

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING

STUDENT CLUBS MANAGEMENT SYSTEM

MUHAMMED OĞUZ
SÜLEYMAN GÖLBOL

SUPERVISOR
PROF. DR. HASARİ ÇELEBİ

GEBZE
2023

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

**STUDENT CLUBS MANAGEMENT
SYSTEM**

MUHAMMED OĞUZ
SÜLEYMAN GÖLBOL

SUPERVISOR
PROF. DR. HASARİ ÇELEBİ

2023
GEBZE

 <p>GEBZE TECHNICAL UNIVERSITY</p>	<p>GRADUATION PROJECT JURY APPROVAL FORM</p>
--	--

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 18/06/2023 by the following jury.

JURY

Member

(Supervisor) : PROF. DR. HASARİ ÇELEBİ

Member : PROF. DR. MEHMET GÖKTÜRK

	<p>GRADUATION PROJECT JURY APPROVAL FORM</p>
---	--

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 18/06/2023 by the following jury.

JURY

Member

(Supervisor) : PROF. DR. HASARİ ÇELEBİ

Member : PROF. DR. MEHMET GÖKTÜRK

Member : RESEARCH ASSISTANT EVREN ÇİFÇİ

ABSTRACT

The Student Clubs Management System is an intuitive web-based application developed to simplify the management of student clubs within an educational institution. The primary objective of this system is to provide a centralized platform that enables club administrators, advisors, and members to effortlessly organize, communicate, and collaborate on a wide range of club-related activities. With modules dedicated to announcements, club registration, event planning, and user management, the system offers a comprehensive toolkit for efficient club administration. Through its user-friendly interface and seamless functionalities, the Student Clubs Management System aims to streamline club processes, enhance communication channels, facilitate event coordination, and cultivate a sense of community among club members. By empowering clubs with accessible tools and fostering meaningful interactions, the system contributes to a vibrant and engaging campus experience that encourages student involvement and strengthens interdisciplinary connections.

Keywords: Education, Club, Management, Event, School, ASP.NET, React, Vite, TypeScript

ÖZET

Öğrenci Kulüpleri Yönetim Sistemi, bir eğitim kurumunda öğrenci kulüplerinin yönetimini kolaylaştırmak amacıyla geliştirilmiş bir web tabanlı uygulamadır. Ana hedef, kulüp yöneticileri, danışmanlar ve üyelerin kulüp faaliyetleriyle ilgili etkinlikleri kolayca düzenlemeleri, iletişim kurmaları ve iş birliği yapmaları için merkezi bir platform sağlamaktır. Sistem, duyurular, kulüp kaydı, etkinlik planlama ve kullanıcı yönetimi gibi modüller içermektedir. Kullanıcı dostu arayüzüyle tasarlanan sistem, tüm kullanıcılar için basitlik ve erişilebilirlik sağlamayı hedeflemektedir. Kulüp yönetim süreçlerini basitleştirme, iletişim kanallarını geliştirme, etkinlik koordinasyonunu kolaylaştırma ve kulüp üyeleri arasında birlikte çalışmayı teşvik etme amacıyla, Öğrenci Kulüpleri Yönetim Sistemi, hoş bir ortam oluşturmayı hedeflemektedir. Kullanıcılara erişilebilir araçlar sunarak anlamlı etkileşimleri teşvik etmek suretiyle, sistem öğrenci katılımını teşvik eden ve disiplinler arası bağları güçlendiren canlı ve etkileyici bir kampüs deneyimine katkıda bulunmaktadır.

Anahtar Kelimeler: Eğitim, Kulüp, Yönetim, Etkinlik, Okul, ASP.NET, React, Vite, TypeScript

ACKNOWLEDGEMENT

We would like to express our gratitude to our member supervisor, Research Assistant Evren Çifçi, for the contribution provided to this project and for guiding us through weekly meetings as well as to Associate Professor Dr. Mehmet GÖKTÜRK for guiding us during presentations.

Additionally, during our education, we would like to express our thanks to our family who provided us with support and knowledge.

Muhammed Oğuz

Süleyman Gölbol

CONTENTS

Abstract	v
Özet	vi
Acknowledgement	vii
Contents	ix
List of Figures	x
1 INTRODUCTION	1
1.1 Application URL	2
1.2 Application Repository URL	2
1.3 Module Details	3
1.3.1 Club	3
1.3.2 Departments	3
1.3.3 Event	3
1.3.4 Users	3
1.3.5 UserClub	4
1.4 Agile Roadmap	4
1.4.1 Phase 1	4
1.4.2 Phase 2	5
1.4.3 Phase 3	5
2 SYSTEM REQUIREMENTS	7
2.1 Server-Side Requirements:	7
2.2 Client-Side Requirements:	7
2.3 Devops Requirements:	8
2.3.1 VPN and Server Configuration	8
2.3.2 Remote Server Setup	8
2.3.3 Database Connectivity	9
2.3.4 Frontend Configuration	9
3 SYSTEM ARCHITECTURE	10
3.1 Activity Diagram	10

3.2	UML Class Diagram	11
3.3	Use Case Diagram	11
3.4	Database Diagram	12
3.5	Flowchart	13
4	SYSTEM IMPLEMENTATION	14
4.1	Client-Side Implementation:	14
4.2	Server-Side Implementation:	15
4.2.1	Swagger APIs	16
5	USER INTERFACE DESIGN	20
5.1	Home Page	20
5.2	Clubs Page	20
5.3	Event Page	21
5.4	Club Home Page	21
6	Conclusions	22
6.1	Success Criterias	22
	Bibliography	24

LIST OF FIGURES

1.1	Logo	2
1.2	QR Code for our web page	2
1.3	Phase1	4
1.4	Phase2	5
1.5	Phase3	6
3.1	Activity Diagramn	10
3.2	UML Diagramn	11
3.3	Use Case Diagramn	12
3.4	Database Diagram	12
3.5	Flowchart	13
4.1	Swagger APIs	16
4.2	Swagger APIs	17
4.3	Swagger APIs	18
4.4	Swagger APIs	19
5.1	Home Page	20
5.2	Clubs Page	20
5.3	Event Page	21
5.4	Club Home Page	21

1. INTRODUCTION

The Student Clubs Management System is a sophisticated web-based application meticulously designed to optimize the management and coordination of student clubs within educational institutions. In the dynamic landscape of academia, student clubs serve as invaluable catalysts in fostering a vibrant campus community, facilitating student engagement, and fostering personal and professional growth. However, the efficient administration of multiple clubs and their diverse activities often presents formidable challenges for club administrators, advisors, and members alike.

Recognizing the imperative for an efficient and centralized solution, the Student Clubs Management System was meticulously conceived to streamline club operations, augment communication channels, and simplify administrative processes. Harnessing the power of cutting-edge technology, this system endeavors to empower club leaders, advisors, and members with an intuitive and seamless platform to collaborate effectively, communicate seamlessly, and organize club-related activities with utmost efficacy.

The paramount objective of this project is to establish a comprehensive and user-friendly platform that effectively addresses the multifaceted needs of student clubs. By offering a centralized hub for managing club information, facilitating meticulous event planning, and nurturing seamless communication among club members, the Student Clubs Management System aims to optimize club operations, enhance overall efficiency, and engender a profound sense of community and belonging.

Within this system, club administrators will enjoy an effortless interface to create and update club profiles, disseminate crucial announcements and news, oversee membership management, and impeccably coordinate club events. Advisors, in turn, will be equipped with the tools to actively engage with clubs, judiciously oversee event approvals, and provide invaluable guidance and support. Simultaneously, club members will relish the ability to explore an extensive array of clubs, access pertinent club-related information, and enthusiastically participate in diverse activities tailored to their interests.

The significance of this project resides in its transformative potential to revolutionize the management and coordination of student clubs, heralding a new era of streamlined operations, enhanced communication, and heightened collaboration among club members. By proffering a centralized platform meticulously tailored to cater to the unique exigencies of student clubs, the Student Clubs Management System aspires to cultivate an inclusive and dynamic campus environment that nurtures holistic student development, fosters interdisciplinary connections, and engenders a profound sense of belonging within the illustrious student community.

In conclusion, the Student Clubs Management System epitomizes a pioneering endeavor poised to revolutionize the management and coordination of student clubs. By empowering club leaders and members, facilitating a thriving campus community, and providing an all-encompassing platform meticulously crafted to promote student growth and engagement, this transformative project endeavors to elevate the club experience to unparalleled heights.



Figure 1.1: Logo

1.1. Application URL

Web Page: <https://ogrenciprojeleri.gtu.edu.tr/KulupYonetim>



Figure 1.2: QR Code for our web page

1.2. Application Repository URL

Github Page: <https://github.com/muhammedogz/CampusClubs>

1.3. Module Details

The system encompasses various modules, including Announcements, Clubs, Departments, Events, Users, UserClub, and UserEvent. The Announcements module allows administrators and club leaders to post important announcements and updates, ensuring that club members stay informed about upcoming events, meetings, and other essential information.

1.3.1. Club

The Clubs module provides a centralized database of all registered clubs within the institution. It enables club administrators to create and manage club profiles, including descriptions, objectives, and contact information. Moreover, the module facilitates the club registration process, allowing students to easily explore and join clubs based on their interests.

1.3.2. Departments

To foster collaboration and interdisciplinary connections, the system includes the Departments module. Club leaders can associate their clubs with specific academic departments, creating opportunities for cooperation and support between clubs and departments.

1.3.3. Event

The Events module offers functionality for planning, organizing, and scheduling club events. Club leaders can create event listings, specifying details such as date, time, location, and description. The system also supports event registration and attendance tracking, ensuring efficient event management and facilitating participation monitoring.

1.3.4. Users

The Users module manages user profiles and authentication within the system. It provides personalized dashboards for students, club leaders, advisors, and administrators, granting access based on assigned roles and permissions. User profiles store relevant information, including contact details and club affiliations.

1.3.5. UserClub

The UserClub and UserEvent modules handle the association between users and clubs/events. Club leaders can invite students to join their clubs, and users can request membership in desired clubs. Additionally, advisors, identified through specific user attributes like "isAdvisor," can approve event proposals and oversee their execution. In summary, the Student Clubs Management System aims to create an inclusive and efficient platform that empowers club administrators, advisors, and members. The system's features promote effective communication, streamlined event planning, and enhanced collaboration among clubs and their members. By providing a user-friendly interface and comprehensive functionalities, the system contributes to building a vibrant campus community that fosters student engagement and interdisciplinary connections.

1.4. Agile Roadmap

1.4.1. Phase 1

During Phase 1 of our Agile roadmap, we laid the foundation for the Student Clubs Management System project. We initiated the project on GitHub, established a comprehensive roadmap, and conducted in-depth discussions with our teachers to analyze the project structure. Additionally, we carefully selected suitable project management tools to enhance collaboration and efficiency. To ensure a solid understanding of the .NET framework, we dedicated time to studying and addressing any knowledge gaps. These activities in Phase 1 set the stage for the subsequent development phases, providing a clear direction and equipping us with the necessary tools and insights to move forward with the project.

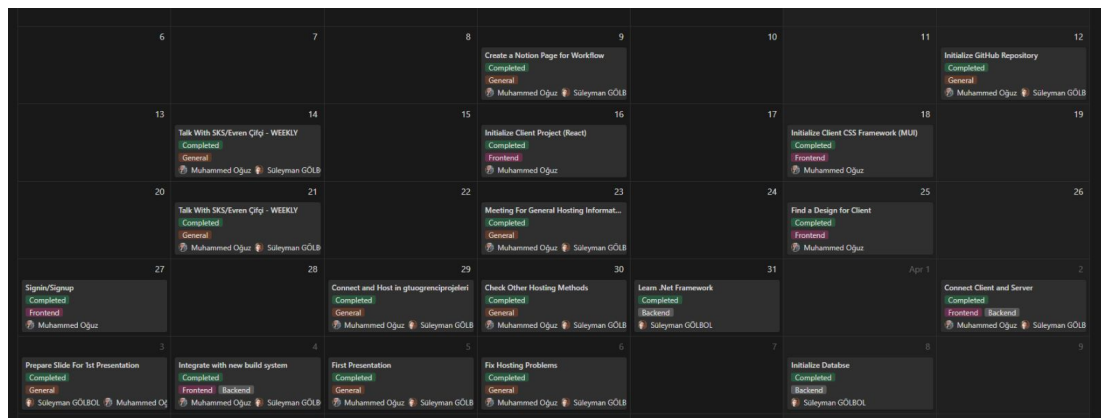


Figure 1.3: Phase1

1.4.2. Phase 2

Phase 2 of our Agile roadmap marked significant progress in the development of the Student Clubs Management System. We began by initializing the database, setting up the server, and configuring the client-side environment. With a focus on user interface design, we conducted thorough research to identify an optimal design approach. Armed with the design specifications, we commenced implementing the UI design into the codebase. Throughout this phase, we maintained consistent weekly meetings to ensure effective communication and collaboration among team members. As the development progressed, we gained a deeper understanding of the system's general structure, laying a solid foundation for the upcoming phases of the project.

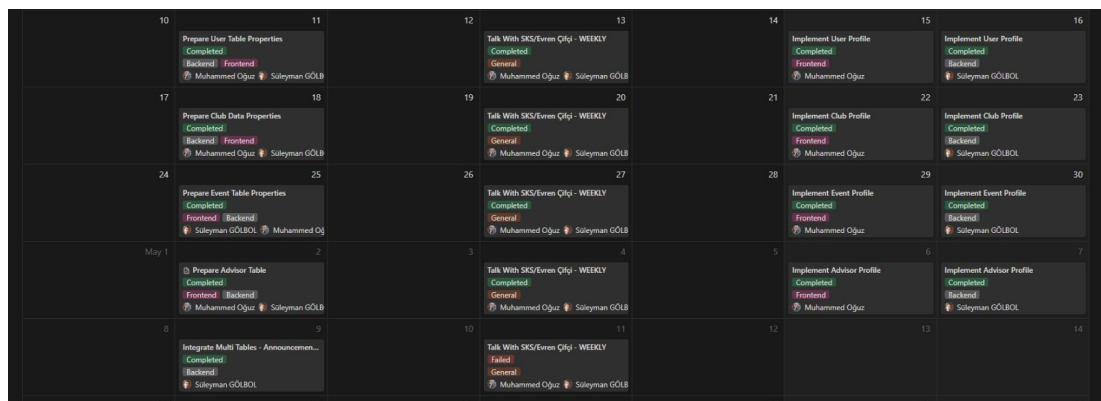


Figure 1.4: Phase2

1.4.3. Phase 3

In the phase 3 of our Agile roadmap we successfully implemented OAuth authentication to handle user authentication, ensuring a secure and streamlined login process. Weekly meetings continued to facilitate effective communication and collaboration within the team. Moreover, we implemented role-based authentication, enabling administrators to manage events and granting appropriate permissions. We established robust API connections, ensuring seamless integration between the server and client-side applications. Database integration and migration were also completed, ensuring data consistency and reliability. Additionally, we focused on optimizing the client-side by making it responsive and compatible with RESTful principles, resulting in a fast and responsive API. These achievements propelled the project forward, setting the stage for the next phase of development.

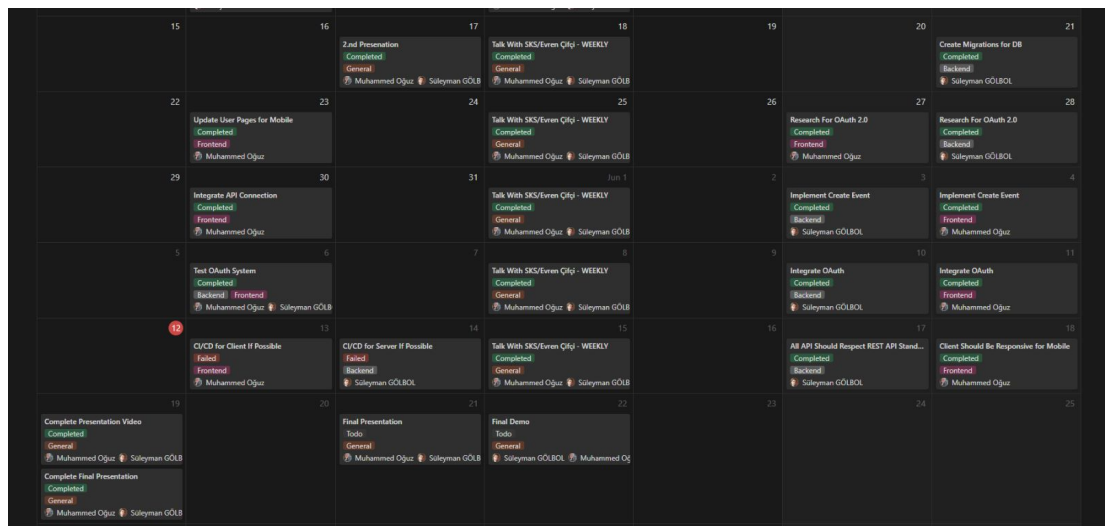


Figure 1.5: Phase3

2. SYSTEM REQUIREMENTS

The successful implementation and operation of the Student Clubs Management System require certain system requirements to be met. These requirements encompass the server-side and client-side components of the system.

2.1. Server-Side Requirements:

- **Operating System:** The server hosting the system should be compatible with a supported operating system Windows.
- **Web Server:** The Student Clubs Management System utilizes the Internet Information Services (IIS) server as the hosting platform. The IIS server is a powerful and reliable web server provided by Microsoft, specifically designed for hosting and managing web applications. [1]
- **Runtime Environment:** The server should have a compatible runtime environment which is ASP.NET or .NET V6 installed to execute the server-side code.[2]
- **Database Management System:** The system utilizes a relational database management system (DBMS) to store and manage data. The supported DBMS Microsoft SQL Server should be installed on the server.[3]
- **Networking:** The server should have a stable internet connection with sufficient bandwidth to handle incoming and outgoing requests from users accessing the system.

2.2. Client-Side Requirements:

- **Web Browser:** The system is designed to be accessed through modern web browsers, such as Google Chrome, Mozilla Firefox, or Microsoft Edge. Users should have a compatible web browser installed on their devices.
- **Operating System:** The client devices, including desktop computers, laptops, or mobile devices, should run a supported operating system, such as Windows, macOS, Linux, or mobile operating systems like Android or iOS. The system is developed responsive.

- **JavaScript Enabled:** The client devices should have JavaScript enabled in their web browsers to ensure proper functionality and interactivity of the user interface.
- **Network Connectivity:** Users accessing the system should have an active internet connection to communicate with the server and access the features and data of the Student Clubs Management System.

It is important to note that the system requirements may vary based on the specific implementation, scale, and technological choices made during the development process. Additionally, regular maintenance and updates may be required to ensure the system remains compatible with the evolving technology landscape and to address any security vulnerabilities that may arise.

Ensuring that the system meets these requirements will facilitate a smooth and efficient operation of the Student Clubs Management System, enabling club administrators, advisors, and members to effectively manage clubs, coordinate events, and foster a vibrant campus community.

2.3. Devops Requirements:

2.3.1. VPN and Server Configuration

To establish a secure connection to the remote server, a VPN (Virtual Private Network) should be set up and configured on the local machine. This involves installing a VPN client, configuring the connection settings with the provided server information, and authenticating with the appropriate credentials. Once the VPN connection is established, it should be tested to ensure proper connectivity to the remote network. After successfully connecting to the remote network via VPN, the server hosting the Student Clubs Management System should be accessed using a remote desktop or similar remote access method. This allows for seamless management and configuration of the server environment. By following these steps and establishing a secure VPN connection, the remote server setup becomes accessible, allowing for efficient hosting and management of the Student Clubs Management System.

2.3.2. Remote Server Setup

The remote server setup for hosting the Student Clubs Management System should involve establishing a VPN connection to the server. Once connected, IIS (Internet Information Services) should be configured to host the project using the drag and drop technique, ensuring that the system is accessible through the specified domain

or IP address. Additionally, for the frontend implementation using React, the project should be set up as a static web app, and the necessary configurations should be made in Dotnet's static web builder. These steps ensure a stable and secure remote server environment for hosting the Student Clubs Management System.

2.3.3. Database Connectivity

The database connectivity for the Student Clubs Management System should involve using Microsoft SQL Server and SQL Server Management Studio (SSMS). Once the VPN connection to the remote server has been established, it is recommended to connect to the database server using SSMS. A new database should be created for the system, and the necessary tables, relationships, and constraints should be configured. The connection string in the project's configuration file should be updated to point to the remote database server. This ensures that the Student Clubs Management System can access and interact with the database effectively, allowing for seamless storage and retrieval of club-related information.

2.3.4. Frontend Configuration

In order to configure the frontend of the Student Clubs Management System, certain steps should be taken. Firstly, the React[4] application should be built and prepared for deployment. This can be achieved by running the appropriate build commands to generate optimized and production-ready files.

Once the React application is built, it should be treated as a static web app. This involves configuring the web server, such as IIS (Internet Information Services), to serve the static files generated by the React build process. The necessary configurations should be made to ensure that the server correctly handles requests for the frontend assets.

Furthermore, if the Student Clubs Management System relies on any backend APIs, the frontend should be configured to communicate with these APIs. This can be done by specifying the appropriate API endpoints and configuring the necessary network settings to enable the frontend to interact with the backend services.

By following these steps, the frontend of the Student Clubs Management System can be properly configured and deployed, ensuring a seamless user experience and efficient communication with the backend services.

3. SYSTEM ARCHITECTURE

The system architecture of the Student Clubs Management System is designed to provide a robust and scalable foundation for the application. The system architecture of the Student Clubs Management System is designed to ensure modularity, scalability, and maintainability. It provides a solid foundation for the development and deployment of the application, enabling efficient management of student clubs within educational institutions.

3.1. Activity Diagram

The activity diagram represents the dynamic behavior of the system by illustrating the flow of activities or processes. It shows the sequential and parallel steps, decision points, loops, and conditions involved in the system's various activities.

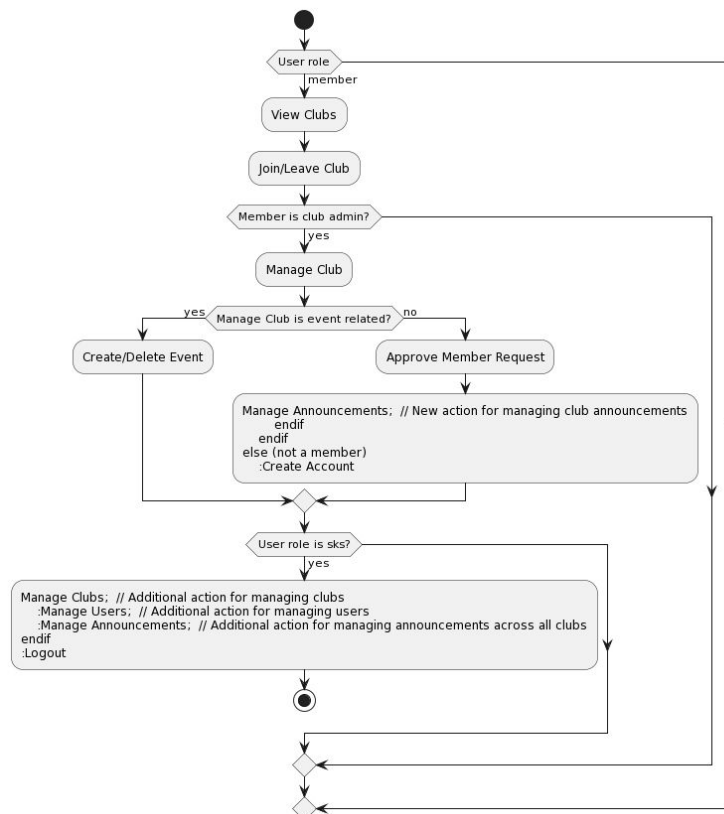


Figure 3.1: Activity Diagramn

3.2. UML Class Diagram

In the context of the Student Clubs Management System, UML diagrams play a crucial role in illustrating the system's architecture, components, and interactions. The class diagram, showcases the static structure of the system by depicting classes, their attributes, methods, and relationships. This diagram allows developers and designers to visualize the organization of the system's codebase and the associations between different classes.

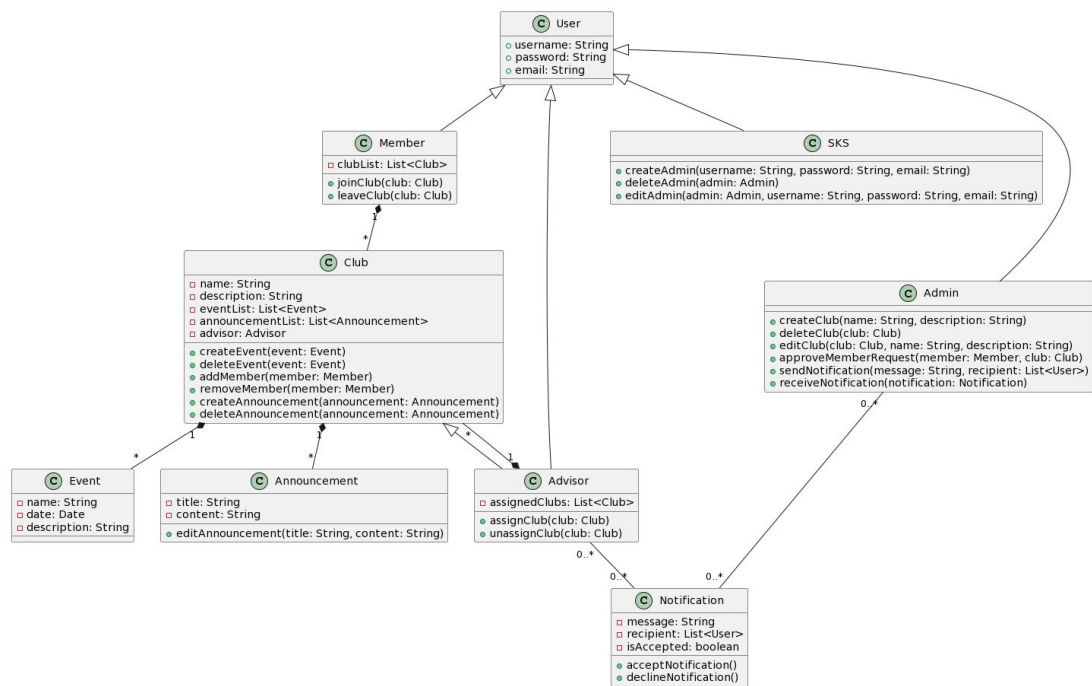


Figure 3.2: UML Diagramn

3.3. Use Case Diagram

Use-case diagram would depict the various use cases or actions that can be performed by different actors, such as club administrators, advisors, and members. These use cases could include functionalities like creating a club, managing events, approving event requests, joining a club, and accessing club information.

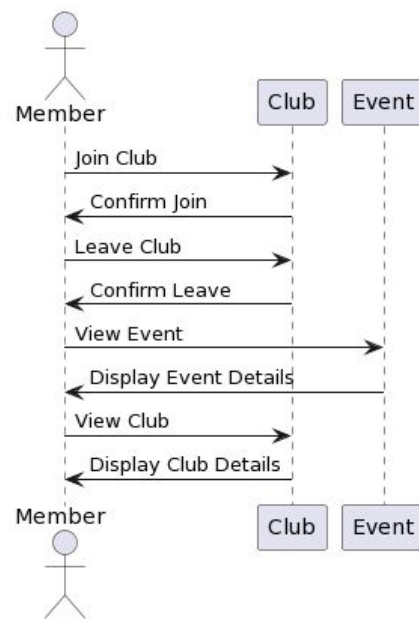


Figure 3.3: Use Case Diagram

3.4. Database Diagram

Database diagram is created with the help of migration which is a way to keep the database schema in sync with the Entity Framework model by preserving data.

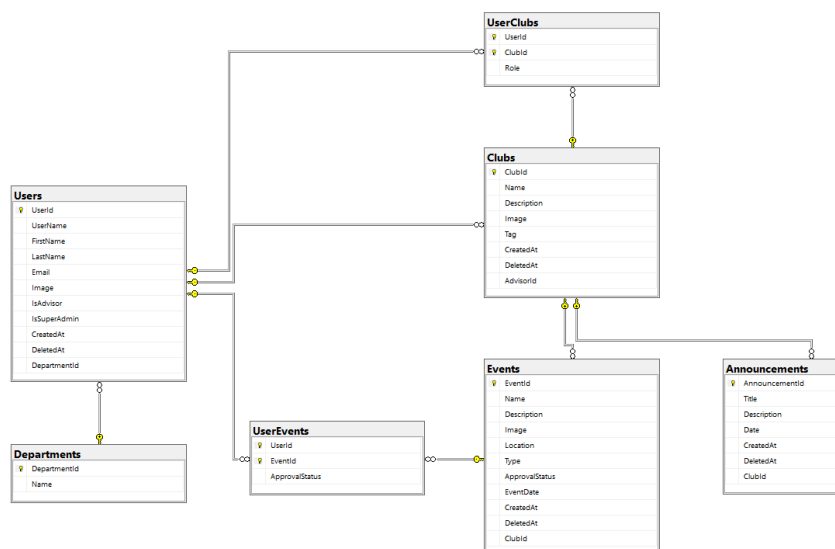


Figure 3.4: Database Diagram

3.5. Flowchart

Flowchart of the project is given below.

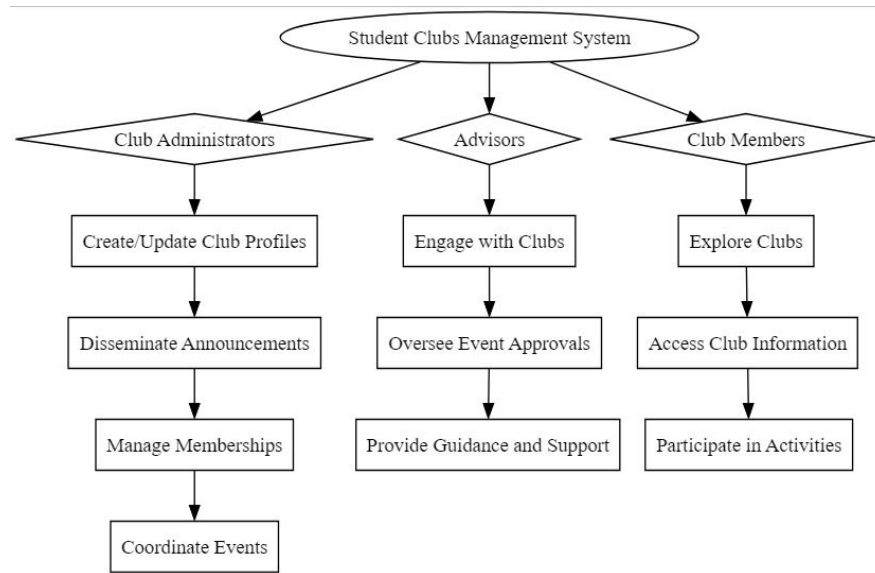


Figure 3.5: Flowchart

4. SYSTEM IMPLEMENTATION

There are 2 sides of implementation. Implementation and operation of the Student Clubs Management System details are given below.

4.1. Client-Side Implementation:

In the Student Clubs Management System, interface design plays a vital role in delivering an intuitive and user-friendly experience for club administrators, advisors, and members.

- **MUI (Material-UI) design library**, We used this which offers a rich set of pre-designed components and styling options that adhere to modern design principles. By leveraging MUI[5], we were able to create visually appealing interfaces with consistent design patterns, enhancing the overall user experience.
- **Vite**: During the development process, the Vite[6] build tool proved to be invaluable. Its fast and efficient module bundling capabilities enabled us to quickly iterate on the project, ensuring rapid feedback and seamless integration of changes. The hot module replacement feature further expedited the development process by instantly reflecting code modifications without the need for manual page refreshing.
- **Local Storage**: To maintain user session state and securely hold authentication tokens, we utilized local storage. This approach allowed us to persist user login information across browser sessions, ensuring a seamless and convenient experience for users interacting with the system.
- **Axios**: For making API requests and handling server-side communication, we relied on the Axios library. Axios simplifies the process of making HTTP requests, providing a convenient and consistent interface for fetching data from the backend. Its robust features, such as request cancellation and interceptors, empowered us to handle various aspects of data retrieval and ensure smooth data flow between the client and server.

In summary, the interface design of the Student Clubs Management System benefitted from the integration of the MUI design library, the efficient build process facilitated by Vite, the utilization of local storage for token management, and the

streamlined API communication achieved through the Axios library. These technological choices collectively contributed to the creation of a visually appealing, responsive, and user-friendly interface that enhances the overall usability and satisfaction of the system.

4.2. Server-Side Implementation:

In the server-side implementation of the Student Clubs Management System, we adopted a specific approach to handle image storage and token management.

- **Image Storage:** To efficiently manage images, we established a dedicated server folder on the remote computer, which was accessible through a VPN connection. This remote server, running IIS (Internet Information Services), served as a centralized repository for storing club-related images. By storing the images on this separate server, we ensured reliable and efficient access to image resources within the system.
- **Json Web Tokens:** Regarding token management and authorization, we employed JSON Web Tokens (JWT) to generate and handle bearer tokens. Within these tokens, we securely stored essential user information such as username, name, ID, email, and role. This allowed us to maintain user session state and facilitate secure authentication and authorization processes throughout the system. By leveraging the capabilities of JWT, we were able to establish a robust and reliable mechanism for user authentication, ensuring that only authorized users could access the system's functionalities and resources.
- **MVC:** In terms of server architecture, we implemented the Model-View-Controller (MVC) pattern, which facilitated the separation of concerns and improved code organization. This architectural approach allowed us to clearly define models, views, and controllers, resulting in a modular and maintainable server-side implementation. Additionally, we employed Data Transfer Objects (DTO) to encapsulate and transfer data between different layers of the application. DTOs enhanced the efficiency and security of data exchange, minimizing unnecessary data exposure and ensuring smooth communication between the client and server components.
- **Migration:** Another thing we used is Migration structure. Migration provided a structured and controlled approach to evolving the database schema over time. It allowed us to apply modifications, such as creating new tables, modifying columns, or adding constraints, in a systematic manner. With each update,

the migration framework ensured that the database schema reflected the latest changes, keeping the system in sync with the evolving requirements.

- **OAuth 2.0:** In the Student Clubs Management System, we implemented OAuth 2.0[7], which is an open standard for secure authentication and authorization. OAuth 2.0 allows users to grant limited access to their protected resources on one website (server) to another website or application (client) without sharing their credentials.

Working closely with Mr. Evren Çifçi, we utilized OAuth 2.0 to handle user authentication and retrieve user information. By integrating an OAuth provider that functions similarly to the E-Devlet authentication system, we were able to authenticate users using their existing credentials from trusted identity providers. Through this process, we obtained user information such as the username, which allowed us to personalize the user experience and provide access to relevant club-related features and functionalities.

Overall, the combination of a dedicated image server folder, protected OAuth 2.0, tokenization with JWT[8] for secure user information storage, and the utilization of MVC and DTOs in the server architecture contributed to the robustness, security, and maintainability of the Student Clubs Management System.

4.2.1. Swagger APIs

The tests are made using Swagger API since it's an interface for RestApi without accessing to source codes.

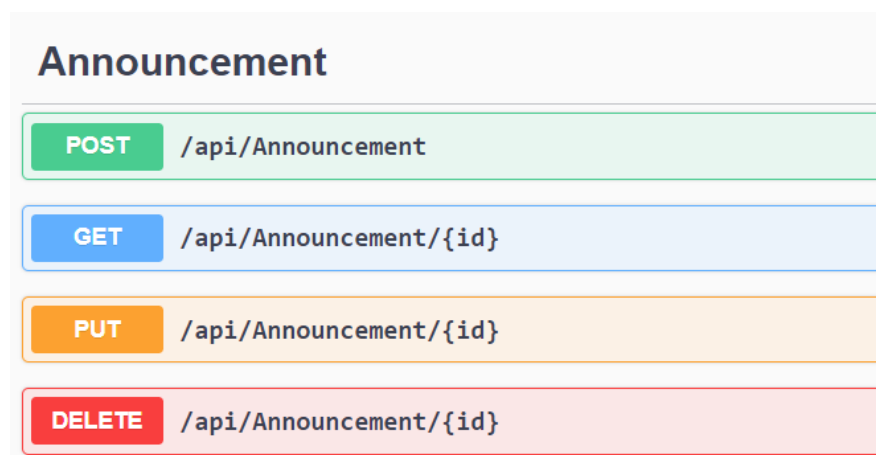


Figure 4.1: Swagger APIs

Auth	
POST	/api/Auth/token
POST	/api/Auth/register
POST	/api/Auth/auth
Clubs	
GET	/api/Clubs
POST	/api/Clubs
GET	/api/Clubs/{id}
PUT	/api/Clubs/{id}
DELETE	/api/Clubs/{id}
POST	/api/Clubs/{clubId}/join
GET	/api/Clubs/{clubId}/pending
PATCH	/api/Clubs/{clubId}/approval/{userId}

Figure 4.2: Swagger APIs

Department	
GET	/api/Department
Events	
GET	/api/Events
POST	/api/Events
PUT	/api/Events/{id}
DELETE	/api/Events/{id}
GET	/api/Events/{id}
POST	/api/Events/{eventId}/join
GET	/api/Events/user/{id}/pending
POST	/approval/{eventId}/user/{userId}
GET	/api/Events/pending
PATCH	/api/Events/approval/{eventId}

Figure 4.3: Swagger APIs

FileUpload	
POST	/api/FileUpload
Memberships	
POST	/api/Memberships
Notification	
GET	/api/Notification
Users	
GET	/api/Users/students
GET	/api/Users/teachers
GET	/api/Users/{id}
PUT	/api/Users/{id}
DELETE	/api/Users/{id}
POST	/api/Users

Figure 4.4: Swagger APIs

5. USER INTERFACE DESIGN

5.1. Home Page

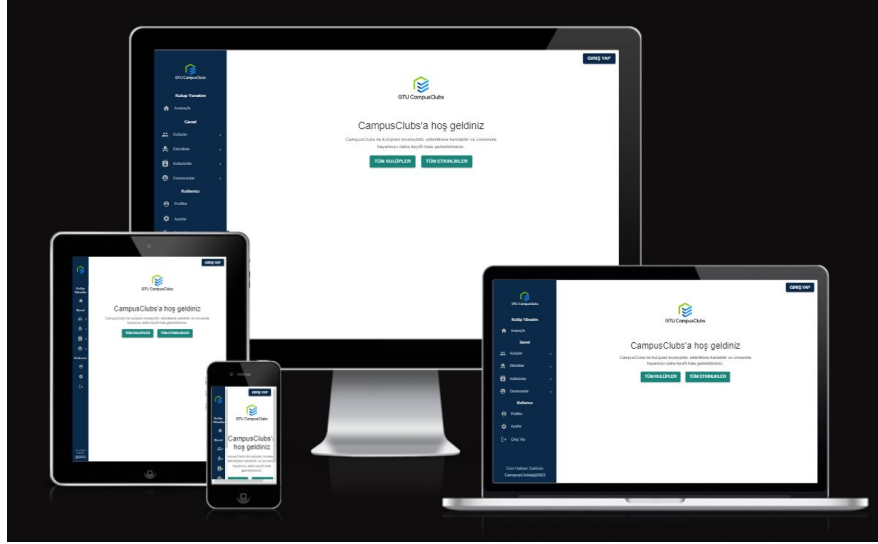


Figure 5.1: Home Page

5.2. Clubs Page

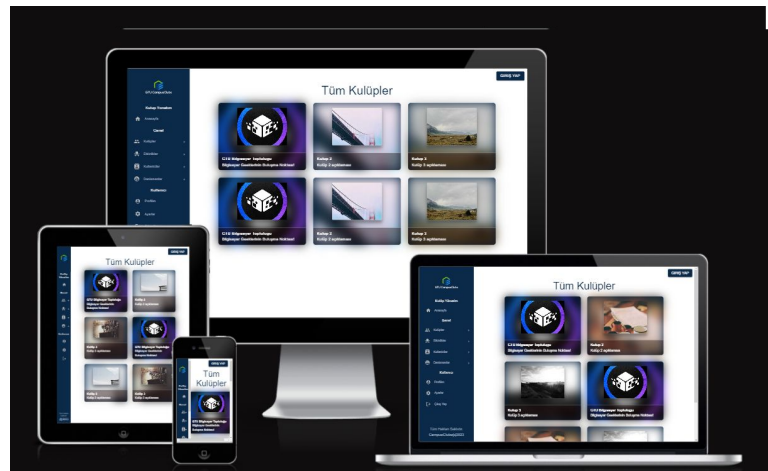


Figure 5.2: Clubs Page

5.3. Event Page

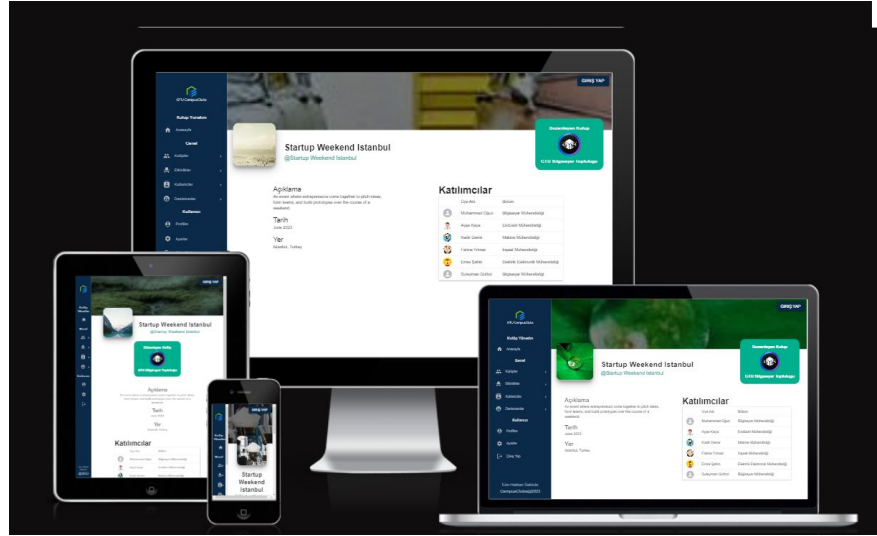


Figure 5.3: Event Page

5.4. Club Home Page

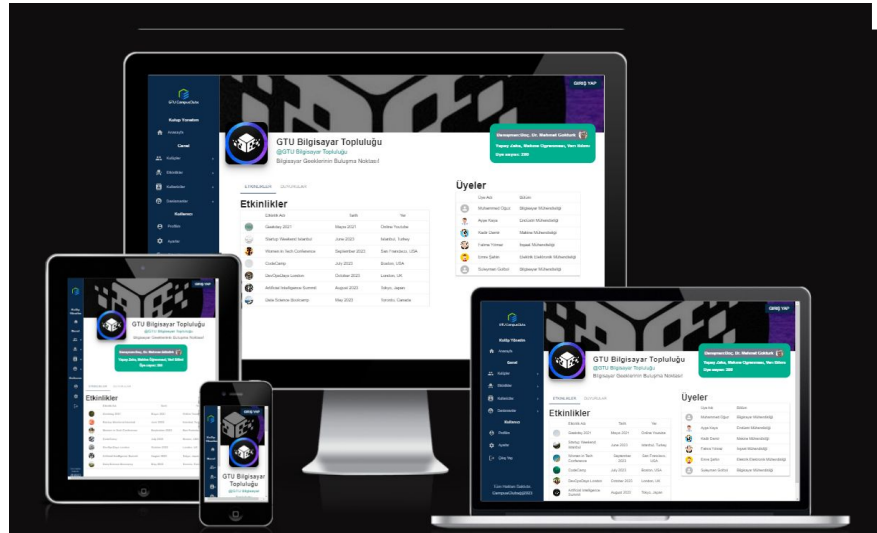


Figure 5.4: Club Home Page

6. CONCLUSIONS

Our development efforts have been driven by a strong commitment to fostering a vibrant campus community and empowering student clubs within educational institutions. By utilizing cutting-edge technologies such as ASP.NET, SQL Server, React, TypeScript, and Vite, we have created a robust and sophisticated system that effectively addresses the needs of student clubs.

Through the Student Clubs Management System, we aim to provide a platform that facilitates efficient club operations, enhances communication channels, and streamlines administrative processes. By leveraging the system's features, student clubs can effectively manage their activities, organize events, share announcements, and collaborate with their members.

We firmly believe that a thriving club ecosystem contributes to the overall success and growth of educational institutions. By providing students with the tools they need to maximize their club experiences, we enable them to develop valuable skills, foster meaningful connections, and make a positive impact on campus.

In conclusion, our Student Clubs Management System stands as a testament to our dedication to creating innovative solutions that bridge the gap between administrative requirements and user needs. We remain committed to supporting the success of student clubs, promoting student engagement, and nurturing personal and professional growth within educational institutions.

6.1. Success Criterias

During the development of the Student Clubs Management System, we set specific success criteria to ensure the system meets the desired objectives. We are proud to announce that we have successfully achieved all of these criteria:

- **Integration with OAuth2 Portal:** We have seamlessly integrated the system with an OAuth2 portal for secure login authentication. This ensures that only authorized users can access the system and its functionalities.
- **Fast Response Time:** One of our primary goals was to provide a highly responsive system. Through efficient coding practices and optimization techniques, we have achieved an impressive response time of between 500 milliseconds - 1 second for most operations. This ensures a smooth and seamless user experience.
- **Mobile Responsiveness:** Recognizing the importance of mobile devices in to-

day's digital age, we have designed the system to be fully responsive across various screen sizes. Whether accessed from a desktop, tablet, or smartphone, users can enjoy a consistent and user-friendly interface.

- **RESTful API Design:** To facilitate seamless communication and integration with other systems, we have implemented well-designed RESTful APIs. These APIs adhere to industry best practices, enabling easy access to the system's functionalities and data.

BIBLIOGRAPHY

- [1] *Iis*, <https://iis.net>.
- [2] *.net*, <https://asp.net>.
- [3] *Mssql dmbs*, <https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms>.
- [4] *React*, <https://react.dev/>.
- [5] *Mui*, <https://mui.com/>.
- [6] *Vite*, <https://vitejs.dev/>.
- [7] *Oauth2.0*, <https://oauth.net/2/>.
- [8] *Jwt*, <https://jwt.io/>.