



# <UNDECIDABLES>

## COSBAS Functional Requirements Documentation

Git: <https://github.com/undecidables/Requirements-Documentation>

GitHub Organisation: <https://github.com/undecidables>

### **The Team:**

Elzahn Botha *13033922*  
Jason Richard Evans *13032608*  
Renette Ros *13007557*  
Szymon Ziolkowski *12007367*  
Tienie Pritchard *12056741*  
Vivian Venter *13238435*

**March 2015**

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Domain Model</b>	<b>3</b>
<b>3</b>	<b>Web Authentication</b>	<b>3</b>
<b>4</b>	<b>Appointments</b>	<b>3</b>
4.1	Approve Appointment . . . . .	3
4.1.1	Description . . . . .	3
4.1.2	Use Case diagram . . . . .	3
4.1.3	Pre-/Post-Conditions . . . . .	4
4.2	Disapprove Appointment . . . . .	4
4.2.1	Description . . . . .	4
4.2.2	Use Case diagram . . . . .	4
4.2.3	Pre-/Post-Conditions . . . . .	5
4.3	Process Specification for Approve and Disapprove Appointment . . . . .	5
4.4	Requesting Appointment . . . . .	5
4.4.1	Description . . . . .	5
4.4.2	Pre/Post conditions . . . . .	5
4.5	Cancelling Appointment . . . . .	6
4.5.1	Description . . . . .	6
4.5.2	Pre/Post conditions . . . . .	6
4.6	Use Case diagram for Requesting Appointment and Cancelling Appointment . .	7
4.7	Functionality for Requesting Appointment and Cancelling Appointment . . . . .	7
4.8	Process Specification for Requesting Appointment and Cancelling Appointment .	8
<b>5</b>	<b>Availability</b>	<b>8</b>
5.1	Use Case Diagram . . . . .	8
5.2	Module Functionality . . . . .	9
5.3	Choose Calendar . . . . .	9
5.3.1	Description . . . . .	9
5.3.2	Pre/Post conditions . . . . .	9
5.4	View Availability . . . . .	9
5.4.1	Description . . . . .	10
5.4.2	Pre/Post conditions . . . . .	10
5.5	Set Availability . . . . .	10
5.5.1	Description . . . . .	10
5.5.2	Pre/Post conditions . . . . .	10
<b>6</b>	<b>Biometric Access</b>	<b>10</b>
6.1	Request Access . . . . .	11
6.1.1	Use Case diagram . . . . .	11
6.1.2	Pre-/Post-Conditions . . . . .	12
6.1.3	Process Specification . . . . .	12
6.2	Register User . . . . .	12
6.2.1	Description . . . . .	12
6.2.2	Use Case diagram . . . . .	13
6.2.3	Pre-/Post-Conditions . . . . .	13
6.2.4	Process Specification . . . . .	14

<b>7</b>	<b>Reporting</b>	<b>14</b>
7.1	Get Visitor Access Exit Times Report . . . . .	14
7.1.1	Description . . . . .	14
7.1.2	Service Contract . . . . .	14
7.1.3	Pre/Post Conditions . . . . .	14
7.2	Get Staff Access Exit Times Report . . . . .	15
7.2.1	Description . . . . .	15
7.2.2	Service Contract . . . . .	15
7.2.3	Pre/Post Conditions . . . . .	15
7.3	Get Not Honoured Appointments Report . . . . .	15
7.3.1	Description . . . . .	16
7.3.2	Service Contract . . . . .	16
7.3.3	Pre/Post Conditions . . . . .	16
7.4	Generate Custom Reports . . . . .	16
7.4.1	Description . . . . .	16
7.4.2	Service Contract . . . . .	17
7.4.3	Pre/Post Conditions . . . . .	17
7.5	Export Report To Specific Format . . . . .	17
7.5.1	Description . . . . .	17
7.5.2	Service Contract . . . . .	17
<b>8</b>	<b>Information Portal</b>	<b>18</b>
8.1	Create Post . . . . .	18
8.1.1	Description . . . . .	18
8.1.2	Functional Requirement . . . . .	19
8.1.3	Service Contract . . . . .	19
8.1.4	Pre/Post Conditions . . . . .	19
8.2	Get Posts . . . . .	20
8.2.1	Description . . . . .	20
8.2.2	Service Contract . . . . .	20
8.3	Delete Post . . . . .	20
8.3.1	Description . . . . .	20
8.3.2	Pre/Post Conditions . . . . .	20
8.4	Edit Post . . . . .	21
8.4.1	Description . . . . .	21
8.4.2	Pre/Post Conditions . . . . .	21
<b>9</b>	<b>Authentication</b>	<b>22</b>
9.1	Authenticate . . . . .	22
9.1.1	Description . . . . .	22
9.1.2	Pre-/Post-Conditions . . . . .	22
9.1.3	Functionality . . . . .	23
9.1.4	Process Specification . . . . .	24
9.2	Log In . . . . .	24
9.2.1	Description . . . . .	24
9.2.2	Pre-/Post-Conditions . . . . .	24
9.3	Log Out . . . . .	24
9.3.1	Description . . . . .	25
9.3.2	Pre-/Post-Conditions . . . . .	25

# 1 Introduction

In this section of the document we will identify and address a high level overview of the COSBAS (Computer Science Biometric Access System) system.

## 2 Domain Model

## 3 Web Authentication

## 4 Appointments

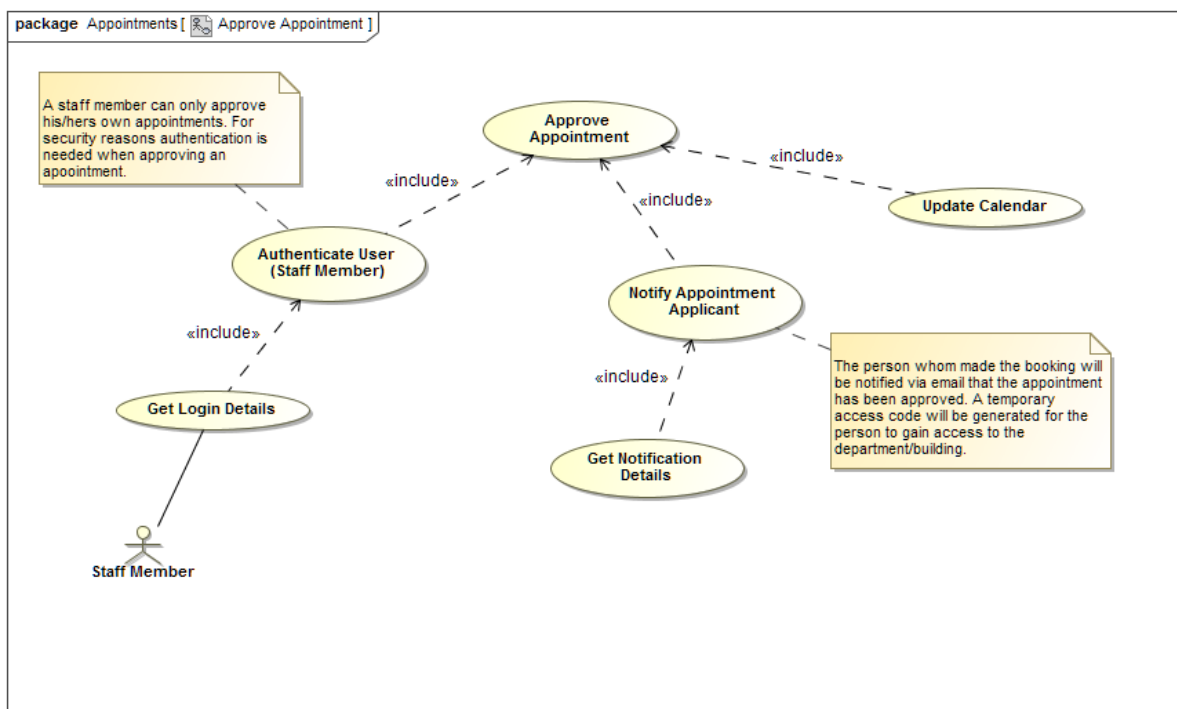
### 4.1 Approve Appointment

*Priority: Important*

#### 4.1.1 Description

A person can book for an appointment with one of the staff members in the department/building. The particular staff member needs to approve such an appointment. A booking needs to be approved by a staff member before the guest/visitor can gain access to the department/building.

#### 4.1.2 Use Case diagram



### 4.1.3 Pre-/Post-Conditions

#### Pre-conditions:

- Staff member needs to be logged in to approve his/her own appointments
- There has to be an appointment booked before it can be approved.

#### Post-conditions:

- The staff member will be booked for that time period, that is the staff member will be unavailable during that time.
- A notification with an temporary access code is sent to the user whom booked for the appointment

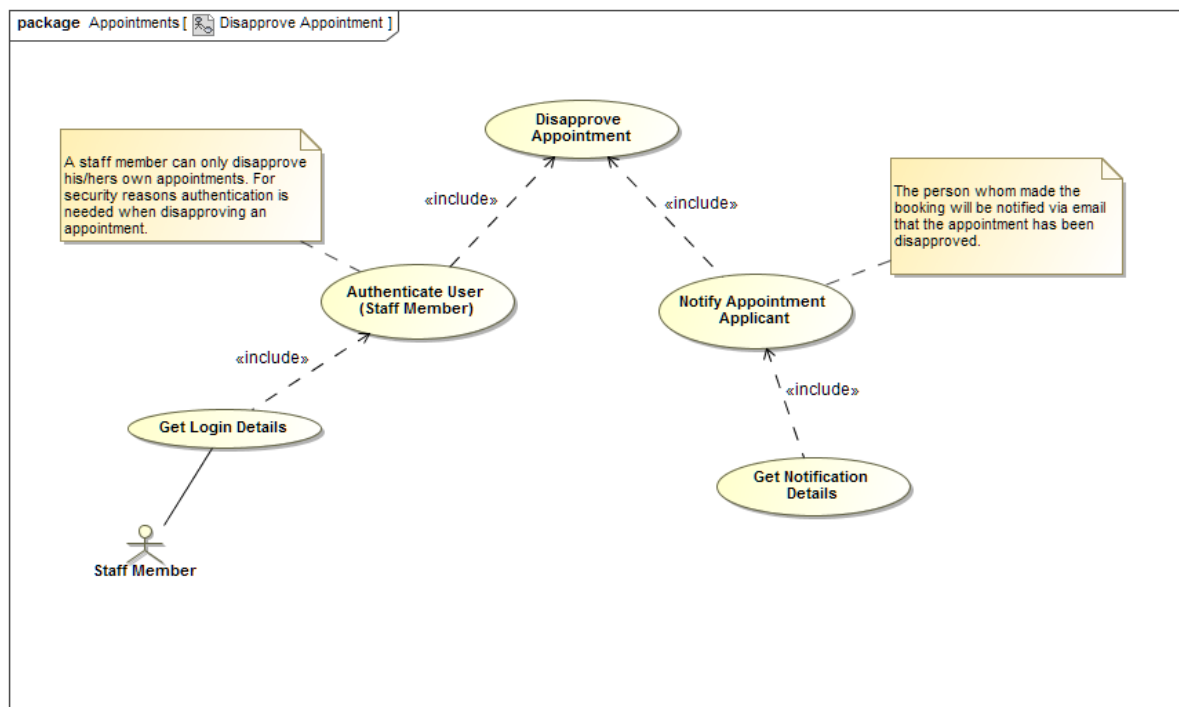
## 4.2 Disapprove Appointment

*Priority: Important*

### 4.2.1 Description

An appointment can be disapproved by a staff member. The appointment application will be notified that the appointment has been disapproved.

### 4.2.2 Use Case diagram



### 4.2.3 Pre-/Post-Conditions

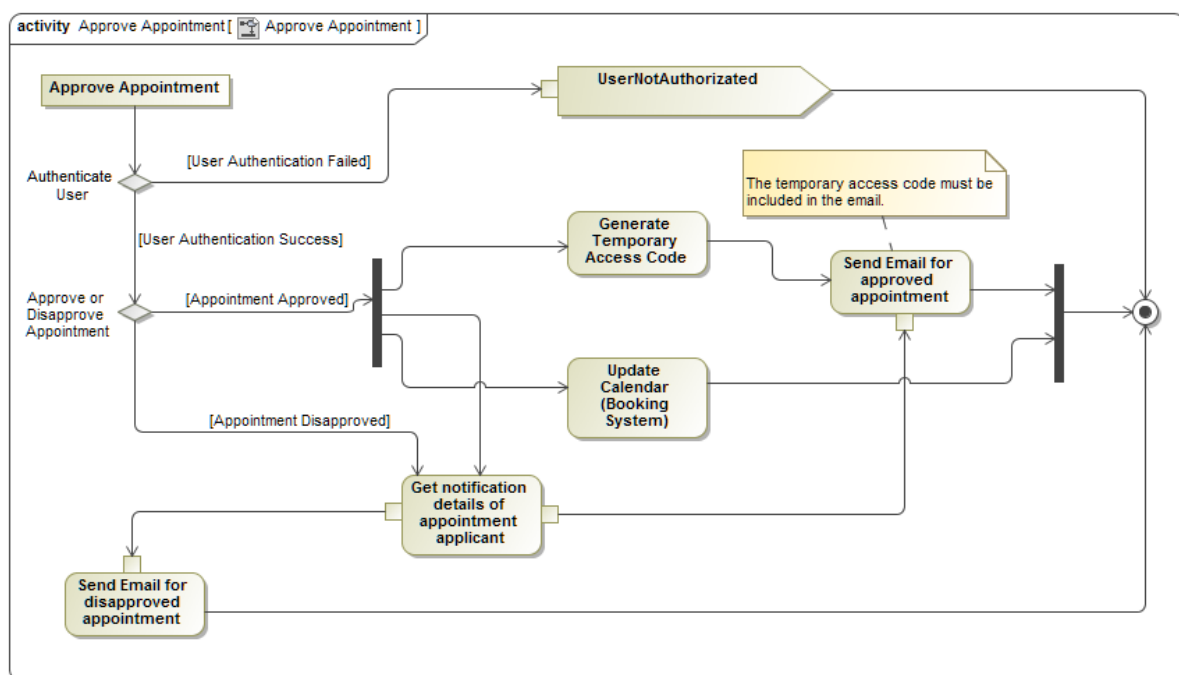
#### Pre-conditions:

- Staff member needs to be logged in to disapprove his/her own appointments
- There has to be an appointment booked before it can be disapproved.

#### Post-conditions:

- A notification informing the appointment applicant that the appointment has been disapproved.

### 4.3 Process Specification for Approve and Disapprove Appointment



### 4.4 Requesting Appointment

Priority: *Important*

#### 4.4.1 Description

The requestAppointment function allows a user of the system to request an appointment with a staff member that is also making use of the system.

#### 4.4.2 Pre/Post conditions

##### Pre-conditions:

- Lecturer must exist.
- Date and time of the requested appointment must be valid entries.

**Post-conditions:**

- Appointment will be saved for the lecturer to approve or deny later on.
- User gets an appointment identifier back.
- Lecturer is notified of the requested appointment.

## 4.5 Cancelling Appointment

*Priority: Important*

### 4.5.1 Description

The `cancelAppointment` function allows the user to cancel an appointment if the user is either the staff member who's appointment it is or the user who made the appointment.

### 4.5.2 Pre/Post conditions

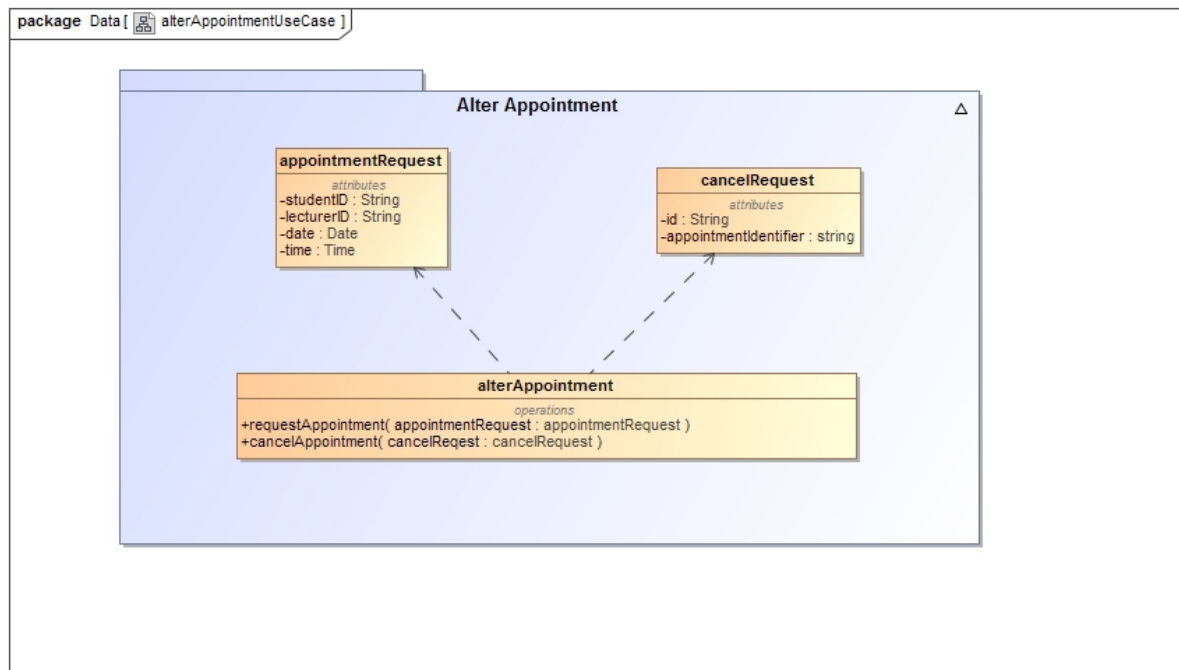
**Pre-conditions:**

- The appointment must exist.
- The user cancelling the appointment has to be the person that the appointment is with or the person who made.

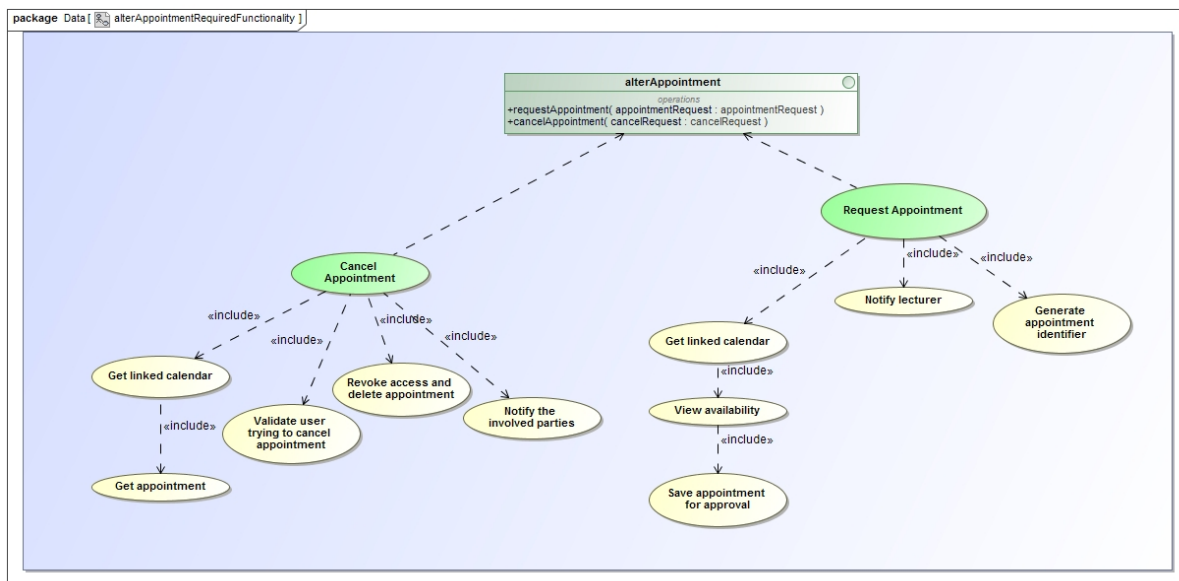
**Post-conditions:**

- The appointment will be cancelled.
- Both parties are notified.
- Access that was granted for the appointment is revoked.

#### 4.6 Use Case diagram for Requesting Appointment and Cancelling Appointment

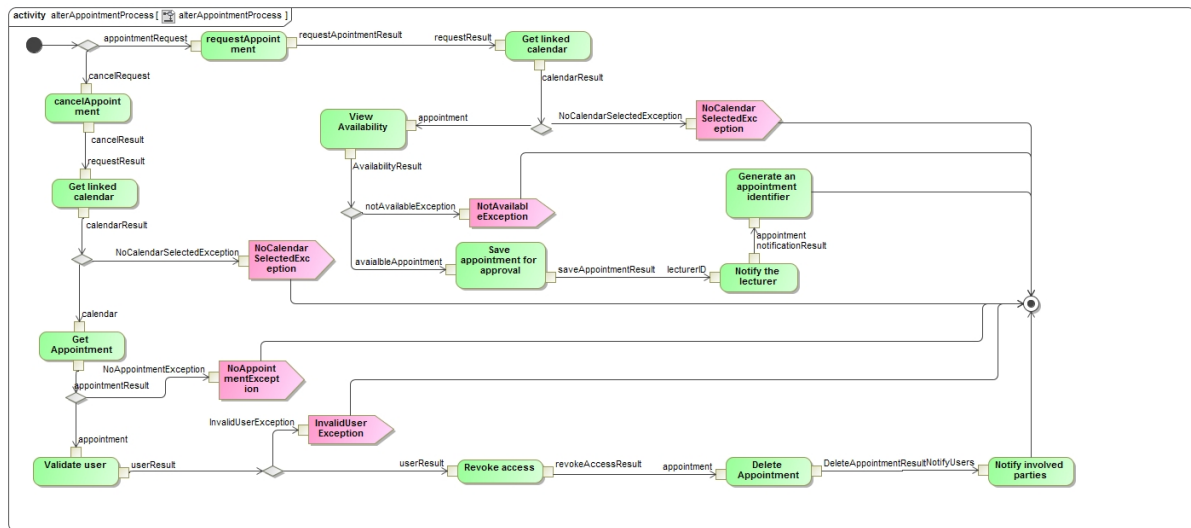


#### 4.7 Functionality for Requesting Appointment and Cancelling Appointment





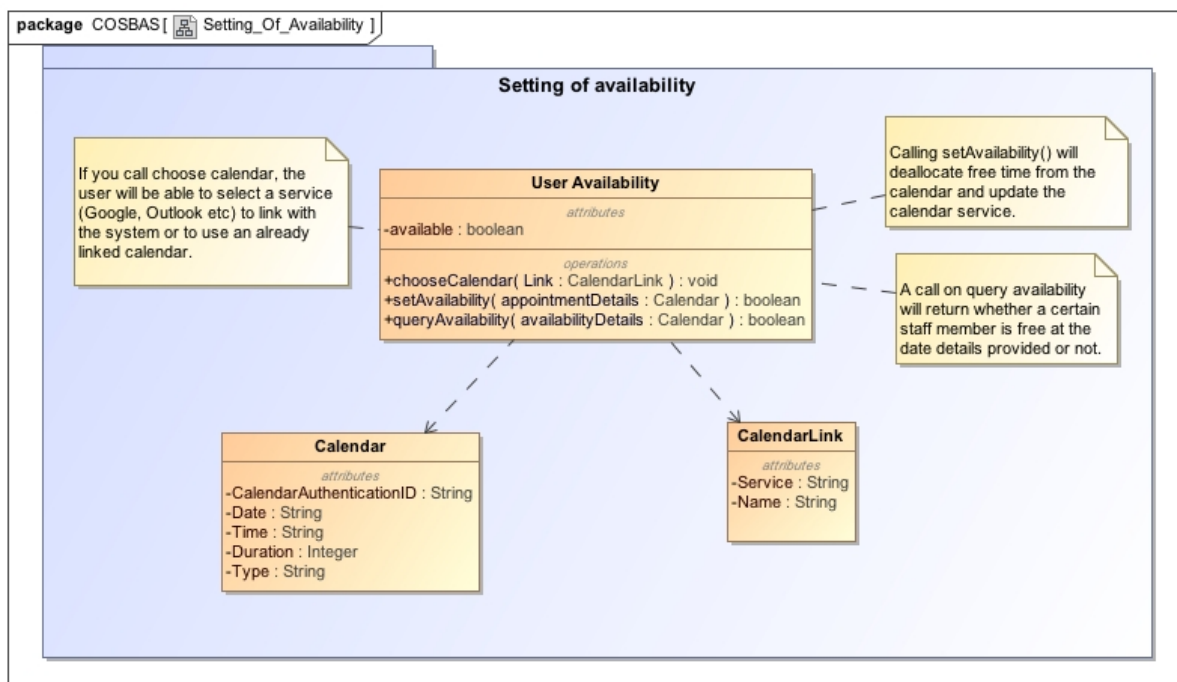
#### 4.8 Process Specification for Requesting Appointment and Cancelling Appointment



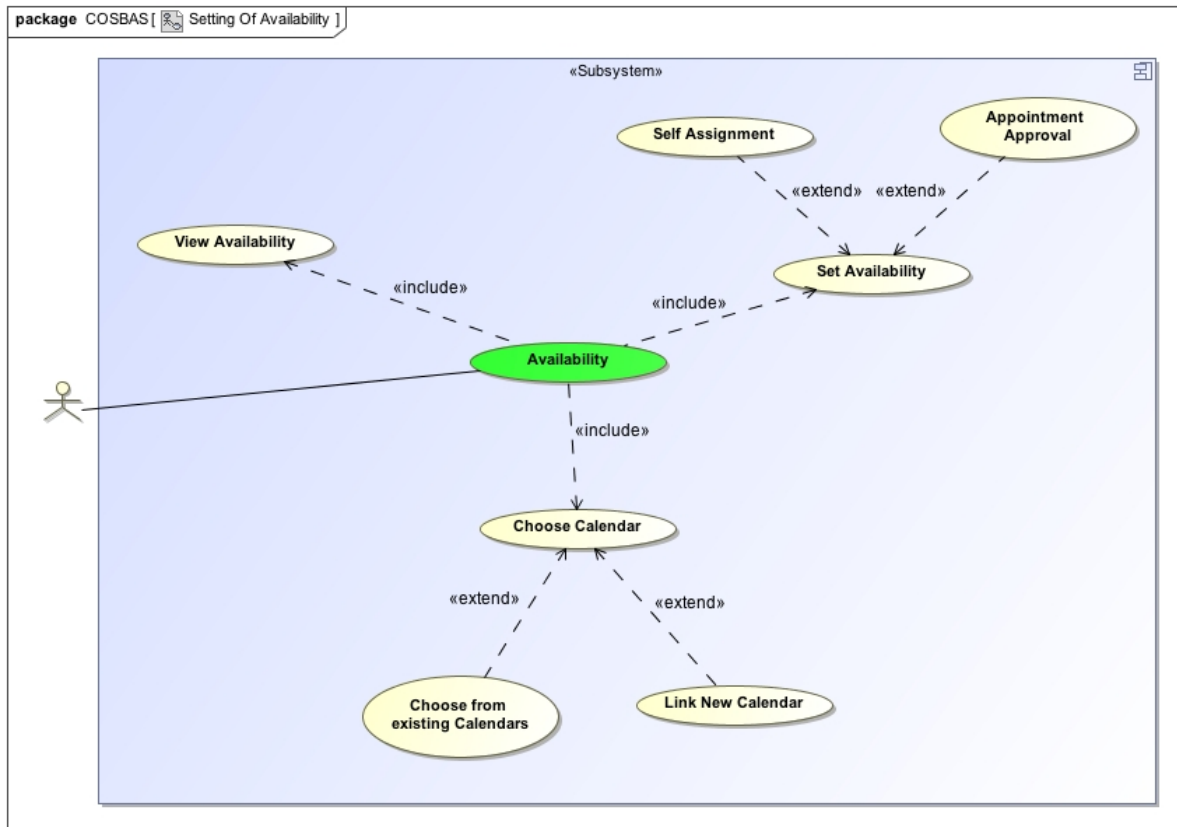
## 5 Availability

The following functionality of creating and setting will be available to only signed in users. Users who are not authorised or signed into the system will only be granted read-only privileges.

## 5.1 Use Case Diagram



## 5.2 Module Functionality



## 5.3 Choose Calendar

Priority: *Important*

### 5.3.1 Description

An authorised user of the system should be able to select which calendar account to associate with the system. If the user chose a linked service, then that service needs to be authenticated.

### 5.3.2 Pre/Post conditions

**Pre-conditions:** The user should be logged into the system and already have a calendar account from the linked services (Google & Outlook).

**Post-conditions:** Once the user has chosen the calendar to associate with the system, all updates in availability will be amended in that specific calendar.

## 5.4 View Availability

Priority: *Important*

### 5.4.1 Description

Both the authorised user and a non-authorised user (or guest) should be able to have a read-only functionality of viewing the availability of the associated staff member on their calender.

### 5.4.2 Pre/Post conditions

**Pre-conditions:** An associated calendar for the staff member should exist.

**Post-conditions:** Be able to see whether the associated staff member has an appointment at a certain date and time or if they are free for appointments.

## 5.5 Set Availability

*Priority: Important*

### 5.5.1 Description

The authorised user should be able to set whether s/he is available at a certain date or time or if they have an appointment at a specific date or time. This can be set by the staff member himself/herself or by the approval of an appointment.

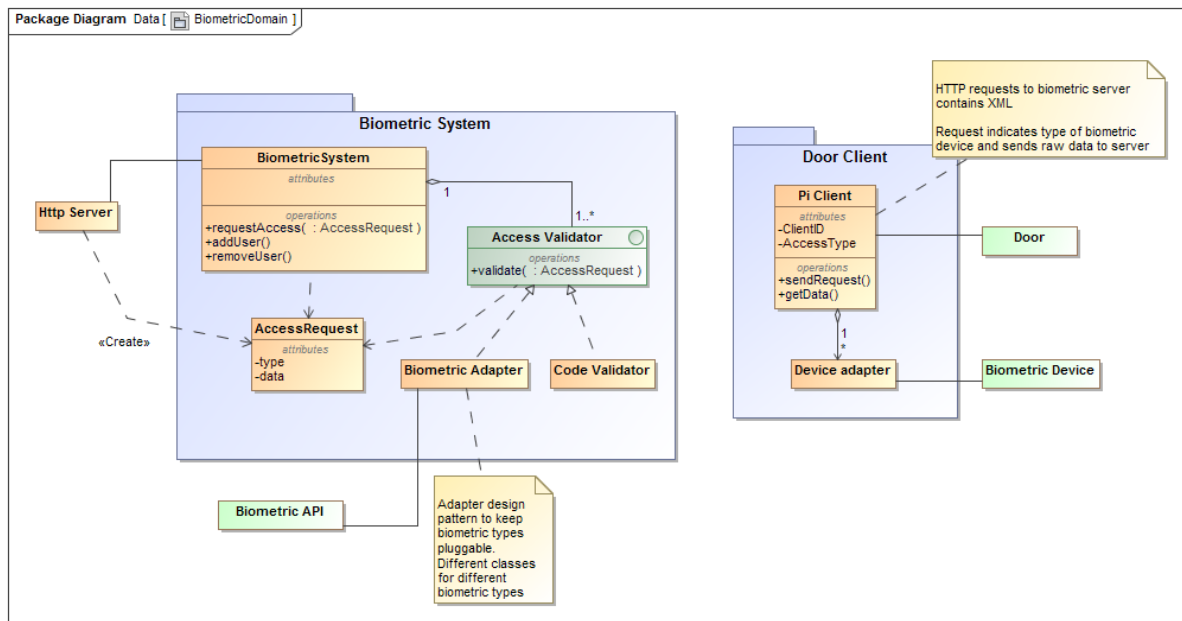
### 5.5.2 Pre/Post conditions

**Pre-conditions:** The user should be authorised and logged in and a calendar should exist.

**Post-conditions:** The linked calendar should be updated with the particulars specified by the authorised user. (CRUD of appointments).

## 6 Biometric Access

This module encapsulates functionality regarding validating biometric data or temporary access code to give staff members or visitors access to the Computer Science Department



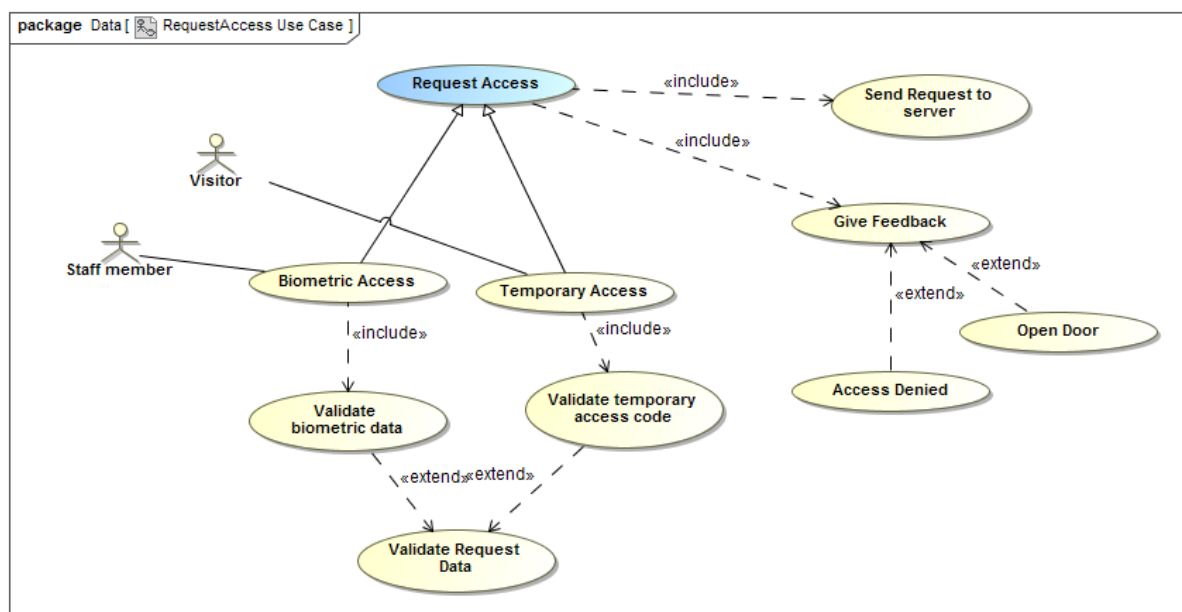
## 6.1 Request Access

Priority: *Critical*

**Staff Access** A staff member can gain access at a door using biometrics.

**Visitor Access** A visitor can enter the department by entering their temporary access code at the door.

### 6.1.1 Use Case diagram



### 6.1.2 Pre-/Post-Conditions

#### Staff Access Pre-conditions:

- Staff member must be registered on system.
- Biometric data must validate correctly

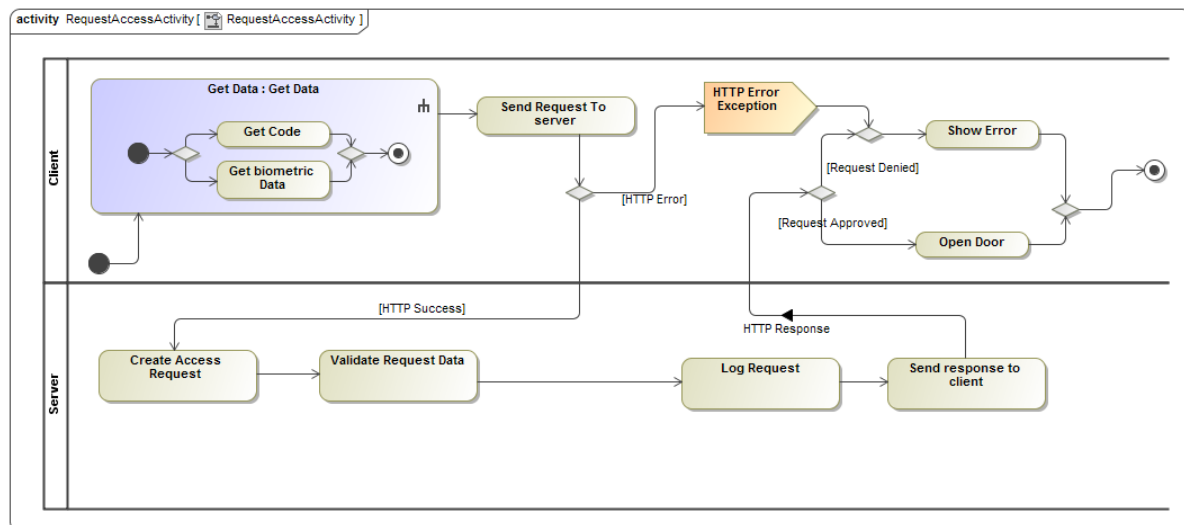
#### Visitor Access Pre-conditions:

- Visitor needs to have an appointment
- Temporary access code should be valid

#### Post-conditions:

- User gains access
- Access logged

### 6.1.3 Process Specification



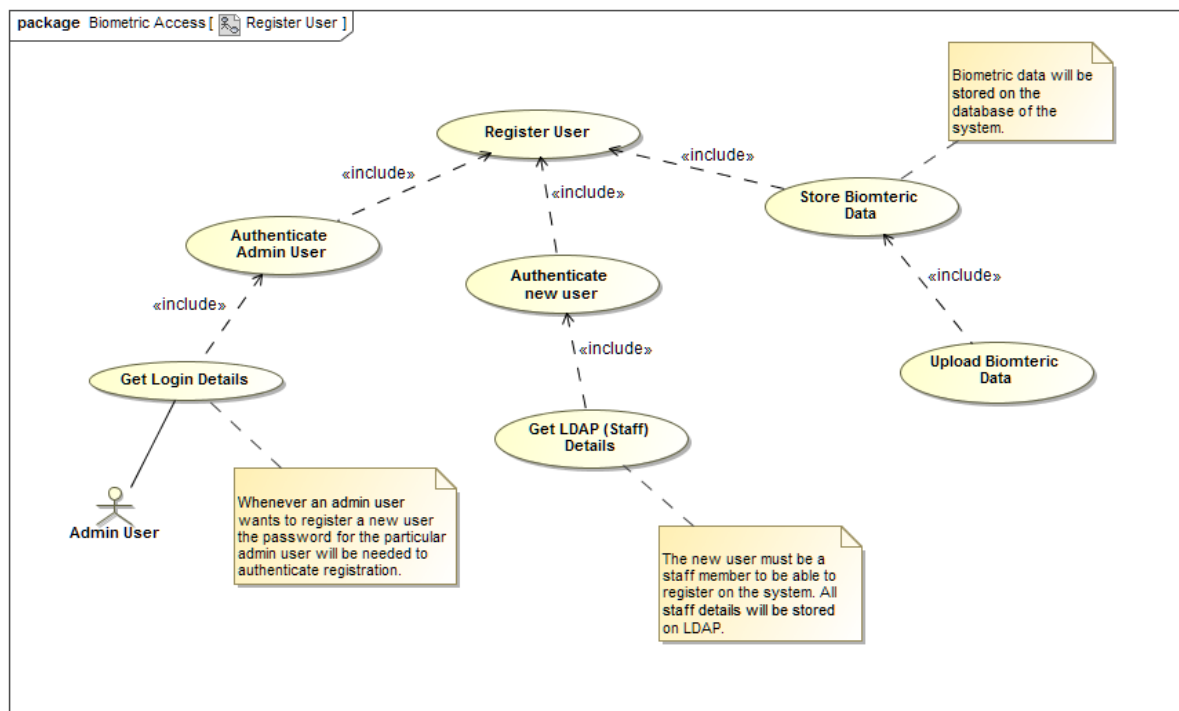
## 6.2 Register User

*Priority: Critical*

### 6.2.1 Description

To register a user on the system is the same as to store biometric data of the user on the database. Staff members need to store their biometric data to gain access to department/building. There will be an administrator(admin user) that will handle all the registering of users on the system.

## 6.2.2 Use Case diagram



## 6.2.3 Pre-/Post-Conditions

### Pre-conditions:

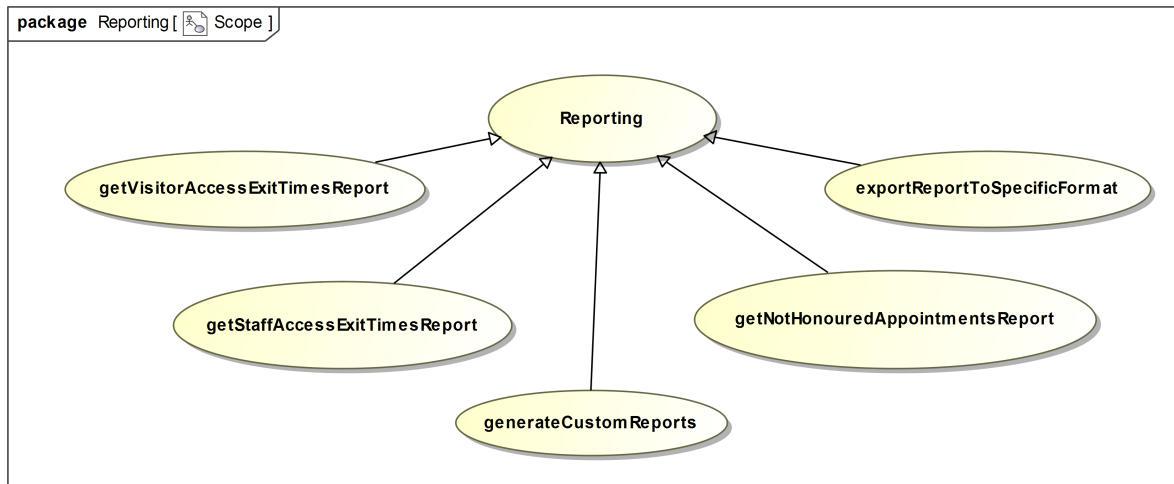
- Administrator (admin user) needs to be logged in to register a user.
- The user being register should be authenticated using LDAP.
- The user being registered must be a staff member.

### Post-conditions:

- The staff member is registered on the system with his/hers own biometric data.
- The staff member will be able to gain access to department/building.

## 6.2.4 Process Specification

# 7 Reporting



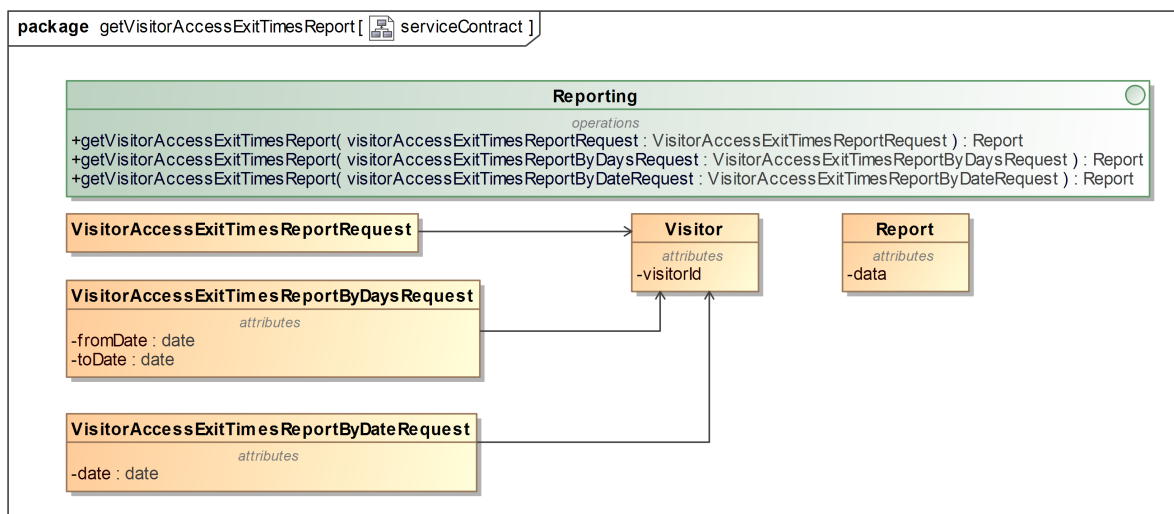
## 7.1 Get Visitor Access Exit Times Report

Priority: *Important*

### 7.1.1 Description

The getVisitorAccessExitTimes function allows a user to query the access and exit times of a specific visitor.

### 7.1.2 Service Contract



### 7.1.3 Pre/Post Conditions

Pre-conditions:

- User must be logged in.
- User must have the correct authorization to make use of this function.
- Visitor must exist.
- Date must be valid.

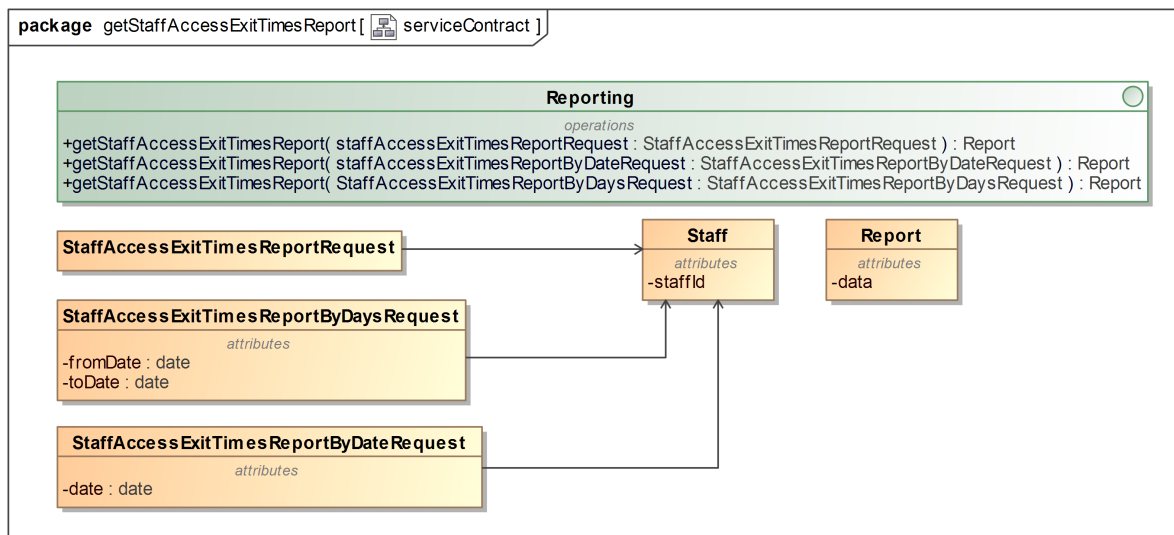
## 7.2 Get Staff Access Exit Times Report

Priority: *Important*

### 7.2.1 Description

The `getStaffAccessExitTimes` function allows a privileged user(e.g: Head of Department) to query the access and exit times of a specific staff member.

### 7.2.2 Service Contract



### 7.2.3 Pre/Post Conditions

Pre-conditions:

- User must be logged in.
- User must have the correct authorization to make use of this function.
- Date must be valid.
- Staff member must exist.

## 7.3 Get Not Honoured Appointments Report

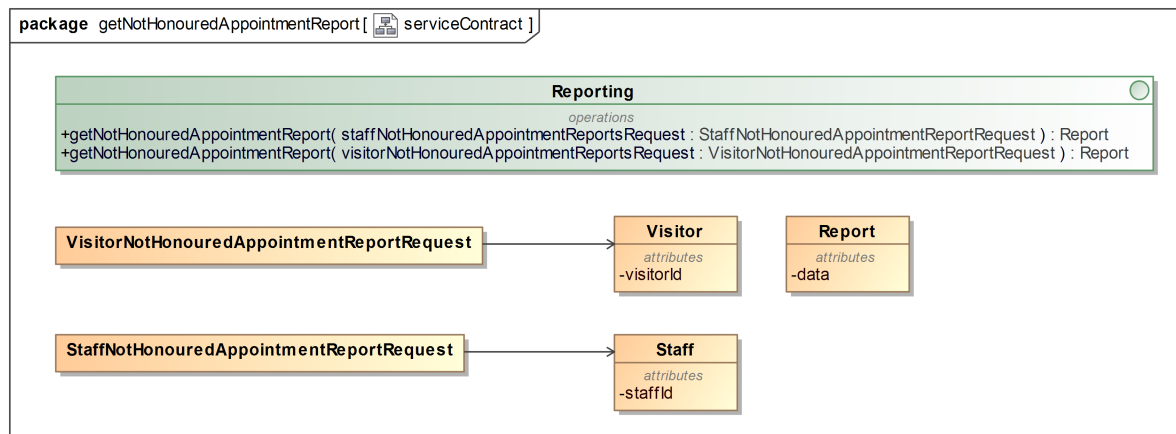
Priority: *Important*



### 7.3.1 Description

The `getNotHonouredAppointments` function provides the user with a means of querying a visitor's or staff member's not honoured appointments.

### 7.3.2 Service Contract



### 7.3.3 Pre/Post Conditions

#### Pre-conditions:

- User must be logged in.
- User must have the correct authorization to make use of this function.
- Visitor or Staff must exist.

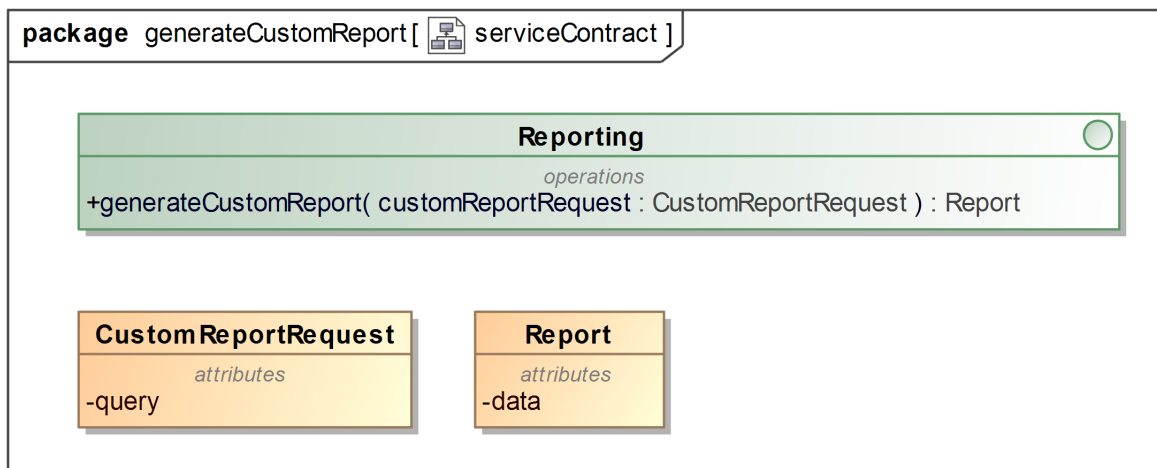
## 7.4 Generate Custom Reports

*Priority: Nice to have*

### 7.4.1 Description

The `generateCustomReports` function allows the user to create a custom report based on a query provided by the user.

### 7.4.2 Service Contract



### 7.4.3 Pre/Post Conditions

- User must be logged in.
- User must have the correct authorization to make use of this function.
- Query must be valid.

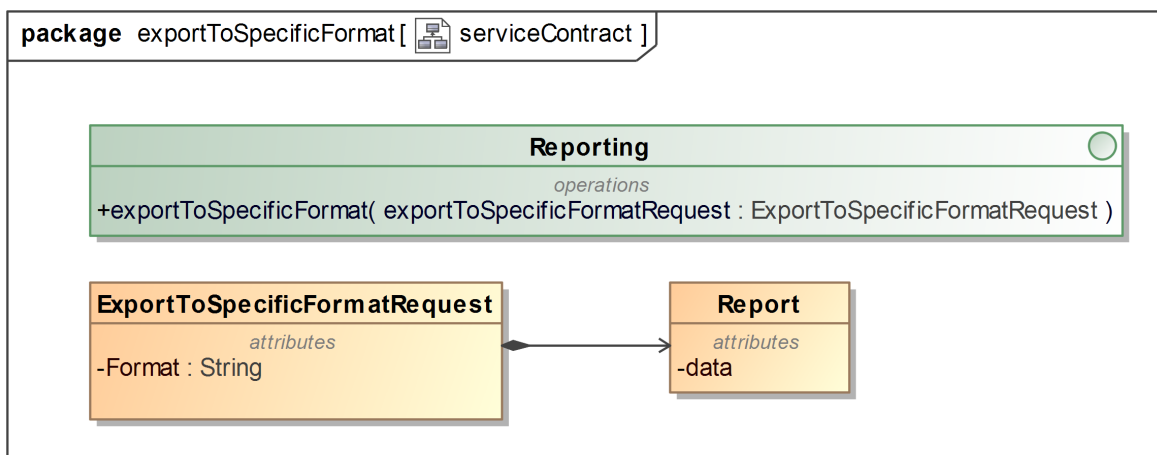
## 7.5 Export Report To Specific Format

Priority: *Important*

### 7.5.1 Description

The `exportReportToSpecificFormat` function will allow the user to export a report to a specific format which will be specified by the user.

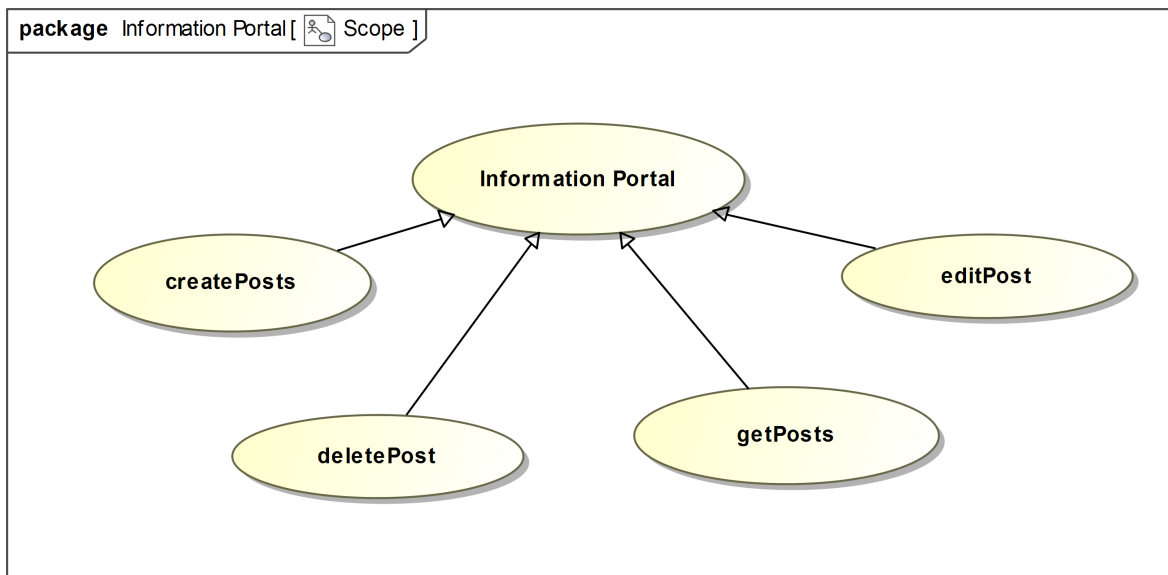
### 7.5.2 Service Contract



### Pre-conditions:

- The format specified by the user is supported by the system.
- User must be logged in.

## 8 Information Portal



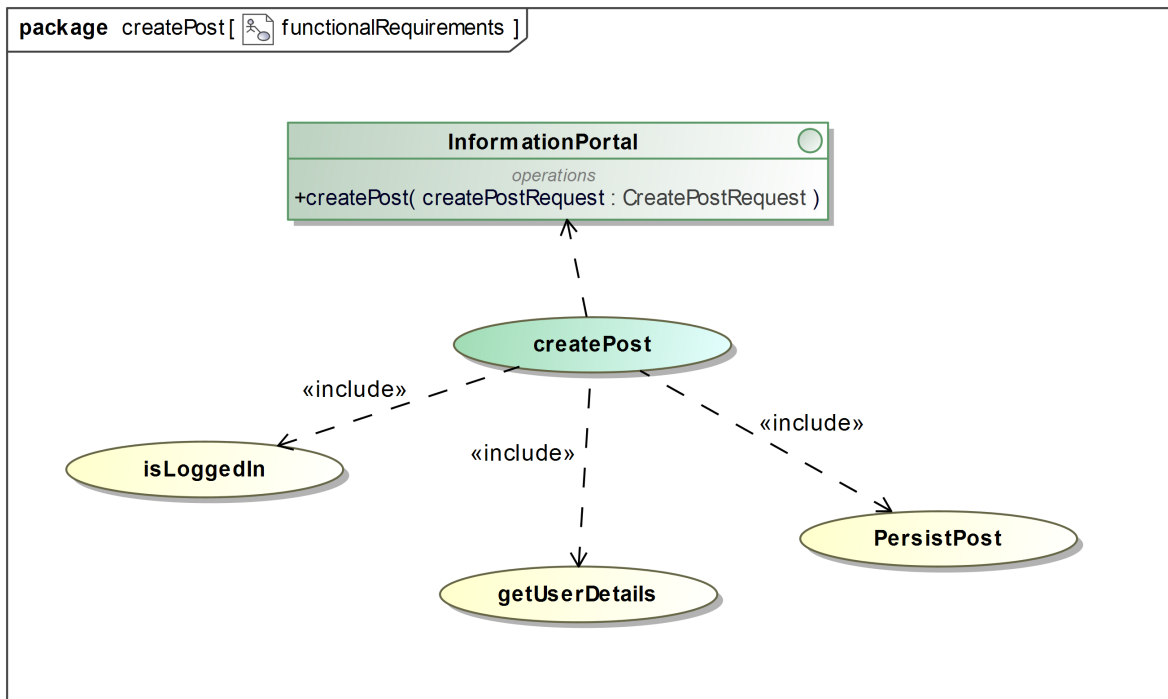
### 8.1 Create Post

*Priority: Important*

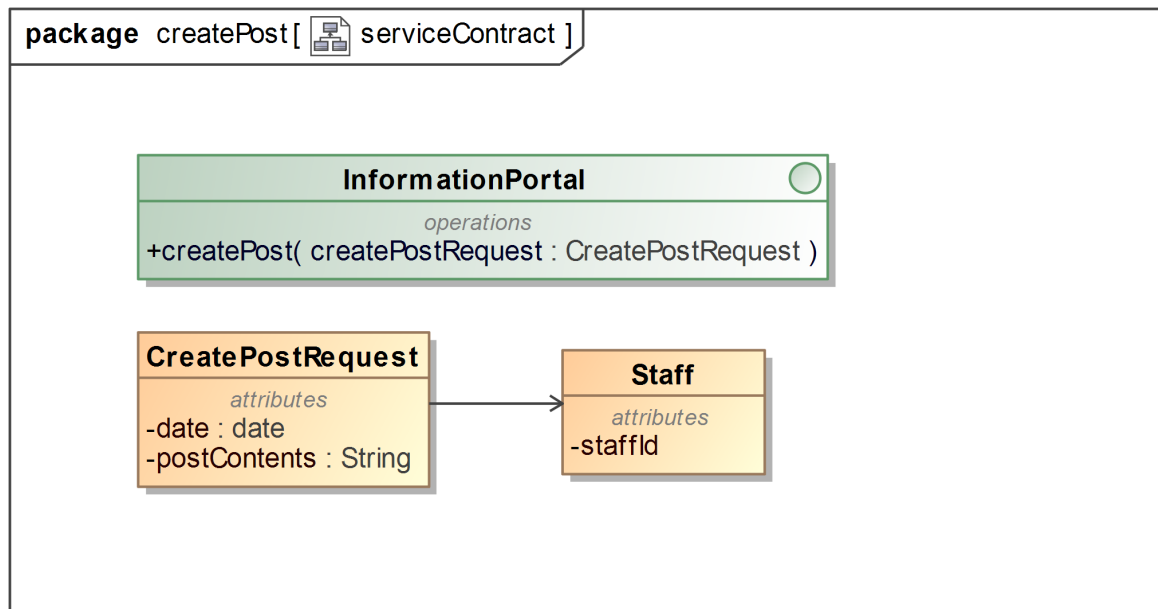
#### 8.1.1 Description

The createPost function allows users of the system to easily create posts which will then be displayed to anyone that visits the Information Portal.

### 8.1.2 Functional Requirement



### 8.1.3 Service Contract



### 8.1.4 Pre/Post Conditions

Pre-conditions:

- User must be logged in.

- User must have the correct authorization to make use of this function.

#### Post-conditions:

- Post will be created and persisted to the database.

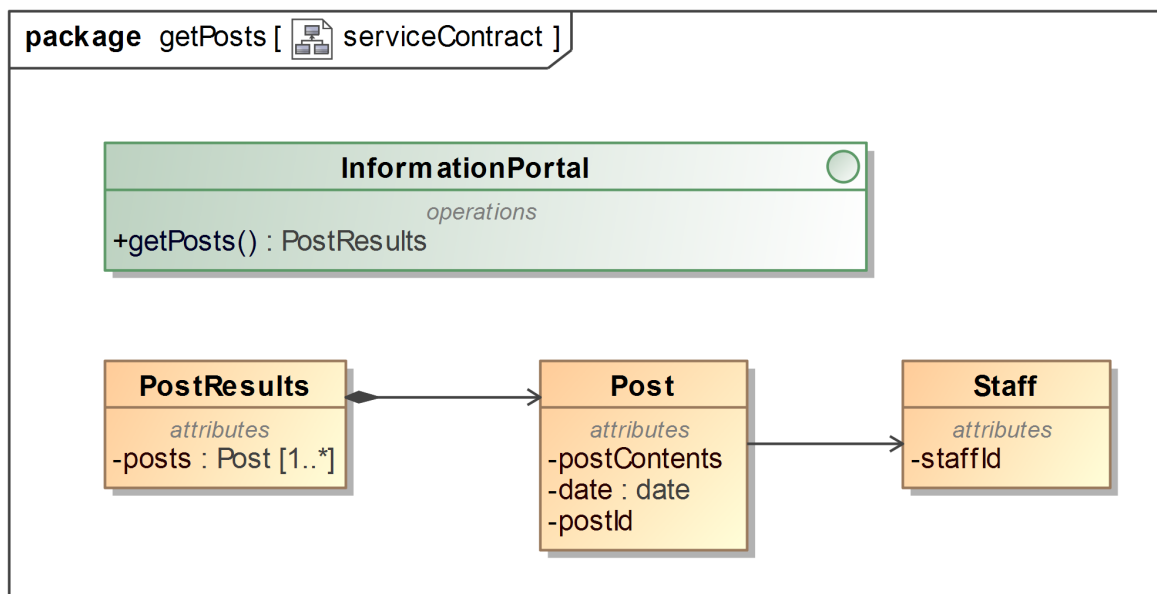
## 8.2 Get Posts

Priority: *Important*

### 8.2.1 Description

The getPost function will retrieve posts from the database which will allow the user to view the posts made.

### 8.2.2 Service Contract



## 8.3 Delete Post

Priority: *Important*

### 8.3.1 Description

The functionality provided by `deletePost` allows the user to remove posts from the Information Portal. User must have necessary permissions.

### 8.3.2 Pre/Post Conditions

#### Pre-conditions:

- User must be logged in.
- User must have the correct authorization to make use of this function.
- Post must exist.

**Post-conditions:**

- Post will be hidden.

## 8.4 Edit Post

*Priority: Important*

### 8.4.1 Description

The editPost provides functionality to the user that allows them to be able to make changes to posts already created.

### 8.4.2 Pre/Post Conditions

**Pre-conditions:**

- User must be logged in.
- User must have the correct authorization to make use of this function.
- Post must exist.

**Post-conditions:**

- Post will be updated.

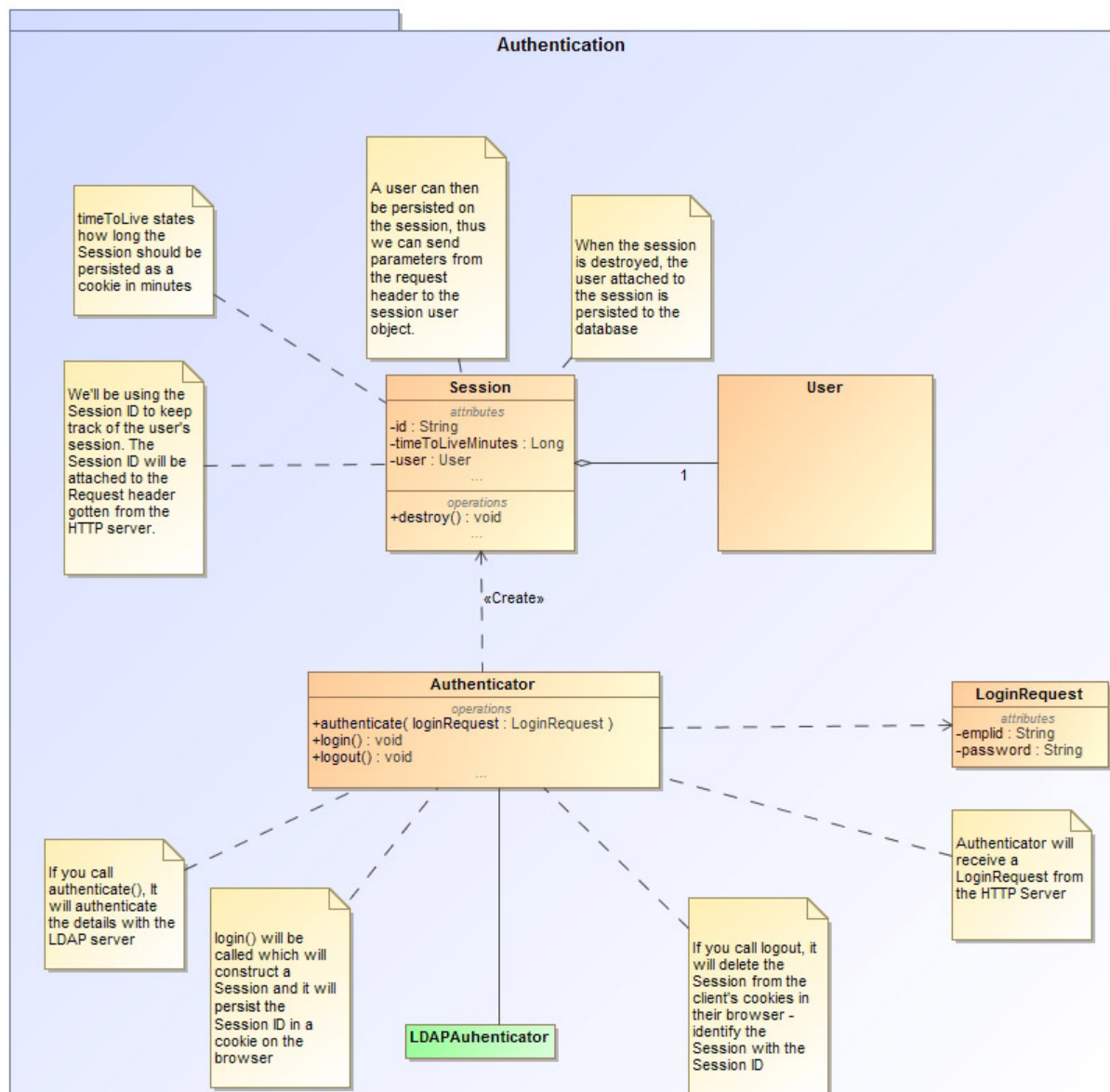
## 9 Authentication

### 9.1 Authenticate

*Priority: Critical*

#### 9.1.1 Description

The following section will describe functionality around logging staff members in and out. Note: The use case, process specification and data structure requirements posted in Authenticate encapsulates all use cases under the authentication functionality.



#### 9.1.2 Pre-/Post-Conditions

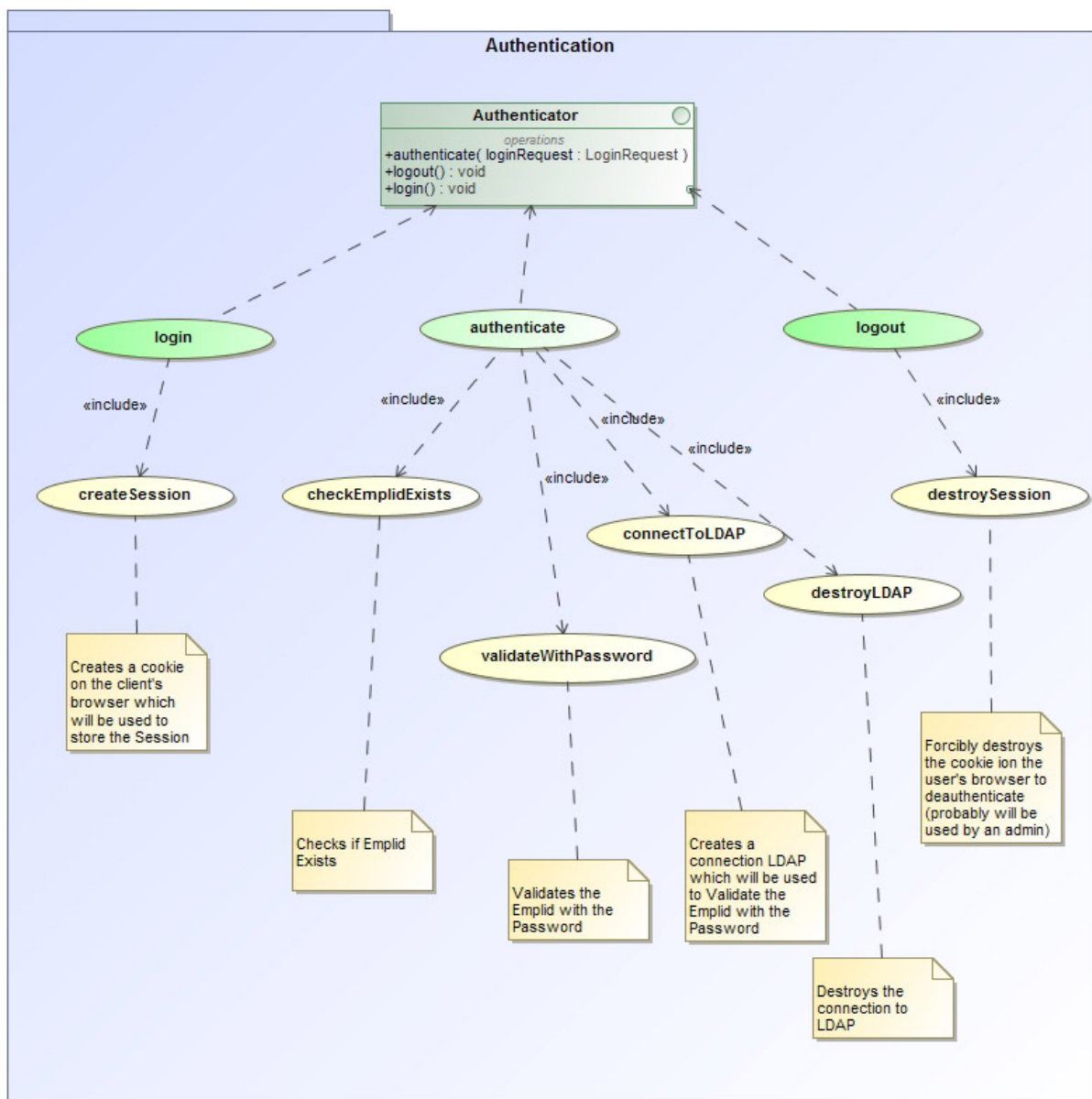
Pre-conditions:

- A user must have an EMPLID from the University of Pretoria
- A user must have an associated password for their EMPLID from the University
- A successful connection to LDAP is important
- A user must be registered as a staff member on LDAP
- A successful validation response after an LDAP authentication is needed to authenticate a user

#### Post-conditions:

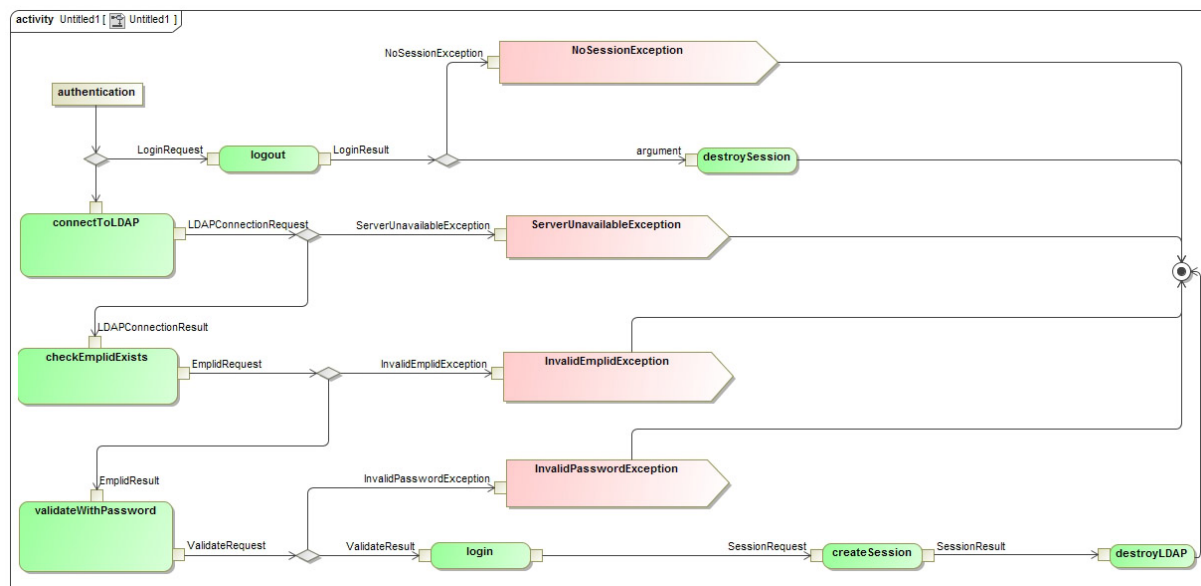
- A user is successfully authenticated on the server

#### 9.1.3 Functionality





### 9.1.4 Process Specification



## 9.2 Log In

*Priority: Critical*

### 9.2.1 Description

A user must be logged into the system once they have been authenticated. A user is logged in by creating a cookie containing the session ID.

### 9.2.2 Pre-/Post-Conditions

**Pre-conditions:**

- A user must have been successfully authenticated by the system to be logged in.

**Post-conditions:**

- A user is successfully logged in and can thus access features which require authentication.
- A user is taken to the booking management page on the website

## 9.3 Log Out

*Priority: Critical*

### **9.3.1 Description**

The system must be able to log a user out. A user is logged out by destroying the cookie containing their session ID.

### **9.3.2 Pre-/Post-Conditions**

#### **Pre-conditions:**

- A user must have been successfully authenticated by the system to be logged in.

#### **Post-conditions:**

- A user is successfully logged in and can thus access features which require authentication.
- A user is taken to the booking management page on the website