ORTA DOĞU TEKNİK ÜNİVERSİTESİ
MIDDLE EAST TECHNICAL UNIVERSITY
KUZEY KIBRIS KAMPUSU ◆ NORTHERN CYPRUS CAMPUS

CNG 495
CLOUD COMPUTING

FALL 2024 – PROJECT PROPOSAL

Team Members	
Ekrem Cagatay Goz	2526374
Haya Arabi Katibi	2542520
Engin Eray Kabalak	2526424

Project Description

The METU NCC Society website is designed to enhance communication between societies, students and administrators. The purpose of this website is to share announcements, event updates, and reservation requests online. Currently, society announcements are shared via "This Week on Campus," a weekly publication. Which prevents societies from changing or updating the events posted. Our platform solves this issue by offering a dynamic, real-time announcement system where societies can share updates instantly and include images in their posts.

This system will also feature a robust, role-based structure:

1. Admin: The admin will approve all the society-related activities including the amphitheater and seminar hall reservations at the CCC. Society Presidents can request room bookings and event slots, which will be approved/denied by the admins.
2. Society Presidents: Society presidents can post announcements, update events, and request event slots and room reservations for the week. Requests of this type go to the admin for approval, enabling flexible scheduling and planning.
3. Students: Students can only view the announcements, allowing them to browse announcements by society or view all announcements at a time.

The homepage will show all societies, and each society will have its specific announcements and events. Additionally, society presidents can access a CCC reservation system that functions similarly to a library booking system allowing them to book a room online.

Cloud Delivery Models

AWS Amplify

Type: SaaS / PaaS

Purpose: This service will provide hosting for the website, authentication and backend API creation.

Amazon S3

Type: IaaS

Purpose: Stores images, documents, and media files.

AWS RDS

Type: PaaS

Purpose: This service will manage the data with a relational database.

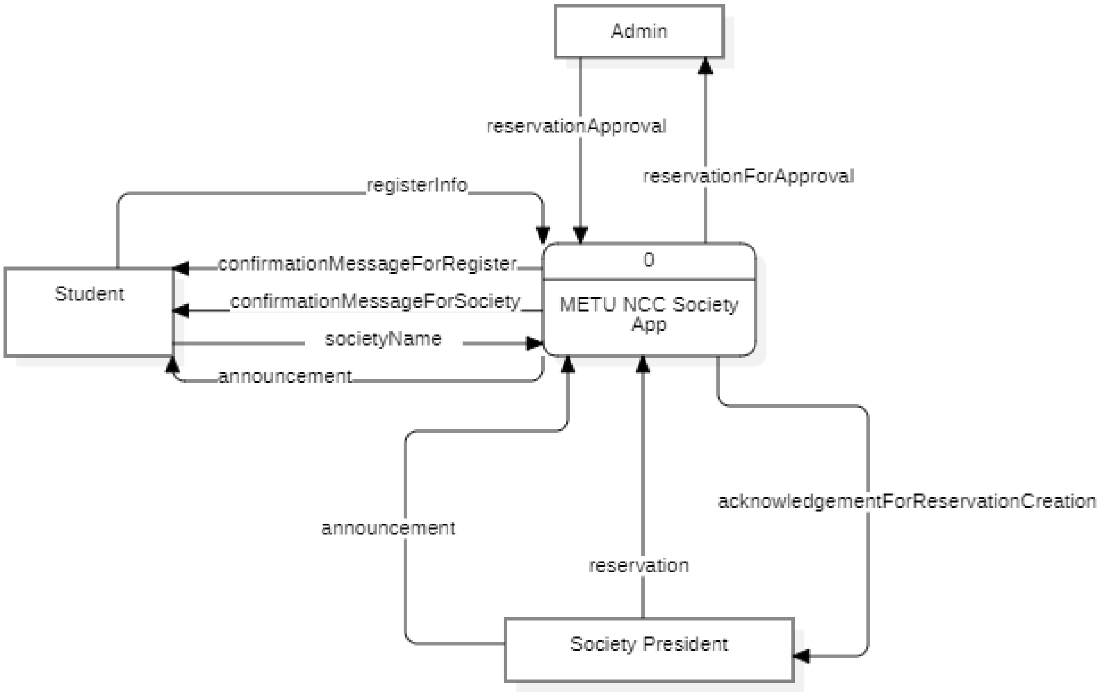
AWS IAM

Type: IaaS

Purpose: Controls permissions and access to AWS resources, managing roles such as admin, society president, student.

Client Server Interaction Diagram

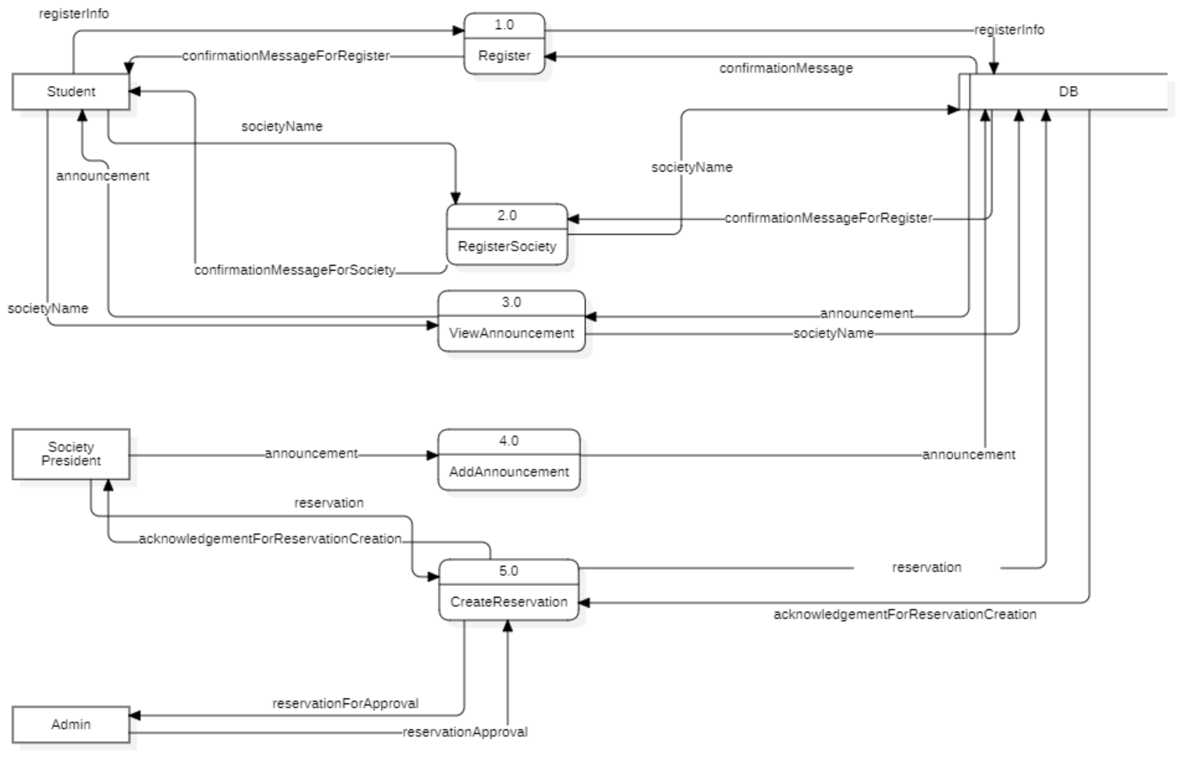
Dataflow Diagram



The context data flow diagram shows what the application needs and what it sends to users.

Student actor sends one’s register information, and the system returns a confirmation message (true or false) as feedback. The purpose of the society name data is to be able to register societies and list the announcements of this society.

Society President can add new announcements into the system. Also, the president can reserve a place, and the system sends this request to the admin. After admin sends response back to the president.



This diagram shows the main processes of our application.

Register process takes the student information as an input, and handles register logic.

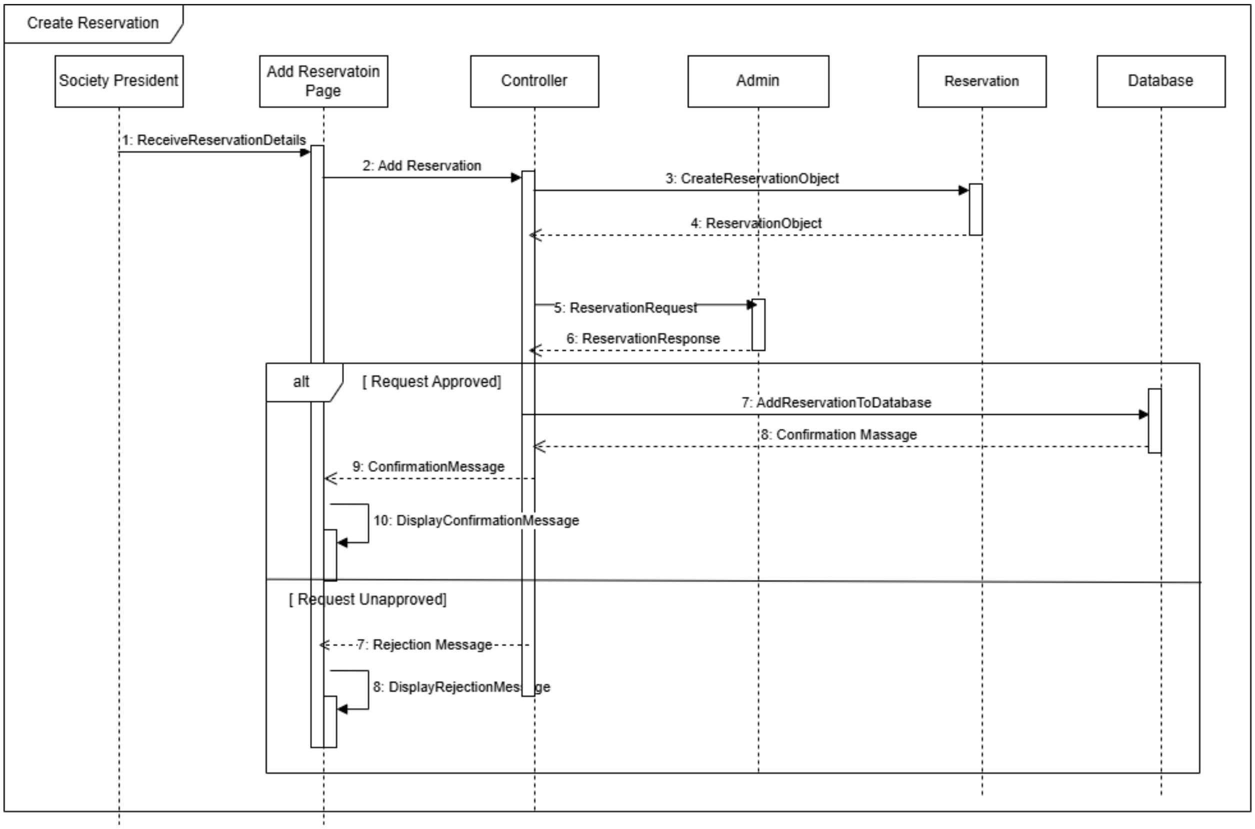
RegisterSociety process takes the society name as input to register this society and sends confirmation message to student.

ViewAnnouncement process handles showing announcements for given society as input.

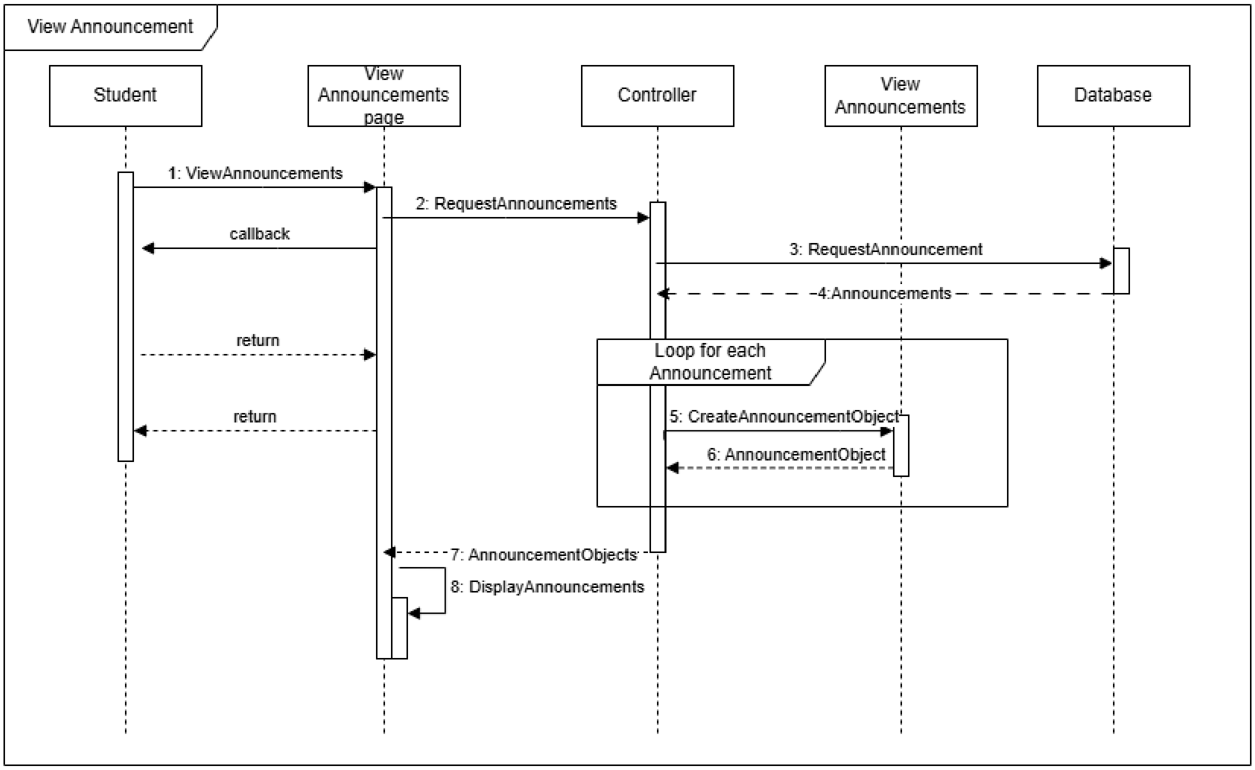
AddAnnouncement process takes announcement data from the president and add this announcement to society page.

CreateReservation process takes reservation information, and this information is sent to the admin to be checked. According to admin's answer, acknowledgement data is sent to the president.

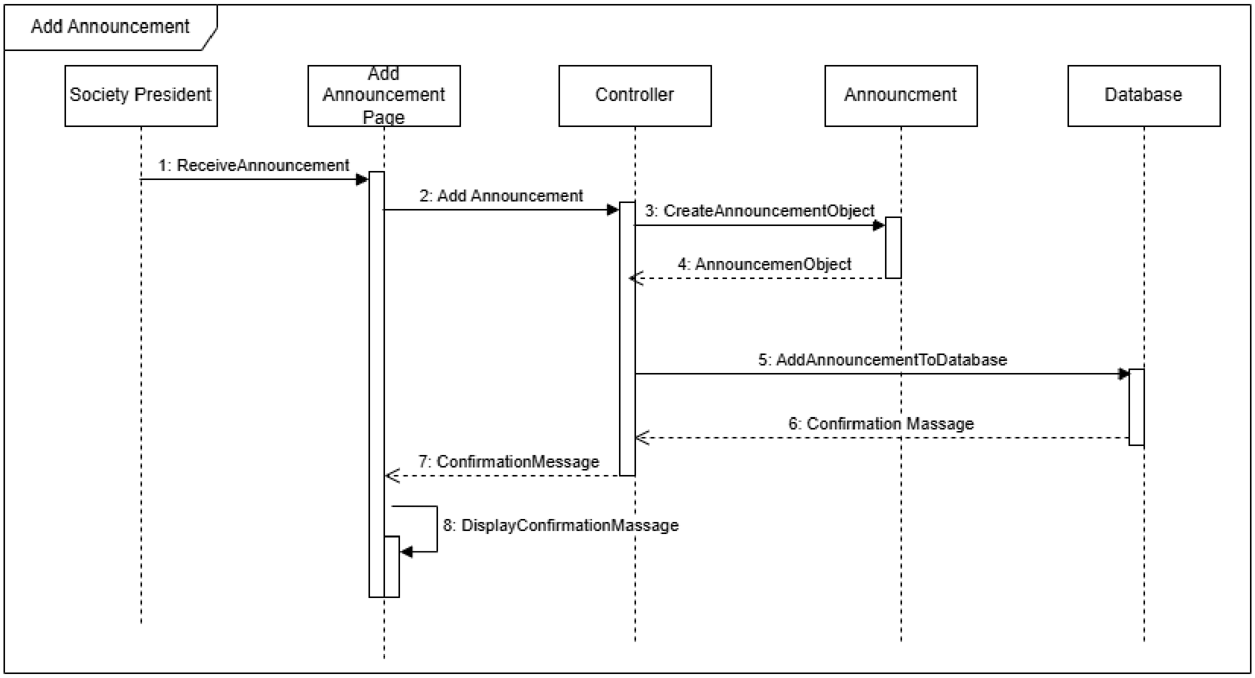
Sequence Diagram



The Create Reservation Sequence diagram outlines the process of society president creating a reservation in the CCC building. It starts with the society president entering the reservation information to the reservations page. The details are then sent to the controller to add the reservation. A request is sent to the admin to confirm or reject the reservation. If the admin confirms, the reservation is added to the database and a confirmation message is displayed. If the admin rejects the reservation, a rejection message is displayed.



The View Announcement sequence diagram highlights the steps needed to show the announcements for the student. It starts with the user requesting to view announcements. The system will retrieve the data from the database. Finally, the announcements will be displayed on the view announcement page.



Add Announcement sequence diagram shows how the society president adding an announcement will be handled in the system. First the society president will send information to the add announcement page which will be directed to the controller. The announcement will be added to the database and a confirmation message will be displayed.

Data Types

1. **Text Data**:
    a. **User Information**: Think of it as the essentials—usernames, email addresses, and those encrypted passwords that keep accounts safe. All these are crucial for letting users log in and for the app to know who's who.
    b. **Society and Room Details**: These are simple and straightforward descriptions, like what the room is called or details about a society. They're important for searching and displaying relevant information easily.
    c. **Reservation Details**: Basically, the "who, what, and when" of room bookings—dates, times, and purposes, all organized to make room management smooth.
    d. **URLs**: Just the links to event pages or extra resources, ready to be accessed when needed.
    e. **Metadata**: Background information like when something was created or updated and details about user roles. It's what keeps the app organized and secure.
2. **Binary Data**:
    a. **Images**: Think of society logos or event flyers. These are stored efficiently in AWS S3.
3. **Numerical Data**:
    a. **IDs**: These are like name tags for users, societies, announcements, and reservations. They keep everything in order behind the scenes.
4. **Boolean Data**:
    a. **Approval Status**: A simple yes/no flag that says if a room reservation is approved or not.
    b. **User Roles**: Quick checks to see if someone is an admin, a society president, or just a regular student.
5. **Mixed Data**:
    a. **Announcements**: These combine both words and pictures, giving announcements a more engaging feel.

Computation

1. **Posting Announcements**:
    a. When a society president wants to post an announcement, they upload a description and maybe an image, too. The app saves the text and uploads the image to AWS S3, generating a unique ID to keep everything tidy. It's all about keeping the data organized and accessible.
2. **Booking a Room**:
    a. Say a president of society wants to book a room. The app checks if the room is available at the desired date and time. If yes, it books it. If not, the user gets a friendly message suggesting other available time slots.
3. **Database Operations**:
    a. The app interacts with the database a lot: adding new records, retrieving announcements, checking room availability, and updating information. All of this happens behind the scenes to make the user experience smooth.
4. **Managing Users and Roles**:
    a. The app checks user credentials to ensure secure logins and identifies user roles, giving or restricting access based on whether they're an admin, society president, or student.
5. **User-Friendly Interface**:
    a. The UI is designed to be intuitive. It handles form submissions, checks for mistakes, and gives immediate feedback. For example, when a user makes a reservation, they instantly see a confirmation or a note about room availability.
6. **Handling Mistakes and Logging**:
    a. The app is prepared for hiccups, like an image upload failing or someone entering invalid data. It provides clear messages to the user and logs these issues so developers can fix problems efficiently.

Expected Contribution for Each Project Group Member

Haya Arabi Katibi: Frontend, Haya will work on designing and building the website's interface, making sure everything looks great and is easy to use.

Engin Eray Kabalak: Backend and Database, Eray will work on the backend, focusing on building the core functionality and managing the database to keep everything running efficiently and securely.

Ekrem Çağatay Göz : Backend and Database Çağatay will work on the backend, focusing on building the core functionality and managing the database to keep everything running efficiently and securely.

All Of Us: All of the members will work together to set up and manage cloud services AWS. This includes tasks like hosting, setting up databases, and so on. We'll make sure the website is reliable, scalable, and runs smoothly.

References

Amazon Web Services Documentation

2-AWS Amplify Documentation: https://docs.aws.amazon.com/amplify

3-Amazon S3 Documentation: https://docs.aws.amazon.com/s3

4-AWS RDS Documentation: https://docs.aws.amazon.com/rds

5-AWS IAM Documentation: https://docs.aws.amazon.com/iam

Web Development References

1-W3Schools - Full Stack Web Development: https://www.w3schools.com

2-MDN Web Docs: https://developer.mozilla.org