UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science

COS 301 - Mini Project

Assignment 1

Software Requirements Specification and Technology Neutral Process Design

---

Team Delta

Mpho Sharon Baloyi (14133670)

Dirk de Klerk (28159102)

Killian Kieck (12252213)

Daniel Malangu (13315120)

Dzilafho Mulugisi (13071603)

Duncan Smallwood (13027205)

Dian Veldsman (12081095)

Contents

1 Introduction

This document aims to set out the functional and non-functional requirements of a system as specified by the Department of Computer Science. The system is required to allow the department to collaborate, share, and maintain research articles in an effective and efficient manner. The document will also serve the purpose of communicating the requirements and specifications as needed by the client.

2 Vision

The client for this project, Department of Computer Science, has called for the design of an application that will allow the department to keep track of all research publications written and published within the department. The main idea behind the project is to alleviate the stress and time required for collaborating on, maintaining, and completing research articles. We have therefore invisioned the following goals:

- Simplify the effort required in maintaining publications as well as pending works.
- Ability to add/remove/specify main author and co-authors.
- Keep track of impact scores of different publications
- Allow authors and staff members to collaborate on articles.
- The ability to add/remove/edit meta-data of said articles.
- Allow different levels of privilege on user accounts.
- Afford privileged users (HODs) the ability to access addtional information on authors, publications and/or a summary page.
- The need to integrate the software in android or web based applications.
- Afford only head authors the ability to remove co-authors as well as transfer authorship.

3 Background

We find ourselves within the Information Age which means we are bombarded with more information than we can handle. This implies the rise of new challenges. How does one deal with this information overload in a timely and effective manner? Technology is the primary cause of the problem but at the same time provides us with multiples solutions in the form of networking.

The very nature of research means that a team or individual will constantly have to revise and apply changes to their work as new information becomes available. Some research efforts are simply to large handle on an individual level as is the case with interdiciplinary and cohort studies. Thus effective team work is required. When team members find themselves in remote locations, this can lead to a serious problem. How does one succesfully collaborate effectively and efficiently?

The above problem is what is currently hindering research efforts within the Department of Computer Science and the world! It is for this reason that the client is interested in developing a research repository that will allow them to share, maintain, and collaborate on various research papers in an effective and timely manner. A system, that is easy to use, is thus suggested to enable remote collaboration on research materials allowing the user to save considerably on time and effort. Such a system would certainly improve the throughput of research efforts, thus allowing authors to focus on quality research rather than deadlines.

4 Architecture Requirements

4.1 Access Channel Requirements

This section specifies and describes different access channels through which the services of the target system will be accessed by humans. The different functions of the system will be highlighted and how different access channels will provide access to these functions. The system must be able to provide the platforms with an interface through which they can access the functions this will be done through the use of a Application Programming Interface (API).

To access the services of the systems users have to be connected to a network preferably the internet instead of a LAN, this is to enable users to access the services of the system from anywhere around the world. The type of devices that will enable access to the system's services are devices capable of connecting to the internet.

To provide access to the services of the system, the client requested that the following platfroms be used:

- A Web Interface
- An Android Application

The above mentioned platforms will have ways of providing the functionality of the system through their methods that are defined in their API's. The access channels will have to provide the users with the ability to view certain data structures , search through the data structures, create new entities (users and publication meta-data), delete existing entities, update attributes of existing entities , to open, view, edit or import certain types files (excel,image files etc), user registration, log in and authentification.

Android Application.

In addition to the device being able to connect to the internet it has be able to run on this operating system. Android provide access to its features via its API, therefore the transfer of data between the system and the platform will be done through the API's. Different versions of the operating system can be used to access the system's services,preferably the recent version 4.4

Android offers a range of libraries and functions that enable its application to send,receive,view data in various formats as well as interfacing with other applications to provide more functionality.

Web interface

For clients who access the system's services via a web interface can use any of the avaliabe web browsers such as firefox, Internet Explorer, Google chrome, Safari, Opera and others. Web browsers make use of web technologies to provide the functionalities that are provided by the system's to the users

Restful web services clients will interact with our system via its RESTful api that makes use of the http request to obtain data from other systems over the internet.

4.2 Quality Requirements

Quality not only refers to how many clients requirements the product fulfils. It also refers to how much effort is put into a product to satisfy the consumers needs and wants compared to other similar product. The aim however is to fulfil, at the least, every need over the wants of the client.

- Performance:
  - Program should be designed to be run as efficient as possible.
  - Execution time should be reduced by avoiding unnecessary comparisons and recursive function calls
  - Database queries should be optimised by avoiding needless joins and using stored procedures
- Reliability:
  - Information stored in the database should remain correct when being transported from the system front-end.
  - System should always be available for use unless the system is intentional shut down for maintenance
- Scalability:
  - System needs to be handle growth in terms of the number of users by:
    - Managing activities running concurrently
    - Storing information in lightweight format to save storage space
- Security:
  - The application should maintain permissions for access and modification of information for the different user groups.
  - Also proper authentication protocols should be implemented for every user group to prevent unauthorised users access at any user group.
- Flexibility:
  - There should be provisions for the addition of new user groups and the modification of existing ones.
  - It should be possible to add new groups of data without rebuilding the system
  - Provision for the modification of data types
- Maintainability:
  - Loose coupling should be encouraged, creating pieces that can be repaired without making huge modification to other parts of the system.
- Auditability:
  - Modifications on database transactions should be stored and logged

    - Information of when and where users access the system should be logged
- Integrability:
    - Program should follow the coding standards specified by the client
- Usability:
    - System should have options grouped in logical manner
    - Help hints should be available for more advanced procedures in the program
    - Where information should be entered in a specific format an example of it should be displayed to the user
    - Manual should have a detailed description of all the functionality in the program

4.3 Integration Requirements

Integration involves merging various subsystems into a single cohesive system. With regard to this project the subsystems may consist the application being developed as well as database management systems.

In order to successfully and safely integrate these subsystems they need to be directly and securely linked to one another. This can be easily achieve by creating a secure connection through the application to the database management system.

The system itself will need to be integrated on two seperate platforms, namely a website and mobile application. In the case of the website it may be desirable to display certain content dynamically as opposed to reloading a new page for each item.

4.3.1 Integration Channels

It is recommended that the following be used to successfully achieve the aforementioned integration:

- PHP: Hypertext Preprocessor

In order to achieve this dynamic display the following technologies will be used:

- AJAX
- JSON

4.3.2 Quality Requirements Integration

Systems consist of both functional, what a system has to do, and non-functional requirements, how a system needs to behave. In other words non-functional requirements refers to qualitative attributes of a system. The following quality requirements will need to be integrated into the system:

- **Low resource consumption(Mobile use):** the user should be kept in mind when developing the system, to ensure that no unnecessary resources are used.
- **Good performance:** The system must achieve what it is designed to do in as little time as possible.
- **Reliability:** The system needs to be stable and provide the user with access at all times.
- **Security:** The materials hosted on the system as well as the user accounts needs to be protected at all times from unauthorised users.
- **Safety:** Integrity of the data hosted on the system needs to be assured.
- **Scalability:** The system needs to accommodate growth.
- **Flexibility:** The system needs to be easily adaptable to change should the client require additional functionality for example.

4.4 Architecture Constraints

The Architecture constraints were indicated on 16.02.2016 in a Client requirements session and lists the following technologies that will be used in the project:

- HTML (Hypertext Markup Language)
- PHP
- AJAX (Asynchronous JavaScript and XML)
- Git (Version Control System)
- Andriod

5 Functional Requirements and Application Design

5.1 Use Case Prioritisation

The Use Case Prioritisation is specified for each use case listed under the next section: "Use Case/Services Contracts".

5.2 Use Case/Services Contracts

- **Registration**

  Description: A user is able to register an account on the publishing website.

  Use Case Prioritisation: Critical

  Pre-Conditions:

  - A user must make use of a unique user name.
  - A user can register an account if they are a staff member.
  - A user who is not a staff member must register through an administrator or Head of department.

  Post-Conditions:

  - The user information is stored in the database.
  - The user receives the login information.
  - The user is able to login the website.

- **Login**

  Description: A registered user is able to log into the website making use of the features available to them.

  Use Case Prioritisation: Critical

  Pre-Conditions:

  - A user must have a registered account in order to login.
  - A user must login with the correct user name and password.

  Post-Conditions:

  - A user has access to certain parts of the website according to their privileges.
  - A user has access to certain functions of the website according to their privileges.

- **Logout**

  Description: User is able to logout of the web page.

  Use Case Prioritisation: Critical

  Pre-Conditions:

  - A user must be logged into the web page in order to log out.

  Post-Conditions:

  - User no longer has privileges of being logged in.

- **Create publication**

  Description: A user can create a publication which holds information about the publishing, also authors who contributed.

  Use Case Prioritisation: Critical

  Pre-Conditions:

  - A user must be logged into the web page in order to create a publication.

  Post-Conditions:

  - At least one user must be assigned to the publication (Creator or author).
  - Meta-data on the publishing must be stored in some form of database.

- **Add author**

  Description: Add author - A user can add authors to their publication who have worked or helped with the publication.

  Use Case Prioritisation: Critical

  Pre-Conditions:

  - A user must be logged into the web page in order to add an author.
  - A user must create a publishing before adding another author to the publishing.
  - The user being added must have a registered account with the website.

  Post-Conditions:

  - Added Author is listed on the publication.

- **Remove author**

  Description: A user can remove authors from their created publications.

  Use Case Prioritisation: Critical

  Pre-Conditions:

  - A user must be logged into the web page in order to delete an author.
  - A user can only delete authors they have added.

  Post-Conditions:

  - Author deleted is removed from the list of authors on the publication.

- **Edit publication**

  Description: A user can change the details of their created publications.

  Use Case Prioritisation: Critical

  Pre-Conditions:

  - A user must be logged into the web page in order to edit a publication.
  - A user can only edit their own publications.

  Post-Conditions:

  - Changes made to the publication meta-data is changed on the data on the web server.

- **View publication**

  Description: A user can view publications they are working on.

  Use Case Prioritisation: Critical

  Pre-Conditions:

  - A user must be logged into the web page in order to view a publication.
  - A user can only view publications they are directly involved in.

  Post-Conditions:

  - Information is correctly diplayed to the user.

- **View profile**

  Description: User can view their own and other user profiles.

  Use Case Prioritisation: Important

  Pre-Conditions:

  - A user must be logged into the web page in order to view a profile.

  Post-Conditions:

  - A user will view and have access to edit their profile.

- **Edit profile**

  Description: User can edit their own user profiles.

  Use Case Prioritisation: Important

  Pre-Conditions:

  - A user must be logged into the web page in order to edit their profile.
  - A user can only edit their own profile.

  Post-Conditions:

  - Any changes made to their profile will take effect.

- **Generate summary**

  Description: Certain users/members can generate information on the publications.

  Use Case Prioritisation: Nice-To-Have

  Pre-Conditions:

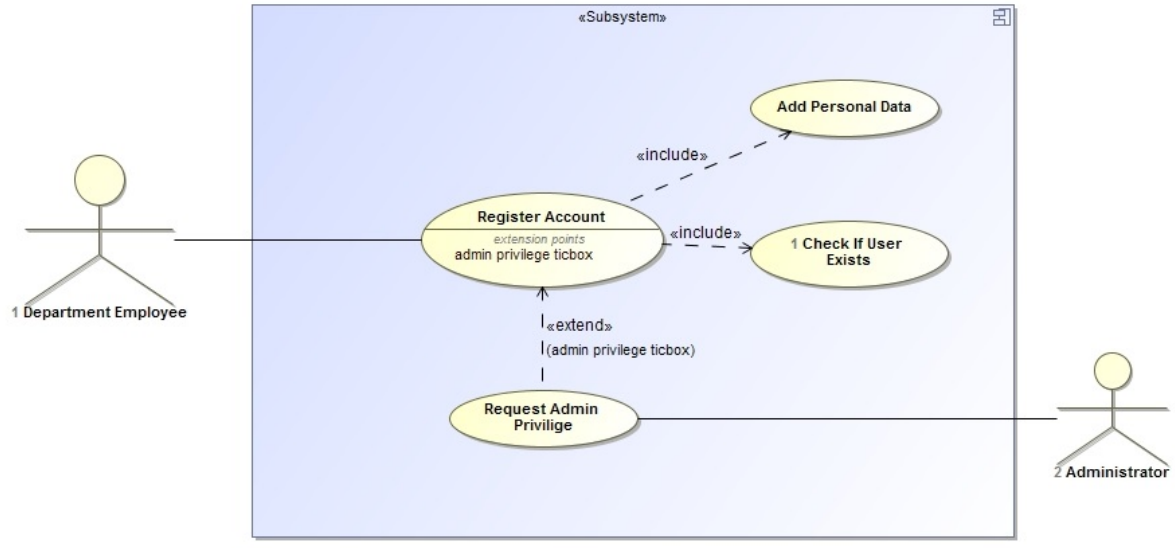
  - A user must be logged into the web page in order to generate a summary on certain publications.
  - A user can only generate a summary on publications they are involved in.
  - Head of department can generate summary on all publications..
  - Research leaders can generate summaries on fields of research.

  Post-Conditions:

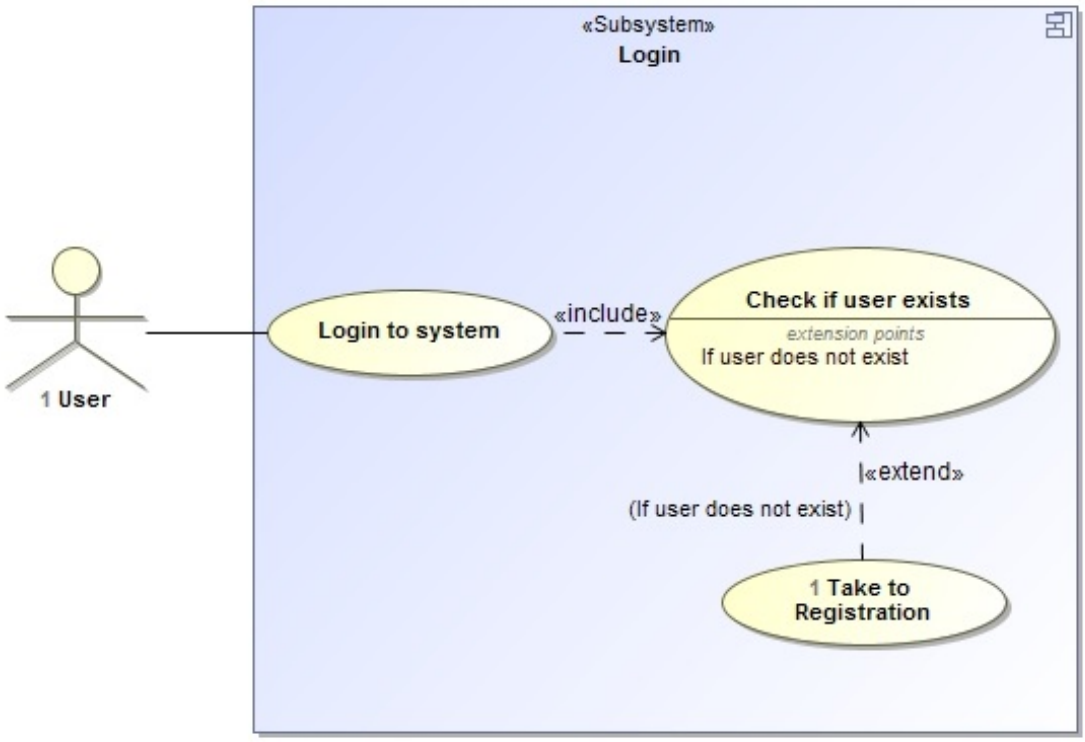
  - Information displayed of selected publications must be accurate.

5.3 Required functionality

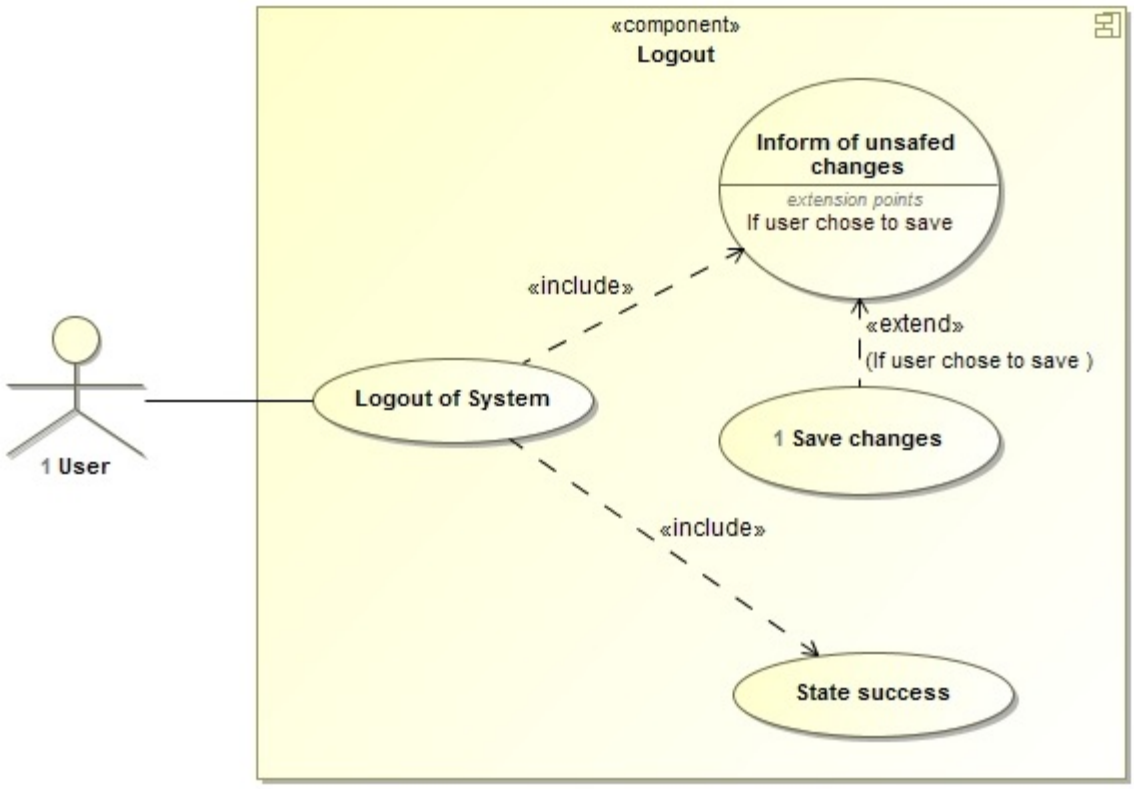
5.3.1 Registration Use Case Diagram



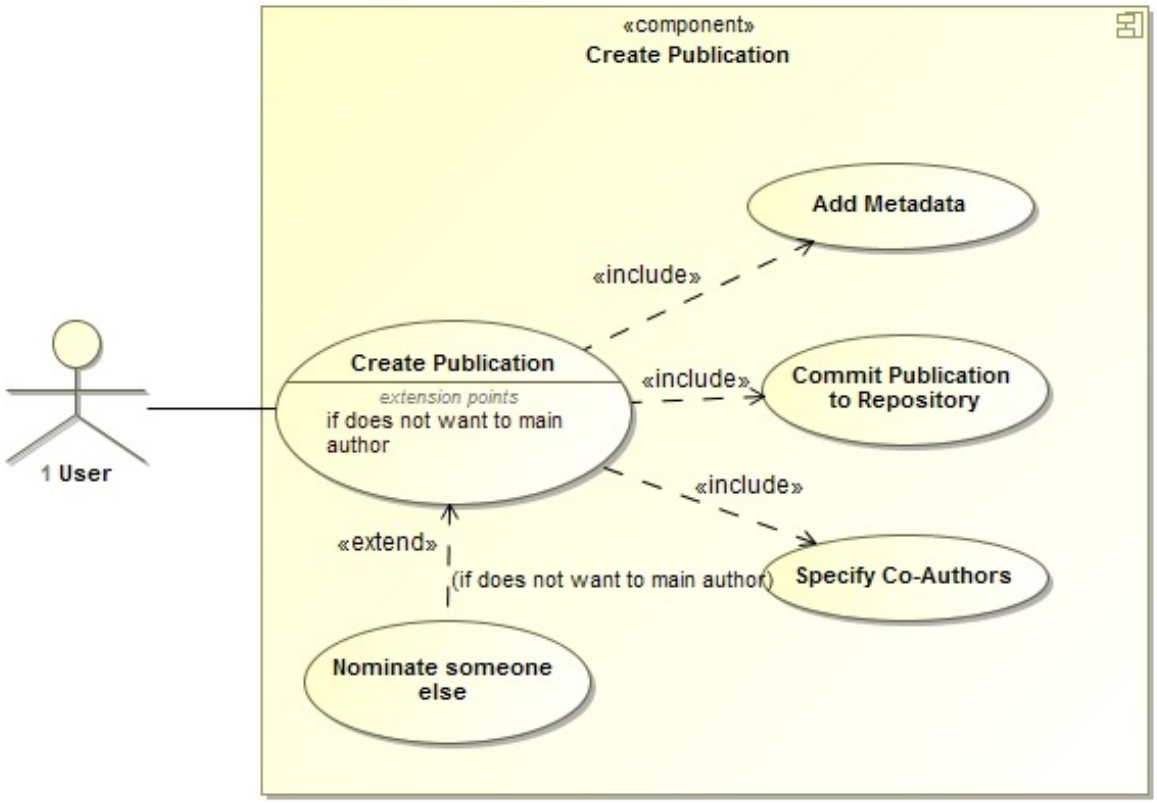
5.3.2 Login Use Case Diagram



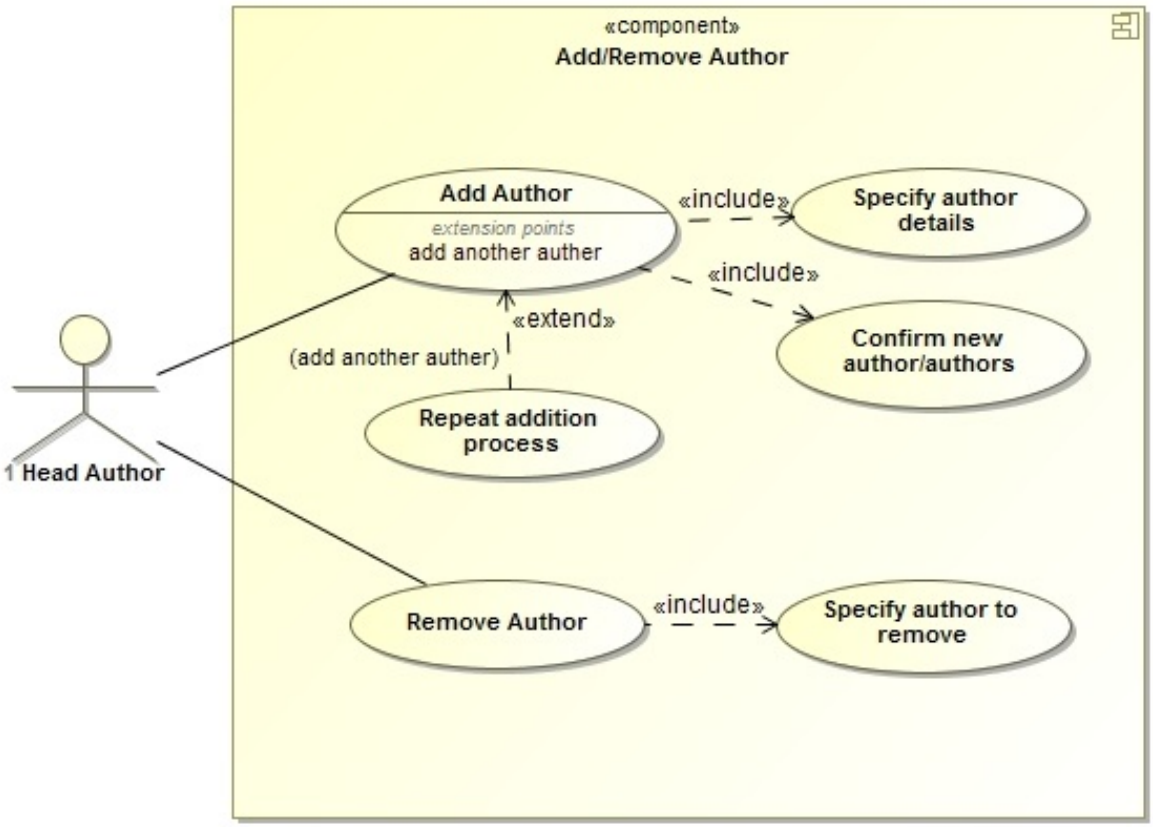
5.3.3 Logout Use Case Diagram



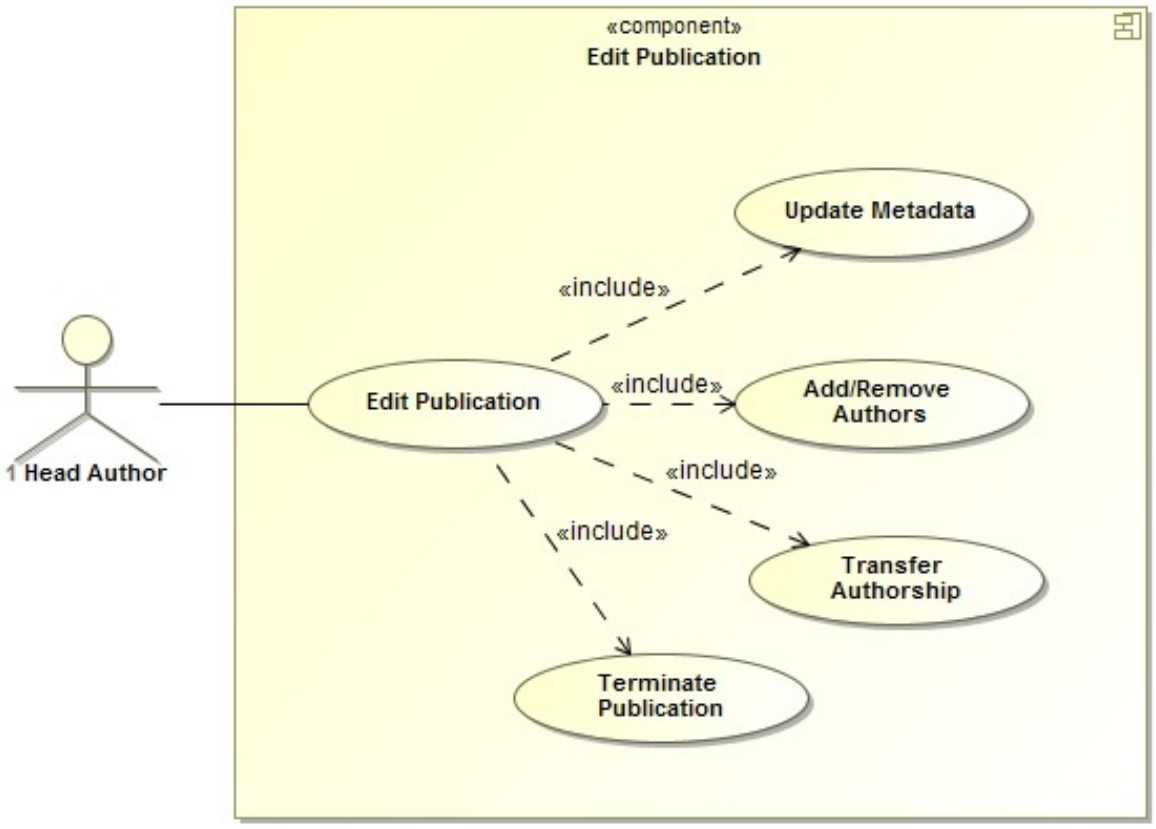
5.3.4 Create Publication Use Case Diagram



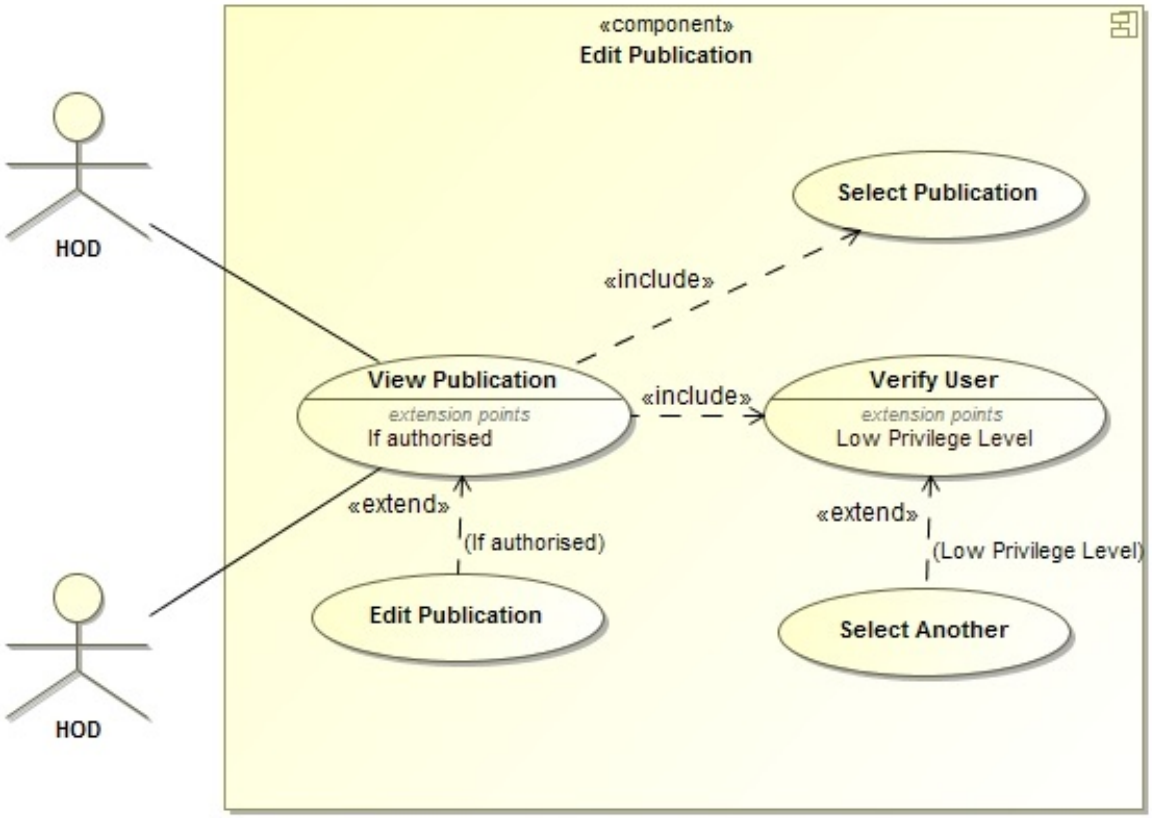
5.3.5 Add/Remove Author Use Case Diagram



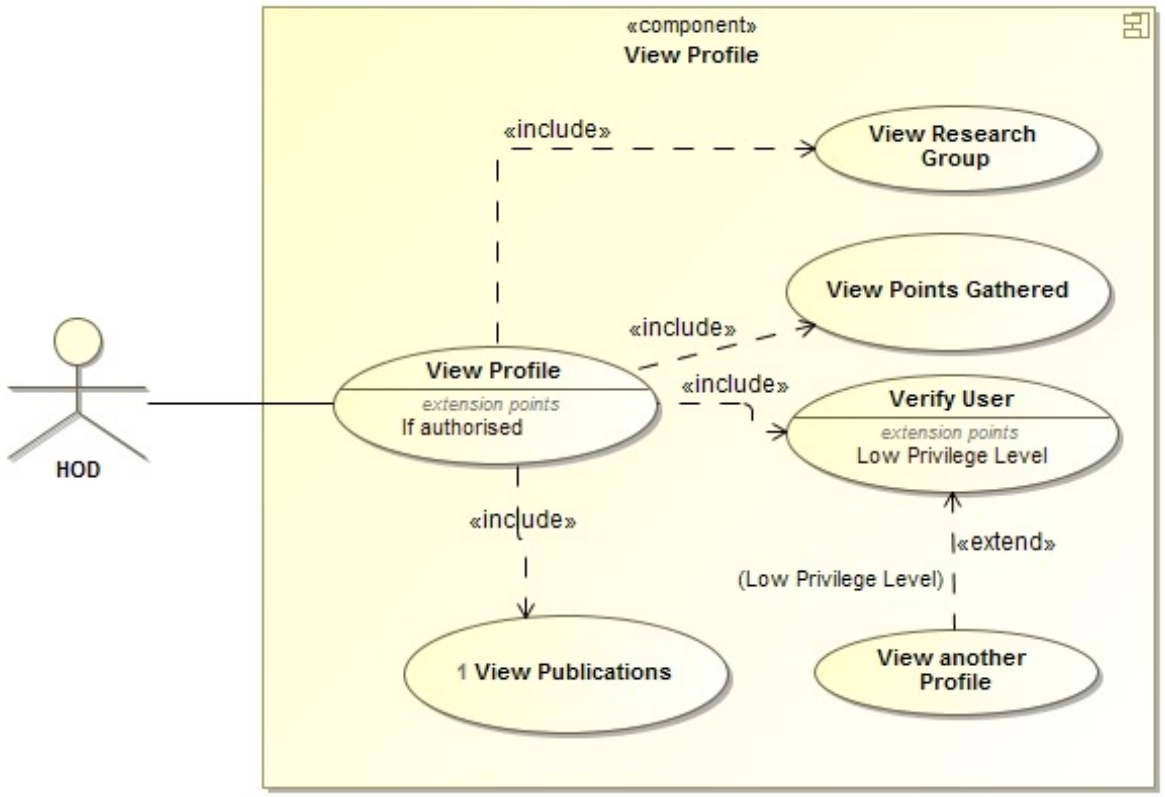
5.3.6 Edit Publication Use Case Diagram



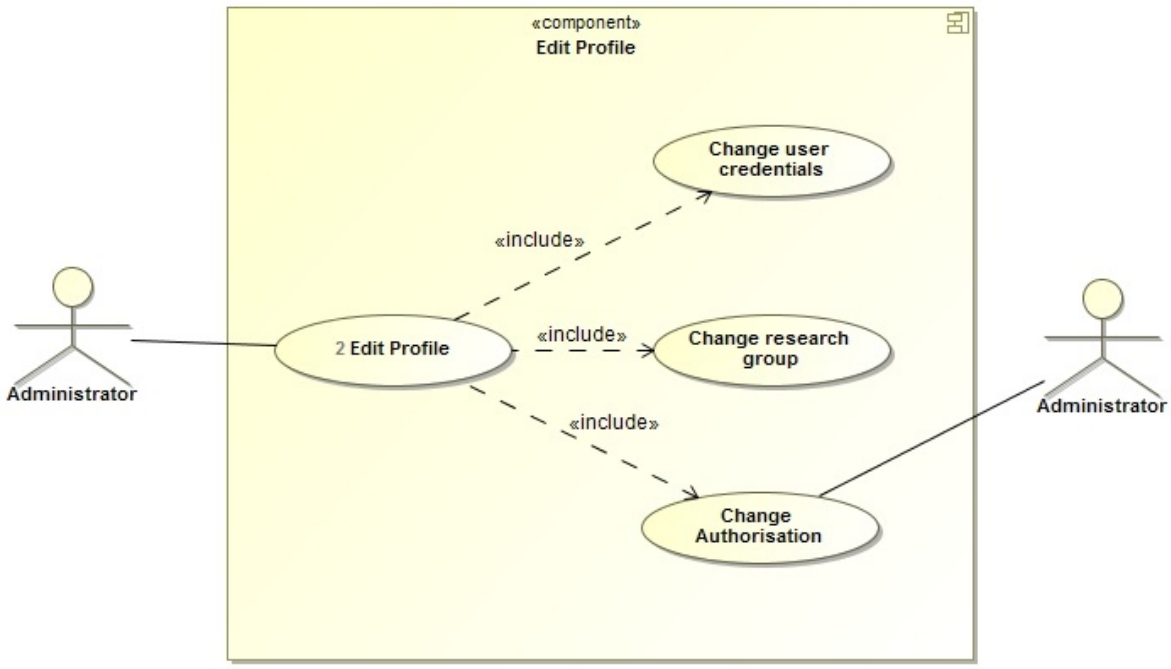
5.3.7 View Publication Use Case Diagram



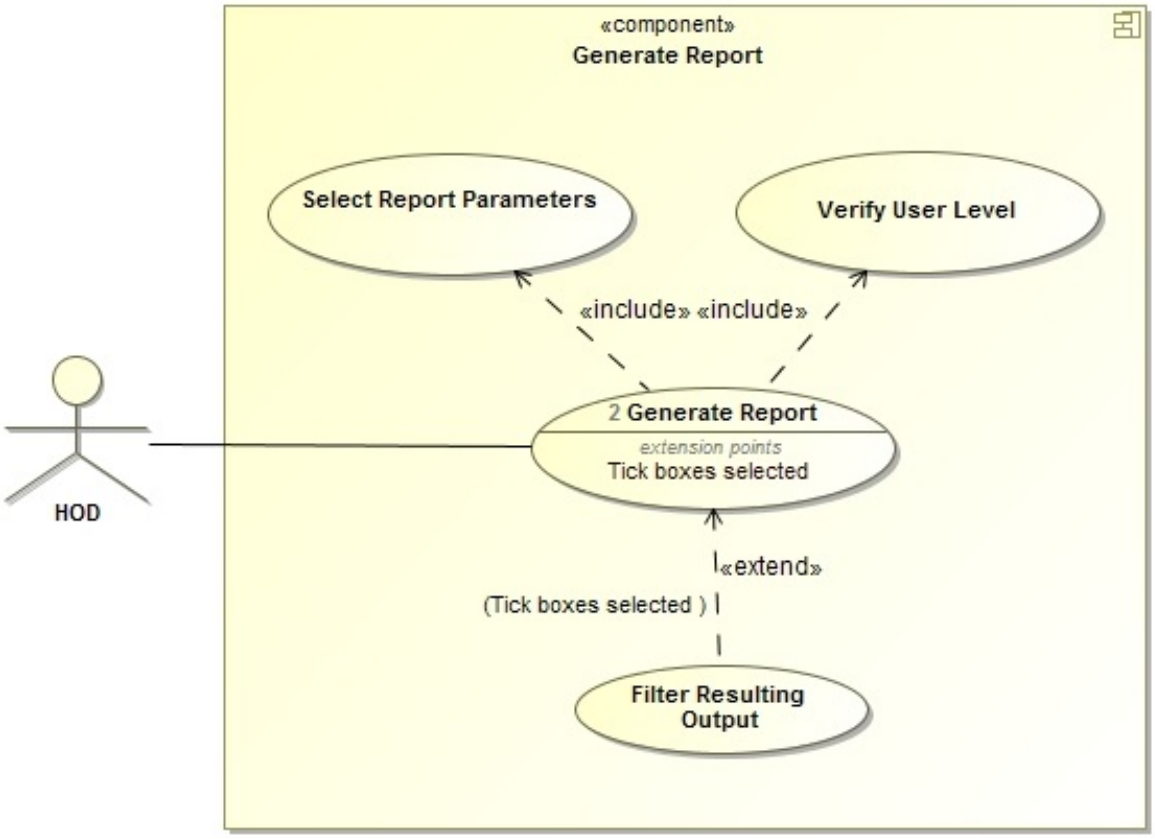
5.3.8 View Profile Use Case Diagram



5.3.9 Edit Profile Use Case Diagram

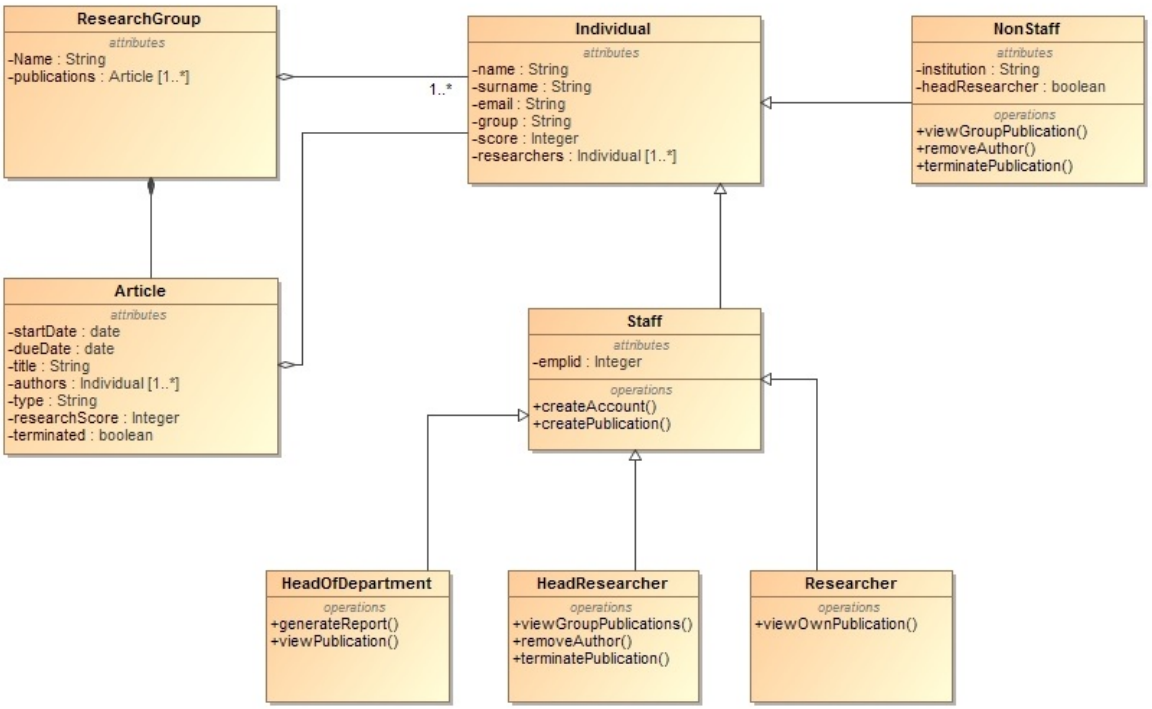


5.3.10 Generate Report Use Case Diagram



5.4 Process specifications

5.5 Domain Model



6 Open Issues