**COMSATS University Islamabad,**

**Abbottabad Campus**

**SOFTWARE DESIGN DOCUMENT  
(SDD)**

**for**

**<University Attendance System>**  
Version 1.0

***By***

**Fawad Iqbal CIIT/FA21-BSE-012/ATD**

**Faizan CIIT/FA21-BSE-011/ATD**

***Supervisor*Prof. Mukhtiar Zamin**

***Bachelor of Science in Software Engineering (2021-2025)***

**Table of Contents**

[Chapter 1 Project Proposal 6](#_Toc139999988)

[1.1 Introduction 6](#_Toc139999989)

[1.2 Vision 6](#_Toc139999990)

[1.3 Business Case 7](#_Toc139999991)

[1.4 Functional Requirements 7](#_Toc139999992)

[1.5 Supplementary Specification 9](#_Toc139999993)

[1.6 Glossary 11](#_Toc139999994)

[1.7 Risk List & Risk Management Plan 12](#_Toc139999995)

[Chapter 2 Use Cases 13](#_Toc139999996)

[2.1 Use Case Diagram 13](#_Toc139999997)

[2.2 Use Case Distribution 13](#_Toc139999998)

[2.3 Brief Level Use Case 13](#_Toc139999999)

[2.4 Fully Dressed Use Cases 13](#_Toc140000000)

[Chapter 3 UI Prototypes 14](#_Toc140000001)

[3.1 Login UI 14](#_Toc140000002)

[3.2 Dashboard 14](#_Toc140000003)

[3.3 Add Attendance 15](#_Toc140000004)

[Chapter 3 Domain Model 15](#_Toc140000005)

[3.1 Domain Model 15](#_Toc140000006)

[Chapter 4 System Sequence Diagram (SSD) 16](#_Toc140000007)

[4.1 SSD 16](#_Toc140000008)

[Chapter 5 Operation Contract (OC’s) 16](#_Toc140000009)

[5.1 Operation Contract 16](#_Toc140000010)

[Chapter 6 Package Diagram 17](#_Toc140000011)

[6.1 Package Diagram 17](#_Toc140000012)

[Chapter 7 Class Diagram 18](#_Toc140000013)

[7.1 Class Diagram: 18](#_Toc140000014)

[Chapter 8 Interaction Diagram 19](#_Toc140000015)

[8.1 Interaction Diagram 19](#_Toc140000016)

[8.2 Sequence Diagram 19](#_Toc140000017)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **sName** | **Date** | **Reason for changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Application Evaluation History**

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*Include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |

**Supervised by**

**<Prof. Mukhtiar Zamin>**

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Chapter 1 Project Proposal

## 1.1 Introduction

Efficient management of attendance records is essential in the dynamic landscape of education. Traditional attendance tracking methods often involve laborious manual record-keeping and administrative efforts, leading to inefficiencies and potential errors. To address these challenges, this thesis presents a groundbreaking desktop application for a university attendance system. The application empowers teachers by providing them with a user-friendly interface to mark, modify, and delete attendance records, revolutionizing the way attendance is managed.

The core objective of this desktop application is to streamline the attendance management process within the university environment. By leveraging modern technology, the application simplifies and automates tasks that were once time-consuming and prone to mistakes. A key feature of the application is its fully functional login system, which ensures secure access, validates user credentials, and authenticates authorized users. This aspect guarantees the integrity and confidentiality of attendance data, providing peace of mind to both teachers and administrators.

With this application, teachers gain a powerful tool to mark attendance accurately and efficiently. They can easily modify attendance records to accommodate makeup classes, correct mistakes, or adapt to changing circumstances. The ability to delete attendance entries in certain situations adds flexibility and ensures accurate reporting. By digitizing the attendance tracking process, the application reduces the burden of manual paperwork and data entry, freeing up valuable time for teachers to focus on their core responsibilities of instruction and mentorship.

## 1.2 Vision

The vision for the desktop application for the university attendance system is to transform the way attendance management is conducted within educational institutions. By leveraging modern technology and intuitive design, the application aims to provide an efficient, accurate, and user-friendly solution for teachers and administrators.

The goal is to streamline attendance tracking processes, reduce administrative burdens, and enhance overall productivity within the academic environment.

## 1.3 Business Case

#### The development and implementation of the desktop application for the university attendance system presents several compelling business cases. These cases revolve around improving operational efficiency, enhancing data accuracy, and facilitating informed decision-making. The key business cases include:

#### Time and Cost Savings: The application reduces manual paperwork, streamlines attendance tracking, and automates administrative tasks. This leads to significant time and cost savings for educational institutions by minimizing the effort required for data entry, record-keeping, and report generation.

#### Improved Accuracy and Compliance: The application ensures accurate and reliable attendance data by eliminating the possibility of human errors inherent in manual processes. It also helps institutions meet regulatory requirements and maintain compliance by providing a systematic and auditable attendance management system.

#### Improved Student Experience: The application contributes to a positive student experience by ensuring accurate attendance records, minimizing disputes, and enabling prompt interventions when necessary. Students benefit from a fair and transparent attendance tracking system that supports their academic journey.

## 1.4 Functional Requirements

### 1.4.1 Admin Requirements

* **Course and Class Management:** The administrator can manage courses and classes within the system. This involves creating and updating course information, assigning teachers to specific courses or classes, and managing class schedules.
* **Attendance Reporting:** The administrator has access to comprehensive attendance reports for analysis and decision-making. They can generate attendance reports for individual students, classes, or courses over specific time periods. The reports may include details such as attendance percentages, late arrivals, and absences.
* **Data Analytics:** The administrator can leverage data analytics functionalities to gain insights into attendance trends and patterns. They can view graphical representations and statistical summaries of attendance data, enabling them to identify areas of concern, track improvement initiatives, and make data-driven decisions.

### 1.4.2 Teacher Requirements

* **Login and Authentication:** The teacher can securely log into the application using their credentials and undergo authentication to access the attendance system.
* **Course and Class Selection:** The teacher can select the relevant course and class for which they want to mark attendance. They can view their assigned courses, class schedules, and associated students.
* **Mark Attendance:** The teacher can mark attendance for students in their assigned classes. They can indicate whether students are present, absent, or late. The system should allow for batch marking of attendance if necessary.
* **Modify Attendance:** The teacher can modify attendance records if changes or corrections are required. They can update attendance statuses, adjust for makeup classes, or rectify any inaccuracies in the attendance data.
* **View Attendance Summary:** The teacher can view attendance summaries for individual students or entire classes. This includes the total number of classes attended, absences, late arrivals, and overall attendance percentages.

### 1.4.3 Student Requirements

* **Login and Authentication:** The student can securely log into the application using their credentials and undergo authentication to access the attendance system.
* **View Attendance Records:** The student can view their own attendance records for the courses they are enrolled in. They can see the number of classes attended, absences, and late arrivals, providing them with a clear overview of their attendance performance.
* **Receive Notifications:** The student can receive notifications or alerts related to attendance, such as reminders about upcoming classes, changes in schedules, or important attendance-related announcements.
* **Access Attendance Reports:** The student can access attendance reports to view their attendance history for specific courses or classes. These reports may include detailed information about individual attendance records and percentages.

### 1.4.4 System Requirements

* **User Authentication and Access Control:** 
  + The system should provide secure login functionality for admins, students, and teachers.
  + Only registered and authorized users should be able to access the attendance system.
  + User roles and permissions should be defined to control access to distinctive features and data within the system.
* **Attendance Marking and Modification:**
  + The system should allow teachers to mark attendance for students, specifying whether they are present, absent, or late.
  + Teachers should have the ability to modify attendance records when necessary, such as for makeup classes or corrections of errors.
* **Data Management and Storage:**
  + The system should securely store and manage attendance records for each student and class.
  + Attendance data should be associated with specific courses, classes, dates, and students.
  + The system should provide efficient methods for storing and retrieving attendance information.

## 1.5 Supplementary Specification

The Attendance System is an application designed to manage and track student attendance within a university environment. This Supplementary Specification outlines additional features and requirements for the system to enhance its functionality and usability.

### 1.5.1 User Interface:

The user interface should be intuitive and user-friendly, allowing teachers to easily mark attendance, modify records, and generate reports. The interface should provide clear instructions and guidance for efficient attendance management.

### 1.5.2 Performance:

The system should be capable of handling a large number of students, classes, and attendance records. It should be able to process attendance data quickly and generate reports in a timely manner. The system should maintain optimal performance even during peak usage periods.

### 1.5.3 Security:

The system should implement robust security measures to protect student attendance data. User authentication and authorization mechanisms should be in place to ensure that only authorized personnel can access and modify attendance records. All data transmissions should be encrypted to maintain data integrity and confidentiality.

### 1.5.4 Reports:

The system should generate comprehensive attendance reports for teachers, administrators, and students. The reports should include attendance percentages and absences for individual students, classes, and courses. The reports should be customizable and exportable in various formats for easy sharing and analysis.

### 1.5.5 Integration:

The attendance system should integrate with other university systems, such as the student information system and course management system. The integration should facilitate the exchange of relevant data, ensuring accuracy and consistency across different platforms.

### 1.5.6 Accessibility:

The system should adhere to accessibility standards and guidelines to ensure that it is accessible to all users, including those with disabilities. The user interface should be designed with considerations for different accessibility requirements, such as screen readers and keyboard navigation.

### 1.5.7 Support and Maintenance:

The system should be accompanied by comprehensive support and maintenance services. This includes user training, technical support, and regular system updates to address any issues or vulnerabilities that may arise.

### 1.5.6 Performance Metrics:

The system should be evaluated based on performance metrics, such as response time, system uptime, and user satisfaction. Regular performance testing and monitoring should be conducted to identify and address any performance bottlenecks or usability concerns.

### 1.5.7 Conclusion:

The Supplementary Specification highlights additional features and requirements for the Attendance System, ensuring its effectiveness and usability in managing student attendance within the university. By incorporating these specifications, the system can meet the specific needs and expectations of the institution, contributing to improved attendance tracking and management processes.

## 1.6 Glossary

1. Attendance System: The software application designed to track and manage student attendance within a university environment.
2. **Admin:** An administrative user with elevated privileges who has access to system configuration, user management, and overall system administration.
3. **Teacher:** A user role representing the faculty member responsible for marking and managing student attendance for their assigned classes.
4. **Student:** An enrolled individual attending classes who is subject to attendance tracking and record-keeping within the system.
5. **Attendance Record:** A specific instance of attendance for a student in a particular class on a given date, indicating whether the student was present or absent.
6. **Absence:** A status indicating that a student was not present for a specific class or session.
7. **Attendance Percentage:** The calculated percentage of classes attended by a student for a specific course or overall attendance.
8. **Makeup Class:** A rescheduled class to compensate for a missed session, typically arranged due to cancellations or special circumstances.
9. **User Authentication:** The process of verifying the identity and credentials of a user before granting access to the attendance system.
10. **Data Encryption:** The technique used to convert sensitive attendance data into a secure and unreadable format to protect it from unauthorized access.
11. **Attendance Report:** A document or presentation displaying attendance data and statistics, such as attendance percentages, absences, and late arrivals for students, classes, or courses.

## 1.7 Risk List & Risk Management Plan

### 1.7.1 Risk List

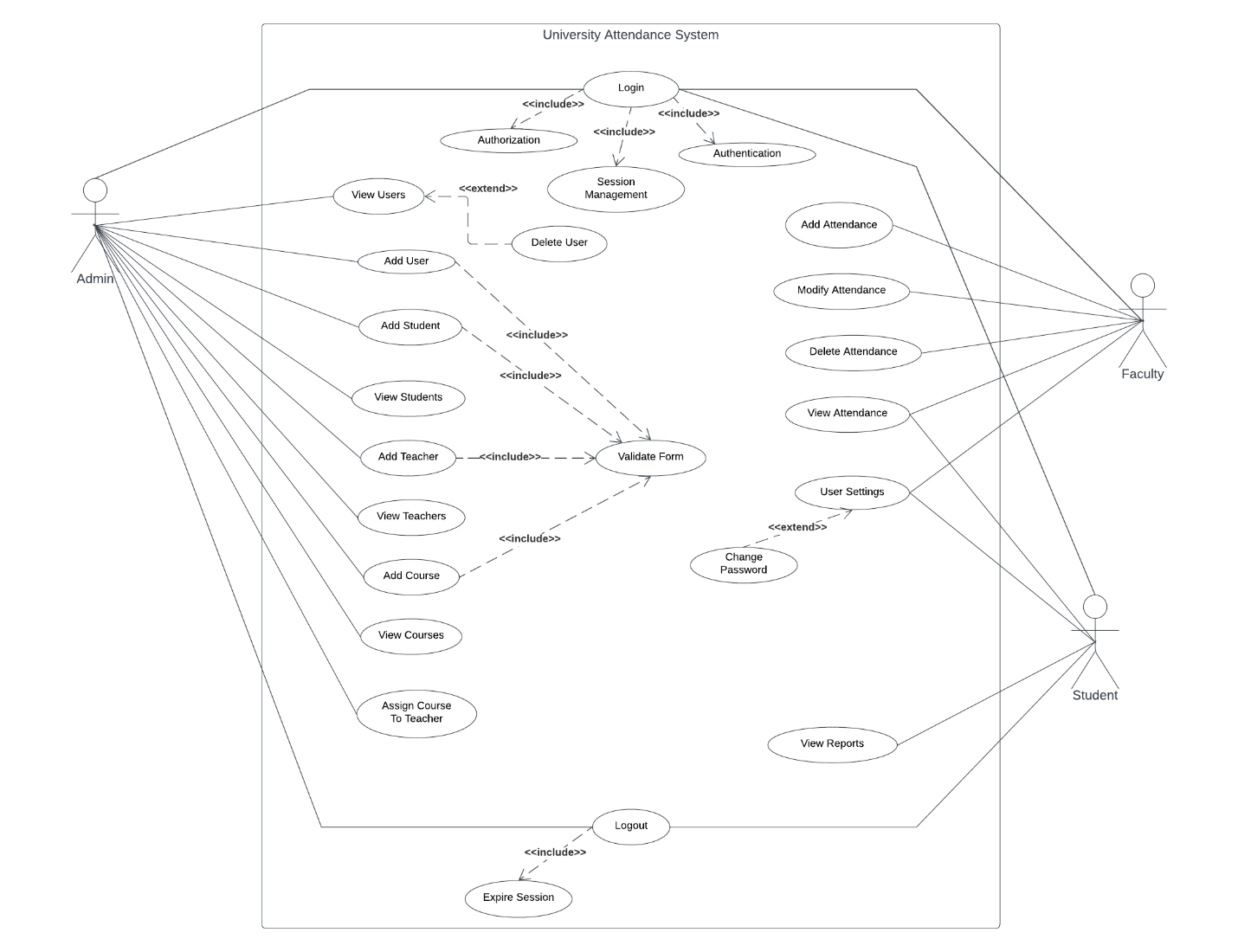
1. Data loss due to system failure or user error.
2. Inadequate system security leads to unauthorized access or data breaches.
3. Incorrect data input by users.
4. Insufficient capacity to handle large volumes of data.
5. Software bugs or errors leading to system crashes or malfunctions.
6. Inability to handle unexpected changes in scheduling or events.
7. Resistance from users to adopt the new system or change their habits.
8. Lack of resources (time, budget, personnel) to implement or maintain the system.
9. Lack of integration with other systems, such as student information systems or other systems.
10. Incomplete or inaccurate attendance data due to non-compliance or bypassing the system.
11. Insufficient user training and support resulting in user confusion or errors.

### 1.7.2 Risk Management Plan

1. Data loss: Implement regular backups and disaster recovery procedures.
2. System security: Use strong access controls, encryption, and regular security audits.
3. Incorrect data input: Implement data validation checks and provide user training.
4. Insufficient capacity: Architect the system to be scalable and conduct capacity planning.
5. Software bugs/errors: Conduct thorough testing and implement effective bug tracking.
6. Handling unexpected changes: Design the system to be flexible and have a change management process.
7. User resistance: Conduct extensive user training, involve users in design, and provide ongoing support.
8. Lack of resources: Thorough project planning and resource allocation.
9. Lack of integration: Establish collaboration, define protocols, and conduct thorough testing.
10. Incomplete/ inaccurate attendance data: Establish clear policies, implement auditing mechanisms, and provide training.
11. Insufficient user training/support: Develop comprehensive training materials, conduct sessions, and offer support channels.

# Chapter 2 Use Cases

## 2.1 Use Case Diagram



## 2.2 Use Case Distribution

## 2.3 Brief Level Use Case

## 2.4 Fully Dressed Use Cases

### 2.4.1 Login

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Login |
| Scope | University Attendance System |
| Level | User goal |
| Primary Actor | Admin, Faculty, Student |
| Stakeholders and Interests | * **System Admin:** wants to easily login to the system to access and manage the services offered efficiently. * **Teacher:** wants a seamless login process to access the system’s services. * **Students:** seek a straightforward and efficient login procedure, allowing them to access the system services. |
| Pre-conditions | * The user must be registered in the system either as a system admin, teacher, or a student. * The user possesses valid credentials, including email and password. |
| Success Guarantee | The users are logged into the system so they can perform within the system. |
| Main success scenario | 1. The user launches the system’s login screen within the application. 2. The system presents a login form to enter the email and password. 3. The user enters their registered email and password into the respective fields. 4. The system verifies the entered credentials against the stored user database. 5. The system identifies the user role (System Admin, Faculty, Student) based on the credentials provided. 6. The system grants appropriate system rights, permissions, and privileges to the user, depending on their role. 7. The system successfully logs the user into the system. 8. After successful login the system redirects the user to their respective dashboard, which is tailored to their role. |
| Exceptions | 1. If the user fails to provide both the email and password, the system will display an error message indicating that both fields are required. 2. If the user provides the password field but leaves the email field blank, the system will display an error message “Email is not valid, Provide valid email with at least 3 characters”. 3. If the user provides the email but leaves the password field blank the system will display an error message “Password is not valid, Provide valid password with at least 3 characters”. 4. If the email entered by the user does not follow the standard email format (e.g., missing '@' symbol, no domain name, etc.), the system should raise an exception indicating that the email is invalid. 5. If there is an error in connecting to the user database during the authentication process, the system will display an error message “Database connections issue please contact customer services”. 6. If the user enters invalid credentials (i.e., incorrect email and password) the system will display an error message “Invalid credentials, check your email and password”. |
| Special Requirements | 1. The login screen must be designed with a responsive layout to ensure the usability of various operating systems. 2. The system should be able to handle a large number of concurrent login requests to support scalability and accommodate peak usage periods. 3. The login process should be completed in a reasonable time frame to provide a smooth user experience. |
| Technology and Data variations | * The "Add User" feature should be compatible with various operating systems, such as windows and Linux. * The back end is developed using java programming language. * The database technology can vary, such as MySQL, PostgreSQL, MongoDB, or Oracle, to store user information. |
| Frequency of occurrences | The frequency of data entry for user addition can vary, ranging from occasional additions to bulk imports when onboarding multiple users. |
| Miscellaneous |  |

### 2.4.2 Add User:

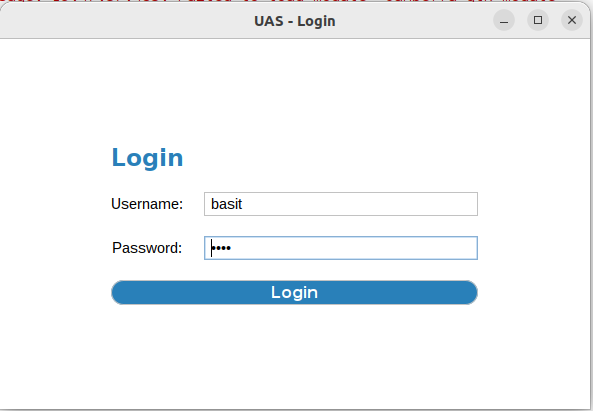
|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Add User |
| Scope | University Attendance System |
| Level | User goal |
| Primary Actor | Admin |
| Stakeholders and Interests | * **System Admin:** The Admin is responsible for managing user accounts creating new users, assigning roles, and maintaining the user database. |
| Pre-conditions | * **Authentication:** The admin must be authenticated and logged in to access the “Add User” functionality. Only authenticated users with admin privileges should have access to this feature. * **User Information Availability:** Before adding a new user, the admin should have all the required information about the user being added, such as their full name, username, email, and initial role assignment. This information is essential for creating the new user account accurately. * **Unique Email:** The admin must ensure that the username and email address for the new user are unique and not already used by another user in the system. Duplicate usernames or emails could lead to conflicts and potential login issues. * **Role Assignment:** The admin should determine the appropriate role and permissions to be assigned to the new user. Roles define the user's access rights and privileges within the system. * **User Policy Compliance:** The addition of new users must comply with any relevant user policies, rules, or regulations defined by the organization or system administrators. |
| Success Guarantee | Upon successful execution, the "Add User" functionality guarantees the creation of a unique, active user account with accurately assigned roles and compliance with system policies. |
| Main success scenario | 1. The administrator accesses the user management section in the system. 2. The administrator selects the option to add a new user. 3. The system presents a user creation form with necessary fields to enter user details, such as email, password, and role. 4. The administrator enters the required user information into the form. 5. The system validates the entered data for accuracy and completeness. 6. The system creates a new user account with the provided details and assigns the specified role. 7. The new user account is now active and can be used by the user to log into the system and access services based on their assigned role. |
| Exceptions | 1. If the user provides the password field but leaves the email field blank, the system will display an error message “Email is not valid, Provide valid email with at least 3 characters”. 2. **Invalid Email Format:** If the email entered by the user does not follow the standard email format (e.g., missing '@' symbol, no domain name, etc.), the system should display an error message “Email format is not valid, Provide a valid email address”. 3. If the admin provides the email but leaves the password field blank the system will display an error message “Password is not valid, Provide valid password with at least 3 characters”. 4. The system should verify that the role assigned to the new user (e.g., System Admin, Faculty, Student) is valid and allowed. If an invalid role is specified, the system should raise an exception and ask the administrator to select a valid role. 5. If there is an error in connecting to the user database during the authentication process, the system will display an error message “Database connections issue please contact customer services”. 6. **Duplicate User:** The system should check whether a user with the same email or username already exists in the database. If a duplicate user is found, the system should raise an exception and inform the administrator that the user already exists. |
| Special Requirements | 1. **Performance:** The "Add User" feature should respond quickly, and the user addition process should be completed within a reasonable time, even under high user loads. 2. **Security:** User data entered during the user addition process, such as passwords and personal information, should be securely stored and transmitted using encryption. Access to the "Add User" feature should be restricted to authorized administrators. |
| Technology and Data variations | The system should support authentication method such as email-password. |
| Frequency of occurrences | Frequent occurrence: During peak usage periods, such as the start of the semester, when many users may attempt to log in simultaneously. |
| Miscellaneous |  |

### 2.4.3 Add Student

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use Case Name | Add Student |
| Scope | University Attendance System |
| Level | User goal |
| Primary Actor | Admin |
| Stakeholders and Interests | * **System Admin:** The Admin is responsible for managing students, adding new students. |
| Pre-conditions | * **Authentication:** The admin must be authenticated and logged in to access the “Add Student” functionality. Only authenticated users with admin privileges should have access to this feature. * **User Information Availability:** It is mandatory that the student should be first registered as user in the system along with their role as a student, the admin then adds the student by providing their necessary information. * **Unique Reg No:** The admin must ensure that each student is added with a unique Registration number. |
| Success Guarantee | Upon successful execution, the "Add Student" functionality guarantees the creation of a unique, active student is added to the system. |
| Main success scenario | 1. The administrator accesses the Admin Dashboard in the system. 2. The administrator selects the option to add a new student. 3. The system presents a student creation form with necessary fields to enter student details, such as reg no, name, father name, DOB, CNIC, phone number and email (the one the user is registered with). 4. The administrator enters the required student information into the form. 5. The system validates the entered data for accuracy and completeness. 6. The system creates a new student account with the provided details. |
| Exceptions | 1. **Missing Required Information:** If the admin attempts to submit the student creation form without filling in all the necessary fields (such as reg no, name, father name, DOB, CNIC, phone number, and email), the system should display an error message indicating that all required information must be provided. 2. **Invalid Date of Birth (DOB):** The system should validate the format and accuracy of the Date of Birth (DOB) entered by the admin. If an invalid or incorrect date format is provided, the system should display an error message asking the admin to enter the DOB in the correct format (e.g., DD/MM/YYYY). 3. **Invalid CNIC Format:** The system should verify that the CNIC (National Identity Card) number entered by the admin follows the correct format (e.g., 12345-1234567-1). If an invalid CNIC format is provided, the system should display an error message prompting the admin to enter a valid CNIC number. 4. **Invalid Phone Number Format:** The system should validate the format of the phone number provided by the admin. If an incorrect format is entered (e.g., missing area code, incorrect number of digits), the system should display an error message asking the admin to enter a valid phone number. 5. **Invalid Email Address:** The system should validate the format of the email address entered by the admin to ensure it follows the standard email format (e.g., user@example.com). If an invalid email address format is provided, the system should display an error message asking the admin to enter a valid email address. 6. **Duplicate Student:** The system should check whether a student with the same registration number, or email already exists in the database. If a duplicate student is found, the system should raise an exception and inform the administrator that the student already exists. 7. **Data Validation:** The system should perform data validation to ensure that the entered student details are accurate and valid. If any data inconsistencies or errors are detected, the system should display appropriate error messages, allowing the admin to correct the information before submitting the form. 8. **Database Connection Issue:** If there is an error in connecting to the database during the student creation process, the system should display an error message notifying the admin about the database connection issue and asking them to try again later or contact customer support for assistance. |
| Special Requirements | 1. **Performance:** The "Add Student" feature should respond quickly, and the user addition process should be completed within a reasonable time, even under high user loads. 2. **Security:** Student data entered during the user addition process, such as Reg no, name, CNIC, phone number, email, should be securely stored and transmitted using encryption. Access to the "Add Student" feature should be restricted to authorized administrators. |
| Technology and Data variations | * The system should support authentication methods such as email-password. * **User Interface:** The user interface for the student creation form can vary based on the technology used, such as a web-based form, a mobile app interface, or a command-line interface for administrative purposes. * **Platform Compatibility:** The system should be designed to work seamlessly on different platforms, such as Windows, macOS, Linux |
| Frequency of occurrences | 1. Semester Start: At the beginning of each academic semester or term, universities typically enroll new students, leading to a significant surge in student additions. During this period, the "Add Student" use case will be frequent as administrators add a large number of new students to the system. 2. Continuous Admissions: Some universities have continuous admissions, allowing students to join throughout the academic year. In such cases, the "Add Student" use case will occur frequently as new students are admitted on an ongoing basis. 3. Late Registrations: There may be instances of late registrations or special cases where students are added after the regular semester starts. These events will result in additional occurrences of the "Add Student" use case. 4. Student Transfers: Students might transfer from one course or program to another within the university, leading to updates or re-enrollment. This could cause additional occurrences of the "Add Student" use case. |
| Miscellaneous |  |

# Chapter 3 UI Prototypes

## 3.1 Login UI



## 3.2 Dashboard

## 3.3 Add Attendance

# Chapter 3 Domain Model

## 3.1 Domain Model

# Chapter 4 System Sequence Diagram (SSD)

## 4.1 SSD

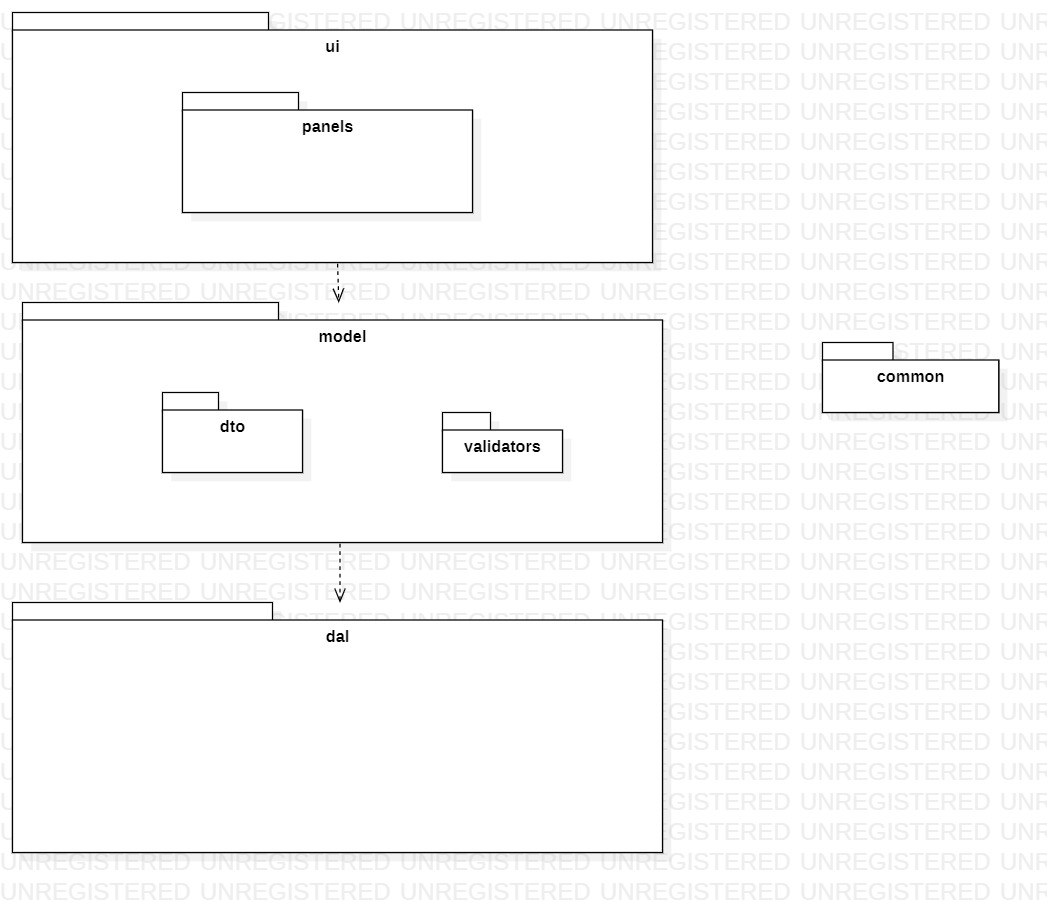


# Chapter 5 Operation Contract (OC’s)

## 5.1 Operation Contract

# Chapter 6 Package Diagram

## 6.1 Package Diagram



# Chapter 7 Class Diagram

## 7.1 Class Diagram:

# 

# Chapter 8 Interaction Diagram

## 8.1 Interaction Diagram



## 8.2 Sequence Diagram