**Software Requirements**

**Specification**

**For**

**University Management System**

**Table of Contents**

Table of Contents

Revision History

1. Introduction

1.1 Purpose

1.2 Product Scope

1.3 Document Conventions

1.4 References

1.5 overview

1.6 Intended Audience

2. Overall Description

2.1 Product Perspective

2.2 Product Functions

2.3 User Characteristics

2.4 Operating Environment

2.5 General Constraints

2.6 User Documentation

2.7 Assumptions and Dependencies

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

3.2.2 Hardware Interfaces

3.3.3 Software Interfaces

3.3.4 Communications Interfaces

3.2 Functional Requirements

3.2.1 Login and authentication

3.2.2 Student management

3.2.3 Course management

3.2.4 Faculty management

3.2.5 Enrollment management

3.2.6 Billing and fee management

3.2.7 Academic record management

3.2.8 Reporting and analytics

3.3Non-functional Requirements

3.3.1 Performance Requirements

3.3.2 Design Requirements

3.3.3 Quality Attributes

3.3.3.1 Usability

3.3.3.2 Reliability

3.3.3.3 Security

4. Other Requirements

4.1 Legal and regulatory requirements.

4.2 Training and support requirements

Appendix A: Analysis Models

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason of change | Version |
|  |  |  |  |
|  |  |  |  |

**1. Introduction**

**1.1 Purpose**

Our objective is to create a comprehensive 'University Management and Scheduling' application that as a bridge between theuniversity administration, faculty, staff, and students, and the development team responsible for building the system.The application aims to enhance the scheduling of courses, on-campus position selection, and provide tools to streamline the teaching process. Our development process will prioritize reliability, maintainability, and data security.

**1.2 Scope**

The scope of the project is to deliver a high-end application which would pragmatically serve many major tasks in a university such as course scheduling, selection of applicants for projects and a number of tools which can be used by the departments or professors. This system would act as a better bridging tool between students, professors and the department also.

**1.3 Definitions , acronyms and abbreviations**

Definitions:

University Management System: A web-based application that provides a comprehensive set of features to students, professors, and the university administration.

Course scheduling: The process of creating a schedule of courses for each semester and allowing students to register for them online.

On-campus position selection: The process of selecting students for on-campus positions such as GA, TA, and RA.

Tools for professors: Features that enable professors to manage their courses, streamline their duties, and make their teaching process smoother.

Acronyms:

SRS: Software Requirements Specification

GA: Graduate Assistant

TA: Teaching Assistant

RA: Research Assistant

RAT : Rationale

DEP : Dependencies

Abbreviations:

Prof: Professor

Dept: Department

Sys: System

Doc: Document

**1.4 References**

[1] Mishra, A., & Ghose, A. K. (2018). Design and implementation of an online university management system. International Journal of Engineering and Advanced Technology, 8(4S), 186-191.

[2]Purohit, S., Thirkateh, A., & Nair, P. University Management and Scheduling Application.

[3] https://www.studocu.com/row/document/nahda-university/computer-science/srs-for-university-management-system/27752841

**1.5 Overview**

The University Management System (UMS) is a comprehensive platform that aims to simplify various university tasks. The system offers features for login and authentication, ensuring secure access to the system. With student management tools, the UMS help students register for courses and manages their academic records. Course management functionalities enable departments to add, modify, and remove course information. Faculty management features help professors communicate with students and manage their courses, including grading and billing. Enrollment management functionalities facilitate the enrollment process and help manage waiting lists. With billing and fee management, the UMS help manage student payments and fees. Academic record management features enable faculty to manage students' academic records, including grading, attendance, and course credits. Reporting and analytics functionalities help the university make data-driven decisions by providing insights into enrollment, course performance, and student outcomes. Overall, the UMS help streamline university operations and enhance the overall student and faculty experience. Additionally, the system will have top-level admin functionality to manage all the departments, professors, and students. The UMS functionalities may be updated based on feasibility and necessity.

**1.6 Intendent Audience**

The University Management System (UMS) caters to all stakeholders involved in running a robust learning community. These include senior-level management charged with ensuring smooth running campus operations; tenured professors working tirelessly towards upholding academic excellence; junior faculty and adjuncts contributing immensely towards shaping tomorrow's workforce; as well as students and support staff all playing integral parts towards making this a success story.

**2. Overall Description**

2.1 Product Perspective

The University Management System (UMS) is a software application that aims to streamline and automate various aspects of university management, such as course scheduling, enrollment management, billing and fee management, academic record management, reporting, and analytics. The system will be developed as an independent module that can integrate with existing university systems. The UMS will be developed using modern programming technologies, and it will be scalable, adaptable, and user-friendly.

2.2 Product Functions

The UMS will include the following core functions: login and authentication, student management, course management, faculty management, enrollment management, billing and fee management, academic record management, reporting, and analytics. These functions will help to simplify various university processes and will enable professors, students, and administrators to work more efficiently.

2.3 User Characteristics

The UMS is designed to meet the needs of various users, including professors, students, department heads, and administrators. The system will be user-friendly, with a simple and intuitive user interface that can be accessed from any device, including smartphones, tablets, and desktops.

2.4 General Constraints

The University Management System (UMS) is subject to certain constraints that must be taken into consideration during the development process. These constraints include the availability of hardware and software resources, budgetary constraints, and time constraints.

2.5 Assumptions and Dependencies

The University Management System (UMS) is dependent on a number of factors and assumptions. These include the assumption that the user has basic computer literacy and can access the system through a reliable internet connection. The system also assumes that the user data will be accurate and up-to-date, and that the system will be able to integrate with existing university systems. It also assumed that the University provides a member\_id to each person belonging to the University to register them into the system.

**3. Specific Requirements**

**3.1 External Interface Requirements**

3.3.1 User Interface: The system should provide a user-friendly interface for both students and professors. The interface should be intuitive and easy to use, and it should have clear navigation options.

3.3.2 Hardware Interface: The system should be compatible with different hardware devices such as computers, laptops, tablets, and mobile devices. The system should also support various operating systems such as Windows, MacOS, and Linux.

3.3.3 Software Interface: The system should be compatible with other software systems used in the university such as Learning Management Systems (LMS), Student Information Systems (SIS), and Human Resource Management Systems (HRMS).

3.3.4 Communication Interface: The system should support different communication protocols such as email, SMS, and push notifications. It should also have the ability to integrate with messaging platforms such as WhatsApp and Telegram.

**3.2 Functional Requirements**

**3.2.1 User Registration**

3.2.1.1 Purpose : To get login access to the system.

3.2.1.2 Actors : Users

3.2.1.3 Pre-Conditions : Must have a member\_id .

3.2.1.4 Inputs:

|  |  |  |  |
| --- | --- | --- | --- |
| Input Element | Input Type | Optional | Rules |
| Member\_id | Alphanumeric | Required | That one provided by the University. |
| Name | Alphabetic | Required | N/A |
| D.O.B | Numeric | Required | The User should be more than 18 years old at the time of registering. |
| Email | Alphanumeric | Required | Should be unique |
| Phone Number | Numeric | Optional | Should be unique |

3.2.1.5 Basic paths :

* User Click on the Register button
* Filled the information like Member\_id, Name , Date of Birth , Email, Phone Number etc.
* System will check the filled information.
* If all the information were correct , the system will register the user to the system .

3.2.1.6 Output: System will show the message that “User X successfully registered to the system. ”

3.2.1.7 Post Condition : User is eligible to log onto the system.

3.2.1.8 ID : FR0

3.2.1.9 DEP: None

**3.2.2 Login and authentication**

3.2.2.1 Purpose: To ensure secure and authorized access to the system for registered users.

3.2.2.2 Actors : Administrators , Students and system developers.

3.2.2.3 Pre-Conditions : Actor must have Register in the system.

3.2.2.4 Inputs:

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Element** | **Input Type** | **Optional / Required** | **Rules** |
| User ID | String | Required | Must have registered username |
| Password | String | Required | The password should match to the actual password. |

3.2.1.5 Basic Path :

* User click on the login button .
* User entered his/her login credentials .
* System verifies the credentials and if correct then let the user log into the system and if wrong then ask the user to reenter the login credentials.
* After Login System will show the administrators administrator dashboard where then not only can view but update information but from student dashboard student can only view the information.

3.2.1.6 Output: The system shows respective dashboard to the users .

3.2.1.7: Post-Condition : The User can access their respective dashboard.

3.2.1.8 ID: FR1

3.2.1.9 DEP: FR0

**3.2.3 Student Management**

**3.2.3.1 Course Registration**

3.2.3.1.1 Purpose: The system shall provide a platform for students to register for courses at the start of each semester.

3.2.3.1.2 Actors : Student

3.2.3.1.3 Pre-Conditions : Actor must have logged in to system.

3.2.3.1.4 Inputs:

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Element** | **Input Type** | **Optional / Required** | **Rules** |
| Course\_ID | Alphanumeric | Required | Course\_ID should be available in the system. |
| Student\_ID | Alphanumeric | Required | User\_ID should be available in the system. |
| SemNo | Numeric | Required | Should be between 1-8 . |
| Dept\_Id | Alphanumeric | Required | Dept\_ID should available in the system. |

3.2.3.1.5 Basic Path :

* User click on the “Select Course” button .
* User entered his/her Student\_ID.
* User entered his/her Course\_ID.
* User entered his/her SemNo.
* User entered his/her Dept\_ID.
* System verifies the credentials and if correct then let the student enrolled for the course.
* System will show the list of courses the student has enrolled .

3.2.3.1.6 Output: The system shows a message “Student X has successfully registered to course Y “ .

3.2.3.1.7: Post-Condition :The Student can see himself enrolled in the course.

3.2.3.1.8 ID: FR31

3.2.3.1.9 DEP:FR1

**3.2.3.2 Generate Student ID Card**

3.2.3.2.1 Purpose: To provide students with an official identification card that can be used for various purposes such as accessing facilities, borrowing books, and sitting for exams.

3.2.3.2.2 Actors : Student

3.2.3.2.3 Pre-Conditions : Actor must have logged in to system.

3.2.3.2.4 Inputs:

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Element** | **Input Type** | **Optional / Required** | **Rules** |
| Member\_ID | Alphanumeric | Required | Should match with the id that university has provided. |
| Student\_ID | Alphanumeric | Required | User\_ID should be available in the system. |
| Degree | Alphabetic | Required | Degree should be provided by the university. |
| Dept\_name | Alphabetic | Required | Dept\_name should available in the system. |
| Photograph | .jpg , .jpeg, .png | Required | Image should be passport size photo , size within 200kb. |
| Phone number | Numeric | Required | The ph no. should not be more than 10 digits. |

3.2.3.2.5 Basic Path :

* User click on the “Generate ID C” button .
* User entered his/her member\_ID.
* User entered his/her Student\_ID.
* User entered his/her Degree
* User entered his/her Dept\_name.
* User Upload his/her photograph.
* User entered his phone number.

3.2.3.2.6 Output: The System provides a downloadable Pdf of Student`s ID CARD.

3.2.3.2.7: Post-Condition :The PDF should be printable.

3.2.3.2.8 ID: FR32

3.2.3.2.9 DEP:FR1

**3.2.4 Course Management**

**3.2.4.1 Add a course**

3.2.4.1.1 Purpose: To enable the administrator to add new courses to the system and make them available for students to register.

3.2.4.1.2 Actors : Administrators

3.2.4.1.3 Pre-Conditions : Administrator login to the system and goes to the Course Management tab.

3.2.4.1.4 Inputs:

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Element** | **Input Type** | **Optional / Required** | **Rules** |
| Course\_ID | Alphanumeric | Required | The Id should follow the format (ex. CSMC102) |
| Course\_name | Alphabetic | Required | Course name should be a valid name. |

3.2.4.1.5 Basic Path :

* Administrator logins into the system and goes to Course management
* User clicked on the “ Add Course “ button
* Enter the course\_id
* Enter the course\_name.
* Press on the “ADD” button.

3.2.4.1.6 Output: The system shows updated list of all course provide by the university.

3.2.4.1.7: Post-Condition :The Student also can see the updated list of courses.

3.2.4.1.8 ID: FR41

3.2.4.1.9 DEP: FR1

**3.2.4.2 Assign Instructor to Course**

3.2.4.2.1 Purpose: To enable the administrator to assign an instructor who is qualified to teach a particular course.

3.2.4.2.2 Actors : Administrators

3.2.4.2.3 Pre-Conditions : Administrator login to the system and goes to the Course Management tab.

3.2.4.2.4 Inputs:

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Element** | **Input Type** | **Optional / Required** | **Rules** |
| Course\_ID | Alphanumeric | Required | The Id should follow the format (ex. CSMC102) |
| Faculty\_id | Alphabetic | Required | Faculty\_id should be available in the system. |
| Faculty\_name | Alphabetic | Required |  |

3.2.4.2.5 Basic Path :

* Administrator logins into the system and goes to Course management
* User clicked on the “ Add Course Instructor “ button
* Enter the course\_id
* Enter the Faculty\_id.
* Enter Faculty Name.
* Press on the “ADD” button.

3.2.4.2.6 Output: The system shows updated list of all assign instructors to a course provide by the university.

3.2.4.2.7: Post-Condition : There could be more than one instructor assigned to a course.

3.2.4.2.8 ID: FR42

3.2.4.2.9 DEP:FR1

**3.2.5 Faculty Management**

**3.2.5.1 Faculty profile**

3.2.5.1.1 Purpose: To ensure that the faculty member's information is up-to-date and accurate.

3.2.5.1.2 Actors : Administrators , Faculty Members

3.2.5.1.3 Pre-Conditions : Administrator/Faculty login to the system and goes to the Faculty Management tab.

3.2.4.1.4 Inputs:

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Element** | **Input Type** | **Optional / Required** | **Rules** |
| Update\_Name | String | Optional | * Must contain name and title, optionally middle name * must not contain numbers or special characters * must be between 3 to 50 letters |
| Update\_Photograph | .jpg, .jpeg, .png | Optional | Image size should be within 100kb -200 kb |
| Update\_Contact | email | Optional | Should be a unique mail id |

3.2.5.1.5 Basic Path :

* Administrator/Faculty logins into the system and goes to Faculty management
* User clicked on the “ Update Profile “ button
* Update Fields if necessary .
* Press on “Save” button to update or on “Cancel” button.

3.2.5.1.6 Output: Updated Profile of the Faculties.

3.2.5.1.7: Post-Condition : A faculty can update his/her information as many time as they want.

3.2.5.1.8 ID: FR51

3.2.5.1.9 DEP:FR1

**3.3 Nonfunctional Requirements**

3.3.1 **Performance Requirements**

The system should be responsive and fast, and should not have any delays in processing user requests. The system should be able to handle a large number of users and requests simultaneously without any performance degradation. The maximum response time for the system should be less than 3 seconds.

3.3.2 **Design Requirements**

* The system shall be designed to be compatible with commonly used web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge.
* The system shall be designed to be responsive and adaptable to different screen sizes and resolutions.
* The system shall be designed using a modular approach to allow for easy maintenance and future upgrades.
* The system shall be designed to follow accessibility guidelines, such as the Web Content Accessibility Guidelines (WCAG) 2.1, to ensure that it is accessible to users with disabilities.
* The system shall be designed to have a user-friendly interface that is intuitive and easy to navigate.
* The system shall be designed to have efficient data processing and retrieval capabilities to ensure smooth and fast user experience.
* The system shall be designed to be scalable and adaptable to accommodate potential growth in user base and additional features.

**3.3.3 Quality Attribute**

3.3.3.1 Usability

The software system should be easy to learn and use for all types of users. It should have a user-friendly interface, clear and concise instructions, and a consistent design. The system should also have the ability to adapt to the user's preferences and provide feedback in case of errors or confusion.

3.3.3.2 Reliability

The software system should be able to perform its intended functions accurately and consistently, without any downtime or unexpected failures. It should also have the ability to recover from errors and maintain data integrity.

3.3.3.3 Security

The software system should ensure the confidentiality, integrity, and availability of data by implementing appropriate security measures such as access control, encryption, and authentication. It should also be able to detect and prevent any unauthorized access or data breaches.

**4. Other Requirements**

4.1 Legal and Regulatory Requirements

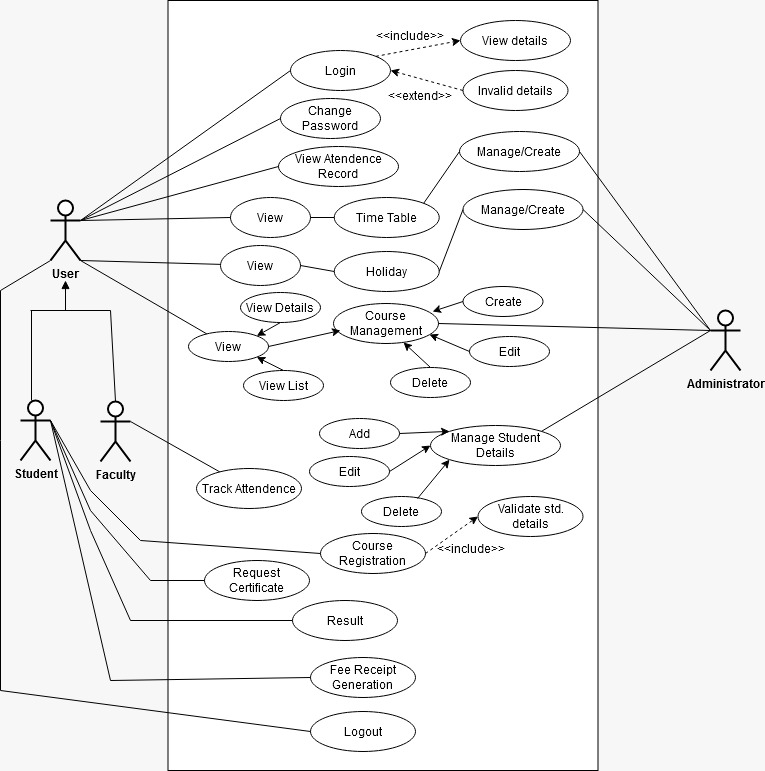
This section identifies any legal and regulatory requirements that the system must comply with, such as data privacy laws or accessibility standards.

4.2 Training and Supporting Requirements

This includes user training for the software and hardware, as well as technical support for system issues. The training and support must be provided by the vendor or the system provider to ensure efficient system usage and maintenance.

**Appendix A : Analysis Models**

**Use Case diagrams**

****