**Software Design Specification (SDS)**

**Habib Unified Portal**

**Version**: 1.0  
**Project Team**:  
Taha Hunaid Ali  
Shavez Niyazi  
**Submission Date**: 25th April 2025

**Document History**

| **Version** | **Name of Person** | **Date** | **Description of Change** |
| --- | --- | --- | --- |
| 1.00 | Taha Hunaid Ali | 20/3/25 | Initial version of the SDS |
| 1.01 | Taha Hunaid Ali | 21/4/25 | Added system architecture and design considerations |
| 1.02 | Taha Hunaid Ali | 23/4/25 | Final version with all design diagrams and details |

**Distribution List**

| **Name** | **Role** |
| --- | --- |
| Ateeb Ahmed | Course Instructor |

**Document Information**

| **Category** | **Information** |
| --- | --- |
| Customer | Habib University |
| Project | Habib Unified Portal |
| Document | Software Design Specification |
| Document Version | 1.0 |
| Identifier | HUP-2025 |
| Status | Draft |
| Author(s) | Taha Hunaid Ali, Shavez Niyazi |
| Approver(s) | Ateeb Ahmed |
| Issue Date | 20/2/25 |

**Table of Contents**

1. **Introduction**  
   1.1. Purpose of Document  
   1.2. Intended Audience  
   1.3. Document Convention  
   1.4. Project Overview  
   1.5. Scope
2. **Design Considerations**  
   2.1. Assumptions and Dependencies  
   2.2. Risks and Volatile Areas
3. **System Architecture**  
   3.1. System Level Architecture  
   3.2. Software Architecture
4. **Design Strategy**
5. **Detailed System Design**  
   5.1. Database Design  
   5.1.1. ER Diagram  
   5.1.2. Data Dictionary  
   5.2. Application Design  
   5.2.1. Class Diagram  
   5.2.2. Sequence Diagram  
   5.2.3. State Diagrams  
   5.2.4. Activity Diagrams
6. **References**
7. **Appendices**

**1. Introduction**

**1.1 Purpose of Document**

This document provides the **design specifications** for the **Habib Unified Portal**, developed using **Flask** as the web framework and **SQLite** for database management. It includes the system’s architecture, design strategies, database design, and other design details for the development team and stakeholders.

**1.2 Intended Audience**

This document is intended for:

* **Project Team**: Developers and architects who will implement and maintain the system.
* **University IT Department**: To understand the technical aspects and assist in future modifications.
* **External Supervisor**: To review the design and provide feedback for improvements.

**1.3 Document Convention**

* **Font**: Calibri, size 11
* **Spacing**: 1.15 line spacing
* **Sections**: Bold section titles and numbered subsections.

**1.4 Project Overview**

The **Habib Unified Portal** is designed to streamline **academic resources**, **appointment scheduling**, **event management**, and **communication** at **Habib University**. It provides a **secure**, **efficient**, and **accessible** platform for students and faculty to manage their academic schedules, events, and appointments.

**1.5 Scope**

* **Academic Support**: Providing course materials, academic discussions, and resource repositories.
* **Faculty Interaction**: Scheduling appointments with faculty and providing a platform for communication.
* **Event Management**: A centralized platform for viewing and registering for academic and extracurricular events.
* **Notifications**: Automatic reminders and alerts for events, appointments, and updates.

**2. Design Considerations**

**2.1 Assumptions and Dependencies**

* The system depends on the university's **local network** and **authentication system** for user management.
* Clients must be using modern web browsers to access the portal.

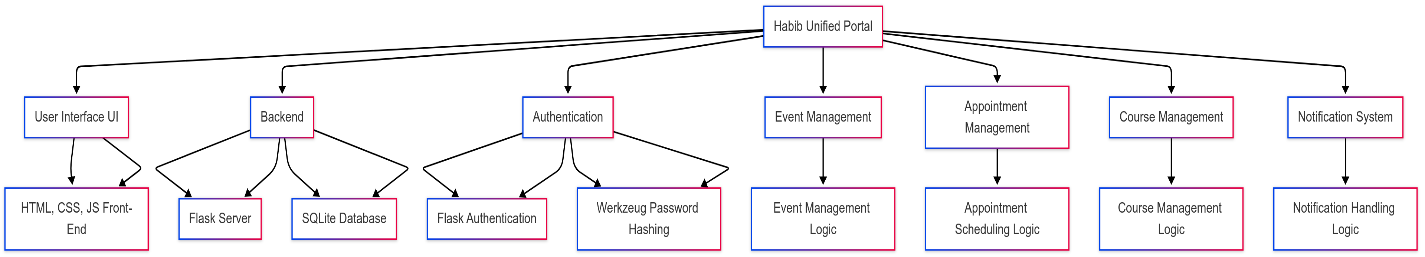
**2.2 Risks and Volatile Areas**

* **User Data Security**: As the portal handles sensitive data, encryption and secure protocols must be implemented to prevent breaches.
* **Server Downtime**: The portal must be highly available, especially during critical times such as exams and event registration periods.

**3. System Architecture**

**3.1 System Level Architecture**

The system will follow a **client-server model**, where the front-end (client) communicates with the back-end (server). The server (running **Flask**) will manage data processing and serve the application, while the client will handle the user interface.



**3.2 Software Architecture**

The **Habib Unified Portal** will have the following major components:

* **User Interface (UI)**: A **web-based** interface created using **HTML**, **CSS**, and **JavaScript**.
* **Backend**: The server-side application will be powered by **Flask**, with **SQLite** as the database for managing user data, events, and appointments.
* **Authentication**: The portal handles user registration and login via **Flask-based authentication**, using **password hashing** via **Werkzeug**.

**4. Design Strategy**

The system will be built following the **Incremental Development Model**, allowing continuous feedback and adjustments after each phase. This strategy ensures that the client is involved throughout the development process, improving transparency and agility.

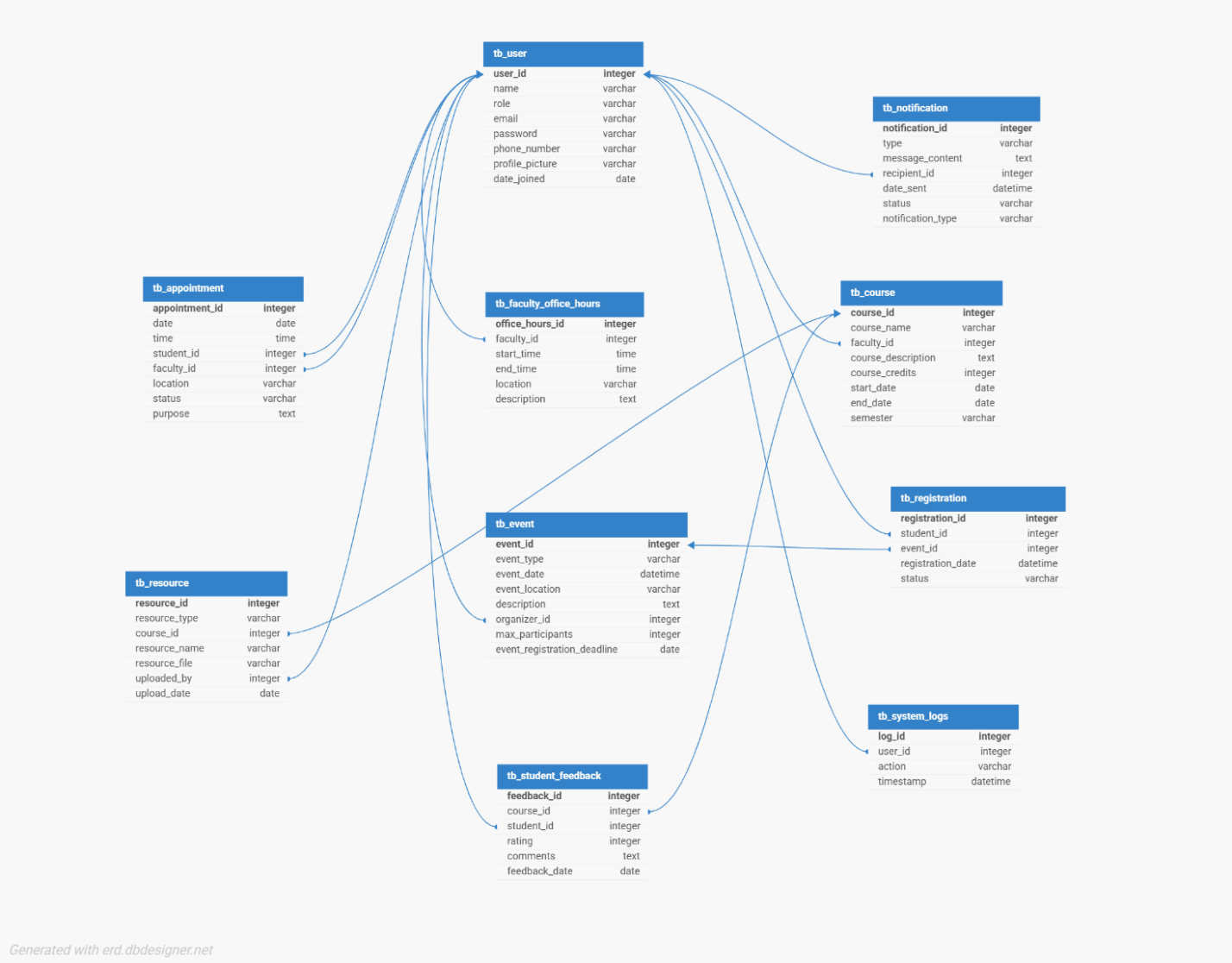
**5. Detailed System Design**

**5.1 Database Design**

**5.1.1 ER Diagram**

The **Entity-Relationship Diagram (ERD)** illustrates the relationships between **Users**, **Appointments**, **Events**, **Courses**, and **Notifications**. The relationships include:

* **Users to Appointments**: One-to-many (A user can have multiple appointments).
* **Users to Events**: Many-to-many (A user can register for multiple events, and an event can have multiple participants).
* **Users to Notifications**: One-to-many (A user can receive multiple notifications).
* **Faculty to Courses**: One-to-many (A faculty member can teach multiple courses).



**5.1.2 Data Dictionary**

The Data Dictionary defines key entities and their attributes:

**User**

* UserID: Unique identifier for the user (Primary Key).
* Name: Full name of the user.
* Role: Role of the user (e.g., Student, Faculty, Administrator).
* Email: User’s email address.
* Password: Encrypted password for authentication.
* PhoneNumber: User’s contact number (Optional).
* ProfilePicture: Link to user’s profile picture (Optional).
* DateJoined: Date when the user registered.

**Appointment**

* AppointmentID: Unique identifier for the appointment.
* Date: Date of the appointment.
* Time: Time of the appointment.
* StudentID: Reference to the student who booked the appointment.
* FacultyID: Reference to the faculty member for the appointment.
* Location: Location of the appointment.
* Status: Status of the appointment (Scheduled, Completed, Cancelled).

**Event**

* EventID: Unique identifier for the event.
* EventType: Type of event (e.g., Academic, Co-Curricular).
* EventDate: Date and time of the event.
* EventLocation: Event location.
* Description: Event details.
* OrganizerID: Reference to the event organizer.

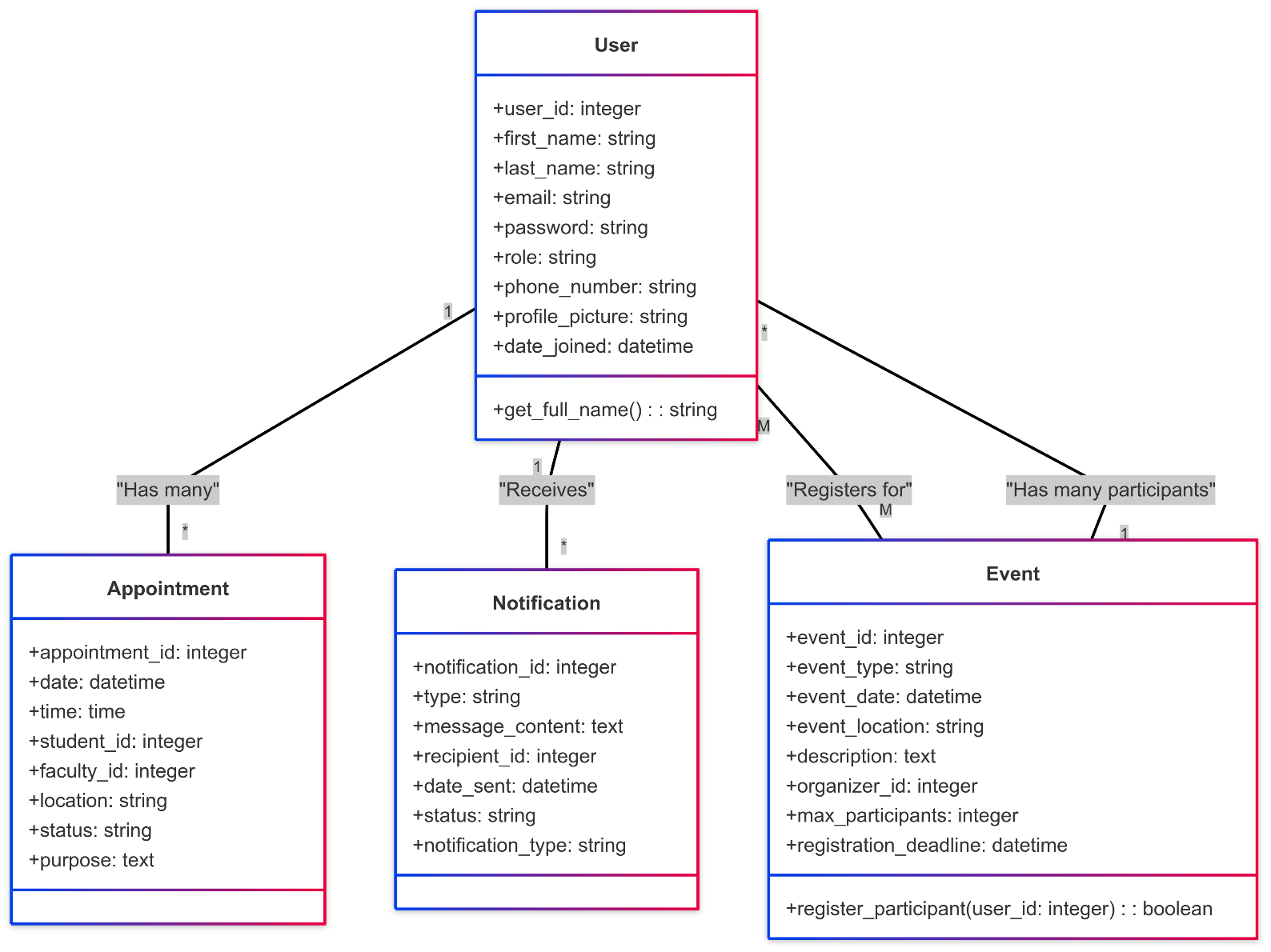
**Notification**

* NotificationID: Unique identifier for the notification.
* Type: Type of notification (e.g., Appointment Reminder).
* MessageContent: Content of the notification.
* RecipientID: Reference to the user receiving the notification.
* DateSent: Date and time when the notification was sent.

**5.2 Application Design**

**5.2.1 Class Diagram**

The **class diagram** includes major classes such as **User**, **Appointment**, **Event**, and **Notification**. Each class has methods and attributes for managing users, events, appointments, and notifications within the portal.



### ****5.2.2 Sequence Diagram****

The **Sequence Diagrams** illustrate the interactions between users and the system, depicting the flow of events for various actions. Below are the primary sequence diagrams:

This sequence diagram represents the interactions between the **Student** and the **Habib Unified Portal** for the following actions:

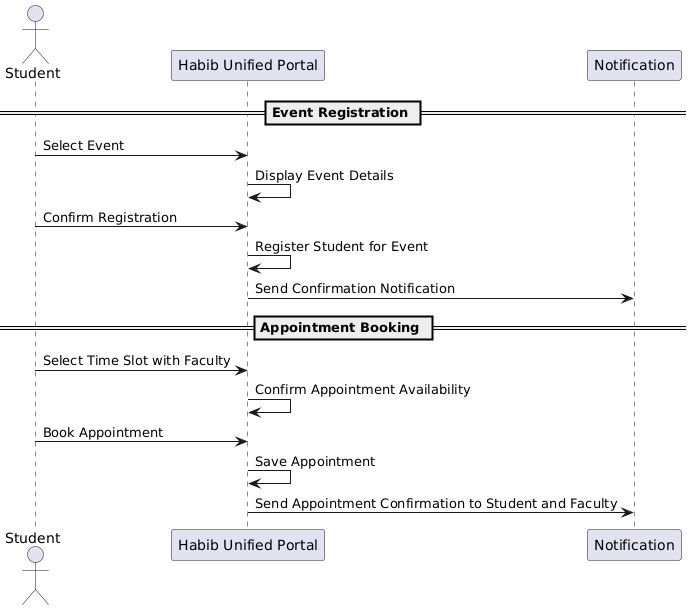
**Sequence Diagram 1 – Client Actor (Student)**:

**Event Registration:**

1. The **Student** selects an **event** from the portal.
2. The **System** displays the **event details**.
3. The **Student** confirms registration.
4. The **System** registers the student for the event and sends a **confirmation notification**.

**Booking an Appointment:**

1. The **Student** selects an available **time slot** with a **faculty member**.
2. The **System** confirms appointment availability.
3. The **Student** books the appointment.
4. The **System** sends a **confirmation notification** to both the **Student** and the **Faculty**



**Sequence Diagram 2 – Admin Actor**

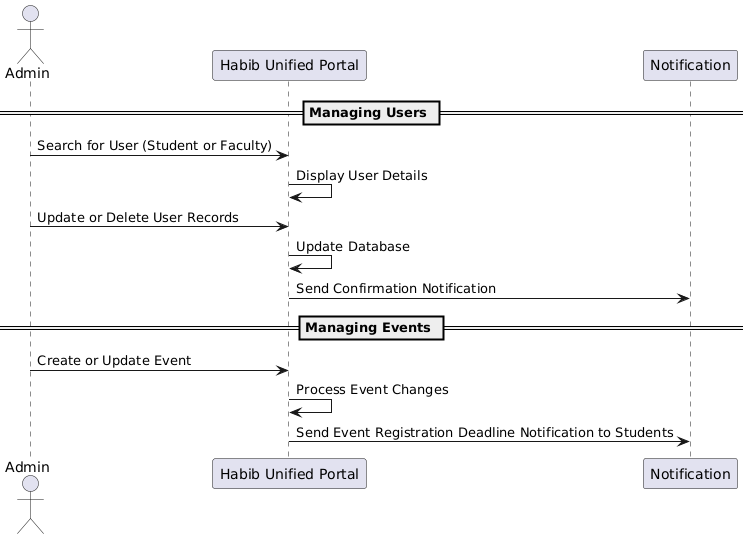
This sequence diagram depicts the interactions between the **Administrator** and the **Habib Unified Portal** for:

**Managing Users:**

1. The **Admin** searches for a **user** (e.g., student or faculty).
2. The **System** displays the **user details**.
3. The **Admin** can **update or delete** user records.
4. The **System** updates the database and sends a **confirmation notification**.

**Managing Events:**

1. The **Admin** creates a **new event** or **updates** an existing event.
2. The **System** processes the changes and updates the event details.
3. The **System** sends notifications to students about **event registration deadlines** and updates.



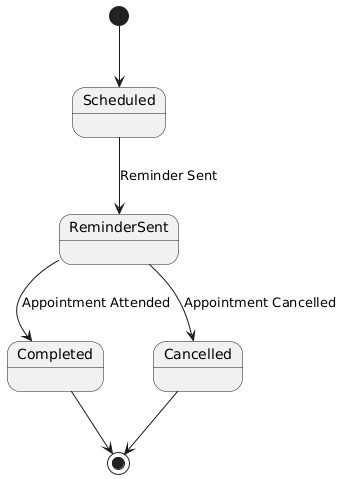
**5.2.3 State Diagrams**

**State Diagram - Appointment Lifecycle**

The state diagram will track the lifecycle of an appointment in the Habib Unified Portal, from creation to completion or cancellation.

**States:**

1. Scheduled: When the appointment is initially booked.
2. Reminder Sent: When the system sends reminders to the student and faculty.
3. Completed: When the appointment is attended by the student and faculty.
4. Cancelled: If the appointment is cancelled by the student or faculty.

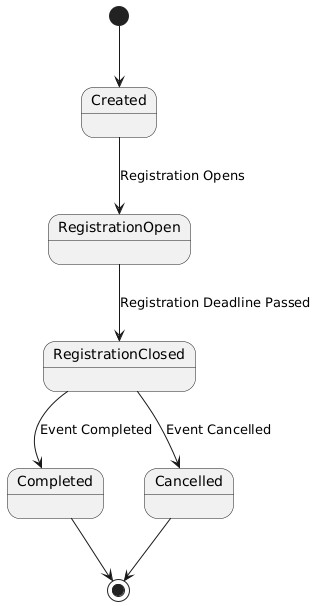


**State Diagram - Event Lifecycle**

This diagram will track the lifecycle of an event, showing the state transitions from creation to closure or cancellation.

**States:**

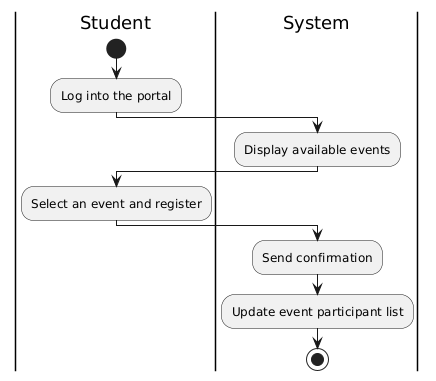
1. **Created**: When the event is created by the admin.
2. **Registration Open**: When registration for the event is open.
3. **Registration Closed**: After the registration deadline passes.
4. **Completed**: When the event occurs.
5. **Cancelled**: If the event is cancelled by the organizer.



**5.2.4 Activity Diagrams**

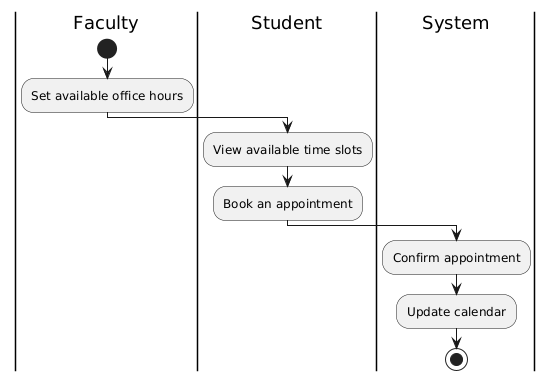
**Student Event Registration:**

* The student logs into the portal.
* The system displays available events.
* The student selects an event and registers.
* The system sends a confirmation and updates the event’s participant list.



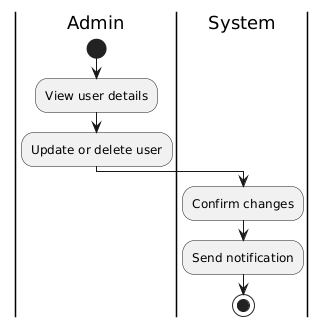
**Faculty Appointment Scheduling:**

* The faculty member sets available office hours.
* The student views available time slots.
* The student books an appointment.
* The system confirms the appointment and updates the calendar.



**Admin User Management:**

* The admin views user details.
* The admin updates or deletes a user.
* The system confirms changes and sends notifications.



**6. References**

* **Flask Documentation**: Official documentation for Flask.
* **SQLAlchemy Documentation**: Official documentation for SQLAlchemy.
* **Werkzeug Documentation**: Official documentation for password hashing.

**7. Appendices**

* **Appendix A**: ERD and Data Models.
* **Appendix B**: Class and Sequence Diagrams.
* **Appendix C**: Detailed User Interface Mockups.