

**Software Engineering and Management**

**Nanjing University of Aeronautics and Astronautics**

**Group name:** CodeBrewers

**Group members:**

Jirapat 许智明 191964107

Tasnim Afra 泰斯 191951601

Salim 萨利姆 191964138

Md Shahariar Idris robin 罗宾 191964134

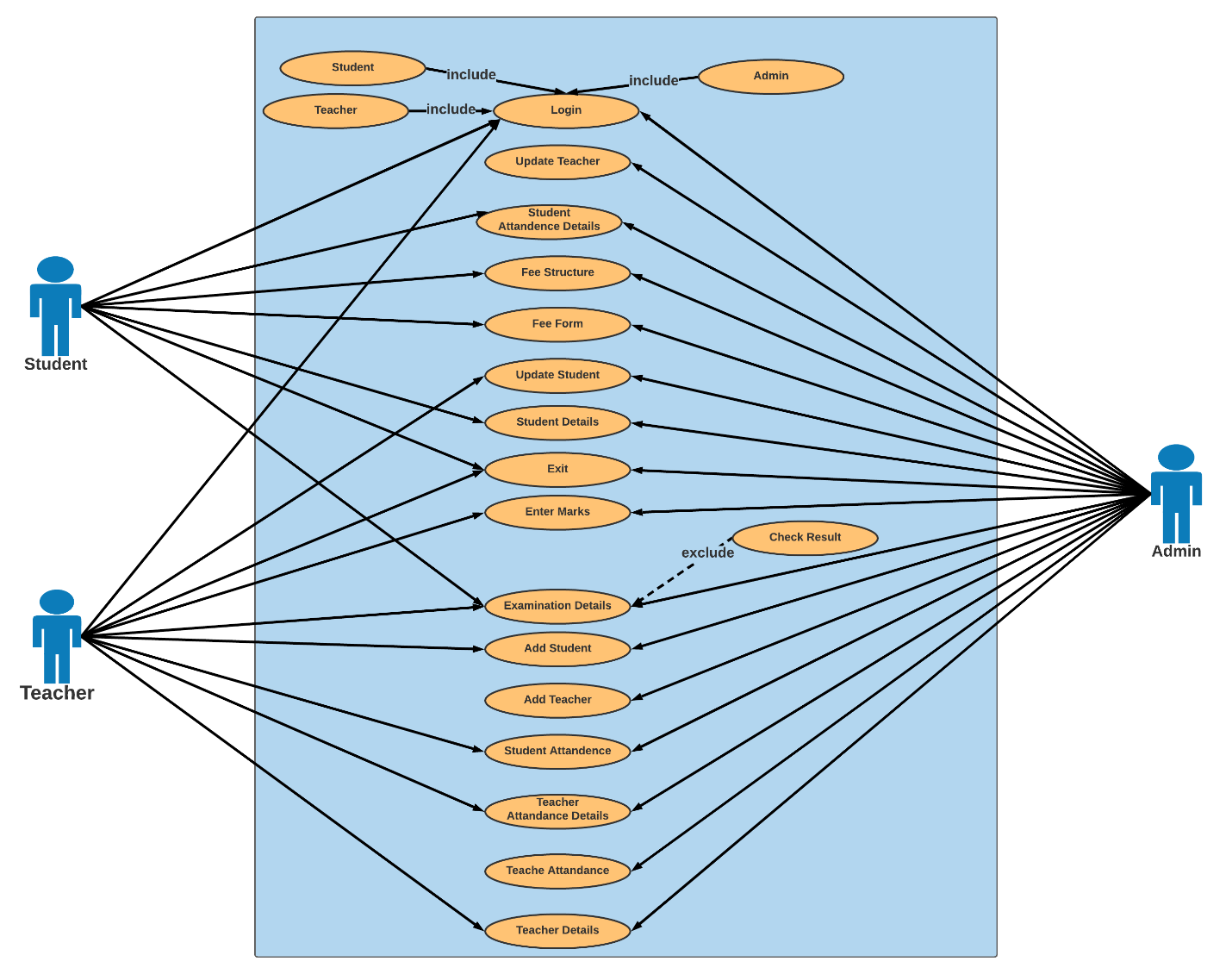
**Teacher’s name:** 王铁鑫

**Course:** Object-Oriented Method & UML Modeling

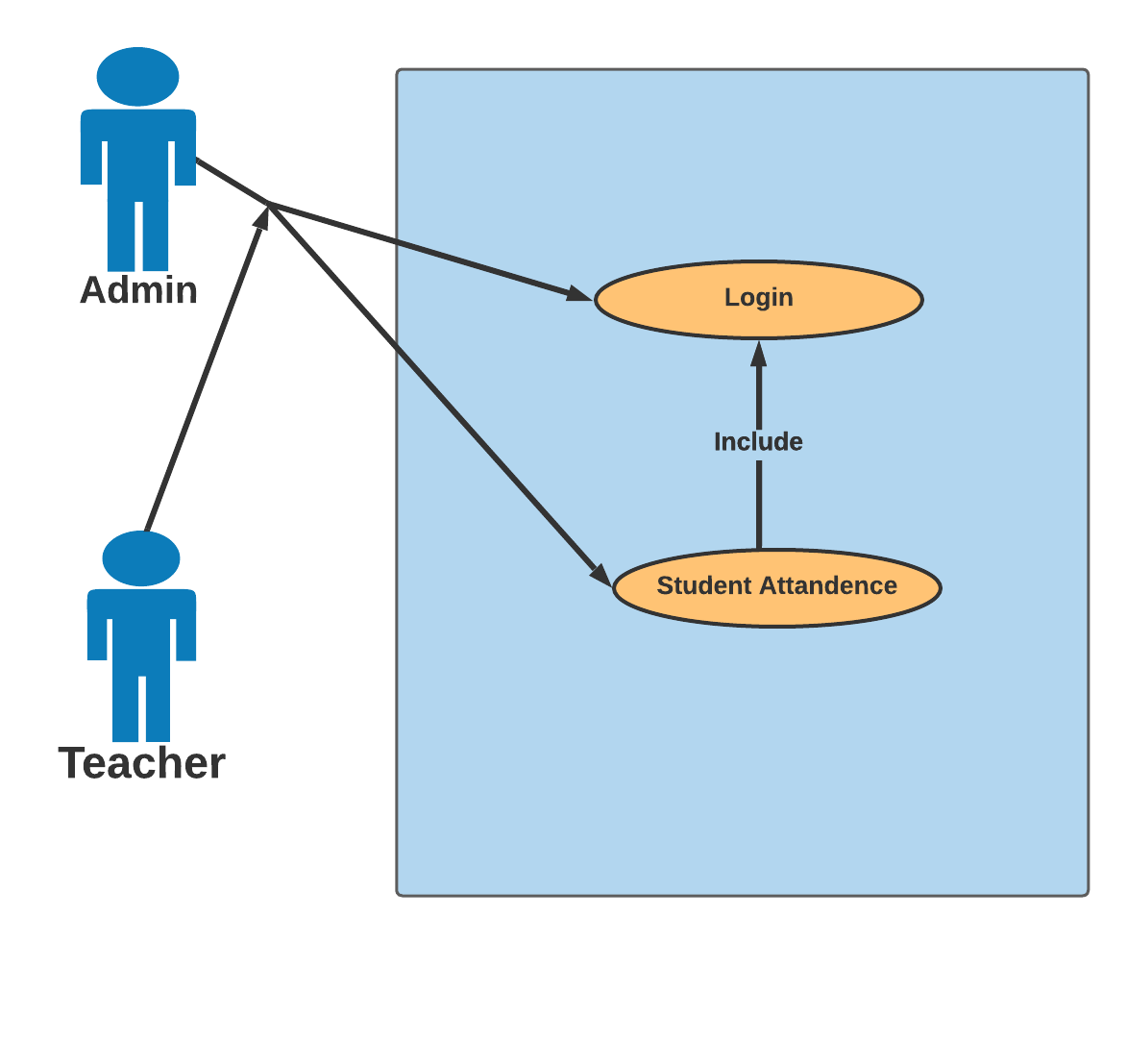
**Date:** 21/12/2021

1. **Use case diagram**

A usecasediagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

****

**Use case 1: Student Attendance**



**Requirement** **specification**:

**Name**: Student attendance

**Description**: Allow admin and teacher to give attendance to the student

**Level**: Admin & Teacher

**Pre**-**Condition**: The user login into the system with the admin login system or the teacher login system.

**Post**-**Condition**:   Select student ID, and select options in the first half second half and submit.

**Basic** **path**: 1. Teacher login into the system

2. Select the attendance option and choose student attendance

3. select specific student ID.

4. select the option for the first and second half (Present, absent, and leave)

5. Click on the ‘Submit’ option.

**Alternative** **Path**: 1. Admin login into the system

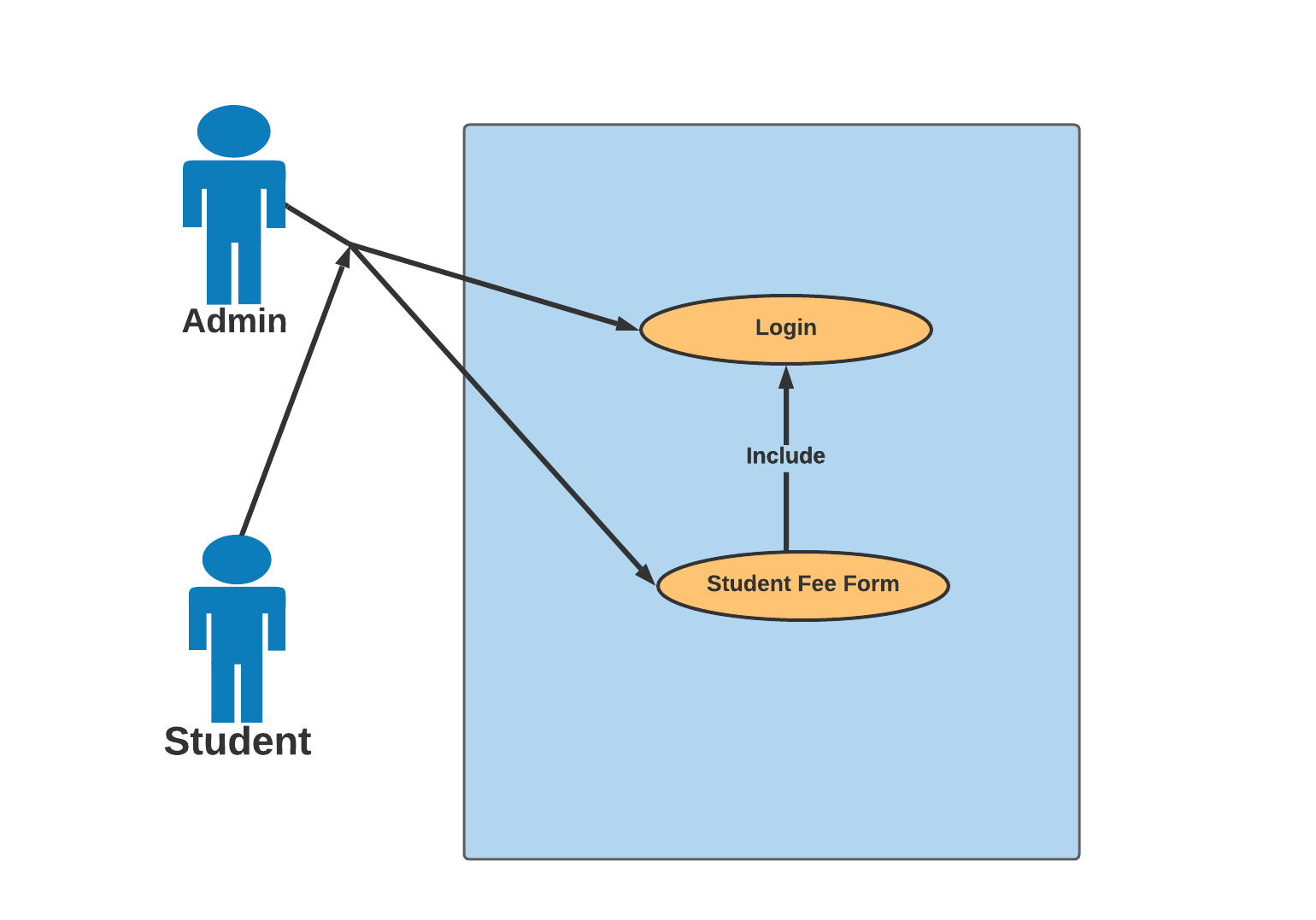
2. Select the attendance option and choose student attendance

3. select specific student ID.

4. select the option for the first and second half (Present, absent, and leave)

5. Click on the submit option.

**Use Case 2: Student fee *form***



**Requirement** **specification**:

**Name**: Student fee form

**Description**: Allow students and admin to apply for the student fee form.

**Level**: Admin & Student

**Pre**-**Condition**: The user login into the system with the student login system or the admin login system.

**Post**-**Condition**:   Select student ID and fill in the rest information

**Basic** **path**: 1. Student login into the system.

2. Click on ‘Fee Details’ and select ‘Student Fee Form’

3. Select student ID and complete the rest requirement.

4. Click on the ‘Pay’ option.

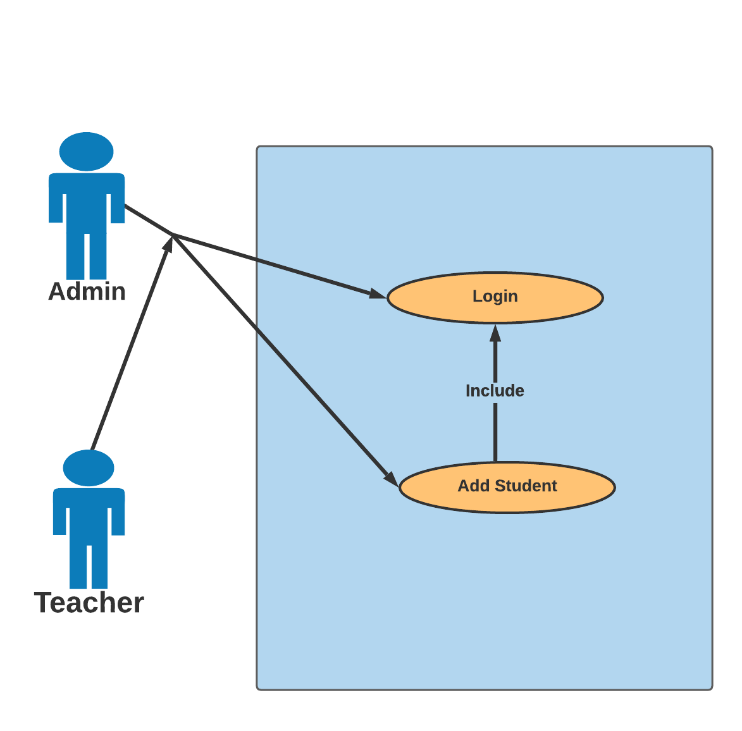
**Alternative** **Path**: 1. Admin login into the system.

2. Click on ‘Fee Details’ and select ‘Student Fee Form’

3. Select student ID and complete the rest requirement.

4. Click on the ‘Pay’ option.

**Use Case 3: Add Student:**



**Requirement** **specification**:

**Name**: Add Student

**Description**: Allow admin to add student in the system.

**Level**: Admin & Teacher.

**Pre**-**Condition**: User must be login in the system

**Post**-**Condition**:   Student details saved in database

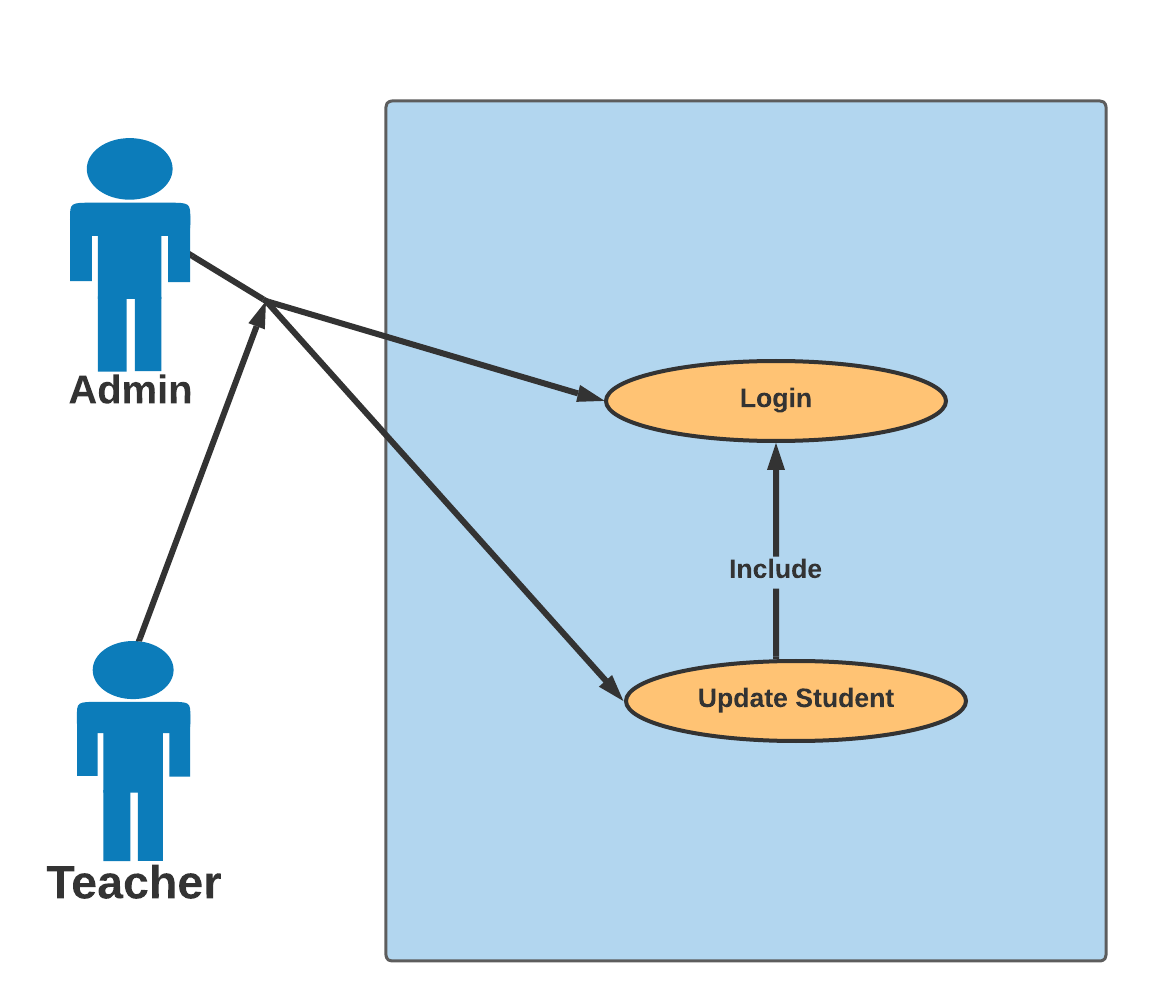
**Basic** **path**:

1. Admin logs on the system.
2. Admin entries all the basic information of student.
3. Select a specific course and Branch.
4. Displays student details inserted successfully

**Alternative** **Path**:

1. Teacher logs on the system.
2. Teacher entries all the basic information of student.
3. Select Id
4. Select a specific course and Branch.
5. Displays student details inserted successfully

**Use Case 4: Update Student**



**Requirement** **specification**:

**Name**: Update Student

**Description**: Admin and teacher allowed to update student.

**Level**: Admin & Teacher.

**Pre**-**Condition**: User must be login in the system and must add student to update specific student.

**Post**-**Condition**:   Student details updated in database and information’s updated in student details

**Basic** **path**:

1. Admin logs on the system.
2. Go to the student update panel
3. Select student Id
4. Add or delete entries
5. Displays student details Successfully updated

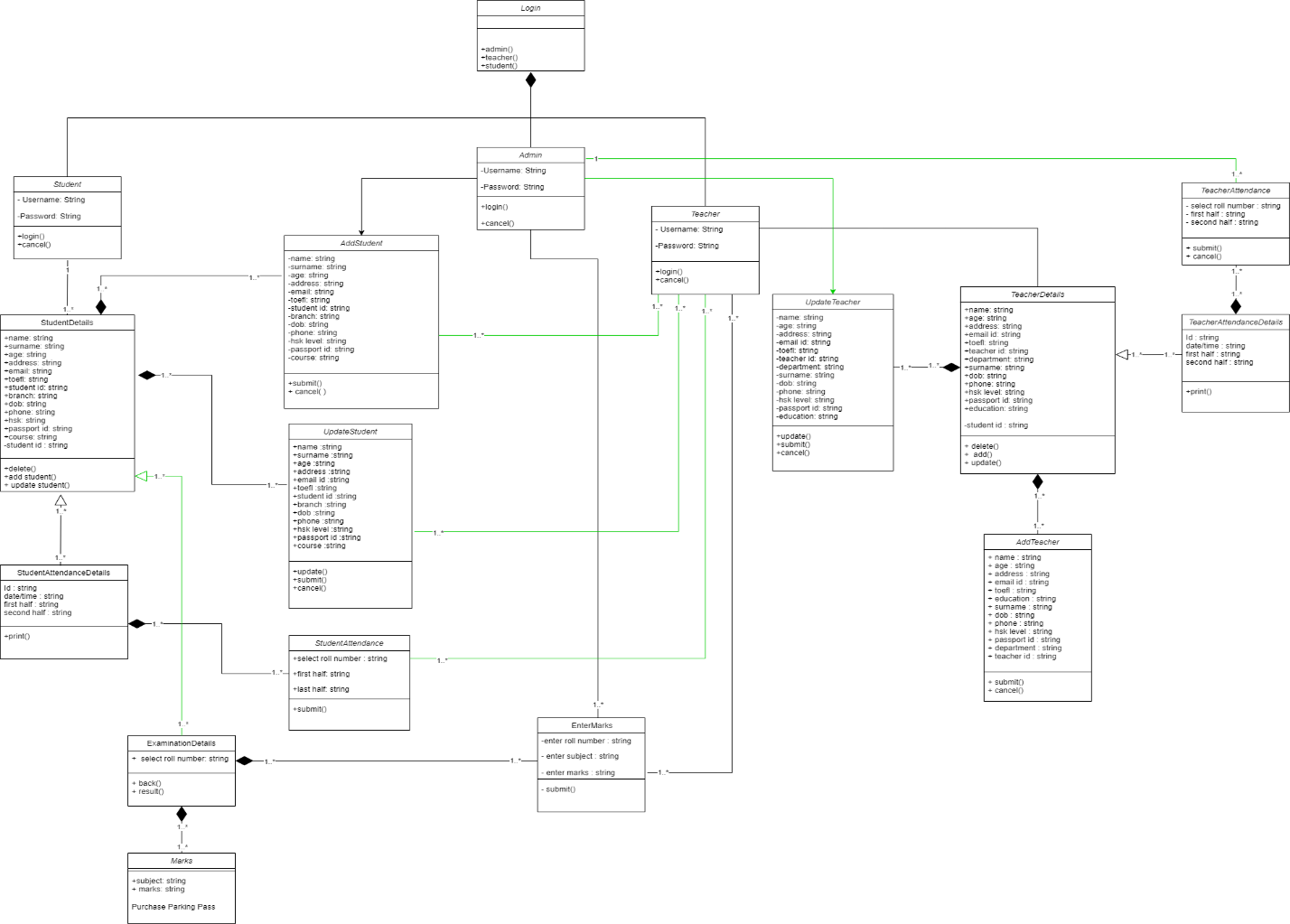
**Alternative** **Path**:

1. Teacher logs on the system.
2. Go to the student update panel Select student Id
3. Add or delete the entries
4. Displays student details Successfully updated
5. **Class Diagram**

The class diagram is the main building block of [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) modeling. It is used for general [conceptual modeling](https://en.wikipedia.org/wiki/Conceptual_model) of the structure of the application, and for detailed modeling, translating the models into [programming code](https://en.wikipedia.org/wiki/Programming_code). Class diagrams can also be used for [data modeling](https://en.wikipedia.org/wiki/Data_modeling).[[1]](https://en.wikipedia.org/wiki/Class_diagram#cite_note-1) The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

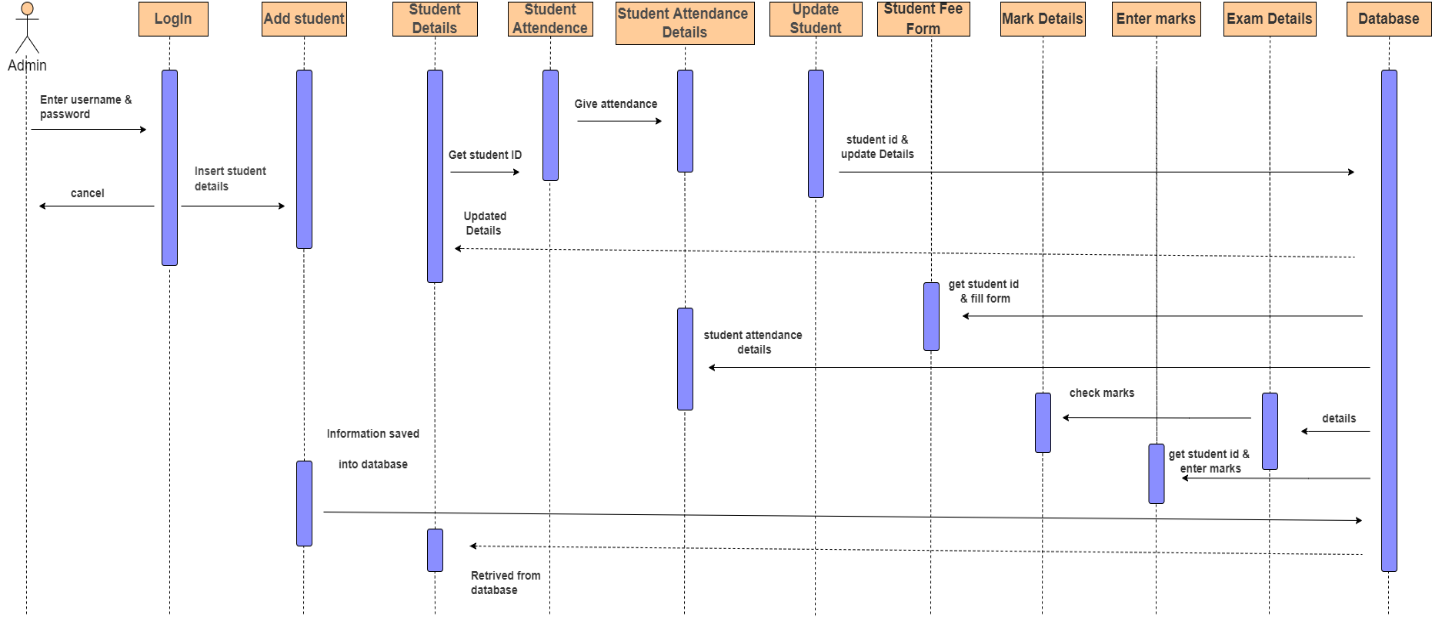
Link for class diagram:

https://drive.google.com/file/d/1e45ZRj8ZM85MymJjZ8VYWZ8QEqMk0DYQ/view?usp=sharing



1. **Sequence diagram**

A sequence diagram or system sequence diagram (SSD) shows object interactions arranged in time sequence in the field of [software engineering](https://en.wikipedia.org/wiki/Software_engineering). It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of scenario. Sequence diagrams are typically associated with use case realizations in the [logical view](https://en.wikipedia.org/wiki/4%2B1_architectural_view_model) of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.



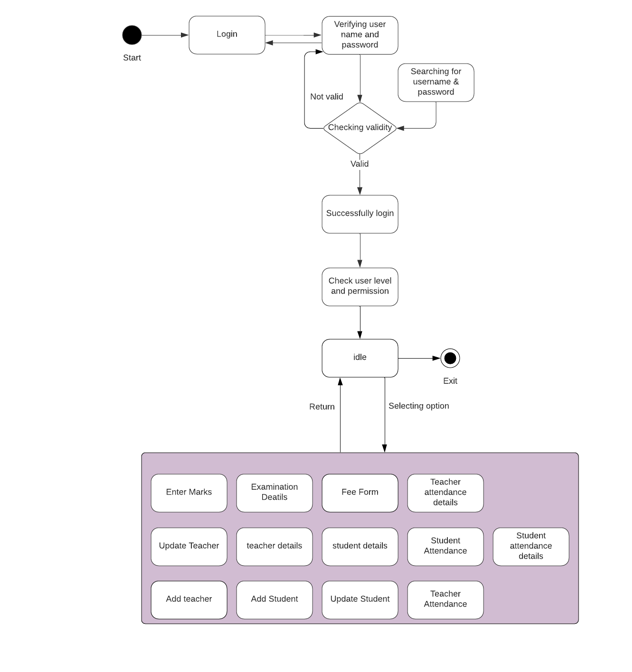
This university management system sequence diagram shows a detailed description of a state. The state is extended with internal actions. In contrast to other actions, these actions can be interrupted.

In the sequence diagram, where admin will be able to login into their account with their credentials. Admin user has the abilities to manage all the operations on ‘add student’, ‘student details’,’ student attendance’, ‘student attendance details’, ‘update student’, ‘student fee form’.

Where after adding students, the information will be saved into the database then retrieve show in the student details class. With the information that has already been stored in the database, the user will be able to use any functions by a quote of the student id. if any update for any activity takes place in the system the data will be stored according to that specific id.

1. **State Chart Diagram**

A state diagram is a type of [diagram](https://en.wikipedia.org/wiki/Diagram) used in [computer science](https://en.wikipedia.org/wiki/Computer_science) and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of [states](https://en.wikipedia.org/wiki/State_(computer_science)); sometimes, this is indeed the case, while at other times this is a reasonable [abstraction](https://en.wikipedia.org/wiki/Abstraction_(computer_science)). Many forms of state diagrams exist, which differ slightly and have different [semantics](https://en.wikipedia.org/wiki/Semantics#Computer_science).



State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. In our program at the first point after verifying the user, the user enters the system. Therefrom the initial state the user can move to any activity.

 In the initial state shows the user will be able to log in using their username and password. After login user will be checked for the user level and permission for access to any state. All the states on both student and teacher can be accessed by the admin user. On the other hand, student and teachers can access only some specific states.