



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

FUNCTIONAL REQUIREMENTS

Project: Figbook

CLIENT: FIGTORY ANIMATION

Team: Creativate

Armand Pieterse *u12167844*

Kgomotso Sito *u12243273*

Jimmy Peleha *u12230830*

Sphelele Malo *u12247040*

Ndivhuwo Nthambeleni *u10001183*

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF PRETORIA

Here's a link to GitHub.

<https://github.com/SpheMalo/COS-301-Main-Project.git>

May 22, 2015

Contents

1	Vision and scope	3
1.1	Project Background	3
1.2	Project vision	3
1.3	Project scope	3
2	Application requirements and design	3
2.1	Authentication	3
2.1.1	scope	3
2.1.2	use cases	3
2.2	Manuscript Management	6
2.2.1	scope	6
2.2.2	use cases	6
2.3	Communication	8
2.3.1	scope	8
2.3.2	use cases	8
2.4	Reporting	10
2.4.1	scope	10
2.4.2	use cases	10

List of Figures

1	Scope for authentication module	3
2	Service contract for registering a new user account	4
3	Service contract for logging a user out of the system	4
4	Service contract for deleting a users account	5
5	Service contract for suspending a users account	5
6	Scope for manuscript management module	6
7	Service contract for creating a new project (manuscript)	7
8	Service contract for deleting a project (manuscript)	8
9	Service contract for linking a project (manuscript) to a social network	9
10	Service contract for editing a project (manuscript)	10
11	Service contract for sending a manuscript	11
12	Service contract for viewing a letter	12
13	Service contract for editing a work schedule	12
14	Service contract for creating a report	13
15	Service contract for viewing a report	13

1 Vision and scope

1.1 Project Background

1.2 Project vision

1.3 Project scope

2 Application requirements and design

This section discusses for each module of the Figbook system the functional requirements as well as the process designs for the use cases and the domain models which must be maintained within persistent storage by that module.

2.1 Authentication

2.1.1 scope

This section provides the details use case requirements for the use cases offered by the Authentication module.

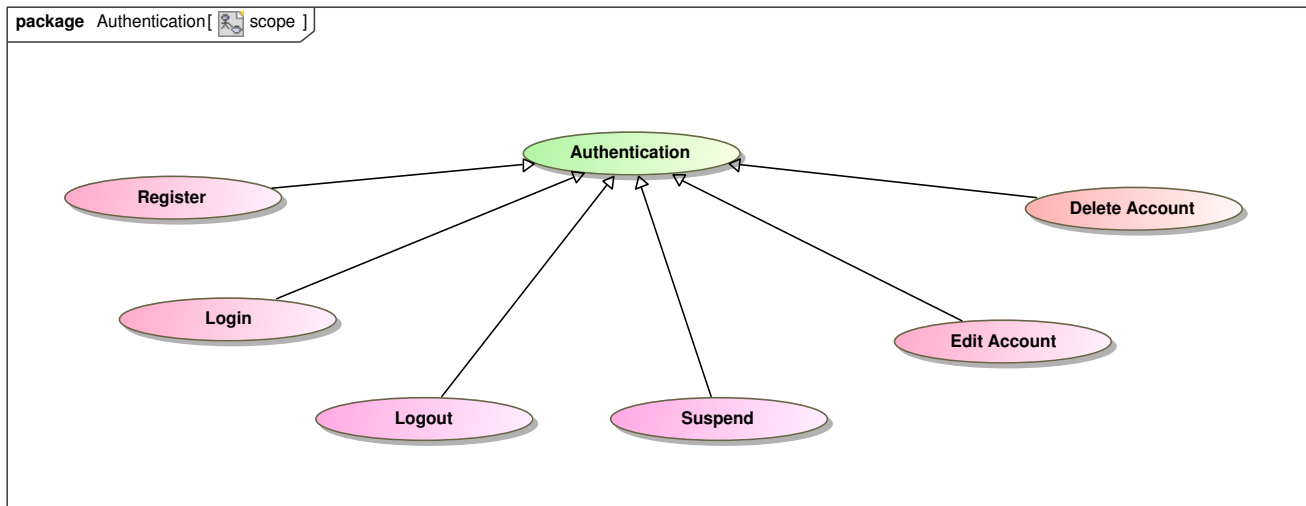


Figure 1: Scope for authentication module

2.1.2 use cases

1. **registerAccount** **priority: important** This use case creates a user account and is persisted through to database.

service contract The service contract for the registerAccount service is shown in figure below. The pre-conditions are enforced (raising the appropriate exception should they not be met) and the requested account is created and persisted through to database.

2. **Login**

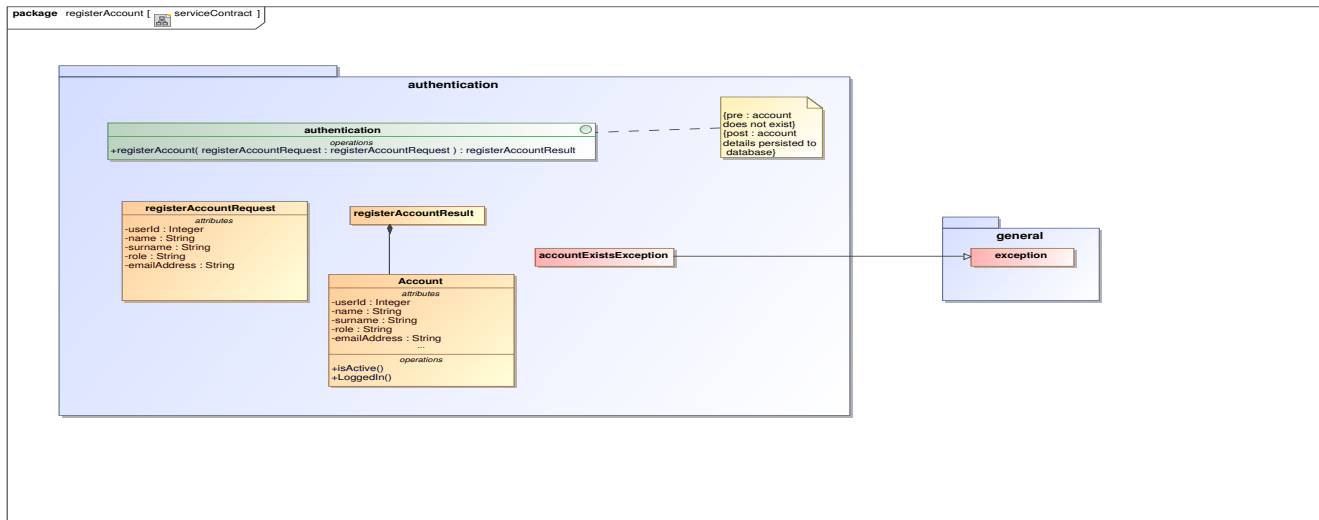


Figure 2: Service contract for registering a new user account

3. Logout - priority: important

This use case allows a user to log out of the system safely.

Service contract: The service contract for logging out is shown below. The preconditions are enforced and the users session is ended, thus logging the user out of the system.

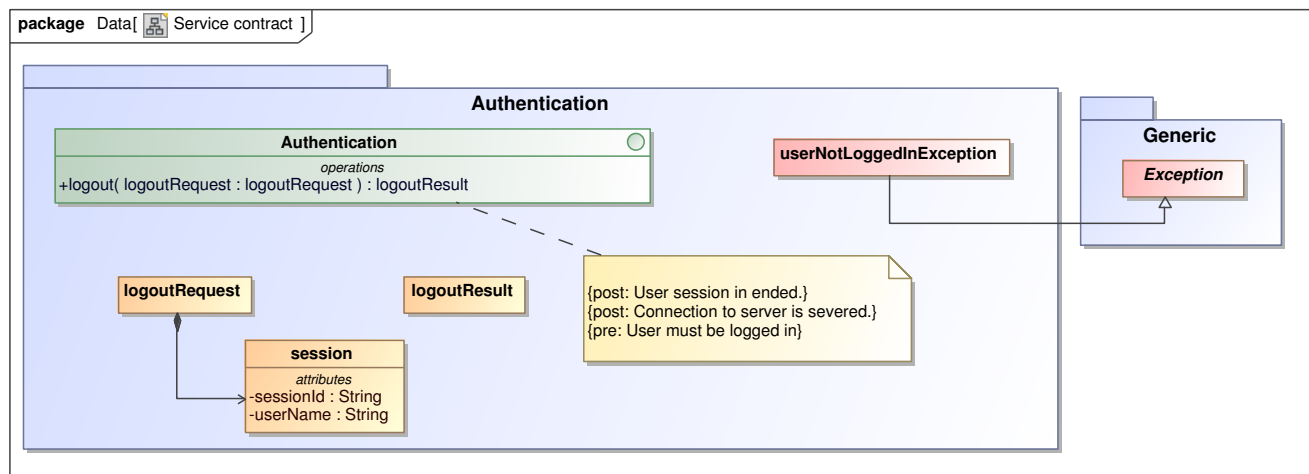


Figure 3: Service contract for logging a user out of the system

4. EditAccount

5. **deleteAccount** **priority: important** This use case creates a user account and is persisted through to database.

service contract The service contract for the deleteAccount service is shown in figure below. The pre-conditions are enforced (raising the appropriate exception should they not be met) and the account of interest is deleted and changes are persisted through to database.

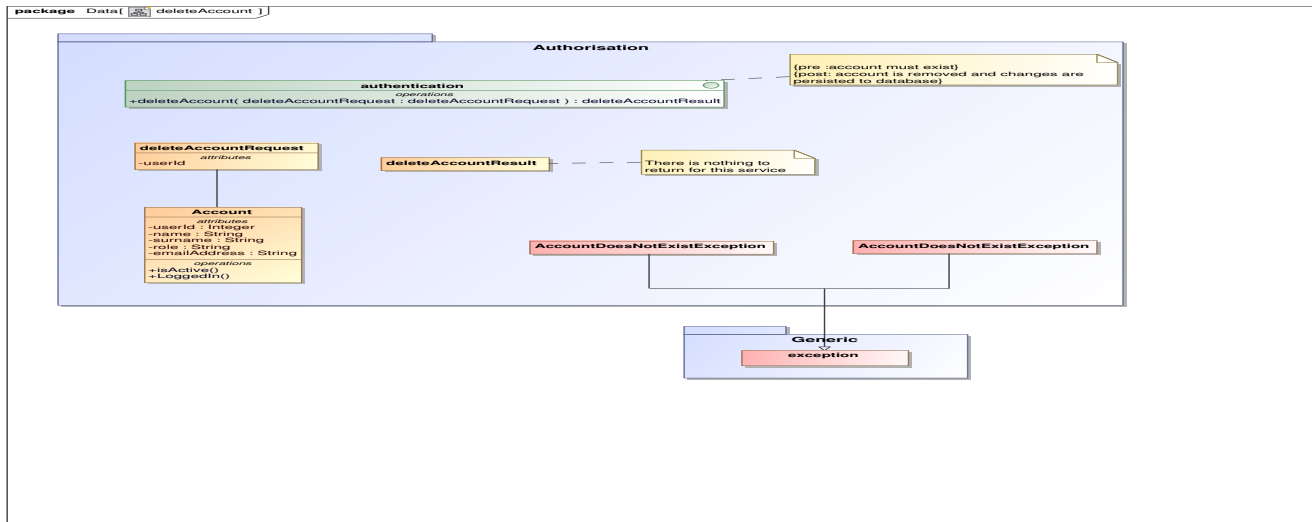


Figure 4: Service contract for deleting a users account

6. SuspendAccount - priority: important

This use case allows the system administrator to suspend a users account for any reason he/she may see as appropriate. A suspended account bars the user from logging in and/or using the account until the administrator lifts the suspension.

service contract: The service contract for the suspendAccount service is shown below.

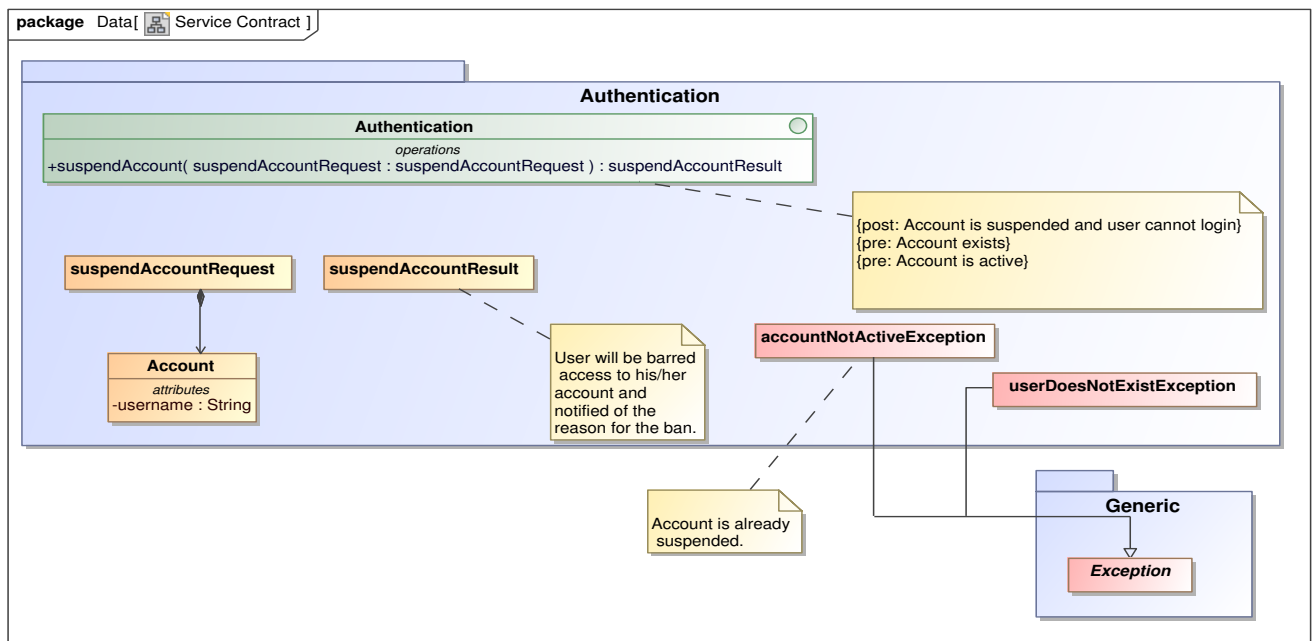


Figure 5: Service contract for suspending a users account

2.2 Manuscript Management

2.2.1 scope

This section provides the details use case requirements for the use cases offered by the Authentication module.

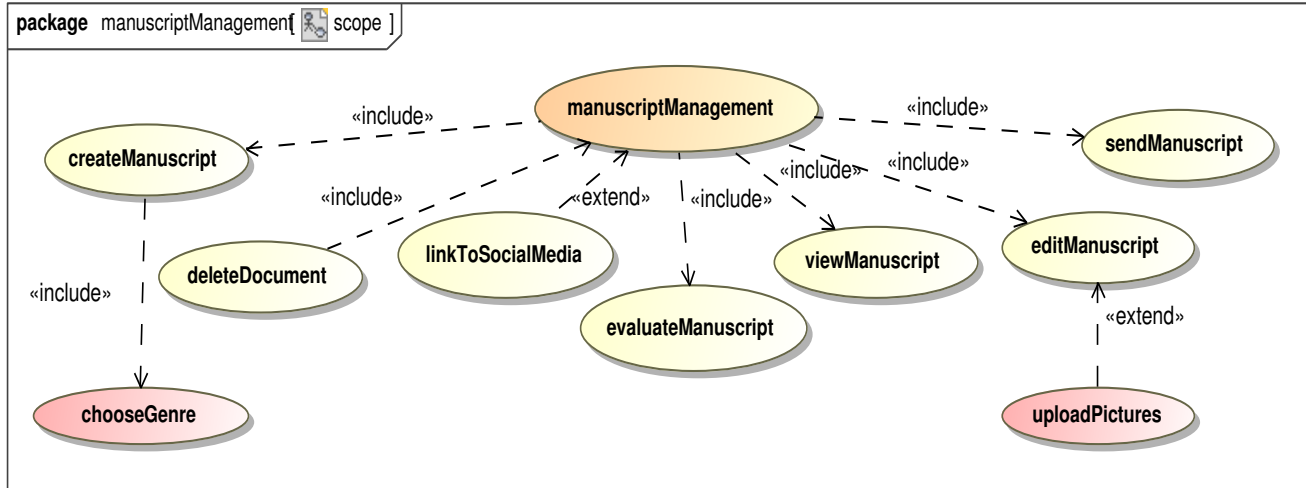


Figure 6: Scope for manuscript management module

2.2.2 use cases

1. Create Manuscript - priority: critical

This use case creates a new project. It allows the user to give the manuscript a name and specify the privacy setting on it.

service contract: below is a figure of the createManuscript service contract.

2. deleteManuscript priority: important

This use case which removes manuscript and persists changes to the database.

service contract: The service contract for the deleteManuscript service is shown in figure below. The pre-conditions are enforced (raising the appropriate exception should they not be met) and the manuscript of interest is deleted and changes are persisted through to database.

3. linkToSocialMedia priority: important

important This use case which link manuscript to social media and create an automated post

service contract: The service contract for the linkToSocialMedia service is shown in figure below. The pre-conditions are enforced (raising the appropriate exception should they not be met) and a post for the manuscript of interest is created.

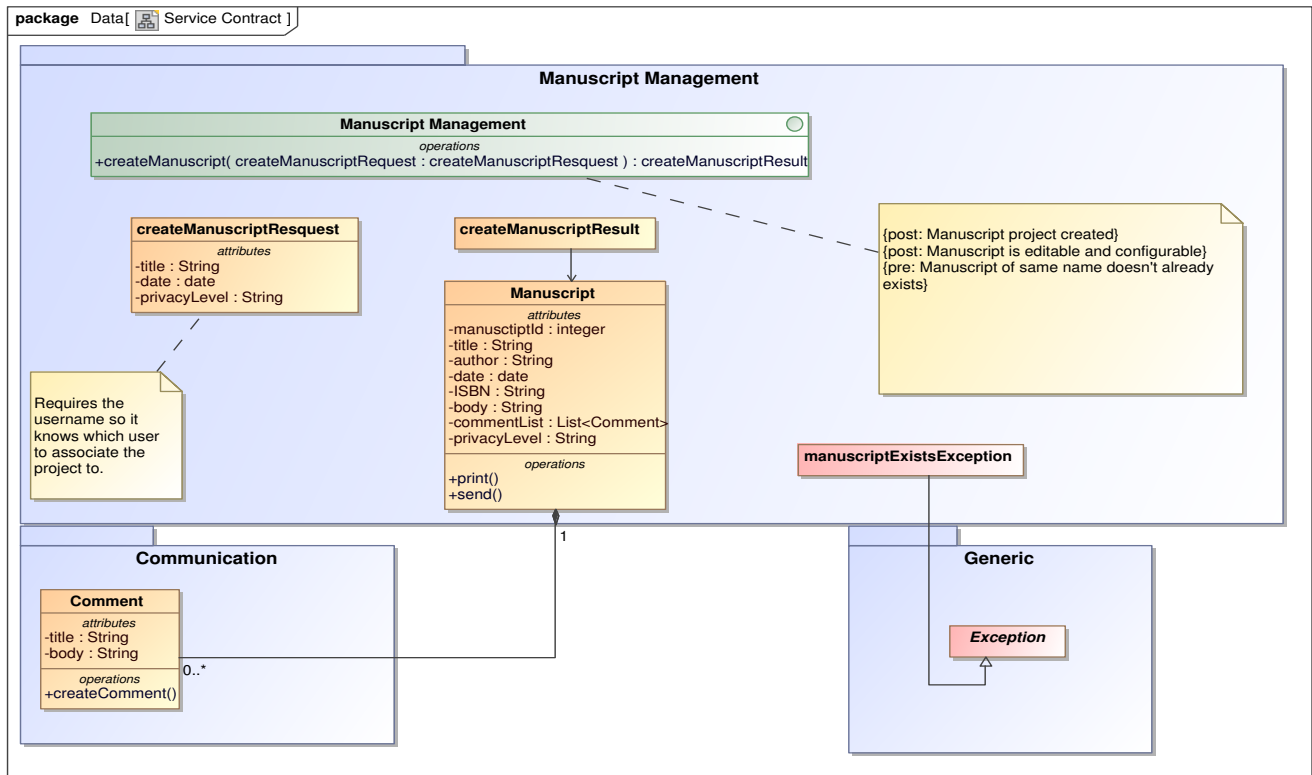


Figure 7: Service contract for creating a new project (manuscript)

4. editManuscript priority: important

This use case which allows one to edit manuscript and changes are persisted to database.

service contract: service contract The service contract for the editManuscript service is shown in figure below. The pre-conditions are enforced (raising the appropriate exception should they not be met) and the manuscript of interest is edited and changes are persisted to database

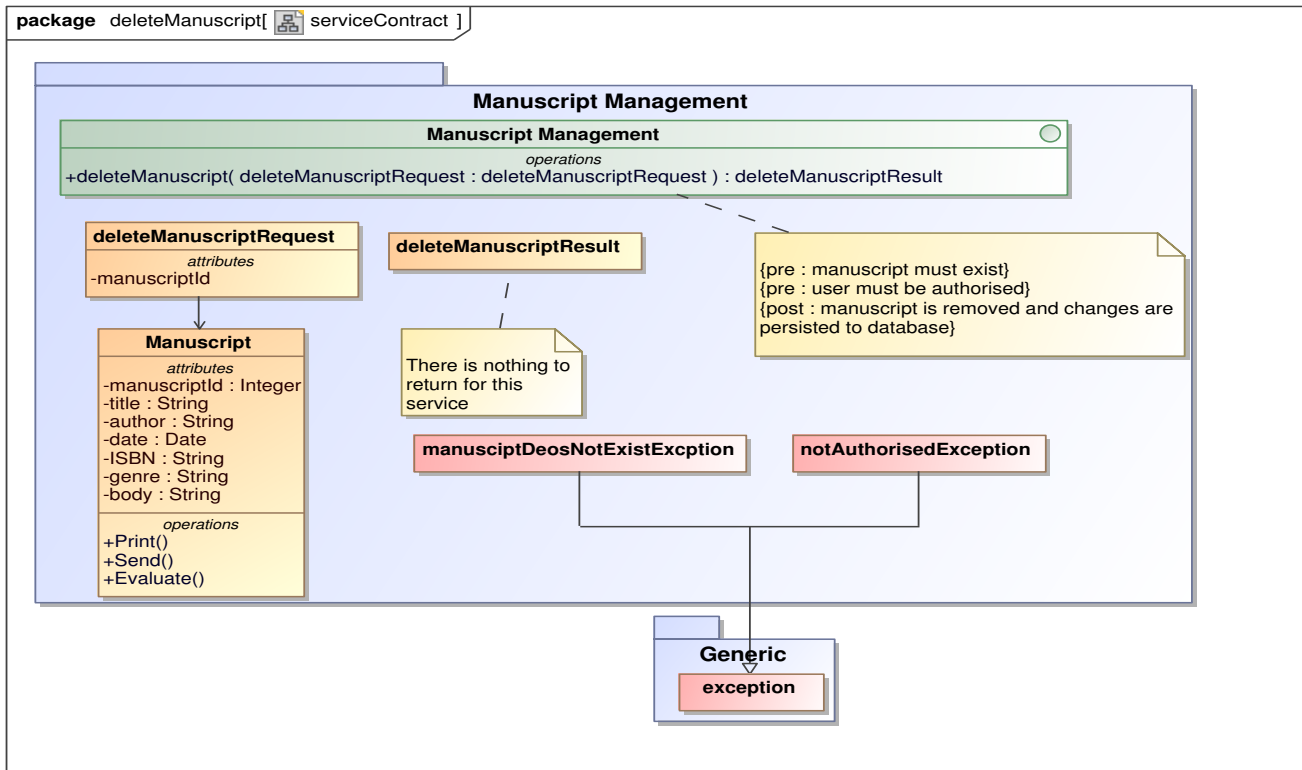


Figure 8: Service contract for deleting a project (manuscript)

5. Evaluate Manuscript

6. Read Manuscript

7. Send Manuscript - priority: important

This use case allows a user to send a another user on the system the manuscript. This allows the recipient to access it and, depending on the priviledges afforded by the owner of the manuscript, be able to read/modify it.

service contract: below is a figure of the sendManuscript service contract.

2.3 Communication

2.3.1 scope

This section provides the details use case requirements for the use cases offered by the Authentication module.

2.3.2 use cases

1. Send Letter

2. View Letter - priority: important

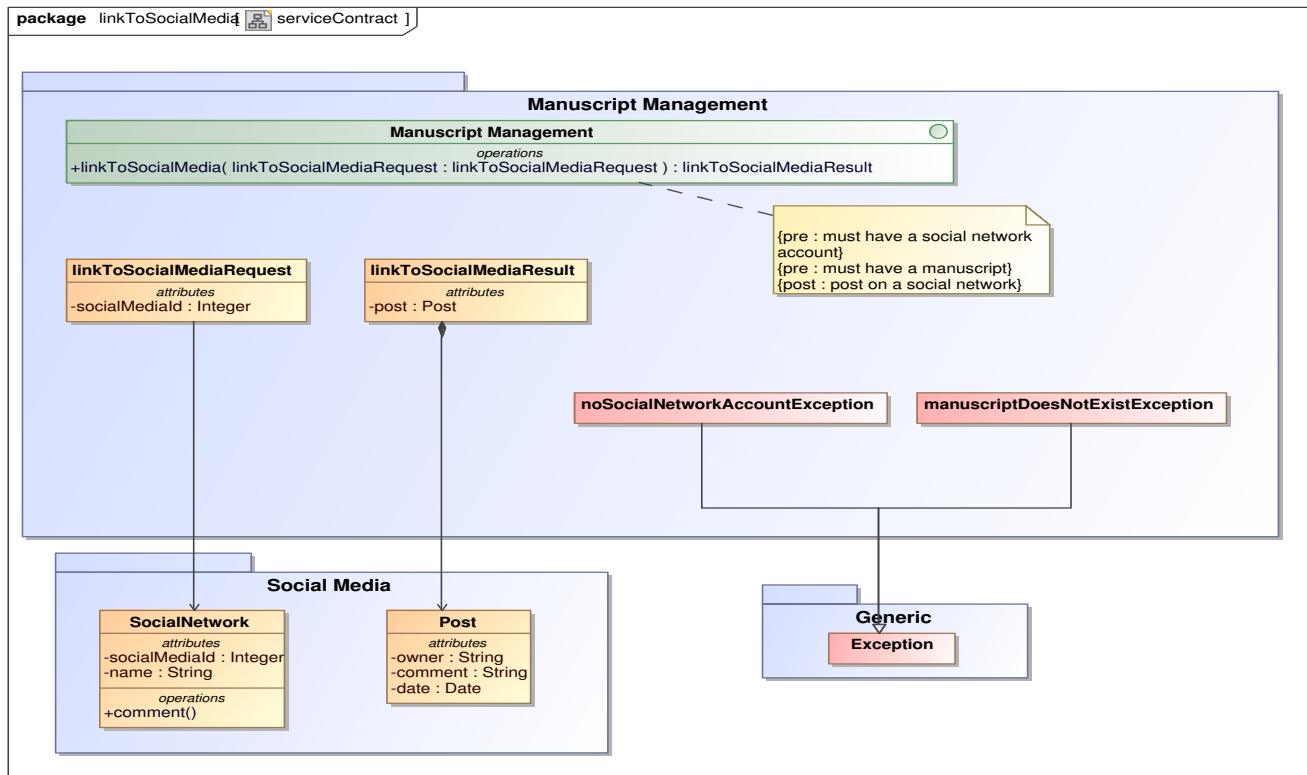


Figure 9: Service contract for linking a project (manuscript) to a social network

This use case allows a user to view one of the letters sent to him/her. The letter can be an editorial letter, contract offer or query letter etc.

service contract: Below is a diagram of a service contract of viewing a letter.

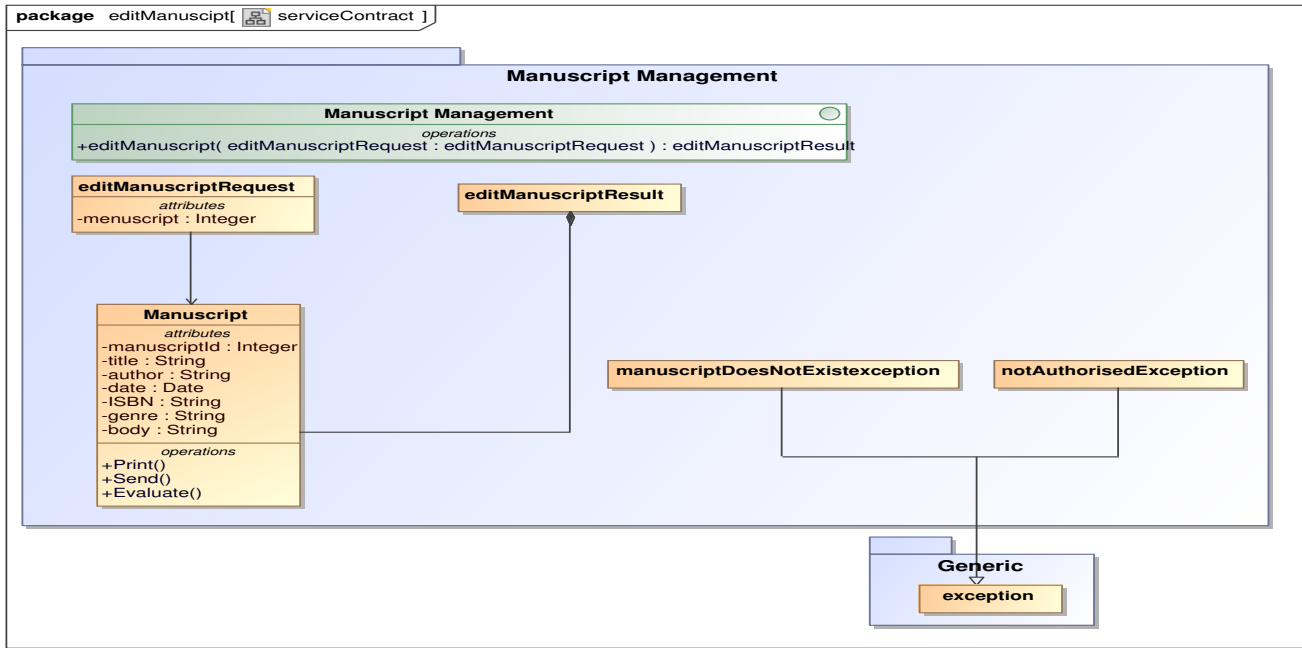


Figure 10: Service contract for editing a project (manuscript)

3. Write Comment

2.4 Reporting

2.4.1 scope

This section provides the details use case requirements for the use cases offered by the Authentication module.

2.4.2 use cases

1. reateSchedule
2. viewSchedule
3. editSchedule - **priority: important**

This use case allows a user to make changes to the work schedule of the book.

service contract: below is the service contract for editing a work schedule

4. **createReport - priority: important**

This use case creates a report based on filters and provides it in printable format.

service contract: The service contract for the createReport service is shown in figure below. The pre-conditions are enforced (raising the appropriate exception should they not be met) and the report is created and return in a printable format.

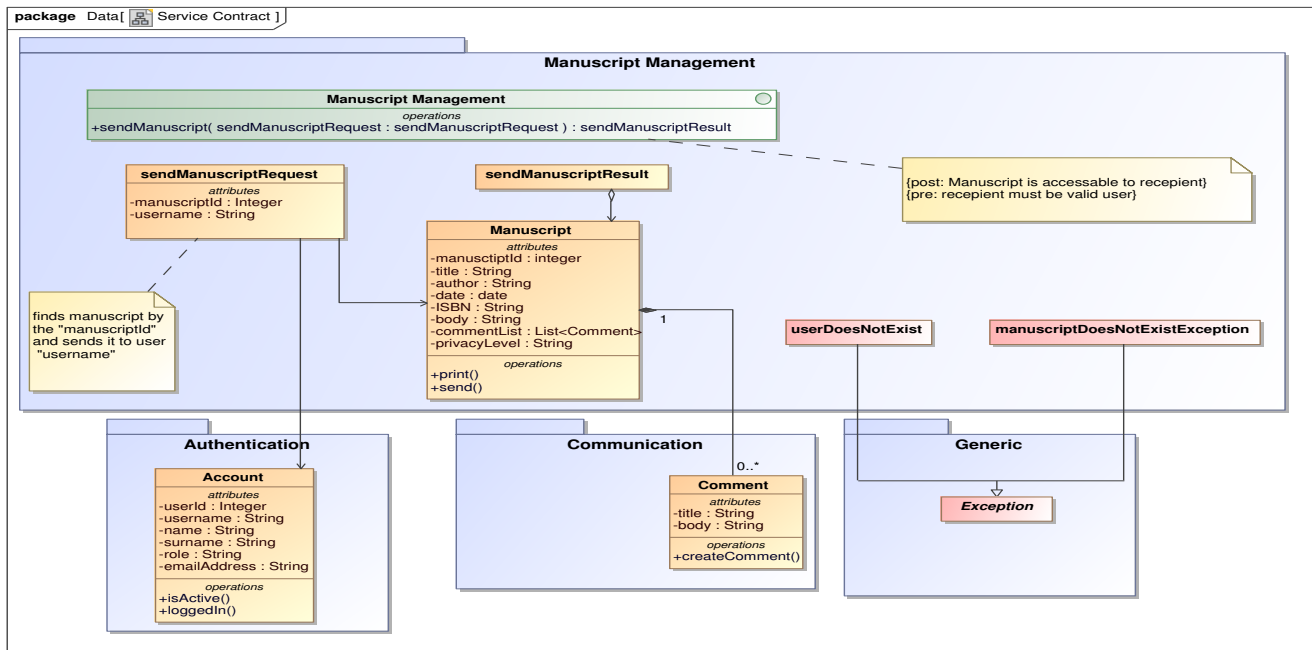


Figure 11: Service contract for sending a manuscript

5. viewReport - priority: important

This use case allows users to view a generated report, on the system (before exporting to say, a pdf for example).

service contract: Below is the service contract for viewing a report.

6. exportReport

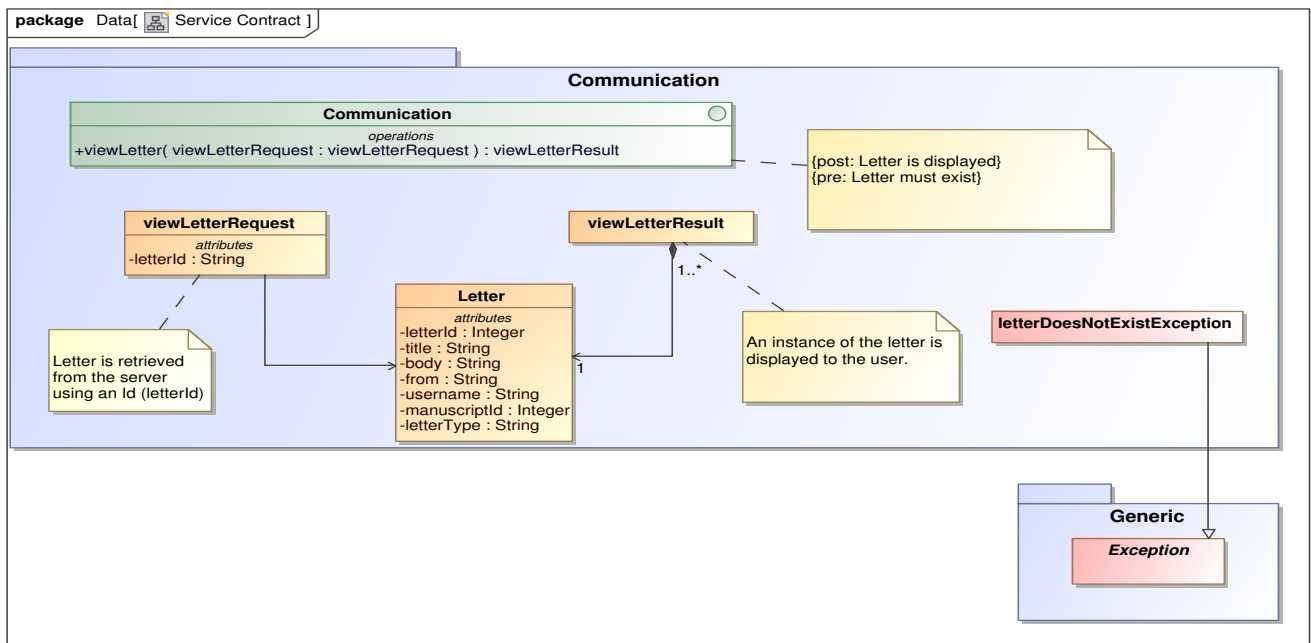


Figure 12: Service contract for viewing a letter

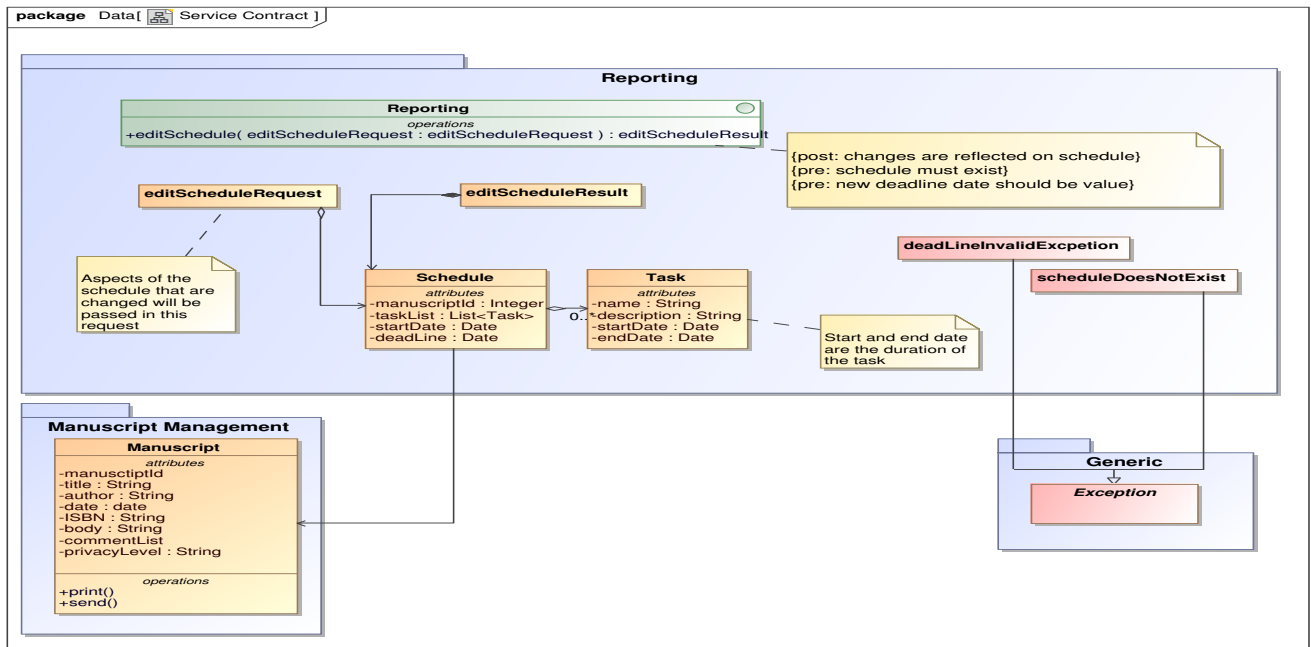


Figure 13: Service contract for editing a work schedule

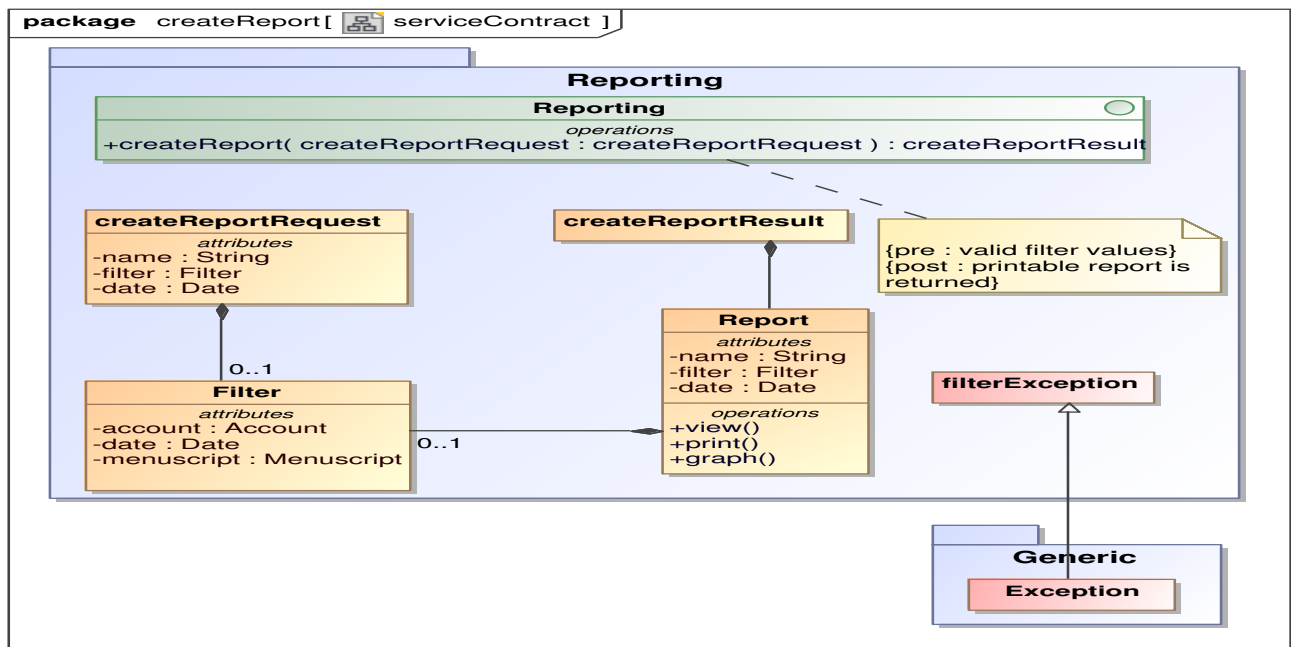


Figure 14: Service contract for creating a report

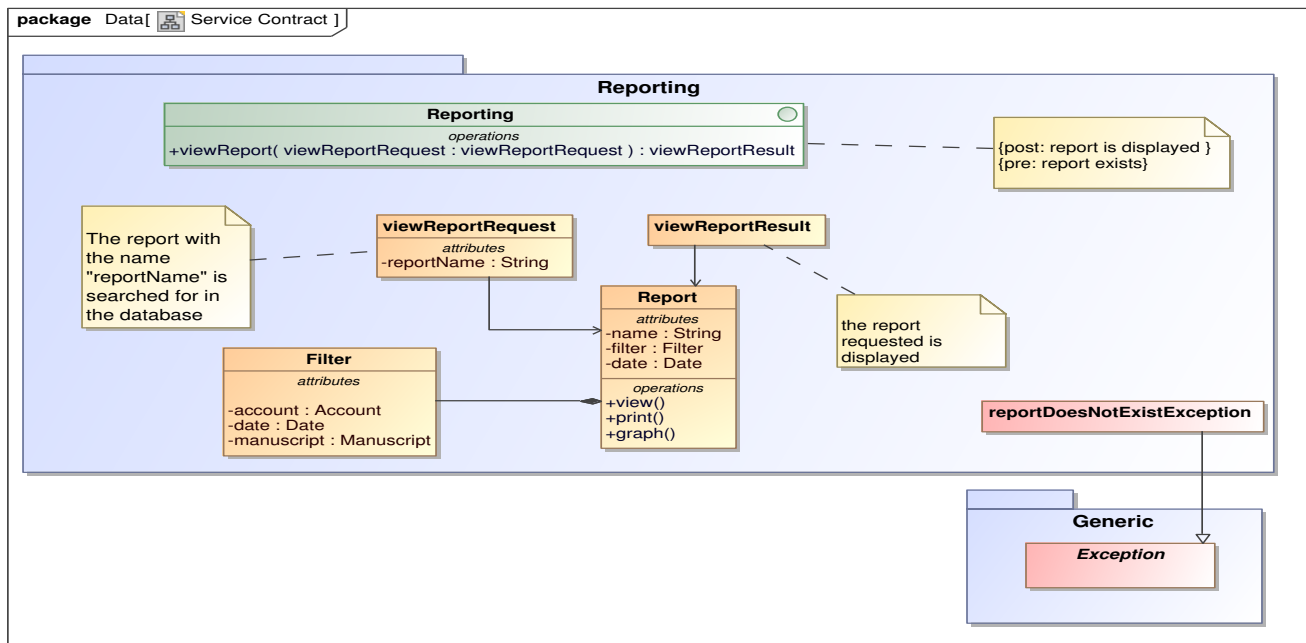


Figure 15: Service contract for viewing a report