Software Requirements Specification
Mark Management System

Version: 1.0

Organization:
University of Pretoria: Group 5

GitHub:

https://github.com/Lecton/Group_2.git

Authors:
Mr Lecton Ramasila (10637291)
Mr Joas Mogale(10354167)
Mr Juandre Barnard (11061015)
Miss Talifhani Sikhitha (10346504)
Mr Kabelo Mamadi (10301004)
Mr Collen Mphabantshi (10404687)
Mr Renato de Jesus (28113561)

March 13, 2014

Contents

1 Software architecture design

1.1 Programing language

Python
The system will be written in python and this language will handle the connections to the server, create the sockets and send and receive requests and response events to the various interfaces thought the different platforms. The program will be written on the Django framework which implies that it will be high performing and elegant and since it is a Web application, it is not restricted by any hardware architecture or software incompatibilities that may prevent other applications from running successfully.

Java
The Java language is used to build the android interface using the trivial android API's as well as several libraries that perform tasks such as animation and smoothness for the application that is to run on any interface that runs the appropriate android operating system or simulation software that can render the application.

1.2 Database

The database of choice is MySQL. This database is very convenient as it is broadly used; hence it is quite convenient and easily pluggable to a lot of applications. The system will use MySQL to do the following.

- The CS MySQL database will be provides exhaustive information to access course and module information.
- Django's object-relational mapper provides the core of the framework, currently supports MySQL. Django generates production-ready CRUD interfaces from the ORM. The automatic creation of database tables and database abstraction layer from Pythonic model definition is really quite elegant and probably Django's most distinctive feature.
- MySQL Audit logging triggers will handle all the audit in the system.

1.3 Application server

Django application server
The Django web server will host and handle the interactions from various interfaces with the python program, thus the service will handle all the requests and response events.

Apache Web server
The current services that are on the computer science web site (www.cs.up.ac.za) run on a Apache Web Server, thus this server will be used as an interface between the computer sciences offered services such as lecturer and tutors names and the python written application that servers as a mark serving utility. The apache HTTP server will manipulate authentication and authorization using three of its process:

- Authentication type
- Authentication provider
- Authorization

These modules implement core directives that are core to all authentic modules.

1.4 Framework

Django application server
The Django framework uses the model view controller pattern (MVC) and it emphasis the component of reusability, preventing one from repeating themselves.

1.5 Application server

- Factory The system different interfaces, factory object helps provide necessary information about the type of object the client needs.
- Model View Controller The purpose of systems is to retrieve data from a data store in the database(LDAP) and display it for the user(marker or lecturer) on provided interfaces. After the user changes the data, the system stores the updates in the data store. Because the key flow of information is between the data store and the user interface, you might be inclined to tie these two pieces together to reduce the amount of coding and to improve application performance

1.6 Protocols

SOAP
The standard protocol HTTP makes it easier for SOAP model to tunnel across firewalls, proxies without any modifications to the SOAP protocol and connection between different channels of the system. SOAP requires less plumbing code for services design, (i.e., transactions, security, coordination, addressing, trust, etc.) hence the system it won't just support complex operations, which require conversational state and contextual information to be maintained. With the SOAP approach, developers need not worry about writing this plumbing code into the application layer themselves
The system needs consistency and integrity in processing information, SOAP has a large number of supporting standards for security, reliability, transactions. This will be applicable with LDAP system used by the department of Computer Science.

HTTPS
It provides authentication that uses an encrypted transport protocol underneath to handle all communication of sensitive data

1.7 Libraries

- MuPDF is also small, fast, and yet complete. It supports PDF 1.7 with transparency, encryption, hyperlinks, annotations, searching and more. It

also reads XPS and OpenXPS documents. MuPDF is written modularly, so features can be added on by integrators if they so desire.

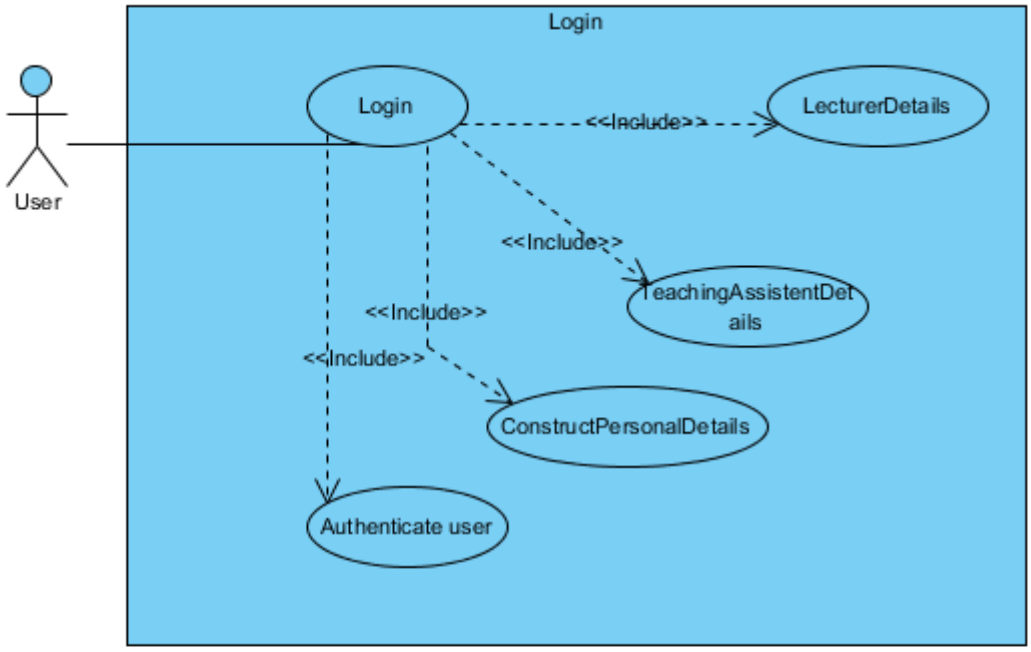
- CSV File Reading and Writing(version 2.3) format and python imports and exports format for spread-sheets and databases of the students mark-sheet.
- Jquery makes HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.

2 Application design

2.1 Specifcations of lower levels of granularity

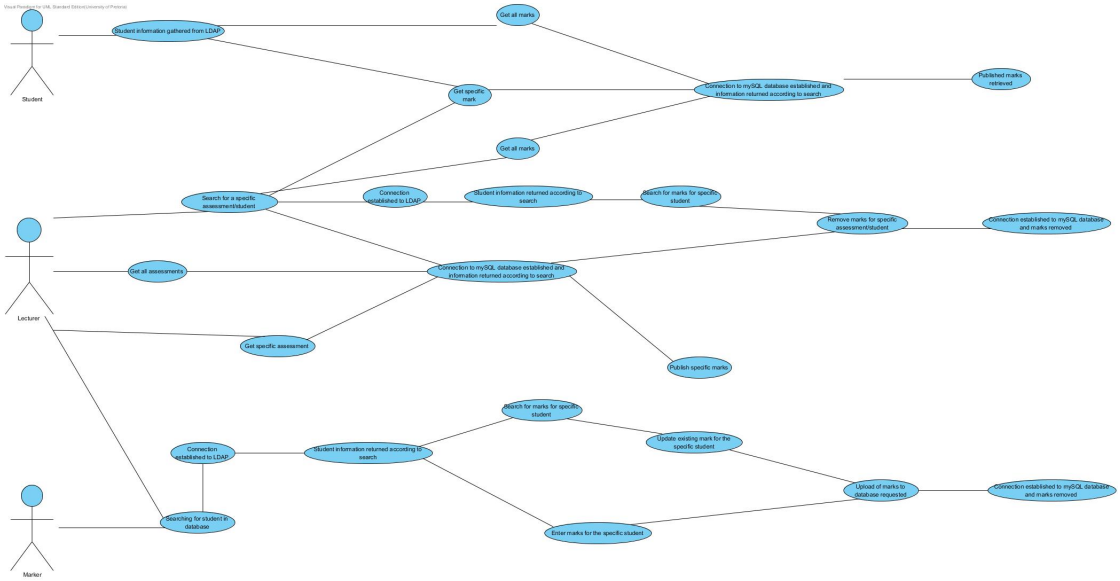
Login diagram

The user have to first login to have full access to the system.



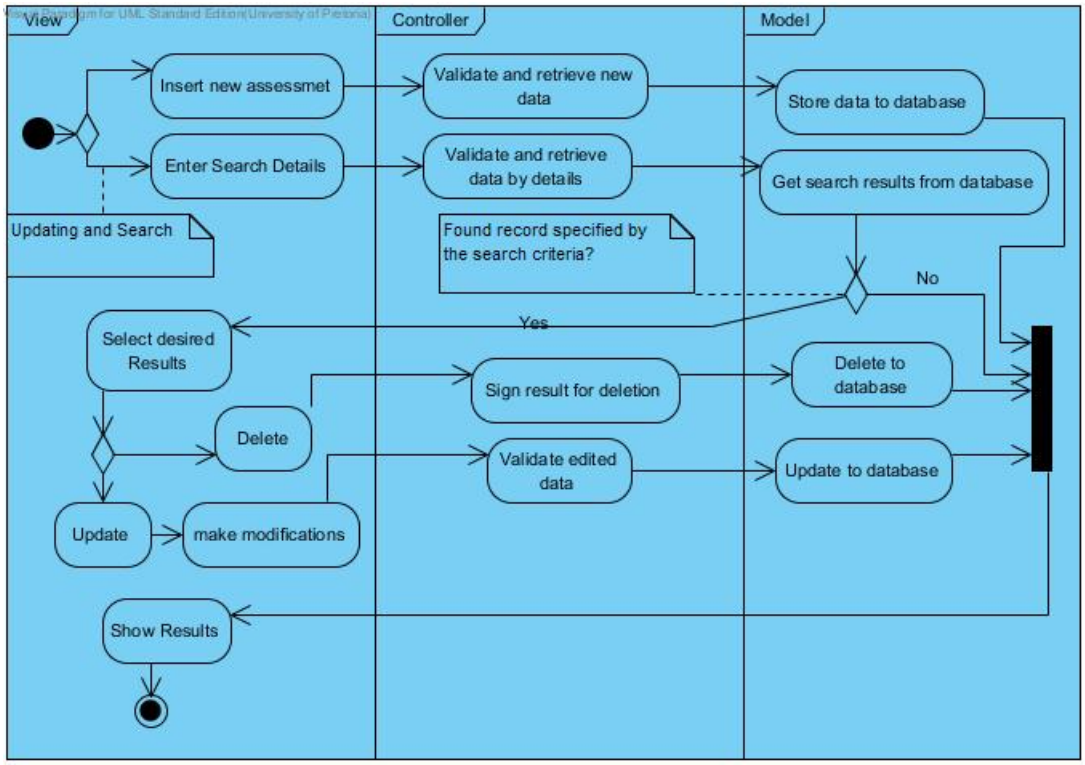
Mark processing

The Marks consists of four subsections: Adding, Editing, Publishing and showing. The students will only take part in the viewing, lectures are the only users that can publish and the editing and adding can be performed by both the marker and lecturers.


Student
Student information gathered from LDAP
Get all marks
Get specific mark
Connection to mySQL database established and information returned according to search
Published marks retrieved
Get all marks
Lecturer
Search for a specific assessment/student
Connection established to LDAP
Student information returned according to search
Search for marks for specific student
Remove marks for specific assessment/student
Connection established to mySQL database and marks removed
Get all assessments
Connection to mySQL database established and information returned according to search
Get specific assessment
Publish specific marks
Marker
Search for marks for specific student
Update existing mark for the specific student
Connection established to LDAP
Student information returned according to search
Upload of marks to database requested
Connection established to mySQL database and marks removed
Searching for student in database
Enter marks for the specific student

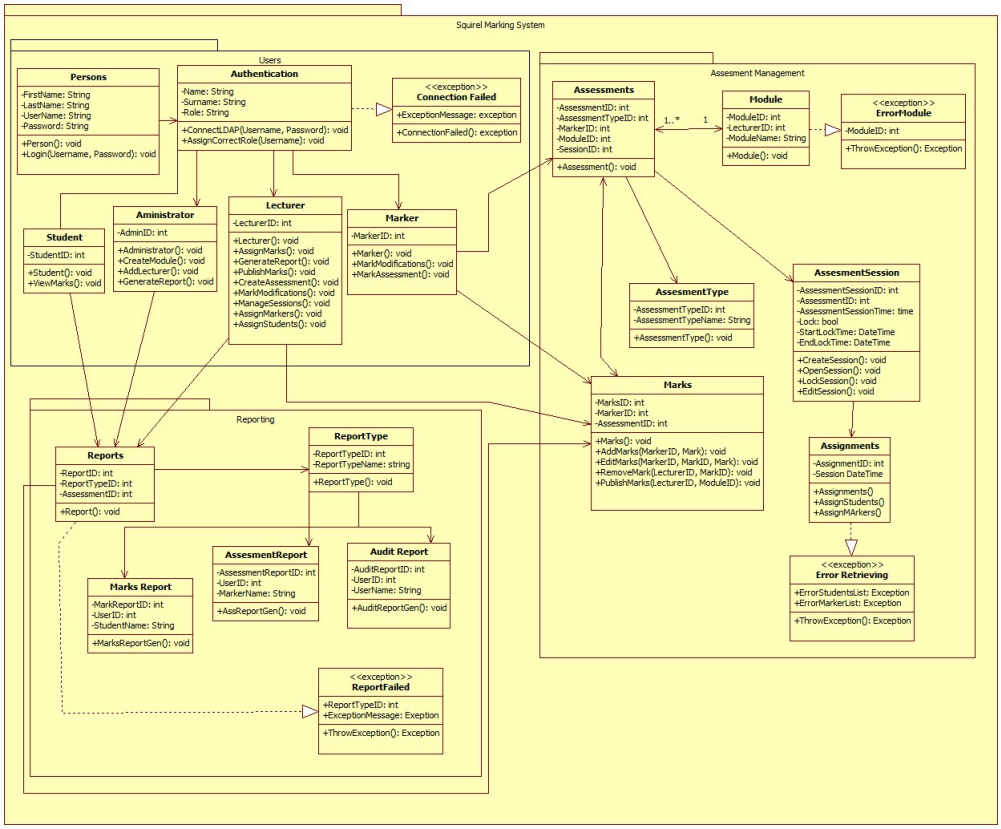
Assessment Activity Diagramg

The system will allow lecturers to create, modify and remove aggregate and leaf assessments. Leaf assessments are assessments to which an atomic mark is assigned. An assessment is by default not published implying that students cannot, by default, query their marks for an assessment.



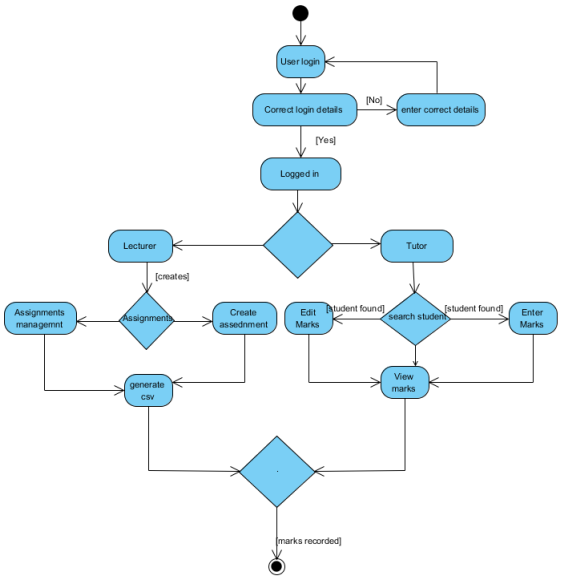
2.2 API specifcations

This demonstrates the API package of the system split into three packages. Namely Users, Reporting, Assessment management.



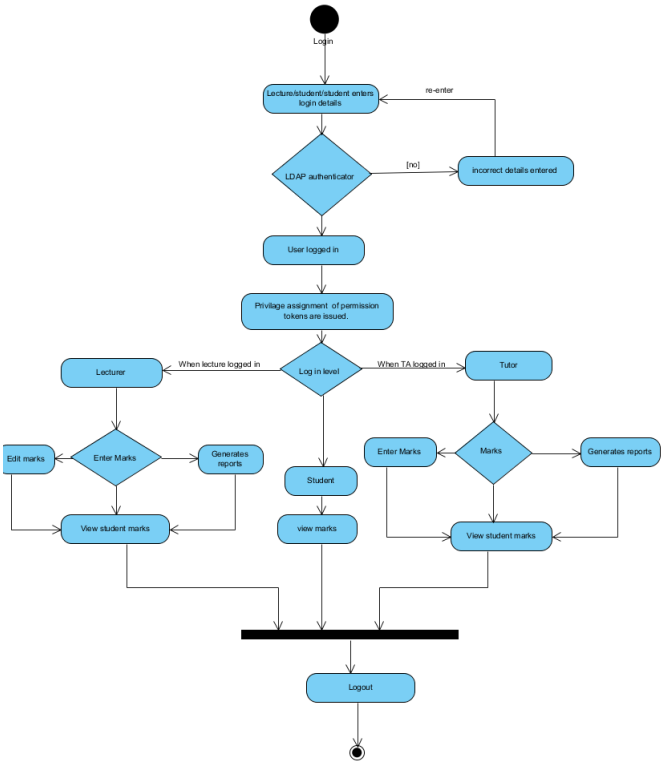
2.3 sequence and/or activity diagrams for detailed system process specifcations

App Interface

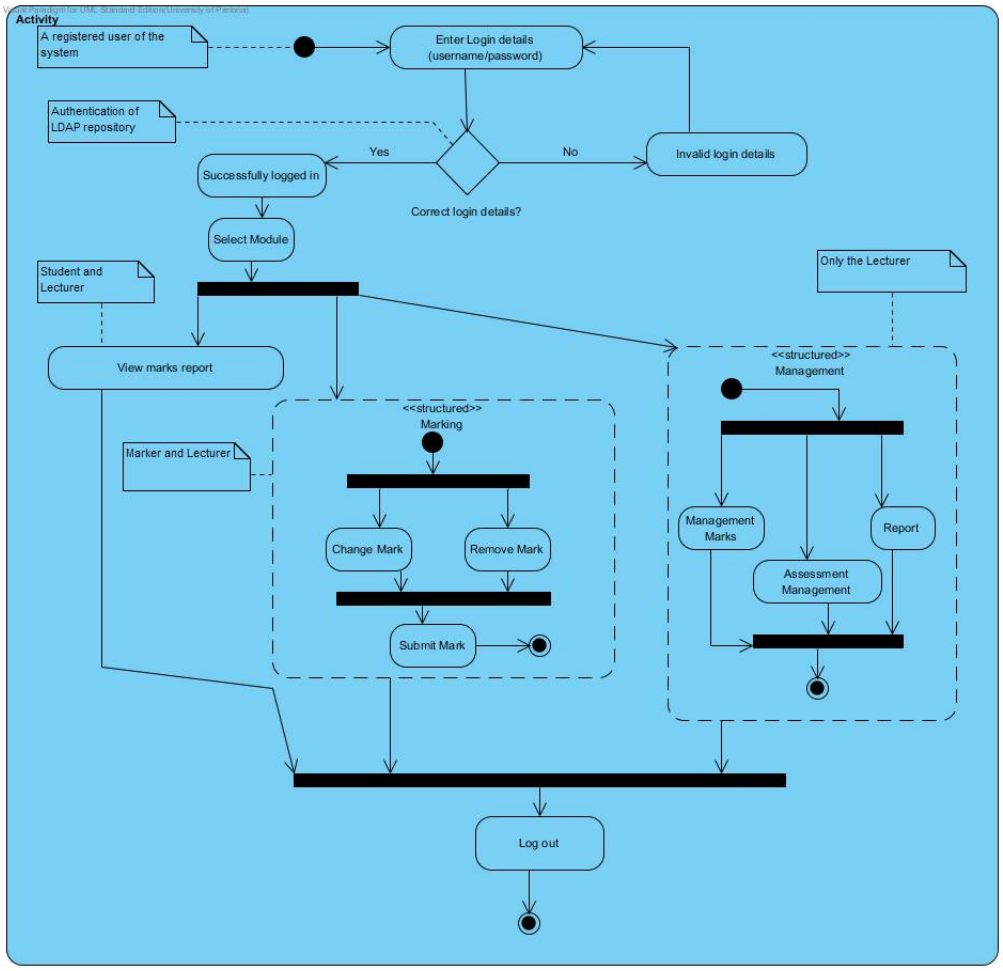


The

Web Interface



2.4 UI screen designs



3 Database Design

The database schematic in ERD(Entity relational database) format