COS 301 ASSIGNMENT
GROUP 2 B

Duran Cole (13329414)
Johannes Coetzee (10693077)
Estian Rosslee (12223426)
Edwin Fullard (12048675)
Herman Keuris (13037618)
Martha Mohlala (10353403)
Motsoape Mphahlele (12211070)
Xoliswa Ntshingila (13410378)

February 2015

1 Introduction

2 Architecture Requirements

2.1 Architectural Scope

2.2 Quality Requirements

2.2.1 Scalability

- Manage resource demand
- Scale out resources
- The system must be able to operate effectively under the load of all registered students within the department of Computer Science. (+-2000 concurrent users)

2.2.2 Performance requirements

- Throughput: The rate at which incoming requests are completed
- Manage resource demand
- Response times

2.2.3 Maintainability

- Regression
- Corrective maintenance: Reactive modification of a software product performed after delivery to correct discovered problems.
- Adaptive maintenance: Modification of a software product performed after delivery to keep a software product usable in a changed or changing environment.
- Perfective maintenance: Modification of a software product after delivery to improve performance or maintainability.
- Preventive maintenance: Modification of a software product after delivery to detect and correct latent faults in the software product before they become effective faults.

2.2.4 Reliability and Availability

- Prevent faults
- Detect faults
- Recover from faults

2.2.5 Security

- Detect attacks
- Resist attacks
- Recover from attack
- Authentication
- Authorization
- Minimize access
- All communication of sensitive data must be done securely using HTTPS.
- All system functionality is only accessible to users who can be successfully authenticated through the LDAP system used by the department of Computer Science.

2.2.6 Monitorability and Auditability

- Logs: Logs system activities such as the time a user logged into/out of the system
- Each action on the system must be recorded in an audit log that can later be viewed and queried.
- Information to be recorded must include:
  - The identity of the individual carrying out the action
  - A description of the action
  - When the action was carried out

2.2.7 Testability

- Controllability: The degree to which it is possible to control the state of the component under test as required for testing.
- Understandability: The degree to which the component under test is documented or self-explaining.
- Test driven development
- Each service provided by the system must be testable through a unit test that tests:
  - that the service is provided if all pre-conditions are met, and
  - that all post-conditions hold true once the service has been provided.

2.2.8 Usability

- Efficiency
- Ease of use
- Learnability
- Satisfaction (How pleasant it is to use the system?)
- The average student should be able to use the system without any prior training.
- Initially only English needs to be supported, but the system must allow for translations to the other official languages of the University of Pretoria to be added at a later stage.

2.2.9 Integrability

- Must be able to intergrate with existing systems and systems which may want to be added.

2.3 Integration and access channel requirements

2.4 Architectural Constraints

3 Architectural patterns or styles

4 Architectural tactics or strategies

5 Use of reference architectures and frameworks

6 Access and integration channels

6.1 Integration Channel Used

6.1.1 REST - Representational State Transfer

- Uses standard HTTP and thus simpler to use.
- Allows different data formats where as SOAP only allows XML.
- Has JSON support
  - faster parsing.
- Better performance and scalability with the ability to cache reads.
- Protocol Independent, can use any protocol which has a standardised URI scheme.

6.2 Protocols

6.2.1 HTTP - Hypertext Transfer Protocol

- Standard web language.
- Easy to write pages.

6.2.2 PHP

- Allows dynamic pages to be built.
- Easy integration of JavaScript and HTML with PHP functions.

6.2.3 IP - Internet Protocol

- Allows Communications between users.
- In charge of sending, receiving and addressing data packets.

6.2.4 SMTP - Simple Mail Transfer Protocol

- Sends emails.
- MIME (Multi-purpose Internet Mail Extensions) which allows SMTP to send multimedia files.

6.2.5 TSL - Transport Layer Security

- Alternative to SSL
- Newer and more secure version of SSL.

7 Technologies