**University Student Management Information System**

**System Requirements Specification**

**Table of Contents**

1 Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definition, Acronym and Abbreviations 4

1.3.1 Definitions 4

1.3.2 Acronyms and Abbreviations 5

2 Overall System Description 6

2.1 Overview 6

2.2 Context Diagram 6

3 Functional Requirements 7

3.1 User Management 7

3.1.1 Add new admin users 7

3.1.2 View admin users 8

3.1.3 Edit admin user details 9

3.1.4 Delete admin user 10

3.2 User Application Management 11

3.2.1 Lecturer Management Requirement 12-16

3.2.2 Student Management Requirement 17-20

3.2.3 Student Management Requirement 21-26

4 Non-functional Requirements 27

4.1 Security Requirements 27

4.2 Performance Requirements 27

4.3 Portability Requirements 27

4.4 Usability Requirements 28

4.5 Manageability Support Requirements 28

5 External Interface Requirements 29

**1.Introduction**

**1.1 Purpose**

The purpose of this document is to explain both functional and nonfunctional requirements and other factors regarding the mini project about University Student Management Information System. After this the project will be called **USMIS.**

**1.2 Scope of management system of faculty of technology**

Our management system helps to manage the entire user profiles, course details, students marks, student attendance, notices, timetable and medicals of faculty of technology. In the system there’re some users with specific authorities those are admin, lecturer, technical officer, and student. Each these users have specific authorities in the system. Some of those authorities are,

* Manage and see academic programs and courses offered.
* Financial management and budgeting
* Manage faculty and staff,
* Student recruitment, admission, and retention
* Manage and see research activities and collaborations.
* Manage facilities and equipment.
* Data management
* Compliance with relevant laws, regulations, and accreditation standards.

As well our system provides brief explanation about each student results, attendance details and ability to attend for examination. Rather than that we can see each student grades for each subject and GPA through this system.

**1.3. Definition, Acronym and Abbreviations**

**1.3.1. Definition**

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfill the needs of all stakeholders such as business and users.

This project defines the TECMIS of University of Ruhuna. TECMIS is the system that manage students’ details, medicals, course details, course registrations, exam registrations, and results. As well as the mentors for the students.

**1.3.2. Acronyms and abbreviations**

GPA – Grade Point Average

CGPA – Cumulative Grade Point Average

SGPA – Semester Grade Point Average

TECMIS – Technology Management Information System

LMS – Learning Management System

**1.Overal System Description**

**2.1 Overview**

University Student Management Information System is a comprehensive system that allows university administrators to track and monitor the progress of their students. It helps in managing the academic, administrative data of students.

Rather than using a paper-based system we can do those tasks without any human related errors. As an advance feature there are authorized parties for tracking the information in securely.

After convert the University Student Management System to web-based application, it provides various features for each user profile, such as Admin, Lecturer, Student and Technical Officer.

**2.1 Context Diagram**

**Diagram

Description automatically generated**

**3. Functional Requirements**

**3.1 Add new admin users.**

**3.1.1 Add new admin users.**

|  |  |
| --- | --- |
| **Use case ID:** | 1 |
| **Use Case Name:** | Add new admin users |
| **Actors:** | Super admin  LMS |
| **Description:** | The purpose of this use case is to allow the super admin user to add low level admin users to the system. |
| **Trigger:** | Super admin clicks the ‘add new user’ button. |
| **Preconditions:** | Logged into the LMS as super admin. |
| **Post – conditions:** | If the user already in the system, display an error message. |
| **Normal Flow:** | 1. Super admin clicks ‘Add new user’ button in LMS. 2. Display a form to get new user details. 3. NIC 4. Username 5. New Password 6. Confirm Password 7. Email 8. Phone 9. Permission Level 10. Read only. 11. Read + Write. 12. Read + Write + Delete. 13. Click the ‘Add’ button to add the new user. |
| **Alternative Flows:** | Press the ‘Cancel’ button. |
| **Exceptions:** | If there is an admin with the same email, NIC displays an error. |
| **Includes:** | N/A |
| **Priority:** | N/A |
| **Frequency of Use:** | When new admin users need to be added to the LMS. |
| **Business Rules:** | N/A |
| **Special Requirements:** | N/A |

**3.1.2 View admin users**

|  |  |
| --- | --- |
| **Use case ID:** | 2 |
| **Use Case Name:** | View admin users |
| **Actors:** | Super admin  LMS |
| **Description:** | The purpose of this use case is to allow the Super admin to view existing admin users. |
| **Trigger:** | When super admin clicks the ‘View user button’. |
| **Preconditions:** | Logged into the LMS as super admin. |
| **Post – conditions:** | If the user is not in the system, display an error message. |
| **Normal Flow:** | 1. Super admin clicks ‘View user’ button in LMS. 2. Display list of registered users with the following details for each user. 3. NIC 4. Username 5. New Password 6. Confirm Password 7. Email 8. Phone 9. Permission Level 10. Read only. 11. Read + Write. 12. Read + Write + Delete |
| **Alternative Flows:** | N/A |
| **Exceptions:** | N/A |
| **Includes:** | N/A |
| **Priority:** | N/A |
| **Frequency of Use:** | When super admin clicks the ‘View user’ button. |
| **Business Rules:** | N/A |
| **Special Requirements:** | N/A |

**3.1.3 Edit admin user details.**

|  |  |
| --- | --- |
| **Use case ID:** | 3 |
| **Use Case Name:** | Edit admin user details. |
| **Actors:** | Super admin  LMS |
| **Description:** | The purpose of this use case is to allow the super admin to edit details of an existing admin user. |
| **Trigger:** | The super admin clicks the ‘edit’ button of a specific user in the ‘view user’ list. |
| **Preconditions:** | Logged into the LMS as super admin. |
| **Post – conditions:** | N/A |
| **Normal Flow:** | 1. Click the edit button of a specific user in the view user list. 2. Display a form with the following details of the selected user.    1. NIC    2. Username    3. New Password    4. Confirm Password    5. Email    6. Phone    7. Permission Level 3. Read only. 4. Read + Write. 5. Read + Write + Delete. 6. Click the CONFIRM button to confirm edit. |
| **Alternative Flows:** | Click the ‘Cancel’ button. |
| **Exceptions:** | N/A |
| **Includes:** | N/A |
| **Priority:** | N/A |
| **Frequency of Use:** | When super admin clicks the ‘Edit’ button. |
| **Business Rules:** | N/A |
| **Special Requirements:** | N/A |

**3.1.4 Delete admin users.**

|  |  |
| --- | --- |
| **Use case ID:** | 4 |
| **Use Case Name:** | Delete admin users |
| **Actors:** | Super admin  LMS |
| **Description:** | The purpose of this use case is to allow the Super admin to delete an existing user from the system. |
| **Trigger:** | Super admin selects the ‘Delete’ button of the specific user in the view user section. |
| **Preconditions:** | Logged into the LMS as super admin. |
| **Post – conditions:** | N/A |
| **Normal Flow:** | 1. Super admin click the delete button of a specific user in the view user section. 2. Display a confirmation dialog. 3. Delete user. |
| **Alternative Flows:** | If the super admin clicks the cancel button, it keeps the data. |
| **Exceptions:** | N/A |
| **Includes:** | N/A |
| **Priority:** | N/A |
| **Frequency of Use:** | Every time the super admin clicks the ‘Delete’ button. |
| **Business Rules:** | N/A |
| **Special Requirements:** | N/A |

* 1. **User Application Management**

**Lecturer Management requirements**

• Lecturer Can update their profile except username and password

• Lecturer Can modify and add materials to courses

• Lecturer Can upload marks for all kind of exams

• Lecturer Can see student details

• Lecturer Can see student eligibility

• Lecturer Can see student marks, grades and GPA

• Lecturer Can see attendance and medical records of students

• Lecturer Can see notices

|  |  |
| --- | --- |
| **Use Case ID** | 1 |
| **Use Case Name** | update their profile except username and password. |
| **Actors** | Lecturer LMS |
| **Description** | The purpose of this use case is to allow update their profile details.  (Preferred language Forum,  preferences Editor,  preferences Course,  preferences Calendar,  preferences Message,  preferences Notification) |
| **Trigger** | Lecturer Enter Username and Password |
| **Preconditions** | Logged into the LMS as Lecturer |
| **Post-conditions** | If the Lecturer is Password or user name is incorrect, display an error message. |
| **Normal Flow** | Lecturer can update his profile Who wanted.  Preferred language Forum,  preferences Editor  preferences Course  preferences Calendar  preferences Message  preferences Notification |

|  |  |
| --- | --- |
| **Use Case ID** | 2 |
| **Use Case Name** | modify and add materials to courses |
| **Actors** | Lecturer LMS |
| **Description** | The purpose of this use case is to Lecturer can modify and add course materials.  (Lecture notes, quizzes, In class activities, upload links, etc.) |
| **Trigger** | Lecturer Enter Username and Password and go through under the course unit. |
| **Preconditions** | Logged into the LMS as Lecturer |
| **Post-conditions** | If the Lecturer is Password or username is incorrect, display an error message. |
| **Normal Flow** | After The modify and add materials to courses updated the system |

|  |  |
| --- | --- |
| **Use Case ID** | 3 |
| **Use Case Name** | upload marks for all kind of exams |
| **Actors** | Lecturer LMS |
| **Description** | The purpose of this use case is to Lecturer Can upload marks for all kind of exams.  Exam marks uploaded with in the grade separate each Students. |
| **Trigger** | Lecturer Enter Username and Password and go through under the Exam Result unit. |
| **Preconditions** | Logged into the LMS as Lecturer |
| **Post-conditions** | If the Lecturer is Password or user name is incorrect, display an error message. |
| **Normal Flow** | After uploaded marks for all kind of exams Student Can see Result in the Student Accounts. |

|  |  |
| --- | --- |
| **Use Case ID** | 4 |
| **Use Case Name** | see student details |
| **Actors** | Lecturer LMS |
| **Description** | The purpose of this use case is to Lecturer Can see student details with and student eligibility and student GPA in what student name, what he or she department, what activities student did, all subjects’ details, etc. |
| **Trigger** | Lecturer Enter Username and Password and go through under the students details unit. |
| **Preconditions** | Logged into the LMS as Lecturer |
| **Post-conditions** | If the Lecturer is Password or user name is incorrect, display an error message. |
| **Normal Flow** | After checking the student's information, the lecturer decides the exams to be retaken by the student and the actions to be taken. And inform student. |

|  |  |
| --- | --- |
| **Use Case ID** | 5 |
| **Use Case Name** | see attendance and medical records of students |
| **Actors** | Lecturer LMS |
| **Description** | The purpose of this use case is to Lecturer Can see attendance and medical records of students |
| **Trigger** | Lecturer Enter Username and Password and go through under the students details unit. |
| **Preconditions** | Logged into the LMS as Lecturer |
| **Post-conditions** | If the Lecturer is Password or username is incorrect, display an error message. |
| **Normal Flow** | Lecturer can see,   * Student attendance in each lecture * See medical reports uploaded by each student.   By this, the lecturer will decide whether the concerned student will face the final exam or not and inform the student. |

**Student management**

* Can update only contact details and profile picture of their profile.
* Can see attendance details.
* Can see medical details.
* Can see their course details.
* Can see their grades and GPA.
* Can see their timetables.
* Can see notices.

|  |  |
| --- | --- |
| **Use case ID:** | 1 |
| **Use Case Name:** | Edit student personal details |
| **Actors:** | Registered student in the LMS |
| **Description:** | The purpose of this use case is to allow to the student to edit and update only contact details and profile picture of their profile. |
| **Trigger:** | Registered student can click “Edit Personal Details” button |
| **Preconditions:** | First registered student to the LMS |
| **Post – conditions:** | If the user is not registered the LMS get an error message |
| **Normal Flow:** | 1. Student clicks ‘Edit Personal Details’ button in LMS. 2. Display to list of personal details. 3. Username 4. New Password 5. Confirm Password 6. Email 7. Phone 8. Profile picture |

|  |  |
| --- | --- |
| **Use case ID:** | 2 |
| **Use Case Name:** | See attendance details |
| **Actors:** | Registered student in the LMS |
| **Description:** | The purpose of this use case is to allow to see the attendance for students (Student attendance percentage) |
| **Trigger:** | Registered students enter the username and password student can see the attendance list to user |
| **Preconditions:** | Logging to the LMS as student |
| **Post – conditions:** | Student Enter the password and username incorrectly display error message |
| **Normal Flow:** | After see the attendance percentages and other details student can maintain the attendance and submit the medicals. |

|  |  |
| --- | --- |
| **Use case ID:** | 3 |
| **Use Case Name:** | See medical details |
| **Actors:** | Registered student in the LMS |
| **Description:** | The purpose of this use case is to allow to the student to see the medical details. |
| **Trigger:** | Registered students enter the username and password after logging “medical details” tab and student can see the medical details to student. |
| **Preconditions:** | Logging to the LMS |
| **Post – conditions:** | Student Enter the password and username incorrectly display error message |
| **Normal Flow:** | After seeing the medical reports students. If students have the any of the changes student can request to change medical details |

|  |  |
| --- | --- |
| **Use case ID:** | 4 |
| **Use Case Name:** | See students course details |
| **Actors:** | Registered student in the LMS |
| **Description:** | The purpose of this use case is to allow to the student to see the course details. |
| **Trigger:** | Registered students enter the username and password after logging “course details” tab and student can see the course details to student. |
| **Preconditions:** | Logging to the LMS |
| **Post – conditions:** | Student Enter the password and username incorrectly display error message |
| **Normal Flow:** | After see the medical reports students. If students can see more about subjects. |

|  |  |
| --- | --- |
| **Use case ID:** | 5 |
| **Use Case Name:** | See students grades and GPA |
| **Actors:** | Registered student in the LMS |
| **Description:** | The purpose of this use case is to allow to the student to see the GPA, grades and this calculate methods for the student. |
| **Trigger:** | Registered student enter the username and password after logging “Result ” tab and student can see the GPA, grades and this calculate methods for the student. |
| **Preconditions:** | Logging to the LMS |
| **Post – conditions:** | Student Enter the password and username incorrectly display error message |
| **Normal Flow:** | After see the Exam results and GPA students. If students, see the result fail student has a button to register those subjects. |

|  |  |
| --- | --- |
| **Use case ID:** | 6 |
| **Use Case Name:** | See student’s timetables |
| **Actors:** | Registered student in the LMS |
| **Description:** | The purpose this case shows the timetables for the student. |
| **Trigger:** | Registered student enter the username and password after logging “time table” tab and student can see their time tables |
| **Preconditions:** | Logging to the LMS |
| **Post – conditions:** | Student Enter the password and username incorrectly display error message |
| **Normal Flow:** | Student can click “Show timetables” button and he/she can see timetable |

|  |  |
| --- | --- |
| **Use case ID:** | 7 |
| **Use Case Name:** | See notices for student |
| **Actors:** | Registered student in the LMS |
| **Description:** | The purpose of this use case is to allow to student see the notices to student. |
| **Trigger:** | Registered student enter the username and password after logging “notices ” tab and students can see the notices in the your course module |
| **Preconditions:** | Logging to the LMS |
| **Post – conditions:** | Student Enter the password and username incorrectly display error message |
| **Normal Flow:** | Students can see notices in the university |

**Technical Officer Management requirements**

• Can update their profile except username and password

• Can add and maintain attendance details of students

• Can add and maintain medical details of students

• Can see notices

• Can see timetables of their department

|  |  |
| --- | --- |
| **Use Case ID** | 1 |
| **Use Case Name** | update profiles except username and password |
| **Actors** | Technical Officer  LMS |
| **Description** | The purpose of this use case is to allow update user’s profiles details. |
| **Trigger** | Technical Officer clicks on update users |
| **Preconditions** | Logged into the LMS as a Technical Officer |
| **Post-conditions** | If one or more fields are empty, display error “Please fill required field” |
| **Normal Flow** | 1. Technical Officer clicks on update users. 2. Display users. 3. Select a user. 4. display update form.   a. NIC  b. Username  c. Email  d. Phone  e. Permission Level  i. Read only  ii. Read + Write  iii. Read + Write + Delete  5.Click the ‘Add’ button to add the new user. |
| **Alternative Flows:** | Press the ‘Cancel’ button. |

|  |  |
| --- | --- |
| **Use Case ID** | 2 |
| **Use Case Name** | Can add and maintain attendance details of students |
| **Actors** | Technical officer  LMS |
| **Description** | The purpose of this use case is to keep attendance details of students |
| **Trigger** | Technical officers click Attendance button |
| **Preconditions** | Logged into the LMS as technical officer |
| **Post-conditions** | Error message “attendant sheet is empty” |
| **Normal Flow** | 1. click on attendance. 2. click on the add button. 3. drop file into upload area. 4. click save button |
| **Alternative Flows:** | Press the ‘Cancel’ button. |

|  |  |
| --- | --- |
| **Use Case ID** | 3 |
| **Use Case Name** | Can add and maintain medical details of students |
| **Actors** | Technical officer  LMS |
| **Description** | The purpose of this use case is to add and maintain medical details of students |
| **Trigger** | Technical officers click on medicals banner |
| **Preconditions** | Logged into the LMS as technical officer |
| **Post-conditions** | Display error” file is empty” when file is empty |
| **Normal Flow** | 1. click on banner” Medicals.” 2. click on “Add “button. 3. drag and drop files. 4. click on “save” button |
| **Alternative Flows:** | Press the ‘Cancel’ button. |

|  |  |
| --- | --- |
| **Use Case ID** | 4 |
| **Use Case Name** | see notices |
| **Actors** | Technical officer  LMS |
| **Description** | The purpose of this use case is to  Retrieve notices |
| **Trigger** | Technical officer view notices under notice section |
| **Preconditions** | Logged into the LMS as technical officer |
| **Post-conditions** | Print error “please select a notice” |
| **Normal Flow** | 1. clicking on the notice 2. Select a notice. |
| **Alternative Flows:** | 1. Press the ‘Cancel’ button. |

|  |  |
| --- | --- |
| **Use Case ID** | 5 |
| **Use Case Name** | see timetables of their department |
| **Actors** | Technical officer  LMS |
| **Description** | The purpose of this use case is to view Timetable of any department |
| **Trigger** | View timetable after clicking timetable option |
| **Preconditions** | Logged into the LMS as technical officer |
| **Post-conditions** | N/A |
| **Normal Flow** | 1. Click on Timetable. 2. Select a department under department list. |
| **Alternative Flows:** | Press the ‘Cancel’ button. |

**4. Non-Functional Requirements**

**4.1 Security Requirements:**

When it comes to the faculty Management information system, security factor should have a major concern. First, we have to look on to giving access to the system. That access privilege allows to change the system data, add and remove data in the system. We have to maintain one Admin to do these changes. That admin has access to all functions in the system. Then the lectures and technical officers. We have to give them some limited access to relating to their job role. Lastly students, they need to have access only on viewing, uploading, and downloading materials.

After the distribution of access, we need to limit the access by implementing a login form. Login can prevent unauthorized access to the system. Then, only relevant people can be access to the system.

**4.2 Performance Requirements:**

Non-functional performance requirements for a Learning Management System (LMS) relate to the system's speed and responsiveness. These requirements ensure that the LMS can handle its users' expectations and provide a positive user experience. The following are some examples of common non-functional performance requirements for an LMS:

**Response time:** The LMS should respond quickly to user requests such as page loading, form submissions, and playing media.

**Load time:** Even in high-traffic situations, the LMS should load pages and media efficiently.

**Concurrent users:** The LMS should be capable of supporting a large number of concurrent users, such as students and instructors, without affecting its performance.

**Data storage:** The LMS should be able to efficiently and effectively store and manage large amounts of data, such as student records and course materials.

**Media playback:** The LMS should be able to play multimedia content such as videos and audio without buffering or lag.

**4.3 Portability Requirements:**

**Operating System Compatibility:** The system must be compatible with the latest Windows, Mac OS, and Linux versions.

**Platform Independence:** The system should be platform-independent and run smoothly on a variety of hardware configurations.

**Code Portability:** The LMS should be designed with open-source technology and programming languages that follow industry standards. The system should be designed in a modular way that allows for easy integration with other applications.

**Data Interoperability:** The LMS should support the import and export of data in standard formats.

**4.4 Usability Requirements:**

**User friendly:** LMS Should have User-friendly interface as such as clean, intuitive, and simple interface for easy navigation to use.

**Mobile compatibility:** Those of students using LMS through Mobile Phone So it should be accessible and usable on multiple devices, including smartphones and tablets.

**Customizable dashboards:** Users should be able to personalize their dashboards and set up custom views to meet their individual needs.

**Reporting and analytics:** The LMS should provide meaningful and actionable data and analytics on student progress and performance.

**Course management:** The LMS should allow for the easy creation, management, and delivery of online courses and content.

**Interactive features:** The LMS should offer interactive features such as discussion forums, quizzes to enhance the learning experience.

**Scalability:** The LMS should be able to accommodate a growing number of users and courses without sacrificing performance or usability.

**Reliability:** The LMS should be dependable and consistently available to users, with minimal downtime for maintenance and updates.

**4.5 Manageability Support Requirements:**

Make sure the LMS supports all the file and learning formats and data collection capabilities user needs.

Docs

PDFs

Standard video file formats (WMV, mp4, vob)

Standard audio file formats (mp3, m4A, WAV, AAC)

Standard image formats (png, jpg, gif)

Proper training materials must have for create courses to users. The LMS must have built-in editor that allows users to create simple test and courses. A powerful authoring tool for building interactive courses, quizzes, and conversation simulations. The system must have the capability to provide reports upon request as soon as possible and clearly. And must maintain an audit log of all user transactions and have the ability to report on those audits.

**5. External interface requirements**

When we have to build the connection between software and the user, we need to implement a user interface. Using that interface, the user can do what they desire simply. This includes screen layouts, buttons, functions on screen.