**COMSATS University Islamabad, Abbottabad Campus**

**Department of Computer Science**

**Project Proposal**

**School Management System**

**CSC392 Object Oriented Software Engineering**

Submitted on: <23-APR-2022>

Group Members:

Afaq (FA20-BSE-057)

Ali Said(FA20-BSE-4B-165)

Muhammad Nawaz Khan (FA20-BSE-073)

Abdullah Javed (FA20-BSE-052)

Ehsanullah (FA20-BSE-068-4B)

Faizan Zaheer (FA20-BSE-045)

Irfan Khan (FA20-BSE-070)

(Ahsan Ali SP20-BSE-060)

Table of Contents

[CHAPTER 1 PROJECT PROPOSAL 3](#_Toc101427096)

[Introduction 3](#_Toc101427097)

[Vision and Business Case 3](#_Toc101427098)

[Use-Case Model 3](#_Toc101427099)

[Supplementary Specification 3](#_Toc101427100)

[Glossary 3](#_Toc101427101)

[Risk List & Risk Management Plan 3](#_Toc101427102)

[CHAPTER 2 USE CASES 4](#_Toc101427103)

[Use Case Diagram 4](#_Toc101427104)

[Brief Level Use Cases 4](#_Toc101427105)

[Student Name 1 (Registration Number 1) 4](#_Toc101427106)

[Student Name 1 (Registration Number 1) 5](#_Toc101427107)

# CHAPTER 1 PROJECT PROPOSAL

## Introduction

School Management system, required for effective software development, to manage the overall data of a school

## Vision and Business Case

In the offline system, it is an overhead to keep the records related to faculty, student, parents, and other school staff on the papers. Everything related to their progress in the system is marked manually. For example A report of a student’s attendance is generated monthly is shown to his/her parents. Now, a regular student, going to school every day, is marked absent for a day by mistake.

It is a burden to take out the register and view the records. As you can see, it is a very time-consuming process and it costs much. So, I thought why I should not help these young guns of the nations to help them to have a bright future and to make an online centralized platform that can be accessed from anywhere in the world.

My other aim is to minimize the paperwork as minimum as I can so that there is no need to cut more and more trees. Indirectly, I will be helping Mother Nature.

## Use-Case Model

The functional requirements of school management system are:

1. Multi-User Account System
2. Student Fee Management
3. Parent Monitoring Feature
4. Homework Document
5. Class Routine Schedule
6. Profile System
7. Exam Marks Management
8. Chart & Graph Analysis of Exams
9. Daily Attendance
10. Internal Messaging

## Supplementary Specification

These are some Non Functional Requirements as we together in this inception phase. Security requirements are important factors in this system as classified data will be stored in the database. User validation will be done during login to insure that the user is valid and that the user only has access to his or her permission data. General users will only have access through the user interface. The system will have consistent interface formats and button sets for all forms in the application, will have a form based interface for all data entry and viewing formats. The system will be easily maintained by authorized trained person and it shall respond as fast as possible in generating report and producing the timetable.

## Glossary

SMS having a dynamic system with a bird view of data and reports can give next level of power and quickness in decision-making for principal or management person. School management System provides extra ordinary Management Dashboard and data reporting functions along with dynamic access rights mechanism which becomes a blessing for management personnel.

## Risk List & Risk Management Plan

**1. Malware**

Digital hackers are watching your every move and trick you to download malware and take control of your computer remotely. They use malware to attack computer networks to perpetrate crimes.  Fraudsters use virus, malware, spyware, spams, and phishing to gain access to your sensitive personal information and commit financial crimes. Defend your data against malware through secure servers, whether physical or in cloud, and shield against vulnerabilities.

**2. Theft & Loss**

Unauthorized users without permissions who have access to sensitive data can cause harm to educational institutions as a result of theft. There is a risk of the sensitive academic data will be leaked by staff. It becomes easy to lose your storage media with backup data due to misplacement or theft. When you suffer data loss due to various incidents such as mechanical damage, power failure, software crash, disasters or loss of your laptops and mobile devices, it is another way of inadvertent data exposure. Keep all your data safe and secure using role-based access control to ensure confidentiality and privacy.

**3. Unsafe data**

If adequate safety precautions are not taken when files and documents are shared in website, smartphones and tablets via internet networks, the information contained on them might gain access to the devices and get exposed to risks. We can make use of cloud deployments to manage the education system better and better.

**4. Negligence**

When data is stored in computers or laptops, it has become so natural that people lose the information when files are accidentally deleted or even it could fall into the wrong hands. Ensure a proper backup strategy to keep your data on important devices and run them smoothly without hassles.

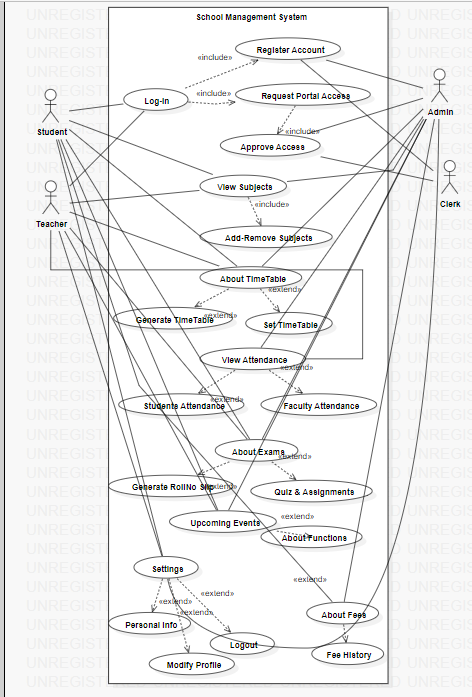
# CHAPTER 2 USE CASES

## Use Case Diagram



**Reverse Engineering:**

In reverse engineering I modify my use case diagram there is sum ambiguities in my use diagram. At Start of the Semester when we plane and draw Use case of our SMS. According to that Use case diagram the student and teacher have permission to register himself in the system. At later during implementation I discussed with my fellow and Teacher. They all give me Suggestion to add clerk in your system for Student and Teacher registration while clerk himself can register by Admin with these flow and concept I draw my use case diagram in reverse engineering.



## Brief level use case:

### Muhammad Nawaz Khan (FA20-BSE-073)

#### Use case: Add faculty attendance:

The faculty mark their attendance using biometric or by entering their id no the real time attendance saved in the database and updated and the system.

#### Use case: Add student attendance:

The teacher can mark the attendance of the student using the system on the specific date and the real time attendance can be updated in the system.

#### Use case: view student attendance:

The student can view their attendance on the specific date. teacher can view the real time attendance of their class students by using the system and the admin can view the real time class wise attendance of students through system.

#### Use case: Modify student attendance:

The teacher can modify the attendance of the students which is already uploaded to the system.

The teacher enter the date to view the attendance on the specific date and can modify the attendance and upload it the new attendance will be saved in the system.

### Ali Said(FA20-BSE-165)

#### Use Case: About Exams

In about exams use case faculty or staff can take exams directly from about exams section and take assignments directly from sub section assignments in about exams. It would be so ease for students as well as teachers to take/give exam or assignment in soft form . Teachers have to feed the questions on the about exams section after that teacher can set the timings for assignments or quizzes and set a deadline for submission. Exams marks would be marked automatically by the system but if teacher wants to change or over write he/she has access to do that. Students have access to upload file for assignments.

### Afaq (FA20-BSE-057)

#### Use Case: Time Table

In school management system time table is necessary so it is strong entity. From time table students can see their time table and takes classes according to their time table slot. Faculty can also take classes according to their time table. Guardians able to see their children time table and keep eye on their class’s times. School administration will set time and classes to each and every class from first grade to matric level. Every class has unique time table and subjects.

### Abdullah Javed (FA20-BSE-052)

#### Use Case: Subjects Management

In this use case students will be asked by the administrators to add class. After adding class there are certain subjects that are taught in that particular class added by student. Administrator is allowed to add or remove subjects from class, while students are bound to see only the subjects list. On the basis of subjects admin can assign instructor in each subjects. Faculty members can also list down the subjects they want to teach. Guardians are also eligible to see the subjects of student.

### **Ehsanullah (FA20-BSE-068)**

#### Use Case: View Result : Student Viewing Result

About Result use case will generate the result for student based on their request for view the result for the subjects. The results of student will be taken by teachers and the record will be submitted on system. The admin and teacher can modify and view the result of student. The guardian and student can only view the result.

#### Use Case: Add Result : Teacher Adding Result

The Add Result use case will create the result for teacher based on their request for Add the result for the subjects. The results of student will be taken by teachers and the record will be submitted on system. The admin and teacher can modify and view the result of student. The guardian and student can only view the result.

### **Faizan Zaheer (FA20-BSE-045)**

#### Use Case: Add Events

In Events Usecase Event Manager Will Add the Event,View events,delete events,monitor events status online,manage series and schedules of events ,monitor Venue and Parcipants records and show the list of parcipants and release the schedules of events and events detail.Event Manager make changes or modify the events and parcipants view the Events and schedule of events.

#### Use Case: Delete Events

In Events Usecase Event Manager Will delete events,monitor events status online, monitor Venue and Parcipants records and show the list of parcipants and release the schedules of events and events detail.Event Manager make changes or modify the events and parcipants view the Events and schedule of events.

#### Use Case: View Events

In Events Usecase Event Manager Will Add the Event,View events,delete events,monitor events status online,manage series and schedules of events ,monitor Venue and Parcipants records.

#### Use Case: Manage Events

In Events Usecase Event Manager Will manage series and schedules of events ,monitor Venue and Parcipants records and show the list of parcipants and release the schedules of events and events detail.Event Manager make changes or modify the events and parcipants view the Events and schedule of events.

### Irfan khan FA20-BSE-070

#### Use Case UC1: register account

User opens the School management system and clicks on the register button on his dashboard for student or teacher register in system. The system takes the user to the registration page for which he want to register. The user is required to enter information asked by the system. The user then enters his first name, last name, id, email, class and password. The system then checks if the user is already registered or not. If he is not registered the system validates and registers him as a student or teacher.

#### Use Case UC2: login

User opens the School management system and clicks on the login button. The system takes the user to the login page. The user is required to enter information asked by the system which is the username and the password. The system then checks if the user is already registered or not. If he is registered the system validates and logs him in.

#### Use Case UC3: Request portal Access

User opens the School management system and clicks on the register button. The system takes the user to the registration page. The user is required to enter information asked by the system. The user then enters his full name, email, address, phone number and password. The System gives to access to log in to System if the user is enter correct and valid information for registration of a user account. If he is register to the system. The System validates and registers him as a User

## Fully dressed use case:

### Muhammad Nawaz Khan (FA20-BSE-073)

#### Use Case UC1: Add faculty attendance

**scope**: mark faculty attendance

**primary Actors**: faculty, admin.

**Stake holder and interest:**

Faculty: it is easy for faculty to mark attendance using biometric and the attendance is updated directly

In the system and there is less chance of errors to forgot attendance.

Admin: managing the teacher’s attendance without this system is very difficult for admin.

The system generates attendance lists and inform the admin that which faculty member is absent.

**Preconditions:** faculty must be authorized through biometric or thorough id no.

**Success guarantee** (or Postconditions): marking the attendance for faculty become easy.

**Main success scenario or Basic flow:**

* Faculty marks their attendance through biometric.
* Teacher marks their attendance system and records is saved in the database.
* The system generates attendance report for admin.
* Admin can view the real time attendance.

**Extension or alternative flows:**

At any time, fingerprint can’t be registered:

* During marking attendance, the fingerprint may not work every time.
* If the fingerprint can’t work teacher should use their ID NO for attendance.

If system does not recover:

* The user will suspend the operation and the system will show an error message.
* The user starts a new operation and continue to his work.

**Special requirements:**

* There should be a biometric attendance machine for faculty attendance.
* The machine should be connected to system

**Technology and Data Variations List**:

The biometric attendance machine should be connected to the system.

#### Use Case UC2: Add Student attendance

**scope**: Mark student attendance

**primary Actors**: faculty.

**Stake holder and interest:**

Teacher: Teacher wants to take attendance easily without any error and don’t want to carry extra register to mark students’ attendance.

In the system and there is less chance of errors to forgot mark the attendance.

Student: students want to get rid of traditional list to check their attendance and the system is to correct information of student attendance

**Preconditions:**

* Teacher must be identified and authenticated.
* The teacher must be the form master of the class

**Success guarantee** (or Postconditions): The process of Taking attendance become easy and the real time attendance is updated in the system.

**Main success scenario or Basic flow:**

* Teacher wants to open school management system to mark the attendance of students.
* The system opens the mark attendance page for the teacher.
* The mark the student attendance of the on the form.
* The teacher saves the attendance and the record saved on the database.

**Extension or alternative flows:**

At any time, system fails:

* The system can fail any time
* only those data will discard which is not saved.
* User will restart the system from the prior state.
* The data will be inserted again

If system Data already exist in the system:

* If The user wants to enter to enter the duplicate data, the system will show an error message.
* Record already exits.

**Special requirements:**

* The user must be authorized and authenticated.

**Technology and Data Variations List**:

The computer should have window install on it.

#### Use Case UC2: View Student Attendance

**scope**: view student attendance

**primary Actors**: Teacher, Admin, Student.

**Stake holder and interest:**

* The students can view can their real time attendance.
* The system generates class wise attendance lists for the admin.
* The teacher can view the real time attendance of the students.

**Preconditions:**

Students, Teachers and admin must be identified and authenticated.

**Success guarantee** (or Postconditions):

* The System View real time attendance to student and teacher.
* Generate Accurate class wise attendance Report for admin.

**Main success scenario or Basic flow:**

* The teacher wants to open the system to view the student’s attendance
* The system will redirect him to teacher view attendance.
* The system loads the attendance with dates and progress bar.
* The want views the class wise attendance
* The system will redirect him to Admin view attendance page
* The system will load the class wise attendance with percentage and progress bar.
* Student wants to view his own attendance
* The system will redirect him to student view attendance page
* The system will load the student attendance with date and progress bar.

**Extension or alternative flows:**

At any time, system fails:

* The system can fail any time
* User will restart the system from the prior state.
* The system will work normally after restart.

If data does not exist in the system:

* If the record does not exist in the system, the system will show an error message
* Record not found

**Special requirements:**

* The user should be authorized and authenticated.

**Technology and Data Variations List**:

#### Use Case UC4: Modify Student Attendance

**scope**: update the existing attendance

**primary Actors**: Teacher.

**Stake holder and interest:**

Teacher: Teacher can update the attendance at any time

In case if there any error or teacher want to modify for some other reason teacher enter and modify attendance at any time

In the system and there is less chance of errors to forgot mark the attendance.

Student: students want to get rid of traditional list to check their attendance and the system is to correct information of student attendance

Preconditions**:**

Students, Teachers, and admin must be identified and authenticated.

**Success guarantee** (or Postconditions):

* The System View real time attendance to student and teacher.
* Generate Accurate class wise attendance Report for admin.

**Main success scenario or Basic flow:**

* The teacher wants to open the system to view the student’s attendance
* The system will redirect him to teacher view attendance.
* The system loads the attendance with dates and progress bar.
* The want views the class wise attendance
* The system will redirect him to Admin view attendance page
* The system will load the class wise attendance with percentage and progress bar.
* Student wants to view his own attendance
* The system will redirect him to student view attendance page
* The system will load the student attendance with date and progress bar.

**Extension or alternative flows:**

At any time, system fails:

* The system can fail any time
* User will restart the system from the prior state.
* The system will work normally after restart.

If Record does not exist in the system:

* If the record does not exist in the system, the system will show an error message
* Record not found

**Special requirements:**

* The user should be authorized and authenticated.

### Ali Said (FA20-BSE-165)

#### Use Case UC1: About Exams

**Scope**: School Management System

**Level:** user goal

**Primary Actor:** Student, Guardian, Faculty, Admin.

**Stakeholders and Interests:**

- **Admin:** Admin is the one who controls the whole school management system, its staff, and every other person or thing that is associated with the school. In other term, the admin is the most powerful entity of the system.

-**Teacher:** The teacher is one of the important entities for a school. The teachers are there to teach the students. The following are the features that will be available to the teachers.

-**Student:** The student is also one of the main parts of the system. The students can log in to the student account anytime they want.

**-Parents/Guardians:** It is important for the parents to be familiar with the status of their child how they are performing in the academy. To make this hassle-free, there is also an option available for the parents.

**Preconditions**: Students are identified and authenticated.

**Success Guarantee** (or Postconditions): Teachers will feed questions and answers and then they will take quiz. Students have to give timeboxed quiz. After that system will automatically mark the quiz on the basis of feed questions.

**Main Success Scenario (or Basic Flow):**

1. Teacher will store the questions on the about exams section before taking quiz/assignment.
2. Teacher will set the deadline/duration of assignment/quiz.
3. Students will be notified about quiz or assignments in notification section.
4. Students will start the quiz or students will download the assignment file.
5. After answering few questions their quiz will end up or they can upload the assignment before deadline.
6. After deadline there will be no option available for quiz/assignment.
7. System will automatically mark the quizzes but in case of assignments teacher have to practically do it.
8. After teacher`s marking result will be sent to the result section and will be publicly available to students.

**Extensions (or Alternative Flows):**

At any time, admin requests can override operation:

1. The teacher set the exam.
2. Admin wants any other operation at that time frame.
3. The admin operation will override the teacher operation.
4. Teacher operation will be set to next timeframe available.

**Special Requirements:**

- Touch screen UI on a large flat panel monitor. Text must be visible from 1 meter.

- Page response within 3 seconds 90% of the time.

- Language internationalization on the text displayed.

**Technology and Data Variations List**:

1. Teachers can set up fingerprint or face unlock.

### Afaq (FA20-BSE-057)

#### **Use Case UC1**: View Timetable

**Scope:** Time Table

**Primary Actor:** Student, Guardians, Faculty, Administration

**Stakeholders and Interests:**

- Student: student can visit his/her time table on daily basis and attend their classes according to their time table

- Teachers: Teachers will also take classes according to their time table.

-Guardians: Guardians can also watch their children time table

- Administration: School administration can modify time table on daily basic according to class rooms available. Each class from grade first to matric has unique time table and subjects.

**Preconditions:** Students, faculty, guardians and administration must be identified and authenticated.

**Success Guarantee** (or Postconditions): Online time table system will help school to be more productive because students and faculty can watch their time table online so there is no chance to miss classes.

**Main Success Scenario (or Basic Flow):**

1. Students will reach to their classes on time because of online table.
2. Faculty can take online attendance according to current time table
3. Administration of school will allot time table to students and faculty before one day so it will save time.
4. Guardians can also watch time table so they keep eye on child activities.
5. Principle of school can also watch time table.

**Extensions (or Alternative Flows):**

\*a. At any time, internet service off.

* During the marking or updating attendance any time the internet connection may be gone.
* The system will be interrupted, and the teacher will not be able to use the system.
* After the fixing the internet, students and faculty will login and continue back to his work.

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Students and faculty restarts System, logs in, and requests recovery of prior state.

**Special Requirements:**

- Touch screen UI on a large flat panel monitor. Text must be visible from 1 meter.

- Credit authorization response within 5% seconds 90% of the time.

### Abdullah Javed (FA20-BSE-052)

#### **Use Case UC1**: Subject Management

Scope: School Management System

Level: user goal

Primary Actor: Student, Admin, faculty, Guardian.

Stakeholders and Interests:

- Admin: Wants only eligible students to enroll for classes , want accurate records of which students are enrolled on which courses, want to know how many students are registered to each course. Admin is also responsible to assign subjects to each student of particular class.

- Student: Wants to enroll for a class and for no eligibility problems to later arise, wants proof of enrollment , wants process to be clear and simple.

- Guardian: Wants to know for his children details about subjects. Guardian is eligible only for viewing details.

- Faculty: teacher is one of the important entities for a school. The teachers are there to teach the students. The following are the features that will be available to the teachers.

**Preconditions:** Student is identified and authenticated.

**Success Guarantee**

Student is aware of enrollment on choosen class. List of students enrolled to course is updated. Student has a proof of enrollment.

**Main Success Scenario (or Basic Flow):**

1. The user requests a list of class currently available for enrollment.
2. The system retrieves the list of courses available and displays the list to the user.
3. System provide the user with the opportunity to select a course or to exit the use case.
4. The user selects a course for which he or she wants to enroll.
5. The system retrieves the details for the chosen course, including the course outline, timetable and eligibility requirements and displays the details to the user.
6. The system checks that the user has taken and passed the pre-requisite courses forthe chosen course
7. The system checks that the user is not already enrolled on a course who setimetable clashes with the chosen course

**Extensions (or Alternative Flows):**

At any time, admin requests can override operation:

1. The teacher view its own subject list.
2. Admin wants any other operation at that time frame.
3. The admin operation will override the teacher operation.
4. Teacher operation will be set to next timeframe available.
5. Admin can change the instructor of the subject.

**Special Requirements:**

User must use computer

User must install that software for access- …

Touch screen UI on a large flat panel monitor. Text must be visible from 1 meter.

**Open Issues:**

The must have computer connected to internet.

**Screen Shot Of GUI:-**

****

### Ehsanullah (FA20-BSE-068-4B)

#### Use Case UC1: View Result : Student Viewing Result

**Scope:** School Management System

**Level:** user goal

**Primary Actor:** Student, Guardian, Faculty.

**Stakeholders and Interests:**

- **Faculty:** Wants accurate, fast entry, and no text errors.

- **Guardian:** Wants to view his/her, son/ daughter result.

- **Student:** Wants easy and fast service with minimal effort. Wants easily visible display of each entered subject result.

- **School Management**: Wants to accurately view result records and satisfy Student, Faculty, interests.

- **Admin:** Wants to be able to quickly perform override operations, and easily debug About Result problems.

**Preconditions:** Student is identified and authenticated.

**Success Guarantee** (or Post conditions): Result is saved. Result is correctly displayed. Result list are updated. Result is generated. Student authorization approvals are recorded.

**Main Success Scenario (or Basic Flow):**

* Faculty marks their result.
* Teacher marks result of students using the system and records is saved in the database.
* The system generates result report for admin.
* System alert guardian if the student is failed.
* Students and guardians can view the real time result.

**Extensions (or Alternative Flows):**

At any time, the internet can be gone:

* During the marking or updating result any time the internet connection may be gone.
* The system will be interrupted, and the teacher will not be able to use the system.
* After the fixing the internet, the user will login and continue back to his work.

At any time, system fails:

* The system can fail any time. The system it will save the work.
* User will restart the system and request for recovery the system will start from the prior state.

If system does not recover:

* The user will suspend the operation and the system will show an error message.
* The user starts a new operation and continue to his work.

**Special requirements:**

* The user should have the computer to use the system.
* The internet must be connected.
* The user should be authorized and authenticated.
* - Touch screen UI on a large flat panel monitor. Text must be visible from 1 meter.
* - Page response within 3 seconds 90% of the time.
* - Language internationalization on the text displayed.

**Technology and Data Variations List**:

The primary actors must have computer connected to internet.

#### Use Case UC1: Add Result : Teacher Adding Result

**Scope:** School Management System

**Level:** user goal

**Primary Actor:** Student, Guardian, Faculty.

**Stakeholders and Interests:**

- **Faculty:** Wants accurate, fast entry, and no text errors.

- **Guardian:** Wants to view his/her, son/ daughter result.

- **Student:** Wants easy and fast service with minimal effort. Wants easily visible display of each entered subject result.

- **School Management**: Wants to accurately view result records and satisfy Student, Faculty, interests.

- **Admin:** Wants to be able to quickly perform override operations, and easily debug About Result problems.

**Preconditions:** Student is identified and authenticated.

**Success Guarantee** (or Post conditions): Result is saved. Result is correctly displayed. Result list are updated. Result is generated. Student authorization approvals are recorded.

**Main Success Scenario (or Basic Flow):**

* Faculty marks their result.
* Teacher marks result of students using the system and records is saved in the database.
* The system generates result report for admin.
* System alert guardian if the student is failed.
* Students and guardians can view the real time result.

**Extensions (or Alternative Flows):**

At any time, the internet can be gone:

* During the marking or updating result any time the internet connection may be gone.
* The system will be interrupted, and the teacher will not be able to use the system.
* After the fixing the internet, the user will login and continue back to his work.

At any time, system fails:

* The system can fail any time. The system it will save the work.
* User will restart the system and request for recovery the system will start from the prior state.

If system does not recover:

* The user will suspend the operation and the system will show an error message.
* The user starts a new operation and continue to his work.

**Special requirements:**

* The user should have the computer to use the system.
* The internet must be connected.
* The user should be authorized and authenticated.
* - Touch screen UI on a large flat panel monitor. Text must be visible from 1 meter.
* - Page response within 3 seconds 90% of the time.
* - Language internationalization on the text displayed.

**Technology and Data Variations List**:

The primary actors must have computer connected to internet.

### Faizan Zaheer (FA20-BSE-045)

| Use Case UC1: Add Events *Use Case UC2: Delete Event.*  *Use Case UC3:* *View Event.*  *Use Case UC4:* *Manage Event.* |
| --- |
| **Scope**: School Management System  **Level**: user goal  **Primary** **Actor**: Event Manager,Parcipants.  **Stakeholders and Interests**:  - Event Manager: Wants to add events and Manages the all the events in school.  -Event manager: are responsible for planning events and ensuring that they run as smoothly as possible.  -Produce detailed proposals for events (for example, time lines,venues, suppliers, legal obligations, staffing and budgets.  - Parcipants: Wants to register and parcipate in events .  - School Management: Wants to give facilities to parcipants for achieving their goals and easly parcipate in events.  **Preconditions**: Event Manager and Parcipants is identified and authenticated. |

**Success Guarantee** (or Postconditions): Information about Events is saved. Events is correctly managed. Schedule and Venue are updated.Clear Objectives, Location of Venue, Financial Resources, Code of Conduct, Marketing and Promotion and lastly, Sponsorship of Event.

**Main Success Scenario (or Basic Flow):**

1. Event manager Add the Events.
2. System give form for Events Details.
3. Event Manager give details of Events.
4. System add the Events.
5. Students will register in events.
6. Students take parcipate in events.
7. Manager Delete the Event.
8. Manager view Event status online.
9. Manager monitor the events status online and update the changes.
10. Parcipants monitor venue and parcipants records.
11. System will Automaticaly send the notifications about events to all Students.

**Extensions (or Alternative Flows):**

**a.At any time, the internet can be gone:**

1. The Event manager add the event.
2. Manager wants any other operation at that time frame.
3. The manager operation will override the parcipant operation.
4. parcipants operation will be set to next timeframe available.

**b. At any time, System fails:**

1. The system can fail any time. The system it will save the work.
2. User will restart the system and request for recovery the system will start from the prior state.

If system does not recover:

* The user will suspend the operation and the system will show an error message.
* The user starts a new operation and continue to his work.

**Special Requirements:**

* The user should have the computer to use the system.
* The internet must be connected.
* The user should be authorized and authenticated.
* Touch screen UI on a large flat panel monitor. Text must be visible from 1 meter.
* Page response within 30 seconds 90% of the time.
* Language internationalization on the text displayed.

**Technology and Data Variations List**:

1. Admin can set up fingerprint or face unlock.
2. The primary actors must have computer connected to internet.

(AHSAN ALI SP20-BSE-060)

Brief Level Use Case

**About Fees:**

This use case will use the student for fee history. A student arrives at a checkout the details of fee. It calculate the all fee challan of the students in institution. Wants to be able to quickly perform override operations, and easily debug students problems.

Fully dressed use case

**Scope:** about fees

**Level:**

**Primary Actors:** Guardians, students, admin

**Stakeholders and Interests**:

Guardian; Guardian wants to know all the fee details without any problem.

Admin: Admin calculate all the fee challan of student in the institution without any error. It can be say that managing the fee problems of students and generate the fee challans. Wants to be able to quickly perform override operations, and easily debug students problems.

Students: Student want to get rid of traditional list to check their details of his fee and this system is very useful for the students for view their updated fee history.

**Main Success Scenario (or Basic Flow):**

1. Guardian wants to know all the fee details without any problem.
2. Admins tells students the total, and asks for payment.
3. System presents receipt
4. Student want to get rid of traditional list to check their details of his fee
5. this system is very useful for the students for view their updated fee history.

**Extensions (or Alternative Flows):**

At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

**Special Requirements:**

* **Fee should be submitted through bank.**
* **Fee section visible to guardian.**

**Technology and Data Variations List**:

1. Languages used is java.
2. Software used to design interface is netbeans, Gui Swing.
3. Mouse and keyboard are required.

Frequency of Occurrence: Could be nearly continuous.

### Irfan khan FA20-BSE-070

#### Use Case UC1: register account

**Scope:** School management system

**Level:** user goal

**Primary Actor:** Student, Admin, Teacher,cLERK

**Stakeholders and Interests:**

* Student: want to register an account in the School management system.
* Teacher: want to register an account in the School management system.
* Admin Register Student and Teacher in the School Management System.

**Preconditions:**

* The System is running correctly
* User is not registered before login.

**Success Guarantee** (or Post conditions):

* Username is not available and account is registered.
* User is registered and Account is created Successfully

**Main Success Scenario (or Basic Flow):**

1. Admin wants to open the School management system to register an account.
2. System redirects him to the registration page
3. System asks him to provide the details.
4. The user then enters his first name, last name, id, email, class and password.
5. System then checks if the account is already registered or not.
6. The account is registered the system asks to provide new information
7. The account is not registered then the system registers the account.
8. System then takes the user to his dashboard where he can access his own profile.

**Extensions (or Alternative Flows):**

\*a. At any time when the user tries signing up:

1. User enter all required information for registration in School management System.
2. User enters the First name, last name, username email address and password is used when registering his account.
3. The system then verifies his credentials when he clicks on the register button.
4. When the credentials matches then he is redirected to his profile.

\*b. At any time when the user tries signing up:

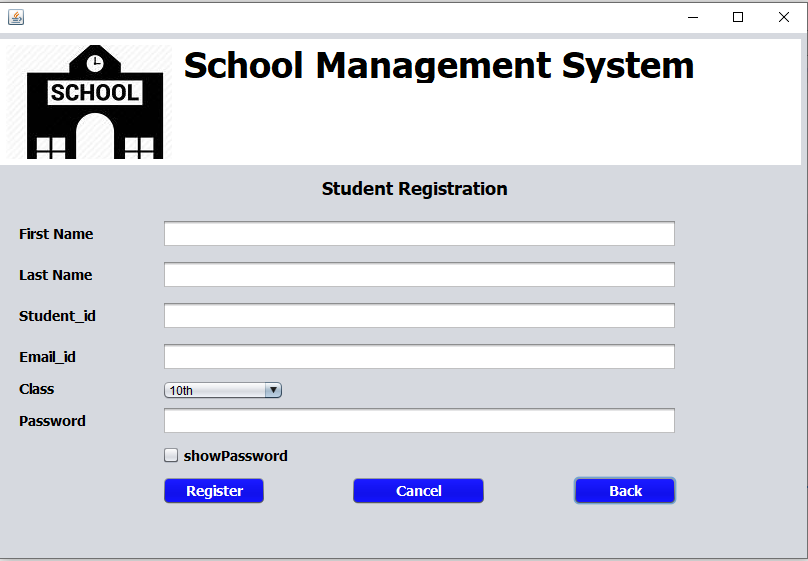
1. User enters the password and username he used when registering his account.
2. The system then verifies his credentials.
3. If the credentials are wrong, then the system displays an error message.
4. If the user not entered credential while registration, then the System displays an error that “enter valid information.”
5. If the credentials don’t match, then the system displays an error message.
6. If the user entered credential while registration and he enter his unique identifier for registration as duplicate entry ,then the System displays an error that “The user is already register. ”

**Special Requirements:**

* + - Text should be visible from 2 meters.
    - Color scheme should be used which is clearly visible.
    - Special characters should be used in username and password.
    - It should take less than 30 second to register an account.
    - Password should be display as \*\*\*\*.

**Technology and Data Variations List**:

1. Languages used is java.
2. Software used to design interface is net beans, Gui Swing.
3. Mouse and keyboard are required.



#### Use Case UC2: login

**Scope:** School management system

**Level:** user goal

**Primary Actor:** Student, admin, Teacher

**Stakeholders and Interests:**

* Student: want to login in the School management system.
* Faculty: want to login in the School management system.

**Preconditions:**

* User is identified and authenticated.
* User is registered and account is exist in System.
* User credentials must entered correctly.

**Success Guarantee** (or Post conditions):

* Username and password are valid, and account is registered.
* User must be able to login.
* User is successfully login to the system.
* System shall redirect to User profile

**Main Success Scenario (or Basic Flow):**

1. User opens the School management system to log in into his account.
2. System redirects him to the log in page
3. System asks him to provide the details.
4. The user enters username, and password
5. System then checks if the account is already registered or not.
6. If the account is registered the system verifies the account.
7. If the account is not registered, then the system asks to register the account.
8. System then takes the user to his dashboard where he can access the System and his own profile.

**Extensions (or Alternative Flows):**

\*a. At any time when the user tries to login:

1. User enters the username and password he used when log in to his account.
2. The system then verifies his credentials when he clicks on the log in button.
3. When the credentials matches then he is redirected to his profile.

\*b. At any time when the user tries to login:

1. User enters the password and username he used when log in his account.
2. The system then verifies his credentials.
3. If the credentials don’t match, then the system displays an error either his password or username is incorrect.
4. If the user not put information in his login id and password, the system should diplay an error message “Enter valid details”.

**Special Requirements:**

* + - Text should be visible from 3 meters.
    - Color scheme should be used which is clearly visible.
    - Special characters should be used in username and password.
    - It should take less than 30 second minutes to register an account.
    - Password should be display as \*\*\*\*.

**Technology and Data Variations List**:

1. Languages used is java.
2. Software used to design interface is netbeans, Gui Swing.
3. Mouse and keyboard are required.

**Screen Shots:**

****

#### Use Case UC3: Request portal Access

**Scope:** School management system

**Level:** user goal

**Primary Actor:** Student, teacher, admin.

**Stakeholders and Interests:**

* Student want to register an account in the School management system.
* The system must approve his registration of an account to access School System.
* Faculty: want to register an account in the School management system.
* The system must approve his registration of an account to access School System.
* Admin must Enter and register the student in the System to give him Access of login to System.

**Preconditions:**

* User is not registered before and User have no Access to the System.
* User must enter correct and valid credential to access the System.

**Success Guarantee** (or Post conditions):

* The system gives Access to User for log in to the system.
* User is redirect to the login page after Successful access and registration of an Account.
* Account is not already registered with the same User name before.

**Main Success Scenario (or Basic Flow):**

* + User wants to open the School management system to Access an account to register to the System.
  + The System redirects him to the registration page to provide accessibility.
  + System asks him to provide the details for Access the System.
  + The user enters username, password, email and address for the creation of an account in System.
  + System then checks if the account is already registered or not.
  + If the account is registered the system asks to provide new information or to login again.
  + If the account is not registered then the system registers the account and provide access to the user.
  + System then takes the user to his dashboard where he can access his own profile.

**Extensions (or Alternative Flows):**

\*a. At any time when the user tries to access for the System:

* User enter all required information for registration to get Account access in School management System.
* User enters the First name, last name, username email address and password is used when registering his account for the Access of School system.
* The system verifies his credentials for accessibility to System when he clicks on the register button.
* When the credentials matches then he is redirected to his login to the System.
* User enters the password and username to use for login when he get access to an account.
* The system then verifies his credentials.
* If the credentials don’t match, then the system displays an error either his password or username is incorrect.
* If the user not entered credential while registration to access the system, then the System displays an error that “enter valid information.”
* While registration and login user must enter correct and valid information.

**Special Requirements:**

* Text should be visible from 2 meters.
* Color scheme should be used which is clearly visible.
* Special characters should be used in username and password.
* It should take less than 30 second to access the system to register an account.

**Technology and Data Variations List**:

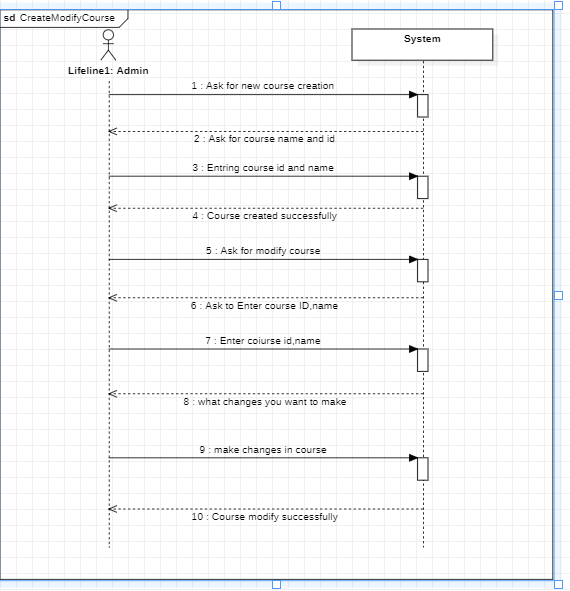
* Languages used is java.
* Software used to design interface is netbeans, Gui Swing.
* Mouse and keyboard are required.

****

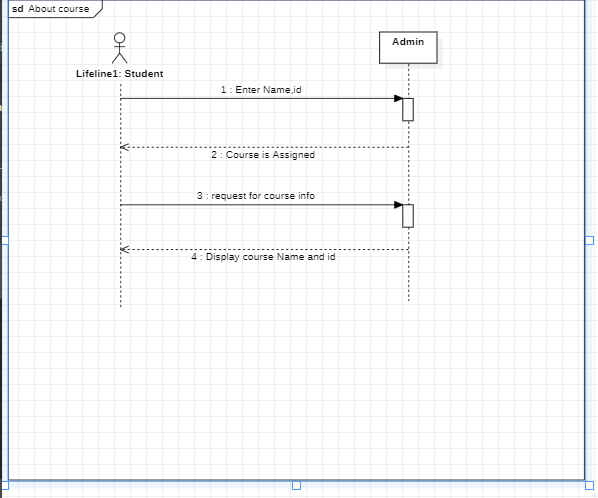
# CHAPTER 3 SSD

## Abdullah javed (FA20-BSE-052):

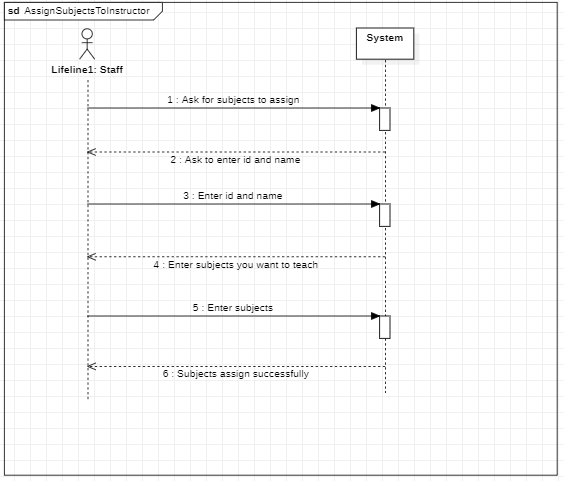
Use Case UC1: create,Modify course



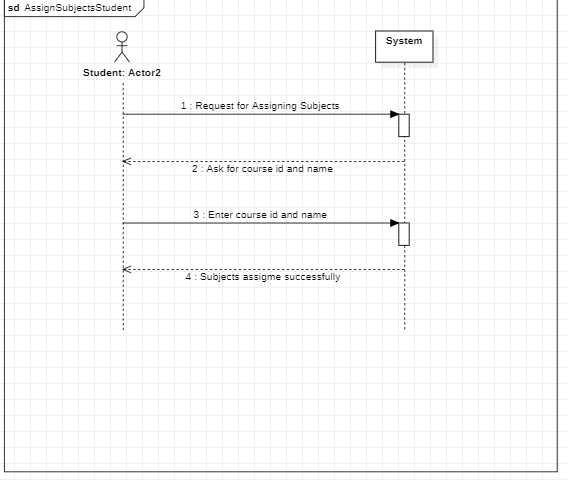
Use Case UC2:About course



Use Case UC3: Assign subjects to instructor:



Use Case UC4:Assign subjects to stuents:



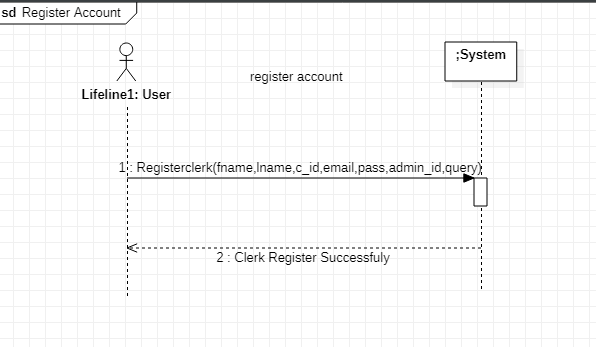
## Irfan khan FA20-BSE-070

#### Use Case UC1: register account

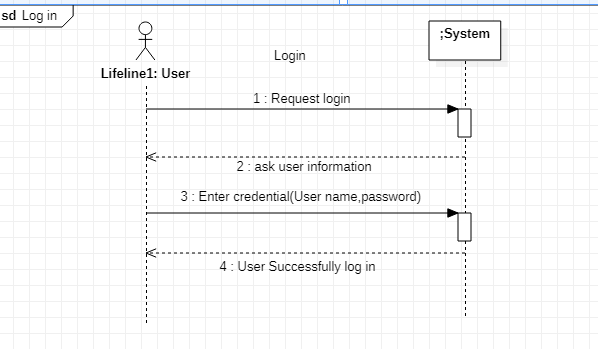


**Reverse engineering:**

**In reverse** I compare my old SSD with code then it was totally wrong so my use case is Register Account then according to my use case diagram Admin can register clerk and clerk register Student and teacher So here Admin Register Clerk So our System event is **Registerclerk** as I show in update SSD diagram.

****

#### Use Case UC2: Login



#### Use Case UC3: Request portal Access



## Faizan Zaheer (FA20-BSE-045)

#### Use Case UC1: Add Events:



## Ahsan Ali SP20-BSE-060

#### Use Case UC1:

**System Sequence Diagram**



## Ehsanullah FA20-BSE-068

#### VIEW RESULT:

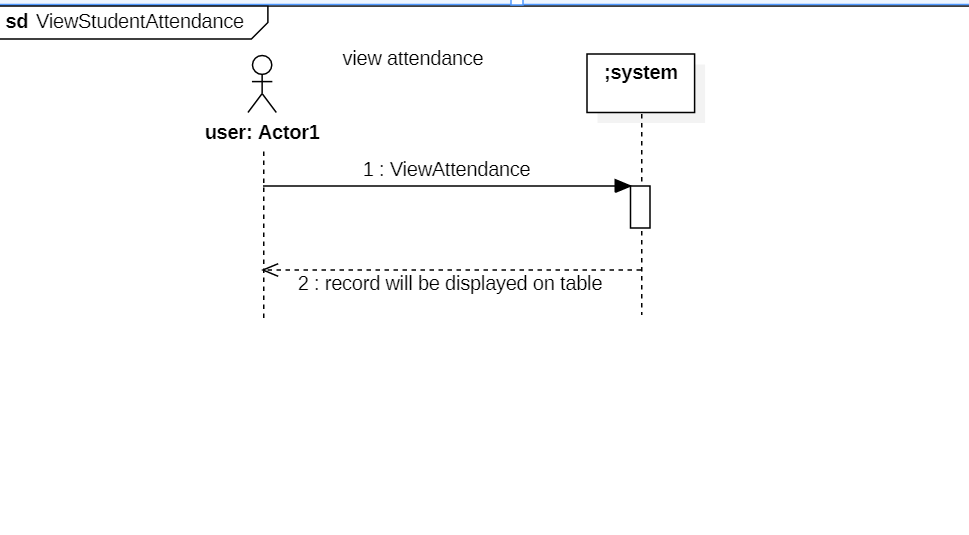


#### ADD RESULT:

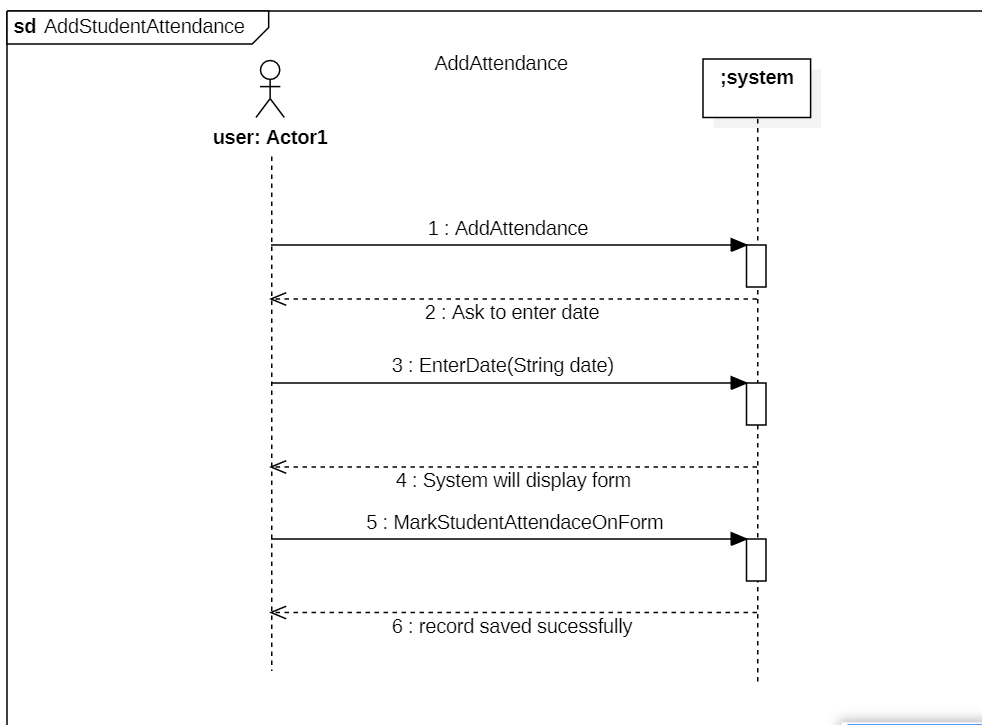


## Muhammad Nawaz Khan FA20-BSE-073

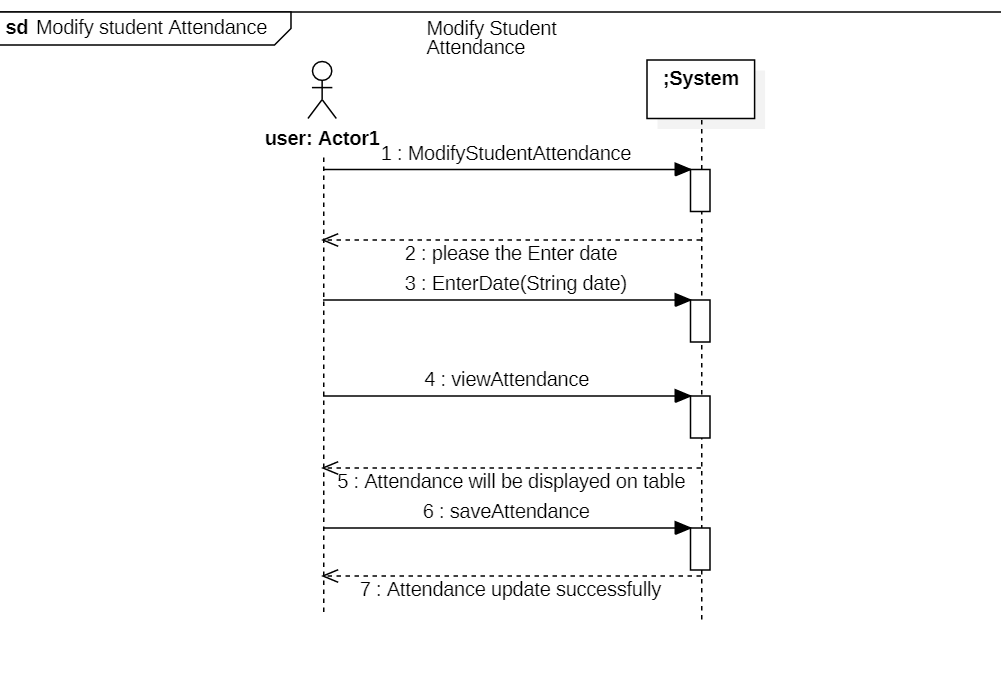
#### View Student Attendance:



#### Add Student Attendance:



#### Modify Student Attendance:



# CHAPTER 4 Operation Contracts – Sections

### Abdullah (FA20-BSE-052)

| Use Case UC1: Assign Subjects |
| --- |

**Contract ID**: Admin assign subjects to students

**Operation:** Admin will assign subjects against course with (subject Id, subject name: String).

**Cross References:** Use cases: Assign Subjects

**Pre-conditions:**

* Course must be registered
* Teachers must be registered against subject.

**Post-conditions:**

* All subjects must be assigned to teachers.
* Students are not able to add or remove subjects.
* Subjects are assigned by admin.

| Use Case UC2: create course |
| --- |

**Contract ID**: Admin create new course in the system.

**Operation:** Admin will create new course after checking its duplication (Course Id, Course name: String).

**Cross References:** Use cases: create\_course

**Pre-conditions:**

* Course must have unique id and name.
* Course must have some subjects

**Post-conditions:**

* Student enroll only one course at a time.
* Each course/class have advisor.

## Irfan khan FA20-BSE-070

#### Use Case UC1: register account

**Contract ID**: Request Registration

**Operation:** Request Registration ()

**Cross References:** Use cases: Register account

**Pre-conditions:**

* User is not registered.
* The System is running correctly.­­

**Post-conditions:­­**

* User is registered and Account is created successfully.

**Contract ID**: Enter Credential

**Operation:** Enter information ()

**Cross References:** Use cases: Register account

**Pre-conditions:**

* User is not registered.
* The System is running correctly.
* All enter information should be in proper valid form.

**Post-conditions:**

* User is registered and Account is created successfully.

**Reverse engineering:**

**In reverse** I compare my old operation contracts with code and my SSD diagram then it was wrong so my use case is Register Account then according to my use case diagram Admin can register clerk and clerk register Student and teacher So here Admin Register Clerk So our System event is **Registerclerk** as I show in update operation contracts .

**Contract ID**: Registerclerk

**Operation:** Registerclerk(fname:String,lname:String,c\_id:String,email:String,pass:String,admin\_id:String,query:String)

**Cross References:** Use cases: Register account

**Pre-conditions:**

* The System is running correctly.
* Clerk is not registered before.
* All enter information should be in proper valid form.
* All text field must be filled.
* c\_id(clerk\_id is user name) must be unique. Otherwise it shows an error message.

**Post-conditions:**

* Clerk was Register Successfully.
* Registerclerk instance was created (instance creation).
* Registerclerk was associated with Student\_registeration.
* Registerclerk was associated with teacher\_registeration

#### Use Case UC2: Login

**Contract ID**: User Login

**Operation:** Request Login ()

**Cross References:** Use cases: Log in

**Pre-conditions:**

* User is registered and account is exist in System.
* User credentials must entered correctly.

**Post-conditions:**

* User is successfully login to the system.
* System shall redirect to User profile

**Contract ID**: Enter information

**Operation:** Enter user log in details (User-name, and password: String)

**Cross References:** Use cases: Log in

**Pre-conditions:**

* User is registered and account is exist in System.
* User credentials must entered correctly.

**Post-conditions:**

* User is successfully login to the system.
* System shall redirect to User profile

#### Use Case UC3: Request portal Access

**Contract ID**: Portal Access

**Operation:** Portal Access ()

**Cross References:** Use cases: Portal Access

**Pre-conditions:**

* User is registered before login to the System.
* The System is running correctly.

**Post-conditions:**

* User Account is Available and Admin give access to the System successfully.

## Ehsanullah FA20-BSE-068

#### Request to View Result

|  |  |
| --- | --- |
| Contract CO2: Request to View Result | |
| **Operation** | Request to View Result |
| **Cross References:** | Use Cases: View Result |
| **Preconditions:** | The Result is uploaded |
| **Postconditions:** | The Result is Displayed |
| Contract CO2: Request to Add Result | |
| **Operation** | Request to Add Result |
| **Cross References:** | Use Cases: Add Result |
| **Preconditions:** | No |
| **Postconditions:** | The Result is Added |
|  |  |

## Muhammad Nawaz Khan FA20-BSE-073

#### Request to View Attendance

**Contract ID:** View Attendance

**Operation:** Request to view Attendance

**Cross References: Use**

**Use case:** view Student Attendance

**Pre-conditions:**

•User should be registered

**Post-conditions:**

Stdlist object will be updated

• The system will load the student attendance with date and progress bar.

#### Request to Add Attendance

**Contract ID:** Add Attendance

**Operation:** Request Add Attendance

**Cross References:**

**Use case:** Add student Attendance

**Pre-conditions:**

•User must be authorized and authenticated

* Teacher must be associated with the class

**Post-conditions:**

* Stdlist object will be updated
* Att object of studentRecord will be updated

• The system will add record in the database.

#### Modify Student Attendance

**Contract ID:** Modify Student Attendance

**Operation:** Request To modify Attendance

**Cross References:**

**Use case:** Modify student Attendance

**Pre-conditions:**

•User must be authorized and authenticated

* Teacher must be associated with the class

**Post-conditions:**

* Stdlist object will be updated
* Att object of studentRecord will be updated
* State of updList Object will be changed

• The system will modify the record in the database.

# CHAPTER 5 Communication Diagram – Sections

## Irfan khan FA20-BSE-070

#### Use Case UC1: register account

******

#### Use Case UC2: Login

******

## Ehsanullah FA20-BSE-068

#### Use Case: View Result

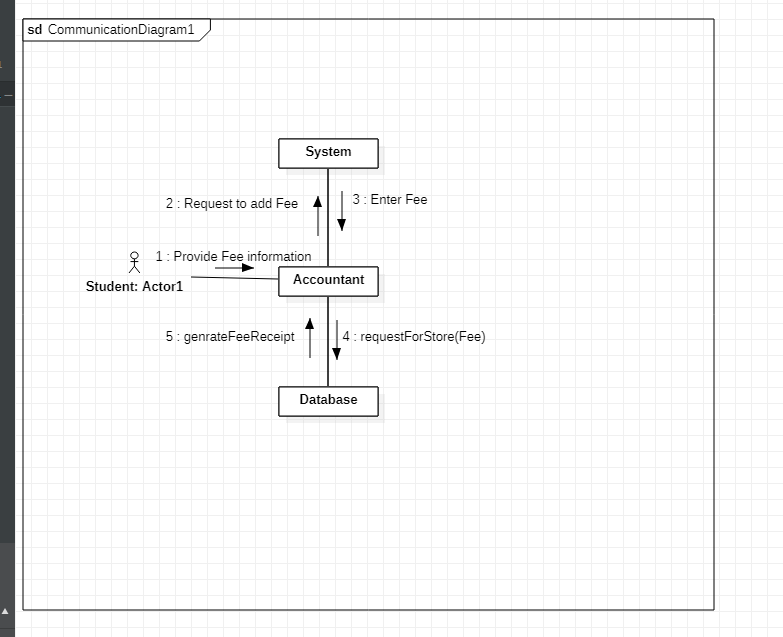
******

#### Use Case: Add Result

******

(AHSAN ALI SP20-BSE-060)

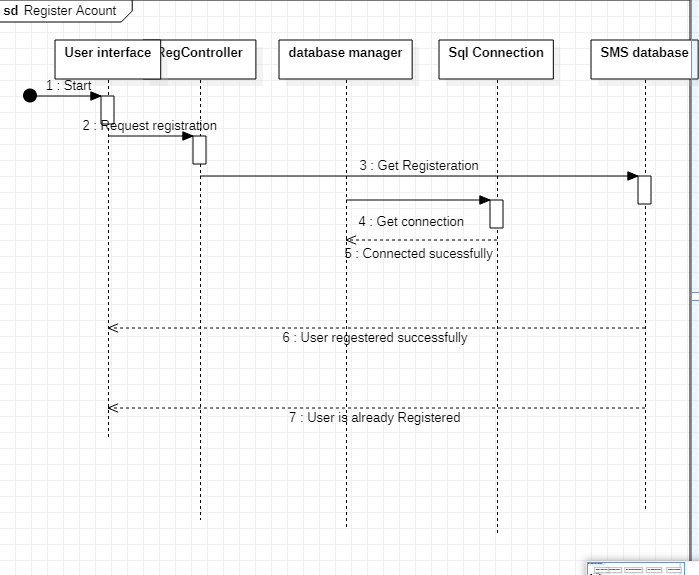
**Communication diagram**



# CHAPTER 6 Interaction diagrams

## Irfan khan FA20-BSE-070

#### Use Case UC1: register account

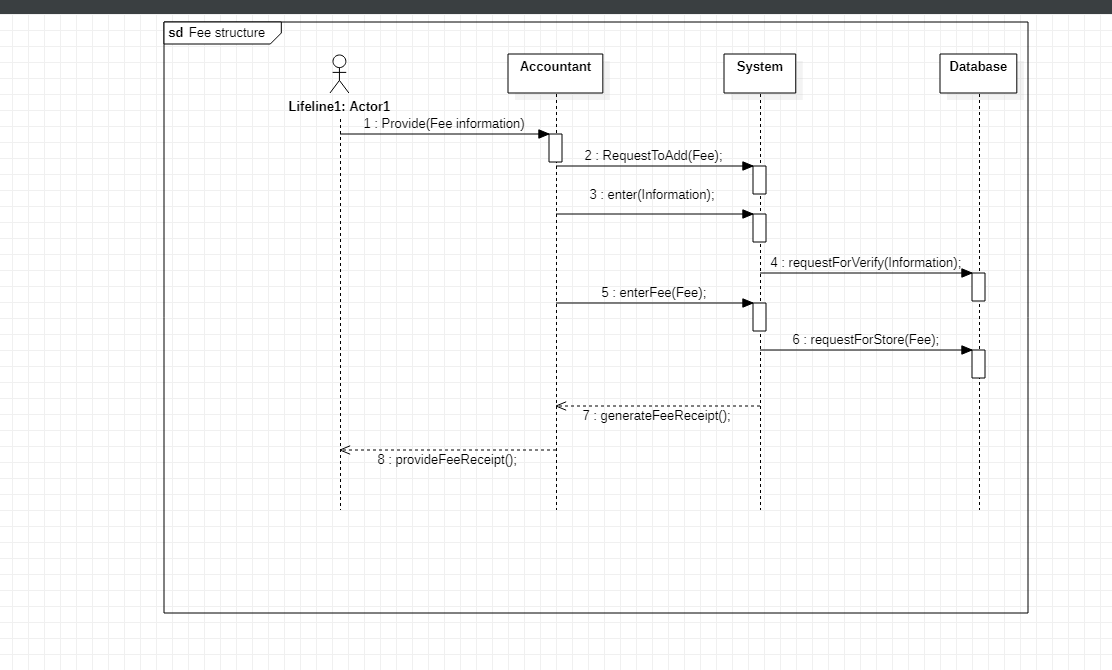


#### Use Case UC2: Login



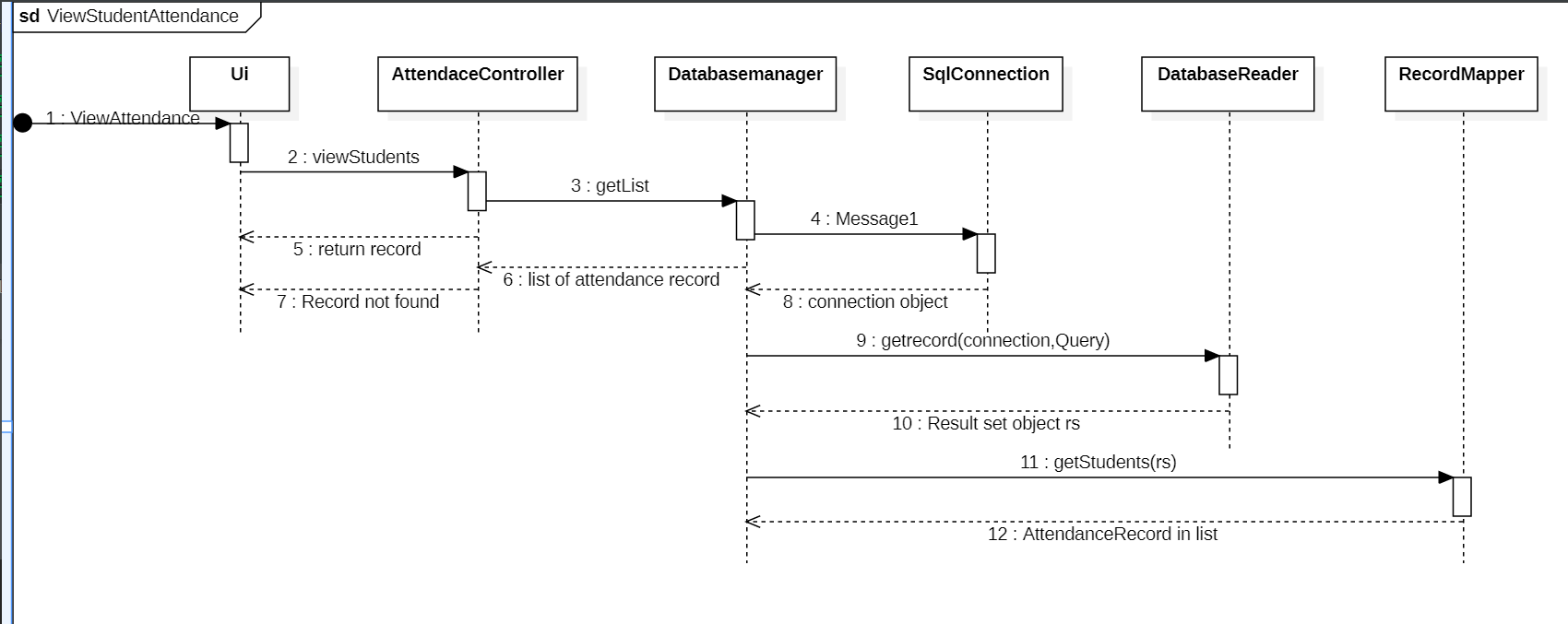
(AHSAN ALI SP20-BSE-060)

**Sequence diagram**

****

## Muhammad Nawaz khan FA20-BSE-073

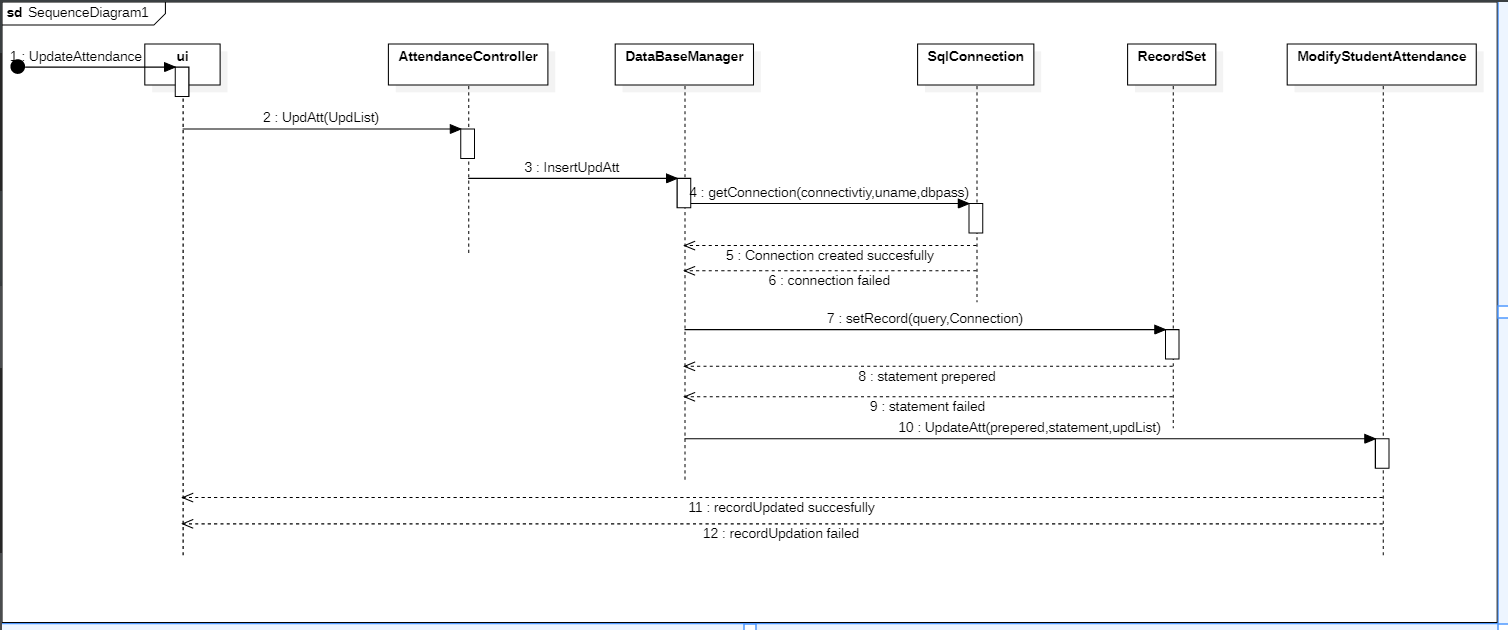
#### Use Case UC1: view Attendance



#### Use Case UC2: Add Attendance

#### 

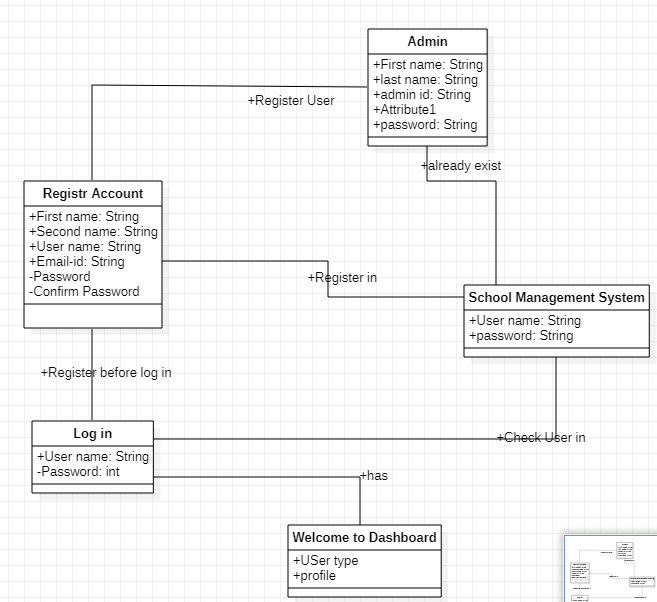
#### Use Case UC3: Modify student Attendance



# CHAPTER 7 Domain Model

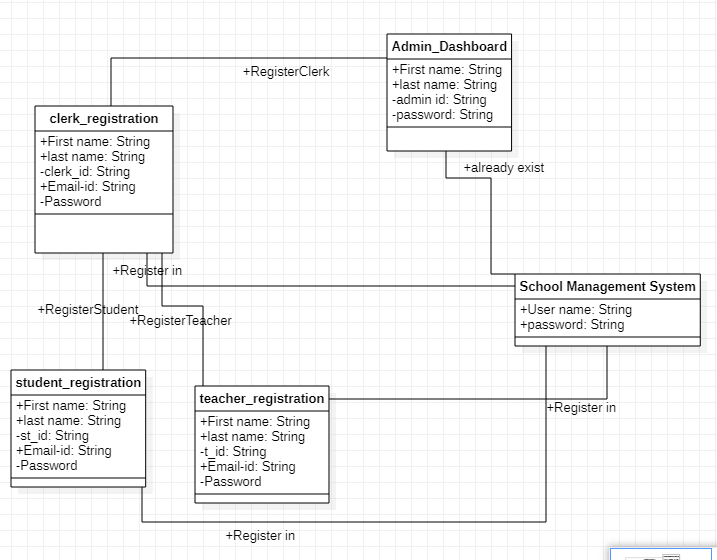
## Irfan khan FA20-BSE-070

#### Use Case UC1: register account, login, portal Access



**Reverse engineering:**

**In reverse** I compare my old Domain model with implementation. my use case is Register Account then according to my use case diagram Admin can register clerk and clerk register Student and teacher So here Admin Register Clerk **according to situation**  Admin register clerk from **Admin\_Dashboard** , then clerk was able to register student and Teacher .



## Muhammad Nawaz khan FA20-BSE-073

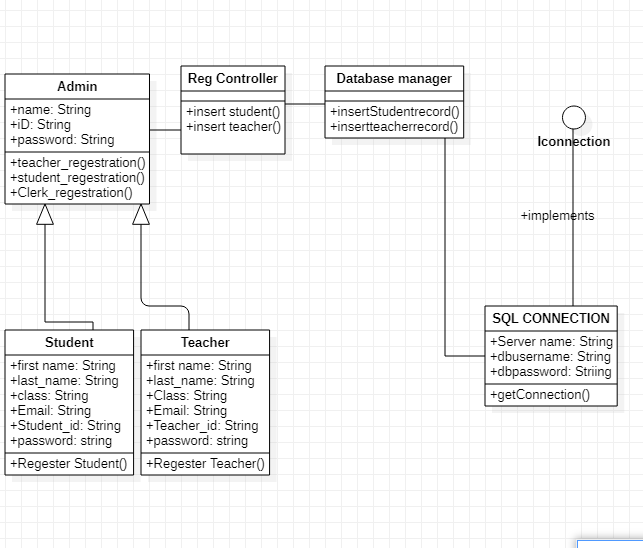
#### Use Case UC1:Add faculty Attendance, Add student attendance, view student Attendance, modify student Attendance

# 

# CHAPTER 7 Class Diagram

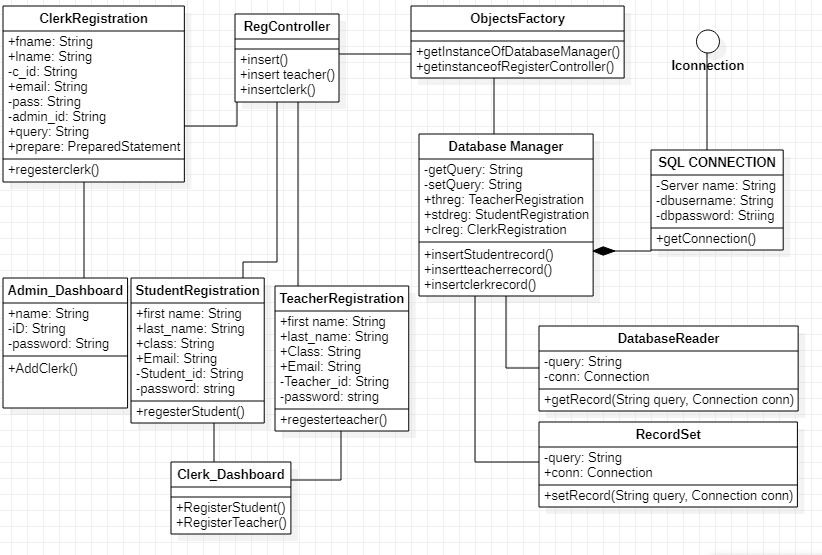
## Irfan khan FA20-BSE-070

#### Use Case UC1: register account, login, portal Access



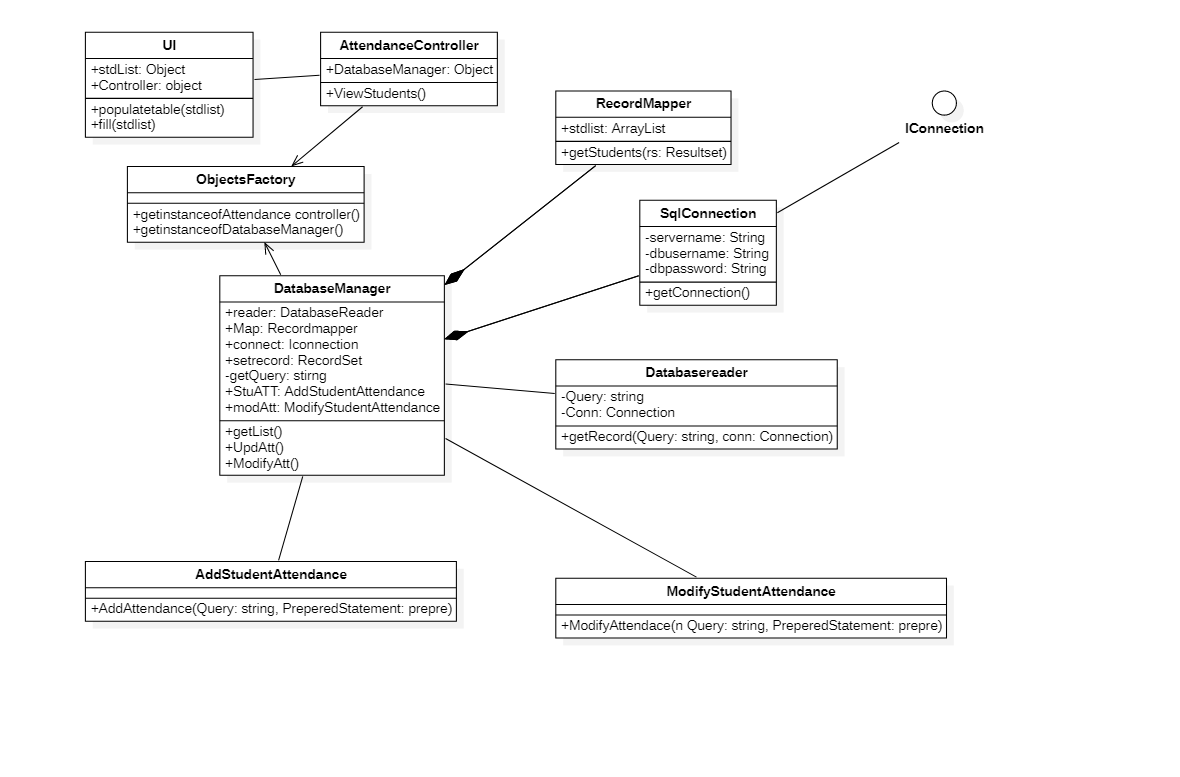
**Updated According to code:**

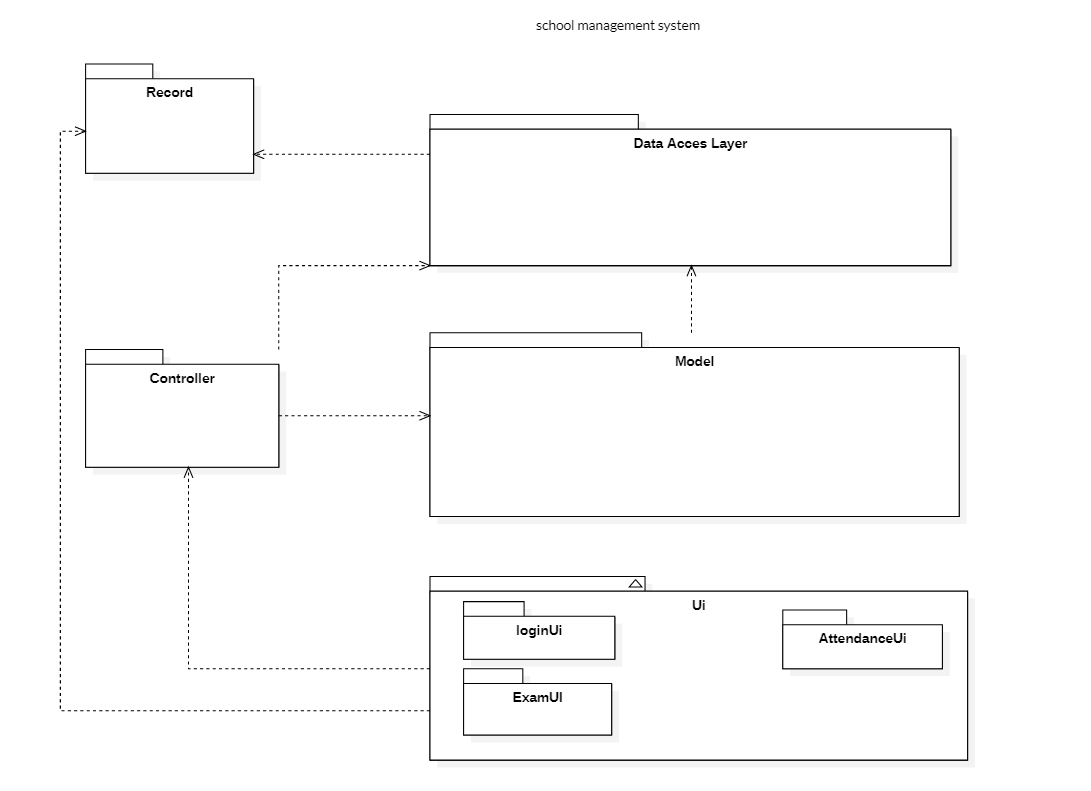
I update my class diagram according to my code



## Muhammad Nawaz khan FA20-BSE-073

#### Use Case UC1: Add student attendance, view studentAttendace, Modify Student Attendance

CHAPTER 8 package diagram:



**Reverse engineering:**

**In reverse engineering** I create Package diagram in which I mixed up a layers of UML package diagrams after coding I realize that the UI layer is become at first layer then second Domain layer and then in last Data access layer comes according to these I create in reverse engineering.

