

|  |  |  |
| --- | --- | --- |
| **NO** | **ID** | **NAME** |
| 1 | 241UC240L4 | LOK HON YONG |
| 2 | 241UC2411S | IVAN LIM ZHE MIN |
| 3 | 1211303991 | MUHAMMAD AMMAR AJWAD BIN MUHAMMAD RUSHDAN |

Software Requirements Specification (SRS)

# Learning Analytics System (LAS)

# Version: 1.0

# Date: 03/09/2024

# 1. Introduction

## 1.1 Purpose

This SRS outlines the requirements needed for the LAS in development. The following document shall be to the benefit of the development team, project stakeholders, and all involved in this project. This is aimed at ensuring that all requirements are understood and met to successfully deliver the LAS.

## 1.2 Scope

This LAS shall be utilized as a virtual platform by the educational institutions to administer and deliver the courses. General system features will include user management, course creation, assignment submissions, grading, and reporting. The system is intended for students, lecturers, and administrators. Integration with existing systems shall include SIS and LTI providers. Development of new tools for content creation is out of scope.

## 1.3 Definitions, Acronyms, and Abbreviations

LAS: Learning Analytics System

SIS: Student Information System

API: Application Programming Interface

LTI: Learning Tools Interoperability

## 1.4 References

ISO/IEC/IEEE 29148:2018 — Systems and software engineering — Life cycle processes — Requirements engineering.

## 1.5 Overview

This document is organized as follows:

Section 2: The overall description of the LAS, including product perspective, features, user classes, operating environment.

Section 3: System Requirements To specify system requirements both functional and non-functional.

Section 4: Quality Attributes Non-functional requirements regarding reliability, security, maintainability, etc.

Section 5: Other Requirements Requirements which cannot be put into one of the headings above.

Section 6: Appendices - A Glossary of the system, Abbreviations, and a History of Documents.

# 2.Overall Description 2.1Product perspective

The LAS exists within a greater ecosystem of educational tools and platforms. The system will interface with the existing SIS and other LTI-compliant tools to provide an integrated user experience. The system shall be web-based with support for mobile devices utilizing responsive design.

## 2.2 Product Features

User Management: Provide management of accounts for students, lecturers, and administrators.

Course Management: Allow for creation, modification, and deletion of courses.

Assignment Submission: Allow students to submit assignments online.

Grading System: Lecturers to grade assignments and exams.

Reporting: Reports on student performance and course progress.

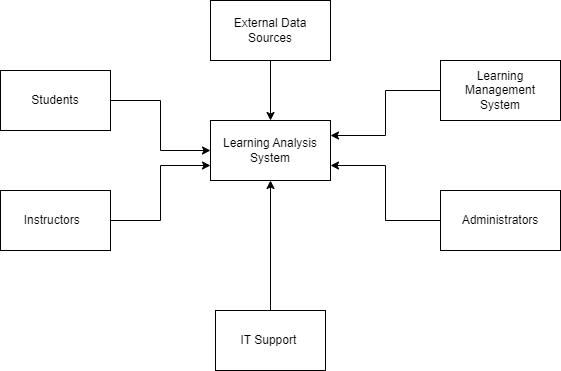
Notifications: Inform users about due dates, grades, and any other important information.

## 2.3 User Characteristics

Students: Major users, who will be able to view course materials, submit assignments, and check grades.

Lecturers: Users who shall lead course creation, disseminate content, and conduct student grading.

Administrators: Users responsible for the administration of the LAS, including the management of user accounts and maintenance of system integrity.

*Figure 1.0 LAS Context Diagram*

# 3.System Requirements 3.1 Functional requirements

1. Login/Registration Function:
   1. Register
      1. The system shall provide the user with a Student/Lecturer ID with password.
   2. Login
      1. The system shall prompt the user to login with Student/Lecturer ID along with password.
      2. If the user forgets their password, the "Forgot Password" feature shall be implemented to allow users to reset their passwords securely.
2. Data Integration Function:
   1. The system shall gather data on student interactions, assignment submissions, quiz results, and course participation.
3. Analytics and Reporting Function:
   1. The system shall analyze student performance, engagement, and learning outcomes across courses.
   2. The system shall provide real time and historical reports to admin, lecturer and students.
4. Dashboard Function:
   1. The system shall provide lecturers with actionable insights to directly inform teaching strategies, class adjustments and targeted interventions that meet students need.
5. Real-time Chat Function:
   1. The system might provide a chat box for users including students and lecturer to communicate with each other.
6. Feedback Function:
   1. The system might provide feedback for students to give feedback to tie those directly to course elements can be valuable.

# 3.2 Quality requirements

1. The loading time of the system shall not be over 2 seconds
2. The codebase should be modular and well-documented, allowing new developers to understand and make changes within a week of onboarding.
3. The system should be able to exchange data with other systems using JSON.
4. The system should have an uptime of 99.9% over a year, with no single downtime period exceeding 10 minutes.

# 3.3 Constraints

1. The system is limited to MMU students and staff only.
   1. This system requires authentication mechanisms (Login & Register) that verify users against the MMU student and staff databases.

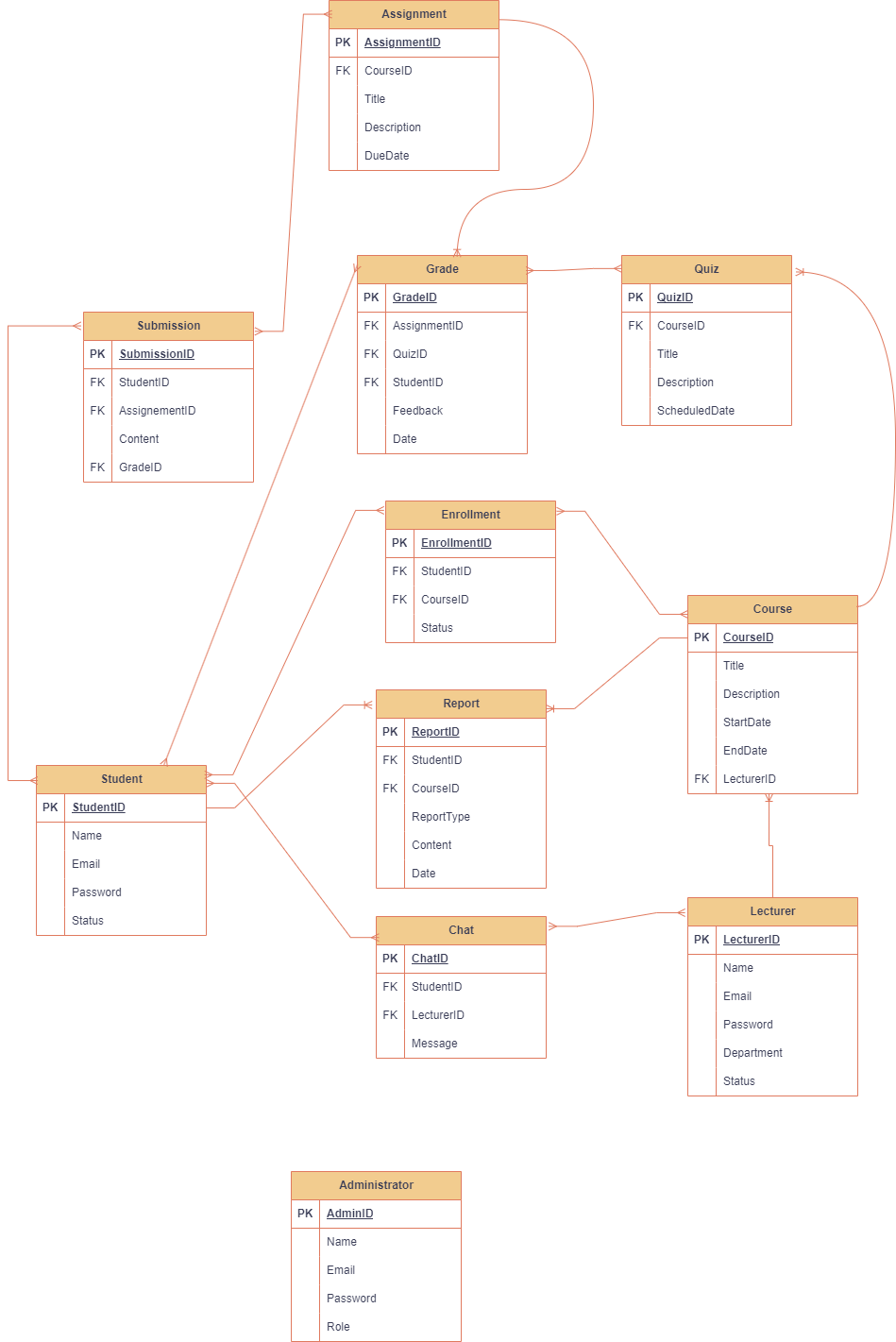
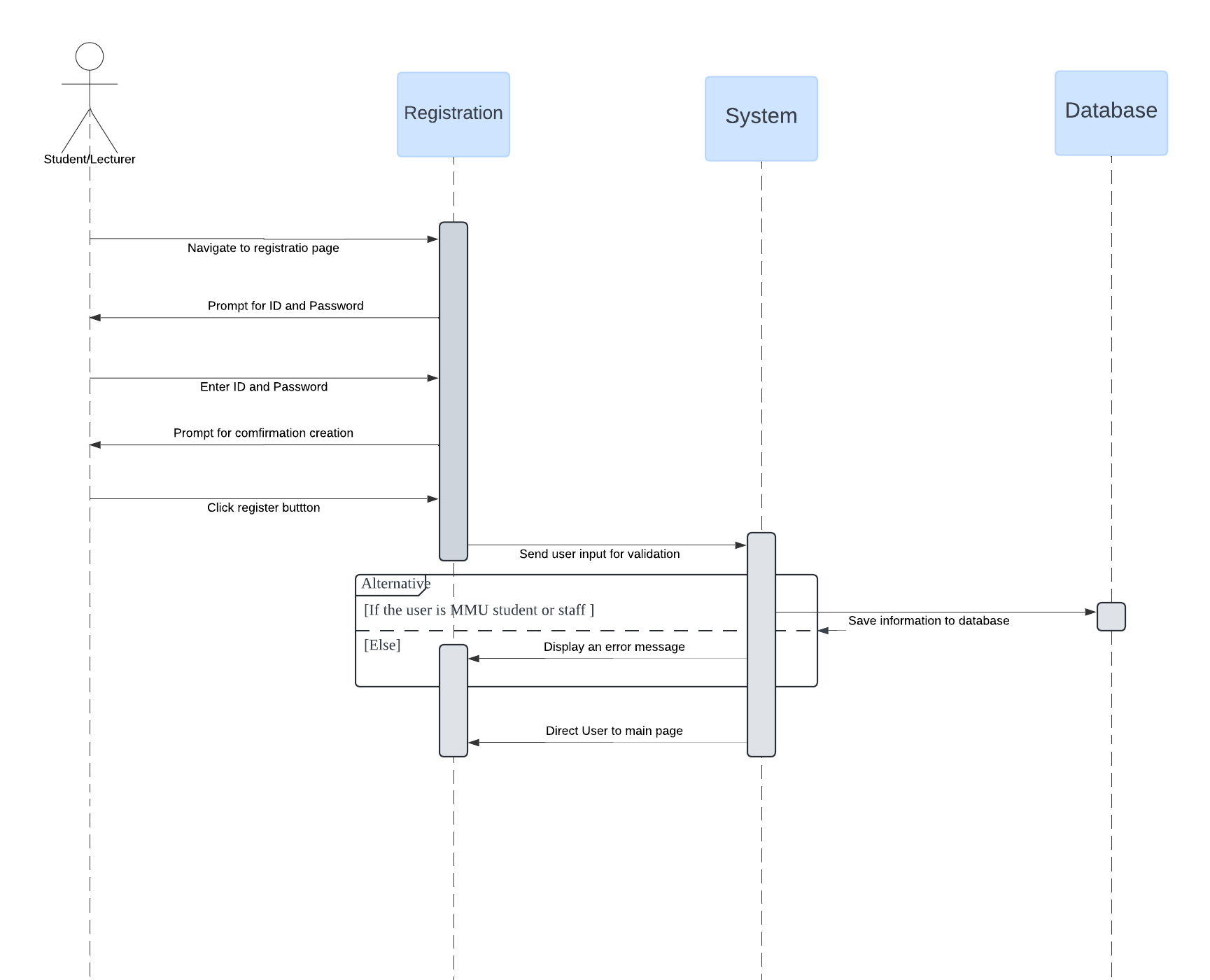


Figure 1.1

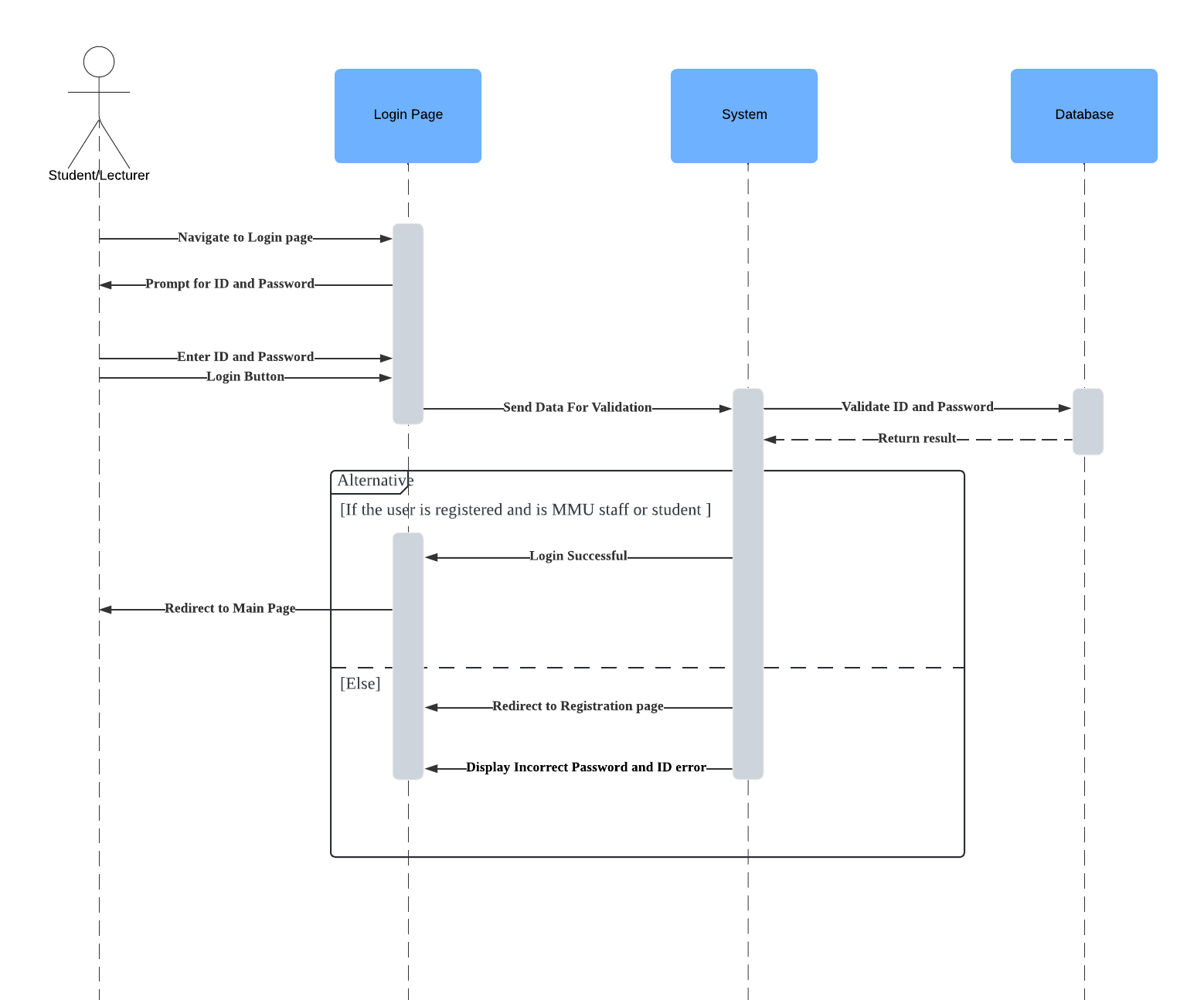
# Use Cases

|  |  |
| --- | --- |
| Use Case ID/Name | UC1/Register user |
| Description | Student and Lecture can register into the System |
| Precondition | All users must be either MMU student or Staff |
| Postcondition | * Registered users will have their information saved on the database. |
| Main Scenario | 1. The user navigates to the registration page of the application. 2. The system prompts the user to enter his/her student/lecturer ID and password to register. 3. The system prompts the user to create a password that meets the security requirements (e.g., minimum 8 characters, including a mix of uppercase, lowercase, numbers, and special characters). 4. The user clicks the "Register" button to complete the registration. 5. The system will validate the user’s input upon register before directing to the main page. 6. The registration process is complete, and the user is now registered in the system. |
| Alternate Scenario |  |
| Exception Scenario | 1. The user enters a weak password (does not meet the security requirements).    1. The system detects the weak password and displays an error message. |



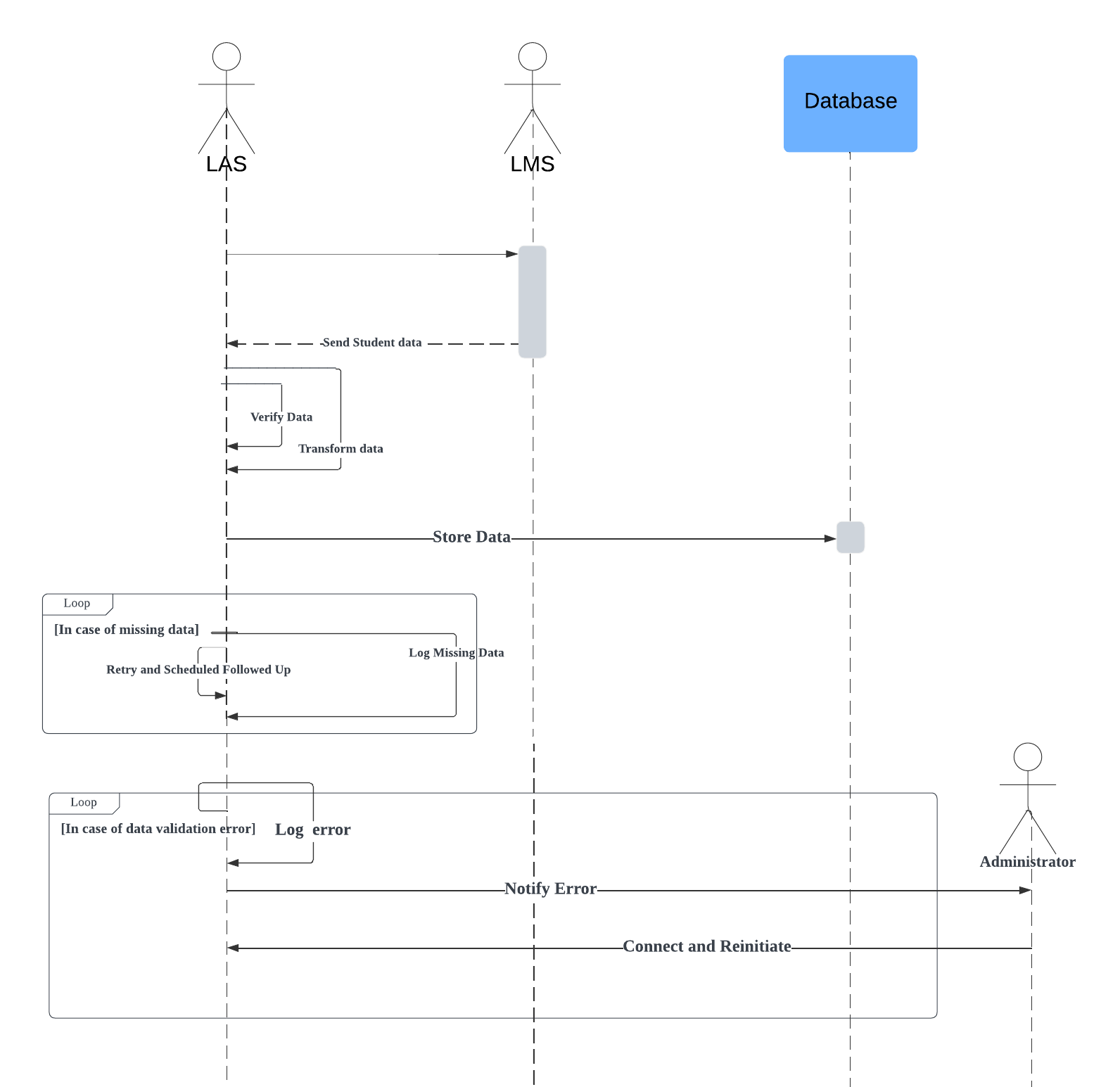
*Figure 1.1.1*

|  |  |
| --- | --- |
| Use Case ID/Name | UC2/Login user |
| Description | Student and Lecturer can log into the System |
| Precondition | All users must be either MMU student or Staff |
| Postcondition | * The users will redirect to the main page. |
| Main Scenario | 1. The user navigates to the Login page of the application. 2. The system prompts the user to enter his/her student/lecturer ID and password to login. 3. The user clicks the "Login" button to complete the Login. 4. The system will validate the user’s input upon login before directing to the main page. 5. The login process is complete, and the user is now registered in the system. |
| Alternate Scenario | 1. The user has not registered.    1. The system navigates the user to register page to register.    2. After the user registered and was able to login to the system. |
| Exception Scenario | 1. The user enters an incorrect password.    1. The system displays an error message and prompts the user to try again or reset their password. 2. The user enters an incorrect ID.    1. The system displays an error message and prompts the user to try again. |



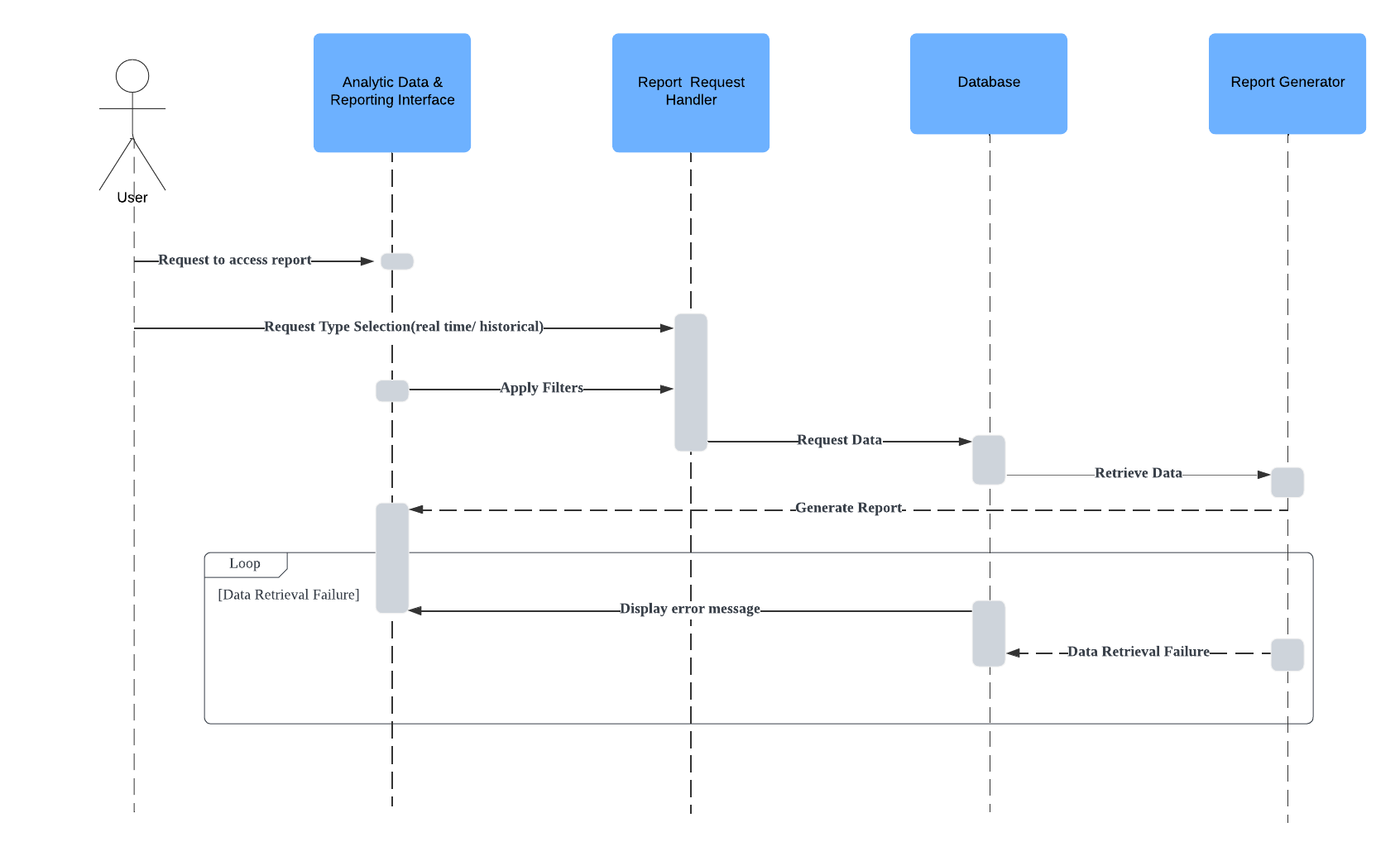
*Figure 1.1.2*

|  |  |
| --- | --- |
| Use Case ID/Name | UC3/Integrate data |
| Description | The system shall gather data on student interactions, assignment submissions, quiz results, and course participation. |
| Precondition | The Learning Analytics System (LAS) is operational and accessible, and the system has the necessary permissions and access credentials to retrieve data from the LMS. |
| Postcondition | * The system successfully stores the gathered data on student interactions, assignment submissions, quiz results, and course participation in its database. |
| Main Scenario | 1. The system initiates the data integration process at a scheduled time or based on a trigger event including new assignment submission or course update 2. The system retrieves data on student interactions, assignment submissions, quiz results, and course participation. 3. The system verifies that all necessary data fields are present and correct. 4. The system transforms the data into a format suitable for its internal database. 5. The system saves the validated and transformed data into its database. |
| Alternate Scenario | 1. The system logs the missing data and generates a notification.    1. The system retries the data request for the missing data types or schedules a follow-up data integration process. |
| Exception Scenario | 1. Data Validation Error    1. Retry the data retrieval process if the issue is suspected to be on the user side.    2. Administrators see the logged errors, correct any data issues if needed, and reinitiate the data integration process if necessary. |



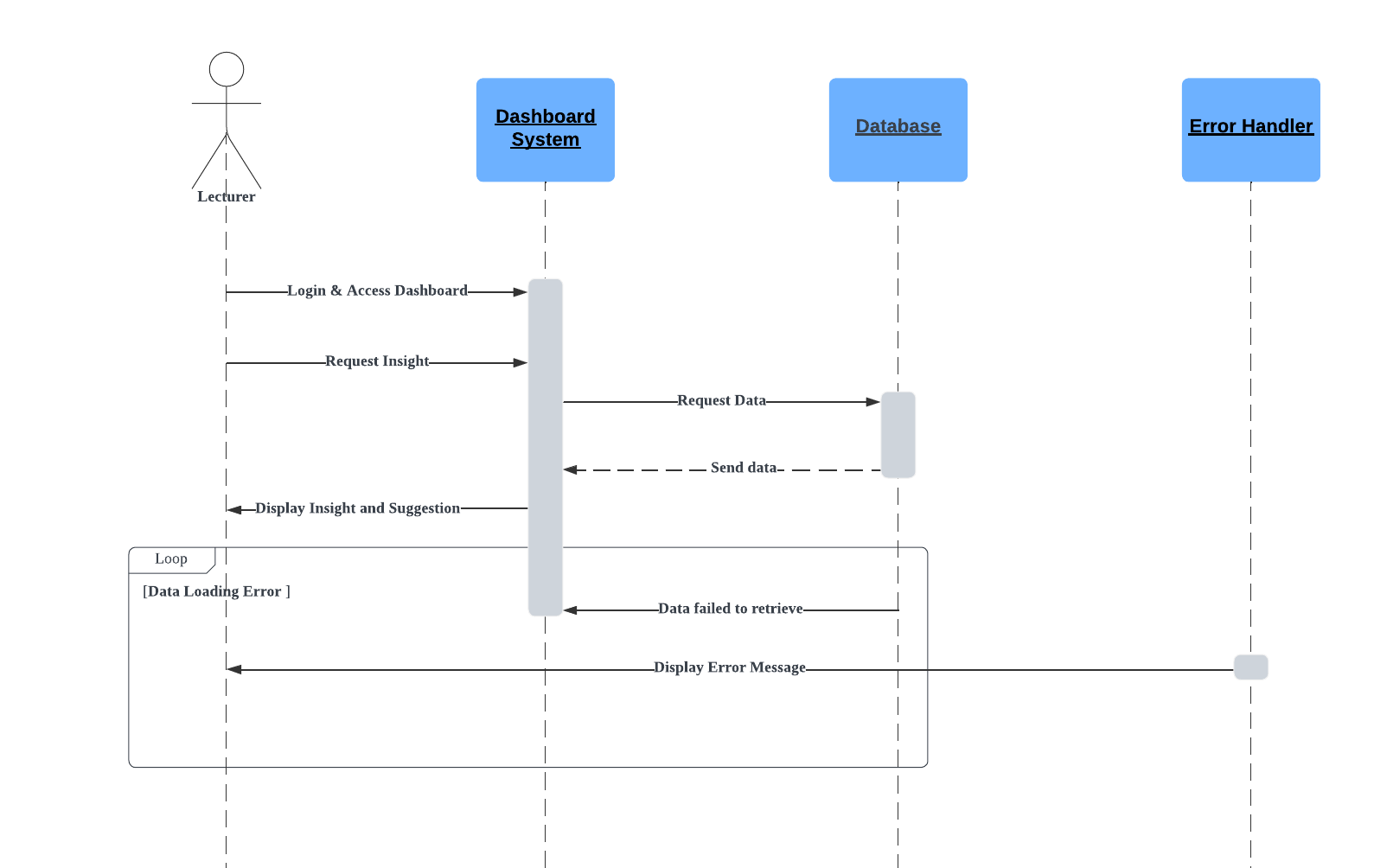
*Figure 1.1.3*

|  |  |
| --- | --- |
| Use Case ID/Name | UC4/ Analytics data and Reporting |
| Description | The system shall analyze student performance, engagement, and learning outcomes across courses. It generates real-time and historical reports that are accessible by administrators, lecturers and students. |
| Precondition | The system has successfully integrated and stored data on student interactions, assignment submissions, quiz results, and course participation. |
| Postcondition | * The user receives the report in real-time or historical format with appropriate data visualizations and summaries. |
| Main Scenario | 1. The user accesses the analytics and reporting feature. 2. The user selects either real-time or historical reports. 3. The user filters the report based on specifies criteria, such as course, time range (for historical reports), grades and engagement levels. 4. The system processes the request and retrieves the relevant data from the database. 5. The system generates the report and displays it to the user. |
| Alternate Scenario | - |
| Exception Scenario | 1. Data Retrieval Failure    1. The system encounters an error while retrieving data from the database.    2. The system displays an error message to the user: "Unable to retrieve data for the requested report. Please try again later or contact support." |



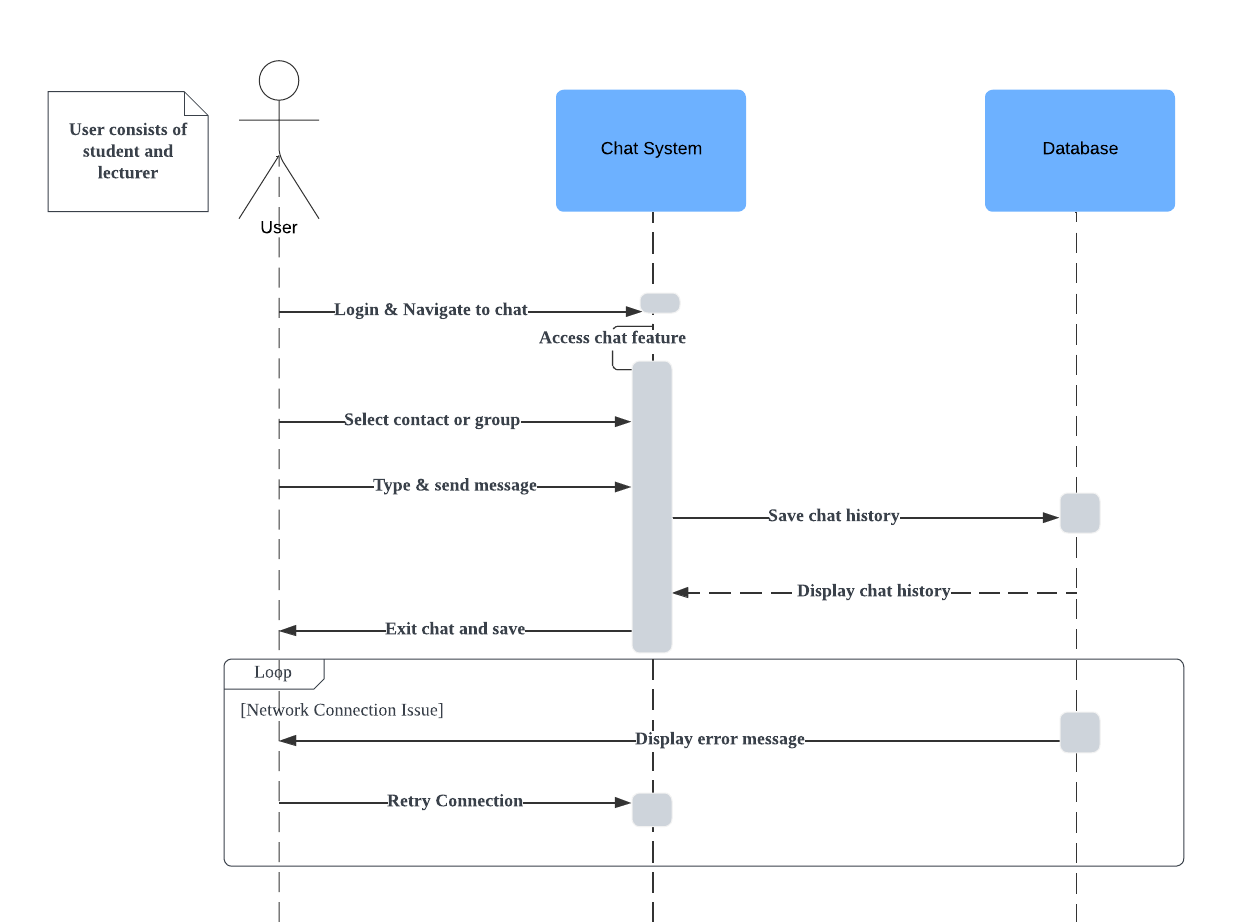
*Figure 1.1.4*

|  |  |
| --- | --- |
| Use Case ID/Name | UC5/Dashboard posting |
| Description | The system provides lecturers with actionable insights into student performance, engagement, and participation. These insights are designed to directly inform teaching strategies, make class adjustments, and implement targeted interventions to meet students' needs. |
| Precondition | Lecturers sign in and have access to the insight's dashboard. |
| Postcondition | * The insights help lecturers adapt their teaching strategies, adjust class content or delivery, and provide targeted support to students who need it. |
| Main Scenario | 1. The lecturer logs into the system and navigates to the insights section of the dashboard. 2. The insights are presented using various data visualizations (charts, graphs, and tables) highlighting areas of concern or trends. 3. The system suggests possible interventions, such as offering additional resources or adjusting teaching methods for specific topics. 4. The lecturer plans adjustments, such as revising lesson plans, creating study groups (etc.TT1L), or scheduling one-on-one meetings with students based on the insights provided. 5. The lecturer makes the necessary class adjustments or interventions based on actionable insights. |
| Alternate Scenario | - |
| Exception Scenario | 1. Data Loading Error    1. The system encounters an error while loading data for the dashboard    2. The system shows an error message to the educator: "Unable to load data now. Please try again later or contact support." |

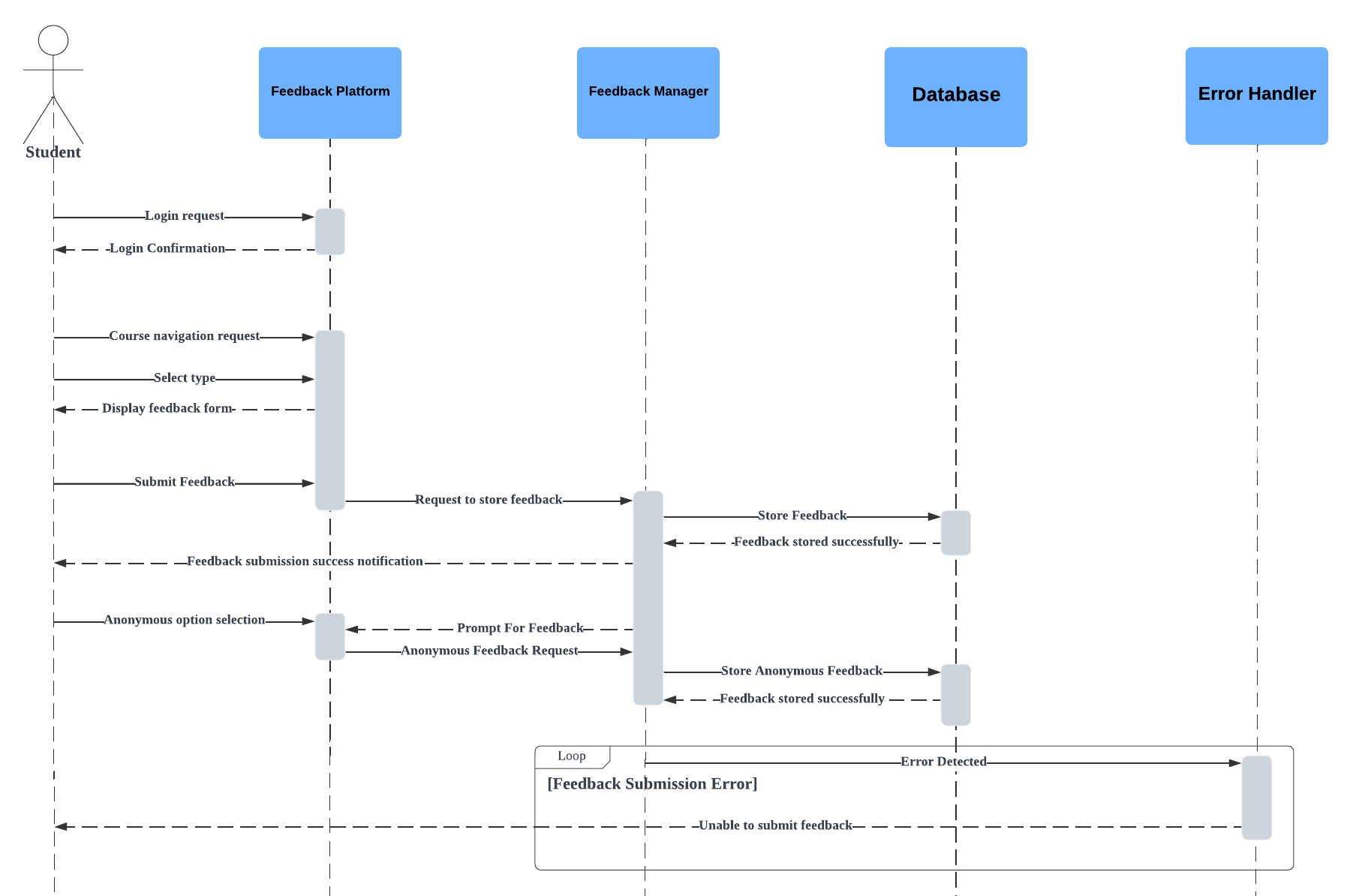


*Figure 1.1.5*

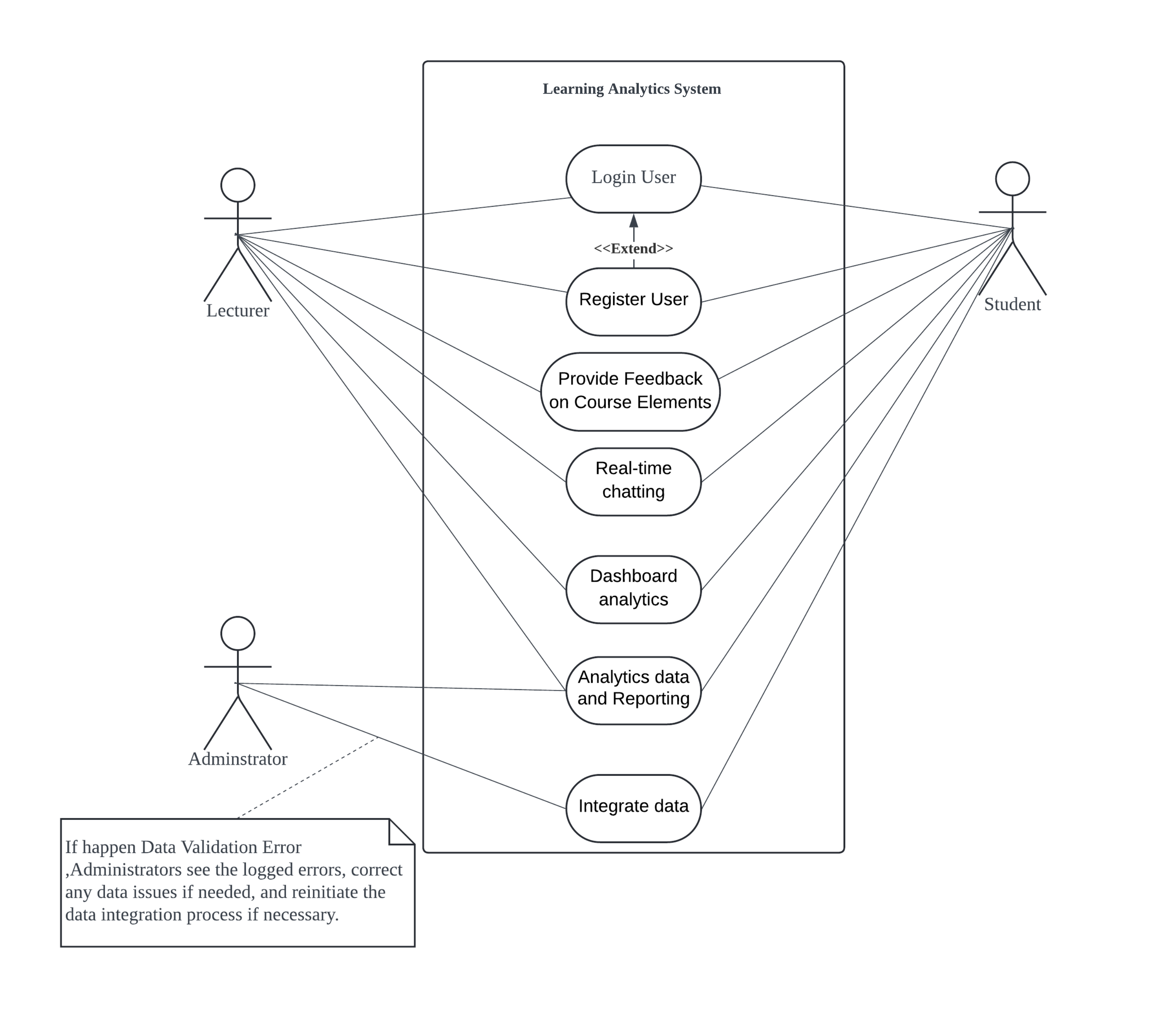
|  |  |
| --- | --- |
| Use Case ID/Name | UC6/ Real-time Chatting |
| Description | The system provides a real-time chat feature that allows students and lecturers to communicate with each other through a chat box. |
| Precondition | Users (students and lecturers) are logged into the system to use the chat functionality. |
| Postcondition | * Users can send and receive messages in real time through the chat box. * Chat history is stored and accessible according to user permissions and system policies. |
| Main Scenario | 1. A user logs into the system and navigates to the chat feature. 2. The user clicks on the chat icon. 3. The user selects a contact or conversation from their list of available users to chat 4. The user types a message into the chat input field and presses "Send." 5. The system immediately transmits the message to the recipient and displays it in the chat box. 6. The recipient receives the message in real time and it appears in their chat box. 7. The user can scroll through the chat history to review previous messages within the conversation. 8. The user closes the chat box or logs out of the system. The chat session is saved |
| Alternate Scenario | 1. A user initiates a group chat. 2. The user selects multiple contacts to include in the group chat 3. Messages sent to the group are visible to all participants. 4. The user can add or remove participants and set group chat preferences if authorized. 5. The user exits the group chat, and the chat history remains accessible to other members. |
| Exception Scenario | 1. Network Connectivity Issue    1. The user encounters network connectivity issues, preventing real-time communication.    2. The system displays an error message: "Network connectivity issue. Please check your internet connection."    3. The user may attempt to reconnect to the chat service by clicking the "refresh” button. |

*Figure 1.1.6*

|  |  |
| --- | --- |
| Use Case ID/Name | UC7/ Provide Feedback on Course Elements |
| Description | The system allows students to give feedback on specific course elements (e.g., assignments, lectures, quizzes). The feedback is directly tied to those course elements, providing valuable input for instructors to improve course content and teaching methods. |
| Precondition | The student is logged into the system, has completed or participated in the course element (e.g., submitted an assignment, attended a lecture). |
| Postcondition | * The system stores the student’s feedback. |
| Main Scenario | 1. The student logs into the system and navigates to the course they are enrolled in 2. The student selects a specific course element (e.g., a lecture, quiz, or assignment) to provide feedback. 3. The system prompts the student to enter feedback through a rating system. 4. The student submits the feedback. 5. The system stores the feedback and links it to the corresponding course element. |
| Alternate Scenario | 1. The student logs into the system and navigates to the course. 2. The student selects the option to provide anonymous feedback. 3. The student provides their feedback 4. System stores as anonymous, without linking the student's identity to the feedback. 5. The feedback is stored anonymously and is accessible to the instructor for review. |
| Exception Scenario | 1. Feedback Submission Error    1. The system encounters an error such as network issue or data processing error.    2. The system shows an error message to the educator: " Unable to submit feedback. Please try again later." |

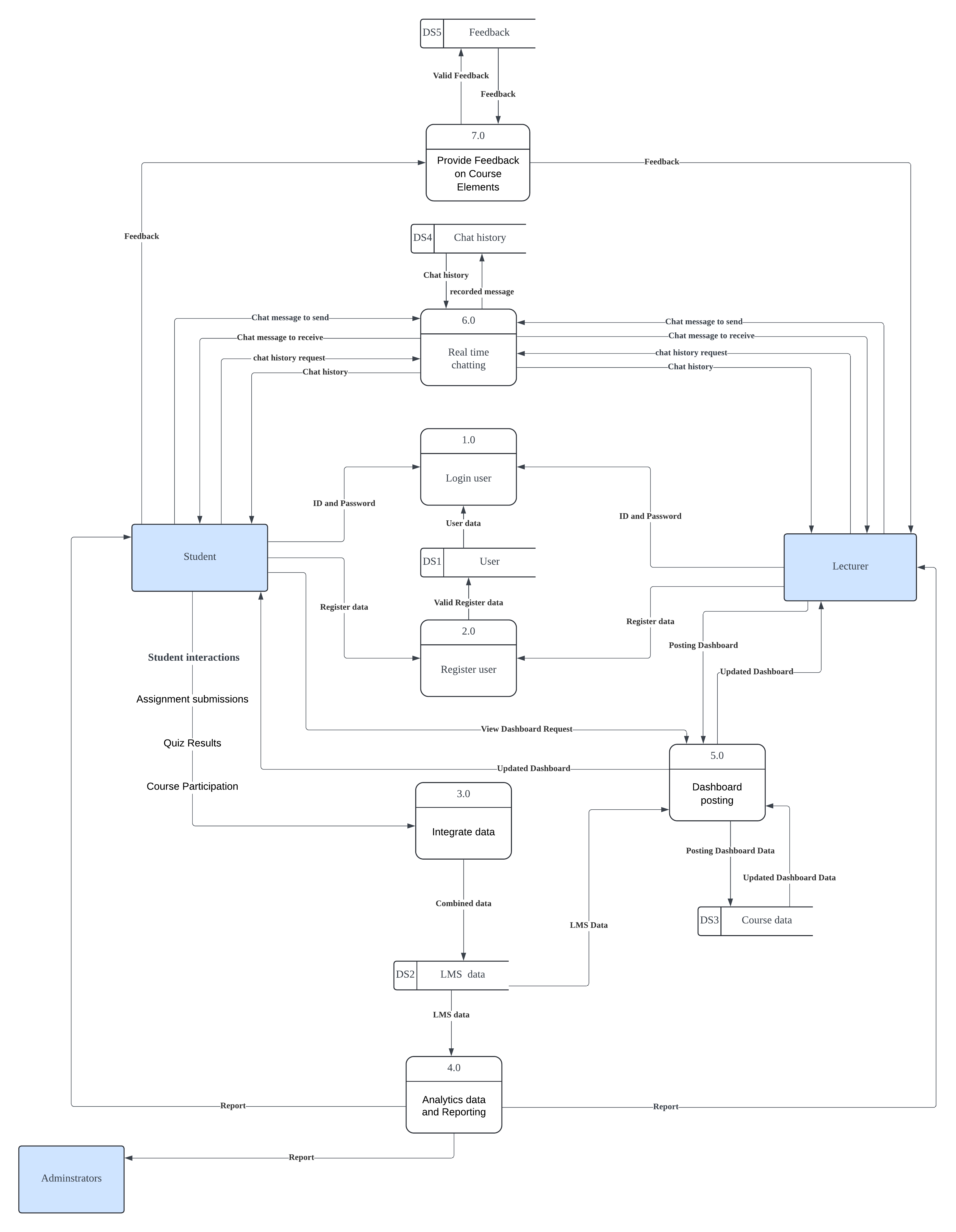
*Figure 1.1.7*

# Use case diagram



*Figure 2.0 Use Case Diagram*

# Data flow Diagram



*Figure 3.0 Data Flow Diagram*