

**COE4112505 - Software Engineering**

**Project Part 3**

**Software Design Specification**

****

**Name of The Team:** Team 6

**Name of All Team Members:** Hilal Türkyılmaz (9A190001), Beyza Şahin (63200033), Esmanur Aytaş (63200005)

**Date:** 18.12.2024

**Advisor:** Muhsin Zahid Uğur

Contents

[**2. Abstract: 3**](#_Toc185285298)

[**3. Introduction 3**](#_Toc185285299)

[**4. High-level and Medium-level Design 4**](#_Toc185285300)

[**4.1 Context Model and Architectural System Diagram 4**](#_Toc185285301)

[**4.1.1 GUI (Graphical User Interface) 5**](#_Toc185285303)

[**4.1.2 UniClub System 5**](#_Toc185285304)

[**4.1.3 Reservation System 6**](#_Toc185285305)

[**4.1.4 Event Management System 6**](#_Toc185285306)

[**4.1.5 Club Management System 6**](#_Toc185285307)

[**4.1.6 User Management System 6**](#_Toc185285308)

[**4.2 Class Diagram 8**](#_Toc185285309)

[**4.3 Class Descriptions 9**](#_Toc185285310)

[**5. Detailed Design 12**](#_Toc185285311)

[**6. Updated Snapshots of the System’s User Interface 18**](#_Toc185285312)

[**7. Glossary of Terms 23**](#_Toc185285313)

# Abstract:

This project aims to develop a University Clubs Website designed to address low student participation in campus activities by offering an intuitive platform for event discovery and booking. The platform will allow students to view, filter, and register for events while enabling club leaders to manage events and interact with participants. The system integrates a frontend interface (HTML, CSS, JavaScript), a Python-based backend, and a MySQL database to ensure efficient event management and seamless user experiences. It focuses on usability, scalability, and compliance with privacy regulations, supporting a vibrant campus community. Key functionalities include user registration, event management, booking, and administrative oversight, all tailored to the unique needs of Istanbul Medipol University. The website's modular design ensures adaptability and scalability for future enhancements, such as multilingual support and advanced analytics.

# **Introduction**

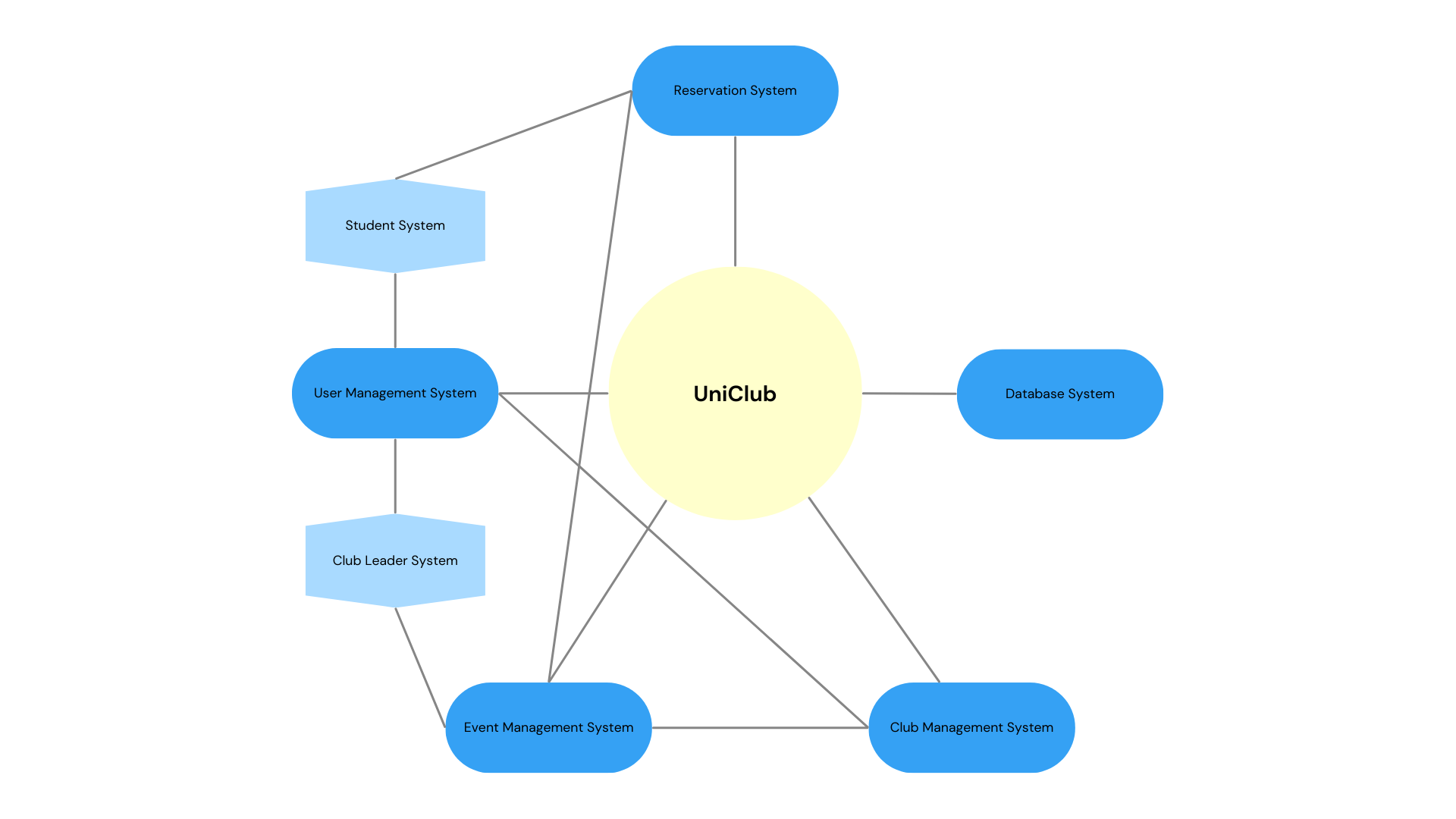
The University Clubs Website Project aims to create a platform where students can discover and participate in events organized by university clubs. The website allows users to search for events, register for them, and manage their bookings. Club leaders can create, update, and manage events, while administrators ensure proper approval and compliance with university policies. This phase of the project builds upon the foundational features implemented in the previous phase, enhancing the overall functionality and design.

Key updates in this phase include:

* Security Features: While not previously focused on, security mechanisms like data encryption and user authentication (e.g., two-factor authentication for administrators) have been integrated to ensure the integrity and confidentiality of user data.
* Context Model and Architectural System Diagram: A detailed model of the system's architecture has been developed to visualize the interaction between different components.
* Class Diagram and Descriptions: The system's core classes and their relationships have been defined to streamline development and ensure maintainability.
* Detailed Design: Several key design diagrams have been added:
  + An event creation activity diagram.
  + An event booking activity diagram.
  + An user registration sequence diagram.
  + An event creation sequence diagram.
  + An event booking sequence diagram.
  + A behavioral state machine diagram.
* UI Updates: The user interface has been refined with updated snapshots showcasing the latest design improvements, making it more user-friendly and aligned with accessibility standards.

# High-level and Medium-level Design

## 4.1 Context Model and Architectural System Diagram



**Figure 1:** Context Model.

## A diagram of a company Description automatically generated

**Figure 2:** Architectural System Diagram.

The UniClub System is a comprehensive system designed to manage various aspects of university clubs, including user management, event management, reservations, and club organization. The system is structured as follows:

### 4.1.1 GUI (Graphical User Interface)

The GUI serves as the primary interface for users to interact with the UniClub System. It connects all the subsystems and provides a user-friendly interface for navigating through various functionalities. The GUI allows users to access different features like reservation management, event scheduling, and club-related activities, offering an intuitive way to manage the entire system.

### 4.1.2 UniClub System

The UniClub System is the core of the architecture and acts as the central hub for managing and integrating all other subsystems. It is responsible for overseeing the entire process of club-related activities, such as event management, user management, and reservations.

Subsystems of UniClub System:

The UniClub System includes the following key subsystems:

* Reservation System
* Event Management System
* Club Management System
* User Management System

### 4.1.3 Reservation System

The Reservation System allows students and club members to make and manage reservations for club events and activities. This subsystem tracks reservation details such as date, time, and status, ensuring that resources (such as space or event slots) are efficiently allocated. It is crucial for managing bookings and availability in club events.

### 4.1.4 Event Management System

The Event Management System oversees the planning and organization of events within the university clubs. This subsystem enables the scheduling of events, coordination with other systems (such as the Club Management System and Reservation System), and management of event-related details, such as capacity, location, and description. It ensures that events are well-planned and executed.

### 4.1.5 Club Management System

The Club Management System is responsible for organizing and managing the structure of different university clubs. This subsystem handles the creation and management of club profiles, the addition of members, and the management of club-specific events. It integrates with the Event Management System to ensure that each club can effectively run and participate in events.

### 4.1.6 User Management System

The User Management System is responsible for handling user-related functionalities, including registration, authentication, and role-based management. This subsystem ensures that users are assigned appropriate roles and can access only the features relevant to their user type.

Subcomponents of User Management System:

* Student System:

The Student System manages the registration and activities of students within the UniClub System. It allows students to make reservations, join clubs, and participate in events. This system ensures that students can engage with the various activities available within the system.

* Club Leader System:

The Club Leader System manages the administrative roles of club leaders within the UniClub System. Club leaders can organize events, manage members, and ensure smooth coordination within the clubs. This system offers features for adding or removing users from clubs, organizing events, and overseeing club activities.

## 4.2 Class Diagram

A diagram of a computer program

Description automatically generated

**Figure 3:** UniClub Class Digram

## 4.3 Class Descriptions

**Register**

| **Attribute/Method** | **Visibility** | **Return Type** | **Description** |
| --- | --- | --- | --- |
| userID | private | int | Unique identifier for the new user. |
| userType | private | string | Type of user being created. |
| userName | private | string | First name of the new user. |
| userSurname | private | string | Last name of the new user. |
| email | private | string | Email for the new account. |
| password | private | string | Password for securing the account. |
| phoneNumber | private | int | Contact phone number for the new user. |
| createUser() | public | boolean | Creates a new user account in the system. |

**Login**

| **Attribute/Method** | **Visibility** | **Return Type** | **Description** |
| --- | --- | --- | --- |
| userType | private | string | Type of user (Student or ClubLeader). |
| email | private | string | Email address for authentication. |
| password | private | string | Password for authentication. |
| authenticateUser() | public | boolean | Verifies user credentials for login. |

**User**

| **Attribute/Method** | **Visibility** | **Return Type** | **Description** |
| --- | --- | --- | --- |
| userID | private | int | Unique identifier for the user. |
| userType | private | string | Specifies the type of user (Student or Club Leader). |
| userName | private | string | First name of the user. |
| userSurname | private | string | Last name of the user. |
| email | private | string | Email address for the user. |
| password | private | string | Password for account authentication. |
| phoneNumber | private | int | Contact phone number for the user. |
| register() | public | boolean | Registers a new user into the system. |
| login() | public | boolean | Authenticates user credentials for login. |

**Student (inherits from User)**

| **Attribute/Method** | **Visibility** | **Return Type** | **Description** |
| --- | --- | --- | --- |
| makeReservation() | public | boolean | Allows a student to make a reservation for an event. |
| cancelReservation() | public | boolean | Allows a student to cancel an existing reservation. |
| joinClub() | public | boolean | Enables a student to join a club. |
| removeClub() | public | boolean | Allows a student to leave a club. |

**ClubLeader (inherits from User)**

| **Attribute/Method** | **Visibility** | **Return Type** | **Description** |
| --- | --- | --- | --- |
| addEvent() | public | boolean | Allows a club leader to add a new event. |
| editEvent() | public | boolean | Enables a club leader to edit event details. |
| deleteEvent() | public | boolean | Removes an existing event from the system. |
| organizeClub() | public | boolean | Organizes and manages club details. |
| addUserClub() | public | boolean | Adds a user to a club. |
| removeUserClub() | public | boolean | Removes a user from a club. |

**Club**

| **Attribute/Method** | **Visibility** | **Return Type** | **Description** |
| --- | --- | --- | --- |
| userID | private | int | Identifier of the club leader. |
| userType | private | string | Type of user associated with the club. |
| userName | private | string | Name of the club leader. |
| userSurname | private | string | Surname of the club leader. |
| clubID | private | int | Unique identifier for the club. |
| clubName | private | string | Name of the club. |
| clubDescription | private | string | A brief description of the club's purpose. |
| organizeEvent() | public | boolean | Organizes events for the club. |
| viewEvents() | public | Event | Displays the list of events organized by the club. |

**Event**

| **Attribute/Method** | **Visibility** | **Return Type** | **Description** |
| --- | --- | --- | --- |
| userID | private | int | Identifier of the club leader who created the event. |
| userType | private | string | Type of user (Student or ClubLeader). |
| userName | private | string | Name of the event creator. |
| userSurname | private | string | Surname of the event creator. |
| reservationID | private | int | Identifier of a reservation associated with the event. |
| clubID | private | int | Identifier of the organizing club. |
| clubName | private | string | Name of the organizing club. |
| clubDescription | private | string | Description of the organizing club. |
| eventID | private | int | Unique identifier for the event. |
| eventName | private | string | Name of the event. |
| eventDescription | private | string | Brief description of the event. |
| eventDate | private | date | Date of the event. |
| eventTime | private | time | Time of the event. |
| eventLocation | private | string | Location where the event will take place. |
| eventCapacity | private | int | Total number of participants allowed. |
| eventStatus | private | string | Current status of the event (e.g., active, canceled). |
| updateDetails() | public | boolean | Updates the details of an existing event. |
| manageCapacity() | public | int | Manages and adjusts the event capacity. |
| checkAvailability() | public | boolean | Checks if there are available spots for the event. |

**Reservation**

| **Attribute/Method** | **Visibility** | **Return Type** | **Description** |
| --- | --- | --- | --- |
| userID | private | int | Identifier for the user making the reservation. |
| userType | private | string | Type of user making the reservation. |
| userName | private | string | First name of the user. |
| userSurname | private | string | Last name of the user. |
| reservationID | private | int | Unique identifier for the reservation. |
| reservationDate | private | date | Date of the reservation. |
| reservationStatus | private | string | Status of the reservation (e.g., confirmed, canceled). |
| getReservationStatus() | public | boolean | Returns the current status of the reservation. |

# Detailed Design

A diagram of a activity diagram

Description automatically generated

**Figure 4:** Event Creation Activity Diagram.

The Event Creation Activity Diagram outlines the process of a club leader creating a new event. The club leader logs in, navigates to the event management page, and selects the "create event" option. The system provides a form to fill in event details. Once the form is submitted, the system validates the information. If correct, the event is saved, and a page is created. If there’s an error, the system prompts corrections.

A diagram with blue squares and black text

Description automatically generated

**Figure 5:** Event Booking Activity Diagram.

The Event Booking Activity Diagram outlines the step-by-step process of the user viewing events and making a reservation. First, the user logs into the system. After logging in, the user views the available events. The user selects the event of interest and enters necessary details (such as the number of participants, date, etc.). The system checks the provided information and verifies the validity of the reservation. If the reservation is successful, the system saves it and sends a success message to the user. If any errors occur, the system displays an error message and directs the user to make corrections.

A diagram of a student

Description automatically generated

**Figure 6:** User Registration Sequence Diagram.

The user registration sequence diagram outlines the step-by-step interactions between the user and the system during the registration process. First, the user submits the registration form to the system. The system then queries the database to verify if the user is already registered. The database checks for an existing user and sends the result back to the system. If the user is not registered, the system sends the new user data to the database for saving. The database then confirms the successful registration to the system. Finally, the system notifies the user that their registration was successful, completing the process.

A diagram of a diagram

Description automatically generated

**Figure 7:** Event Creation Sequence Diagram.

The club leader logs in by submitting their email and password. After successful login, they navigate to the event management page and choose "Create Event." The system presents a form for event details, which the club leader fills out and submits. The system validates the information, ensuring all fields are correct. If there are errors, the system waits for the club leader to make corrections. Once the details are valid, the system saves the event to the database and generates a dedicated event page.

A diagram of a student

Description automatically generated

**Figure 8:** Event Booking Sequence Diagram.

This diagram illustrates the interactions between the system and the database when a student makes an event reservation. After the student logs in, they view the events, check the availability, and confirm the booking. The system communicates with the database to process the reservation and notifies the student upon successful completion.

A diagram of a company

Description automatically generated

**Figure 9:** Behavioral State Machine Diagram for University Club ManagementSystem.

The Behavioral State Machine Diagram for the University Club Management System models the lifecycle of an event, from creation to completion or cancellation. The event progresses through states such as Draft, Published, Registration Open, Registration Closed, Ongoing, Completed, and Canceled. Transitions between these states occur based on actions like saving the event, opening or closing registration, starting or ending the event, and canceling it. This diagram helps ensure logical state transitions, handles edge cases like cancellations, and supports a consistent event lifecycle. It aids developers and stakeholders in managing the system's behavior effectively.

# Updated Snapshots of the System’s User Interface

Figure 10 displays events with essential details like event title, club name, date, time, location, and a "Book Now" button for easy registration. It also features navigation options for searching, viewing clubs, bookings, and user profiles."Create Event," button enhances accessibility for club administrators.

**metin, ekran görüntüsü, web sitesi, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Figure 10:** Home page.

The Events page in the figure 11 provides a layout where students can discover upcoming events organized by university clubs. This section allows users to easily browse, filter, and engage with campus activities, promoting active student participation and community involvement.

metin, ekran görüntüsü, web sayfası, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Figure 11:** Events page.

Figure 12 allows users to access their personalized accounts on the UniClubs platform. By signing in, students can manage event bookings, view their profile, and explore club activities tailored to their interests.

metin, ekran görüntüsü, çizgi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Figure 12:** Sign in page.

The Log In page in the figure 13 provides users with secure access to their accounts on the UniClubs platform. It enables students to manage bookings, track upcoming events, and engage with university club activities.

metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Figure 13:** Log in page.

The Clubs section in the figure 14 showcases a list of all university clubs, allowing students to explore various organizations, learn about their activities, and join communities that match their interests.

metin, ekran görüntüsü, yazı tipi, doküman, belge içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Figure 14:** Clubs page.

# Glossary of Terms