Application Requirements and Design

Real-time Geospatial Data Processor and Visualiser

Client: Werner Raath

COEUS

Members:
Nsovo Baloyi 12163262
Maluleki Nyuswa 13040686
Keletso Molefe 14222583
Kamogelo Tswene 12163555

23 October 2016
v0.4

Contents

Document History **2**

1 Introduction **3**

2 Vision **3**

3 Background **3**

4 Functional Requirements and Application Design **4**
4.1 System Overview 4
4.2 Use Case Prioritization 4

5 Modular System **6**
5.1 Disaster Management Module 6
5.1.1 Scope 6
5.1.2 viewDisaster 7
5.1.3 queryDisaster 9
5.1.4 notifyNewDisaster 12
5.1.5 trackDisaster 14
5.1.6 addDisasterType 17
5.1.7 modifyDisasterType 19
5.2 Climate Management Module 21
5.2.1 Scope 21
5.2.2 viewWeather 22
5.3 User Module 25
5.3.1 Scope 25
5.3.2 registerUser 26
5.3.3 login 28
5.3.4 forgotPassword 31
5.3.5 changeProfile 33
5.3.6 logout 36
5.3.7 addPreferences 38
5.3.8 editPreferences 40
5.3.9 viewPreferences 42
5.4 Domain Model 45

6 Open Issues **46**

Document History

Version	Date	Changed By	Summary
v0.1	27 May 2016	Nsovo Baloyi Maluleki Nyuswa Keletso Molefe Kamogelo Tswene	First Draft
v0.2	29 July 2016	Maluleki Nyuswa Kamogelo Tswene	Second Draft
v0.3	9 September 2016	Nsovo Baloyi Maluleki Nyuswa Keletso Molefe Kamogelo Tswene	Third Draft
v0.4	23 October 2016	Nsovo Baloyi Maluleki Nyuswa Keletso Molefe Kamogelo Tswene	Final Draft

1 Introduction

The purpose of this document is to outline the core functionality of the Real-time Geospatial Data Processor and Visualiser system in a technology neutral design specification. The design is not necessarily a depiction of what the final system will look like as during the development of the system it is possible that new functionalities are added, or old ones removed.

In this document we will thoroughly discuss and layout the project's functional requirements to provide a clear overview of the system as a whole. An agile approach is being followed which involves an interactive and incremental method of managing and designing the system as described by the scrum methodology.

2 Vision

The aim of this project is to provide a fully functional system to facilitate the processing and visualisation of geospatial data in real-time[1], including but not limited to climate and fire disasters. The system should be maintainable, with detailed supporting documentation for the Real-time Geospatial Data Processor and Visualiser system. When implemented, the system should be able to store, process and visualise large amounts of data in real-time.

3 Background

In the business of disaster management, being able to monitor a disaster's spread and change in near-real time is essential for assessing damages e.g. wildfire burned area, flooded area, etc. and risk to human lives and assets. There are numerous mature open-source technology for processing the voluminous real-time data, but systems to visualise this data is woefully incomplete and its development remains at an infancy stage. Thus we need to contribute to this quest for knowledge. The need for systems such as the one suggested is very high. The system can save lives and serve as a tool to do research amongst other things.

[1] real-time

4 Functional Requirements and Application Design

The purpose of this section is to outline the functional requirements and application design. The application requirements and design specify the the core functional requirements, i.e. the user's requirements around use cases and not secondary functional requirements arising from realizing non-functional requirements.

4.1 System Overview

Figure 1 shows a high-level overview of the system.

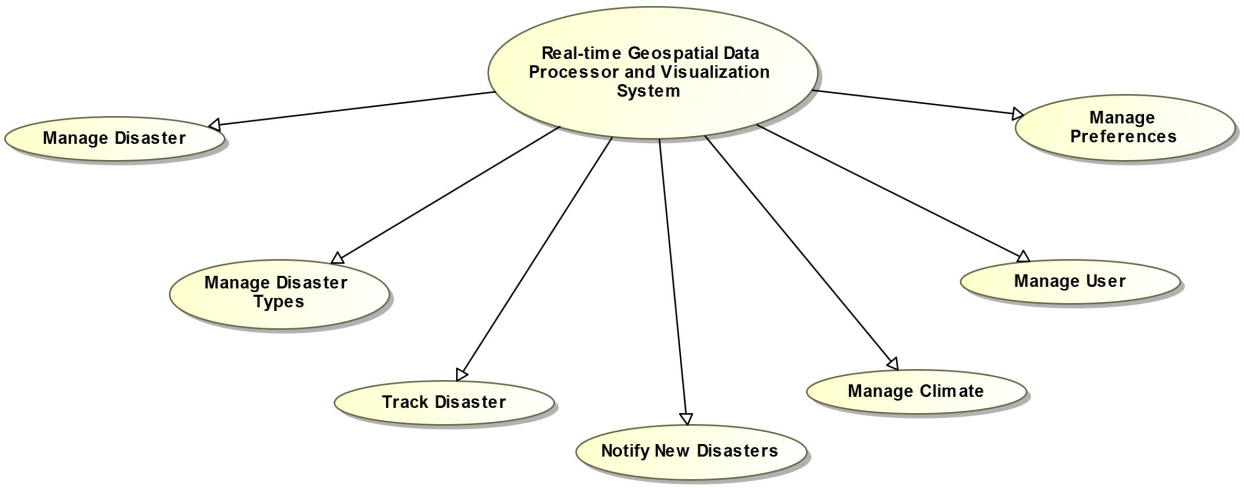

Figure 1: Overview of the System

4.2 Use Case Prioritization

The system use cases are characterised as follows:

- **Manage Disaster - Critical**
  This use case is critical since the main functionality of the system requires disaster management.
- **Manage Disaster Types - Important**
  An authorized user should be able to add a new disaster type or remove a redundant disaster type.

- **Track Disaster - Important**
  Users need to be able to track an active disaster that has occurred for feedback purposes.

- **Notify New Disaster - Important**
  A user that is currently online should be able to get instantaneous notification of a newly occurring disaster.

- **Manage Climate - Important**
  A user should be able to view normal weather climate at any time they wish to.

- **Manage User - Important**
  A user should be able to register with the system because as a registered user they are able to access all the functionality provided by the system. An authorized user (admin) once logged in is able to access and/or modify relevant system data.

- **Manage Preferences - Important**
  A registered user should be able to add, edit or view their weather and disaster preferences.

5 Modular System

5.1 Disaster Management Module

5.1.1 Scope

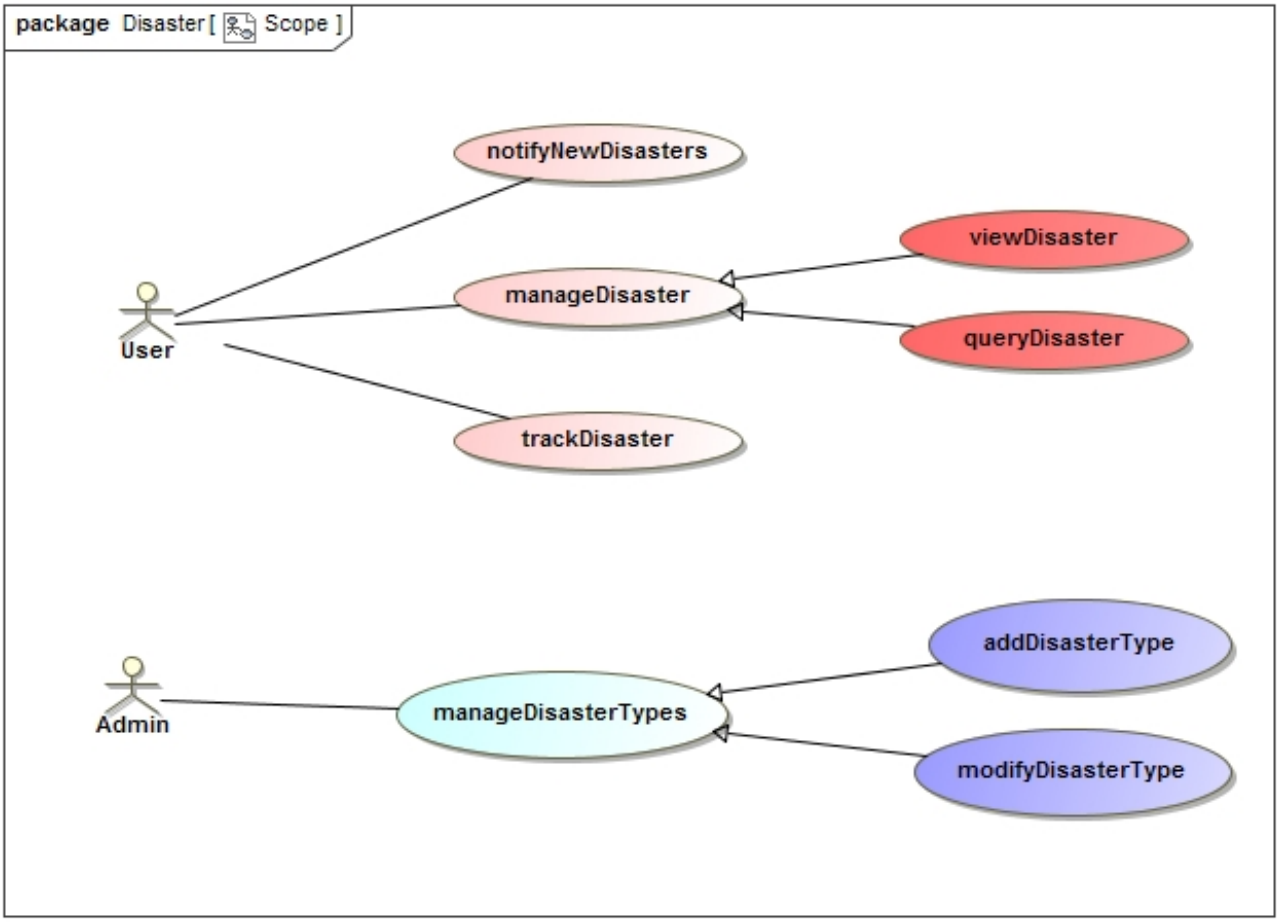

Figure 2: The scope of functionality required from the disaster module

Admin can add and/or modify the different kinds of disaster types that are required whilst users are able to track and manage disasters as well as get notifications for new disasters.

5.1.2 viewDisaster

A user, given that they are logged in, can view disasters by going on to the disasters tab on the system and viewing all the disasters, active or not, that are occurring throughout the world. Below are the service contract, activity diagram and functional requirements diagram for viewDisaster.

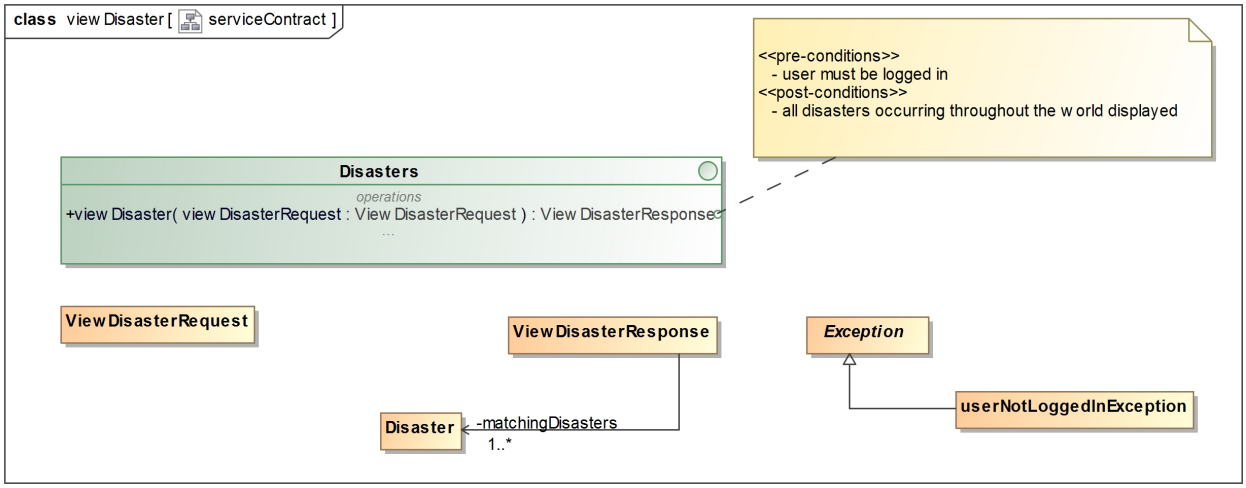

Figure 3: The service contract for viewDisaster

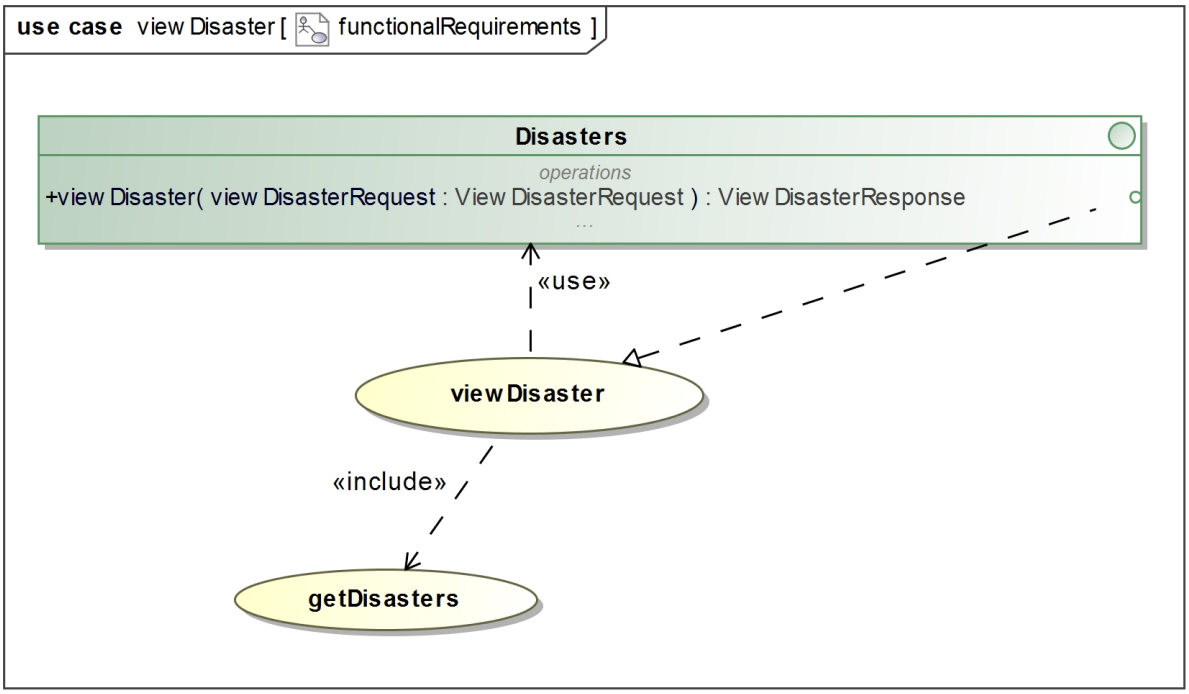

Figure 4: The functional requirements diagram for viewDisaster

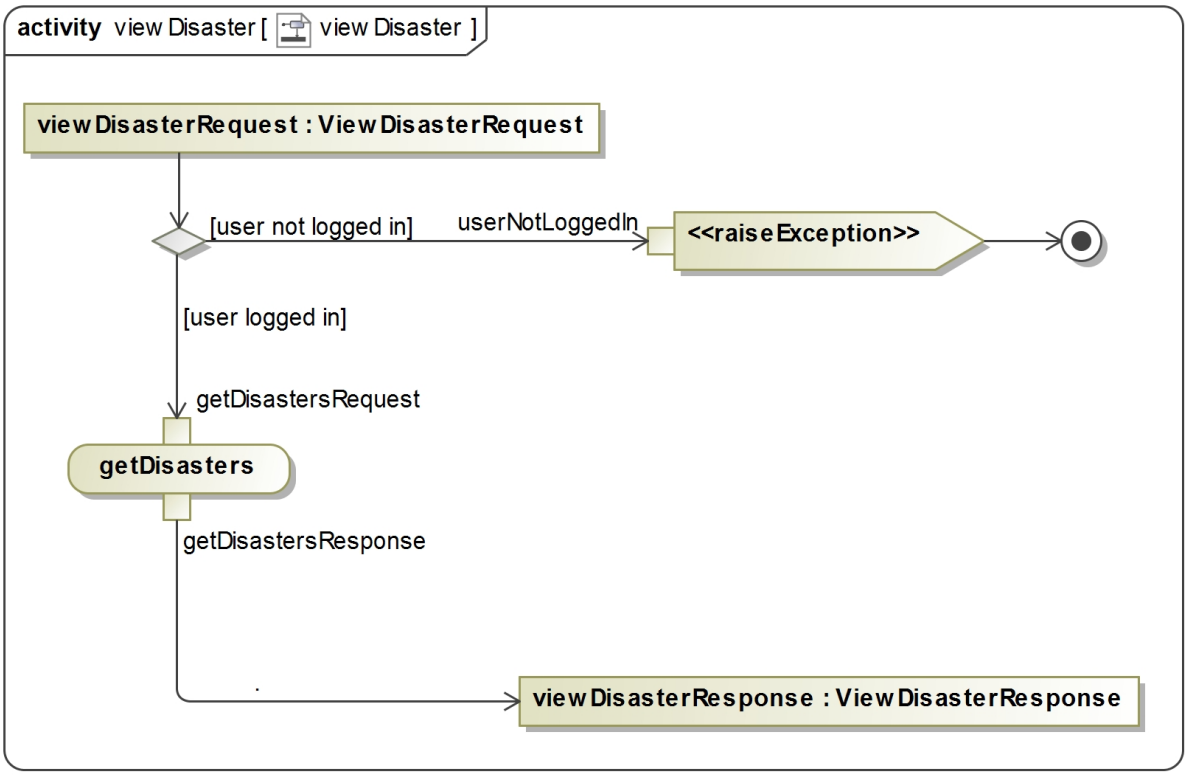

Figure 5: The activity diagram for viewDisaster

5.1.3 queryDisaster

A user, given that they are logged in, will be able to query the system for specific disasters, by entering search criteria such as a date, location or a disaster feature like magnitude level for earthquakes. That way only disasters matching the specified criteria are returned to the user. Below are the service contract, activity diagram and functional requirements diagram for queryDisaster.

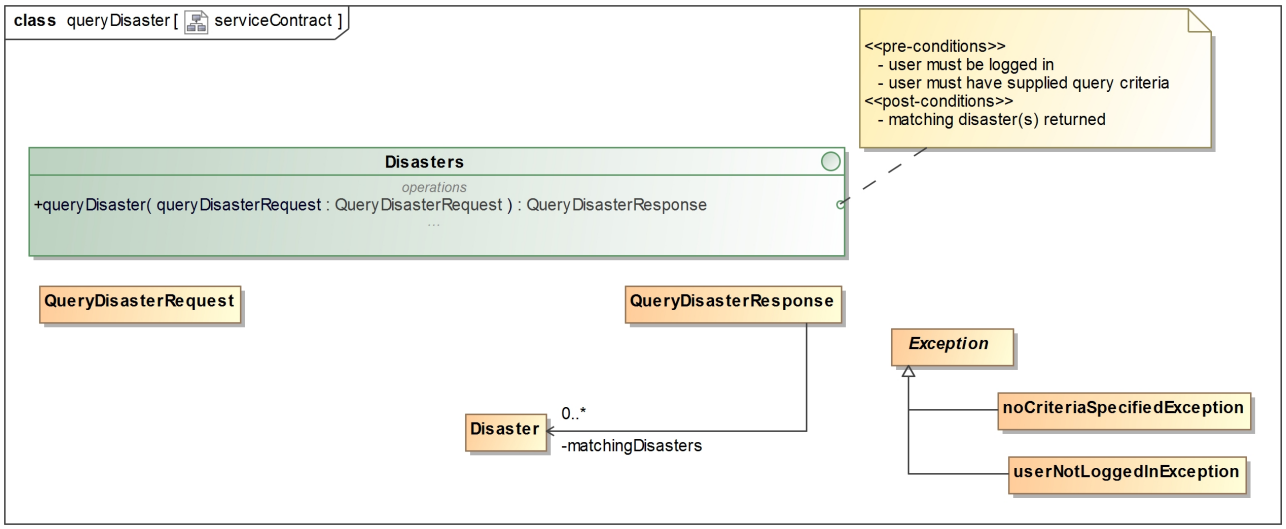

Figure 6: The service contract for queryDisaster

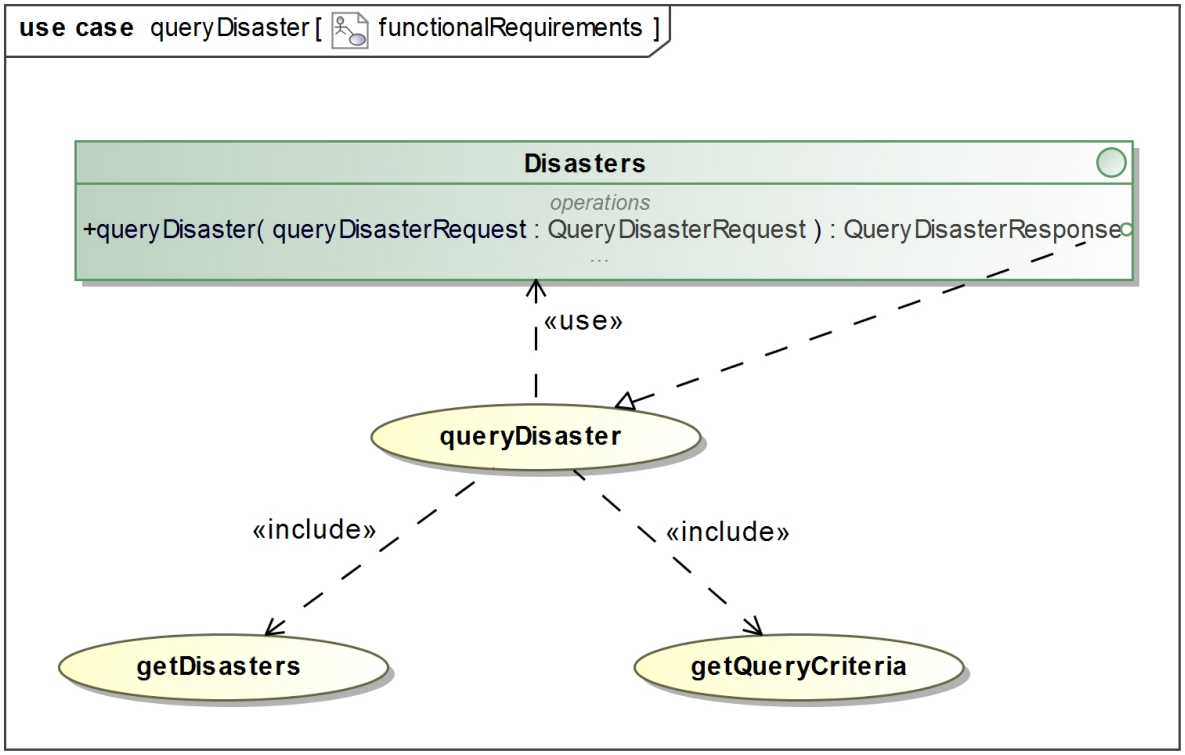

Figure 7: The functional requirements diagram for queryDisaster

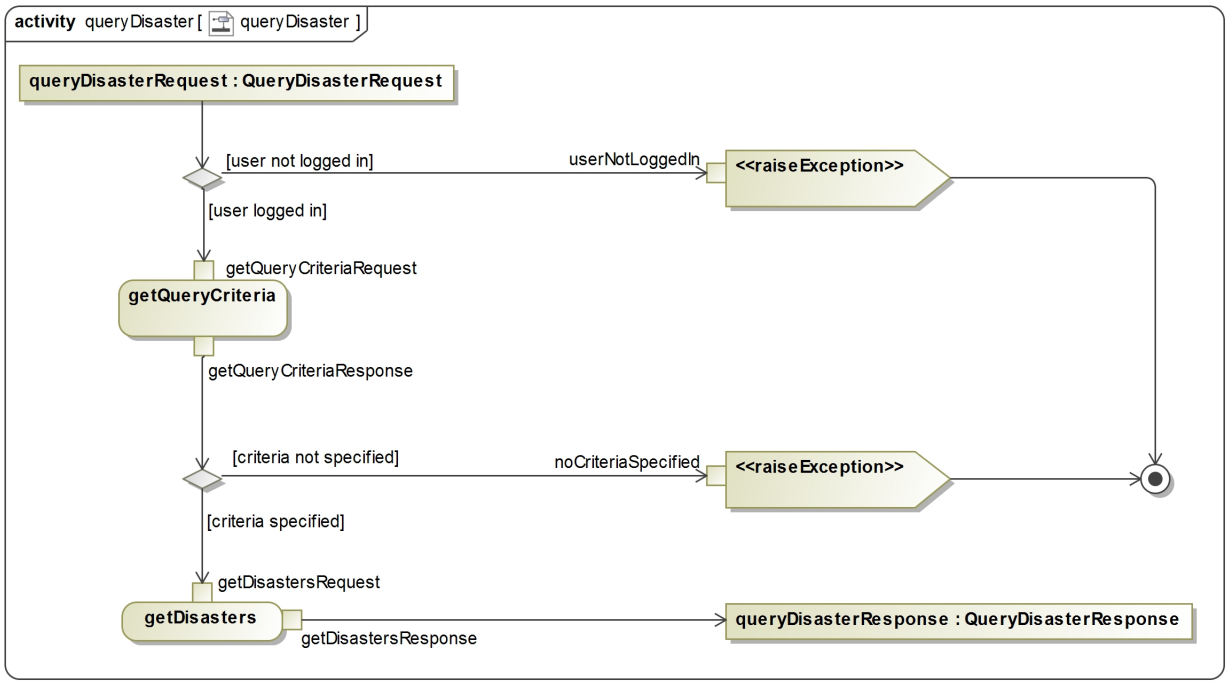

Figure 8: The activity diagram for queryDisaster

5.1.4 notifyNewDisaster

A user, given that they are logged in, will be able to be notified of any new disasters that are occuring as of the time they are using the system as long as that disaster is not visualized on the system yet. Below are the service contract, activity diagram and functional requirements diagram for notifyNewDisaster.

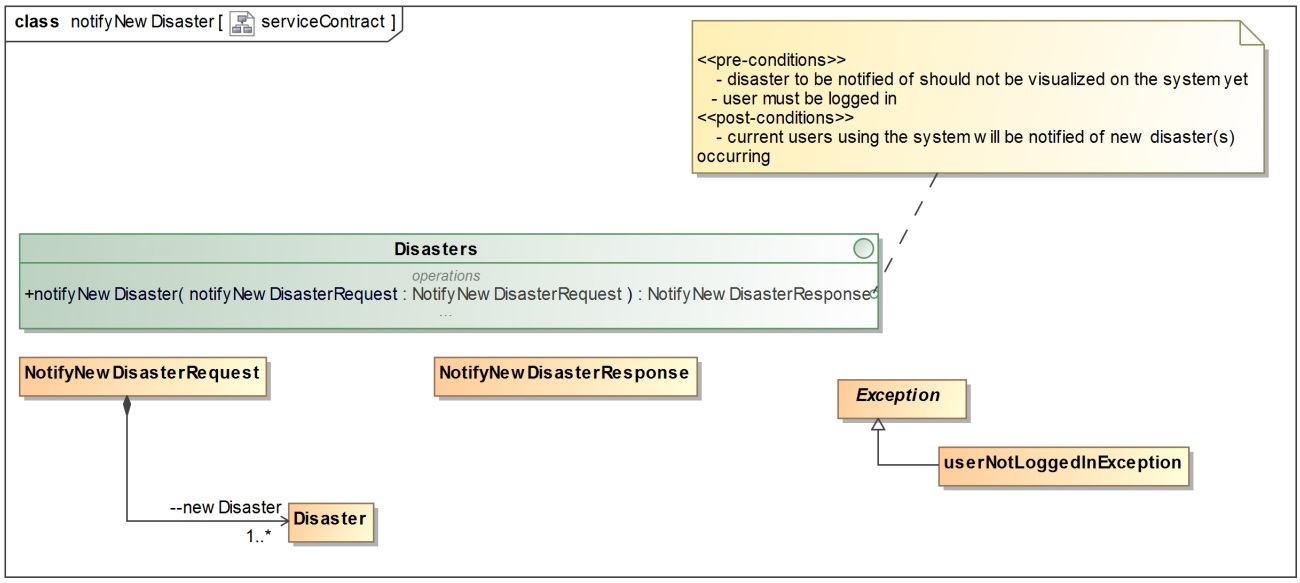

Figure 9: The service contract for notifyNewDisaster

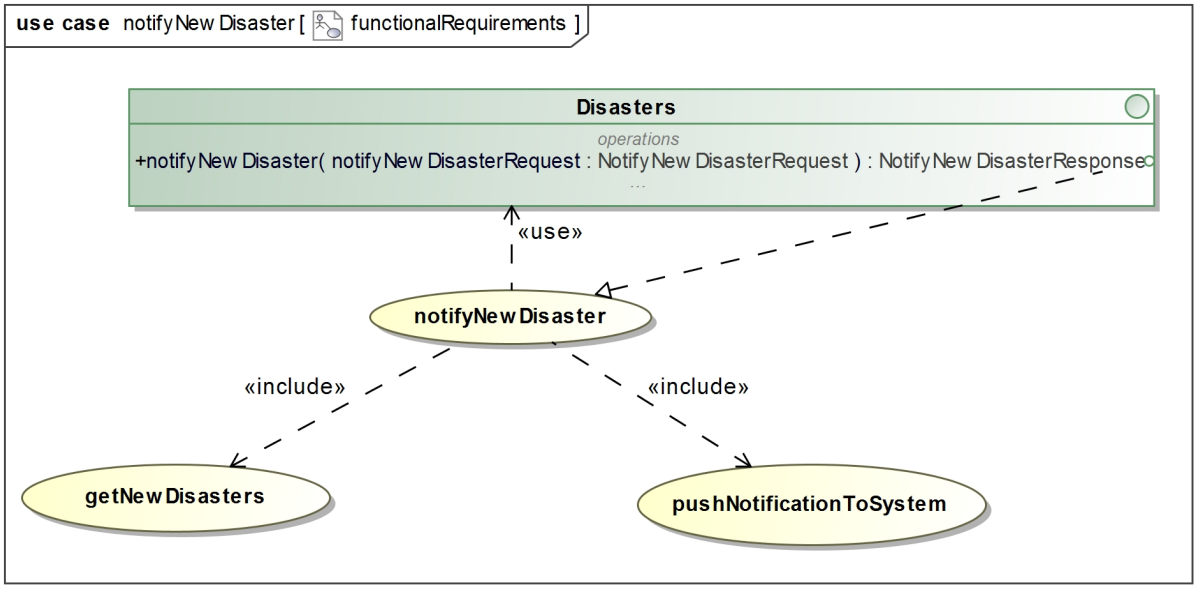

Figure 10: The functional requirements diagram for notifyNewDisaster

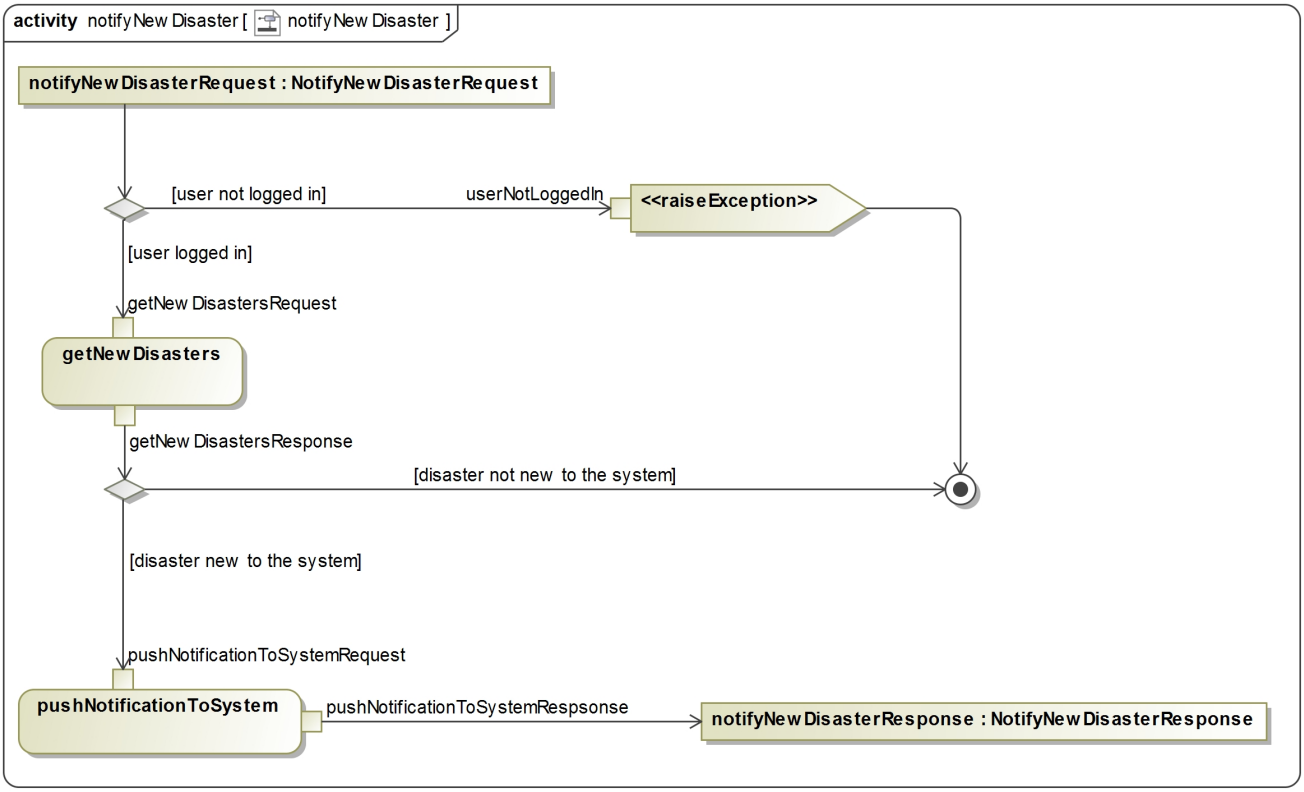

Figure 11: The activity diagram for notifyNewDisaster

5.1.5 trackDisaster

A user, given that they are logged in, is able to track a disaster as long as the disaster is still active and data for the disaster is updated regurlarly. Below are the service contract, activity diagram and functional requirements diagram for trackDisaster.

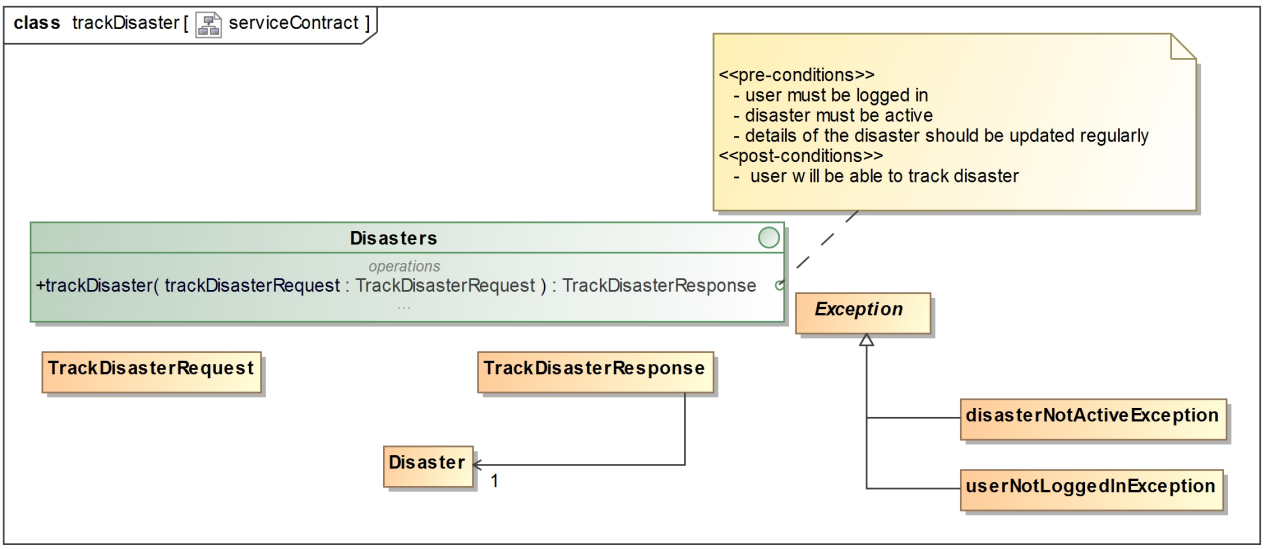

Figure 12: The service contract for trackDisaster

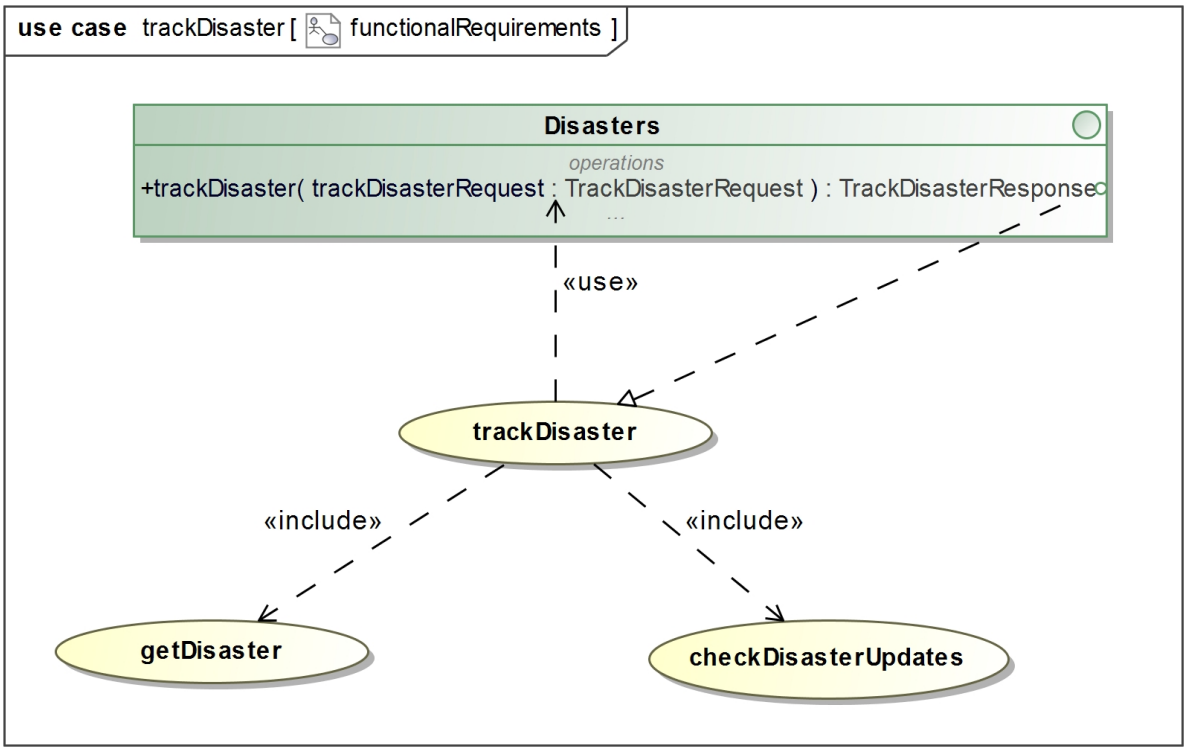

Figure 13: The functional requirements diagram for trackDisaster

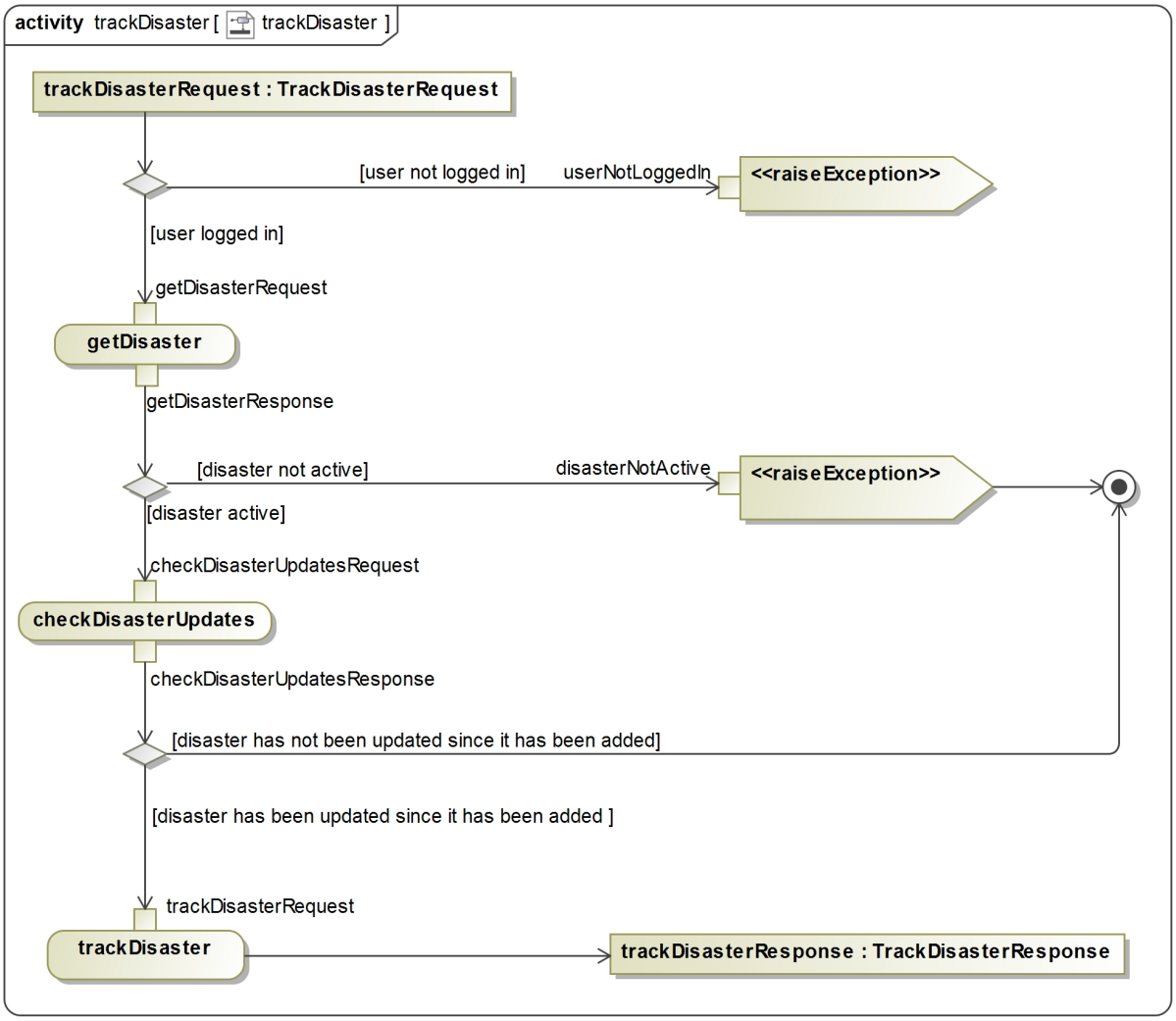

Figure 14: The activity diagram for trackDisaster

5.1.6 addDisasterType

Adding a disaster type requires that the user has admin rights and that there should not be a disaster type of the type being added that exists. If the user is admin and the disaster type being added is unique then a new disaster type will be added to the system. Below are the service contract, activity diagram and functional requirements diagram for addDisasterType.

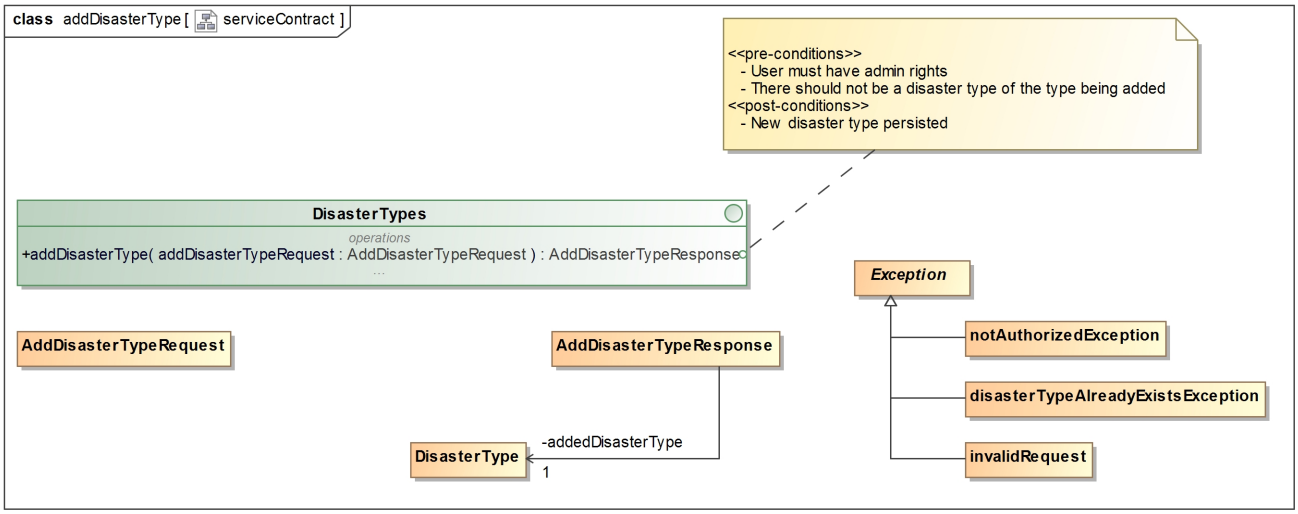

Figure 15: The service contract for addDisasterType

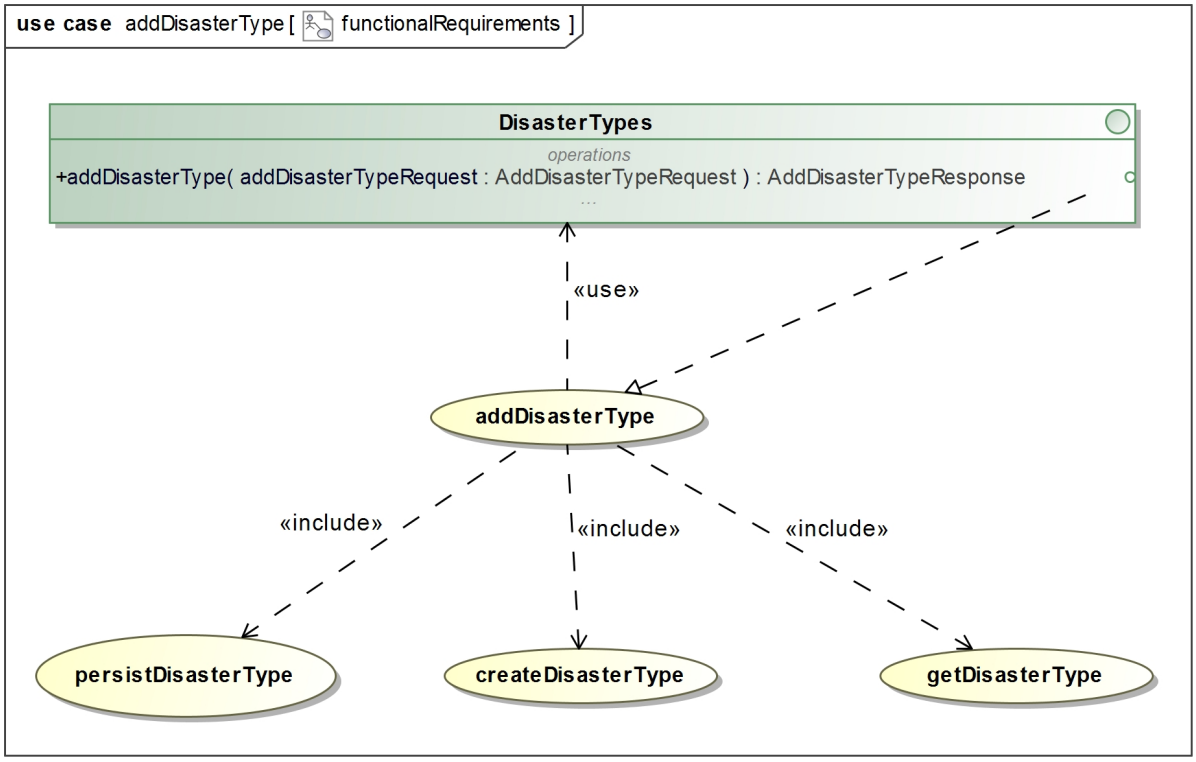

Figure 16: The functional requirements diagram for addDisasterType

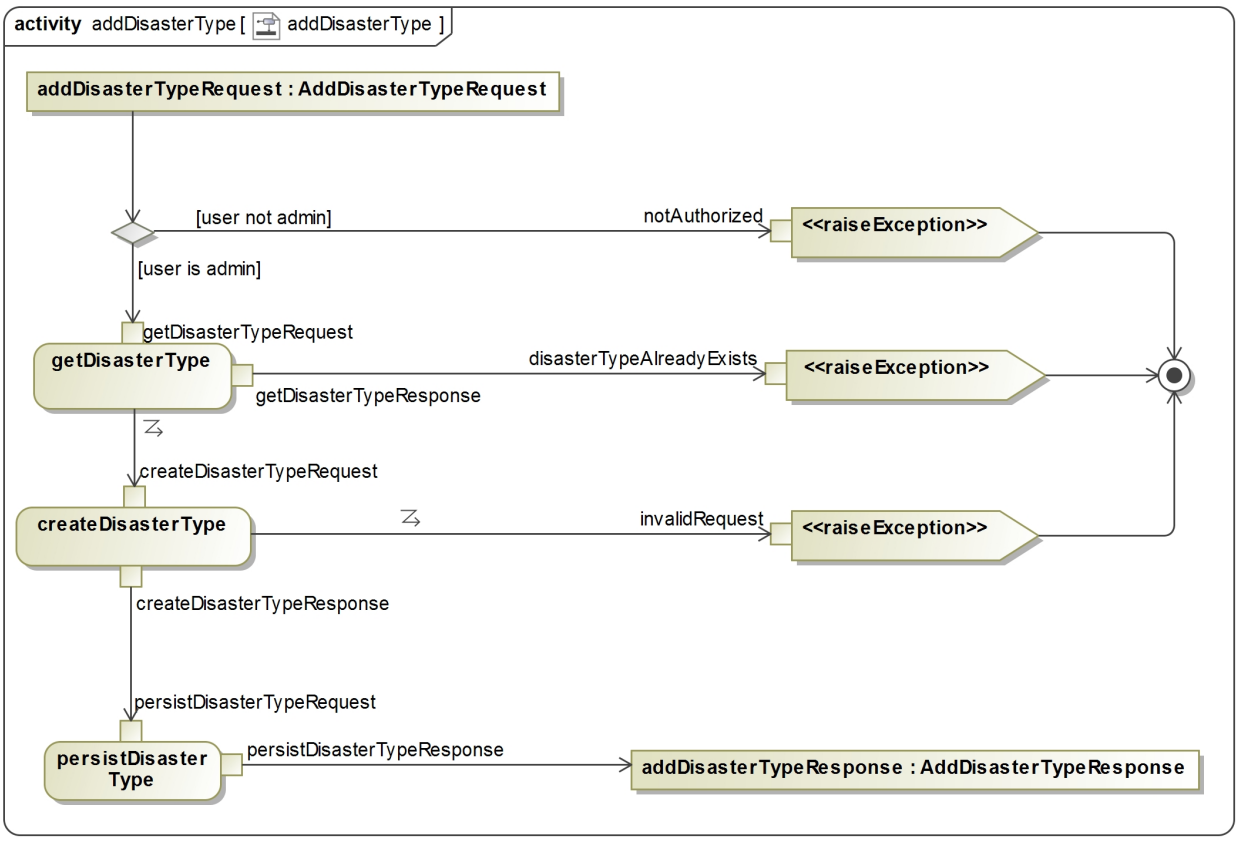

Figure 17: The activity diagram for addDisasterType

5.1.7 modifyDisasterType

Modifying a disaster type requires that the user has admin rights and that there should not be a disaster type of the same type as the modified one. If the user is admin and the modified disaster type is unique then the modified disaster type will be persisted to the database. Below are the service contract, activity diagram and functional requirements diagram for modifiyDisasterType.

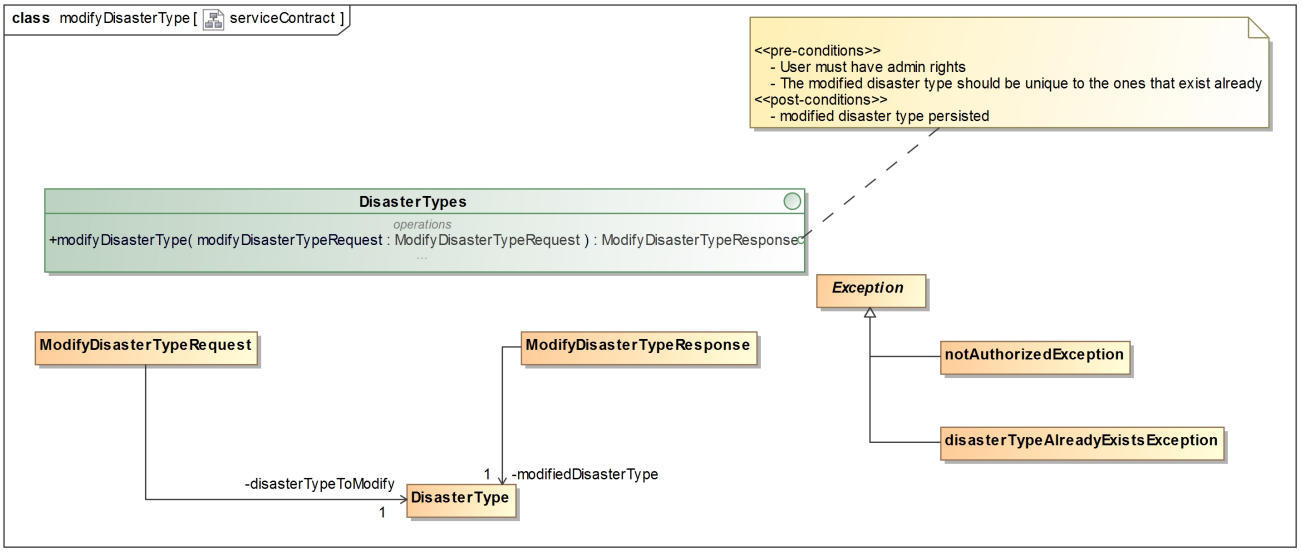

Figure 18: The service contract for modifyDisasterType

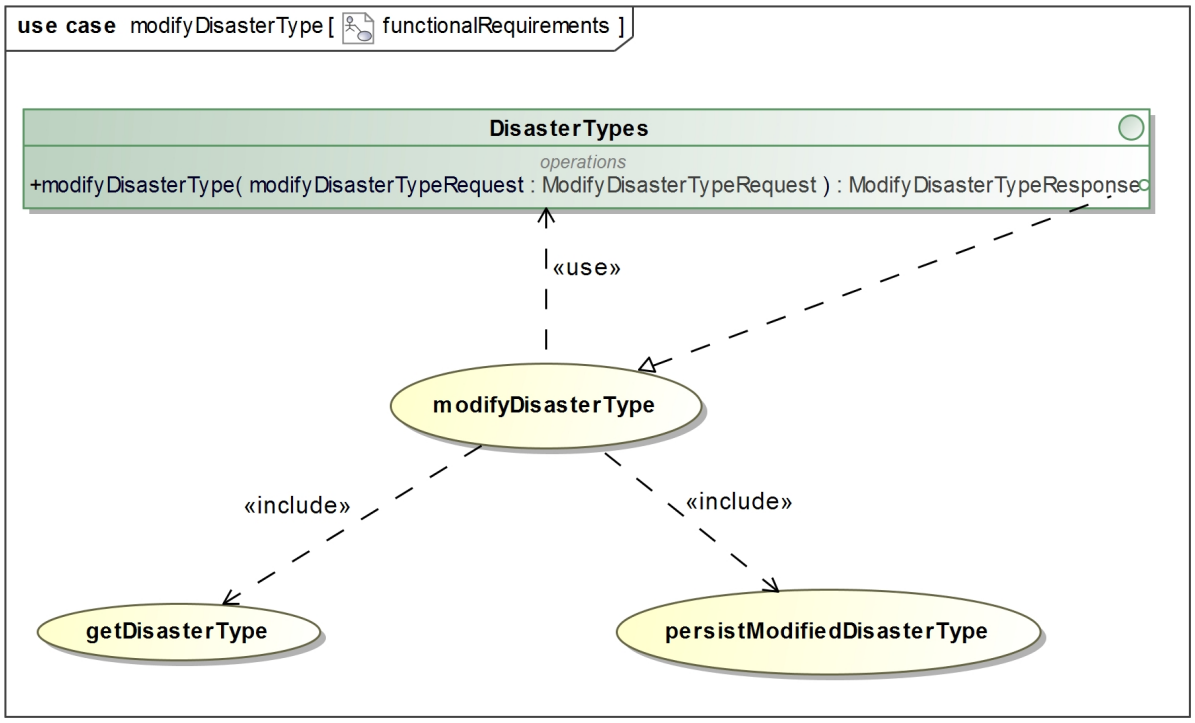

Figure 19: The functional requirements diagram for modifyDisasterType

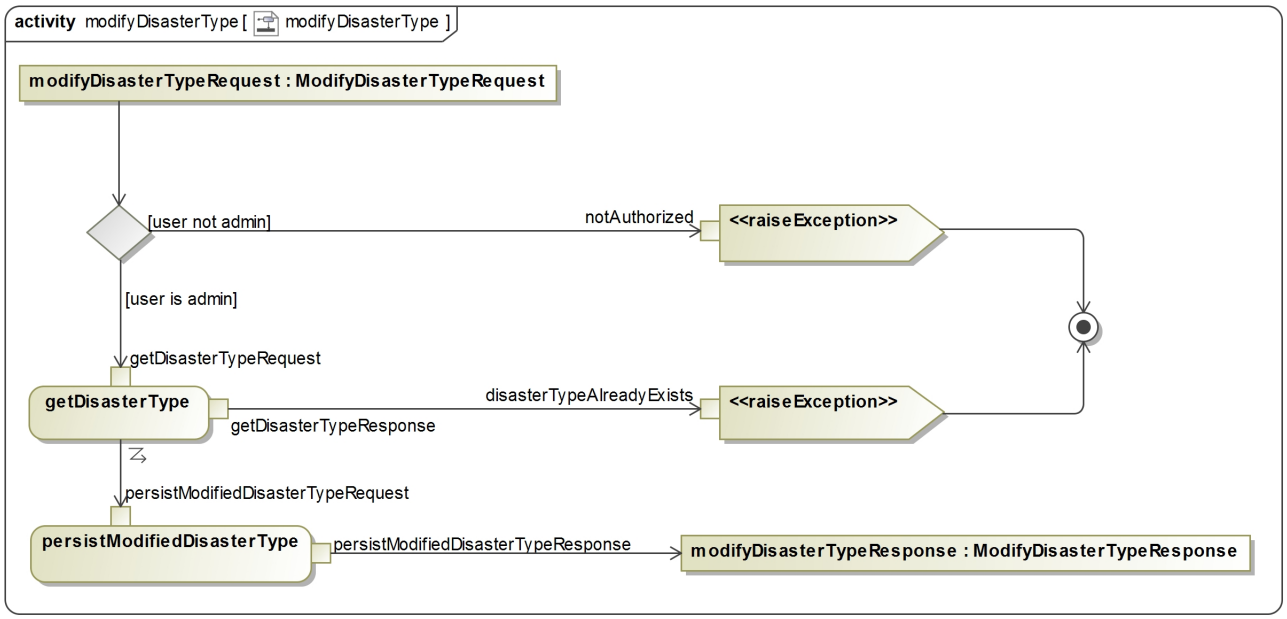

Figure 20: The activity diagram for modifyDisasterType

5.2 Climate Management Module

5.2.1 Scope

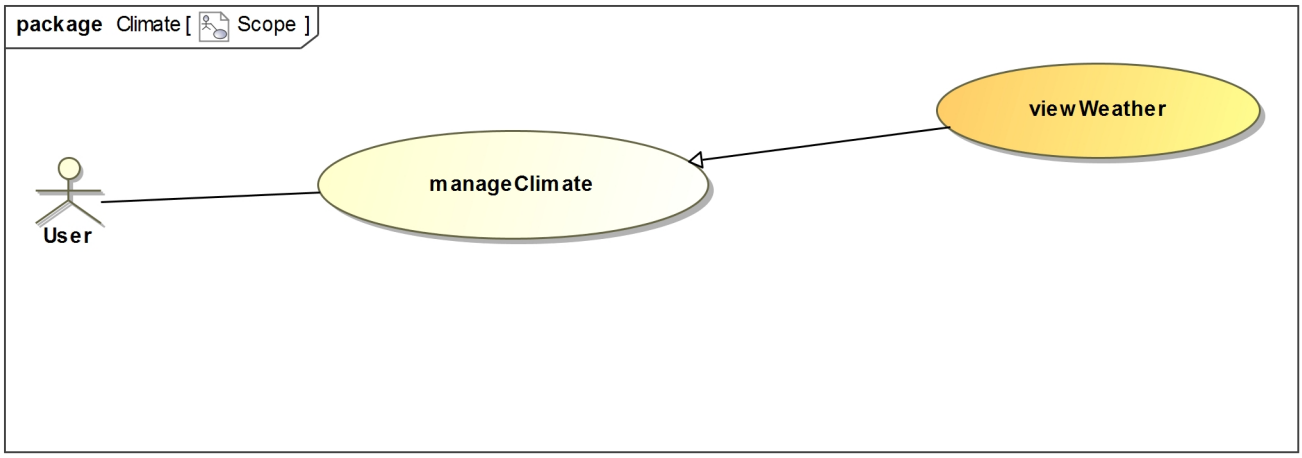

Figure 21: The scope of functionality required from the climate module

Users are able to view weather.

5.2.2 viewWeather

A user can view weather by going on to the weather tab on the system. The user is able to see weather for a specific place by specifying the name of that place in a search bar. Below are the service contract, activity diagram and functional requirements diagram for viewWeather.

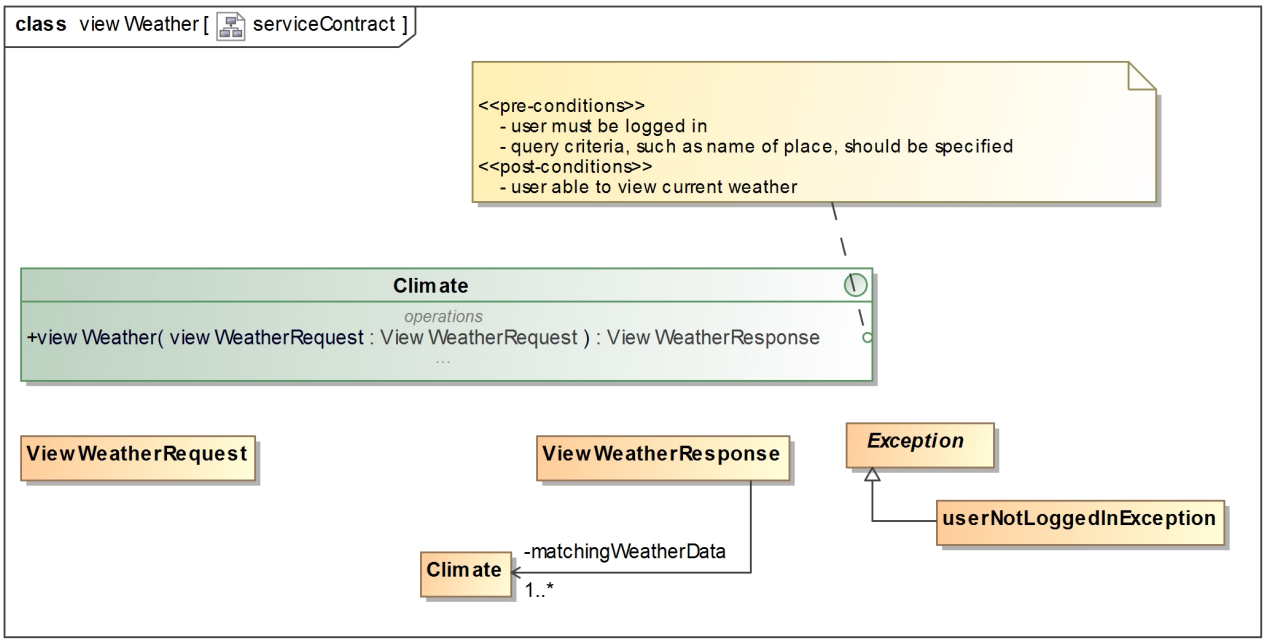

Figure 22: The service contract for viewWeather

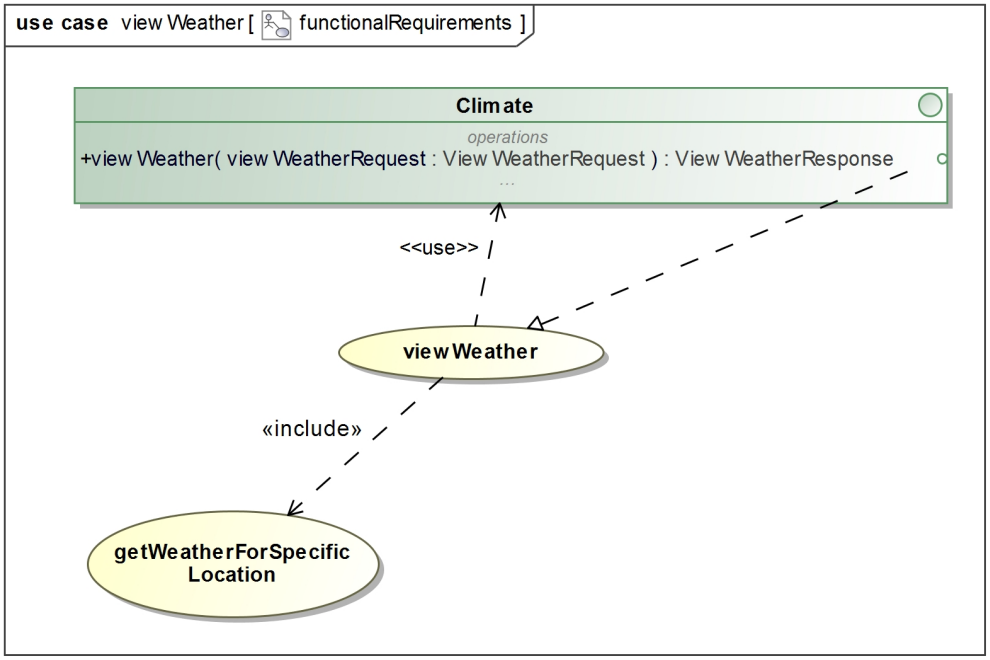

Figure 23: The functional requirements diagram for viewWeather

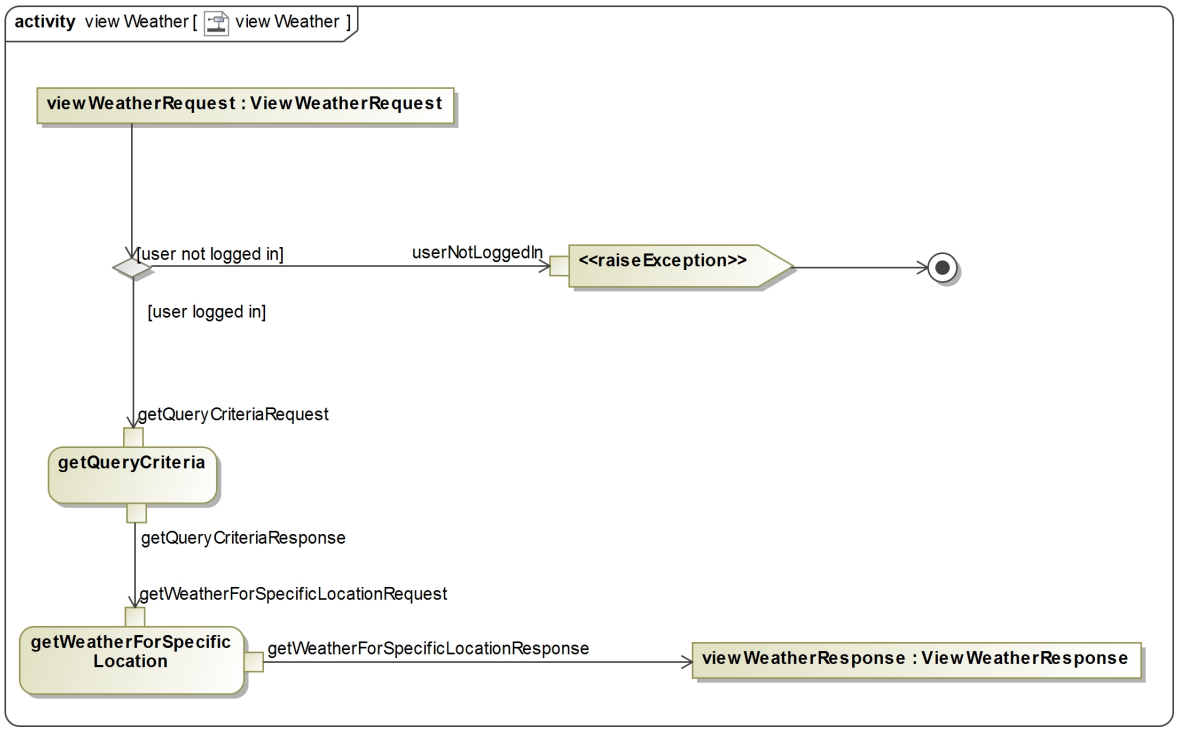

Figure 24: The activity diagram for viewWeather

5.3 User Module

5.3.1 Scope

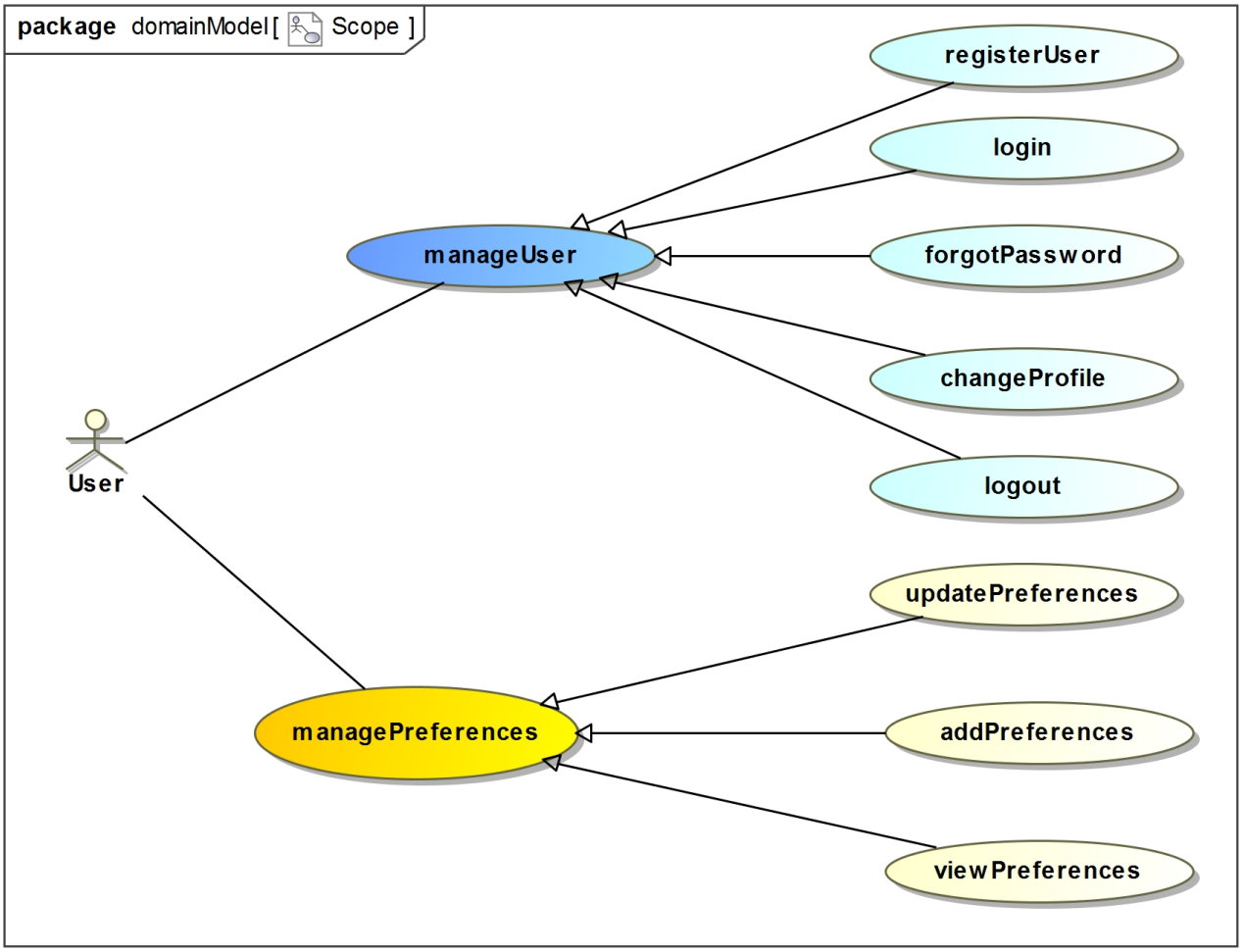

Figure 25: The scope of functionality required from the user module

A user can register and login in to the system to get access to functionality such as changing their profile and adding,viewing and editing preferences of weather or disaster data that they would like to be at their disposal. A user is also able to have their password sent to them if they have forgotten it and also logout of the system once they are done using the system.

5.3.2 registerUser

A user can register with the system as long as they are not already registered. Below are the service contract, activity diagram and functional requirements diagram for registerUser.

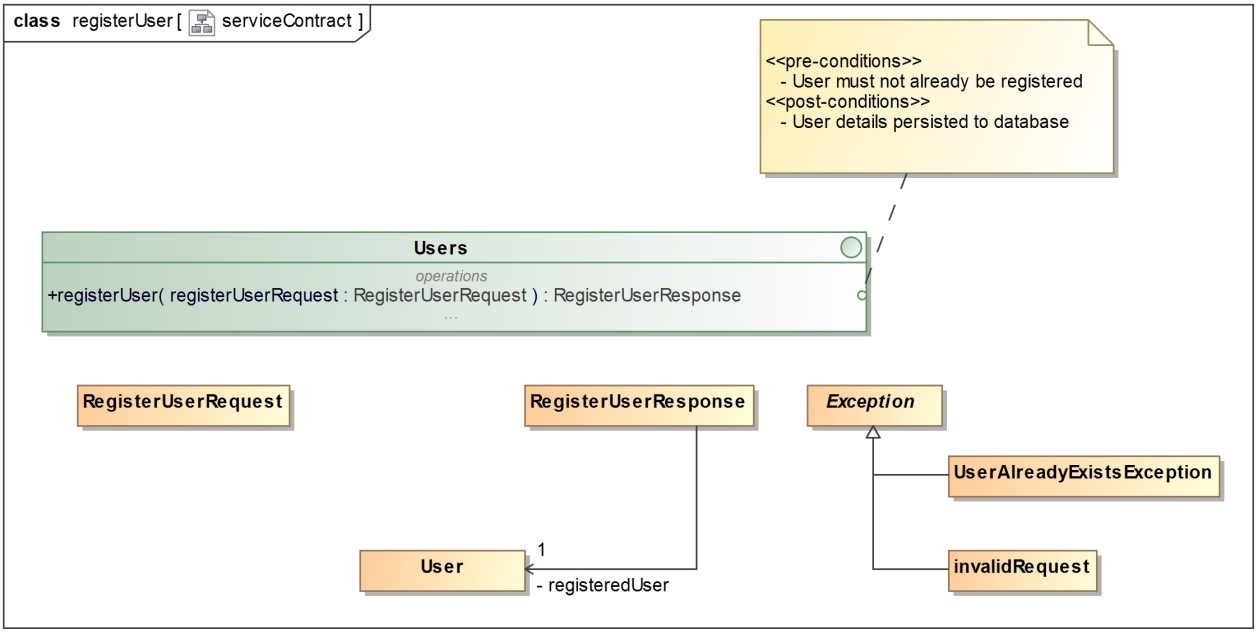

Figure 26: The service contract for registerUser

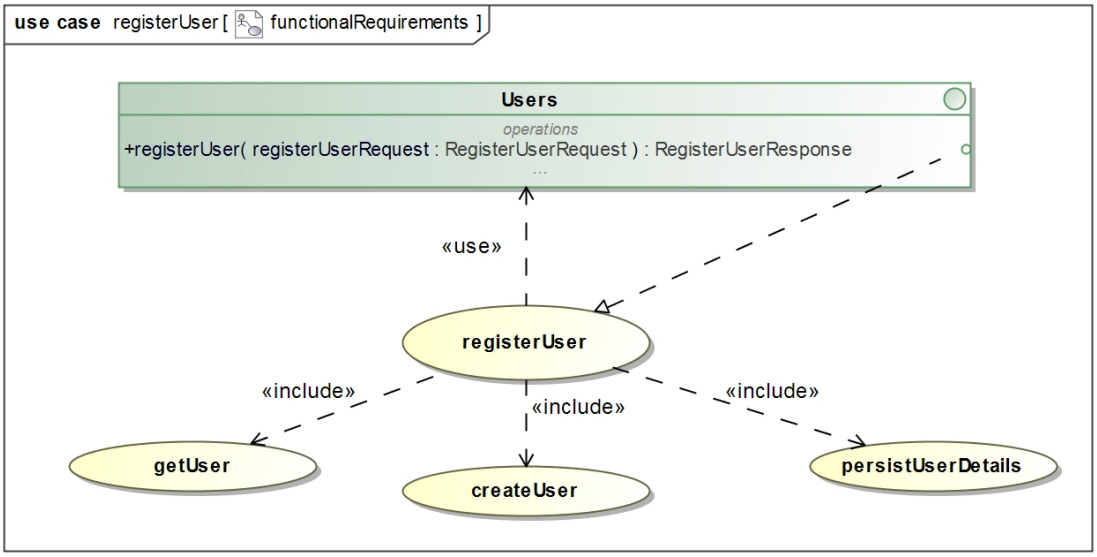

Figure 27: The functional requirements diagram for registerUser

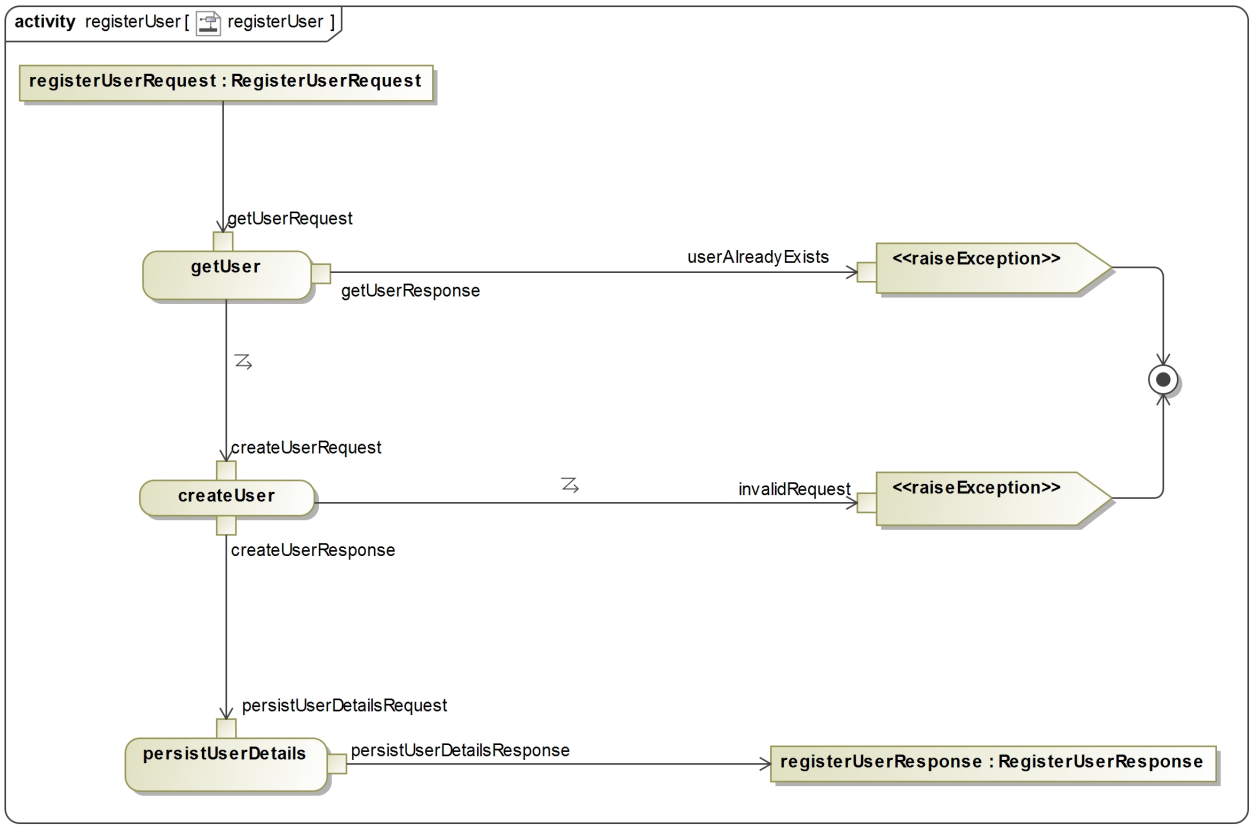

Figure 28: The activity diagram for registerUser

5.3.3 login

A registered user, given that they provide correct details to authenicate them and they are not already logged in, is able to access functionality such as adding and editing weather and disaster preferences once logged in. Below are the service contract, activity diagram and functional requirements diagram for login.

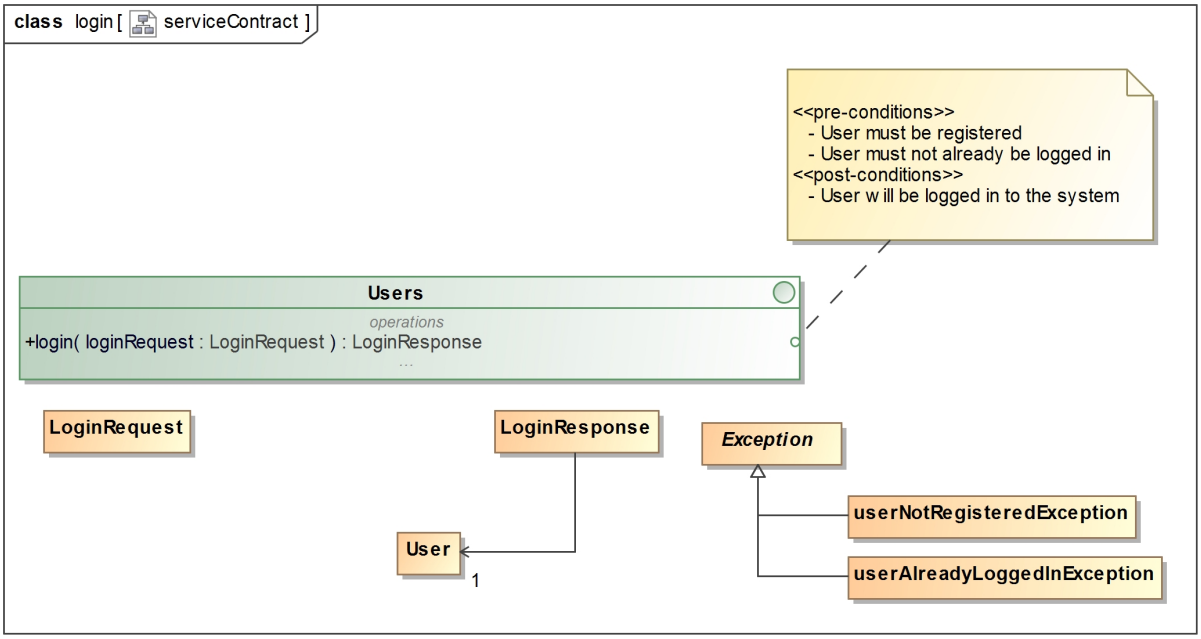

Figure 29: The service contract for login

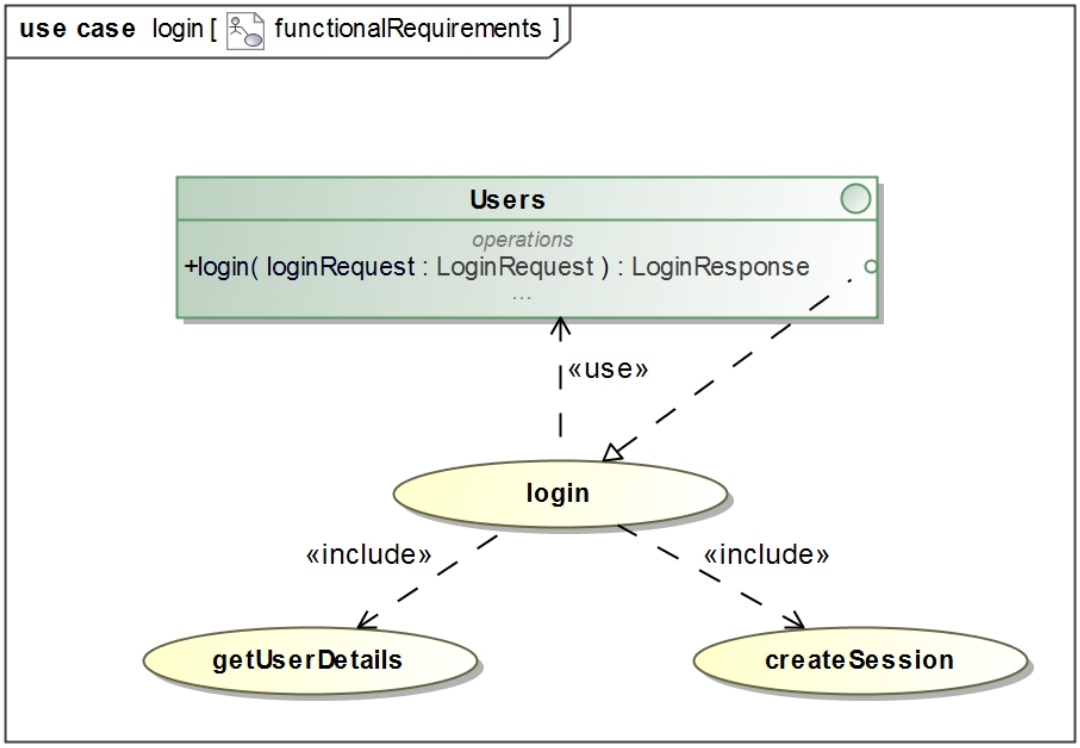

Figure 30: The functional requirements diagram for login

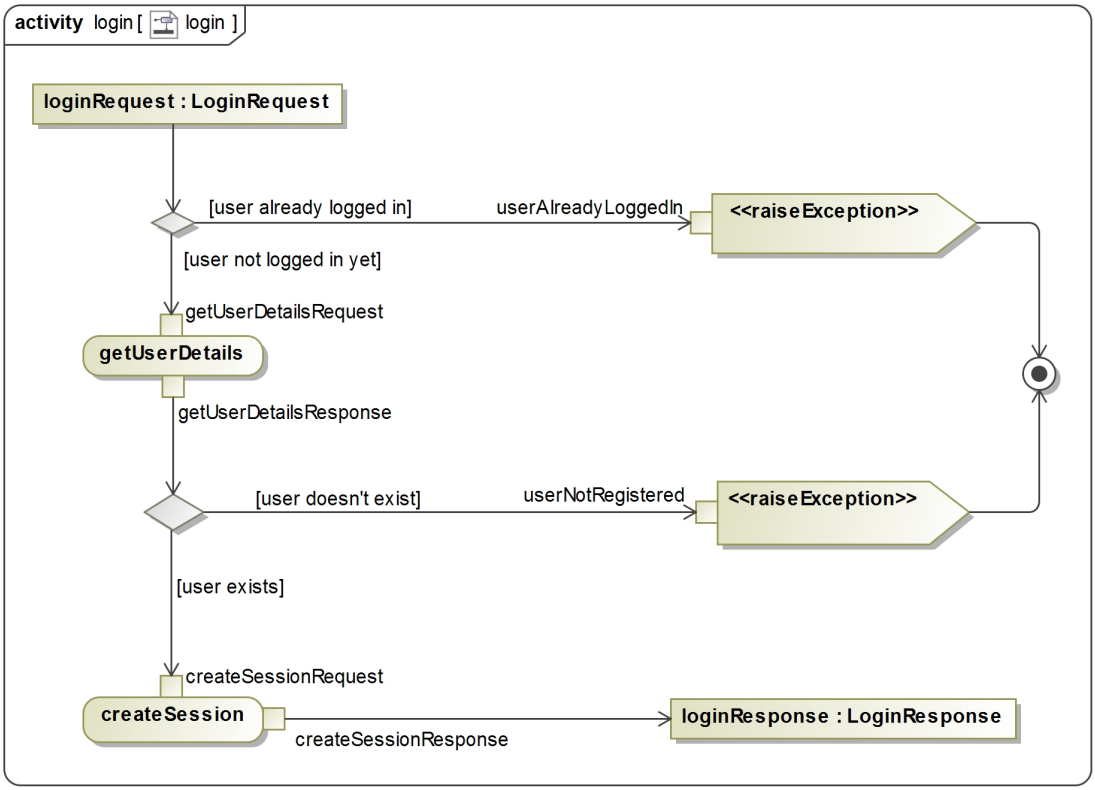

Figure 31: The activity diagram for login

5.3.4 forgotPassword

A user who has forgotten their password is able to have it sent to them via email as long as they provide a working email address that is registered with the system. Below are the service contract, activity diagram and functional requirements diagram for forgotPassword.

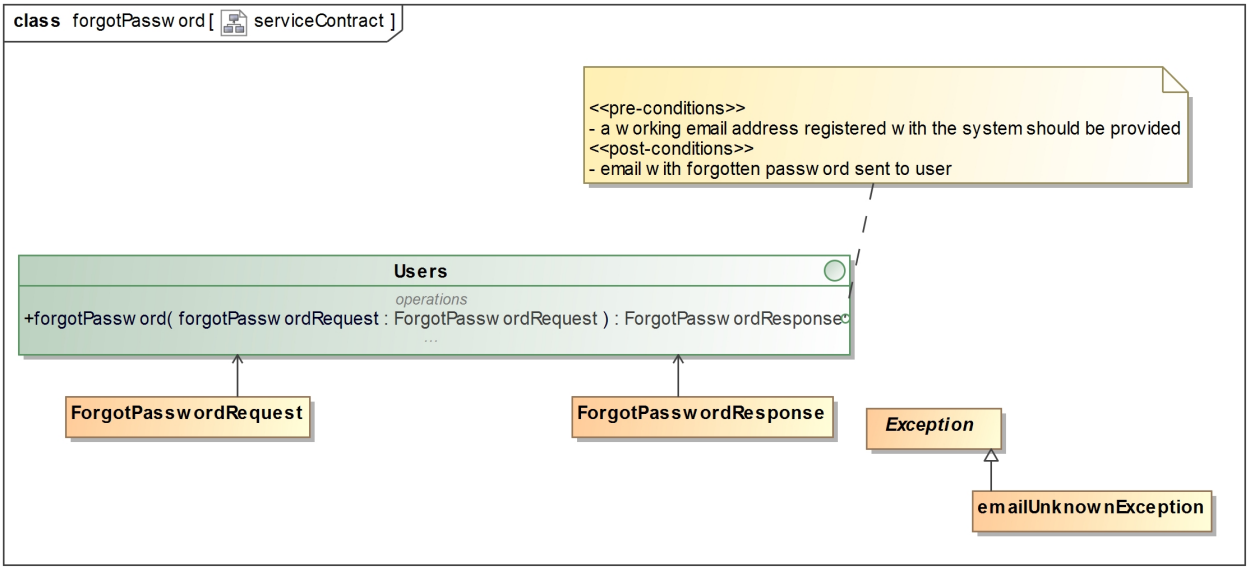

Figure 32: The service contract for forgotPassword

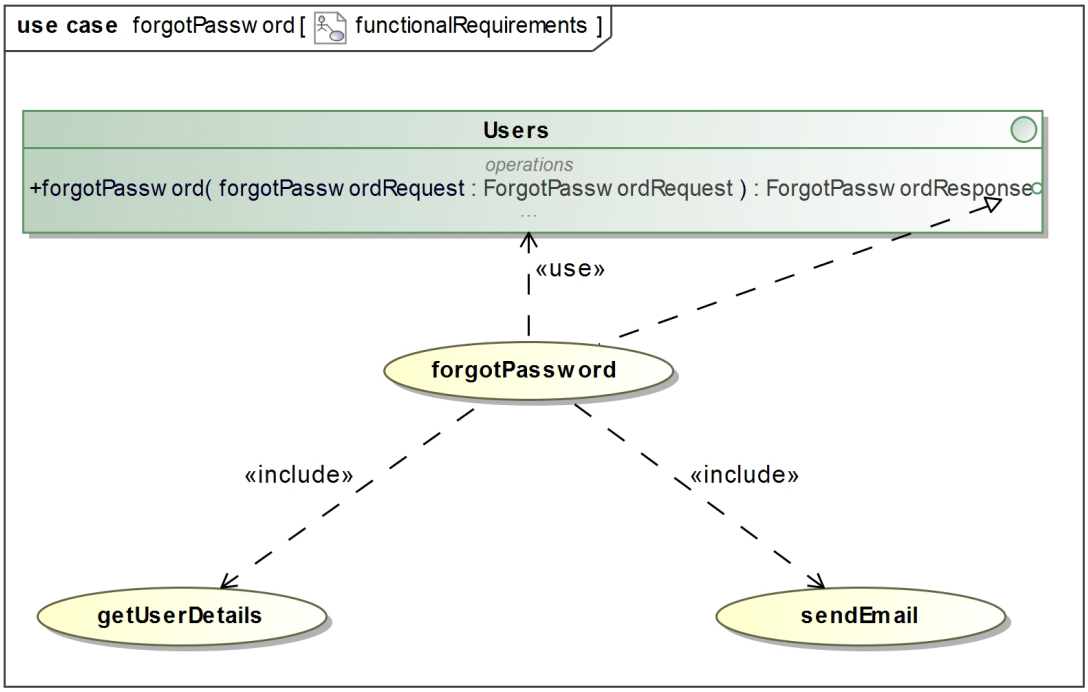

Figure 33: The functional requirements diagram for forgotPassword

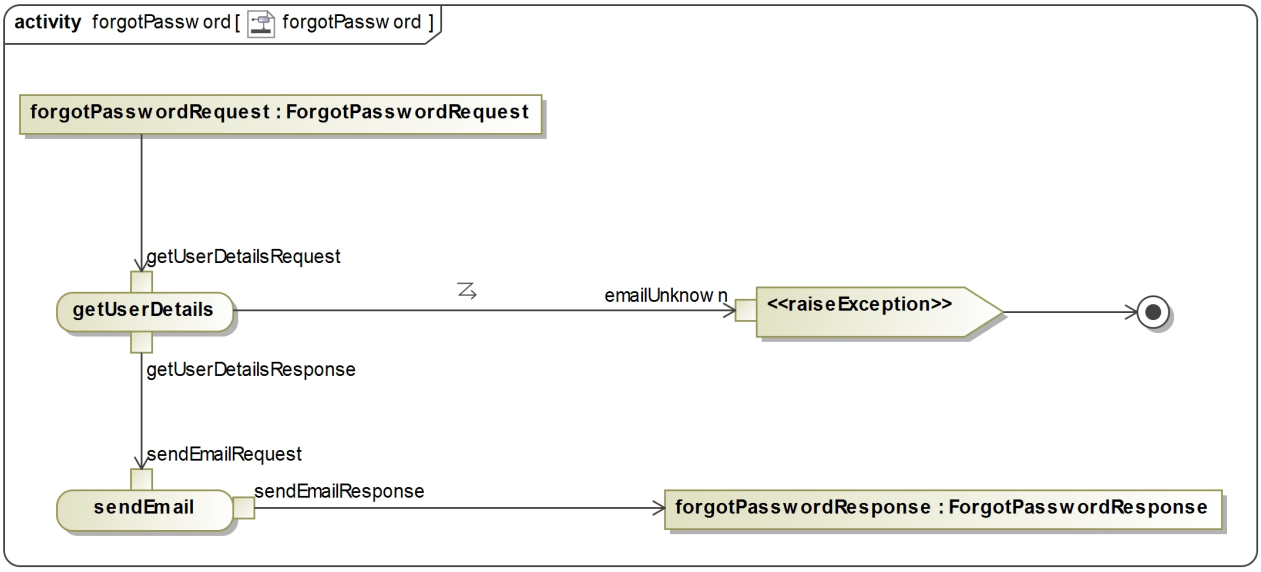

Figure 34: The activity diagram for forgotPassword

5.3.5 changeProfile

A user, given that they are logged in, is able to change their user profile. Below are the service contract, activity diagram and functional requirements diagram for changeProfile.

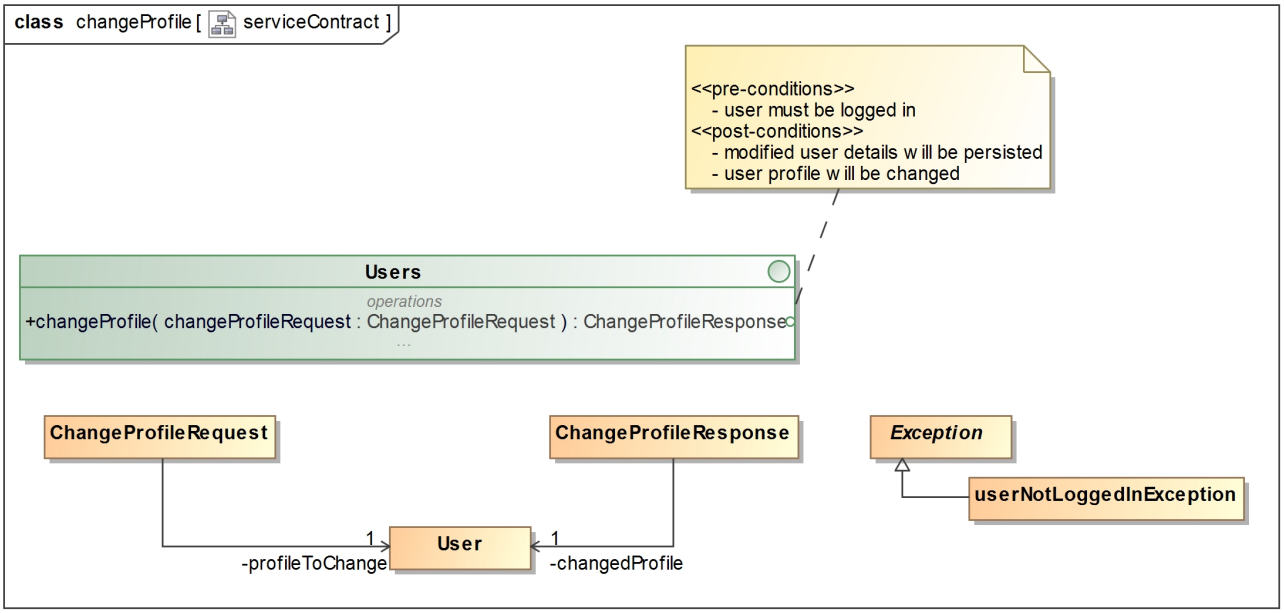

Figure 35: The service contract for changeProfile

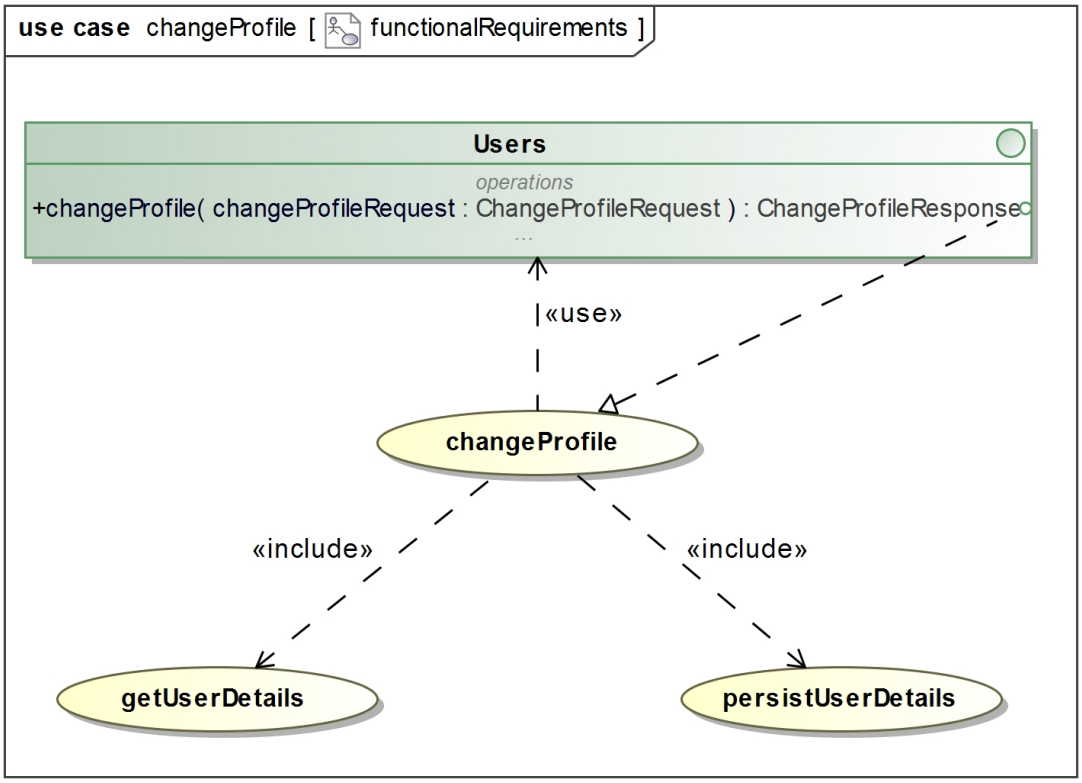

Figure 36: The functional requirements diagram for changeProfile

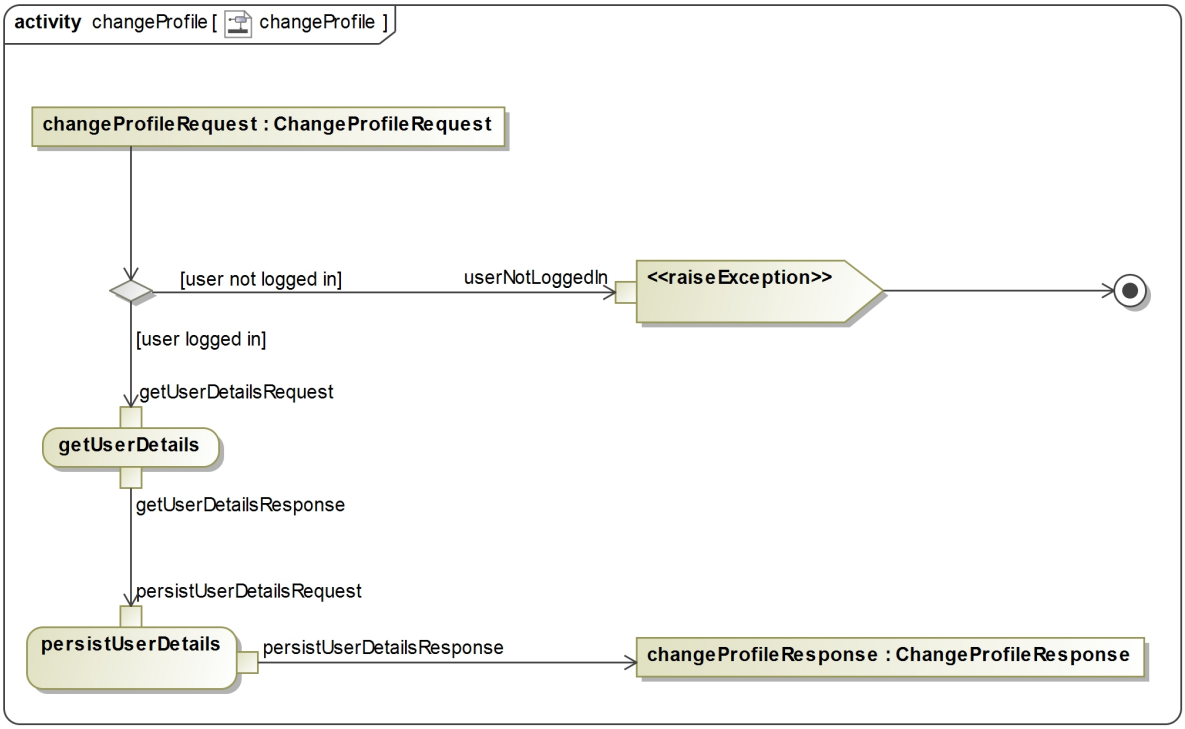

Figure 37: The activity diagram for changeProfile

5.3.6 logout

A user, given that they are logged in to the system, is able to logout of the system once done. Below are the service contract, activity diagram and functional requirements diagram for logout.

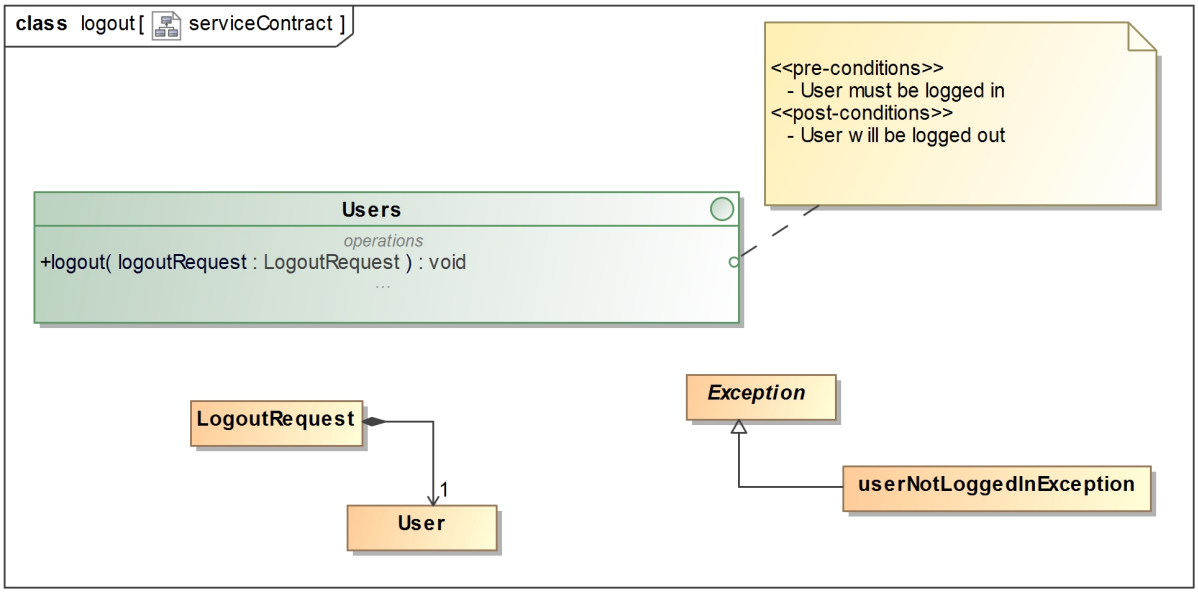

Figure 38: The service contract for logout

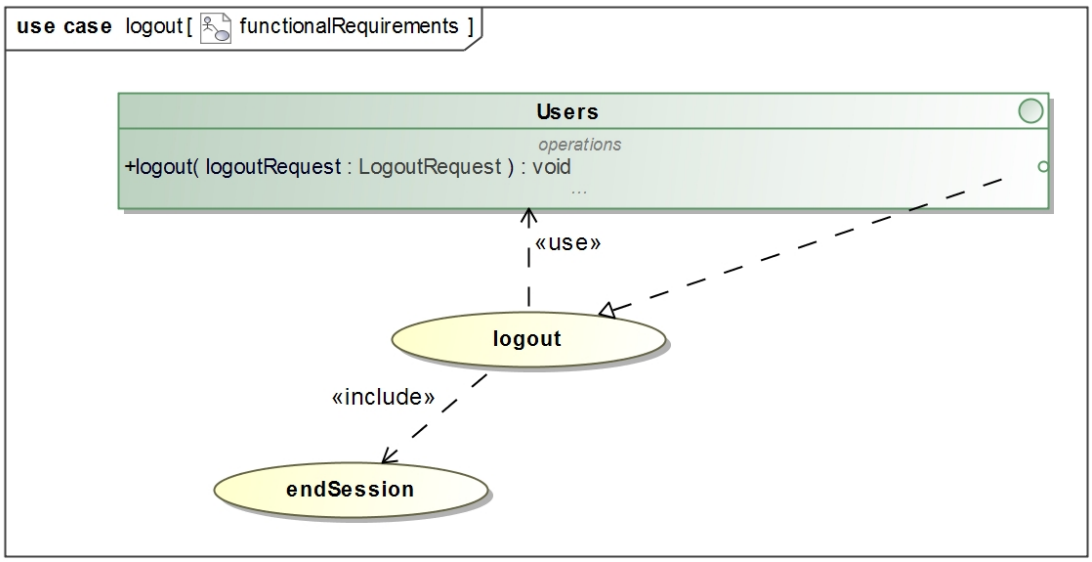

Figure 39: The functional requirements diagram for logout

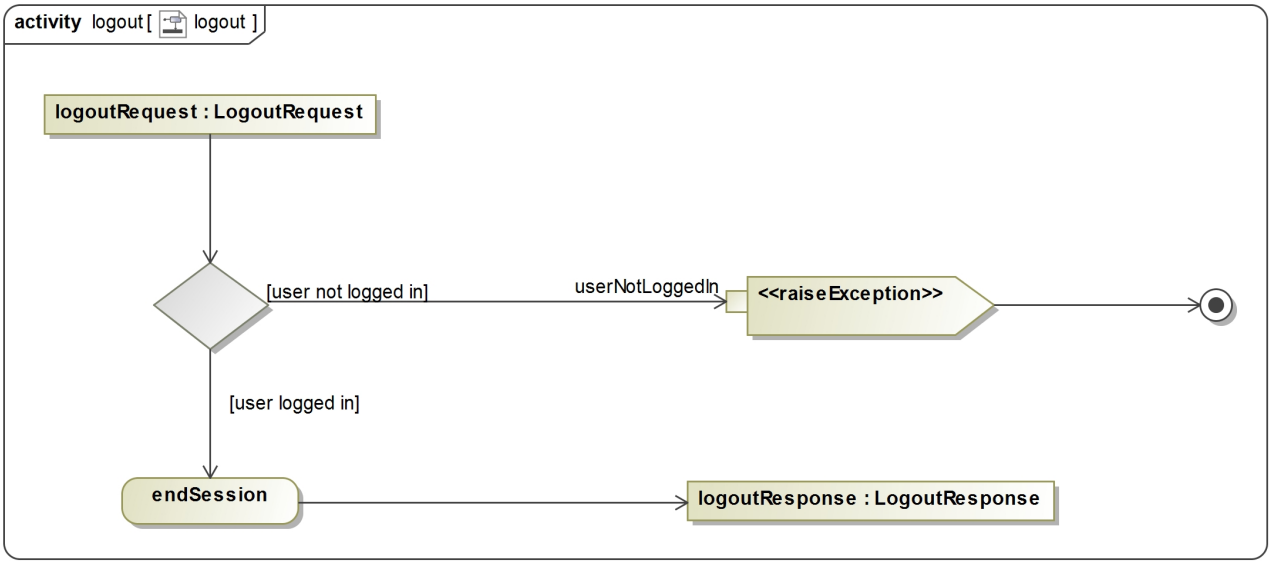

Figure 40: The activity diagram for logout

5.3.7 addPreferences

A user, given that they are logged into the system, is able to add their weather and/or disaster data preferences as long as the user has specified the preference details correctly and those preference details do not already exist for the user. Below are the service contract, activity diagram and functional requirements diagram for addPreferences.

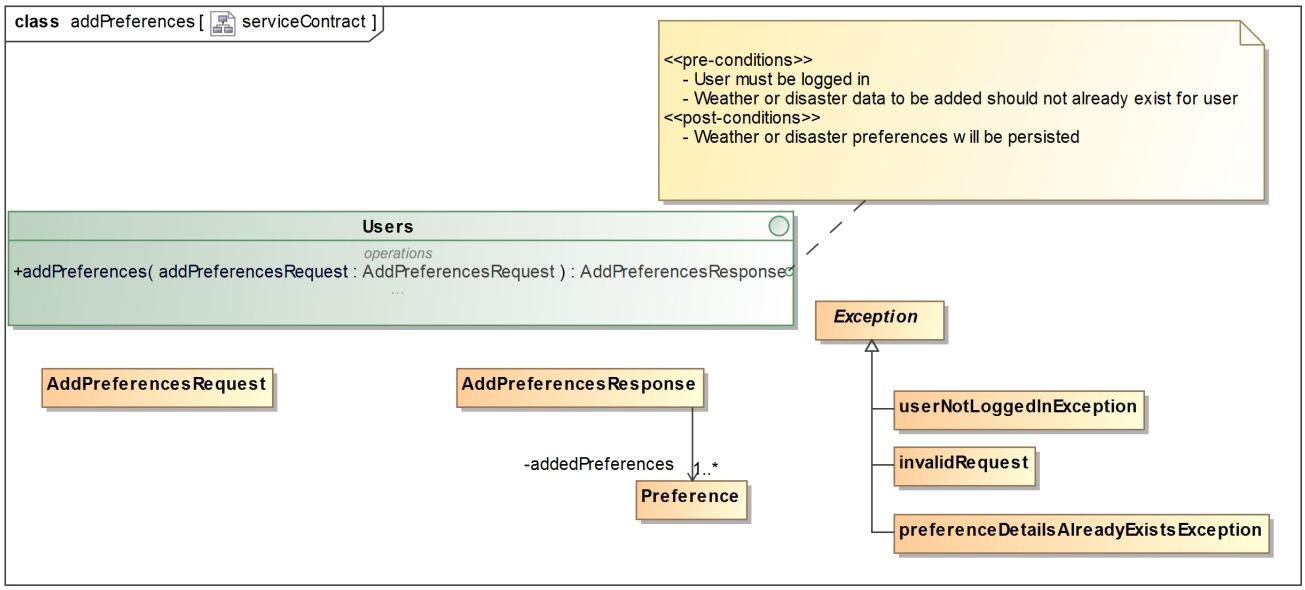

Figure 41: The service contract for addPreferences

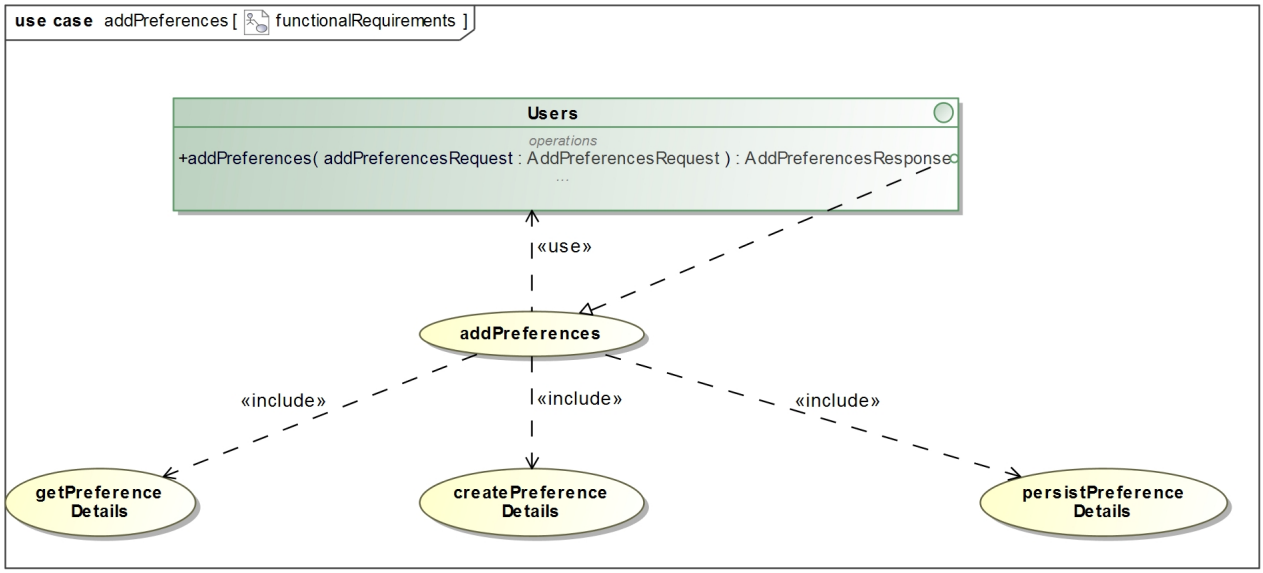

Figure 42: The functional requirements diagram for addPreferences

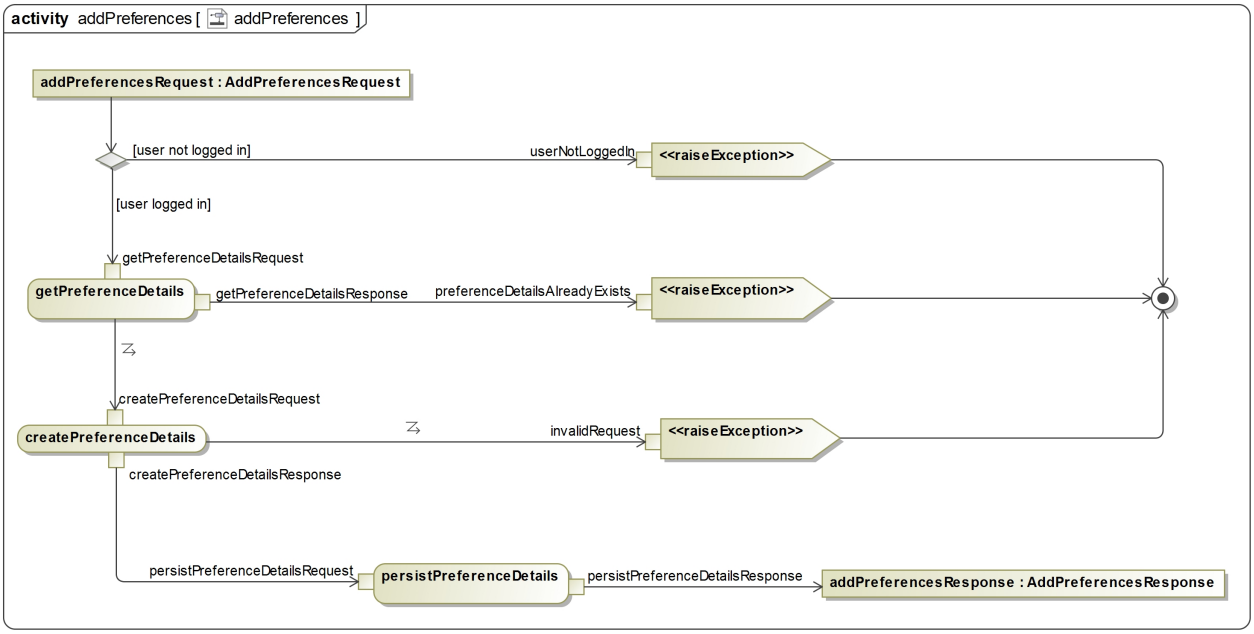

Figure 43: The activity diagram for addPreferences

5.3.8 editPreferences

A user, given that they are logged into the system, is able to edit any of their weather and/or disaster data preferences as long as the preference details to update exist and that the preference details when updated are not like any of the ones that already exist for the user. Below are the service contract, activity diagram and functional requirements diagram for editPreferences.

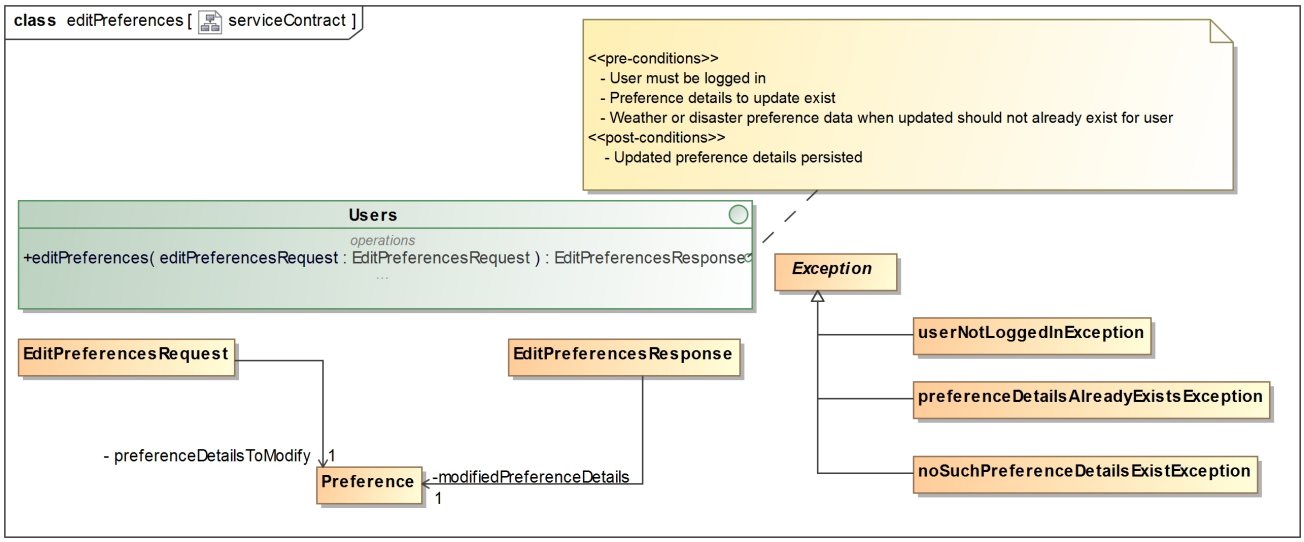

Figure 44: The service contract for editPreferences

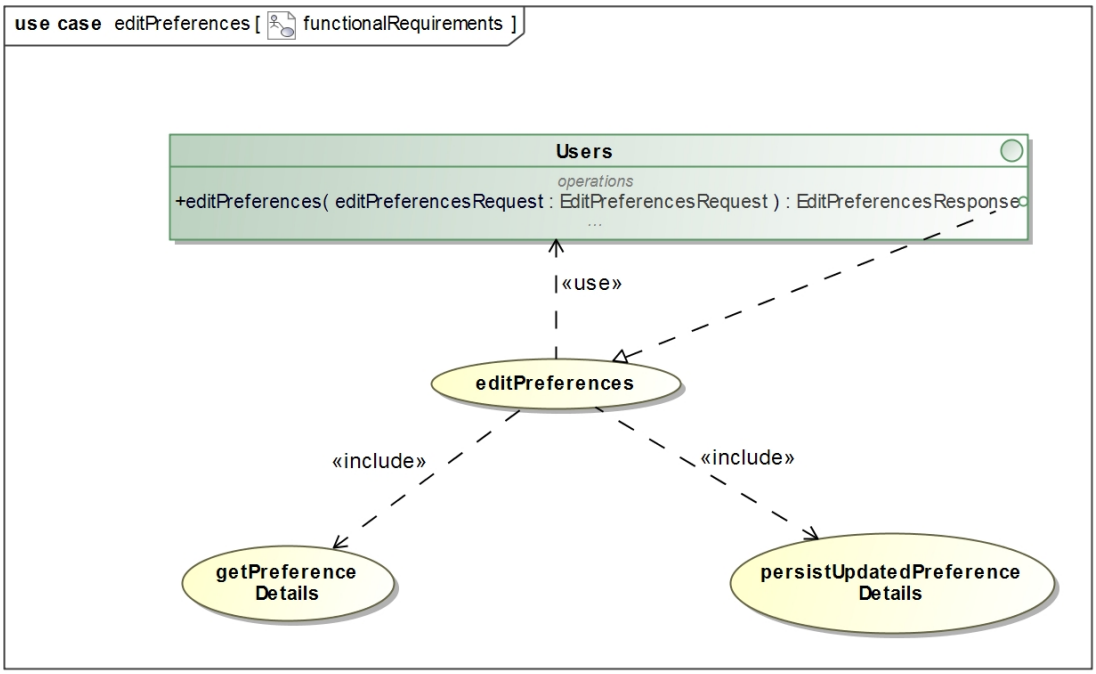

Figure 45: The functional requirements diagram for editPreferences

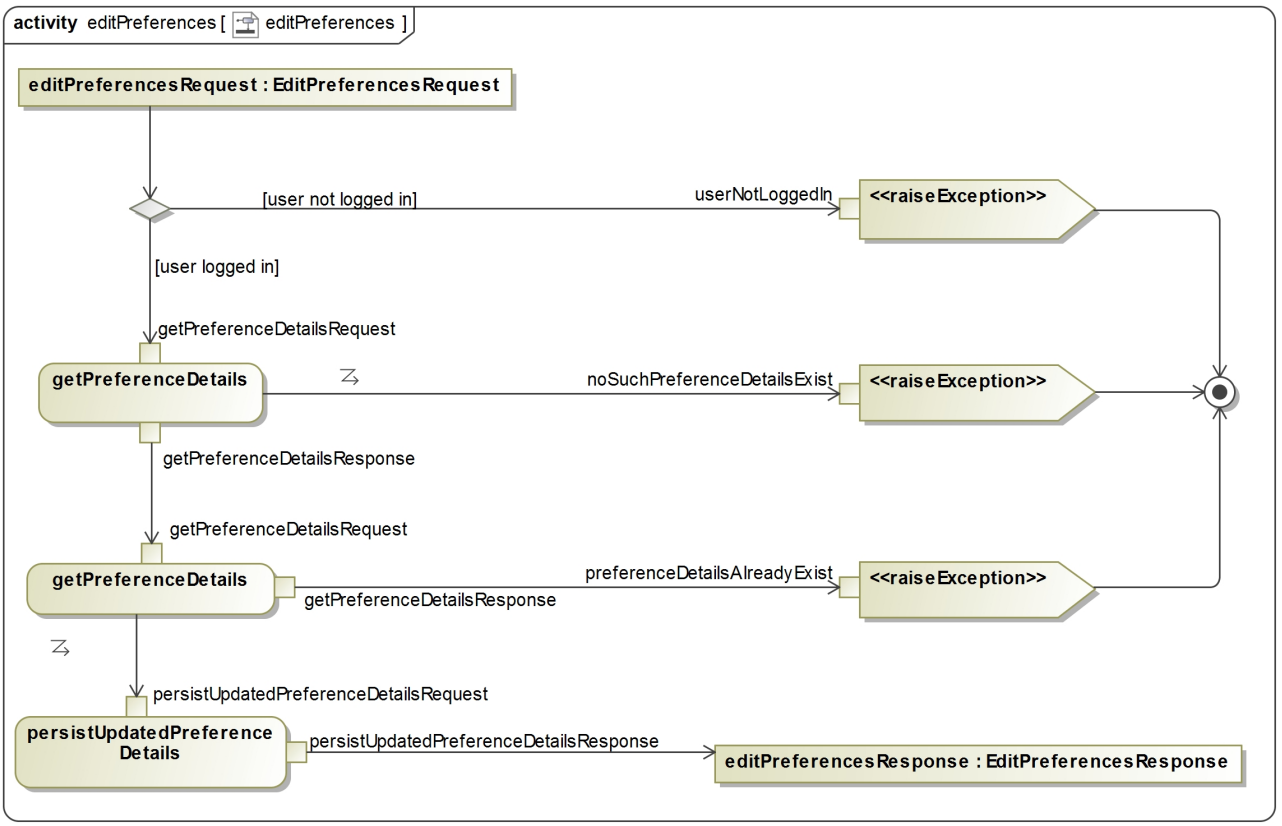

Figure 46: The activity diagram for editPreferences

5.3.9 viewPreferences

A user, given that they are logged into the system, is able to view any of their weather and/or disaster preferences as long as they have already added those preferences. Below are the service contract, activity diagram and functional requirements diagram for viewPreferences.

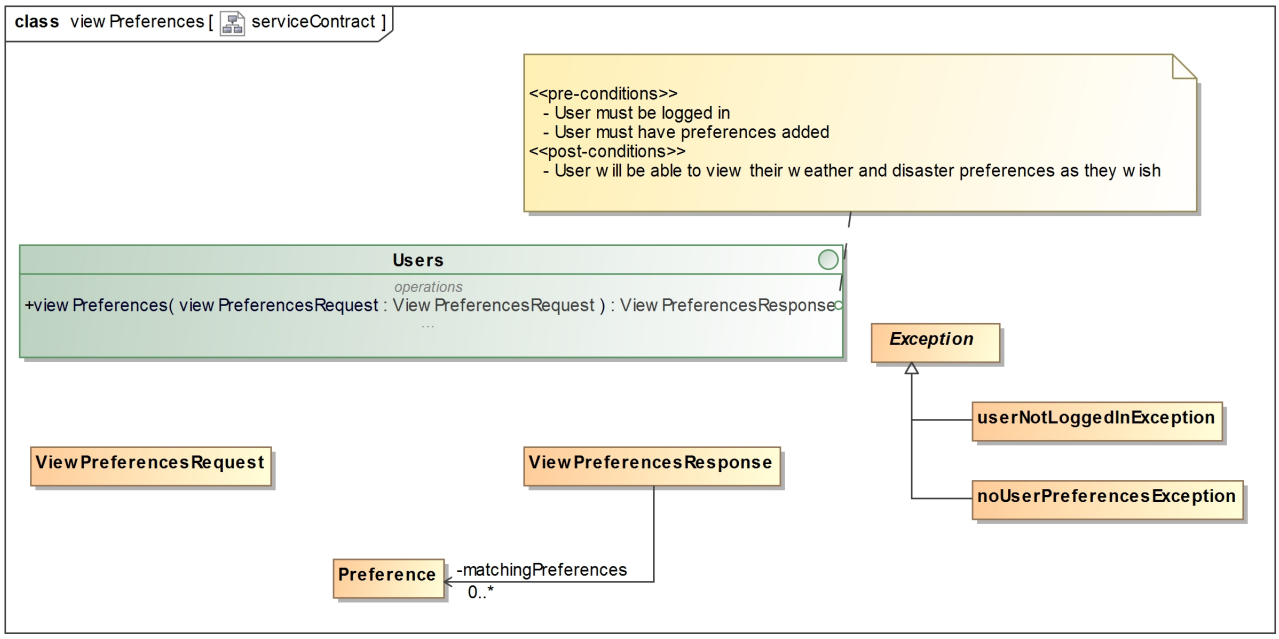

Figure 47: The service contract for viewPreferences

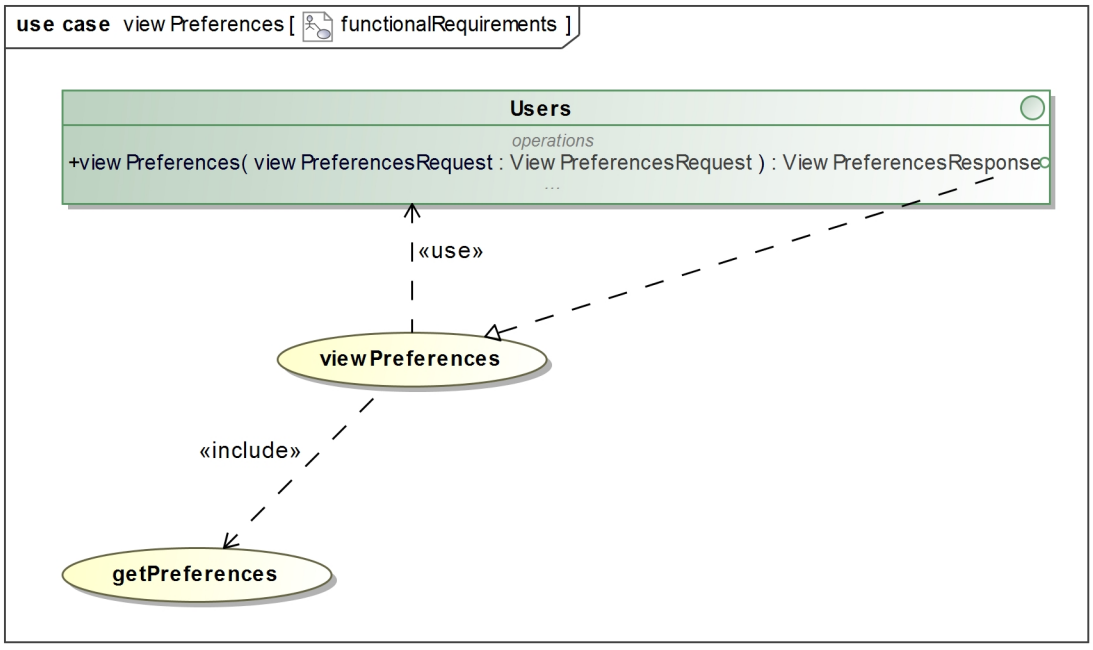

Figure 48: The functional requirements diagram for viewPreferences

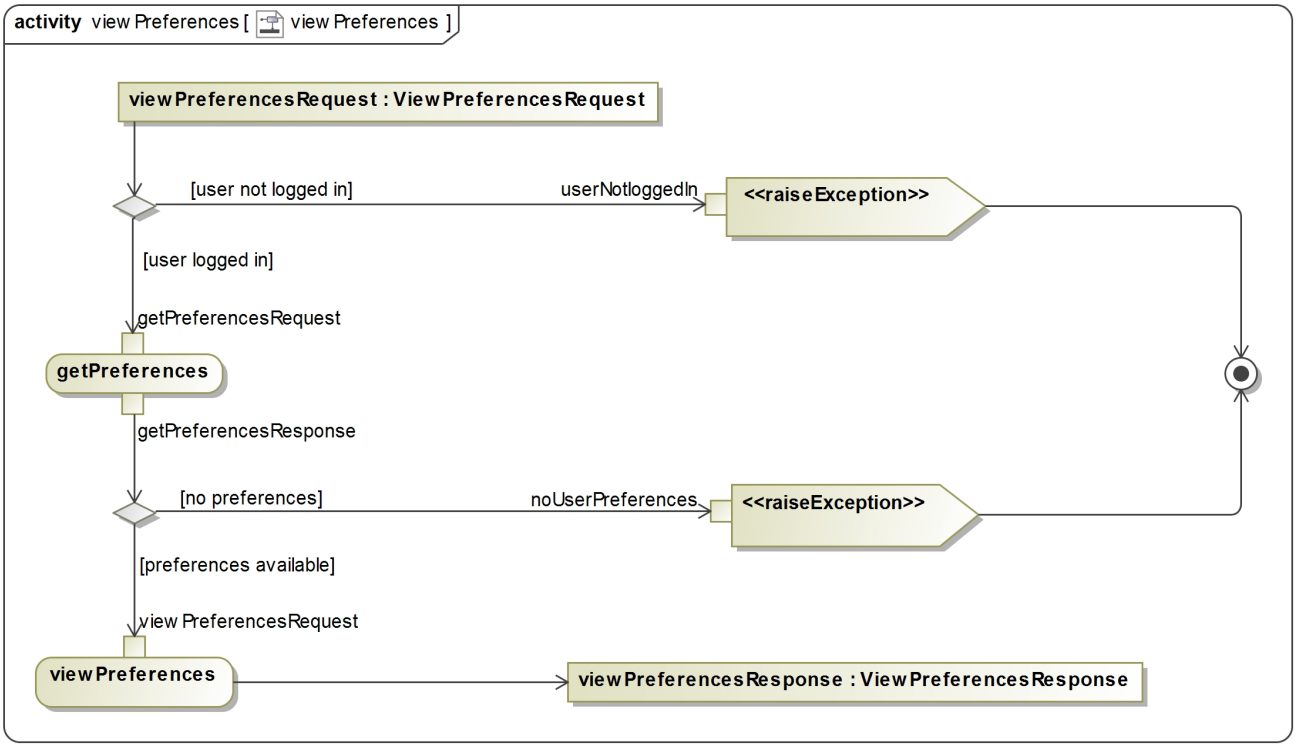

Figure 49: The activity diagram for viewPreferences

5.4 Domain Model

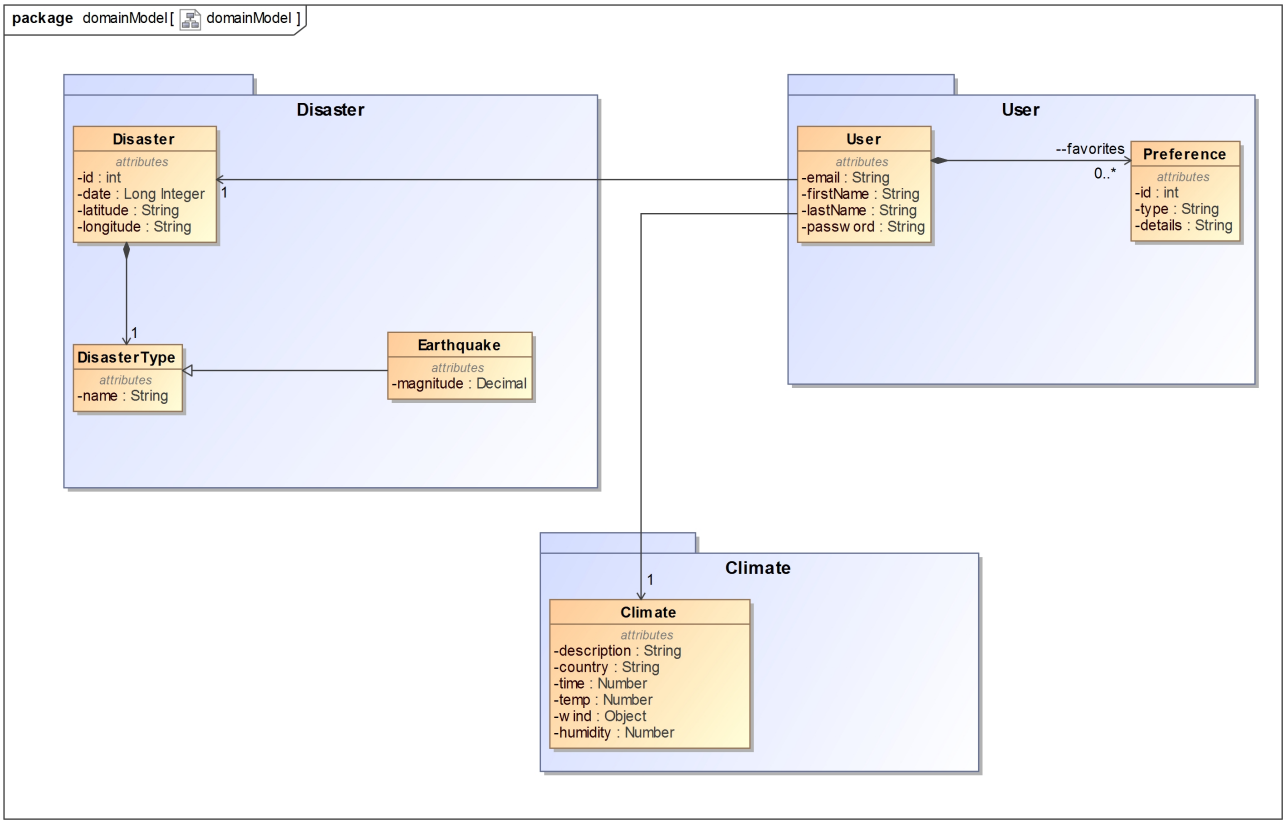

Figure 50: The domain model for the climate, disaster and user modules

Each user has a first name, last name, password and an email address. A user can also have 0 or more favourites that they can save by specifying the type of preference (disaster or weather) as well as other required details identifying the type of preference. A user, who is a client, is able to request services from the disaster and climate system components. A disaster which has an id, latitude, longitude, date and extra details also has a type. A disaster type could either be a fire, earthquake, flood or etc but at current the system has earthquake as the only type. Climate, on the other hand has a description, and values describing the country, time, temperature, wind and humitidty of a specific climate.

6 Open Issues

There are no issues

List of Figures

1 Overview of the System 4
2 The scope of functionality required from the disaster module 6
3 The service contract for viewDisaster 7
4 The functional requirements diagram for viewDisaster 8
5 The activity diagram for viewDisaster 9
6 The service contract for queryDisaster 10
7 The functional requirements diagram for queryDisaster 11
8 The activity diagram for queryDisaster 12
9 The service contract for notifyNewDisaster 13
10 The functional requirements diagram for notifyNewDisaster . 13
11 The activity diagram for notifyNewDisaster 14
12 The service contract for trackDisaster 15
13 The functional requirements diagram for trackDisaster 16
14 The activity diagram for trackDisaster 17
15 The service contract for addDisasterType 18
16 The functional requirements diagram for addDisasterType . . 18
17 The activity diagram for addDisasterType 19
18 The service contract for modifyDisasterType 20
19 The functional requirements diagram for modifyDisasterType 20
20 The activity diagram for modifyDisasterType 21
21 The scope of functionality required from the climate module . 21
22 The service contract for viewWeather 22
23 The functional requirements diagram for viewWeather 23
24 The activity diagram for viewWeather 24
25 The scope of functionality required from the user module . . 25
26 The service contract for registerUser 26
27 The functional requirements diagram for registerUser 27
28 The activity diagram for registerUser 28
29 The service contract for login 29
30 The functional requirements diagram for login 30
31 The activity diagram for login 31
32 The service contract for forgotPassword 32
33 The functional requirements diagram for forgotPassword . . . 32
34 The activity diagram for forgotPassword 33

35 The service contract for changeProfile 34
36 The functional requirements diagram for changeProfile 35
37 The activity diagram for changeProfile 36
38 The service contract for logout 37
39 The functional requirements diagram for logout 37
40 The activity diagram for logout 38
41 The service contract for addPreferences 39
42 The functional requirements diagram for addPreferences . . . 39
43 The activity diagram for addPreferences 40
44 The service contract for editPreferences 41
45 The functional requirements diagram for editPreferences . . . 41
46 The activity diagram for editPreferences 42
47 The service contract for viewPreferences 43
48 The functional requirements diagram for viewPreferences . . . 43
49 The activity diagram for viewPreferences 44
50 The domain model for the climate, disaster and user modules 45