



Software requirement specification Version 0.9

Soft**Serve** Group

Members:

Kgothatso Phatedi Alfred Ngako (12236731)

Tokologo “Carlo” Machaba (12078027)

Mathys Ellis (12019837)

Change log			
Date	Version	Description	Person
12/05/2014	v 0.0	Document created	Mathys Ellis
13/05/2014	v 0.1	Added Quality Requirements, Access Channel requirements	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 0.4	Added more Quality Requirements	Carlo Machaba

Contents

1	Document description:	4
1.1	Document purpose:	4
1.2	Documentation methodology	4
1.3	Document conventions:	5
1.4	References:	5
2	Architecture requirements	6
2.1	Access channel requirements	6
2.2	Quality requirements	6
2.2.1	Availability:	6
2.2.2	Security requirements	6
2.2.3	Scalability requirements	6
2.2.4	Testability:	7
2.2.5	Auditability:	7
2.2.6	Usability requirements	8
2.2.7	Robustness:	8
2.2.8	Flexibility:	9
2.2.9	Reusability:	9
2.2.10	Maintability:	9
2.2.11	Efficiency:	9
2.3	Integration requirements	9
2.4	Architecture constraints	10
3	Glossary:	11

List of Figures

1 Document description:

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

1.1 Document purpose:

The Architecture Requirements Specification provides a set of quantitative statements that outline what the Post-Doctoral Application must do in order to comply with the architecture required by the client. 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 project functional requirements.

1.2 Documentation methodology

The documentation and software development methodology used by the project adhere to the guidelines set out by the agile method. 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

1.3 Document conventions:

- Documentation formulation tool: LaTeX
- ERD Crow-Foot notation
- UML 2.0

1.4 References:

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

2 Architecture requirements

2.1 Access channel requirements

All stakeholders: 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. The web interface will allow different stakeholders access to different sections of the system based on the roles assigned to their accounts.

2.2 Quality requirements

2.2.1 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.

2.2.2 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.

2.2.3 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 is 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 efficiently and effectively since the system requires multiple stakeholders to partake in the application process while there can be multiple applications occurring at the same time.

2.2.4 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.

2.2.5 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.

2.2.6 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.

2.2.7 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.

2.2.8 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.

2.2.9 Reusability:

The system is designed in such a way that it will eventually be added to the existing PeopleSoft system currently used by the University. Thus all the components should be have the ability to be reused by the current application system.

2.2.10 Maintability:

In the case where the system has a defect correcting the defect will require

2.2.11 Efficiency:

2.3 Integration requirements

The system has to integrate with the following systems:

- Must be able to create exportable data packages containing the information of approved fellowships that can easily be loaded into the CSC PeopleSoft system.
- Must be able to query CSC PeopleSoft system to retrieve the account information of internal individuals/group Stakeholders to create a uniform login experience.
- Must assign Prospective fellows with a id number in such a way that is integrable with UP Emplid. SoftServe suggests: "f" + 8 digit code.
- Must be able to access and utilize the NRF researcher ratings.

2.4 Architecture constraints

The following architecture constraints have been selected as being suitable for the system.

1. Database technology: MySQL
2. Development technology: Java EE
3. Web server: Apache Tomcat
4. Application server: GlassFish
5. Web interface protocol: HTTPS

3 Glossary:

- **CV** - Curriculum Vita
- **PDF** - Portable Document Format file
- **NRF** - National Research Foundation
- **API** - Application Programming Interface
- **Spreadsheet** - A special type of computer document that is used to represent data in rows and columns.
- **HTML** - Hyper Text Mark-up Language
- **Java EE** - Java Enterprise Edition
- **Use case** - A visual depiction of a service or group of services.
- **Application** - Both a renewal and new fellowship are seen as applications.
- **PhD** - A doctoral degree in a particular field of study.