

***Software Requirement and Design Specifications***

***School Management System***

***Version: 1.0***

|  |  |
| --- | --- |
| *Course Code* | *SE 2002* |
| *Instructor* | *Syeda Rubab Jaffar* |
| *Project Team* | *21k-3848 (Muneeb Hasan)*  *21k-3845 (Taha Siddiqui)* |
| *Submission Date* | *30th-April-2023* |



*Table of Contents*

[1. INTRODUCTION 5](#_Toc20720)

[1.1. Purpose of Document 5](#_Toc20721)

[1.2. Intended Audience 5](#_Toc20722)

[2. OVERALL SYSTEM DESCRIPTION 6](#_Toc20723)

[2.1. Project Background 6](#_Toc20724)

[2.2. Project Scope 6](#_Toc20725)

[2.3. Not In Scope 6](#_Toc20726)

[2.4. Project Objectives 6](#_Toc20727)

[2.5. Stakeholders 6](#_Toc20728)

[2.6. Operating Environment 6](#_Toc20729)

[2.7. System Constraints 6](#_Toc20730)

[2.8. Assumptions & Dependencies 6](#_Toc20731)

[3. EXTERNAL INTERFACE REQUIREMENTS 7](#_Toc20732)

[3.1. Hardware Interfaces 7](#_Toc20733)

[3.2. Software Interfaces 7](#_Toc20734)

[3.3. Communications Interfaces 7](#_Toc20735)

[4. FUNCTIONAL REQUIREMENTS 8](#_Toc20736)

[4.1. FUNCTIONAL HIERARCHY 8](#_Toc20737)

[4.2. Use Cases 8](#_Toc20738)

[4.2.1. School Management System 8](#_Toc20739)

[5. NON-FUNCTIONAL REQUIREMENTS 9](#_Toc20740)

[5.1. Performance Requirements 9](#_Toc20741)

[5.2. Safety Requirements 9](#_Toc20742)

[5.3. Security Requirements 9](#_Toc20743)

[5.4. User Documentation 9](#_Toc20744)

[SDS 10](#_Toc20745)

[6. SYSTEM ARCHITECTURE 11](#_Toc20746)

[6.1. SYSTEM LEVEL ARCHITECTURE 11](#_Toc20747)

[6.2. SOFTWARE ARCHITECTURE 11](#_Toc20748)

[7. DESIGN STRATEGY 12](#_Toc20749)

[8. DETAILED SYSTEM DESIGN 13](#_Toc20750)

[8.1. DATABASE DESIGN 13](#_Toc20751)

[9. APPLICATION DESIGN 15](#_Toc20752)

[10. REFERENCES 15](#_Toc20753)

[11. APPENDICES 17](#_Toc20754)

## 1. Introduction

#### 1.1. Purpose of Document

The purpose of this document to explain what our project is supposed to do. It gives complete overview of the functionalities provided by our web application which provide ease for School to manage all the internal records. Our customers are the students, teachers. Our system will also help parents to check their children’s progress in the school.

#### 1.2. Intended Audience

1. *Admin*
2. *Parent*
3. *Teacher*
4. *Student*

**1.3. Definition of Terms, Acronyms and Abbreviations**

Not Applicable

***1.4 Document Convention***

*Font used is Times New Roman*

*Font Size used is 12, 14, 16.*

## 2. Overall System Description

#### 2.1. Project Background

The Project is used by school administration. In the project school can do multiple functions. Developer is responsible for: developing the system, hosting the software on the client’s hardware, conducting any user training that might be needed for using the system through soft copy of technical manual/report and presentation slides, and maintaining the system for a period of one year after the hosting.

#### 2.2. Project Scope

**Administrator:**

The admin of the school deals with the students, teachers’ registration and also configure them. He also keeps track of attendance status of all students then calculate the percentage of each student’s Attendance and also manages subjects of particular class.

**Student/Guardian:**

Students can also access the web application. They can view their personal and Academic information on the home page, check their overall Attendance percentage and also check their attendance of daily basis. Guardians also have rights to access student account to view their children progress.

**Teacher:**

Teacher can view their profile, take and modify attendance of students.

#### 2.3. Not In Scope

I will work further on Students Marks and Fees details functionalities in future.

#### 2.4. Project Objectives

Our objectives for this project are that user can login to a School and then can perform multiple functionalities. There is typically a primary purpose for any schoolwebsite to provide timely, engaging, and current information to parents, students and teachers.

#### 2.5. Stakeholders

Admin: Have all authorities.

Teacher: Can Mark and modify attendance of Students of his Class.

Student: Can View Academic Records.

Parent: Can View his Children’s Progress.

#### 2.6. Operating Environment

Hardware requirement is Computer with proper wifi connection on any windows and system. The system should run have any Window 32 or 64 bit, 250MB plus RAM, 40GB HDD. The most important is user should have internet access.

* Software requirement is any browser (such as: Chrome, firefox, etc).

#### 2.7. System Constraints

* The system should run have any Window 32 or 64 bit, 250MB plus RAM, 40GB HDD. The most important is user should have internet access.
* Software requirement is browser (Chrome, firefox, etc).
* The main user of this system will be Parents, Student, Teacher, and Admin. Who are somewhat literate with computers and know what to do.
* Software constraints: Only those people can access the website functionalities who are related to school only.
* Hardware constraints: Only run on those devices which support web application.
* Cultural constraints: This website uses English language only.
* Legal constraints: Only those users can use the website whose id and password are provided by the admin.
* User constraints: people age >=3

#### 2.8. Assumptions & Dependencies

* Internet should be available and server should be running properly

## 3. External Interface Requirements

This website which we are building can be accessible by any one which have proper internet connection and connection on our local host server.

#### 3.1. Hardware Interfaces

* Computer as Server
* Computer as Client
* PHP API for connection and data retrieval.

#### 3.2. Software Interfaces

* Tools used are HTML, CSS, JavaScript, PHP for connection.
* Tools used for Database is MySQL.
* Communication is done by client sending signal to server and data is retrieve through route calls and shown on his profile page.

#### 3.3. Communications Interfaces

* *Not Applicable.*

## 4. Functional Requirements

### *4.1. Functional Hierarchy*

* The admin can view and add Users Data. Student and Teachers can view their given data which admin makes.
* Admin can add/modify/delete Students, Classes, teachers, Sections, Attendance.
* Only Admin and teachers can mark Attendance of Classes as well as Individual Students.
* Teacher can login to Teacher’s Portal by inserting his email and password.
* Teachers can also change their password by providing previous(old) password.
* Students and Parents can login by providing student’s email address and date of birth.
* Students and Parents can view Student’s Personal information, Attendance Status, Class detail, Subjects information.
* Any correction in Student or Teacher’s Profile can only be made through admin.

#### 4.2. Use Cases

##### 4.2.1. School Management System

##### 

|  |  |
| --- | --- |
| ***Use Case Description*** | |
| ***Use Case name:*** | *School Management System* |
| ***Use Case Description:*** | |
| ***Primary actor:*** | ***Other actors:*** *Parent* |
| ***Stakeholders:*** | Admin, Teacher, Student. |
| ***Relationships*** ***Includes:*** *-*   ***Extends:***  *-* | |
| ***Flow of Events:***   1. The admin has a right to update the Teacher and student information. 2. Students can view their Personal details, Subjects details, class details and Attendance Status. 3. Teacher can take Attendance of their Class Students, can also view his Personal information, class information. 4. Parents can view their child’s Details, such as: personal info, class, subjects, attendance. | |

## 5. Non-functional Requirements

#### 5.1. Performance Requirements

The System is very high speed and feasible to the user, it is not a complex system it has a different section for every work so user can navigate through it easily which take very less time give a plus point of speed efficiency. This System is also secure, no user can access the personal info of any other user, for each and every work it maintains database and give a positive point of concurrency and provide concurrent data to user.

#### 5.2. Safety Requirements

Not applicable

#### 5.3. Security Requirements

Initially only the registered user can login the system whether it is a teacher student or Admin. It also validates all types of form whether user input valid details or not. The system is secure in all aspect.

#### 5.4. User Documentation

Not applicable

***SDS***

# 6. System Architecture

The hierarchical overview of the system there are three section Admin, Teacher and Student each are working separately from each other some functionalities as Students can view Attendance of daily basis that can be modify by teacher and Admin.

Except that all three portions (Admin, teacher, and student) work in isolation with each other’s.

### 6.1. System Level Architecture

The hierarchical overview of the system there are three section Admin, Teacher and Student each are working separately from each other some functionalities as Students can view Attendance of daily basis that can be modify by teacher and Admin.

Except that all three portions (Admin, teacher, and student) work in isolation with each other’s.

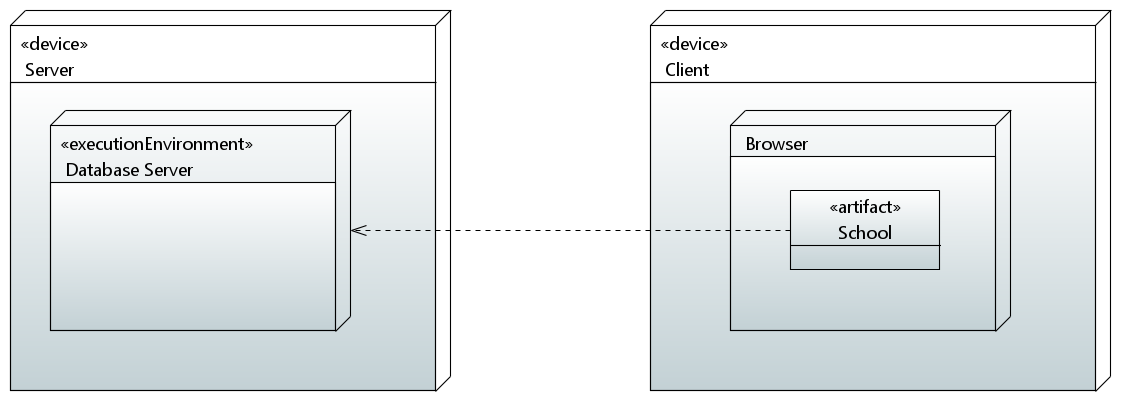
*NOTE: You will use UML diagrams (Deployment and Component diagrams) to document the overall system architecture.*

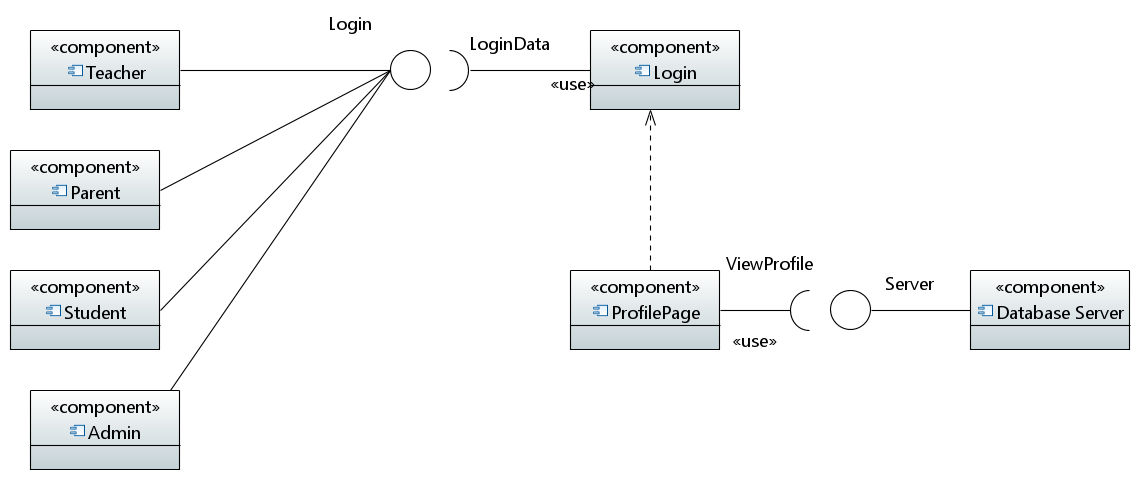
### 6.2. Software Architecture

We have used HTML for structure and CSS for design purpose, also we have used JavaScript for responsive on our website PHP is main language in our system plus we have used PHP and MySQL as core for our backend function i.e., route calling and fetching data from XAMPP server Database.

The main requirement for client is that, they should have XAMPP Local Host connected else website will not run because we have used XAMPP for our database storage.

***Deployment Diagram:***



***Component Diagram:*** 

# 7. Design Strategy

Many different adaptations, tests, and experiments have been left for the future due to lack of time (i.e., the experiments with real data are usually very time consuming, requiring even days to finish a single run). Future work concerns deeper analysis of particular mechanisms, new proposals to try different methods, or simply curiosity. There are some ideas that I would have liked to try during the description and the development such as System more complex than FLEX.

* To sum up our system is like Flex and Slate somehow but not implemented fully, by given more time we can build a fully hybrid system which will be powerful and fast.
* We can use AI to our system and make it better and cheaper.
* We can build mobile app and desktop app for our website for convenient.

# 8. Detailed System Design

**Schemas Of Database:**

**Admin:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Username | Password | Email\_address | Phone\_no |

***Student\_info:***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | Name | Father\_Name | Phone\_no | Email\_address | Date\_of\_birth | Class\_id |

**Teacher\_info:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Teacher\_id | Name | Email\_address | Phone\_no | Date\_of\_birth | Salary | Subject\_id | password |

**Class\_info:**

|  |  |  |
| --- | --- | --- |
| Class\_id | Class\_teacher | No\_of\_Students |

**Subject\_info:**

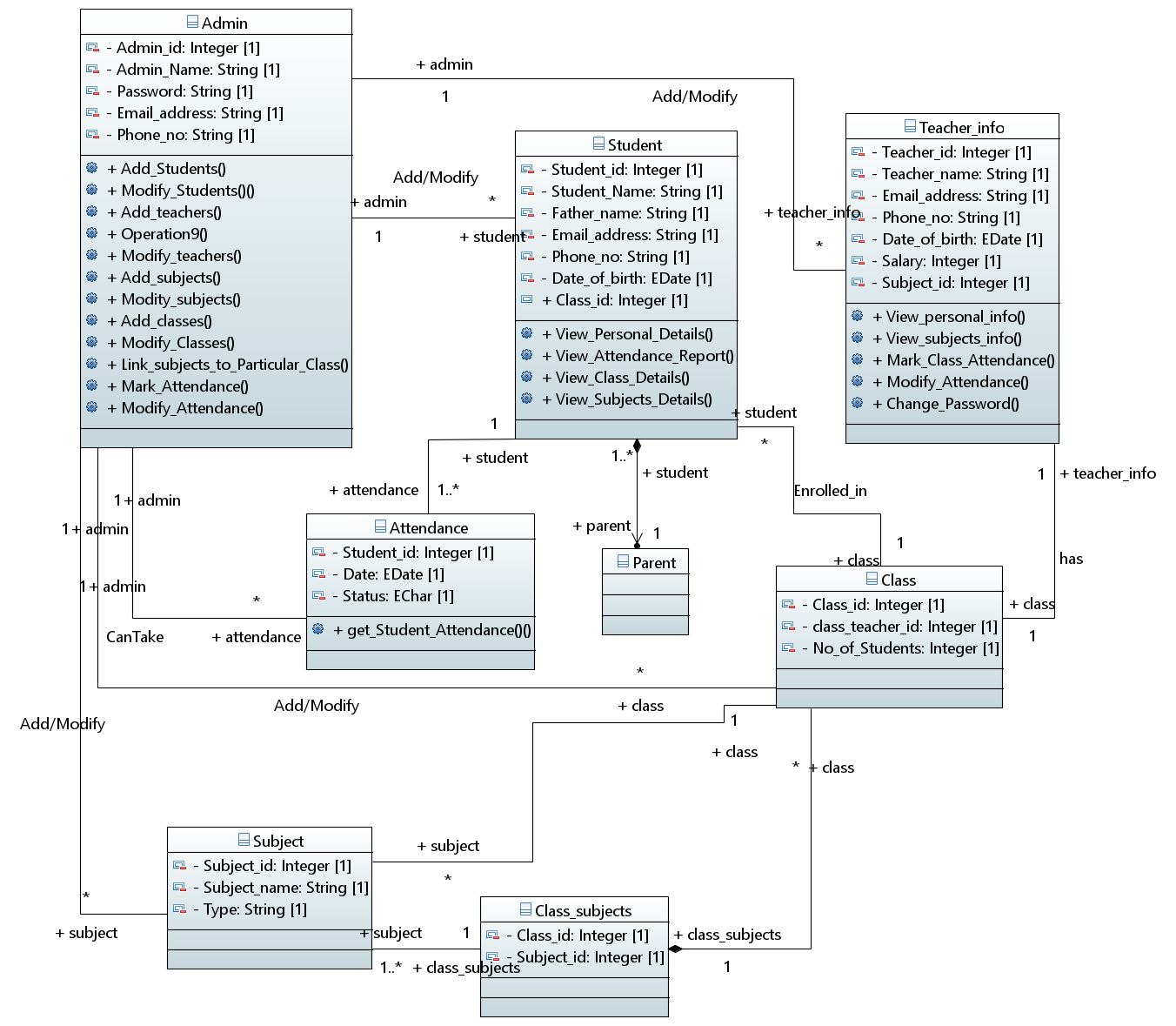
|  |  |  |
| --- | --- | --- |
| Subject\_id | Subject\_name | Type |

**Class\_subjects:**

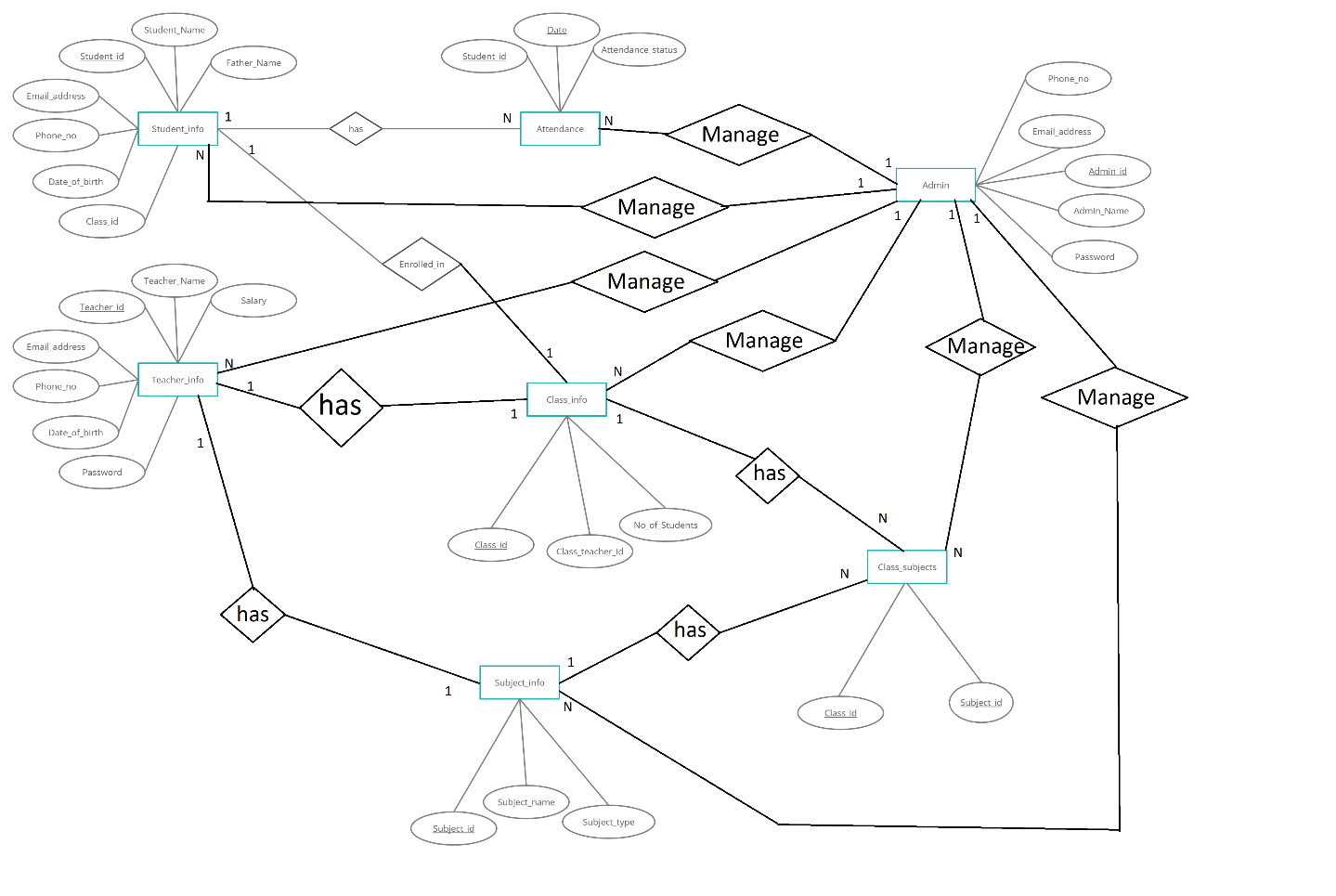
|  |  |
| --- | --- |
| Class\_id | Subject\_id |

**Attendance\_info:**

|  |  |  |
| --- | --- | --- |
| Class\_id | Class\_teacher | No\_of\_Students |

**Class Diagram:** 

### 8.1. Database Design

***8.1.1. ER Diagram:***

***8.1.2. Data Dictionary***

***8.1.2.1 Student\_info***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Student Information** | | | | | | | |
| ***Name*** | | *Student\_info* | | | | | |
| ***Alias*** | | *Student\_info* | | | | | |
| ***Where-used/howused*** | | *ViewPersonalDetails, ViewAttendanceReport, CheckClassDetails, ViewClassSubjects* | | | | | |
|  | | | | | | | |
| ***Column Name*** | ***Description*** | | ***Type*** | ***Length*** | ***Null able*** | ***Default Value*** | ***Key Type*** |
| *id* | *Unique number id for all Students* | | Integer | *11* | *No* | *None* | *PK* |
| Name | *Name of Student* | | *text* | - | *No* | *None* | *-* |
| Father\_Name | *Father Name of Student* | | *Text* | - | *No* | *None* | *-* |
| Phone\_no | *Phone Number of Student* | | *Text* | - | *No* | *None* | *-* |
| Email\_address | *Unique Email\_address for all students* | | *Text* | - | *No* | *None* | *-* |
| Date\_of\_Birth | *Date of Birth of Student* | | *Date* | - | *No* | *None* | *-* |
| Class\_id | *Class number in which Student is Currently studying.* | | *Integer* | 11 | *No* | *None* | *FK* |

8.1.2.2. Teacher Information

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Teacher Information** | | | | | | | |
| ***Name*** | | *Teacher info* | | | | | |
| ***Alias*** | | *Teacher\_info* | | | | | |
| ***Where-used/howused*** | | *MarkAttendanceOfClass, ModifyAttendance, ViewPersonalDetails, ChangePassword, ViewSubjectInfo.* | | | | | |
|  | | | | | | | |
| ***Column Name*** | ***Description*** | | ***Type*** | ***Length*** | ***Null able*** | ***Default Value*** | ***Key Type*** |
| *Teacher\_id* | *Unique number id for all Teachers.* | | Integer | *11* | *No* | *None* | *PK* |
| Name | *Name of Teacher.* | | *text* | - | *No* | *None* | *-* |
| Email\_address | *Unique Email\_address for all teachers* | | *Text* | - | *No* | *None* | *-* |
| Phone\_no | *Phone Number of teacher* | | *Text* | - | *No* | *None* | *-* |
| Date\_of\_Birth | *Date of Birth of teacher* | | *Date* | - | *No* | *None* | *-* |
| Salary | *Salary of Teacher* | | *Integer* | 11 | *No* | *None* | *-* |
| Subject\_id | *Subject number of subject that teacher teaches.* | | *Integer* | 11 | *No* | *None* | *FK* |
| Password | *Password of Teacher.* | | *Text* | - | *No* | *None* | *-* |

8.1.2.3. Admin Information

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Admin Information** | | | | | | | |
| ***Name*** | | *Admin* | | | | | |
| ***Alias*** | | *Admin\_info* | | | | | |
| ***Where-used/howused*** | | Add/Modify Students, Add/Modify Teachers, Add/Modify Subjects, Add/Modify Classes, Add/Modify Subjects of ParticularClass, Mark/Modify Attendance. | | | | | |
|  | | | | | | | |
| ***Column Name*** | ***Description*** | | ***Type*** | ***Length*** | ***Null able*** | ***Default Value*** | ***Key Type*** |
| *id* | *Unique id number for all Admin.* | | Integer | *11* | *No* | *None* | *PK* |
| Username | *Name of Admin* | | *Text* | - | *No* | *None* | *-* |
| Password | *Pasword for Admin* | | *Text* | - | *No* | *None* | *-* |
| Email\_address | *Unique Email\_address for all Admins.* | | *Text* | - | *No* | *None* | *-* |
| Phone\_no | *Phone Number of teacher* | | *Text* | - | *No* | *None* | *-* |

8.1.2.4. Attendance Information

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Attendance Information** | | | | | | | |
| ***Name*** | | *Attendance* | | | | | |
| ***Alias*** | | *Attendance* | | | | | |
| ***Where-used/howused*** | | *GetStudentAttendance* | | | | | |
|  | | | | | | | |
| ***Column Name*** | ***Description*** | | ***Type*** | ***Length*** | ***Null able*** | ***Default Value*** | ***Key Type*** |
| *Date* | *Date of Attendance for a student.* | | Date | *-* | *No* | *None* | *Composite PK,*  *Also, FK* |
| Student\_id | *Student id for Attendance for that Date* | | *integer* | 11 | *No* | *None* | Composite PK,  Also, FK |
| Status | *Attendance Status (i.e., “Present” or “Absent”)* | | *Varchar* | 1 | *No* | *None* | - |

8.1.2.5. Class Information

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Class Information** | | | | | | | |
| ***Name*** | | *Class* | | | | | |
| ***Alias*** | | Class\_info | | | | | |
| ***Where-used/howused*** | | *Managing records of Individual Classes.* | | | | | |
|  | | | | | | | |
| ***Column Name*** | ***Description*** | | ***Type*** | ***Length*** | ***Null able*** | ***Default Value*** | ***Key Type*** |
| *Class\_id* | *Unique number id for all Classes.* | | Integer | *11* | *No* | *None* | *PK* |
| Class\_teacher | *Unique class\_teacher\_id*  *For every class.* | | *Integer* | 11 | *No* | *None* | FK |
| Students | *Number of Students currently Enrolled in that class.* | | *Integer* | 11 | *No* | *None* | *-* |

8.1.2.6. Subject Information

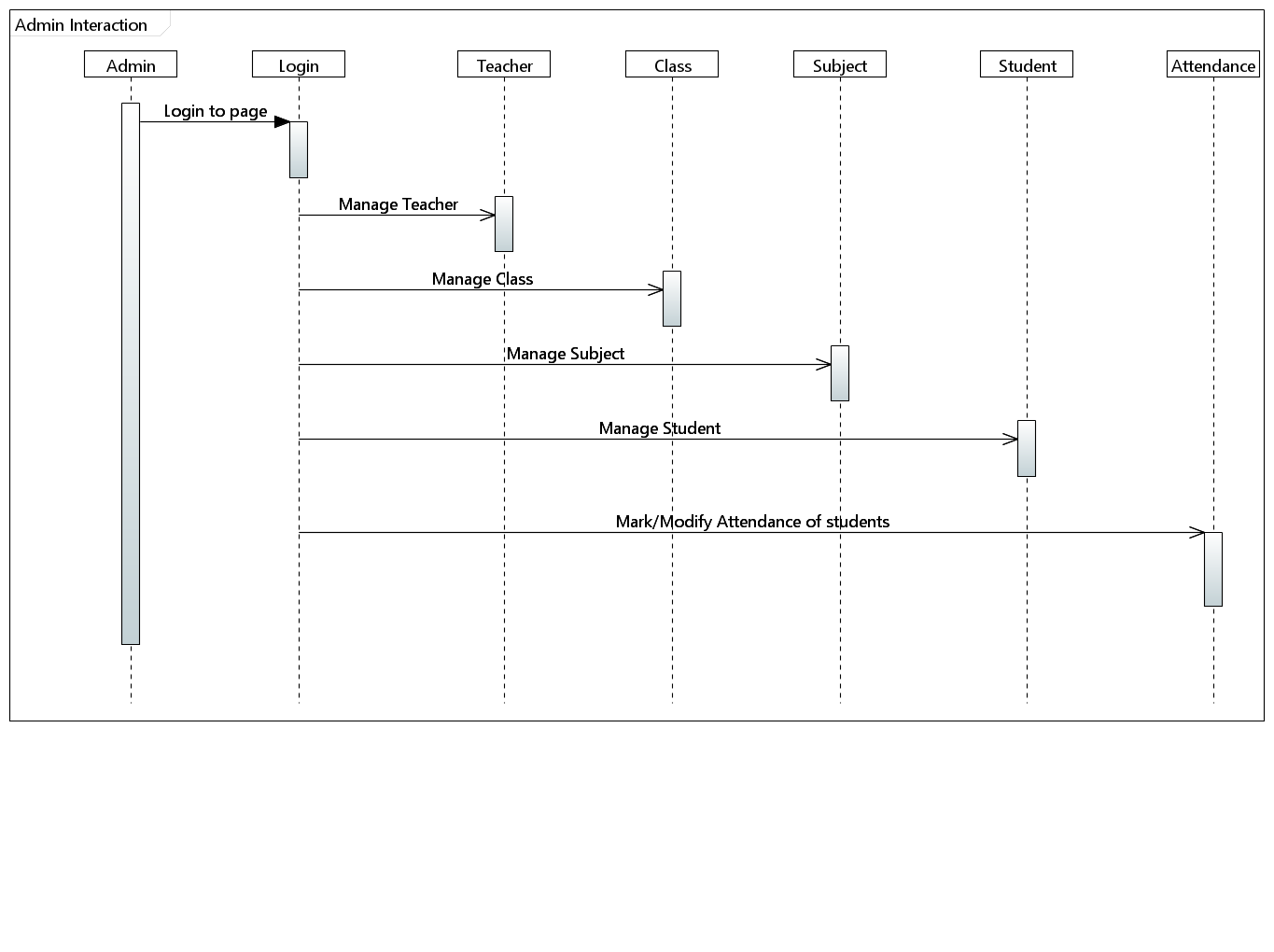
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Subject Information** | | | | | | | |
| ***Name*** | | *Subject* | | | | | |
| ***Alias*** | | *Subject\_info* | | | | | |
| ***Where-used/howused*** | | *Managing Records of All Subjects.* | | | | | |
|  | | | | | | | |
| ***Column Name*** | ***Description*** | | ***Type*** | ***Length*** | ***Null able*** | ***Default Value*** | ***Key Type*** |
| *Subject\_id* | *Unique number id for all Subjects.* | | Integer | *11* | *No* | *None* | *PK* |
| Subject\_name | *Name of Subject.* | | *text* | - | *No* | *None* | *-* |
| Type | *Type of Subject (i.e., Theory or Practical)* | | *Text* | - | *No* | *None* | *-* |

8.1.2.7. Class Subject Information

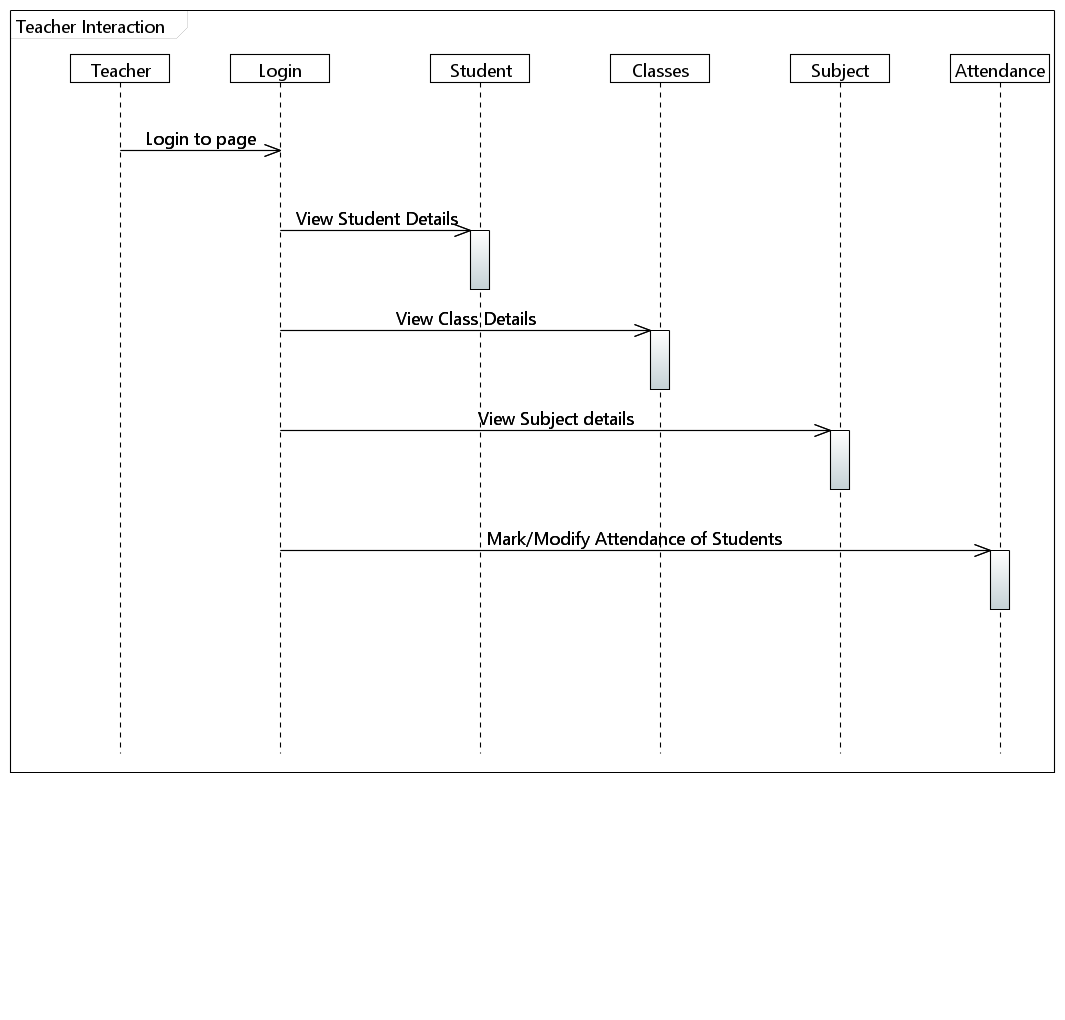
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **SubjectsOfClass Information** | | | | | | | |
| ***Name*** | | *Subjects of a particular class* | | | | | |
| ***Alias*** | | Class\_subjects | | | | | |
| ***Where-used/howused*** | | Keeps Records of Subjects of every class. | | | | | |
|  | | | | | | | |
| ***Column Name*** | ***Description*** | | ***Type*** | ***Length*** | ***Null able*** | ***Default Value*** | ***Key Type*** |
| *Class\_id* | *Id of a Class.* | | Integer | *11* | *No* | *None* | FK, and Composite PK |
| Subject\_id | *Subject ids of that above class\_id.* | | *Integer* | 11 | *No* | *None* | *FK, and Composite PK* |

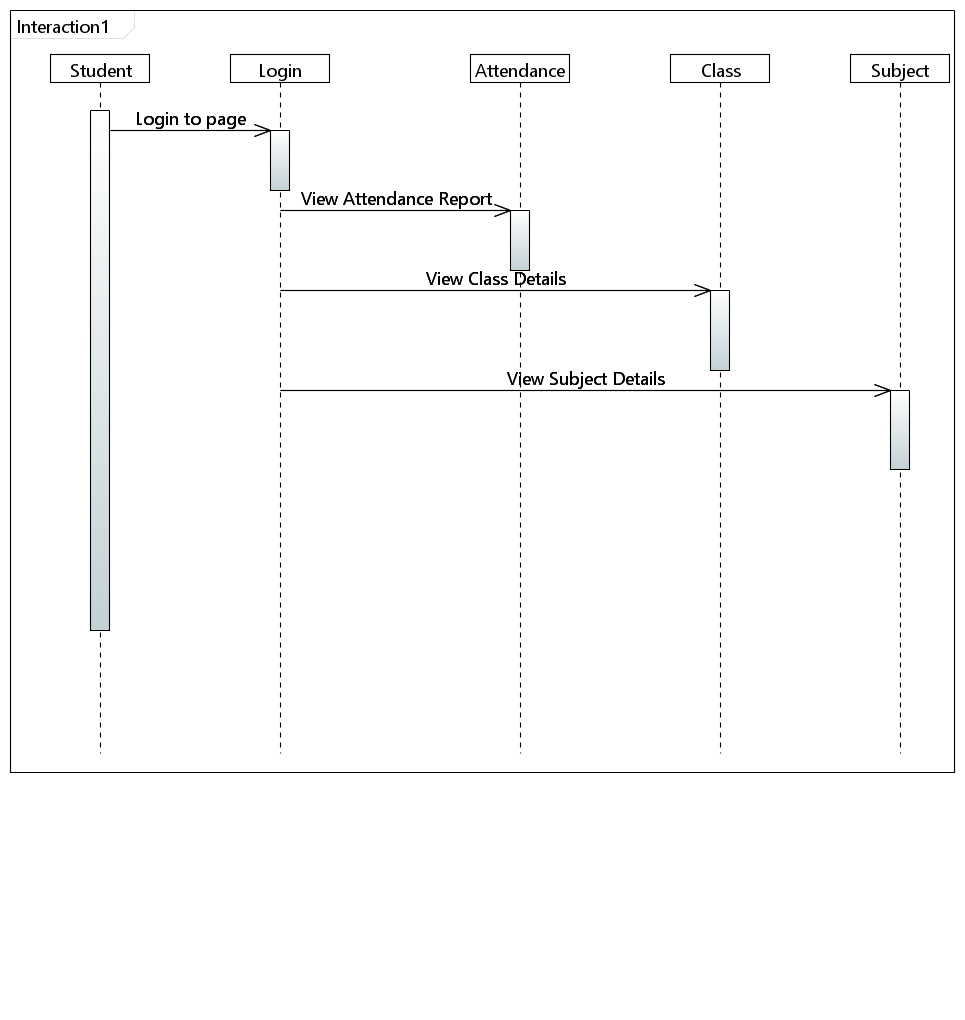
## 9. Application Design

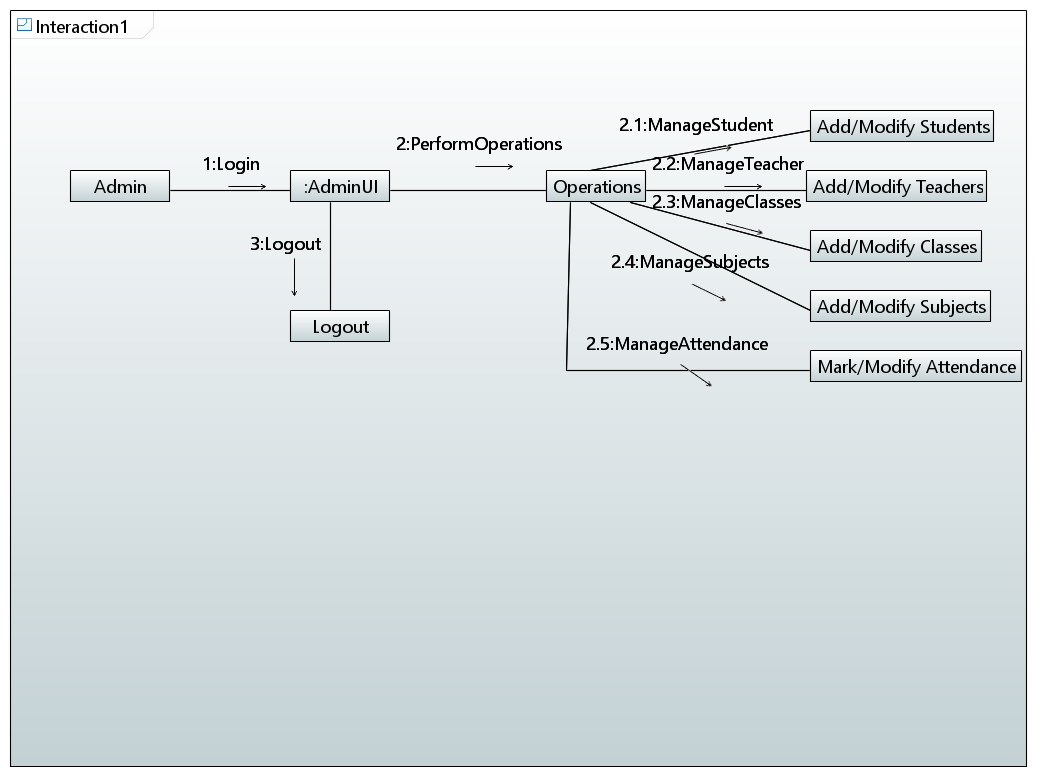
***9.1.2. Sequence and Collaboration Diagram***

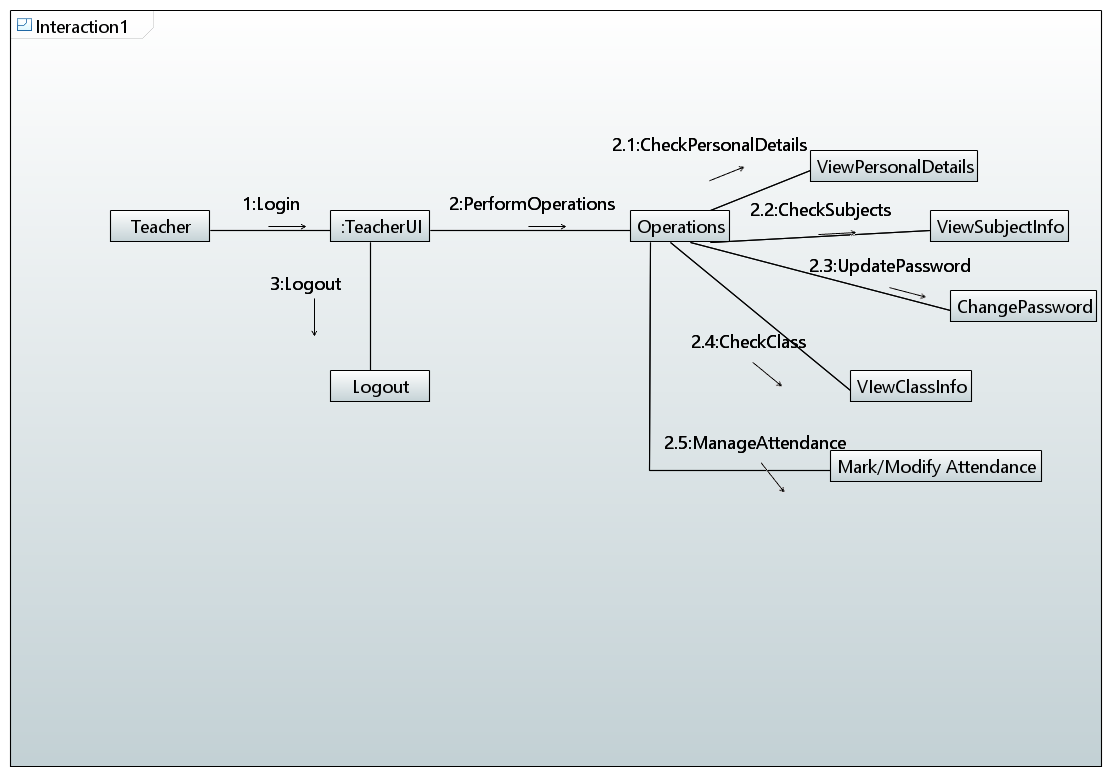
***9.1.2.1 Sequence Diagram for Admin:*** 

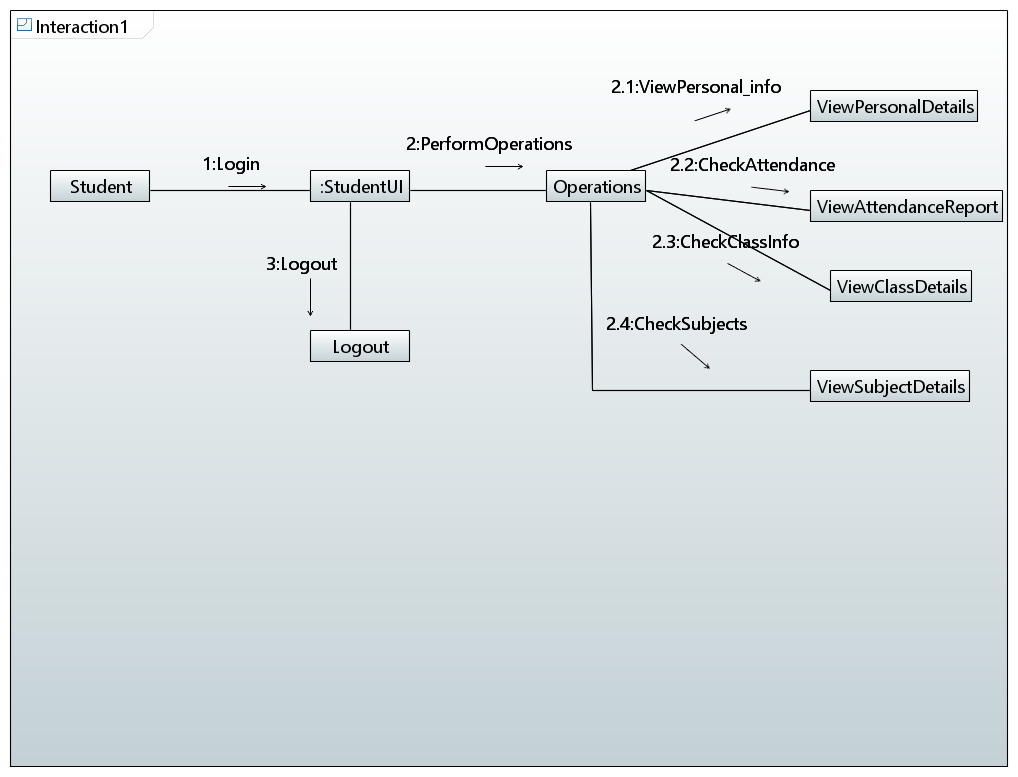
*In the above sequence diagram of School management system which shows the interaction between Admin and the objects of Teacher, Classes, Subject, Student. The instance of class objects involved in this UML sequence diagram of school management system are as follows: Admin Object, Teacher Object, Subject Object, Classes Object, Student Object.*

***9.1.2.2 Sequence Diagram for Teacher:*** 

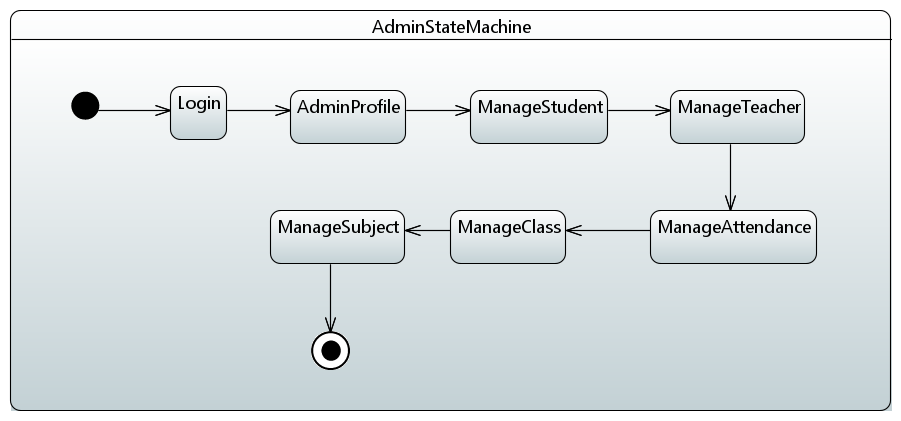
***9.1.2.3 Sequence Diagram for Student:*** 

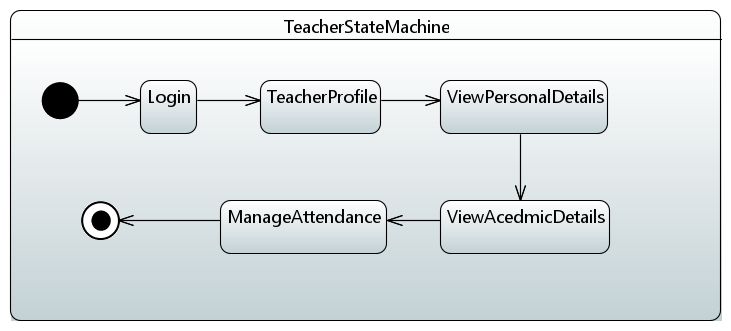
***9.1.2.4 Collaboration Diagram for Admin:*** 

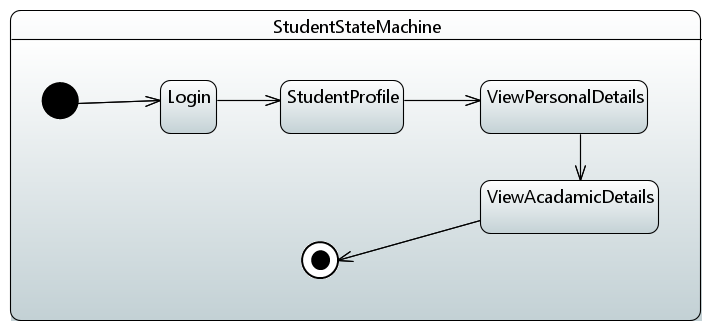
***9.1.2.4 Collaboration Diagram for Teacher:*** 

***9.1.2.4 Collaboration Diagram for Student:*** 

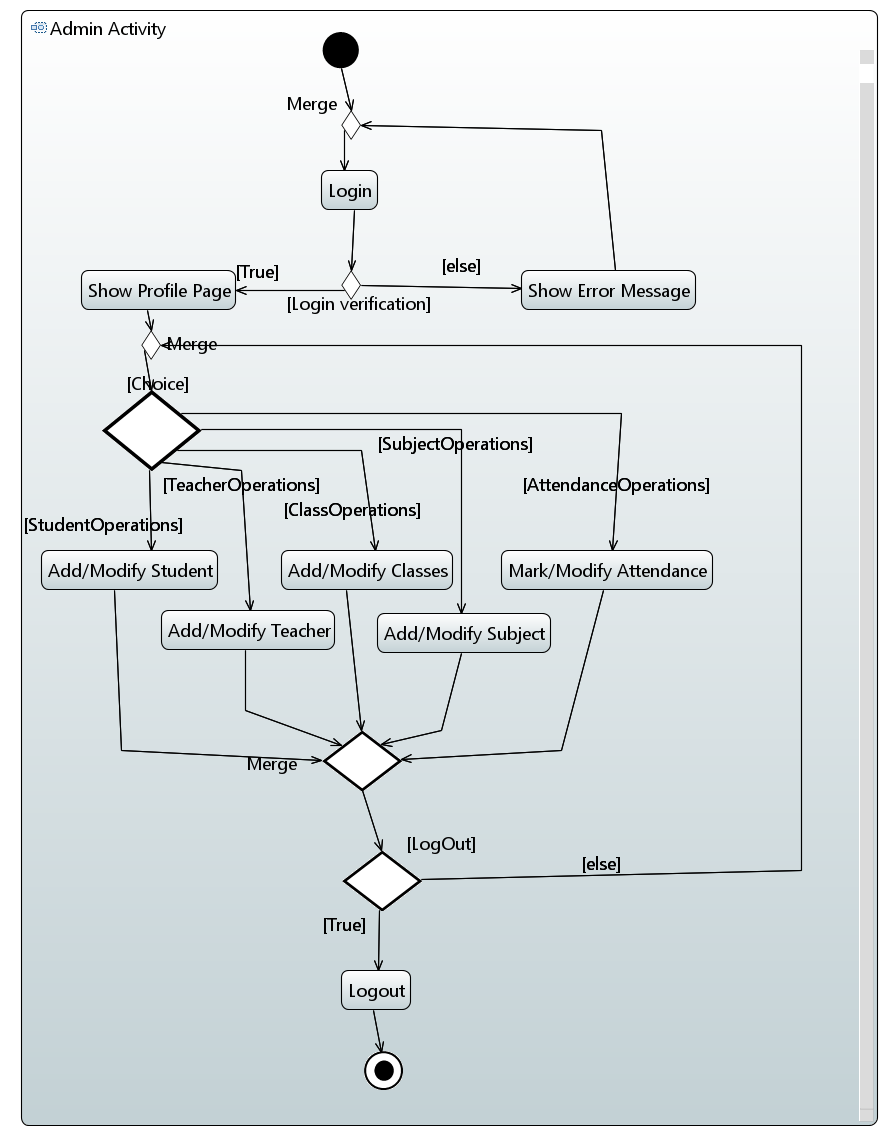
***9.1.3. State Diagram***

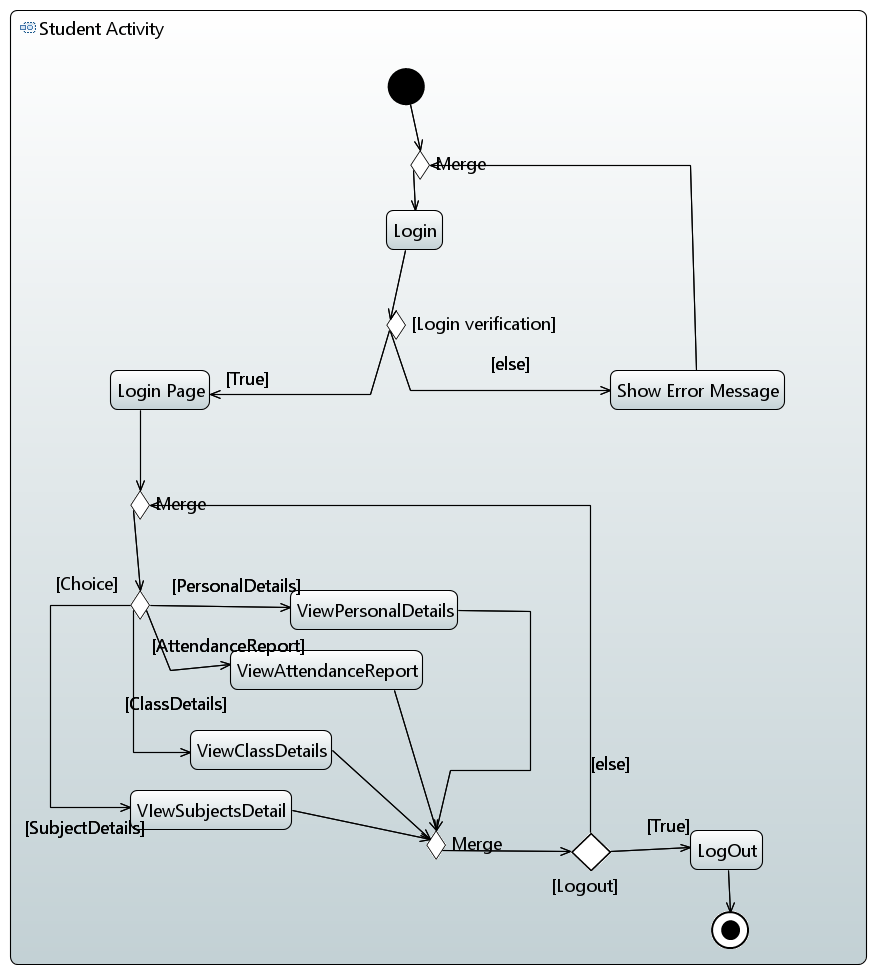
***9.1.3.1 State Diagram for Admin:*** 

9.1.3.2 State Diagram for Teacher: 

***9.1.3.3 State Diagram for Student:*** 

***9.1.4. Activity Diagram***

***9.1.4.1 Activity Diagram for Admin:*** 

***9.1.4.2 Activity Diagram for Student:*** 

## 9.1.4.3 Activity Diagram for Teacher:

## 10. References

*Not Applicable*

## 11. Appendices

*Not Applicable*