



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

REQUIREMENTS AND DESIGN SPECIFICATION

Mobile Monitoring App

Emilio Mumba

The 5 Concurrent Nodes

Khathutshelo Shaun Matidza
Sylvester Sandile Mpangane
Thabang Michael Letageng
Aluwani Augustine Simetsi
Matthew Nel

Department of Computer Science, University of Pretoria

Aug 2015

Contents

1	Vision and Scope	3
1.1	Project Vision	3
1.2	Project Scope	4
2	Application Requirements and Design	4
2.1	Modular System	4
2.2	MobileMonitoringApp - Accounts module	5
2.2.1	Scope	5
2.2.2	Use-Cases	5
2.2.3	Domain Model	9
2.2.4	Design Pattern	10
2.3	MobileMonitoringApp - Data module	11
2.3.1	Scope	11
2.3.2	Use-Cases	11
2.3.3	Domain Model	15
2.4	MobileMonitoringApp - Reports module	16
2.4.1	Scope	16
2.4.2	Use-Cases	17

1 Vision and Scope

1.1 Project Vision

Digital forensics is defined as the use of scientifically derived and proven methods towards the preservation, collection, validation, identification, analysis, interpretation and presentation of digital evidence derived from digital sources for the sole purpose of facilitating or furthering the reconstruction of events found to be criminal or helping to anticipate the unauthorized actions shown to be disruptive to planned operations [3]. Readiness is considered as the process of being prepared for a digital investigation before an incident has occurred [2].

The proposal of a mobile monitoring application will promote readiness in digital forensics and protect mobile users from malicious entities and activities. It aims to provide a proactive measure that is undertaken by the mobile device user or mobile device owner. Having this application installed on mobile devices will proactively ensure that relevant digital evidence is made ready and available before an incident occurs. The mobile monitoring application is expected to monitor user activities on a mobile device and report application data/logs to a dashboard on a desktop computer. It will generate reports giving the investigator quick and comprehensive data/logs that provide a starting point during a mobile device investigation.

The objective of the mobile monitoring application is to collect data/logs and assist in understanding the activities performed by a mobile user as well as shedding more light into the behaviour of the mobile user. Combining activities from the various applications promotes a proactive approach which in turn enforces proactive (readiness) measures.

1.2 Project Scope

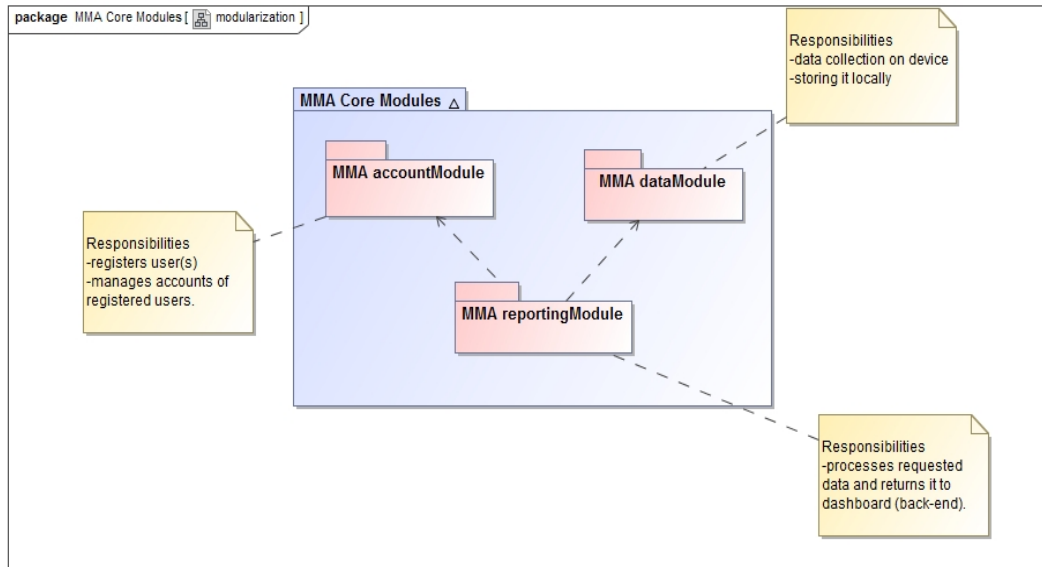


Figure 1: High level system modules

The high level modules of the Mobile Monitoring App is as indicated on figure 1 above. The responsibilities of each module are also noted.

2 Application Requirements and Design

The following section will explain each module in the Mobile Monitoring App. It will discuss the use-case design and functional requirements extracted from the use-case; along with the selected service contracts.

2.1 Modular System

The application is going to use a modular design. This will allow the following:

- add new functionality in the future
- decouple the system

2.2 MobileMonitoringApp - Accounts module

2.2.1 Scope

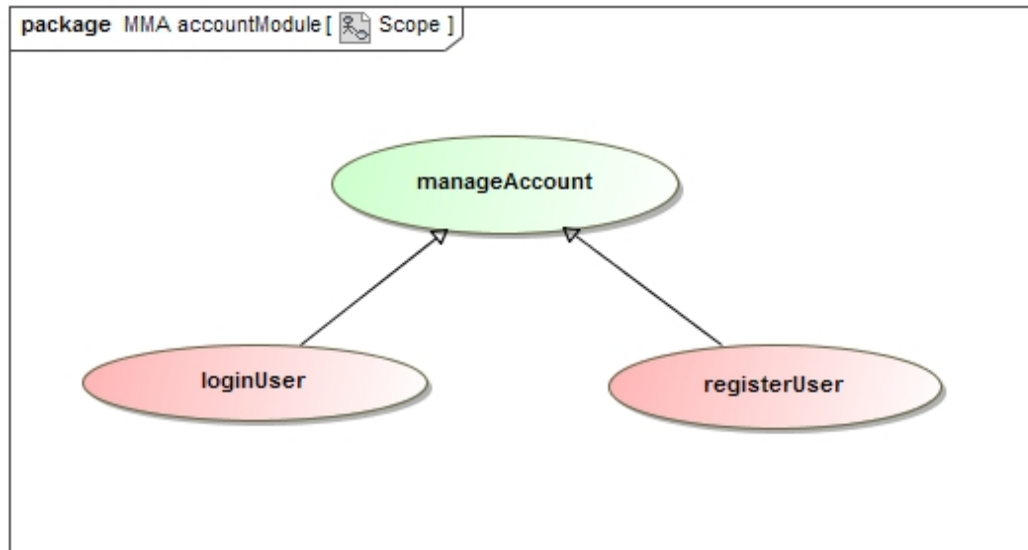


Figure : Scope of mmaAccounts module

2.2.2 Use-Cases

This section provides details on the use-case requirements for the use-cases offered by this module.

2.2.2.1 registerUser - priority: important This use-case registers a user on initial installation after user accepts terms and conditions of use.

Service Contract: The service contract for registerUser is shown in the figure below. The pre-conditions are enforced (raises an exception if not met) and on success the user is registered and the device and user data is persisted through the local database.

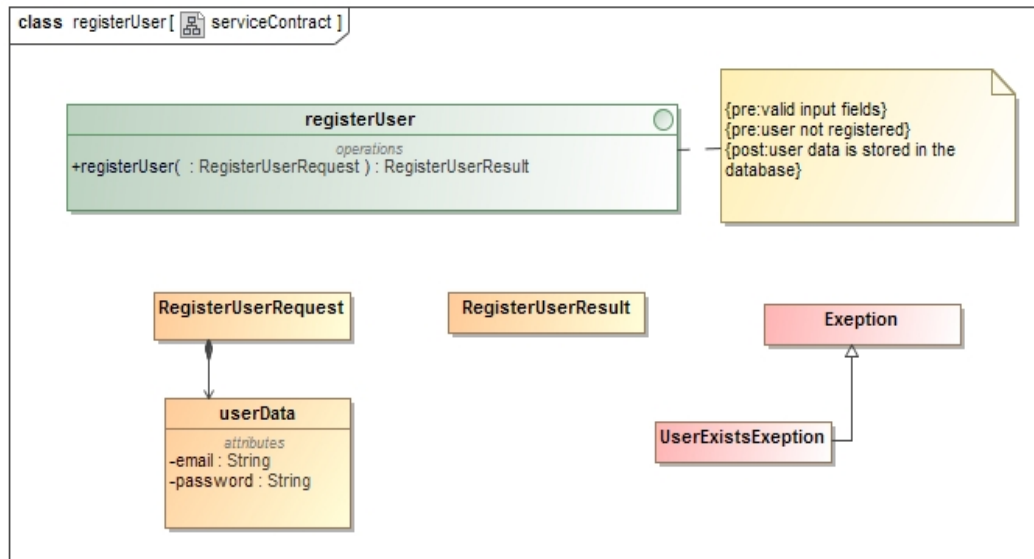


Figure : The service contract for registering a new user on the app

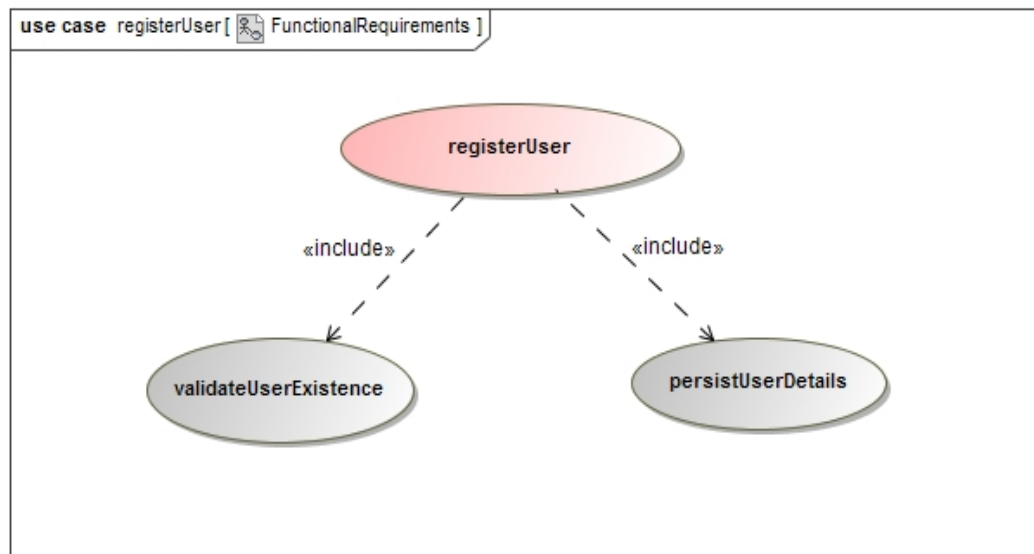


Figure : The functional requirements for registerUser

2.2.2.2 Process specification: When a request is made for a user to register a connection to the database must be established. If no connection to the database can be made then an exception is thrown. Once the connection to the database is made then validateUserExistence is called to ensure that the user does exist on the database already. If the user does exist on the database then an exception is thrown. After everything checks out the user is registered.

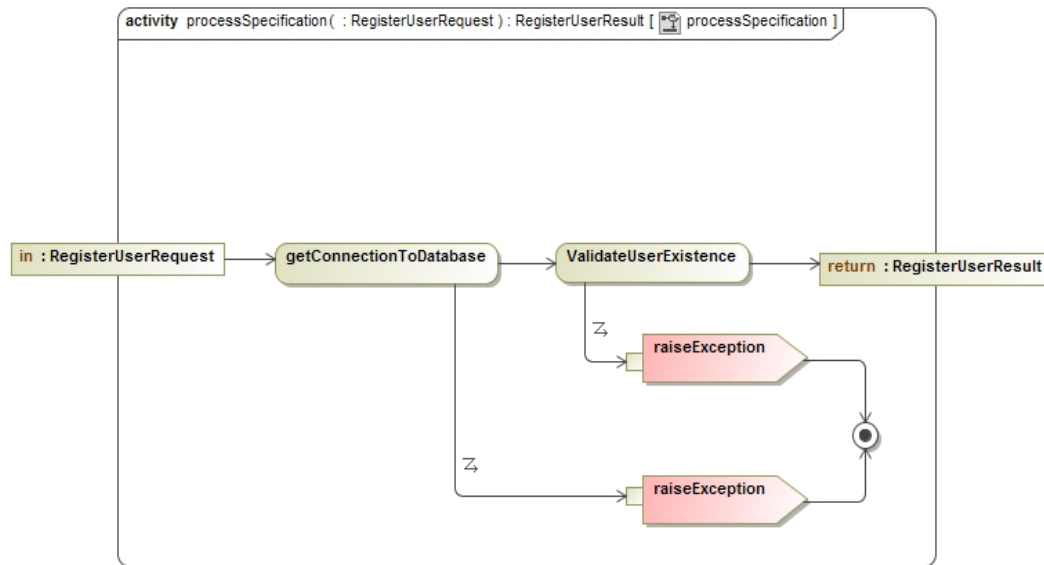


Figure : The process specification for the registerUser use case.

2.2.2.3 loginUser - priority: important This use-case logs a user on initial installation after user accepts terms and conditions of use.

Service Contract: The service contract for loginUser is shown in the figure below. The pre-conditions that are enforced (raises an exception if not met) and on success the user is logged in and the device and user data is persisted through the local database.

2.2.2.4 Process specification: When a request is made for a user to login a connection to the database must be established. If no connection to the database can be made then an exception is thrown. Once the connection to the database is made then validateUserExistence is called to ensure that the user is not already logged in. If the user does not exist on the database then an exception is thrown. After everything checks out the user is then logged in.

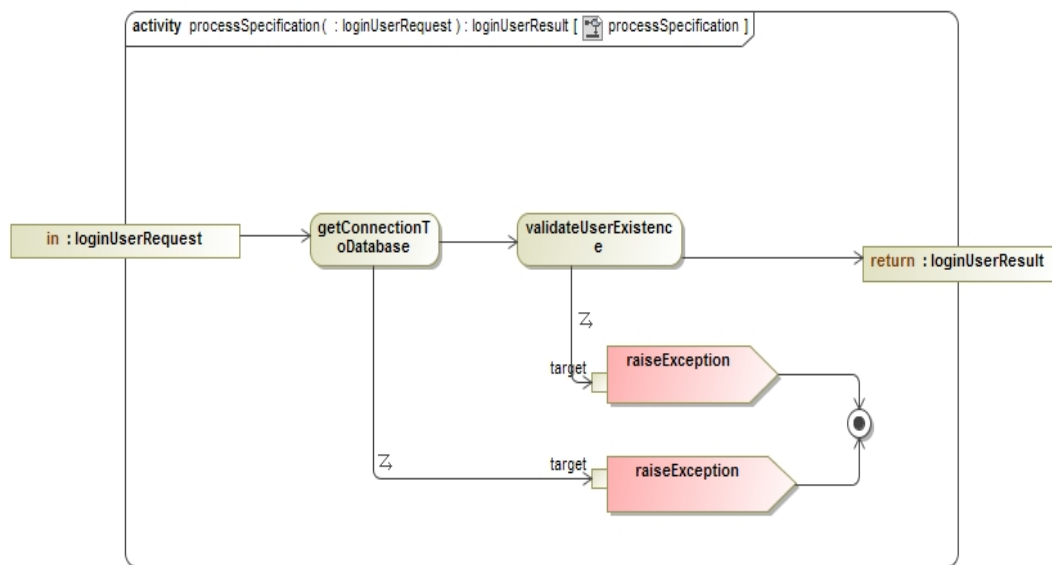


Figure : The process specification for the loginUser use case.

2.2.3 Domain Model

The domain model for this module is very simple and straight forward. It only requires a valid service request.

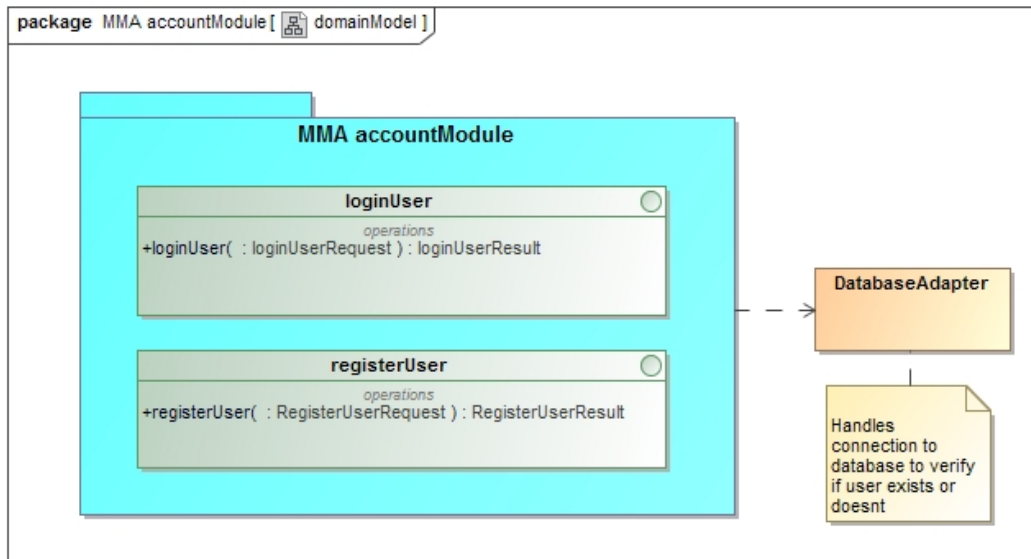


Figure : The domain model for the Account module.

2.2.4 Design Pattern

In order to depict the behavior of this module, we use a State design pattern. This highlights the transitions between a user login and user register.

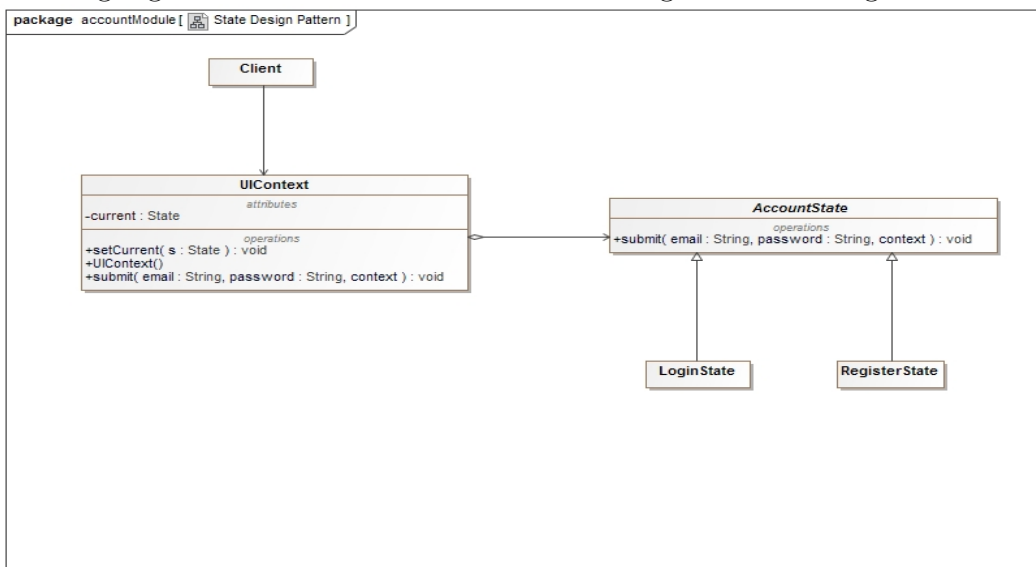


Figure : State Design Pattern.

2.3 MobileMonitoringApp - Data module

2.3.1 Scope

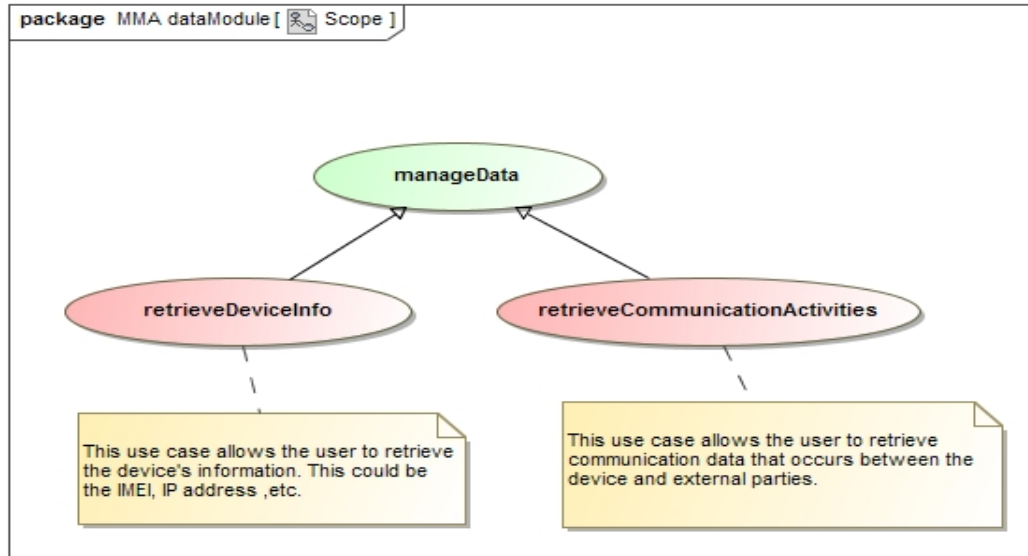


Figure : Scope of MMA dataModule

2.3.2 Use-Cases

This section provides details on the use-case requirements for the use-cases offered by this module.

2.3.2.1 retrieveDeviceInfo - priority: important The retrieveDeviceInfo use case retrieves all the relevant device information from the device.

Service Contract: The service contract for retrieveDeviceInfo is shown in the figure below. The retrieveDeviceInfo receives a retrieveDeviceInfoRequest object that specifies the type of data to be retrieved and stored locally on to a database.

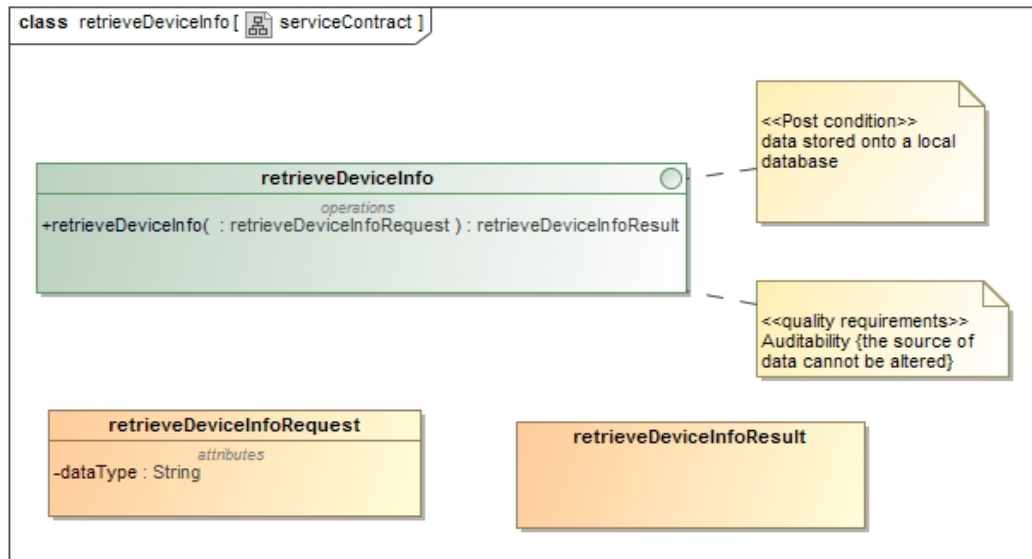


Figure : The service contract for retrieveDeviceInfo on the device

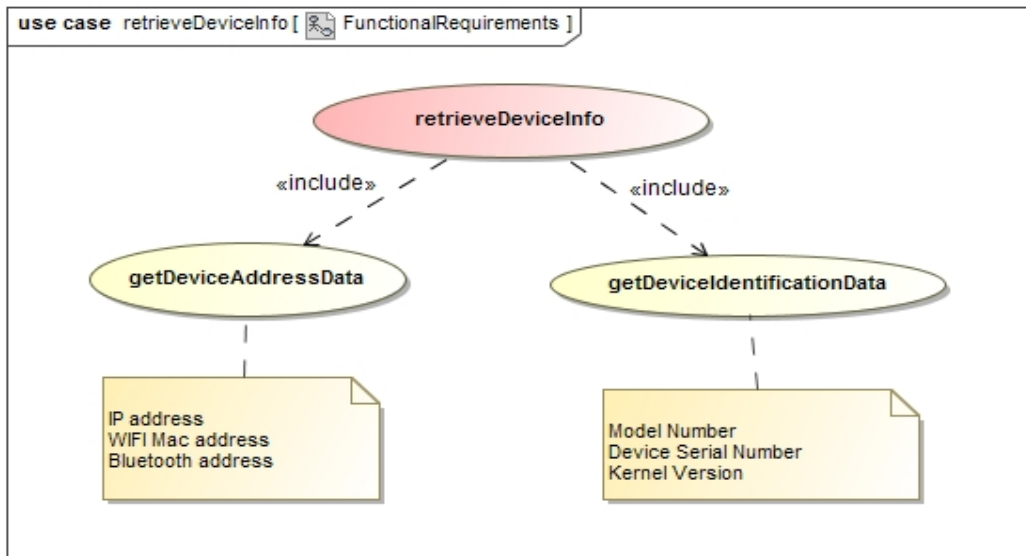


Figure : The functional requirements for retrieveDeviceInfo

2.3.2.2 Process specification: When a request is made for device data to be retrieve. The service checks for the type of data to be retrieved and follows the required channels to retrieve the data.If the service is not suc-

cessful then an exeption is raised. If the service is successfull then an object with the required data is returned as a result of the query.

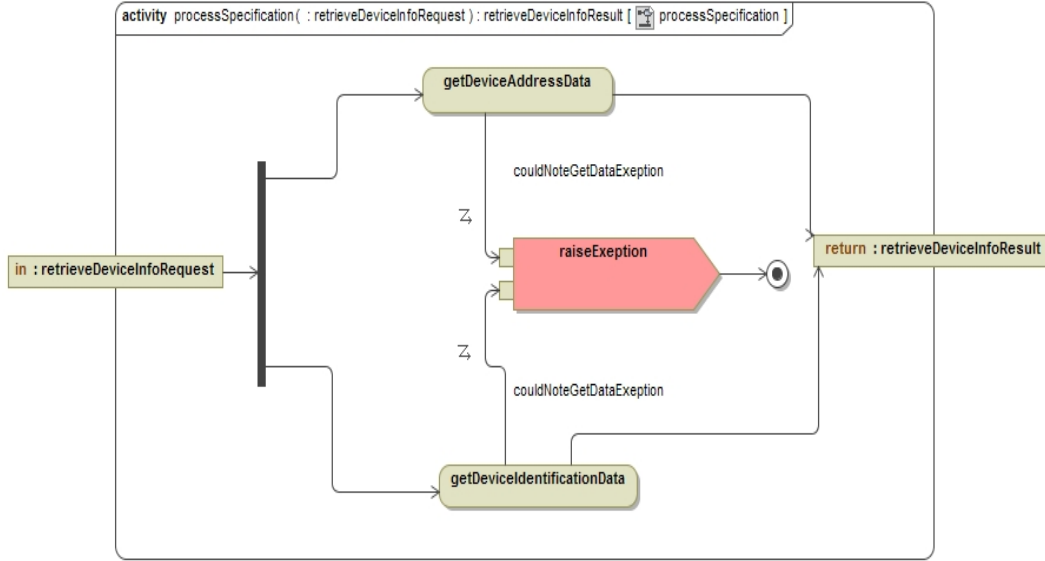


Figure : The process specification for the retrieveDeviceInfo use case.

2.3.2.3 retrieveCommunicationActivities - priority: important The retrieveCommunicationAcitivites use-case retrieves all the users communication activities from the various apps on the device.

Service Contract: The service contract for retrieveCommunicationAcitivites is shown in the figure below. The retrieveCommunicationActivitiesRequest requests the data regardless of the data type and retrieveCommunicationActivitiesRequest retrieves the data where it packages the data and stores it on the database locally.

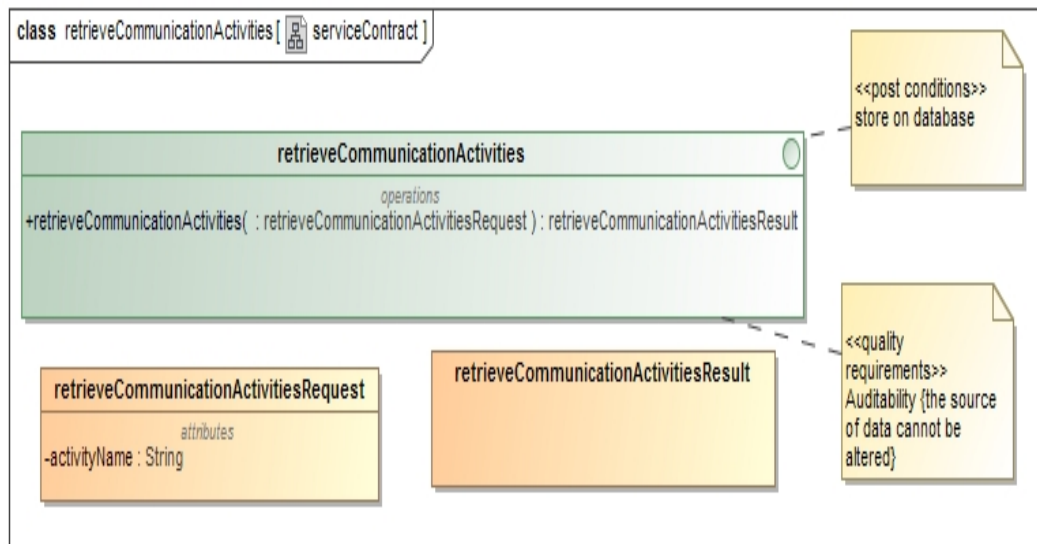


Figure :The service contract for retrieveCommunicationAcitivites on the device

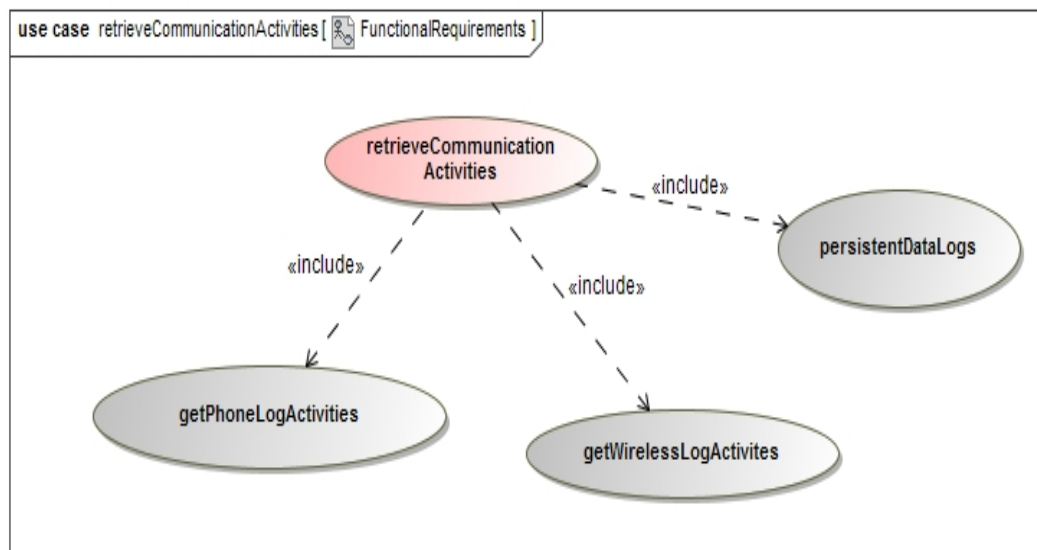


Figure : The functional requirements for retrieveCommunicationActivities

2.3.2.4 Process specification: The service receives a request that specifies the type of activity the service needs to collect data from. The service will retrieve the data from that activity and if it fails then an exception will be raised. The result from the function will be an object that wraps the required data.

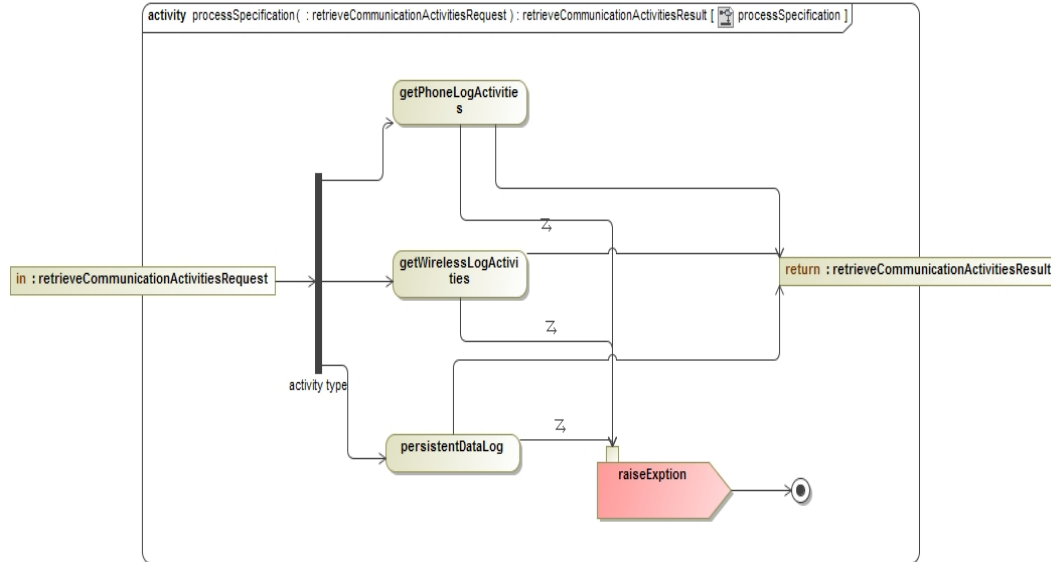


Figure : The process specification for the retrieveDeviceInfo use case.

2.3.3 Domain Model

The domain model for this module is very simple and straight forward. It only requires a valid service request.

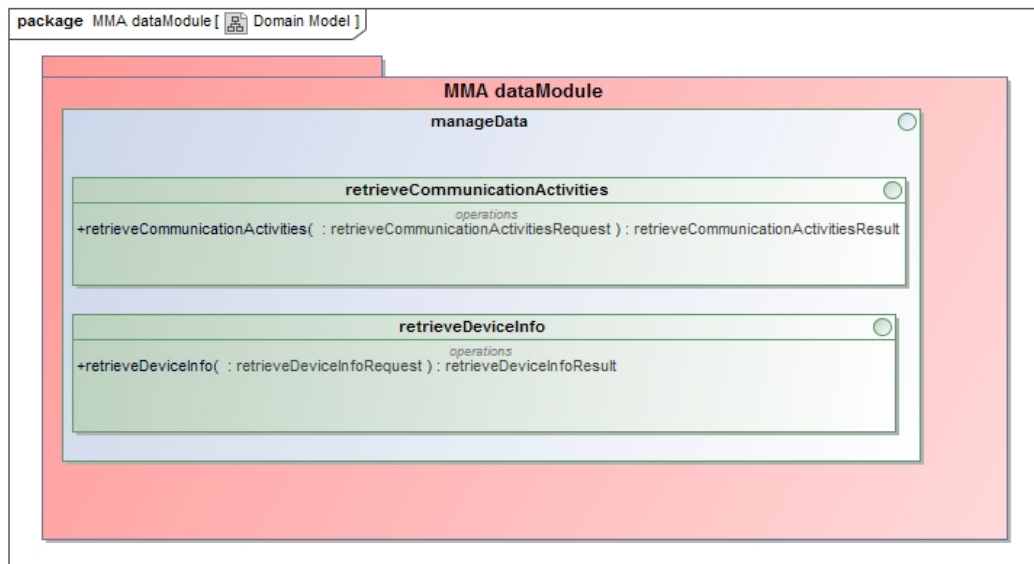


Figure : The domain model for the Data module.

2.4 MobileMonitoringApp - Reports module

2.4.1 Scope

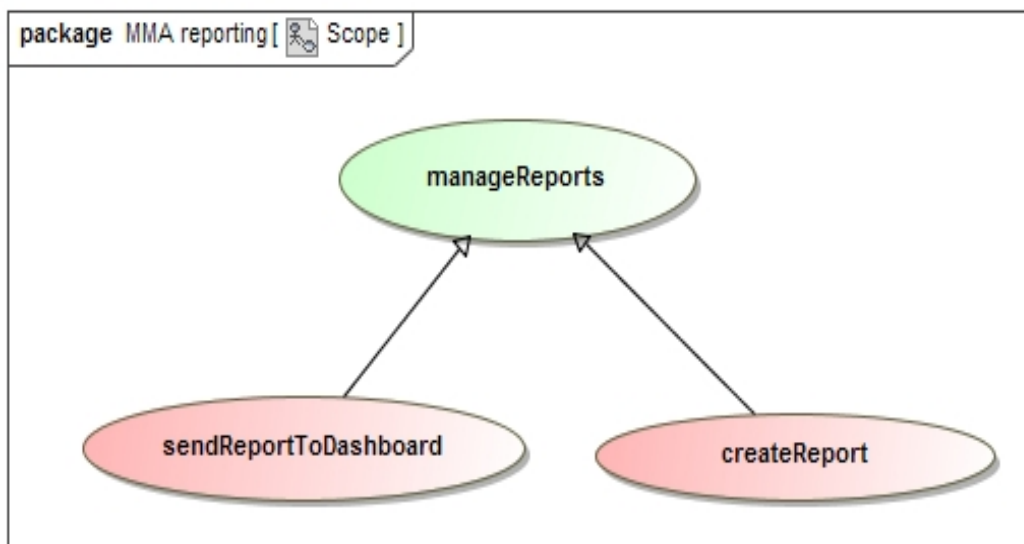


Figure : Scope of mmaReporting module

2.4.2 Use-Cases

This section provides details on the use-case requirements for the use-cases offered by this module.

2.4.2.1 createReport - priority: important This use-case retrieves logs from the device and creates a report from those specific logs.

Service Contract: The service create report is shown in the figure below. The pre-condition is enforced.

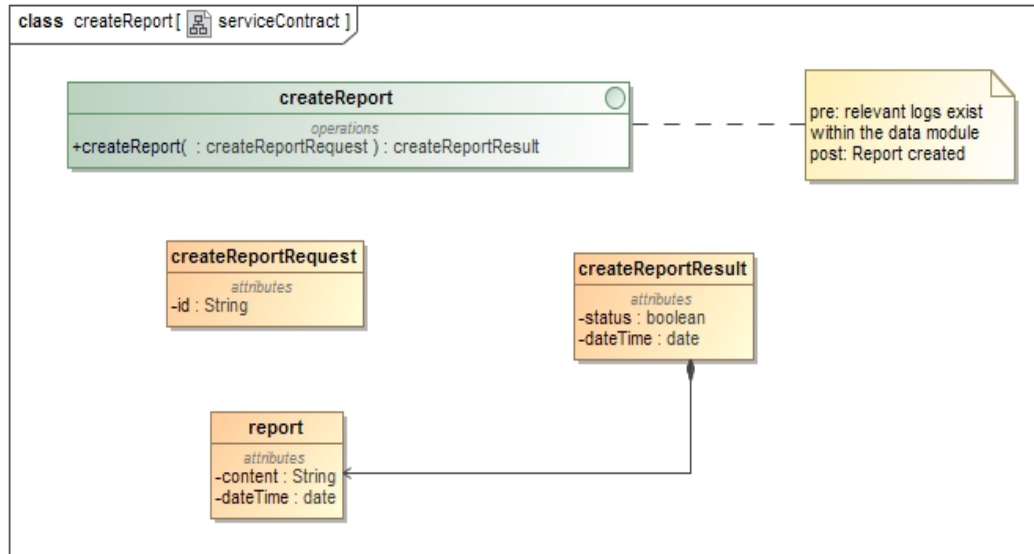


Figure : The service contract for createReport

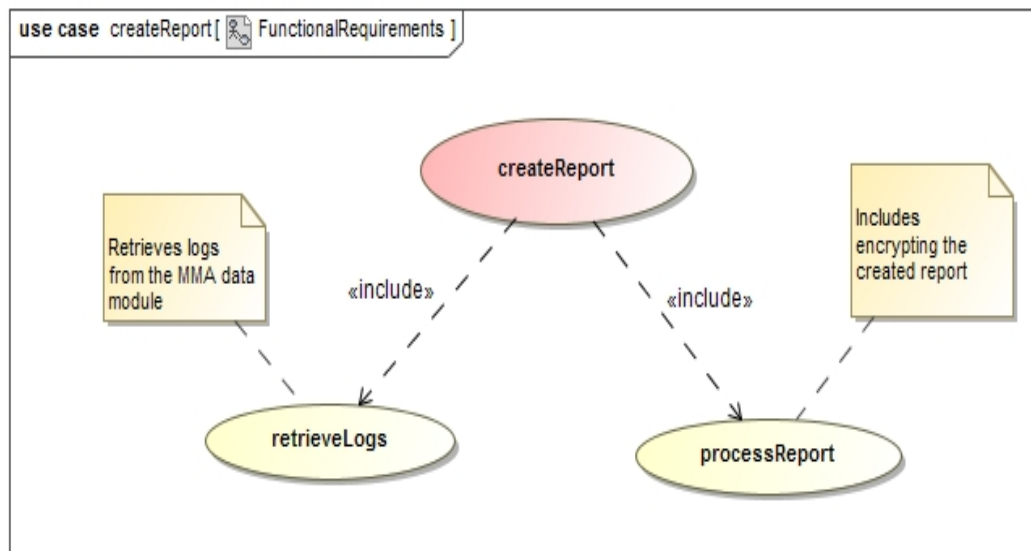


Figure : The functional requirements for createReport

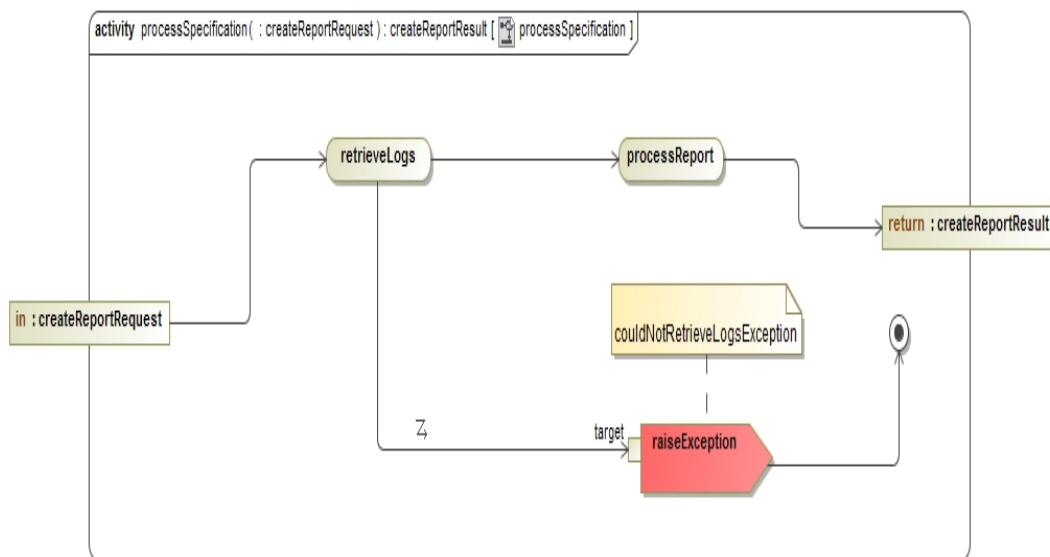


Figure : The Process Specification for createReport

2.4.2.2 sendReportToDashboard - priority: important This module sends the report onto a server where they will be saved on a database to be displayed later on a dashboard .

Service Contract: The service contract for sendReportToDashboard is shown in the figure below. The pre-condition is not enforced i.e. If the app fails to establish a connection with the server, the report will be saved temporarily on the device until a connection is established.

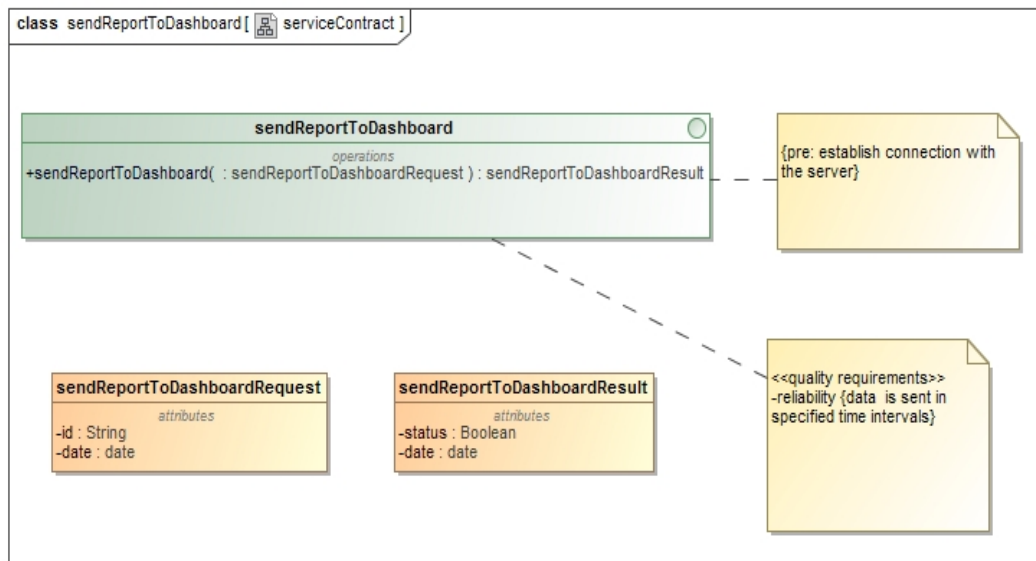


Figure : The service contract for sendReportToDashboard

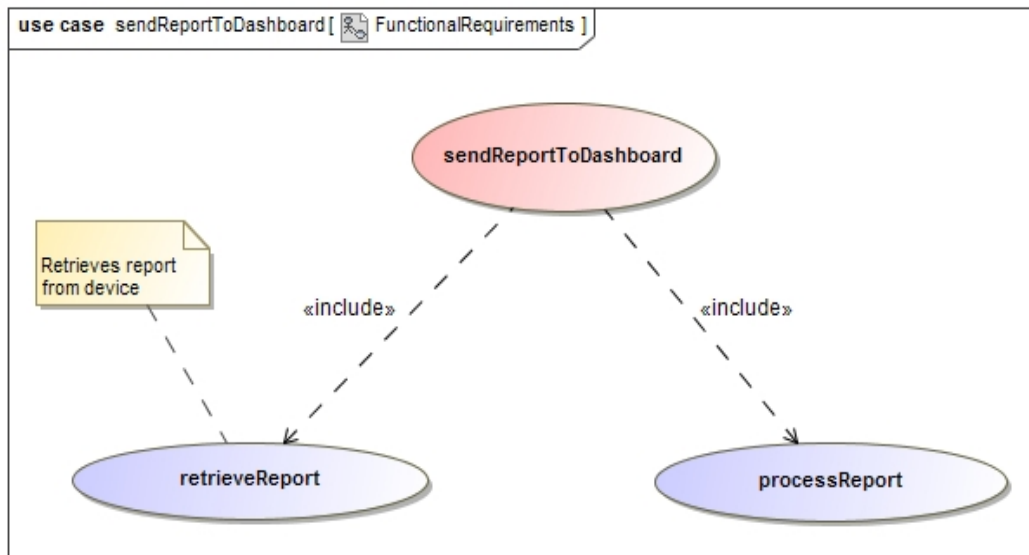


Figure : The functional requirements for sendReportToDashboard

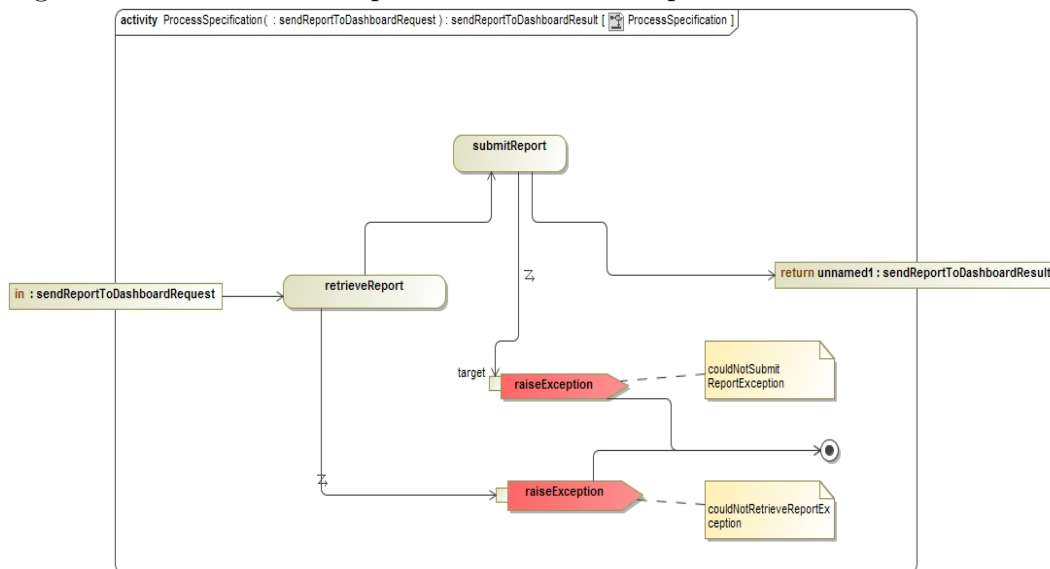


Figure : The Process Specification for sendReportToDashboard

2.4.2.3 Domain Model: This section contains the domain model for the reporting module. This module retrieves logs from the data module.

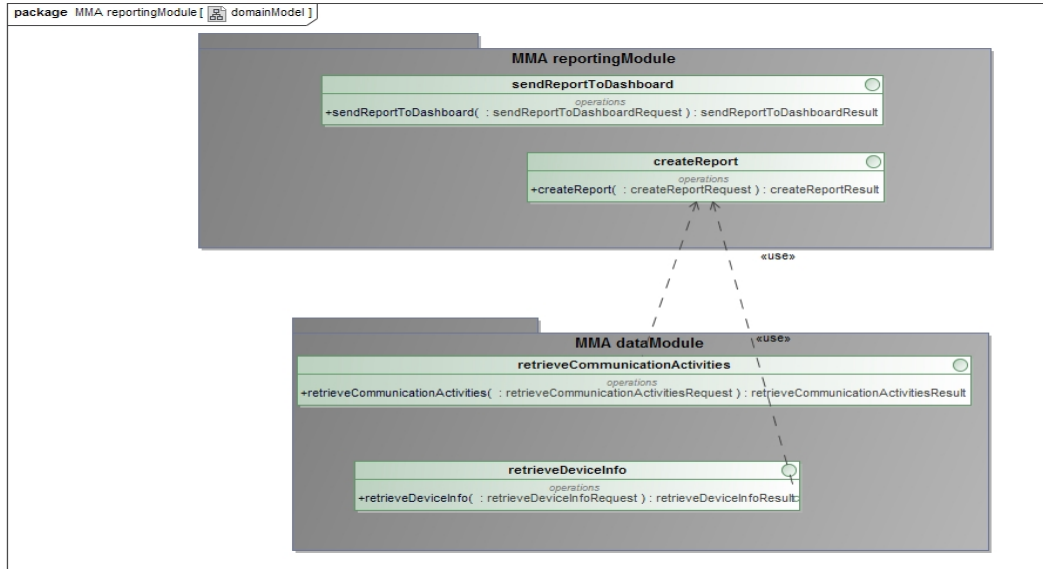


Figure : The domain model for the reporting module.