



# Design Document



## Teams Clone Project

as a part of Microsoft Engage Mentorship Program 2021.

Github : [/shobitnair](#)

Website : [[Link1\(AWS\)](#)] [[Link2\(heroku\)](#)]



Prepared by: Shobit Nair

3rd Year Undergraduate, IIT Ropar



# Introduction

- This design document covers all information needed to understand, make and update the Teams Clone application.
- The application aims in providing a very user-friendly video call and chatting experience for users using any form of device.
- The project covers both the features (ability to video call and Message) demanded in the mentorship program as well as many amazing UI/UX and function features.
- This project aims to produce a public application that can be used by a broad spectrum of people and devices in a user-friendly manner without compromise on any feature.

## Software Overview

- Language and Frameworks used
  - React framework using HTML, CSS, Javascript, JSX
  - Redux for local user state management
- Twilio Programmable Video SDK
  - video, audio, and data streams management
  - for seamless data transfer between a group of users.
- Google Firebase and Firestore database
  - user authentication system
  - for handling user data, messages, channel history
- AWS Amplify console (primary deployment)
- Heroku Cloud Platform (backup deployment)

# Testing Application

*The below are the steps needed to run the application locally on your device and test it.*

- There will be two folders, frontend, and server in the Github repository.
- Clone **only the frontend folder**
- Run **npm install** in CLI to set up the required libraries
- Run **npm start** in CLI after the installation of libraries
- Make sure you have React and Node installed on your device.

## Advanced

- If you want to set up your own Twilio server, you need to follow the documentation given [here](#).
- If you want to set up your own Firebase server, you need to edit the [firebase.js](#) file in the src/ChannelPage folder and add in your own custom API ID and server KEY obtained while making the project in the firebase console.

# Functional Features

- **Channels ( A personal space for a group of users)**
  - User Authentication (using Google Sign-in)
  - Each channel is **password protected**
  - A group of users can **share messages** in the channel.
  - Each user will have their own dashboard with only the channels they are members of. This works even if the user **logs on from another device** as data is stored in firebase.
  - The messages are stored in the firebase and hence you can scroll up to check out any past message communicated to your channels anytime, anywhere.
    - Clicking a message will show an added message details including user Avatar, email, and time of message delivery.
      - **Option to Delete a message**
      - **Option to copy message to clipboard**
- **Host / Join Meetings**
  - This feature **does not require** you to have a google Sign-in
  - **Any user can host and receive a unique hash-key** that can be sent to your friends (**using the inbuilt channel** in the application or via any external social messaging platform).
  - Other users can join the meeting if they enter the valid hash-key
  - Additional meeting features
    - Messaging via WebRTC based Twilio server ( Lasts only during the meeting)
    - Screen Sharing
    - mute / unmute Video and Audio streams
    - Participant Section (keeps track of the participants)
    - Copy to Clipboard feature to easily copy ROOM ID.


# UI/UX Features

- Immense care has been given to UI/UX, Every page and component is **responsive and adjusts to any device**.
- Separate **application layout for mobile devices** with **bottom navigation** to easily translate across sections.
- Subtle animations like **Collapse , Fade , Slide** have been set to make the application look lively.
  - CSS-based **loading animations and progress bars**.
  - Scrolling is **momentum-based** (depends on how strong you pull)
  - **Relevant Toast notifications and alerts**
    - during errors an input of **wrong credentials**
    - when a person **enters/leaves a meeting**
    - many other miscellaneous browser alerts.
- The color combination deeply follows the [Teams Color Pallete](#).
- Flat Vector graphic images have been used to give a minimal and meaningful representation of each page in the application.
- Option to switch between **Dark / Light theme**. (Just click the hovering bubble in the left border of the screen)
- **ToolTips** have been added to each icon for improved user understanding. Although commonly practiced and familiar icons have been used.
- In the Meeting,
  - Videos are **user resizable and scrollable**.
  - **ScreenShare preview** appears in the bottom left under the participant's panel.
  - **Dynamic Message Bubbles**
    - The chat bubbles resize and wrap as per the text content and don't overflow.
    - Sent and received messages have been given unique styles for distinct identification.

# How to use? | Channel

- This is a group chatroom that requires google sign-in authentication to work properly.
- A new user may feel the dashboard a bit empty, Use the "+" (plus) button in the sidebar to open create channel panel.
  - add a channel name and password and press create to create a channel. ( [may throw an error](#) if you use a channel name that already exists )
  - Once the channel is created (it will be initially having 0 participants), [you'll need to join it using the credentials you used while creating it.](#) You can pass on these credentials to other users whom you would like to join your channel.
  - If the credentials are correct, a [new channel will appear in your sidebar](#) and you can communicate with your friends there.
  - There is no limit on the number of channels or messages per user.
- Clicking on messages will show you the [additional message details](#).
- The [host / Join button](#) in the channel header will redirect you to the video meeting feature using Twilio in a new browser tab.
- You can use this channel to share the meeting ID for the users from your channel to join.

# How to use? | Meeting

- This is a video meeting that can be used [without the need for any user authentication](#).
- Host a meeting by just entering your name and you are good to go.
  - you can copy the Meeting ID using the  icon. (this copies the unique hash-key to the clipboard.
  - Send it to your friends using the [channel](#) or via any other external messaging platform.
- Other users must go to the join section and enter this hash-key / Meeting Id to enter the meeting.
- The participant section on the left shows the number of other participants currently in the meeting.
- Explore different functionalities using the icons available in the dock.
- You can [contract/expand video](#) using the corner icons near participant videos. [The videos, participants, and chats are scrollable](#).
- Chat with other users in the meeting using the chat section in the right panel.
  - these messages are not stored in history ([since video calling is available even for users without user authentication](#)), you can parallelly use the channel (with google sign-in) if you want your chats to remain saved.

# Limitations and Info

- As of now only google sign-in is available for the channel feature, [so it is required for the tester to have a google account](#). However, simple video hosting using the Meeting doesn't require google authentication.
- **Preferably use AWS Link to test the application, Heroku cloud platform may sometimes sleep the app if no one is using it to free its server, so there may be a chance that the app may not load on the first try since the dynos are sleeping. However, the app will start running once the server starts. There are chances that Heroku may say the app cant be reached on the first try, but that will get fixed once you refresh and try again in a minute or so.**
- Twilio API limits participants to up to 50 users and costs around 0.004\$ per minute for using its STUN/TURN servers.
  - (Initial balance is around 15\$ )
  - So there will be around 60-70 hours of testing possible with multiple users. [\(which certainly shouldn't get over easily\)](#).
- Try to **end calls using the end call button** , else it may take a minute for the application to figure out the user is disconnected and you may just remain frozen for a minute for other participants before you get disconnected.



# The Sprint

- **Week 1 :**
  - Planning and Design
    - Choosing the best software that meets my requirements.
- **Week 2:**
  - Research and Learning.
  - Develop with minimum features.
- **Week 3:**
  - Adding more functional features.
  - Improving UI / UX
  - Final Polish before last phase.
- **Week 4:**
  - Adapt and Develop the new feature.
  - Test and Review
  - Deploy