OBJECT ORIENTED ANALYSIS AND DESIGN PROJECT PHASE-II
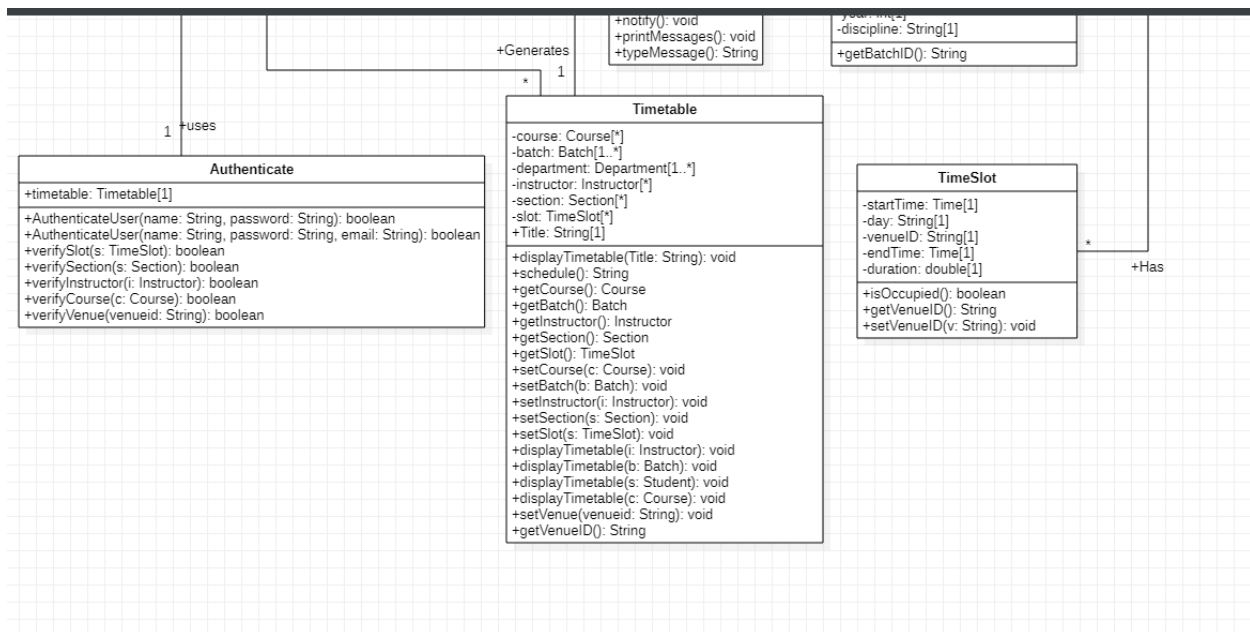# Time Table Generation System

Team#2

Group Members:

1. Ramin Rafi (17L-4284) – Group Leader
2. Tehreem Mushtaq (17L-4325)
3. Laraib Amjad (17L-4156)
4. Akash Bajwa (17L-6341)
5. Umair Ahmad (17L-4023)
6. Umer Aziz (16L-4326)
7. Michelle Akaishi (17L-4104)

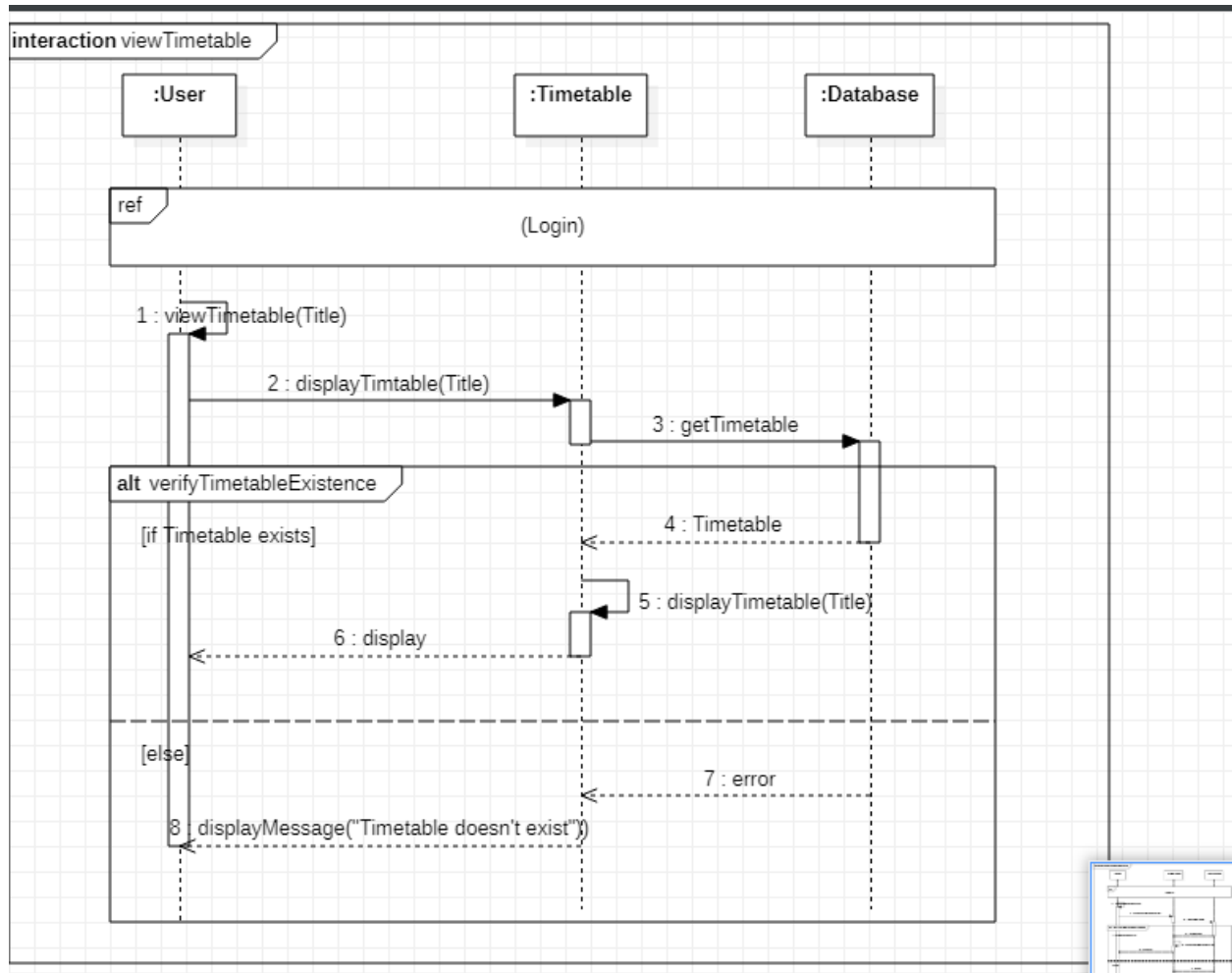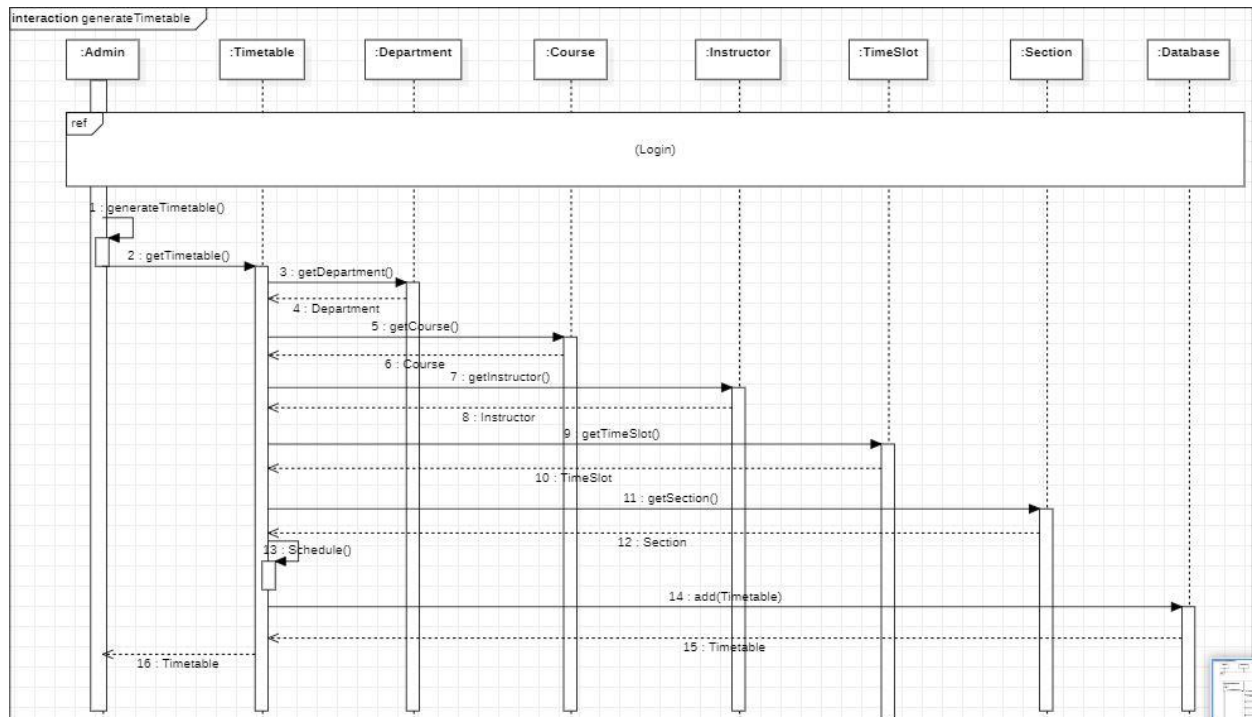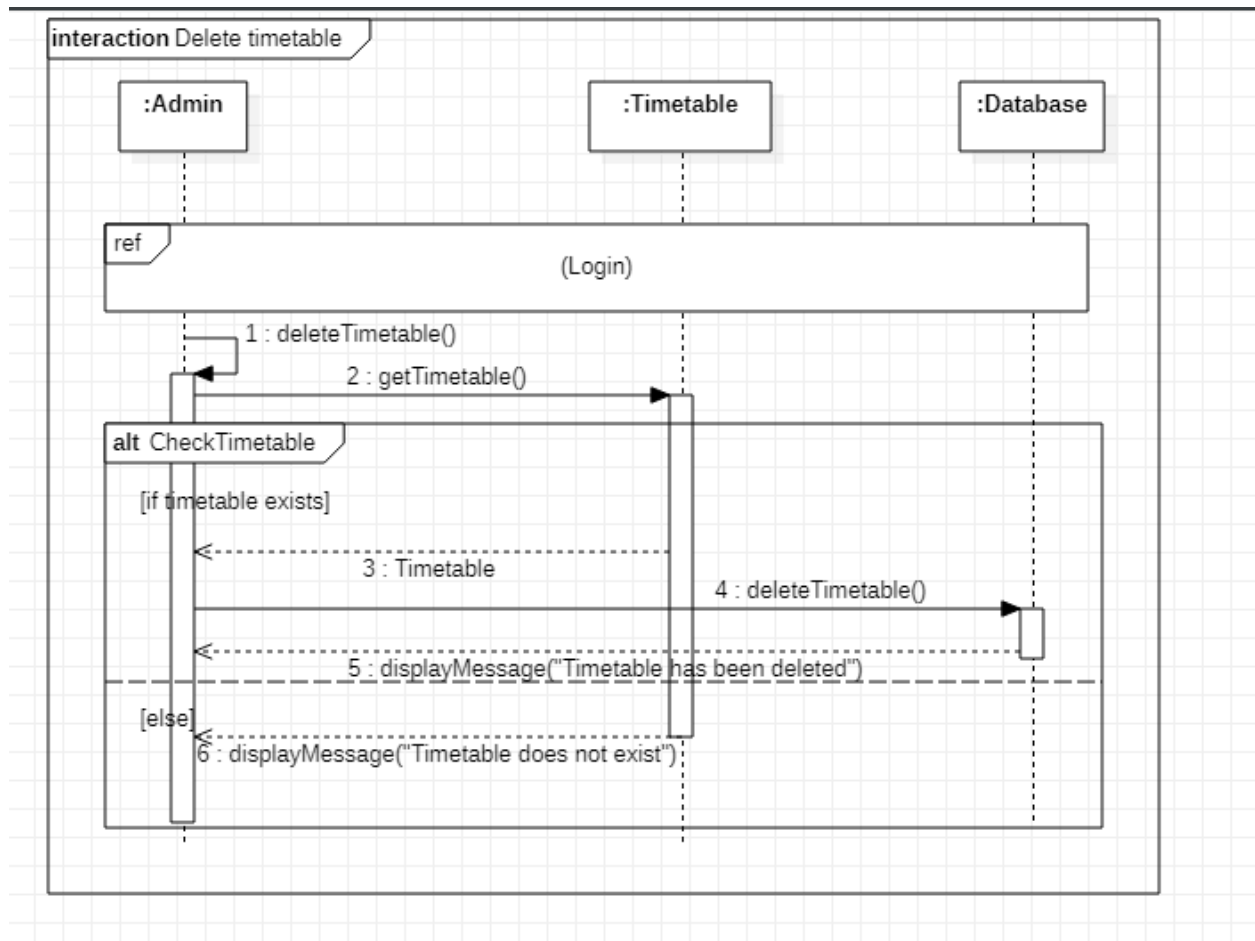**SPRING 2020**

Submission Date: **10-04-2020**

Overall Design Class Diagram:

User

#firstname: String[1]
#lastname: String[1]
#department: String[1]
#email: String[1..*]
#phoneNum: String[1..*]
#userID: Sring[1]
#password: String[1]

+viewTimetable(Title: String): void
+signup(fname: String, lname: String, mail: String, pass: String, dept: String, phone: String): String
+login(userID: int, password: String): void
+logout(): void
+viewTimetableOfDepartmentBatchwise(b: Batch): void
+viewTimetableCoursewise(c: Course): void
+viewTimetableInstructorwise(i: Instructor): void
+isRegistered(): boolean
+viewTimetableStudentwise(s: Student): void
+getEmail(): String
+getPassword(): String
+getUsername(): String

Department

-name: String[1]
-departmentID: String[1]
-coursesOffered: Course[*]

+getCoursesOffered(): Course

+Offers

Course

-courseID: String[1]
-name: String[1]
-creditHrs: int[1]
-timeduration: double[1]
-instructor: Instructor[1..*]
-prereq: String[1]
-sections: Section[1..*]

+getDuration(): double
+getInstructor(): Instructor
+getPrereq(): String
+getCoursename(): String
+setInstructor(i: Instructor): void

+Studies

+BelongsTo

Student

-numOfCourses: int[1]
-CGPA: double[1]
-batch: Batch[1]
-course: Course[*]
-currentCreditHrs: int[1]

+getnumOfCourses(): int
+getBatch(): Batch
+getCourse(): Course
+getcurrentCreditHrs(): int
+setBatch(b: Batch): void
+setCourse(c: Course): void

Section

-currentSeats: int[1]
-maxCapacity: int[1]
-course: Course[1]
-name: char[1]

+isAvailable(): boolean
+getCourse(): Course
+setCourse(c: Course): void

+Has

Admin

+generateTimetable(): Timetable
+deleteTimetable(timetable: Timetable): boolean
+addSlot(s: TimeSlot): boolean
+addSection(s: Section): boolean
+addInstructor(i: Instructor): boolean
+deleteSlot(s: TimeSlot): boolean
+deleteSection(s: Section): boolean
+deleteInstructor(i: Instructor): boolean
+addCourse(c: Course): boolean
+deleteCourse(c: Course): boolean
+getTimetable(Title: String): Timetable
+deleteVenue(venueID: String): boolean

Instructor

-numOfCourses: int[1]
-course: Course[*]

+getCourse(): Course
+getnumOfCourses(): int
+setCourse(c: Course): void

+MessagesTo

MessagingThread

+messages: String[*]
+status: String[1]

+notify(): void
+printMessages(): void
+typeMessage(): String

Batch

-batchID: String[1]
-year: int[1]
-discipline: String[1]

+getBatchID(): String

+BelongsTo

+Views

+Generates

Authenticate

+timetable: Timetable[1]

+AuthenticateUser(name: String, password: String): boolean
+AuthenticateUser(name: String, password: String, email: String): boolean
+verifySlot(s: TimeSlot): boolean
+verifySection(s: Section): boolean
+verifyInstructor(i: Instructor): boolean
+verifyCourse(c: Course): boolean
+verifyVenue(venueid: String): boolean

+uses

Timetable

-course: Course[*]
-batch: Batch[1..*]
-department: Department[1..*]
-instructor: Instructor[*]
-section: Section[*]
-slot: TimeSlot[*]
+Title: String[1]

+displayTimetable(Title: String): void
+schedule(): String
+getCourse(): Course
+getBatch(): Batch
+getInstructor(): Instructor
+getSection(): Section
+getSlot(): TimeSlot
+setCourse(c: Course): void
+setBatch(b: Batch): void
+setInstructor(i: Instructor): void
+setSection(s: Section): void
+setSlot(s: TimeSlot): void
+displayTimetable(i: Instructor): void
+displayTimetable(b: Batch): void
+displayTimetable(s: Student): void
+displayTimetable(c: Course): void
+setVenue(venueid: String): void
+getVenueID(): String

TimeSlot

-startTime: Time[1]
-day: String[1]
-venueID: String[1]
-endTime: Time[1]
-duration: double[1]

+isOccupied(): boolean
+getVenueID(): String
+setVenueID(v: String): void

+Has

Close ups:

User
```
#firstname: String[1]
#lastname: String[1]
#department: String[1]
#email: String[1..*]
#phoneNum: String[1..*]
#userID: Sring[1]
#password: String[1]
```
```
+viewTimetable(Title: String): void
+signup(fname: String, lname: String, mail: String, pass: String, dept: String, phone: String): String
+login(userID: int, password: String): void
+logout(): void
+viewTimetableOfDepartmentBatchwise(b: Batch): void
+viewTimetableCoursewise(c: Course): void
+viewTimetableInstructorwise(i: Instructor): void
+isRegistered(): boolean
+viewTimetableStudentwise(s: Student): void
+getEmail(): String
+getPassword(): String
+getUsername(): String
```

+BelongsTo

Department
```
-name: String[1]
-departmentID: String[1]
-coursesOffered: Course[*]
```
```
+getCoursesOffered(): Course
```

+Offers

Course
```
-courseID: String[1]
-name: String[1]
-creditHrs: int[1]
-timeduration: double[1]
-instructor: Instructor[1..*]
-prereq: String[1]
+sections: Section[1..*]
```
```
+getDuration(): double
+getInstructor(): Instructor
+getPrereq(): String
+getCoursename(): String
+setInstructor(i: Instructor): void
```

+Studies

Student
```
-numOfCourses: int[1]
-CGPA: double[1]
-batch: Batch[1]
-course: Course[*]
-currentCreditHrs: int[1]
```
```
+getnumOfCourses(): int
+getBatch(): Batch
+getCourse(): Course
+getcurrentCreditHrs(): int
+setBatch(b: Batch): void
+setCourse(c: Course): void
```

+Has

Section
```
-currentSeats: int[1]
-maxCapacity: int[1]
-course: Course[1]
-name: char[1]
```
```
+isAvailable(): boolean
+getCourse(): Course
+setCourse(c: Course): void
```

Admin
```
+generateTimetable(): Timetable
+deleteTimetable(timetable: Timetable): boolean
+addSlot(s: TimeSlot): boolean
+addSection(s: Section): boolean
+addInstructor(i: Instructor): boolean
+deleteSlot(s: TimeSlot): boolean
+deleteSection(s: Section): boolean
+deleteInstructor(i: Instructor): boolean
+addCourse(c: Course): boolean
+deleteCourse(c: Course): boolean
+getTimetable(Title: String): Timetable
+deleteVenue(venueID: String): boolean
```

Instructor
```
-numOfCourses: int[1]
-course: Course[*]
```
```
+getCourse(): Course
+getnumOfCourses(): int
+setCourse(c: Course): void
```

+MessagesTo

+Views

MessagingThread
```
+messages: String[*]
```

+BelongsTo

Batch

Student (continued)
```
-batch: Batch[1]
-course: Course[*]
-currentCreditHrs: int[1]
```
```
+getnumOfCourses(): int
+getBatch(): Batch
+getCourse(): Course
+getcurrentCreditHrs(): int
+setBatch(b: Batch): void
+setCourse(c: Course): void
```

Section
```
-currentSeats: int[1]
-maxCapacity: int[1]
-course: Course[1]
-name: char[1]
```
```
+isAvailable(): boolean
+getCourse(): Course
+setCourse(c: Course): void
```

Admin
```
+generateTimetable(): Timetable
+deleteTimetable(timetable: Timetable): boolean
+addSlot(s: TimeSlot): boolean
+addSection(s: Section): boolean
+addInstructor(i: Instructor): boolean
+deleteSlot(s: TimeSlot): boolean
+deleteSection(s: Section): boolean
+deleteInstructor(i: Instructor): boolean
+addCourse(c: Course): boolean
+deleteCourse(c: Course): boolean
+getTimetable(Title: String): Timetable
+deleteVenue(venueID: String): boolean
```

Instructor
```
-numOfCourses: int[1]
-course: Course[*]
```
```
+getCourse(): Course
+getnumOfCourses(): int
+setCourse(c: Course): void
```

+MessagesTo

+Views

MessagingThread
```
+messages: String[*]
+status: String[1]
```
```
+notify(): void
+printMessages(): void
+typeMessage(): String
```

+BelongsTo

Batch
```
-batchID: String[1]
-year: int[1]
-discipline: String[1]
```
```
+getBatchID(): String
```

+Generates

+uses

Authenticate
```
+timetable: Timetable[1]
```
```
+AuthenticateUser(name: String, password: String): boolean
+AuthenticateUser(name: String, password: String, email: String): boolean
+verifySlot(s: TimeSlot): boolean
+verifySection(s: Section): boolean
+verifyInstructor(i: Instructor): boolean
+verifyCourse(c: Course): boolean
+verifyVenue(venueid: String): boolean
```

Timetable
```
-course: Course[*]
-batch: Batch[1..*]
-department: Department[1..*]
-instructor: Instructor[*]
-section: Section[*]
-slot: TimeSlot[*]
+Title: String[1]
```
```
+displayTimetable(Title: String): void
+schedule(): String
+getCourse(): Course
+getBatch(): Batch
+getInstructor(): Instructor
+getSection(): Section
+getSlot(): TimeSlot
```

TimeSlot
```
-startTime: Time[1]
-day: String[1]
-venueID: String[1]
-endTime: Time[1]
-duration: double[1]
```
```
+isOccupied(): boolean
+getVenueID(): String
+setVenueID(v: String): void
```

+Has

+notify(): void
+printMessages(): void
+typeMessage(): String

-discipline: String[1]
+getBatchID: String

+Generates   1

*

Timetable

-course: Course[*]
-batch: Batch[1..*]
-department: Department[1..*]
-instructor: Instructor[*]
-section: Section[*]
-slot: TimeSlot[*]
+Title: String[1]

+displayTimetable(Title: String): void
+schedule(): String
+getCourse(): Course
+getBatch(): Batch
+getInstructor(): Instructor
+getSection(): Section
+getSlot(): TimeSlot
+setCourse(c: Course): void
+setBatch(b: Batch): void
+setInstructor(i: Instructor): void
+setSection(s: Section): void
+setSlot(s: TimeSlot): void
+displayTimetable(i: Instructor): void
+displayTimetable(b: Batch): void
+displayTimetable(s: Student): void
+displayTimetable(c: Course): void
+setVenue(venueid: String): void
+getVenueID(): String

1   +uses

Authenticate

+timetable: Timetable[1]

+AuthenticateUser(name: String, password: String): boolean
+AuthenticateUser(name: String, password: String, email: String): boolean
+verifySlot(s: TimeSlot): boolean
+verifySection(s: Section): boolean
+verifyInstructor(i: Instructor): boolean
+verifyCourse(c: Course): boolean
+verifyVenue(venueid: String): boolean

TimeSlot

-startTime: Time[1]
-day: String[1]
-venueID: String[1]
-endTime: Time[1]
-duration: double[1]

+isOccupied(): boolean
+getVenueID(): String
+setVenueID(v: String): void

*

+Has

Sequence Diagrams:

Design Sequence Diagram of UC1: view timetable

Design Sequence Diagram of UC2: generate timetable

Design Sequence Diagram of UC3: delete timetable

interaction Delete timetable

:Admin

:Timetable

:Database

ref

(Login)

1 : deleteTimetable()

2 : getTimetable()

alt CheckTimetable

[if timetable exists]

3 : Timetable

4 : deleteTimetable()

5 : displayMessage("Timetable has been deleted")

[else]

6 : displayMessage("Timetable does not exist")

Design Sequence Diagram of UC4: add slot

interaction AddSlot

:Admin   :Authenticate   :Timetable   :Database

ref
(Login)

1 : addSlot(slot)

2 : verifySlot

3 : getSlot

alt

4 : setSlot(slot)

[slot doesn't exist]

5 : true

6 : true

7 : displayMessage("Slot has been added")

8 : error

[else]

9 : displayMessage("Slot already exists")

Design Sequence Diagram of UC5: add section

Design Sequence Diagram of UC6: delete section

Design Sequence Diagram of UC7: login

Design Sequence Diagram of UC8: logout

Design Sequence Diagram of UC10: add course

Design Sequence Diagram of UC11: delete course

interaction delete course

| :Admin | :Authenticate | :Timetable | :Database |

ref
(Login)

1 : deleteCourse(course)

2 : verifyCourse(course)

3 : getCourse()

alt   [course exists]

4 : deleteCourse(course)

5 : true

6 : true

loop   [for each section]

7 : deleteSection(section)

8 : displayMessage("Course removed successfully")

9 : error

[else]   10 : displayMessage("Course does not exist")
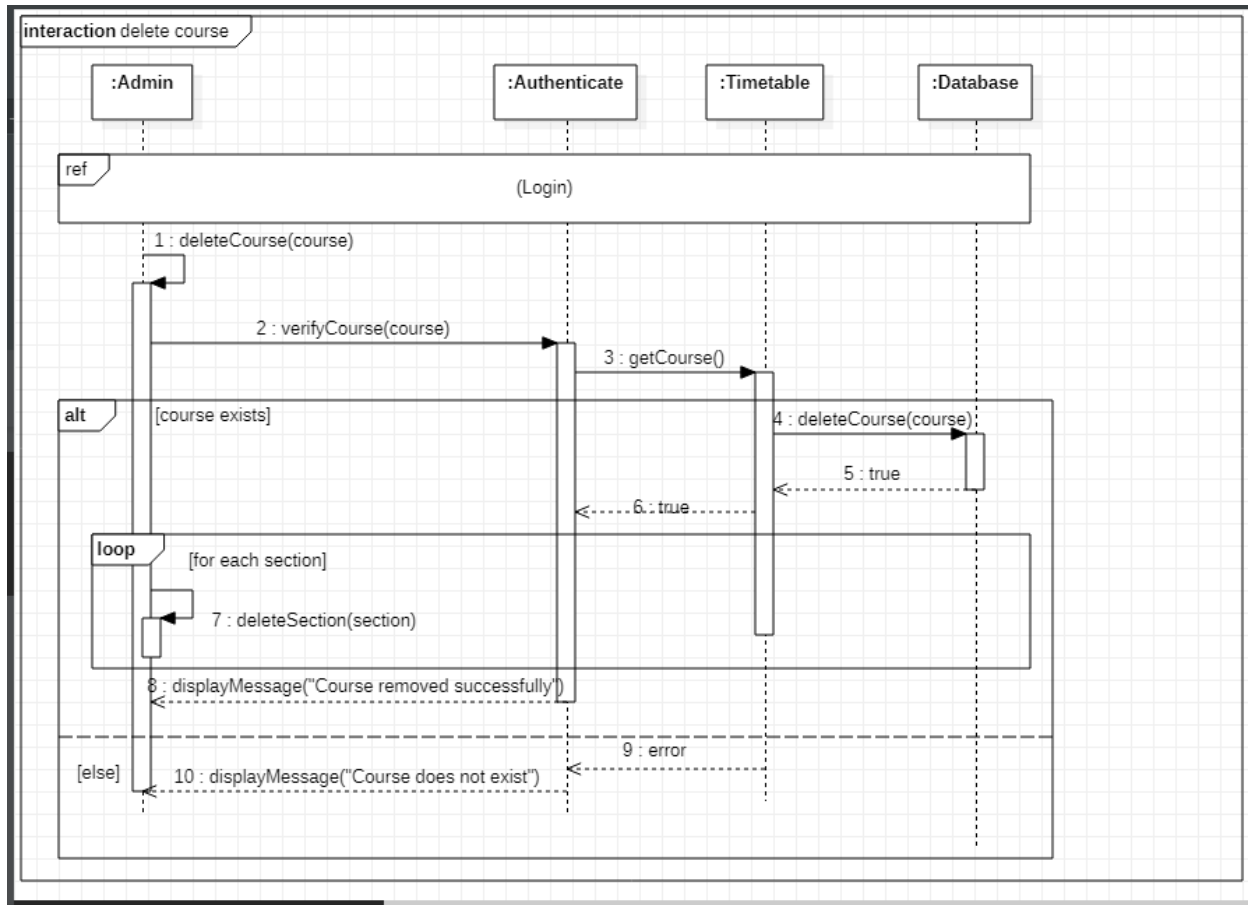
Design Sequence Diagram of UC12: add venue

interaction Addvenue

:Admin   :Authenticate   :Timetable   :Database

ref

(Login)

1 : addVenue(venueID)

2 : verifyVenue   3 : getVenueID

alt checkVenue

[If venue doesn't exist]

4 : setVenue(venueID)

7 : displayMessage("Venue is added")   6 : true   5 : true

[else]

8 : error

9 : displayMessage("Venue already exist")
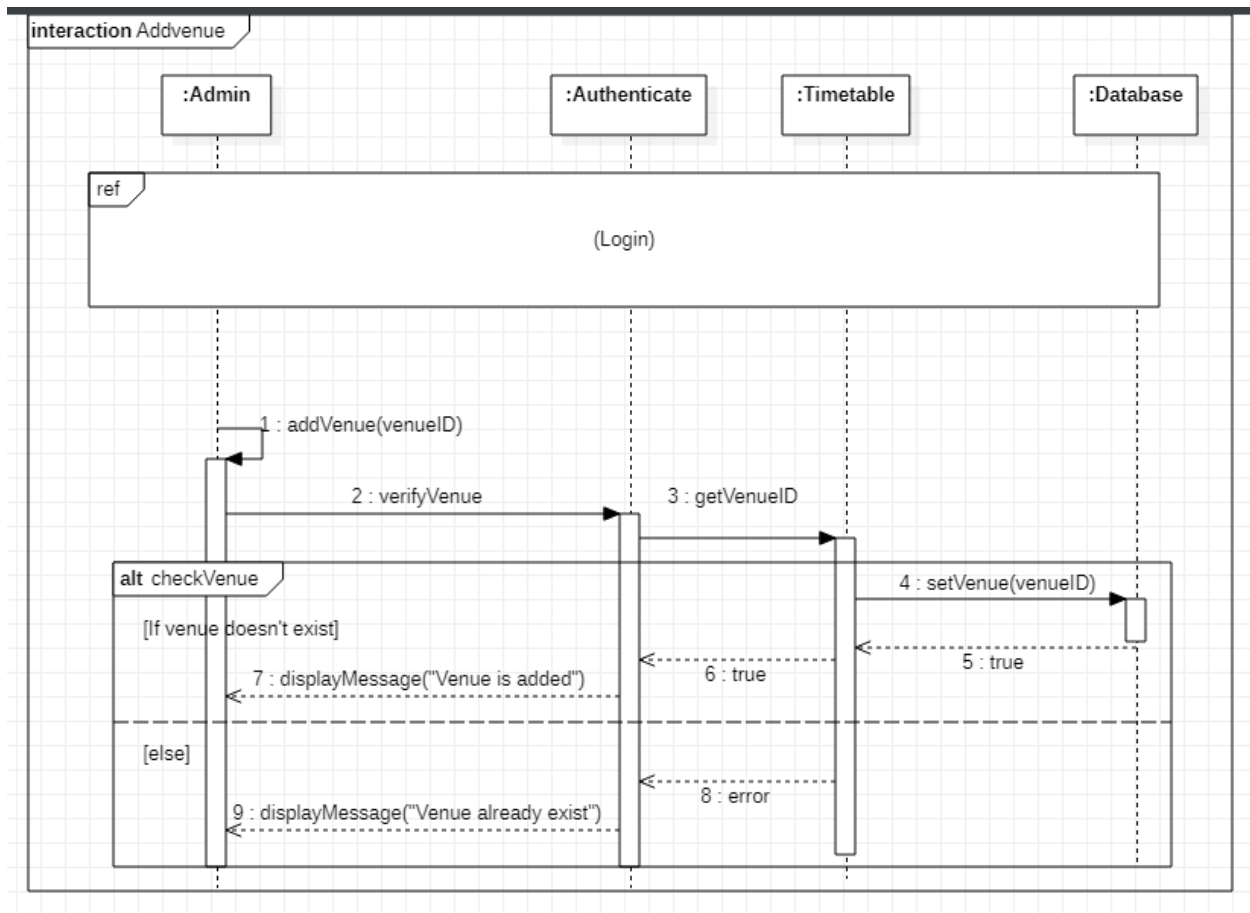
Design Sequence Diagram of UC13: delete venue
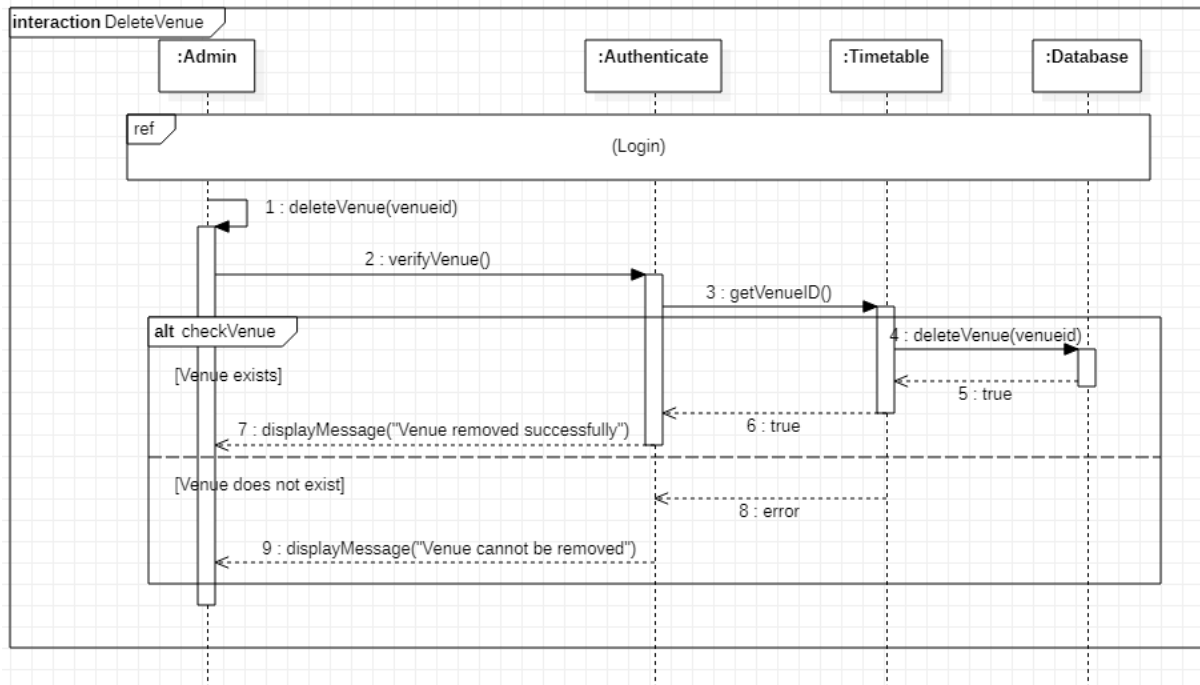
Design Sequence Diagram of UC14: delete instructor

interaction delete instructor

:Admin   :Authenticate   :Timetable   :Database

ref

(Login)

1 : deleteInstructor(Instructor)

2 : verifyInstructor(instructor)

3 : getInstructor()

alt CheckInstructor

[Instructor Exists]

4 : delete(instructor)

5 : true

6 : true

7 : displayMessage("Instructor removed successfully")

[else]

8 : error

9 : displayMessage("Instructor cannot be removed ")
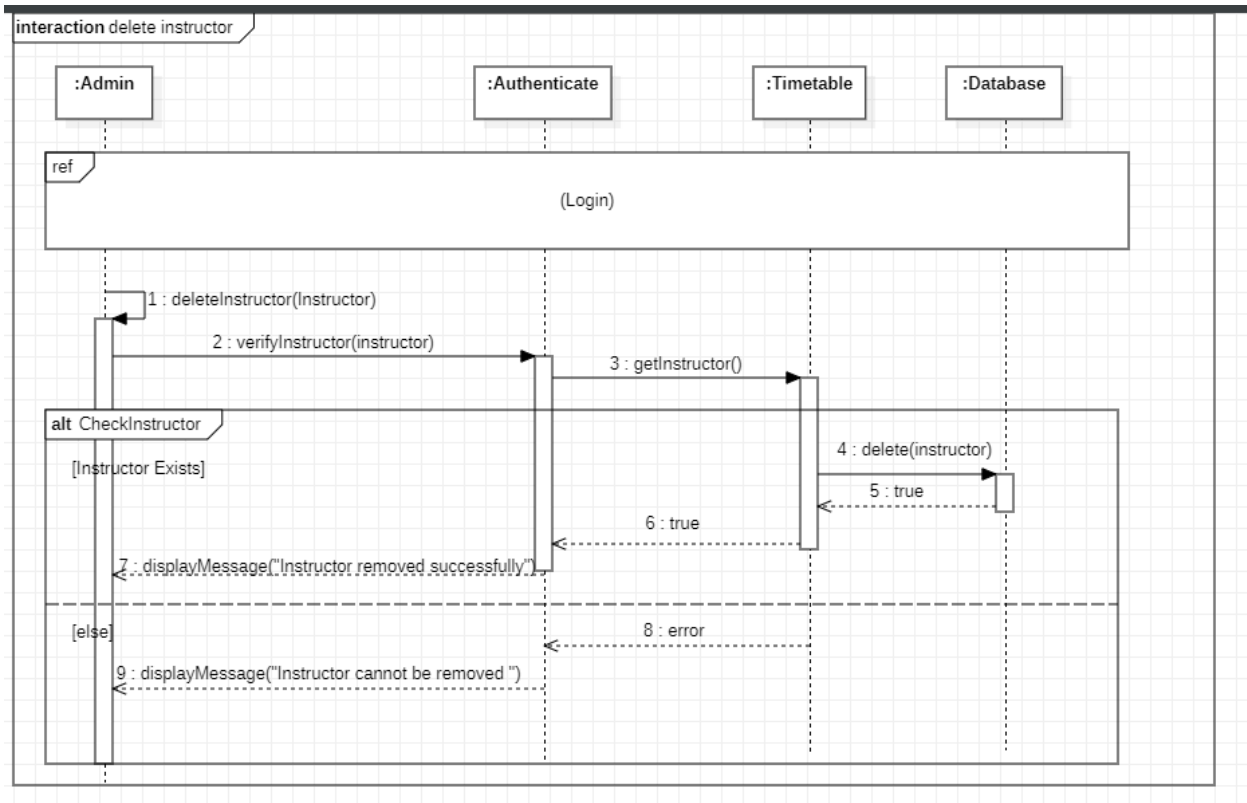
Design Sequence Diagram of UC15: add instructor
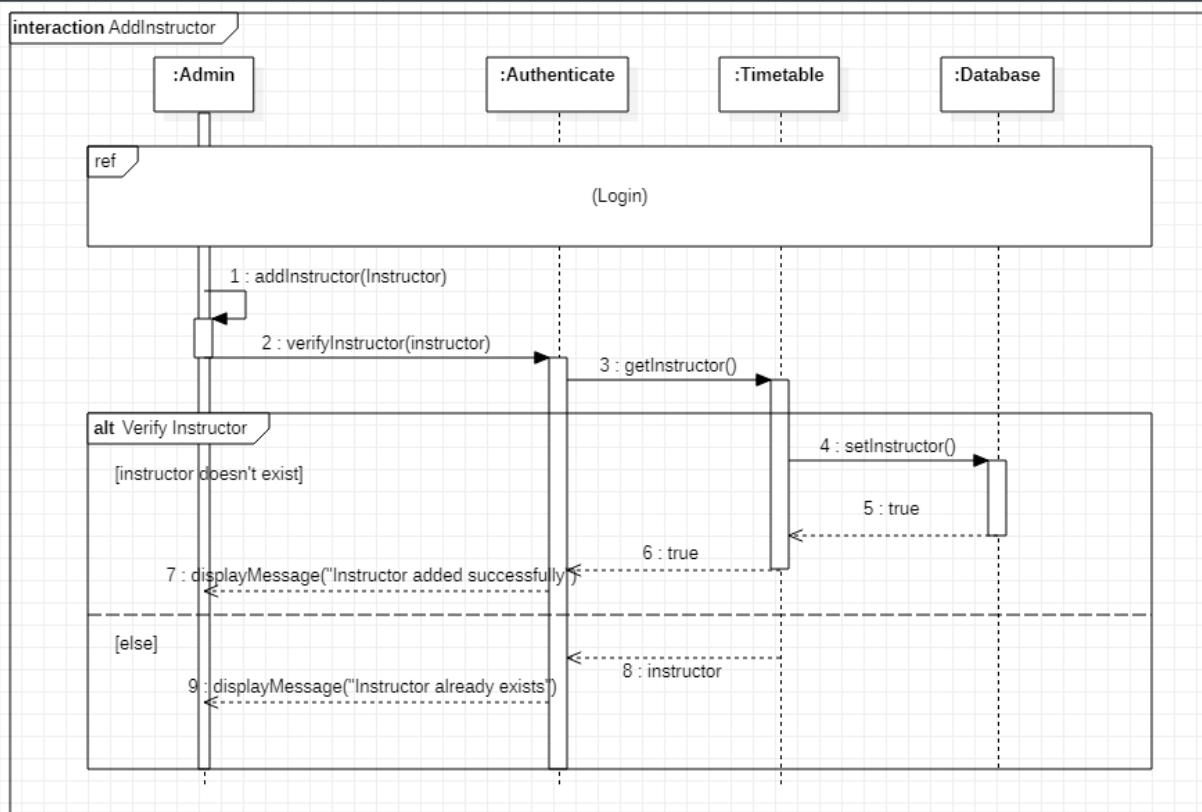
Design Sequence Diagram of UC16: view timetable of department batch wise

interaction view timetable of department batchwise

:User
:Timetable
:Database

ref
(Login)

1 : viewTimetableOfDepartmentBatchwise(batch)

2 : displayTimetable(batch)

3 : getTimetable()

alt [timetable exists]

4 : timetable

5 : displayTimetable(batch)

6 : display

[else]

7 : error

8 : displayMessage("Timetable does not exist")
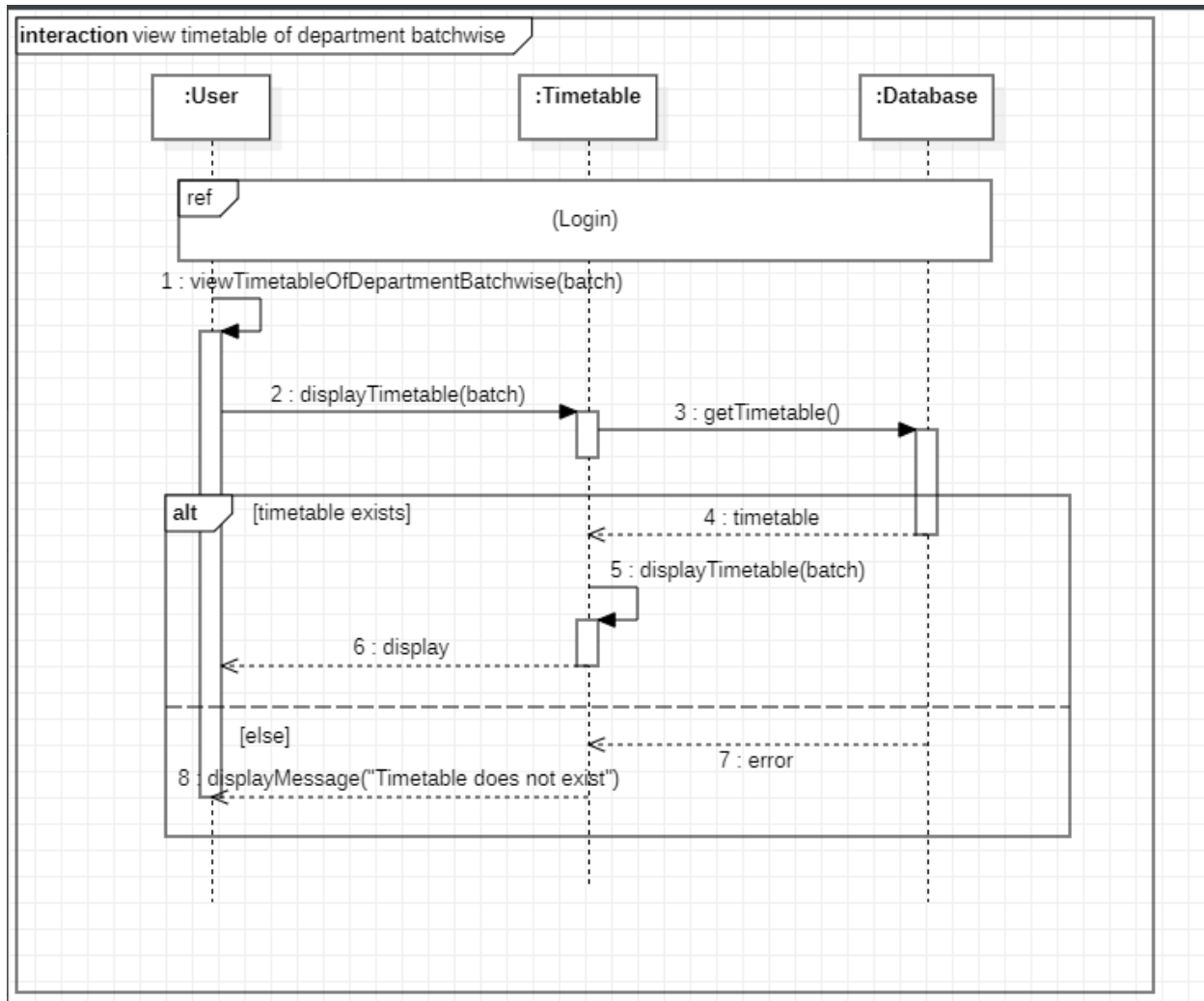
Design Sequence Diagram of UC17: view timetable course wise

interaction view timetable coursewise

:User	:Timetable	:Database

ref
(Login)

1 : viewTimetableCoursewise(course)

2 : displayTimetable(course)

3 : getTimetable()

alt [timetable exists]

4 : timetable

5 : displayTimetable(course)

6 : display

[else]

7 : error

8 : displayMessage("Timetable does not exist")
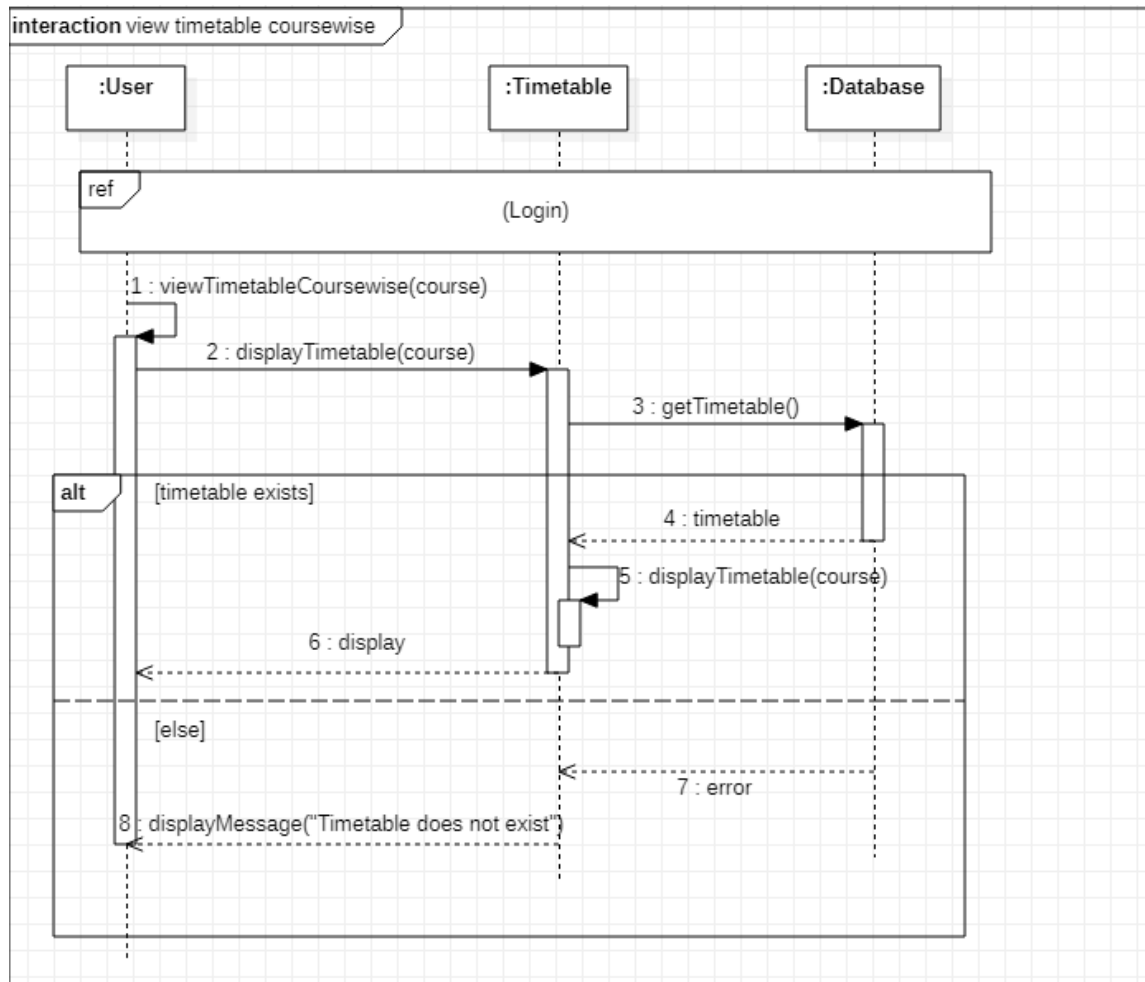
Design Sequence Diagram of UC18: view timetable instructor wise

interaction view timetable instructor wise

:User   :Timetable   :Database

ref
(Login)

1 : viewTimetableInstructorwise(instructor)

2 : displayTimetable(instuctor)

3 : getTimetable()

alt   [timetable exists]

4 : timetable

5 : displayTimetable(instructor)

6 : display

[else]

7 : error

8 : displayMessage("Timetable does not exist")
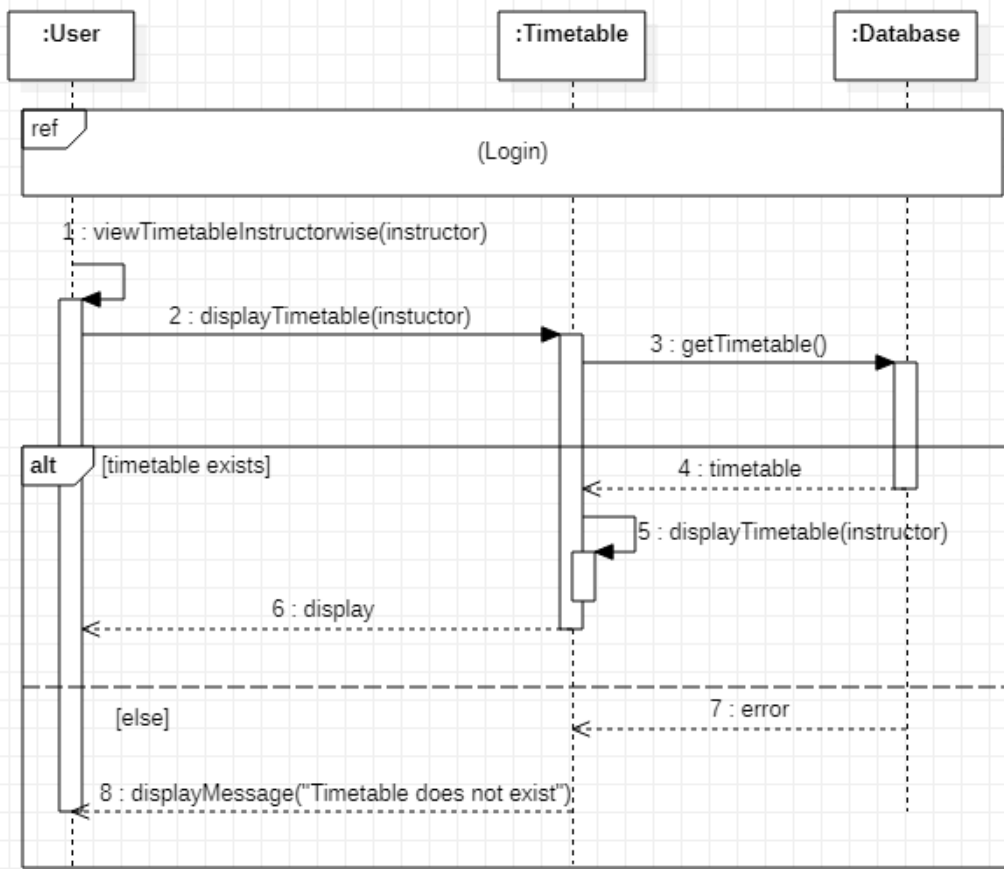
Design Sequence Diagram of UC19: view timetable student wise

interaction view timetable studentwise

| :User | :Timetable | :Database |

ref

(Login)

1 : viewTimetableStudentwise(student)

2 : getTimetable()

alt [timetable exists]

3 : timetable

4 : displayTimetable(student)

5 : display

[else]

6 : error

7 : displayMessage("Timetable does not exist")
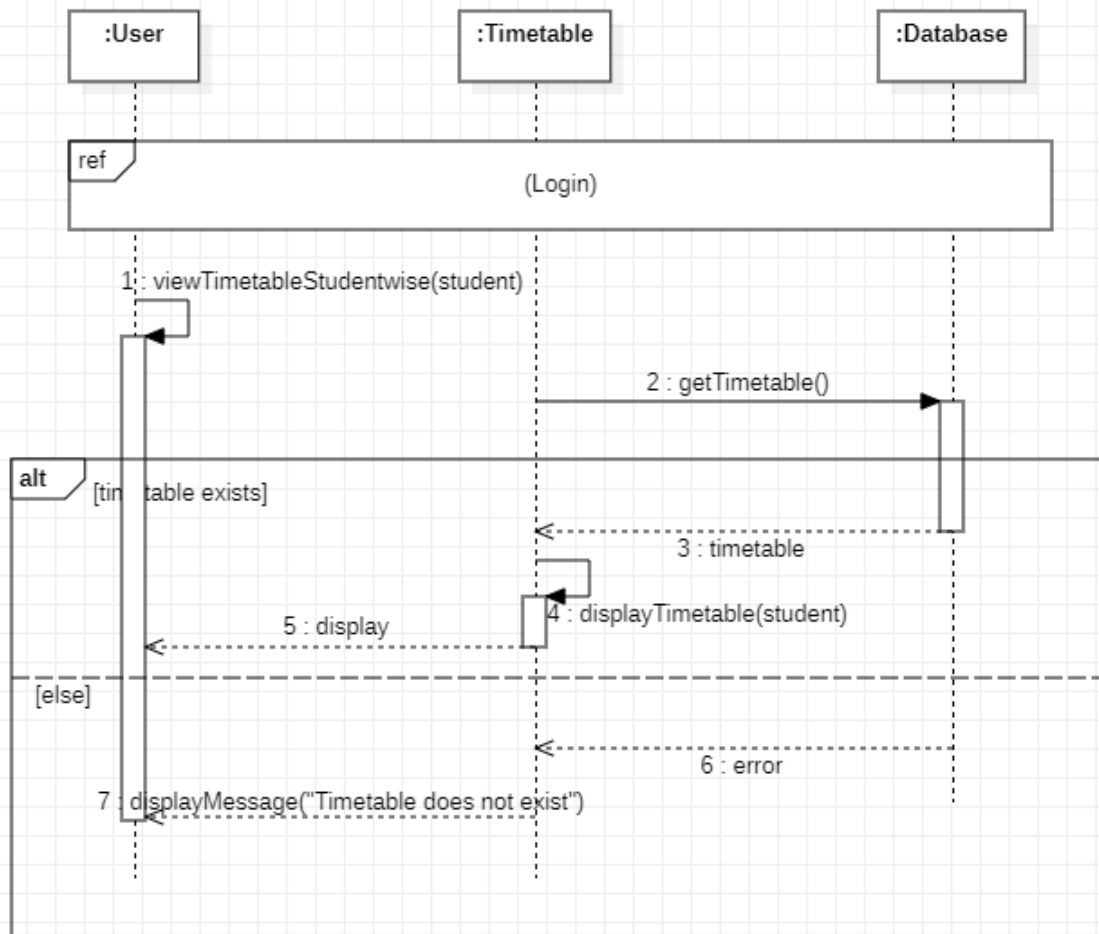
Design Sequence Diagram of UC20: delete slot
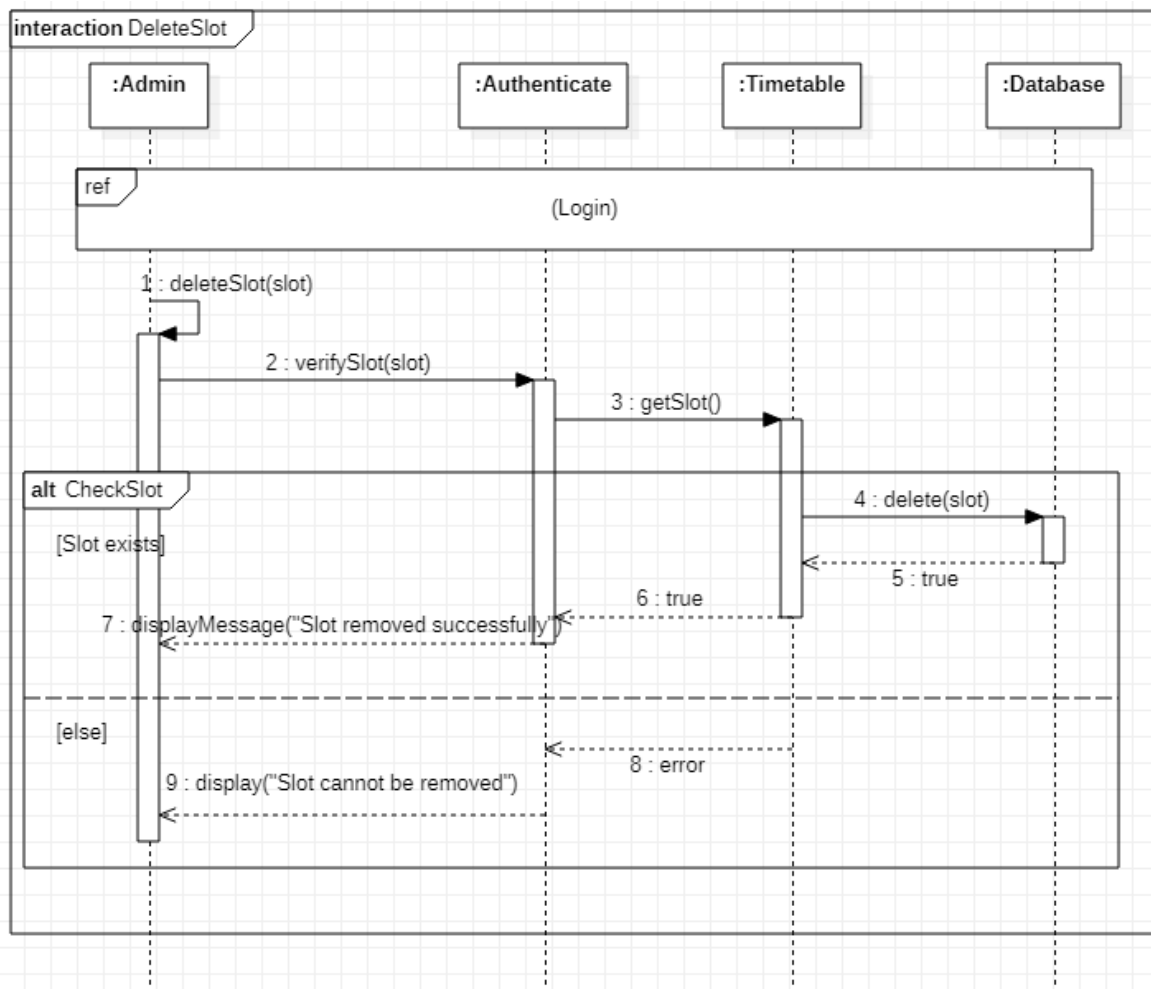
Design Sequence Diagram of UC21: signup
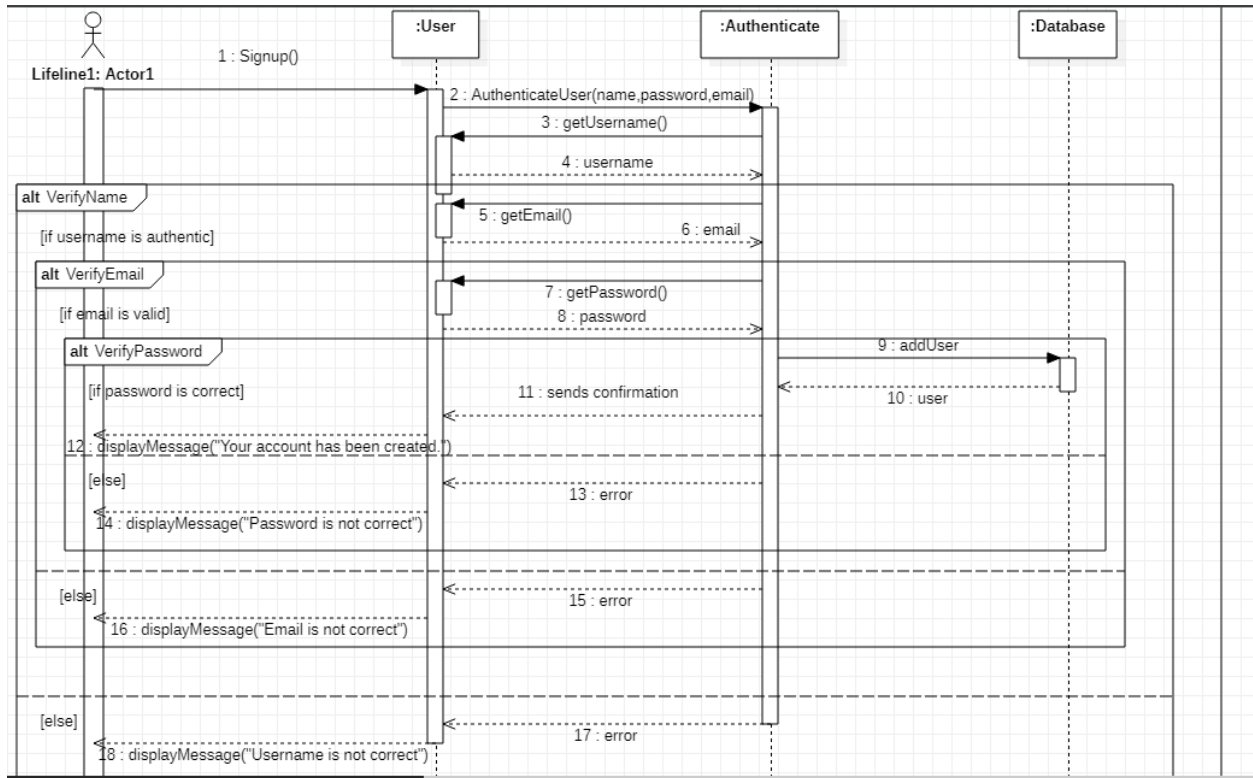
Prototype:

Home Page:
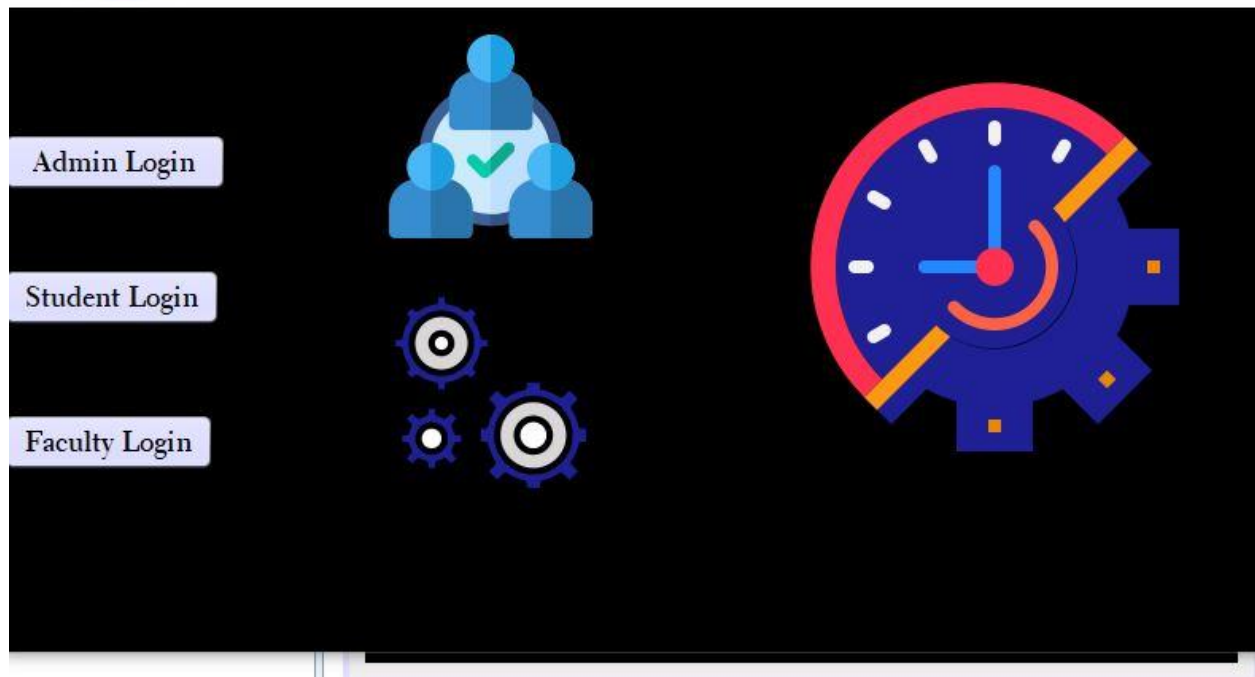
Screen of UC21: signup

SignUp

First Name:	Last Name:
Email:	Phone:
Password:	Dept.

Sign up

Screen of UC7: Login

Admin Login

Admin Id:

Password:

Login

Student Login

Student Id:

Password:

Login

Admin Portal:
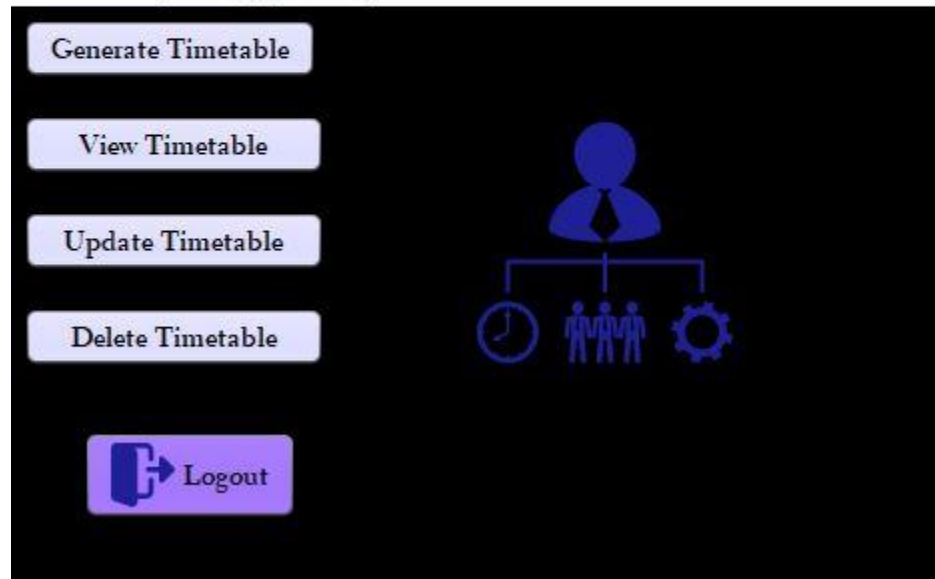
Student Page:
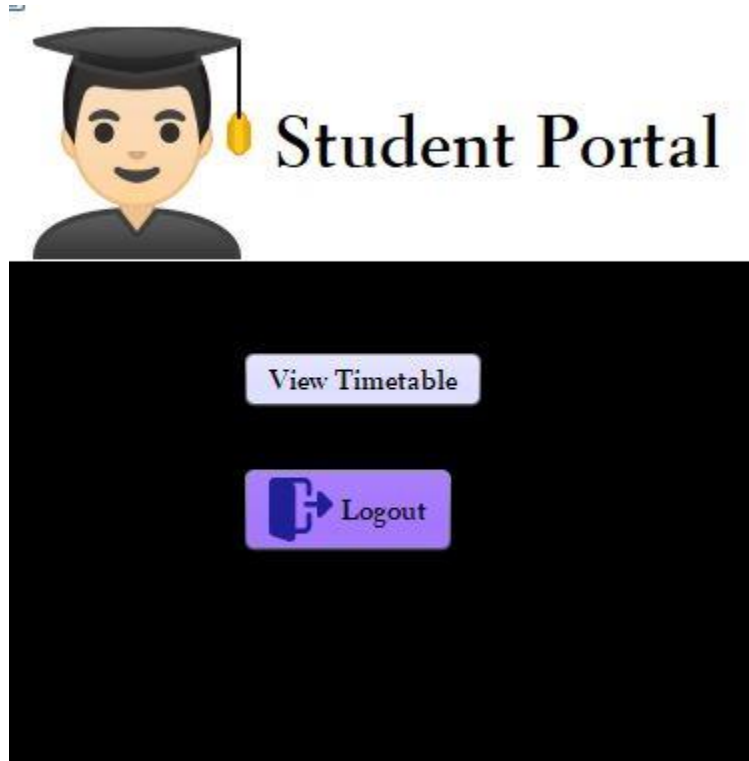
Instructor Page:
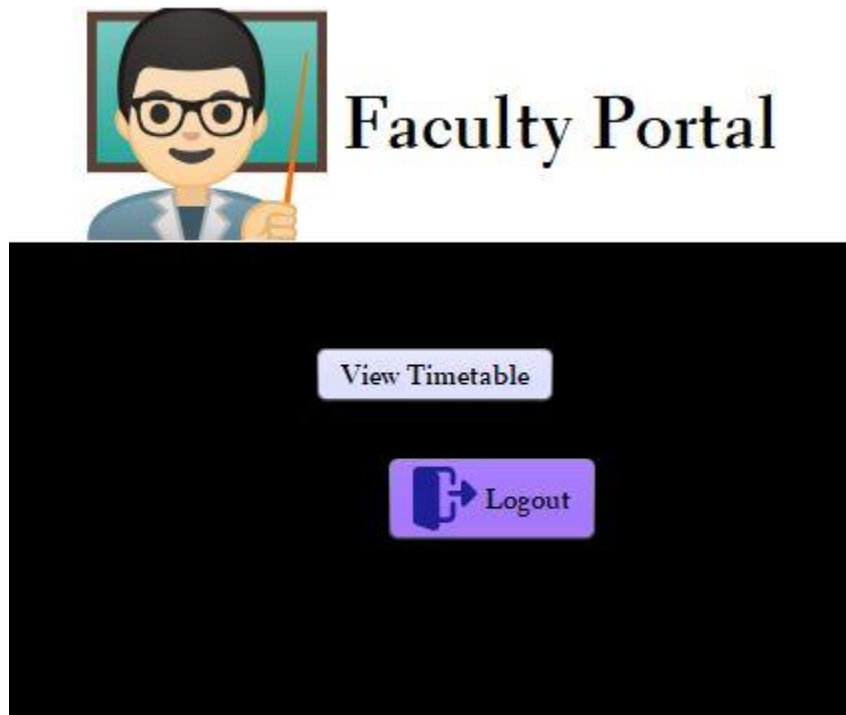
Screen of UC2: generate timetable
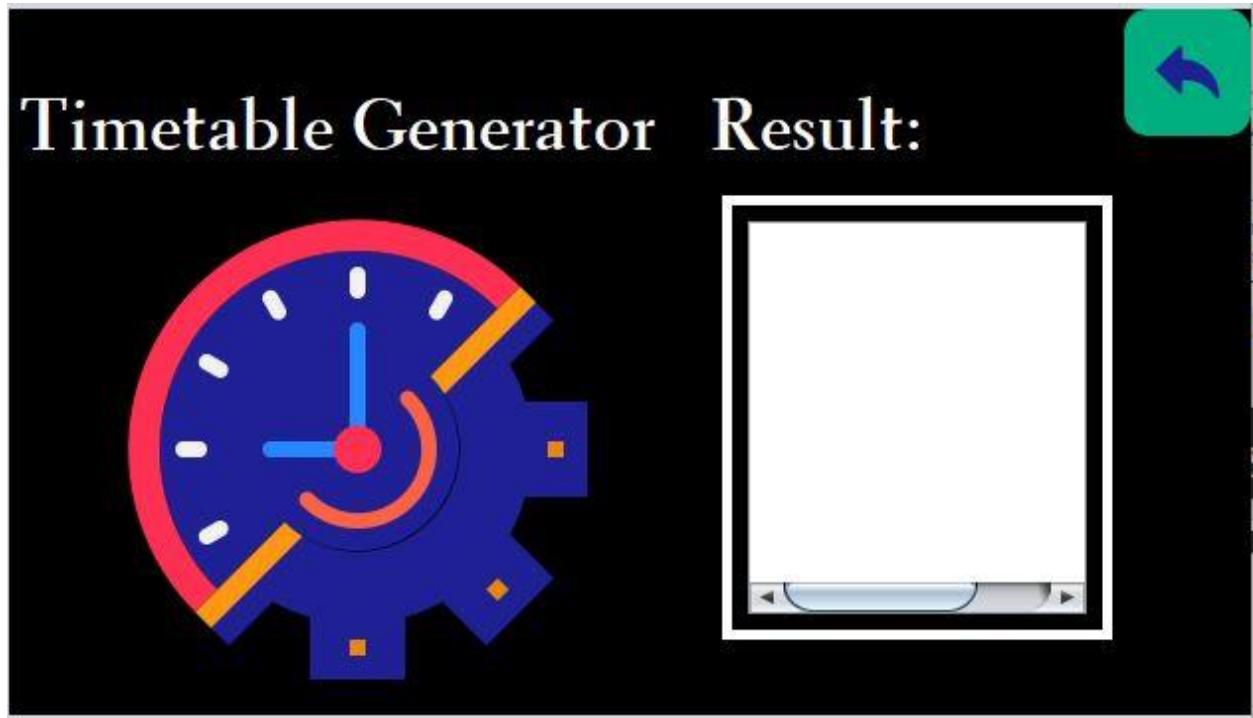
Screen of UC3: delete timetable
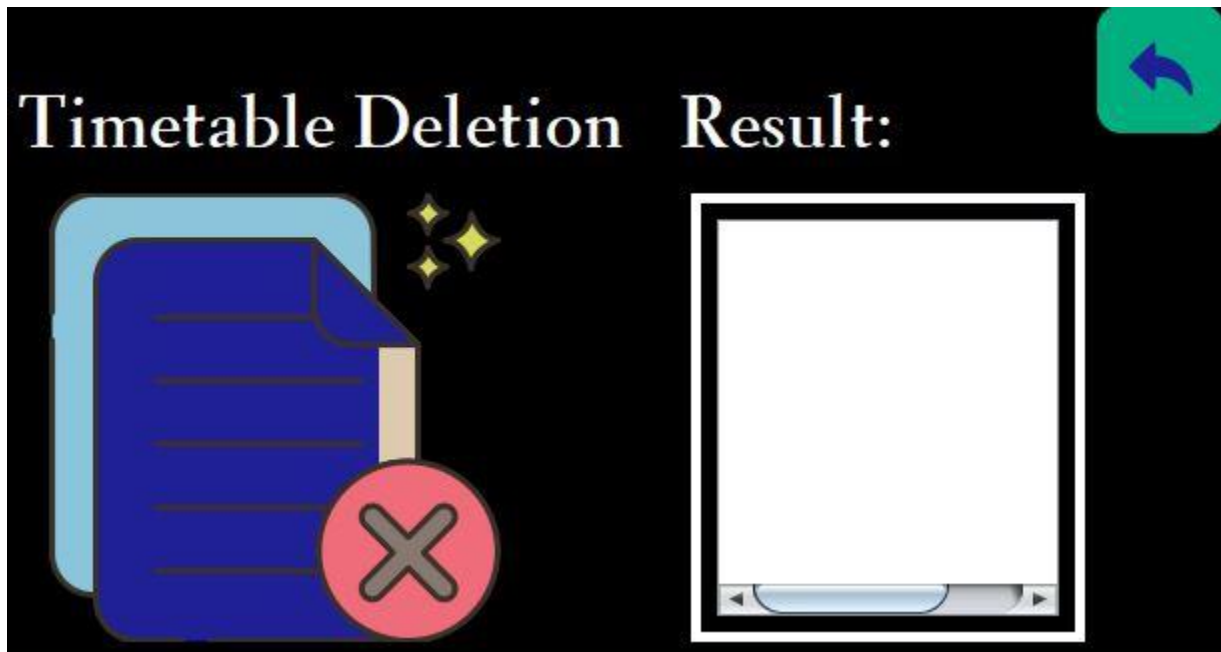
Screen of UC1: view timetable
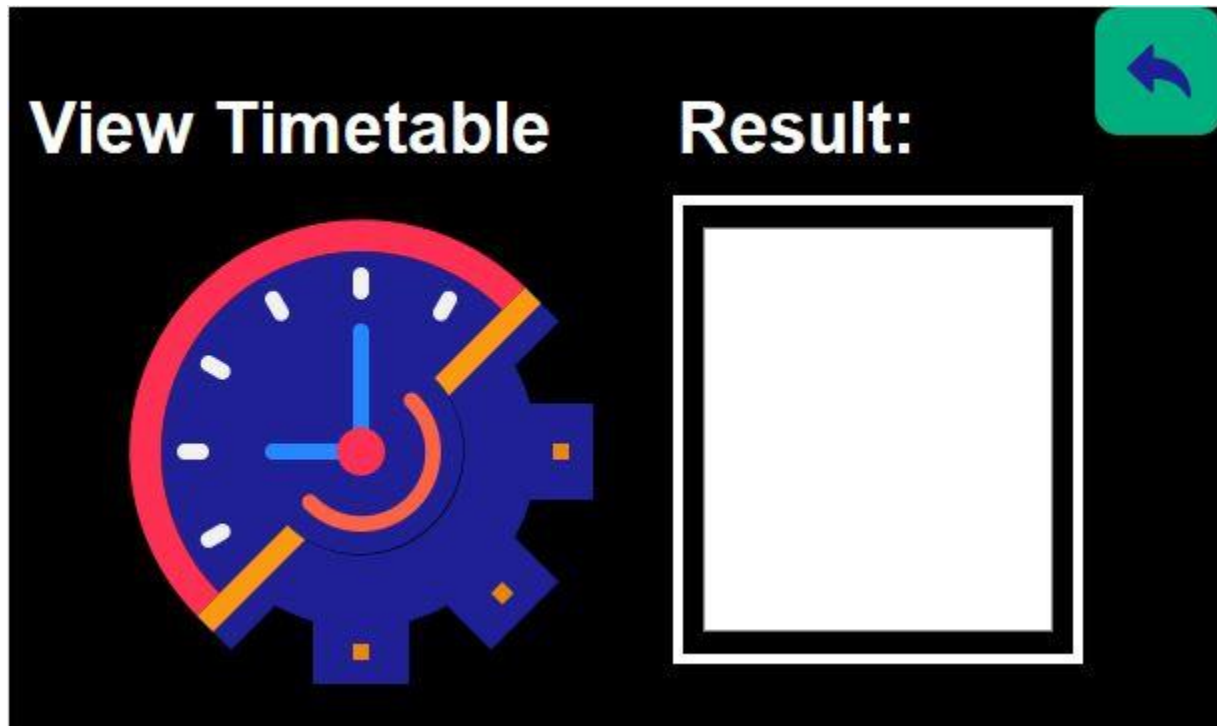
Screen of UC16: view timetable of department batch wise

Screen of UC17: view timetable course wise

Screen of UC18: view timetable instructor wise
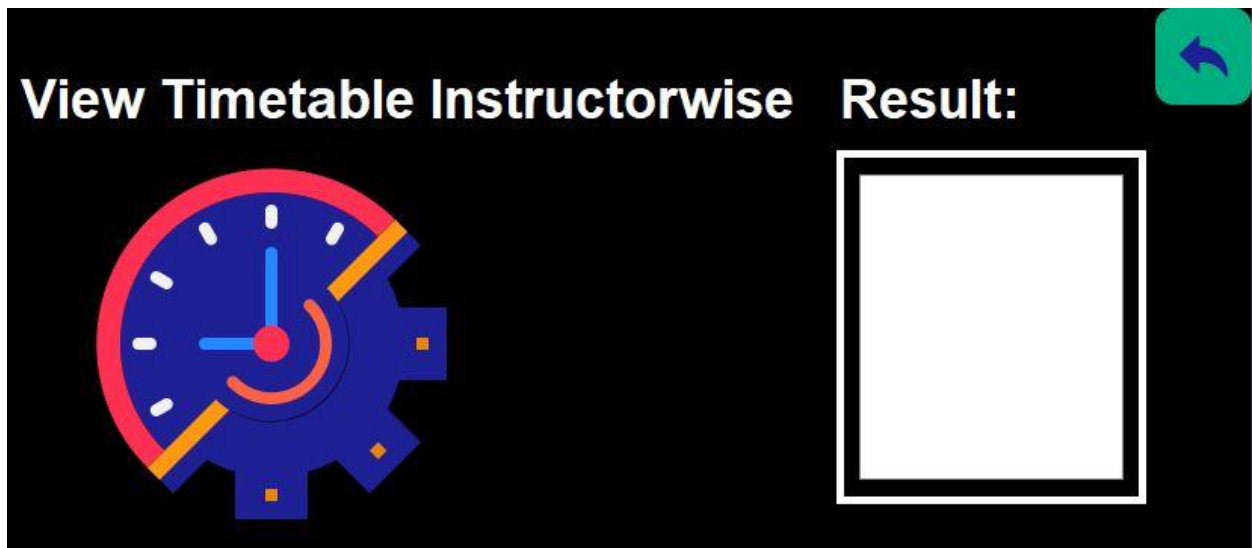
Screen of UC19: view timetable student wise

View Timetable Studentwise   Result:
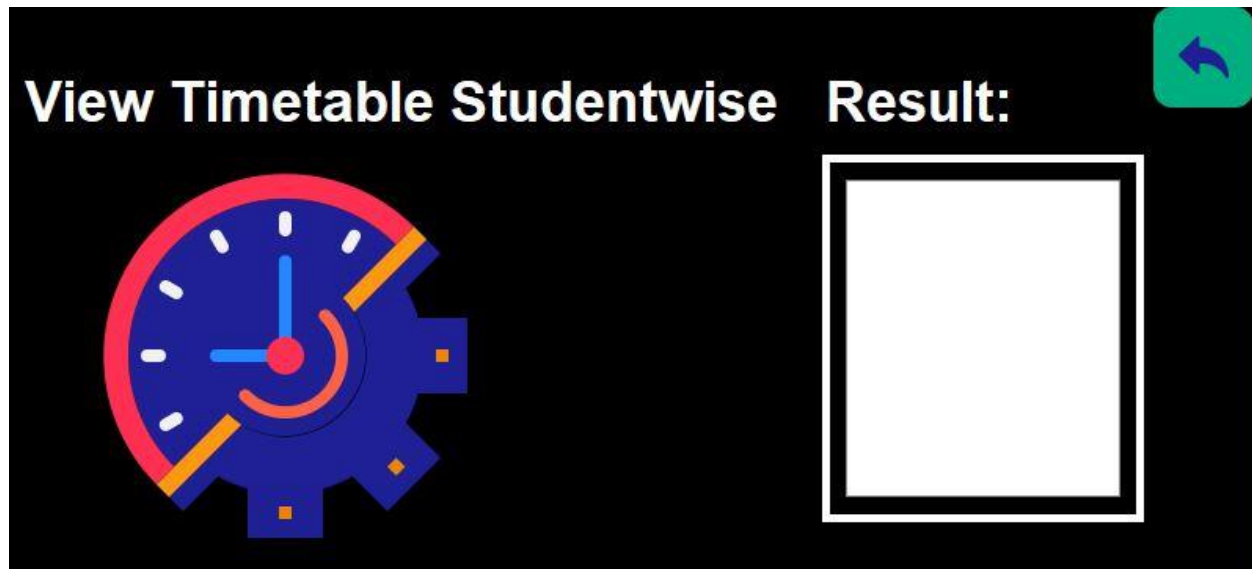
Update Timetable Page:

Screen of UC4: add slot
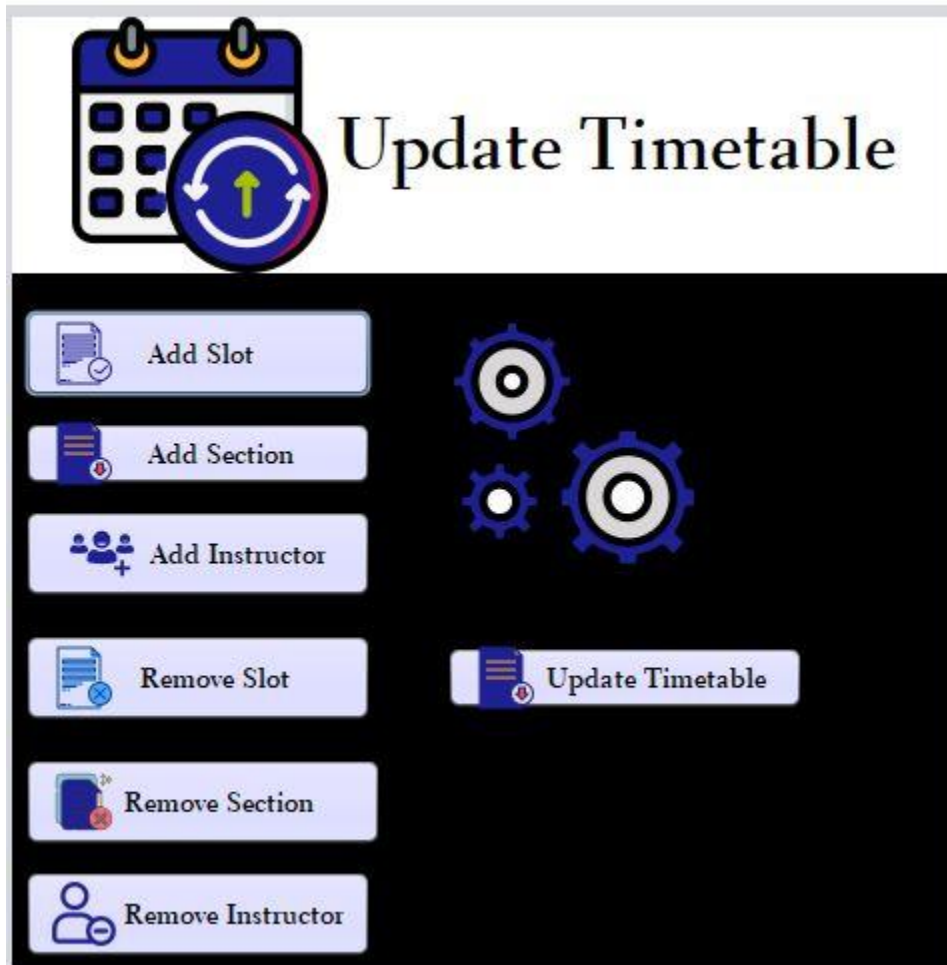
Screen of UC20: delete slot

Remove Slot:

Enter Start Time:

Enter End Time:

Enter Date:

Enter Duration:

Enter Venue:

Result:
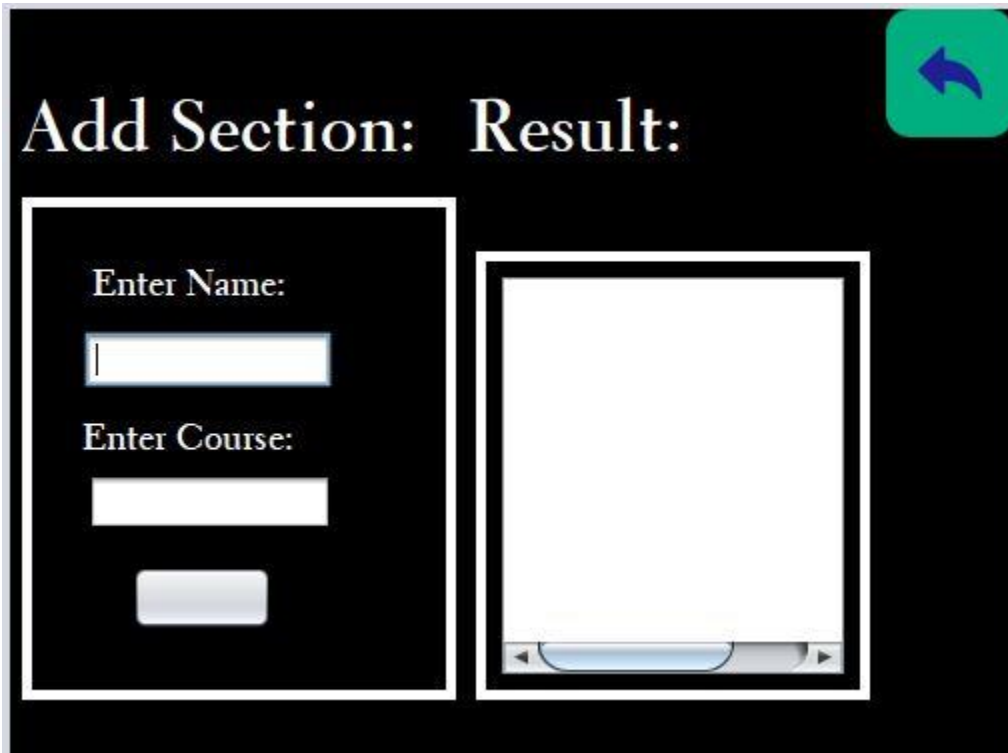
Screen of UC5: add section

Screen of UC6: delete section

Screen of UC15: add instructor

Screen of UC14: delete instructor

Remove Instructor: Result:

Instructor Id:

Course:

Updated Timetable Screen:
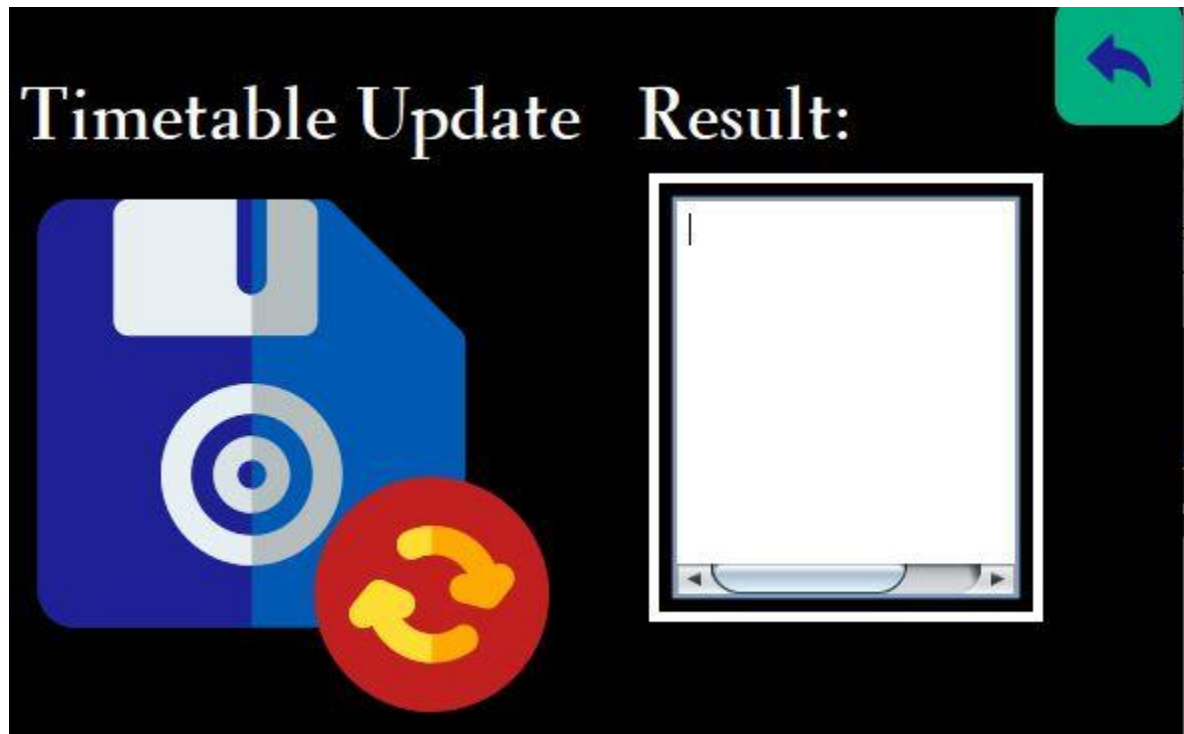
Screen of UC10: add course

Screen of UC11: delete course

Screen of UC12: add venue

Screen of UC13: delete venue

Screen of UC8: logout