



Kalafong Provincial Tertiary Hospital

Gynaecological Patient Information management System:

Functional Requirements

Team Pentec:

Ruth Ojo 12042804
Liz Joseph 10075268
Trevor Austin 11310856
Maria Qumayo 29461775
Lindelo Mapumulo 12002862

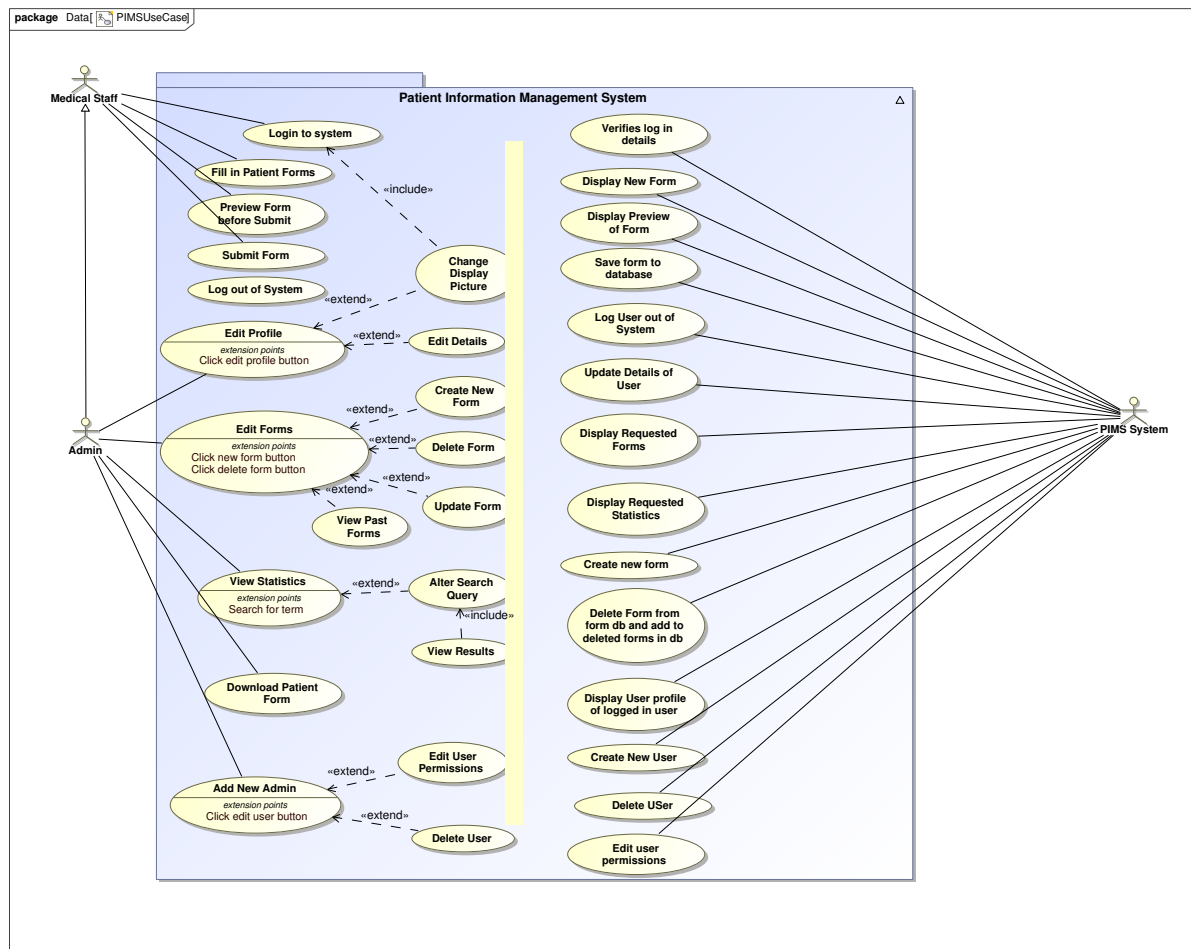


Final Version

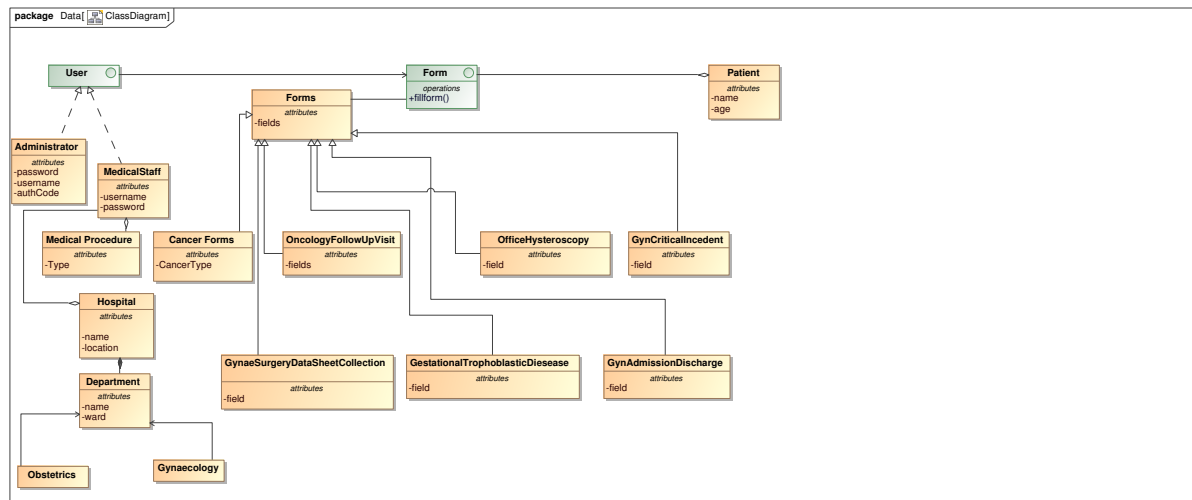
July 17, 2015

Contents

1 Use Case Diagram



2 Domain Model



3 Introduction

This document contains detailed specifications of the functional requirements for the Gynaecological Patient Information management System.

It describes and defines major use cases identified as given by the client's original project specification. As well as a clear overview of the system represented as a domain model

The following topics are addressed in detail with supporting diagrams for each use case:

- Use case prioritization.
- Pre and Post Conditions
- Service contracts.

4 User Handling

4.1 User login

Description: This use-case enables a user to log into the system so as to insert medical data.

4.1.1 Prioritization:

Critical

4.1.2 Conditions and Data Structures:

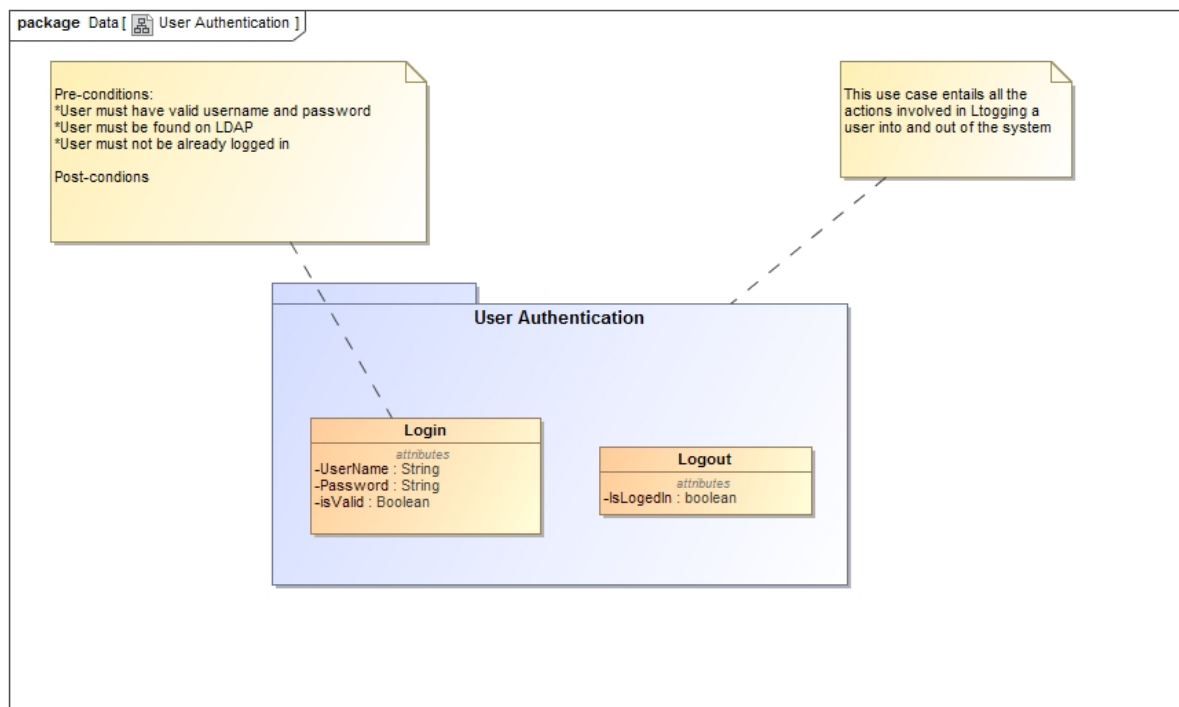
Pre-Conditions:

- , The user must be found on LDAP, thus have a valid username and password.
- The user must not already be logged in.

Post-Conditions:

- The user is logged in.
- The user is authenticated.
- The user can carry out system activities.

4.1.3 Process Specifications:



4.2 Logout

Description: User Is logged out of system and is not allowed to access system until logged in again

4.2.1 Prioritization:

Crucial

4.2.2 Conditions and Data Structures:

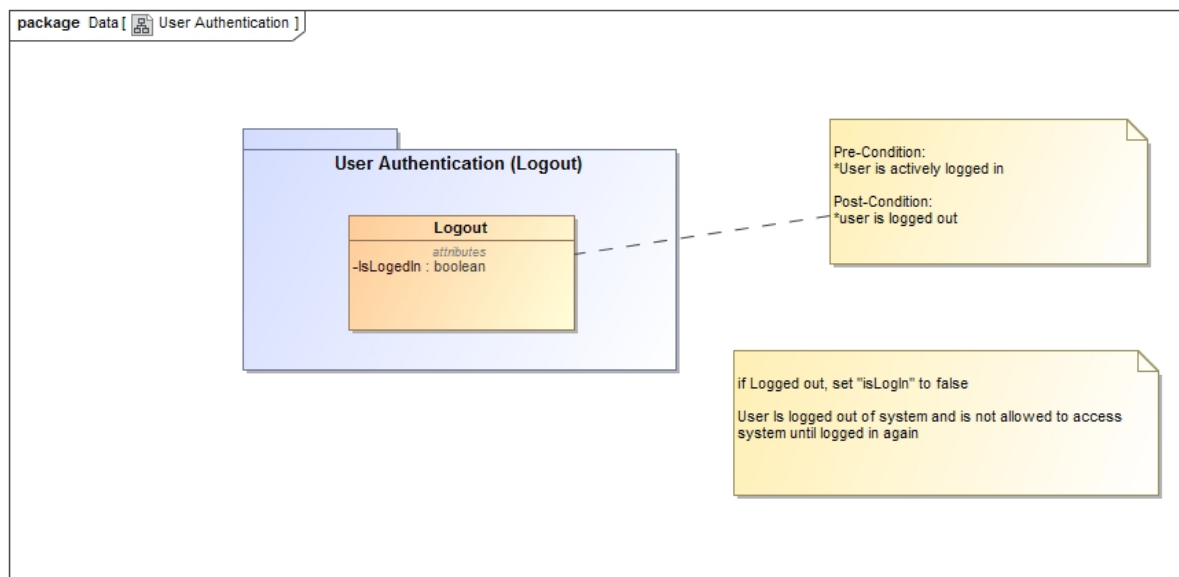
Pre-Conditions:

- The user must be actively logged in

Post-Conditions:

- The user is logged out of the system

4.2.3 Process Specifications:



5 Form

5.1 createNewForm

Description: This use-case allows for the addition of a new form on to the system by first building the form with a form builder tool.

5.1.1 Prioritization:

Important

5.1.2 Conditions and Data Structures:

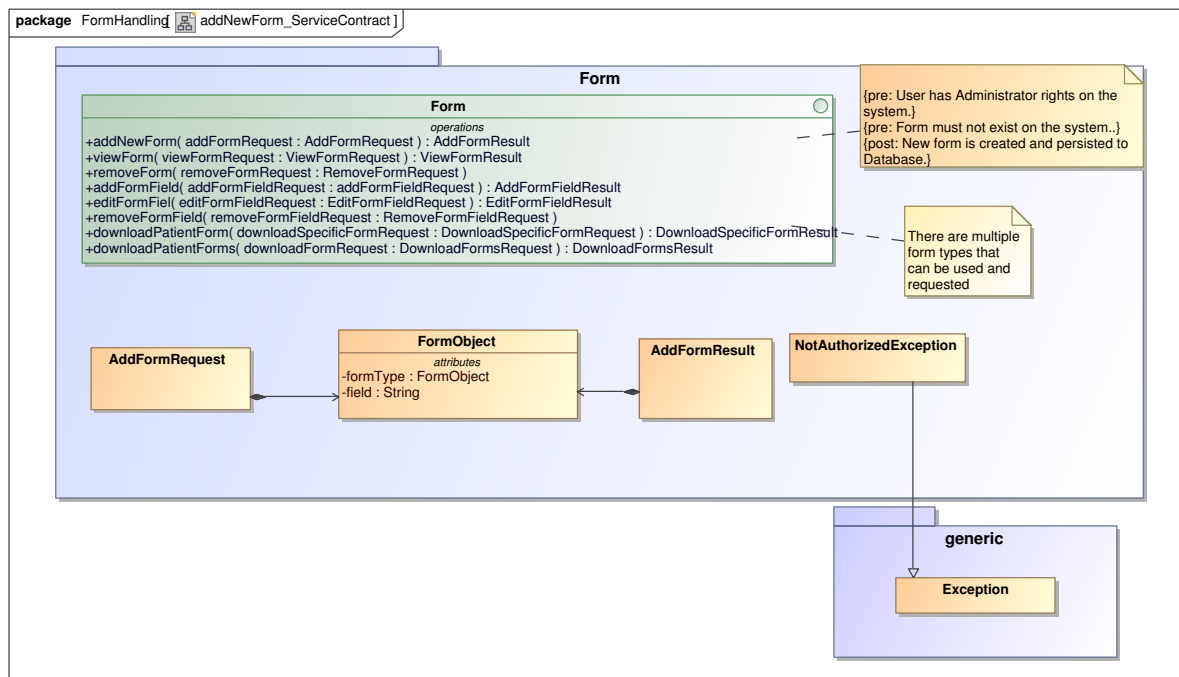
Pre-Conditions:

- User has Administrator rights on the system.
- Form must not exist on the system.

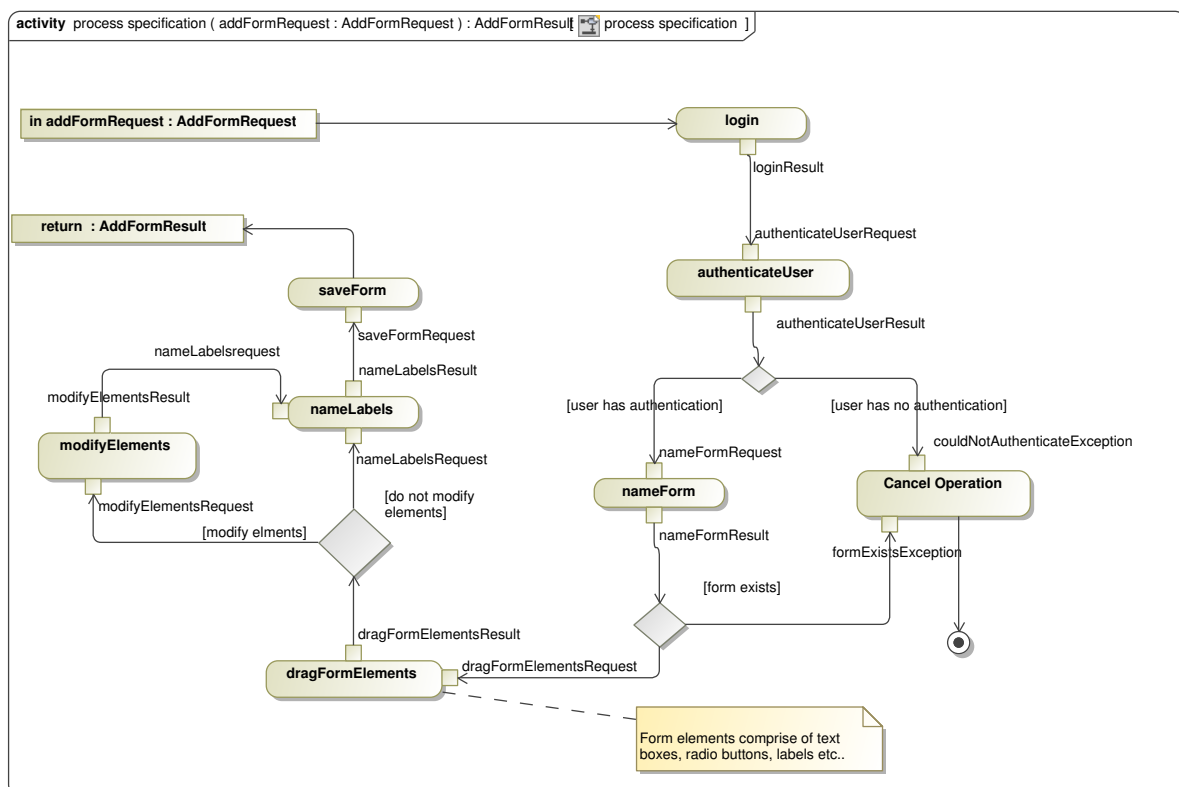
Post-Conditions:

- New form is created and persisted to Database

5.1.3 Service Contract:



5.1.4 Process Specifications:



5.2 viewForm

Description: This use-case allows for viewing of a form.

5.2.1 Prioritization:

Important

5.2.2 Conditions and Data Structures:

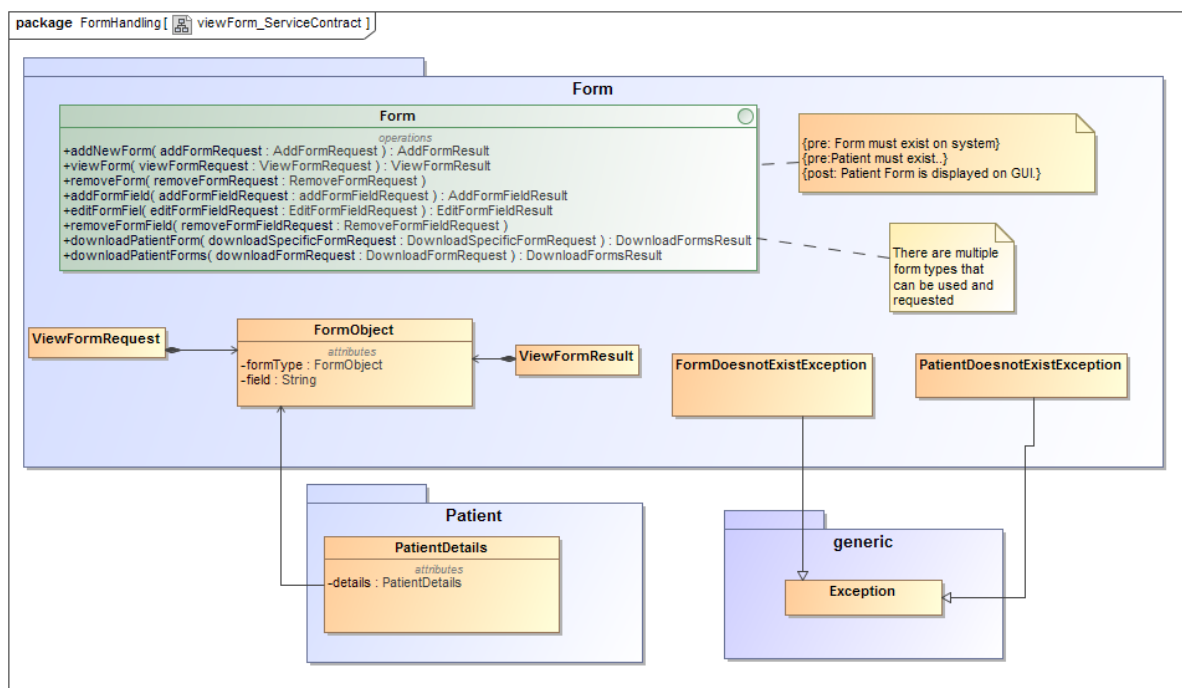
Pre-Conditions:

- Form must exist

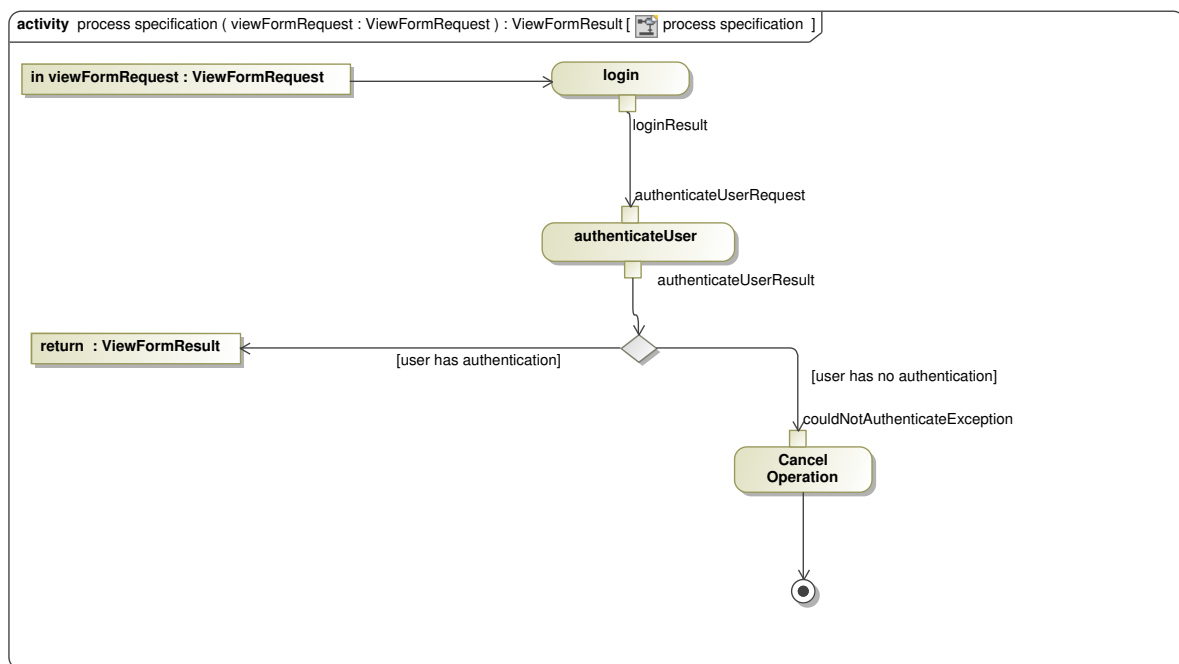
Post-Conditions:

- Form is displayed on GUI

5.2.3 Service Contract:



5.2.4 Process Specifications:



5.3 removeForm

Description: This use-case allows for the logical removal of a form, in that the form will be archived on the system so that it can be retrieved later if needed.

5.3.1 Prioritization:

Important

5.3.2 Conditions and Data Structures:

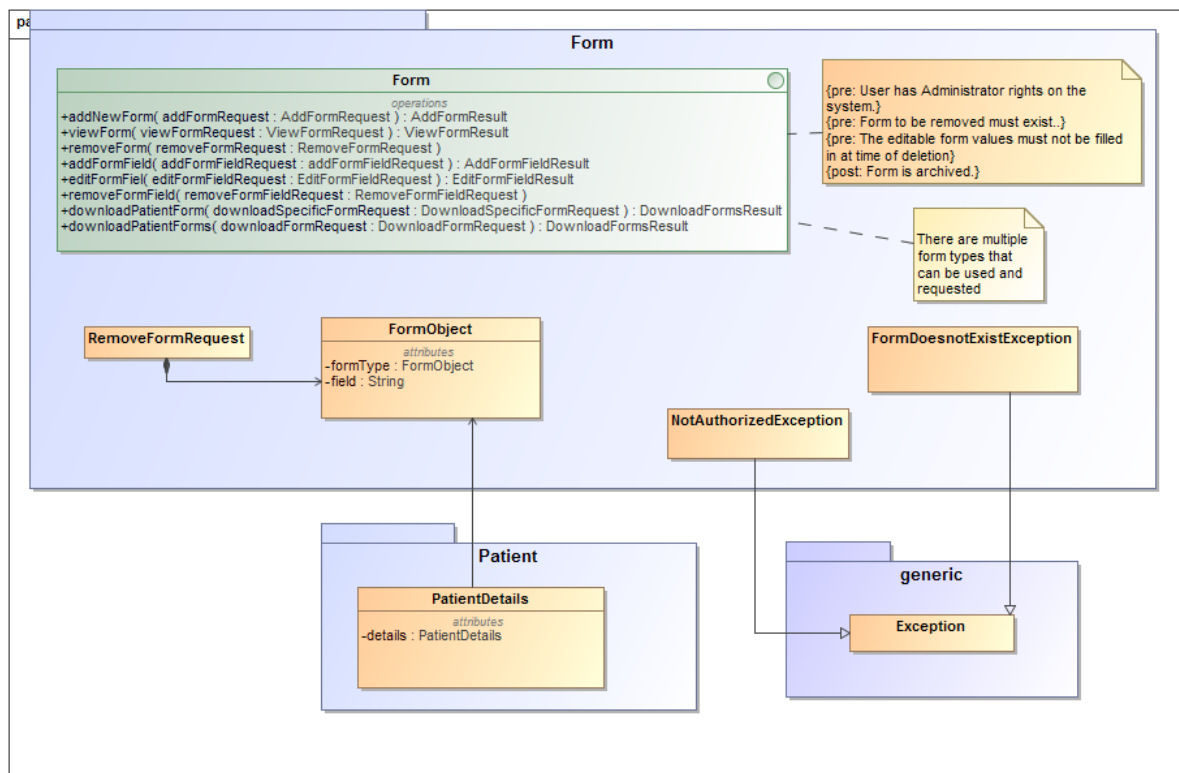
Pre-Conditions:

- User has Administrator rights on the system.
- Form to be removed must exist.
- The editable form fields must not be filled in at time for deletion

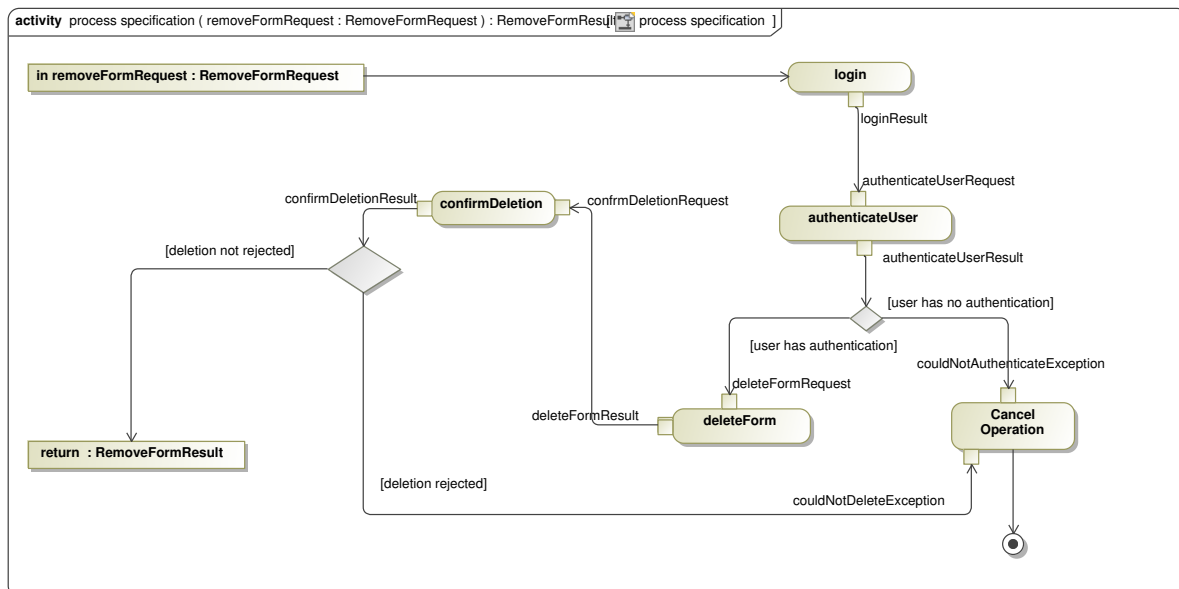
Post-Conditions:

- Form is archived

5.3.3 Service Contract:



5.3.4 Process Specifications:



5.4 addFormField

Description: This use-case allows for the addition of a new form field into a currently existing form

5.4.1 Prioritization:

Nice-to-Have

5.4.2 Conditions and Data Structures:

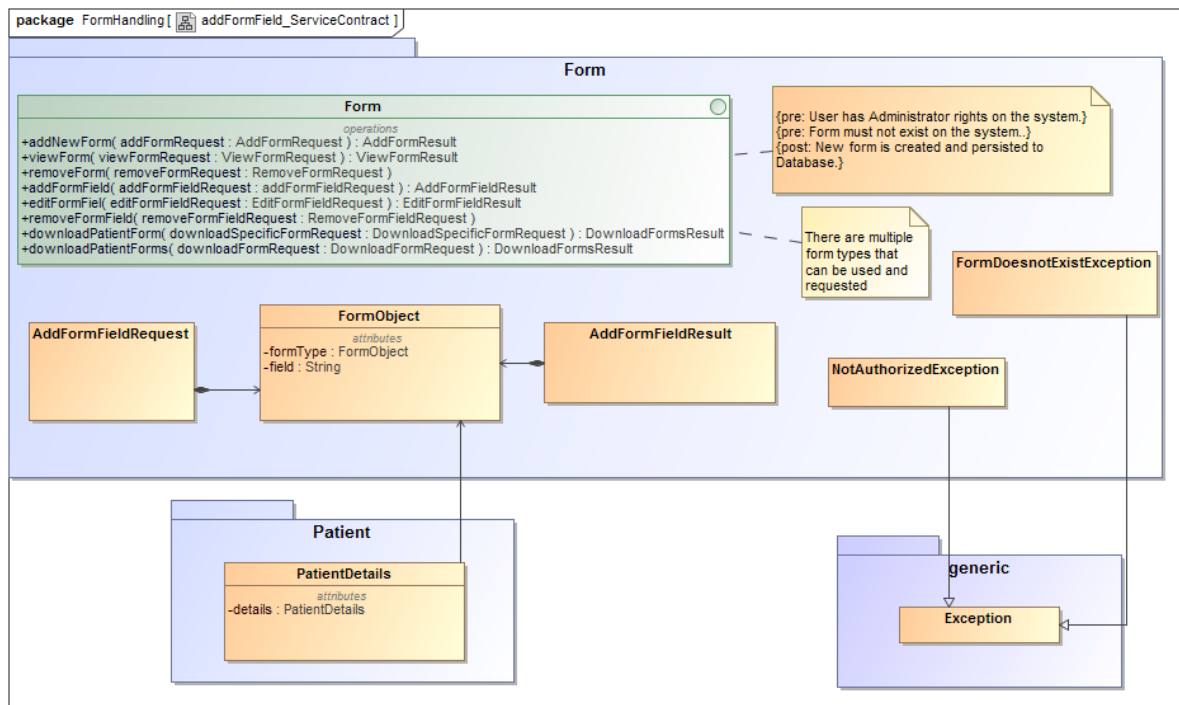
Pre-Conditions:

- User has Administrator rights on the system.
- Form to be edited must exist on system.

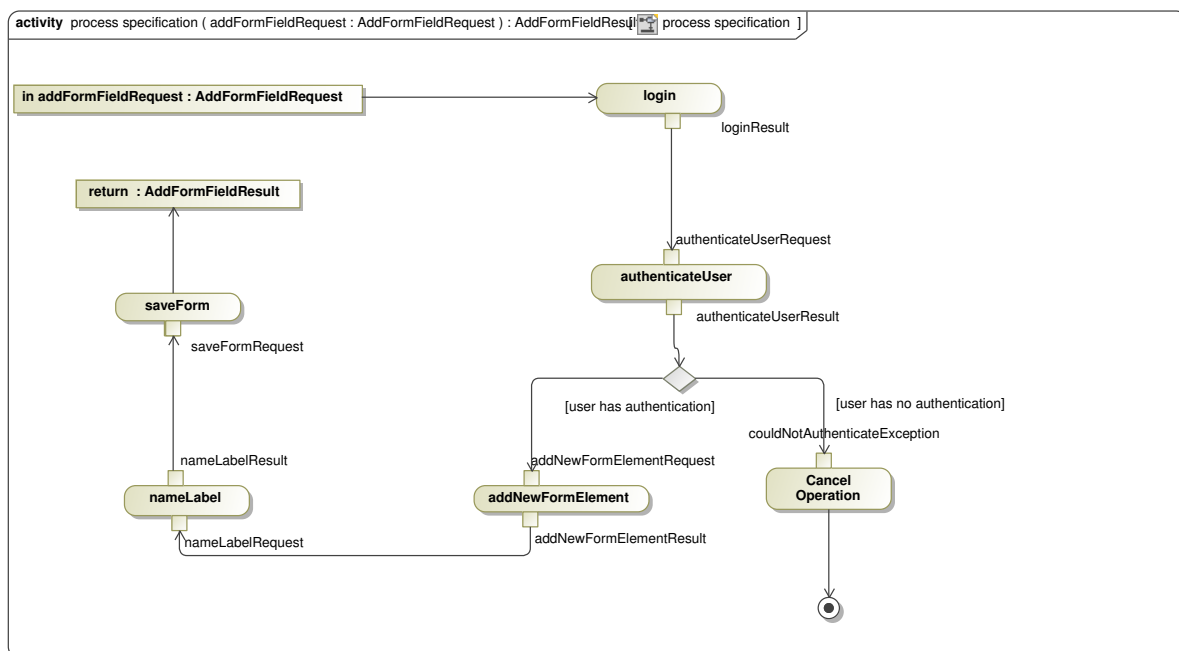
Post-Conditions:

- Field is added to GUI and persisted to DB

5.4.3 Service Contract:



5.4.4 Process Specifications:



5.5 editFormField

Description: This use-case allows for the editing of the field of a specific form that is on the system.

5.5.1 Prioritization:

Nice-to-Have

5.5.2 Conditions and Data Structures:

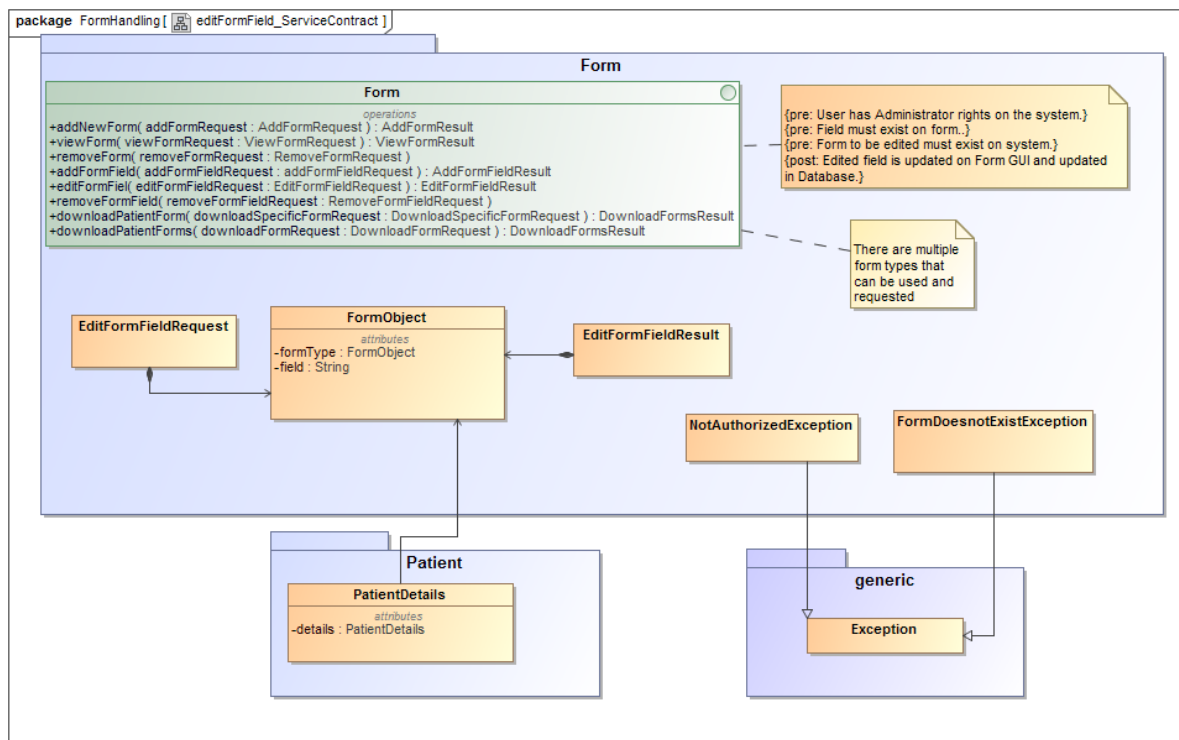
Pre-Conditions:

- User has Administrator rights on the system.
- Field must exist on form.
- Form to be edited must exist on system.

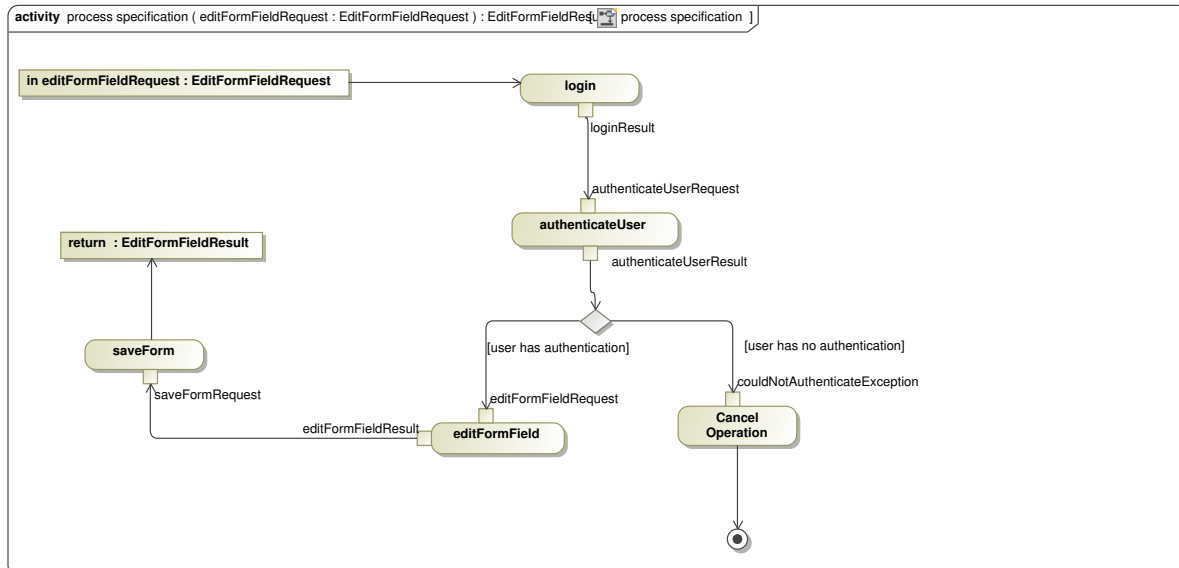
Post-Conditions:

- Edited field is updated on Form GUI and updated in Database

5.5.3 Service Contract:



5.5.4 Process Specifications:



5.6 removeFormField

Description: This use-case allows for the logical deletion of a form field in that the field will no longer be shown on the GUI but would still be present in the Database.

5.6.1 Prioritization:

Nice-to-Have

5.6.2 Conditions and Data Structures:

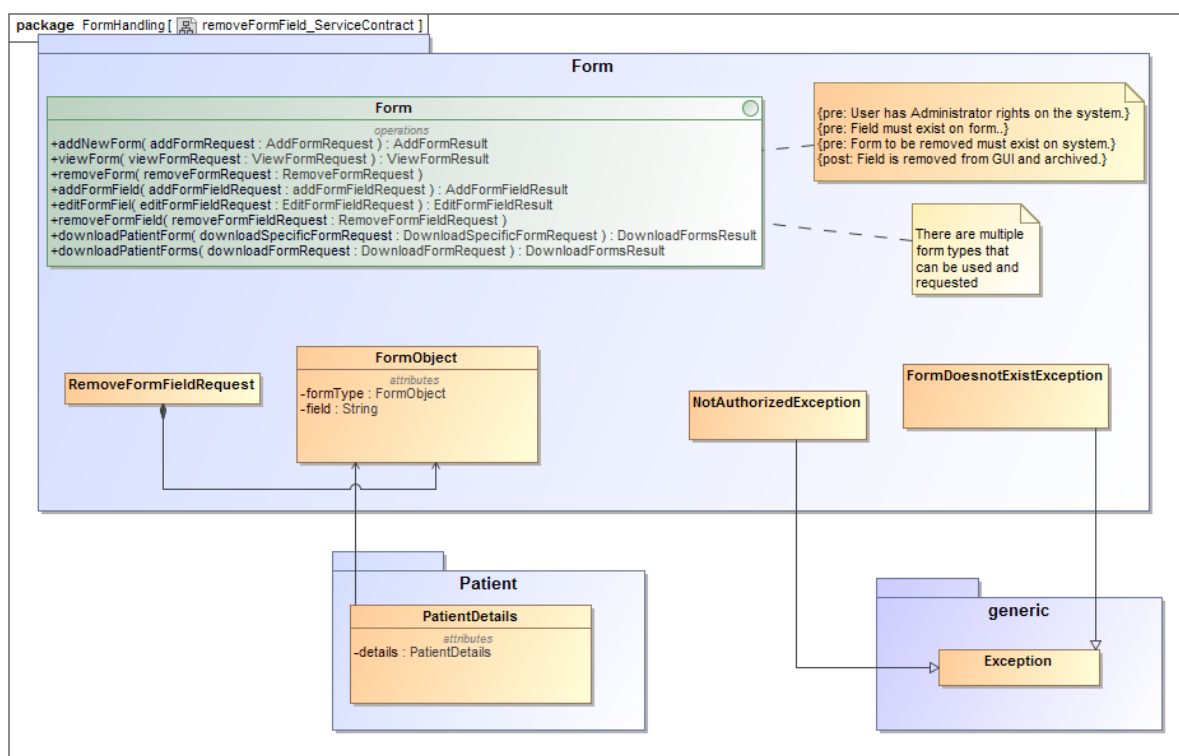
Pre-Conditions:

- User has Administrator rights on the system.
- Field must exist on form.
- Form to be removed must exist on system.

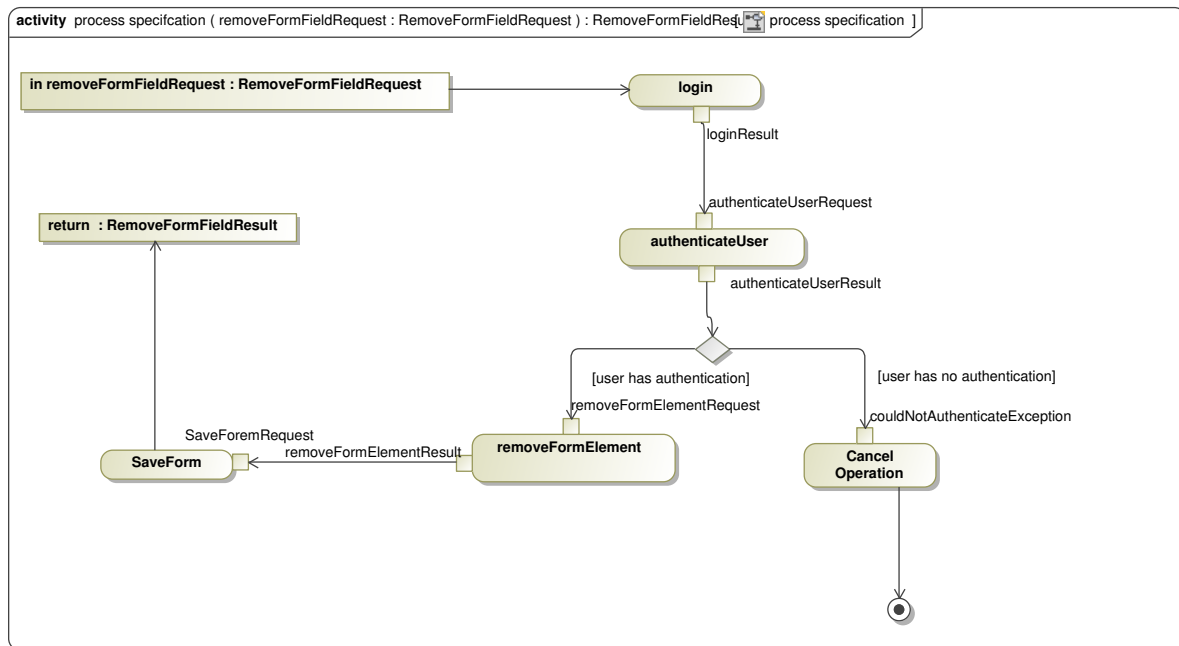
Post-Conditions:

- Field is removed from GUI and archived.

5.6.3 Service Contract:



5.6.4 Process Specifications:



5.7 downloadPatientForm

Description: This use-case allows for downloading of a particular form associated with the specified patient.

5.7.1 Prioritization:

Nice-to-Have

5.7.2 Conditions and Data Structures:

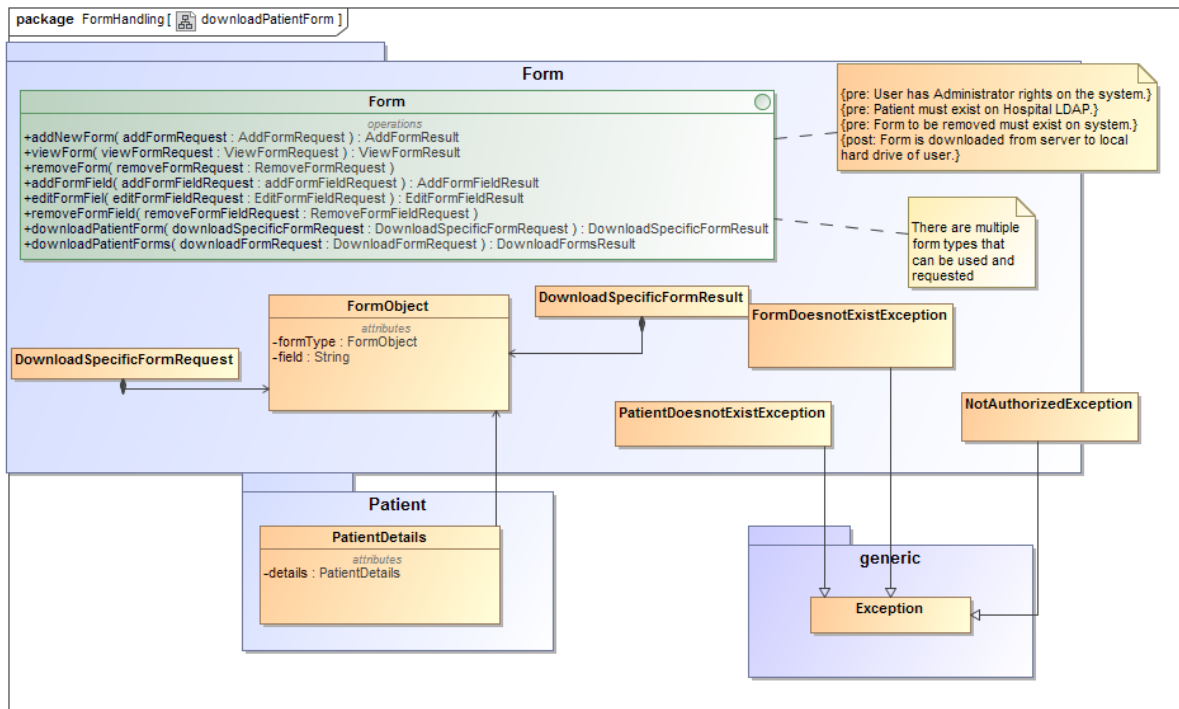
Pre-Conditions:

- User has Administrator rights on the system.
- Form must exist on the system.
- Patient must exist on Hospital LDAP.

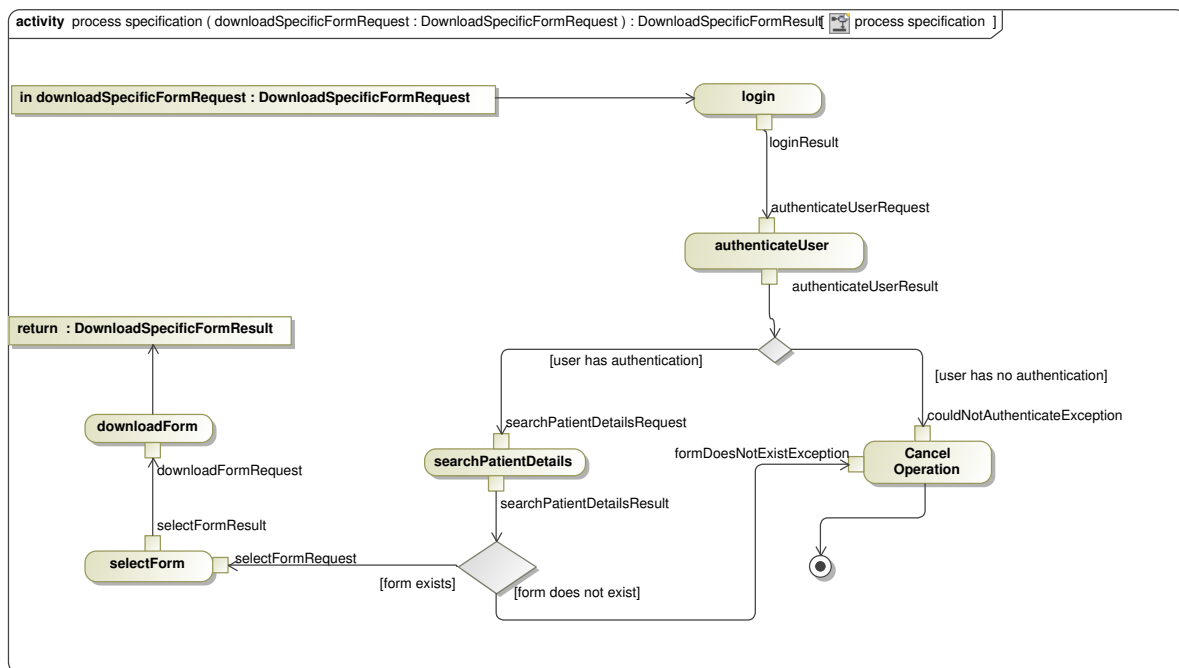
Post-Conditions:

- Form is downloaded from server to local hard drive of user

5.7.3 Service Contract:



5.7.4 Process Specifications:



5.8 downloadPatientForms

Description: This use-case allows for downloading of all forms associated with the specified patient.

5.8.1 Prioritization:

Nice-to-Have

5.8.2 Conditions and Data Structures:

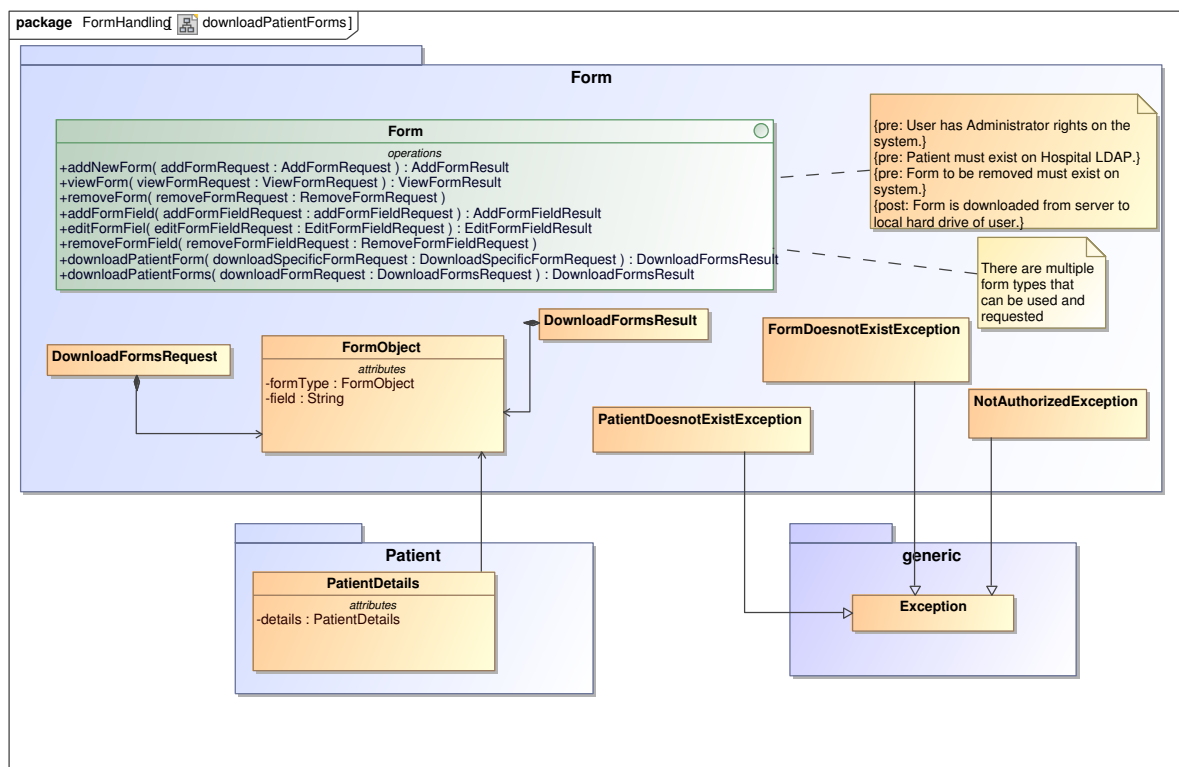
Pre-Conditions:

- User has Administrator rights on the system.
- Form must exist on the system.
- Patient must exist on Hospital LDAP.
- Patient information must exist.

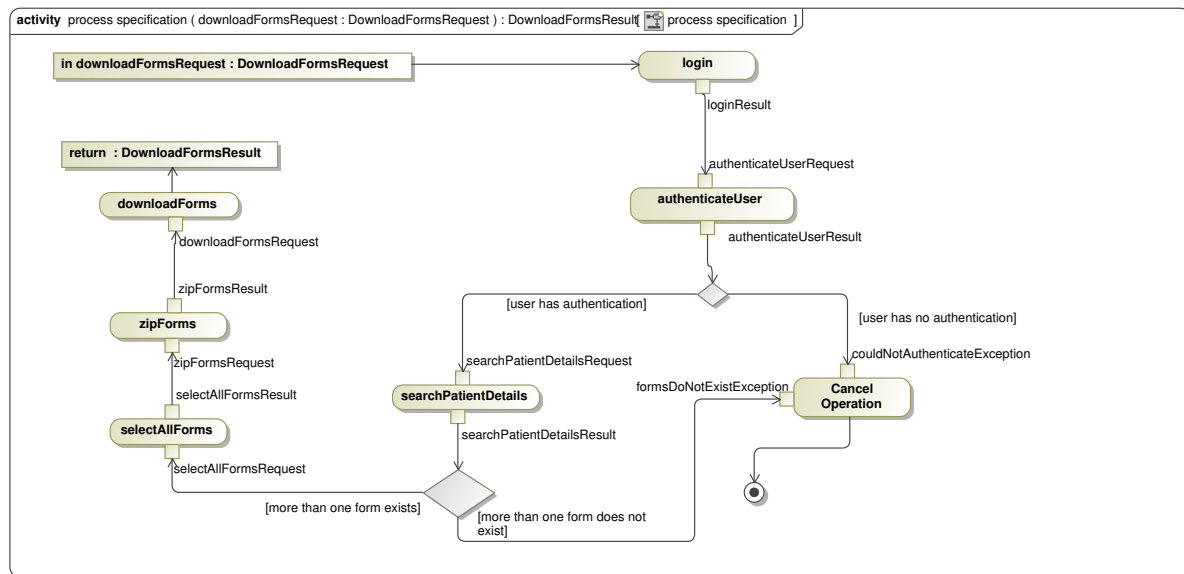
Post-Conditions:

- Forms are zipped and downloaded from server to local hard drive of user

5.8.3 Service Contract:



5.8.4 Process Specifications:



6 Patient Details

6.1 View patient details

Description: This allows the administrator to view the details of a specific patient.

6.1.1 Prioritization:

Critical

6.1.2 Conditions and Data Structures:

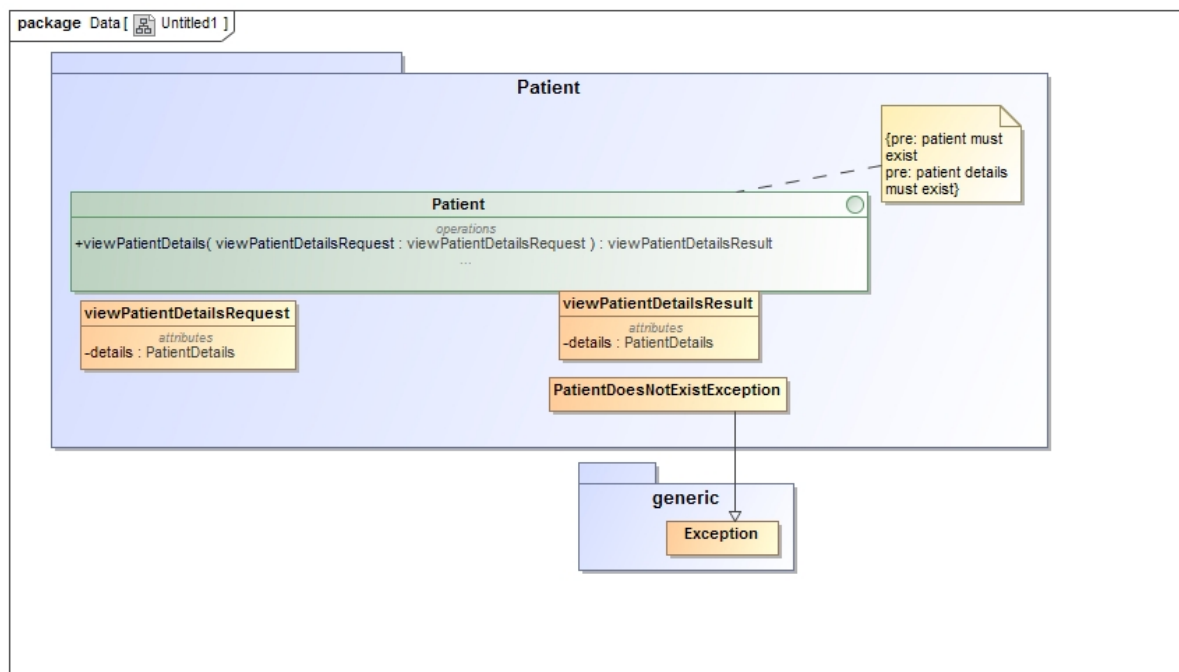
Pre-Conditions:

- The user must be authorised to view the patient's details
- The patient must exist

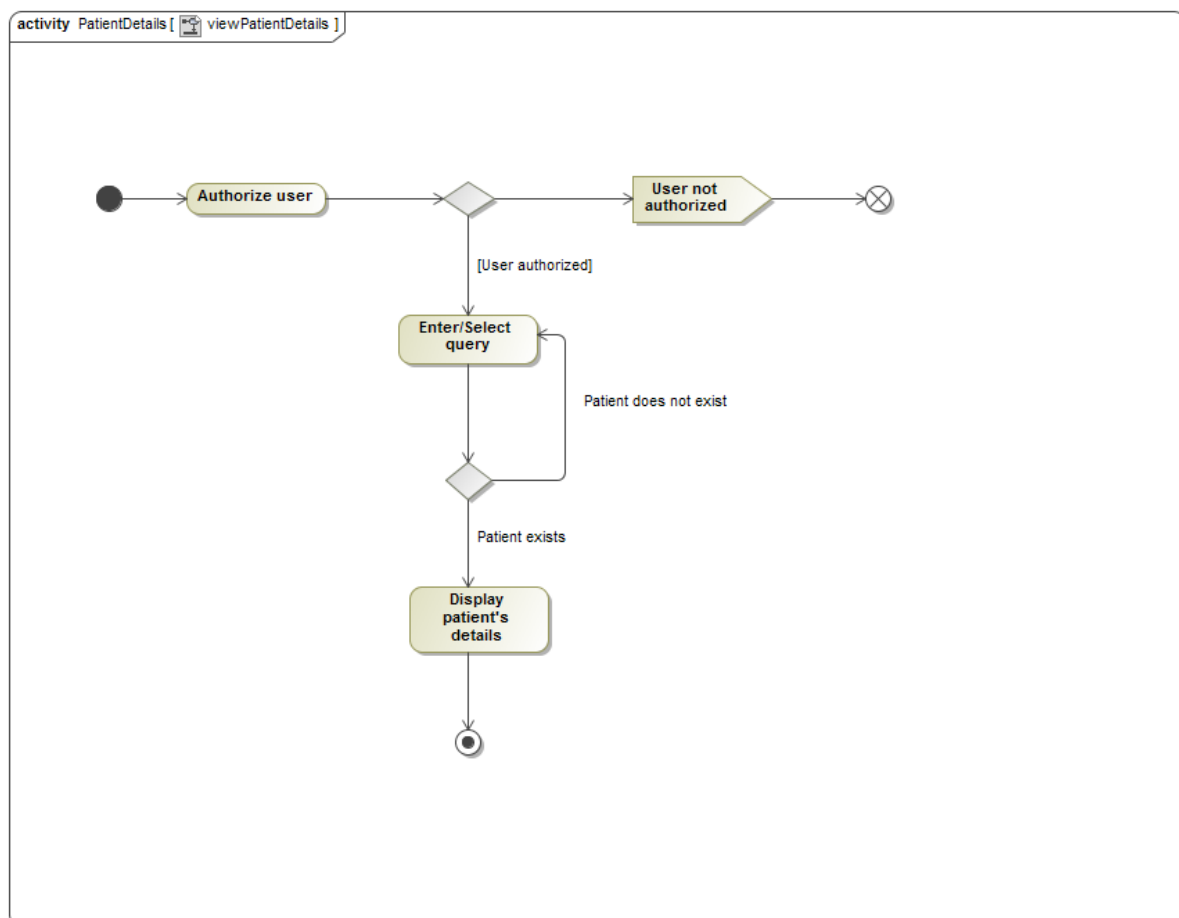
Post-Conditions:

- The user's details are displayed, as a form.

6.1.3 Service contract:



6.1.4 Activity diagram:



6.2 Update patient details

Description: This allows the administrator to update the details of a patient.

6.2.1 Prioritization:

Critical

6.2.2 Conditions and Data Structures:

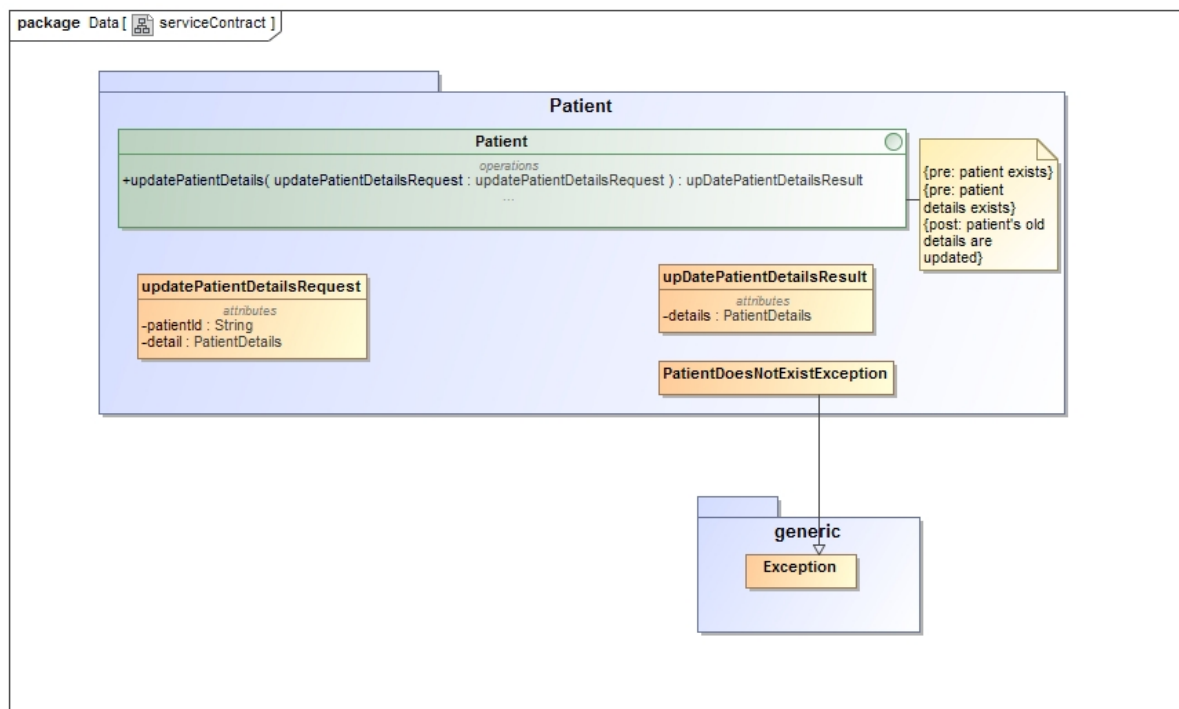
Pre-Conditions:

- The user must be authorised to update the patient's details
- The patient must exist
- Patient's details must exist

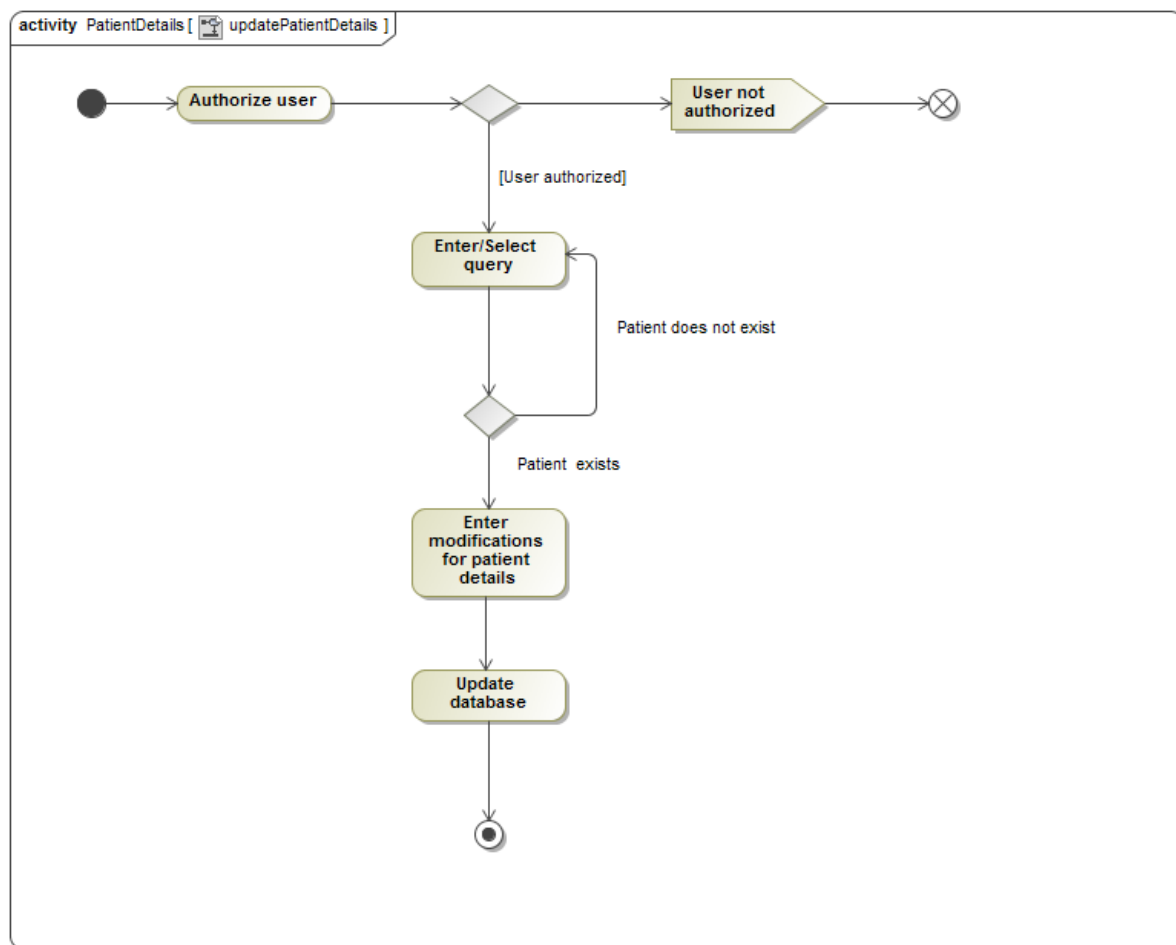
Post-Conditions:

- Patient's old details are updated

6.2.3 Service contract:



6.2.4 Activity diagram:



6.3 Query All Patient Details

Description: This allows the administrator to query the database of patients; predefined functions are created.

6.3.1 Prioritization:

Critical

6.3.2 Conditions and Data Structures:

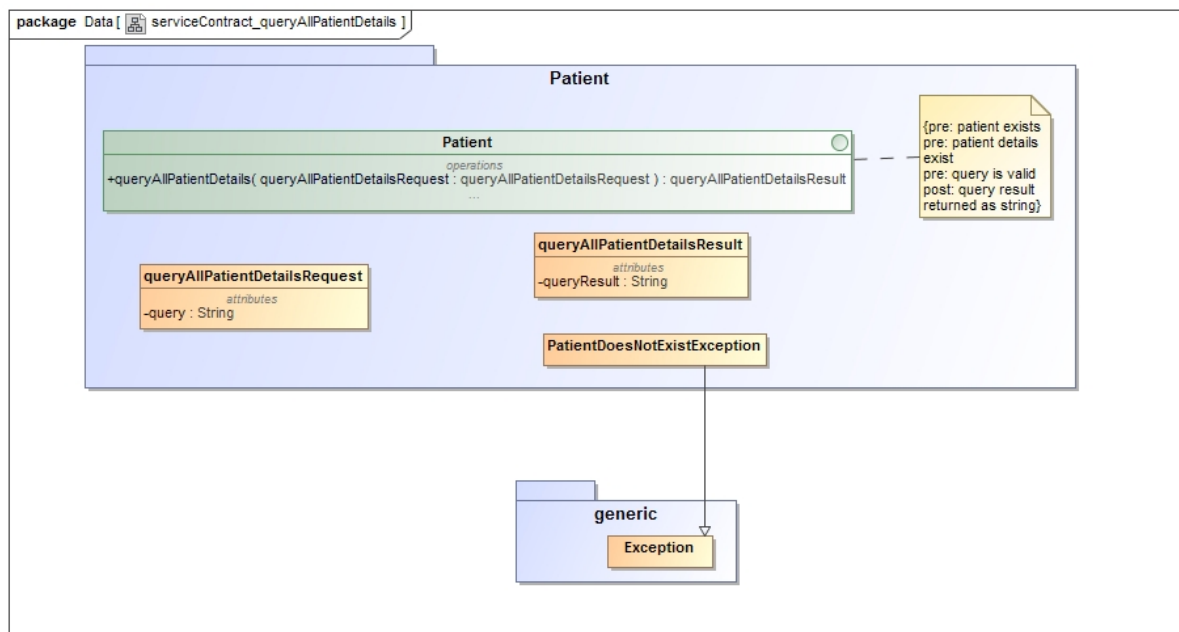
Pre-Conditions:

- The user must be authorised to query the database
- The patient must exist
- Query must be valid

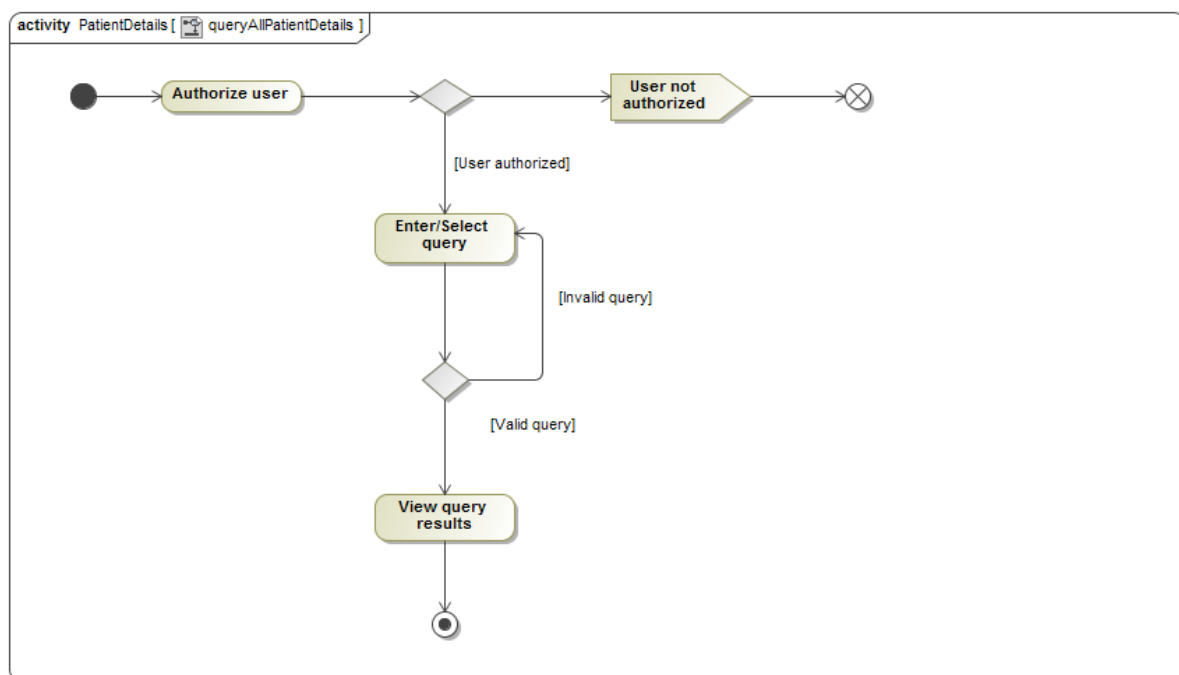
Post-Conditions:

- Query results are returned as a string

6.3.3 Service contract:



6.3.4 Activity diagram:



7 CRUDUser

7.1 Add new user

Description: This allows the administrator to add a new user to PIMS.

7.1.1 Prioritization:

Critical

7.1.2 Conditions and Data Structures:

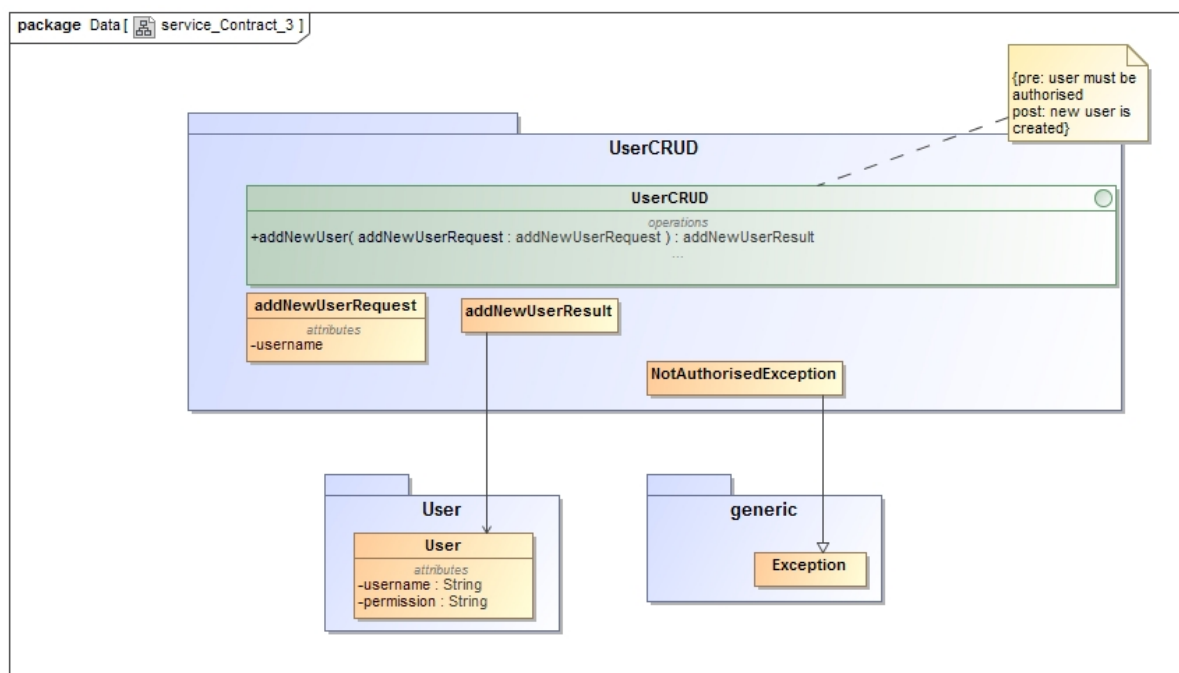
Pre-Conditions:

- The user must be authorised to add a new user

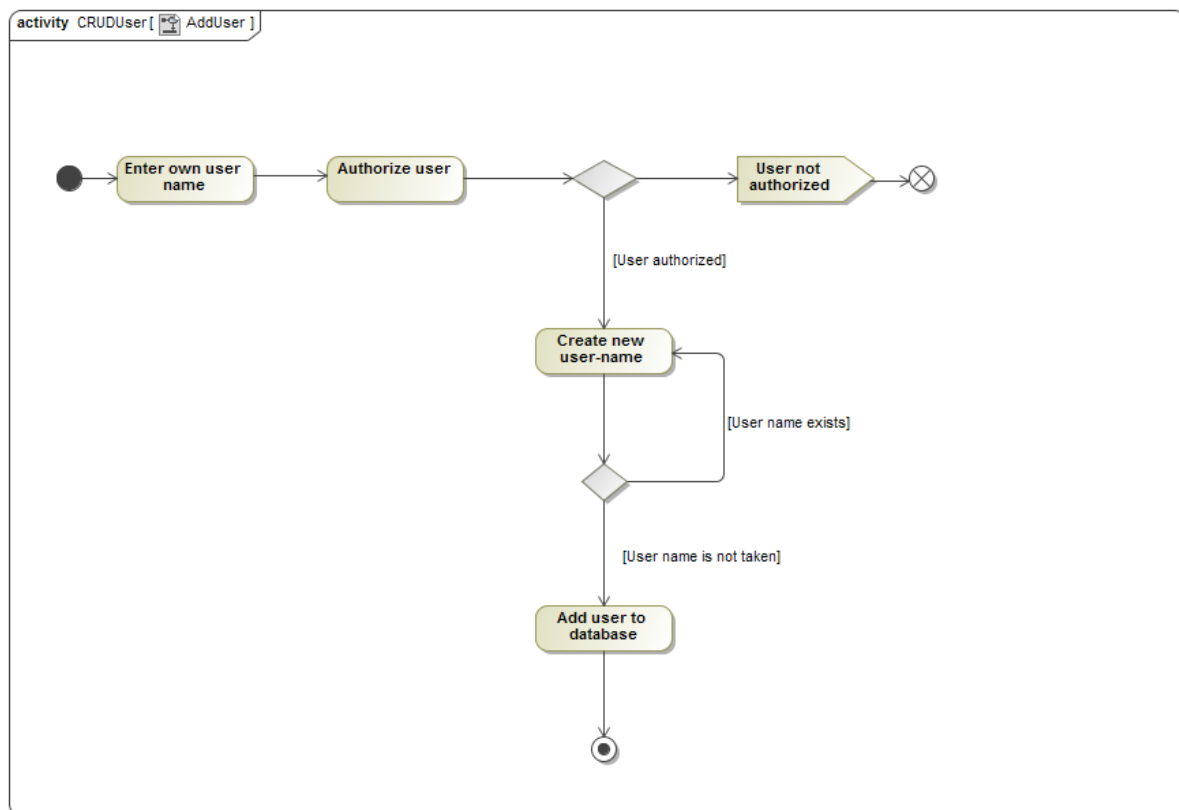
Post-Conditions:

- New user is added to PIMS.

7.1.3 Service contract:



7.1.4 Activity diagram:



7.2 Change user permissions

Description: This allows the administrator to promote or demote a user's access privilege.

7.2.1 Prioritization:

Important

7.2.2 Conditions and Data Structures:

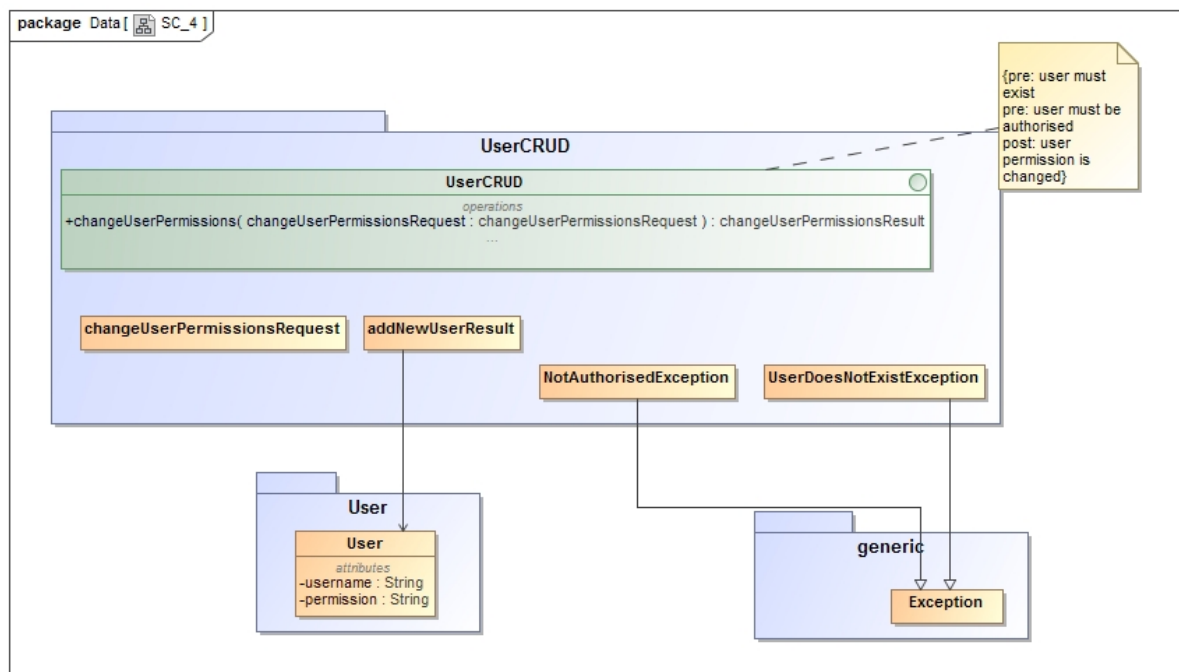
Pre-Conditions:

- The user must be authorised to change the permissions
- The user, whose permission is being changed, must exist

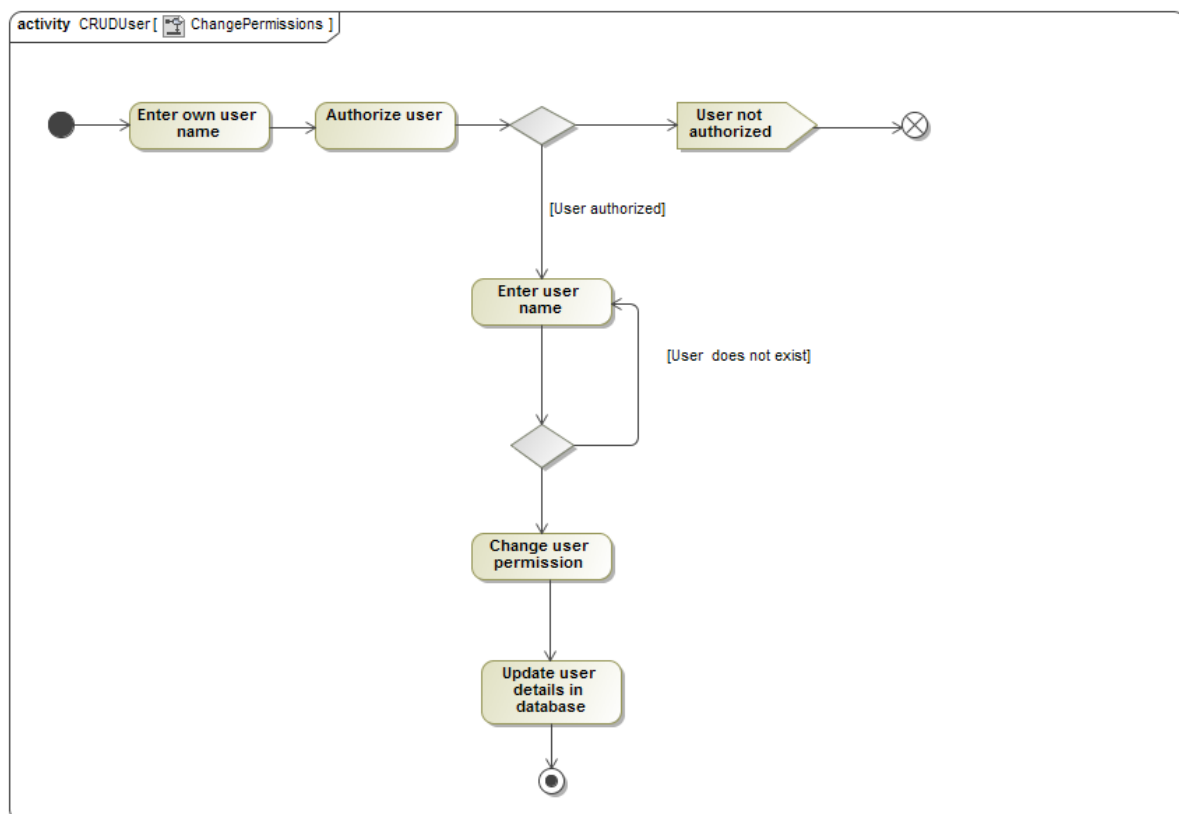
Post-Conditions:

- User's permissions is changed

7.2.3 Service contract:



7.2.4 Activity diagram:



7.3 Delete User

Description: This allows the administrator to remove a user from PIMS.

7.3.1 Prioritization:

Important

7.3.2 Conditions and Data Structures:

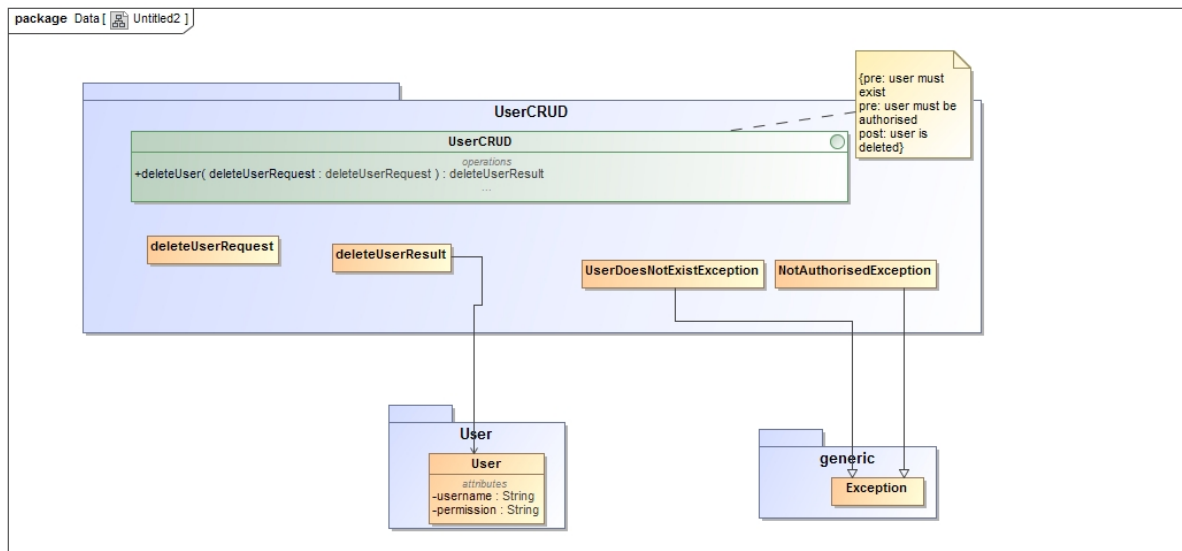
Pre-Conditions:

- The user must be authorised to remove users
- User must exist

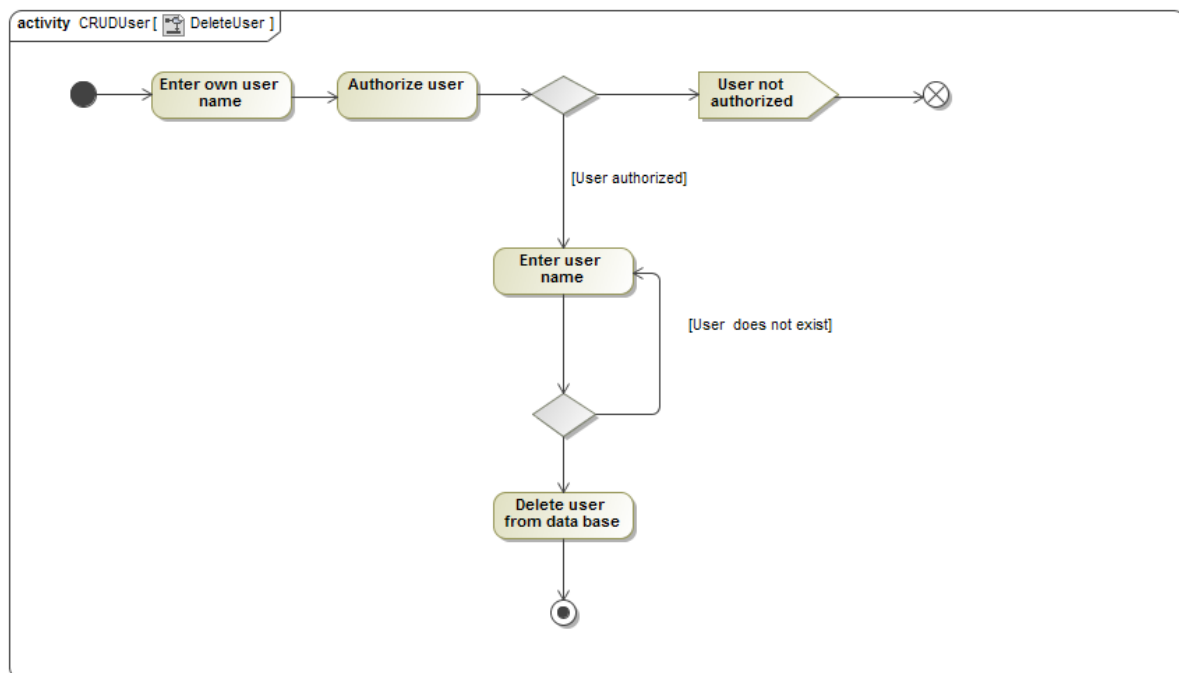
Post-Conditions:

- User is removed from PIMS.

7.3.3 Service contract:

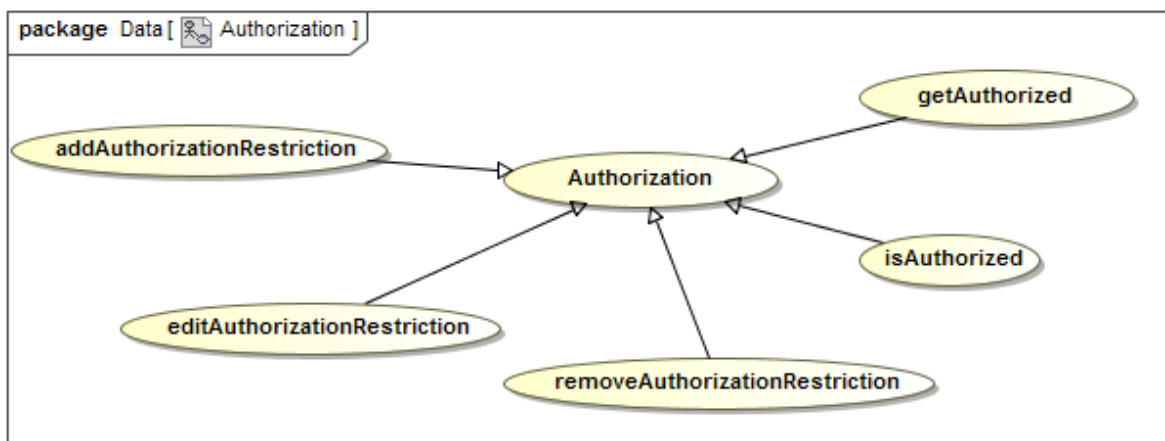


7.3.4 Activity diagram:



8 Authorization

8.1 Scope



Administration users will be able to add, remove, and edit authorization restrictions. Authorization restrictions refer to the services that a user cannot access. If a certain user group has a restriction on a service they will not be able to access that service. It can be assumed that if there is no restriction on a service then all users can access that service.

8.2 addAuthorized

Description: This use case allows the administration user to add an authorization restriction on a user.

8.2.1 Prioritization:

Critical

8.2.2 Conditions and Data Structures:

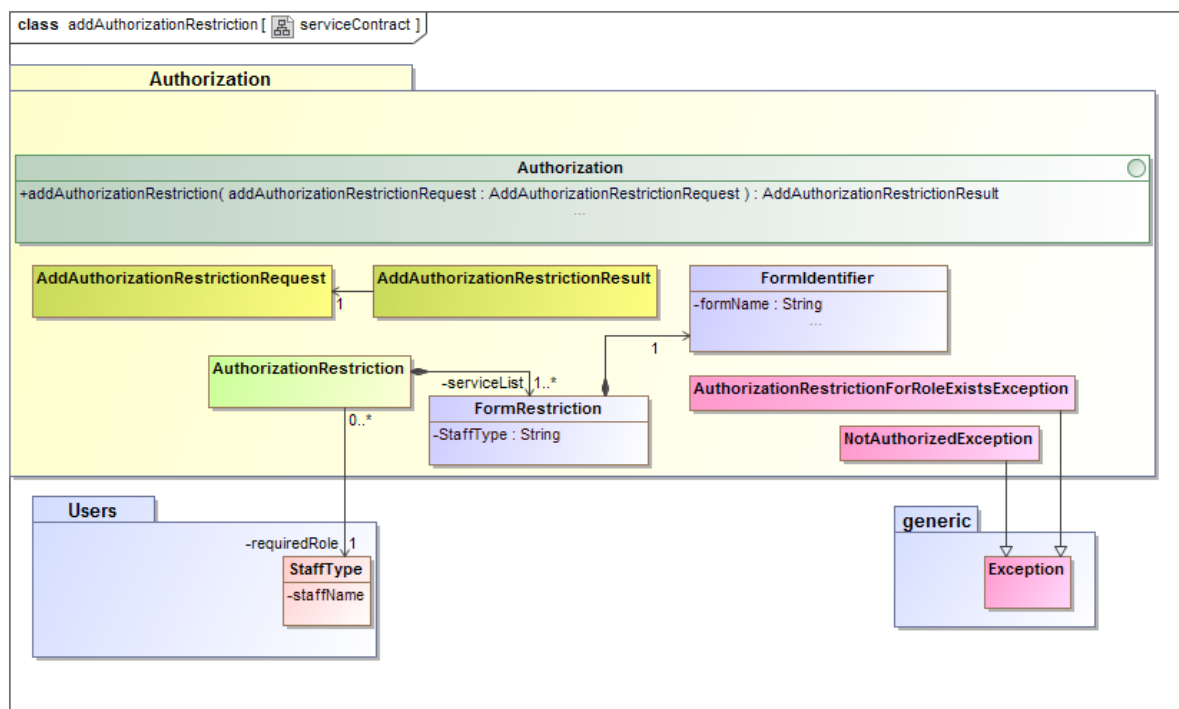
Pre-Conditions:

- User must have Administrator rights on the system.
- Restriction must not exist.

Post-Conditions:

- Restrictions must persist until removed

8.2.3 Service Contract:



8.3 getAuthorised

Description: This use case allows an admin user to get all the authorization features.

8.3.1 Prioritization:

Important

8.3.2 Conditions and Data Structures:

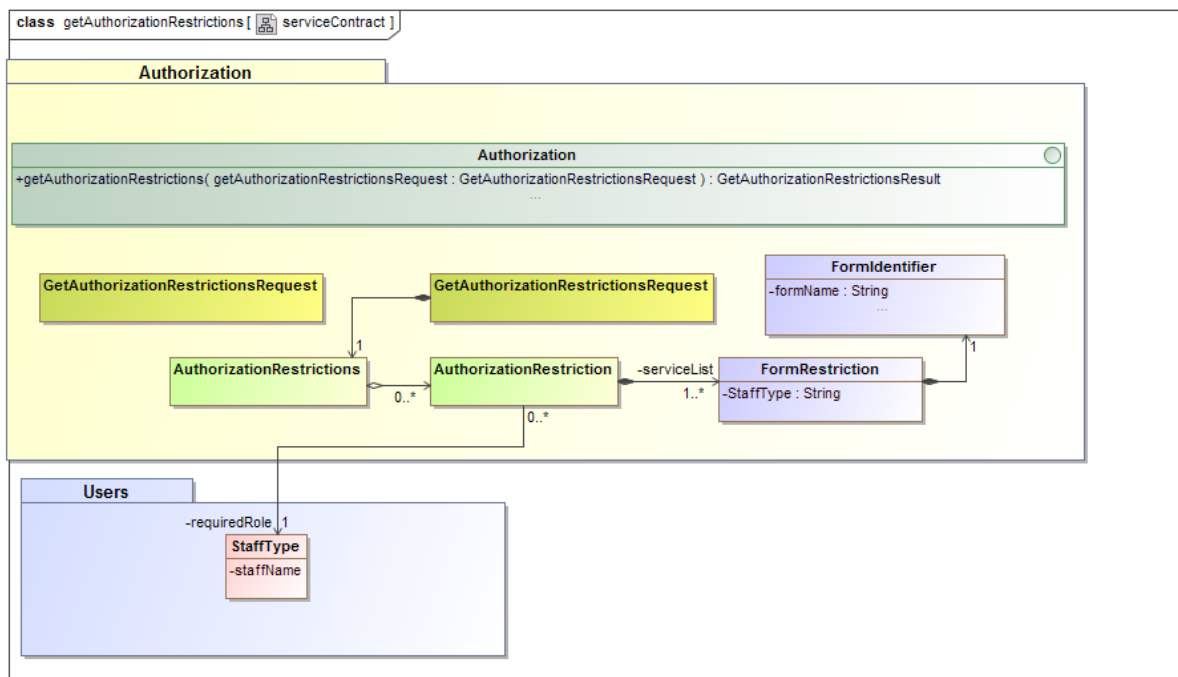
Pre-Conditions:

- User is an administrator.

Post-Conditions:

- Results are returned to the administrator.

8.3.3 Service Contract:



8.4 removeAuthorized

Description: This use case allows the administration user to remove an authorization restriction on a user.

8.4.1 Prioritization:

Critical

8.4.2 Conditions and Data Structures:

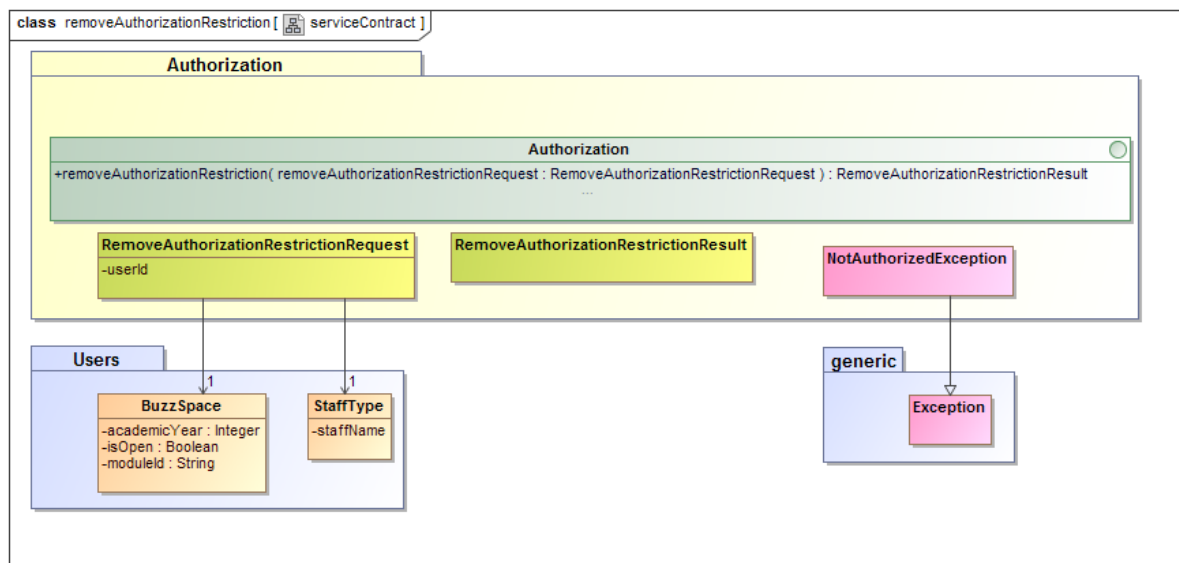
Pre-Conditions:

- User has an administration role

Post-Conditions:

- Authorization restriction is removed from the data base

8.4.3 Service Contract:



8.5 editAuthorized

Description: This use case allows the administration user to edit authorization restrictions on users

8.5.1 Prioritization:

Important

8.5.2 Conditions and Data Structures:

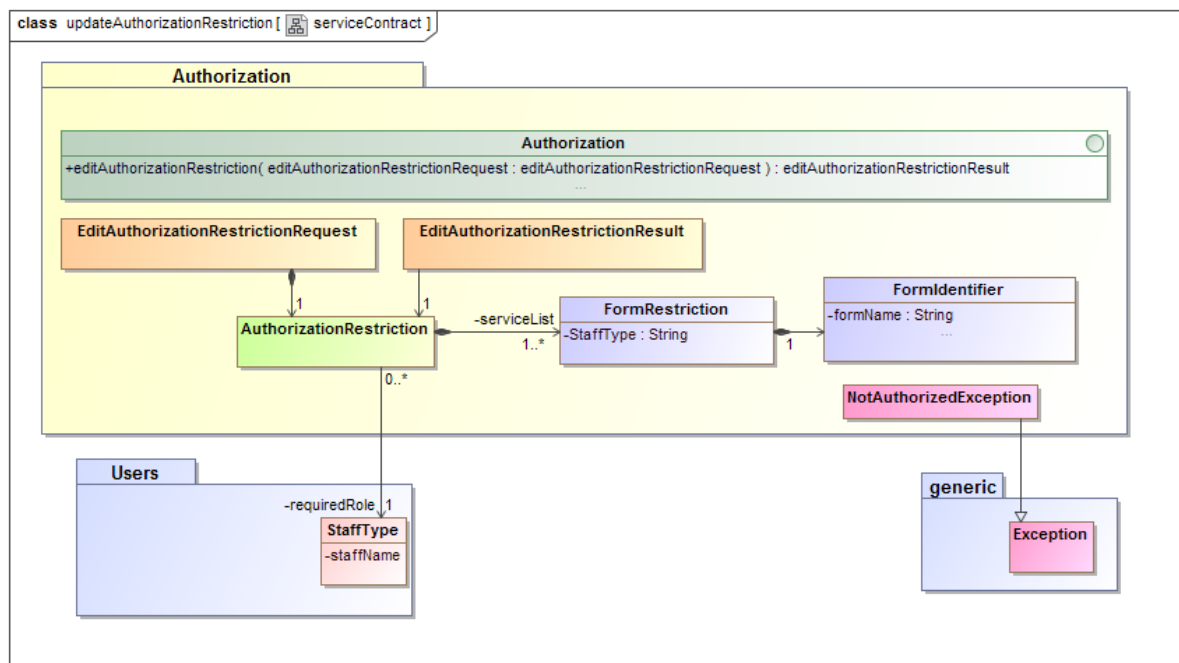
Pre-Conditions:

- User has an Administration Role

Post-Conditions:

- Updated authorization restrictions are stored in database

8.5.3 Service Contract:



8.6 isAuthorized

Description: This use case tests if a user is authorized to access a specific service

8.6.1 Prioritization:

critical

8.6.2 Conditions and Data Structures:

Pre-Conditions:

- User is authorized to access service

Post-Conditions:

- User is permitted to access the service.

8.6.3 Service Contract:

