Design Documentation
CS Marks System

Version: 3.0

Christo Brits u11080923
Zuhnja Riekert u12040593
Mamelo Soepela u12094847
Eduan Bekker u12214834
Christopher Crossman u10134842
Chris Moodley u10457489

URL - https://github.com/eduanb/COS301$_Phase2$

March 13, 2014

Contents

1 Software architecture design

1.1 Choices of Technologies

- LDAP
  - LDAP will be used to authenticate users of the University.
  - Security
    - Users will only be able to gain access to the system after they have been authenticated through the LDAP system, which would contain all the users of the University of Pretoria, guaranteeing security in the system.
- Python, Django
  - This will provide a channel to the database from the database to the web and mobile interfaces.
- Android, Java
  - This will provide a mobile channel to the system.
  - Accessibilty
    - Since the system needs to be accessible on Android, the Android SDK, using Java to program it will be used.
- MySQL, SQL
  - MySQL will be used to store the roles of the users, the users marks as well as the audit logs.
  - Security
    - The system will require an audit log, this will be stored in a database that cannot be edited by anyone.
    - The Database will contain all the roles of all the users as well as containing all the marks.
- HTML 5 and CSS 3
  - This will provide the front end of the web interface.
  - Usability
    - HTML 5 as well as CSS 3 will be used to ensure that the front end of the system is structured properly to ensure a legible and usable Web Interface.

- CSV
  - CSV will be used to import and export information to and from the system.
  - Scalability
    - The information supplied by lectures will need to be processed quickly, so data inputed into the system should be in the form of a CSV file to ensure fast processing.
- PDF
  - Performance
    - The system will have to be able to deliver reports at high speed, particularly at no more than 10 seconds, therefore PDF is the most appropriate format for this performance requirement.

1.2 Chosen Frameworks

- Django web framework
  - Enforces the Model-View-Controller Design Pattern.
  - It will help ease the complexity of writing a database driven web application.
- Django Object-Relational Mapper
  - This will be used to guarantee persistence to a relational database.

1.3 Chosen Protocols

- SOAP
  - Simple Object Access Protocol. This will be used to exchange structured information in the implemantation of web services.
  - SOAP relies on XML, HTTP and SMTP to function to its full extent.
- XMLP
  - Extensible Markup Language Protocol. This will be used to encapsulate XML data that allows for distributable extensibility.

  - XML is a markup language that is both human- and computer-readable.

- HTTPS
  - Hypertext Transfer Protocol Secure. This is used for secure communication over a computer network. HTTPS is the result of layering HTTP on top of TLS.

- HTTP
  - Hypertext Transfer Protocol. This will be used for the exchange and transfer of structured text that uses logical links known as hyperlinks, this is also known as hypertext.

- TLS
  - Transport Layer Security, previously known as Secure Sockets Layer. This is designed to provide communication security over the internet.

- SMTP
  - Simple Mail Transfer Protocol. This will be used for electronic mail transmission.

- LDAP
  - Lightweight Directory Access Protocol. This will be used for accessing and maintaining distributed directory information.

1.4 Chosen Libraries

- Generating PDFs
  - iText is a PDF library that allows you to create, adapt, inspect and maintain documents in the Portable Document Format.
  - iText is supported by Java and Android applications with PDF functionality.

- LDAP Integration
  - UnboundID LDAP SDK for Java is fast, powerful, user-friendly, and free. It is also supported by Android. This will be very efficient to use for Java and Android integration with LDAP.

- XML Marshalling
  - Spring is a robust Java application framework that contains an O/X Mapping feature that translates Java objects into an XML document and vice versa. Spring also supports Model-View-Controller pattern.

2 Application design

2.1 Back-end System

Generating PDFs

- Database Design: [width=4in, height=4in]./DatabaseDiagram/Backend$_{D}atabase.jpg$

2.2 Web Application

2.3 Android Application