UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COS301 Mini Project

Software Requirements Specification

Group 5B

Keagan Thompson *u13023782*
Matthew Russell *u12047822*
Maret Stoffberg *u11071762*
Patience Mtsweni *u11116774*
John-Latham van der Walt *u13222580*
Name(s) Surname *u********
Name(s) Surname *u********
Name(s) Surname *u********

Github Repository
`https://github.com/keagsthom/COS301_Group5B_Phase2`

1 History

Date	Version	Description
04-03-2015	Version 0.1	Document Template created - Keagan

Contents

List of Figures

2 Introduction

2.1 Purpose

2.2 Background

3 Architecture Requirements

3.1 Access Channel Requirements

3.2 Quality Requirements

3.2.1 Scalability

The software system must be able to handle the registration and data of all the users registered on the LDAP server. If several users attempt to do the same action simultaneously, like commenting in a thread, the actions must appear to happen one at a time. The system must not have a maximum or minimum amount of users that may use the system at a time

3.2.2 Performance requirement

3.2.3 Maintainability

The system must be easy to update and maintain.

3.2.4 Reliability and Availability

The system must be usable on any (modern) web browser and operating system.

3.2.5 Security

3.2.6 Monitorability and Auditability

3.2.7 Testability

3.2.8 Usability

The Buzz sytem must be easy to use. It can make use of UI metaphors like a having a "post-it" symbol for the post operation or an envelope for the messages link. The most common used operations must perform very quickly.

The system must be easy to learn so new users can adapt to it without help. The validation and error messages must be clear and understandable.

3.2.9 Integrability

The system mut be integrated with the LDAP server. All the data on the server like names, photos registered modules, etc are used in the buzz system.

3.3 Integration Requirements

3.4 Architecture Constraints

The architecture constraints were indicated in the briefing document and the following is a list of technologies we are going to use in the project:

- Linux (Operating System): Linux is more stable than windows on servers. It is free and does not need added costs. Thus it makes a good OS to use. There is a lot of free software that runs on it, but there is also software that only runs on Windows.

- Git (Version Control System): Git is good for group projects and information share. It can show the progress and the groups efforts in completing the project. It is not always easy to use, sometimes it can break and work need to be restored from a previous version.

- JavaEE (Java Platform Enterprise Edition): For server programming Java EE would suffice, it is reliable and easier to use. Even though it is a bit slower and is resource intensive. It makes a good framework that incorporates JPA, JPQL and JSF.

- JPA (Java Persistence AP): JPA uses JPQL so it would be better to use this, it operates against entity object rather than with database tables. Most of the developers have not yet practised with this technology which leads to inexperience with this.

- JPQL (Java Persistence query language): Is used to make queries against entities stored in a relational database and additionally retrieve objects from the database on the server. It would be used to retrieve user data and etc. Most of the developers have not yet practised with this technology which leads to inexperience with this.

- JSF (Java Server Faces): Is a Java specification for building component-based UI for web applications and exposing them as server side Polyfills. This works with AJAX. Most of the developers have not yet practised with this technology which leads to inexperience with this.
- HTML (Hypertext Mark-up Language): This is going to be code that has to be generated for the website where the user can navigate through, the browser takes this code, interprets it and give a UI for the user.
- AJAX (Asynchronous JavaScript and XML): It is easier to use, rather than having to make our own function for everything. This is client side processing so that we dont hassle the servers with little tests.
- CSS (Cascading Style-sheet): This is the coding platform we are going to use to make the website more attractive and to fill the emptiness of the website.

4 Conclusion

5 References