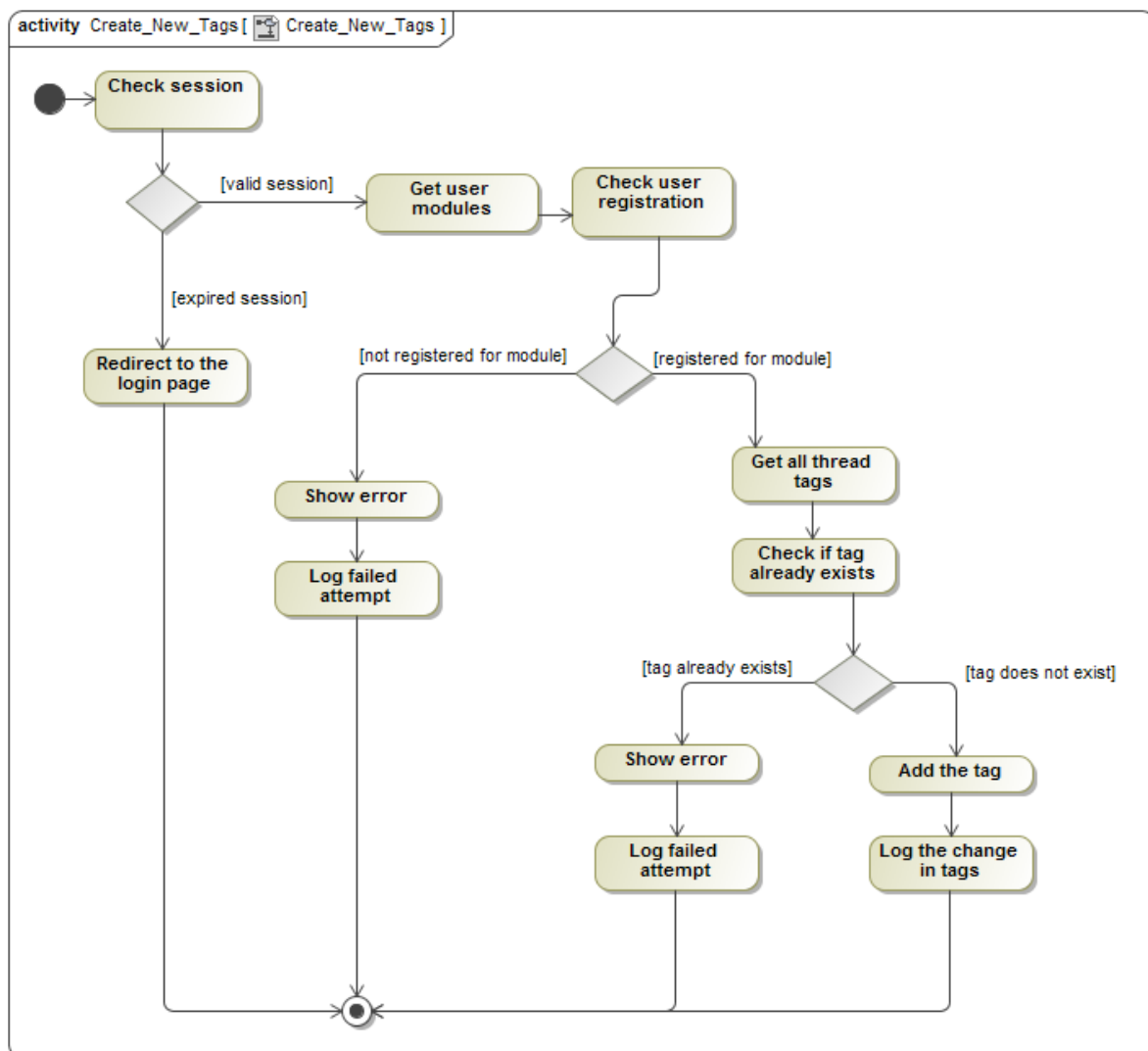


Figure 29: Create New Tags activity diagram.

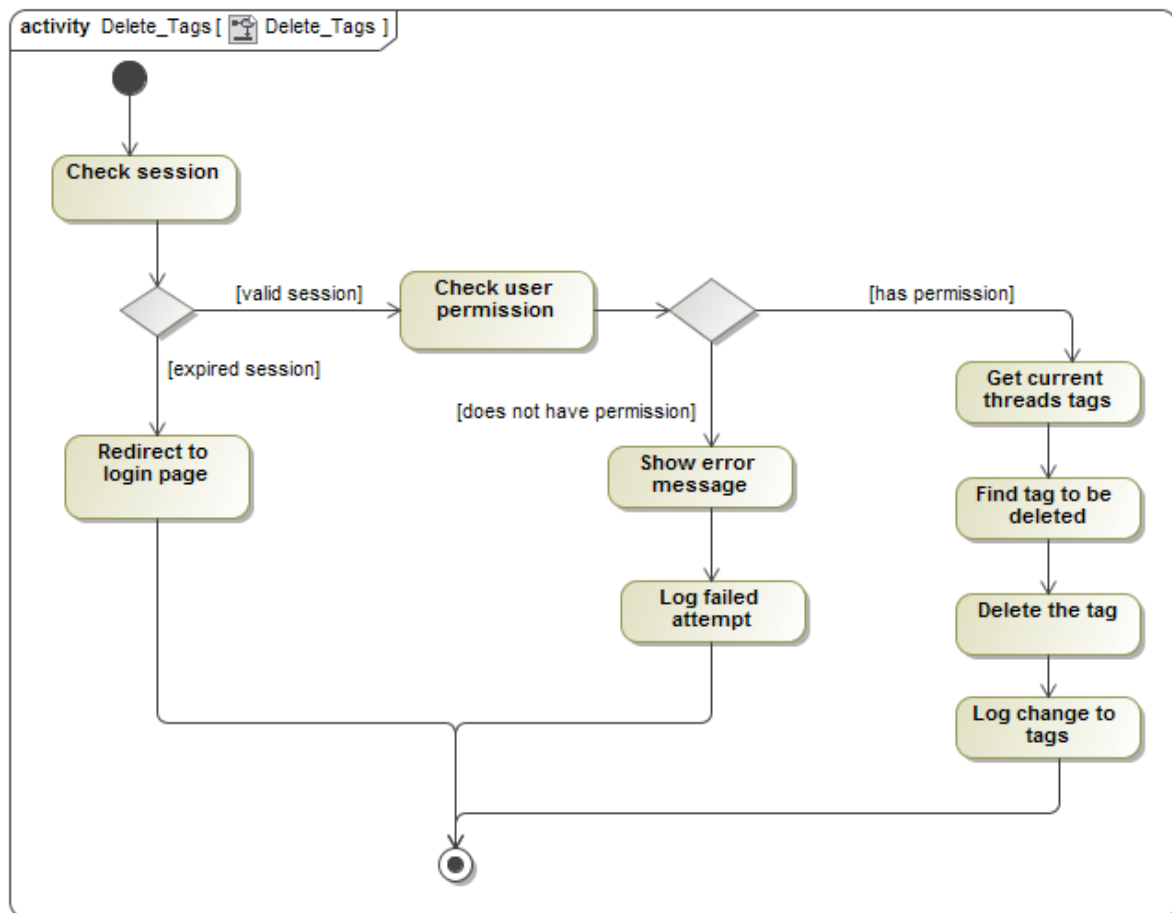


Figure 30: Delete Tags activity diagram.

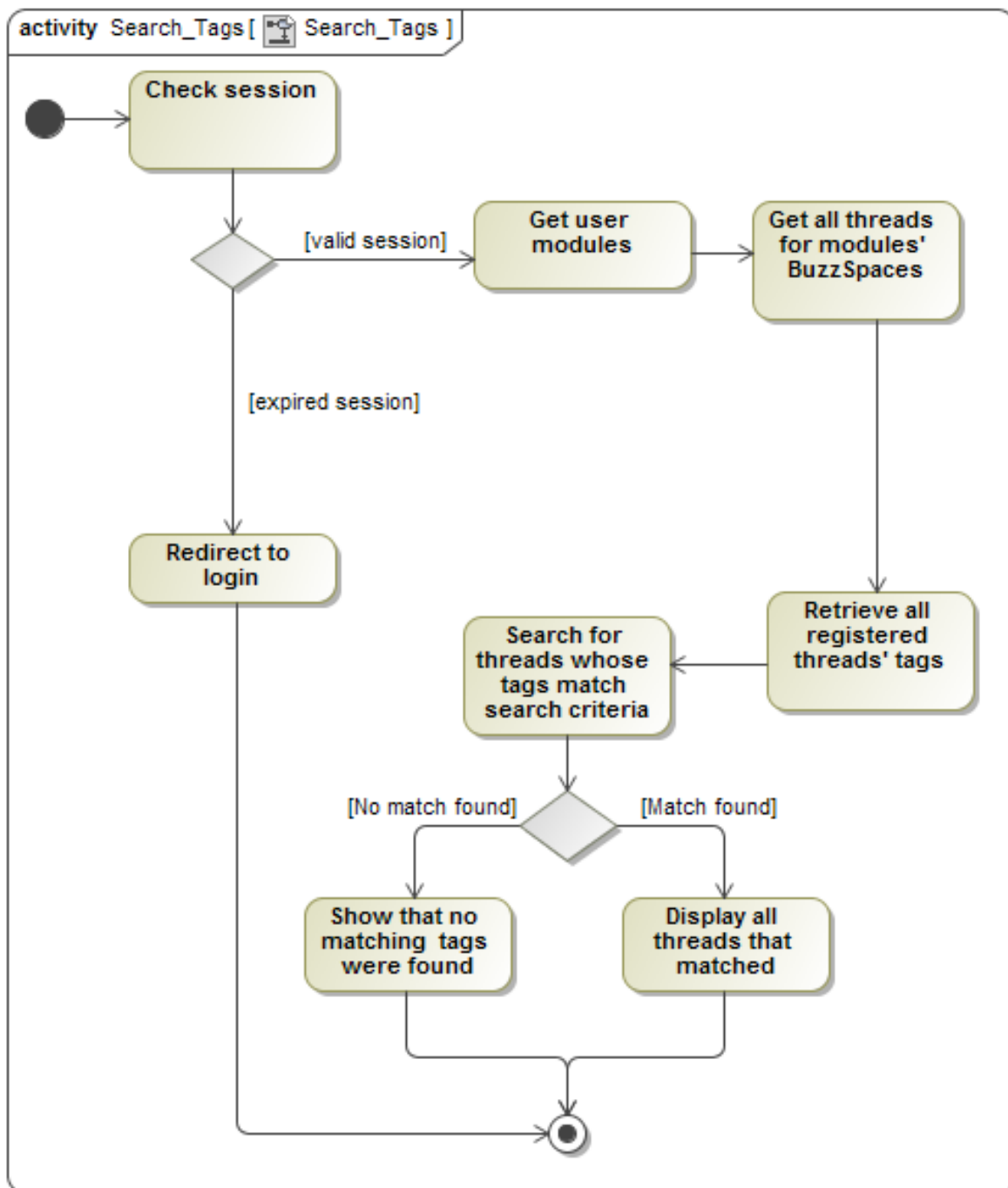


Figure 31: Search Tags activity diagram.

2.4.5 Thread

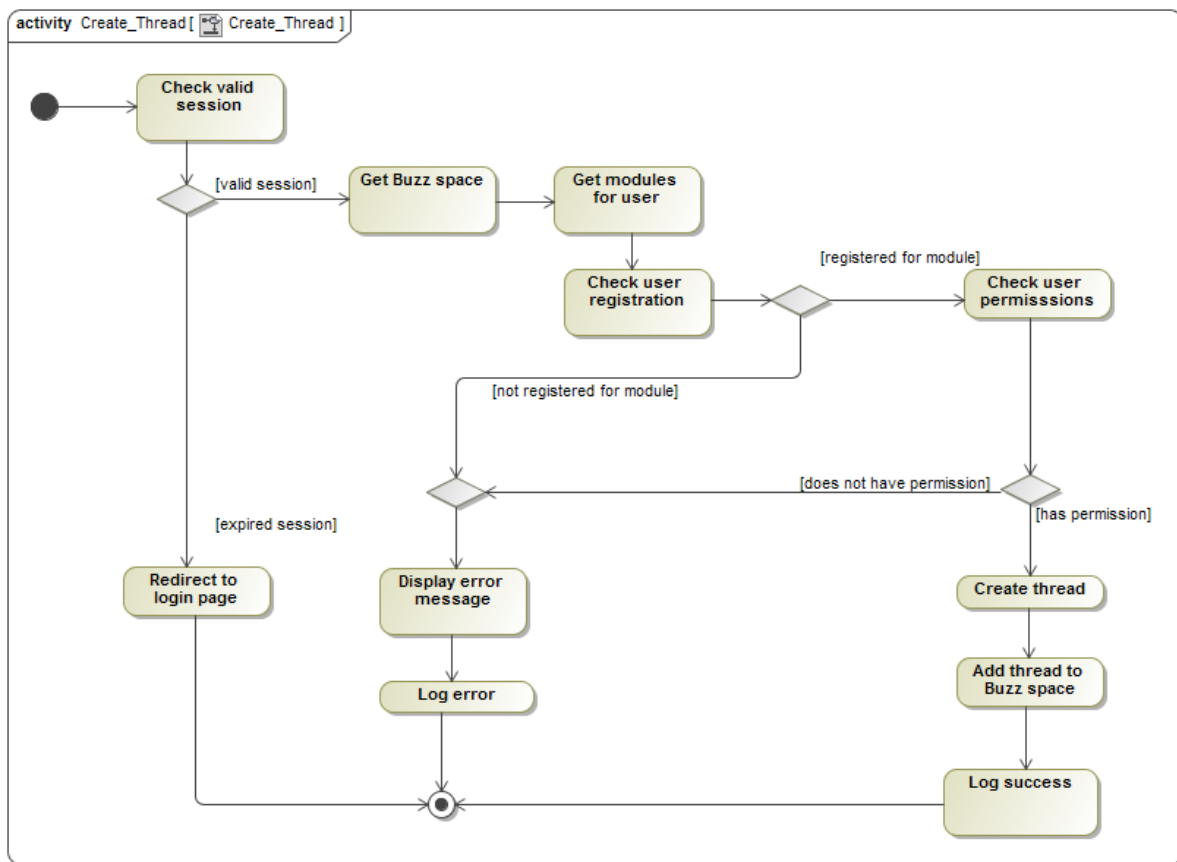


Figure 32: Create thread activity diagram.

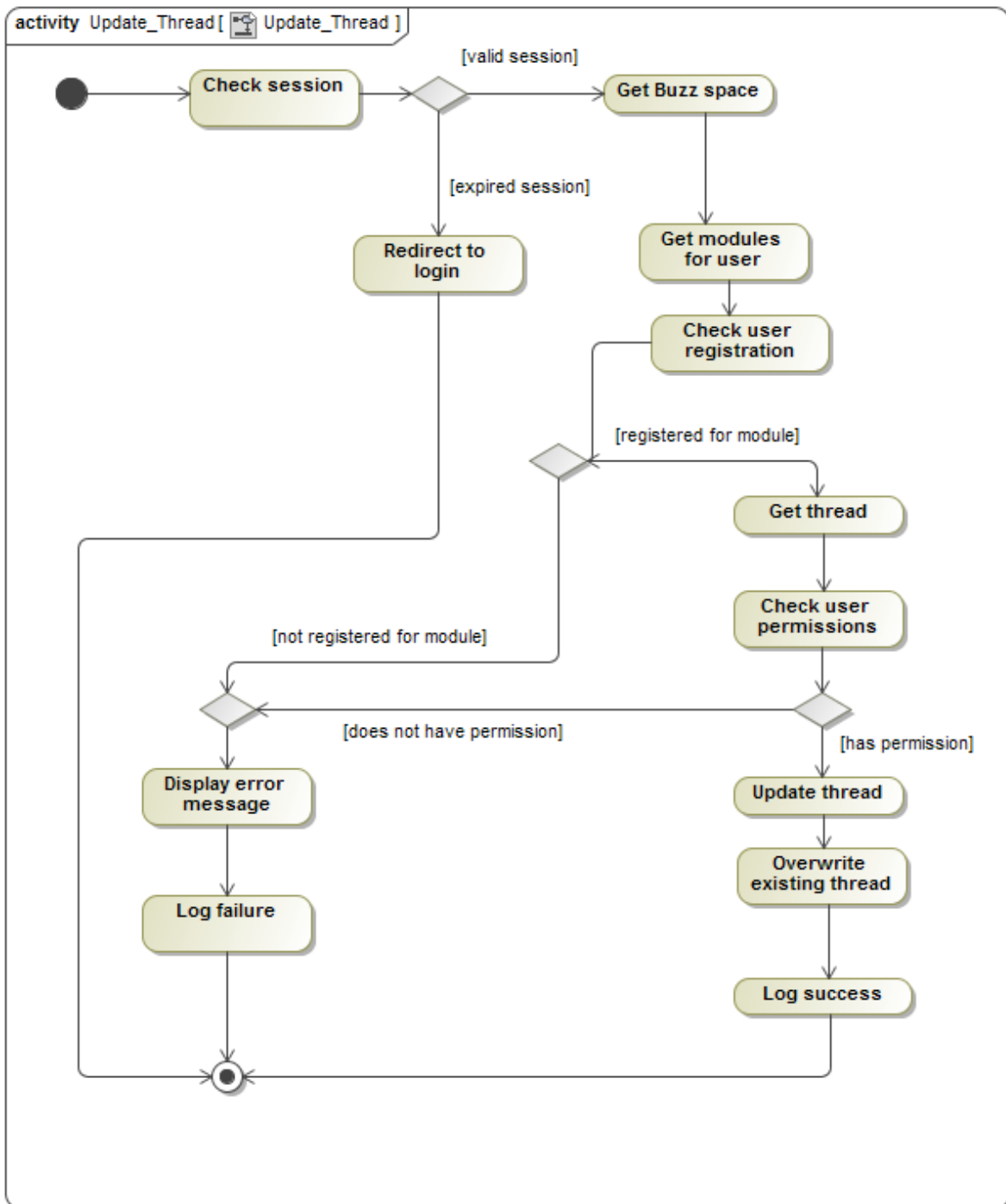


Figure 33: Update thread activity diagram.

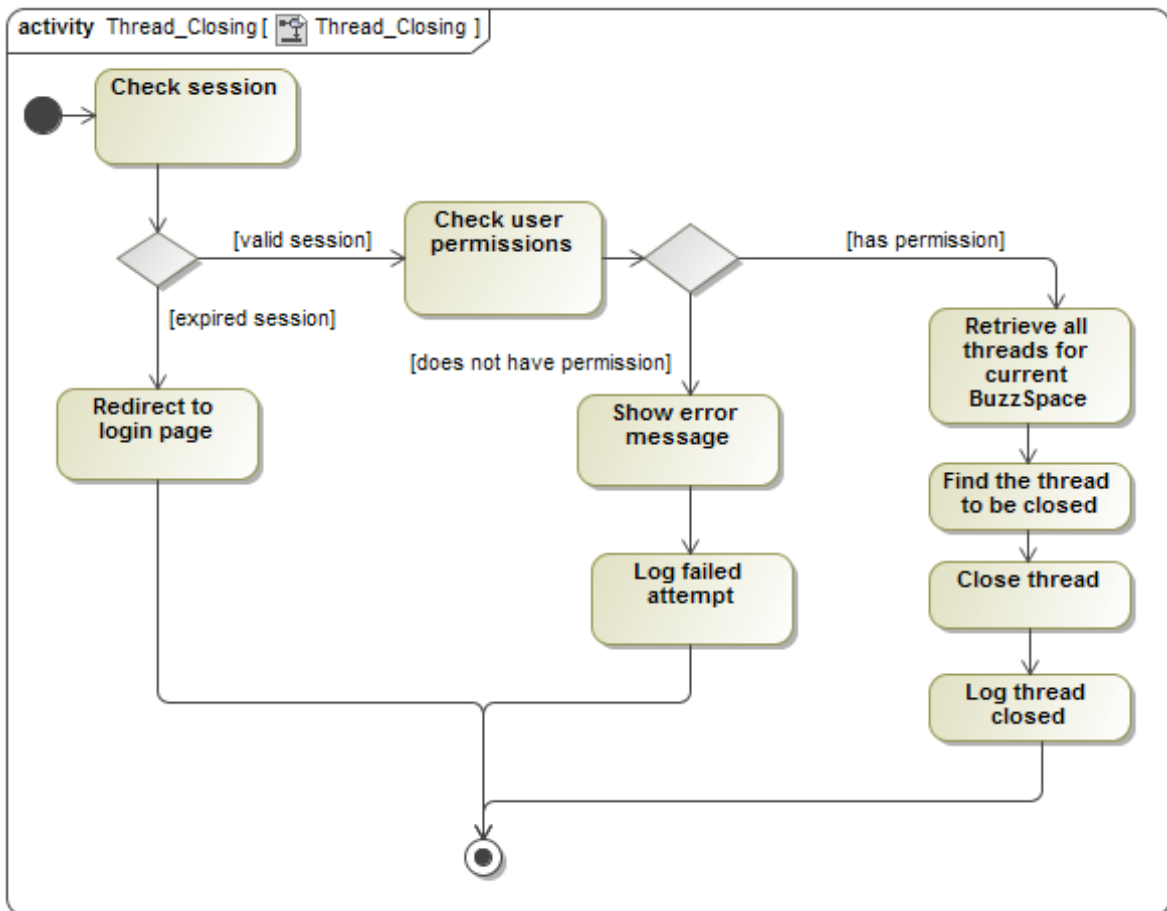


Figure 34: Close thread diagram.

2.4.6 Thread posts

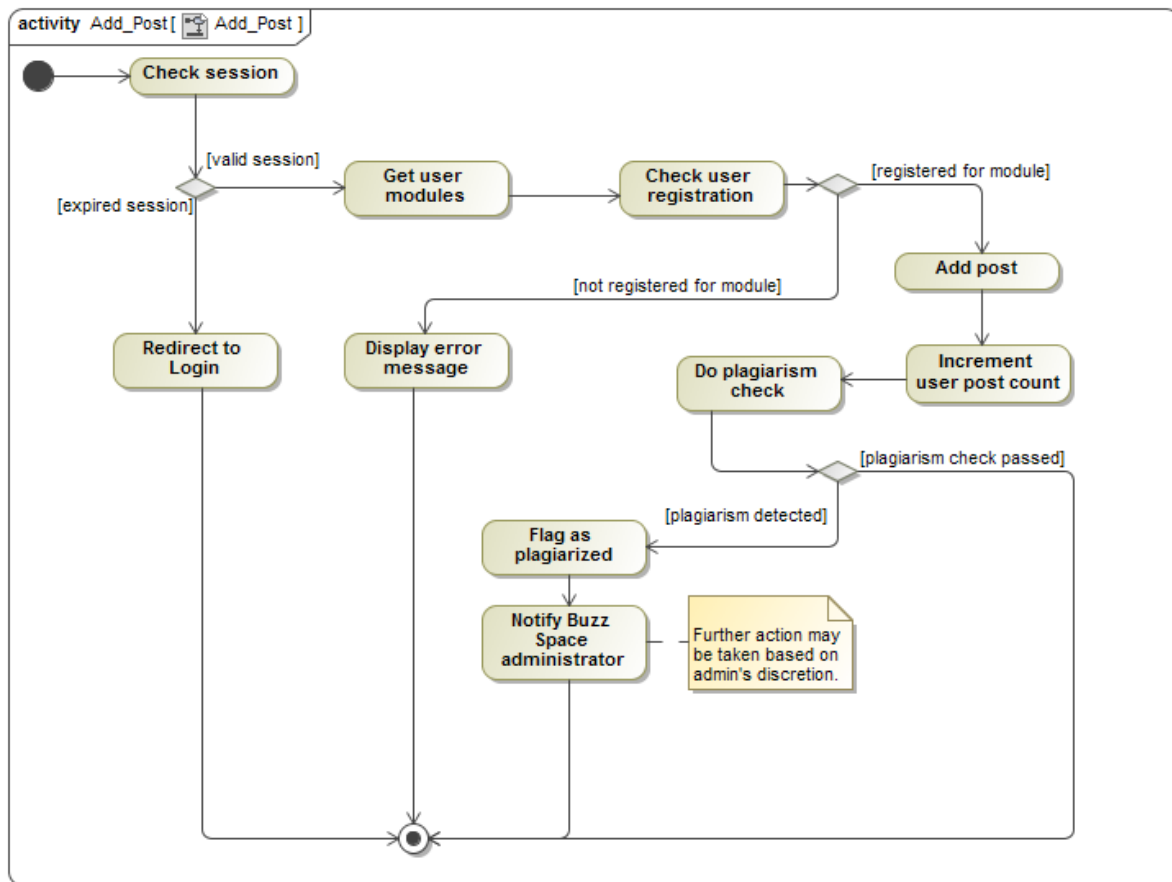


Figure 35: Add post to thread activity diagram.

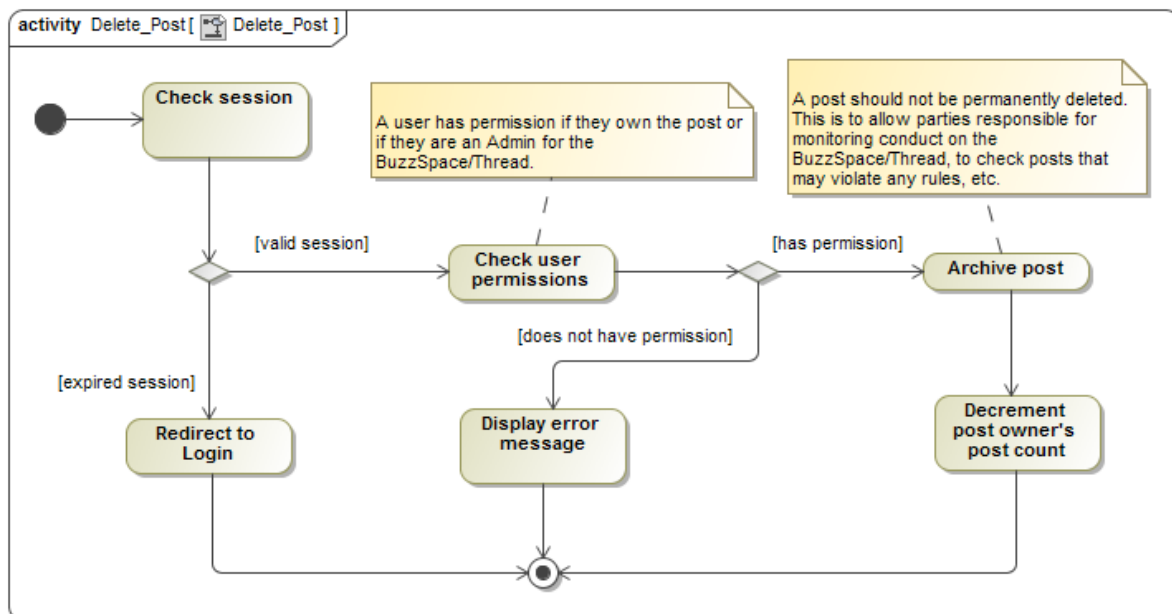


Figure 36: Delete thread post activity diagram.

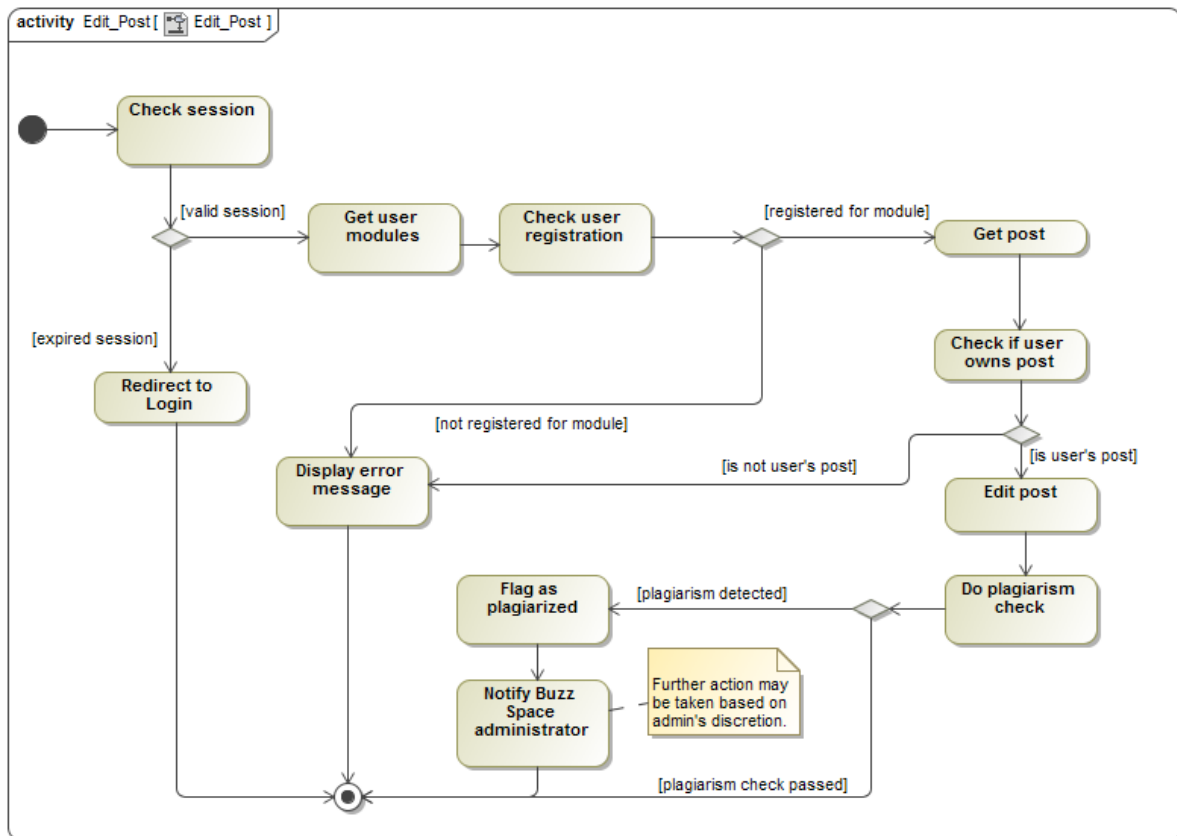


Figure 37: Edit thread post activity diagram.

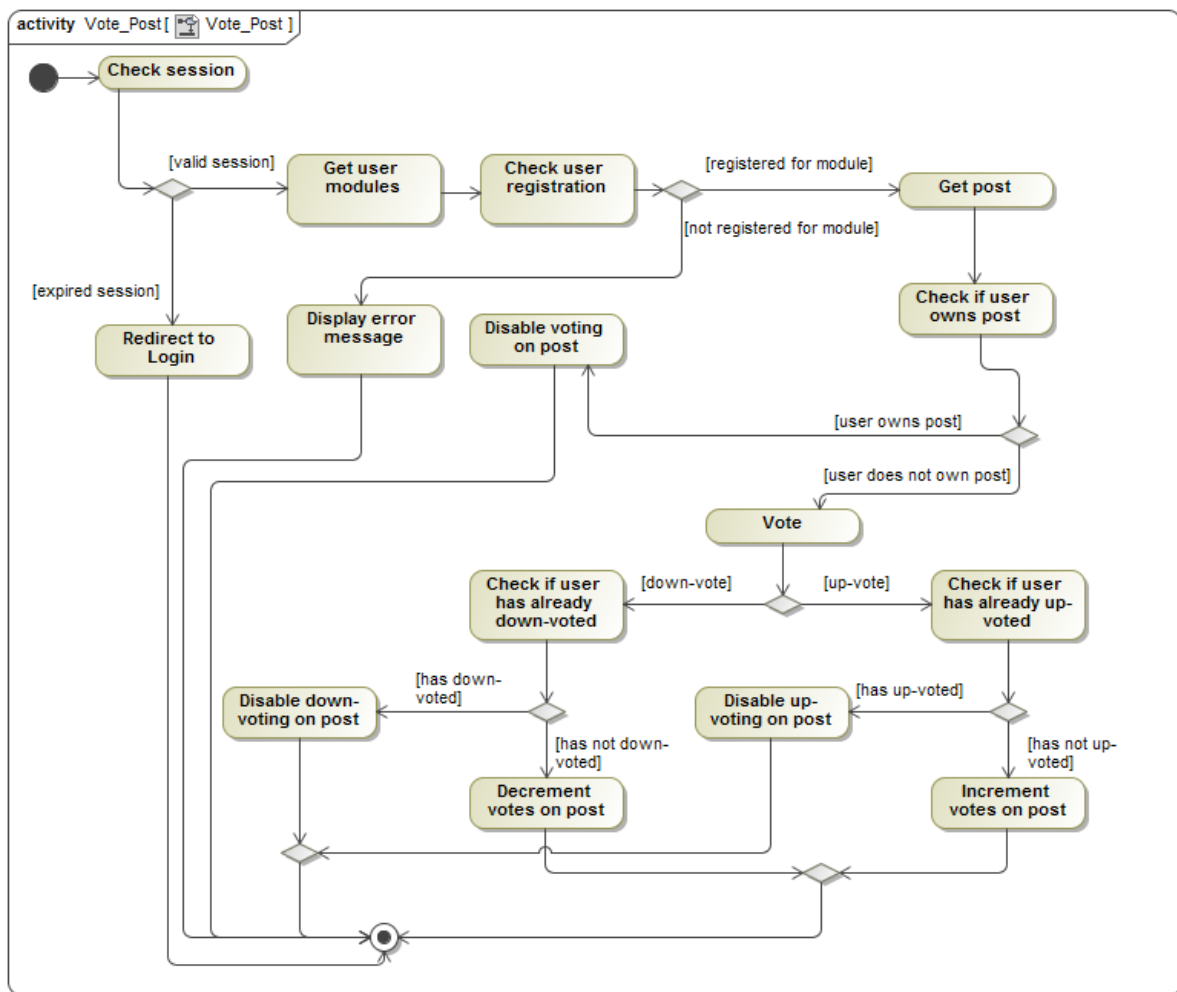


Figure 38: Vote for post activity diagram.

2.5 Domain Model

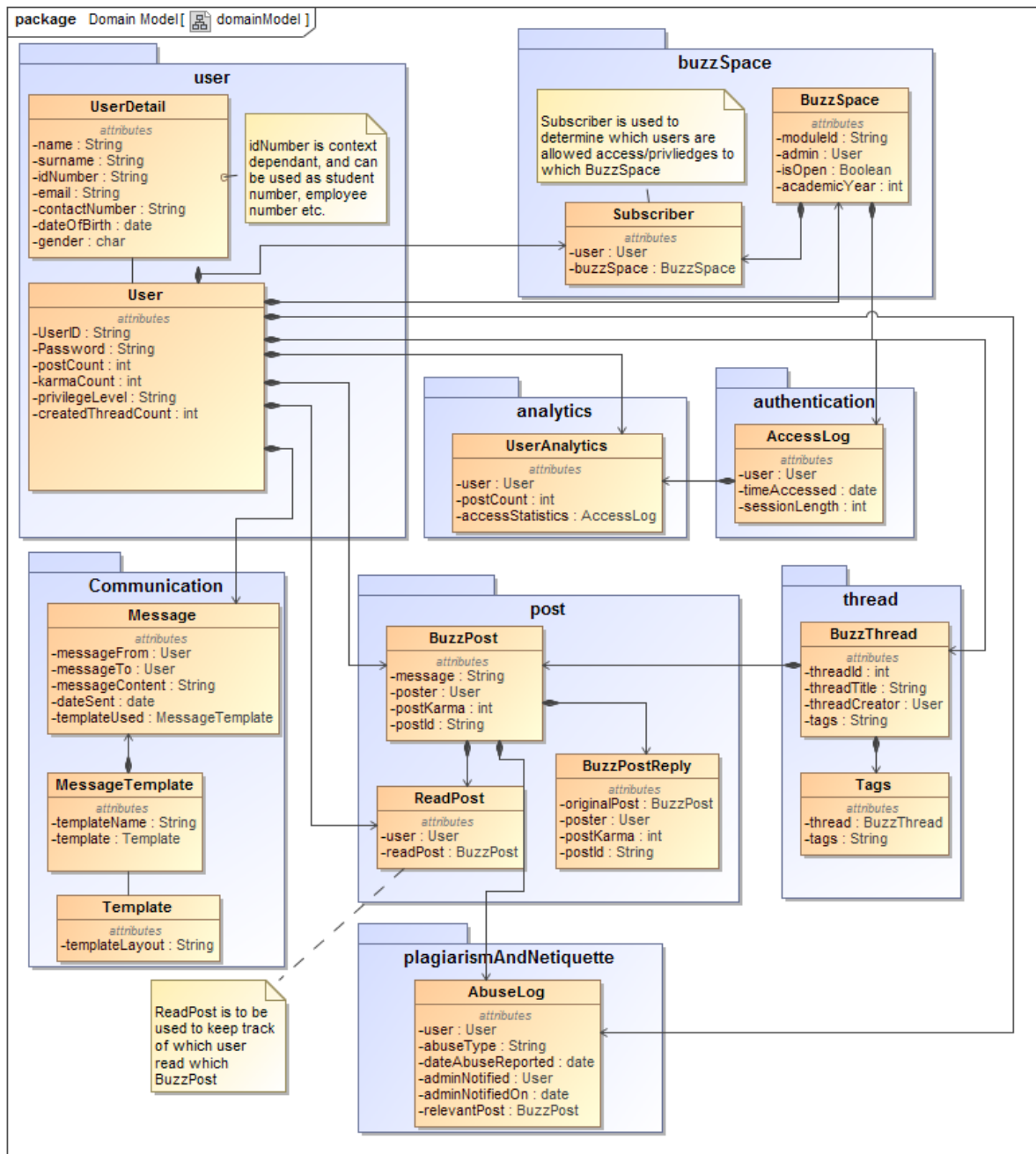


Figure 39: Domain model for the discussion board.