### **Usama Ijaz (FA21-BSE-158)**

#### **Use Case UC 1: Create Timetable**

**Scope:** Timetable Management System

**Level:** User goal

**Primary Actor**: Admin

**Stakeholders and Interests:**

• Admin: Wants to create a clear and conflict-free timetable efficiently.

**Preconditions:**

The Admin is authenticated and authorized to create a new timetable.

**Success Guarantee (or Postconditions):**

The new timetable is created and published to all users that can have access without any conflicts.

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

1. The Admin navigates to the "Create Timetable" feature in the system.
2. A createTimetable.fxml has been initialized.
3. The system shows the form to select the following fields (year/session, department, section, Slot).
4. The above data in parameters are saved in controller
5. The slots are populated by retrieving data from database.
6. The timetable has been generated randomly using given constraints.
7. In the grid, classes are populated.

**Extensions (or Alternative Flows):**

The system does not create a new timetable until the entered details are validated and corrected.

**Special Requirements:**

• The system should be able to validate the entered details accurately.

• The system should be able to handle errors and exceptions gracefully.

**Technology and Data Variations List:**

• The form for entering timetable details may vary based on the system design and user interface.

• The validation and conflict-checking algorithms may vary based on the system design and implementation.

• The publishing system may vary based on the system design and implementation.

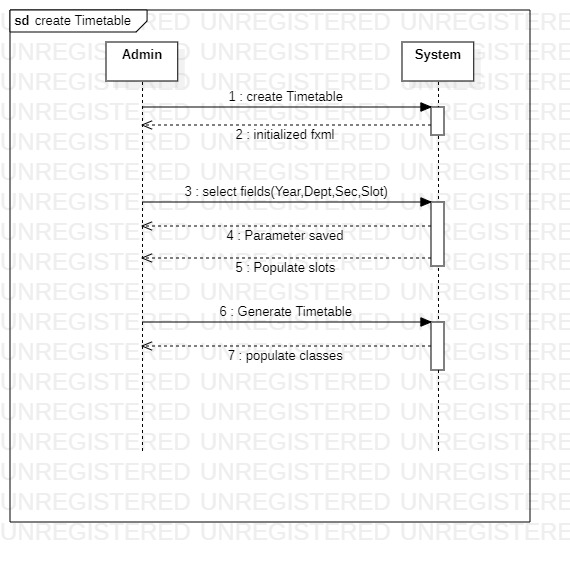
Frequency of Occurrence: The creation of new timetables could happen multiple times a year, depending on the frequency of slots.

**Open Issues:**

• The system should be able to handle multiple users creating new timetables simultaneously without conflicts.

• The system should be able to integrate with other modules of the timetable management system for seamless updates.

**SSD:**

****

**Operation Contract:**

|  |  |
| --- | --- |
| Operation | createTimetable() |
| Cross References | Create timetable |
| Preconditions | * The system is initialized and running. * The user has administrative privileges. * There are available slots, teachers, room and courses in the database. |
| Postconditions | * A new timetable is created with assigned slots, teachers, room and courses. * The timetable reflects the scheduling of classes for the specified slots, teachers, room and courses. |

### **Saud Khan (FA21-BSE-033)**

#### **Use Case UC 7: Create Slot**

**Scope:** Timetable Management System

**Level:** User goal

**Primary Actor:** Admin

**Stakeholders and Interests:**

* Admin: Wants to create a new time slot accurately and efficiently without any errors.

**Preconditions:** The Admin is authenticated and authorized to create a new slot.

**Success Guarantee (or Postconditions):**

* The new time slot is created and added to the timetable without any conflicts by entering its type and interval of slot.

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

1. The Admin navigates to the "Create Slot" feature in the system.
2. The system presents a form for entering the slot details, type of slot and interval.
3. The Admin enters the required details in the form.
4. The system adds the slot to the database.

**Extensions (or Alternative Flows):**

If the Admin enters incorrect or invalid details in the form:

1. The system displays an error message to the Admin, prompting them to correct the errors and re-submit the form.
2. The system does not create a new slot until the entered details are validated and corrected.

**Special Requirements:**

* The system should be able to validate the entered details accurately.
* The system should be able to provide notifications to the Admin upon successful creation of a new slot.

**Technology and Data Variations List:**

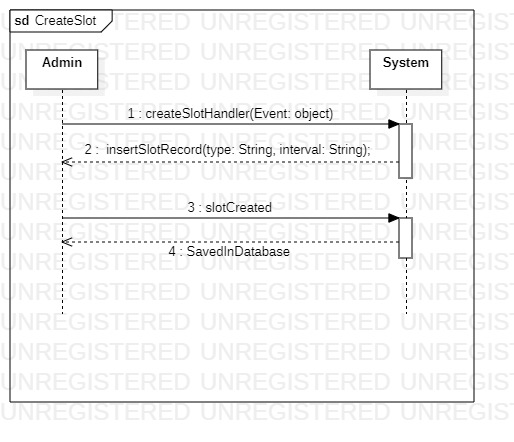
* The form for entering slot details may vary based on the system design and user interface.
* The validation and conflict-checking algorithms may vary based on the system design and implementation.
* The notification system may vary based on the system design and implementation.

Frequency of Occurrence: The creation of new slots could happen multiple times in semester, depending on the frequency of types of slot and timetable updates.

**Open Issues:**

* The system should be able to integrate with other modules of the timetable management system for seamless updates.

**SSD:**

****

**Operation Contract:**

|  |  |
| --- | --- |
| Operation | createSlot() |
| Cross References | Create Slot |
| Preconditions | * The system is initialized and running. * The user has administrative privileges. * The slot table should be exist in the database |
| Postconditions | * A new Slot object is created with the specified interval * The Slot object is forwarded to the createTimetable use case. * If the createTimetable use case is successful, the Slot object is added to the timetable. |

### **Zakariya Anwar Khan (FA21-BSE-042)**

#### **Use Case UC 13: Login**

**Scope:**

Timetable Management System

**Level:**

User goal

**Primary Actor:**

User

**Stakeholders and Interest:**

* **Admin:** The admin can login to system to create timetable and slots.

**Preconditions:**

* The Timetable Management System is installed and running.
* The admin has valid email and password

**Success Guarantee:**

* The user is authenticated and granted access to the system.

**Main Success Scenario:**

* The admin opens the login page of the Timetable Management System
* The system presents a form with fields for the email and password.
* The admin enters their email and password.
* The system verifies the entered credentials with the stored user data.
* If the credentials are correct, the system grants the admin access to the system and redirects them to the home page.

**Extensions:**

* If the admin enters an incorrect email or password, the system displays an error message and asks the user to re-enter their credentials.

**Special Requirements:**

* The system should use encryption to securely store user credentials.
* The system should have a password recovery mechanism in case the user forgets their password.

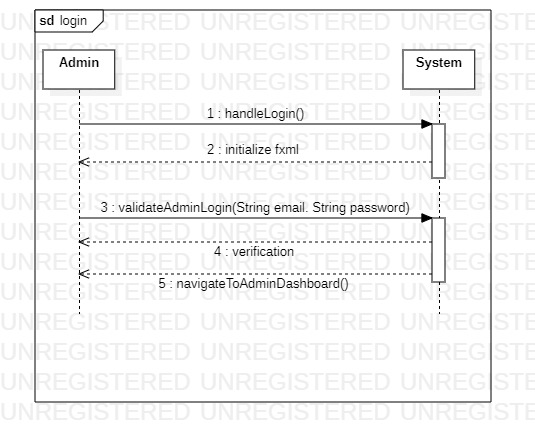
**Technology and Data Variation List:**

* The system may use different databases to store user data.

**Open Issues:**

* The Login use case is executed every time a user wants to access the Timetable Management System.

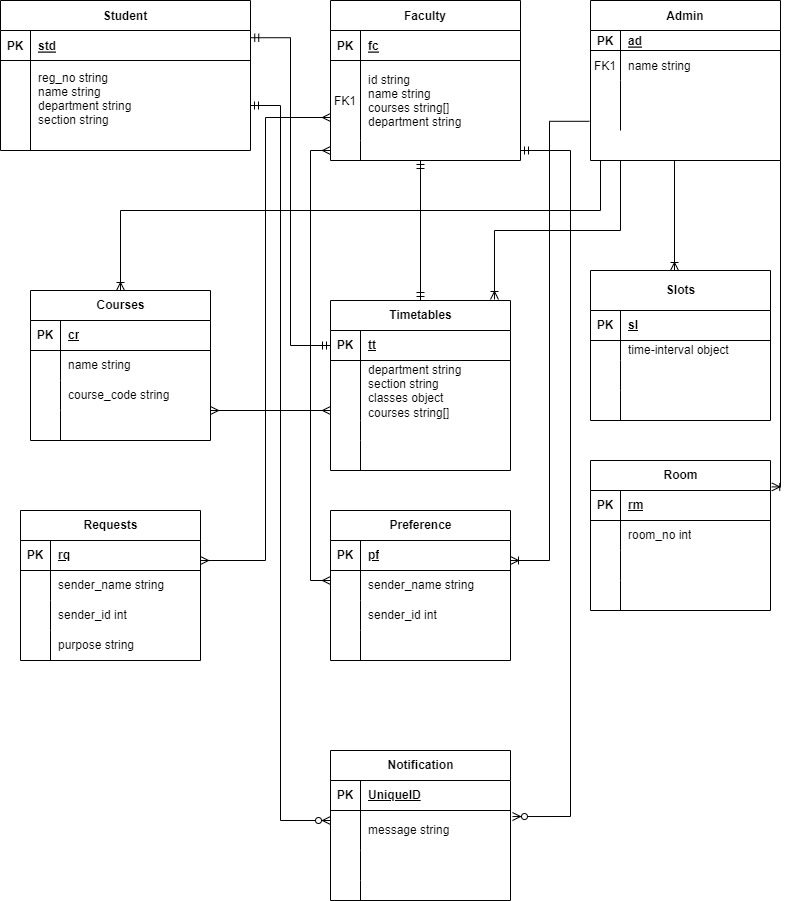
**SSD:**

****

**Operation Contract:**

|  |  |
| --- | --- |
| Operation | login() |
| Cross References | Login |
| Preconditions | * The system is initialized and running. * The admin must be registered. * The admin has valid email and password |
| Postconditions | * The admin is able to create slot and create timetable. * The timetable is displayed with all the relevant slots, teachers, room and courses. |

**ERD:**

****