Software Design Specification

(SDS)

Access Control Framework for

Cloud

Version 1.0

Prepared by Fizza Shafiq, Umer Ahmad Cheema & Ismail Tirmizi

NUST-SEECS

20 Nov 2016

Contents

[1. Introduction 4](#_Toc469909916)

[1.1 System Purpose 4](#_Toc469909917)

[1.2 System Scope 4](#_Toc469909918)

[1.3 Definitions, Acronyms, and Abbreviations 5](#_Toc469909919)

[2. System Overview 5](#_Toc469909920)

[3. Analysis Model 6](#_Toc469909921)

[3.1 Interface Objects 6](#_Toc469909925)

[3.1.2 Acquaintance Association interface objects 7](#_Toc469909926)

[3.1.3 Containership 7](#_Toc469909927)

[3.1.3.2 Add Interface 8](#_Toc469909928)

[3.1.3.3 Update Interface 8](#_Toc469909929)

[3.1.3.4 Delete Interface 9](#_Toc469909930)

[9](#_Toc469909931)

[3.2 Sub System Diagram 11](#_Toc469909932)

[3.3 Class Diagram 12](#_Toc469909933)

[3.4 Control Objects 13](#_Toc469909934)

[4. Design Model 16](#_Toc469909936)

[4.1 Sequence Diagrams 16](#_Toc469909937)

[4.1.1 Database Population 16](#_Toc469909940)

[4.3.2 Policy Creation 21](#_Toc469909944)

[4.3.3 Policy Generation 27](#_Toc469909945)

[4.3.4 Manage Pre-Updates 28](#_Toc469909947)

[4.3.5 Manage OnGoing-Updates 28](#_Toc469909948)

[4.3.6 Manage Post-Updates 30](#_Toc469909949)

[4.3.7 Request Interception 31](#_Toc469909950)

[4.3.8 Attribute Update 32](#_Toc469909951)

[4.3.9 Access Request Re-evaluation 33](#_Toc469909952)

[4.3.10 Policy Evaluation 34](#_Toc469909953)

[4.4 Communication Diagrams 35](#_Toc469909954)

[4.4.1 Database Population 35](#_Toc469909955)

[4.4.2 Policy Creation 38](#_Toc469909956)

[4.4.3 Policy Generation 42](#_Toc469909957)

[4.4.4 Manage Pre-Updates 43](#_Toc469909958)

[4.4.5 Manage On-Updates 43](#_Toc469909959)

[4.4.6 Manage Post Updates 44](#_Toc469909960)

[4.4.7 Request Interception and Policy enforcement 44](#_Toc469909961)

[4.4.8 Enforce Attribute Update 45](#_Toc469909962)

[4.4.9 Access Request Re-Evaluation 46](#_Toc469909963)

[4.4.10 Policy Evaluation 47](#_Toc469909964)

[4.5 Activity Diagrams 48](#_Toc469909965)

[4.6 Deployment Diagram 48](#_Toc469909966)

[4.7 Component Diagram 48](#_Toc469909967)

[4.8 Package Diagram 48](#_Toc469909968)

# 

# Introduction

## System Purpose

After completing the SRS document we are done with the requirement phase of the project so next we must move onto the architecture and design phase of the document. This SDS document will serve as a means to explain minute details of the architecture and design of our ‘Rabt’ Website Application. To present our work we have decomposed the project into numerous components and this document will dive into detail to explain them. The top level architecture of the system gives a higher level overview, design strategies, detailed system design, various design views, UML diagrams and deployment architecture that will be described in this document.

## System Scope

The ‘*Rabt’* Website Application will extend its use to those users who are excessively involved in project development. They will be able to find experts and people with experience in the project field that they are currently working on. These experts can then help the project developers in many ways. The web application is developed to assist the students in universities and the freelancer community, which are mostly in need of guidance and abstract help in the start of the project. When these guys dive in their questions and their problems become very much bespoke and specific they require people who have previously done such projects for help. Our web app plays a critical role in finding such experts for the users. The website application also provides the user an interface where when a user is done with his project, they can also promote his status as an expert in that project ensuring the return of the help he received from the site.

The application will assist user by giving them accounts and working space where they can organize and optimize their work. It also includes module which will allow users to set deadlines for their work, upload project files to share and a newsfeed area to communicate with multiple people.

This *Software Design Specification* document describes the details of the system design decisions. The design of graphical user interface for the system administration and CSCs is discussed in Software Requirement Specification document, so they are not addressed in this document.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **ABAC** | Attribute Based Access Control |
| **ACaaS** | Access Control-as-a-Service |
| **CSC** | Cloud Service Consumer. |
| **CSP** | Cloud Service Provider |
| **FGAC** | Fine-Grained Access Control |
| **PAPaaS** | Policy Administration Point |
| **PDPaaS** | Policy Decision Point |
| **PEPaaS** | Policy Enforcement Point |
|  |  |
| **PIP** | Policy Information Point |
| **RBAC** | Role Based Access Control |
| **SaaS** | Software–as-a-Service |
| **SAML** | Security Assertion Markup Language |
| **UCON** | Usage-based access CONtrol |
| **XACML** | eXtensible Access Control Markup Language |

# System Overview

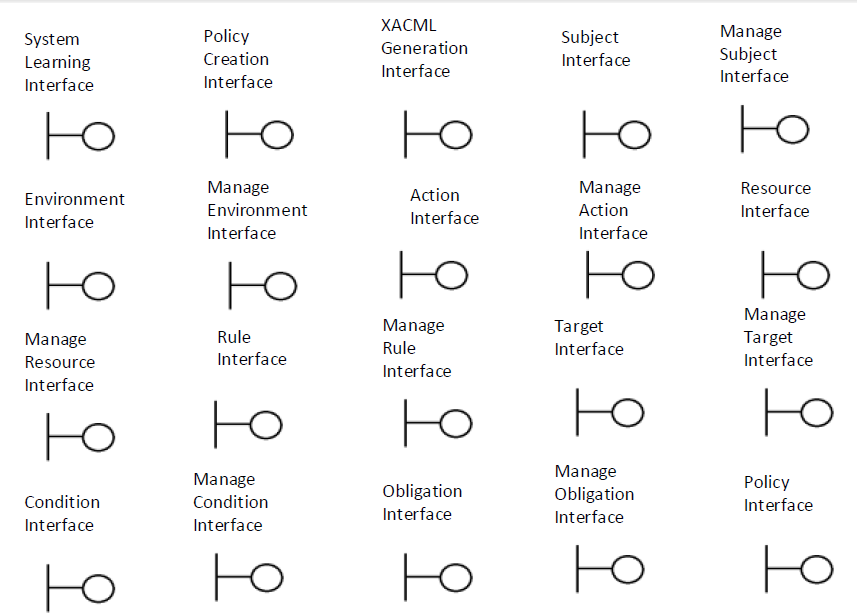
The final product is implemented as an API with deployment of components as web services on Cloud. SaaS layer where Cloud consumers can access the authorization services. The major components of the framework are Policy Administration Point (PAPaaS), Policy Enforcement Point (PEPaaS) and Policy Decision Point (PDPaaS) which are deployed as a service on Cloud platform. First major component of the framework is PAPaaS, which provides a web based flexible and user-friendly graphical interface for policy creation and management. The PAPaaS provides interfaces to add or remove policies and update various policy related parameters and attributes. In addition to policy creation, a mechanism is provided for storing these policies in a policy repository. Another important component is PDPaaS, which is responsible for evaluating the access control policies and making the access decisions accordingly. PEPaaS is provided as a web service that acts as a gateway for all the authorization requests send to the framework. When an end user wants to access the application’s resources, an access request is sent to the PEPaaS for policy enforcement. The PEPaaS forwards the access request towards PDPaaS of the framework. The PDPaaS retrieves the applicable policy for evaluation of authorization request received from PEPaaS. After the policy evaluation, the final authorization decision is returned to PEPaaS for the enforcement of access control and obligations. On the basis of the PDPaaS’s decision which is either permit or deny, PEPaaS enforces the access control on application.

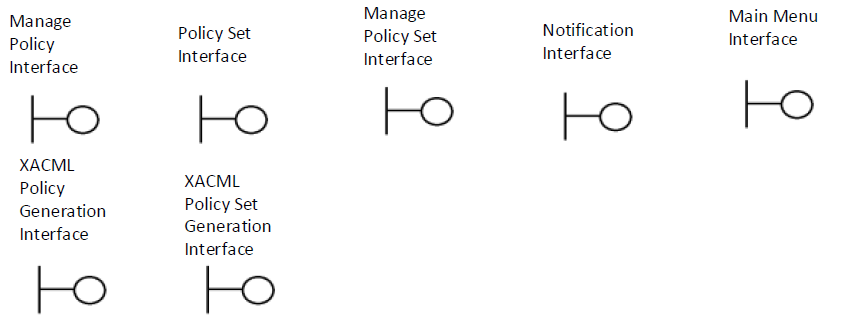
# Analysis Model



## Interface Objects

The interface objects represent the main interfaces of the system. Given below is the list of different interface objects used in our system.





### Acquaintance Association interface objects

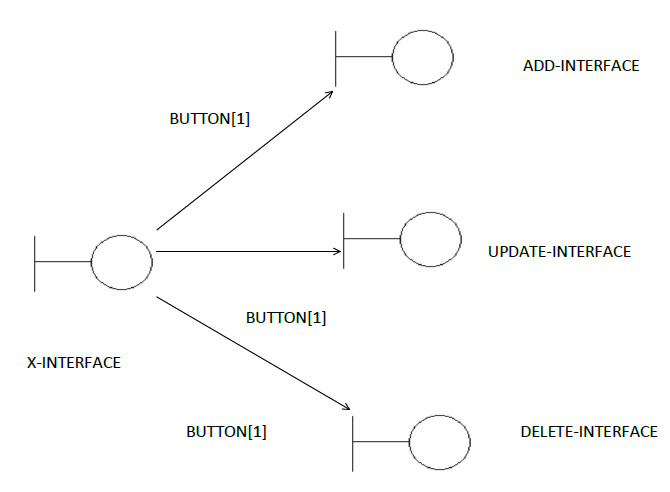
**System Learning**: In Add-X-interface, Delete-X-interface and Update-X-Interface, X can be Subject, Action, Resource or Environment.

**Policy Creation:** In Add-Y-Interface, Delete-Y-Interface and Update-Y-Interface, Y represents Target, Condition, Rule, Obligation, Policy, and PolicySet.

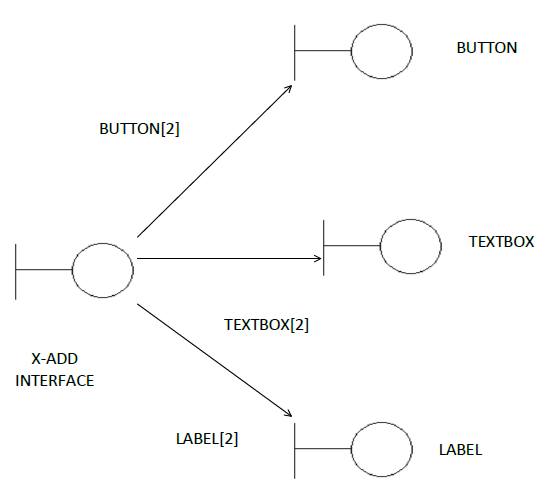
**Policy Generation:** In Generate-Z-Interface, the Z represents Policy and PolicySet

### Containership

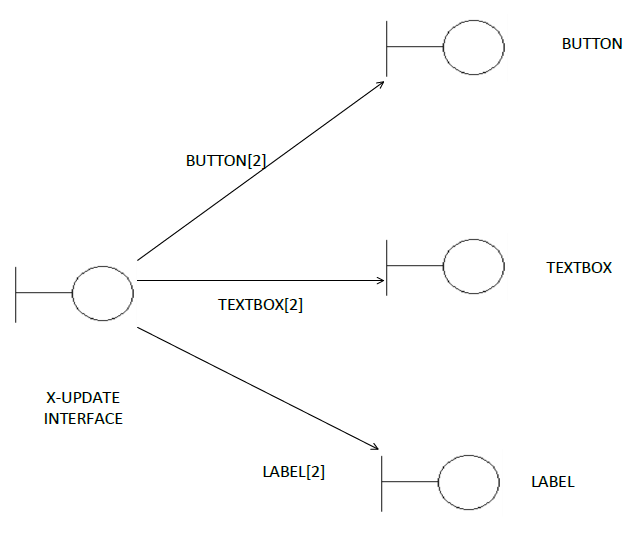
* + - 1. Initial Interface



# Add Interface

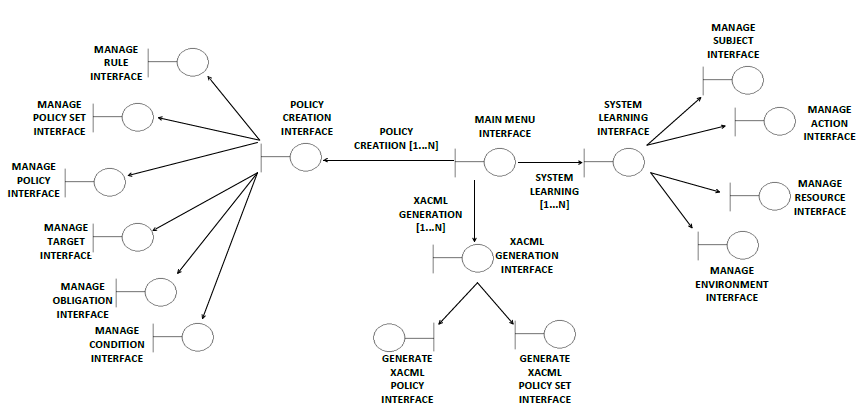


# Update Interface

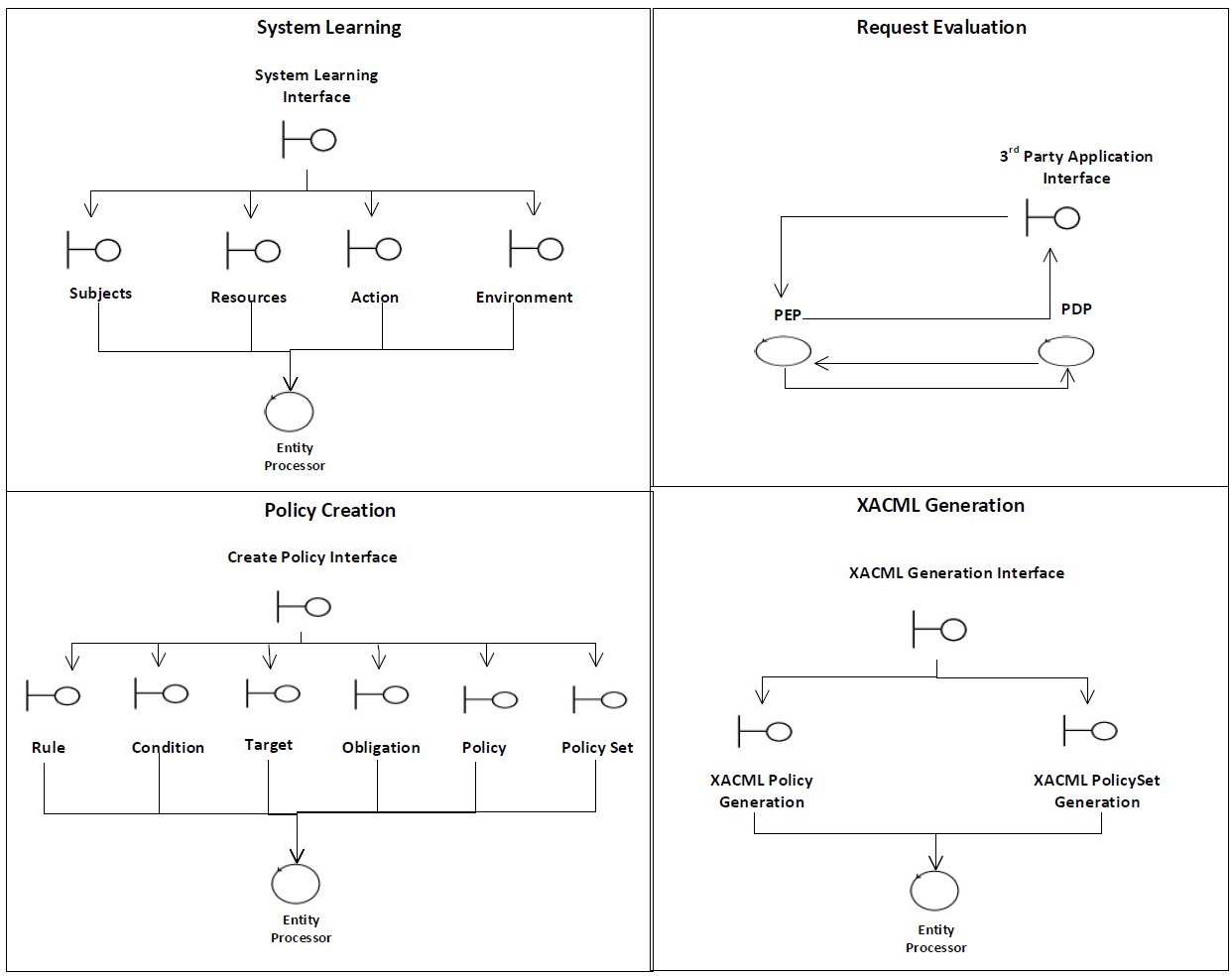


# Delete Interface

# 



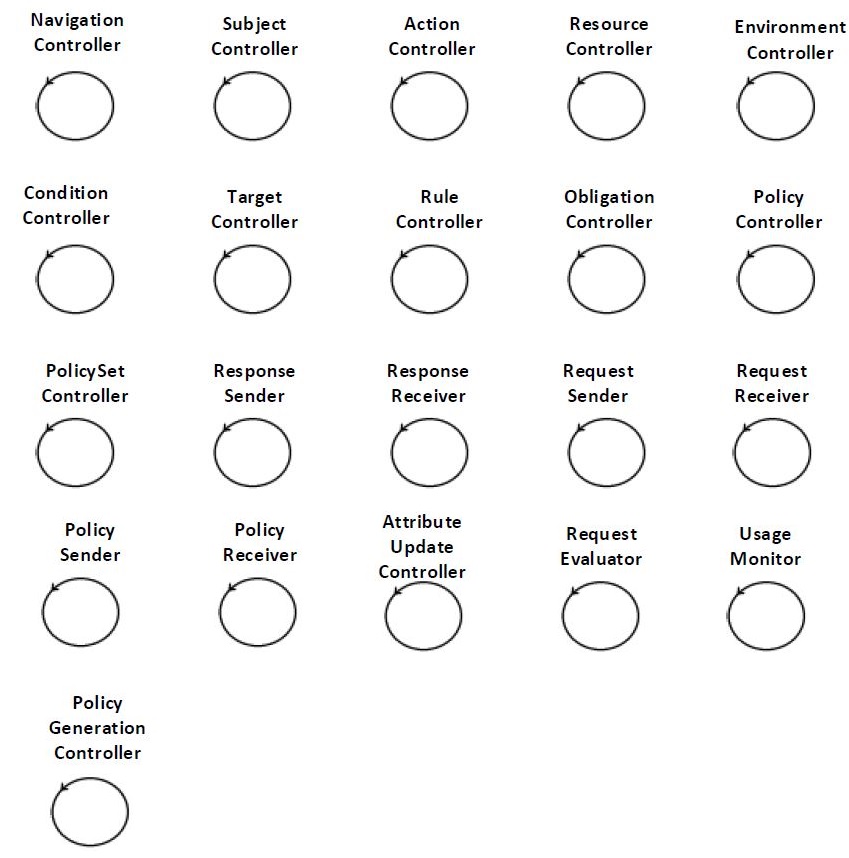
## Sub System Diagram



## Class Diagram



## Control Objects

Control objects are responsible for computation or processing tasks. Following are the main control objects in the proposed framework.

**Description of Control Objects**

1. **Navigation Controller:**

This control object is responsible for the navigation among interfaces.

1. **Subject Controller:**

This control object is used to manage policy *Subject* parameter. It handles all the Subject Add, Update and Delete operations as per user request.

1. **Action Controller:**

This control object is used to manage policy *Action* parameter. It handles all the Action Add, Update and Delete operations as per user request.

1. **Resource Controller:**

This control object is used to manage policy *Resource* parameter. It handles all the Resource Add, Update and Delete operations as per user request.

1. **Environment Controller:**

This control object is used to manage policy *Environment* parameter. It handles all the Environment Add, Update and Delete operations as per user request.

1. **Condition Controller:**

This control object is used to manage policy *Condition* parameter. It handles all the Condition Add, Update and Delete operations as per user request.

1. **Target Controller:**

This control object is used to manage policy *Target* parameter. It handles all the Target Add, Update and Delete operations as per user request.

1. **Rule Controller:**

This control object is used to manage policy *Rule* parameter. It handles all the Rule Add, Update and Delete operations as per user request.

1. **Obligation Controller:**

This control object is used to manage policy *Obligation* parameter. It handles all the Obligation Add, Update and Delete operations as per user request.

1. **Policy Controller:**

This control object is used to manage policy *Policy* parameter. It handles all the Policy Add, Update and Delete operations as per user request.

1. **Policy Set Controller:**

This control object is used to manage policy *Policy Set* parameter. It handles all the Subject Add, Update and Delete operations as per user request.

1. **Response Receiver:**

This control object is used to receive XACML based policy response from PDPaaS component.

1. **Response Sender:**

This control object is used to send XACML based policy response to PEPaaS component.

1. **Request Receiver:**

This control object is used to receive XACML based policy request at PDPaaS component.

1. **Request Sender:**

This control object is used to send XACML policy request from PEPaaS component.

1. **Policy Sender:**

This control object is used to send XACML policy from PAPaaS to policy repository.

1. **Policy Receiver:**

This control object is used to receive XACML policy from PAPaaS in policy repository.

1. **Attribute Update Controller:**

This control object is used to update the values of attributes in UCON specific Pre, Post and OnGoing Update requests.

1. **Request Evaluator:**

This control object is used to evaluate the XACML policy evaluation request.

1. **Usage Monitor:**

This control object is used to invoke the access request re-evaluation request in case of OnGoing and Post attribute updates.

1. **Policy Generation Controller:**

This control object is responsible for the generation and storage of XACML based policy.

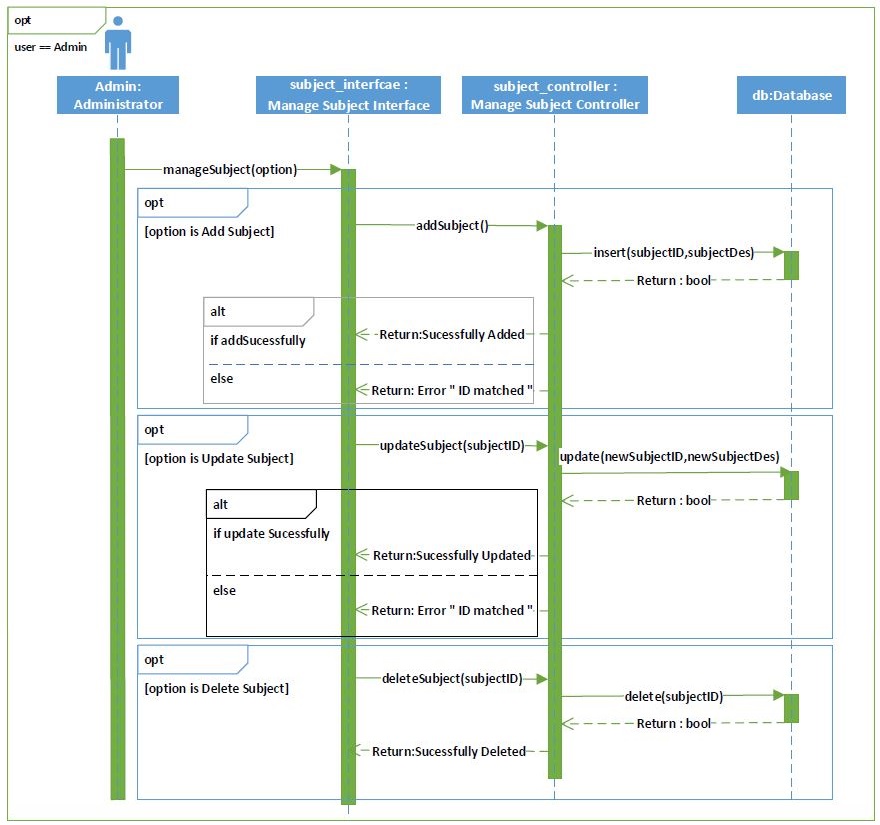
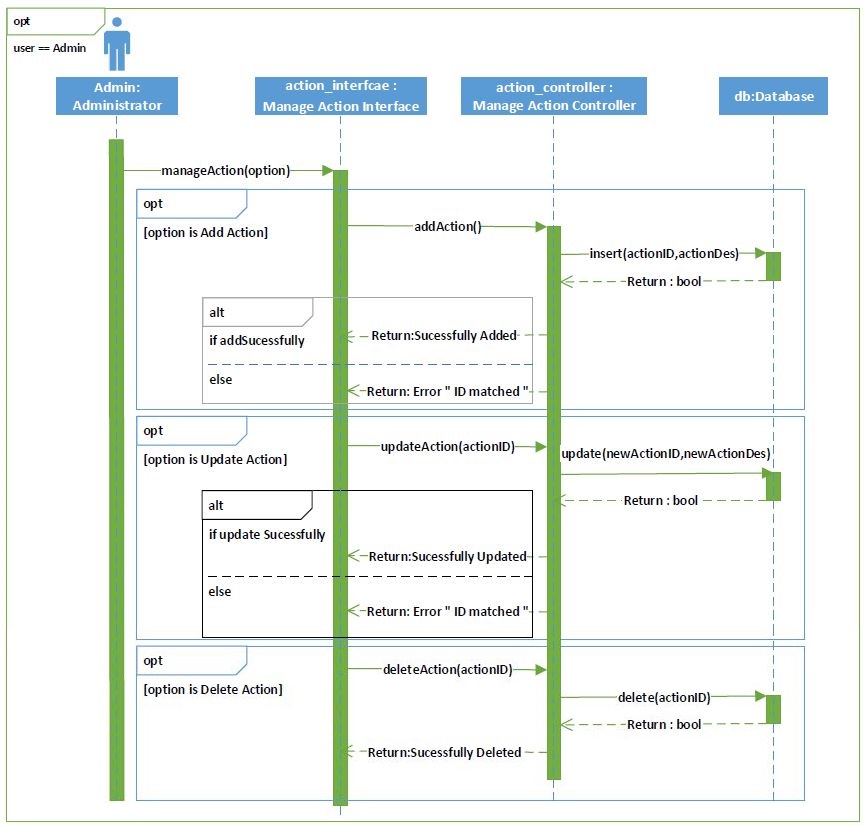
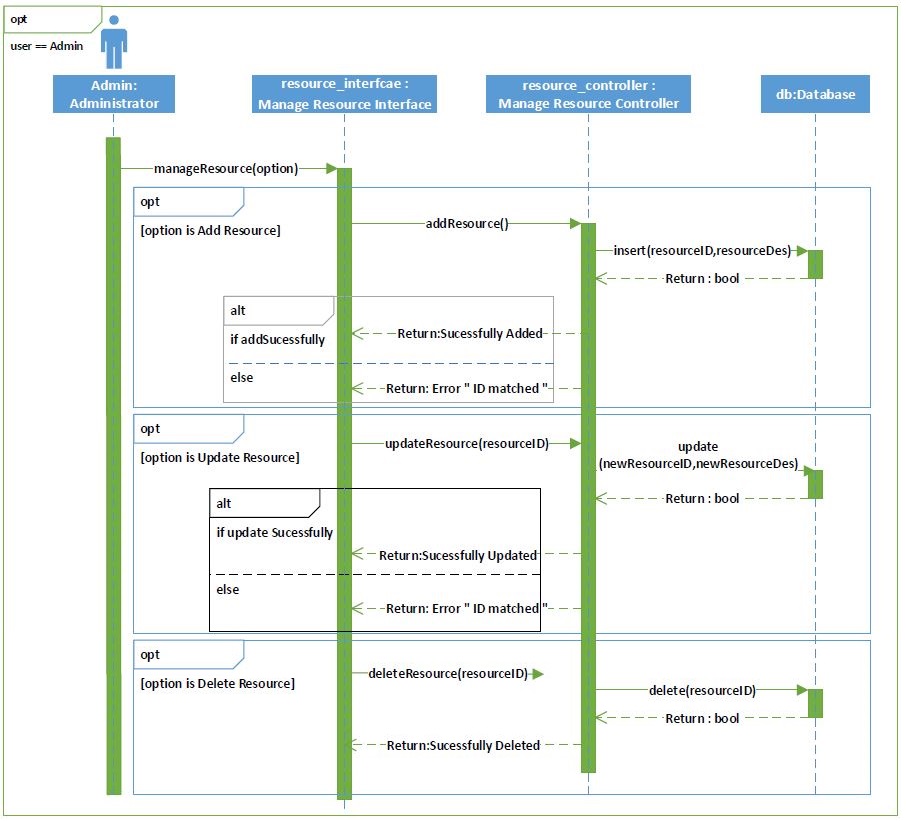


# Design Model

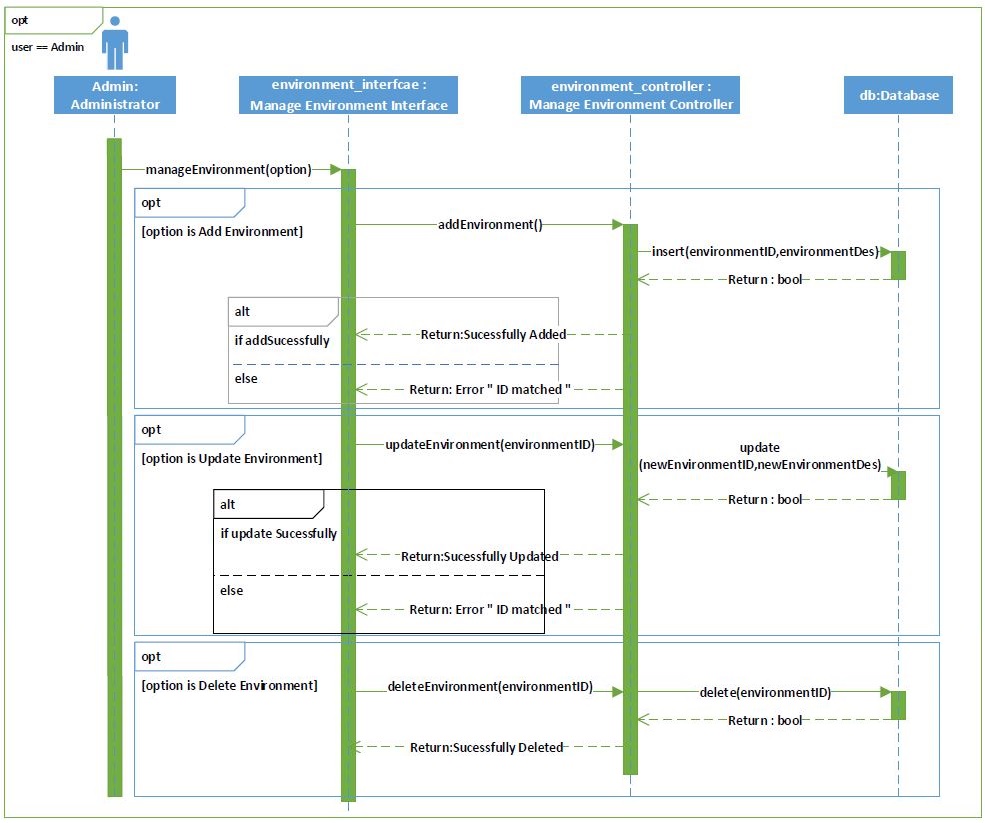
## Sequence Diagrams



### Database Population

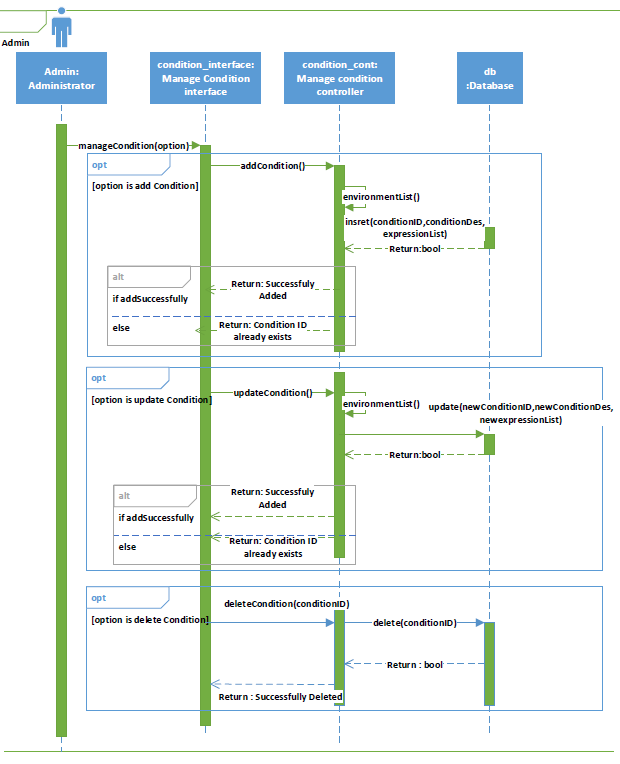
* + - 1. **Manage Subjects**
      2. **Manage Action**
      3. **Manage Resource**

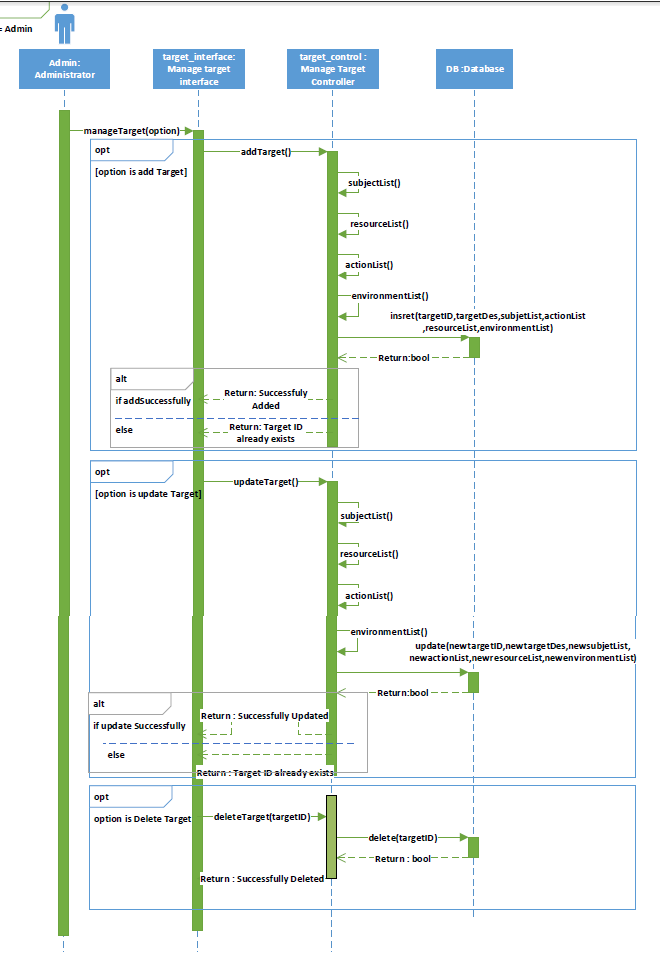
1. * + 3. **Manage Environment**

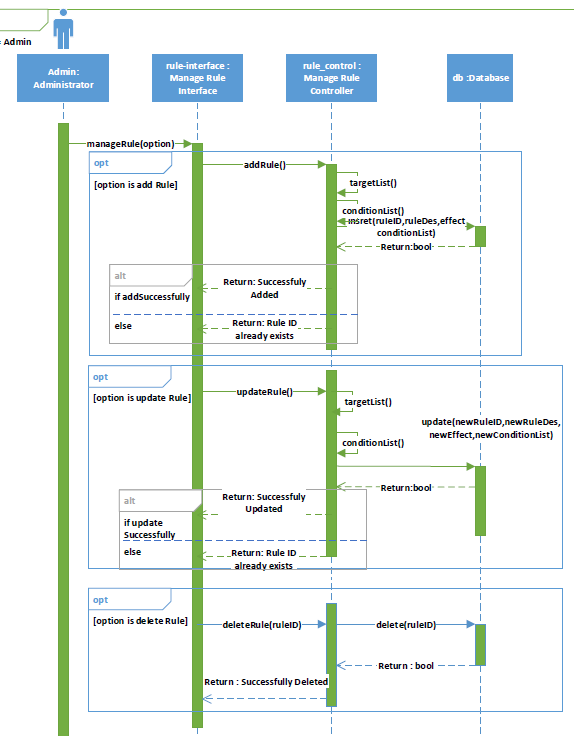


### Policy Creation

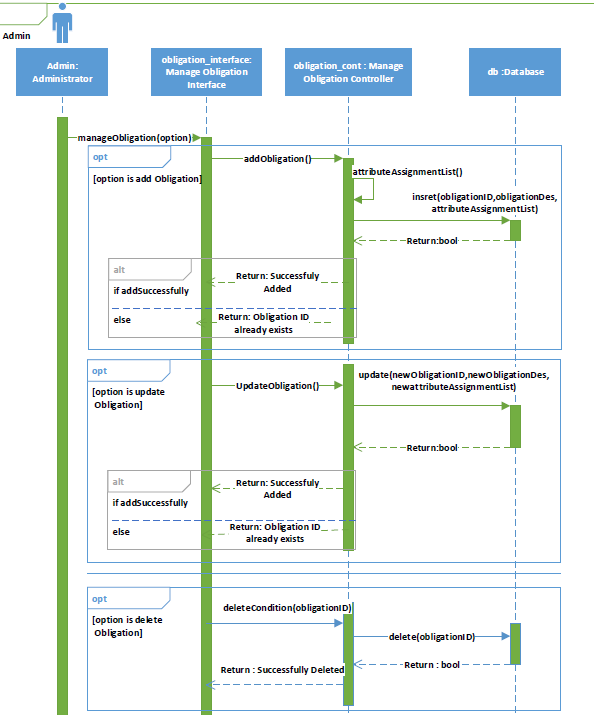
* + - 1. **Manage Condition**

****

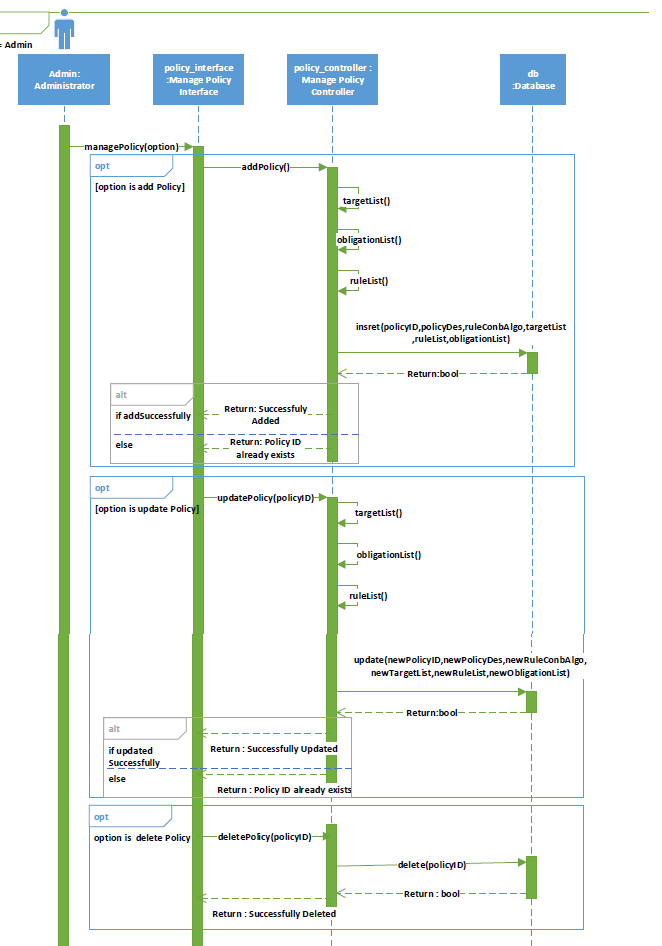
* + - 1. **Manage Target**
      2. **Manage Rule**

****

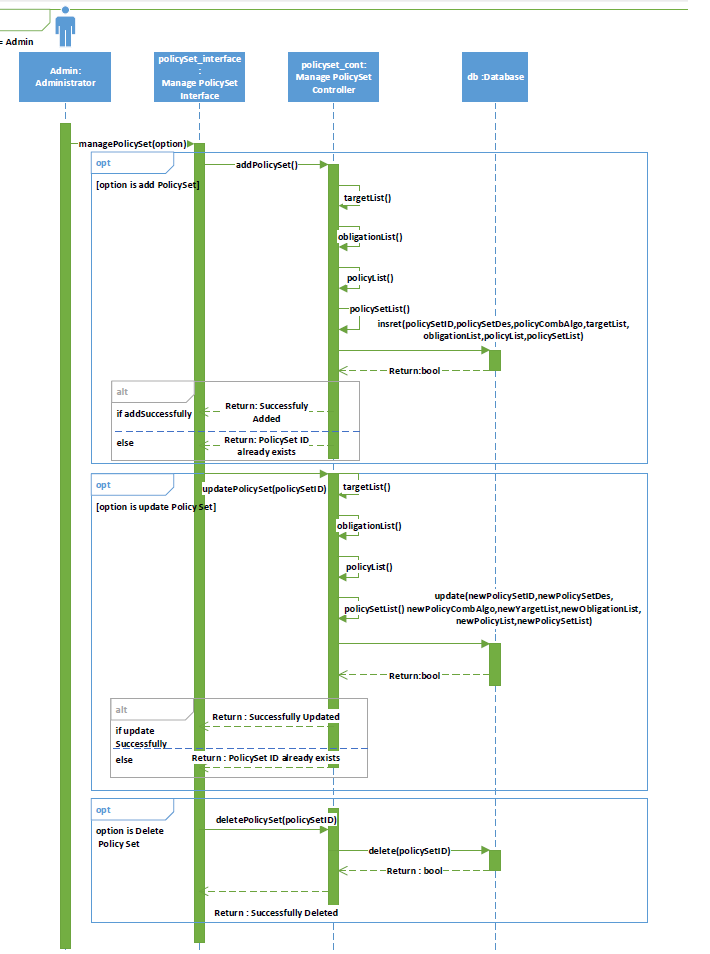
* + - 1. **Manage Obligation**



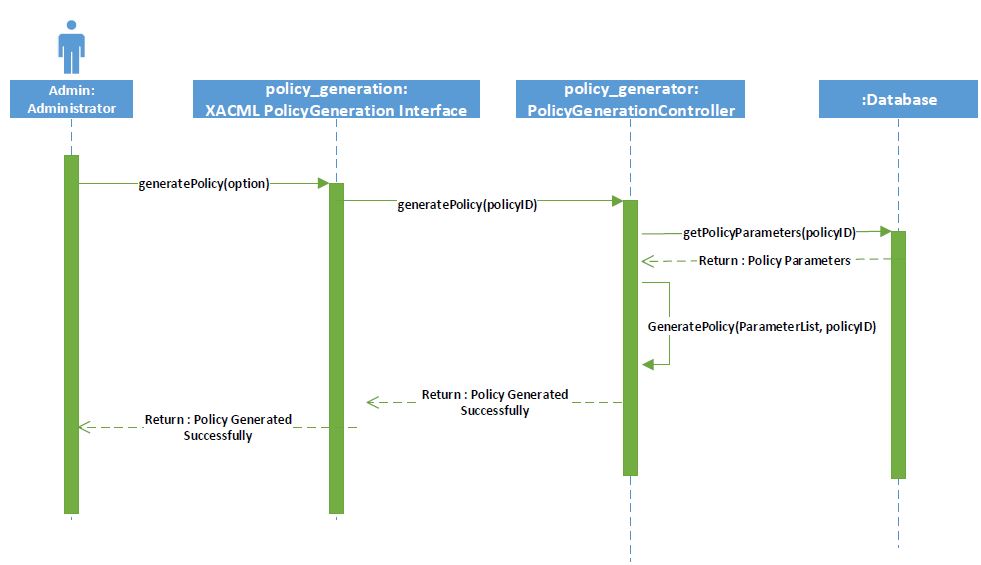
* + - 1. **Manage Policy**

****

* + - 1. **Manage Policy Set**

****

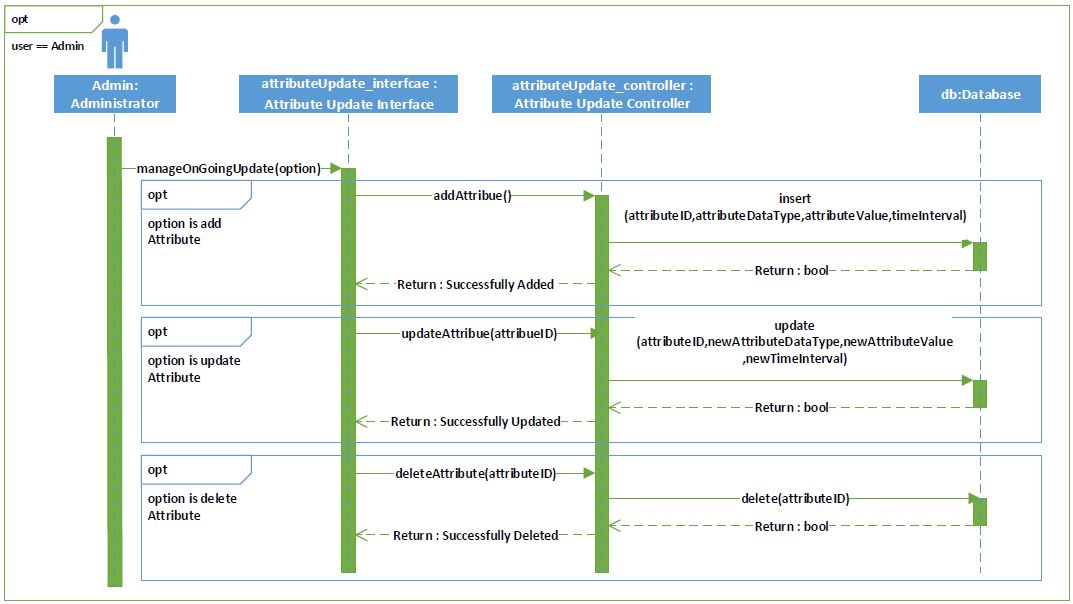
### Policy Generation



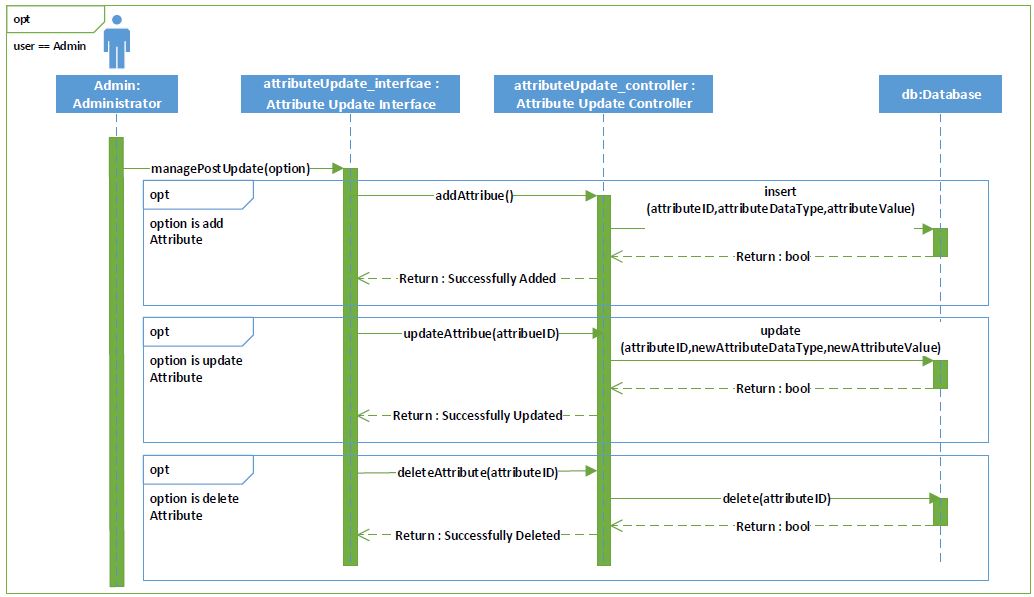


### Manage Pre-Updates

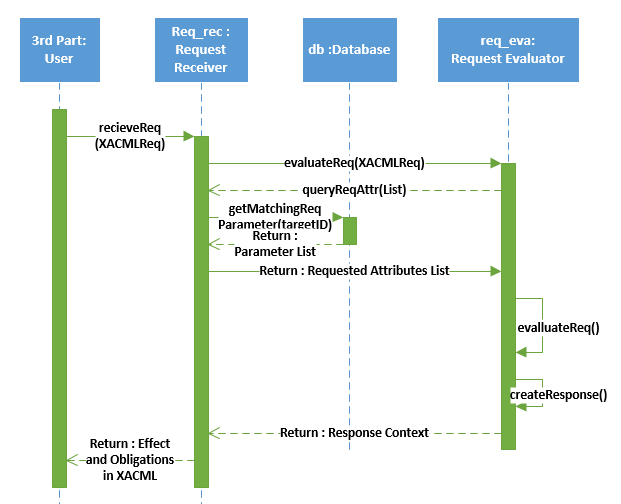
### Manage OnGoing-Updates



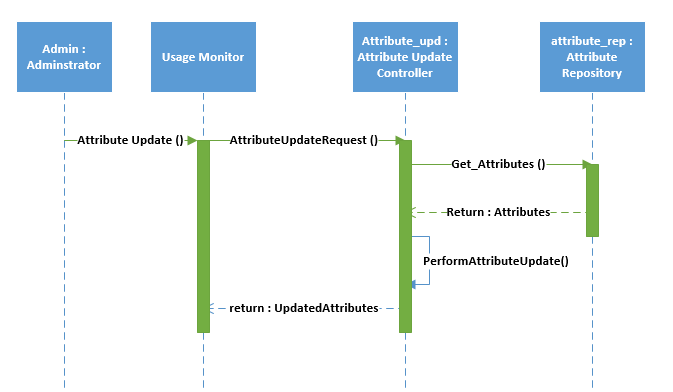
### Manage Post-Updates



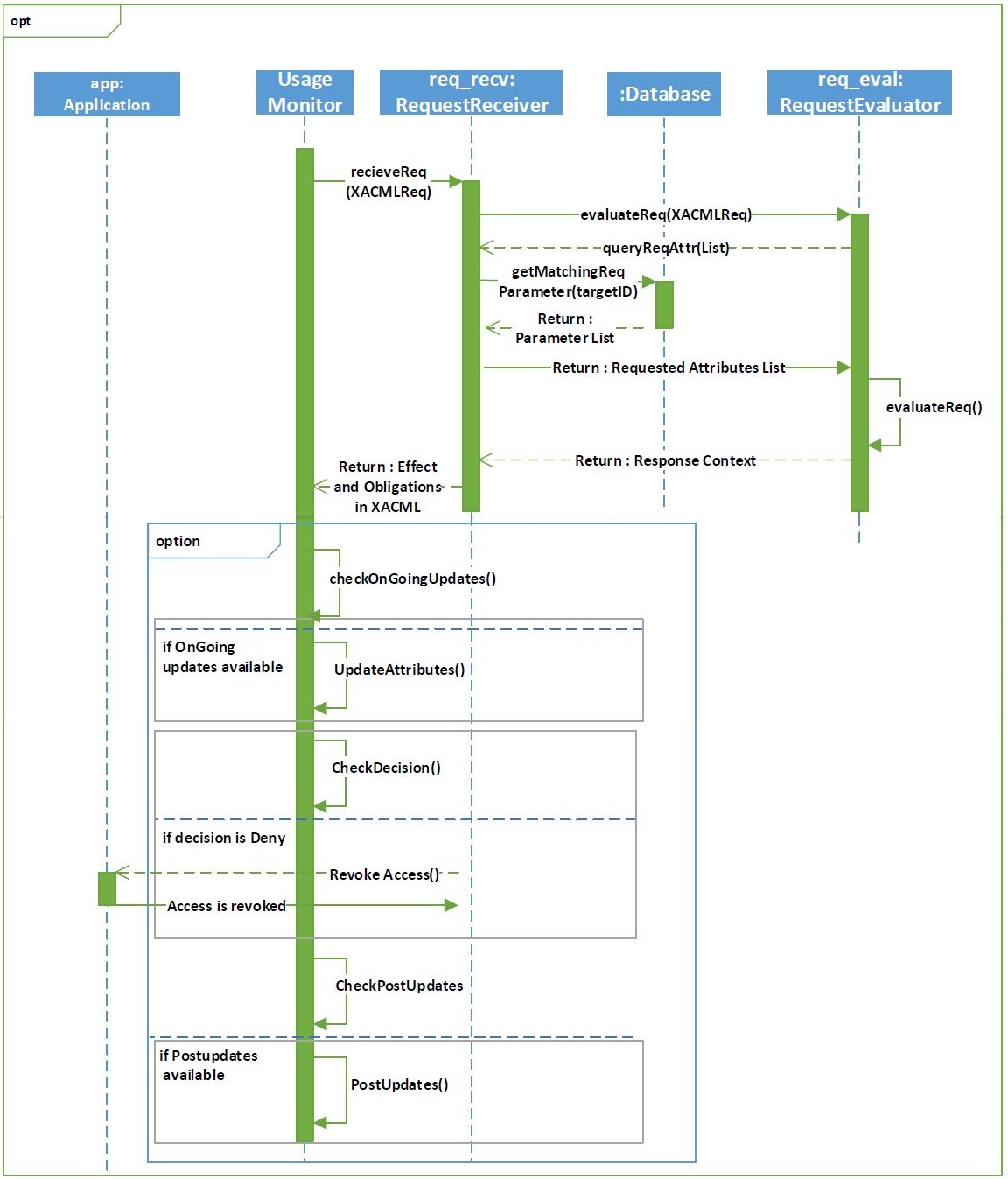
### Request Interception



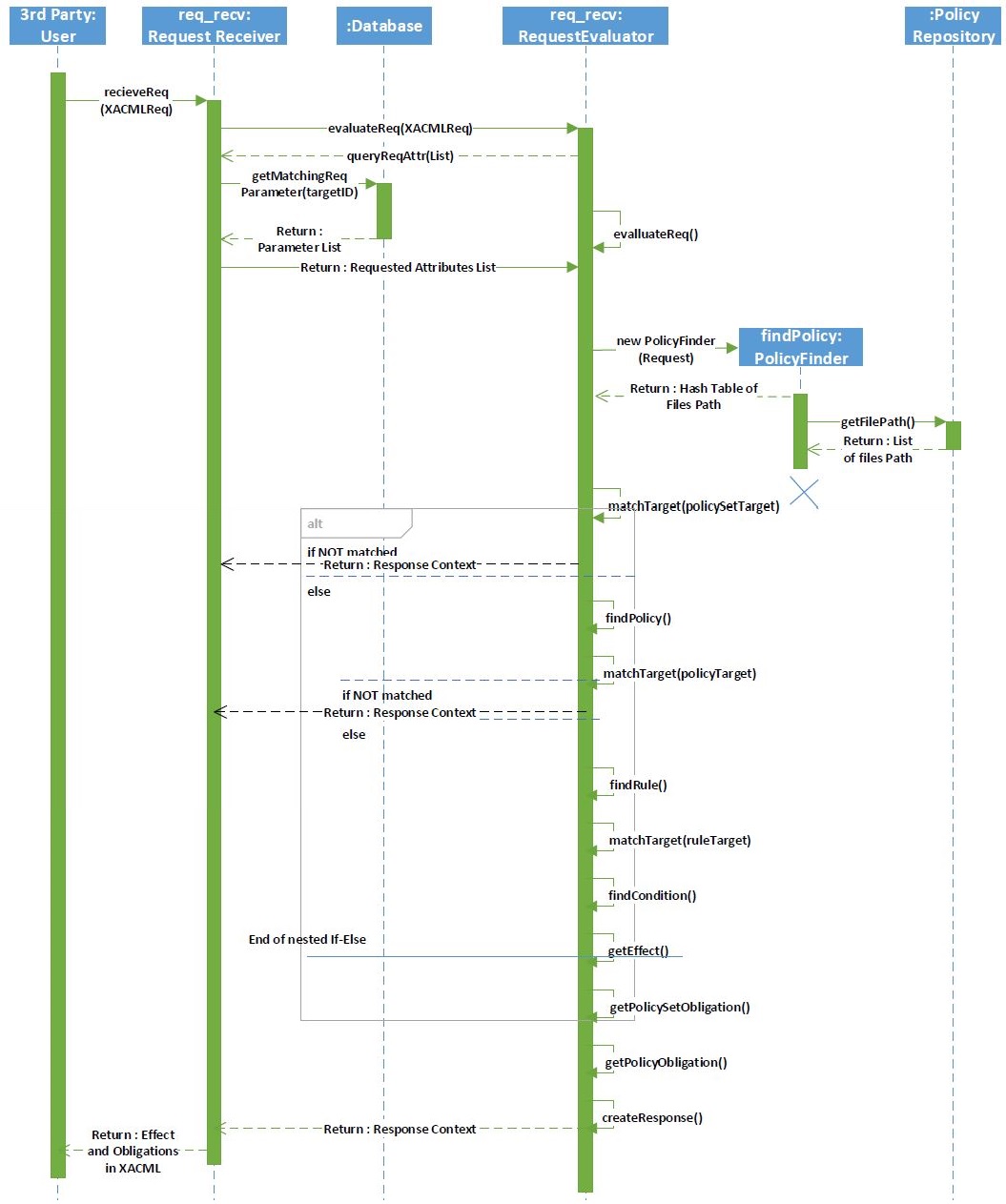
### Attribute Update



### Access Request Re-evaluation



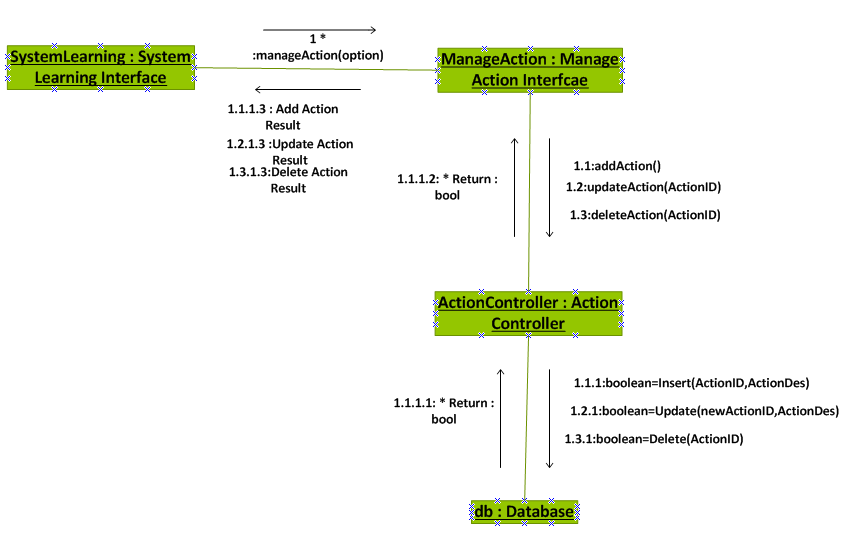
### Policy Evaluation



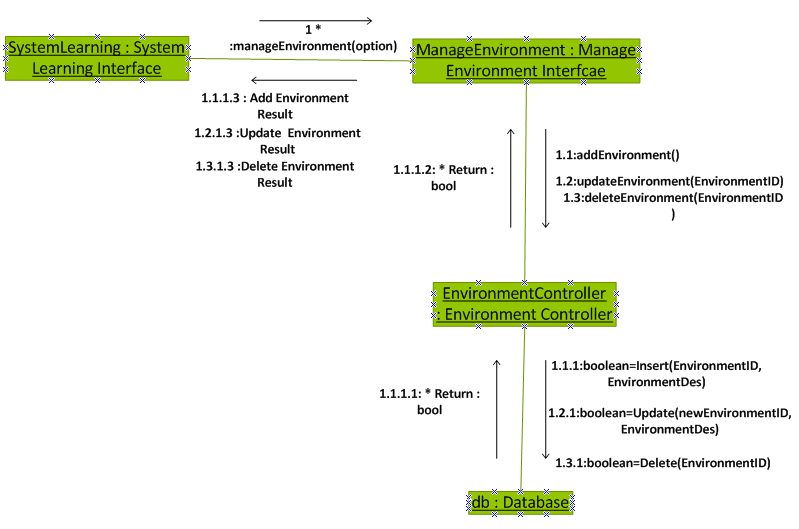
## Communication Diagrams

### Database Population

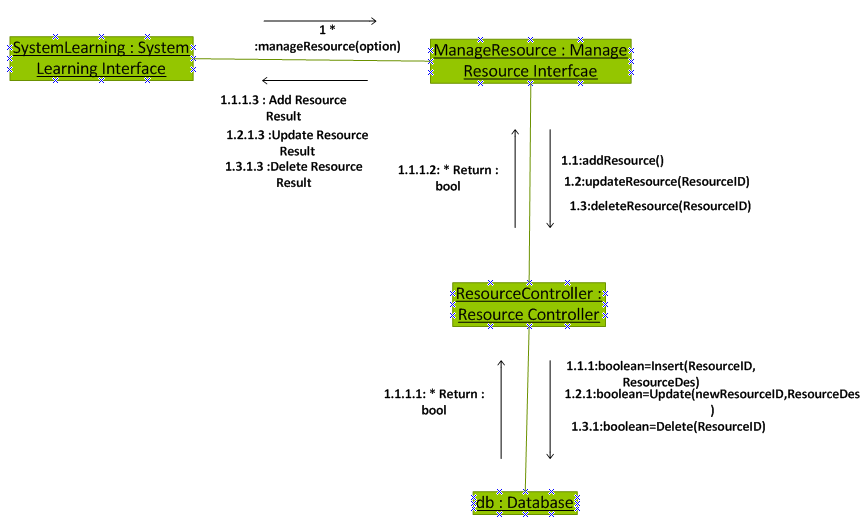
* + - 1. **Manage Action**



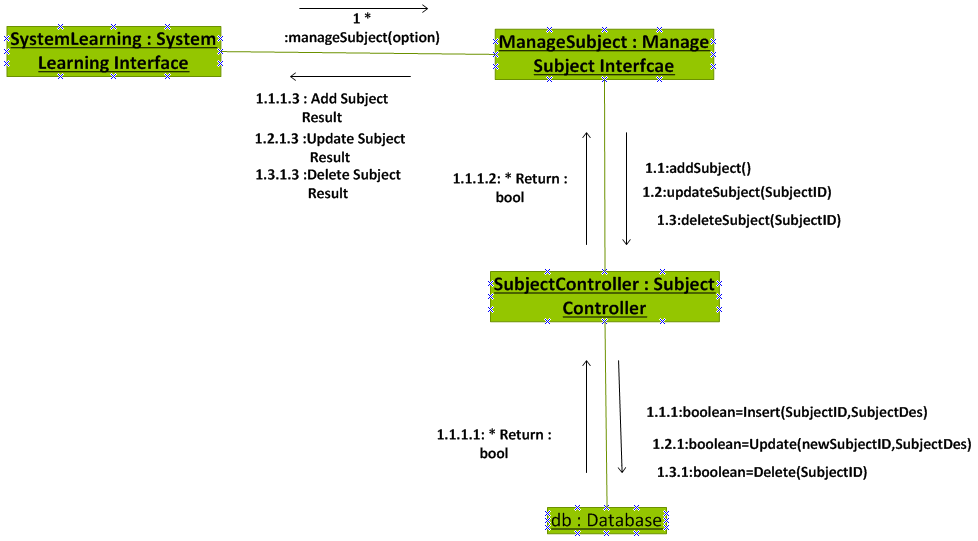
* + - 1. **Manage Environment**



* + - 1. **Manage Resource**

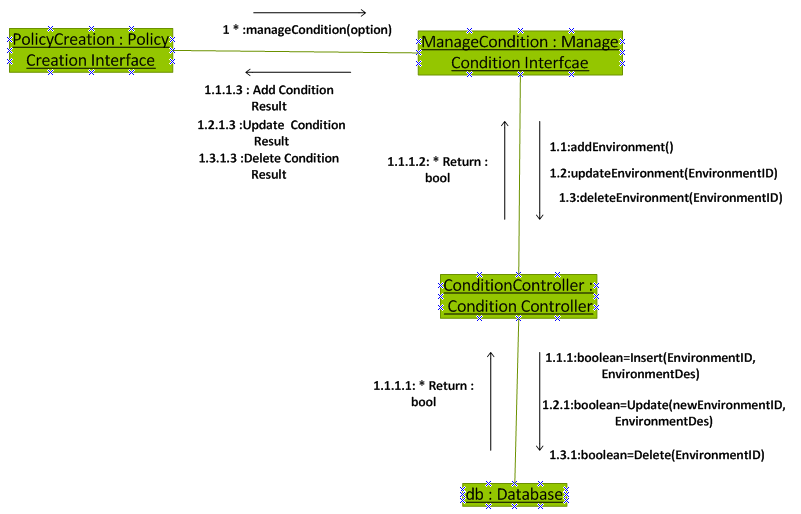


* + - 1. **Manage Subject**

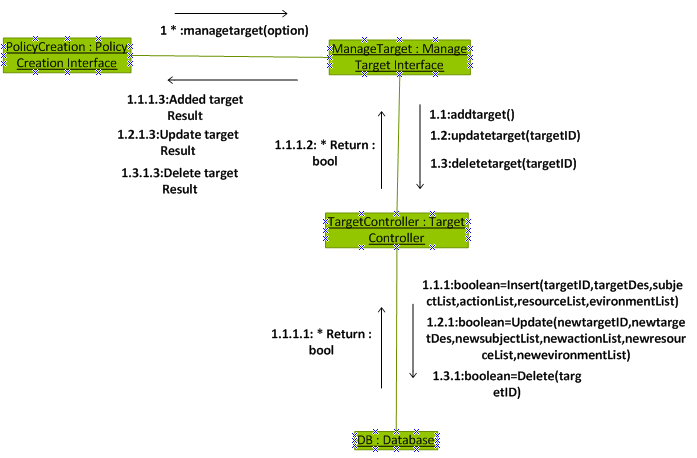


### Policy Creation

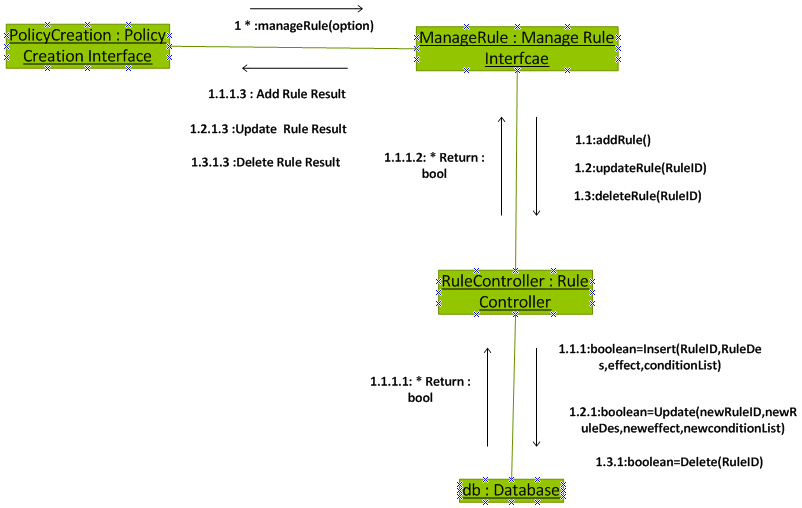
* + - 1. **Manage Condition**



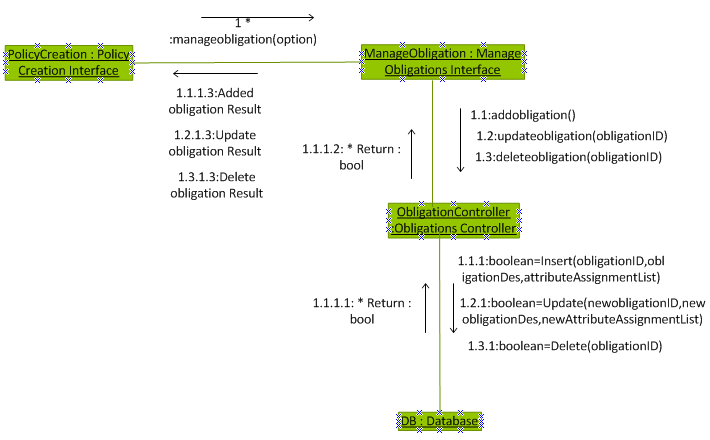
* + - 1. **Manage Target**



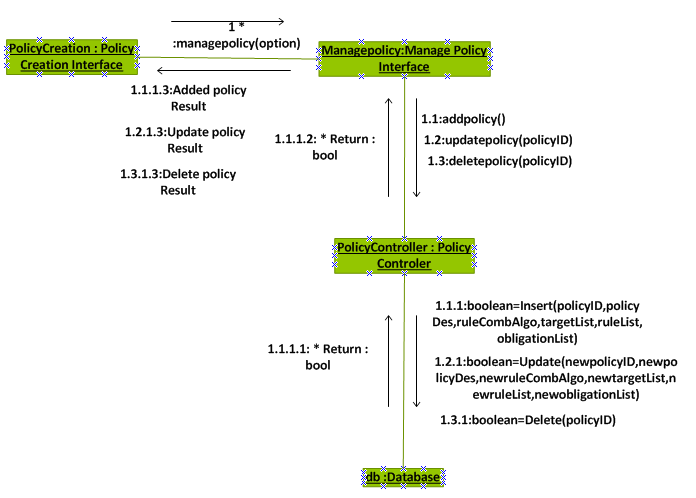
* + - 1. **Manage Rule**

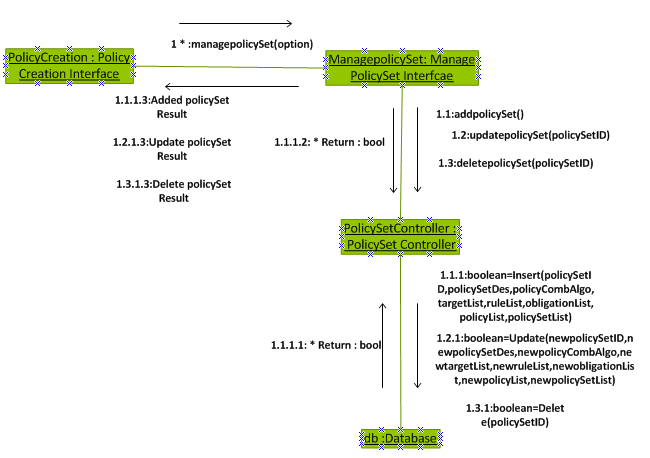


* + - 1. **Manage Obligation**

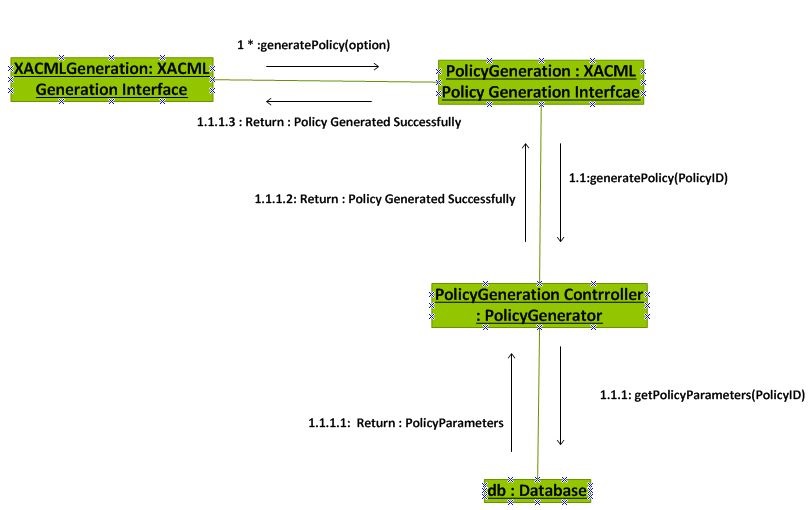


* + - 1. **Manage Policy**



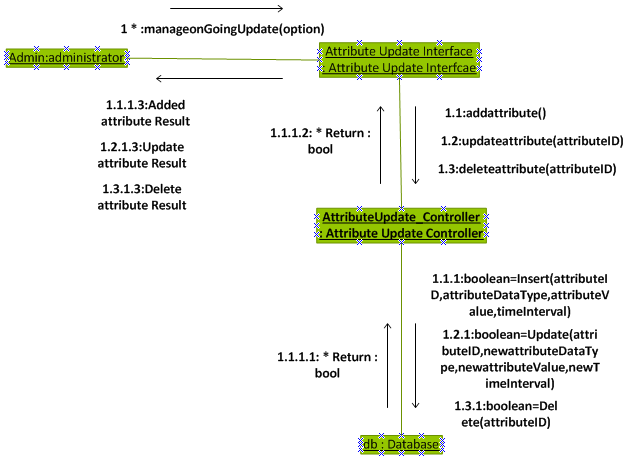
* + - 1. **Manage Policy Set**

### Policy Generation

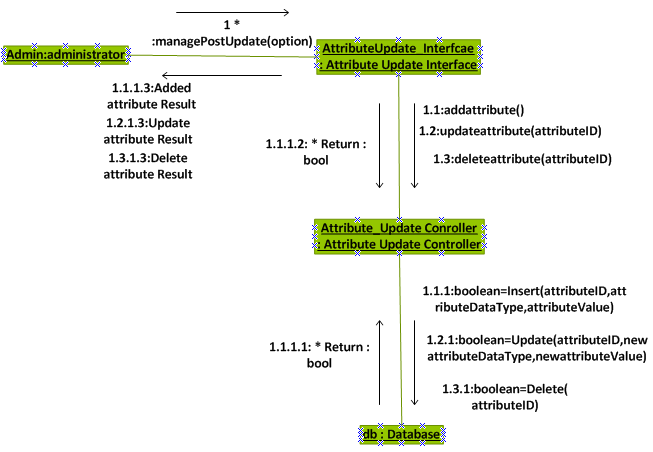


### Manage Pre-Updates

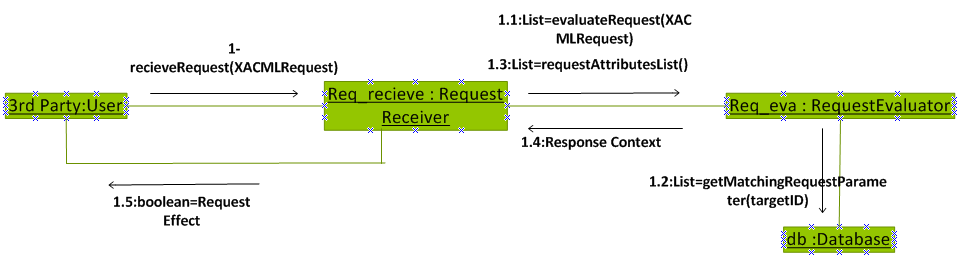
### Manage On-Updates



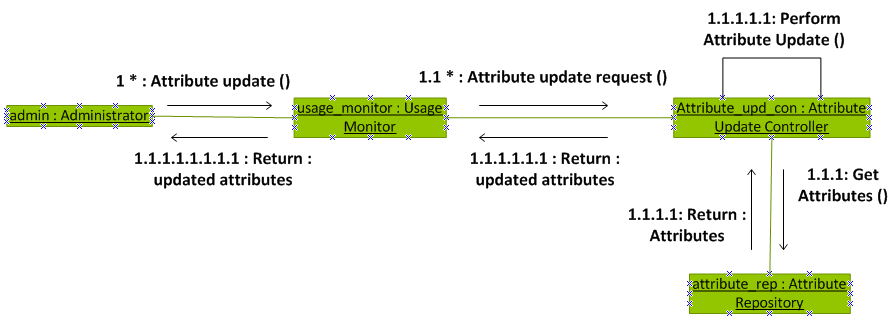
### Manage Post Updates



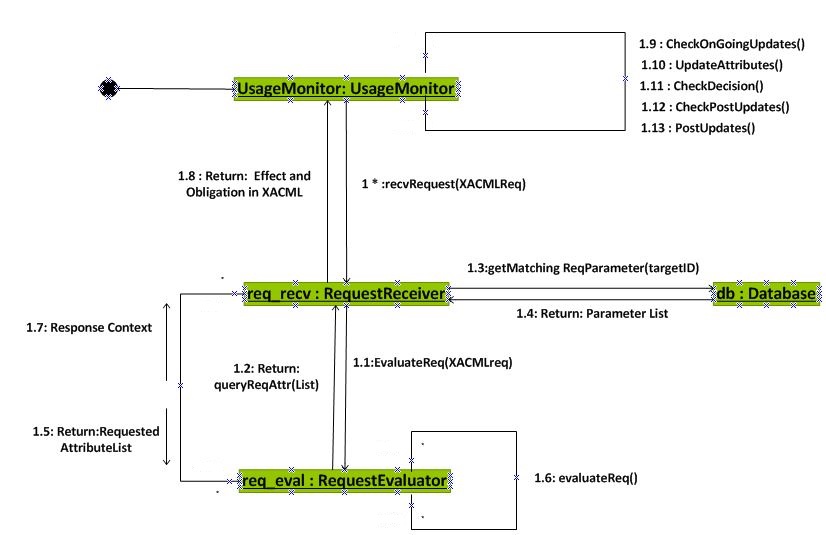
### Request Interception and Policy enforcement



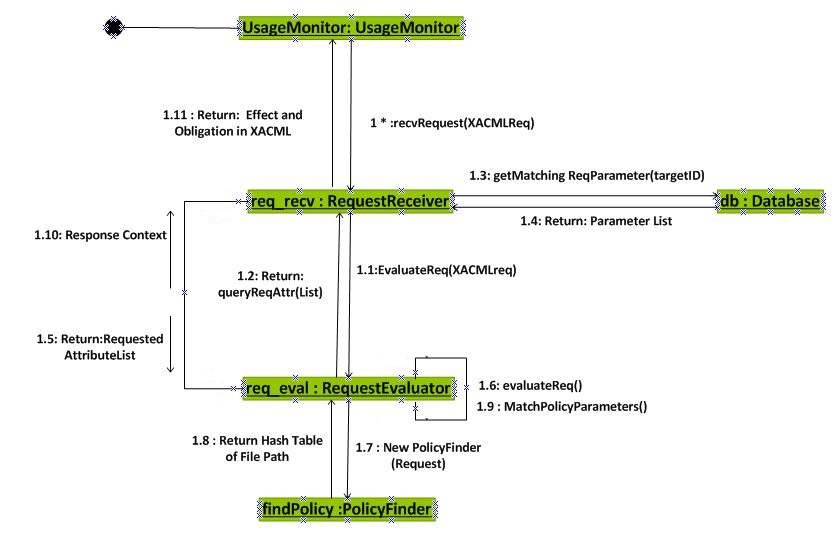
### Enforce Attribute Update

****

### Access Request Re-Evaluation



### Policy Evaluation



## Activity Diagrams

Put your activity diagrams here.

## Deployment Diagram

Put your deployment diagram here.

## Component Diagram

Put your Component Diagram here.

## Package Diagram

Put your package diagram here.