

Team Group 11

TiaoTiaoXiong TV Web Application

Software Design Document

Names: Hongkai Huang, Chengkai Yang

Lab Section: CmpE 131 Sec-02

Workstation: ...

Date: (03/25/2021)

TABLE OF CONTENTS

1. INTRODUCTION	2
1.1 Purpose	2
1.2 Scope	2
1.3 Overview	2
1.4 Reference Material	2
1.5 Definitions and Acronyms	2
2. SYSTEM OVERVIEW	3
3. SYSTEM ARCHITECTURE	3
3.1 Architectural Design	3
3.2 Decomposition Description	4
3.3 Design Rationale	5
4. DATA DESIGN	5
4.1 Data Description	5
4.2 Data Dictionary	5
5. COMPONENT DESIGN	6
6. HUMAN INTERFACE DESIGN	7
6.1 Overview of User Interface	7
6.2 Screen Images	8
6.3 Screen Objects and Actions	8
7. REQUIREMENTS MATRIX	9

1. INTRODUCTION

1.1 Purpose

This software design document describes the architecture and system design of TiaoTiaoXiong TV Web Application. This document will be offered to software developers who will implement this web app design. It will be easy for developers to understand the software design of the application.

1.2 Scope

This web application is designed for customers who want to share their movies during COVID-19 age. It will provide users a platform which allows users to watch and share movies with their friends and family. It is good for human relationship that was built on internet/software but focused on humanity

1.3 Overview

This document is about software design, and it contains 7 parts, which is introduction, system overview, system architecture, data design, component design, human interface design and Matrix requirements. Each part will help developers to better understand the software design of the application.

1.4 Reference Material

N/A

1.5 Definitions and Acronyms

Front-end: The frontend of a software program or website is everything with which the user interacts.

Back-end: The server, application and database that work behind the scenes to deliver information to the user.

UI - User Interface

2. SYSTEM OVERVIEW

This web app system shall be used by customers who want to share their movies/videos to their family or friends during COVID-19 age. Because people are isolated at their home alone, people need some interpersonal interaction through some online entertainment. Watching movies shall be a nice way but theaters are closed due to quarantine.

Functionality: The web app can generate a media room with a unique name and a password upon users' request; Users shall be able to join a media room after they enter the correct name and password; Users shall be able to quit the media room while they are in one; The web app shall be able to receive media files uploaded by users and stream it in the media room; The web app shall be able to display the list of media files uploaded in the media room while in that room; The web app shall be able to play or delete the media files uploaded in the media room depends on users' request.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design

The software's architecture will split into 2 big parts, front-end and back-end, they will connect each other with a Restful HTTP connection.

Front-end provides a user-friendly interface, grand user's input and return feedback. It will be divided into 4 main subsystems, connection handler, data handler, account handler, media player. In front architectural. Data handler will handle most of the data transfer between subsystems. And all of the requests to the backend are sent to the connection handler to process. Connection handler also sends the response from the backend to each subsystem to process.

Backend provides a restful interface for frontend to access, it manages the logic of data flow in this application. It reads the requests from the front-end and sends back responses, it is also in charge of the database access. It acts as a middleware between the user interface and datas. It can be decomposed into connection handler and database handler

3.2 Decomposition Description

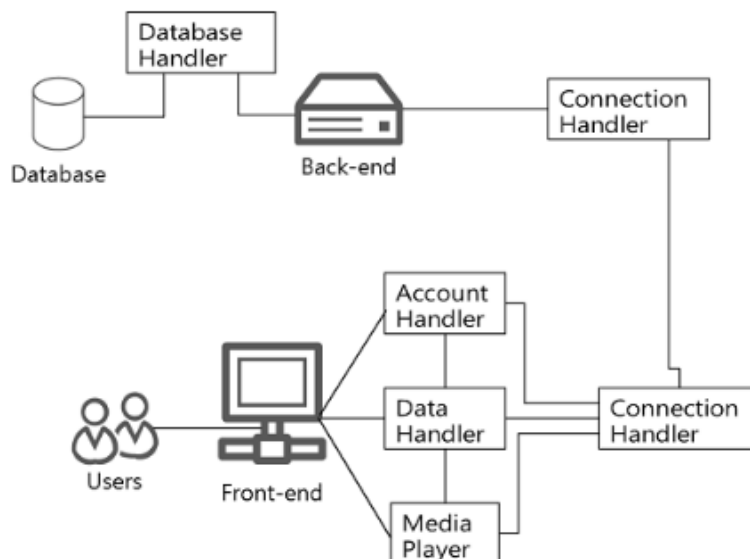
The connection handler manages all the connections to the backend, including handling RESTful requests and responses and media stream socket. This component makes sure the communication between client and server are modular and standardized, so that the application can be more maintainable and stable.

The data handler acts the same way as the connection handler in its purpose. But in terms of function, it handles data of user, room and media. Data will be accessed and displayed by user interface and served to the user.

Account handler helps to manage the user's operation on the user account. This subsystem will handle user's login and logout, account creation and secure authentication

Connection handler has very similar functionality as the connection handler in front-end

Database handler manages database's CRUD and response to frontend's request correspondingly.



3.3 Design Rationale

We are using the Model-View-Controller(MVC) architectural for our web application.

Model–view–viewmodel(MVVM): a software architectural pattern that facilitates the separation of the development of the graphical user interface. It is a popular architecture, we didn't choose it because it is over complex for our web application. It is hard to read.

4. DATA DESIGN

4.1 Data Description

This application Contains 3 data models.

1. User: distinguish users helps the authentication process and it is an important part of the account system.
2. room: the function of application is to ,
3. media file: users rely on uploading videos and storing in our service. It is a big part of functions.

Database will contain three tables.

The first one is User, it stores user id, username, password and token. Second one stores each room's name, password and a unique ID. The third one stores each media file uploaded with the ID of the room it belongs and its stored

4.2 Data Dictionary

String Password

String RoomID

String Username

String Text

displayMainPage();

uploadVideo();

```
displayVideo();
showFullScreen();
inviteFriends();
showSubtitles();
showPlayList();
changeVolume();
VideoList();
sendChat();
EnableVoiceChat();
DisableVoiceC();
SignIn();
SignOut();
```

5. COMPONENT DESIGN

The connection handler: manages all the connections to the backend, including handling restful requests and responses and media stream socket. This component makes sure the communication between client and server are modular and standardized, so that the application can be more maintainable and stable.

The data handler: acts the same way as the connection handler in its purpose. But in terms of function, it handles data of user, room and media. Data will be accessed and displayed by user interface and served to the user.

Main page:

If userA selects display video

 Load the video into display window

Else if userA selects Upload

 Upload video into Playlist

Else if userA selects Play/Pause

 Play or Pause the video

Else if userA selects Resolution

 Change the resolution and fps

- Else if userA selects subtitles
 - Display subtitles
- Else if userA selects volume
 - Change volume
- Else if userA selects view in full screen
 - Enlarge display window to full screen
- Else if userA selects voice chat
 - Enable voice chat
- Else if userA selects send text
 - Send text in chat
- Else if userA selects a title in Playlist
 - Load the video into display window
- Else if userA selects Friends
 - Show list of friends
 - If userA selects invite someone in the list
 - Invite friends to the room or copy the link of the room

Account handler: helps to manage the user's operation on the user account. This subsystem will handle user's login and logout, account creation and secure authentication

- Login page:
 - Username = "hzwz"
 - Password = "12345"
 - If userA chooses Sign_in
 - Login user
 - Enter main user page (media room)
 - Else if userA chooses Sign_out
 - Logout user

6. HUMAN INTERFACE DESIGN

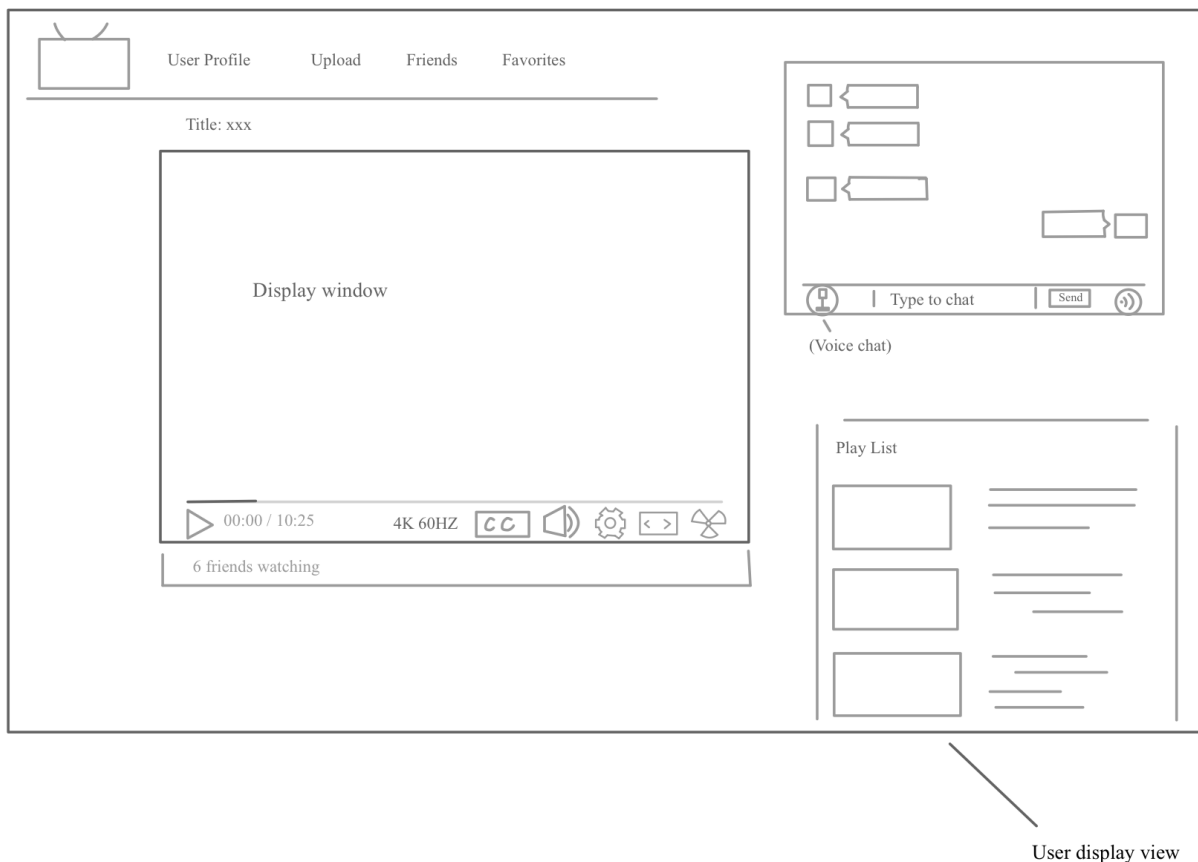
6.1 Overview of User Interface

The front end of our web app contains two pages, which is the login page and media room page. Pages use JavaScript to send requests and listen to responses. And the media player components load the media file from the path contained in the response. Here

users shall be able to create a media room with a unique name and a password; Users shall be able to join a media room after they enter the correct name and password; Users shall be able to quit the media room while they are in one; Users should be able to upload media files to the media room while in the room; Users should be able to see the list of media files uploaded in the media room while in that room; Users should be able to play or delete the media files uploaded in the media room while in that room.

6.2 Screen Images

Main user interface (PC web end)



6.3 Screen Objects and Actions

In the display window, these buttons for Display/Pause, resolution and fps choices, subtitles, volume, video setting, view in widescreen and in full screen.

Top side of user view is for user account management and social interaction.

Top right is a chat window for text chat and voice chat while watching video with friends.

Bottom right is the Playlist showing video titles the user uploaded.

7. REQUIREMENTS MATRIX

Functional Requirements (From SRS)	System Components (To satisfy Functional Requirements)
A. Users shall be able to create a media room with a unique name and a password.	Account handler
B. Users shall be able to join a media room after they enter the correct name and password.	Account handler Data handler
C. Users shall be able to quit the media room while they are in one.	Connection handler Account handler
D. Users should be able to upload media files to the media room while in the room.	Data and database handler Connection handler
E. Users should be able to see the list of media files uploaded in the media room while in that room.	Data and database handler
F. Users should be able to play or delete the media files uploaded in the media room while in that room.	Data and database handler Connection handler