**COMSATS University Islamabad,**

**Abbottabad Campus**

**SOFTWARE REQUIREMENTS SPECIFICATION   
(SRS DOCUMENT)**

**for**

**<University Attendance System>**  
Version 1.0

***By***

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

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

**Ahmed Ali Khan CIIT/SP20-BSE-016/ATD**

***Supervisor*Prof. Mukhtiar Zamin**

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

**Table of Contents**

**Revision History 3**

**1.** **Introduction 5**

1.1 Purpose 5

1.2 Scope 5

**2.** **Overall description 5**

2.1 Product perspective 5

2.2 Operating environment 5

**3.** **Requirement identifying technique 6**

3.1 Use Case Diagram 6

3.2 Use Case Description 6

**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\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Introduction**

A login system is a basic yet essential feature for any web application or software that requires user authentication. It allows users to securely access their accounts by providing their login credentials such as usernames and passwords.

This project aims to create a login system with default features, which includes a user registration page, login page, and a dashboard that displays the user's profile and basic information. The login system will have strong security measures in place, such as password encryption and multi-factor authentication, to ensure the safety of user data.

When a user registers for the system, they will be asked to provide their basic information and set up a username and password. Once registered, the user will be able to log in to the system using their login credentials. Upon logging in, the user will be directed to their dashboard, which displays their personal information, such as their name, email, and account status.

The default features of this login system can be extended to include additional functionality such as password reset, email verification, and social login integration, depending on the specific needs of the project. Overall, the login system with default features will provide a secure and reliable way for users to access their accounts and manage their information.

**Overall description**

**Product perspective**

Overall, taking a product perspective involves looking at a product from multiple angles, including the customer's needs, technology advancements, and market trends. It requires a deep understanding of the target market and a willingness to adapt to changing market conditions to ensure the product's success.

**Design and implementation constraints**

* java
* SQL Database

**Use case diagram:**

**Requirement identifying technique**

**Table 1 Show the detail use case template**

|  |  |
| --- | --- |
| **Use Case ID:** | UC1 |
| **UseCase Name:** | Create slot |
| **Actors:** | Teacher only |
| **Description:** | The teacher will setup their slot according to their schedule which is provided by the university or organization |
| **Trigger:** | remind the class to the teacher for attendance |
| **Preconditions:** | User must have their email address and password to use the system  Must have known the credentials  Teacher should have the schedule time table which is provided by the university  Teacher should have created a class |
| **Post conditions:** | The alerts message will display on the screen |
| **Normal Flow:** | Teacher first should login the app  Then click on the create class/course  Browse the file into the PC  Select and import the student file and select the slot provided by the organization |
| **Alternative Flows:** | 1.1 can setup multiples slots  1. teacher can add or modify a specified selected slot |
| **Exceptions:** | 1.0. if teacher enter the wrong credentials the error will be display |
| **Business Rules** | BR-1 alerts will generate on the base of selected time day  BR-2 display of alerts will bailing max 10mints. |
| **Assumptions:** |  |

|  |  |
| --- | --- |
| **Use Case ID:** | UC2 |
| **UseCase Name:** | Take attendance |
| **Actors:** | Teacher only |
| **Description:** | The teacher will take the attendance of the student by checking the checkbox. |
| **Trigger:** | Save the time and avoid the manual file system |
| **Preconditions:** | User must have their email address and password to use the system  Must have known the credentials  Teacher should have the source of attendance  Student record must be saved in data base |
| **Post conditions:** | The attendance list must have save  Viewable and downloadable |
| **Normal Flow:** | Teacher first should login the app  Check the time table of the class  Mark attendance |
| **Alternative Flows:** | 1.1 can modify the attendance may time  1.2 teacher can add or modify the attendance |
| **Exceptions:** | 1. if teacher enter the wrong credentials the error will be display |
| **Business Rules** | BR-1 attendance time will takes some seconds  BR-2 attendance list will be generated as fast as possible. |

|  |  |
| --- | --- |
| **Use Case ID:** | UC3 |
| **UseCase Name:** | View attendance |
| **Actors:** | Teacher only |
| **Description:** | The teacher will take the attendance of the student by marking attendance. |
| **Trigger:** | Save the time and avoid the manual file system |
| **Preconditions:** | User must have their email address and password to use the system  Must have known the credentials  Teacher should have the source of attendance  Student record must be saved in data base |
| **Post conditions:** | : Teacher can mark /unmark the student in the attendance sheet and also see them. |
| **Normal Flow:** | Teacher first should login the app  Check the time table of the class  Teacher will be able to mark attendance and after taking the attendance teacher will be able to see or view the attendance |
| **Alternative Flows:** | 1.1 can modify or view the attendance may time  1.2 teacher can add or modify the attendance |
| **Exceptions:** | 1: Attendance sheet does not be shown. Because of internet issues and server down. |
| **Business Rules** | BR-1 attendance sheet will open with few seconds  BR-2 if changes possible then teachers are able to changes it manually |
| **Assumptions:** |  |

|  |  |
| --- | --- |
| **Use Case Section** | **UC4** |
| Use Case Name | Update Attendance |
| Primary Actor | Teacher |
| Preconditions | After successful mark the attendance, we can also update the attendance within 15 days. |
| post condition | Update the attendance sheet after marking. |
| Normal flow | Teacher has rights to update the wrong attendance. |
| Alternative flow |  |
| Exteceptions | After 15 days later, attendance will not be update.  Teacher is not correct the wrong attendance. |
| Business rule | 1: It should be reliable.  2: it has high performance and secure database.  3: User friendly: user can easily understand and handle in first use. |

**Functional Requirements**

* System should be able save the teacher login details
* System should be able to create a class of the students
* System should be able to import the student list in the excel form
* System should be able to generate alerts for the attendance reminder
* Users should be able to set up slots according to their schedule.
* System should be able to save the attendance of each student per unique id.
* System should be able to save, update and delete the attendance

**Non-Functional Requirements**

**Usability:**

The application shall allow a user to take the attendance easily with a single interaction.

**Security**

The software should be handled only by the administrator and authorized users.

**Performance**

System shall be able to take attendance within 2 seconds

System shall be able to load a page within 1 seconds

**Scalability:**

The system must be scalable enough to support 1,000,000 visits at the same time while maintaining optimal performance.

**Portability:**

The application must support windows.

**Reliability:**

The system must perform without failure in 95 percent of use cases during a month.

**Maintainability**

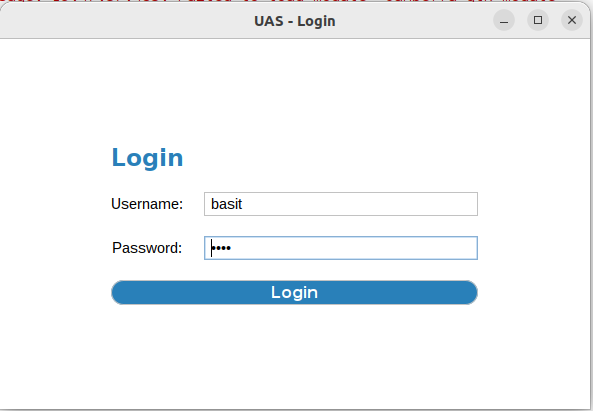
The mean time to restore the system (MTTRS) following a system failure must not be greater than 10 minutes

**Availability:**

The application dashboard must be available to user’s 99.98 percent of the time every month.

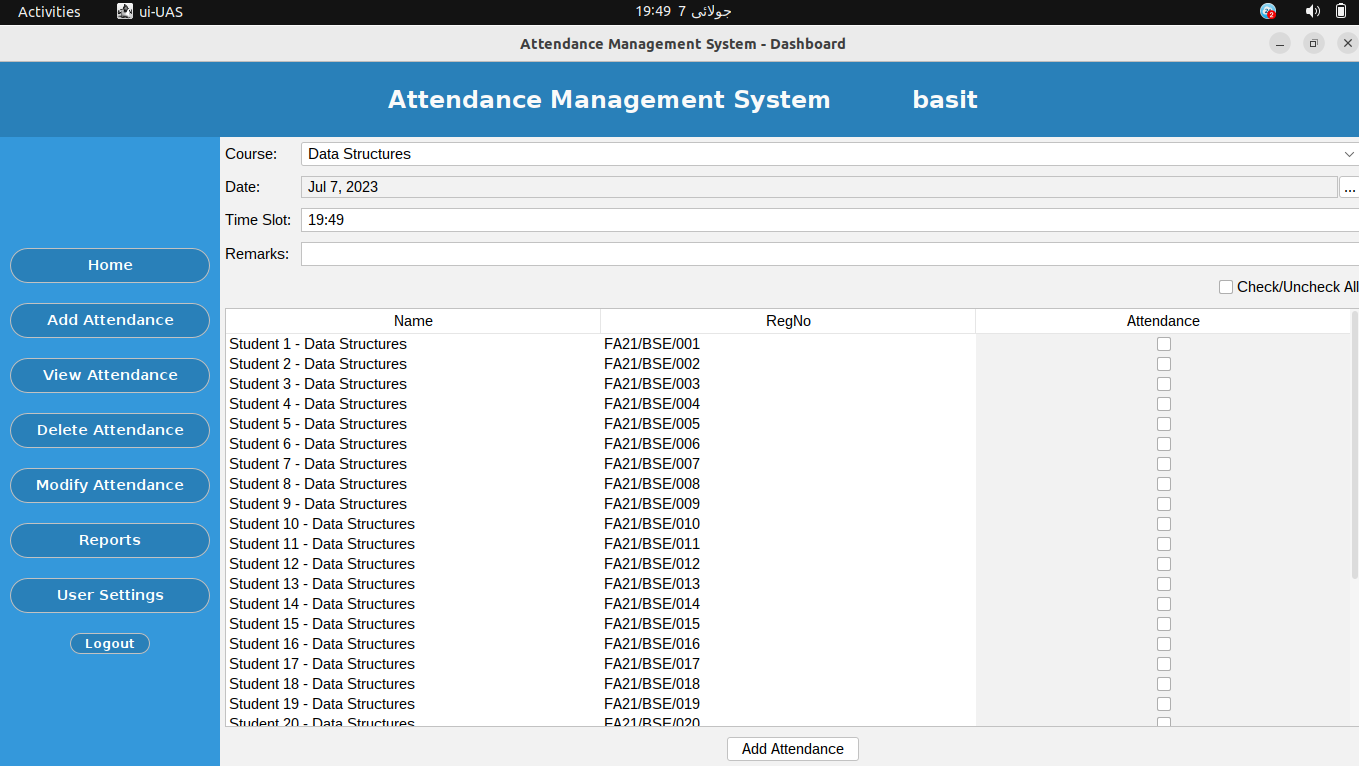
**UI Prototypes:**

## Login UI:



## **Dashboard:**

## ADD Attendance:



# UML Designs:

## Class Diagram:

## Package Diagram:

## 

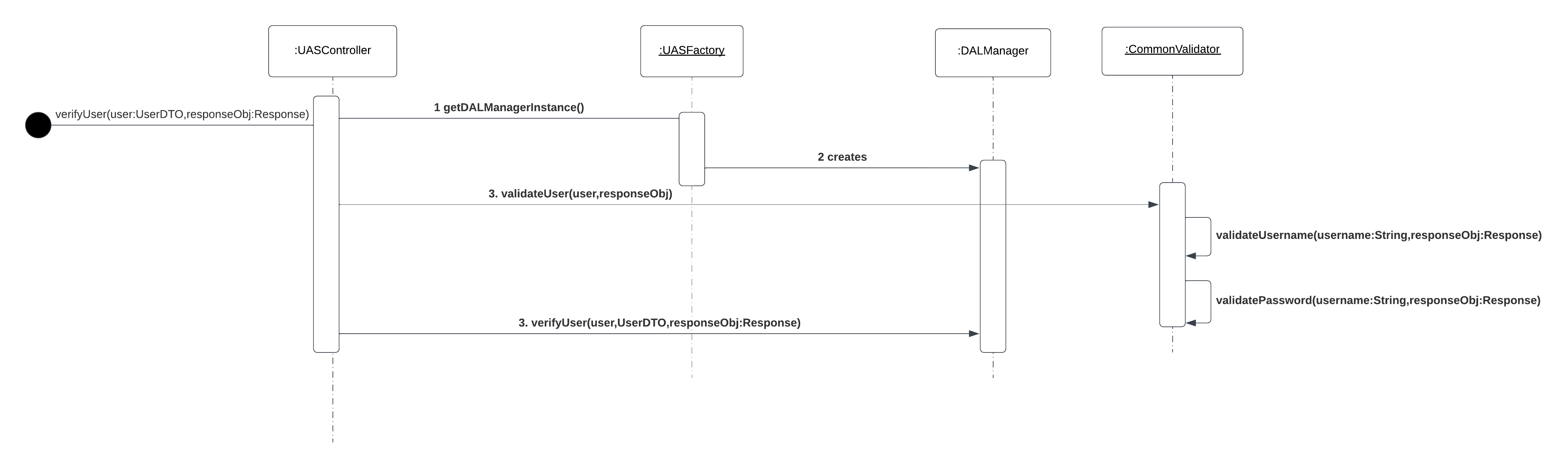
## System Sequence Diagram:



## Interaction Diagram:



## Sequence Diagram:

Login: