



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	2
2	Vision	3
3	Background	4
4	Architecture Requirements	5
4.1	Access Channel Requirements	5
4.2	Quality Requirements	6
4.3	Integration Requirements	6
4.3.1	Integration Channels	6
4.3.2	Quality Requirements Integration	6
4.4	Architecture Constraints	7
5	Functional Requirements and Application Design	8
5.1	Use Case Prioritisation	8
5.2	Use Case/Services Contracts	8
5.3	Required functionality	8
5.3.1	Registration Use Case Diagram	8
5.3.2	Login Use Case Diagram	9
5.3.3	Logout Use Case Diagram	10
5.3.4	Create Publication Use Case Diagram	11
5.3.5	Add/Remove Author Use Case Diagram	12
5.4	Process specifications	12
5.5	Domain Model	12
6	References	13

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 interdisciplinaty 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 platforms 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 authentication.

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 avaiabe 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

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

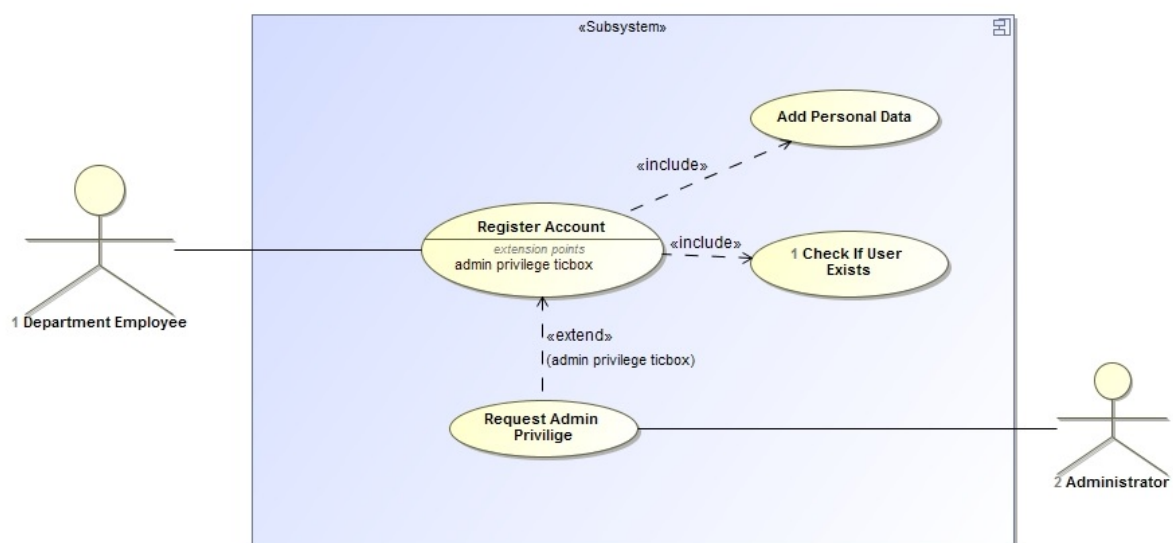
5.1 Use Case Prioritisation

- Registration
- Login
- Logout
- Create publication
- Add author
- Remove author
- Edit publication
- View publication
- View profile
- Edit profile
- Generate summary

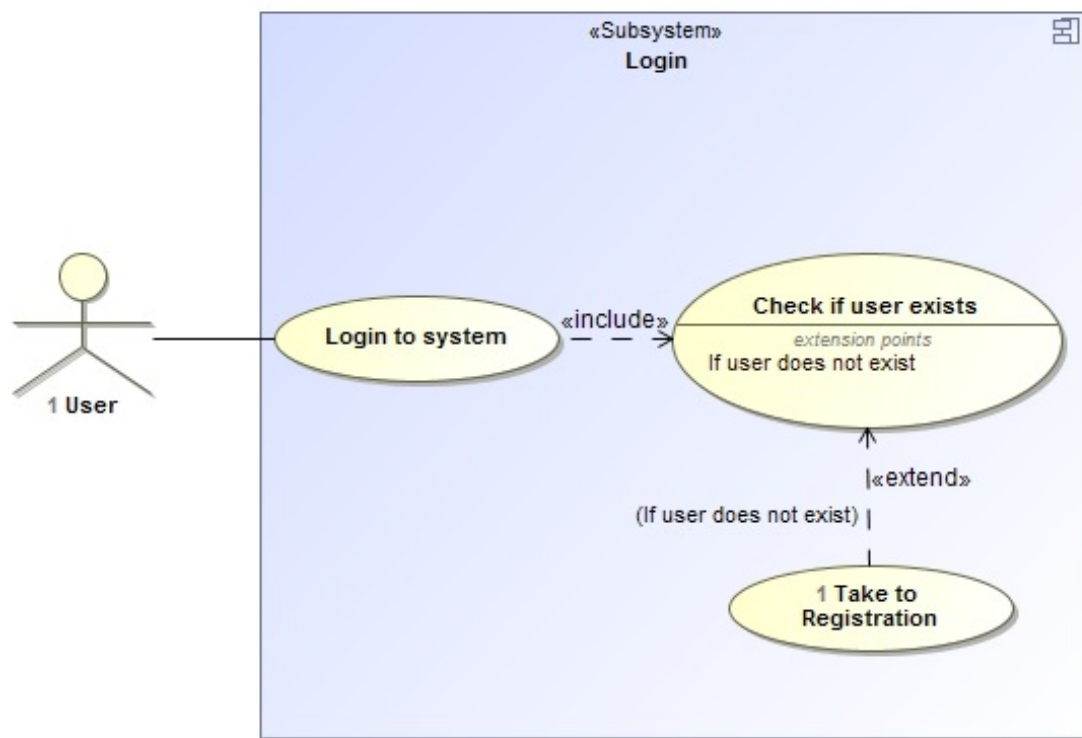
5.2 Use Case/Services Contracts

5.3 Required functionality

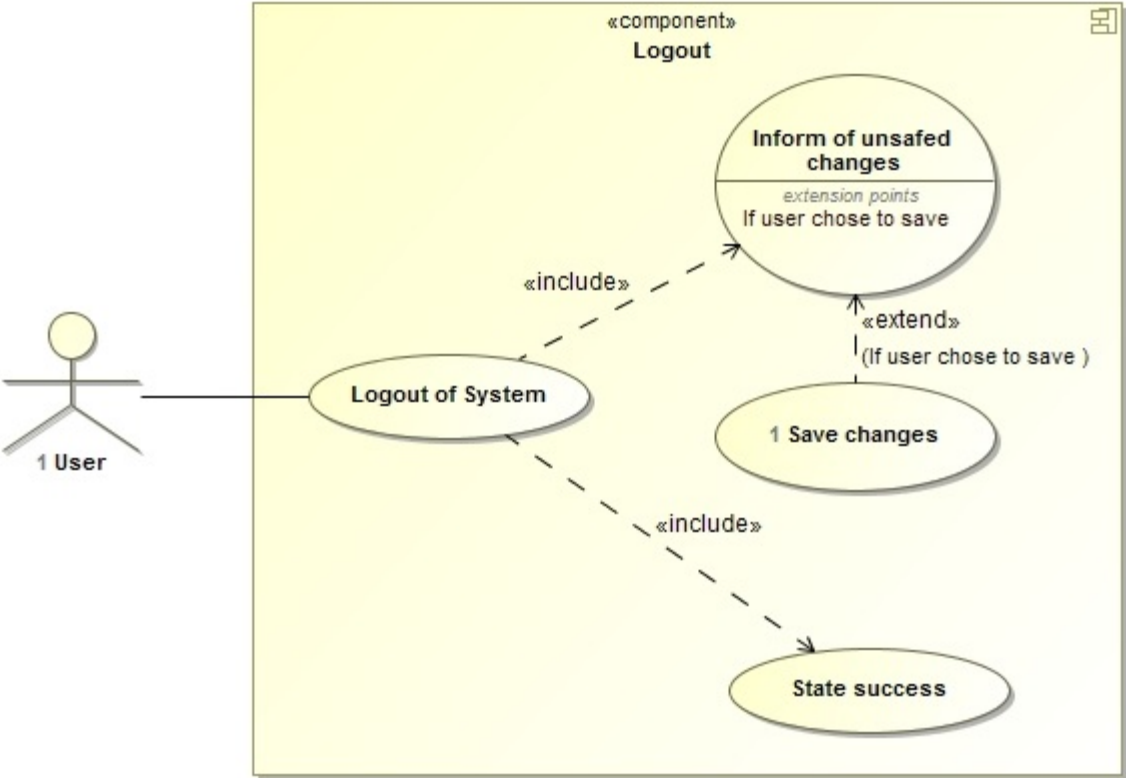
5.3.1 Registration Use Case Diagram



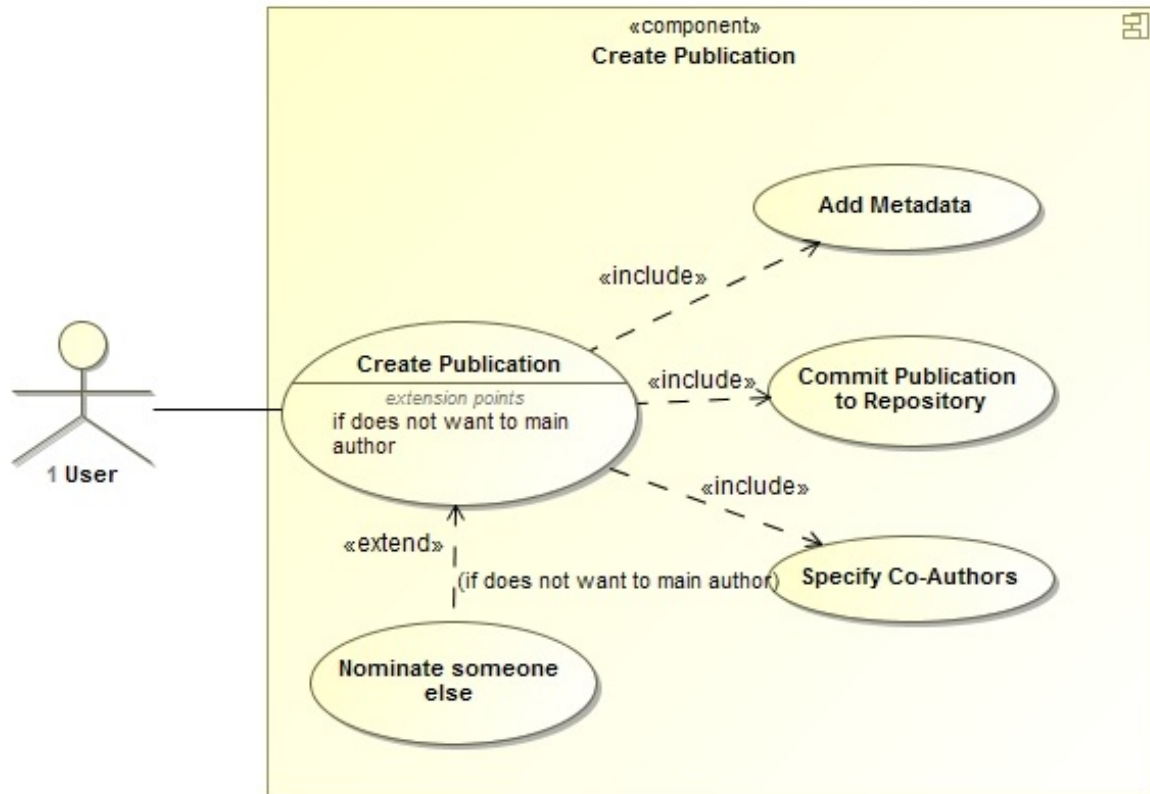
5.3.2 Login Use Case Diagram



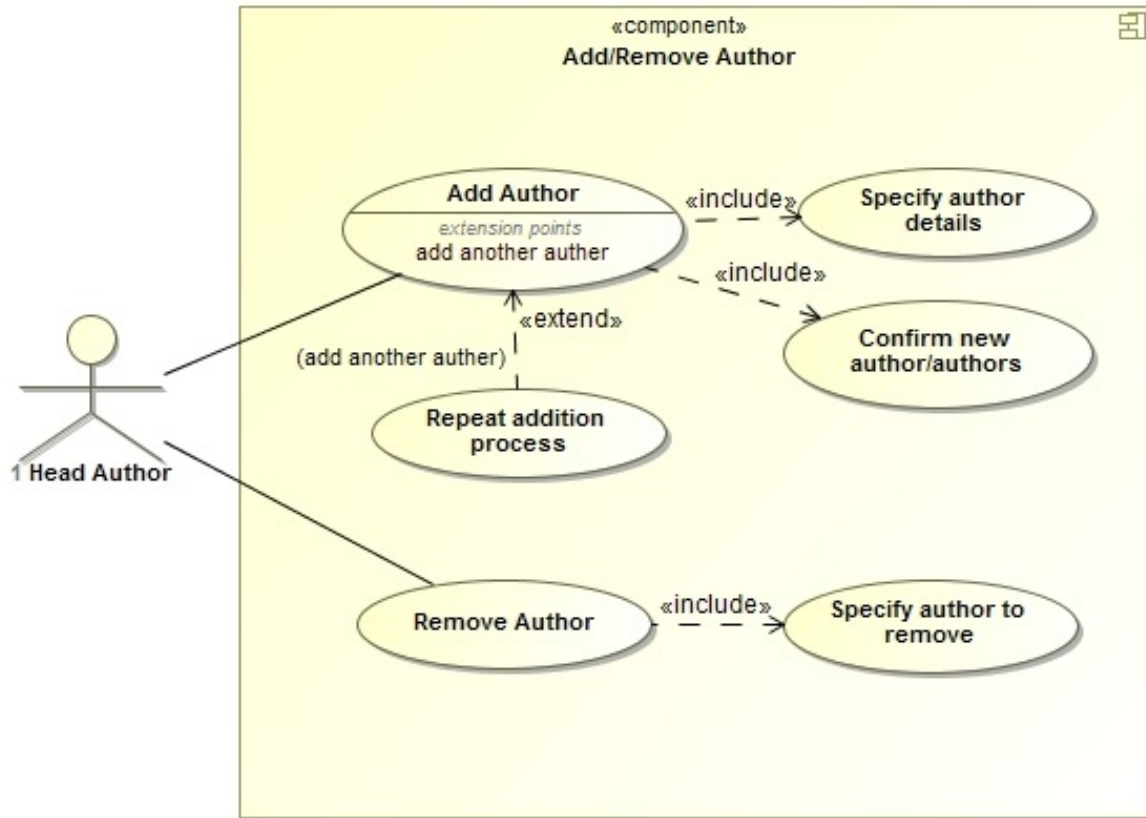
5.3.3 Logout Use Case Diagram



5.3.4 Create Publication Use Case Diagram



5.3.5 Add/Remove Author Use Case Diagram



5.4 Process specifications

5.5 Domain Model

6 References