# CAREER EPISODE 03

## MEET NOW! (Cloud Based Web Conferencing)

### INTRODUCTION

### CE 3.1

After graduating from college, I was accepted to [redacted] [redacted] Computer Science and Engineering program. One of the projects I worked on for this degree was designing a cloud-based web conferencing platform, which I was asked to work on in my seventh semester, [redacted] under the course title “Cloud Computing”. I asked my course instructor to be my mentor and project supervisor, and he was really helpful throughout the entire task. I decided to use all the resources available to me, both inside and outside of my university.

### BACKGROUND

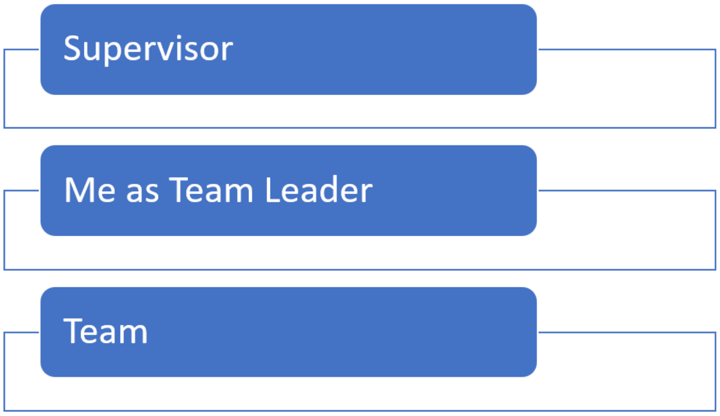
### CE 3.2.1

I knew that it might be challenging to arrange a time and location for in-person conferences that work for everyone. Several well-known companies employ video conferencing software that requires extra hardware to implement and encode audio and video streams. It is difficult to transport those hardware devices everywhere, and stream secrecy is compromised. Because personal computers, laptops, and mobile phones have limited storage capacity, many consumers these days do not wish to download any applications. They require any web-based application that can operate without downloading, and they also don't want to carry their sensitive information on their devices in case they misplace it. Businesses wish to protect their streams from other outside conferencing services.

### CE 3.2.2

This project is specifically designed to address all of these problems and save time. A secure cloud-based web conferencing application called MEET NOW was proposed to improve the interactive atmosphere in any kind of business. This application's goal is to assist employees in setting up safe meetings, communicating with one another, and managing all streams within the company. The application is client-server based. There are no geographical restrictions because this web program is simply accessible from any computer, laptop, or Android phone with internet connectivity and is dependent only on device capacity and bandwidth. Meeting invitations are sent to the appropriate members by this application. In addition to storing their critical data in the program, users can use text messages and audio/video feeds to communicate during meetings. A meeting can be started by the administrator, who can also allow participants access. No additional hardware is needed because I handled all of the processing, including encryption, decryption, compression, and decompression, within the program. All devices can use the "MEET NOW" because of its responsive and easy-to-use interface. I used my knowledge that I obtained from different workshops and seminars to finish this project.

## PROJECT REPORTING HIERARCHY



## PERSONAL ENGINEERING ACTIVITY

### CE 3.3.1

I thoroughly reviewed the literature using every resource at my disposal. Creating a dependable, affordable, cloud-based conferencing solution and enabling remote virtual meetings were among my primary goals. I designed this platform to assist users in keeping their most critical files in the appropriate panels. During the conference, participants could text each other in a group chat. Multiple members might use the webcam capability simultaneously. The goal was to create a device-independent program and to offer a single platform for virtual meetings, communication with other users, text message sharing, and cloud data storage with only a fast internet connection. An organization's total growth could be boosted by this application, which also eliminated the need to set up separate meeting spaces.

### CE 3.3.2

This application I've put all the features into place by using any internet-connected device to handle meetings and create a better atmosphere for communication between managers and staff members. In contrast to current apps, the suggested application allowed for secure audio and video streaming. The encoder supported the minimum latency. This program offered users a cost-effective alternative. Organizations must pay for the things they utilize. Each meeting had a minimum cost that they must pay. I started working on its client/server architecture. Because servers were devoted to handling disk drivers or network traffic, I knew from prior experience that I could rely on them for assistance.

**CE 3.3.3**

I decided to divide my work into several parts and utilize the incremental strategy to construct this project. Each of these modules were then subdivided into easier-to-manage parts that comprise the actual requirements implementation. This model was used because it makes it simple to test and troubleshoot the final result during revisions. I decided to ask clients questions before beginning to develop its design. I also concentrated on requirements that were both functional and non-functional. I designed this specific process for collecting requirements depending on sophistication of application.

**CE 3.3.4**

I made the decision to interview clients in order to get their requirements and information. I was aware that requirements analysis was the process of organizing, projecting, and researching the application's overall requirements. Based on the literature, I came to the conclusion that an application should give functional requirements. I defined a service statement. Clients requested an administrator to handle user data as a result. I decided that meeting members should be managed by the administrator, who should also be allowed to start and stop meetings. Auto-response services should be generated by the application. All of the data that has been pulled from a centralized server ought to be handled by the administrator. The application must handle the backends' daily data expansion and streams. A single bitrate stream was encoded into a multi-bitrate stream by the application.

**CE 3.3.5**

The limitations or checks on the services and functions that an application offered, such as time limits and development standards/process limitations, were known as non-functional requirements. MEET NOW's non-functional objectives, such as the need for an application that offers improved performance and reaction, were examined. A high-speed internet connection was needed to enable the application to stream live content in high quality. I used different standard authentication to secure the application, and Azure Media Services' cutting-edge technology, such as encryption, AES, and DRM content protection, safeguard the streaming services.

**CE 3.3.6**

The application was able to stream high-quality live material with a fast internet connection. The user must be able to interact with the program. I defined the use cases, the inputs and outputs to certain other actors, and the way inputs are converted into outputs. The use case typically outlines any potential problems during the specific behavior as well as the beneficial actions that the program will take. Below is the user login:

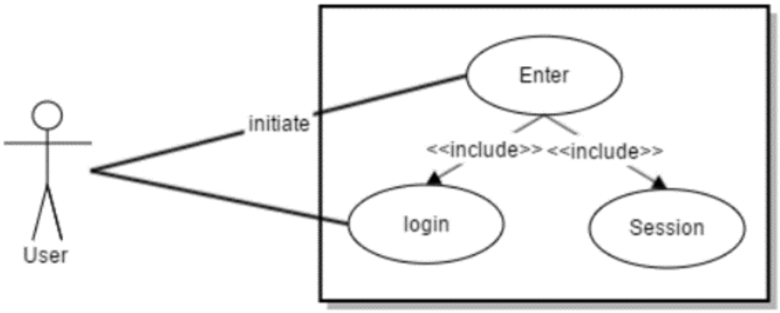


*Figure 1 Login Use Case*

**CE 3.3.7**

The administrator must have already registered the user in the program as a prerequisite for this login. To log in, the user needs to input their email address and password correctly. The ability to access the primary functionalities of the program was one of the post-conditions. The application confirms that the password and email address are registered and formatted correctly. The application shows the content of every other user as well as the session that the specific user started, if the inputs entered are accurate. The program will reroute to the main page if the password and email address entered are invalid or incorrect.

**CE 3.3.8**

My next task was to register administrator

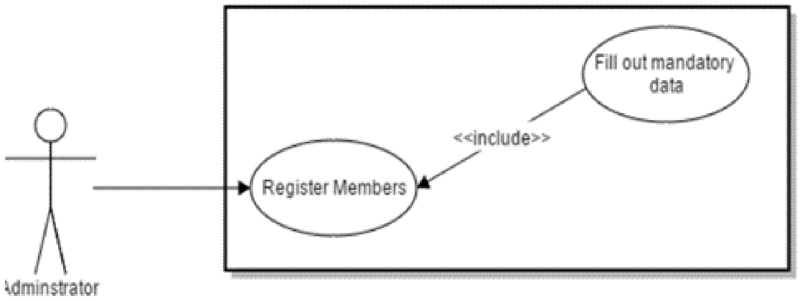


*Figure 2 Register Members*

I wanted to use administrator account to manage all of the member's required information. In the project directory, a member-specific folder was created. To carry out this task, the administrator could sign in to the application. User data was entered by the administrator. The "Create Account" button was clicked by the administrator. The application confirmed that the inputs submitted were legitimate. The application created the member account detail file once the supplied information was verified.

**CE 3.3.9**

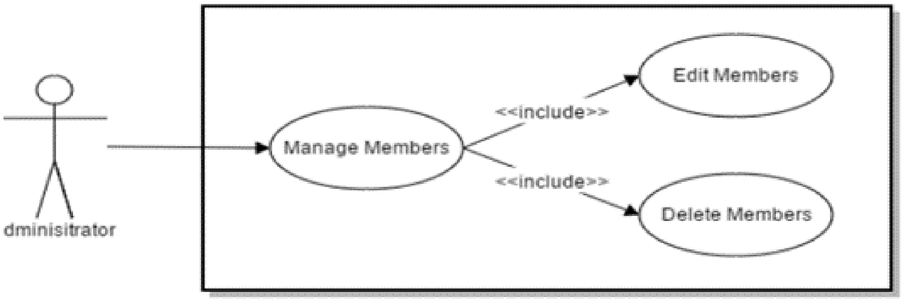


*Figure 3 Manage Members*

It was necessary to update and remove the member data. To carry out this task, the administrator could sign in to the application. After reviewing each member's details, the administrator chooses one to carry out these tasks. Admin must enter the most recent information about a particular member, then click the "Update" button. The application updates and confirms that the inputs entered were valid. Applications show a confirmation notification when a member's information was deleted. The administrator could either click the delete button to remove that member or cancel the deletion procedure by using the BACK TO DETAILS page button. The database was updated by the application.

**CE 3.3.10**

This was done for data storage and password changes. Data was kept in the project directory and database. A specific member was able to view uploaded data on the My Storage page. To do this task, the member must be logged into the program. The "MY STORAGE" button is selected by the application's user. Web server shows member's stored data. Only the member's files were visible. By selecting the download button that appeared on the left side of the file, members were able to download files. By inputting the data's name, members searched uploaded data. The application displayed an error notice if a member attempted to upload a video or large file larger than 4 MB. The application displayed an error notice if a file with the same name already existed. The application displayed an error notice if the data was in an invalid format.

**CE 3.3.11**

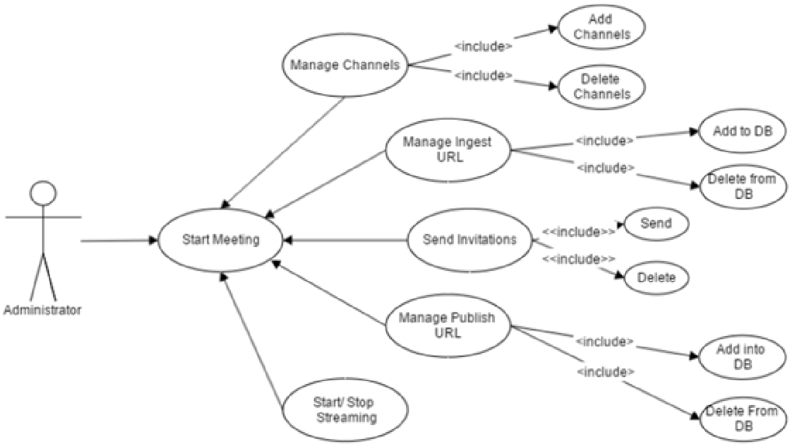


*Figure 4 Start/End Meetings*

An Azure Media Services (AMS) account was required for this. A channel ought to be established. The specific members received an invitation to the meeting. To create a channel in the Azure Portal for a specific member's live broadcast, the administrator inputs channel information into the application's channel creation page. The application established a channel on the AMS Account administrator's portal.

**CE 3.3.12**

I used a case of entering the meeting as shown below:

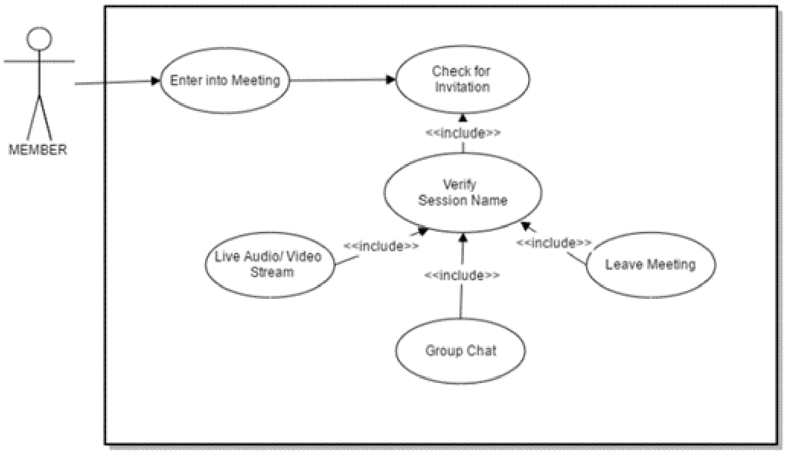


*Figure 5 Entering meeting*

After this, I designed sequence diagram as shown below:

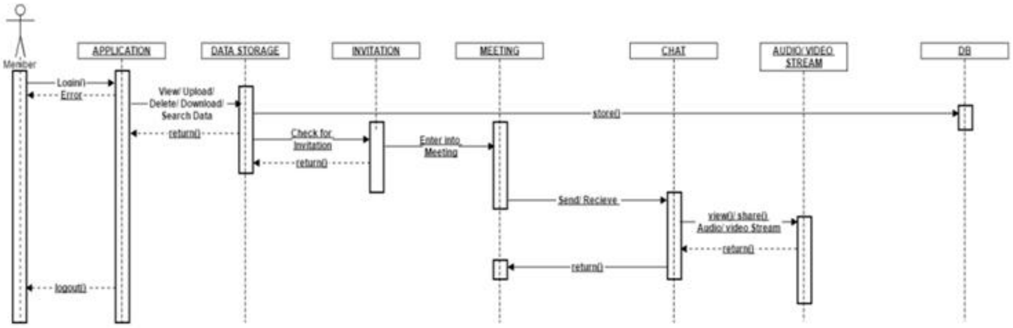


*Figure 6 Sequence Diagram*

## CE 3.3.13

To illustrate the kind of data that is delivered as input and output from the system, and where it will be stored, I created a data flow diagram.

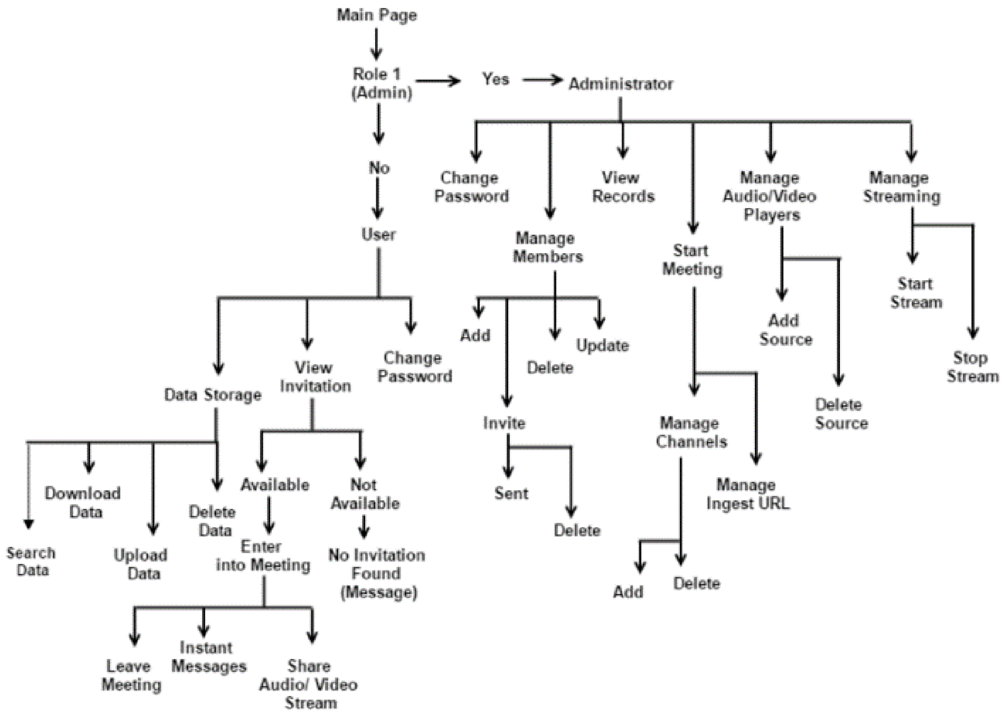


*Figure 7 Diagram of Data Flow*

I knew from my previous knowledge that I needed to design an entity relationship diagram as shown below:

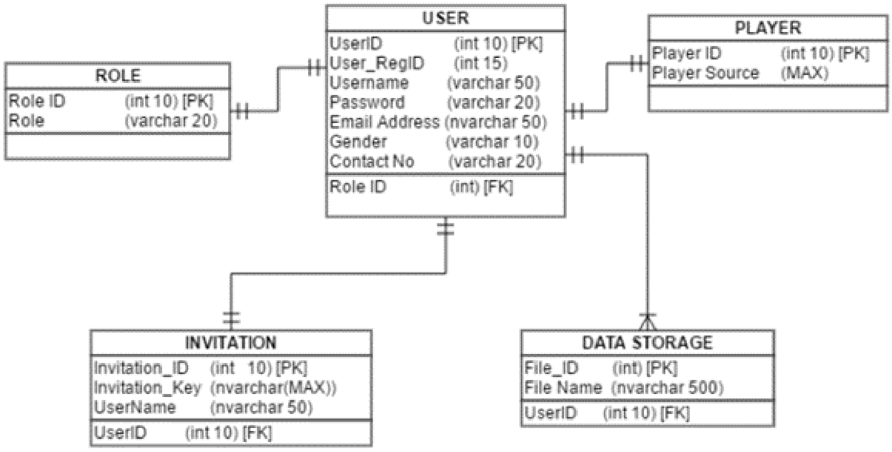
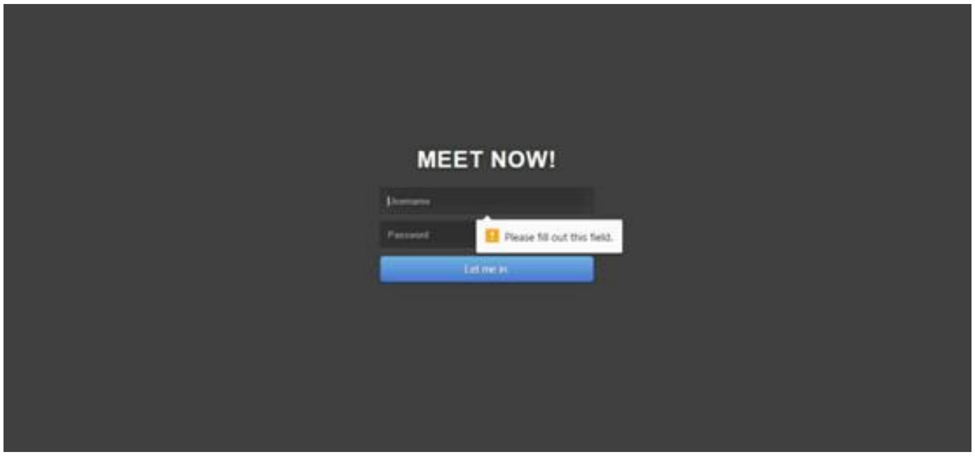


*Figure 8 Entity Relationship Diagram*

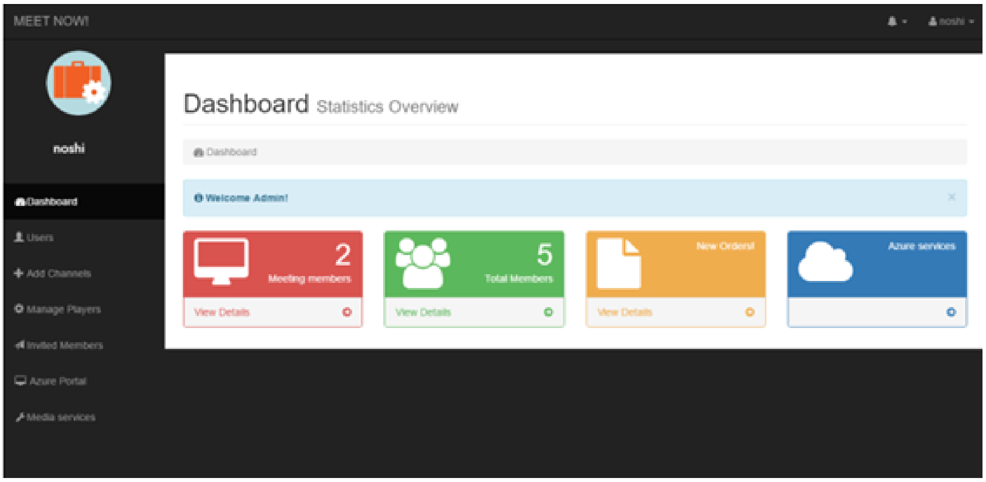
**CE 3.3.15**

The development of the application is the primary objective of this phase. Converting design specifications into executable programs was my next task.



Please fill out this field.

*Figure 9 Meet Now Login*



Welcome Admin!

*Figure 10 Dashboard for Administrator*

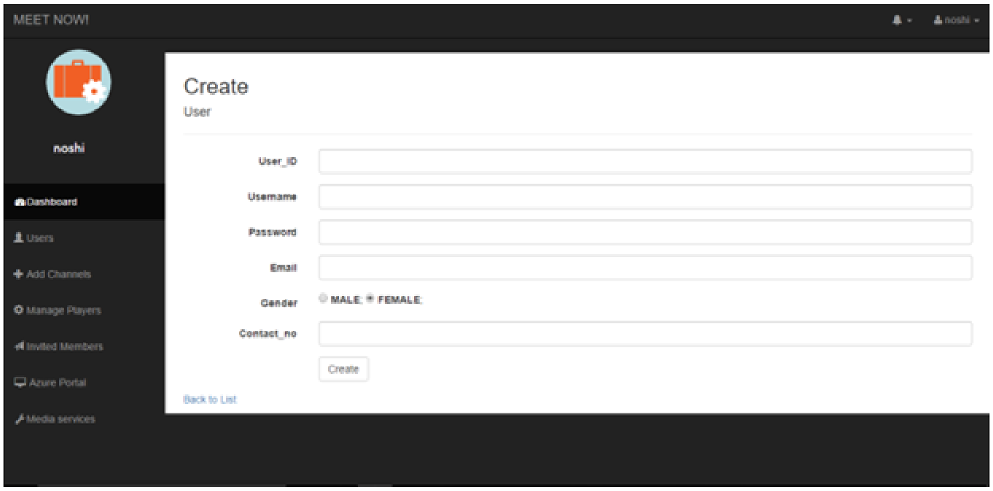


*Figure 11 Adding Users in Dashboard*

## CE 3.3.16

After developing this, I conducted various tests on the "Meet Now" application to assess its effectiveness. Throughout the sessions, I examined a variety of test plans. I started with unit testing, where I looked for faults by concentrating on modules separately. I was able to identify logical and code mistakes in every module. Entering data and determining whether the value matched the type and size that C# supports were part of this testing. Tests are conducted on the different controls to make sure that modules function as intended. I used system testing to test the entire program. As the most important component of any application's success, I looked for user acceptability. By continuously communicating with application users during development and making necessary modifications, the application under consideration is tested for user approval.

## CE 3.3.17

Following testing, I determined that every option in the graphical user interface displayed effectively. After creating new accounts and verifying that the user had entered a working email address and password, the user was signed in. Using the database, the administrator may view, add, remove, and update users. Every detail was displayed in relation to a particular user's ID. By entering accurate channel information, I was able to construct channels. I decided to test this application on many platforms, and every test passed.

## SUMMARY

## CE 3.4

With a web application specifically made for communities, educational institutions, and small and large companies, "Meet now (cloud based web conferencing)" is developed for a variety of platforms. This application's primary goal was to facilitate virtual meetings, cloud-based storage, communication through the sharing of cost-effective voice and video live streams and text messages, maintain stream confidentiality, encode streams with low latency, and offer dynamic packaging. It does not require the installation of software or the setup of hardware devices; instead, it relies only on the user's device capacity and internet connection. It will shield consumers from travel costs and enhance the effectiveness of such organizations.