Software Requirements Specification and Technology Neutral Process Design

P.A.P.E.R.S (Publication And Papers Electronic Repository System)

University of Pretoria - Team Charlie

Created By:

Claudio Da Silva - 14205892
Arno Grobler - 14011396
Dillon Heins - 14035538
Charl Jansen Van Vuuren - 13054903
Priscilla Madigoe - 13049128
Bernhard Schuld - 10297902
Keorapetse Shiko - 12231992

GitHub Repository - Team Charlie

For more information, please click here

17 February 2016

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to give a detailed explanation and description of the P.A.P.E.R.S 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 P.A.P.E.R.S 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 system will be a web based program accessible via browser with an API for Android access.
- The system may not be browser or operating system specific, and should be able to run on any system chosen.
- The system should be handled locally and should not rely on outside internet sources in order to function.

- The database used must be a relational database as the information being stored requires both a standard structure as well as relations on fields.
- Modules must be decoupled as far as possible, allowing as much pluggability and further editing as need be when the system grows larger.
- A proper framework such as Django will probably be used in favour of a standard PHP approach during the program's design.
- The technology used may not be proprietary and should be free and available to use for all.

5 Functional Requirements and Application Design

5.1 Use Case Prioritization

5.1.1 Paper Sub-System

Handles the adding, editing and terminating of papers.

User:

- *Critical:*
  - add paper entry
  - edit own paper entry
  - add/remove authors from a paper
  - add new author
  - search authors from database
  - terminate own paper
  - add/remove conference/journal to paper
  - add new conference/journal
  - search conferences/journals in database
- *Important:*
  - edit own paper progress

Head of Research:

- *Critical:*
  - search all papers within research group

Admin:

- *Critical:*
  - search all papers on system
- *Important:*
  - purge paper from the system

5.1.2 User Sub-System

Handles all actions associated with user accounts such as login, viewing and editing profiles, as well as privileged actions such as adding/removing users to the system and adding/removing users from a research group.

User:

- *Critical:*
  - login
  - view own user profile

Head of Research:

- *Critical:*

  - add/remove users from research group
  - search all users in research group
  - view all user profiles in research group

Admin:

- *Critical:*
  - add new user
  - search all users
  - view all user profiles
- *Important:*
  - set user as active/inactive
- *Nice-to-Have:*
  - purge user from system

5.1.3 Notification Sub-System

Handles the adding and editing of notifications within the system, as well the process of creating and sending notifications to relevant users of the system.

User:

- *Important:*
  - set paper deadline
  - edit paper deadline

System:

- *Important:*
  - send deadline notification
  - send deadline update notification

5.1.4 Reporting Sub-System

Handles the generation of reports with regards to all relevant information contained by the system.

Head of Research:

- *Critical:*
  - generate report for research group

Admin:

- *Critical:*
  - generate report for department
  - dump database to file

5.1.5 Group-Control Sub-System

Handles the creating and removing of research groups, as well as the allocating of heads of research groups and the editing of individual research groups associated metadata.

Head of Research:

- *Critical:*
  - edit research group information

Admin:

- *Critical:*
  - add/remove research groups
  - allocate head of research group

5.2 Use Case/Service Contracts

5.2.1 Paper Sub-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.
  - **Request and Results Data Structures:**
- **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.
  - **Request and Results Data Structures:**
- **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.
  - **Request and Results Data Structures:**
- **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.
  - **Request and Results Data Structures:**

- **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.
  - **Request and Results Data Structures:**

- **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.
  - **Request and Results Data Structures:**

- **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.
  - **Request and Results Data Structures:**

- **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.
  - **Request and Results Data Structures:**

- **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.
  - **Request and Results Data Structures:**

- **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.
  - **Request and Results Data Structures:**

Head of Research:

- **Search All Papers within Research Group** *Priority: Critical*
  - The head of a research group should have access to all the papers associated with their particular research group.
  - **Pre-conditions:**
    * The user must be logged in.
    * The user must have the privilege rights assigned to a head of research.
    * The user must be the head of the particular research group they would like to observe.
  - **Post-conditions:**
    * The user should be able to view all metadata associated with the papers of their research group.
    * The action should be logged.
  - **Request and Results Data Structures:**

Admin:

- **Search All Papers on System** *Priority: Critical*
  - An admin user should be able to view all papers and their metadata on the system.
  - **Pre-conditions:**
    * The user must be logged in.
    * The user must have the privilege rights assigned to an admin.
  - **Post-conditions:**
    * The user should be able to view all metadata associated with all the papers stored on the system.
    * The action should be logged.
  - **Request and Results Data Structures:**
- **Purge Paper From the System** *Priority: Important*
  - An admin user should be able to completely remove a paper from the system if they find the need to do so.
  - **Pre-conditions:**
    * The user must be logged in.
    * The user must have the privilege rights assigned to an admin.
  - **Post-conditions:**
    * The paper and all of its metadata should be permanently removed from the system.
    * The action should be logged.
  - **Request and Results Data Structures:**

5.2.2 User Sub-System

Handles all actions associated with user accounts such as login, viewing and editing profiles, as well as privileged actions such as adding/removing users to the system and adding/removing users from a research group.

User:

- **Login** *Priority: Critical*
  - A registered user of the system must be able to login to the system using their private credentials.
  - **Pre-conditions:**

    - The user must not be logged in.
    - The identifying field used (email address) must match one of an active, registered user of the system.
    - The password entered must match the password associated with the identifying field within the system.
  - **Post-conditions:**
    - If the credentials were incorrect a notification must be displayed, as well as an option to reset passwords.
    - If the credentials were correct:
      - The system must begin a session for the user.
      - The user must gain access to the system according to the set of privileges associated with their account.
    - In both cases the action should be logged.
  - **Request and Results Data Structures:**

- **View Own User Profile** *Priority: Critical*
  - A logged in user should be able to see their own profile and all details associated with it. The user should also be able to view all papers associated with their account.
  - **Pre-conditions:**
    - The user must be logged in.
    - The user's profile being viewed must be the same as the logged in user.
  - **Post-conditions:**
    - The information associated with the user's account should be displayed to them.
    - The action should be logged.
  - **Request and Results Data Structures:**

Head of Research:

- **Add/Remove Users from Research Group** *Priority: Critical*
  - The head of a research group should be able to manage the users associated with that group.
  - **Pre-conditions:**
    - The user must be logged in.
    - The user must have the privileges associated with a Head of Research.
    - The user must be the Head of Research of the particular group which they are wanting to manage.
  - **Post-conditions:**
    - The user must have been able to add other registered users into their research group.
    - The user must have been able to remove other registered users from their research group.
    - The action should be logged.
  - **Request and Results Data Structures:**

- **Search All Users in Research Group** *Priority: Critical*
  - The head of a research group should be able to search all of the users within their research group.
  - **Pre-conditions:**
    - The user must be logged in.

    - The user must have the privileges associated with a Head of Research.
    - The user must be the Head of Research of the particular group which they are wanting to search.
  - **Post-conditions:**
    - The user must have been able to view all of the users associated with their research group.
    - The action should be logged.
  - **Request and Results Data Structures:**

- **View All User Profiles in Research Group** *Priority: Critical*
  - A research group leader should be able to access the profiles and information associated with the users within their research group. In particular the Head of Research should be able to view all paper entries associated with each user.
  - **Pre-conditions:**
    - The user must be logged in.
    - The user must have the privileges associated with a Head of Research.
    - The user must be the Head of Research of the particular group which they are wanting to view profiles of.
  - **Post-conditions:**
    - The head of the research group can access and view all profile information (including paper entries) for each of the users associated with their research group.
    - The action should be logged.
  - **Request and Results Data Structures:**

Admin:

- **Add New User** *Priority: Critical*
  - Admin users should have the ability to add new users into the system.
  - **Pre-conditions:**
    - The user must be logged in.
    - The user must have the privileges associated with an admin account.
  - **Post-conditions:**
    - A new user with all their metadata as well as privileges should be added into the system.
    - The action should be logged.
  - **Request and Results Data Structures:**

- **Search All Users** *Priority: Critical*
  - Admin users should be able to search through all of the users within the system.
  - **Pre-conditions:**
    - The user must be logged in.
    - The user must have the privileges associated with an admin account.
  - **Post-conditions:**
    - The admin user should have been able to search and locate particular registered users on the system.
    - This action should be logged.
  - **Request and Results Data Structures:**

- **Purge User from System** *Priority: Nice-to-Have*
  - An admin user must be able to permanently remove a user and all of their associated information from the system.
  - **Pre-conditions:**
    * The user must be logged in.
    * The user must have the privileges associated with an admin account.
    * The user must confirm they want to continue with the action before the system executes the command.
  - **Post-conditions:**
    * The user and all of their associated information should be removed from the system.
    * The action should be logged.
  - **Request and Results Data Structures:**
- **Set User as Active/Inactive** *Priority: Important*
  - An admin user should be able to set the status of a user to active or inactive, allowing the user to login or preventing them from continuing to interact with the system.
  - **Pre-conditions:**
    * The user must be logged in.
    * The user must have the privileges associated with an admin account.
  - **Post-conditions:**
    * The selected user should have been marked as active/inactive, active allowing the user to login and use the system, inactive preventing them from login in to the system and making changes.
    * The action should be logged.
  - **Request and Results Data Structures:**
- **View All User Profiles** *Priority: Critical*
  - An admin user should be able to view all the information associated with each of the registered users within a system, especially their paper entries and all of the metadata associated with them.
  - **Pre-conditions:**
    * The user must be logged in.
    * The user must have the privileges associated with an admin account.
  - **Post-conditions:**
    * All information, paper entries and their metadata should be available for viewing to the admin user. Upon selecting a particular user all information associated with that user is displayed.
    * The action should be logged.
  - **Request and Results Data Structures:**

5.2.3 Notification Sub-System

Handles the adding and editing of notifications within the system, as well the process of creating and sending notifications to relevant users of the system.

User:

- **Set Paper Deadline** *Priority: Important*

- Upon creating a paper entry a user must be able to set the deadline for when the paper needs to be published by.
- **Pre-conditions:**
  - The user must be logged in.
  - The user must be the author or a co-author of the paper which they would like to set the deadline for.
- **Post-conditions:**
  - A deadline will have been set and stored in the system. It will be associated with the particular paper on which it was set.
  - The action should be logged.
- **Request and Results Data Structures:**

- **Edit Paper Deadline** *Priority: Important*
  - A user must be able to edit the deadline associated with their papers.
  - **Pre-conditions:**
    - The user must be logged in.
    - The user must be the author or a co-author of the paper on which they are editing the deadline.
    - The paper must not have been terminated.
  - **Post-conditions:**
    - The change in the deadline should reflect within the system and be associated with the paper on which it was set.
    - The action should be logged.
  - **Request and Results Data Structures:**

System:

- **Send Deadline Notification** *Priority: Important*
  - The system must be able to notify an author as well as co-authors of a paper of the deadline for the paper.
  - **Pre-conditions:**
    - The user must be logged in.
    - A deadline must have been set for a paper.
    - The deadline must not have passed already.
    - The authors being notified must be associated with the paper the deadline has been set on.
  - **Post-conditions:**
    - Authors of a particular paper are notified via the use of email as to what the deadline for the paper is.
    - The action should be logged.
  - **Request and Results Data Structures:**

- **Send Deadline Update Notification** *Priority: Important*
  - The system must be able to notify an author as well as co-authors of a paper when the deadline for a paper is updated.
  - **Pre-conditions:**
    - The user must be logged in.

    - A new deadline must have been set for a paper.
    - The deadline must not have passed already.
    - The authors being notified must be associated with the paper the deadline has been updated on.
  - **Post-conditions:**
    - Authors of a particular paper are notified via the use of email as to what the updated deadline for the paper is.
    - The action should be logged.
  - **Request and Results Data Structures:**

5.2.4 Reporting Sub-System

Handles the generation of reports with regards to all relevant information contained by the system.

Head of Research:

- **Generate Report for Research Group** *Priority: Critical*
  - The head of a research group should be able to generate a summarised report of all important information related to their research group in terms of the users in the group as well as their papers and all metadata associated with their papers.
  - **Pre-conditions:**
    - The user must be logged in.
    - The user must have the privileges associated with a Head of Research.
    - The user must be the head of the research group which they would like to generate a report on.
  - **Post-conditions:**
    - The head of the research group should have all relevant information returned to him/her in a summarised and easily readable manner.
    - This action should be logged.
  - **Request and Results Data Structures:**

Admin:

- **Generate Report for Department** *Priority: Critical*
  - An admin user should be able to generate a summarised report of all important information related to their department as a whole. This includes summaries of each research group and associated papers, as well as of each individual user of the system and their associated papers.
  - **Pre-conditions:**
    - The user must be logged in.
    - The user must have the privileges associated with an admin.
  - **Post-conditions:**
    - The admin should have all relevant information returned to him/her in a summarised and easily readable manner.
    - The action should be logged.
  - **Request and Results Data Structures:**
- **Dump Database to File** *Priority: Critical*
  - An admin user should be able to at any point dump all of the information within the database into a file for safekeeping, such as a CSV file.

- **Pre-conditions:**
  - The user must be logged in.
  - The user must have the privileges associated with an admin user.
- **Post-conditions:**
  - All information contained in the database should be dumped into a CSV file and be made available to download by the admin user.
  - This action should be logged.
- **Request and Results Data Structures:**

5.2.5 Group-Control Sub-System

Handles the creating and removing of research groups, as well as the allocating of heads of research groups and the editing of individual research groups associated metadata.

Head of Research:

- **Edit Research Group Information** *Priority: Critical*
  - A head of a research group should be able to edit the information associated with their research group.
  - **Pre-conditions:**
    - The user must be logged in.
    - The user must have the privileges associated with a research leader.
    - The user must be the head of the research group which they are attempting to edit the information of.
  - **Post-conditions:**
    - The information associated with the particular research group should have been updated within the system.
    - This action should be logged.
  - **Request and Results Data Structures:**

Admin:

- **Add/Remove Research Groups** *Priority: Critical*
  - An admin user should be able to add or remove research groups from the system.
  - **Pre-conditions:**
    - The user must be logged in.
    - The user must have the privileges associated with an admin user.
  - **Post-conditions:**
    - In the case of adding a new research group which is able to have users added to or removed from should exist within the system.
    - In the case of removing a research group the group should be removed from the system as well as all relationships related to users contained within the group.
    - In both instances the action should be logged.
  - **Request and Results Data Structures:**
- **Allocate Head of Research Group** *Priority: Critical*

- An admin user should be able to allocate a registered user to be the head of a particular research group, in turn allowing for this user to have more privileges within the system.
- **Pre-conditions:**
  - The user must be logged in.
  - The user must have the privileges associated with an admin user.
- **Post-conditions:**
  - The allocated user should now have the necessary rights and privileges to manage and view their allocated research group.
  - This action should be logged.
- **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.

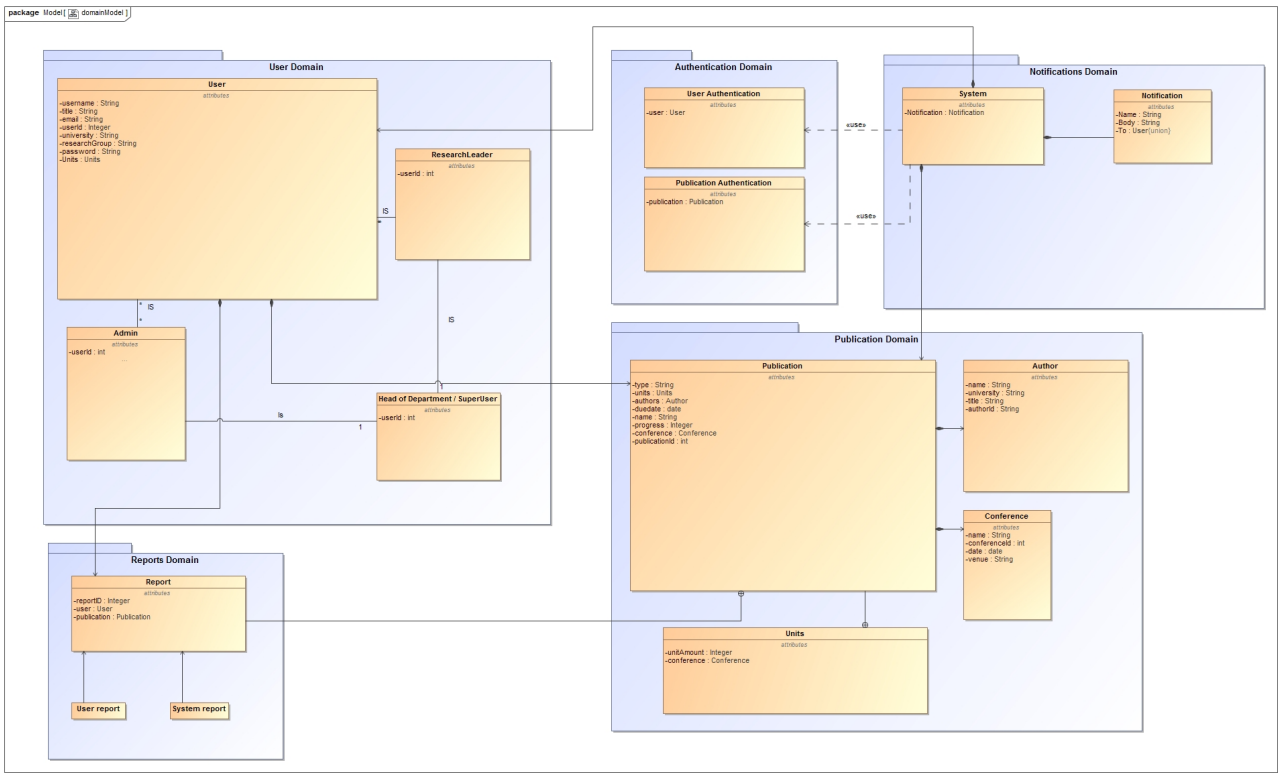

Figure 1: Domain model - Classes related to each domain

Domains include :

- **User:**
  - Classes:
    - User
    - Admin
    - Research Leader
    - Head Of Department
    - Super User
- **Authentication:**
  - Classes:
    - User Authentication
    - Publication Authentication
- **Reports:**
  - Classes:
    - Report
- **Notifications:**
  - Classes:
    - System
    - Notification
- **Publication:**
  - Classes:
    - Publication
    - Conference
    - Units
    - Author

6 Open Issues

Issues that weren't discussed in the client requirements meeting:

-
-