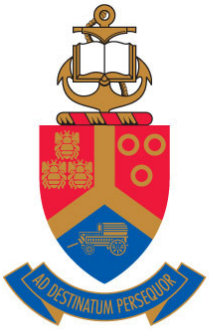

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science

COS 301 - Software Engineering

COS 301 - Mini Project

Author:	*Student number:*
Sean Hill	u12221458
Liz Joseph	u10075268
Jessica Lessev	u13049136
Kale-ab Tessera	u13048423
Sphelele Malo	u12247040
Diana Obo	u13134885
Armand Piterse	u12167844

March 4, 2015

DECLARATION OF ORIGINALITY

UNIVERSITY OF PRETORIA

The University of Pretoria places great emphasis upon integrity and ethical conduct in the preparation of all written work submitted for academic evaluation.

While academic staff teach you about referencing techniques and how to avoid plagiarism, you too have a responsibility in this regard. If you are at any stage uncertain as to what is required, you should speak to your lecturer before any written work is submitted.

You are guilty of plagiarism if you copy something from another author's work (e.g. a book, an article or a website) without acknowledging the source and pass it off as your own. In effect you are stealing something that belongs to someone else. This is not only the case when you copy work word-for-word (verbatim), but also when you submit someone else's work in a slightly altered form (paraphrase) or use a line of argument without acknowledging it. You are not allowed to use work previously produced by another student. You are also not allowed to let anybody copy your work with the intention of passing if off as his/her work.

Students who commit plagiarism will not be given any credit for plagiarised work. The matter may also be referred to the Disciplinary Committee (Students) for a ruling. Plagiarism is regarded as a serious contravention of the University's rules and can lead to expulsion from the University.

The declaration which follows must accompany all written work submitted while you are a student of the University of Pretoria. No written work will be accepted unless the declaration has been completed and attached.

Full names of students: _____

Student numbers: _____

Topic of work: _____

Declaration

1. I understand what plagiarism is and am aware of the University's policy in this regard.
2. I declare that this assignment report is my own original work. Where other people's work has been used (either from a printed source, Internet or any other source), this has been properly acknowledged and referenced in accordance with departmental requirements.
3. I have not used work previously produced by another student or any other person to hand in as my own.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.

SIGNATURES: _____ DATE: _____

Architectural Requirements

Buzz Space link

For further references see gitHub. March 4, 2015

Contents

1 Change Log

2 Introduction

Designing and implementing a system as the one in question is a vast amount of work and requires a lot of planning and thinking. All the functional requirements needed for the system were covered in the previous phase. In this phase all the non-functional requirements will be covered. These are all categorized as architectural requirements.

2.1 Purpose

The main purpose of this document is to decompose the system at hand and fully describe the decomposition in terms of subsystem responsibilities, the mapping of the subsystem to hardware and the dependencies among the subsystems. Policy decisions that need to be made is also described in this document. This included decisions such as access control, data storage and control flow.

The document will cover all the non-functional aspects of the requirements including:
the architectural scope comprising the responsibilities,
the quality requirements for the software system,
the access and integration requirements, and
the architectural constrains.

2.2 Project Scope

The scope of this phase is understanding the functional requirements covered in the previous phase and using them to build and derive the architectural requirements needed for the implementation of the system.

3 Access Channel Requirements

3.1 Platforms

4 Quality Requirements

4.1 Performance

Expectation:

Prioritization:

Example in System:

4.2 Reliability

4.3 Scalability

Expectation:

System should be able to handle (at the very least) the number of registered students on the CS LDAP system.

Prioritization:

Critical

Example in System:

Number of simultaneous posts/comments, number of online users etc.

4.4 Security

4.5 Flexibility

4.6 Integratability

Expectation:

System must integrate well with any host website.

Prioritization:

Important

Example in System:

Users of the host site should be able to access the forum easily. It should seem as if its merely an extension of the host site.

4.7 Maintainability

Expectation:

System must be easy to maintain/modify

Prioritization:

Important

Example in System:

Modular separation of concerns, low coupling and reduced dependencies.

4.8 Auditability

Expectation:

System must be auditable

Prioritization:

Important

Example in System:

The plagiarism and netiquette checker should have no problem editing the page to ensure that every post is appropriate for the website.

4.9 Monitorability

4.10 Cost

4.11 Usability

Expectation:

The system must be easy to use by the average user.

Prioritization:

Critical

Example in System:

Navigating the site should be straight forward. Buttons and links should be clearly visible and placed in easy to find locations. Naming convention should be descriptive and unambiguous.

5 Integration Requirements

5.1 Integration Channel

5.2 Protocols

5.3 API Specifications

5.4 Quality Requirements

6 Architecture Constraints

6.1 System

The BuzzSpace has to be integrated into the CS departments website.

Reason: The users credentials from the CS LDAP will be used for authentication purposes in the BuzzSpace e.g. to register on the BuzzSpace to login.

6.2 User

- Only users registered for a module from the CS department will be allowed to participate in discussions on the discussion forums.
- Users who are registered as guests on the BuzzSpace will only be allowed to read and observe the discussion forums.

  Reason: To manage who has access to the BuzzSpace.

6.3 Time

If a user is logged in and remains inactive for more than 30mins the user will have to login again to post on the forum.

Reason: To ensure security for user Authentication.

6.4 Technologies

- HTML

  To create the skeleton of the BuzzSpace's front-end to enable users to login/logout create forums and participate in discussions.
- Javascript and AJAX

  Used for the BuzzSpace's front-end to verify the login details of each user and will keep track of user login and particiaptions on discussion forums.
- PHP

  Allows the front end to communicate with the CS LDAP database to access user credentials for authentication purposes for logging in to the BuzzSpace.
- JavaEE
- JPA and JPQL

6.5 Architectural Patterns/Frameworks

- Layer (object-oriented design)
- MVC (Model View Controller)
- Peer-to-Peer Network
- Services Oriented Architectures

7 Policies

8 Traceability Matrix

9 References

PIETERSE, V. 2015. COS301: Mini project Buzz. In: Lecture notes issued online. University of Pretoria. Pretoria, South Africa

10 Glossary