



Post-Doctoral Application Management System

Architecture requirements specification

May 21, 2014

Version 1.0

Iteration 1

Prepared for Ms. Cathy Sandis (UP Research Office)
by SoftServe Group

Group members

Kgothatso Phatedi Alfred Ngako (12236731)

Tokologo “Carlo” Machaba (12078027)

Mathys Ellis (12019837)

Change log			
Date	Version	Description	Person
06/03/2014	v 0.0	Added quality requirements in old SRS document	Mathys Ellis
12/05/2014	v 0.0	Document created	Mathys Ellis
13/05/2014	v 0.1	Transferred quality Requirements, Access Channel requirements to document	Carlo Machaba
14/05/2014	v 0.2	Added Integration requirements	Carlo Machaba
15/05/2014	v 0.3	Added to Glossary	Carlo Machaba
15/05/2014	v 0.4	Added Document description, purpose, methodology, conventions and references	Carlo Machaba
17/05/2014	v 0.5	Added more Quality Requirements	Carlo Machaba
20/05/2014	v 0.5	Added more Quality Requirements	Carlo Machaba

Contents

1	Project Repository	4
2	Document description:	5
2.1	Document purpose:	5
2.2	Documentation methodology	5
2.3	Document conventions:	6
2.4	References:	6
3	Architecture requirements	7
3.1	Access channel requirements	7
3.2	Quality requirements	7
3.2.1	Security requirements	7
3.2.2	Availability:	8
3.2.3	Testability:	8
3.2.4	Scalability requirements	8
3.2.5	Auditability:	9
3.2.6	Usability requirements	9
3.2.7	Robustness:	10
3.2.8	Flexibility:	10
3.2.9	Reusability:	10
3.2.10	Maintability:	10
3.3	Integration requirements	11
3.4	Architecture constraints	12
4	Glossary:	13

List of Figures

1 Project Repository

<https://github.com/mox1990/Project-Postdoc.git>

2 Document description:

This document provides the documenting for the software architecture requirements which the application functionality is deployed and executed.

2.1 Document purpose:

The Architecture Requirements Specification provides the architectural requirements that the project will need to follow as well as a set of quality requirements that must be met by the system. The quality requirements have been quantified in order to provide the SoftServe group and the client with a means to test the system for non-functional compliance. This allows the SoftServe group to set up tests in the non-functional testing document that verify that the system complies with the architecture and quality requirements specified by this document. Architecture in this document's context refer to the technological basis and software development styles of the project. Thus this document serves as a contract between SoftServe and the client, Mrs Cathy Sandis of the DRIS of the University of Pretoria in terms of what technologies the project should incorporate as-well as what quality the system should be of once the project is complete.

2.2 Documentation methodology

The documentation and software development methodology used by the project adhere to the guidelines set out by the scum agile methodology. Thus this document has undergone and will undergo various iterations that may extend or reduce the contents of the document.

This document was created using the requirement elicitation techniques and requirement definitions as specified by Klaus Pohl's book Requirements Engineering: Fundamentals, Principles, and Techniques [Dr.Phol, K., 2010]. The requirements, vision and scope were elicited from the following sources:

- Numerous interviews with the client.
- On-line research into UP Post doctoral applications.
- Correspondence with the UP IT department.
- Collecting and analysing various documents such as:
 - The initial project request document
 - Application forms

- Renewal forms
- CV templates
- Approval and recommendation forms

2.3 Document conventions:

- Documentation formulation tool: LaTeX

2.4 References:

- Dr.Phil, K., 2010, *Requirements Engineering: Fundamentals, Principles, and Techniques*, Springer, Heidelberg.

3 Architecture requirements

3.1 Access channel requirements

The only two access channel requirements specified by the client are as follows:

- **All stakeholders:** Users of the system will access the system through a HTML 5 web browser client that is locally installed on a user's computer system or mobile platform. Support for HTML 4.0.1 will also be implemented due to the clients request for compatibility of the system with a older internet enabled systems. The web interface will allow different stakeholders access to different sections of the system based on the roles assigned to their accounts.
- **System Administrator:** The system administrator will require the same as the above but will need to be able to conduct any administration duties through some form of web portal. Above this the system administrator will need to have access to the MySQL database employed by the system. This is mainly for maintenance reasons but also as an alternative if certain unexpected system failures or system generated errors occur.

3.2 Quality requirements

These are the quality requirements as specified by the client and are also in the order of precedence. Where the first takes the highest precedence.

3.2.1 Security requirements

The system will need to be fully secured since the system deals with confidential information such as person information, application statuses, financial data and meeting information. Also since the systems main goal is to provide stable and audible application and renewal process flow the system may not be vulnerable to data tampering or any tampering whatsoever.

The system will have to provide different security roles to the registered users on the system. Any number of roles should be assignable to any user by a administrator with the correct role to allow for flexibility. But in essence a stakeholder may only have access to their section of the application process. The system administrator should be able to view all the sections in the system and should be able to modify them except where they may not.

3.2.2 Availability:

The system's availability on designated networks will depend on the availability of the University of Pretoria's servers that host the system. If the University of Pretoria's servers hosting the system are active and provide access over a designated network then the system must be available over that designated network. The designated networks are defined as the internet and the campus network of the University of Pretoria.

3.2.3 Testability:

The system must be testable. This will be done using unit testing and following the test plan that will be laid out in the testing document of this project.

Unit testing will test each unit in regard to:

- **Preconditions**
- **Post conditions**

The project will also have two phases of testing:

- **Offline:** This is the initial phase of testing and debugging which will be done with pseudo data.
- **Online:** This is the final phase of testing and debugging which will be done with active real time data.

3.2.4 Scalability requirements

The current aim is to create a scalable system that can support 500 to 1000 applicants per year with possible growth. This is in line with the figures given by the client and the growth in the research sector of the university.

The system needs to be scalable in regard to the following factors:

- **Performance:** This is regarded as the speed and responsiveness of the system. The system needs to be able to handle report queries in less than 10 seconds. It should be able to handle any application section processing in less than 3 seconds.
- **Storage:** This is regarded as the growth and shrinking of the data that is stored. The system will need to be able to handle a database that is in the range of 1 GB to 15 GB that has the potential to grow even larger. The reason for this stems from the requirement that the system will support archival functionality and archived data will store the data for long periods of time.

- **Concurrency:** This regarded as the amount of active users on the system at the same time. The system will need to support at least 100+/- concurrent users efficient and effectively since the system requires multiple stakeholders to part take in the application process while there can be multiple applications occurring at the same time.

3.2.5 Auditability:

The system needs to provide an audit trail of all critical actions that occur in the system. Critical actions are considered: user account management operations, login action, logout action and any operation by a user that leads to a change in application data of a particular prospective fellow.

The Audit trail will be in the form of a read-only table stored in the database. It can only be viewed by a user with the correct security role. The system is the only entity that can modify the audit trail where this modification can only be the addition of entries.

3.2.6 Usability requirements

The primary language of the system will be South African English. Any other language support is not considered part of the requirements but the system will be designed to allow for such development in the future.

The system's UI will only consider 2 types of user categories with regard to usability:

- **Trained user:**

This type of user will have to have training in understanding how the system functions and how to use it. Their computer skills will be assumed to be in the range of basic to intermediate. Thus the user interface can allow for certain complexities but these complexities must be kept at a minimal. This user will be regarded as a system administrator. The stakeholders who fall under this category is the DRIS staff members overseeing the application process.

- **Normal user:**

This type of user will have no training. Their computer skills will be assumed to be none or minimal. Therefore the UI that they will have access to will be simplistic and will be as user friendly as possible. The stakeholders that fall under this category will be Prospective fellows, Grant holders, HODs, Deans and Post-doctoral committee members.

3.2.7 Robustness:

The system needs to be robust, with the following areas of focus. It will have the ability to detect if the incorrect input has been entered into the system, the user will be notified immediately and not allowed to continue. If there are any defects with any other components within the system and a task cannot be completed the user will also be notified immediately with the appropriate message sent to them. Users will also re-authenticate using a session object. Emails will be used for this authentication.

3.2.8 Flexibility:

The system is designed in such a way that adding new features can be done without extensive code restructuring. The process of adding new features will require the programmer to create a new module for the new functionality.

3.2.9 Reusability:

The System's components will be used again where applicable. The Login credentials is an example of areas where the system will be reused.

3.2.10 Maintainability:

The system administrator will have the responsibility of adding updates that a programmer may wish to add to the system.

3.3 Integration requirements

The client's primary integration requirement is with regards to the import and export of data that can be used by the PeopleSoft management system of UP as well as other parties. The client's secondary integration requirement is with regards to the complete integration of the system with UP's PeopleSoft management system. With the regards the vision and scope document it should be noted that the knowledge the SoftServe group has of the Peoplesoft system is very limited due to UP's IT department's prerequisite for the project. Above that their are a few other integration requirements that SoftServe in conjunction with the client have purposed. The system has to integrate with the following systems as follows:

- Must be able to create exportable data packages containing the information of prospective and current research fellows that can easily be loaded into the PeopleSoft system or used by other parties. The technology interchange format purposed is CSV. This is to meet the primary requirement of the client. In terms of quality requirements:
 1. **Performance:** The system should be able to handle the export and importing of application information at a reasonable rate of about at least 10 full application data packages in a second. The system should be able to import and generate on-demand user accounts at a rate of 50 per second. For exporting it should be able to do so at 60 accounts per second.
 2. **Scalability:** The system should be able to export or import as much data as specified by the user as long as it does not exceed the size of the database or the size of users secondary storage.
 3. **Audibility:** Any export and import should be recorded by the system.
- Must be able to query PeopleSoft system to retrieve the account information of internal individual or group Stakeholders to create a uniform login experience. As specified by the vision and scope document this will not be possible at the time of writing. Therefore no API can be giving. But the system will most likely make use of the above in point to achieve this requirement.
- Must assign Prospective fellows with a id number in such a way that is integrable with UP Emplid. SoftServe suggests: "f" + 8 digit code.
- Stakeholders that have a Peoplesoft account must use their UP emplid id as a user name. The password may vary.
- Must be able to provide access to the NRF researcher ratings list but not utilize it as the client has stipulated human error is possible in maintaining of those ratings.

The above integration requirements except for the second requirement does need to make use of any protocols or another system's API.

3.4 Architecture constraints

The following architecture constraints have been selected by SoftServe group as being suitable for the system. Please note due to the client's inexperience with software development and the technologies that exist she has allowed the SoftServe group to use their expertises, to decide on the appropriate set of constraints that can meet the above requirements.

1. Database technology: MySQL
2. Programming paradigm: Object-Oriented
3. Programming language: Java
4. Development technology: Java EE
5. Web server software: GlassFish 4.0 server
6. Web interface protocol: HTTPS

4 Glossary:

- **NRF** - National Research Foundation
- **Spreadsheet** - A special type of computer document that is used to represent data in rows and columns.
- **GlassFish** - GlassFish is a web server software package that is very flexible and compatible with Java EE applications.
- **HTML** - Hyper Text Mark-up Language
- **HTTPS** - Hyper Text Transfer Protocol Secure is a higher level network oriented communication rule set that is highly secure and is used by all web browsers.
- **Java EE** - Java Enterprise Edition
- **MySQL** - Is a relational persistence database package that provides all the necessary management tools to run and manage a database server.
- **Object-Oriented** - A programming language style that encapsulates everything as an object instance of a particular class of attributes and methods.
- **Application** - Both a renewal and new fellowship are seen as applications.