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

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. Readiness is considered as the process of being prepared for a digital investigation before an incident has occurred.

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 undertaking 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

The high level modules of the iCrawler monitoring app is as indicated on the figure below. The responsibilities of each module are also noted.

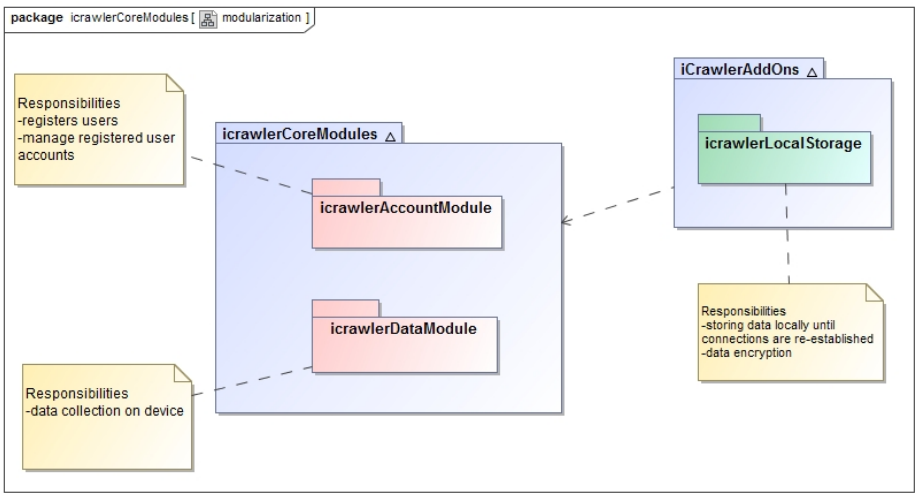

Figure 1: High Level App Modules

2 Application Requirements and Design

The following section will explain each module in the mobile monitoring app. The use-case design, functional requirements extracted from the use-case along with the selected service contracts will also be discussed.

2.1 Modular System

The application uses modular design approach. This allows the following to be archived:

- add new functionality in the future
- decouple the system

2.2 iCrawler - Account Module

2.2.1 Scope

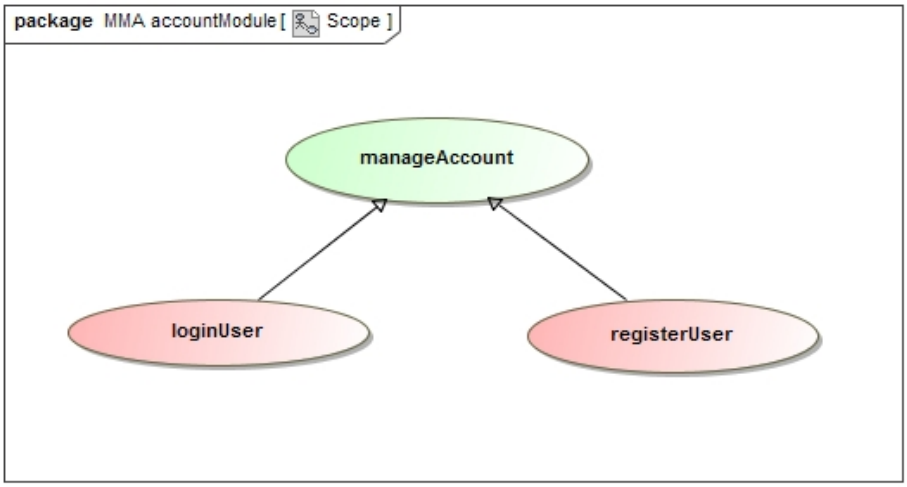

Figure 2: Scope - mmaAccounts module

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.

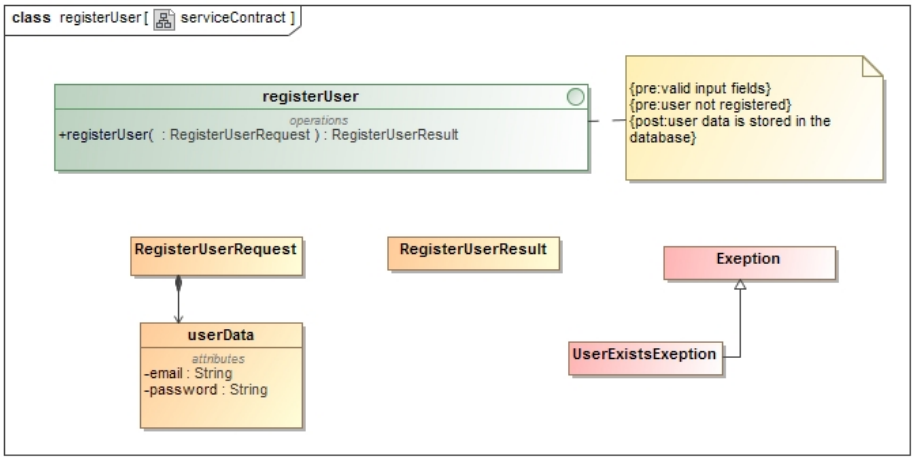

Figure 3: Service Contract - Register user

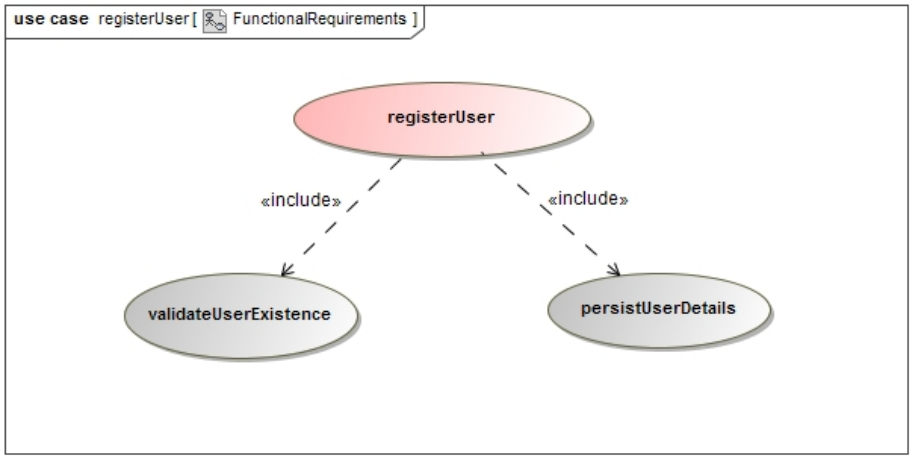

Figure 4: Functional Requirements - Register user

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 validateUserExistance is called to ensure that the user does exist on the database already. If the user does exist on the database then and exception is thrown. After everything checks out the user is registered.

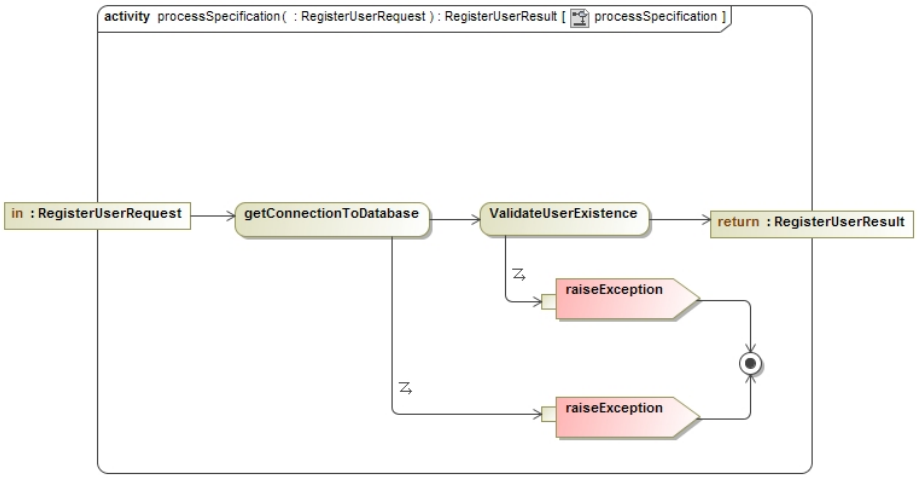

Figure 5: Process Specification - Register user

2.2.2.3 loginUser - priority: important This use-case logins 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 and exception if not met) and on success the user is logged in and the device and user data is persisted through the local database.

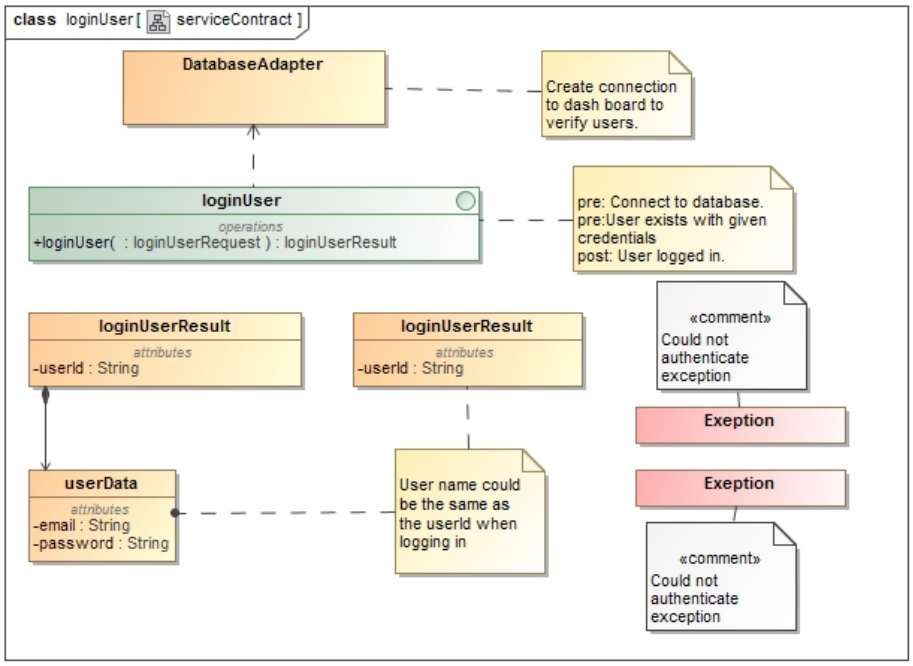

Figure 6: Service Contract - Login user

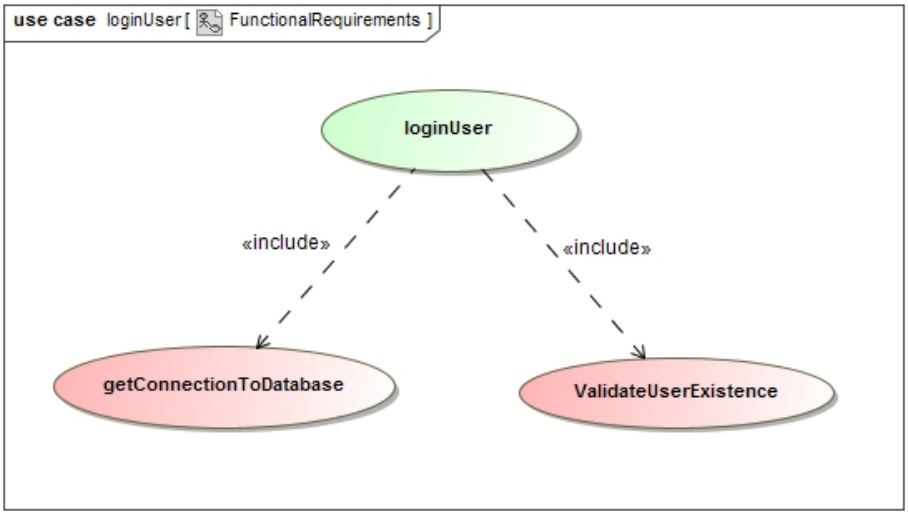

Figure 7: Functional Requirements - Login user

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 validateUserExistance is called to ensure that the user is not already logged in. If the user does not exist on the database then and exception is thrown. After everything checks out the user is then logged in.

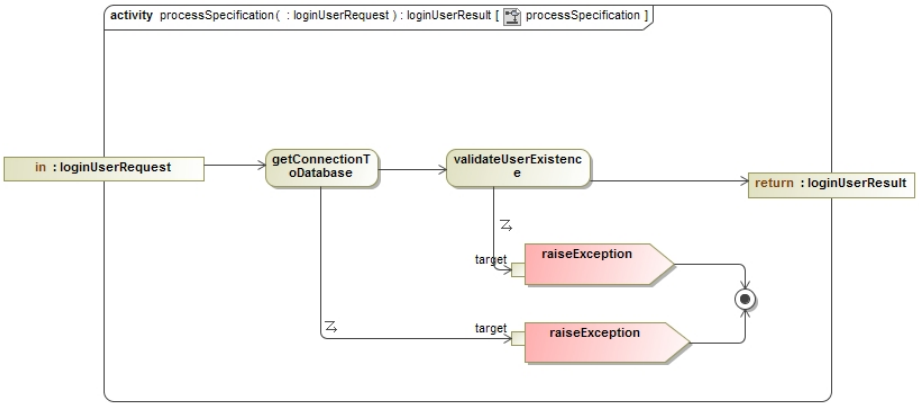

Figure 8: Process Specification - Login user

2.2.3 Domain Model

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

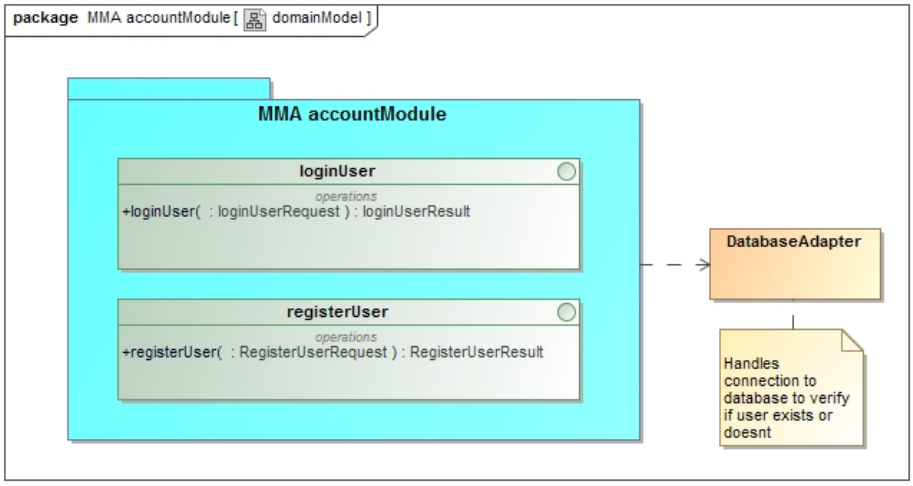

Figure 9: Domain Model - Account Module

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.

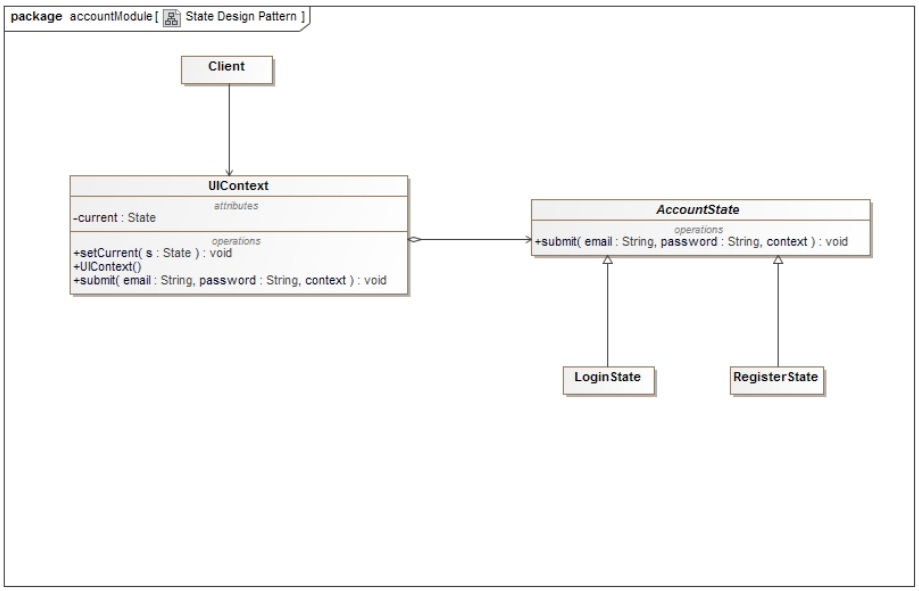

Figure 10: Design Pattern - State

2.3 MobileMonitoringApp - Data module

2.3.1 Scope

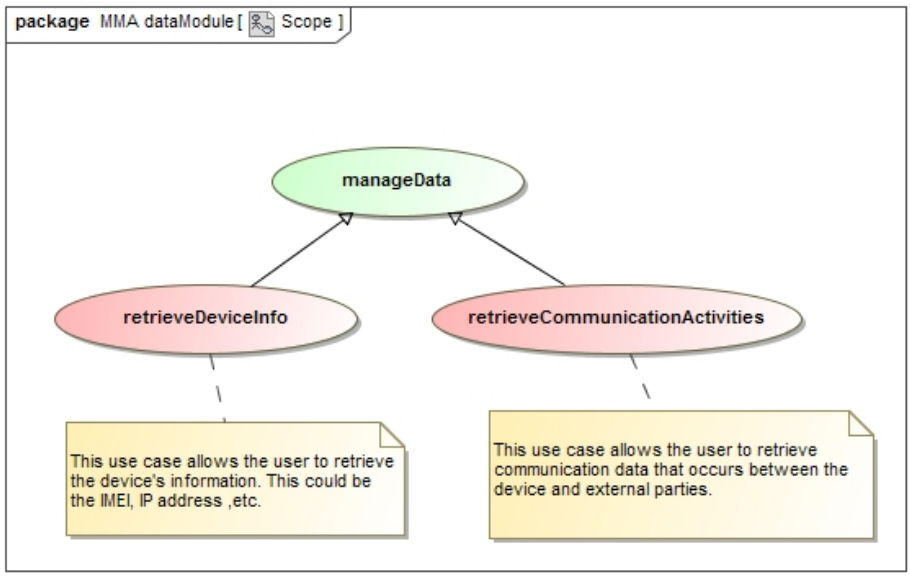

Figure 11: Scope - dataModule

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 recieves a retrieveDeviceInfoRequest object that specifies the type of data to be retrieved and stored locally on to a database.

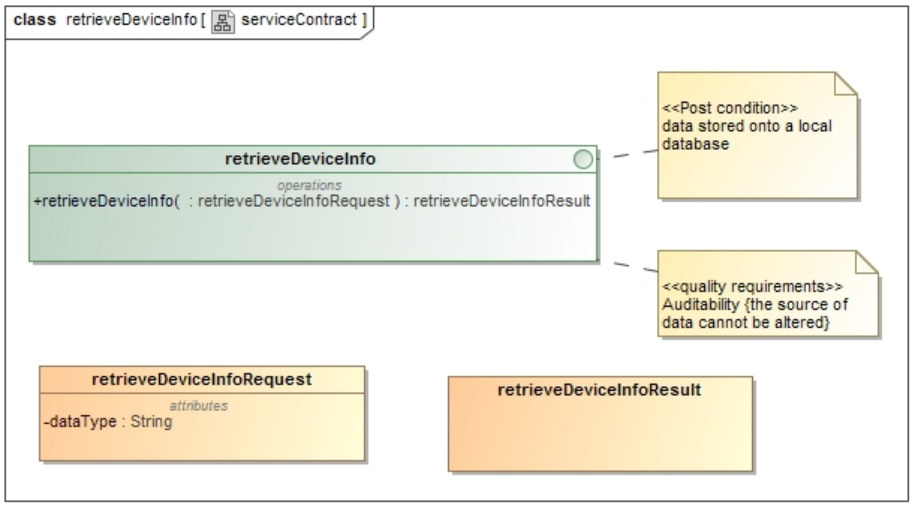

Figure 12: Service Contract - retrieveDeviceInfo

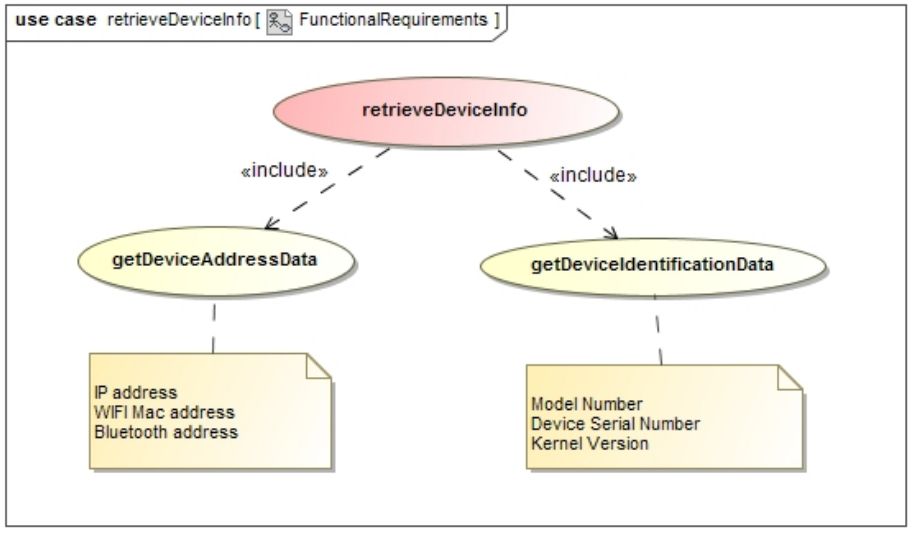

Figure 13: Functional Requirements - retrieveDeviceInfo

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 successful 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.

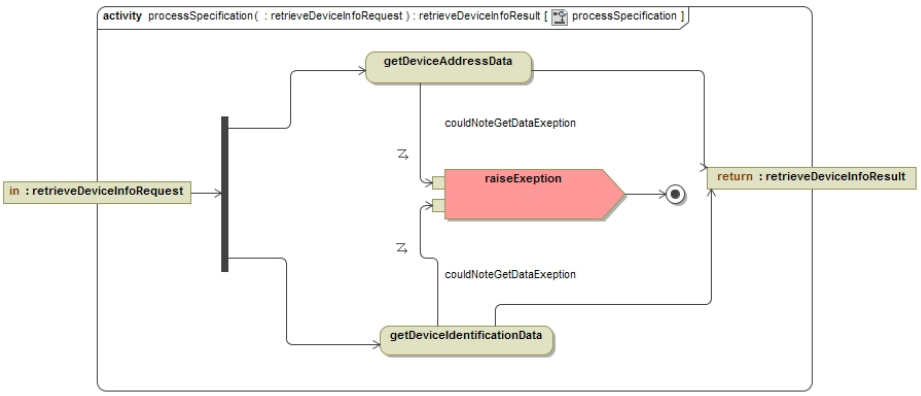

Figure 14: Process Specification - retrieveDeviceInfo

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 retrieveCommunicationActivities-Request 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.

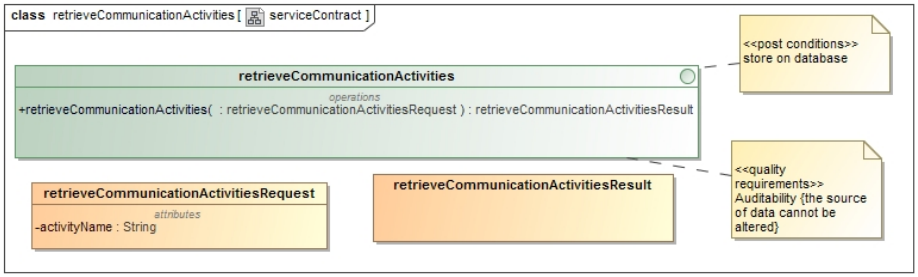

Figure 15: Service Contract - retrieveCommunicationAcitivites

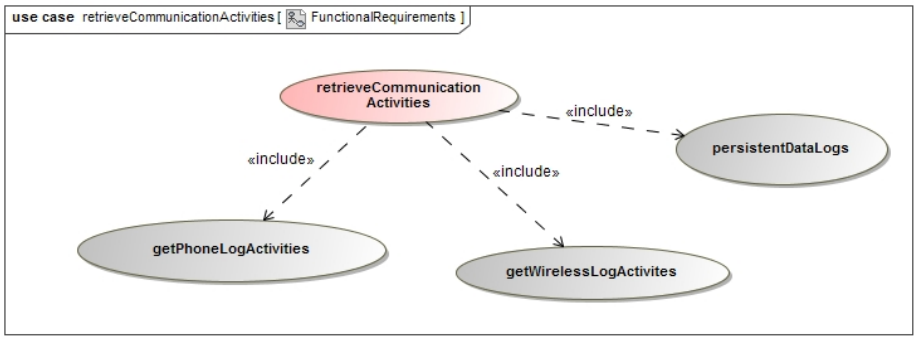

Figure 16: Functional Requirements - Retrieve Communication Acitivites

2.3.2.4 Process specification: The service recieves 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 exption will be raised.The result from the function will be an object that wraps the required data.

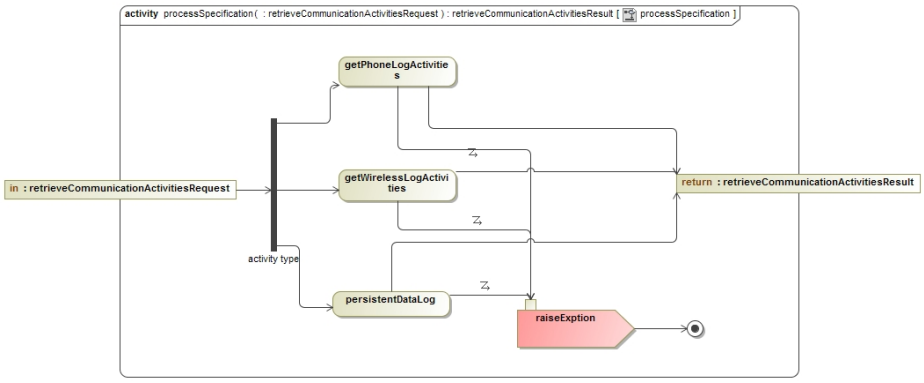

Figure 17: Process Specification - Retrieve Communication Acitivites

2.3.3 Domain Model

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

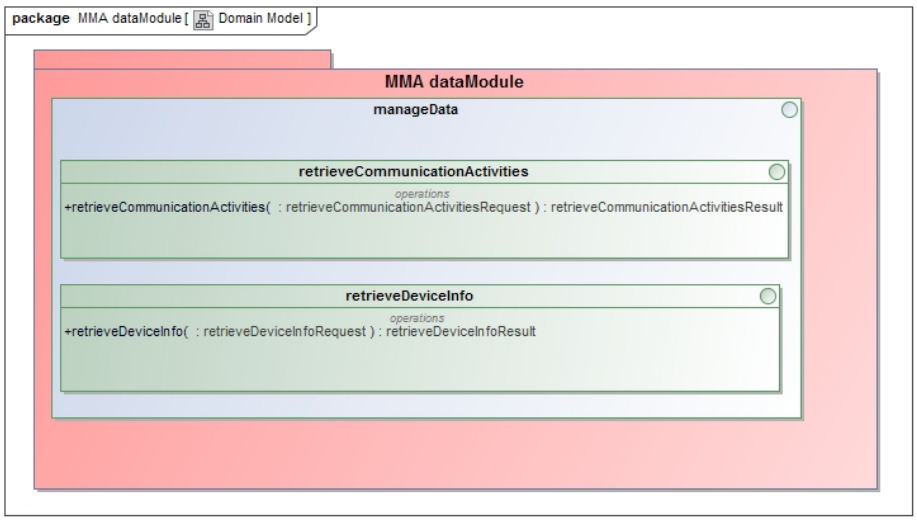

Figure 18: Domain Model - Data Module

2.4 MobileMonitoringApp - Reports module

2.4.1 Scope

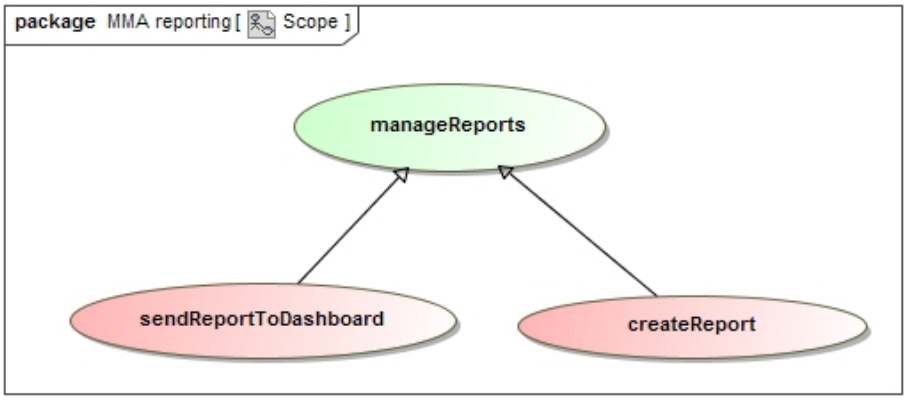

Figure 19: Scope - Reporting Module

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.

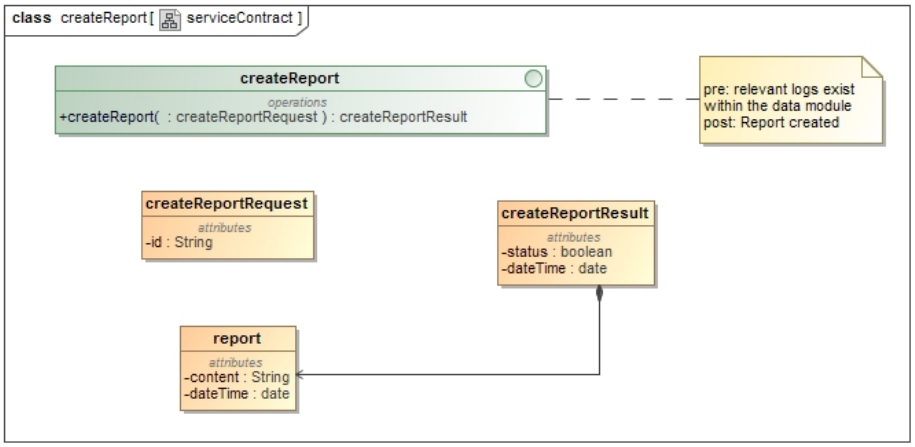

Figure 20: Service Contract - Create Report

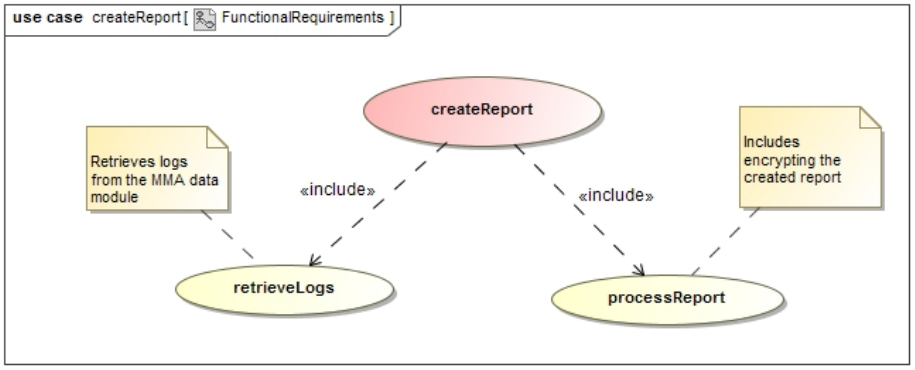

Figure 21: Functional Requirements - Create Report

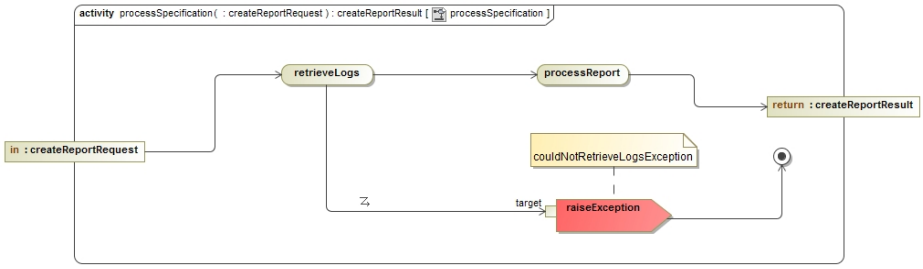

Figure 22: Process Specification - Create Report

2.4.2.2 sendReportToDashboard - priority: important This module sends the report onto a server where they will be saved on a database to de 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 estabilish a connection with the server, the report will be saved temoparily on the device until a connection is estabilished.

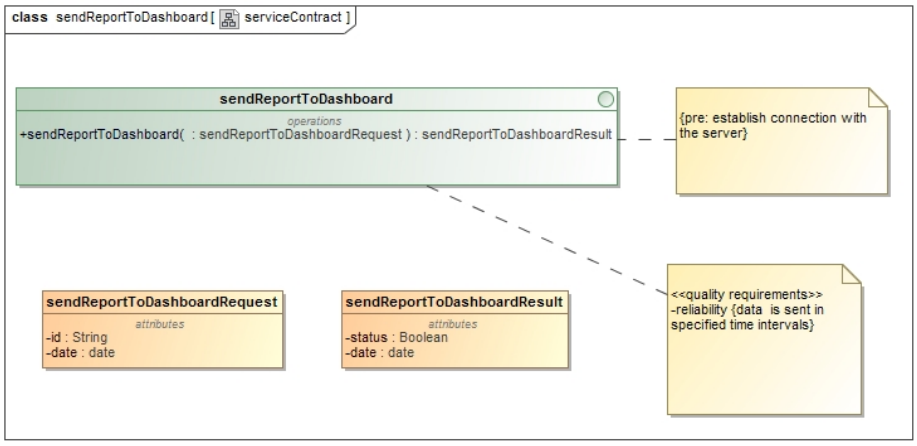

Figure 23: Service Contract - Send Report To Dashboard

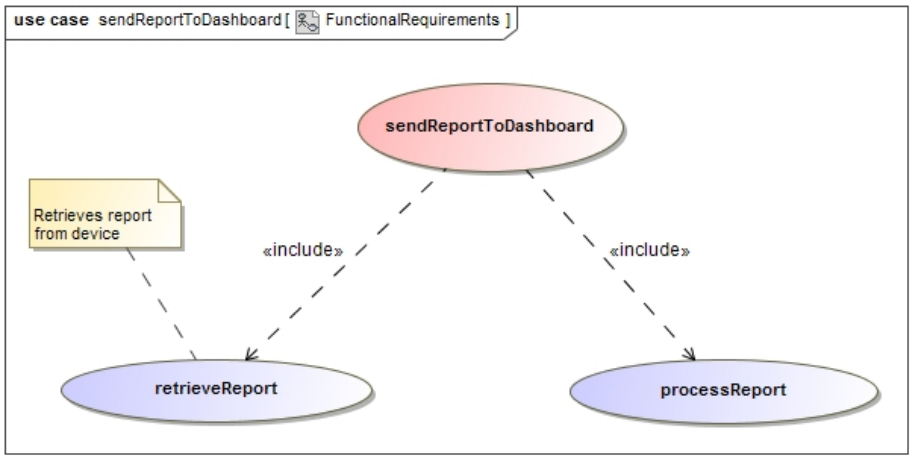

Figure 24: Functional Requirements - Send Report To Dashboard

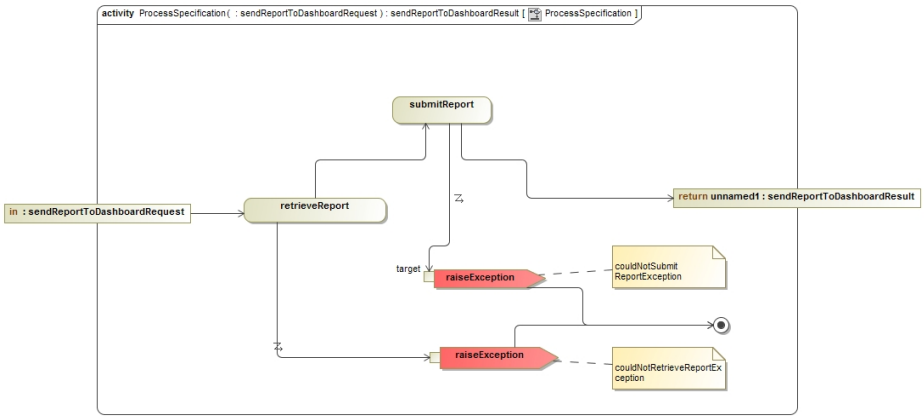

Figure 25: Process Specification - Send Report To Dashboard

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

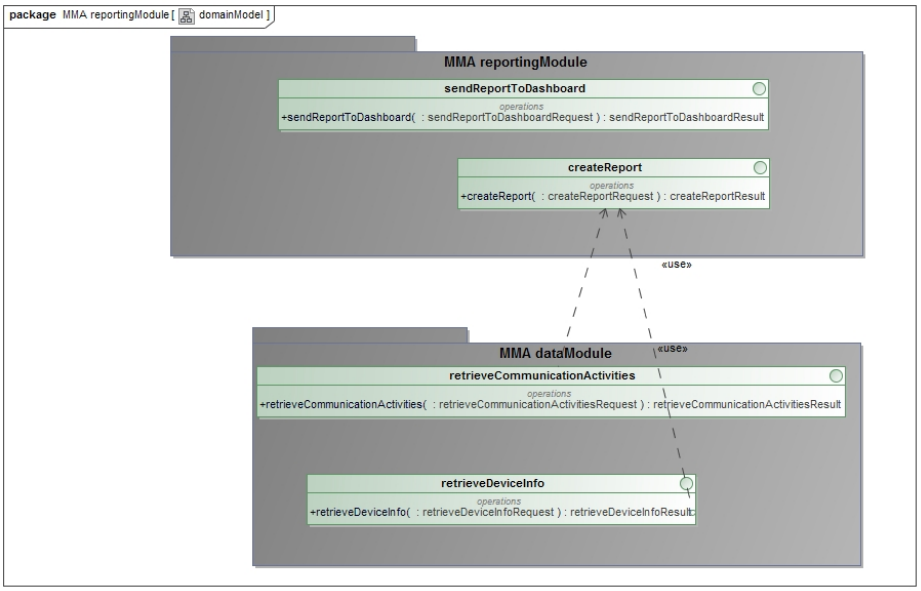

Figure 26: Domain Model - mmaReporting Module