Software Requirements Specification and Technology Neutral Process Design

(Name of System)

University of Pretoria - Team Charlie

Created By:

Claudio Da Silva
Arno Grobler
Dillon Heins
Charl Jansen Van Vuuren
Priscilla Madigoe
Bernhard Schuld
Keorapetse Shiko

17 February 2016

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to give a detailed explanation and description of the [Name] system. This document will illustrate the purpose as well as the features of said system, the interfaces of the system, the functionality of the system, the constraints under which it must operate and how the system will integrate with external systems. This document has been created for use by the developers of the system, the proposed client as well as any other additional third party collaborators who require to understand the software specifications of the [Name] system.

2 Vision

The client has requested a system which allows researchers at the *University of Pretoria*, specifically within the Computer Science Department, to keep track of the publications which they are currently actively involved with or working on.

The system is required to keep track of historical publications so as to allow researches to maintain an archive of their work.

The system should support the management of the multiple research groups within the department as well as allow the acting heads of the individual research groups to manage their group's members and publications.

Ultimately this system is to be used by the acting Head of Department so as to be able to view all the research groups and their research output. It is a way for the department to ensure that the researchers are meeting their goals as well as the department's goals so as to ensure future funding for the department.

The typical usage scenarios for the desired output from this system would be:

- A UP staff member submitting a research paper to a conference, technical report or conference.
- The submission and acceptance of such a paper is what allows researchers to earn units.
- These units correspond with academic prestige as well as funding for the University of Pretoria and its researchers.
- Departments have predetermined goals which they set out to achieve each academic year.
- The ultimate desired output from this system is the ability to monitor the CS Department's researchers and their contribution towards earning these units.
- This allows the acting Head of Department to award researchers who achieve as well as take note of those who do not.

3 Background

Reasons for the development of this project include but are not limited to:

- Research opportunities:
  - Through the monitoring of units earned by staff members it enables the progression of research opportunities for the *University of Pretoria's* Computer Science Department. By monitoring units earned the department is able to ensure that it meets its goals and is able to secure funding opportunities.
- Opportunities to simplify some aspect of work:
  - The method in use by the department currently is to have all researchers edit the same Microsoft Excel document as their manner of managing and tracking publications and units earned.
  - This method is inefficient as well as error prone and has hence lead to the need to create this system as a means to replace it.
  - This system aims to allow for all researchers to be able to manage their own publications in their own user space. It also allows for the Head of Department to no longer have to use an Excel document to create reports from, instead he/she would be able to use the system to the work for him/her in a far more accurate and efficient manner.
- Problems the client is currently facing:
  - The problem of having all members of a department trying to collaborate on a single Excel document.
  - The problem of having personal and academic information visible to all who have access to this document.
  - The problem of managing this data in such a way as to get valuable and meaningful information out of it quickly and accurately.

4 Architecture Requirements

4.1 Access Channel Requirements

The different access channels through which the system's services will be made available to users as well as other systems are as follows:

- An Application Program Interface residing on a server which will be interfaced with by clients in order to supply services to them. Clients referring to:
  - Human users via an interface
  - External systems using the services provided by the API
- Human users can interface with the system via the use of:
  - a web-based application service
  - an android based mobile application

The interface is required to be lightweight.

4.2 Quality Requirements

- Performance
  - Workload is a maximum of 100 users concurrently.
  - No implementation of concurrent editing of document entries - last saved edit is written to the database.
  - The system should be able to support 100 users updating information at the same time as updating is more intensive than reading.
  - The response time of the system should be fast enough that a user is able to complete their work without frustration. Due to the system being off-line it is reasonable to expect the system's response time to be only limited by the speed of the network.
- Reliability
  - The system should not fail whilst providing critical or important use cases.
  - The system should not fail at all within a time period of at least 6 months.
- Scalability
  - Ability for multiple external systems to connect to the system's API.
  - The system should be able to support a large amount of historical document entries being added to the database.
- Security
  - A hierarchical system will be used to determine the security privileges of users of the system.
  - Passwords are to be hashed using at least sha256 and should be stored as such within the database along with a salt.
  - An inactive user session should be terminated after a period of 10 minutes with no activity.
  - A user who has forgotten their passwords can use a password reset option which will send a one time password to their registered email address so that they may login once using it and reset their password.

- Flexibility
  - The client has stated that the system is not needed to be able to extend to accommodate a greater number of departments.
- Maintainability
  - The system should have as few bugs as possible so as to prevent having to constantly maintain it in the future.
  - The system should be built in a modular way so that all services are decoupled in such a manner that allows for the extension of the system at a later stage.
- Auditability/monitorability
  - Every action performed by a user should be logged and all details about said action should be stored.
  - These actions should be visible to admin users.
- Integrability
  - User's document entries should not be able to be deleted, if it is a case where the document will not be completed it should remain in the system and be terminated.
  - A user with no admin rights should not have access to admin privileges so that the system's data may remain integrable and safe.
  - The system should not ever be in a state where it is under pressure and the data is at risk of becoming corrupted. The system should be designed to handle the pressure for which it has been specified to handle.
- Cost
  - All software used should not be proprietary but rather open source so as to minimise cost as much as possible.
- Usability
  - The interface should be lightweight.
  - The interface should be intuitive to use as well as obey Human Computer Interaction guidelines so that it is efficient and easy to use.

4.3 Integration Requirements

- The first item
- The second item
- The third etc ...

4.4 Architecture Constraints

- The first item
- The second item
- The third etc ...

5 Functional Requirements and Application Design

5.1 Use Case Prioritization

5.1.1 Paper System

Handles the adding, editing and terminating of papers.

User:

- **Add Paper Entry** *Priority: Critical*
  - A user of the system should be able to add a paper entry associated with their user account into the system.
  - **Pre-conditions:**
    - A user should have a registered account and be logged in.
  - **Post-conditions:**
    - A paper entry and all its details should be entered into the database of the system and be associated with a particular registered user.
    - The act of creating a new paper entry should be logged.
- **Edit Own Paper Entry:** *Priority: Critical*
  - A user must be able to edit the metadata associated with their own paper entries.
  - **Pre-conditions:**
    - A user must be logged in.
    - The logged in user must be an author or co-author of the paper.
    - The paper must not have been terminated.
  - **Post-conditions:**
    - The metadata in the database associated with the paper should be updated.
    - The action of editing and all relevant changes to the entry must be logged.
- **Add/Remove Authors from a Paper:** *Priority: Critical*
  - A user should be able to add and remove authors from a paper which they have created or are a co-author of.
  - **Pre-conditions:**
    - The user must be logged in.
    - The user must be the author or a co-author of the paper.
  - **Post-conditions:**
    - There should be an ordered list of unlimited size associated with the paper consisting of the details of the author and co-authors stored in the database.
    - The action should be logged.
- **Add New Author:** *Priority: Critical*
  - A user should have the option to add the details of a new author into the system if the author does not already exist within the system.
  - **Pre-conditions:**
    - The user must be logged in.

    * The author must not already exist.
  - **Post-conditions:**
    * The author and his/her associated details have been added to the system's database.
    * The action should be logged.

- **Search Authors from Database:** *Priority: Critical*
  - The user should be able to search or select from a list of authors within the database to associate the authors' entries with their paper.
  - **Pre-conditions:**
    * The user is logged in.
    * The author is within the system.
    * The author is not within the system.
  - **Post-conditions:**
    * If the author is within the system a result matching the search term or the selected item should be returned.
    * If the author is not within the system no result should be returned and it should be indicated as such. An option to add this author to the system's database should be offered to the user.
    * The action should be logged.

- **Edit Own Paper Progress:** *Priority: Important*
  - A user should have the ability to indicate in terms of a percentage how far they are currently in terms of progress on their paper.
  - **Pre-conditions:**
    * A user should be logged in.
    * The logged in user must be an author or co-author of the paper.
    * The paper must not have been terminated.
  - **Post-conditions:**
    * The field indicating progress within the database must have been updated.
    * The change in progress should be depicted visually to the user.
    * The action should be logged.

- **Terminate Own Paper:** *Priority: Critical*
  - A user must be able to terminate (make inactive) a paper which they feel they are not going to be completing in the future.
  - **Pre-conditions:**
    * A user must be logged in.
    * The logged in user must be an author or co-author of the paper.
  - **Post-conditions:**
    * The paper has been marked as terminated within the database.
    * The action of terminating the paper must have been logged.
    * The paper must no longer be able to be edited by the user.

- **Add/Remove Conference/Journal to Paper:** *Priority: Critical*

  - Whilst creating or editing a paper a user should be able to associate a particular conference or journal with their paper as well as be able to remove an already associated conference or journal whilst editing a paper.
  - **Pre-conditions:**
    * The user must be logged in.
    * The user must be an author or co-author of the paper in question.
  - **Post-conditions:**
    * The paper should be associated with a particular conference or journal or the paper should have had its association removed.
    * The action should be logged.

- **Add New Conference/Journal:** *Priority: Critical*
  - A user must be able to fill in the details of a particular conference or journal if it does not already exist within the system.
  - **Pre-conditions:**
    * The user must be logged in.
    * The conference or journal must not already exist.
  - **Post-conditions:**
    * The conference or journal and all its details have been added to the system's database.
    * The action has been logged.

- **Search Conferences/Journals in Database:** *Priority: Critical*
  - The user should be able to search or select an already existing conference/journal from a list populated through the searching of the database.
  - **Pre-conditions:**
    * The user must be logged in.
    * The conference/journal is within the system.
    * The conference/journal is not within the system.
  - **Post-conditions:**
    * If the conference/journal is within the system a result matching the search term or the selected item should be returned.
    * If the conference/journal is not within the system no result should be returned and it should be indicated as such. An option to add this conference/journal into the system should also be offered.
    * The action should be logged.

Head of Research:

- **xxx** *Priority: xxx*
  - xxx
  - **Pre-conditions:**
    * xxx
  - **Post-conditions:**
    * xxx
    * xxx

5.2 Use Case/ Service Contracts

Each use case is discussed in detail in this section

5.2.1 "use case name"

Description: "description of the use case"

Pre-conditions:
-
-

Post-conditions:
-
-

Request and Results Data structures:
-
-

5.3 Required Functionality

5.4 Process Specifications

Certain use cases require further information with regards to their function.
These use cases are specified further by means of process specification.

5.5 Domain Model

The domain model is described in terms of class diagram.
Class diagrams contain information on the current class such as attributes and relationships to other classes.

6 Open Issues

- The first item
- The second item
- The third etc ...