Project Documentation of Quarantine&Chill

By

Alex Jackson

20161119

An Assignment Submitted in Partial Fulfilment of The Requirements

For the Course Ctpr490: Information Science Adv. Project

Instructor: Mr. Damion Mitchell

Date: May 7, 2020

Contents

[Project Proposal 4](#_Toc39768742)

[Executive Summary 4](#_Toc39768743)

[Problem Definition 4](#_Toc39768744)

[Solutions Concept 6](#_Toc39768745)

[Deliverables 6](#_Toc39768746)

[Technical Document 7](#_Toc39768747)

[Technical Architecture Design & Environment 7](#_Toc39768748)

[Other Technical Documentation Requirements 10](#_Toc39768749)

[Entity Relationship Diagram 10](#_Toc39768750)

[Data Flow Diagrams 10](#_Toc39768751)

[Source files 10](#_Toc39768752)

[Source files 11](#_Toc39768753)

[Systems Requirements 12](#_Toc39768754)

[Innovation Overview 12](#_Toc39768755)

[Conclusion 12](#_Toc39768756)

[References 13](#_Toc39768757)

[Appendix 13](#_Toc39768758)

Project Proposal

# Executive Summary

This reason for the development for this project was due the lock down of a similar site that carried out similar functions. The development interest spiked when the COVID-19 pandemic was declared.

The pandemic caused a major problem in our regular activities. Persons now must practice social distancing and the pandemic also causes curfews and lockdown of business, such as the movie theatre.

This would prevent people from partaking in social activities, going to the movies and even simply, spending time with each other.

This platform has the solution to the social distance problems by enabling persons to be in a room and watch videos, listen to music, chat and video chat simultaneously.

In the more detailed stage of the technical documentation, expect to see interactive and well-structured system. Also, expect to see solutions concept, deliverables, technical architecture designs & environments, system requirements and innovation overview.

# Problem Definition

The COVID-19 pandemic has impacted effect of the problems we faced before, and those problems include being able to interact without being physically together. Another problem is that travel is restricted, and persons are confined to their homes more than ever, they cannot go out and watch movies with friends because the movie theatres are closed. The system wholeheartedly, solves the problem of social distancing.

The initial decision to undertake this project came from seeing people, especially couples in, long-distance relationships; find bonding than just video and voice calling. However, with the effect of the COVID-19 pandemic. It is because of the COVID-19 this problem occurs AND it occurs when a group or groups of individuals in different locations wants watch videos together.

There are similar systems available which solves the same problem. Zoom is a collaboration tool which allows users to video chat, present and share screen content. Google meet is another collaboration platform that allows users to video chat and share screen content.

Improvements to the current solutions are being able to watch videos without depending on the share screen module. This method is lags much and depends directly on all parties to present quality content.

# Solutions Concept

The project will achieve the goals of bring people together and allowing users to watch videos, send messages and video chat simultaneously. The system will have a room, where the users interact with each other. The room will have video controls, chat section and an add friends to room section. The intended users are teenagers, adults, parents and their children and couples.

# Deliverables

The project will have watch rooms.

People will be able to join the room. The room has two built-in players, YouTube and HTML5, along with video, voice and text chatting.

The player allows the host to choose any show of any choice they want to watch.

The voice and video call that is incorporated allows users to react and respond like you would in person.

the project achieves – bring people together during the COVID-19 pandemic.

Technical Document

# Technical Architecture Design & Environment

The online streaming application will be develop using the language JavaScript. The development tools will be socket.io, node Js, Express framework for Node.js, YouTube data API and WebRTC (for video, voice and messaging).

Diagram Showing video syncing software technical Architecture.

A screenshot of a cell phone

Description automatically generated

A major part of the system functionality focuses on web sockets, Socket.io specifically. It works by using a socket is created when a client is connects to the server. The room host will enter the room name. The input from the host is then sent back to the server and it creates a room with the Socket.io. The host socket is marked when a room is created which allows users to join and interact with the users in the said room. The host socket manages the video information and distributes it any new socket that enters the room.

Socket.io, node.js, express framework, WebRTC YouTube data APIs is used as the frontend of the application and html5 css7 and JavaScript will be used as the backend.

The implementation issues and challenges are displaying all active rooms in real time. Showing active rooms, because active rooms are separate from the sockets. Another implementation of use the webserver to manage the bandwidth used for video, voice and streaming instead of having the device handling it and I think that will be another challenge. Finally, to stream video chat content through the web sockets.

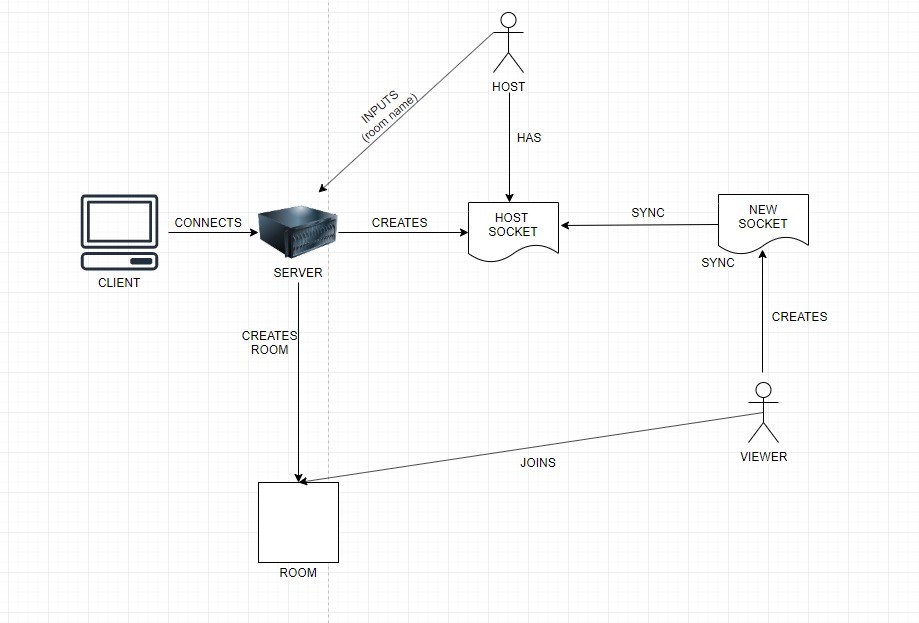


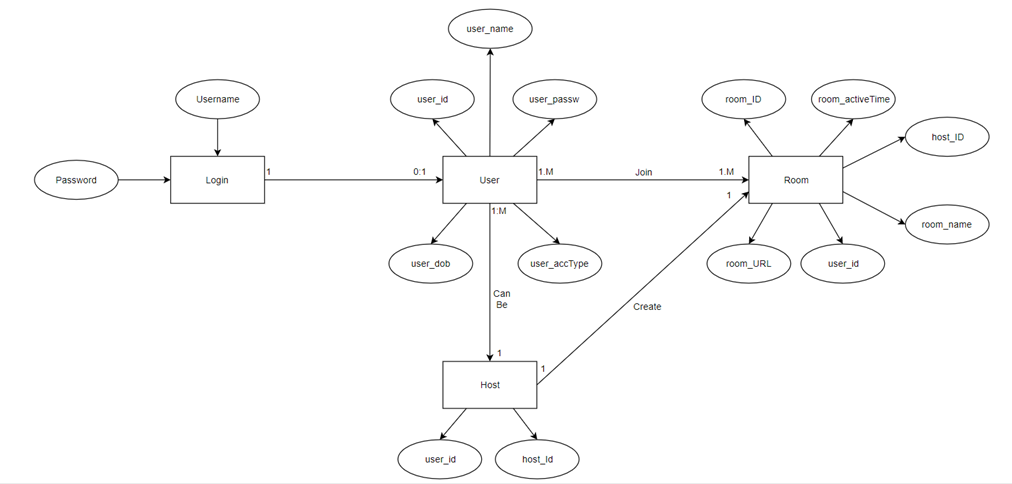
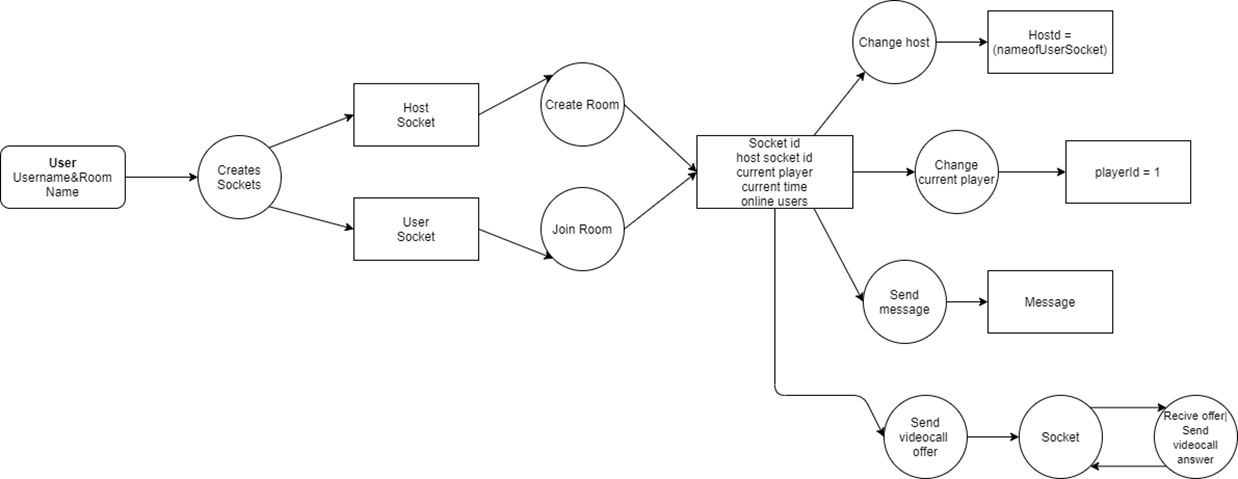
Diagram showing system functionality.

# Other Technical Documentation Requirements

### Entity Relationship Diagram

### Data Flow Diagrams

### Source files



**Diagram Showing Data Flow**

**Diagram showing the Entity Relationship Diagram**

## Source files

Animate.css – gives access to common animations such as fadeIn, fadeOut, bounce, flash, tada, and other animation styles.

Bundle.js – webpack that holds the Js for video chatting

Events.js - functions play or pause the player Created for event listeners

Function.test.js – this is the file that holds function tests.

Host.js – This manipulates the hosts, such as set host, change host, auto host

Html5.js – event listener for HTML 5 player

html5player.css – stylesheet for html 5

Index.html

Notify.js - Holds the popup notifications functions

Mobile-style.css – stylesheet for mobile phones

Small-business.css – default stylesheet

Player.js – holds the player functions

Packet.json – holds the info for the program

Package-lock.json - holds in depth info for the program

Sync.js – enables the players to be sync

Server.js – contains the server files for the project

Yt.js – event listener for the YouTube player

# Systems Requirements

Users on any up to date browser.  
Webcam and microphone

# Innovation Overview

The uniqueness of this project stems from, being able to transmit video feed and sync up with players without having to share screen. This method is independent of the party’s that are viewing it unlike other platforms.

The development was done using mainly node Js and other notable modules such as socket.io, express, moment, simple peer, expect and mocha. The functionalities that these modules provide puts an edge over traditional websites.

# Conclusion

With the presence of the COVID-19 pandemic the Quarantine&Chill app was created to enable people to be social, while practicing social distance.

The pandemic caused a major problem in our regular activities. Persons now must practice social distancing and the pandemic also causes curfews and lockdown of business.

Persons are prevented from partaking in social activities, going to the movies and even simply, spending time with each other.

This platform has the solution to the social distance problems by enabling persons to be in a room and watch videos, listen to music, chat and video chat simultaneously.

# References

Web Security Readings, by Ferruh Mavituna Why Framework Choice Matters in Web Application Security, accessed on 11/09/2019 [Online] https://www.netsparker.com/blog/web-security/why-framework-choice-matters-in-web-application-security/

VB, by DEAN TAKAHASHI, Rabbit lets you remotely watch online videos with your friends, accessed on 11/09/2019 [Online], https://venturebeat.com/2018/11/08/rabbit-lets-you-remotely-watch-online-videos-with-your-friends/

# Appendix

Promote Clear Usage

Include all the words needed to avoid ambiguity for a person reading code where the name is used.

Name variables, parameters, and associated types according to their roles, rather than their type constraints.

Protocols that describe what something is should read as nouns (e.g. Collection).

Protocols that describe a capability should be named using the suffixes able, ible, or ing (e.g. Equatable, ProgressReporting).

The names of other types, properties, variables, and constants should read as nouns.