



UNIVERSITEIT VAN PRETORIA  
UNIVERSITY OF PRETORIA  
YUNIBESITHI YA PRETORIA

---

Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

---

## Software Requirements Specification and Technology Neutral Process Design

---

*Members:*

Matthew Botha 14214742

Gershon Maluleke

Nathan Dunkley

Aiden Malan

Sello Thosago

Nsovo Baloyi

Matheu Botha

**Team India**

March 4, 2016

v0.1

# 1 Introduction

TODO!! This still needs to be completed.

# 2 Vision

This projects aim is to provide a system to facilitate the administration of postgraduate publications, including but not limited to journal articles and conference papers. It should be able to store information on Users, Authors and Publications as well as employ a authentication system to ensure only specified parties within the system have access to operations on the stored information. The purpose of keeping track of the publications is, in part, to keep track of the DoE and UP weighted units earned by Users and/or authors. The reason for this is that these users/authors are required to publish a certain amount of units each year in order to determine pay, research funding, and maintaining their research position.

# 3 Background

Currently the publications are kept track of with a complicated excel spreadsheet. The purpose of the spreadsheet is to keep track of the status of publications in order to generate a report detailing the units earned by a researcher. There are many drawbacks to this system. It is tedious to maintain the spreadsheet. Data is spread over multiple sheets. The larger the database gets the slower it performs. The information is displayed in a raw textual format which makes it a daunting task to sit through and analyse. It is difficult to add new functionality and change or enhance existing functionality. When pulling in data from multiple sheets it becomes difficult to debug the Lookups.

# 4 Architecture

## 4.1 Access Channels

This service should be accessable via web based clients. i.e. web based clients, mobile browsers. As well as an Android application.

## 4.2 Quality Requirements

- Reliability - The reports generated from the data provided to the system should be
- Scalability - It is estimated that there will be approximatly one hundred users. The system should be able to handle each user being active at the same time even though it is unlikely.

- **Cost** - Generating a report for all data in the system will be the worst case scenario in this system and thus have a high cost. The system should however be able to produce subsets of the report at a relatively low cost. For example a report for all researchers under a specific research group.
- **Security** - There should be strict access right applied to each of the sub-systems and their components. This means that if a user, by definition of the business rules, is not allowed to access a certain functionality, then they will be denied access.
- **Auditability** - Actions performed on the system should be logged in a manner that would allow one to trace back the history of actions performed on the system.

### 4.3 Intergration Requirements

The system will operate in a standalone fashion. In the future, however, it may require intergration with google calender.

### 4.4 Architecture Constraints

Client has not provided any architecture constraints.

## 5 Functional Requirements and Application Design

### 5.1 Use case Prioritisation

#### 5.1.1 User Subsystem

Register User: Critical  
 Update User: Important  
 Login User: Critical  
 Logout User: Critical  
 Add to Publication: Important

#### 5.1.2 Conference Subsystem

**Add Conference:** important  
**Update Conference:** important  
**Delete Conference:** important

#### 5.1.3 Journal Subsystem

**Add Journal:** important  
**Update Journal:** important  
**Delete Journal:** important

#### 5.1.4 Thesis Subsystem

**Add Thesis:** important

**Update Thesis:** important

**Delete Thesis:** important

### 5.2 Use case/Services contracts

#### 5.2.1 User Subsystem

**Pre-conditions:**

**Register user:** The user can not be registered to the system already and the user registering a new user has to be admin.

**Update user:** The user has to be logged in first before they can update their profile and they can only update their own profile.

**Login user:** The user has to be registered to the system first.

**Logout user:** The user has to be logged in first.

**Add to publication:** The user has to be logged in first and the publication has to exist in the system

**Post-conditions:**

**Register user:** The user is now registered and added to the database.

**Update user:** The user's profile is updated and the changes are saved to the database.

**Login user:** The user is logged into the system.

**Logout user:** The user is logged out of the system.

**Add to publication:** The user's name is now added to the author's of the selected publication.

### 5.2.2 Conference Subsystem

#### Add Conference:

**Pre-conditions:** The user must be logged in.

**Post-conditions:** The conference is added.

#### Update Conference:

**Pre-conditions:** The user must be logged in and a the conference must be added prior to this use case.

**Post-conditions:** The conference is updated.

#### Delete Conference:

**Pre-conditions:** The user must be logged in and a the conference must be added prior to this use case.

**Post-conditions:** The conference is deleted.

### 5.2.3 Journal Subsystem

#### Add Journal:

**Pre-conditions:** The user must be logged in.

**Post-conditions:** The journal is added.

#### Update Journal:

**Pre-conditions:** The user must be logged in and a the journal must be added prior to this use case.

**Post-conditions:** The journal is updated.

#### Delete Journal:

**Pre-conditions:** The user must be logged in and a the journal must be added prior to this use case.

**Post-conditions:** The journal is deleted.

### 5.2.4 Thesis Subsystem

#### Add Thesis:

**Pre-conditions:** The user must be logged in.

**Post-conditions:** The thesis is added.

#### Update Thesis:

**Pre-conditions:** The user must be logged in and a the thesis must be added prior to this use case.

**Post-conditions:** The thesis is updated.

#### Delete Thesis:

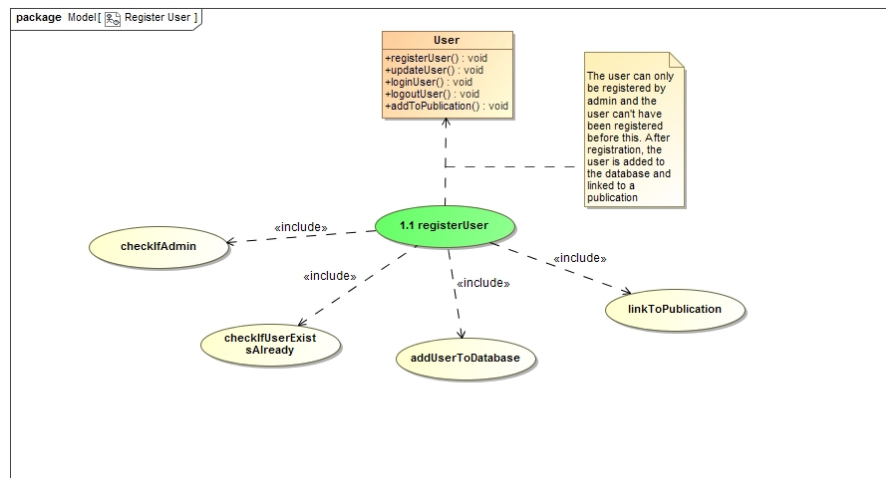
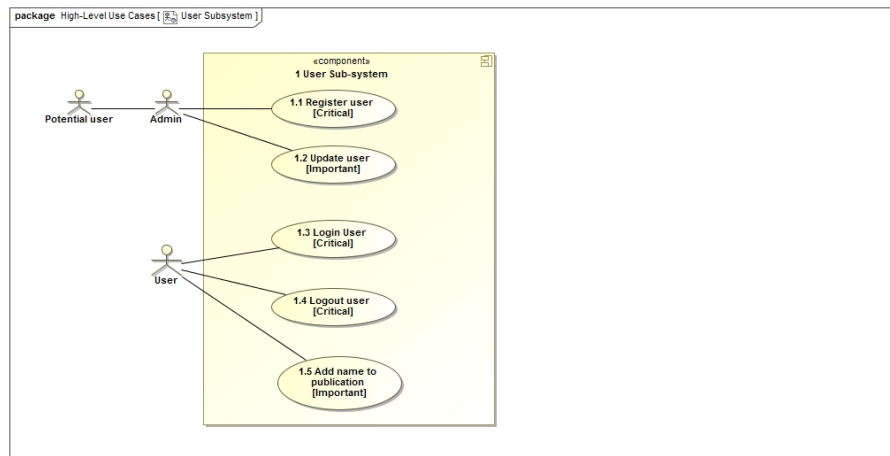
**Pre-conditions:** The user must be logged in and a the thesis must be added prior to this use case.

**Post-conditions:** The thesis is deleted.

### 5.2.5 Conference, Journal and Thesis Subsystem Request and Results Data Structures

## 5.3 Required Functionality

### 5.3.1 User Subsystem Use Case



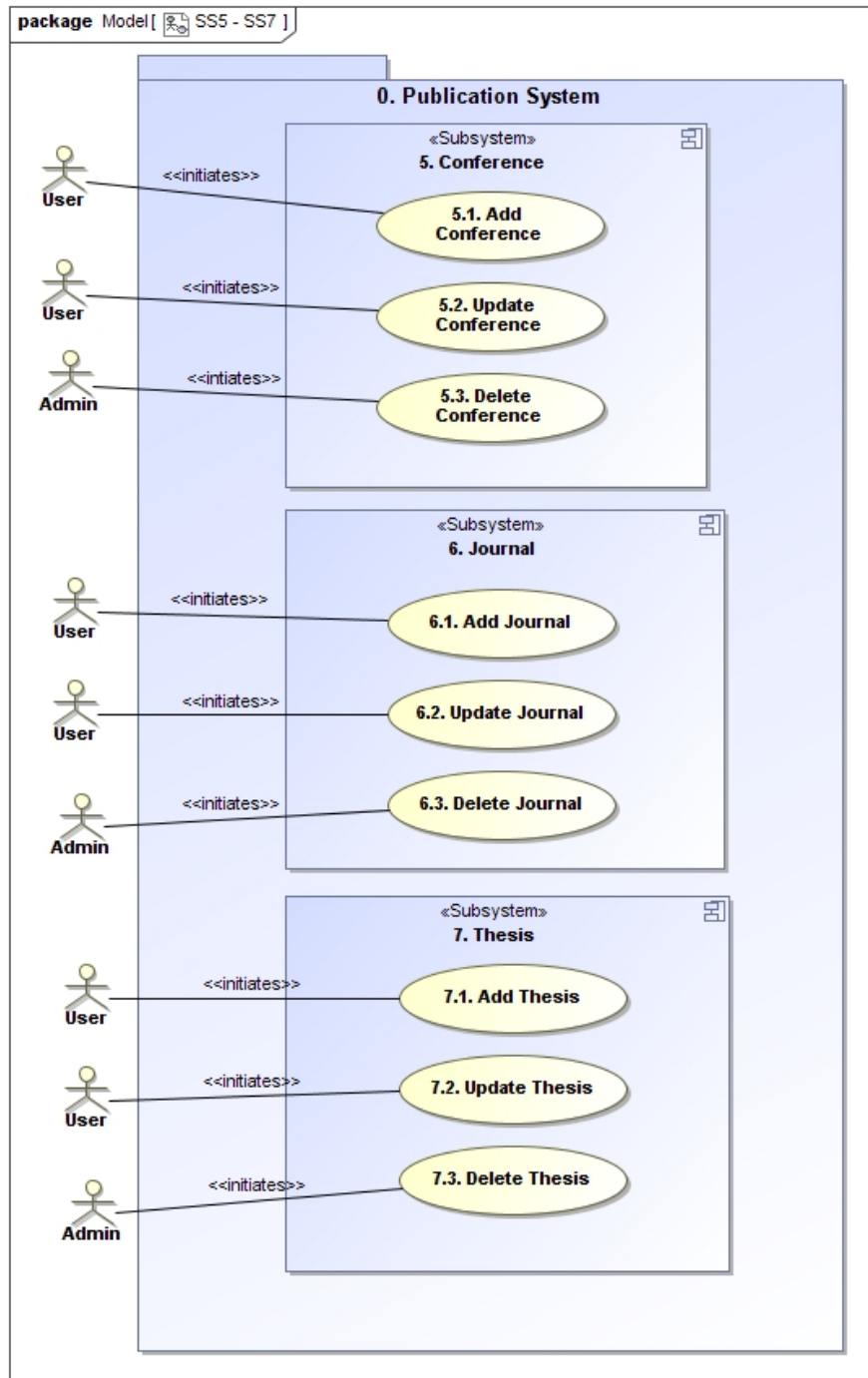
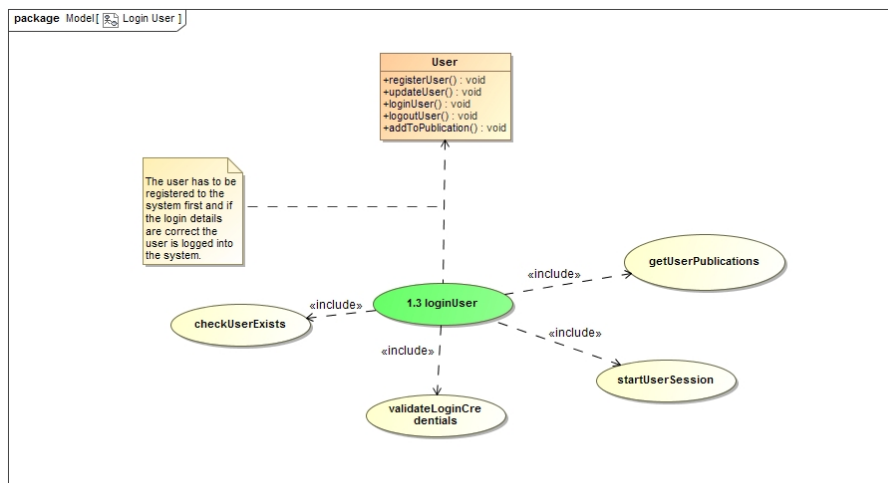
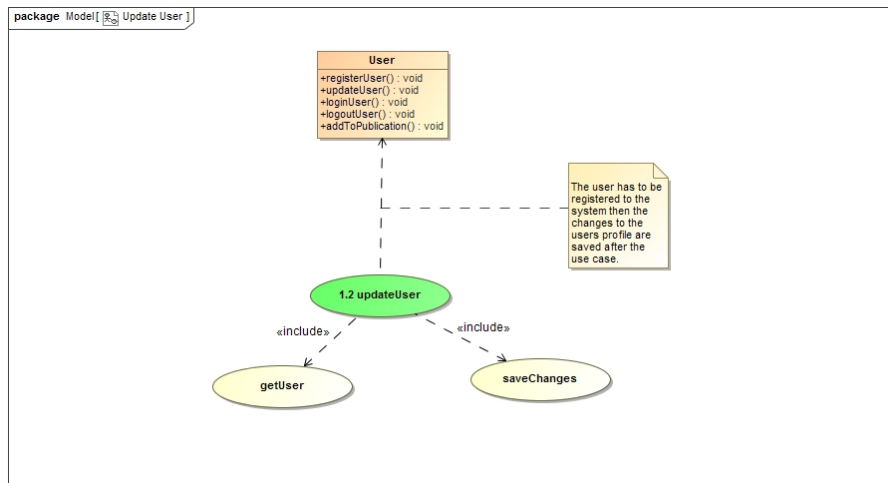
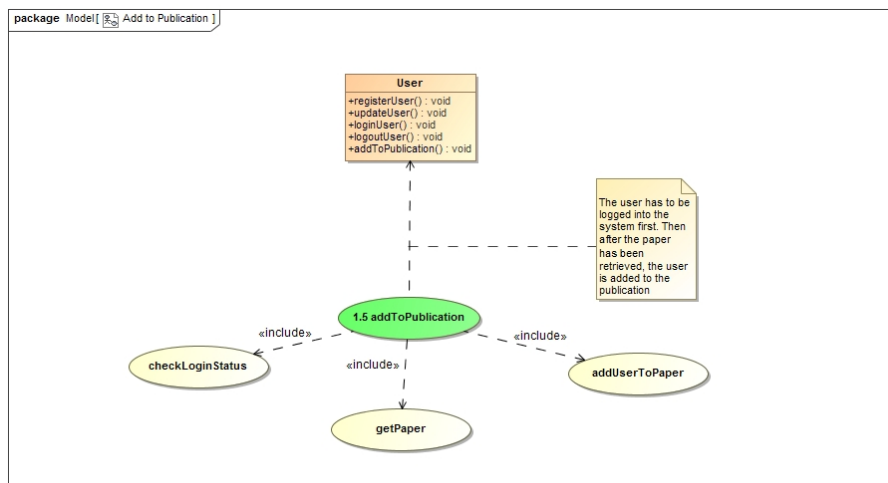
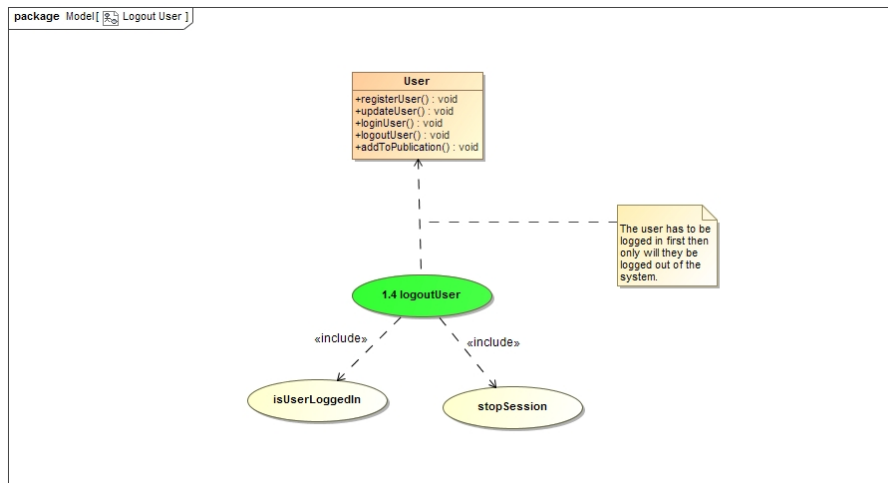


Figure 1: Use case diagram for the Conference, Journal and Thesis Subsystems







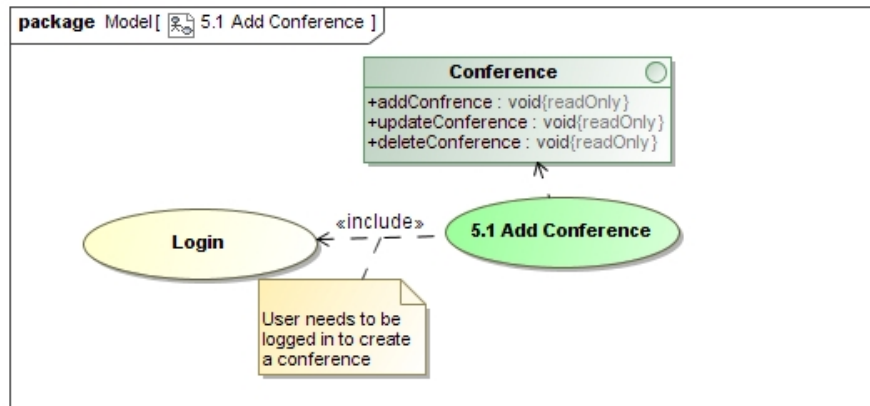
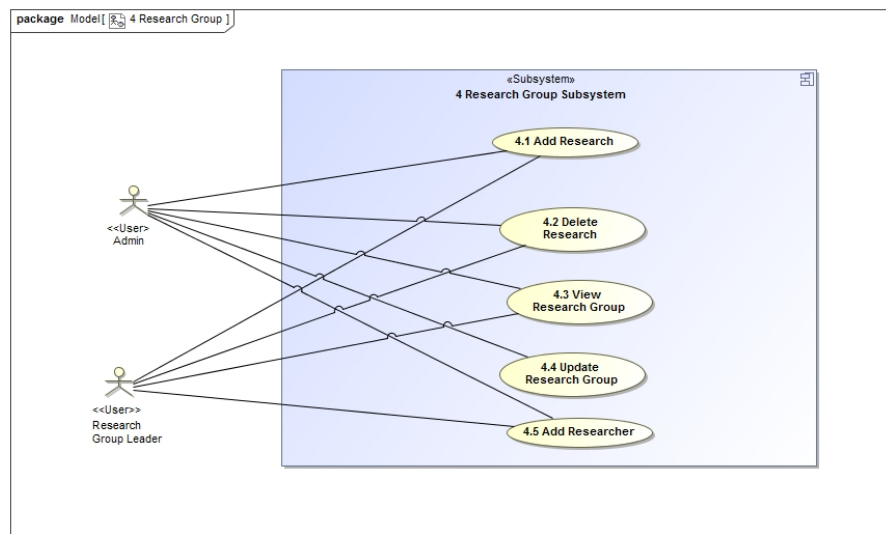


Figure 2: Add Conference

### 5.3.2 Research Group Subsystem Use Case



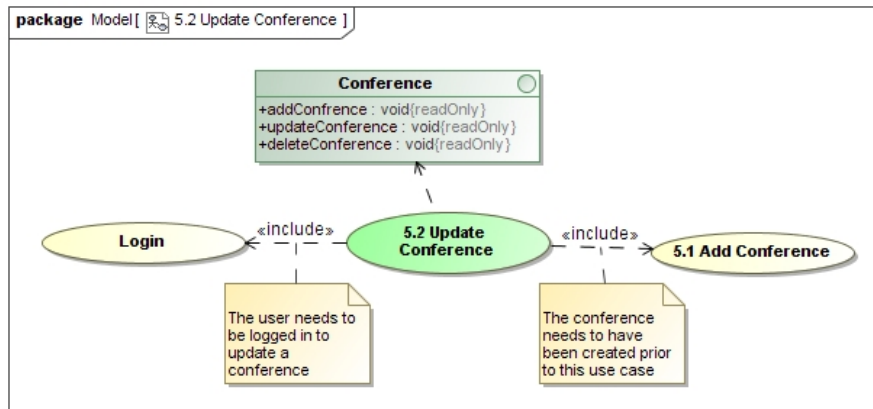


Figure 3: Update Conference

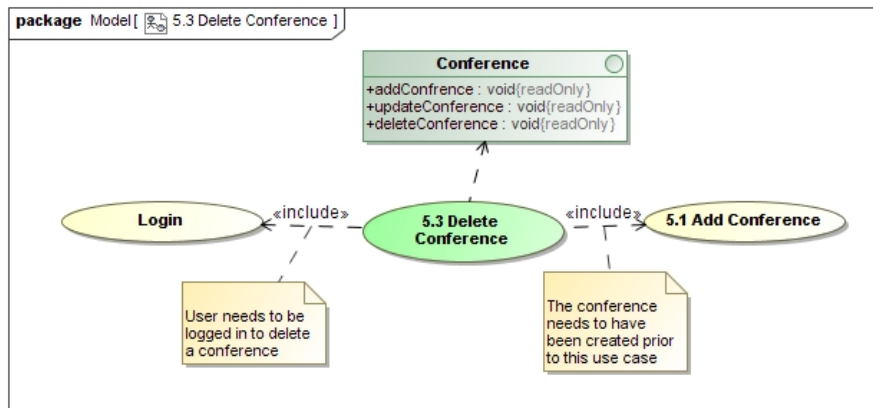


Figure 4: Delete Conference

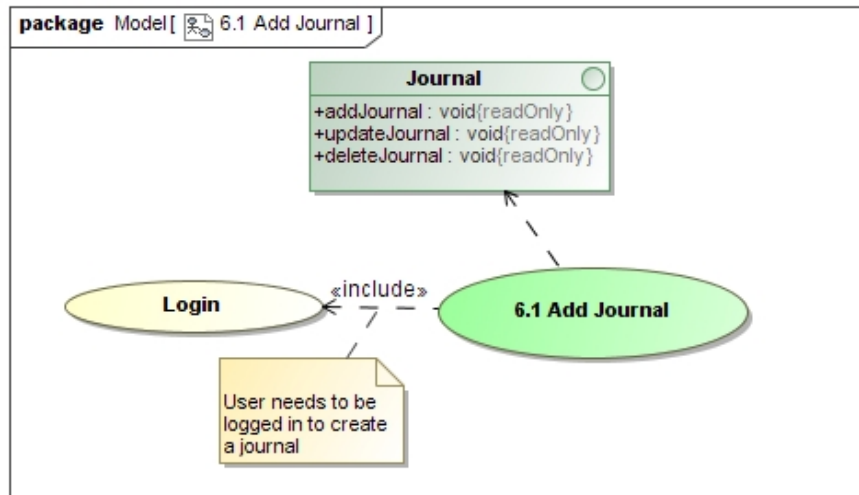


Figure 5: Add Journal

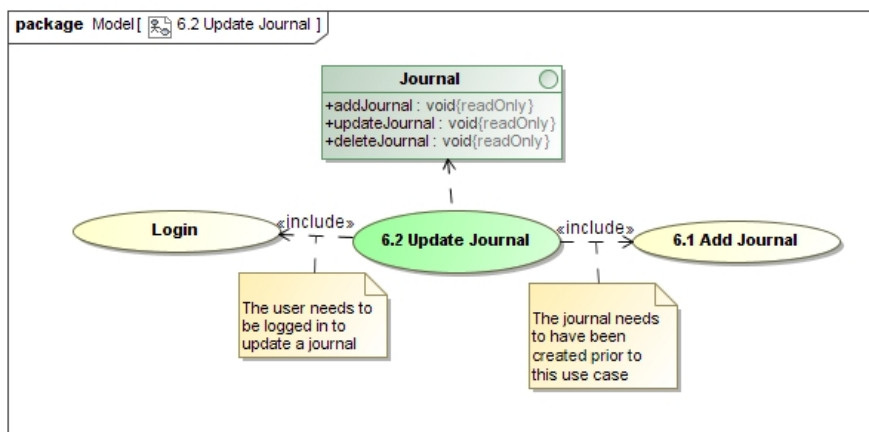


Figure 6: Update Journal

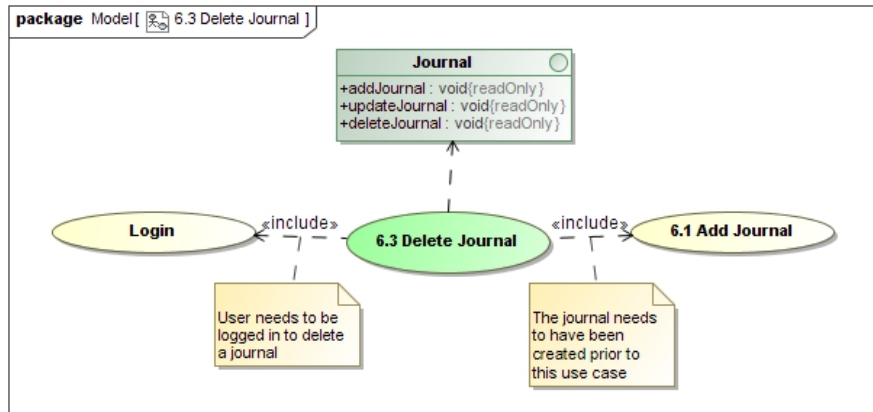


Figure 7: Delete Journal

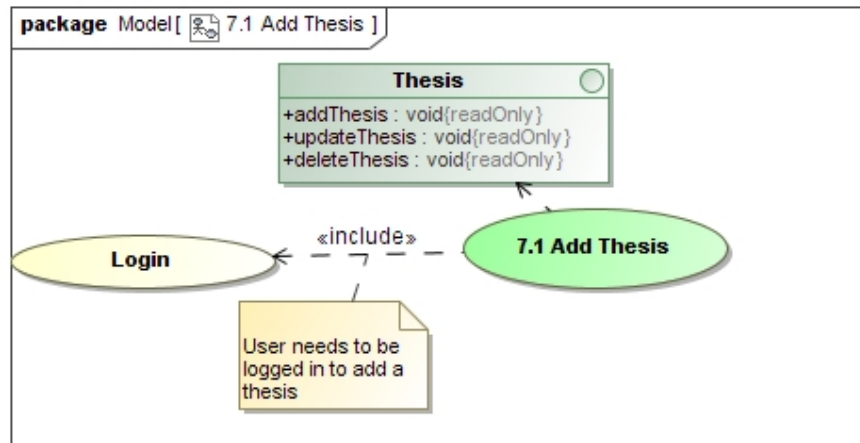


Figure 8: Add Thesis

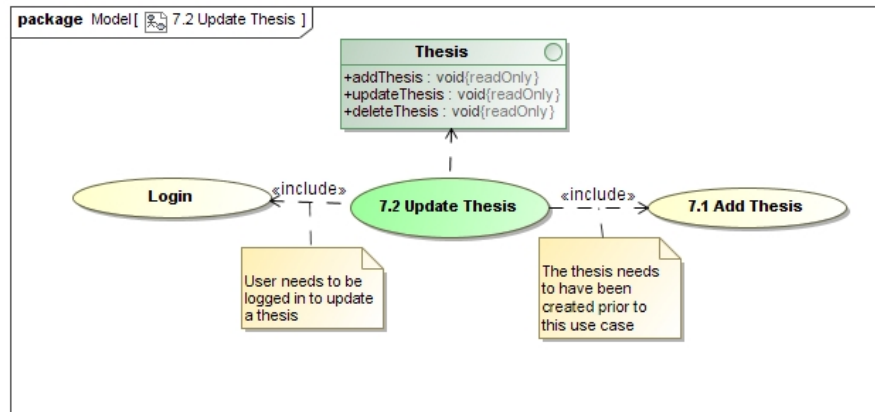


Figure 9: Update Thesis

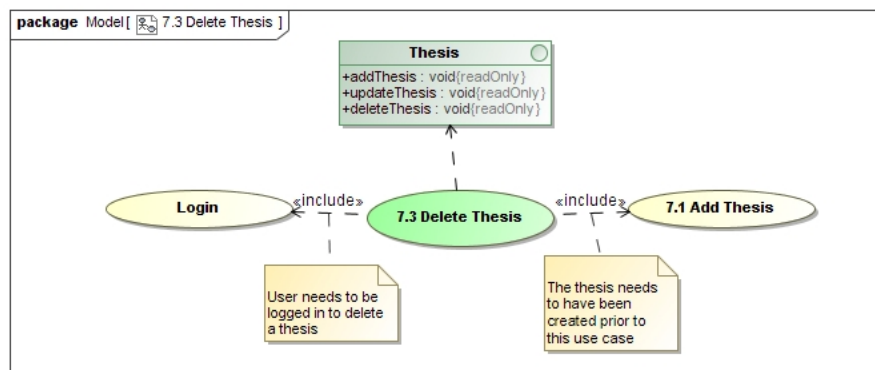


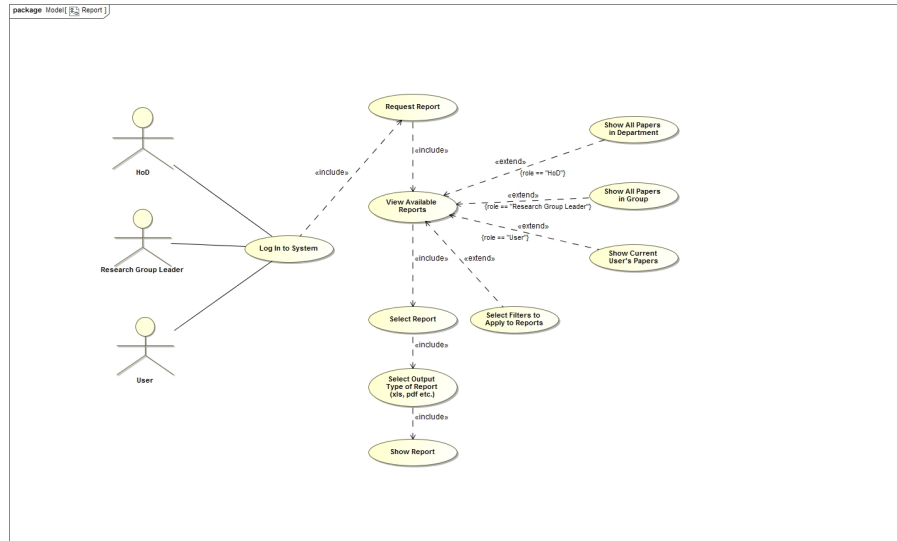
Figure 10: Delete Thesis

### 5.3.3 Conference Subsystem Use Case

### 5.3.4 Journal Subsystem Use Case

### 5.3.5 Thesis Subsystem Use Case

### 5.3.6 Report Subsystem Use Case



## 5.4 Process Specifications

### 5.4.1 Conference Subsystem

### 5.4.2 Journal Subsystem

### 5.4.3 Thesis Subsystem

## 6 Open Issues

- What does DoE stand for?
- Specifics with regard to the DoE and UP weighted contributions
- Researcher funding

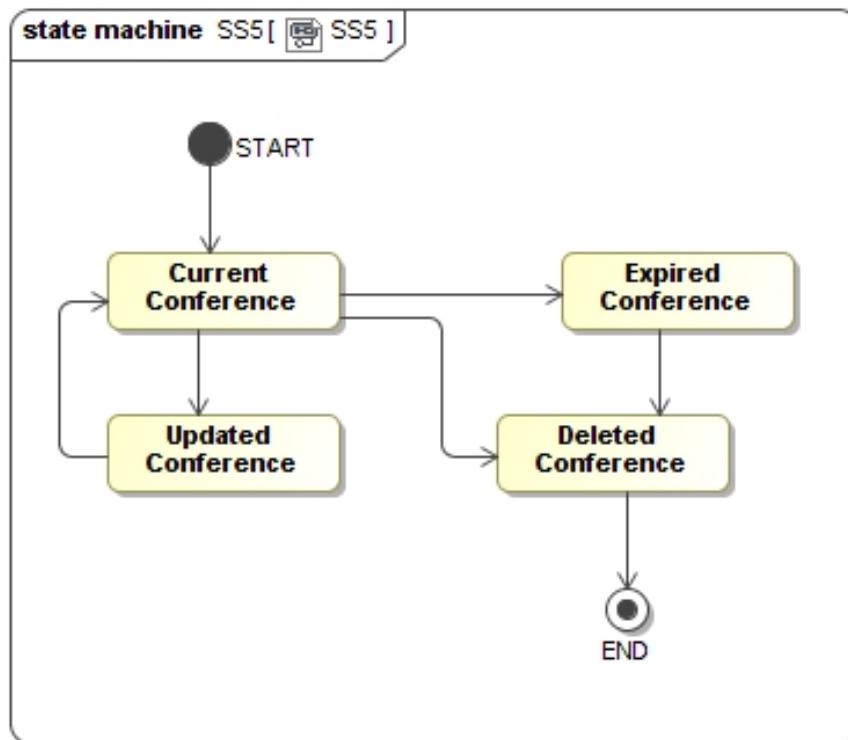


Figure 11: Conference State Diagram



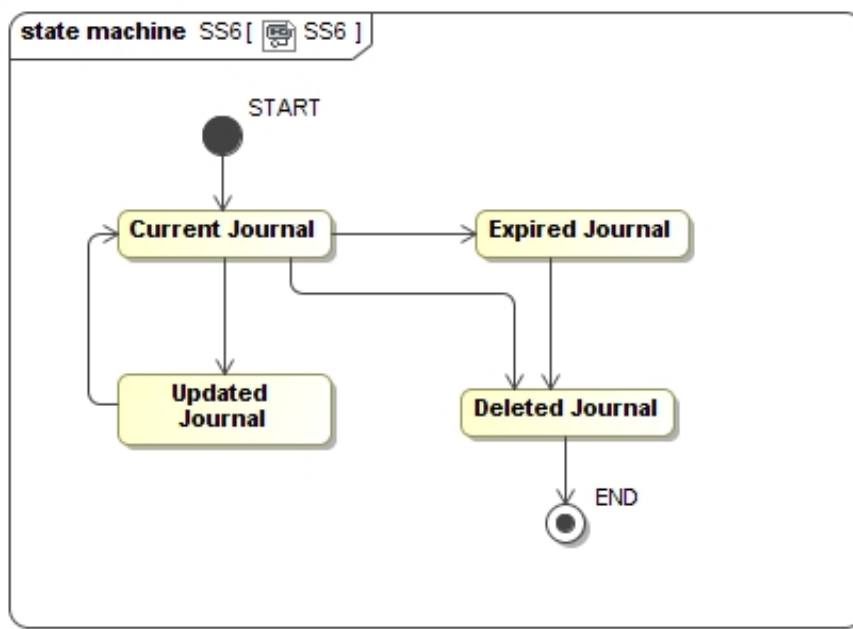


Figure 12: Journal State Diagram

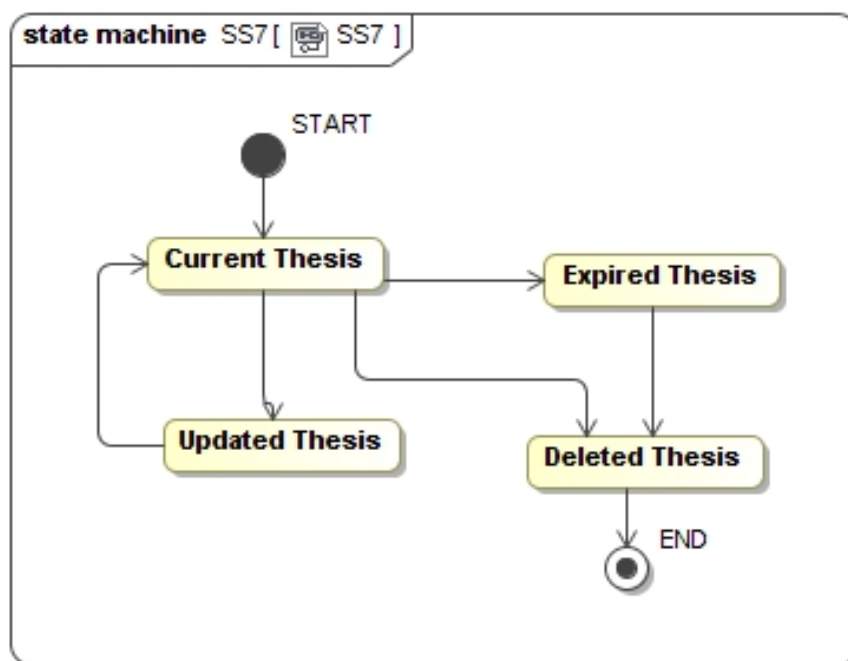


Figure 13: Thesis State Diagram