ELEN4009 - SOFTWARE ENGINEERING LABORATORY EXERCISE 2

Julio Baeta (710066), Nomakhosi Ndebele (671480), Ryan Robinson (453764), Timothy Rokebrand (458960)
School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

I. SOFTWARE REQUIREMENTS SPECIFICATION

In order to ensure the success of the final software produced, it is necessary to assemble a document that is known as a *Software Requirements Specification*. Such a document will set out exactly what the software is capable of, what interfaces the user will be interacting with when using the software, what the user will require to run the software including hardware requirements and other programs that will need to be installed as well as a variety of other aspects regarding the software.

The software requirements specification for the Online Postgraduate Application Approval System to be used by the School of Electrical and Information Engineering at the University of the Witwatersrand can be found in the same repository as this document.

II. THE USER INTERFACE - FRONT END

The complete application is received, from SIMS, by the School's page. The application is first assessed by the postgraduate officer (Ms. Adam) to ensure that all the information is present. If not, a message is sent back to SIMS to inform them that the application is incomplete. When all the information is present the postgraduate officer gives the relevant supervisors access to the applicants details and proposal. It is now up to the supervisor to make a decision on whether the applicant has been successful. If the application is rejected reasons must be given. Once the supervisor has made his decision, the head supervisor (Professor Hofsajer) is able to view the decision and either finalise the decision or query it with the relevant supervisor.

1) Applying Student Interface (Timothy Rokebrand):
HTML This module will consist of coding, the main responsibility to take the proposed design and ensure that the functionality is as expected (i.e. the buttons do what is expected, hyperlinks navigate to the correct pages etc.), linking the html code with CSS and assisting with the linking of the front-end to the back-end

2) Administrative Officer Interface (Nomakhosi Ndebele):
This aspect will be broken down into: CSS coding, the main responsibility is the design and layout of the page i.e. what everything will look like and what needs to be on each page, as well as linking the html code with CSS.

III. DATABASE - THE BACK END

1) Student Information Database (Ryan Robinson):
This database will be used to store all of the information pertaining to the student such as their name, student number, academic history and contact details. For the course of this project, the assumption will be made that all of the applying student's information is already present on the university's database. This implies that the student's information has already been entered by the Wits Students Information Management System (SIMS) once an application has been received by the School of Electrical and Information Engineering.

The database management system that will be used for the student's information will be MySQL. This database management system will also be used in conjunction with Apache webserver software. Since the webpage (user) interface or the 'front-end' of the system will be making use of HyperText Markup Language (HTML), the scripting language Hypertext Preprocessor (PHP) will be used as a means of creating a link between the webpage and the relevant databases.

2) Document Storage Database (Julio Baeta):
This database will be used to store any relevant documents that are required to be submitted by the student as part of their postgraduate application process. This includes a curriculum vitae (CV), a copy of their degree as well as a copy of academic records if they are applying from a different university (did not complete their undergraduate degree at the University of the Witwatersrand). The assumption that all documentation has already been entered into the university's database by SIMS will also be made for this portion of the database aspect. The database management system that will be used for the student's documentation will be Solr. Alternatively, MySQL could be used for this purpose as well by making use of Binary Large Objects or BLOBs. However, this has been known to cause what is known as 'table bloat,' whereby MySQL uses too much memory and causes performance issues in addition to resulting in a variety of other problems [1].

A Solr and Lucene combination was selected since it is open source, has a large community and are packaged together as a single product by Apache. Solr also provides an advanced full text search, which will allow postgraduate officers to find the proposal though key word searches as opposed to tediously searching though the database. Solr uses XML, JSON and HTTP. Apache Zookeeper makes the database easily

scalable. This fact, combined with the open source nature should greatly extend the product's life and allow it to be quickly changed according to the client's needs. Solr also allows the use of security SSL/TLS, which is a necessary requirement for connecting to the Internet in modern times [2]. Thus the most optimal back-end configuration will make use of MySQL to create the basic database, which will then be used in Solr/Lucene with its extended features.

IV. SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

In order to give the project the best possible chance of succeeding, it is essential that a choice of software development methodology be selected. Due to the nature of the limited timespan allocated to the project and the subsequent need to complete tasks in the quickest possible manner, the SDLC methodology employed was that of SCRUM. This methodology enabled the team to consistently reevaluate the way in which they worked between (weekly) sprint meetings and to ensure that the most effective way was adopted in subsequent sprints.

V. BUILDING THE REQUIREMENTS BACKLOG

To ensure that the project would be considered by the 'customer' (the School of Electrical and Information Engineering at the University of the Witwatersrand) to be successful, it was essential that a meeting was set up with the primary user of the system. For this project, this would be the postgraduate officer within the school. As such, a meeting was arranged with the postgraduate officer and a subsequent product backlog constructed by the software development team.

VI. REFERENCES

[1] Voelkel, J. (2014). Performance: WP-Options Table Bloat and It's Performance Effects. Available from: http://justinvoelkel.me/problem-solved- wp-options- table-bloat- and-its- performance-effects/. Last accessed 7th March 2016.

[2] The Apache Software Foundation. Solr Features. Avaliable from: http://lucene.apache.org/solr/features.html. Last accessed 6th March 2016.