Software Requirements Specification for

ENHANCED COURSE FEEDBACK SYSTEM

Version 1.0

Tanayça Ünlütürk

Eda Kocaman

Contents

1. Introduction

1.1 Purpose

This document presents the specification for the application called “Enhanced Course Feedback System.” This application aims to improve the quality of courses by allowing interaction between both teachers and students.

1.2 Scope

Enhanced Course Feedback System is a mobile application that allows teachers to submit a form to the system and collect feedback from students and send it to ChatGPT to receive the highlights of the feedback collected.

1.3 Definitions, Acronyms, and Abbreviations

ECFS: Enhanced Course Feedback System

GPT: ChatGPT

SRS: Software Requirement Specification

API: Application Programming Interface

1.4 References

This document adheres to the standards outlined in the IEEE SRS template.

1.5 Overview

This software requirements document serves as a comprehensive guide for the development of ECFS. It outlines the purpose and scope of the project, defining its boundaries and specifying the features and functionalities to be included. The document also provides a curated list of definitions, acronyms, and abbreviations for clarity, ensuring a consistent understanding of terminology among stakeholders. This overview sets the stage for a detailed exploration of the project's general and specific requirements, encompassing aspects such as product perspective, user characteristics, functional and non-functional requirements. With a focus on clarity and precision, this document aims to serve as a foundational reference for all project stakeholders, fostering a unified vision and understanding throughout the project lifecycle.

2. General Description

2.1 Product Perspective

This product is a stand-alone mobile application and a self-contained product. This SRS does not contain any larger system or the functionality of the ECFS, nor any interface of it.

2.2 Product Functions

ECFS App allows users to perform following functions:

- Design Feedback Request Form
- Generate a unique URL for the feedback request form
- Send students the feedback request form via e-mail
- Summarize feedback through GPT
- List previous feedback

2.3 User Characteristics

ECFS app is designed for the use of Çankaya University teachers and students who need university accounts to use the application to request and receive feedback about courses opened in the semesters. There is no need of technical expertise to use the product.

2.4 General Constraints

ECFS is only designed for specific usage of Çankaya University students' and teachers' need of feedback on courses. The teachers must have .edu account extensions to use the application and students can be directed to the feedback request form in the app via the URL sent to the student e-mail.

The app will run on both smartphones and tablets running Android or iOS.

The generated URL will be active for 48 hours for students to fill in and submit the feedback form.

There will be GPT API integration to the ECFS that runs after 48 hours to summarize collected feedback.

Concerning databases, MySQL will be employed for the application.

For language, English and Turkish languages will be supported.

Since the app is for specific usage, it is not scalable or does not contain any other usages.

Security will be provided through third party companies.

The maintenance will also be provided through third party companies.

The system should be able to handle 1000 students at a given time.

There should not be memory leaks and memory usage should be optimized.

ECFS will take overall a week to develop.

The budget will be limited to 10.000 $ in total.

2.5 Assumptions and Dependencies

This application is only meant for Çankaya University teachers and students usage.

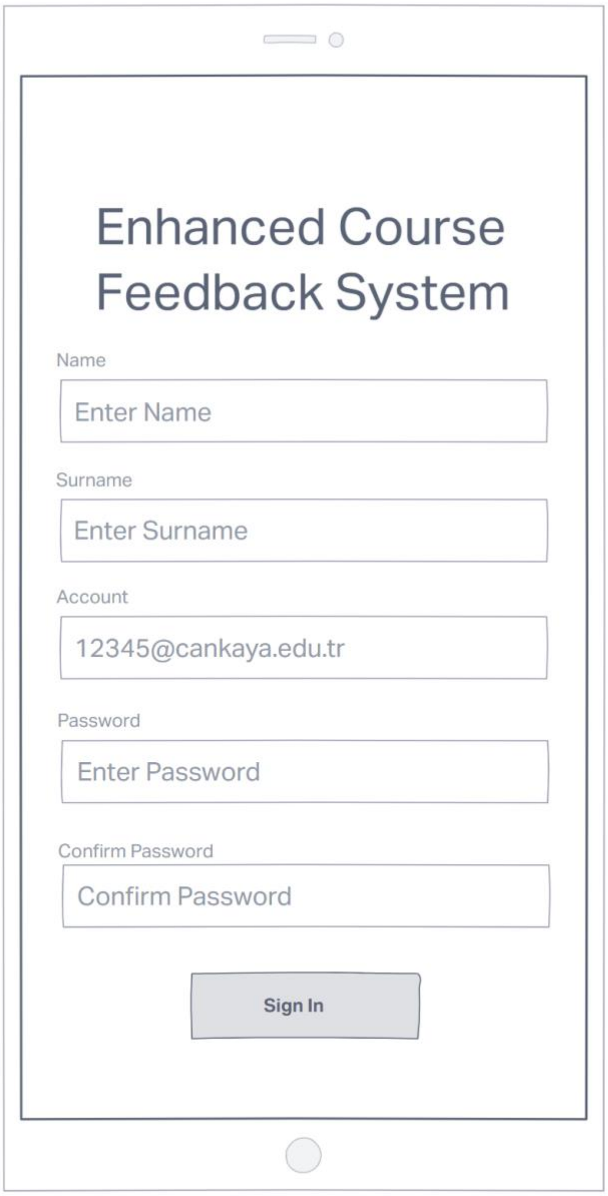
The feedback summary will be acquired through ChatGPT integration which necessitates an API for this usage.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.1.1 Teacher Sign in Page



Enhanced Course Feedback System

Name

Enter Name

Surname

Enter Surname

Account

12345@cankaya.edu.tr

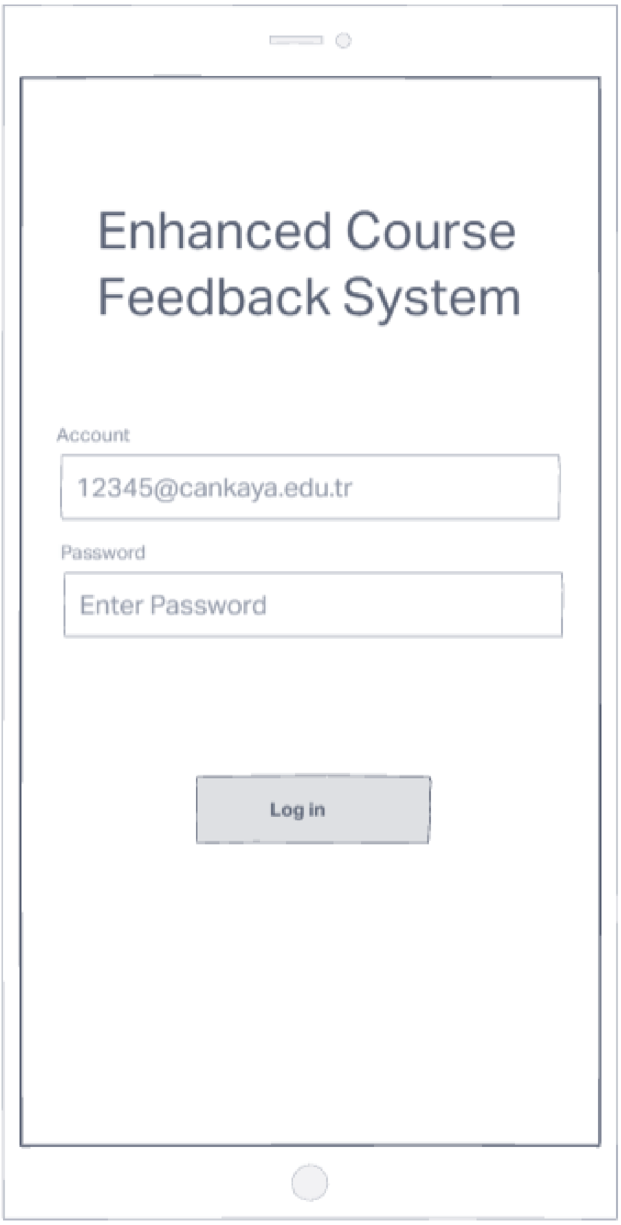
Password

Enter Password

Confirm Password

Confirm Password

3.1.1.2 Teacher Log in Page



Enhanced Course Feedback System

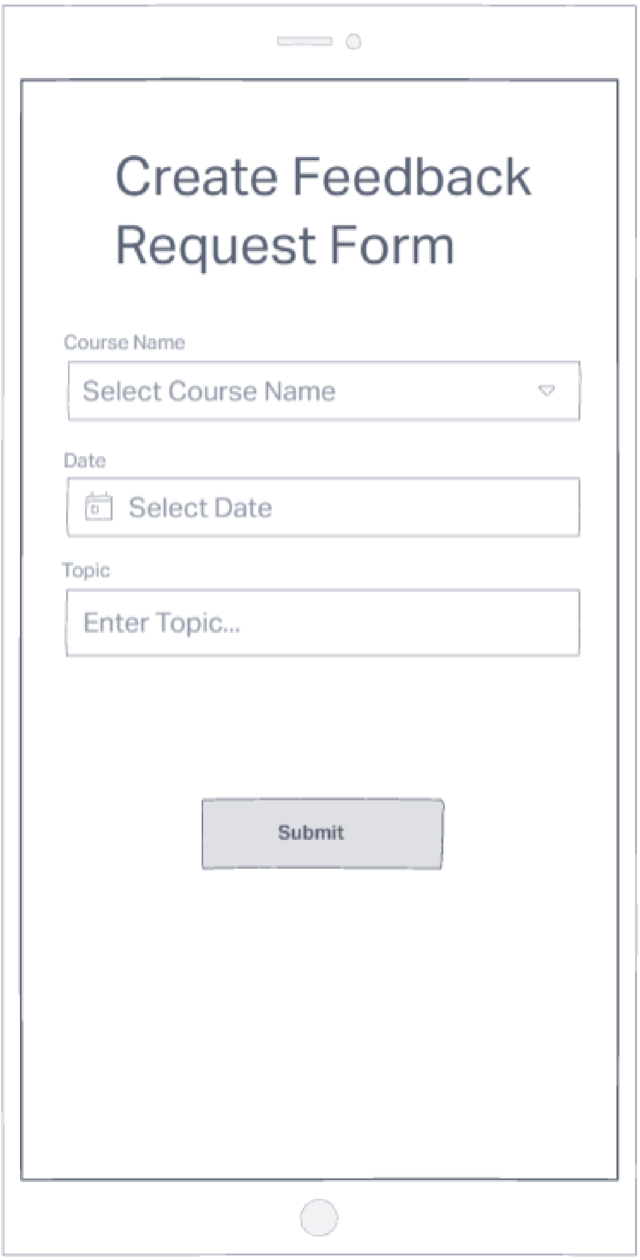
Account

12345@cankaya.edu.tr

Password

Enter Password

3.1.1.3 Teacher Feedback Request Form



Create Feedback Request Form

Course Name

Select Course Name

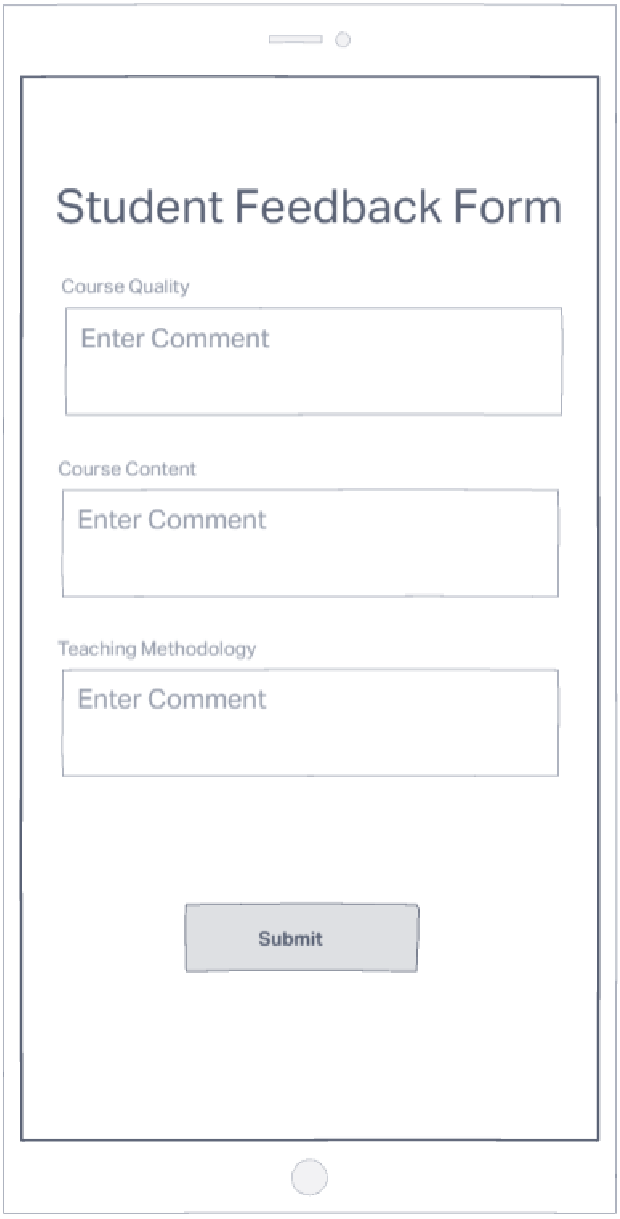
Date

Select Date

Topic

Enter Topic...

3.1.1.4 Student Feedback Form



Student Feedback Form

Course Quality

Enter Comment

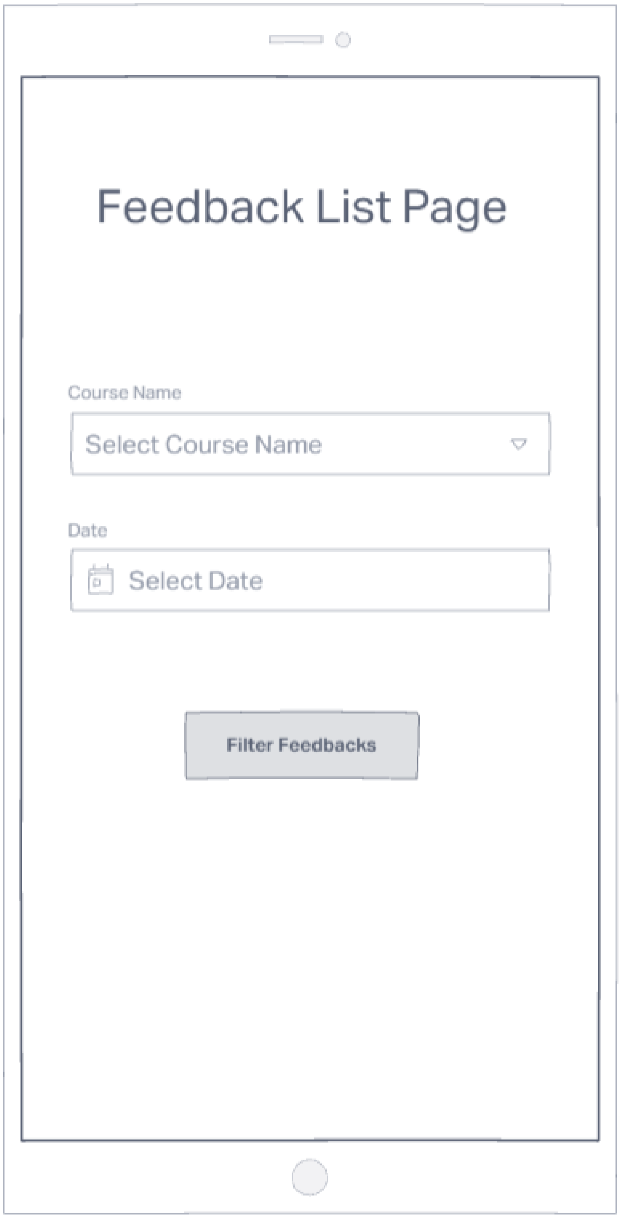
Course Content

Enter Comment

Teaching Methodology

Enter Comment

3.1.1.5 Feedback List Page



Feedback List Page

Course Name

Select Course Name

Date

Select Date

3.1.2 Hardware Interfaces

The Application will run on both smartphones and tablets.

ECFS will work in the same way on both tablets and smartphones concerning screen size.

ECFS requires Wi-Fi or mobile data connection.

3.1.2 Software Interfaces

The application will run on both IOS and Android operating systems without version limitations.

Concerning databases, MySQL is used with JSON file format.

For teachers' authentication, Firebase Authentication will be used.

Data storage will be provided through third party server companies.

The application will support both Turkish and English language.

3.1.4. Communication Interfaces

Network protocols consist of HTTPS.

ECFS works integrated with GPT; feedback collected from students will be directed to GPT API to summarize and highlight important aspects of feedback. To do that, JSON data exchange format will be employed.

For the authentication Firebase Authentication will be integrated to the system.

3.2 Functional Requirements

Diagram 1: Use Case Diagram

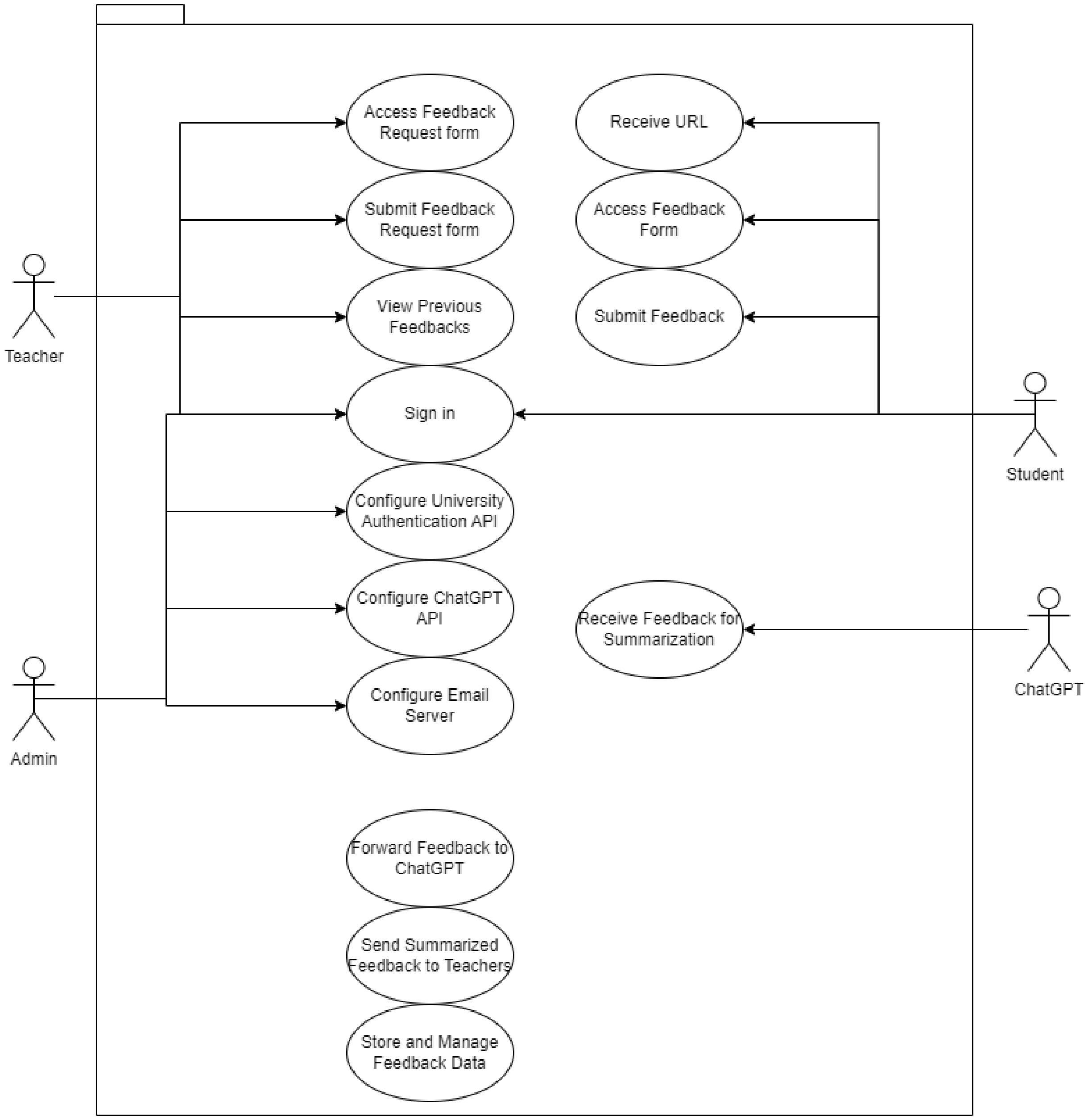

Access Feedback Request form
Receive URL
Submit Feedback Request form
Access Feedback Form
Teacher
View Previous Feedbacks
Submit Feedback
Sign in
Student
Configure University Authentication API
Configure ChatGPT API
Receive Feedback for Summarization
ChatGPT
Admin
Configure Email Server
Forward Feedback to ChatGPT
Send Summarized Feedback to Teachers
Store and Manage Feedback Data

Diagram 2: Activity Diagram

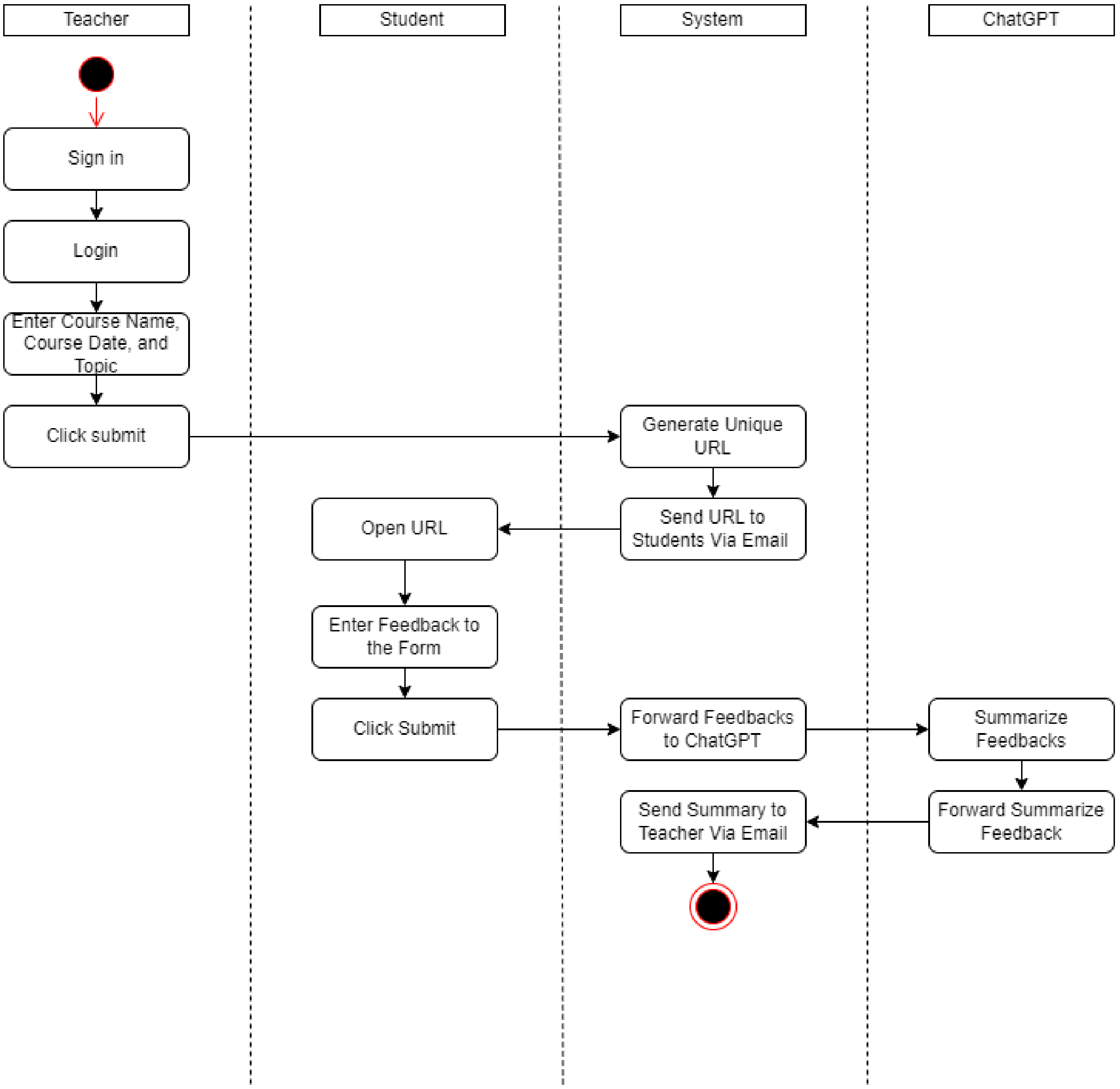

Teacher
Student
System
ChatGPT
Sign in
Login
Enter Course Name, Course Date, and Topic
Click submit
Generate Unique URL
Send URL to Students Via Email
Open URL
Enter Feedback to the Form
Click Submit
Forward Feedbacks to ChatGPT
Summarize Feedbacks
Forward Summarize Feedback
Send Summary to Teacher Via Email

Diagram 3: Sequence Diagram

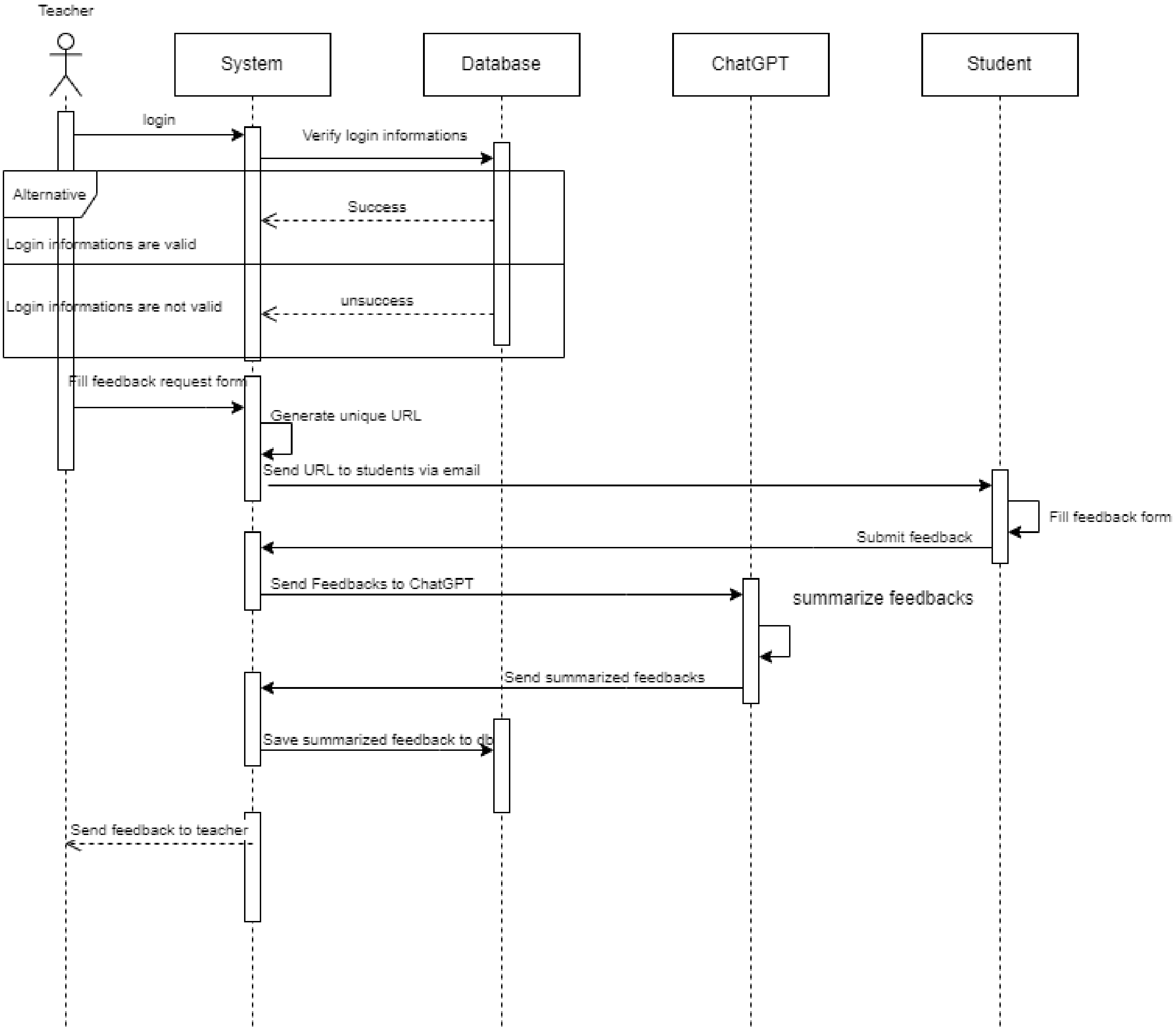
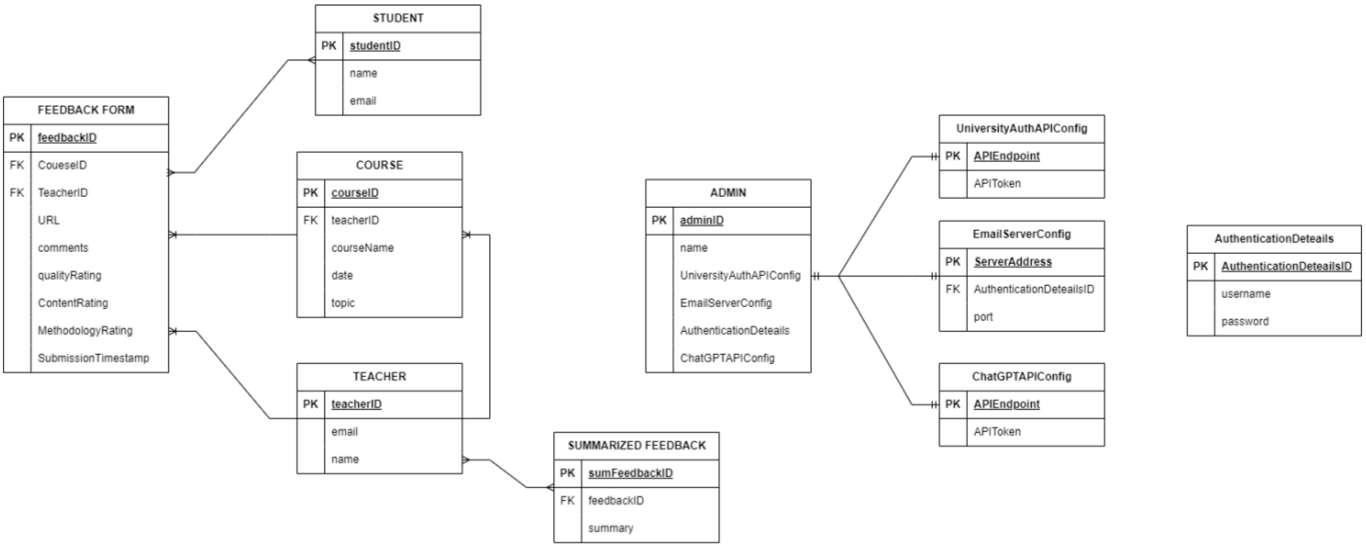

Teacher
System
Database
ChatGPT
Student
login
Verify login informations
Alternative
Success
Login informations are valid
unsuccess
Login informations are not valid
Fill feedback request form
Generate unique URL
Send URL to students via email
Fill feedback form
Submit feedback
Send Feedbacks to ChatGPT
summarize feedbacks
Send summarized feedbacks
Save summarized feedback to db
Send feedback to teacher

Diagram 4: Data Model



3.2.1. Teacher Sign In

3.2.1.1 Description and Priority

The ECFS will allow teachers to sign into the system by entering a name, surname, university account and creating a password. This is a high priority feature.

3.2.1.2 Stimulus/Response Sequences

1. User: Enters name, surname, university account, password, and confirmation password to the designated fields.
2. User: Clicks the sign in button.
3. System: Saves user info to the database.

3.2.1.3 Functional Requirements for Teacher Sign In

REQ. 1: User will be able to enter name, surname, university account, password, and confirmation password to the designated fields.

REQ. 2: The system should check if the user has filled in necessary fields.

REQ. 3: The system should check whether the user exists in the database.

REQ. 4: If the user does not exist in the database, the system should add user to database.

3.2.2 Teacher Log In

3.2.2.1 Description and Priority

The ECFS will allow teachers to log in to the system via their university accounts and password. This is a high priority feature.

3.2.2.2 Stimulus/Response Sequences

1. User: Enters name, surname, university account, password, and confirmation password to the designated fields.
2. User: Clicks the sign in button.
3. System: Saves user info to the database.

3.2.2.3 Functional Requirements for Teacher Log In

REQ. 1: User will be able to enter university account and password to the designated fields.

REQ. 2: The system should check if the user has entered correct information.

REQ. 3: The system should allow the user to log in.

3.2.3. Create a Feedback Request Form

3.2.3.1 Description and Priority

The user can create a feedback request form for students to fill in by choosing course name, course date and writing down the topic. This is a high priority feature.

3.2.3.2 Stimulus/Response Sequences

1. User: Enters course name, selects date from calendar and writes a topic to the necessary fields, clicks submit button.
2. System: Generates a unique URL for the request form with a 48-hour limit.
3. System: Sends URL to student e-mails.

3.2.3.3 Functional Requirements for Create a Feedback Request Form

REQ. 1: The user should be able to enter course name, select date from calendar and write a topic to the necessary fields.

REQ. 2: The user should be able to click the submit button.

REQ. 3: The system should generate unique URL to send the students that becomes unavailable after 48 hours.

REQ. 4: The system should send URL to students' university e-mails.

3.2.4. Fill in Student Feedback Form

3.2.4.1 Description and Priority

The students can click on the sent URL to their e-mail address and be directed to the Student Feedback Form page and enter their comments about course's quality, content, and teaching methodology. This is a high priority feature.

3.2.4.2 Stimulus/Response Sequences

1. User: Clicks on the URL that is sent via e-mail.
2. System: Directs user to the app, to the Student Feedback Page.
3. User: Enters comments in necessary fields.
4. System: Saves entered info to the database.

3.2.4.3 Functional Requirements for Fill in Student Feedback Form

REQ. 1: User should be able to click the sent URL.

REQ. 2: User should be directed to Student feedback page.

REQ. 3: User able to fill in the necessary forms.

REQ. 4: System should save the entered information to the database.

3.2.5. ChatGPT Interaction

3.2.5.1 Description and Priority

The system should send collected feedback to the GPT and receive summaries and highlighted texts from it. This is a high priority feature.

3.2.5.2 Stimulus/Response Sequences

1. System: Sends collected feedback to the GPT through API.
2. System: Receives summaries and highlights from GPT through API.

3.2.5.3 Functional Requirements for ChatGPT Interaction

REQ. 1: The system should be able to send collected feedback to GPT through API.

REQ. 2: The system should be able to collect summaries from GPT through API.

3.2.6. Feedback Delivery

3.2.6.1 Description and Priority

Summarized feedback will be promptly sent to respective teachers via e-mail. This is a medium priority feature.

3.2.6.2 Stimulus/Response Sequences

1. System: Sends collected feedback summaries to the teacher via e-mail.

3.2.6.3 Functional Requirements for ChatGPT Interaction

REQ. 1: The system should be able to send collected feedback summaries to the teacher via e-mail.

3.2.7. See List of Previous Feedback

3.2.7.1 Description and Priority

Teachers can view the previous feedback via a list page. They can filter the feedback with course name and date. This is a high priority feature.

3.2.7.2 Stimulus/Response Sequences

1. User: Clicks on View Feedback List.

2. System: Directs user to Feedback List Page.

3. User: Clicks on Filter List.

4. User: Selects filter options by Course Name and Date.

5. System: Filters Feedback by selected criteria.

3.2.7.3 Functional Requirements for See List of Previous Feedback

REQ. 1: User should be able to click on View Feedback List.

REQ. 2: The system should be able to direct user to Feedback List Page.

REQ. 3: User should be able to filter feedback by course name and date.

REQ. 4: The system should be able to filter feedback and show filtered list to the user.

3.2.8. Configure University Authentication

3.2.8.1 Description and Priority

Admin will be able to configure the university authentication API connection info and credentials. This is a medium priority feature.

3.2.8.2 Stimulus/Response Sequences

1. The admin initiates the configuration process by accessing the admin settings or configuration section of the application.
2. The system displays the configuration interface specifically designed for setting up the university authentication API connection.
3. The admin enters the required information for the university authentication API connection.
4. The system validates the entered information to ensure its correctness and completeness. If validation is successful, the admin is prompted to save the configuration.
5. The administrator confirms and saves the entered configuration.

6. The application stores the configured university authentication API connection information securely.
7. The admin provides the necessary credentials for accessing the university authentication API.
8. The system validates the provided credentials. If they are valid, the admin is prompted to save the credentials.
9. The admin confirms and saves the entered credentials.
10. The application securely stores the provided credentials for future use.

3.2.8.3 Functional Requirements for Configure University Authentication

REQ 1: The system shall provide an administrative interface for configuring the university authentication API connection information and credentials.

REQ 2: The admin shall be able to access the configuration interface through the application's administrative settings or configuration section.

REQ 3: The application shall respond by displaying a user-friendly form for entering university authentication API connection details.

REQ 4: The admin shall be able to input necessary connection information, including API endpoint URLs, authentication methods, and any other required parameters.

REQ 5: The system shall validate the entered connection information to ensure correctness and completeness before allowing the administrator to proceed.

REQ 6: The admin shall have the capability to save the configured university authentication API connection information.

REQ 7: The application shall securely store the configured university authentication API connection information for future use.

REQ 8: The admin shall be able to input the required credentials for accessing the university authentication API.

REQ 9: The system shall validate the entered credentials and only allow saving if they are deemed valid.

REQ 10: The admin shall have the capability to save the provided credentials for accessing the university authentication API.

3.2.9. Configure ChatGPT API Connection

3.2.9.1 Description and Priority

Admin will be able to configure the ChatGPT API connection info and credentials. This is a high priority feature.

3.2.9.2 Stimulus/Response Sequences

1. The admin accesses the admin settings or configuration section of the application.

2. The application responds by displaying the configuration interface specifically designed for setting up the ChatGPT API connection.

3. The admin enters the required information for the ChatGPT API connection.

4. The application validates the entered information to ensure its correctness and completeness.

5. The administrator confirms and saves the entered configuration.

6. The application securely stores the configured ChatGPT API connection information.

7. The administrator provides the necessary credentials for accessing the ChatGPT API.

8. The application validates the provided credentials. If they are valid, the admin is prompted to save the credentials.

9. The administrator confirms and saves the entered credentials.

10. The application securely stores the provided credentials for future use.

3.2.9.3 Functional Requirements for Configure University Authentication

REQ. 1: The system shall provide an administrative interface within the application to facilitate the configuration of the ChatGPT API connection information and credentials.

REQ 2: The admin shall be able to access the ChatGPT API configuration interface through the application's administrative settings or configuration section.

REQ. 3: The system shall display a user-friendly form prompting the administrator to input ChatGPT API connection details, including API endpoint URLs, authentication methods, and any other relevant parameters.

REQ. 4: The system shall validate the entered ChatGPT API connection information to ensure correctness and completeness before allowing the administrator to proceed with the configuration.

REQ. 5: The administrator shall have the capability to save the configured ChatGPT API connection information.

REQ. 6: The system shall securely store the configured ChatGPT API connection information for future use.

REQ. 7: The administrator shall be able to input the required credentials for accessing the ChatGPT API.

REQ. 8: The system shall validate the provided ChatGPT API credentials. Configuration should only proceed if the credentials are valid.

REQ. 9: The administrator shall have the capability to save the provided credentials for accessing the ChatGPT API.

REQ. 10: The system shall securely store the provided ChatGPT API credentials for future use.

3.2.10. Configure E-mail Server Connection

3.2.10.1 Description and Priority

Admin will be able to configure the e-mail server connection info and credentials. This is a high priority feature.

3.2.10.2 Stimulus/Response Sequences

1. The administrator initiates the configuration process by accessing the admin settings or configuration section of the application.

2. The application responds by displaying the configuration interface specifically designed for setting up the e-mail server connection.

3. The administrator enters the required information for the e-mail server connection, including server address, port number, and any other relevant details.

4. The application validates the entered information to ensure its correctness and completeness. If validation is successful, the admin is prompted to save the configuration.

5. The administrator confirms and saves the entered e-mail server configuration.

6. The application securely stores the configured e-mail server connection information.

7. The administrator provides the necessary credentials for accessing the e-mail server.

8. The application validates the provided credentials. If they are valid, the admin is prompted to save the credentials.

9. The administrator confirms and saves the entered credentials.

10. The application securely stores the provided e-mail server credentials for future use.

3.2.10.3 Functional Requirements for Configure E-mail Server Connection

REQ. 1: The system shall provide an administrative interface within the application to facilitate the configuration of the e-mail server connection information and credentials.

REQ 2: The administrator shall be able to access the e-mail server configuration interface through the application's administrative settings or configuration section.

REQ. 3: Upon accessing the configuration interface, the system shall display a user-friendly form prompting the administrator to input e-mail server connection details, including server address, port number, and any other relevant parameters.

REQ. 4: The system shall validate the entered e-mail server connection information to ensure correctness and completeness before allowing the administrator to proceed with the configuration.

REQ. 5: The administrator shall have the capability to save the configured e-mail server connection information.

REQ. 6: The system shall securely store the configured e-mail server connection information for future use.

REQ. 7: The administrator shall be able to input the required credentials for accessing the e-mail server.

REQ. 8: The system shall validate the provided e-mail server credentials. Configuration should only proceed if the credentials are valid.

REQ. 9: The administrator shall have the capability to save the provided credentials for accessing the e-mail server.

REQ. 10 The system shall securely store the provided e-mail server credentials for future use.

4. Other Non-Functional Requirements

4.1 Performance Requirements

1. The system shall respond to user interactions (e.g., submitting feedback, loading pages) within 2 seconds under normal operating conditions.
2. The system shall support a minimum of 500 concurrent users without a degradation in performance.
3. Pages within the application shall load within 3 seconds.
4. Database queries shall execute and return results within 1 second, ensuring efficient data retrieval for various system operations.
5. The system shall maintain a high level of performance, with no more than 5 minutes of downtime per month, excluding scheduled maintenance periods.
6. The application shall utilize no more than 80% of the available server CPU and memory resources during peak usage times.
7. Batch processes, such as generating reports or processing feedback data, shall be completed within 30 minutes for a data set containing up to 100,000 records.
8. The system shall support file uploads and downloads with speeds averaging 2 MB per second for files up to 10 MB in size.
9. Mobile applications shall respond to user interactions within 3 seconds and provide an optimal user experience on devices with varying screen sizes.
10. Security measures (e.g., encryption, authentication) shall introduce no more than a 5% overhead on system performance.
11. Teachers receive summarized feedback promptly, enabling quick adjustments to course content and methodologies.

4.2 Security Requirements

1. The system shall implement a secure and robust authentication mechanism to verify the identity of users accessing the system.
2. User passwords shall be stored securely using industry-standard encryption algorithms.
3. The system shall enforce role-based access control (RBAC) to ensure that users can only access information and perform actions appropriate to their roles.
4. Administrators shall have the authority to assign and revoke access privileges based on user roles.

5. Communication between clients and the server, as well as between internal system components, shall be encrypted using industry-standard protocols (e.g., TLS/SSL).
6. The system shall maintain an audit trail to log all significant events, including user logins, data modifications, and security-related incidents.
7. User inputs and sensitive data transmitted over the network shall be encrypted to prevent eavesdropping and data interception.
8. The system shall implement secure session management to protect against session hijacking and ensure the confidentiality of user sessions.
9. The system shall implement mechanisms to ensure the integrity of stored data, preventing unauthorized modification or tampering.
10. The system shall enforce password policies, including minimum complexity requirements, expiration periods, and account lockout mechanisms.
11. Regular data backups shall be performed to prevent data loss in the event of system failures, and backup procedures shall be periodically tested for reliability.
12. The system shall have an incident response plan in place to address security incidents promptly, including procedures for reporting, investigating, and mitigating security breaches.
13. Physical access to servers and other critical infrastructure components shall be restricted to authorized personnel, and appropriate physical security measures shall be in place.
14. The system shall comply with relevant data protection regulations, and measures shall be in place to ensure the privacy and protection of user data.

4.3 Software Quality Attributes

1. The system shall be highly reliable; it shall process and store data, with a low probability of errors or data corruption.
2. The system shall provide 48 hours of availability after the form is created.
3. The user interface shall be intuitive and user-friendly, with clear navigation and a minimal learning curve for users.
4. Efficient feedback mechanisms foster a better learning environment and stronger teacher-student relationships.
5. The system shall be designed with modular and well-documented code to facilitate easy maintenance and updates.

6. The system shall integrate seamlessly with other relevant systems and APIs, fostering interoperability and data exchange.
7. The system shall be designed with testability in mind, allowing for effective testing of individual components and overall system behavior.
8. The system shall comply with relevant legal and regulatory requirements, including data protection and privacy regulations.
9. The user interface and system behavior shall be consistent across different modules and functionalities.
10. The system shall demonstrate resilience by recovering gracefully from failures and disruptions, minimizing downtime and data loss.
11. The system shall maintain detailed and tamper-evident audit logs to facilitate tracking and auditing of user actions and system events.