COS 301 Software Requirements Specification

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

1 Introduction

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2 Vision

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3 Background

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

4 Architecture Requirements

4.1 Access Channel Requirements

The system must be able to be accessed via the following channels:

- A desktop application available in the form of a **Windows (7/8/10)** client, a **Linux** client and binaries ready to be built on either system with an interactive GUI
- A web version compatible with all major browsers (eg. *Mozilla Firefox, Google Chrome or Opera*)
- A mobile application developed for Android and compatible with all current and upcoming versions thereof

This will be accomplished by making use of RESTful web services (these being based on the REST, or REpresentational State Transfer architecture). The system itself will accept HTTP requests from any of these channels and create responses in the form of JSON strings, a format easily handled in any one of the aforementioned access systems.

Additionally, the following additional access channels can be added:

- A command line (terminal) based version of the desktop application, which could be suitable for the target audience, who are very technologically capable
- A mobile application developed for Windows Phone and/or iOS.

4.2 Quality Requirements

The following assurances must be made in terms of quality:

- Performance
  1. The server must always provide the minimum data required to fulfill a request. That is to say, were a user to log in to the system, the server should only send the data pertaining to that user to be displayed, no Papers related to his or her Co-Authors or other related parties.
  2. The system must be created with the most minimal and efficient coding practices possible, given that the result must still be reliable and robust.
  3. No actual files are to be stored in the system, lest it negatively affect the performance components of the system itself.
- Reliability
  1. The system must be thoroughly tested on both the client and server side, to ensure it will not cause faults or problems. It is important that no data is lost, thus the coding used to create the system must be defensive and thorough.
- Scalability
  1. The system must be designed such that:
     a-The client is able to handle and display details of a large, potentially infinite number of Publications.
     b-The server is able to handle, display details pertaining to large, potentially infinite number of Users, Authors and Publications.
  2. Modular programming should be used in order to ensure that there are no restrictions in terms of the system's ability to be extended and improved upon at later stages.
- Security
  1. It should not be possible for individuals other than the actual Users to access or modify the system. This means that security has to be ensured in terms of password storage, secure login methods and user management (methods such as re-obtaining password via email should be very carefully guarded).
  2. It should not be possible for Users to make changes to other Users' details, as it is with non-User Authors, unless they are one of a select few Super Users or Administrators.
  3. A publication should not be able to be removed from a system, only edited, unless it is removed by an aforementioned Super User.
  4. A User should not be capable of viewing or editing a publication for which they are not on the list of Authors.
- Flexibility
  1. The system should be capable of reacting quickly to different stimuli. This means (as an example) that if multiple users are concurrently using the system and performing vastly differing tasks which make use of completely different parts of the same system, there should not be any noticeable loss of performance.
  2. The system should be able to perform well even under bulk loads, without loss of data on the way.
  3. It should be possible to add new components or fields to existing components in the system without making major changes.
- Maintainability
  1. The system should be developed with current and maintained technologies, so as to avoid loss of support for as long as possible.
  2. The system should be well documented so as to ensure future developers on the system are capable of maintaining the system without worry.
  3. When changes are made in current technologies, the system should be updated as soon as possible to reflect relevant changes.

4. The modular design of the system must be such that if changes must be made to a part of the system, only that part itself should be changed.

- Monitorability
  1. **All** actions taken that have any affect on the databases stored server side are to be logged.
  2. All logs, current connections and current activity must be viewable by the Super Users in charge of the system.
- Integrability
  1. The system should be designed in such a manner (with modularity and common interfacing methods) that it is capable of having pieces or services plugged in and catered to with minimal effort, such as that of Google Calender, which could be a logical future addition to the system for the sake of deadline maintenance.
- Cost
  1. The tools used to design the system should, as far as possible, be open source, free and not require a license.
  2. In certain cases, paid and licensed software may be suitable for some individual pieces of the system, such as having a Database Management System (DBMS) to handle the storage of data as best possible.
  3. Costs may be created in the form of external hosting for the web service and database storage, should the client desire it to be so.
- Usability
  1. The Users, being staff members, must have easy access from any channel.
  2. The system should be designed in such a manner that the interface is easy to learn and use.
  3. The system should be minimal and avoid having unneeded visuals that could impair a User's ability to use the system comfortably.

4.3 Integration Requirements

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

4.4 Architecture Constrains

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

5 Functional Requirements and application design

5.1 Use case prioritization

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

5.2 Use case/Services contracts

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

5.3 Required functionality

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

5.4 Process Specifications

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

6 Open Issues

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.