Assignment 2: Software Requirements Specification and Technology Neutral Process Design
COS 301 Team Alpha Project
Version 1.0

Amy Lochner 14038600
Avinash Singh 14043778
Christiaan Nel 14029368
Christiaan Saaiman 12059138
Gerard van Wyk 14101263
Marc Antel 12026973
Themba Mbhele 14007950

https://github.com/AvinashSingh786/COS301-Alpha.git

University Of Pretoria

February 2016

Contents

1 Software Architecture Documentation

This document defines

1.1 Architecture requirements

In this section extract the architectural requirements from the software requirements including

1.1.1 Architectural scope

1.1.2 Quality requirements

1.1.3 Integration and access channel requirement

As it is possible that the system may have concurrent users, it is favourable that there be interfaces such as a web application or website, as well as a mobile platform applications for the various different mobile devices through which the system can be accessed.

1.1.4 Architectural constraints

It is desired that this system will encourage authors to collaborate with other authors on similar topics and to expand the users base knowledge of ongoing research projects.

2 Architectural patterns or styles

2.1 Architectural tactics or strategies

2.2 Use of reference architectures and frameworks

2.3 Access and integration channels

Specify and quantify each of the quality requirements which are relevant to the system

- Performance - How well the system can cope with extreme load.
- Security - Minimising the possibility of leaking information, maintaining data integrity, and avoiding session hijacking.
- Maintainability - Can the system be managed without downtime.
- Scalability - Can the system be used for large amount of users without it affecting performance.
- Efficiency - Can the system be optimized to produce faster and better results.

- Flexibility - Can the system be easily changed or modified.
- Reliability - Is the system able to cope with the load in order to provide constant access, ensure the system is always active and can provide all functionality.
- Integrability - Will the system be able to integrate with other technologies.
- User Friendly - Does a user easily understand how to use the system.
- Concurrency - Can multiple users perform actions at the same time.
- Low Cost, Reduced data usage - Is it suitable for users with low budget and capped Internet.
- Updatability - Can the system have version updates to introduce new features or functionality whilst maintaining old data.

  - The different Web protocols used.
  - API - UML Interfaces.
  - Google Calender and Email integration.
  - Mobile Scalability and functionality Integration.
  - Venue and Publication integration.

2.4 Technologies

This specifies any constraints, the client may require, to be placed on the system architecture. Such constraints may be:

- HTML (Hypertext Markup Language)
- AJAX (Asynchronous JavaScript and XML)
- JavaEE (Java Platform Enterprise Edition)
- JavaScript (Functionality to HTML)
- PHP (Server Side Scripting)
- MySQL (Database Manager)
- Android (Mobile Devices)
- IOS (Mobile Devices)
- Apache (Web Server)
- Linux/Windows (Operating System)