

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Requirement Specification

CS MARKING SYSTEM (MINI-PROJECT)

Pieter le Roux (10454862)
Chistopher Moodley (10457489)
Collen Mphabantshi (10404687)
Bernhard Müller (11037157)
Machocho Shika (12127877)
Gerhard Smit (12282945)
Handre Watkins (10115405)

V1.0

February 27, 2014

GutHub Repository

Change log/History

Date	Author	Description	Version
20 Feb 2014	Machocho Shika	Skeleton and First draft	0.1
22 Feb 2014	Gerhard Smit	Introduction and Scope	0.1.1
24 Feb 2014	Bernhard Müller	New Template document, section documents and Background	0.2
24 Feb 2014	Bernhard Müller	Functional Scope and Limitations/Exclusions	0.2.1
24 Feb 2014	Collen Mphabantsi	Functional requirements	0.3
24 Feb 2014	Pieter Le Roux	Architecture Constraints	0.4
26 Feb 2014	Machocho Shika	Completed Process Specifications	0.5.1
26 Feb 2014	Machocho Shika	Edited Open Issues	0.5.2
26 Feb 2014	Machocho Shika	Added use-case diagram to section 5.2 (Functional Scope and Limitations/Exclusions)	0.5.3
26 Feb 2014	Handre Watkins	Integration requirements	0.6
26 Feb 2014	Christopher Moodley	Access channel requirements	0.7.1
26 Feb 2014	Christopher Moodley	Quality requirements	0.7.2
26 Feb 2014	Machocho Shika	Functional Scope and Limitations/Exclusions	0.8
27 Feb 2014	Machocho Shika	Open Issues	0.8.1
27 Feb 2014	Pieter le Roux	Service contracts diagrams	0.9
27 Feb 2014	Bernhard Müller	Latex Conversion Document Finalisation	1.0

Contents

1	Introduction	1
2	Vision	1
3	Background	1
4	Architecture requirements	2
4.1	Access channel requirements	2
4.2	Quality requirements	3
4.3	Integration requirements	4
4.4	Architecture constraints	6
5	Functional requirements	8
5.1	Introduction	8
5.2	Scope and Limitations/Exclusions	8
5.3	Required functionality	10
5.4	Use case prioritisation	23
5.5	Use case/Services contracts	24
5.6	Process specification	27
5.7	Domain Objects	30
6	Open Issues	31
7	Glossary	32

1 Introduction

This document serves to fully specify and outline the requirements of the marking-system in detail. The document also serves to give the client and developers a clear description and elaboration of the system to be implemented in its totality. Furthermore, this document formulates an agreement between the client, Mr Jan Kroeze from the Department of Computer Science at the University of Pretoria, and developers with regard to the system to be built.

2 Vision

The marking system aims to provide lecturers with the opportunity to centralize and digitalize the marking process for practical assessments carried out and evaluated. The system will also allow Teaching Assistants (TAs) and Tutors to access the system during a marking session for a module, to record the marks of students onto the system, through the use of their mobile smartphones or computers. The system will also allow the student to access his or her mark at a later stage to see how much they have acquired. The aim of this system is to remove tedious paperwork and to prevent the loss of marks.

3 Background

This project is due to the Computer Science department of the University of Pretoria currently facing an abundance of paper work and a potential loss of mark sheets and the opportunity of utilising the COS 301 course learning opportunity for the creation of an online marking system. The system is intended to improve the marking procedures currently in place within the Computer Science department and simplify the process of processing and distributing marks as they are released. The system also provides the COS 301 students with the opportunity to learn more with regards to the procedure used for creating, designing and developing projects for businesses while also providing the University with a potentially new system that may in future be used throughout the entire university.

4 Architecture requirements

4.1 Access channel requirements

Students - Viewing marks

-] Access to marks through either the web interface, browser clients, or through the application as mobile client.
-] Need to be registered for the course.

Tutors/TA - Entering/Modifying marks

-] Either through the web interface, browser clients, or through the application as mobile client.
-] Need to be assigned as a TA/Tutor on the system.

Administrators

-] Through the web interface as browser as RESTful web clients.
-] Must have authority

Database Administrators

-] Through an SQL back-end.
-] Requires credentials.

Mobile application and Web interface (Presentation Tier)

-] Will use the CS server as the Logic-Tier.
-] Will connect via Wi-Fi or Ethernet.

The CS server

-] Will use the CS Database as well as the university's student database as the Data-Tier

4.2 Quality requirements

Performance

-] The system must retrieve and update information in real time.
-] The mobile application is lightweight and will require minimal hardware resources paired to an android operating system, it will not be dependent on using a newer, more powerful device

Reliability

-] If the system goes off-line or the mobile device cannot maintain a connection after a marking sheet has been opened, marking must still be possible, all changes will be stored locally on the device with the use of time stamps and later synchronised when a connection is made.
-] The system will not be functional whilst the server is down for maintenance.

Scalability

-] The system must scale in terms of the number of clients, both web and mobile application, concurrently using the system and should cater for all students and staff.
-] Performance, reliability, audibility and usability must not be dependent on the number of users on the system.

Security

-] The systems functions will be limited to respected classes of users, with authentication by means of a username and password.
-] Without the system approving credentials, the user will not be able to access delicate functions of the system such as a marking sheet or an audit log.

Flexibility

-] The system must be accessible from either a web or mobile application interface.
-] The web interface must not be dependent on the type of browser or operating system for full functionality.

Maintainability

-] The user must be able to effortlessly upgrade their mobile application to a newer version from the application itself.
-] The efficiency of the system must be testable with simulations.

Auditability / Monitorability

-] All modifications of data on the system must be recorded in an audit log.
-] Nobody has the authority to modify the audit logs.
-] Only the relevant HOD has the authority to access the audit log if needed.

Integrability

-] Each of the layers must integrate seamlessly with each other and not be dependent on regular human attention to function.

Cost

-] The cost of the mobile application and using the web interface must be free.

Usability

-] Both the web and mobile application interfaces must be simple, straight forward and self-explanatory with on screen guidance.

4.3 Integration requirements

The Integration channel section will cover all information in regards to the program interfacing with the multiple systems, what systems will be involved in the program and how this interfacing will be done. Integration channels refer to the endpoints that are created in the system and the interfaces and interface channels to realize this program. Integration with the multiple areas of the system will play a major part in the connection of the device, website and the database aspects.

Integration channel

Integration will be facilitated between the website, mobile application and databases. The database will serve as the endpoint in most of the user cases and only be seen as a middle step in the retrieval of marks form the student and lecturer side. The hosted database will be a MySQL database (as per client specification) and will interface to the ALDOP to retrieve student, tutor and lecturer information.

All applications will have internet requirements as with the mark, mark sheet and change logs will be uploaded via the internet to a local database that stores mark information. Mobile device, internet connection will be used in this regard to connect to the DNS of the local storage server. The website will be hosted in concert with the database and PHP can be used for server side data transfer.

Both the website and mobile application is required to do real time uploads to the database if internet connection can be made and will require a constant syncing of information. The information stored on the database will be accessible from both website and mobile application to view individual marks and final marks. This data will be used by the reporting tools

to generate the graphs and statistical information. A failsafe must be included for the mobile application if internet connectivity is not available the application must continue until the end of the current mark sheet. This must allow a batch upload of the marks. The batch size must be no more than the size of the current mark sheet and has to be checked against the timestamp that is included in the version of the database at that time.

Dynamic roles will be allocated from the data interface to the ALDOP system. The marking process can only be started by lecturers and tutors. Viewing of marks will be available to students in a personal capacity, a student can only see his or her mark. A lecturer will only be able to view the marks in a group capacity.

Protocols

The protocols include all integration tools that will be used in the development of the program.

-] FTP
-] SOAP
-] HTTPS

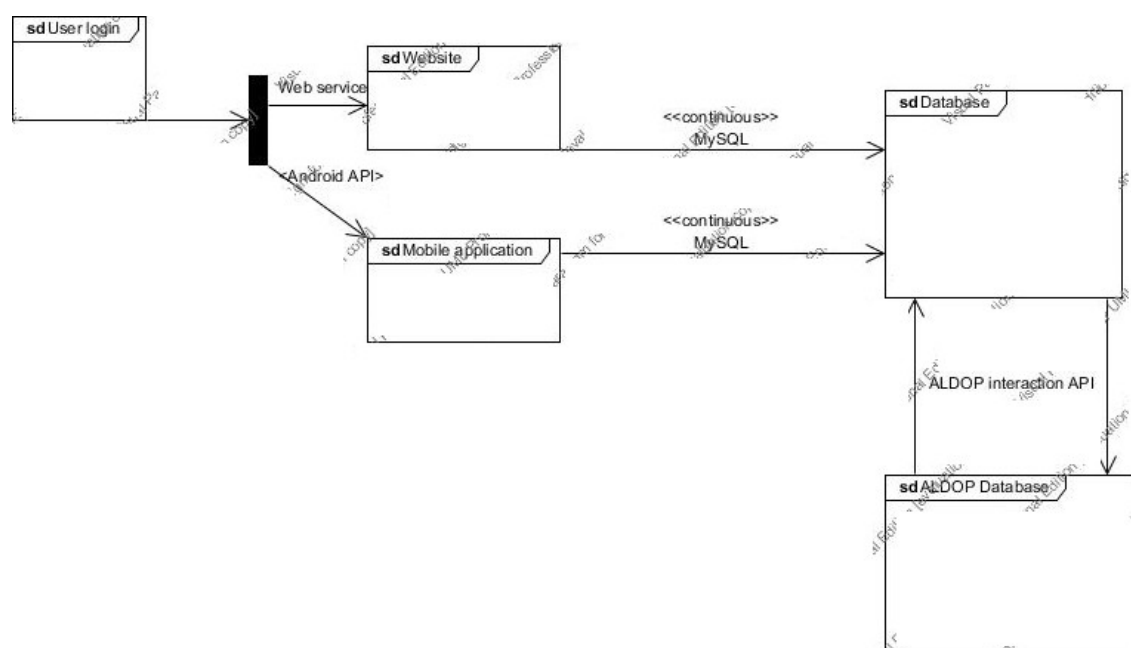
API

API (Application Programming Interface)

Android

MySQL

ALDOP



Quality requirements

Performance

The interaction with the database must be able to host all students, tutors, teaching assistance within the IT faculty. This integration must be done so that there is no decrease in performance

Security

The interface must be secure and only interaction from a registered user via the mobile application or the website will be allowed. All interactions must be logged in a way that is accessible by staff but cannot be edited.

Usability

All registered personnel at the faculty of IT must be able to use the website or application as long as the minimum requirements have been met.

The user must:

- Be a student, tutor, teaching assistant or lecturer at the Faculty of Information Technology
- Have an active internet connection via Pc or a mobile device running android (Minimum API level to be confirmed)

4.4 Architecture constraints

The constraints regarding the architecture of the system gives us a boundary or barrier outlining the system, and specifying our design within certain limitations. According to the client, constraints in the system exist both on the hardware and software of the application.

Hardware Constraints

Firstly the system needs to be able to function on any personal computer. On this type of hardware the user will access the system via a browser using the internet.

The system also needs to be able to run on any tablet available and should also be accessible through a browser, but more likely through an Android Application.

Lastly, the system will need to function on a mobile phone. Older phones might only be able to access the system from a browser, where newer smartphones should be able to access it both from a browser as well as an Android Application.

Software Constraints

The software constraints of the system are discussed a bit more and form a larger part of the boundary, so that we may better understand where and how we are limited. The technologies that must be used in this system are specified as the following:

-] On web based systems, the application needs to function on all browsers available as well as all the versions of these browsers.
-] Android is the only app based system that an app needs to be designed for.
-] MySQL will be used as database to store all the information such as marks and other student related information.
-] GitHub will be used for a web based repository to store all of the information regarding the project.
-] LDAP (Lightweight Directory Access Protocol) will be used as a user database that requires only a single sign on password.
-] Python programming language should be used to create the system.
-] Django must be used as a web framework, which is written in python, to complete the web based systems.
-] SOAP (Simple Object Access Protocol) will be used as interface for exchanging information on the web services and across the network.
-] UTF-8 must be used for encoding to keep information safe and secure.
-] Audit trails must be followed for all access made to the system.

5 Functional requirements

5.1 Introduction

The system will require the user to be a Student, Teaching Assistant, Lecturer or HOD at the Department of Computer Science at the University of Pretoria. It will allow the students to review their progress of marks obtained from practical assignments and progress marks of the semester as well. The Teaching Assistants will be allowed to mark the students during a practical and upload the marks and then to be added to the students mark list, the marks are also editable by the TA provided the mark sheet is not locked by the Lecturer. The Lecturer will be able to set up a practical and mark sheet that can be edited during the allocated time of the practical, the Lecturer will be able to review the students submissions and the marks and make changes where it is required. The HOD will be able to add and remove Lecturers from modules provided they are employed at the Department of Computer Science; he/she will have access to any module and review the content of any practical. Any changes made will be updated to the audit log.

5.2 Scope and Limitations/Exclusions

Scope

The scope of the Marking System project can be encapsulated as a solution that allows the users of the system to:

1. Record the practical assignments of the students.
2. To view the marks of the practical assignments at a later stage.
3. Keep all the marks on a centralized database.

The system shall be designed primarily for use by heads of departments, lecturers, teaching assistants, tutors, and students for the core and sole purpose of viewing, inputting and modifying student marks in a centralized environment.

Exclusions

The following exclusions are made explicit:

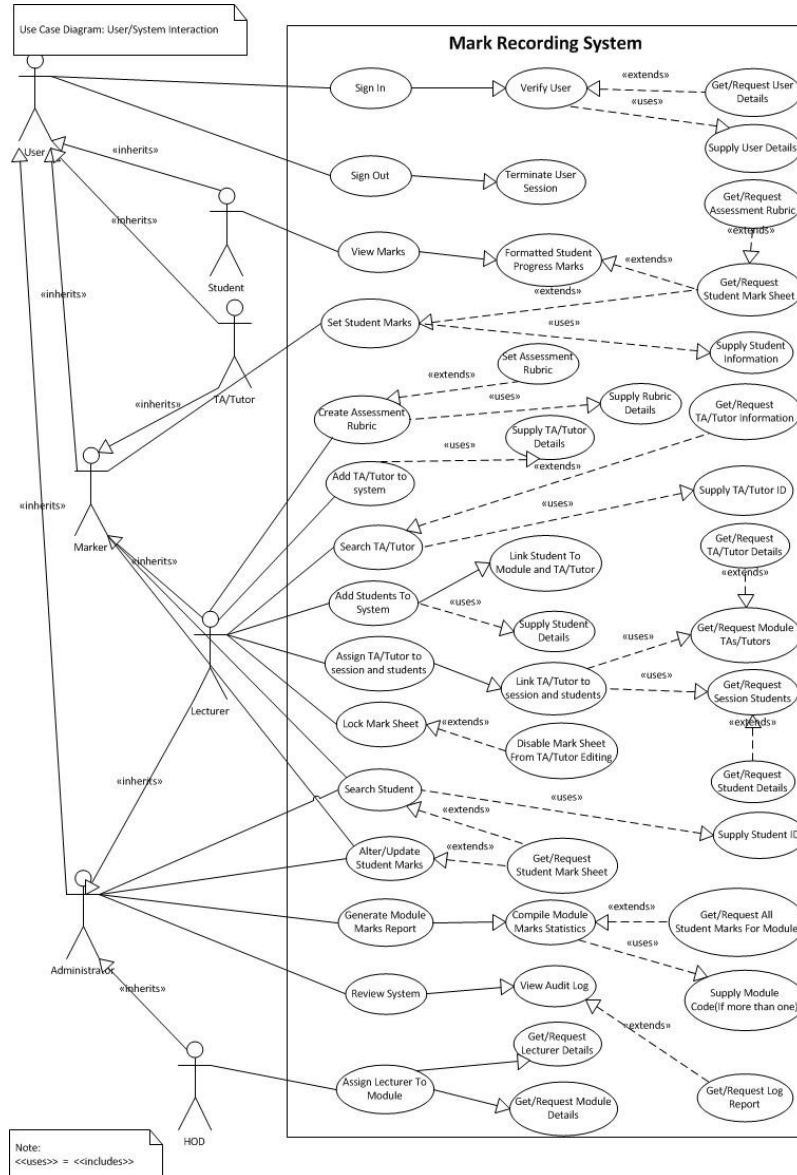
-] The system does not allow for assessments to be uploaded onto the system for auto marking.
-] The system strictly allows for mark-inputting by markers.

Limitations

The following limitations are made explicit:

- The HOD cannot add lecturers not employed by the university to the system.
- Lecturers cannot add individuals not registered or employed by the university on the system.
- The lecturer cannot add students not registered with the university to the system.
- Lecturers cannot view module reports for modules they do not lecture.

Use-Case Diagram

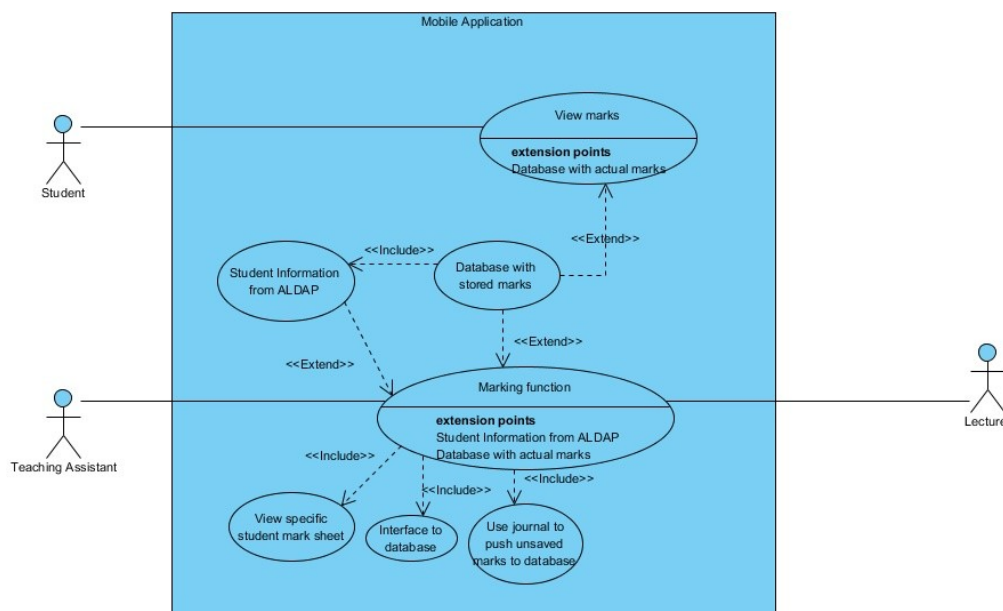


5.3 Required functionality

The following system processes describe the functional requirements of the system.

1. Downloading mobile application and installation

- a) Elaboration - Users shall be able to download the mobile version of this application from Google play store. The application is free.
- b) Importance - 4.
- c) Dependency level - Without downloading the application, users cannot make use of the applications functionality. Hence this process is a critical process.
- d) Pre-condition - User does not have mobile application.
- e) Post-condition - User has mobile application downloaded and installed on their mobile phones.
- f) Requestor - Client

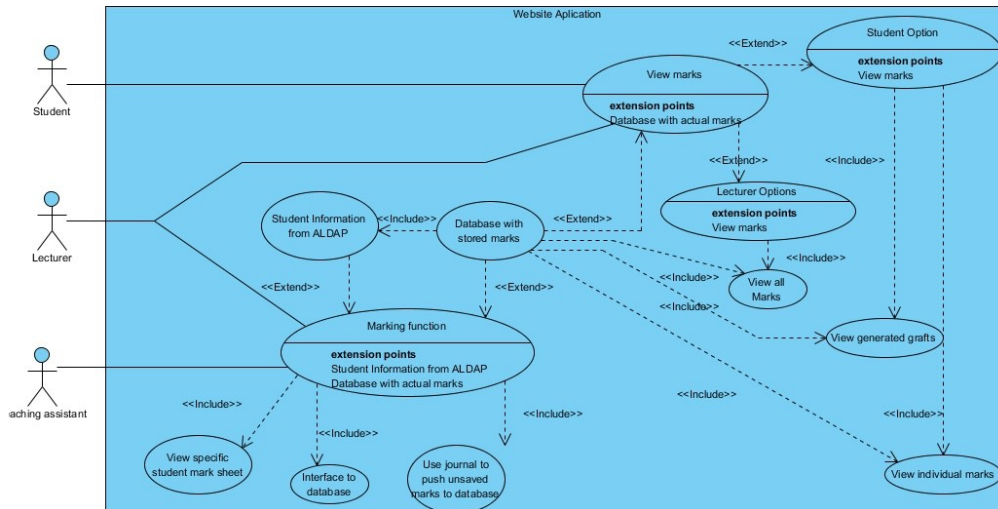


2. Online access(WEB APPLICATION)

- a) Elaboration - Users shall be able to access online the web version of MMS from the specific link provided e.g. www.mms.up.ac.za
- b) Importance - 4.
- c) Dependency level - Without having access to internet, users cannot make use of the web application functionality. Hence this process is a critical process.
- d) Pre-condition - User does not have internet access on his/her computing device.

e) Post-condition - User has internet access on their computing devices.

f) Requestor - Client



3. Signing In

a) Elaboration - Users of this system should be able to sign in in order to verify them as valid users of the system and allow them their given privileges on the system.

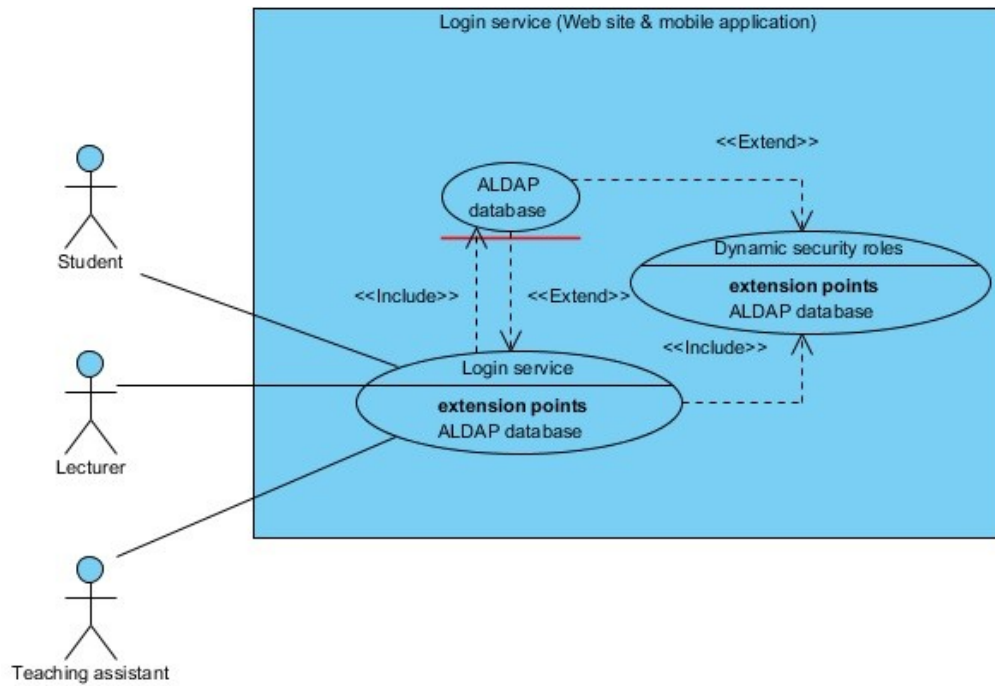
b) Importance - 4.

c) Dependency level - Without this process, users of this system will not have access to the system, which will render the system unusable.

d) Pre-condition - This feature on the system will be properly implemented using appropriate technology.

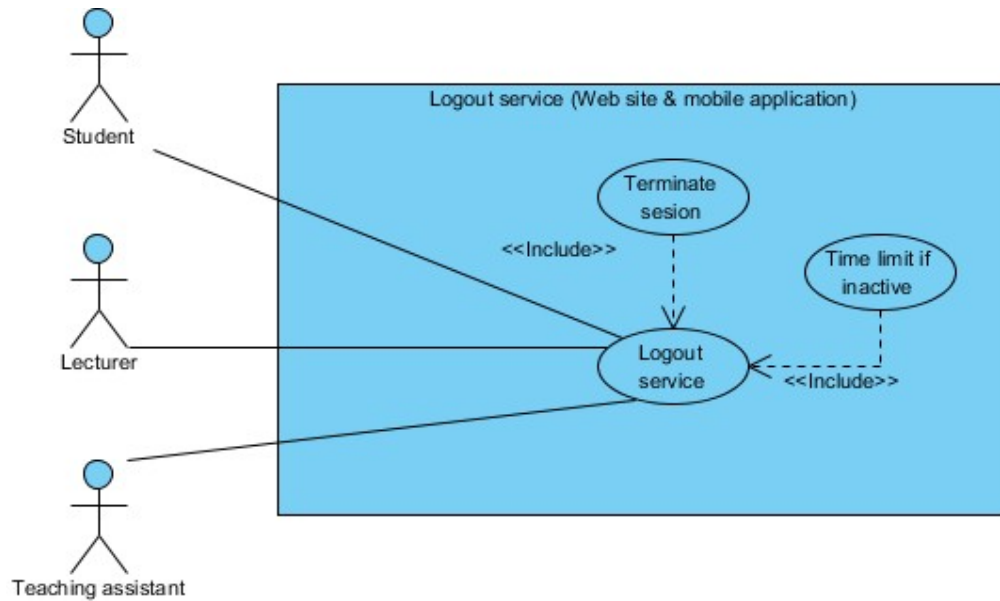
e) Post-condition - This feature functions as it should with no defects or shortcomings.

f) Requestor - Client.

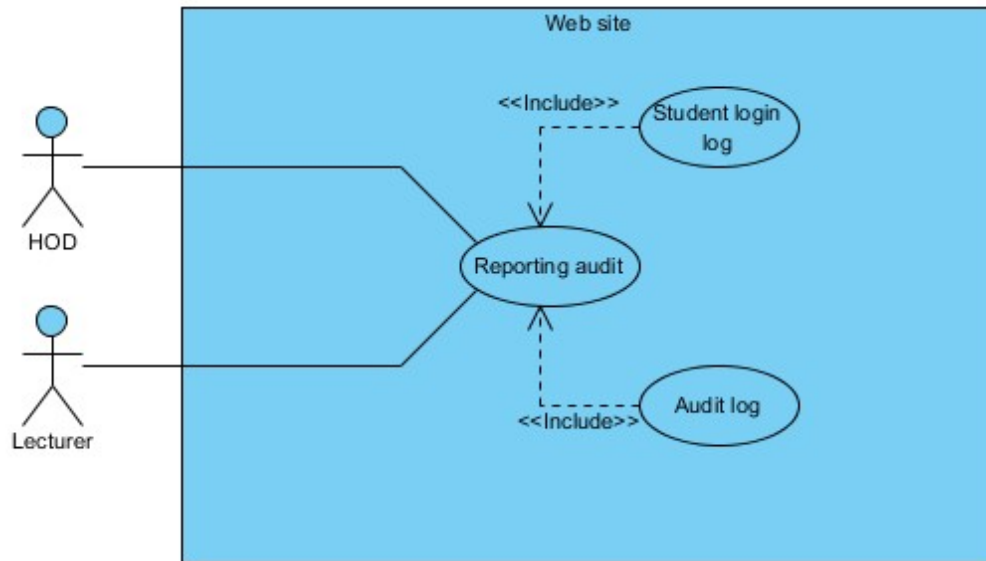


4. Signing out

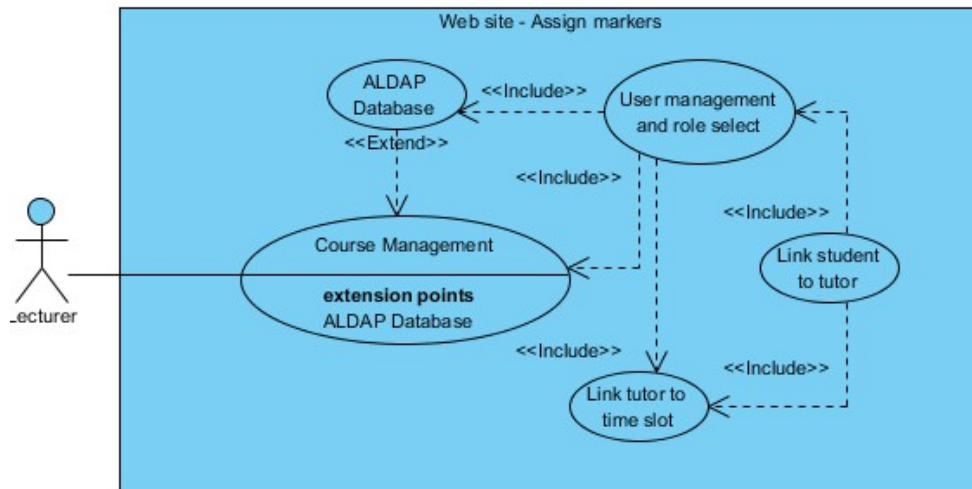
- a) Elaboration - Users of this system should be able to sign put in order to end their active usage session of the system for security reasons.
- b) Importance - 4.
- c) Dependency level - Without this process, users will not be able to sign out of their active sessions, which poses a great threat to the security of the system.
- d) Pre-condition - This feature on the system will be properly implemented using appropriate technology.
- e) Post-condition - This feature functions as it should with no defects or shortcomings.
- f) Requestor - Client.



5. Adding lecturers to the system and assigning them to relevant modules.
 - a) Elaboration - This process will allow the HOD to add lecturers in his department to the system and validate them as lecturers for their respective modules.
 - b) Importance - 4.
 - c) Dependency level - Without this feature, the entire system is unusable as lecturers cannot assign TAs and tutors to mark students using the system, hence the client request is not satisfied.
 - d) Pre-condition - This feature on the system will be properly implemented using appropriate technology.
 - e) Post-condition - This feature functions as it should with no defects or shortcomings.
 - f) Requestor - Client.

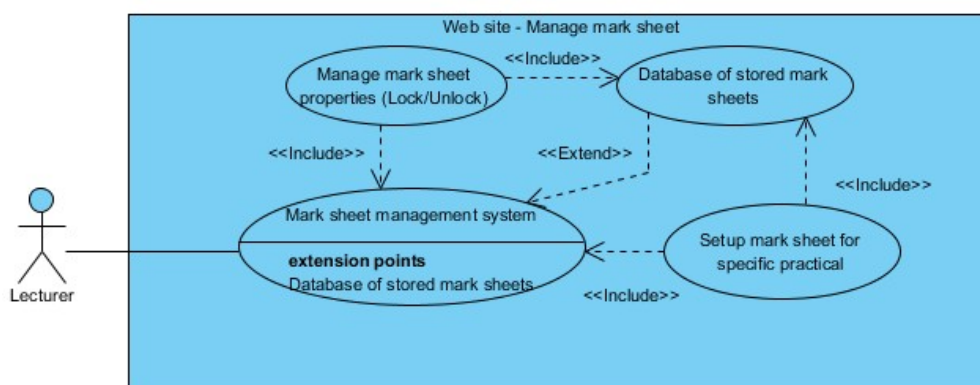


7. Assigning markers to practical sessions and assigning students to markers.
- a) Elaboration - This process allows lecturers to assign markers to marking sessions and to assign that marker a selected group of students.
 - b) Importance - 4.
 - c) Dependency level - Without this feature, the system becomes ineffective as there are no markers assigned to record student marks.
 - d) Pre-condition - This feature on the system will be properly implemented using appropriate technology.
 - e) Post-condition - This feature functions as it should with no defects or shortcomings.
 - f) Requestor - Client.



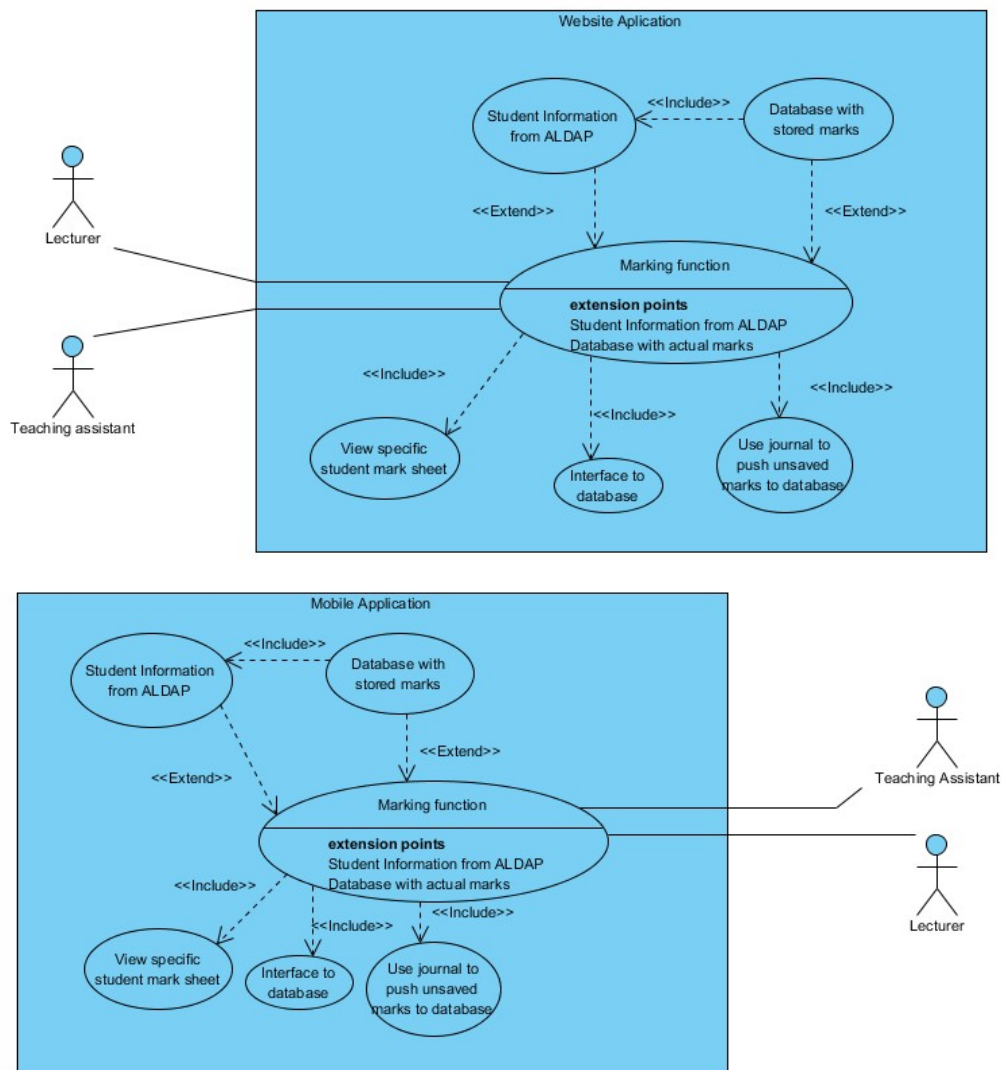
8. To lock and finalize mark sheets for marking sessions once the session is complete.

- a) Elaboration - This process involves the system automatically locking a mark-sheet after practical session is complete, or the lecturer locking the mark-sheet himself/herself.
- b) Importance - 3.
- c) Dependency level - Without this process, markers may alter mark sheets whenever they wish, even long after the assessment was assessed, which poses security and validity threats.
- d) Pre-condition - This feature on the system will be properly implemented using appropriate technology.
- e) Post-condition - This feature functions as it should with no defects or shortcomings.
- f) Requestor - Client.



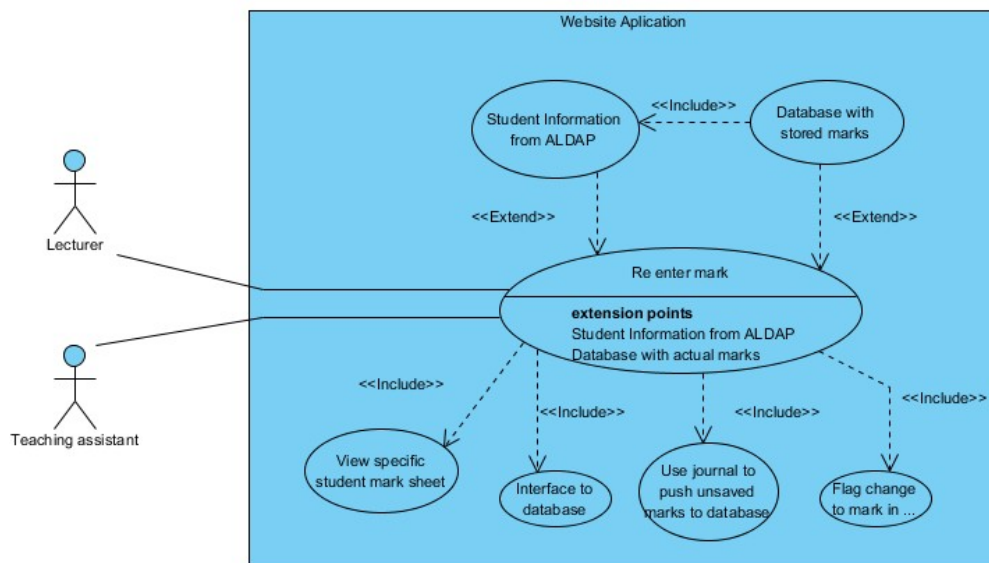
9. To record students marks onto the system

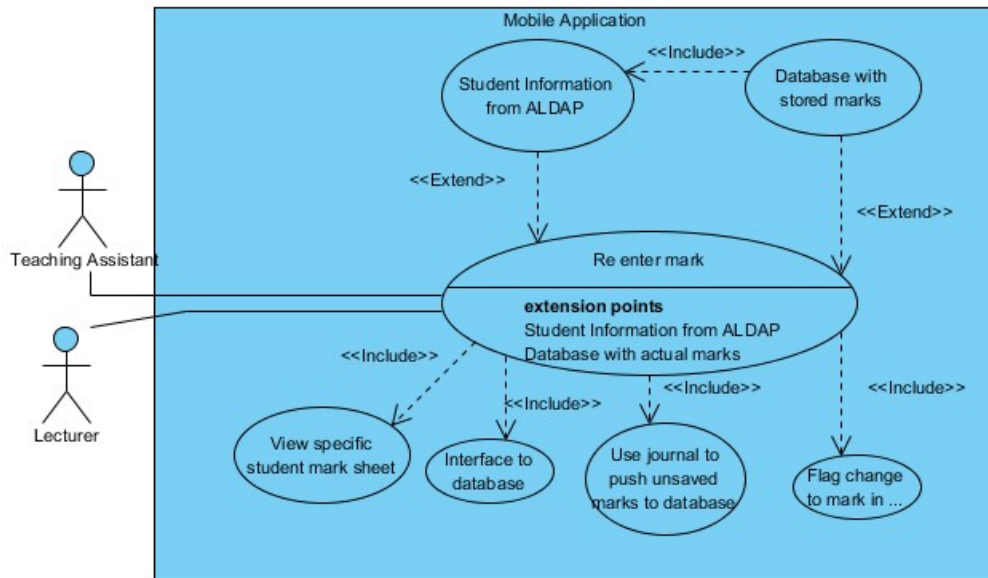
- a) Elaboration - This process involves the marker entering a students mark into the system.
- b) Importance - 4.
- c) Dependency - Without this process, the purpose of the system is defeated.
- d) Pre-condition - This feature on the system will be properly implemented using appropriate technology.
- e) Post-condition - This feature functions as it should with no defects or shortcomings.
- f) Requestor - Client.



10. To alter students marks on the system

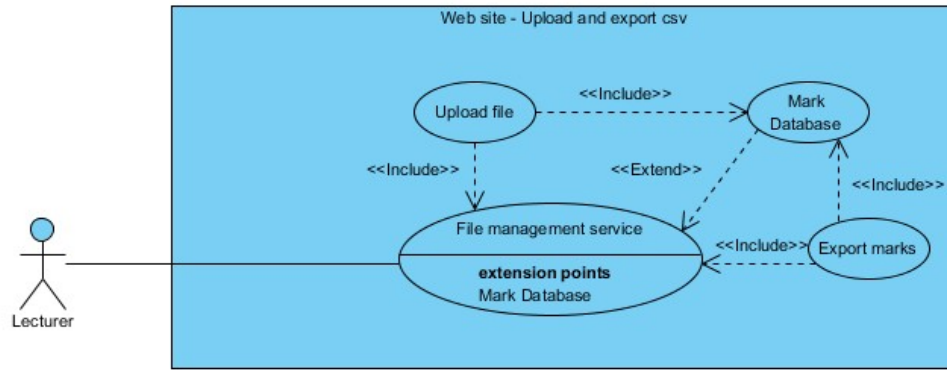
- a) Elaboration - This process describes a marker altering a student mark on the system after having already recorded a mark for the student. There shall be an audit book that will keep records of every edit that is being done on the mark sheet(s). It will only record changes that happens when the marks are being edited on the mark sheet(s).
- b) Importance - 3.
- c) Dependency level - This process allows markers to correct a students mark which has been incorrectly recorded on the system. As the client wishes that lecturers have minimal to zero interaction with the mark-recording-process, this process is rather necessary.
- d) Pre-condition - This feature on the system will be properly implemented using appropriate technology.
- e) Post-condition - This feature functions as it should with no defects or shortcomings.
- f) Requestor - Client.





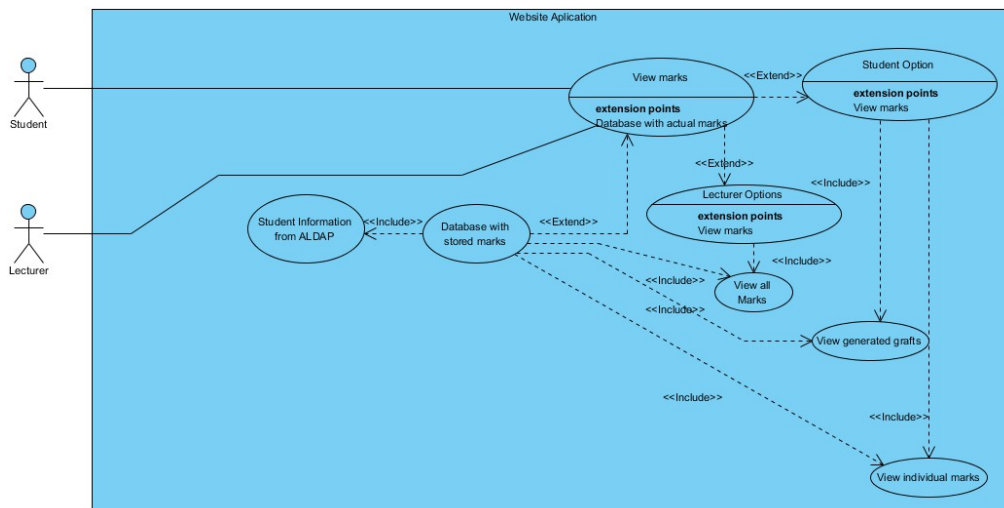
11. Import/Export .csv files

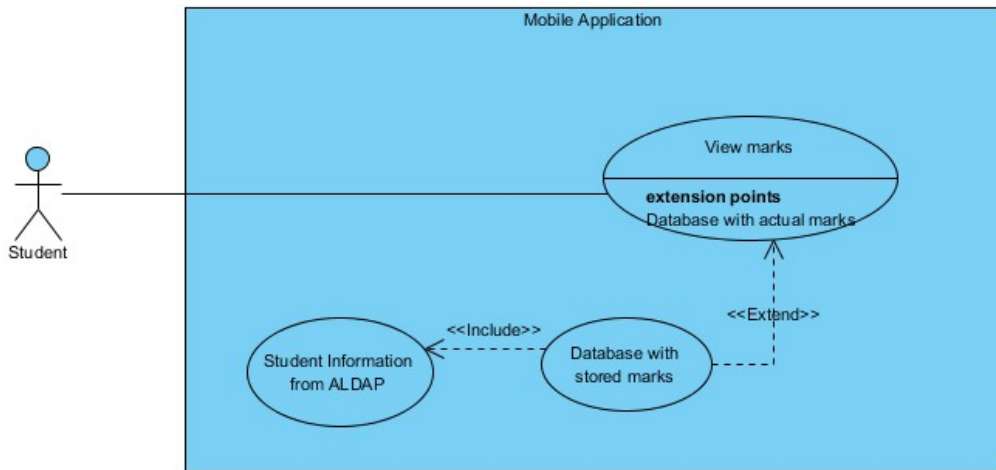
- a) Elaboration Users of this system shall be able to import or export marks to and from the system in csv format.
- b) Importance - 2
- c) Dependency level - The system is not highly dependent on this process, and can still function to its purpose without this process.
- d) Pre-condition -
 - i. User wishes to download mark sheet in csv format.
 - ii. User wishes to upload mark sheet in csv format.
- e) Post-condition -
 - i. User successfully downloaded mark sheet in csv format.
 - ii. User successfully uploaded marks in csv format.



12. To view marks on the system

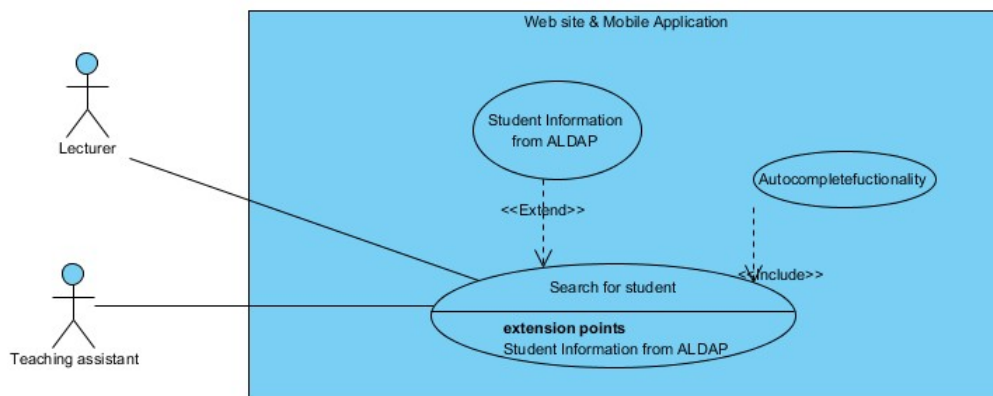
- a) **Elaboration** This process describes students, lecturers, and the HOD being able to view a students marks on the system. Student marks should be visible as individual and cumulative marks till any given point in time. These marks should also have statistical measures and graphical representations such as graphs.
- b) **Importance** 4
- c) **Dependency level** Without this process, the purpose of the system is defeated.
- d) **Pre-condition** This feature on the system will be properly implemented using appropriate technology.
- e) **Post-condition** This feature functions as it should with no defects or shortcomings.
- f) **Requestor** Client.





13. Search ability

- The system shall be able to allow users to search using key words such as name, surname and students number. The keeps the history of what has been searched for on that session.
- Importance 2
- Dependency level This feature will make the system more easy to use and simple to use.
- Pre-condition nothing has been typed on the search bar, no search is made.
- Post-condition keyword has been typed on the search bar, the system automatically filters content with relevance to the keyword.
- Requestor Client

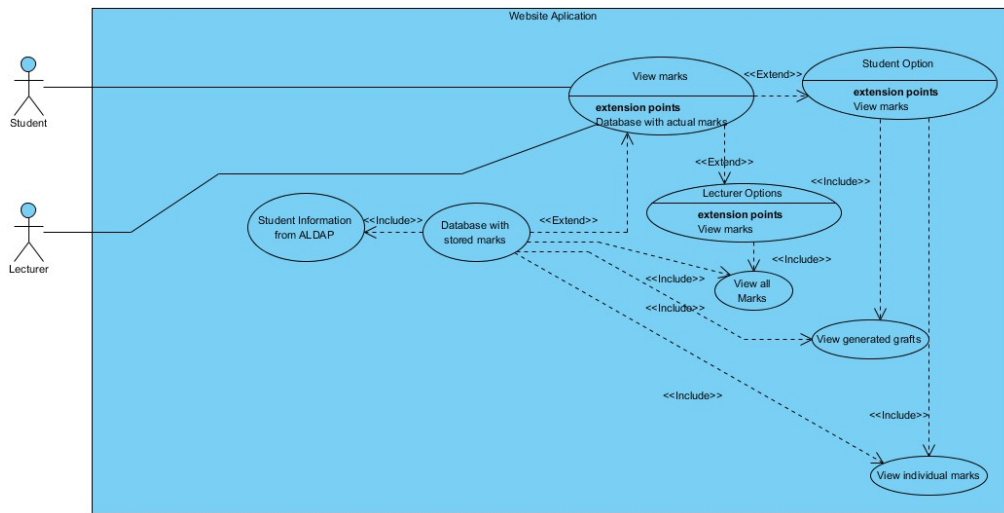


14. Generating report

- HODs and Lectures shall be able to generate report on their modules at any level of precedence. The report is in pdf format and it contains graphs and other

necessary records about the module.

- b) Importance 3
- c) Dependency level this feature is very useful since it adds value to the system of producing clear indication of what is happening on modules to lectures and HODs.
- d) Pre-condition no request has been made on any level of precedence.
- e) Post-condition request has been made of the report at some level of precedence, it will be generated in pdf format.
- f) Requestor Client



5.4 Use case prioritisation

Critical

-] System runs on all web browser platforms. Application runs on android.
-] All marks are stored on a centralized database; no data is stored on any other nodes.
-] Audit log cannot be edited at all.
-] SOAP interface.

Important

-] TAs can edit and add marks during practical sessions.
-] Lectures can set up a lock for mark sheets.
-] Students can only view their marks after the practical.
-] Should be an audit log for every change and action done.
-] Students can only be marked by the TA assigned to them.
-] If signal is lost while trying to update marks, should be saved by the browser temporarily till signal is restored.
-] User must be logged in to access.
-] Daily batch to the audit log to record what was changed.
-] Progression and individual practical session marks.
-] Reason for editing marks.
-] Report system, display of graphs.
-] Fitchfork marks exported.
-] Automatically decide the best set of marks.
-] Search for individual students.
-] Journal of audit log.

Nice to have

-] Kick user out for inactivity.
-] When marks are edited after practical lectures are emailed about changes.
-] Specify weight of marks.
-] Auto complete student number.

5.5 Use case/Services contracts

Pre-conditions:

A few conditions must be met before any of the users can access and use the system.

-] The user must have a functioning device capable of running the system. Devices may be a personal computer, a tablet or a mobile phone.
-] The user must work on a software platform capable of running the system. That would be all the web browsers or Android operating system to run the application.
-] The user must have network or internet access.
-] Students may only view their own marks and information.
-] If the user of the system is a student, the student must be registered for the course and have his/ her own user name and password to view their marks.
-] If the user is a tutor, they have to be granted special permission to access and modify the information on the system.
-] Tutors may only change and add marks for the practical session they are booked to mark.
-] Tutors need to receive permission from the lecturers or Head of Department to change marks.
-] The lecturers also need a username and password granting special permission to modify and view all of the information.
-] The Head of Department will also need a username and password that has permission to alter the information all across the system.

Post-conditions:

Post-conditions are the conditions that need to hold after access has been granted to the system to keep on using the system.

-] The user needs to have fixed access to the network or internet to keep on using the system.
-] The user needs to have permission for that specific user type to use user-specific functions in the system.
-] The user needs to be automatically signed out after a set period of time when the user is inactive.

Request and Results Data Structures

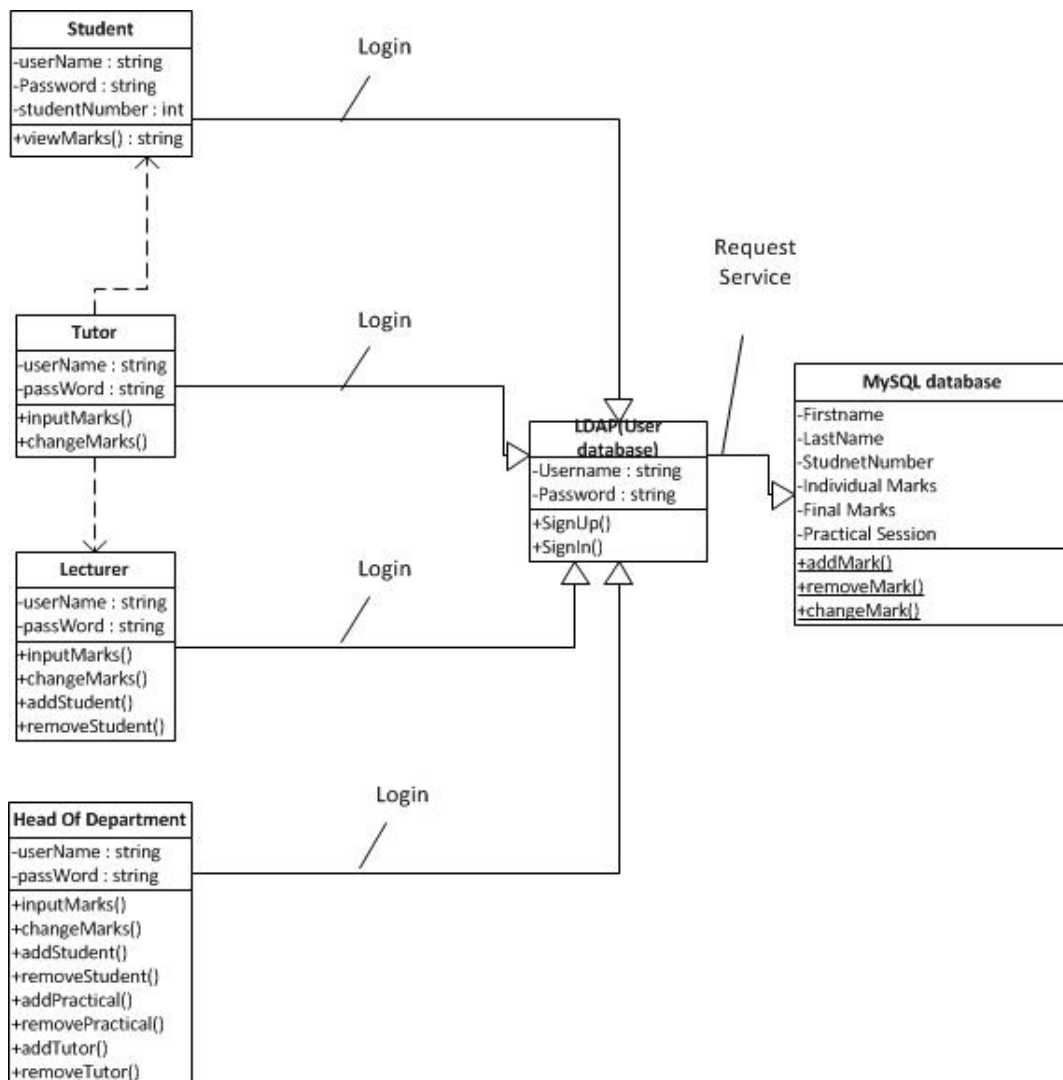
The data structures that are being used in the system are closely linked and interleaved to form the structure of the system and give it the most core of functionality. The system will need to receive some form of input from the users to process and generate the expected output.

The requests of the system will be the basic input that will need to be provided by the users, whether it is the students, tutors or lecturers. A few basic and fundamental inputs will be:

-] Students will have to provide their username and password to LDAP to gain access to the system.
-] Also students will have to choose which mark to view and in what format it should be viewed.
-] Tutors will also have to provide their specially assigned username and password to LDAP to access the system.
-] The tutors will need to provide the current students student number that is being marked, along with the allocated mark for the determined practical.
-] The tutors need to provide a reason for changing the mark of a student along with the changed mark.
-] The lecturers will once again need to provide their username and password to LDAP to gain access to the system.
-] The lecturers will have to provide marks when adding and editing them along with the students student number and what individual mark needs to be altered.
-] The lecturer will also need a newly added students information to add new students as well as removing them from the system.
-] The head of department will once again need to provide their username and password to LDAP to gain access to the system.
-] The head of department will have to provide marks when adding and editing them along with the students student number and what individual mark needs to be altered.
-] The head of department will also need a newly added students information to add new students as well as removing them from the system.
-] The head of department needs to provide a time and location to create a practical and add it to the system.
-] The head of department will have to provide details when he/she wants to add a new feature to the system.
-] The head of department will have to provide the tutor details that needs to be added or removed from the system.

The result of the system are much simpler as it produces more or less what the user wants to achieve from the system. The results of the system will be:

-] The student will be able to view his/her marks individually as well as calculated together.
-] The tutor will be able to view the marks of the student, and also know if the alteration and adding of marks was successful.
-] The lecturer will be able to view all of the database information of the students and an audit trail of any alteration of marks on the system.
-] The head of department will be able to view all information regarding the marks and students as well as an audit trail for alteration of marks.
-] The head of department will also receive reports regarding the system in a timely manner.



5.6 Process specification

Below are the specific process requirements needed for some of the use-cases:

Signing In

Summary:

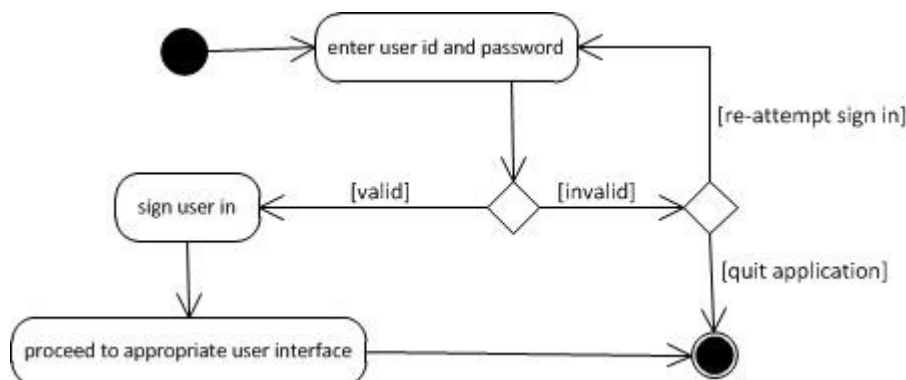
On signing in, users will present their user id and password and the system will validate them before signing them into the system and awarding them their privileges on the system.

Requirement needed for process:

User id and password should be valid.

Basic Sequence:

1. User enters user id and password
2. System validates user by checking user sign in details against the database.
3. If the user is valid, sign the user into the system and display the appropriate user interface. If the user is invalid, reject sign in attempt and request user to re-attempt sign in or exit the application



Adding Lecturer to the system

Summary:

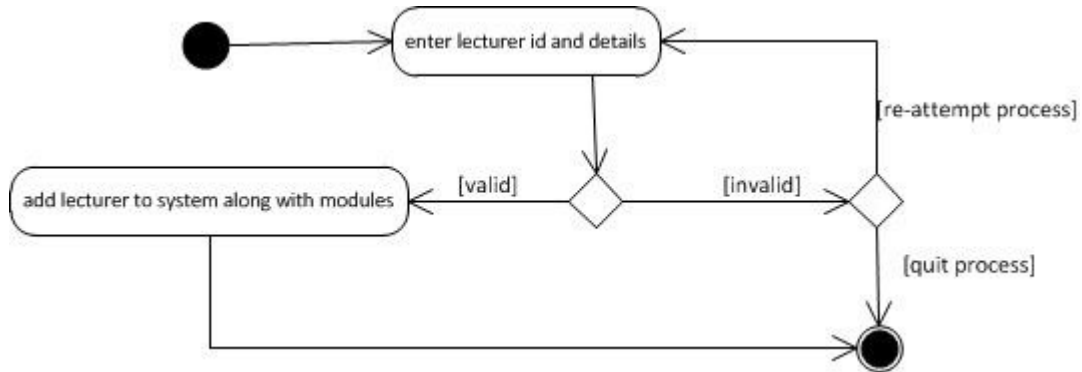
The HOD shall be able to add a lecturer to the system, however this lecturer has to be a valid lecturer at the university.

Requirement needed for process:

Lecturer should be valid lecture at the university.

Basic Sequence:

1. HOD will input lecturer id and details.
2. System will confirm lecturer against university database.
3. If lecturer is valid on university database, lecturer is added to the system along with his/her module. If the lecturer is invalid, the HOD can either re-attempt or quit the process.



Adding TA/Tutor to the system

Summary:

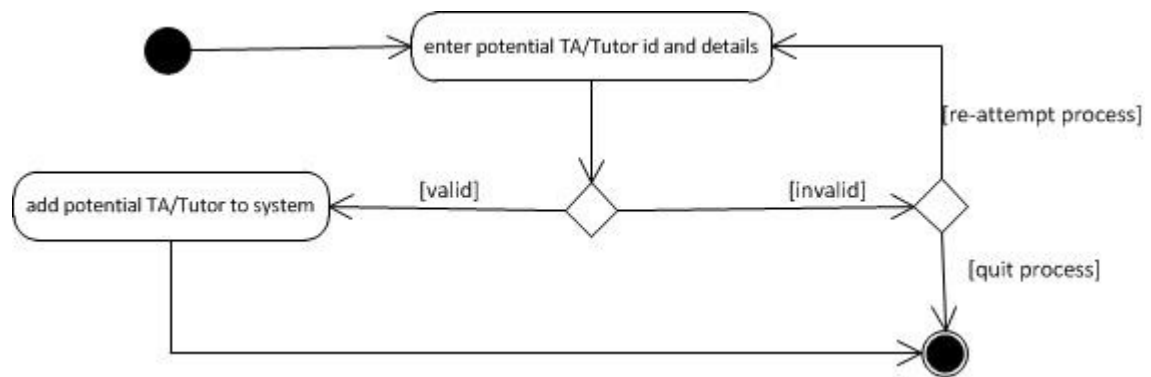
Lecturers shall be able to add TAs and Tutors to the system. However these TAs and Tutors need to be either valid students or lecturers (or employees) at the university

Requirement needed for process:

TA/Tutor is a valid student or lecturer (or employee) at the university.

Basic Sequence:

1. Lecturer attempts to add potential TA/Tutor to the system.
2. System validates potential TA/Tutor against university student and employee database.
3. If potential TA/Tutor exists on the university database, TA/Tutor is added to the system. If TA/Tutor does not exist on the database, the Lecturer may either re-attempt or quit the process.



Locking and finalizing mark-sheet

Summary:

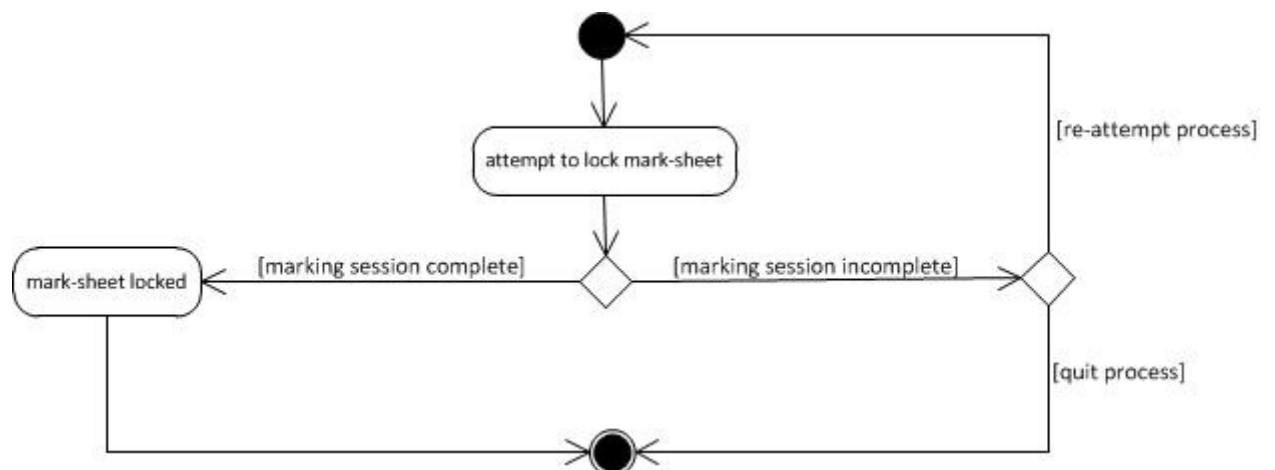
Lecturers shall be able to lock and finalize mark sheets. However, this process is dependent on the actual marking session being completed. Hence the lecturer should not be able to lock a mark sheet while a marking session is in progress.

Requirement needed for process:

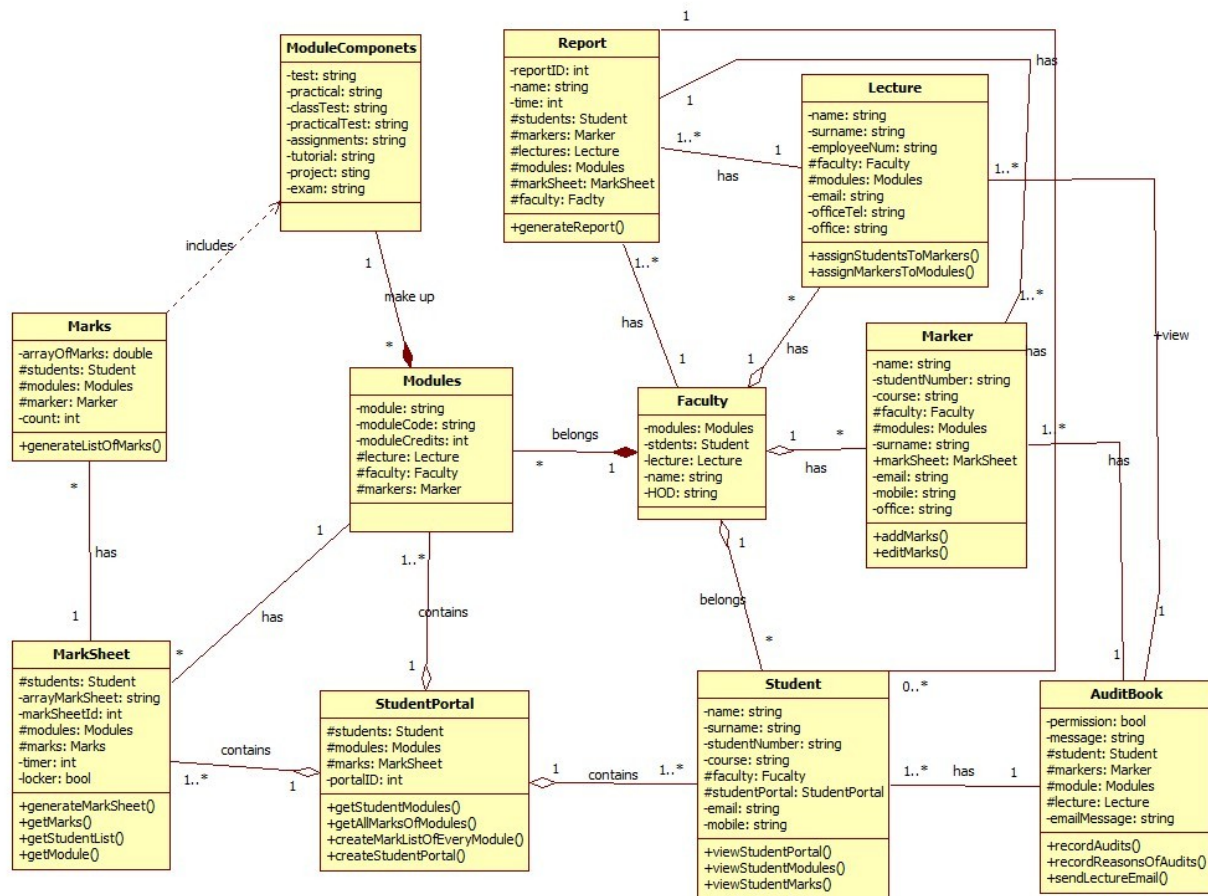
Marking session should be complete.

Basic Sequence:

1. Lecturer attempts to lock mark-sheet.
2. If marking session is complete, mark-sheet is locked. If marking session is not complete, mark-sheet lock attempt fails, and lecturer may re-attempt until marking session is complete.



5.7 Domain Objects



6 Open Issues

Requirements still to be specified

No further requirements need be specified.

Clarification Required

The following points still need to be clarified with the client:

-] Regarding the pooled-notifications for students, should a batch be sent periodically to students or should students simply check their marks on the system when they need and want to?
-] Who can upload .csv files to the system? Only the lecturer or all markers?

Requirements inconsistencies

The following inconsistencies were identified in the requirements:

-] TAs and Tutors can alter student marks by providing reasons when they do so. The client has indicated that Lecturers can decide on the privileges assigned to TAs and Tutors. However, TAs and Tutors only have two core roles which define their purpose on the system; recording student marks and altering/updating student marks (with reason). What privileges can the Lecturer then decide on, as removing any of these roles from a TA or Tutor possibly defeats the role of a TA or Tutor on the system, which in turn possibly defeats the client's purpose for the system, which is to have the TAs and Tutors deal with as much of the mark recording as possible.

7 Glossary

TA - Teaching Assistant.

HOD - Head of Department.

COS - Computer Science.

COS301 - Software Engineering module.

Auto Marking - Marking is done by a third party application, without any human insight.

Markers - Lecturers, Tutors and Teaching Assistants.

SOAP - Simple Object Access Protocol.

Audit log - A log keeping track of all updates and modifications to marks and the system itself.

Android - Operating System used on smartphones designed by Google.

Fitchfork - automated marking system to mark a practical assignment.

Locking of a mark sheet - prevents any marks to be edited or added on a practical.