

Design Documentation

Group 2

Group Members:

Zuhnja Riekert (12040593)

Mamelo Soepela (12094847)

Chris Moodley (12019837)

Eduan Bekker (12214834)

Christopher Crossman (10134842)

Christopher Moodley (10457489)

Version 3.0

GitHub repository

Contents

1	Software architecture design	2
1.1	Choices of Technologies	2
1.2	Chosen Frameworks	3
1.3	Chosen Protocols	3
1.4	Chosen Libraries	4
2	Application design	5
2.1	Back-end System	5
2.1.1	Database Design	5
2.1.2	Back-end Class Diagram	6
2.2	Web Application	7
2.3	Android Application	10
2.3.1	Detailed System process specification	10

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.
 - Accessibility
 - * 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 inputted 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 implementation 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

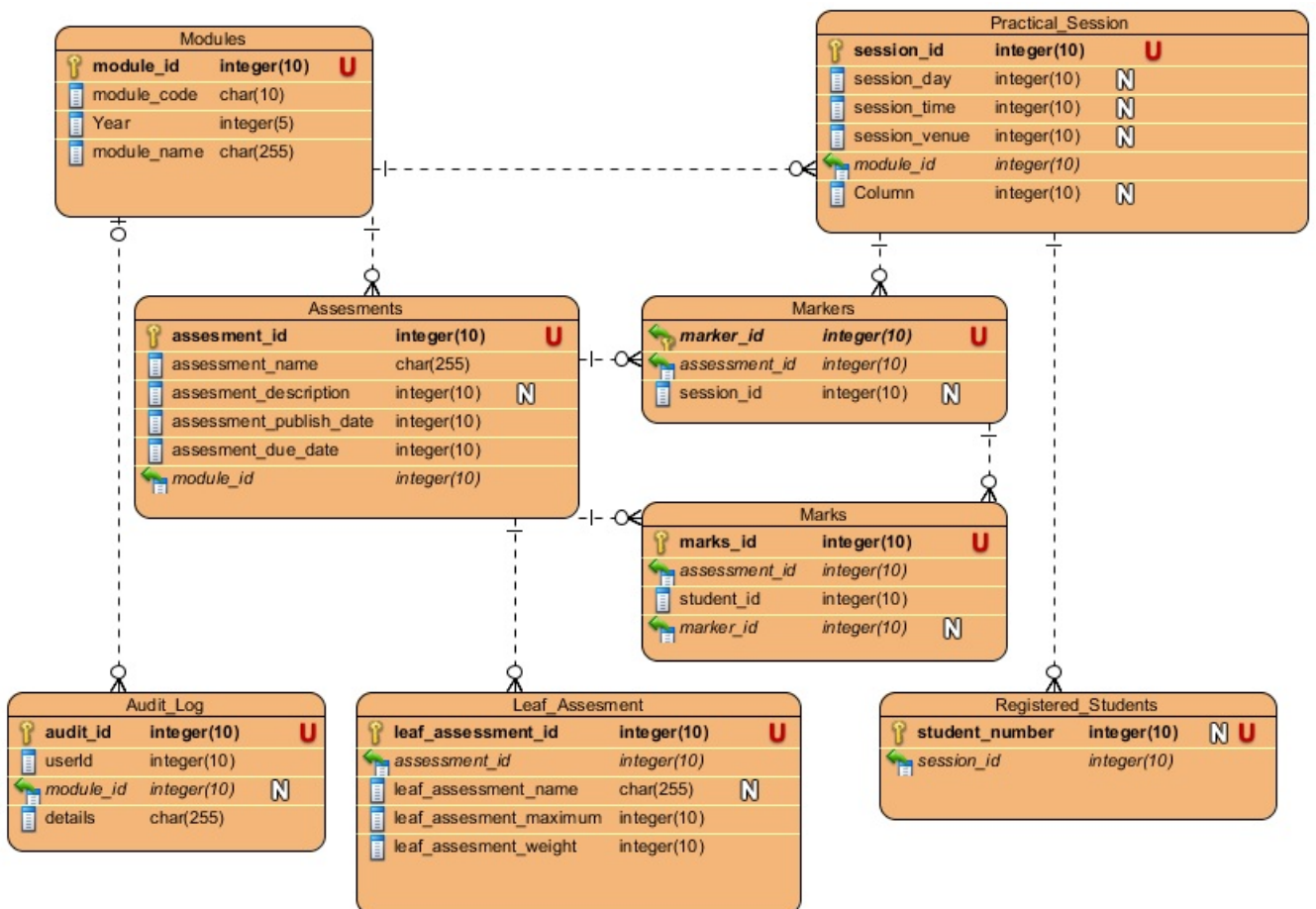
- Generating PDFs
 - iText is a PDF library that allows you to create, adapt, inspect and maintain documents in the Portable Document Format.
 - iText is supported by Java and Android applications with PDF functionality.
- LDAP Integration
 - UnboundID LDAP SDK for Java is fast, powerful, user-friendly, 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

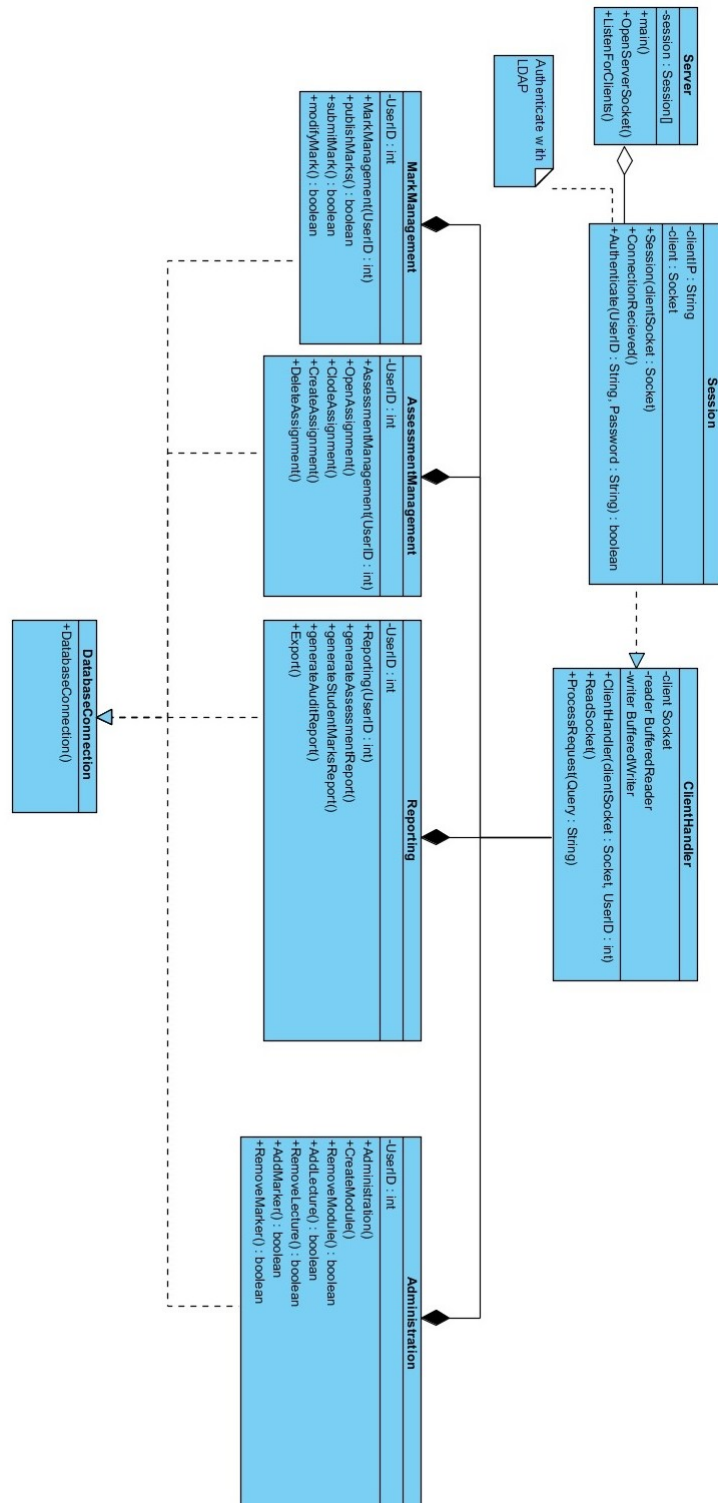
2.1 Back-end System

2.1.1 Database Design

Squirrel should be incorporated into current running systems of the university. All information on lecturers and students are already in the university's databases. No data should be duplicated. The aim of the database is only to store data that is not already stored by the University. This is mostly information on Assignments and marks. The database should also store all the audit log information.

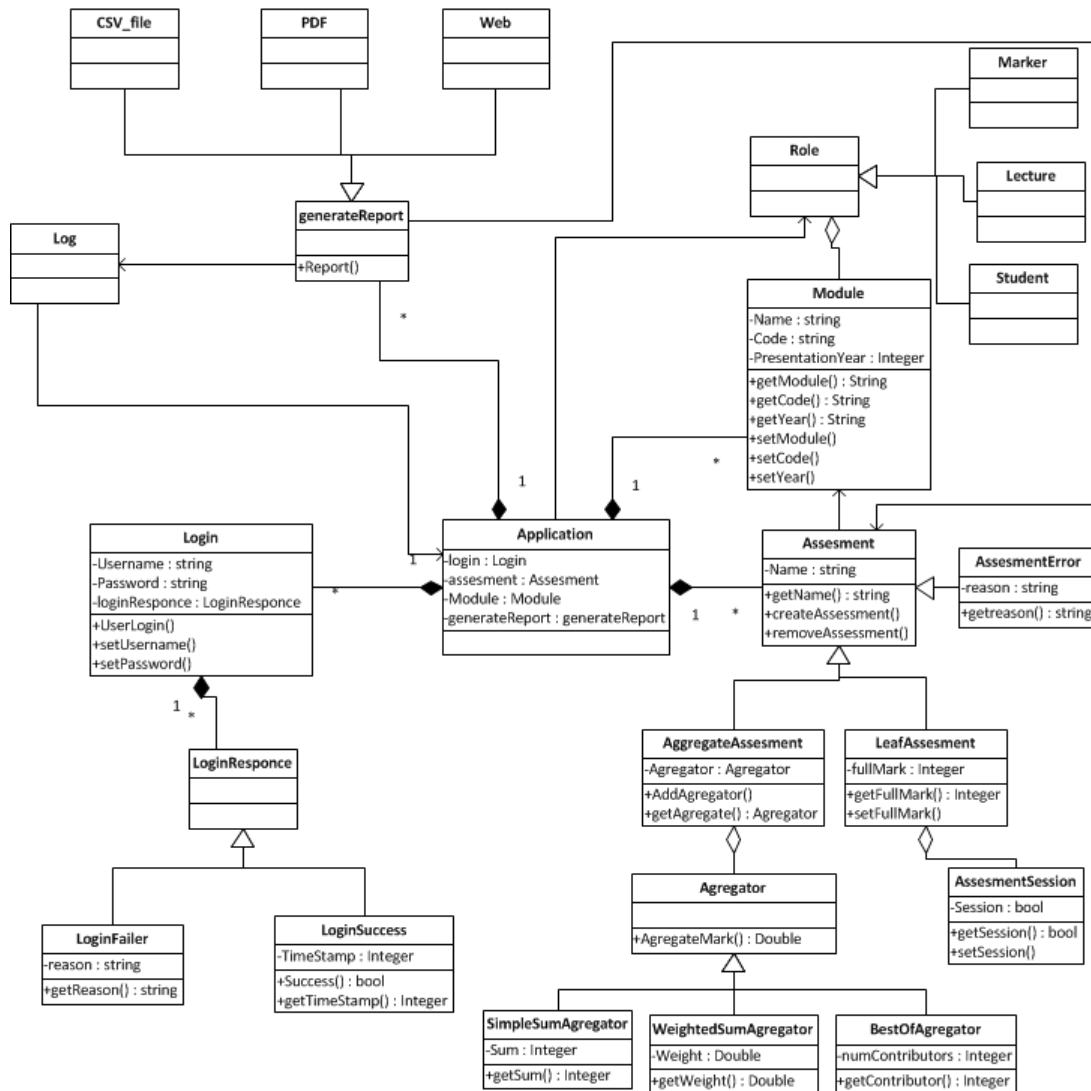


2.1.2 Back-end Class Diagram



2.2 Web Application

The web application provides a browser based interface into our application as described in this section. The class diagram below provides the implementation-independent description of the types of processes that are used in the system and passed between its components. It conveys the input and output and possible exceptions that could occur within the application.



2.2.1 The class Diagram of the web Application.

Description of the class diagram

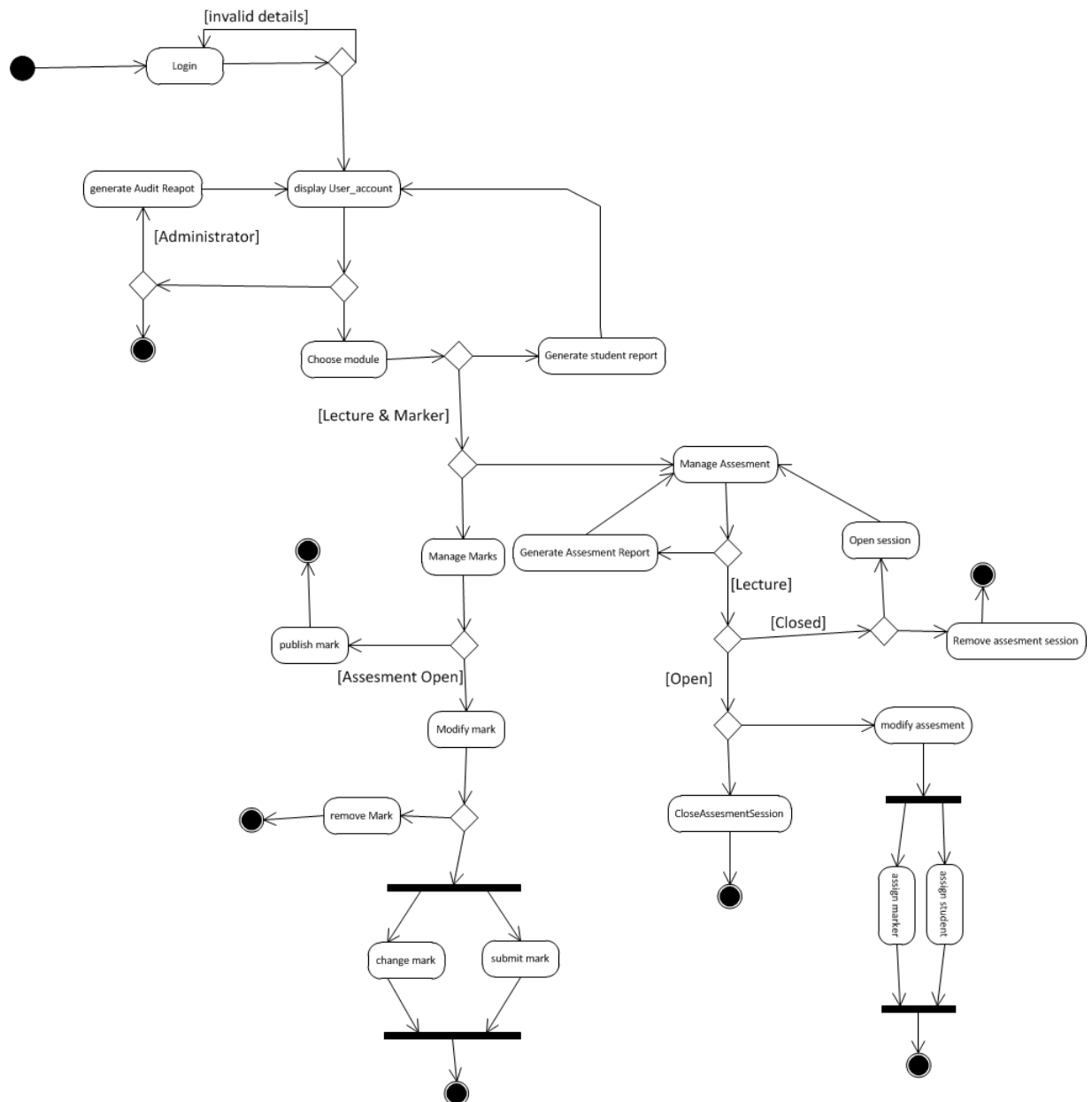
The user provides his Login details and they are authenticated with the LDAP and if the details are incorrect an exception is thrown, else the users account is loaded to the application. Only the modules and assessment that the user is registered for can be viewed from the application. Through the application if the user is a student he can view his marks by generating a report. The lecture can create and maintain assessment by allocating student and markers to the leaf assessment and he is able of opening and closing an assessment as

required. If a marker tries to modify or enter marks for an assessment session that he is not registered for, the application will throw an exception acknowledging the user of the error encountered.

The lecture can also maintain marks by allocating marks for student or editing marks and all of this actions can take place while the assessment session of the mark sheet is open. The Lecture can generate an assessment report, student mark report and the audit log report when logged in as an administrator. When generating the report assessment or student mark report, the user can aggregate the marks by weight, and or by best of aggregate. The default aggregate is the sum aggregate and the reports can be generated as pdf, a CSV file or be displayed in the web application.

The Activity Dargram

Below is the activity diagram that describes the process specification of the web application



2.2.2 The Activity diagram of the web Applicaion

The process specification

The user has to Login first in order to have access to the application, his details are authenticated and if details are invalid the user is returned to the application Login page, and an error displayed allowing the user to try inserting the correct details. If the login details are correct the home page is displayed with all the modules and assessment that the user is registered for. If the user is registered as an administrator, he can monitor and generate an audit log report.

The user can generate a student mark report and if a user is a lecture, he can manage

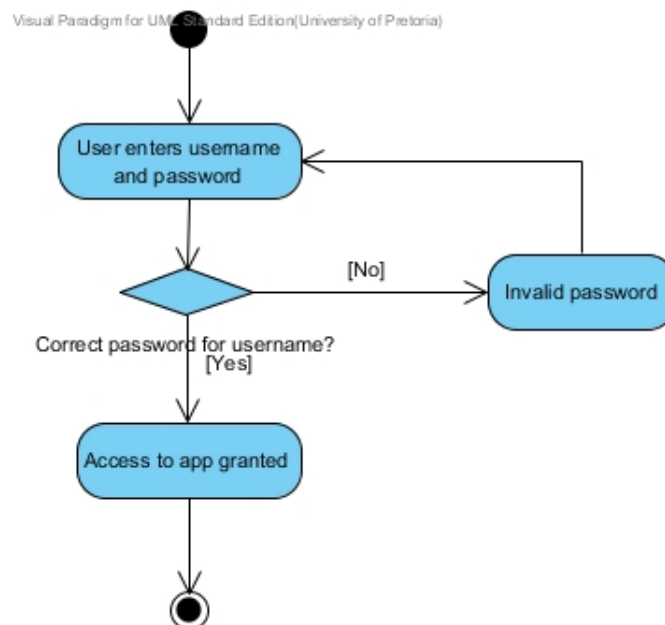
marks as well as manage assessment. When managing marks the user can remove mark sheet and or publish it and if the assessment session of the mark sheet is open the lecture or assigned marker can change and submit student marks. The lecture can also create, modify and remove assessment session. He is also able to associate an assessment session to a group of students and markers. The lecture is also responsible for opening and closing assessment sessions.

2.3 Android Application

2.3.1 Detailed System process specification

All users need to enter a username and password when first opening the application on their android device. This is necessary, for we do not want just anyone to be able to access the features and information that only specific users should have access to.

Login:



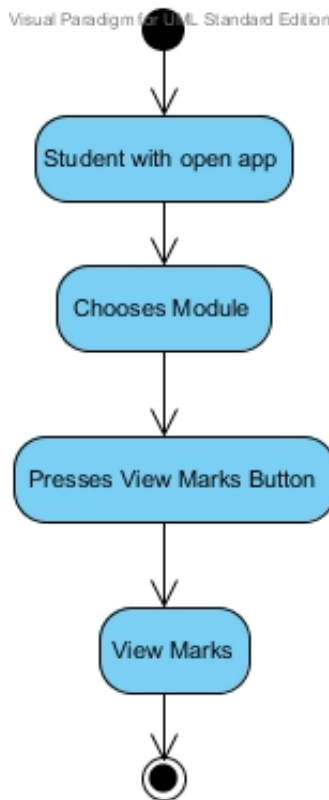
After a user has gained access to the app, the user type (student/ teaching assistant/ lecturer) is detected. The app only allows certain features to be available according to the user type's permissions.

A student only has access to their own marks, thus when they choose to view marks, only their marks for the selected module are loaded.

Student views mark:

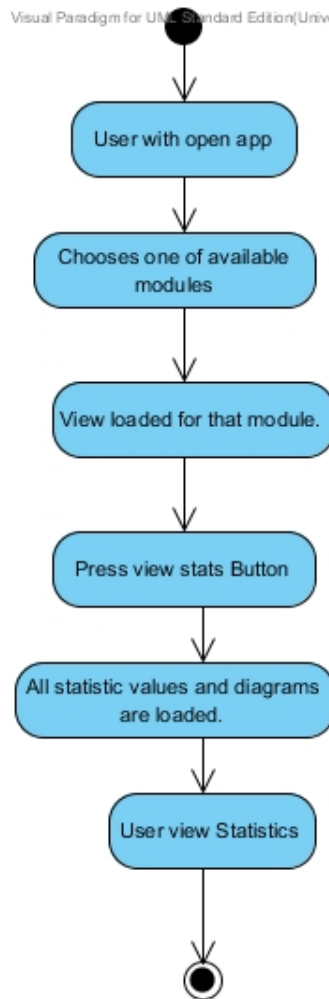
Students, like all users are however allowed to view the statistics for the selected module.

View statistics:



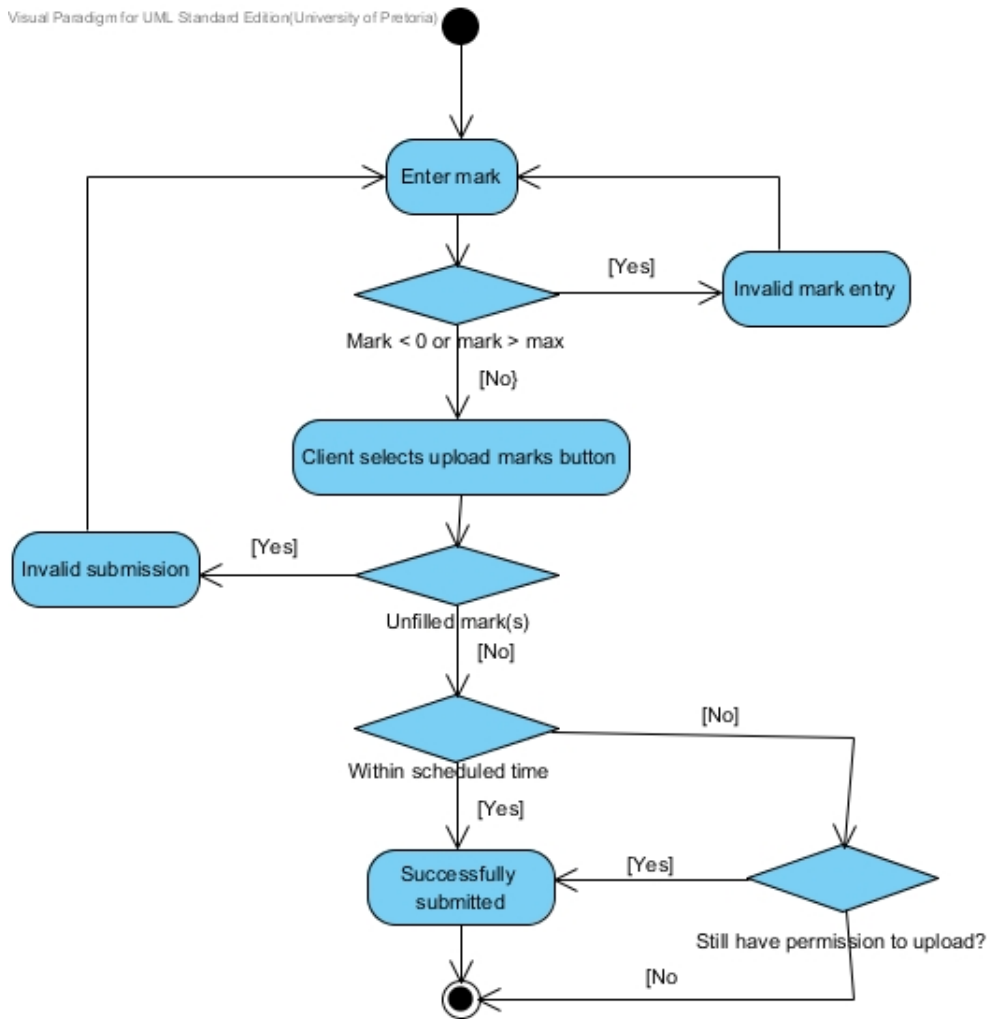
Both lecturers and TAs should have the option to add marks for a student during the required specified time limit.

Adding marks:



Only lecturers for a specific module should be able to edit marks after it has been up-loaded. Teaching will first need the permissions from the lecturer to be able to edit any marks.

Edit marks:



Only lecturers should be able to view all of the student's marks and give teaching assistants certain permissions.

Permissions and view all marks:

