



UNIVERSITEIT VAN PRETORIA  
UNIVERSITY OF PRETORIA  
YUNIBESITHI YA PRETORIA

---

COS301 MINI PROJECT  
ARCHITECTURAL REQUIREMENTS  
SPECIFICATION  
Group 5a

Matthew Gouws *u11008602*  
Tsepo Ntsaba *u10668544*  
Werner Mostert *u13019695*  
Semaka Malapane *u13081129*  
Andrew Parkes *u12189139*  
Name Surname *uXXXXXXXXXX*  
Name Surname *uXXXXXXXXXX*  
Name Surname *uXXXXXXXXXX*

Version  
March 9, 2015

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Vision</b>	<b>3</b>
<b>3</b>	<b>Background</b>	<b>3</b>
<b>4</b>	<b>Architectural Requirements</b>	<b>3</b>
4.1	Access channel and integration requirements . . . . .	3
4.1.1	Access channels . . . . .	3
4.1.2	Integration Channels . . . . .	3
4.2	Architectural responsibilities . . . . .	5
4.3	Quality requirements . . . . .	6
4.3.1	Scalability . . . . .	6
4.3.2	Performance requirements . . . . .	6
4.3.3	Maintainabilty . . . . .	6
4.3.4	Reliability and Availability . . . . .	6
4.3.5	Security . . . . .	6
4.3.6	Monitorability and Auditability . . . . .	6
4.3.7	Testability . . . . .	6
4.3.8	Usability . . . . .	6
4.3.9	Integrability . . . . .	6
4.4	Architecture constraints . . . . .	7

# 1 Introduction

text goes here

# 2 Vision

text goes here

# 3 Background

text goes here

# 4 Architectural Requirements

## 4.1 Access channel and integration requirements

### 4.1.1 Access channels

**Human access channels**

**System access channels**

### 4.1.2 Integration Channels

**Integration Channel**

- The Enterprise service bus will be used. The bus is a subsystem that facilitates the communication between other subsystems. This means that subsystems only need to communicate through the ESB. This significantly lowers the amount of necessary connections and it processes the messages, meaning other subsystems need not waste time processing messages. If a subsystem is replaced or added, the only changes that need to take place is the creation of a new interface between the new system and the bus. This makes the system much more flexible and makes communication much more simple.

**Protocols Used**

- Hypertext Transfer Protocol  
HTTP is the standard way and method to transfer method of web pages over the internet by using the protocols TCP/IP (Transmission Control Protocol and Internet Protocol) to manage the Web transmission. In use of Buzz all posting, registering and even just access to the forums will make use of these protocols.

- Mail Protocols POP3 and SMTP

Post Office Protocol (POP) will probably not be used within the system but is used in receiving emails. Simple Mail Transfer Protocol (SMTP) will be used though, the protocol is used in the sending of emails by making use of a SMTP server. In the case of Buzz all notifications that are not present within Buzz will be sent through email such as account verification and notification of changes within a users account. Finally if a user loses account details all information and lost password would be sent over email.

- File Transfer Protocol

File Transfer Protocol (FTP) is the method of copying files over the internet or a network but more simply put allows for the transfer of information from one computer to a remote computer. In terms of Buzz we are not creating a File Server but rather with all the images profile pictures and any other documentation that are uploaded to different profiles for other users to use make use of the File Transfer Protocol.

## 4.2 Architectural responsibilities

- The System must be able to provide concurrent clients to read threads, post threads and update threads.
- The system should be able to store all threads and posts, as well as who posted, and whether a thread has been deleted.
- The system should provide an integration environment to allow for multiple deployment
- The system could allow for persistent data storage for easy 'Remember Me'
- Storage of archived thread

## **4.3 Quality requirements**

### **4.3.1 Scalability**

### **4.3.2 Performance requirements**

### **4.3.3 Maintainability**

### **4.3.4 Reliability and Availability**

### **4.3.5 Security**

### **4.3.6 Monitorability and Auditability**

### **4.3.7 Testability**

### **4.3.8 Usability**

### **4.3.9 Integrability**

## 4.4 Architecture constraints

### Reference Architecture

- Java EE (Enterprise Edition) is specified as the chosen Reference Architecture to use.

#### Details

Java Platform, Enterprise Edition or Java EE is Oracle's enterprise Java computing platform. The platform provides an API and runtime environment for developing and running enterprise software, including network and web services, and other large-scale, multi-tiered, scalable, reliable, and secure network applications. Java EE extends the Java Platform, Standard Edition (Java SE), providing an API for object-relational mapping, distributed and multi-tier architectures, and web services.

#### Comments

### Other Software Technologies

- JPA (Java Persistence API)

#### Details

The Java Persistence API (JPA) is a Java programming language application programming interface specification that describes the management of relational data in applications using Java Platform, Standard Edition and Java Platform, Enterprise Edition.

#### Comments

- JPQL (Java Persistence Query Language)

#### Details

The Java Persistence Query Language (JPQL) is a platform-independent object-oriented query language defined as part of the Java Persistence API (JPA) specification. JPQL is used to make queries against entities stored in a relational database. It is heavily inspired by SQL, and its queries resemble SQL queries in syntax, but operate against JPA entity objects rather than directly with database tables.

#### Comments

- JSF (JavaServer Faces)

#### Details

JavaServer Faces (JSF) is a Java specification for building component-based user interfaces for web applications and exposing them as

server side Polyfills.] It was formalized as a standard through the Java Community Process and is part of the Java Platform, Enterprise Edition.

#### **Comments**

- HTML (HyperText Markup Language)

#### **Details**

HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages.

#### **Comments**

- AJAX (Asynchronous JavaScript and XML)

#### **Details**

Ajax is a group of interrelated Web development techniques used on the client-side to create asynchronous Web applications. With Ajax, web applications can send data to and retrieve from a server asynchronously (in the background) without interfering with the display and behavior of the existing page.

#### **Comments**

- CSS

#### **Details**

Cascading Style Sheet will be used for the formatting and styling of the discussion forum. It will ensure separation of the content from presentation.

#### **Comments**

- Github

#### **Details**

Github will be the platform used by the developers to collaborate. This will be where all the code will be pushed so that the developers can be able to work together from remote locations and be able to combine all their code/work.

#### **Comments**

### **Operating System**

- Linux

#### **Details**

The forum will be designed to work on any Operating system. However, Linux is ideal for certain functionalities therefore the forum will work optimally on Linux Operating Systems. The



web server will be hosted on a Linux machine as Linux is widely considered to be the best operating system for web servers.

**Comments**

**Deployed Environments**

- Web Based Interface

**Details**

Users will be able to navigate to the application using a web browser from any compatible device, such as a tablet; smart-phone or computer.

**Comments**

This is a very suitable environment since it allows for a large measure of ease of access.

- Android (out of scope - for future reference)