



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 25, 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	4
2.1	Authentication	4
2.1.1	scope	4
2.1.2	Use Cases	5
2.1.3	Domain Model	10
2.2	Manuscript Management	11
2.2.1	Scope	11
2.2.2	Use Cases	12
2.2.3	Domain Model	19
2.3	Communication	20
2.3.1	scope	20
2.3.2	Use Cases	21
2.3.3	Domain Model	24
2.4	Reporting	25
2.4.1	Scope	25
2.4.2	Use Cases	25
2.4.3	Domain Model	28
2.5	Production	29
2.5.1	scope	29
2.5.2	Use Cases	30
2.5.3	Domain Model:	36
2.6	Design	37
2.6.1	scope	37
2.6.2	Use Cases	38

List of Figures

1	Scope for authentication module	4
2	Service contract for registering new user account	5
3	Service contract for logging into a user account	6
4	Service contract for logging a user out of the system	6
5	Service contract for editing a users account	7
6	Service contract for deleting a users account	8
7	Service contract for suspending a users account	9
8	Domain model of Authentication module	10
9	Scope for manuscript management module	11
10	Service contract for creating a new project (manuscript)	12
11	Service contract for deleting a project (manuscript)	13
12	Service contract for linking a project (manuscript) to a social network	14
13	Service contract for editing a project (manuscript)	15
14	Service contract for evaluating a manuscript	16
15	Process Specification for evaluating a manuscript	16
16	Service contract for reading a manuscript	17
17	Service contract for sending a manuscript	18
18	Domain model for managing a manuscript	19
19	Scope of Communication module	20
20	Service contract for sending a letter	21
21	Service contract for viewing a letter	22
22	Service contract for writing a comment a manuscript	23
23	Service contract for writing a comment a manuscript	24
24	Scope for reporting module	25
25	Service contract for creating a work schedule	25
26	Service contract for viewing a work schedule	26
27	Service contract for editing a work schedule	26
28	Service contract for creating a report	27
29	Service contract for viewing a report	27
30	Service contract for exporting a report	28
31	The domain model for the reporting module.	28
32	Scope for Production module	29
33	Service contract for registering a manuscript's ISBN number.	30
34	Service contract for selecting a copy editor	32
35	Service contract for selecting a proof reader	33
36	Service contract for selecting a design team	34
37	Service contract for evaluating a publish	35
38	Domain model for managing a manuscript	36
39	Scope of Design module	37
40	Service contract for structuring a menuscrypt	38
41	Service contract for creating a template design	39
42	Service contract for creating a book cover	40
43	Service contract for uploading a graphic	41

1 Vision and scope

1.1 Project Background

In the world of publishing; how books, magazines, graphic novels and other academic text are produced predominantly using traditional means of publishing. There has not been much drive to leverage the latest developments in technology.

The traditional manner of publishing allows limited room for collaboration and makes the production life-cycle for a book long and tedious. The process of producing the book is still limited to the editor and sub-editor working in collaboration, quality assurance remains the sole responsibility of the publisher.

This has constrained the amount of books publishing houses produce over a set period of time. The inability to seamlessly collaborate on text and quality assurance extends the production life cycle of a book.

1.2 Project vision

Figbook aims to address the issue mentioned above. We intend to provide an application that will allow authors, agents, editors, copy-editors and designers to work on the book in an environment that encourages collaborative efforts and simplifies the life-cycle.

The system will allow multiple authors to concurrently work on a project together so as to reduce the time spent on multi-authored works.

1.3 Project scope

The core of this project is to allow multiple authors to work on a book concurrently. The system should allow smooth communication between different user roles at the different phases of the books life cycle.

The application will allow the authors to work on and see their project go from its infancy to the design and finally publishing phase.

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 module takes care of everything involving the users account. This includes functionality like logging in and out of the system, editing of user accounts etc. Further description is provided in the use cases below

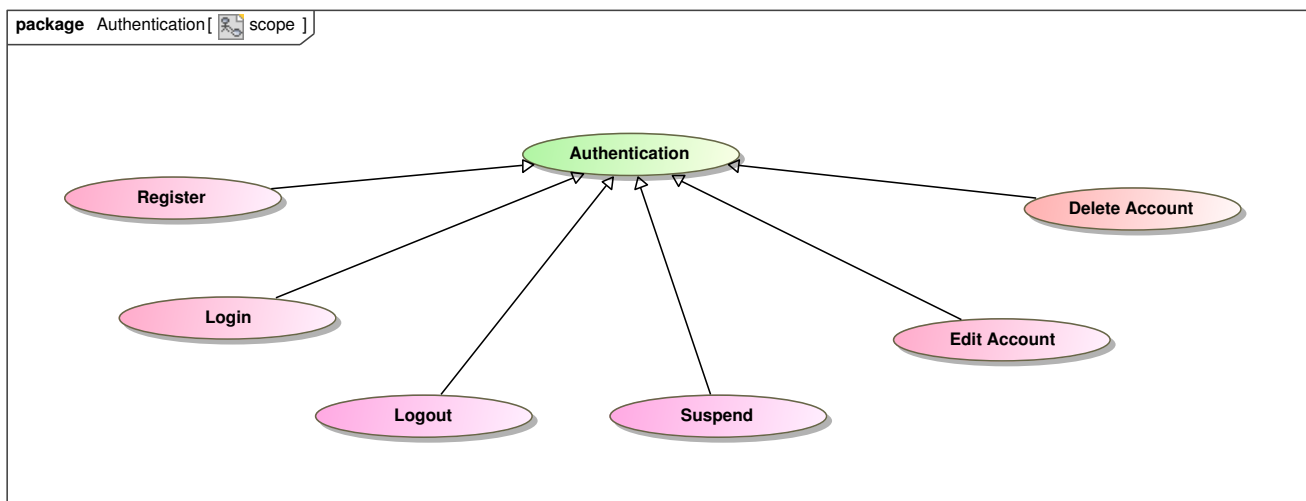


Figure 1: Scope for authentication module

2.1.2 Use Cases

1. **registerAccount** **priority: critical** This use case creates a user account that gets persisted to database.

Service Contract:

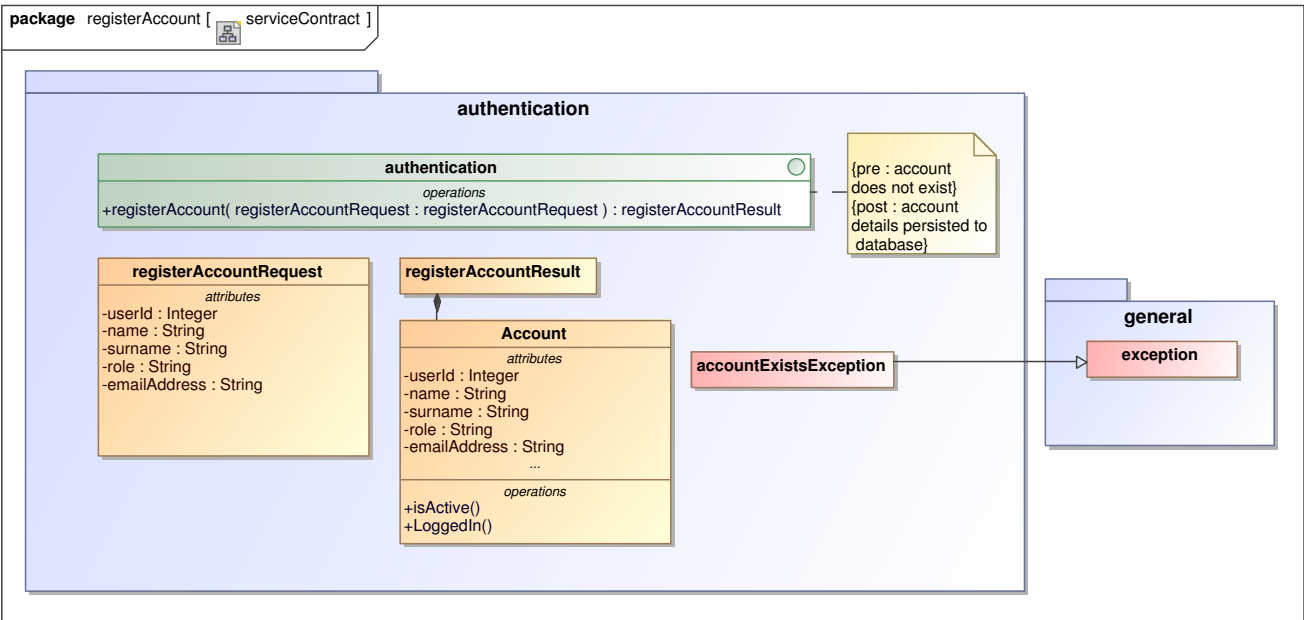


Figure 2: Service contract for registering new user account

2. **Login - priority: critical** This use case allows a user to log into an existing account.

Service Contract The service contract for the login service is shown in figure below.

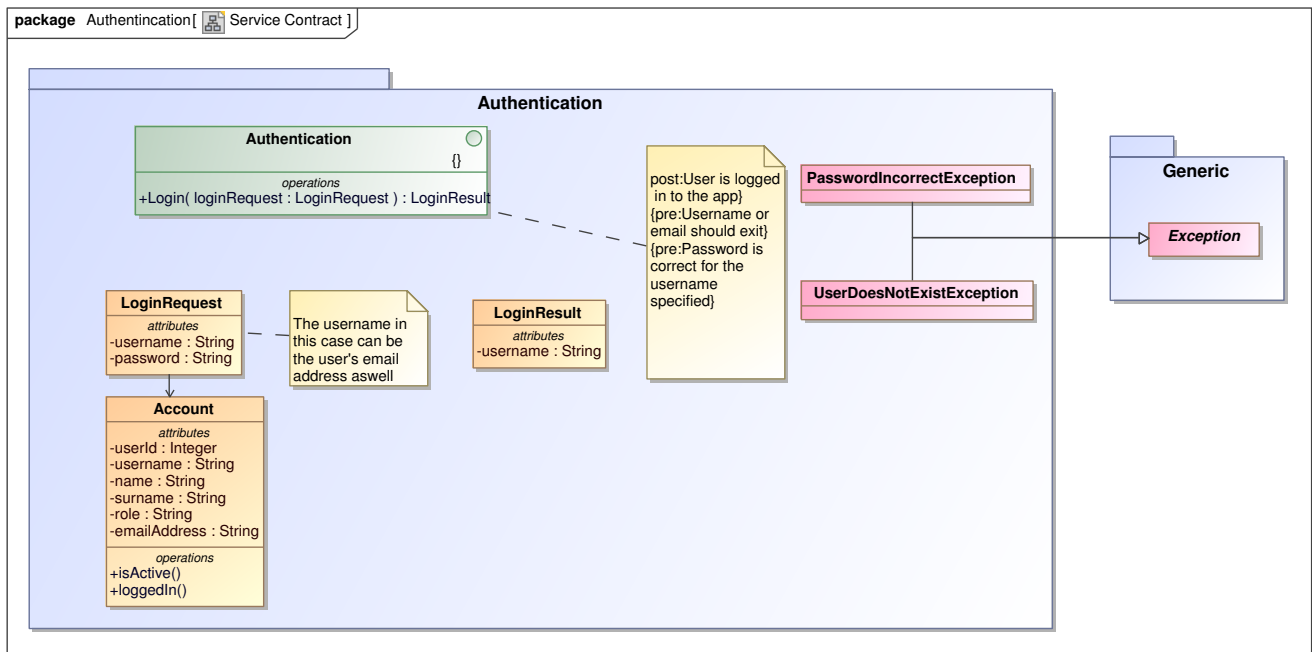


Figure 3: Service contract for logging into a user account

3. **Logout - priority: important**

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

Service Contract:

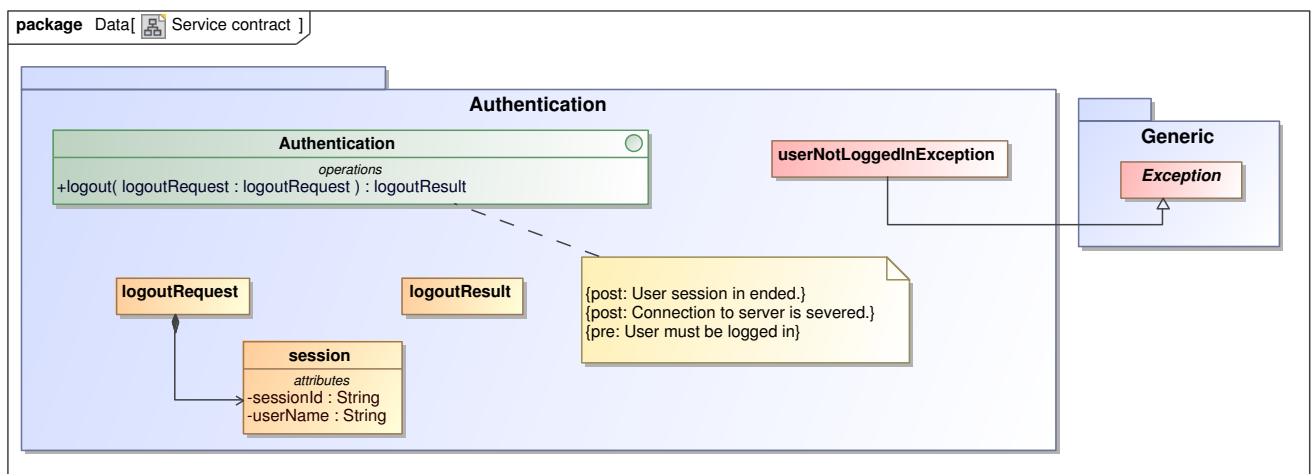


Figure 4: Service contract for logging a user out of the system

4. editAccount - priority: important

This allows a user to make changes to their profile details. They will be allowed to edit the details they added when creating the account.

Service Contract:

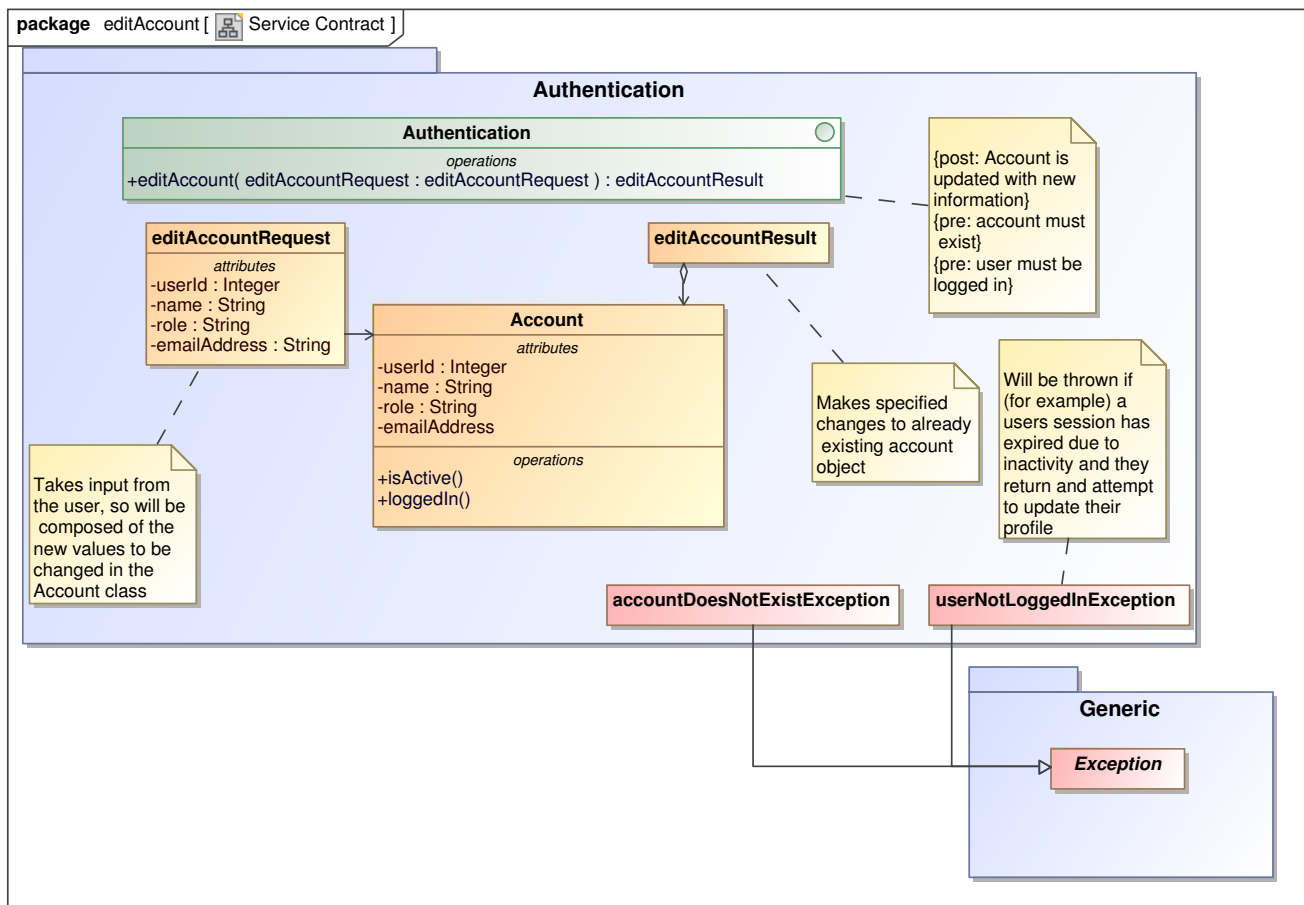


Figure 5: Service contract for editing a users account

5. **deleteAccount** **priority: important** This use case creates a user account that gets persisted to database.

Service Contract:

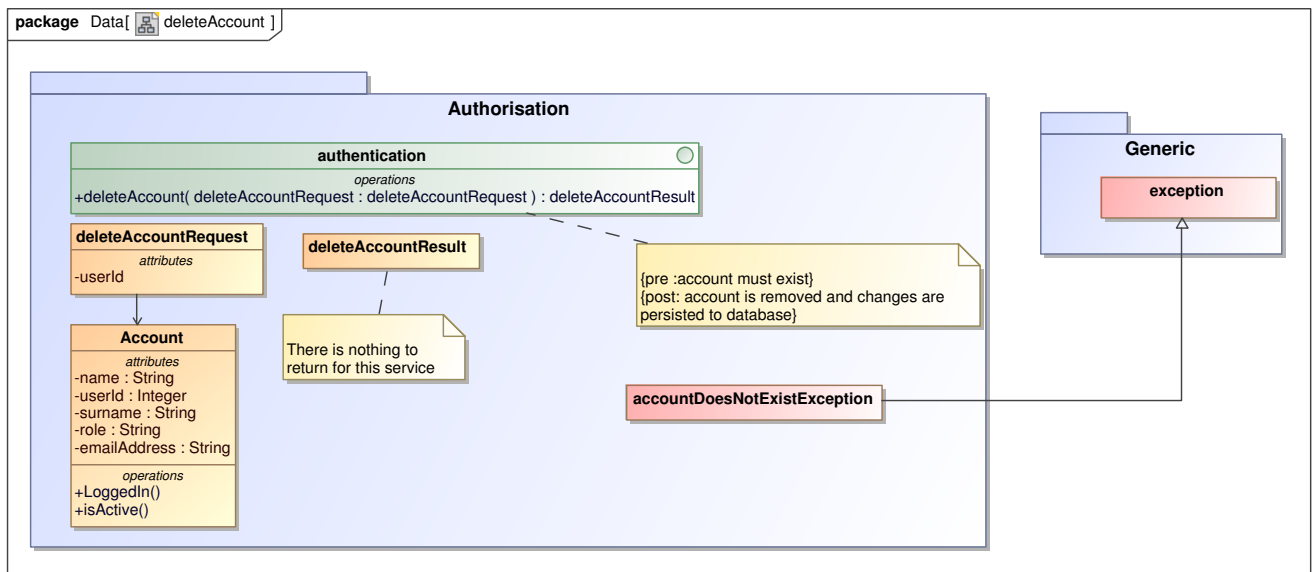


Figure 6: Service contract for deleting a users account

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:

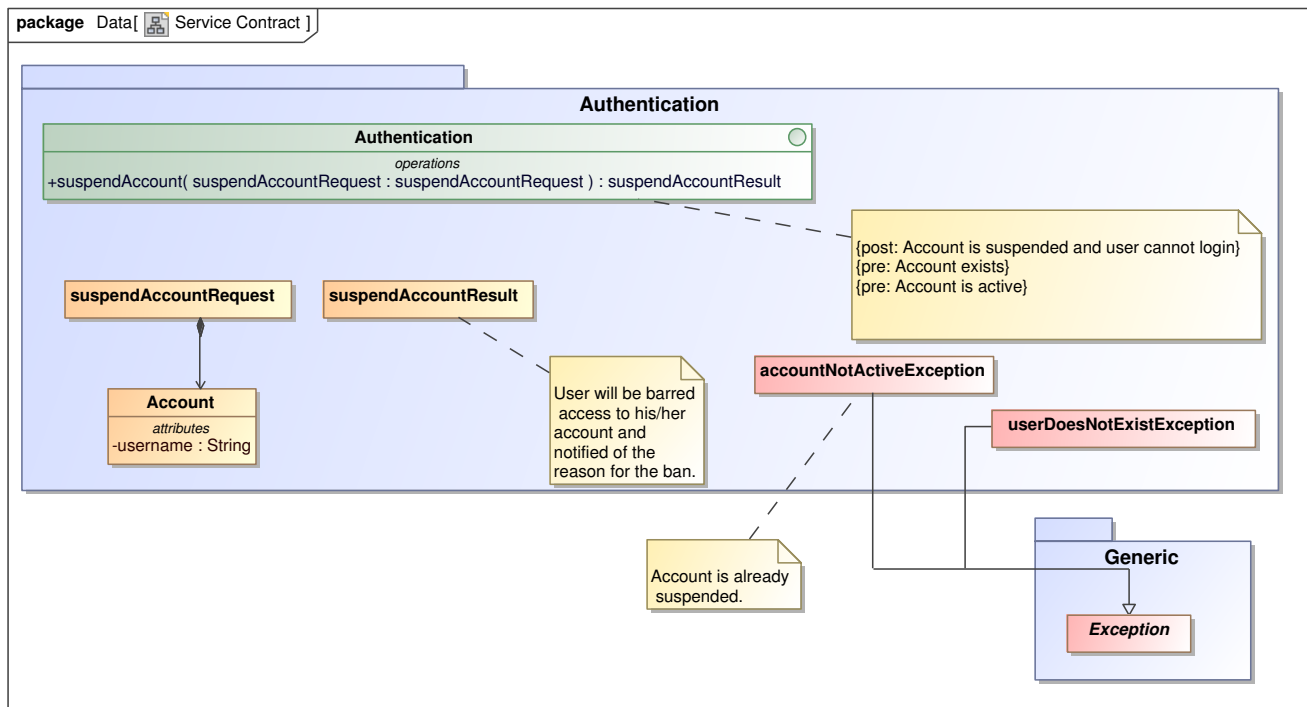


Figure 7: Service contract for suspending a users account

2.1.3 Domain Model

The Authentication module will keep Account objects in the database. Any part of the system that checks for user authorisation will check against the account table to see if that particular user has authorisation to do whatever it is they are attempting to do.

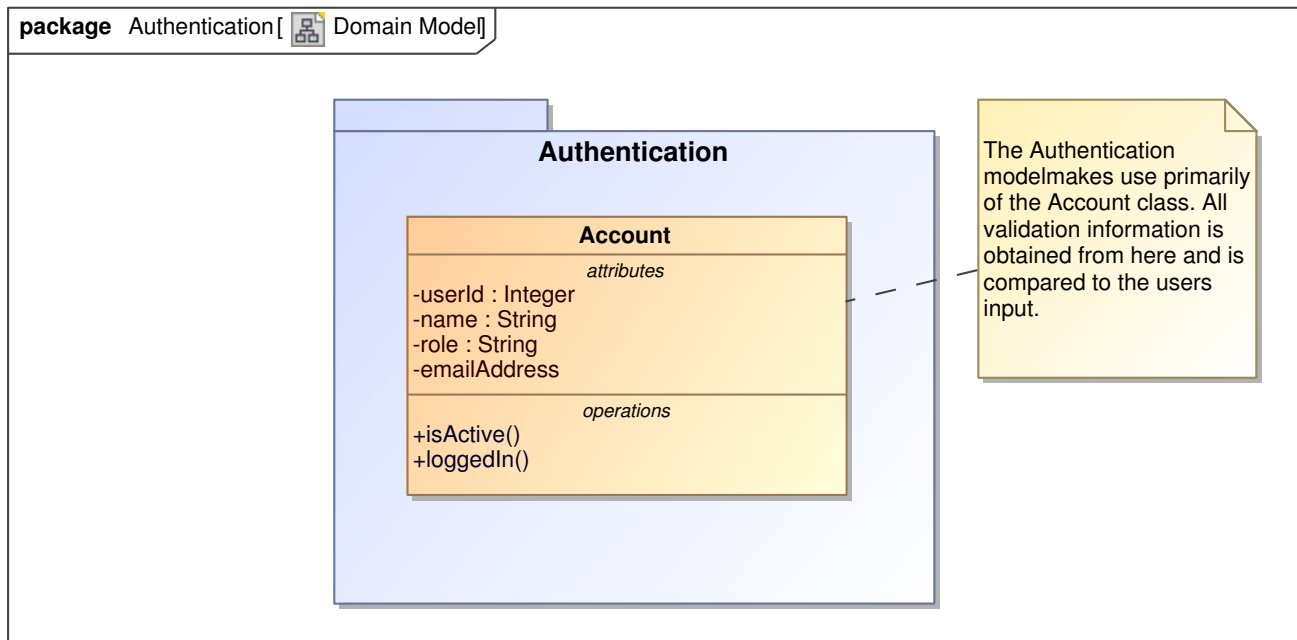


Figure 8: Domain model of Authentication module

2.2 Manuscript Management

2.2.1 Scope

This section provides the detailed service contracts and activity diagrams(when needed) for the use cases offered by the Manuscript Management module. Its services includes all the needed functionality to create, read, evaluate, edit and to send it to the next role in the life-cycle of creating a quality book.

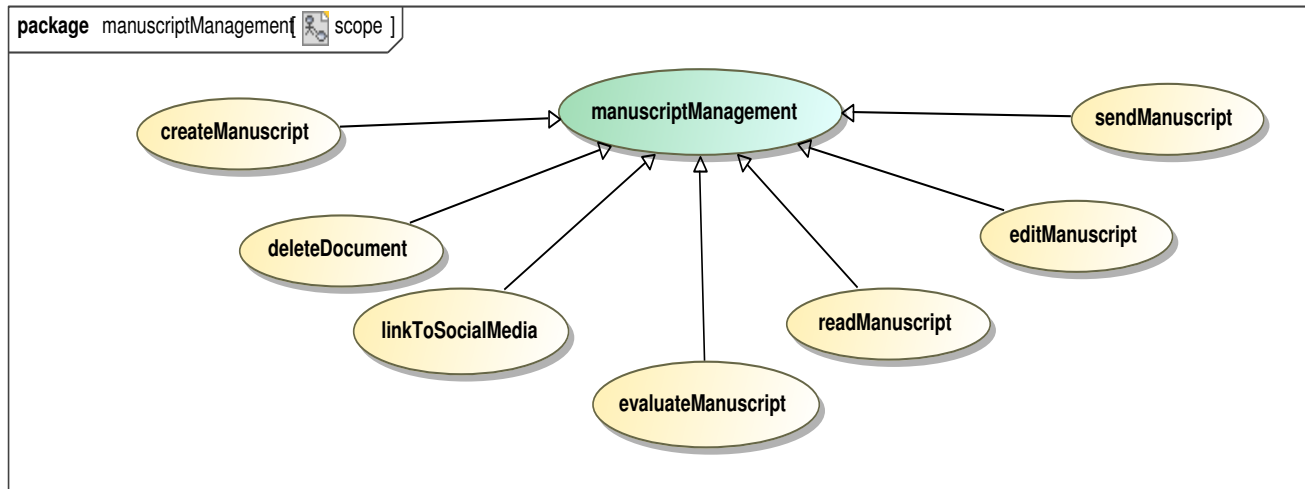


Figure 9: Scope for manuscript management module

2.2.2 Use Cases

1. Create Manuscript - priority: critical

This use case allows an author to create a new project, by creating a new manuscript. The author will be able to give the manuscript a name(title) and specify the privacy setting on it. There may be one or multiple authors collaborating on a manuscript. A manuscript's title should be unique.

Service Contract:

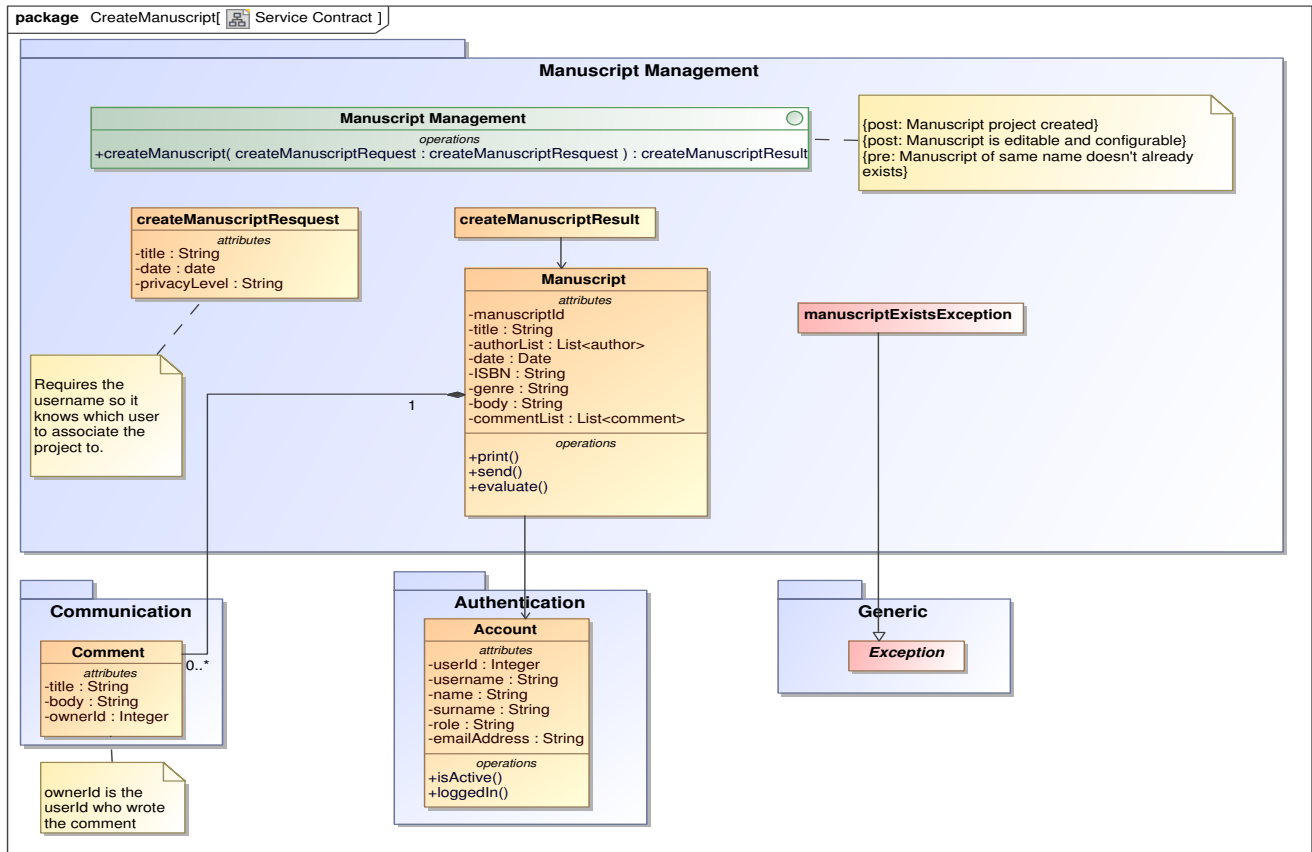


Figure 10: Service contract for creating a new project (manuscript)

2. Delete Manuscript priority: important

This use case removes an existing manuscript. 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.

Service Contract:

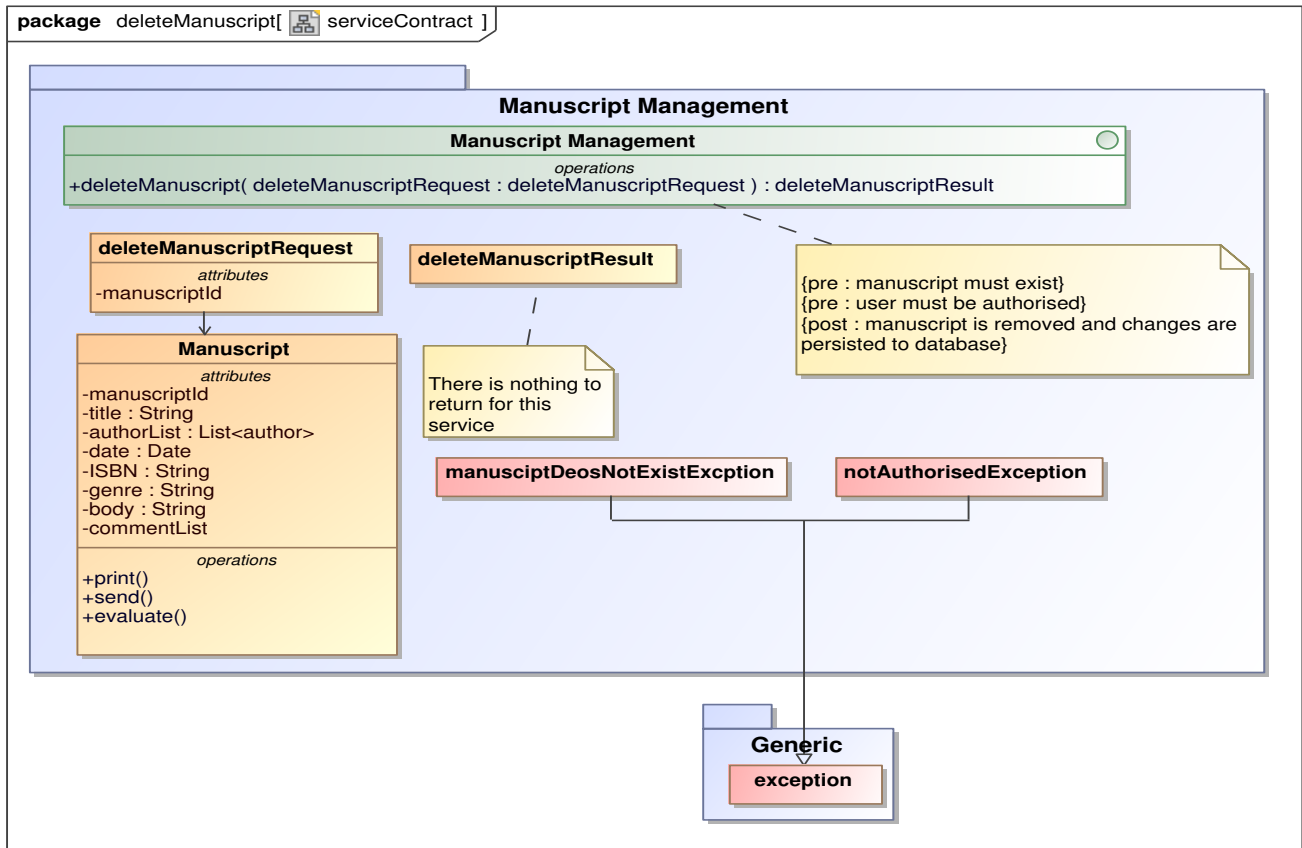


Figure 11: Service contract for deleting a project (manuscript)

3. Link To Social Media priority: important

This use case links a project(manuscript) to a social media of choice if conditions are met and creates an automated post on that social media.

Service Contract:

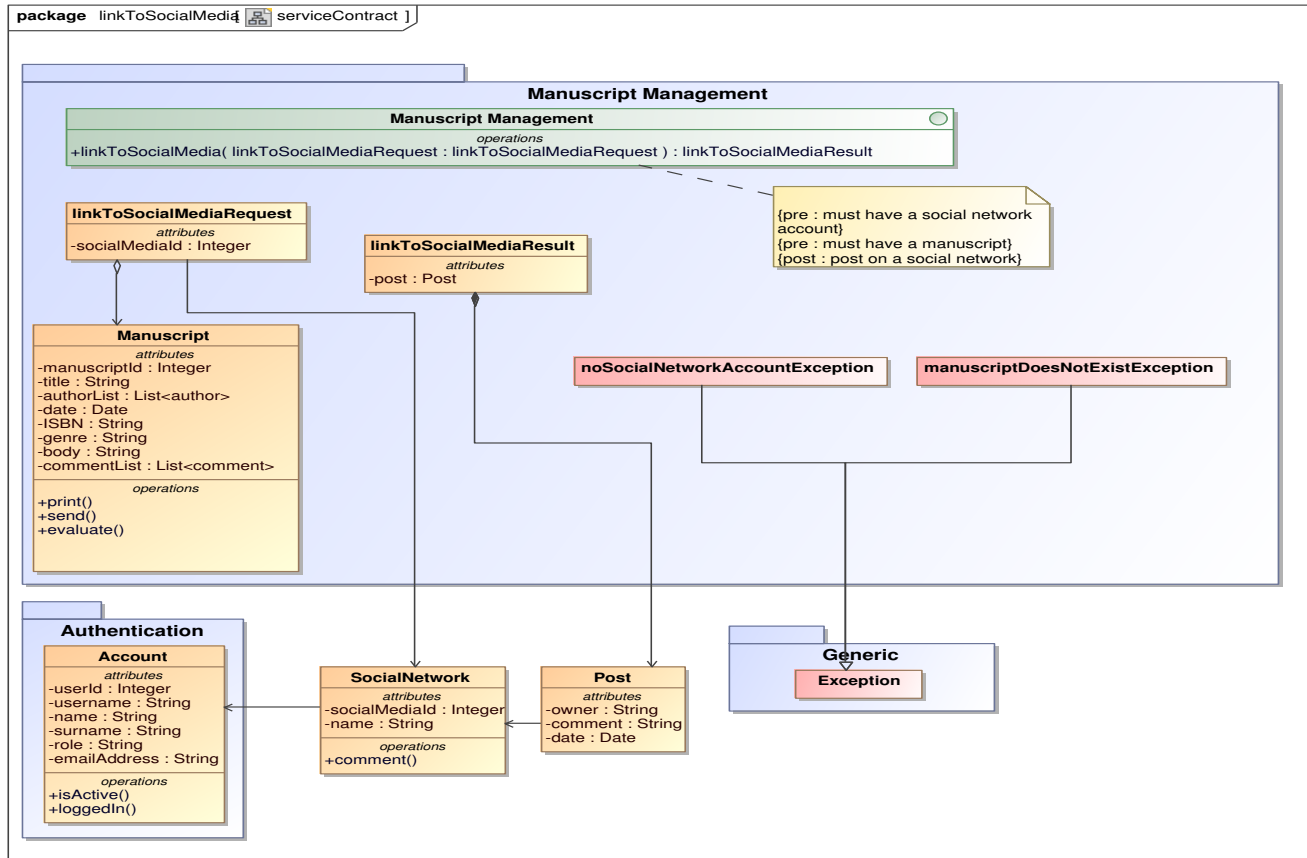


Figure 12: Service contract for linking a project (manuscript) to a social network

4. Edit Manuscript priority: critical

This use case which allows one to edit a manuscript if authorized and afterwards changes are persisted to database.

Service Contract:

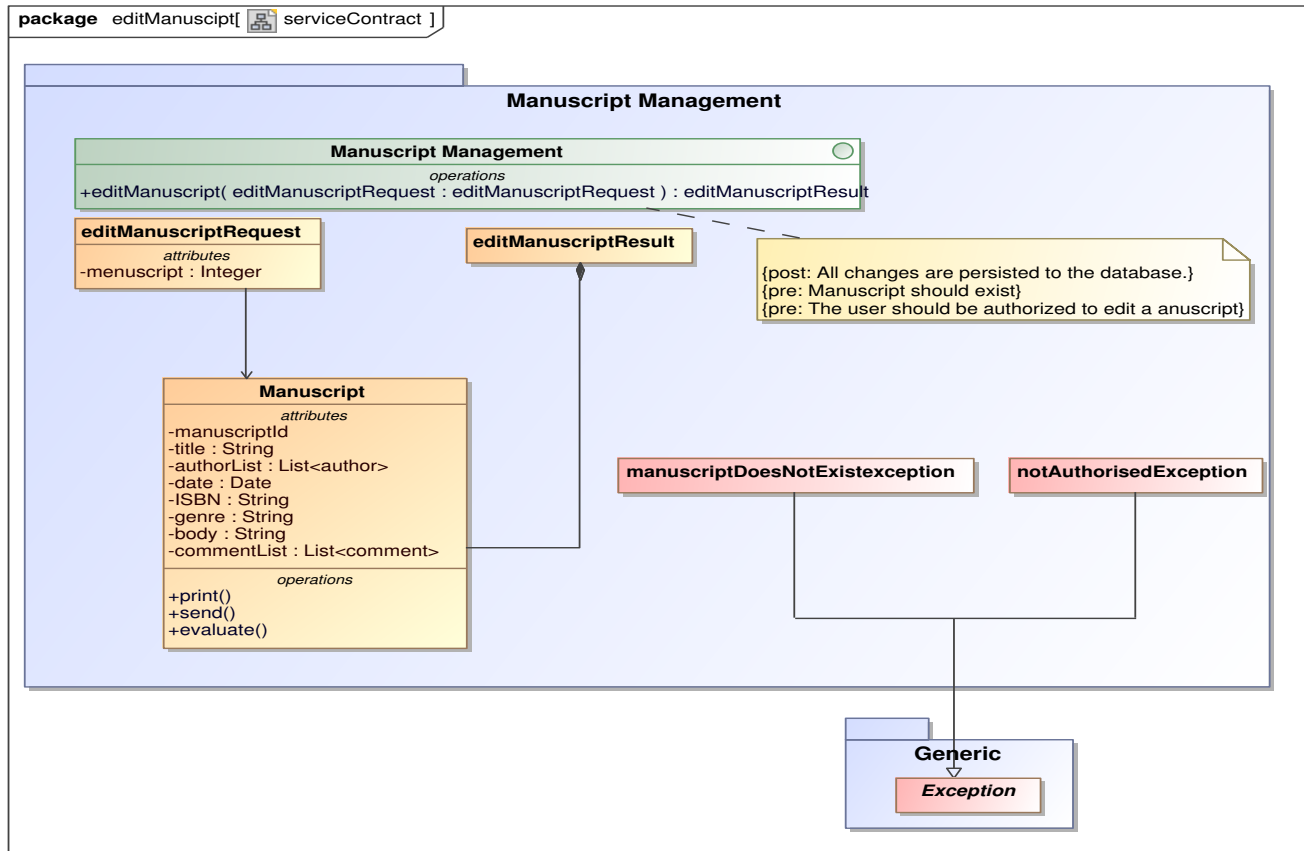


Figure 13: Service contract for editing a project (manuscript)

5. Evaluate Manuscript - priority: critical

This use case allows an agent or an editorial to evaluate the manuscript and to give the according feedback to the author, such as a comment or editorial letter.

Service Contract:

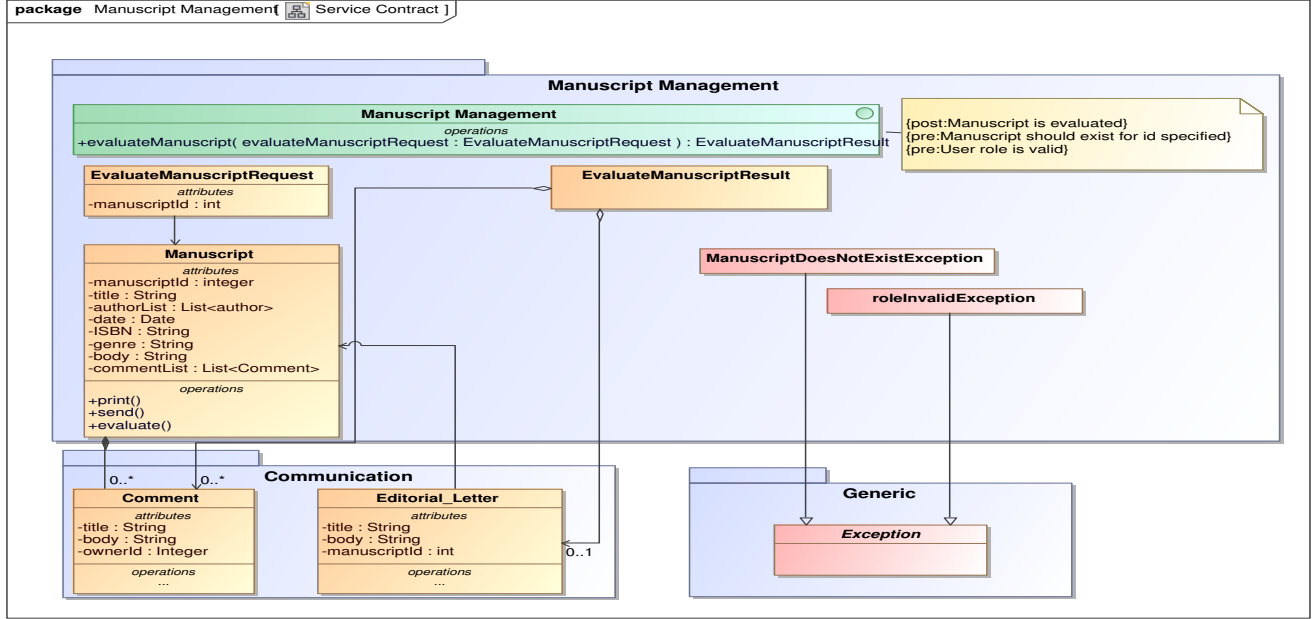


Figure 14: Service contract for evaluating a manuscript

Process Specification (Activity diagram):

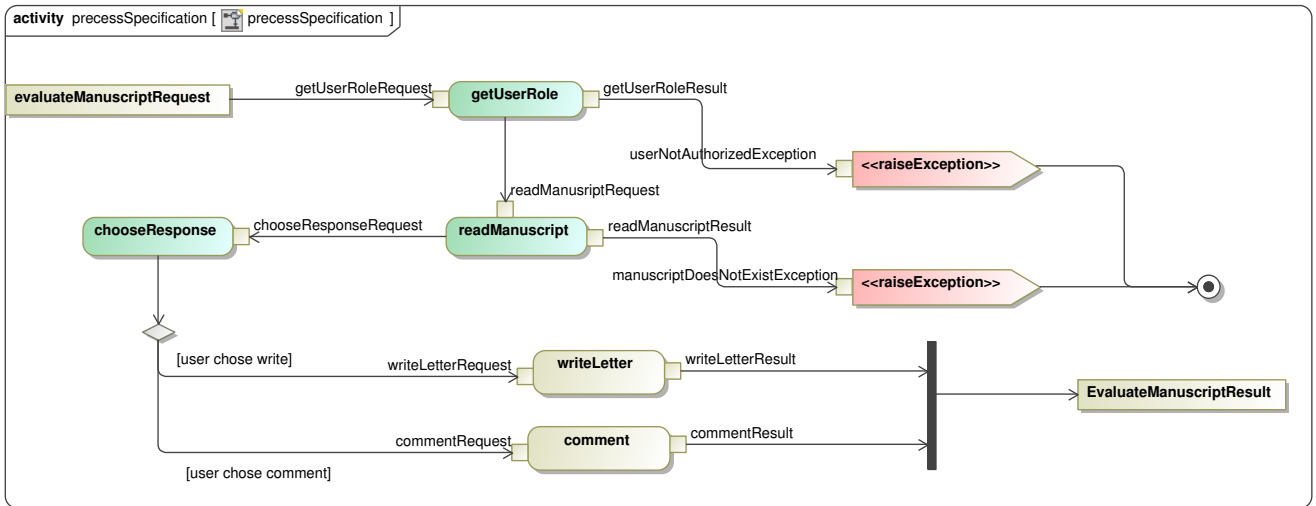


Figure 15: Process Specification for evaluating a manuscript

6. Read Manuscript - priority: critical

The use case grants access to users to view/read a manuscript, under the conditions that the user is authorized and that the manuscript still exists.

Service Contract:

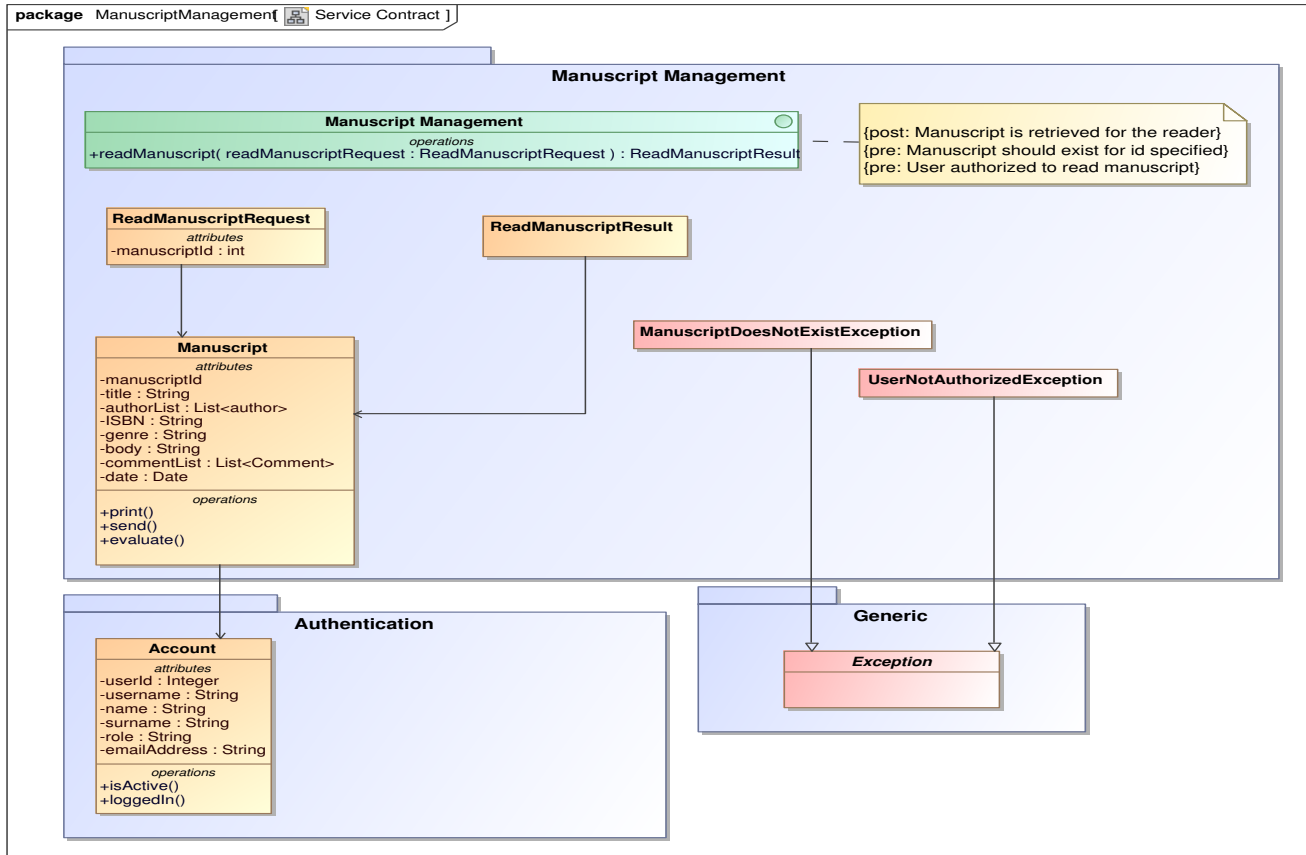


Figure 16: Service contract for reading a manuscript

7. Send Manuscript - priority: critical

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

Service Contract:

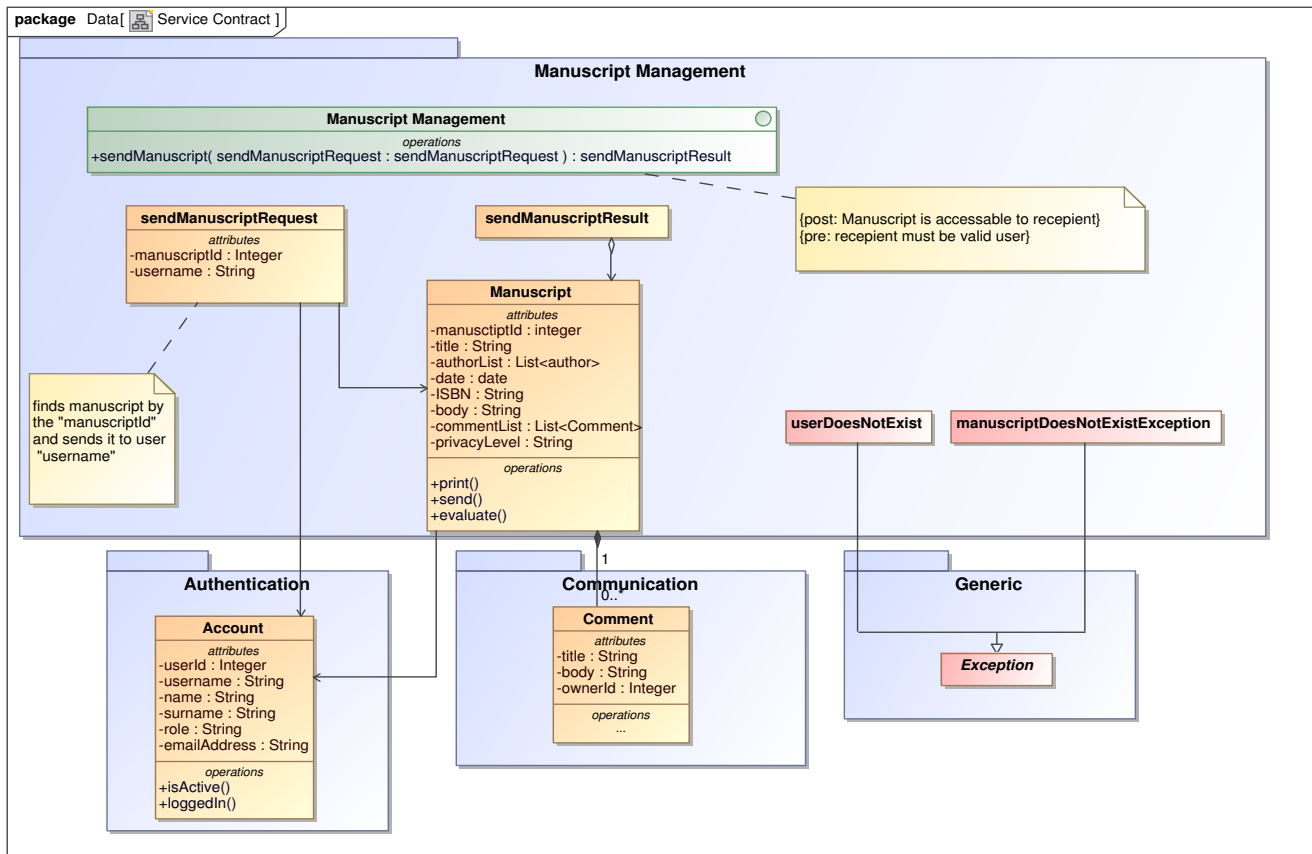


Figure 17: Service contract for sending a manuscript

2.2.3 Domain Model

This module will keep a manuscript in the database as well as be connected to all the other modules needed to perform the necessary tasks such as communication, adding authors, editors, agents etc, the modules connected to this module at the current stage of management are: Communication, Authentication as well as the external classes required for linking the manuscript to social media.

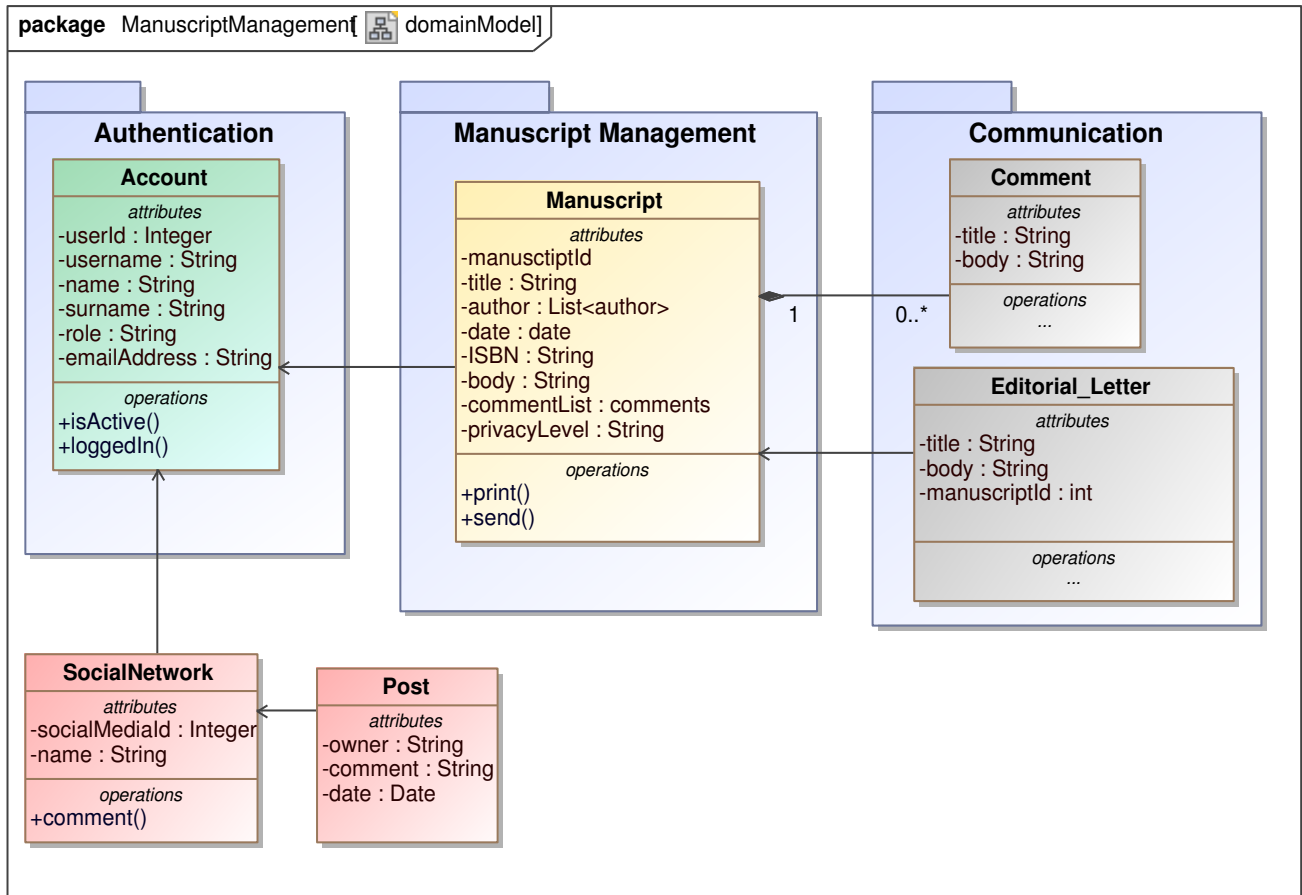


Figure 18: Domain model for managing a manuscript

2.3 Communication

2.3.1 scope

In the life cycle of the book, there is plenty communication that occurs. This module takes care of all communication that may occur between different parties (author to editor, editor to agent agent to author etc)

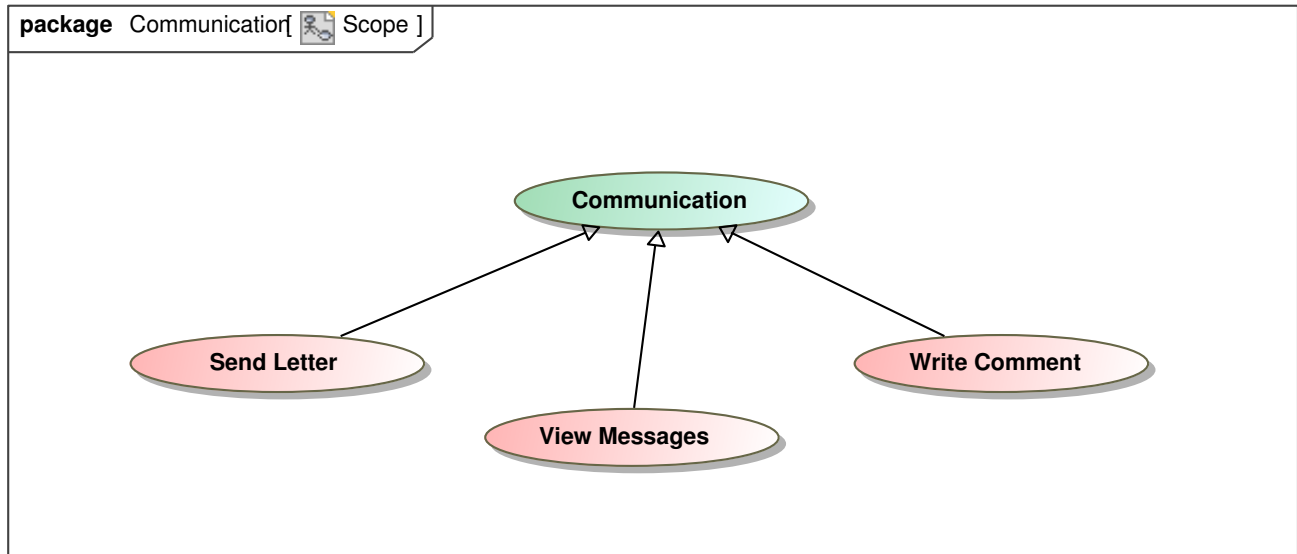


Figure 19: Scope of Communication module

2.3.2 Use Cases

1. Send Letter - priority: critical

This use case allows the writing and sending of an appropriate letter/message to the corresponding user, such as a query letter from author to agent. Available letters are: editorial letter, contract offer and query letter

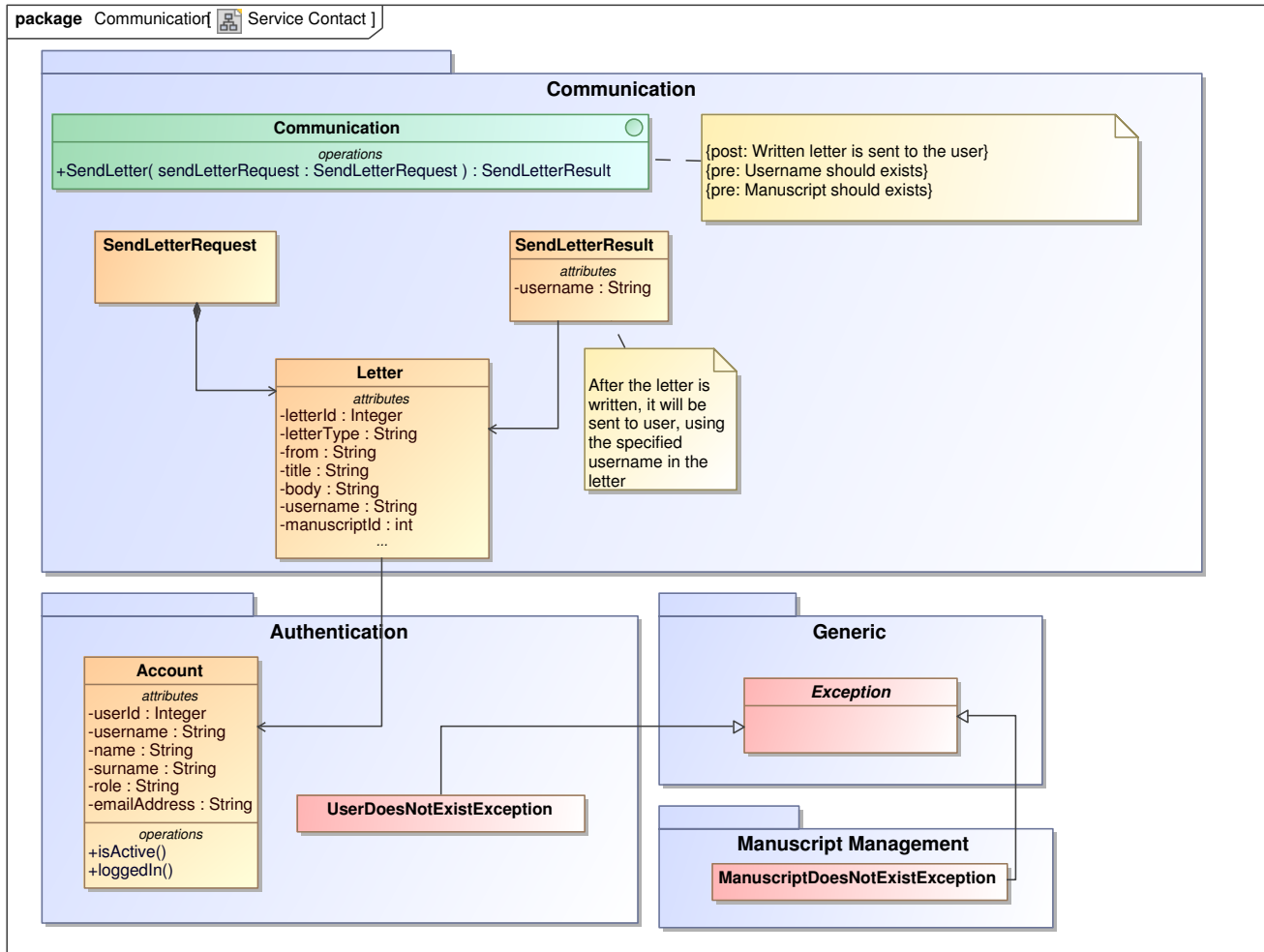


Figure 20: Service contract for sending a letter

2. View Letter - priority: critical

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

service contract:

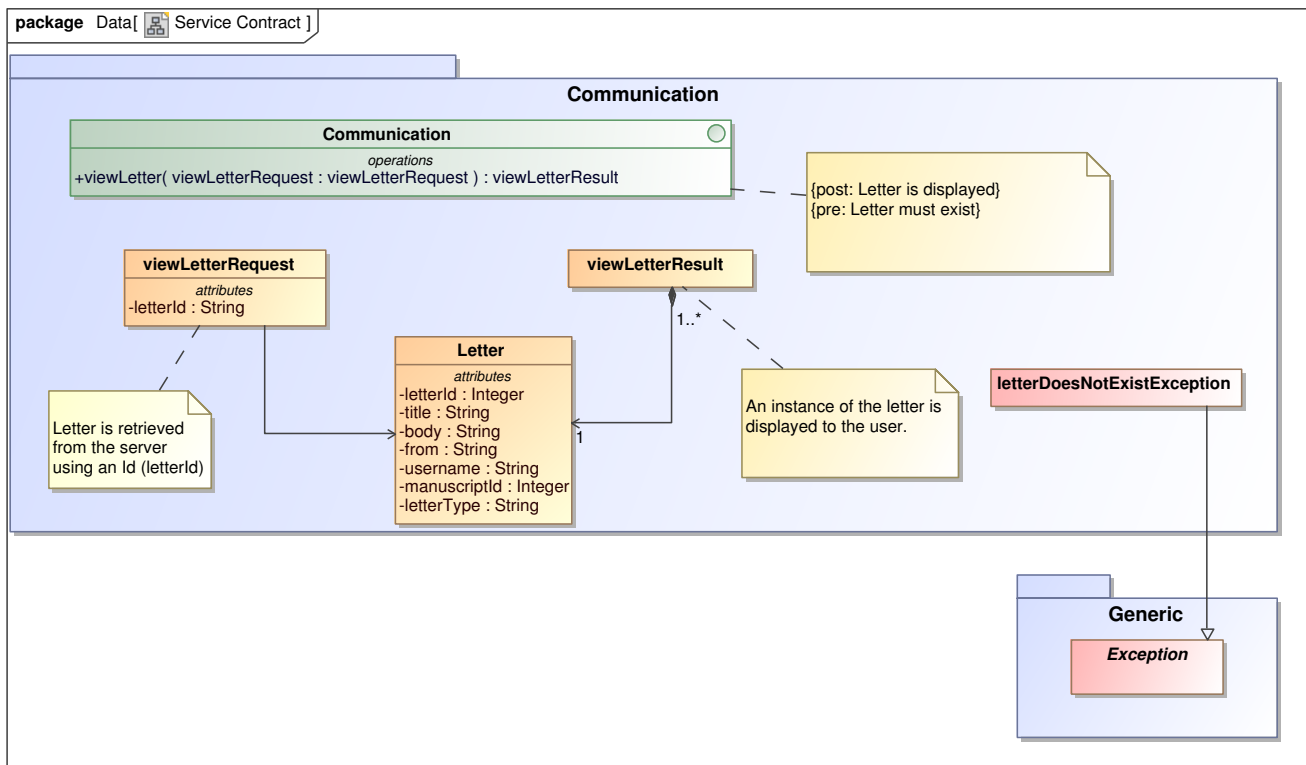


Figure 21: Service contract for viewing a letter

3. Write Comment priority: nice to have

This allows a user (with the editor role) to add comments on the manuscript. Comments can be little notes to the author to indicate what the editor was thinking about that portion of the manuscript.

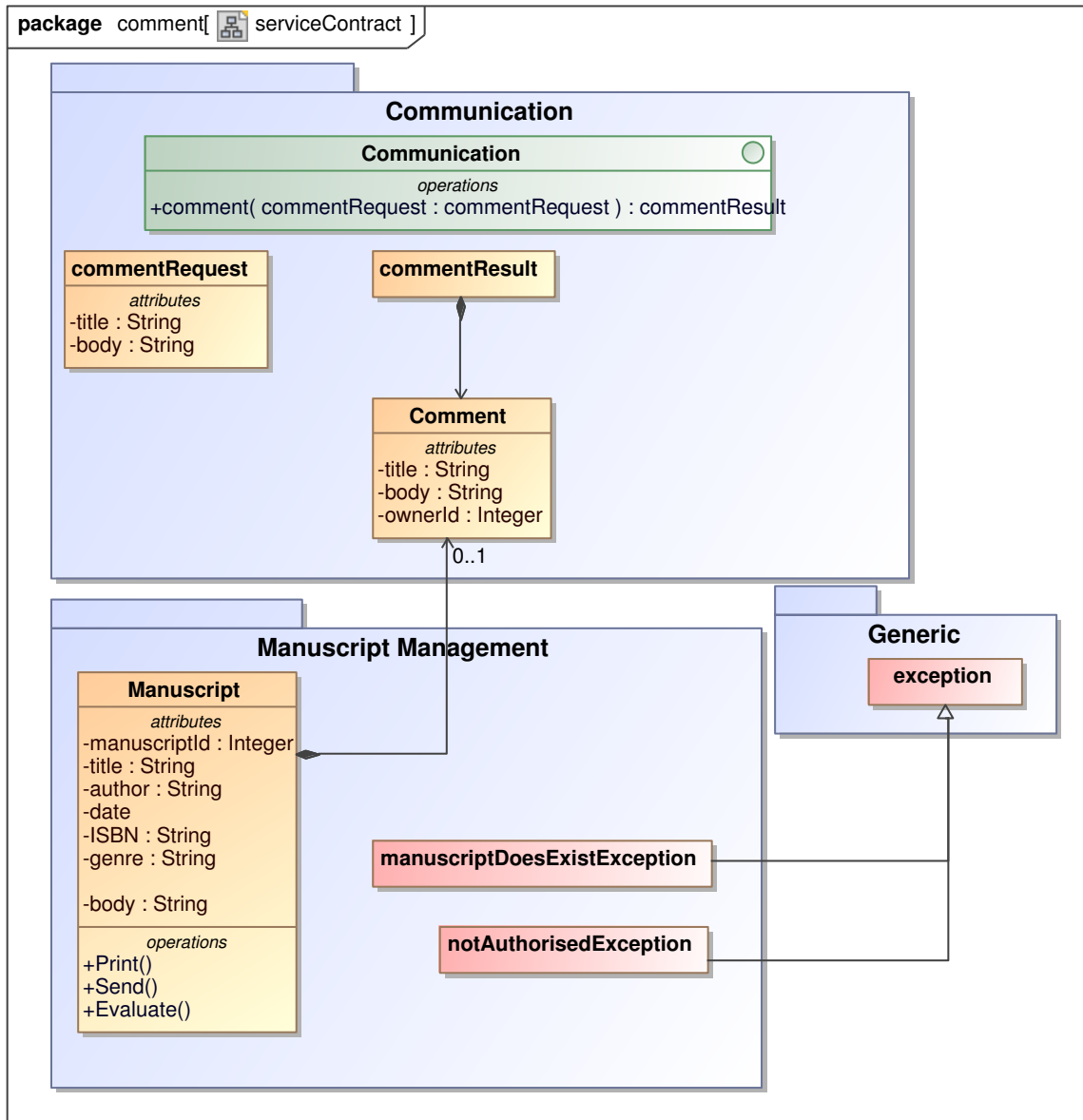


Figure 22: Service contract for writing a comment a manuscript

2.3.3 Domain Model

Every time a user wishes to communicate with another user on the system they will make use of functionality from this module. This module accommodates the sending, receiving and viewing of any messages on the system between users.

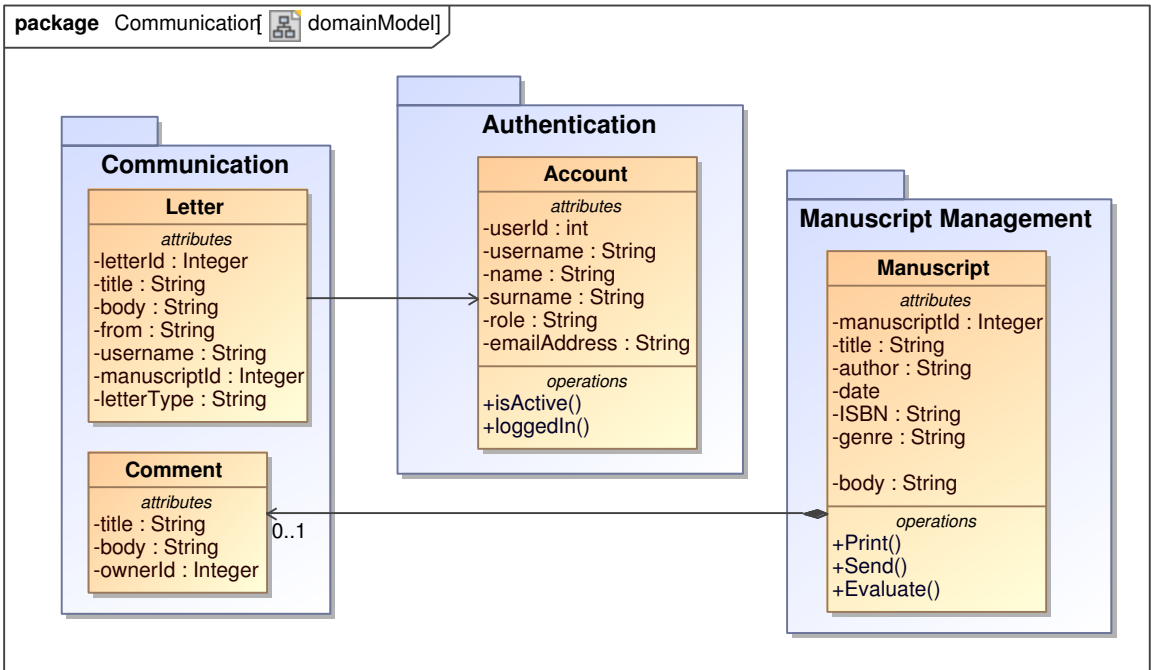


Figure 23: Service contract for writing a comment a manuscript

2.4 Reporting

2.4.1 Scope

The reporting module allows for better control and understanding of what has been done and what still needs to be done as well as the time constraints that apply to all tasks and the overall schedule.

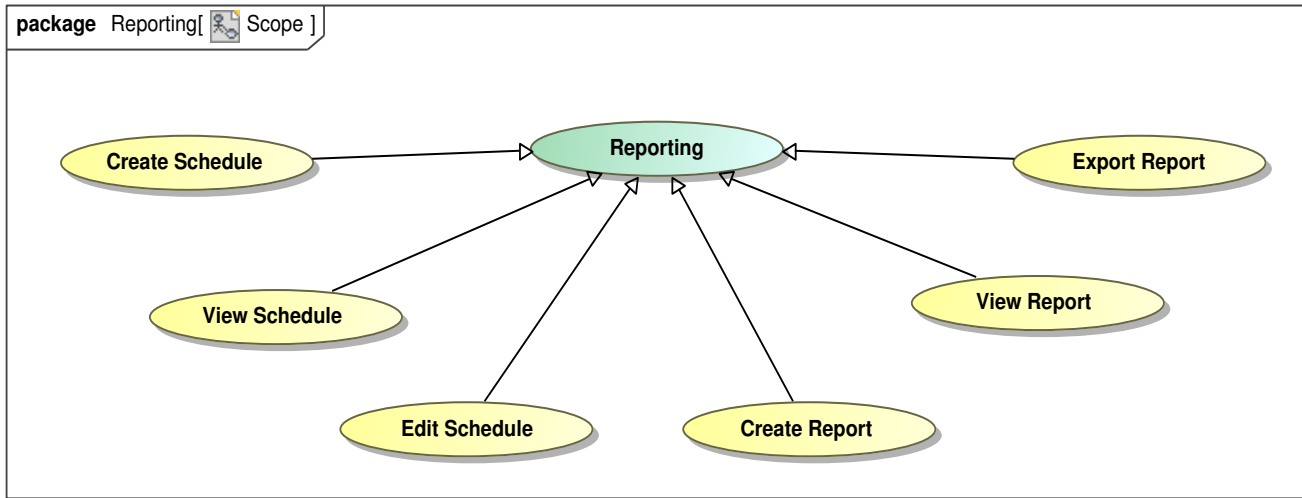


Figure 24: Scope for reporting module

2.4.2 Use Cases

1. Create Schedule - priority: critical

This use case allows a user to create a work schedule for the book that may consist of multiple tasks, so as to facilitate concurrent work(Tasks). **Service Contract:**

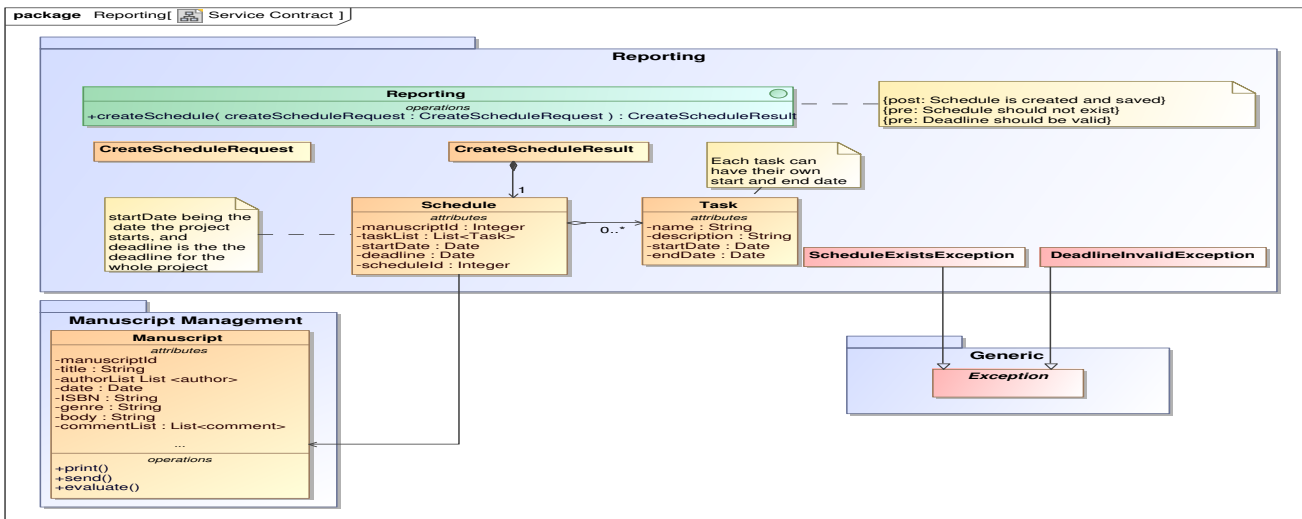


Figure 25: Service contract for creating a work schedule

2. View Schedule - priority: critical

This use case allows a user to view the work schedule for the book.

Service Contract:

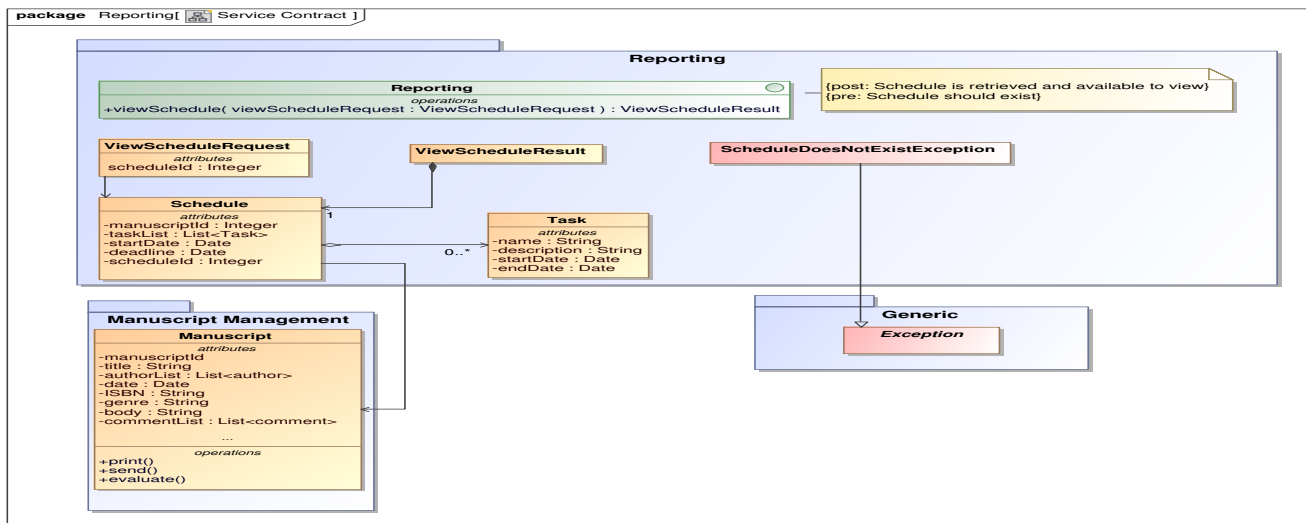


Figure 26: Service contract for viewing a work schedule

3. Edit Schedule - priority: critical

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

Service Contract:

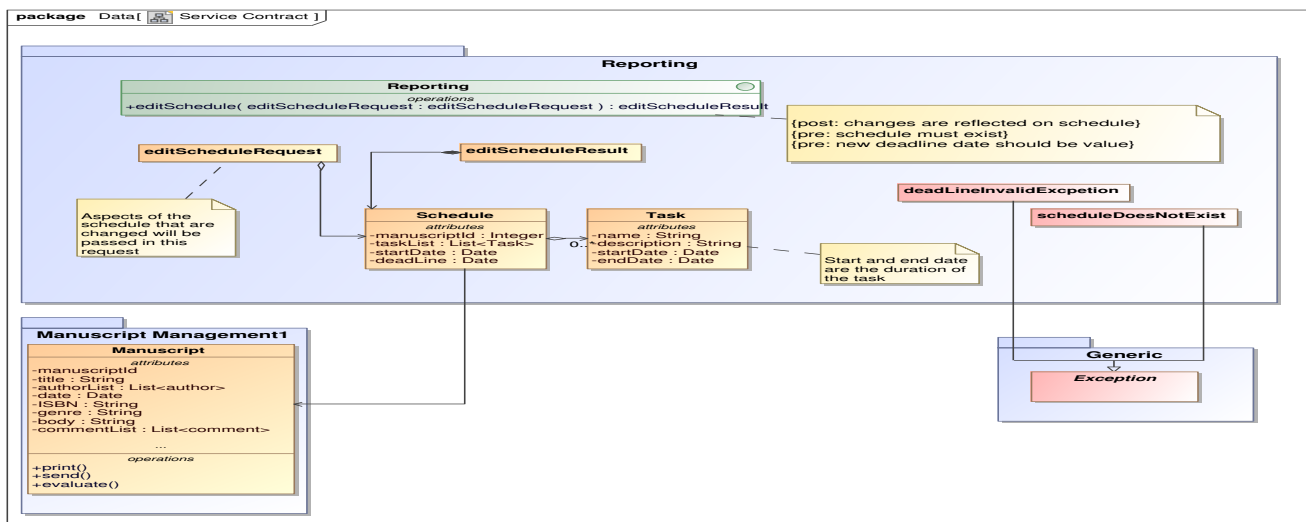


Figure 27: Service contract for editing a work schedule

4. Create Report - priority: important

This use case creates a report based on filters, therefore allowing it to be in printable format.
Service Contract:

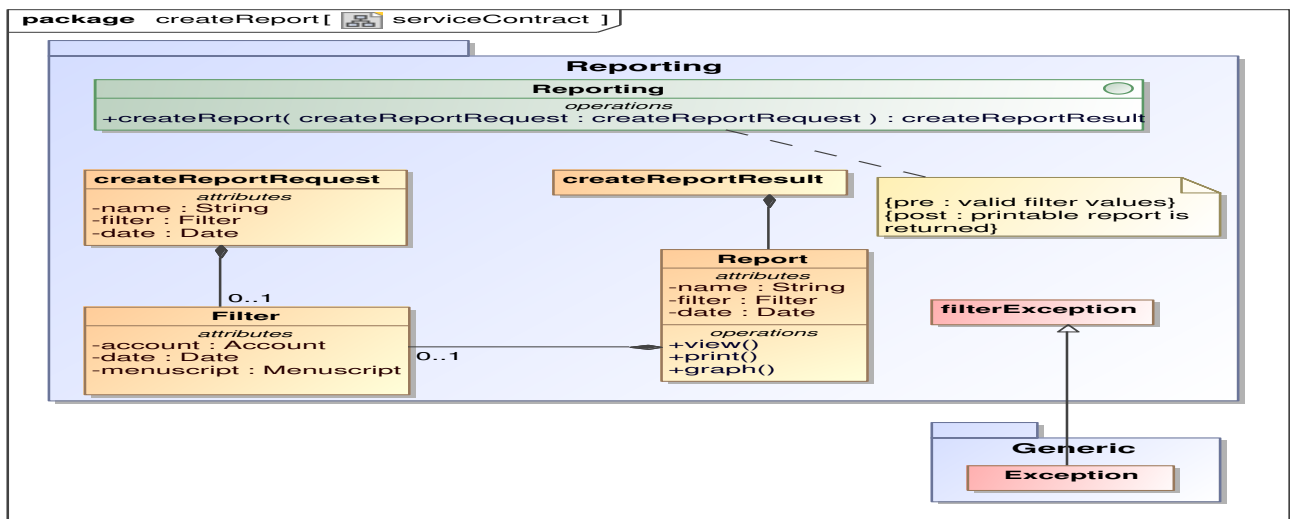


Figure 28: Service contract for creating a report

5. View Report - 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:

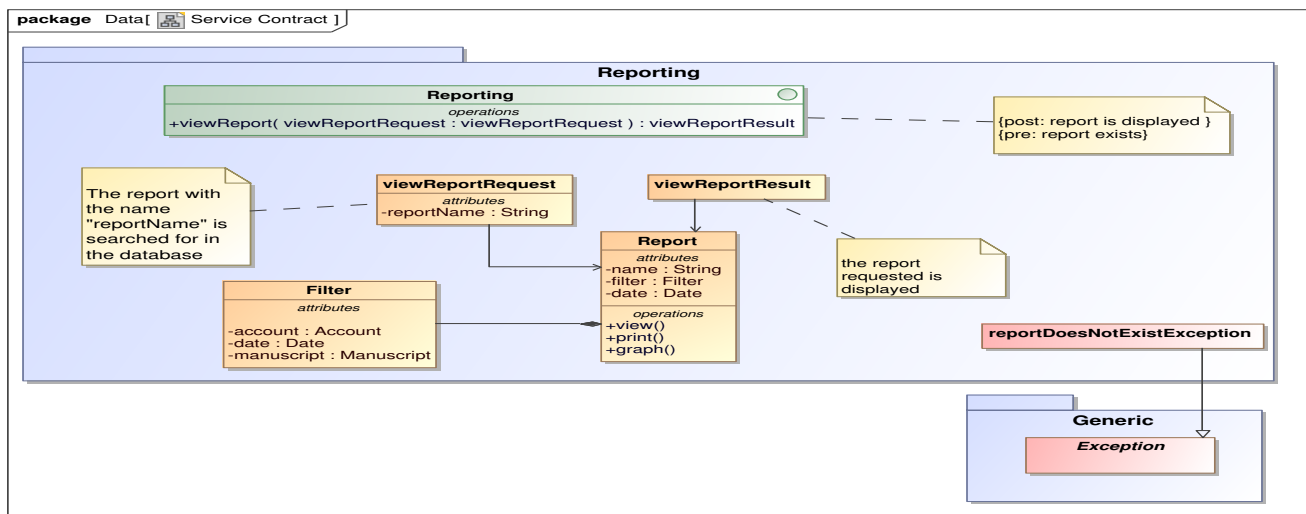


Figure 29: Service contract for viewing a report

6. Export Report - priority: important

This use case which allows one to export a manuscript
Service Contract:

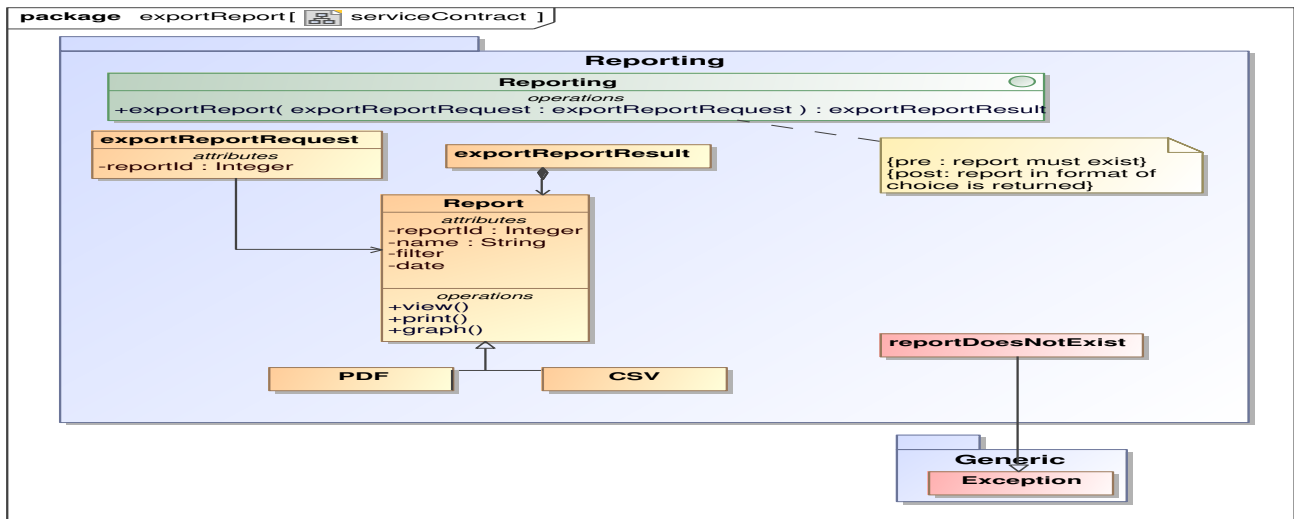


Figure 30: Service contract for exporting a report

2.4.3 Domain Model

The reporting module is responsible for keeping track of all task, schedules and progress. A schedule class is setup to consist of various tasks, starting with none and they can be added over time. The overall schedule has a starting and a deadline date, while the inner tasks also consist of their own unique start and end dates. A report can be generated by using a user specified filter that can consist of multiple fields.

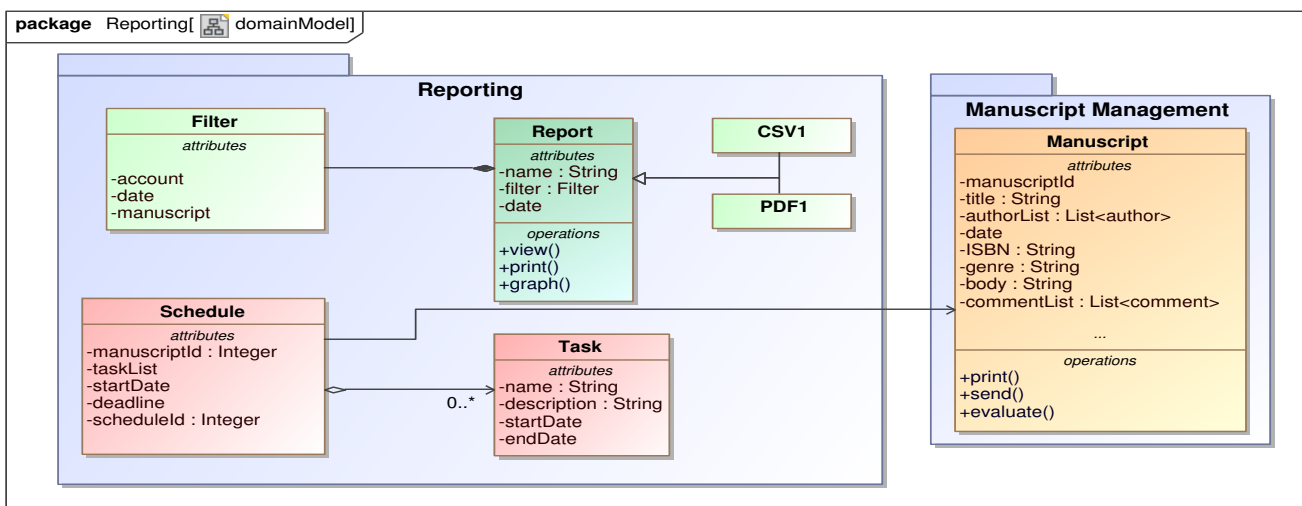


Figure 31: The domain model for the reporting module.

2.5 Production

2.5.1 scope

This section provides the detailed use case requirements for the use cases offered by the Production module.

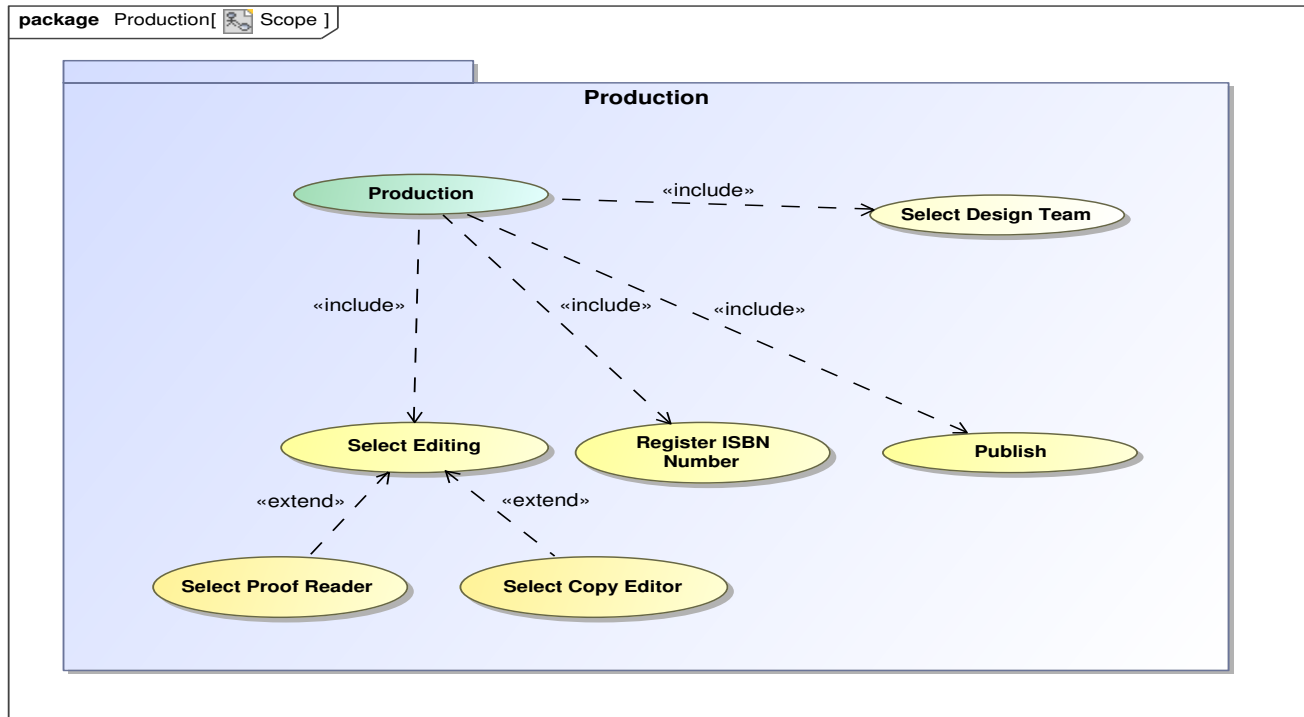


Figure 32: Scope for Production module

2.5.2 Use Cases

1. Register ISBN Number - priority: important

This use case will allow the user to fill in an application form to register the ISBN number of the manuscript. Most of the inputs on this application form will be auto-completed and an email requesting the ISBN Number will be sent.

Service Contract: Below is a figure of the registerISBNNumber service contract.

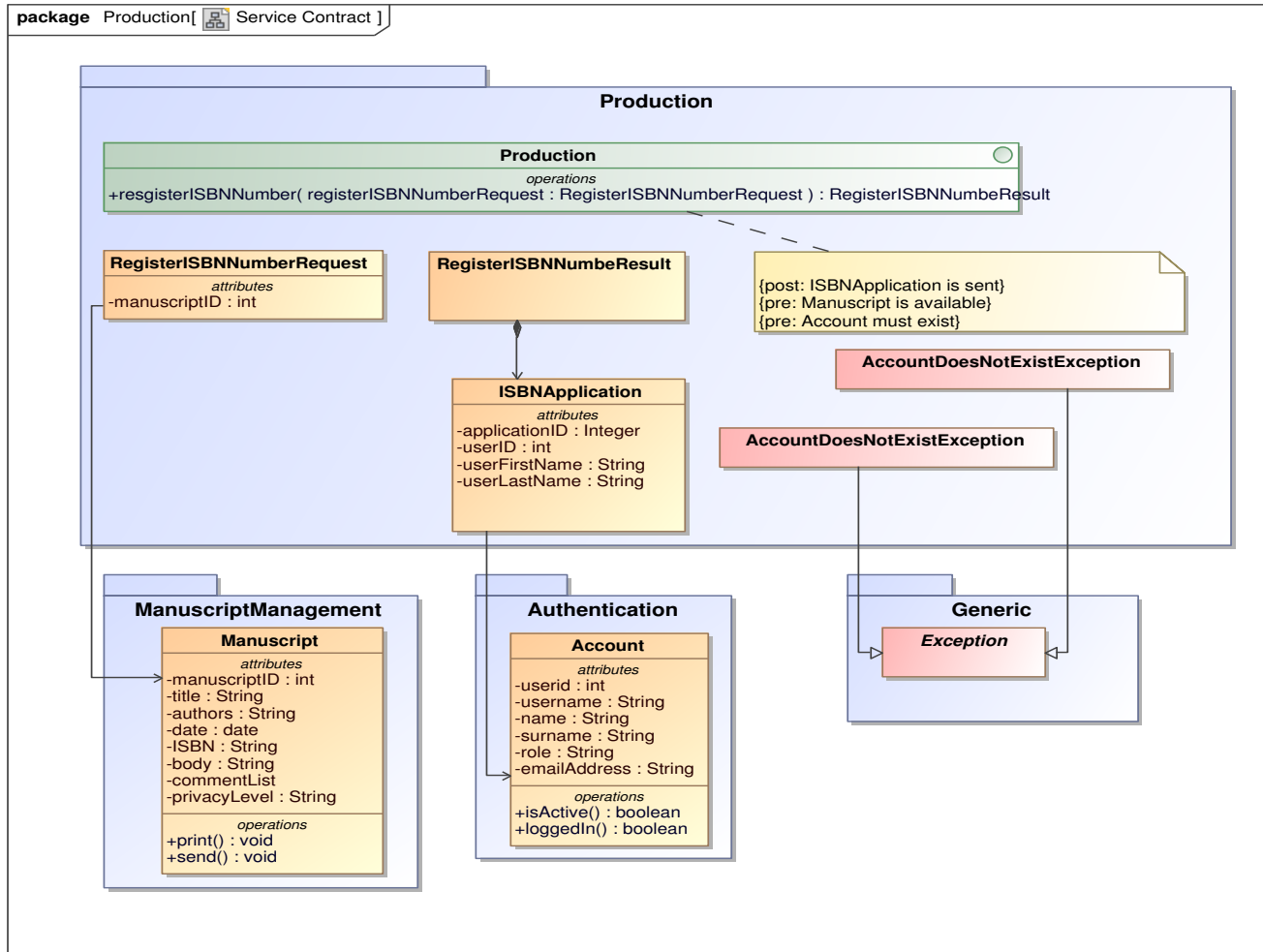


Figure 33: Service contract for registering a manuscript's ISBN number.

2. Select Copy Editor - priority: critical

This use case allows a user to select a copy-editor from the available copy copy-editors registered in the system.

Service Contract: The service contract for selecting a copy editor service is shown in the figure below. The pre-conditions are enforced (raising the appropriate exception should they not be met) and the required copy editor is then notified of the user's request.

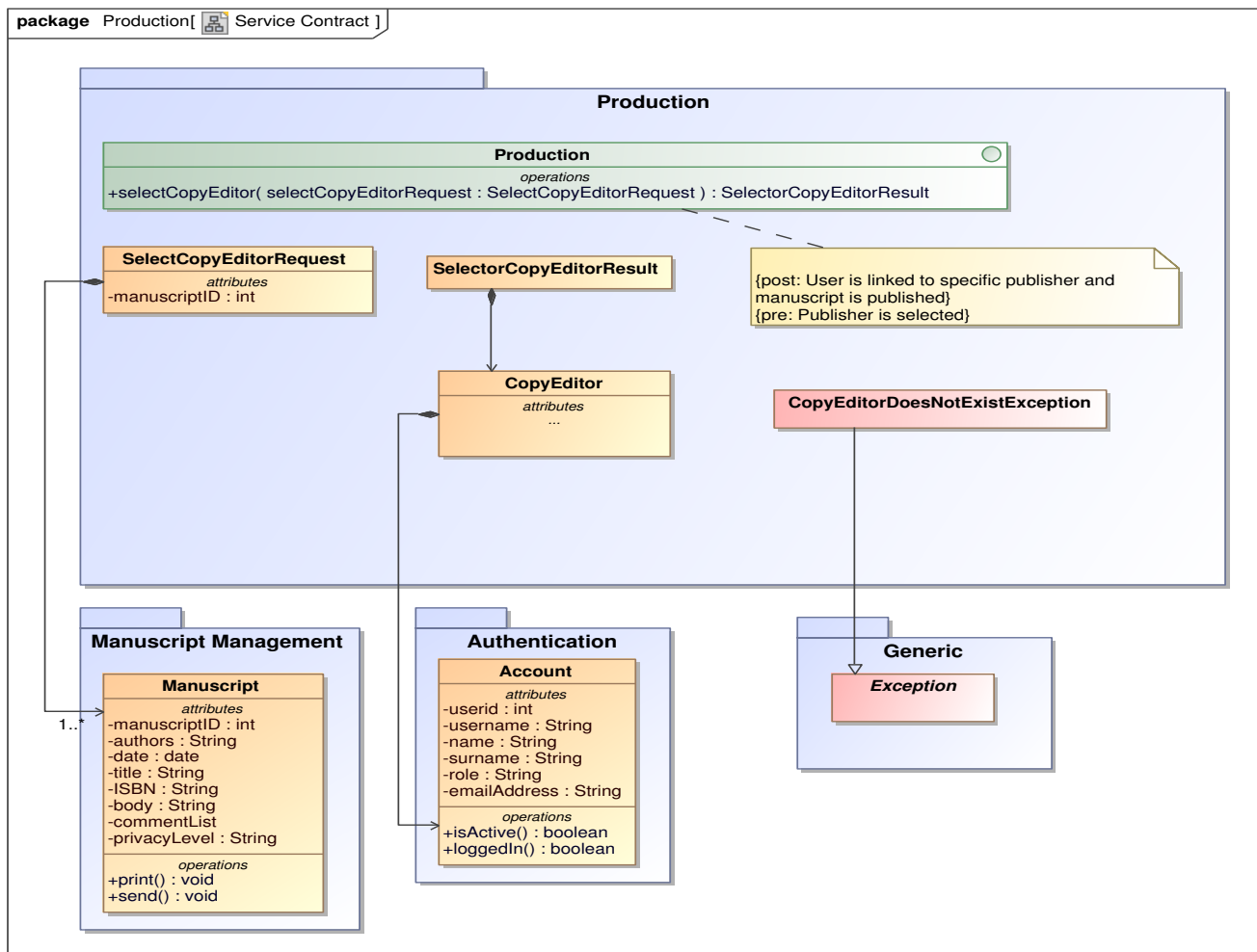


Figure 34: Service contract for selecting a copy editor

3. Select Proof Reader priority: important

This use case allows a user to select a proof reader from the available proof readers registered in the system.

Service Contract: The service contract for selecting a proof reader service is shown in the figure below. The pre-conditions are enforced (raising the appropriate exception should they not be met) and the required proof reader is then notified of the user's request.

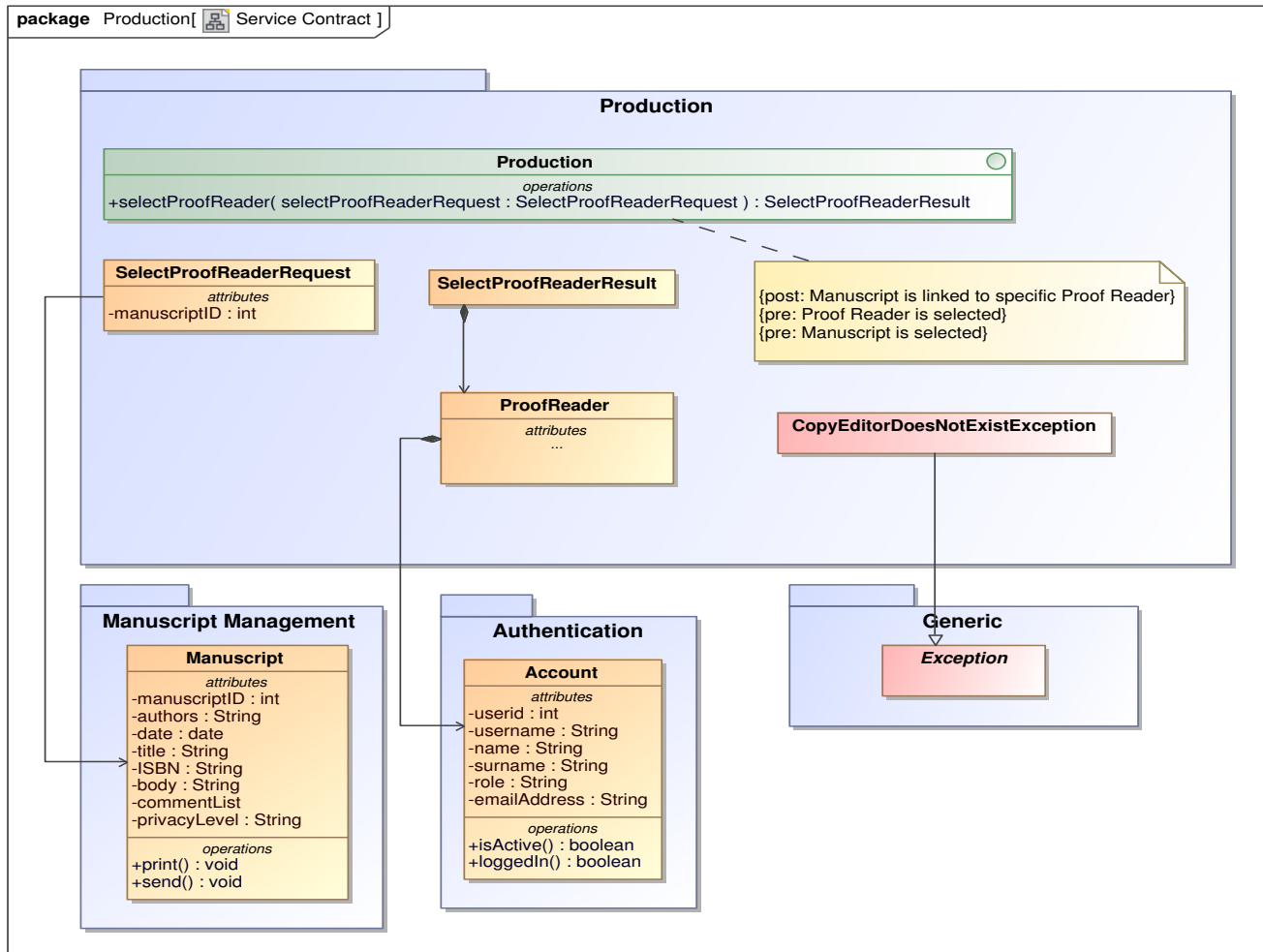


Figure 35: Service contract for selecting a proof reader

4. Select Design Team priority: important

This use case allows a user to select, from the registered designers, a design team that will be responsible for the book cover and styling.

Service Contract: The service contract for the selectDesignTeam service is shown in figure below. The pre-conditions are enforced (raising the appropriate exception should they not be met) and the required design team is notified.

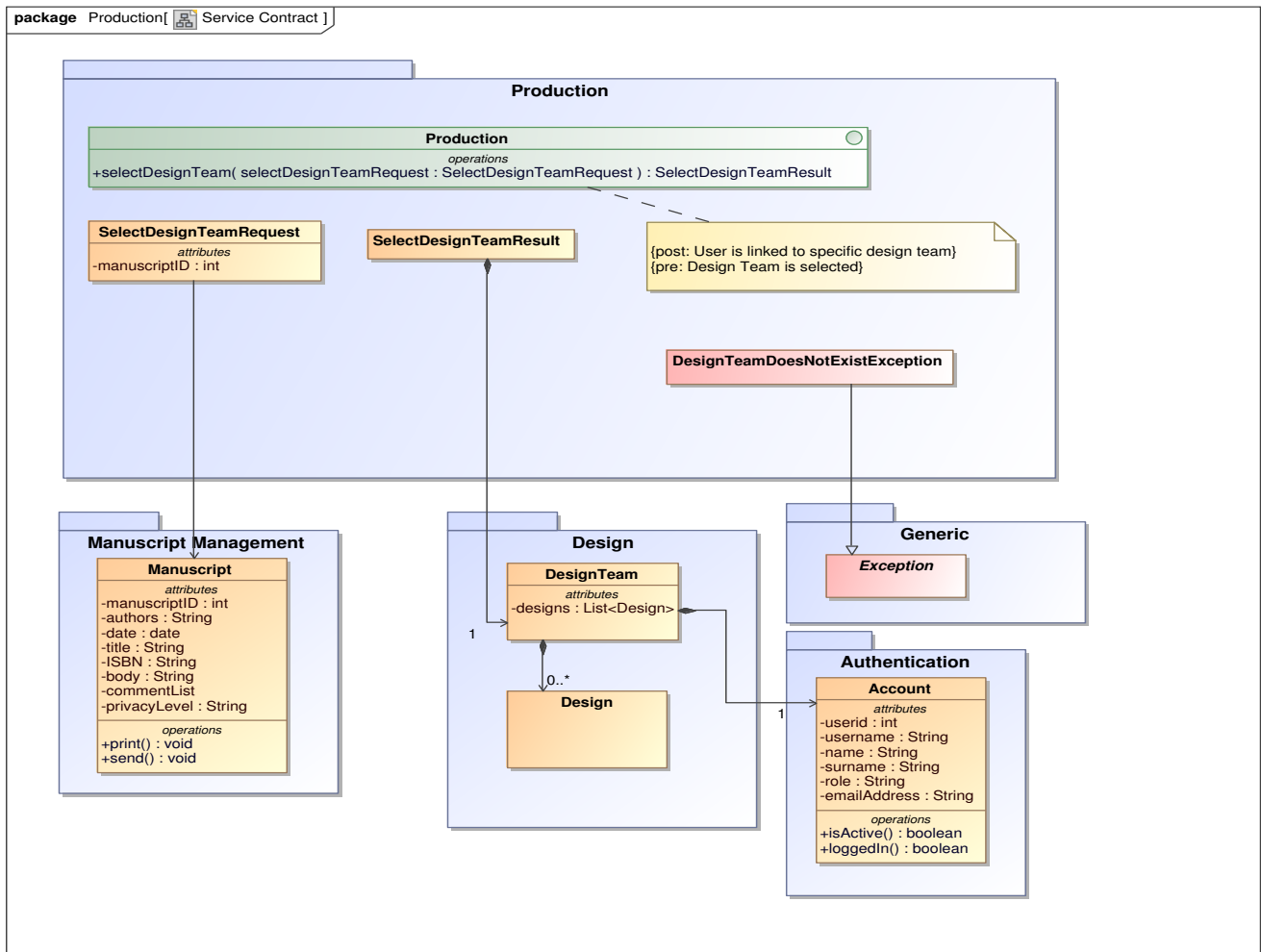


Figure 36: Service contract for selecting a design team

5. Publish - priority: critical

This use case allows an agent or an editorial to select a publisher and request that the final manuscript be published.

Service Contract: The service contract for publish is shown in the diagram below. The pre-conditions are enforced (raising the appropriate exception should they not be met) and a preferred publisher is notified of the publishing request.

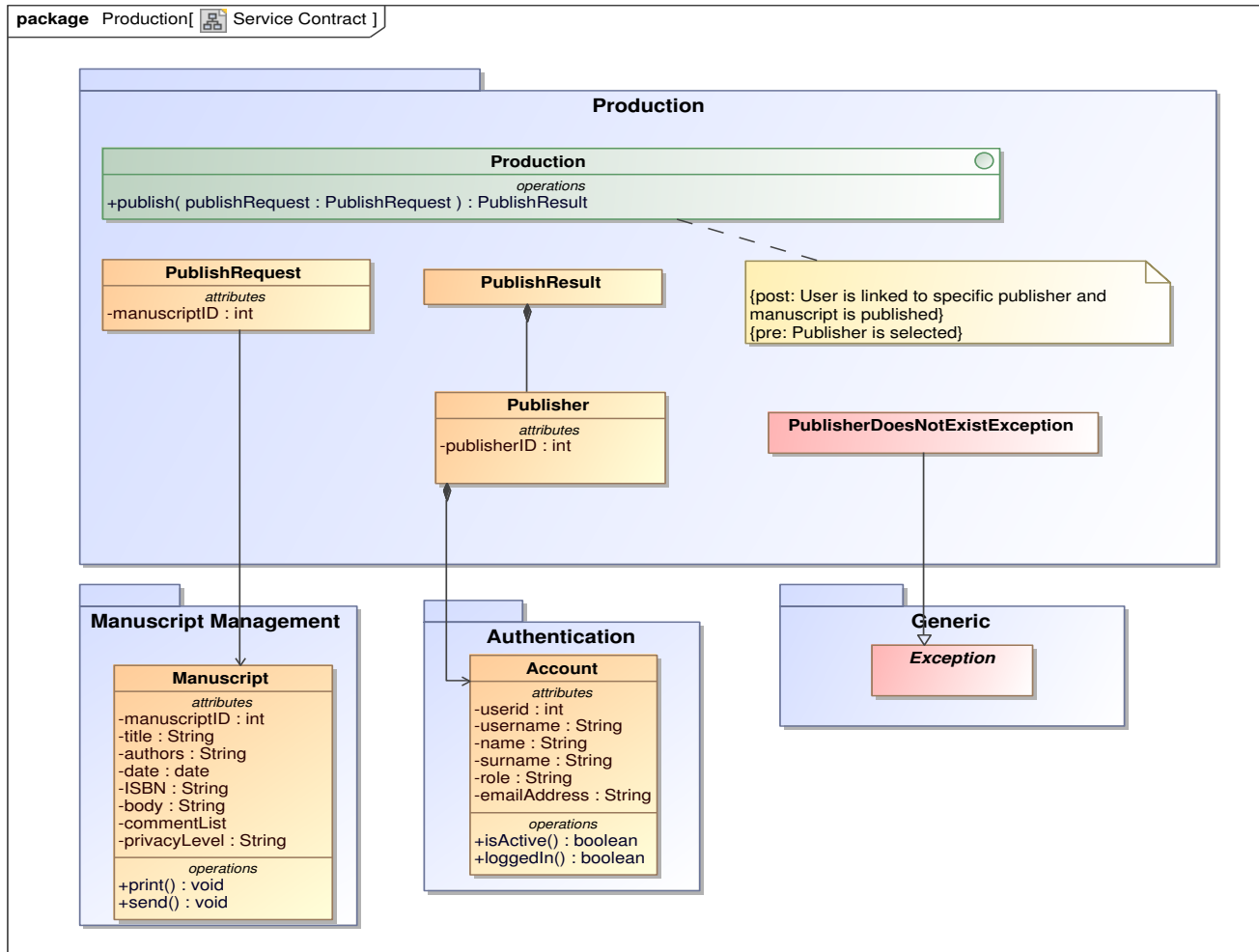


Figure 37: Service contract for evaluating a publish

2.5.3 Domain Model:

Below is a figure for the domain model of manuscript management.

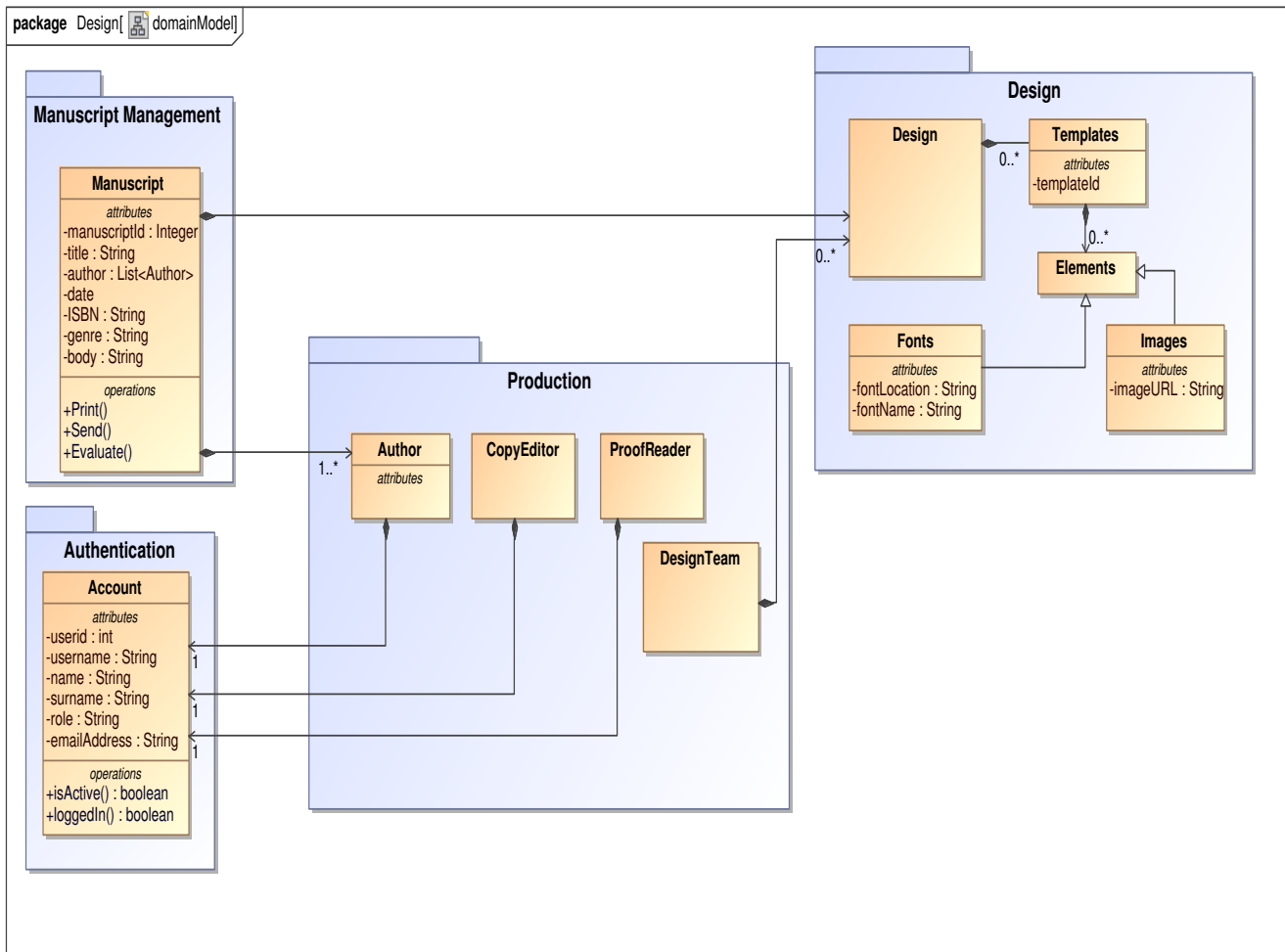


Figure 38: Domain model for managing a manuscript

2.6 Design

2.6.1 scope

The design module allows the designer/design team to structure the layout of the book. This is done by creating or choosing a template, as well as uploading all desired graphics and designing the books cover pages.

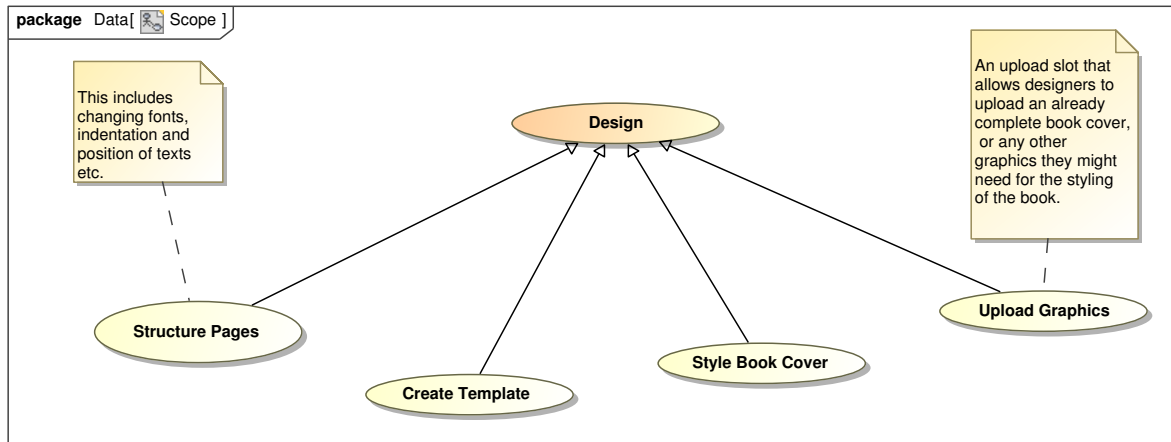


Figure 39: Scope of Design module

2.6.2 Use Cases

1. Structure Pages - priority: critical

This use case which allows one to structure the manuscript.

Service Contract:

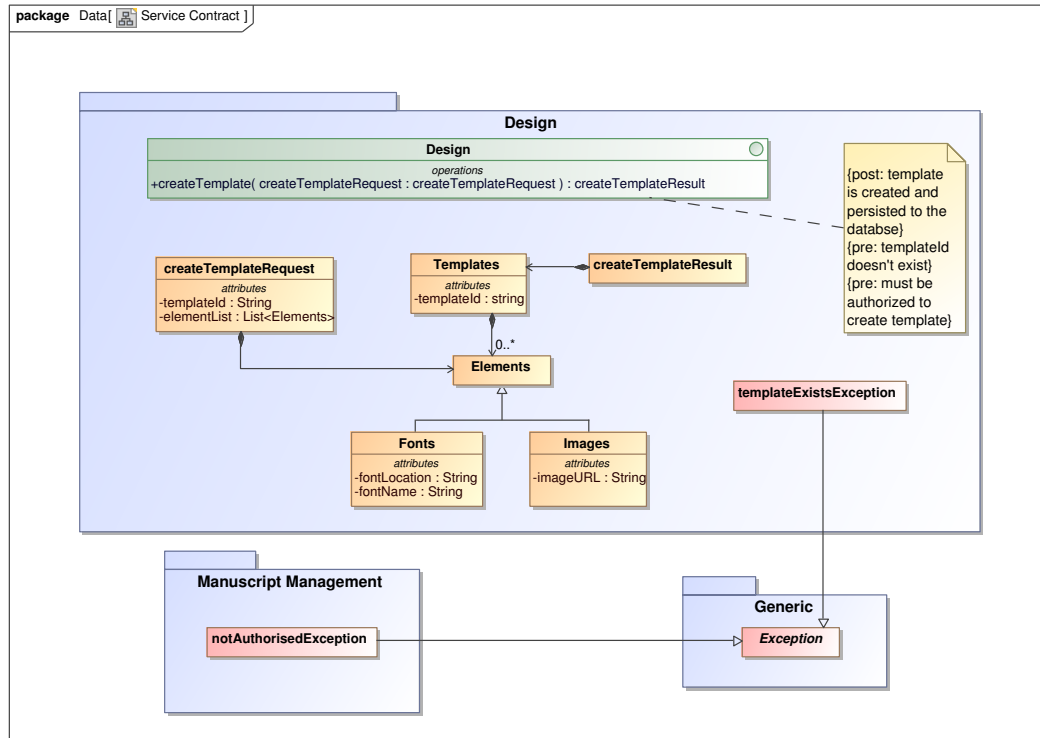


Figure 40: Service contract for structuring a manuscript

2. Create Template - priority: nice to have

This module allows a user to create a template design for the book.

Service Contract:

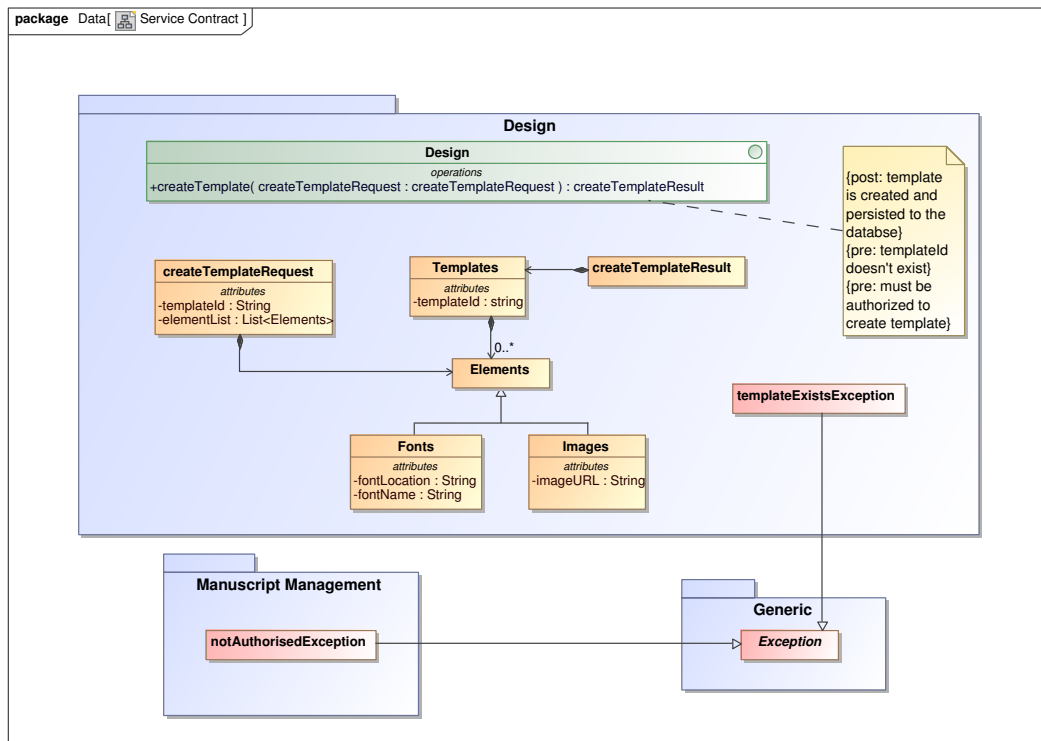


Figure 41: Service contract for creating a template design

3. Create Book Cover - priority: critical

This use case which allows create a book cover for a manuscript.

Service Contract:

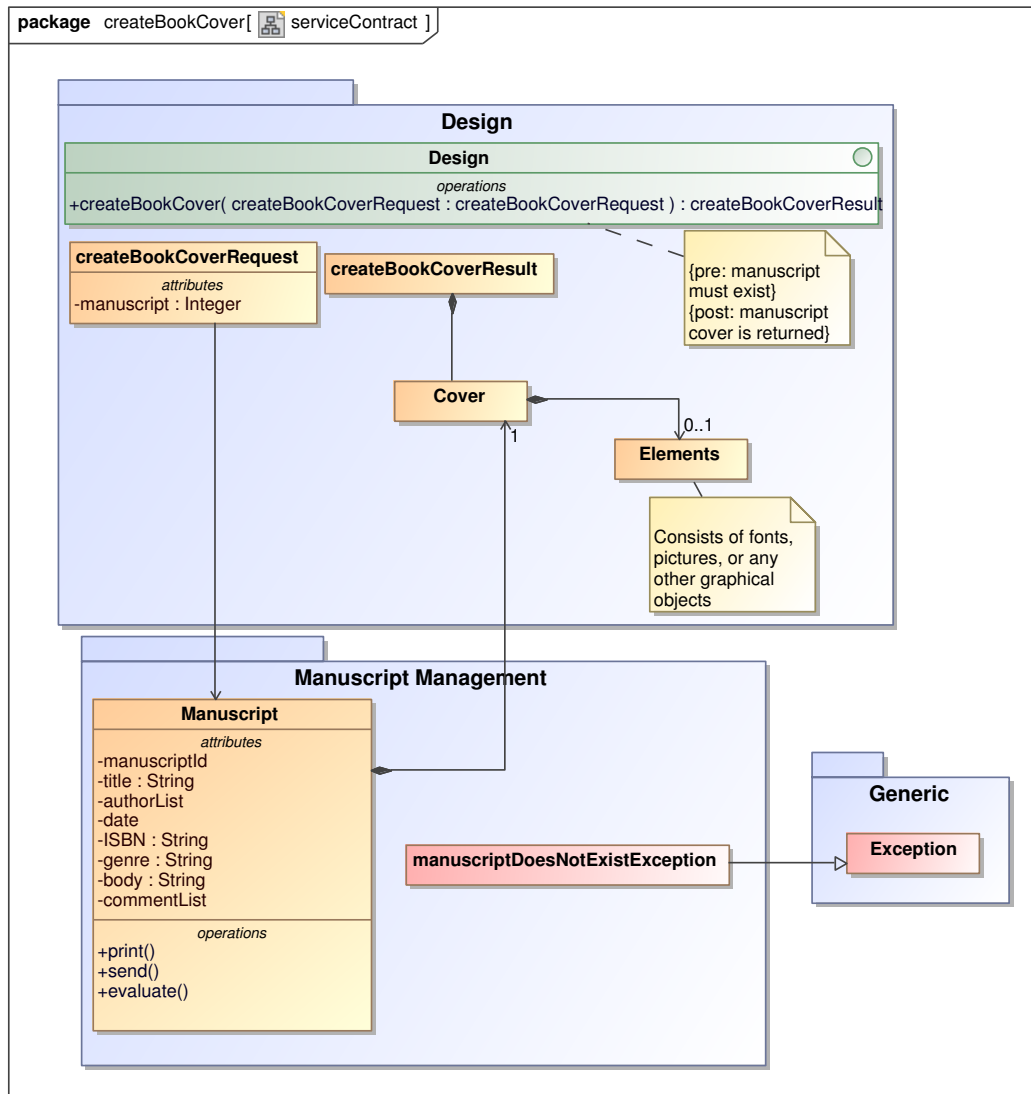


Figure 42: Service contract for creating a book cover

4. Upload Graphic - priority: important

This use case allows a user to open any graphic they choose. This may be a new font, or image they wish to use, or a completed cover for the book. Size and format of a graphic are restricted however.

Service Contract:

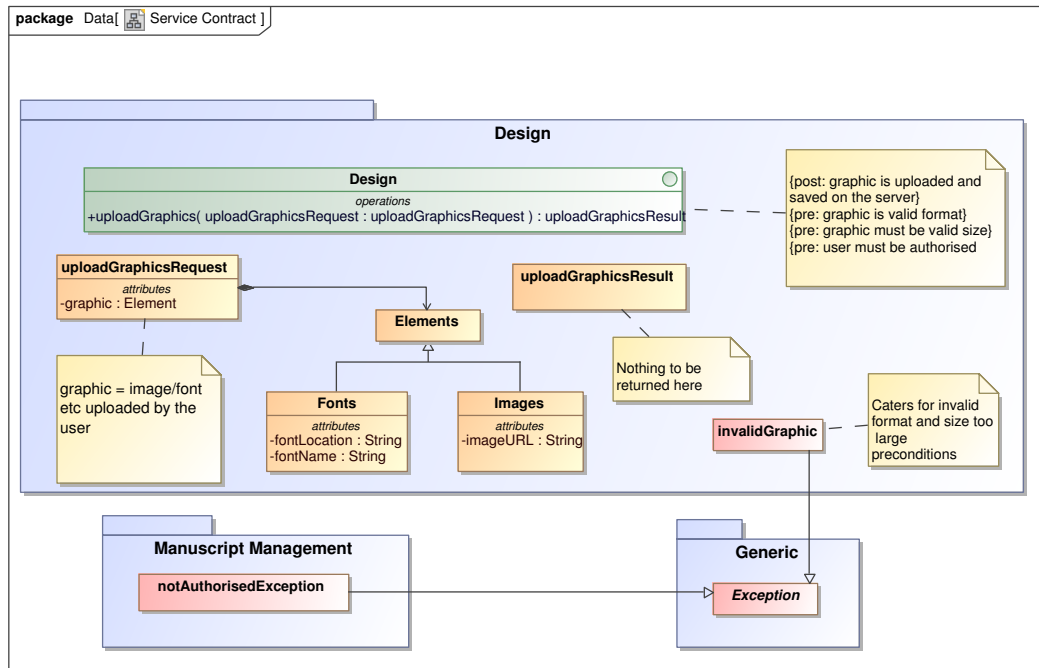


Figure 43: Service contract for uploading a graphic