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 Tatctics 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

Persistance.

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 auditibility 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:**
- **Security:**
- **Monitorability and Auditability:**
- **Testability:**
- **Usability:**
- **Integrability:**

1.1.5 Architecture Constraints

1.2 Architectural Patterns or Styles

1.3 Architectural Tatctics or Strategies

1.4 Use of Reference Architectures and Frameworks

1.5 Access and Integration Channels

1.6 Technologies