Academic Session: 20232024/1

SECR3104-03 – APPLICATIONS DEVELOPMENT

System Documentation (SD)

Project: Google Drive (DUX)

Prepared By:

Group Nemo

Lecturer: DR.MOHD RAZAK BIN SAMINGAN

Group Members:

No	Name	Matric No
1	MUHAMMAD FITRI BIN ISMAIL	A21EC0076
2	MUHAMMAD ADAM HAIKAL	A21EC0062
3	AFIF BIN GENARI@AZHARI	A21EC0003
4	MUHAMAD FAIZ BIN ABDUL MUTALIB	A21EC0059
5	AMOS KEAGAN HOSEA	A21EC0161

Table Of Contents

1. Introduction

1.1 Purpose

The purpose of this Web Portal Documentation (WPD) is to comprehensively outline the development and functionality of the web portal project aimed at addressing the inefficiencies and complexities associated with the current method of sharing Google Drive (GD) resources. The WPD encompasses the System Requirements Specification (SRS), System Design Document (SDD), and System Testing Document (STD), providing a detailed roadmap for developers, testers, project managers, and end-users.

Our project seeks to revolutionize the way users interact with shared GD resources by introducing a user-friendly desktop web system. The existing approach, relying on email messages and the default GD page, has proven ineffective and cumbersome for users to trace shared resources. The primary objective of our system is to offer an enhanced and more convenient method for users to view and manage GD shared resources seamlessly.

1.2 Scope

The scope of our project, the development of the innovative web platform named DUX, aims to empower users in efficiently managing their shared files. Drawing inspiration from current Google Drive applications, DUX will serve as a comprehensive platform where all the shared Google Drive links will be sort according to the category in the DUX platform.

The key functionalities of the project include:

- **File Naming:**
  DUX provides a user-friendly interface for naming files, ensuring clarity and organization. Users can follow standard naming conventions effortlessly.

- **File Tracing:**
  DUX tackles the challenge of tracking shared files by offering a simple tracing mechanism. Users can quickly locate and trace files within the platform, eliminating the complexities of traditional methods.

- **File Sorting:**
  DUX introduces sorting feature, categorizing shared files based on customizable criteria. This enhances overall accessibility, enabling users to find and manage resources swiftly.

1.3 Overview

We were tasked to create a web portal that will solve the following problem: the current approach of sharing Google Drive (GD) resources, which primarily relies on email messages and the default GD page (https://drive.google.com/), has proven to be ineffective and cumbersome for users to trace shared resources. To rectify this issue, our project aims to provide a user-friendly desktop web system that offers an improved and more convenient way to view and manage GD shared resources. In order to ensure a simpler and more efficient resource sharing process, our system will also include features that lead users methodically. Examples of these features include adherence to standard file and folder naming conventions and the consistent use of created information structures.

2. Specific Requirements

2.1 System Features

DUX seamlessly organizes shared Google Drive links based on categories, providing users with an intuitive web platform for efficient management of their shared files.

2.1.1 Use Case Diagram

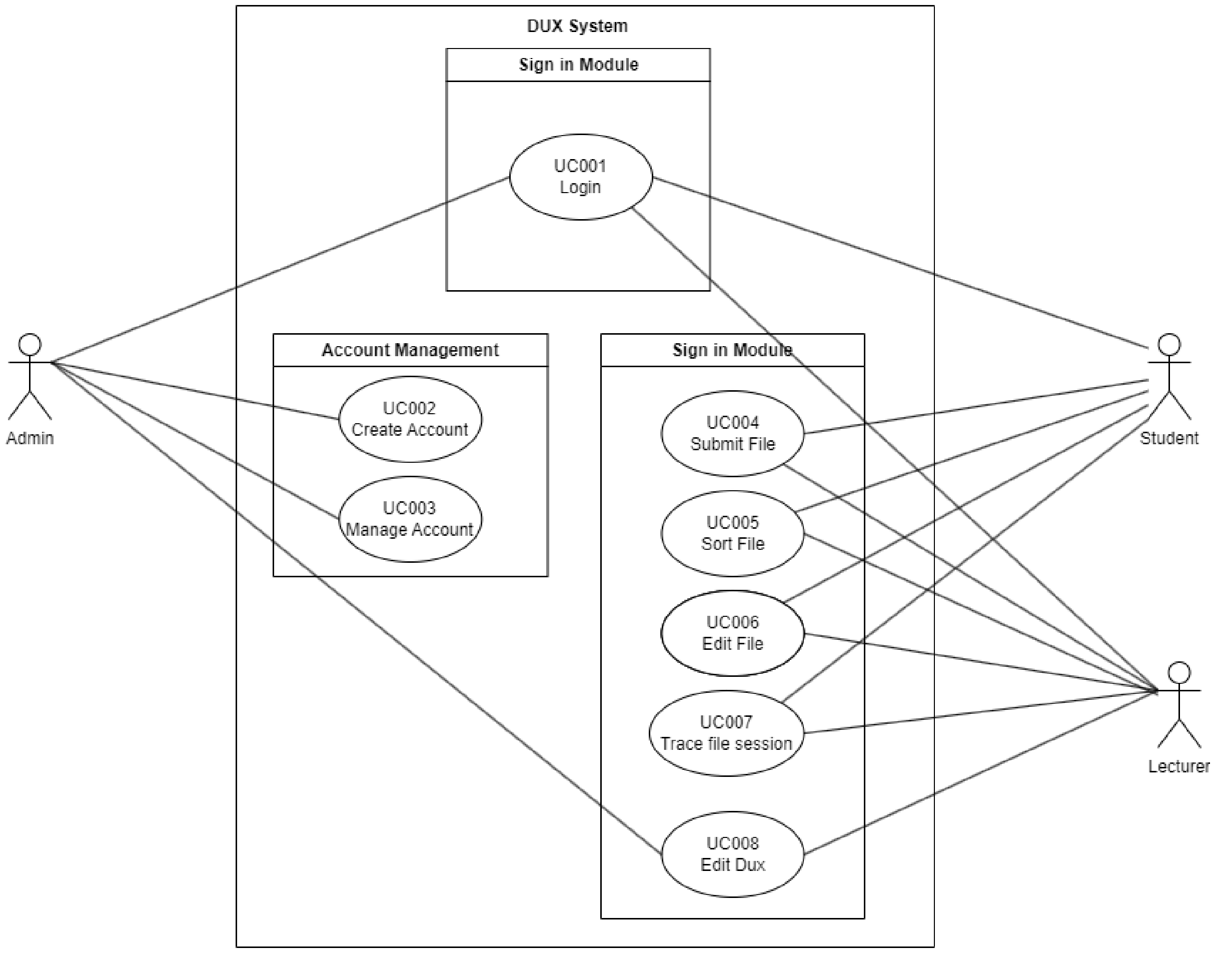


Figure 1.0: Use Case Diagram for <DUX>

2.1.1.1 Updated Use Case Diagram

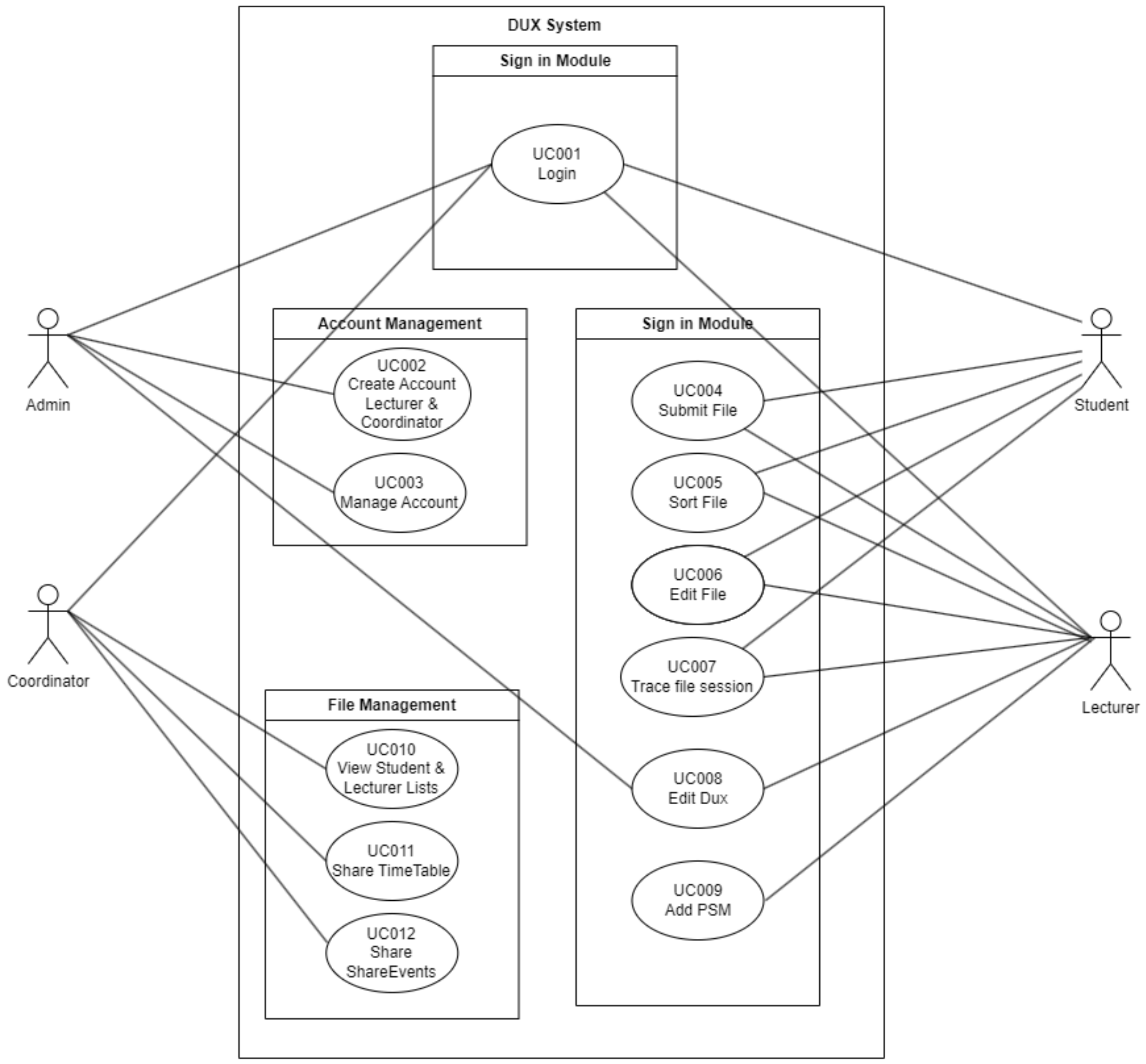


Figure 1.1: Updated Use Case Diagram for <DUX>

Table 1.0: Description of Module and Functions for <DUX>

Module	Function	Description
Sign In Module	UC001 - login	Allows users to securely sign in to the DUX platform, ensuring the protection of their accounts and enabling access to personalized features.
Account Management Module	UC002 - CreateAccount	Provides functionalities for users to create new accounts
	UC003 - ManageAccount	Provides functionalities for users to manage existing ones, including profile settings, security options, and account preferences.
File Management Module	UC004-SubmitFile	Users can easily upload and share files on the DUX platform, streamlining the process of adding content to the system.
	UC005 - SortFile	Organize shared files effortlessly by categorizing them, ensuring a neat and accessible file structure for efficient navigation.
	UC006 - EditFile	Edit the content of shared files directly within DUX, providing a convenient way to make real-time updates without external applications.
	UC007-TraceFile	Track the history of a file's activity, offering users insights into who accessed the file and when, enhancing transparency and collaboration.
	UC008-EditDux	Personalize the DUX platform by adjusting settings, preferences, and configurations to tailor the user experience according to individual needs and preferences.
	UC009 - Add PSM	Facilitate the addition and management of sharing PSM within the DUX platform between the lecturers and students.
	UC010-View Students and Lecturer List	Provide a comprehensive view of the list of students and lecturers associated with a course or project, enabling users to

		easily access and manage participant information for better coordination and communication.
	UC011 - Share TimeTable	Coordinator being able to share course or project timetables through DUX, allowing users to coordinate schedules, view upcoming events.
	UC012 - Share Events	Enable users to share and manage events directly within the DUX platform, ensuring that all stakeholders are informed about upcoming activities, deadlines, and important milestones

2.1.2 Entity Relationship Diagram

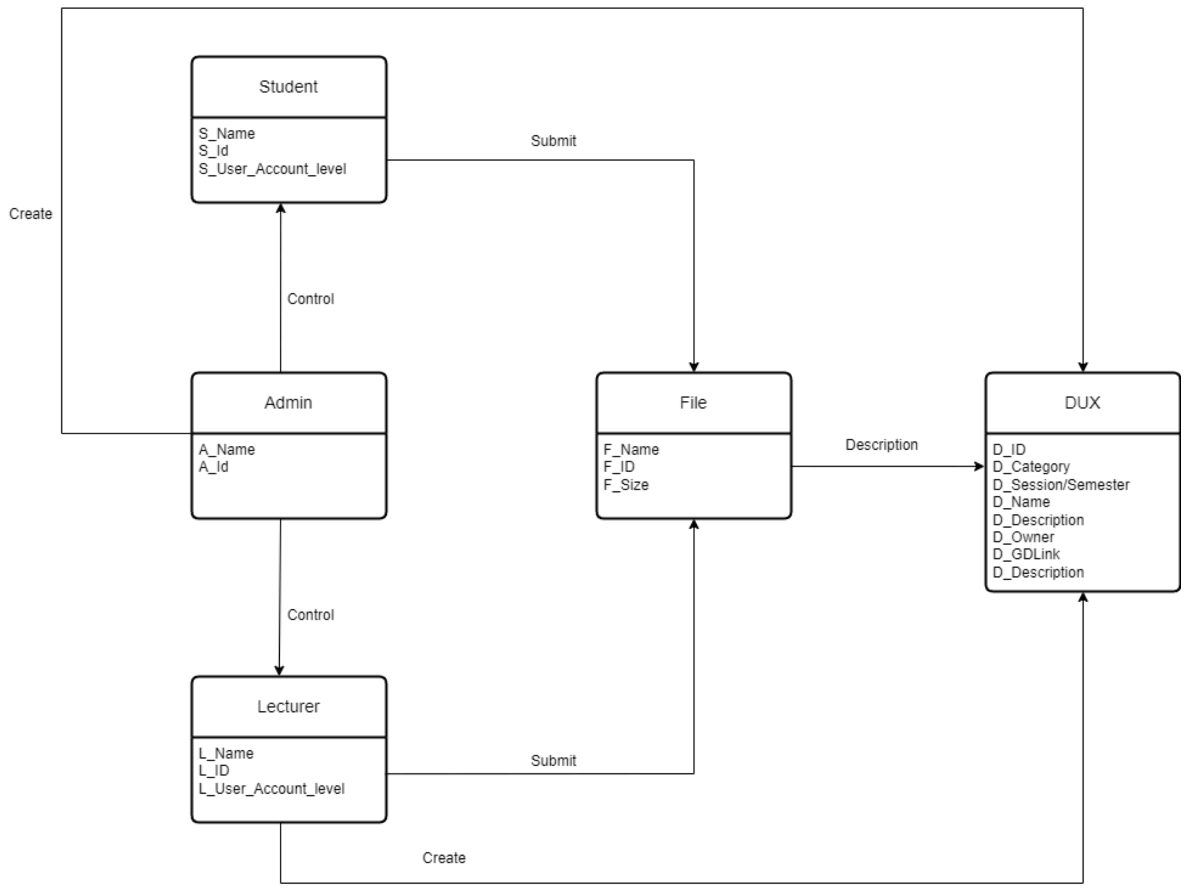


Figure 2.0: ERD for <DUX>

2.1.2.1 Updated Entity Relationship Diagram

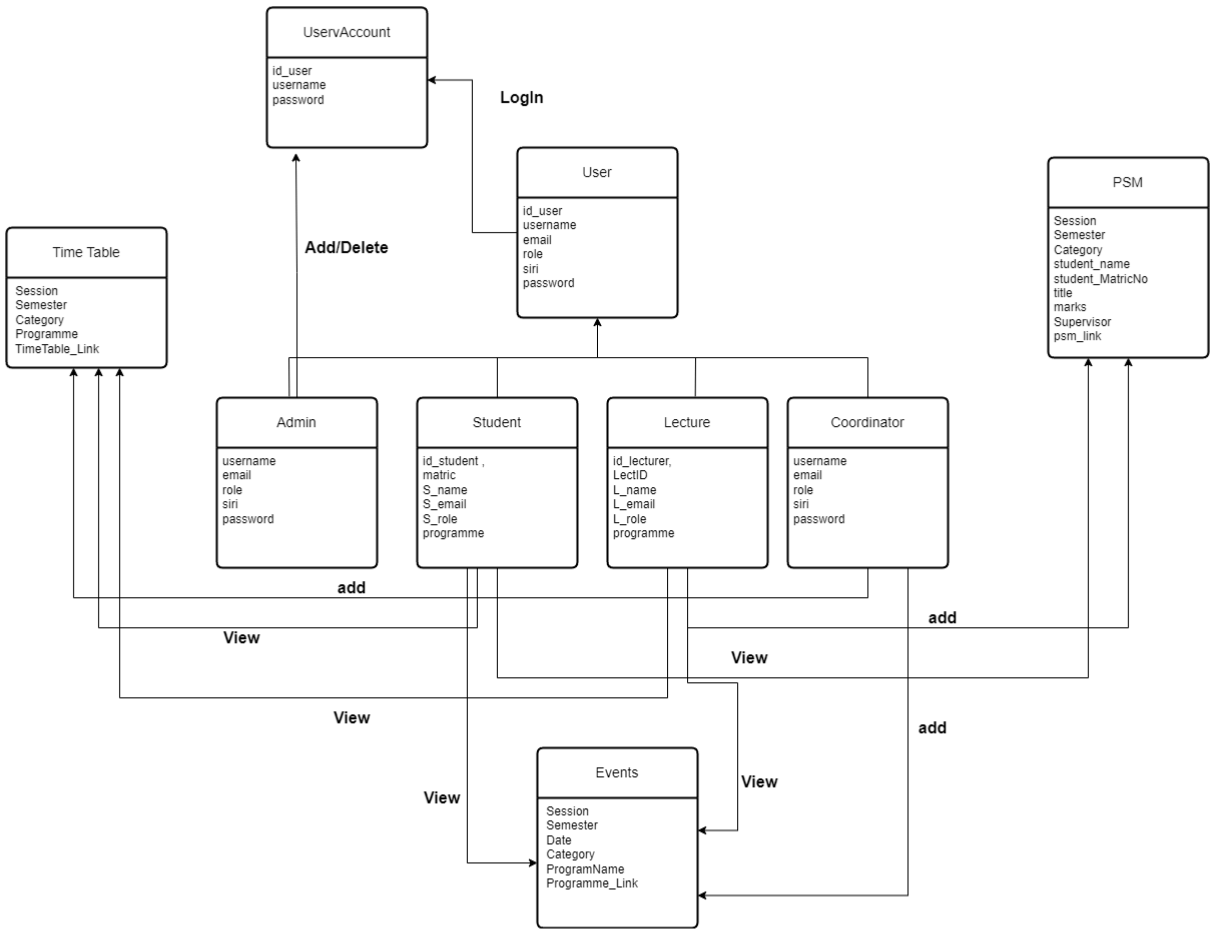


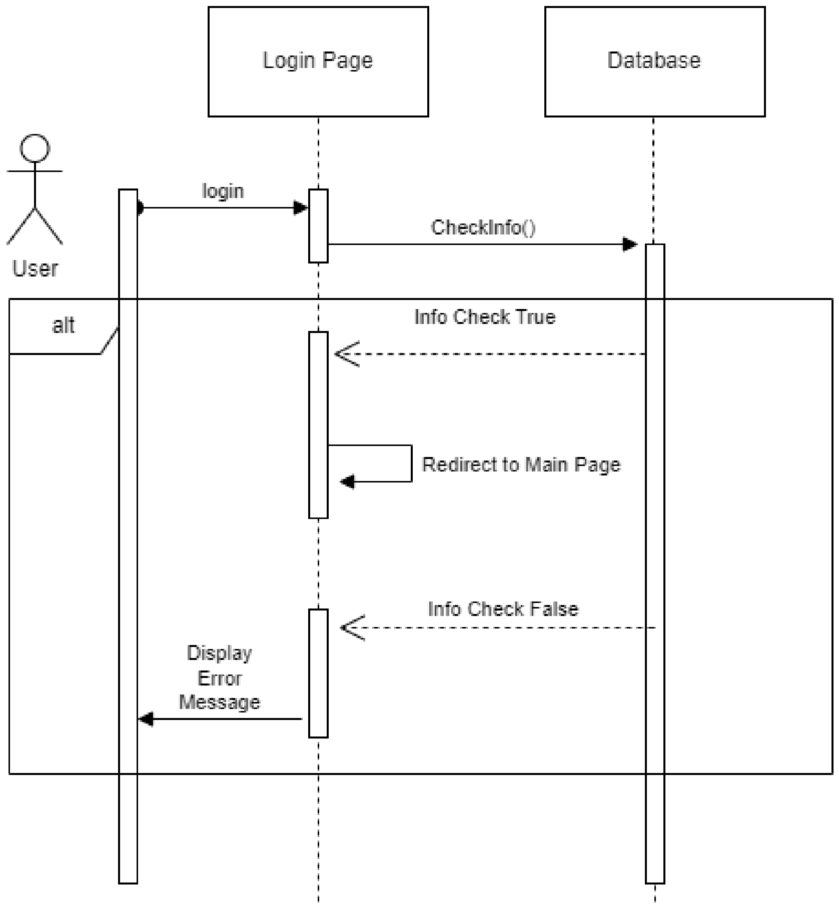
Figure 2.1: Updated ERD for <DUX>

2.1.3 Use case: UC001 - login

Table 2.0: Use Case Description for **<login>**

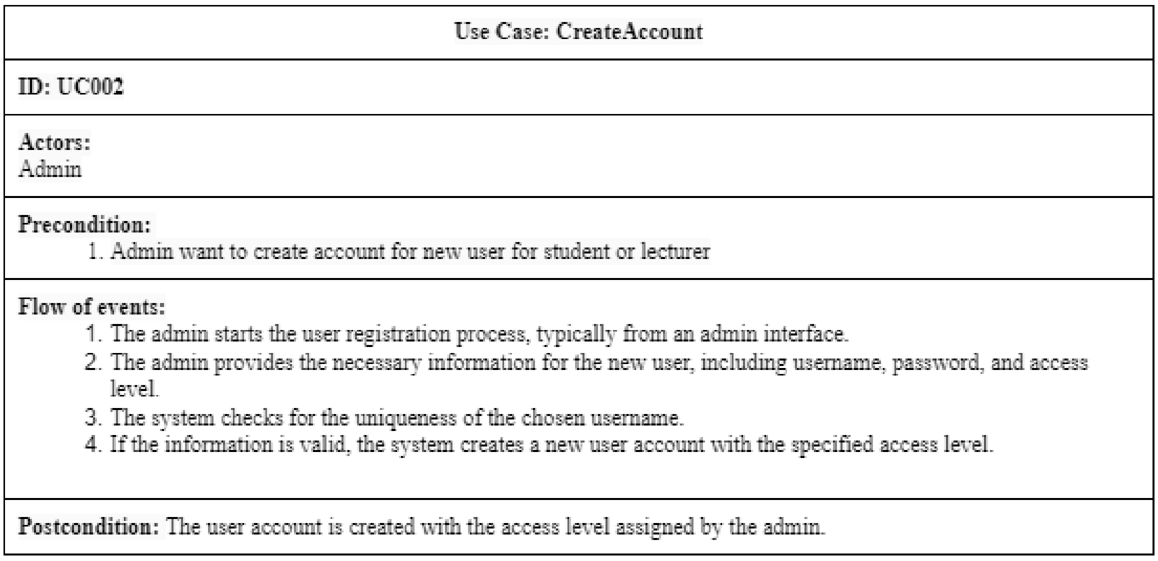
Use Case: Login
ID: UC001
Actors: Users (Admin, Student or Lecturer)
Precondition: 1. A valid user account must already exist in the system.
Flow of events: 1. The user initiates the login process by providing their credentials like matric number and password. 2. The system checks the entered credentials against the stored user account information. 3. If the credentials are valid, the system grants access to the user. 4. If the credentials are invalid, the system denies access and notifies the user of the authentication failure.
Postcondition: The user gains access to their account and is now authenticated within the system.

Figure 3.0: Sequence Diagram for **<Login>**



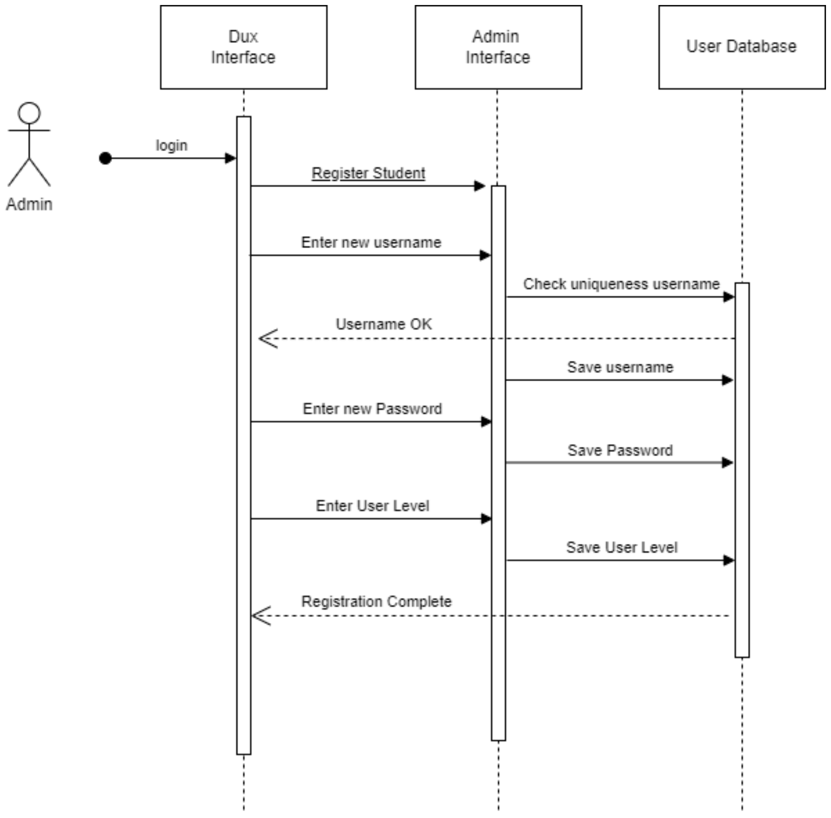
2.1.4 Use case: UC002 - CreateAccount

Table 3.0: Use Case Description for <CreateAccount>



Use Case: CreateAccount
ID: UC002
Actors: Admin
Precondition: 1. Admin want to create account for new user for student or lecturer
Flow of events: 1. The admin starts the user registration process, typically from an admin interface. 2. The admin provides the necessary information for the new user, including username, password, and access level. 3. The system checks for the uniqueness of the chosen username. 4. If the information is valid, the system creates a new user account with the specified access level.
Postcondition: The user account is created with the access level assigned by the admin.

Figure 4.0: Sequence Diagram for <CreateAccount>

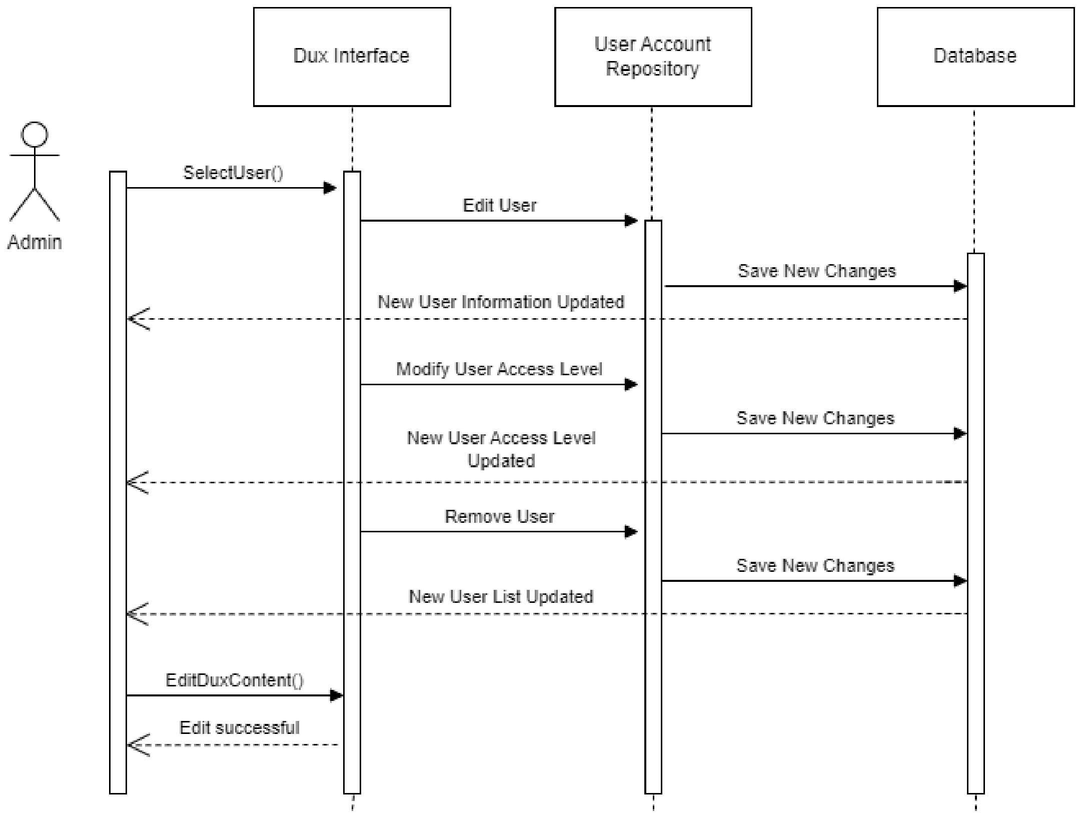


2.1.5 Use case: UC003 - ManageAccount

Table 4.0: Use Case Description for <ManageAccount>

Use Case: ManageAccount
ID: UC003
Actors: Admin
Precondition: 1. The admin initiating account management must have valid credentials.
Flow of events: 1. The admin logs into the system using their credentials. 2. The admin navigates to the "Manage Account" section of the admin interface. 3. The admin selects a specific user account to manage, typically from a list. 4. The admin can view and edit user information, including username, email, and access level. 5. The admin modifies the access level of the selected user (e.g., changing from student to lecturer or vice versa). 6. The admin has the option to remove the selected user account from the system. 7. Additionally, the admin may have the ability to edit system content, such as category, session/semester or other relevant information.
Postcondition: Any changes made by the admin, including edits to user access levels, user removal, or system content, are successfully applied.

Figure 5.0: Sequence Diagram for <ManageAccount>

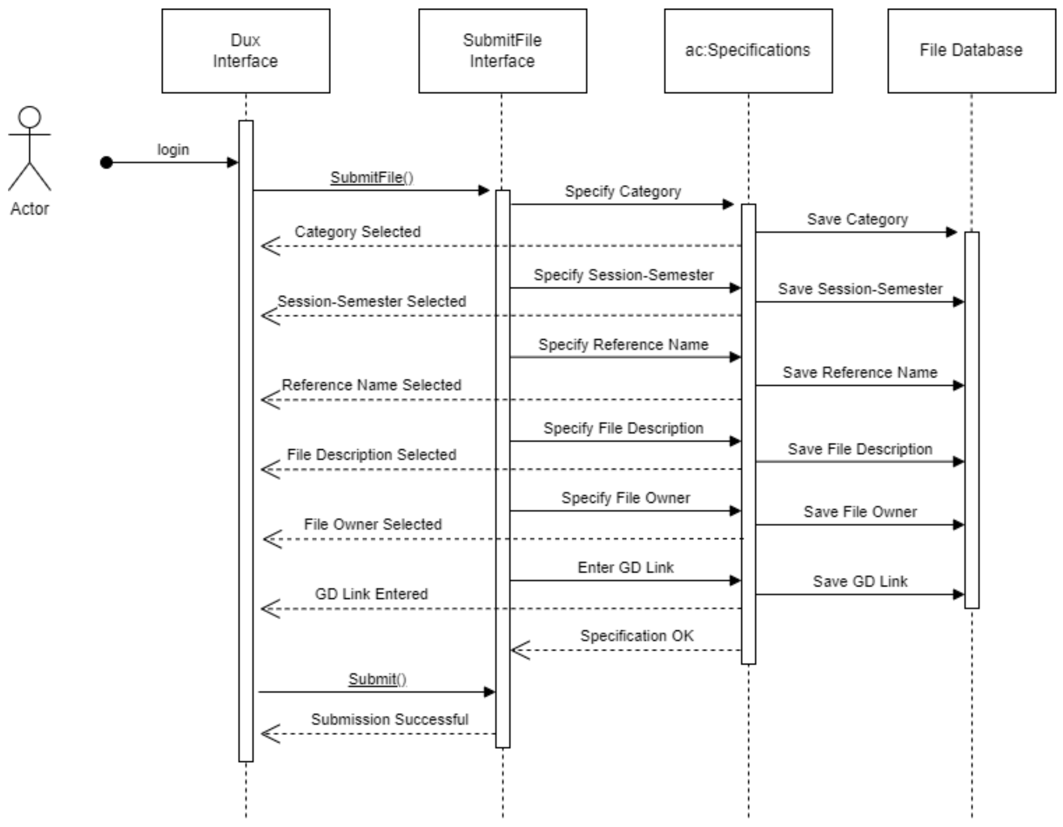


2.1.6 Use case: UC004-SubmitFile

Table 5.0: Use Case Description for **<SubmitFile>**

Use Case: SubmitFile
ID: UC004
Actors: Users(Students and Lecturers)
Precondition: 1. The user must login into the system using their credentials
Flow of events: 1. The user navigate to "Submit File" section of the user interface 2. The user need to specify the category of file that need to be submitted 3. The user need to specify the session-semester of file that need to be submitted 4. The user need to specify the reference name of file that need to be submitted 5. The user need to specify the description of file that need to be submitted 6. The user need to specify the owner of file that need to be submitted by enter their own email 7. The user need to enter the link of Google Drive file that wanted to be shared 8. The user the need to click on the submit button
Postcondition: Notifications will pop up indicating that the file have been submitted

Figure 6.0: Sequence Diagram for **<SubmitFile>**

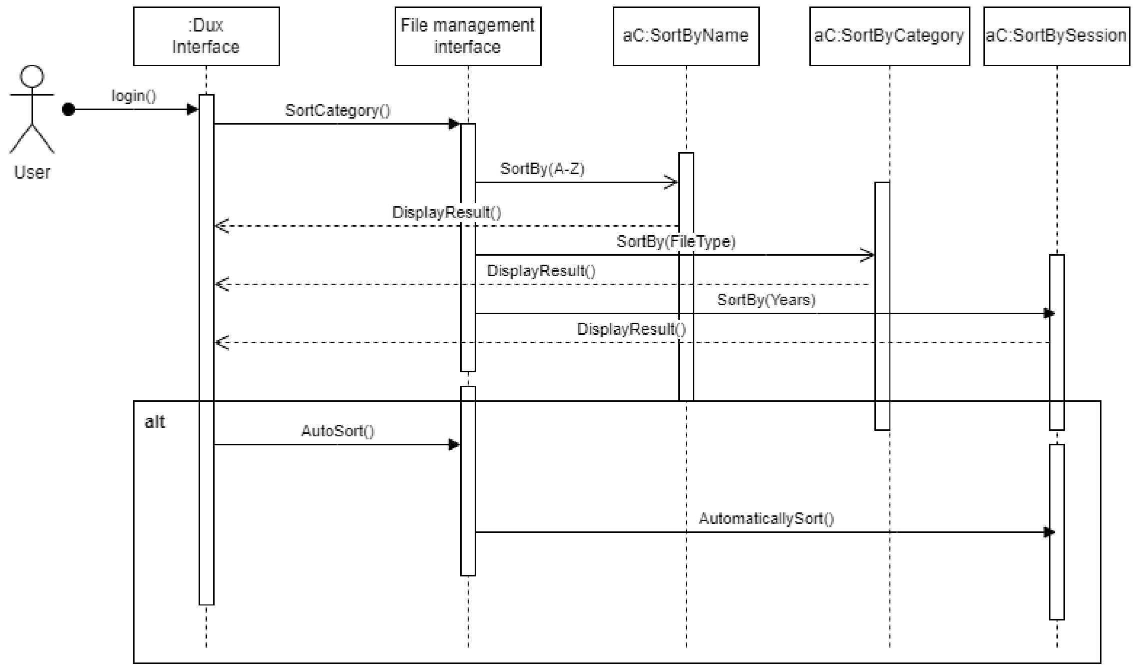


2.1.7 Use case: UC005 - SortFile

Table 6.0: Use Case Description for <SortFile>

Use Case: SortFile
ID: UC005
Actors: Users (Admin, Student or Lecturer)
Precondition: 1. The user must be logged into the system.
Flow of events: 1. The user navigates to the "SortFile" feature within the platform. 2. The user selects the preferred sorting option(such as sort by Folder Name, Category, or Session-Semester.). 3. If the user selects "Sort by Folder Name": The system will sorts files and folders in alphabetical order of their folder names. 4. If the user selects "Sort by Category": The system will sorts files and folders according to their assigned categories. 5. If the user selects " Session-Semester": The system will sorts files and folders based on the session and semester associated with them. 6. The system retrieves and displays the files and folders according to the chosen sorting criteria.
Postcondition: The files and folders are displayed to the user based on the selected sorting criteria.

Figure 7.0: Sequence Diagram for <SortFile>

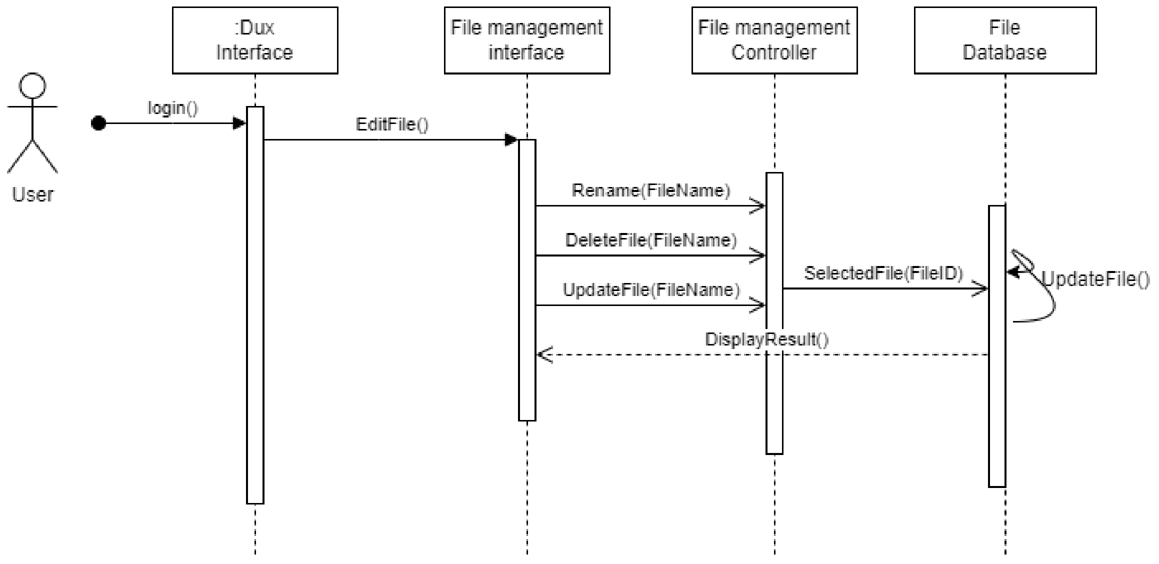


2.1.8 Use case: UC006 - EditFile

Table 7.0: Use Case Description for <EditFile>

Use Case: EditFile
ID: UC006
Actors: User (Admin, Lecturer, or Student)
Precondition: 1. The user must be logged into the system. 2. User has access to the file they want to edit.
Flow of events: 1. The user navigates to the "Edit File" section within the platform. 2. The user clicks on the "Edit" button or option associated with the selected file. 3. If the selected item is a file, the system provides options to edit the file, such as renaming, updating content, or attaching additional information. 4. If the selected item is a folder, the system allows the user to modify the folder name and add or remove files within the folder. 5. The user clicks on the "Save" button or option to apply the made changes and save the updated file. 6. The system updates the file with the user's changes and reflects the updated version in the file management system.
Postcondition: The selected file or folder is successfully edited with the changes made by the user and the system reflects the modifications in real-time for all users who have access to the edited file or folder.

Figure 8.0: Sequence Diagram for <EditFile>

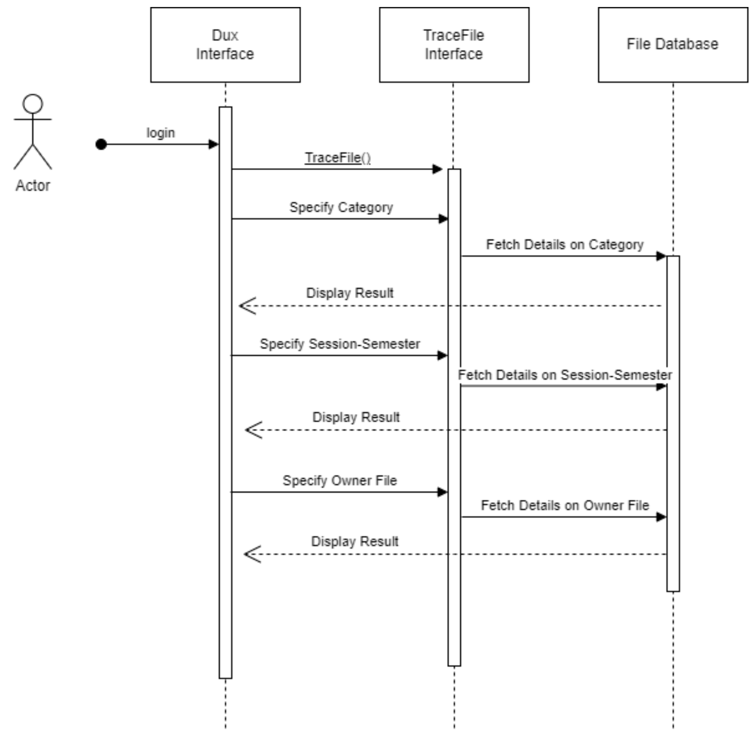


2.1.9 Use case: UC007-TraceFile

Table 8.0: Use Case Description for **<TraceFile>**

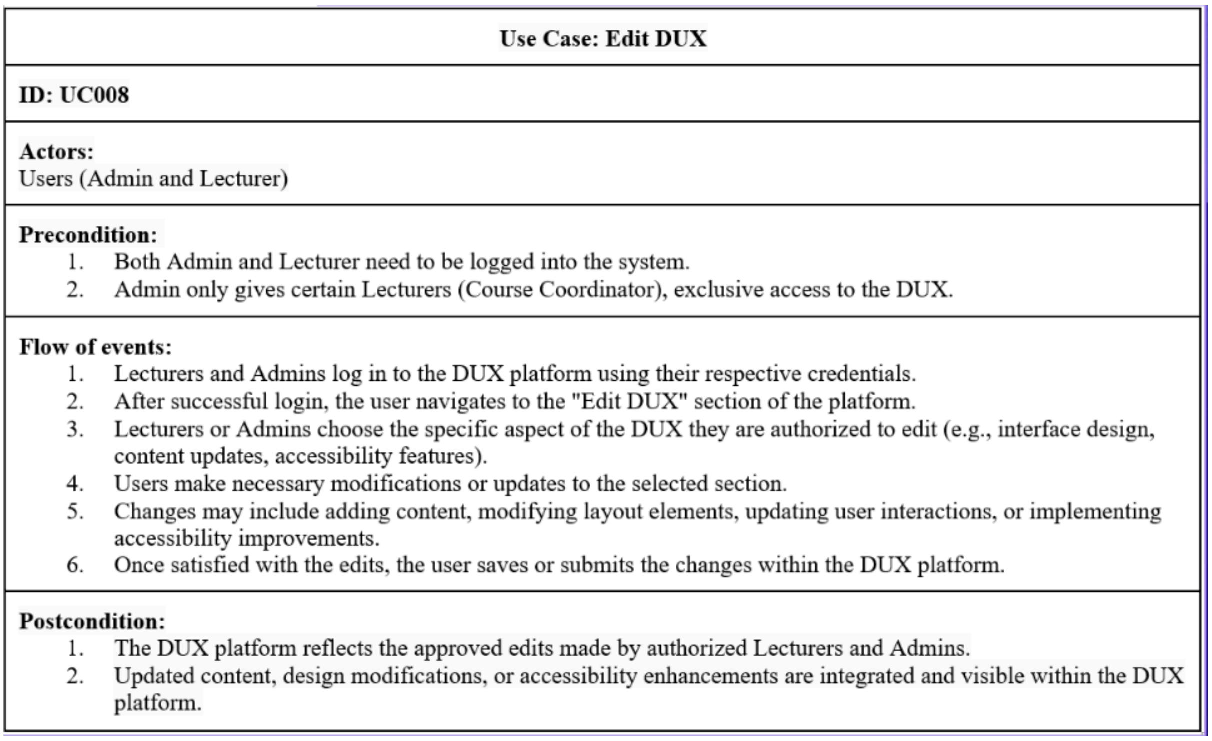
Use Case: TraceFile
ID: UC007
Actors: Users(Students and Lecturers)
Precondition: 1. There must be at least one shared file submitted to the platform
Flow of events: 1. The user log into the system using their credentials. 2. The user navigate to the DUX user interface 3. The user can trace the file based on their category, session-semester and the owner. 4. If user select session-semester: 1. User need to select which session-semester of the file they want to view 2. The system will then provide all the files based on the session-semester that the user have chosen including the owner of the file and the category of the file 5. The system displays the category , reference name , session-semester , description , owner and link of the file
Postcondition: User can now trace the file based on their input

Figure 9.0: Sequence Diagram for **<TraceFile>**



2.1.10 Use case: UC008-EditDux

Table 10.0: Use Case Description for **<EditDux>**



Use Case: Edit DUX

ID: UC008

Actors:
Users (Admin and Lecturer)

Precondition:

1. Both Admin and Lecturer need to be logged into the system.
2. Admin only gives certain Lecturers (Course Coordinator), exclusive access to the DUX.

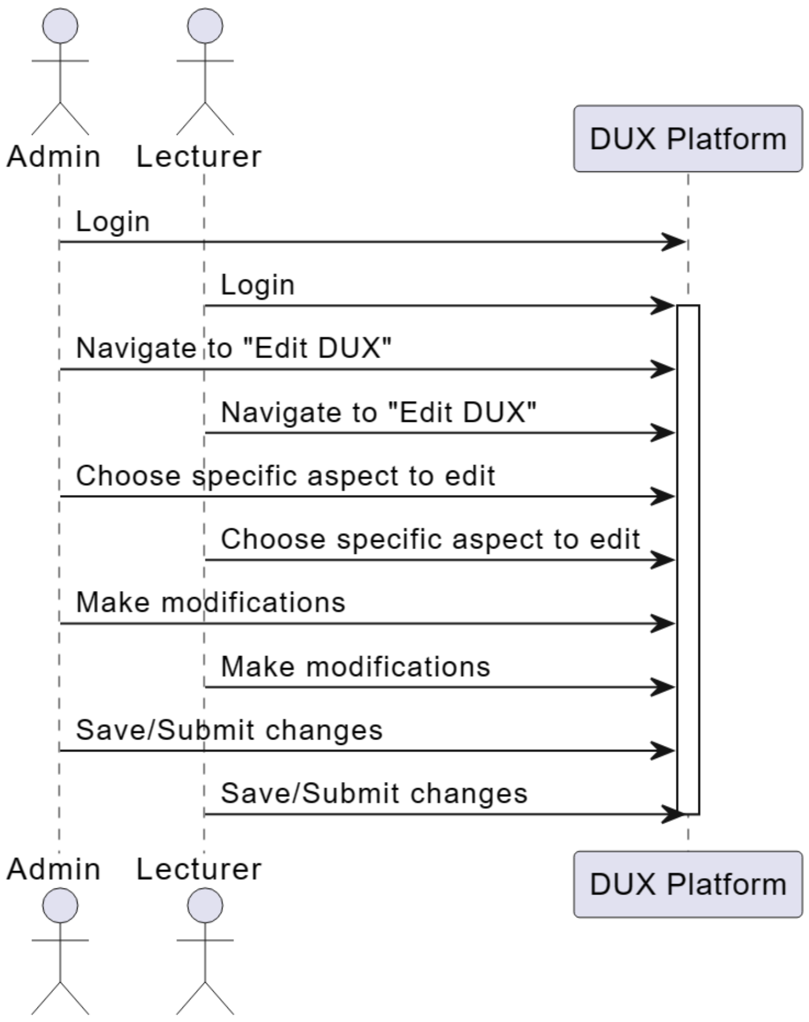
Flow of events:

1. Lecturers and Admins log in to the DUX platform using their respective credentials.
2. After successful login, the user navigates to the "Edit DUX" section of the platform.
3. Lecturers or Admins choose the specific aspect of the DUX they are authorized to edit (e.g., interface design, content updates, accessibility features).
4. Users make necessary modifications or updates to the selected section.
5. Changes may include adding content, modifying layout elements, updating user interactions, or implementing accessibility improvements.
6. Once satisfied with the edits, the user saves or submits the changes within the DUX platform.

Postcondition:

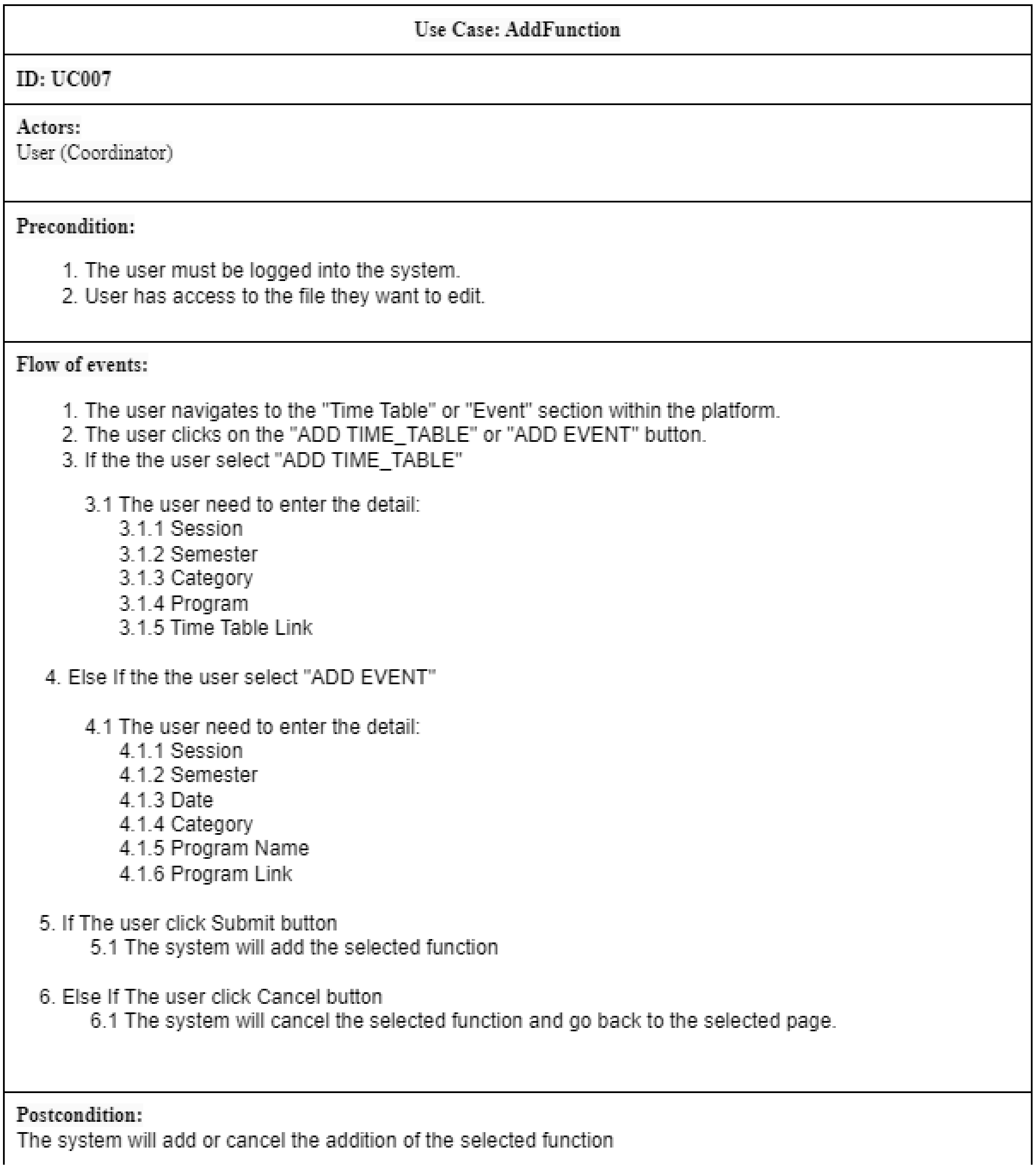
1. The DUX platform reflects the approved edits made by authorized Lecturers and Admins.
2. Updated content, design modifications, or accessibility enhancements are integrated and visible within the DUX platform.

***Figure 10.0: Sequence Diagram for* <EditDux>**



2.1.11 Use case: UC009-AddFunction

Table 11.0: Use Case Description for **<AddFuntion>**



Use Case: AddFunction

ID: UC007

Actors:
User (Coordinator)

Precondition:

1. The user must be logged into the system.
2. User has access to the file they want to edit.

Flow of events:

1. The user navigates to the "Time Table" or "Event" section within the platform.
2. The user clicks on the "ADD TIME_TABLE" or "ADD EVENT" button.
3. If the the user select "ADD TIME_TABLE"

   3.1 The user need to enter the detail:
      3.1.1 Session
      3.1.2 Semester
      3.1.3 Category
      3.1.4 Program
      3.1.5 Time Table Link

4. Else If the the user select "ADD EVENT"

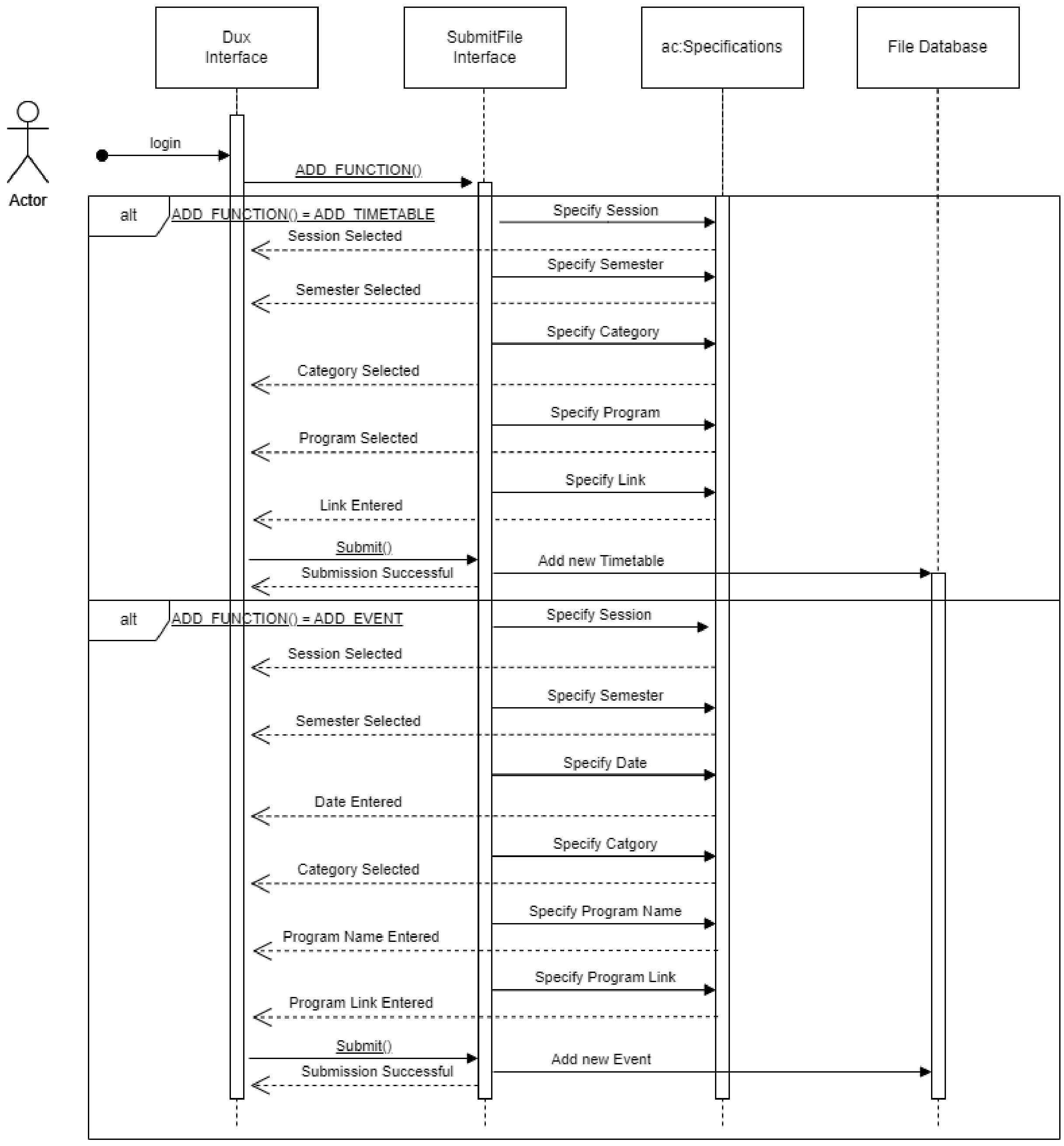
   4.1 The user need to enter the detail:
      4.1.1 Session
      4.1.2 Semester
      4.1.3 Date
      4.1.4 Category
      4.1.5 Program Name
      4.1.6 Program Link

5. If The user click Submit button
   5.1 The system will add the selected function

6. Else If The user click Cancel button
   6.1 The system will cancel the selected function and go back to the selected page.

Postcondition:
The system will add or cancel the addition of the selected function

Figure 11.0: Sequence Diagram for <AddFunction>



2.2 Software System Attributes, Performance and Other Requirements

The DUX system product needs to be dependable, user-friendly and simple to maintain. It should function well, have a large capacity and quick reaction times,and interact with existing systems. Maintaining security, legal compliance and recordkeeping is crucial. The System also needs to handle users, backup data and be regularly maintained.

Software System Attributes:

- Usability : To ensure interaction for student, lecturer, coordinator and admin, the DUX system product should have a user-friendly interface and intuitive navigation.

- Maintainability : The system should be created in a way that is modular and well-structured, making it easy to maintain, update and improve as necessary.

- Reliability : The system must be reliable , ensuring constant accessibility and accurate data management throughout the entire system process.

- Portability : Users should be able to access and use the DUX system regardless of their choice of platform because it should be compatible with a wide range of operating systems, web browsers and devices.

- Compatibility : To ensure effective data exchange and collaboration , the system should be readily compatible with other systems such as google drive platform.

Performance Requirements :

- Response Time : The DUX system product should aim to respond as quickly as possible to users actions, such as loading pages and requesting information.

- Throughput : The system must be capable of supporting a significant amount of concurrent users to guarantee lag-free performance even during busy times like processing customer requests.

- Capacity : The DUX system should be able to manage and store a significant amount of data such as user information, timetable information, event information as well as file information.

- Availability : To enable users to access and use the system whenever they would like, it should have a high uptime percentage and aim for almost continuous availability.

Other Requirements :

- Security : To protect the privacy of users data , the DUX system product should provide strong security measures in place . These measures should include user authentication , data encryption and access control.

- Legal Regulatory : To ensure confidentiality of users information , the DUX system product must comply with all applicable regulations , laws and data protection requirements.
- Documentation : The system should offer complete documentation, such as user manuals, system guides, and technical documentation, to help users understand the functionalities and processes of the system.

2.3 Design Constraints

Design constraints are the limitations or restrictions imposed on a design due to internal or external factors. The final product is heavily reliant on the design, as it depends on the aesthetic taste as well. Thus, it is important that the constraints are well known first before proceeding.

- User constraints: Users need to register as an individual or organization in order to use it. Furthermore, the users must have their correct usernames and passwords to enter the DUX system.
- Website constraints: The system needs to be presented in a simple way so that users with basic knowledge would be able to traverse the site.
- Performance constraints: The response time to load for the request should not be longer than 2 seconds.
- Traffic constraints: The web portal needs to be able to handle a big amount of capacity as the database might need to queue the requests, which would subsequently increase the amount of time needed to fetch data.
- Database constraints: All user information must be stored in a central database that is accessible by the DUX system.

3. System Architectural Design

3.1 Architecture Pattern and Rationale

For the development of our sharing Google Drive(GD) resources platform, we picked the **Model-View-Controller (MVC)** architectural pattern. The Model, View, and Controller are the three components of the MVC pattern. These elements are linked and work together to form an organized and modular structure.

Model:

- Manages the structure, storage , and retrieval of DUX system information , user data and system functionality.
- Represents the data and business logic of the DUX system.

View:

- Represents the user interface and presentation layer.
- Displays student listings, detailed information and facilitates user interactions
- Ensures a clean separation from underlying data and logic

Controller:

- Manages user inputs and interactions.
- Orchestrates the flow of data between the Model and View.
- Handle user requests,process data, and update the Model and View accordingly.

3.2 Component Model

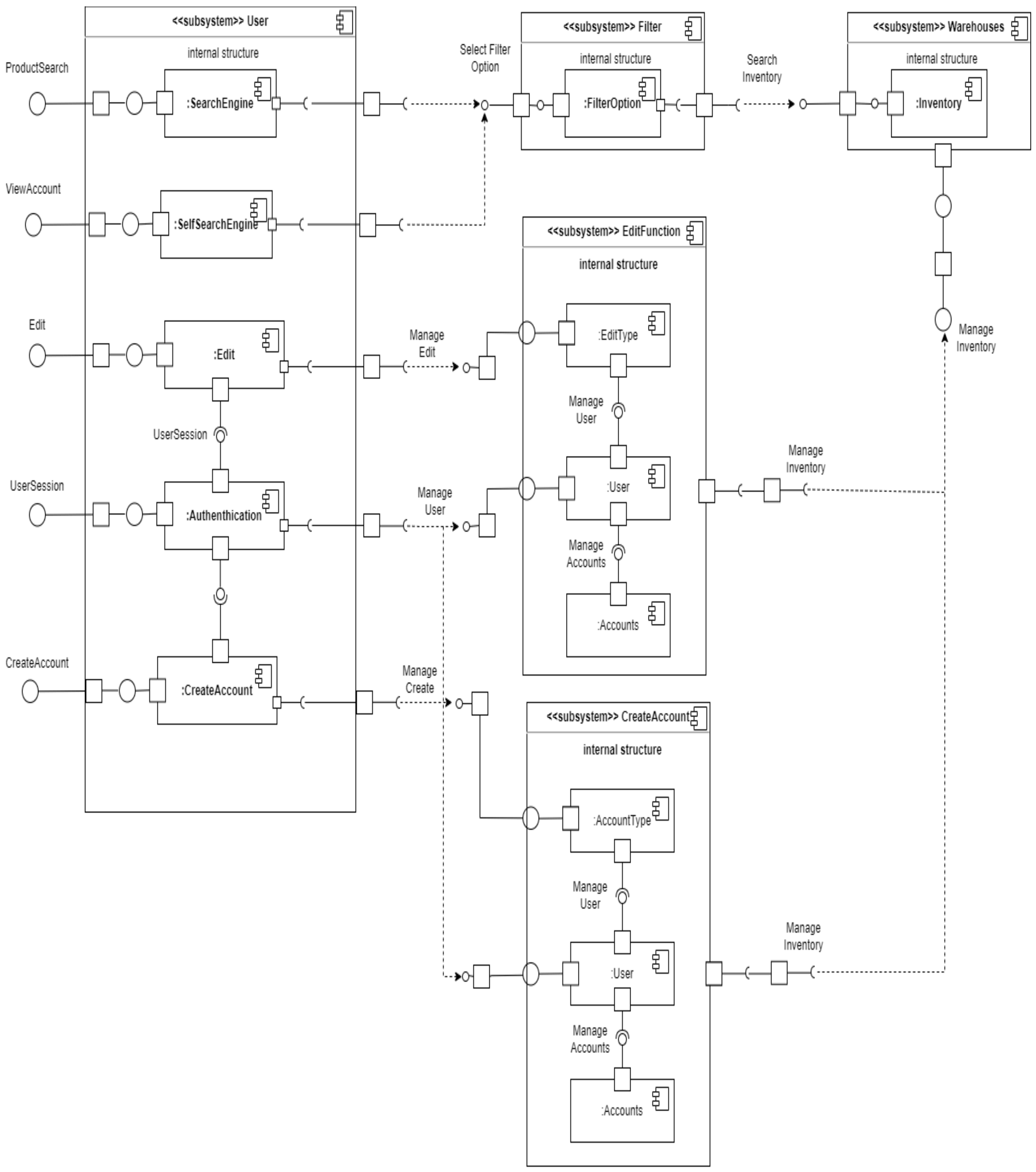


Figure 12.0: Component Model for Dux System

3.3 Complete Package Diagram

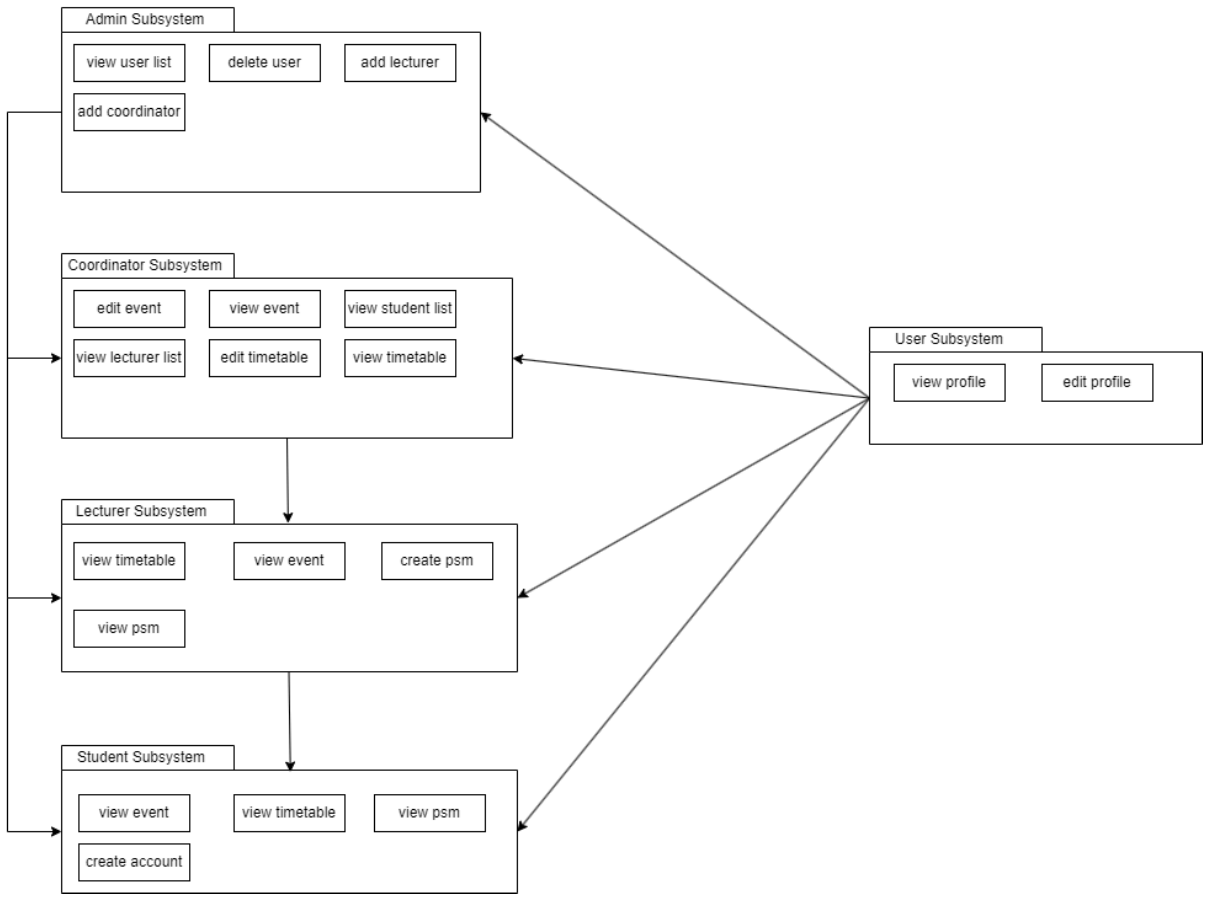


Figure 13.0: Package Diagram for Dux System

3.4 Class Diagram

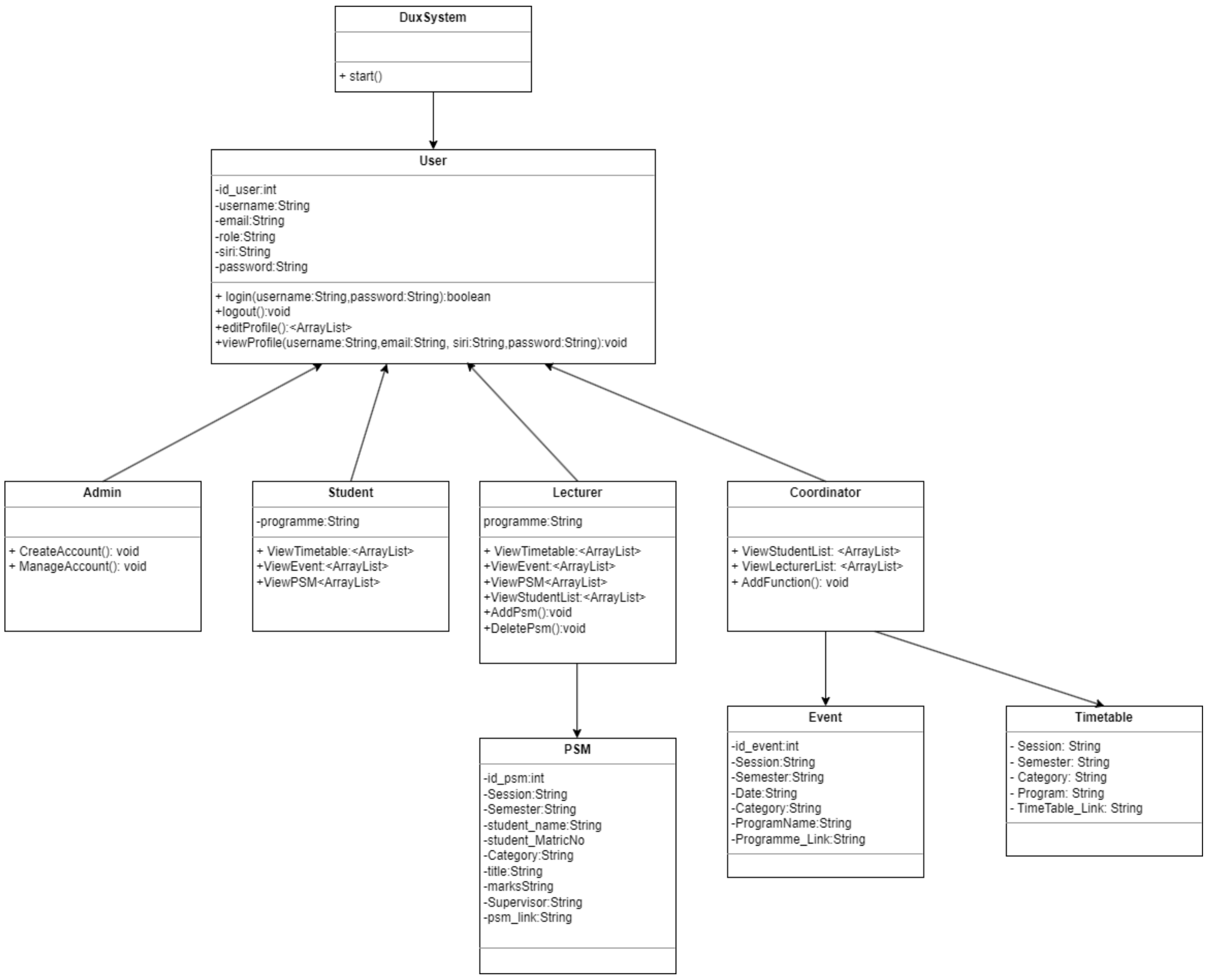


Figure 14.0: Class Diagram for Dux System

4. Data Design

4.1 Data Description

The major data or systems entities are stored into a relational database named as DuxSys, processed and organized into 6 entities as listed in Table 4.1.

Table 4.1 Description of Entities in the Database

No	Entity Name	Description
1.	user	holds basic information of the user registered into the system
2.	student	user registered as student that can use the platform to view the data
3.	lecturer	user registered as lecturer that can use the platform to view and edit data
4.	timetable	entity which gives information to both lecturer and student
5.	event	entity which gives information about any programme being held
6.	PSM	entity which lecturer can gives mark to student for their final year project

4.2 Data Dictionary

4.2.1 Entity:user

Attribute Name	Type	Description
id_user	int	Unique user ID. Primary Key
username	Varchar	Unique username
email	Varchar	Email of user
role	Varchar	Role of user (lecturer,student,admin,coordinator)
siri	Varchar	Matric Number of user
password	Varchar	Password of user

4.2.2 Entity:student

Attribute Name	Type	Description
matric	Varchar	Unique student ID. Primary Key
S_name	Varchar	Student name
S_email	Varchar	Student Email
S_role	Varchar	Their role
programme	Varchar	Student’s Programme (SECRH,SECJH)

4.2.3 Entity:lecturer

Attribute Name	Type	Description
LectID	Varchar	Unique lecturer ID. Primary Key
L_name	Varchar	Lecturer name
L_email	Varchar	Lecturer Email
L_role	Varchar	Their role
programme	Varchar	Lecturer’s Programme (SECRH,SECJH)

4.2.4 Entity:timetable

Attribute Name	Type	Description
id_TimeTable	Int	Unique TimeTable ID. Primary Key
Session	Varchar	Year of the timetable (2019/2020)
Semester	Varchar	Semester of the timetable (sem 1 or sem 2)
Category	Varchar	Used for undergraduate or postgraduate
Programme	Varchar	Majoring programme (SECRH, SECJH)
TimeTable_Link	Varchar	Link to be view by the user when clicked

4.2.5 Entity:event

Attribute Name	**Type**	**Description**
id_event	Int	Unique event ID. Primary Key
Session	Varchar	Year of the timetable (2019/2020)
Semester	Varchar	Semester of the timetable (sem 1 or sem 2)
Date	Varchar	The date of the event that are going to be held
Category	Varchar	Used for undergraduate or postgraduate
ProgramName	Varchar	The name of the event that are going to be held
Programme_Link	Varchar	Link to be view by the user when clicked

4.2.6 Entity:PSM

Attribute Name	**Type**	**Description**
id_psm	Int	Unique psm ID. Primary Key
Session	Varchar	Year of the timetable (2019/2020)
Semester	Varchar	Semester of the timetable (sem 1 or sem 2)
Category	Varchar	Used for undergraduate or postgraduate
student_name	Varchar	name of the student
student_MatricNo	Varchar	matric number of the student
title	Varchar	title of the final year project
marks	Varchar	marks of the project
Supervisor	Varchar	name of the supervisor
psm_link	Varchar	Link to be view by the user when clicked