

National University of Computer and Emerging Sciences



Deliverable #4

2024/05/05

Legendary Sentinel of Attendance

Group Members:

Name	Roll Number	Sub-section
Ahmed Abdullah	22L-7503	A
Ibtehaj Ali	22L-7476	A

Fundamentals of Software Engineering
Spring 2024
Department of Computer Science
FAST-NU, Lahore, Pakistan.

Introduction

The project, ‘Attendance++’, is a desktop application. Our desktop application utilising C# for an attendance system. The system incorporates face recognition technology to register and track students' attendance efficiently.

Unlike ‘Traditional attendance systems’ that often rely on manual methods, leading to inefficiencies and inaccuracies in recording attendance data, this project leverages face recognition technology to automate the attendance tracking process.

User Stories

1. Authentication:

As a user, I want to be able to securely log in to the application using my username and password to access the system functionalities.

2. Register Faces:

As a user, I want to register my face for attendance recognition using the application's webcam, allowing me to participate in the attendance system.

3. Facial Recognition Attendance:

As a user, I want the application to utilise facial recognition technology to mark my attendance when I open the attendance system, allowing for a seamless and efficient attendance recording process

4. Data Visualization

As a user, I want to be able to view graphs of my data so that I can see attendance data for all subjects associated with the teacher ID in a single interface. I want the attendance tracking system to handle files efficiently, so that I can easily view attendance data for all subjects associated with my teacher ID and view them in a single interface

6. Dashboard:

As a teacher, I want a personalised dashboard that displays my detailed information, so that I can easily view my profile and associated teaching responsibilities

7. Feedback System:

As a user, I want to be able to submit feedback through a unified feedback system so that I can easily provide feedback on the application and be able to see feedback from other people who use this application.

8. Student Data Management

As a system user, I want to fetch the data of a student from a CSV file based on the recognized face captured through the live camera, so that I can access accurate information about the student when required. Once the student's data is found in the CSV file, the application should display the data in an easy-to-read format.

9. Teacher Data Management

As an administrator, I want to add a new teacher to the system by providing their data, so that they can be included in the list of teachers and assigned to teach specific classes.

10. Admin Data Management

Administrators can add, update, or delete teacher, student, and admin profiles, along with assigning roles and permissions.

11. Marking Attendance

The system automatically marks attendance using facial recognition from live webcam captures, prompting manual marking when faces are not recognized.

12. Display Attendance

Users access attendance data via a dashboard, viewing records by class, date, or student, with graphical representations available for trend analysis and export options for further analysis.

Sprint 3 (Sprint Backlog Included)

I. Fetch Student Data

As a system user, I want to fetch the data of a student from a CSV file based on the recognized face captured through the live camera, so that I can access accurate information about the student when required.

Sub-User Stories:

1.1 Face Recognition: The application should recognize the face from the live camera feed and associate it with a student ID.

1.2 Student Data Lookup: The application should use the recognized student ID to search for the student's data in the CSV file.

1.3 Display Student Data: Once the student's data is found in the CSV file, the application should display the data in an easy-to-read format.

1.4 Error Handling: If the recognized face does not match any record in the CSV file, the application should display an appropriate error message to the user.

II. Student Attendance Graph

As a teacher, I want to view a graph of the monthly average attendance of a particular student, filtered by the student's section and subject, so that I can easily understand their attendance trends over time.

Sub-User Stories:

1.1 Fetch Student Details: Upon clicking "Fetch Detail," the application should retrieve the student's data, including their section and subject.

1.2 Get Monthly Average: The application should retrieve the monthly average attendance for the student based on the retrieved data.

1.3 Generate Attendance Graph: The application should generate a graph that displays the student's monthly average attendance for the specified section and subject.

1.5 Display Attendance Graph: The application should display the generated graph to the user in an easy-to-understand format.

1.6 Error Handling: If there is insufficient data to generate the graph, the application should notify the user with an appropriate message.

III. Feedback System File Handling:

As a user, I want the feedback system to store and handle feedback data in a text file, so that feedback can be easily recorded, saved, and displayed as required.

Sub-User Stories:

1.1 Store Feedback: The application should store each feedback entry in a text file named "feedback" in the format "name: feedback" on each line.

1.2 Retrieve Feedback: The application should read the feedback entries from the text file.

1.3 Display Feedback: The application should display the retrieved feedback entries in a clear and organised manner

1.4 Error Handling: The application should handle errors gracefully if the text file is unavailable.

1.5 Append New Feedback: The application should be able to append new feedback entries to the text file without overwriting existing data.

IV. Add Teacher

As an administrator, I want to add a new teacher to the system by providing their data, so that they can be included in the list of teachers and assigned to teach specific classes.

Sub-user stories:

1.1 Input Teacher Data: The application should provide a form for entering the teacher's data, including ID, password, name, father's name, date of birth, CNIC, email, specialisation, and section of class.

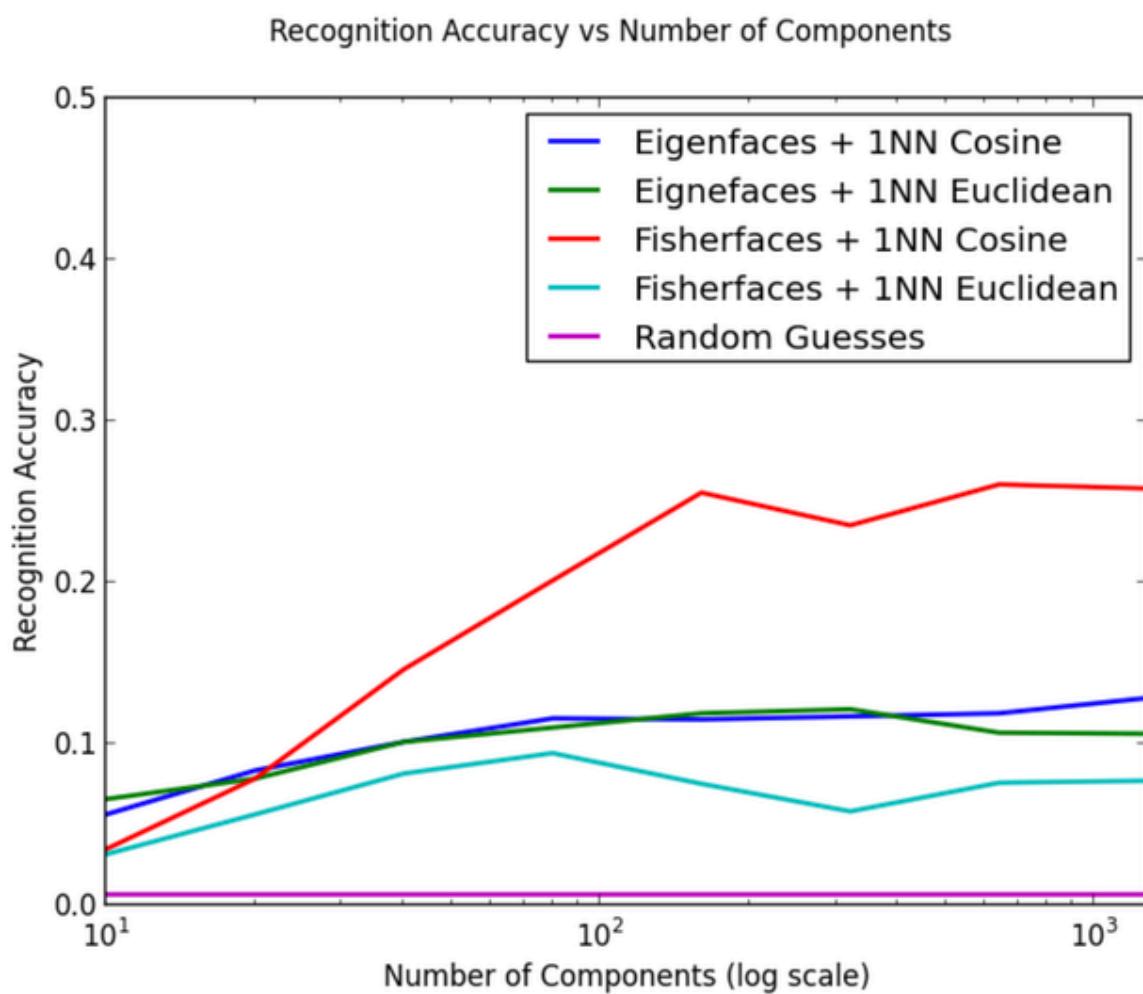
1.2 Validate Data: The application should validate the inputted data to ensure it meets the required format and constraints (e.g., unique ID, valid email format, etc.).

1.3 Store Teacher Data: Once the data is validated, the application should store the teacher's data

1.4 Confirmation Message: The application should provide a confirmation message to the user upon successfully adding the teacher.

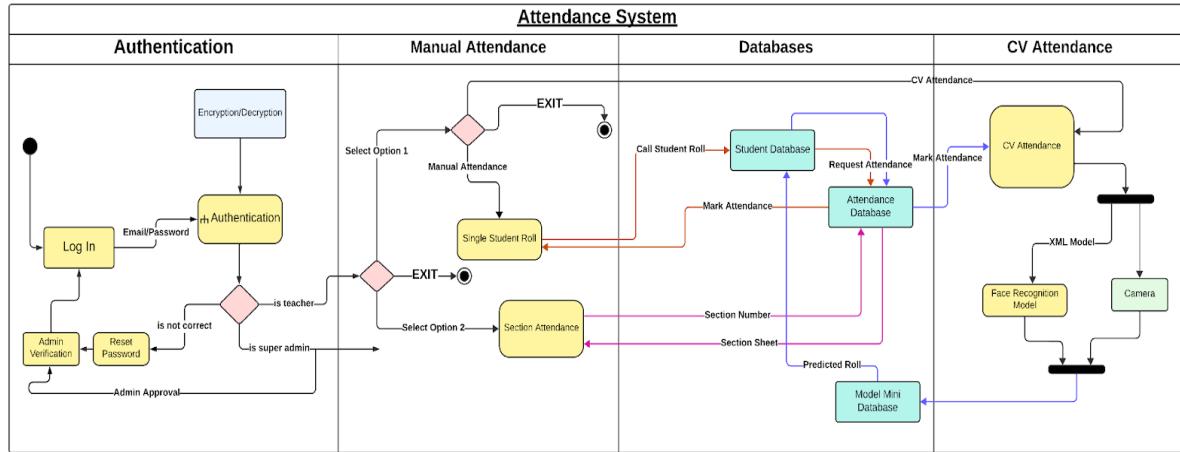
1.5 Error Handling: If there are issues with the data input, the application should notify the user with clear error messages.

Metrics

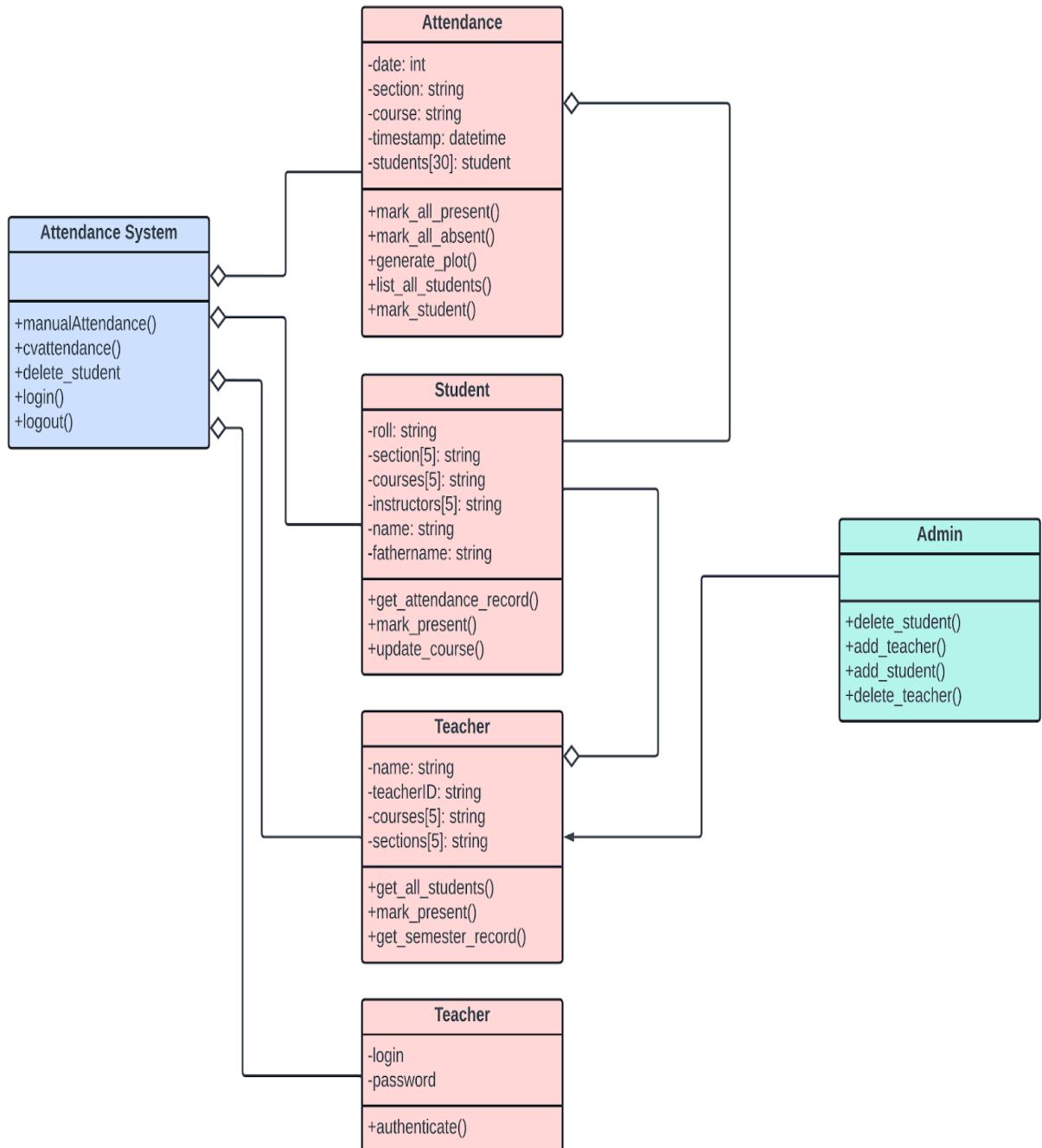


Design/Diagram

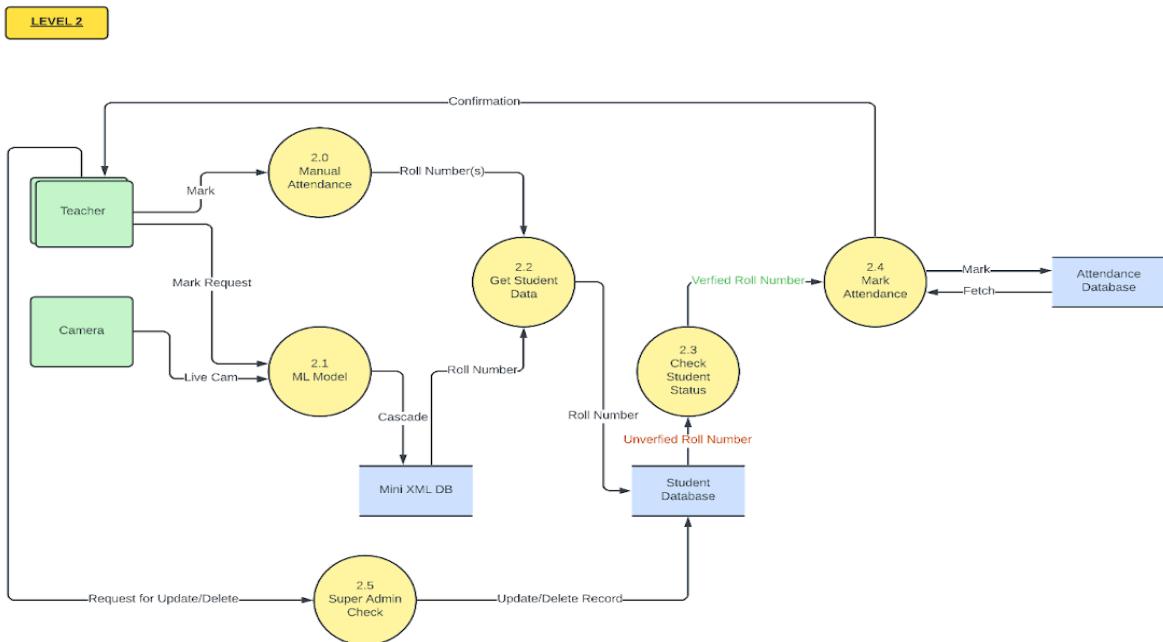
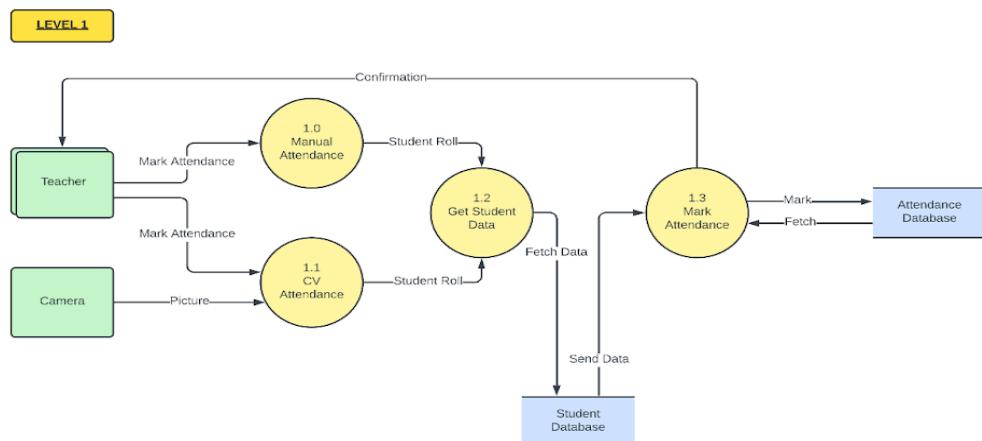
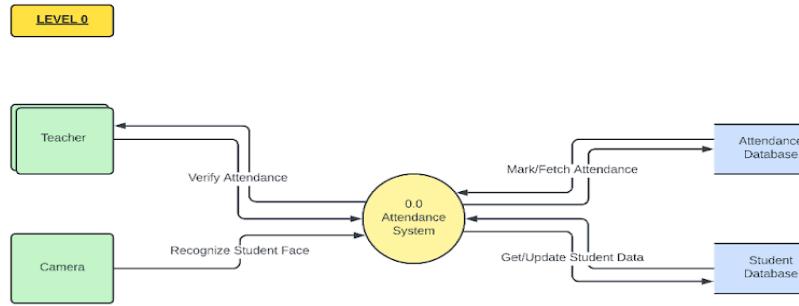
Swimlane



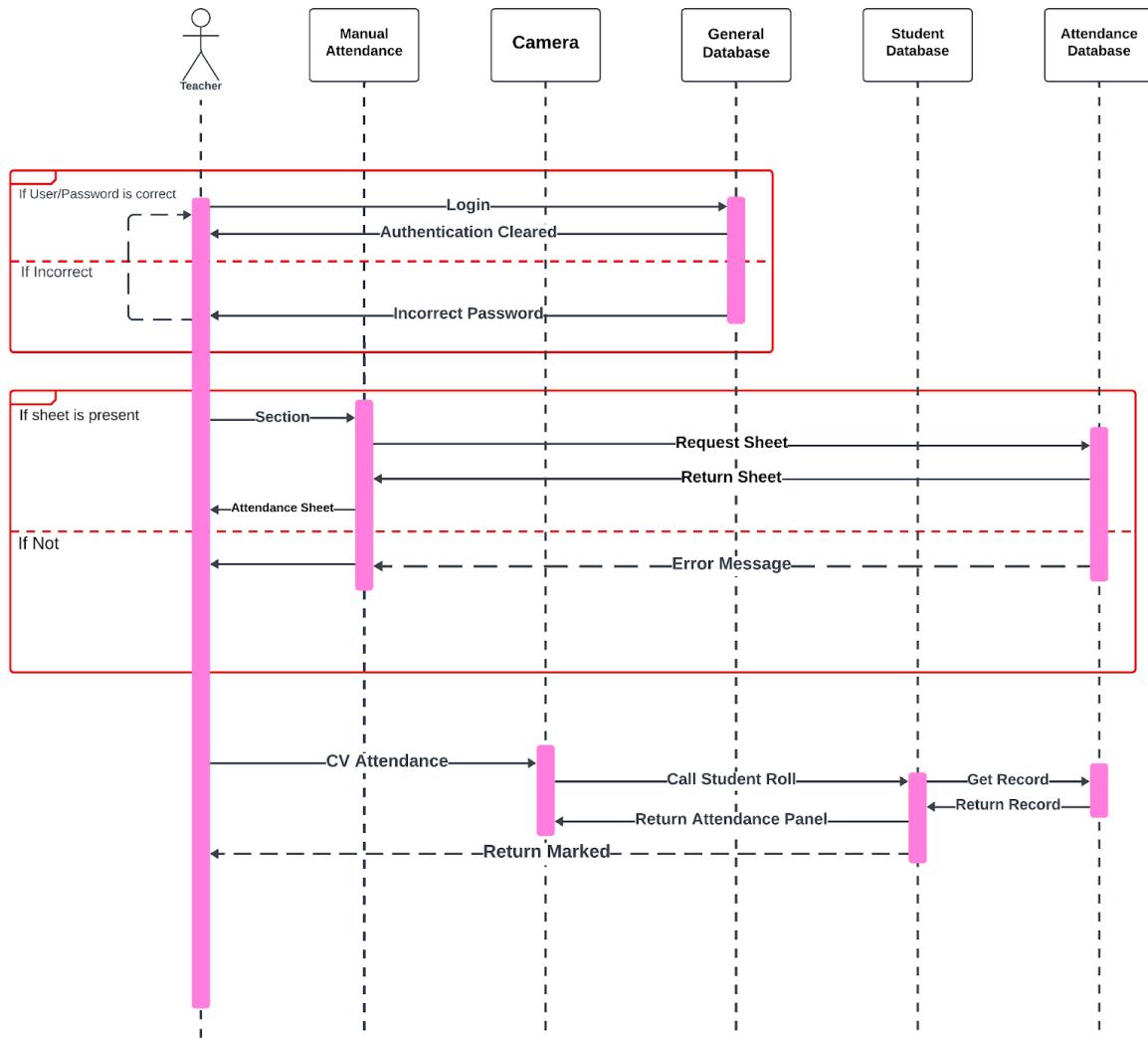
Class Diagram



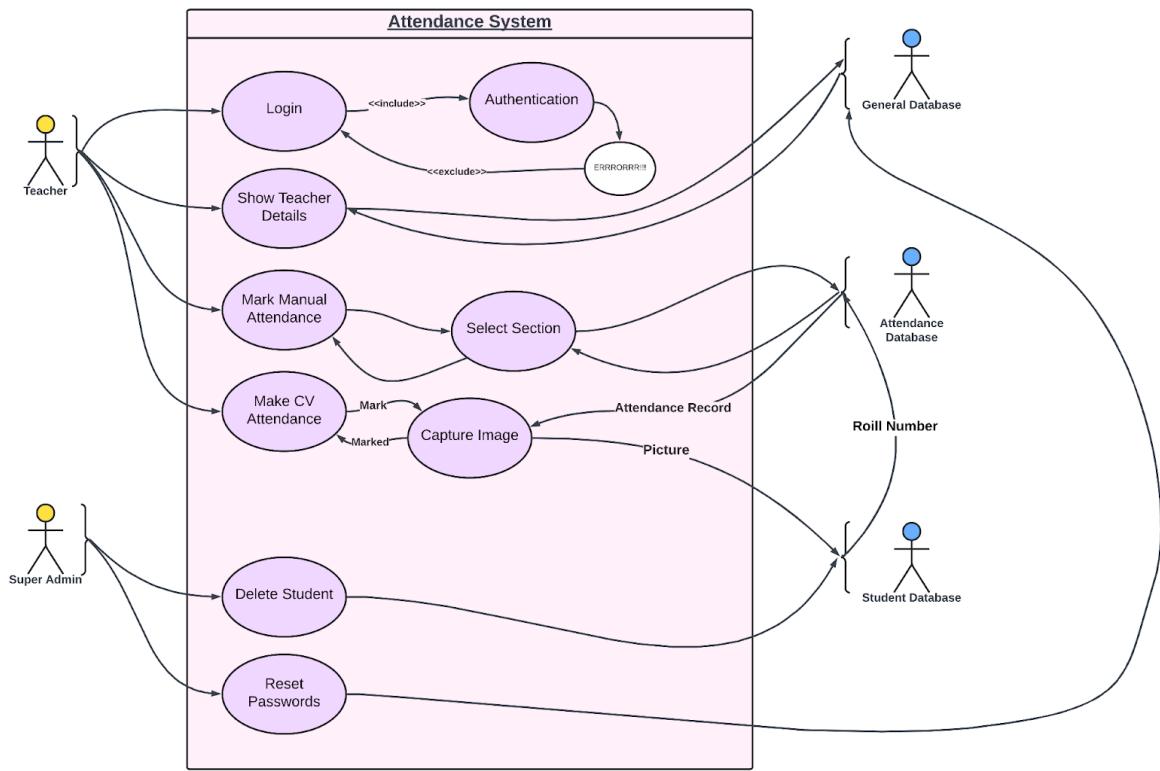
Data Flow Diagram



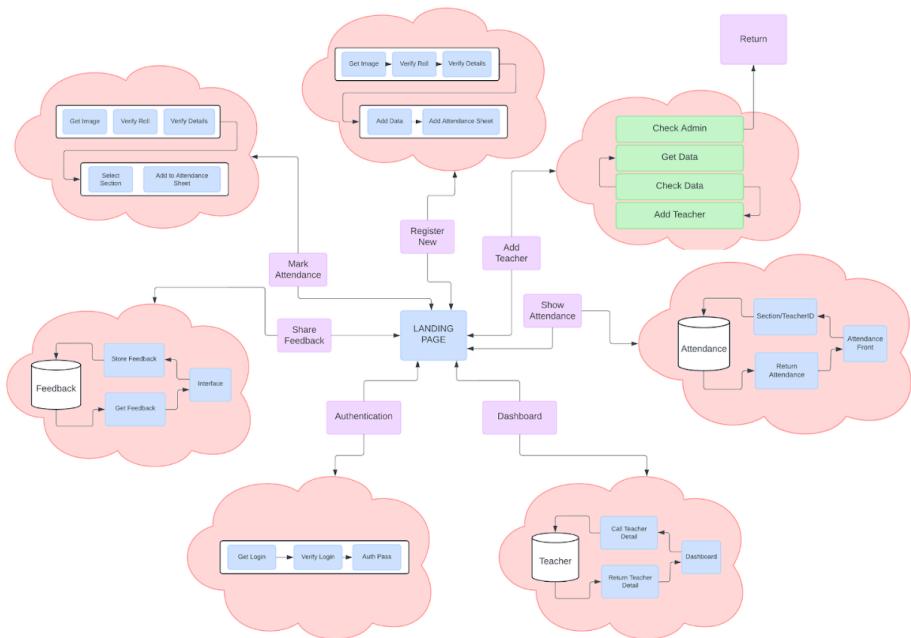
Sequence Diagram



Use Case Diagram



Architecture



Implementation

The screenshot shows the login interface of the Attendance++ application. The top bar displays the title "ATTENDANCE++" and the date "03/05/2024". A blue button labeled "OpenMenu" is in the top right. The main area has a green gradient background with two large semi-circular shapes. It contains fields for "TeacherID" (Admin) and "Password" (Admin123), a checked "Remember Me" checkbox, and a pink "Log In" button.

TeacherID
Admin

Password
Admin123

Remember Me

Log In

03/05/2024

ATTENDANCE++

HOME

REGISTER NEW

ATTENDANCE++

GRAPH

SHOW ATTENDANCE

ADD TEACHER

FEEDBACK

LOG OUT

DASHBOARD

Name: Emma Brown

TeacherID: Admin

Father Name: Brown

Email: emma.brown@example.com

CNIC: 246810-1357924-8

DOB: 08/05/1996

Specialization: Computer Science

Sections: BDS-2A;BDS-5C;BCS-6A

HideMenu

ATTENDANCE++

- [HOME](#)
- [REGISTER NEW](#)
- [ATTENDANCE++](#)
- [GRAPH](#)
- [SHOW ATTENDANCE](#)
- [ADD TEACHER](#)
- [FEEDBACK](#)
- [LOG OUT](#)

03/05/2024

REGISTER NEW STUDENT

CAMERA

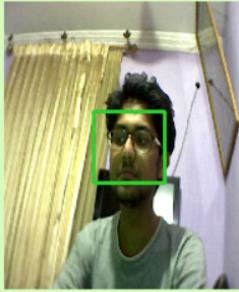


Image Saved Successfully: 8/8

Roll Number: L 22 7503
Name: Ahmed
Father Name: Aslam
Batch: 22 Fall
CGPA: 2.7

OPEN CAMERA **CAPTURE** **SAVE DETAILS**

[HideMenu](#)

ATTENDANCE++

- [HOME](#)
- [REGISTER NEW](#)
- [ATTENDANCE++](#)
- [GRAPH](#)
- [SHOW ATTENDANCE](#)
- [ADD TEACHER](#)
- [FEEDBACK](#)
- [LOG OUT](#)

03/05/2024

ATTENDANCE++

CAMERA



Roll Number: L227503
Name: Ahmad
Father Name: Aslam
Batch: Fall-2022
CGPA: 2.88

OPEN CAMERA **MARK ATTENDANCE**
FETCH DETAILS

Section Subject

L227503(6)

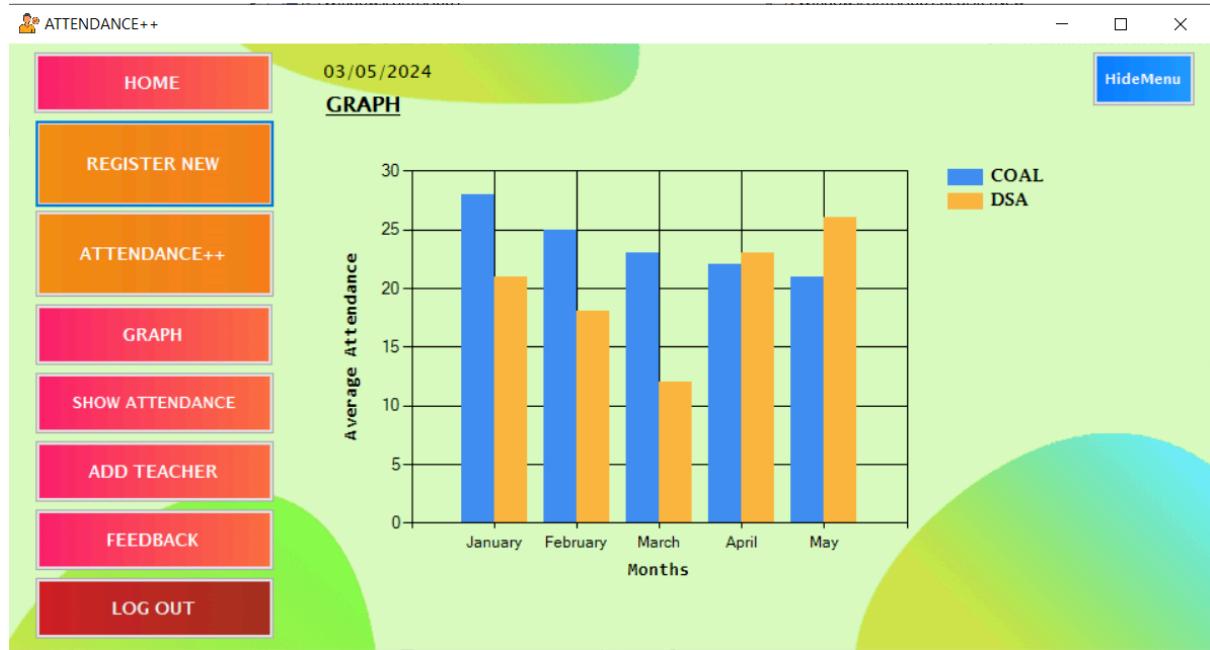
Attendance



Overall Attendance

Month	Attendance
January	25
February	25
March	25
April	18
May	16

[HideMenu](#)



ATTENDANCE++

03/05/2024

CHECK ATTENDANCE

BCS-2C | CS2001 (PF)

Friday , 3 May 2024

FETCH ATTENDANCE

	Attendance Status	Roll Number(s)
▶	P	L227503
	P	L227504
	P	L227243
	P	L227476
	P	L227223

HOME

REGISTER NEW

ATTENDANCE++

GRAPH

SHOW ATTENDANCE

ADD TEACHER

FEEDBACK

LOG OUT

ATTENDANCE++

03/05/2024

FEEDBACK

Arif: Arigato!!
Ibto: Oni Chan!!!
Arif: Let me see!
Admin: Is it working?
L227503: Oh Wow!!
Anonymous: Working Anonymously?

I can write something here!!!!

Anonymous? Sus.

SUBMIT FEEDBACK

HOME
REGISTER NEW
ATTENDANCE++
GRAPH
SHOW ATTENDANCE
ADD TEACHER
FEEDBACK
LOG OUT

HideMenu

ATTENDANCE++

03/05/2024

ADD TEACHER

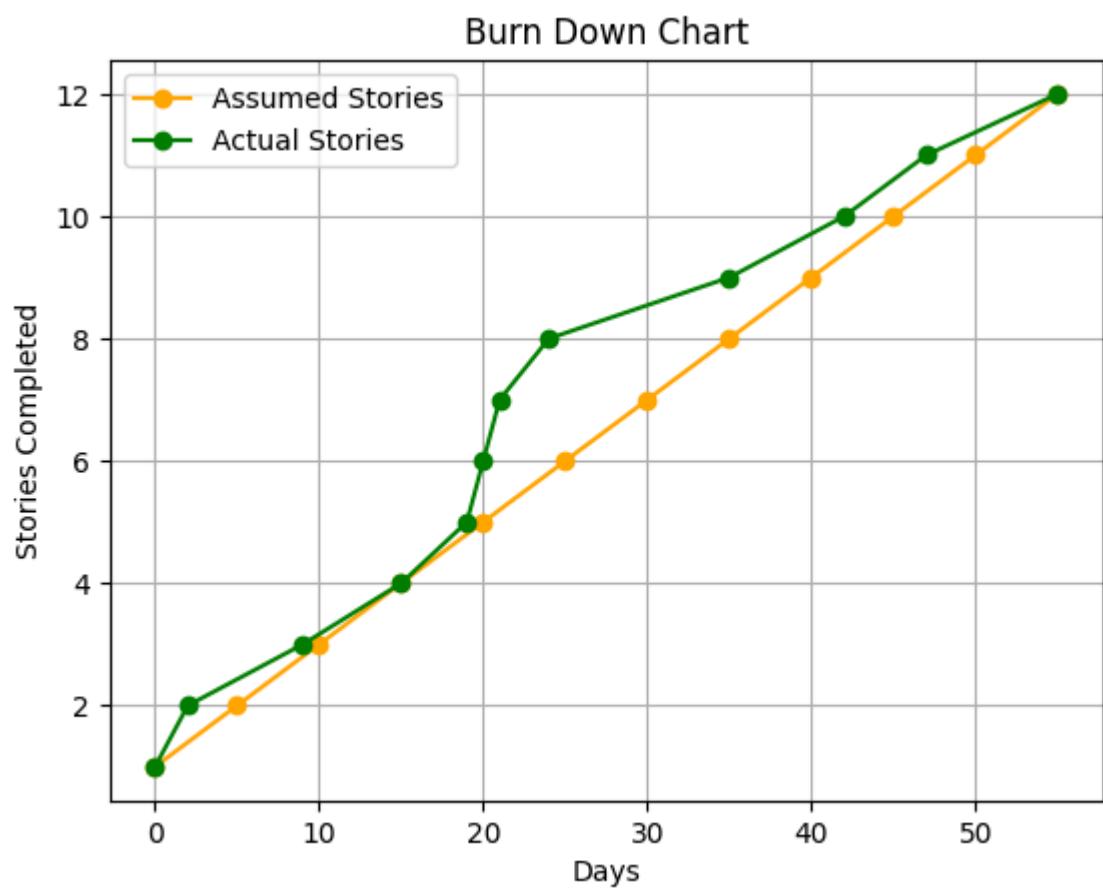
TeacherID: L 18 1112
Password: pancakes123\$\$
Name: Dr. Ibtehaj Alvi
Father Name: Obaid
Email: I181122 @lhr.nu.edu.pk
CNIC: 12336 9797993 1
DOB: Wednesday, 31 December 2003
Specialization: Data Scrapping
Section(s): BDS-3A;BDS-2A;BSE-2A

SAVE DETAILS

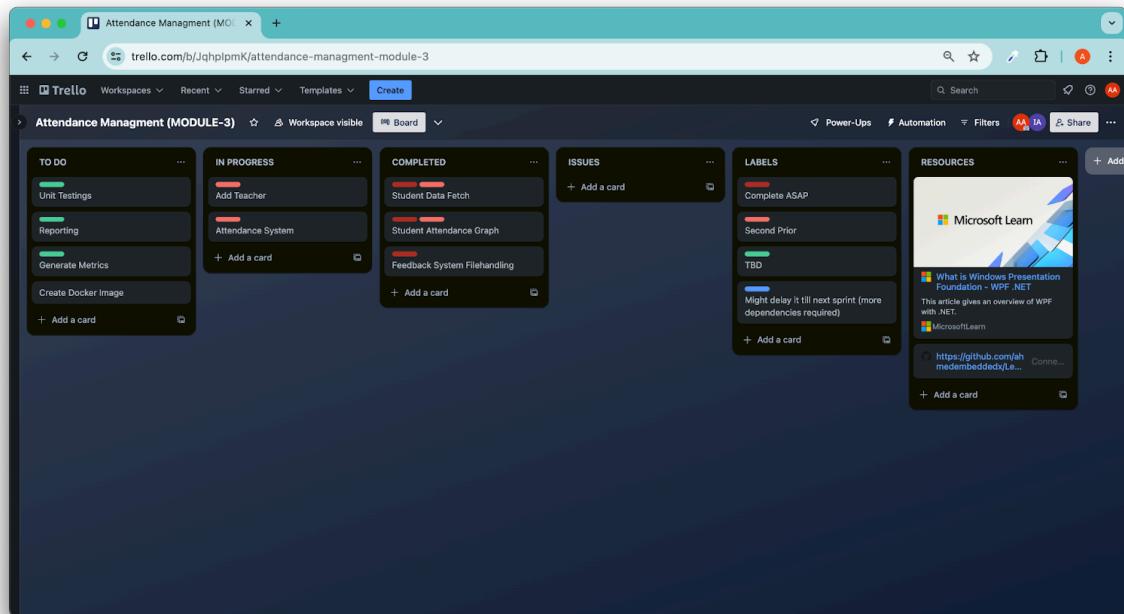
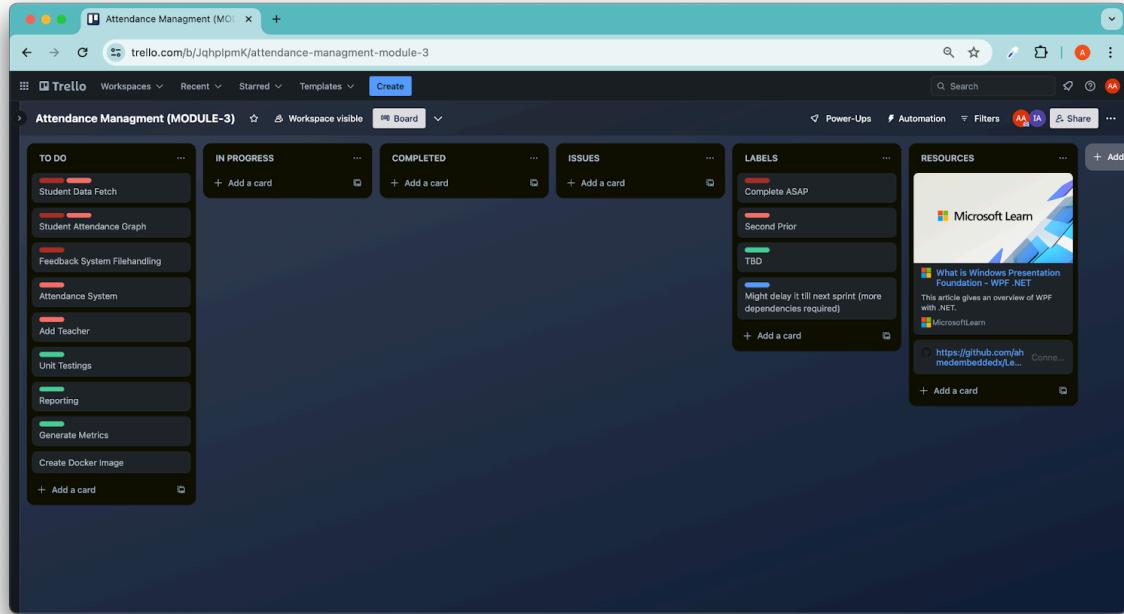
HOME
REGISTER NEW
ATTENDANCE++
GRAPH
SHOW ATTENDANCE
ADD TEACHER
FEEDBACK
LOG OUT

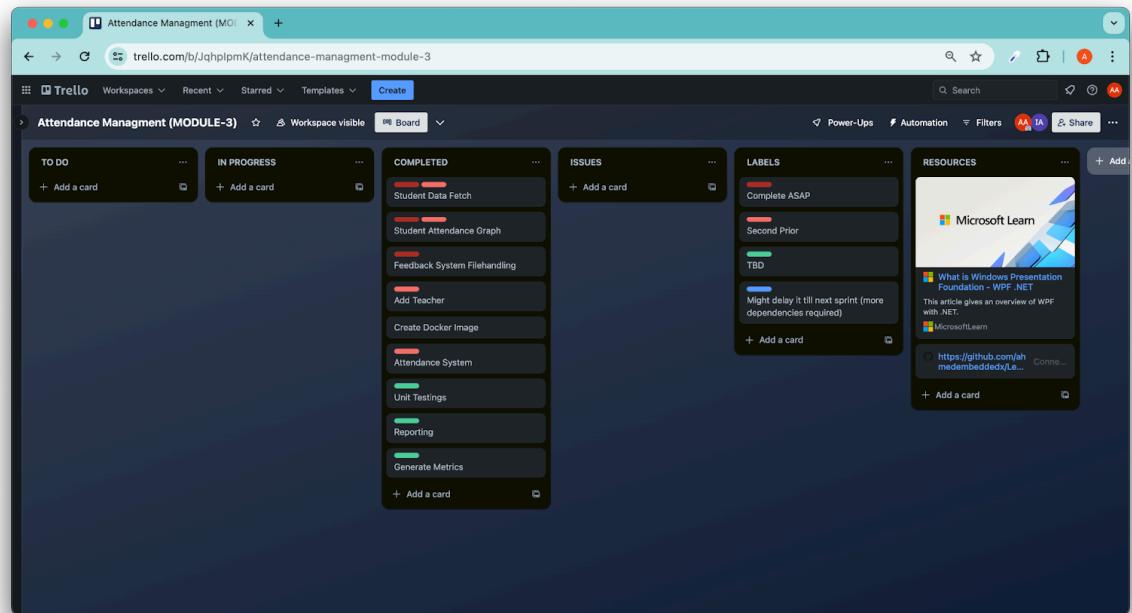
HideMenu

Burn Down Chart



ScrumBoard





Boundary value analysis testing

1. AddTeacher

Input Field	Test Case	Input	Expected outcome	Actual outcome	Analysis
Name	Min Length(1)	"A"	System accepts the input	System accepts the input	The system behaves as expected
	Max Length(40)	Str with len(40)	System accepts the input	System accepts the input	The system behaves as expected
	just below MinL(0)	""	System rejects the input	rejects and displays invalid input	The system behaves as expected
	just above MaxL(41)	Str with len(41)	does not take more than 40 inputs	does not take more than 40 inputs	The system behaves as expected
ID	Min length(1)	1	System accepts the input	System accepts the input	The system behaves as expected
	Max length (4)	1231	System accepts the input	System accepts the input	The system behaves as expected
	just below Min(0)	null	System rejects the input	rejects and displays invalid input	The system behaves as expected
	just below Max(5)	12345	does not take more than 4 inputs	does not take more than 4 inputs	The system behaves as expected
Mail	Min Length	1	System accepts the input	System accepts the input	The system behaves as expected
	Max Length	7	System accepts the input	System accepts the input	The system behaves as expected
	just below MinL	0	System rejects the input	rejects displays invalid input	The system behaves as expected
	just below MaxL	8	does not take more than 7 inputs	does not take more than 7 inputs	The system behaves as expected
Date of Birth	earliest date	Jan 1953	System accepts the input	System accepts the input	The system behaves as expected
	latest date	December 2005	System accepts the input	System accepts the input	The system behaves as expected
	just before earliestD	Dec 1952	Does not take input of prior date	Does not take input of prior date	The system behaves as expected
	just after latest D	June 2023	Does not take input of later date	Does not take input of later date	The system behaves as expected

2. Register Students

Input Field	Test Case	Input	Expected outcome	Actual outcome	The system behaves as expected
Name	Min Length(1)	"A"	System accepts the input	System accepts the input	The system behaves as expected
	Max Length(40)	Str with len(40)	System accepts the input	System accepts the input	The system behaves as expected
	just below MinL(0)	""	System rejects the input	rejects and displays invalid input	The system behaves as expected
	just above MaxL(41)	Str with len(41)	does not take more than 40 inputs	does not take more than 40 inputs	The system behaves as expected
ID	Min length(1)	1	System accepts the input	System accepts the input	The system behaves as expected
	Max length (4)	4	System accepts the input	System accepts the input	The system behaves as expected
	just below Min(0)	0	System rejects the input	System does not save any details	The system behaves as expected
	just below Max(5)	5	does not take more than 4 inputs	does not take more than 4 inputs	The system behaves as expected
Number of Pictures	Min Length(8)	8	System accepts the input	System accepts the input	The system behaves as expected
	Max Length(8)	8	System accepts the input	System accepts the input	The system behaves as expected
	just below MinL(7)	7	System rejects the input	System does not save any details	The system behaves as expected
	just below MaxL(9)	9	does not take more than 8 inputs	does not take more than 8 inputs	The system behaves as expected
CGPA	Min CGPA(0.1)	0.1	System accepts the input	System accepts the input	The system behaves as expected
	Max CGPA(4.0)	4	System accepts the input	System accepts the input	The system behaves as expected
	just below Min(0.0)	0	Does not allow below 0.1 gpa	Does not allow below 0.1 gpa	The system behaves as expected
	just above Max(4.1)	4.1	Does not allow above 4.0 gpa	Does not allow above 4.0 gpa	The system behaves as expected

3. Feedback

Input Field	Test Case	Input	Expected outcome	Actual outcome	Analysis
Feedback length	Min Length(1)	"A"	System accepts the input	System accepts the input	The system behaves as expected
	Max Length(50)	Str with len(50)	System accepts the input	System accepts the input	The system behaves as expected
	just below MinL(0)	""	System rejects the input	rejects and displays invalid input	The system behaves as expected
	just above MaxL(51)	Str with len(51)	does not take more than 50 inputs	does not take more than 50 inputs	The system behaves as expected

4. Login Page

Input Field	Test Case	Input	Expected outcome	Actual outcome	Analysis
Userame length	Min Length(1)	"A"	System accepts the input	System accepts the input	The system behaves as expected
	Max Length(20)	Str with len(20)	System accepts the input	System accepts the input	The system behaves as expected
	just below MinL(0)	""	System rejects the input	System rejects the input	The system behaves as expected
	just above MaxL(21)	Str with len(21)	does not take more than 20 inputs	does not take more than 20 inputs	The system behaves as expected
Passwo rd length	Min Length(1)	"A"	System accepts the input	System accepts the input	The system behaves as expected
	Max Length(20)	Str with len(20)	System accepts the input	System accepts the input	The system behaves as expected
	just below MinL(0)	""	System rejects the input	System rejects the input	The system behaves as expected
	just above MaxL(21)	Str with len(21)	does not take more than 20 inputs	does not take more than 20 inputs	The system behaves as expected

MODULE	T1	R1	T2	R2	T3	R3		OVERALL
LOGIN	--	ERROR POP UP	"123....XYZ"	ERROR POP UP	{VALID}	PASSED		COMPLETED
REGISTER NEW	--	ERROR POP UP	"123....XYZ"	ERROR POP UP	{VALID}	ADDED STUDENT		COMPLETED
MARK ATTENDANCE	--	ERROR POP UP	"123....XYZ"	ERROR POP UP	{VALID}	MARKED ATTENDANCE		COMPLETED
SHOW ATTENDANCE	--	PASS	"123....XYZ"	PASS	{VALID}	DISPLAY ATTENDANCE		COMPLETED
FEEDBACK	--	ERROR POP UP	"123....XYZ"	PASS	{VALID}	FEEDBACK SUBMITTED		COMPLETED
ADD TEACHER	--	ERROR POP UP	"123....XYZ"	ERROR POP UP	{VALID}	ADDED TEACHER		COMPLETED
GRAPH	--	PASS	"123....XYZ"	PASS	{VALID}	DISPLAY GRAPH		COMPLETED

MODULE	DEVELOPER'S RATING	TEACHER'S RATING	INDEPENDENT SYSTEM	ADMIN'S RATING	STUDENT'S RATING	AVG. MODULE RATING
LOGIN	10	8	7	8	-	8.25
REGISTER NEW	9	8	8	-	-	8.33
MARK ATTENDANCE	7	6	7	-	9	7.25
SHOW ATTENDANCE	8	6	8	-	-	7
FEEDBACK	9	9	-	-	9	9
ADD TEACHER	3	-	8	8	-	8
GRAPH	10	10	10	9	-	9.75
Avg. Reviewer's Rating	8	7.83	8	8.33	9	8.23

Work Division

Name	Work
Ibtehaj Ali	<ul style="list-style-type: none"> - Content Writing and Documentation - Pair Programmer (Developer in second week) - File Handling fro Graphs - File Handling for Dashboard - Design for Feedback Form
Ahmed Abdullah	<ul style="list-style-type: none"> - Pair Programmer (Developer in first week) - Diagram Designer - Graph Designer - Feedback Form Designer - Improved Login Form

Lesson learnt by group

Working on a face recognition attendance system has taught us a lot of lessons. While our first iteration didn't go as planned, offering us a challenging start, we carried on through teamwork. The need to modify user stories on the go was hectic, yet it pushed us to remain flexible.

Through pair programming and open communication, we managed to overcome the obstacles and deliver a successful project in the end. This experience underscored the importance of clear planning and the value of adapting our methods as needed.

We learned that accuracy can vary depending on the situation. While high accuracy is good, we also need to balance it with speed, efficiency, and user experience. This helps us set realistic expectations for future projects.

Not all pretrained models are of good quality. We found that some models didn't meet our needs, so we learned to evaluate and choose pretrained models carefully.

We struggled with fine-tuning models in C#, which limited our ability to optimise the system effectively. This taught us the importance of learning how to fine-tune models in the chosen language to improve performance.

We learned that using the right technology for the task is crucial.

Our experience showed us that C# may not be the most efficient language for machine learning tasks. It lacks the robust support and libraries available in other languages such as Python

These lessons will help us do better in future projects as we keep improving. By working hard and staying united, we achieved our goals and created a functional and useful system. This system is more beneficial to the people who use it.
