General Sir John Kotelawala Defense University Faculty of Computing Department of Computer Science

BSC DEGREE IN COMPUTER SCIENCE/ COMPUTER ENGINEERING/ SOFTWARE ENGINEERING

REQUIREMENTS ENGINEERING – CS2052

INTAKE 36

SYSTEM REQUIREMENTS SPECIFICATION

Group Details		
Group Number	**Student Number**	**Student Name**
Group 9	D/BSE/19/0004	C S Wanigasooriya
	D/BSE/19/0003	MFAR Fernando
	D/BSE/19/0005	SHCKD Silva
	D/BCE/19/0001	PSM De Silva
	D/BSE/19/0001	PH Perera
	C/BCE/5714	RLBJS Jayawardhana
Project Details		
Project Title	Result Management System	
Supervisor	Mr. Budditha Hettige	
Co-Supervisor		

Name – C S Wanigasooriya

Reg No – D/BSE/19/0004

Table of Contents

1. Introduction

1.1 Purpose

The purpose of the Software Requirements Specification document is to clearly define the system under development, namely the University Result Management System (URMS). The intended audience of this document includes the Computing Faculty Head of Department, respective supervisors, and the end users of the URMS. Other intended audience includes the potential developers in the development group.

1.2 Scope

The Faculty of Computing of the Kotelawala Defense University (KDU) conducts its end semester examinations bi-annually. However, the time lapse between the conclusion of the examination and the release of results is disproportionately large. Therefore, the new URMS will allow the following functionality online: to get summarized reports of student result, to rent videos, to modify user information, and to update results. Authorized Department personnel may use the URMS to process the Grade Point Average (GPA) or Semester Grade Point Average (SGPA), to add or remove marks, and to update marks. The URMS is intended to decrease the University's results processing period by centralizing all required actions to one system.

1.3 Definitions. acronyms and abbreviations

Term	**Definition**
KDU	*Kotelawala Defence University*
CS	*Computer Science*
GPA	*Grade Point Average*
HOD	*Head of Department*

1.4 References

[1] IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.

[2] Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990 IEEE). Ieeexplore.ieee.org. 2020. [online] Available at: <http://ieeexplore.ieee.org>

1.5 Overview

Section 2 of the SRS describes the product in more detail. Section 3 provides a complete list of the functional requirements and non-functional requirements along with the class diagram, and the use case diagram. The appendices appear next.

2. Overall Description

2.1 Product Perspective

The URMS is a web-based system. The system interacts with other systems, the University's student information system and the browsers used by URMS authorized users. The system provides a secure environment for all result managements and for the storing and retrieving of confidential student information.

2.2 Product Functions

The URMS allows authorized users such as the administrator staff and HOD to process summarized reports of individual students, to calculate and allocate grades from CAS marks and End Semester marks. Administrator staff or the HOD should assign Moderators and Setters using the URMS. Upon becoming a assigned as a setter or moderator and logging into the URMS, the URMS provides the functionality for allocating marks, viewing allocated marks, and updating marks with the confirmation of HOD. The setter and moderator are Lectures that are allocated for each subject for reviewing marks. The setter can enter finalized marks into the URMS, while the moderator validates the results.

The administrator staff Computing faculty use URMS to process the return of students' final grades and GPA. The HOD uses URMS to add setters or moderators into the system, remove setters or moderators from the system, modify information and has privileges of administrative staff.

2.3 User Characteristics

The two main groups of URMS users are Administrative staff and lectures. A lecturer can view, search and monitor student results. The amount of product training needed for a lecturer is none since the level of technical expertise and educational background is unknown. The only skill needed by a lecturer is the ability to browse a website.

Lecturers have two categories Setters and Moderators have more privileges and subject allocated lectures. Marks allocation and modification is only allowed to the assigned Setter or Moderator.

HOD is a subset of the administrative staff with additional rights. Result modification or retrieval is done with the confirmation of the HOD. The amount of product training is high since the proper management of results is crucial for the release of results.

2.4 Constraints

This system provides web access for all customer and member functions. The user interface will be intuitive enough so that no training is required by lecturers, but training is required for the administrative staff to manage the systemically. All credentials and results of students will be done in a secure environment. Accessing the system outside of the premises should be done responsibly. User Negligence might lead to revertible consequences, therefore proper awareness of the system should be trained.

2.5 Assumptions and dependencies

URMS is mainly developed in PHP and therefore requires no additional software to be installed on the user's system expect for a web browser. URMS requires a stable and secure internet connection. This applies to Windows and Linux users.

3. Specific Requirements

3.1 External Interfaces

1. Login UI

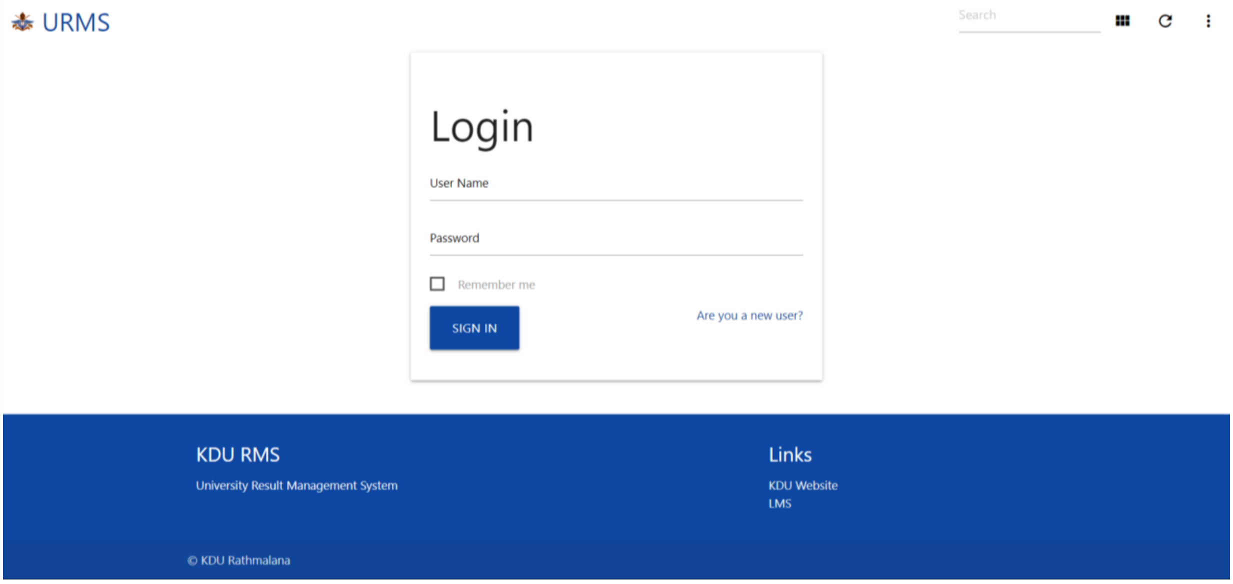


Fig 1.1 Login Screen of URMS

2. Main UI

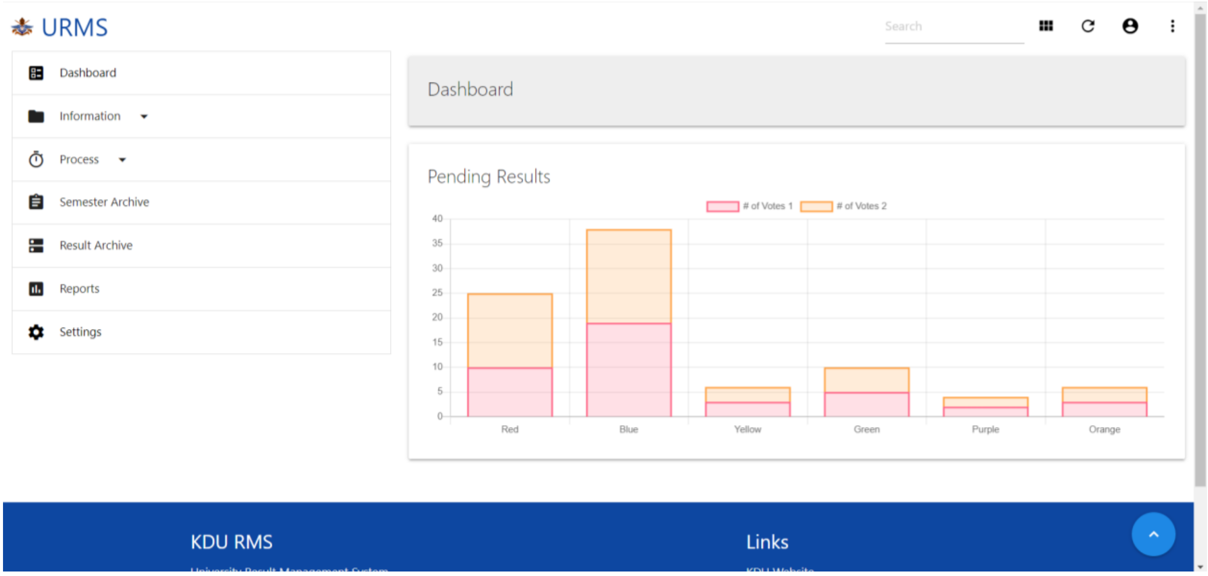


Fig 1.2 Main UI of URMS

3. User Information Update UI

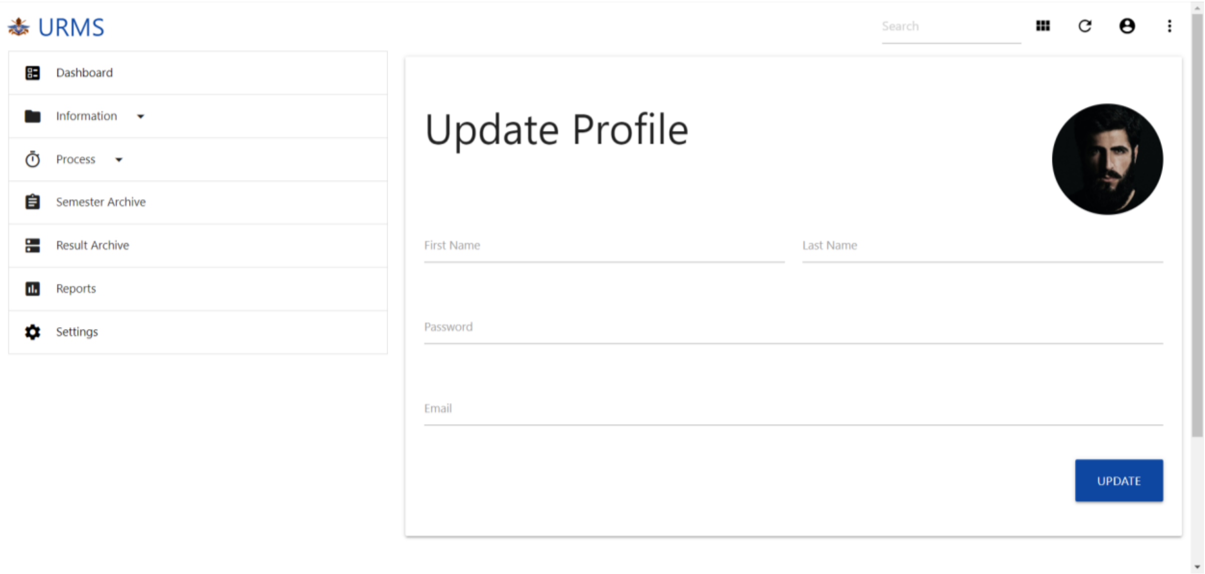


Fig 1.3 Profile Update UI

3.2 Functional Requirements

Functional Requirements describe the tasks expected to be performed using our system. In our web application we are going to provide following features:

- A new document to be filled which is a collaboration of the three sheets- detailed mark sheet, marks returned sheet and comments sheet. This sheet contains all relevant fields which make the document contain less redundant fields and substitute certain fields which simply can be generated automatically. (e.g.: - **Total** which can be calculated by adding marks for each question)
- Faster transfer of documents to and from relevant people
- Providing summarized reports for each individual student
- Ability to confirm setter and moderator
- Preparing Class/Group-wise Result
- Graphical view of students' performance

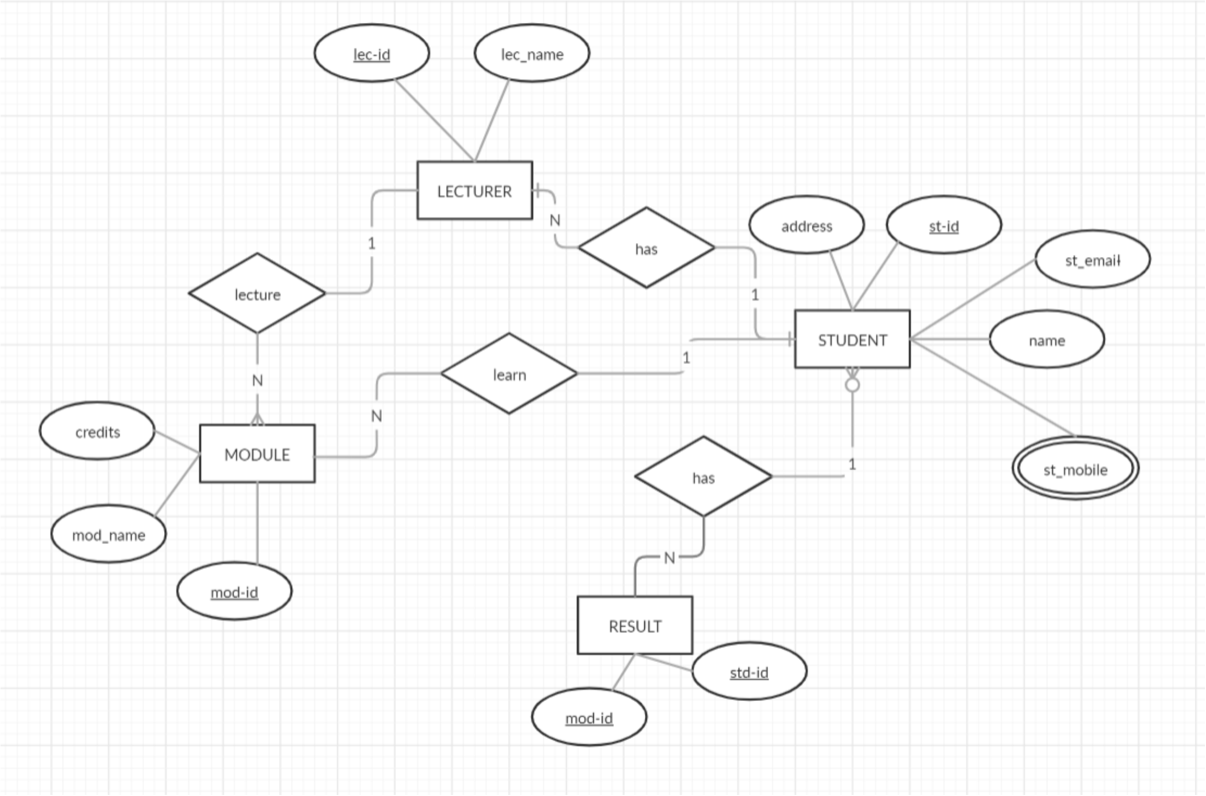
- Allows to generate result of students for different examination and different types of exams during the year as oral, practical, written, etc. [1]
- Manage the results of the university students and keep a record of results of the students throughout the year.

3.3 Non-Functional Requirements

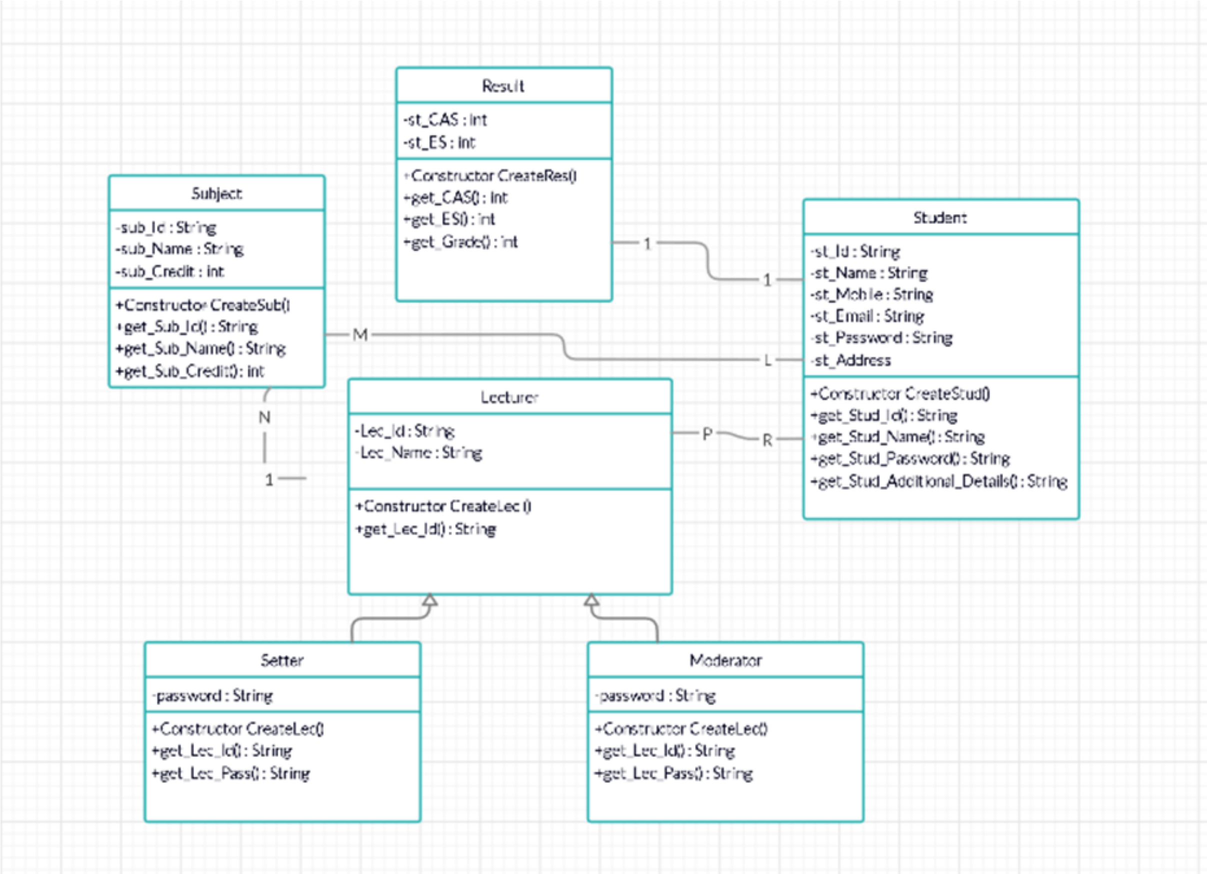
- Usability: The web app should be able to use without any extra effort. The initial configurations should be easy to learn.
- Reliability: since we are providing students information and some notifications to the lecturers, that information must be reliable, and the user should be able to depend on those details.
- Start -up-time: The start-up time of the app should be minimum in order to increase the efficiency.
- Security: Outside users cannot access student and lecturer details. Need to take database backup in case of crashing database due to virus or OS failure.
- Scalability: It can be easily run in any platform.
- Application size: The web app size must be compact to download.
- Responsiveness: The web app should complete the task according to the given time.
- Efficiency: User can easily access the system and search very fast.
- Delivery: The entire system is expected to be delivered in one year of time.

3.4 System Architecture

ER Diagram

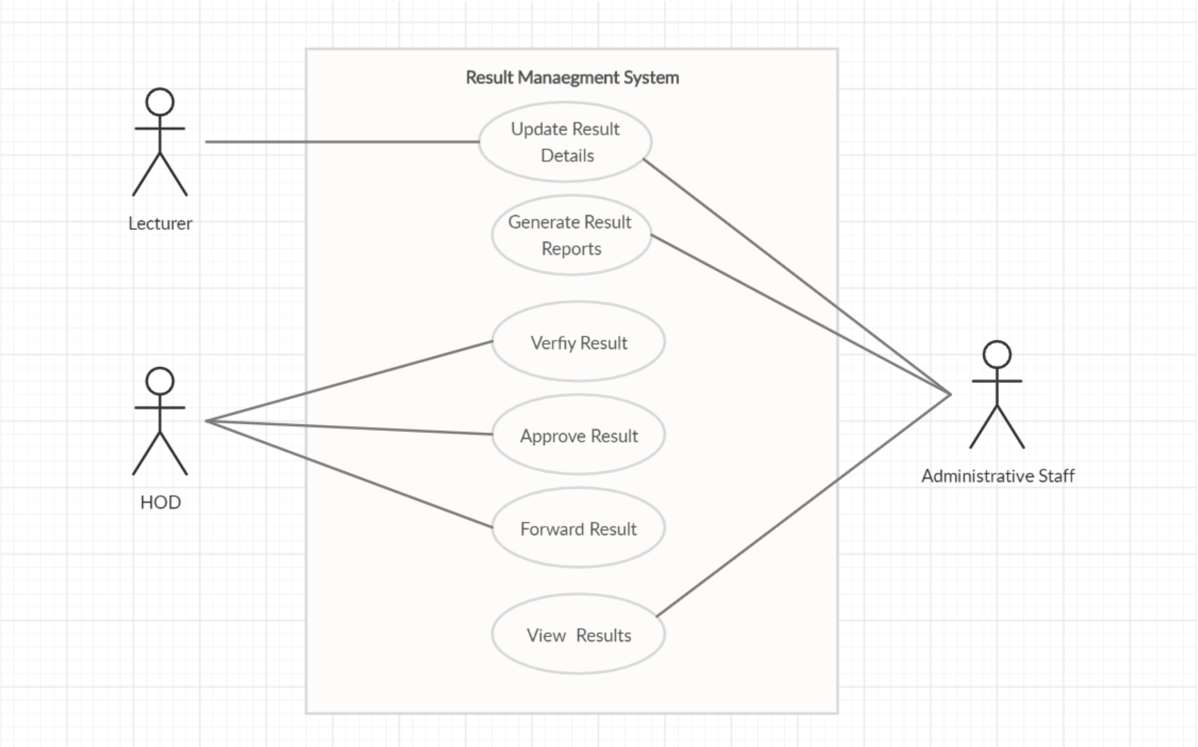


Class Diagram

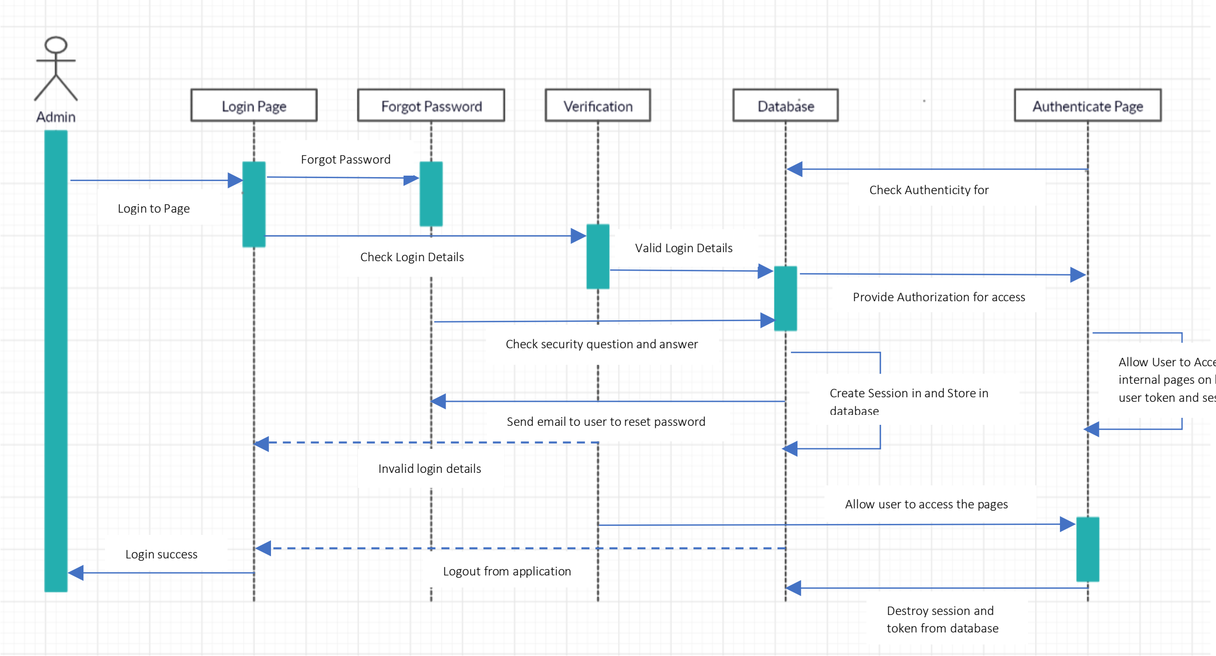


3.5 System Model

Use Case Diagram



Sequence Diagram


Admin
Login Page
Forgot Password
Verification
Database
Authenticate Page
Forgot Password
Login to Page
Check Authenticity for
Check Login Details
Valid Login Details
Provide Authorization for access
Check security question and answer
Allow User to Access the internal pages on base of user token and session
Create Session in and Store in database
Send email to user to reset password
Invalid login details
Allow user to access the pages
Login success
Logout from application
Destroy session and token from database

3.6 Performance Requirements

URMS requires a system with at least a 500-megahertz CPU and 128 megabytes of RAM and an OPENGL 1.2 compatible graphics card. However, these requirements can support URMS a stable and secure internet connection with a browser with v8 engine is compulsory.

3.7 Design Constraints

- Design of result management functionality, and all grade allocation procedures shall conform to standard university practices.
- Report formats shall offer a standard configuration but must be completely customizable by the user.
- Data naming shall conform to a standard data naming convention.

3.8 Appendix

Interview with Head of Department:

The following information was collected over the course of one interview:

General Information

Purpose to evaluate:

- whether current information system can generate sufficiently accurate results
- whether current system maintains backups
- whether current system has some type of archival mechanism
- whether current system is secure
- the importance of accuracy on issuing results
- general complaints about the current system

Name: Dr. Pradeep Kalansooriya

Job Title: Head of Department

Number of Years in position: Unknown

Questions asked about the current system

What is the scope behind our software?

Since this system is a fragile and sensitive component of the university and considering that the system, we are trying to build is at the development stage, we chose to make the system only focused on the faculty of computing. For now, our scope is limited with in the department.

Note: As soon as the system has been declared a success then and only then will we broaden the scope towards all faculties of the university.

Who gets to mark the relevant papers?

First and foremost, setter checks the paper and fills the detailed mark sheet, then later the moderator re-checks the paper and fills another detailed mark sheet.

What is the process behind the current results management system?

The current system is mainly processed using hard photocopies of the four sheets (Detailed Mark Sheet x2, Marks Return Sheet, and Comment Sheet) which is not time effective again since the listed sheets must be hand delivered to each relevant person. Another problem that has been faced to persuade our team to proceed with this project is that lecturers don't get a summarized report of each individual student.

Steps are shown as below.

1.Setter checks the paper and fills the detailed mark sheet.

2.Moderator again checks the paper and fills another detailed mark sheet.

3.Both mark sheets are sent to the HOD.

4.HOD cross checks the two given sheets to see if there is an unacceptable pair of marks.

5.If there is a big difference, then the HOD adjusts the marks as he/she sees fit.

6.The mark sheets (including the mark sheet containing the final marks for each student) are then sent back to the setter for him/her (Setter) to continue to proceed on to the marks return sheet and the comment sheet.

7.Then the marks return sheet and the comment sheet are sent to the examination dept.

Our system is based on issuing end-semester results, calculating the GPA of each student of the Faculty of Computing at General Sir John Kotelawala Defence University and finalizing it at the end of the final semester. This system's purpose is also to further diminish the long-time consumption for lecturers as well as when entering the marks into the database. It will also reduce the amount of paperwork for each lecturer, the time taken to pass-on the said paperwork to the other relevant persons.

What are the additional features we could add into the software?

Providing a summarized report on the performance of each individual student.

Who are the authorised respective individuals for each stage?

Paper marking: Each module's respective setter and moderator.

Semi-Final Grade evaluation: Faculty head of department.

Final Grade evaluation: Setter.