# Software Requirement Specification

## For

# Faculty Management System

Prepared by Group 4

ADARSH M S - 5
RINU JAUHAR - 58
SREEKANTH CS - 65
DEVANARAYANAN - 21

14 - MARCH - 2023

# Table of Contents

# Revision History

| Name | Date | Reason For Changes | Version |
|---|---|---|---|
| | | | |
| | | | |
| | | | |

# 1. Introduction

## 1.1 Purpose

The Faculty Management System is a software application that aims to streamline the administrative and academic activities of educational institutions. Many educational institutions struggle to maintain faculty data and its effective organizing. It becomes difficult when we require searching, sorting etc for many purposes like NAAC and NBA accreditations. To address these challenges, there is a need for a faculty management system that enables educational institutions to make an easy and efficient way for faculty management.

The scope of a faculty management system can vary depending on the specific needs and requirements of the organization or accrediting organization implementing it. However, the core goal is to automate the manual processes involved in managing faculty members.

## 1.2 Document Conventions

- Use Arial font for the entire report – Chapter/Section Title – Times New Roman 18, Bold; Heading 2 – Arial 14, Bold; Body- Arial 11, Normal.
- Line Spacing – Between Heading 2 – 3 lines, between lines in paragraph 1.5 lines.
- Alignments – Chapter/Section Title – Left, heading 2 & 3 should be Left Aligned. Ensure that all body text is paragraph justified.

## 1.3 Intended Audience and Reading Suggestions

Readers of this document include developers, project managers, testers, and documentation writers. Readers include the panel of professors and the team members consisting of developers and documentation writers. The document consists of a detailed description of all the functional and non-functional requirements and various perspectives of the product. It has all the features that are to be implemented in the product.

### 1.4 Product Scope

The product scope of faculty management system would include the following:

1. User Interface: The system needs to have a user-friendly interface that allows users to add and retrieve data easily. This could be in the form of a web or mobile application.
2. Authentication Mechanism: A secure authentication mechanism is needed to ensure that only verified faculties and other members have access to our website. This could be achieved through a variety of mechanisms including university-issued identity verification, or other secure methods.
3. Faculty profile management: This feature allows the institution to manage the profiles of all its faculty members, including their personal information, educational qualifications, research interests, and contact details.
4. Course and timetable management: This feature allows the institution to manage course schedules, class timings, and faculty assignments for each course.
5. Attendance and performance tracking: This feature allows the institution to track the attendance and performance of each faculty member, including their teaching evaluations, research outputs, and other metrics.
6. Leave and absence management: This feature allows the institution to manage the leave and absence requests of faculty members, including sick leave, vacation time, and personal days.
7. Administration Panel: The system needs to have an administration panel that allows authorized personnel to manage the other user's activities like login and all. setting up login credentials for the users.
8. Professional development: This feature allows the institution to track the professional development of each faculty member, including their attendance at conferences, workshops, and other events.
9. Accreditation Evaluation: This feature will allow the accrediting board to evaluate various parameters required for the accreditation process.

### 1.5 References

## 2. Overall Description

### 2.1 Product Perspective

It is not a new system. It is an extended version of our existing products like etlab and other faculty management systems. The product perspective of a faculty management system would focus on the design, development, and functionality of the product itself. This would include considerations such as the database and technology used in the product, the user interface and experience, and the security features in place to protect against tampering and hacking.

From a product perspective, an faculty management system would need to be designed to meet a range of technical requirements, such as the ability to allow faculty members to create and manage their profiles, including personal information, educational qualifications, research interests, publications, and professional affiliations, should provide tools for scheduling courses, assigning faculty members to courses, and managing course-related information, such as course descriptions, syllabus, and grading policies, tools for tracking attendance, grades, and other performance metrics for individual students and entire classes.

Overall, the product perspective of a faculty management system would need to balance the need for efficiency, accessibility, and ease of use and accuracy. This would require careful design, rigorous testing, and ongoing monitoring and maintenance to ensure that the product is both reliable and trustworthy.

## 2.2 Product Functions

The primary function of the faculty management system is to enable the faculties to add their data into the database in a structured manner, and to accurately retrieve those data in the required manner. To achieve this, a faculty management system typically includes several key product functions, such as:

- User Interface: A faculty management system must provide an intuitive and user- friendly interface that allows users to add and display the data in a standard manner. This might include navigation buttons and separate pages for each section of data.

- Faculty profile management: The system should allow faculty members to create and manage their profiles, including personal information, educational qualifications, research interests, publications, and professional affiliations.

- Course scheduling and management: The system should provide tools for scheduling courses, and managing course-related information, such as course descriptions, syllabus, and grading policies.

- Attendance and grade tracking: The system should provide tools for tracking attendance, grades, and other performance metrics for individual students and entire classes.

- Notifications: The system must be able to manage and display notifications and announcements related to academics.

Overall, a faculty management system should be user-friendly, flexible, and customizable to meet the unique needs of each institution. It should also be scalable and secure to handle the growing demands of modern education.

## 2.3 User Classes and Characteristics

There are several user classes and characteristics of an faculty management system, including:

- Faculty Members: This user class includes all the teaching staff of the institution, such as professors, lecturers, and assistants. They have access to features like course management, attendance tracking, grade management, and communication tools.

- Administrators: This user class includes the institutional management team, such as department heads, deans, and academic directors. They have access to features like faculty management, course scheduling, and analytics and reporting.

- Admin:There is an admin for a specific university who can control the login and access of the users of our product.

- Accreditation Body: They have special access to our system who can access the data of specific departments and they can analyse the whole activity and performance of the department in an efficient manner.

Overall, the characteristics and needs of each user class will vary depending on their role in the university. Designing a faculty management system that is accessible, secure, and user- friendly for all user classes is an important consideration in ensuring the goal of implementing automated faculty management system.

## 2.4 Operating Environment

The operating environment of a faculty management system refers to the physical and technical conditions in which the product is designed to operate. Some key factors that can impact the operating environment of a faculty management system:

- Hardware Requirements: The system should be able to run on a variety of hardware configurations, including desktop computers, laptops, tablets, and smartphones. It should also support different operating systems, such as Windows, macOS, Linux, iOS, and Android.

- Network Infrastructure: The system should be accessible over a network, such as a local area network (LAN), wide area network (WAN), or the internet. It should also support different network protocols, such as HTTP, HTTPS, TCP/IP, and SSL.

- Technology Stack: The system may use a combination of programming languages, frameworks, libraries, and databases to implement its features. Our product uses a web-based architecture with HTML, CSS, and JavaScript for the front-end and Node js for the back end. It also uses a relational database management system (RDBMS) MySQL Server to store and manage data.

- Software Dependencies: The system may depend on various third-party software and services, such as web servers, application servers, email servers, messaging services, authentication and authorization services, and analytics and reporting tools. These dependencies should be carefully managed to ensure compatibility, security, and performance.

- Security Considerations: The system should be designed with security in mind, including authentication and authorization mechanisms, data encryption, access control policies, and backup and recovery procedures.

- User Accessibility: The operating environment must be accessible to all users who have valid login id and password.

Overall, the operating environment of a faculty management system should be reliable, scalable, and secure, to support the needs of educational institutions and their faculty members.

## 2.5 Design and Implementation Constraints

There are several design and implementation constraints that must be considered when designing and implementing a faculty management system. Some of the most important constraints include:

- Security: One of the most important design constraints for faculty management system is security. The software must be designed to prevent unauthorized access, tampering, or hacking, and must incorporate robust cybersecurity measures to protect against data breaches or other security threats.

- Accessibility: Another important design constraint is accessibility. The service must be designed to accommodate all faculties in a college.

- Cost: Cost is another important constraint that must be considered when designing and implementing a faculty management system. The software must be cost-effective to manufacture and deploy, while still meeting the necessary security and accessibility requirements.

- Technical Complexity: Faculty management systems are complex systems that require a high level of technical expertise to design, implement, and maintain. The design must strike a balance between complexity and usability, ensuring that the software can be operated and maintained effectively without sacrificing security or functionality.

- Legal and Regulatory Compliance: Faculty management system must comply with a range of legal and regulatory requirements, including UG guidelines, data privacy laws, and cybersecurity regulations. The design and implementation must ensure that the software meets all necessary compliance requirements while still providing a secure and accessible voting platform.

- Infrastructure: Faculty management system require a certain level of infrastructure to operate effectively, including server-side service, network connectivity, and data storage and processing capabilities. The design and implementation must take into account the availability and reliability of these infrastructure components, as well as any associated costs or logistical considerations.

Overall, designing and implementing a Faculty management system that meets these constraints requires a careful balance of technical expertise, user-cantered design principles, and compliance with legal and regulatory requirements.

## 2.6 User Documentation

User documentation for a Faculty management system is an important aspect of ensuring that faculty can understand and use the software effectively. Some key elements that should be included in the user documentation for a Faculty management system include

- Overview and Purpose: The user documentation should provide an overview of the purpose and functionality of the Faculty management system
- Setup and Installation: The user documentation should provide clear and concise instructions for setting up faculty profiles, including any necessary data required.
- User Interface: The user documentation should provide a detailed description of the user interface for the website, including how to navigate menus, select options, and change/update data.
- Accessibility: The user documentation should provide information about any accessibility features of the website, including options for faculties of roles.
- Troubleshooting: The user documentation should include troubleshooting information, including a list of common issues and how to resolve them, as well as information about how to contact technical support if needed.
- Security: The user documentation should provide information about the security measures in place to protect the integrity and confidentiality of the data management, including any measures to prevent unauthorized access or tampering.
- Privacy: The user documentation should provide information about how the Faculty management system handles faculty data and any measures in place to protect faculty privacy.
- Training demo tour: The user documentation should include a training demo tour for faculty and other staff, including detailed instructions for setting up and operating the website, as well as any relevant legal or regulatory information.

Overall, the user documentation for a Faculty management system should be clear, concise, and easy to understand, with a focus on providing the information necessary for voters to use the software effectively and securely.

## 2.7 Assumptions and Dependencies

Assumptions and dependencies are important considerations when designing and implementing a Faculty management system. Some key assumptions and dependencies that may impact the functionality and usability of a Faculty management system include:

- Faculty Behavior: The design and functionality of a Faculty management system assumes that faculty will follow the proper operating procedures and use the website as intended. Any deviations from expected faculty behavior could impact the accuracy and integrity of the Faculty management system.

- Infrastructure Availability: Faculty management system rely on a range of infrastructure components, including server support, network connectivity, and data storage and processing capabilities. Any issues with these components, such as server or network failures, could impact the ability of the Faculty management system to operate effectively.
- Technical Expertise: The design and implementation of a Faculty management system require a high level of technical expertise, both in terms of front-end and back-end development and cybersecurity. Any issues with technical expertise could impact the security and functionality of the website.
- Legal and Regulatory Compliance: Faculty management system must comply with a range of legal and regulatory requirements, including UG regulations, data privacy laws, and cybersecurity regulations. Any changes to these requirements could impact the functionality and usability of the website.
- Accessibility Requirements: Faculty management system must be designed to accommodate all faculties in a college. Any issues with accessibility could impact the ability of some voters to use the machine.
effectively
- Trust in the System: The effectiveness of a Faculty management system relies on the trust of faculty and other staff in the security and efficiency of the system. Any issues or concerns about the security or efficiency of the machine could impact the trust and confidence of faculty in the Faculty management system.

Overall, assumptions and dependencies play an important role in the design and implementation of a Faculty management system, and should be carefully considered to ensure that the website is functional, secure, and accessible for all faculty

# 3. External Interface Requirements

## 3.1 User Interfaces

The login page will have the following features:

- Options for selecting role
- Label for username and password
- Forgot password option

For each profile, its username and initial passwords are created by admins of colleges. After login the faculty/staff can change the password.

The home page will have the following features

- Photo and personal details
- Options for changing details
- Hyperlinks for Professional experience, Teaching, Personal, Research & Examination/Valuation

## 3.2 Software Interfaces

- React is used for building user interfaces.
- Node.js allows to run JavaScript code outside of a web browser.
- CSS is used to style web pages.
- SQL database is used to store relational data

# 4. System features

This illustrates the functionalities and capabilities the system must possess to meet the requirements of its users. These features typically describes the actions that the system must be able to perform and the data it must process.

## 4.1 Faculty profile management

The system should allow the faculty members to create and manage their profiles, including educational, professional, personal, awards, publications and research data.

## 4.2 Attendance management

The system should be able to track the attendance of the faculty members. The attendance module should be easily accessible by both the faculty and administrators with a user-friendly interface.

### 4.3 Report generation

The system should be able to automatically generate the document based on NBA criteria 5 by using the faculty data previously obtained. The document must be available in a printable pdf form which can be downloaded.

### 4.4 Faculty document management

The system will be able to store various documents like appointment letters, degree certificates, promotion letters etc as proofs of the data submitted. The document will be saved in a cloud-based file system with the security provided by the AWS S3.

### 4.5 Security and access control

The system will implement proper security by providing role-based access control. Different roles like faculty, admin, etc have access to only the data and functionalities they are allowed. The roles of the user are checked properly while using the system. The database security is provided by using proper hashing algorithms to store the user credentials and private information.

## 5. Other Non-functional Requirements

### 5.1 Performance Requirements

The system should have a response time of less than 2 seconds for all operations.
The system should be able to handle a minimum of 100 concurrent users.

### 5.2 Security Requirements

The system should use encryption to protect the faculty members personal information. The system should have role-based access control to restrict access to sensitive data.

### 5.3 Software Quality Attributes

- Reliability: The system should be reliable, ensuring that it operates correctly and consistently, and can handle a high volume of transactions.
- Scalability: The system should be scalable, allowing it to handle an increasing number of users and transactions without compromising performance.
- Usability: The system should be user-friendly, easy to use, and accessible to all eligible faculties, regardless of technical expertise or disabilities.
- Maintainability: The system should be maintainable, allowing for efficient and effective maintenance and updates to the system as needed.
- Interoperability: The system should be interoperable, allowing it to communicate and exchange data with other systems and platforms.
- Performance: The system should perform well, ensuring fast response times and minimal latency, even under high traffic conditions.
- Security: The system should be secure, protecting against unauthorized access, tampering, or other malicious activities.

# 6. Other requirements

## 6.1 Database schema

**Personal**

| | |
|---|---|
| PEN | string |
| university_id | string |
| name | string |
| dob | date |
| staff_type | string |
| email_personal | string |
| email_official | string |
| do_join | date |
| address | string |
| dept | string |
| mobile | integer |
| aadhar | integer |

**Education**

| | |
|---|---|
| degree | string |
| branch | string |

| | |
|---|---|
| **specialization** | **string** |
| **university** | **string** |
| **date_of_acquiring** | **date** |
| **marks** | **number** |

### Professional

| | |
|---|---|
| **type** | **string** |
| **institution** | **string** |
| **designation** | **string** |
| **role** | **string** |
| **start_date** | **date** |
| **end_date** | **date** |

### Publications

| | |
|---|---|
| **title** | **string** |
| **type** | **string** |
| **name_of_journal** | **string** |
| **publisher_name** | **string** |
| **issn** | **integer** |
| **index** | **integer** |

| | |
|---|---|
| **impact_factor** | **string** |