

Architecture Requirements for the Buzz System

Git: https://github.com/FrikkieSnyman/Phase2_Group3B

COS301 Group 3b

Andreas du Preez 12207871

Jason Evans 13032608

Sebastian Gerber 12213749

Baruch Molefe 12260429

Kyhle Ohlinger 11131952

Renette Ros 13007557

Michelle Swanepoel 13066294

Frikkie Snyman 13028741

March 2015

Contents

1	Software Architecture	2
1.1	Architectural Requirements	2
1.1.1	Scope of Architectural Responsibilities	2
1.1.2	Access and Integration Requirements	2
1.1.3	Architectural Responsibilities	2
1.1.4	Quality Requirements	2
1.1.5	Architecture Constraints	2
1.2	Architectural Patterns or Styles	2
1.3	Architectural Tactics or Strategies	2
1.4	Use of Reference Architectures and Frameworks	2
1.5	Access and Integration Channels	2
1.6	Technologies	2

1 Software Architecture

1.1 Architectural Requirements

1.1.1 Scope of Architectural Responsibilities

Persistence.

Database:

In the scope of the BUZZ system and taking into consideration the type of data the system should store, we have concluded to make use of a Relational Database Management System (RDBMS). The BUZZ system will store only structured data with strong relations between content (e.g. Threads and Social Tags). Persistence of data is also closely related to the auditability of the BUZZ system, hence deleted threads will not be removed completely from the database, but instead be archived (marked as hidden) for possible later retrieval.

1.1.2 Access and Integration Requirements

1.1.3 Architectural Responsibilities

1.1.4 Quality Requirements

- **Scalability:**
- **Performance Requirements:**
- **Maintainability:**
- **Reliability and Availability:** The reliability and availability of the system is important for all users, i.e. The students, tutor, lecturers, and administrative staff.

If the system is unavailable, then users would not be able to access the required information on the system. This could have serious implications on all users.

Reliability and Availability of a system is essential, this could be achieved by:

- Identifying ways to prevent system failure, and if the system does fail, have measures in place to start a failover, so that the system is still accessible.
- Identifying ways to handle the system when external systems, to which the system is connected, i.e communication networks, external databases, are unavailable. This could be done by having some sort of offline system functionality.
- **Security:** The security of the system is important for all users, i.e. The students, tutor, lecturers, and administrative staff.

The purpose of security is to protect the information stored in the system, whether it be the systems information or user data, and prevent unauthorised access to and/or modification of the information.

In order to enforce security:

- The system should authentication and authorization to prevent spoofing of users identities.
- Input validation is important in preventing damage caused by malicious input.

- Sensitive data should be encrypted and user activity, i.e. Guest and Authorised users, should be monitored to prevent loss or damage of data.
- The system should log all user interaction with the system, this would be beneficial when auditing the system.
- The system should have multiple safe guards in order to protect access to data.
- System timeouts could be considered, in the unlikely event of DOS or DDOS attacks.
- **Monitorability and Auditability:** The system will be monitored by the administrative staff and users that are specifically assigned the role of maintaining the system.

This will help ensure that users abide by the netiquette and plagiarism policies. Any infringement of these policies should be captured/logged for later use by the administrative staff.

The audit logs would be made accessible to the administrative staff through specific requests to the system.

- **Testability:**
- **Usability:**
- **Integrability:**

1.1.5 Architecture Constraints

1.2 Architectural Patterns or Styles

1.3 Architectural Tactics or Strategies

1.4 Use of Reference Architectures and Frameworks

1.5 Access and Integration Channels

1.6 Technologies