**Faculty Management System**

Use Case (Requirement) Document

2nd Jan 2025

Ver0.0Draft1

Document Identification

| Project Code |  |
| --- | --- |
| Project Name | Faculty Management System |
| Document Id |  |
| Project Start Date | 09-01-2024 |
| Project End Date | 10-01-2025 |
| Project Type |  |
| Project Manager | Mr. Nilesh Kumar |
| Project Manager Cell No |  |

Client Detail

| Client Name |  |
| --- | --- |
| Contact Person |  |
| Contact Person Email Id |  |
| Contact Person Cell No |  |

Version History

| Version | Date | Author(s) | Comment | Reviewed By |
| --- | --- | --- | --- | --- |
| Ver0.0Draft1 | 2nd Jan, 2025 | Nishant Jha | Initially Draft. |  |

Contents

[1 Introduction 4](#_Toc186811700)

[1.0 Purpose 4](#_Toc186811701)

[1.1 Scope 4](#_Toc186811702)

[1.2 Target Audience 4](#_Toc186811703)

[1.3 Document Management 4](#_Toc186811704)

[1.4 Reference Document 4](#_Toc186811705)

[2 Overview 5](#_Toc186811706)

[2.0 Assumptions 5](#_Toc186811707)

[2.1 Constraints 5](#_Toc186811708)

[2.2 Technologies 5](#_Toc186811709)

[2.3 Exclusions 5](#_Toc186811710)

[3 Use Cases 6](#_Toc186811711)

[3.0 UC-000: High Level Use Case Association 6](#_Toc186811712)

[3.1 UC-001: View User 7](#_Toc186811713)

[3.2 UC-002: Add User 9](#_Toc186811714)

[3.3 UC-003: Update User 10](#_Toc186811715)

[3.4 UC-004: Delete User 12](#_Toc186811716)

[4 Processing Details 13](#_Toc186811717)

[4.1 PD01: Faculty Management System 13](#_Toc186811718)

[4.1.1 PD01A: Add User 13](#_Toc186811719)

[4.1.2 PD01A: Update User 13](#_Toc186811720)

# Introduction

## Purpose

The purpose of this document is to provide insight into the use cases related to the “Faculty Management System”, which will be the baseline document for design, development and testing of the application.

## Scope

This document covers the business use case of “Faculty Management System Phase-1.0” and assumption which is used at the time of designs & development of project. In “Faculty Management System Phase”, following are new requirements/modules to be implemented:

1. Functionality to view a user details.
2. Functionality to add a user or multiple users.
3. Functionality to delete a user or multiple users.
4. Functionality to update a user or multiple users.

## Target Audience

The target audience for the Faculty Management System includes businesses, organizations, and HR departments aiming to streamline their Faculty management, payroll, and administrative processes.

## Document Management

The requirements in this use case document shall be traced to the appropriate deliverables in the development and testing phases to ensure that all requirements are properly implemented and tested.

# Overview

The Objective of Faculty Management System Phase-1.0 is to

1. Functionality to view a user details.
2. Functionality to add a user or multiple users.
3. Functionality to delete a user or multiple users.
4. Functionality to update a user or multiple users.

## Assumptions

1. There will be only one admin.
2. Each user will have a unique identifier, that is the Faculty ID.

## Constraints

* Java 17 will be used as the baseline version for development.
* This system does not support multi-language interface.

## Technologies

**Languages & Tool:** Escort Designer,Java v17, Spring Boot Framework, Vue 3, and JavaScript

**Operating Systems/Platform:** Windows 11, Microsoft Edge

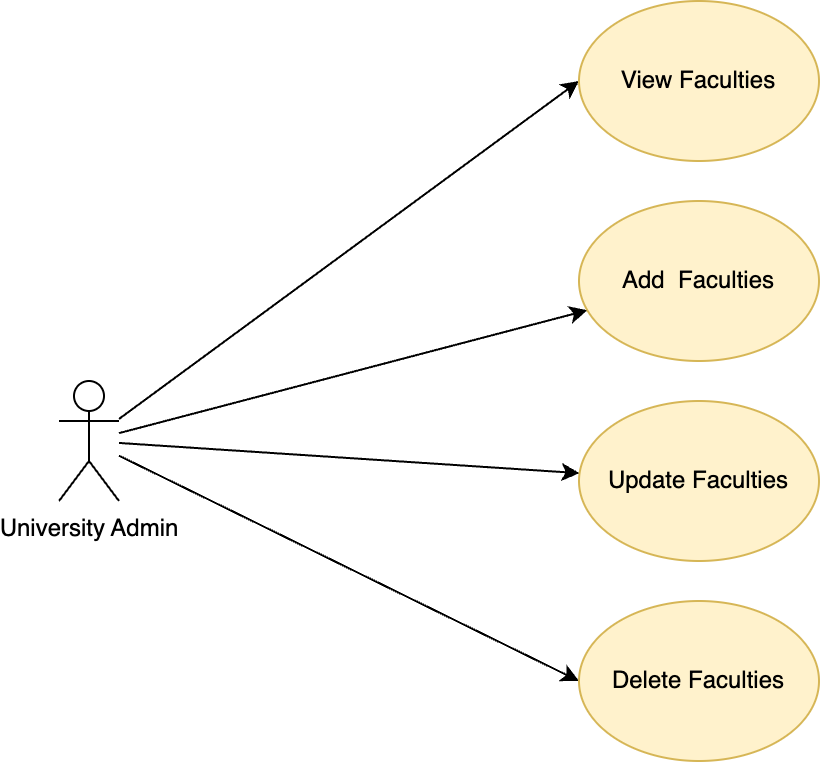
**Device Databases:** PostgreSQL

## Exclusions

* The execution of the project would be as stated in the above sections. Any other aspect or deviation introduced during the project development will be considered as Out of Scope or Change request and would call for impact analysis in terms of added timeframe and additional charges.

# Use Cases

## UC-000: High Level Use Case Association



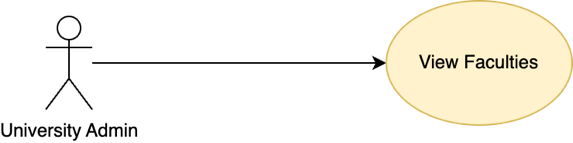
## UC-001: View Faculties

**Purpose**

* + - This use case describes how the user can visit the dashboard page of the Faculty Management System.

**Actors**

* + - The actor(s) that interact and participate in this use case is Admin.



**Pre-Conditions**

* + - The Faculty Management application should be running and accessible.

**Post-Conditions**

* + - A list of Faculties is displayed to the admin.

**Basic Flow**

* + - The admin accesses the Faculty Management System.
    - The system displays an editable table with the following sections:

1. **Faculty ID:** A mandatory field to input a unique Faculty ID.
2. **Faculty Name:** A mandatory field to input or modify the Faculty's name.
3. **Profile Image:** Displays the Faculty's profile image, allowing the admin to upload a new one.
4. **Age:** A mandatory field to input or update the Faculty's age.
5. **Contact Number:** A mandatory field to input or update the Faculty's contact number.
6. **Email:** A mandatory field to input or modify the Faculty's email address.
7. **Qualifications:** Editable field to input or modify the Faculty's qualifications.
8. **Courses:** A mandatory field to input or modify the courses taught by the Faculty.
   * + The system displays an editable table with the following sections:
9. **Edit any field inline** (e.g., Faculty Name, Age, Contact Number, etc.) and save changes using the **Save** button.
10. **Add a new Faculty** by clicking the **Add** button, which creates a new, editable row in the table.
11. **Delete selected Faculty(s)** by clicking the **Delete** button.
12. **Upload a profile image** for a specific Faculty by clicking the **Upload** button in that row and selecting an image file.
    * + Pagination options allow the admin to navigate between pages of Faculty data, and a search bar or filters enable quick access to specific records.

**Failure Case**

* The table may fail to load if there is a loss of connection to the database, resulting in no data being retrieved.
* Large or unsupported image formats might cause loading delays or broken images.

**Screen Design**

A screenshot of a computer

Description automatically generated

**Database Tables Involved**

* Facultys\_master

## UC-002: Add Faculties

**Purpose**

* + - This use case describes how the admin can add one or more new Faculties to the Faculty Management System by clicking the **New button**, editing the Faculty details in an editable row, and then saving the changes.

**Actors**

* + - The actor that interacts and participates in this use case is: Admin.



**Pre-Conditions**

* + - The Faculty Management application should be running and accessible.
    - The admin should have the necessary permissions to add users.

**Post-Conditions**

* + - New Faculty(s) are added to the system and displayed in the table.

**Basic Flow**

* + - Admin Accesses the Faculty Management System: The admin is on the Faculty Management System dashboard.
    - The system displays an editable table with the following sections:

1. The admin clicks the Add button to create a new user.
2. Clicking the button creates one or more new rows in the editable table, depending on how many times the admin clicks the **New button**.
   * + **Editing the New Row(s):** Each new row is empty by default, allowing the admin to input data for each Faculty. The admin can edit any field in the row(s) as necessary.
     + After editing, the admin clicks the **Save button** to save the new Faculty(s) that are selected.
     + The system validates the entered data for correctness (e.g., required fields, correct formats). If the data is valid, the system sends the new Faculty(s) to the backend for storage in the database.
     + After the new data is saved, the system retrieves the updated list of Facultys, including the new Faculty(s). The table dynamically updates to show the newly added Facultys.

**Failure Case**

* If the system cannot connect to the backend or database, the new data will not be saved, and an error message will be shown.
* After saving, if the system cannot retrieve the updated list of Facultys due to backend or database issues, the table will not update with the new Faculty(s).

**Screen Design**

A screenshot of a computer

Description automatically generated

**Database Tables Involved**

* Facultys\_master

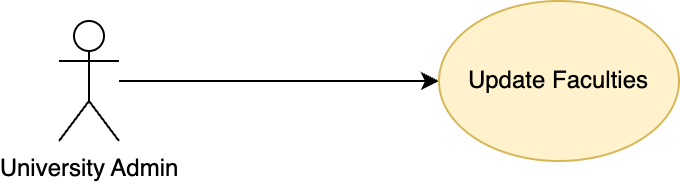
## UC-003: Update Faculties

**Purpose**

* + - This use case describes how the admin can update the details of an existing Faculty in the Faculty Management System by editing the values in the table and saving the changes.

**Actors**

* + - The actor involved in this use case is Admin



**Pre-Conditions**

* + - The Faculty Management application should be running and accessible.
    - The admin should have the necessary permissions to update user details.
    - The Faculty records to be updated should already exist in the system.

**Post-Conditions**

* + - The updated Faculty details are saved in the system and reflected in the table.

**Basic Flow**

* + - **Admin Accesses the Faculty Management System**: The admin accesses the dashboard where Faculty records are displayed.
    - **Editing an Existing Faculty Record**:

1. The admin selects an existing Faculty’s row from the table.
2. The system displays the current details of the Facultys in the editable fields.
3. The admin can update the fields for the selected Faculty.
   * + **Saving the changes:** After editing the values, the admin clicks the **Save button** to save the updated Faculty details. The system validates the edited data to ensure correctness (e.g., required fields, correct formats).
     + **Data gets updated and retrieved**: If the data is valid, the updated Faculty information is sent to the backend and saved in the database. After the new data is saved, the system retrieves the updated list of Facultys, including the new Faculty(s). The table dynamically updates to show the newly added Facultys.

**Failure Case**

* **Database Error:** If the system cannot connect to the backend or database, the new data will not be saved, and an error message will be shown.
* After updating, if the system cannot retrieve the updated list of Facultys due to backend or database issues, the table will not update with the new Faculty(s).

**Screen Design**

A screenshot of a computer

Description automatically generated

**Database Tables Involved**

* + - NA

## UC-004: Delete Faculties

**Purpose**

* + - This use case describes how the admin can delete one or more existing Faculty records from the Faculty Management System by selecting rows with checkboxes and clicking the **Delete button**.

**Actors**

* + - The actor involved in this use case is: Admin



**Pre-Conditions**

* + - The Faculty Management application should be running and accessible.

**Post-Conditions**

* + - Selected Faculty(s) are removed from the system, and the updated list is displayed in the table.

**Basic Flow**

* + - **Admin Accesses the Faculty Management System**: The admin accesses the dashboard where Faculty records are displayed.
    - **Selecting Facultys for Deletion**: The admin selects one or more rows in the table by checking the checkboxes next to each Faculty record to be deleted.
    - After selecting the rows, the admin clicks the **Delete button** to initiate the deletion process.
    - The system sends a request to the backend to delete the selected Faculty(s) from the database. If the deletion is successful, the selected rows are removed from the table.
    - After successful deletion, the system retrieves the updated list of Facultys from the backend. The table dynamically updates to reflect the deletion, showing only the remaining Faculty records.

**Failure Case**

* **Database Error:** If there is an error in the database, the deletion process will fail.
* **Network Error:** If there is a network error, the deletion process might not be completed successfully may fail due to server issues or validation errors.

**Screen Design**

A screenshot of a computer

Description automatically generated

**Database Tables Involved**

* + - NA

# Processing Details

## PD01: Faculty Management System

### PD01A: Add User

* + - Each field in the user row (name, team, etc.) must be checked for completeness.
    - Username must be unique across the system to avoid confusion and ensure proper identification.
    - Username must be a minimum of 5 characters and maximum of 28 characters.
    - Username must not contain prohibited characters or patterns, such as spaces, slashes, or any other special characters other than underscore and hyphen.

### PD01A: Update User

* + - Validate each updated field for correctness (e.g., name must not be empty, DOB must be a valid date, etc.).
    - Ensure any updated username adheres to the same rules as for adding a new user.
    - Username must be unique across the system to avoid confusion and ensure proper identification.
    - Username must be a minimum of 5 characters and maximum of 28 characters.
    - Username must not contain prohibited characters or patterns, such as spaces, slashes, or any other special characters other than underscore and hyphen.