Software Requirements Specification - Marking System

Change Log

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Who | What | Version |
|  |  | Skeleton | 0.1 |
|  |  | Introduction and Scope | 0.2 |
|  |  | System Requirements | 0.3 |
|  |  | Sequence Diagrams | 0.4 |
|  |  |  | 0.5 |
|  |  |  |  |
|  |  |  |  |

To Do:

Use Case Diagram / Sequence Diagram

Improve them maybe compile them into a single diagram

Class Diagram

Basic classes for what the system needs. Basic variables,

only what we know it would need nothing more.

Open Issues

Make a list of what could be problematic. For example the security issues with Mobile App

Glossary

Any and all abbreviations used for the entire thing

Requirement Matrix (Bernhard)

Just all the requirements put into separated tables for easy viewing. +-5 word descriptions of each requirement.

HERE ARE SOME THINGS THAT I BELIEVE NEED DOING. PLEASE PUT YOUR NAME NEXT TO WHAT YOU WISH TO BE INCHARGE OF. SOME OF THEM ARE SMALLER AND DON’T NEED MUCH TIME AT ALL.

Table of Contents

[Introduction 3](#_Toc380681031)

[Purpose 3](#_Toc380681032)

[Vision 3](#_Toc380681033)

[Scope 3](#_Toc380681034)

[General Description 3](#_Toc380681035)

[User Problem Statement 3](#_Toc380681036)

[User Characteristics 4](#_Toc380681037)

[User Objectives 4](#_Toc380681038)

[Constraints 4](#_Toc380681039)

[Developmental and technological constraints 5](#_Toc380681040)

[General Constraints 5](#_Toc380681041)

[Assumptions and Dependencies 5](#_Toc380681042)

[Assumptions 5](#_Toc380681043)

[Dependencies 5](#_Toc380681044)

[System Requirements 6](#_Toc380681045)

[Functional Requirements 6](#_Toc380681046)

[System processes 6](#_Toc380681047)

[User/System Interaction Sequence Diagrams 9](#_Toc380681048)

[Non-functional Requirements 11](#_Toc380681049)

[Interface Requirements 11](#_Toc380681050)

[Graphical User Interface 11](#_Toc380681051)

[Hardware interface 11](#_Toc380681052)

[Technical Requirements 11](#_Toc380681053)

# Introduction

## Purpose

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 and developers with regard to the system to be built.

## Vision

The marking system aims to afford lecturers an opportunity to centralize and digitalize the marking process for practical assessments carried evaluated during practical sessions in the informatorium. The system will also allow TAs and tutors to . The system will do so by allowing TAs and tutors to record students’ practical marks onto the system, using either their mobile phones or computers provided in the practical session venues. The system will also allow TAs and tutors to record students’ marks for other non-practical assessments onto a centralized database. This aims to avoid tedious paper-based recording of marks and all sundry processes associated with it.

## Scope

In essence, this marking-system serves to achieve one core task:

1. Recording of student assessment grades onto 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.

# General Description

This section gives an overall general description of the system to be developed based on the client’s explanation and description of the system.

## User Problem Statement

The user problem is the current mark-recording system being used within the Department of Science and other departments at the University of Pretoria. The current system heavily involves the module lecturer, and the client requires that a digital solution be made to reduce and if possible eliminate the lecturers’ involvement in the mark-recording process.

The current mark-recording system involves the following process:

* For practical sessions in the informatorium;
  + Students complete their practical assignments and bring them to their booked practical sessions.
  + The marker (TA or tutor) then traverses through all the students, assessing their practical assignments and awarding marks by recording these marks on a piece of paper populated with a rubric for the particular practical assignment and all the students who are supposed to be in that practical session.
  + After the practical, the tutor then hands the mark-sheet to the module lecturer who then records the marks onto the master mark document.
* For written and other assessments not assessed in the informatorium;
  + Students submit their work either by uploading the work onto the Computer Science website, or by handing in a hard copy of their work to the lecturer or TA or tutor.
  + The TAs and tutors then grade the assessments and then record the marks onto a mark sheet, which they then give to the lecturer who then records the marks onto the master mark document.

It is clear from the above processes how tedious and lengthy the process of assessing and recording marks onto a master mark document can be, hence the request for a more effective and efficient digital solution which will eliminate most of the sundry steps and internal processes required to record and document students’ marks, which will essentially reduce the involvement of the lecturer in the mark recording process.

## User Characteristics

This system will be used by the following users:

* HOD - The Head of Department for which ever department that is

using the system.

* Lecturers - The module lecturers responsible for a module with

assessments graded using this system.

* Teaching Assistants - The individuals assigned by lecturers responsible for

grading and recording students’ assessment mark.

* Tutors - The individuals assigned by lecturers responsible for

grading and recording students’ assessment mark.

* Students - Students registered for modules using this marking system.

## User Objectives

Each user of the system is meant to use the system to accomplish the following objectives:

* HOD
  + To administer and review the system.
  + To assign lecturer’s to their respective modules on the system.
  + To view student marks.
* Lecturers
  + To assign markers to marking sessions and assign students to markers (i.e. practical sessions and other sessions where students’ assessments are graded).
  + To lock and finalize mark sheets for marking-session once the marking session are complete.
  + To review relevant audit logs.
  + To view student marks.
  + To grade and record students’ marks onto the system when required.
  + To alter students’ marks on the system if necessary when required.
* Teaching Assistants
  + To grade and record students’ marks onto the system.
  + To alter students’ marks on the system if necessary.
* Tutors
  + To grade and record students’ marks onto the system.
* Students
  + To view their marks for assessments already graded.

## Constraints

Each user of the system is restricted in the following manner:

* HOD - May only administer the system by reviewing and not

editing relevant audit logs.

- Restricted from doing anything else not mentioned in the user objectives for the HOD.

* Lecturers - May only view and not edit relevant audit logs.

- Restricted from doing anything else not mentioned in the user objectives for the lecturer.

* Teaching Assistants - Should provide reason if altering a student’s marks.

- Should only have access to mark sheets for practicals, assessments and students they are assigned to.

- Restricted from doing anything else not mentioned in the user objectives for the teaching assistants.

* Tutors - Should provide reason if altering a student’s marks.

- Should only have access to mark sheets for practicals, assessments and students they are assigned to.

- Restricted from doing anything else not mentioned in the user objectives for the teaching assistants.

* Students - Restricted from doing anything else not mentioned in the user objectives of the student.

### Developmental and technological constraints

In developing this system, the following developmental and technological constraints must be noted as requested by the client:

* System should have web-interface.
* System should have mobile-interface, android interface in particular.
* System should interface with LDAP.
* System should run with Django.
* System should make use of SOAP-Interface.

### General Constraints

All users of the system are restricted from doing the following:

* Edit any audit log on the system.

## Assumptions and Dependencies

This section outlines the assumptions and dependencies made regarding the system.

### Assumptions

Assumptions regarding users:

* All users of the system are technology friendly and can easily use the system for its purpose.

Assumptions regarding use-environment:

* This system assumes that markers (TAs and tutors) will either have a web-enabled mobile phone or a computer with internet access in the venue where they will be recording the marks.
* This system assumes the venue or environment where markers (TAs and tutors) are using this system has second or third-generation network coverage.

### Dependencies

This section outlines the dependencies which the system relies on in order to function effectively and efficiently.

The following are deemed as dependencies:

* Markers (TAs or tutors) will have the necessary technology in their marking venues to enable them to use the system.

# System Requirements

This section outlines all the functional, non-functional and other requirements of the system.

Scale of importance:

1 - Not very important. Item/feature can be eliminated from the system

without causing harm to the system.

2 - Not necessarily important. Item/feature can be eliminated from the system

but only causing negligible harm.

3 - Important. Items/features that cannot be eliminated from the system

without causing significant harm and are somewhat critical to the proper and effective functioning of the system.

4 - Critical. Items/features that are entirely critical to the system and cannot be

eliminated for whatever reason and under any circumstance.

## Functional Requirements

### System processes

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

1. Downloading mobile application
   1. Elaboration – Users shall be able to download the mobile version of this application.
   2. Importance – 4.
   3. Dependency level – Without downloading the application, users cannot make use of the applications functionality. Hence this process is a critical process.
   4. Pre-condition – User does not have mobile application.
   5. Post-condition – User has mobile application downloaded and installed on their mobile phones.
   6. Requestor - Client
2. Signing In
   1. 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.
   2. Importance – 4.
   3. Dependency level – Without this process, users of this system will not have access to the system, which will render the system unusable.
   4. Pre-condition – This feature on the system will be properly implemented using appropriate technology.
   5. Post-condition – This feature functions as it should with no defects or short-comings.
   6. Requestor – Client.
3. Signing out
   1. 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.
   2. Importance – 4.
   3. 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.
   4. Pre-condition – This feature on the system will be properly implemented using appropriate technology.
   5. Post-condition – This feature functions as it should with no defects or short-comings.
   6. Requestor – Client.
4. Adding lecturers to the system and assigning them to relevant modules.
   1. 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.
   2. Importance – 4.
   3. 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.
   4. Pre-condition – This feature on the system will be properly implemented using appropriate technology.
   5. Post-condition – This feature functions as it should with no defects or short-comings.
   6. Requestor – Client.
5. Administering system.
   1. Elaboration – This process will allow the HOD to administer the system and ensure that the system is operating smoothly.
   2. Importance – 2.
   3. Dependency level – Without this process, the system becomes open to fault that may possibly hinder the effective functioning of the system.
   4. Pre-condition – This feature on the system will be properly implemented using appropriate technology.
   5. Post-condition – This feature functions as it should with no defects or short-comings.
   6. Requestor – Client.
6. Reviewing the system.
   1. Elaboration – This process will allow the HOD and lecturers to view user activity on the system by reviewing audit logs to ensure that no misuse of the system occurs. It will also allow the HOD and lecturers to follow up on cases that may arise relating to student marks by reviewing user activity on the audit logs.
   2. Importance – 4.
   3. Dependency level – Given that no misuse occurs, this feature does not prove fatal to the operation of the system. However, Murphy’s Law suggests that if anything can go wrong, it will, hence it’s the auditability, security and effectiveness of the system may rely heavily on this feature.
   4. Pre-condition – This feature on the system will be properly implemented using appropriate technology.
   5. Post-condition – This feature functions as it should with no defects or short-comings.
   6. Requestor – Client.
7. Assigning markers to practical sessions and assigning students to markers.
   1. Elaboration – This process allows lecturers to assign markers to marking sessions and to assign that marker a selected group of students.
   2. Importance – 4.
   3. Dependency level – Without this feature, the system becomes ineffective as there are no markers assigned to record student marks.
   4. Pre-condition – This feature on the system will be properly implemented using appropriate technology.
   5. Post-condition – This feature functions as it should with no defects or short-comings.
   6. Requestor – Client.
8. To lock and finalize mark sheets for marking sessions once the session is complete.
   1. Elaboration – This process involves the system automatically locking a mark-sheet after practical session is compete, or the lecturer locking the mark-sheet himself/herself.
   2. Importance – 3.
   3. 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.
   4. Pre-condition – This feature on the system will be properly implemented using appropriate technology.
   5. Post-condition – This feature functions as it should with no defects or short-comings.
   6. Requestor – Client.
9. To record students’ marks onto the system.
   1. Elaboration – This process involves the marker entering a student’s mark into the system.
   2. Importance – 4.
   3. Dependency – Without this process, the purpose of the system is defeated.
   4. Pre-condition – This feature on the system will be properly implemented using appropriate technology.
   5. Post-condition – This feature functions as it should with no defects or short-comings.
   6. Requestor – Client.
10. To alter students’ marks on the system.
    1. Elaboration – This process describes a marker altering a student mark on the system after having already recorded a mark for the student.
    2. Importance – 3.
    3. Dependency level – This process allows markers to correct a student’s 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.
    4. Pre-condition – This feature on the system will be properly implemented using appropriate technology.
    5. Post-condition – This feature functions as it should with no defects or short-comings.
    6. Requestor – Client.
11. Import/Export .csv files
    1. Elaboration – Users of this system shall be able to import or export marks to and from the system in csv format.
    2. Importance – 2
    3. Dependency level – The system is not highly dependent on this process, and can still function to its purpose without this process.
    4. Pre-condition – 1. User wishes to download mark sheet in csv format.

2. User wishes to upload mark sheet in csv format.

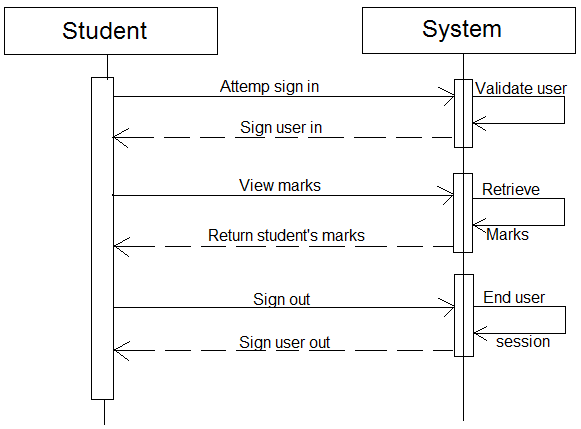
* 1. Post-condition – 1. User successfully downloaded mark sheet in csv format.

2. User successfully uploaded marks in csv format.

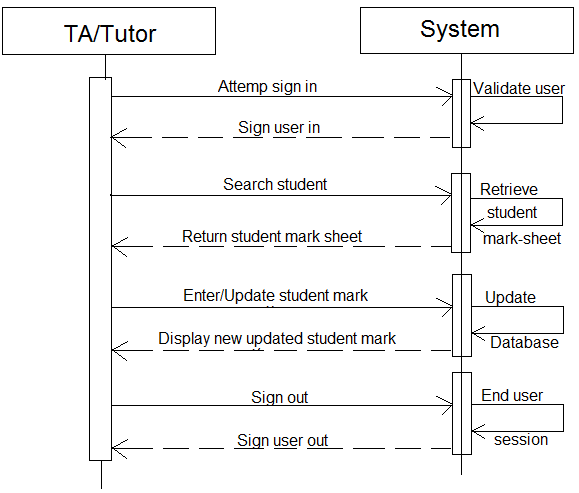
1. To view marks on the system.
   1. Elaboration – This process describes students, lecturers, and the HOD being able to view a student’s marks on the system. Student marks should be visible as individual and cumulative marks till any given point in time. These marks should also have statistical measures and graphical representations such as graphs.
   2. Importance – 4
   3. Dependency level – Without this process, the purpose of the system is defeated.
   4. Pre-condition – This feature on the system will be properly implemented using appropriate technology.
   5. Post-condition – This feature functions as it should with no defects or short-comings.
   6. Requestor – Client.

### User/System Interaction Sequence Diagrams

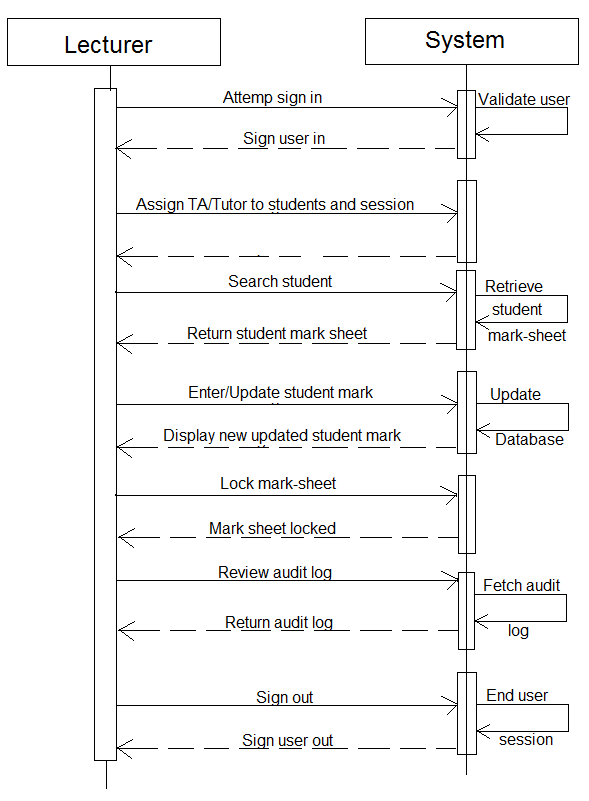
Student/System Interaction



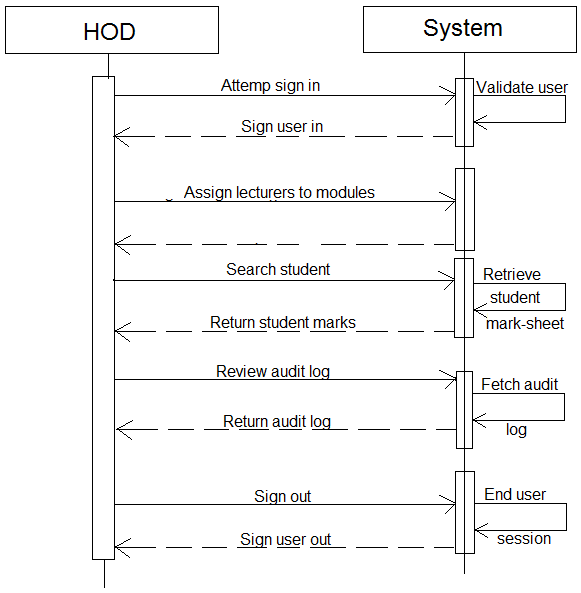
TA and Tutor/System Interaction



Lecturer/System Interaction



HOD/System Interaction



## Non-functional Requirements

1. Security - The system should provide adequate and reliable security features.
2. Authentication - the system should provide adequate authentication for all its’ users.
3. Auditability - All actions taken on the system should be easily traceable to the

source of the action.

1. Testability - The system should be testable and should be tested using Django.
2. Scalability - The system should provide a high level of scalability in that it should

operate at an acceptable level of performance.

## Interface Requirements

### Graphical User Interface

All graphical user interfaces developed for this system should be simple and user-friendly to all users of the system.

### Hardware interface

The system will operate on android mobile devices along with internet-enabled computers.

## Technical Requirements

The following technical specifications are enforced and required by the client:

* A web and mobile application version of the system should be available.
* System should interface with aldap.
* System should be tested with Django.
* System should accommodate SOAP interface.
* System should operate in real-time against the database.