UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COS301
Mini Project
Architectural Requirements Specification

Group 1B

David Breetzke	*u12056503*
Tienie Pritchard	*u12056741*
Ephiphania Munava	*u10624610*
Paul Engelke	*u13093500*
New Member	*uxxxxxxxx*
New Member	*uxxxxxxxx*
New Member	*uxxxxxxxx*
New Member	*uxxxxxxxx*

GitHub: Visit

Version 1
March 7, 2015

Contents

6 Architecture Constraints 11

1 Access and Integration Requirements

[insert text]

2 Access Channels

[insert text]

3 Integration Channels

[insert text]

4 Architectural Responsibilities

[insert text]

5   Quality Requirements

5.1   Scalability

[insert text]

5.2   Performance Requirements

[insert text]

5.3   Maintainability

[insert text]

5.4   Reliability and Availability

[insert text]

5.5   Security

[insert text]

5.6   Monitorability and Auditability

[insert text]

5.7   Scalability

[insert text]

5.8   Testability

5.8.1   Type of Quality:

System Quality

5.8.2   Priority:

Critical

5.8.3 Description:

Testability measures how easy it is to create testing standards for a system and its individual components, theses standards are tested to evaluate if a criteria has been met. Thus software testability is the point to which the software system supports testing in some context. Hence if the software testability is high finding faults in the system is easier.

5.8.4 Stake Holder:

- Persons who operates the system : Administrator, Maintenance Operator and Tech-team.

- Persons who benefits from the system : Lectures, Teaching Assistance, Tutors, Students and Guest.

5.8.5 Context:

- Stimulus : The testing is performed by tester (these might be system testers, integration testers and even the end user).

- Artifact : The target of the attack can be the system or the data in the system.

- Environment This attack can come from the user of the system or an outsider like a hacker.

- Response : The system has to authorize certain actions and responses for each of the given tasks.

- Response Measure : The measure of the system and it functionality before, during and after the attack.

5.8.6 Measurable Specification:

- Understand-ability : The point at which the component of the system that being tested is self-explanatory.

- Separation of concerns : The point, at which the component of the system that's being tested has a well-defined responsibility.

- Observe-ability : The point, at which the component of the system thats being tested become possible to discern the test results.

Component Under Test
Is a test that restrictions the scope of the used software to a ration of the system that is being tested.

- Controllability : The point, at which the system thats being tested becomes possible to control the state of the component under test as required.

- Isolate-ability : The point, at which the system thats being tested becomes possible for the component under test to be tested in isolation.

5.9 Usability

5.9.1 Type of Quality:

User Quality

5.9.2 Priority:

Critical

5.9.3 Description:

Usability describes how the system meets the requirements of the stake holders by being instinctive on condition that good access for incapacitated users is provided, and resulting overall great user experience. Thus software usability refers to the ease of use and learn-ability of the system. In other words how user-friendly is it.

5.9.4 Stake Holder:

- Persons who benefits from the system : Lectures, Teaching Assistance, Tutors, Students and Guest.

5.9.5 Context:

- Stimulus : The stake holder wants to use the system efficiently.

- Artifact : The target of use which is the system.

- Environment : This stake holders action with which the usability quality is concerned.

- Response : The system provides the stake holder with features that the stake holder will or might need.

- Response Measure : The response of the system and it functionality is measured by the number of errors, number of problems encountered, user satisfaction and time taken per task.

5.9.6   Measurable Specification:

Cognitive Modelling Methods
Cognitive Modelling Methods involves creating computational method in order to estimate the time it will take people to perform given tasks.

- Human Processor Model : This model was developed to calculate how long it takes an individual to perform a task. A table is given with amount of times a user would take to execute an action i.e. move eye to look at the screen 230ms.

- Keystroke level modelling : Very much like the GOMS version but simplifies assumptions so that calculation time and complexity is reduced.

- Heuristic Evaluation : This measurable method involves bringing in a set of experts that will evaluate the usability of your system based on their prior knowledge and research.

5.10   Integrability

5.10.1   Type of Quality:

User Quality

5.10.2   Priority:

Critical

5.10.3   Description:

The capability of making components of a single system that is isolated and developed separately work together correctly.

5.10.4   Stake Holder:

- Persons who operates the system : Administrator, Maintenance Operator and Tech-team.

5.10.5   Context:

- Stimulus : The stake holder wants to use to configure, maintain, update and use the system.

- Artifact : The target of use which is the system.

- Environment : This stake holders action with which the system has to conform to.

- Response : The system allows the stake holders to interact with it.

- Response Measure : The response of the system and it functionality after the system has been refactored.

5.10.6   Measurable Specification:

- Spread load across time : This can be achieved by making use of Queuing as a tactic.

- Reduce communication load : This can be accomplished by using strategies like compression, batching and course grained services.

- Fault prevention : This can be addressed by making use of persistent messaging.

- Component Application : We encounter naming service, trader service and interface/ contract repository which is addressed with integerability.

- Security : The use of encryption and restricting accessibility will address the security insures we might come across.

6 Architecture Constraints

[insert text]