

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**ARCHITECTURAL DESIGN SPECIFICATION
CSE 4316: SENIOR DESIGN I
FALL 2021**



TEAM NAME : THE STREAMERS

PRODUCT NAME : SWaP - Social Watch Party

**Manish Phuyal
Lahana Maharjan
Subash Sunar
Sunbash Dahal
Subash Shahi
Rajkumar Jirel**

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	08.01.2021	MT,LH,SS,SS,SD,RJ	document creation
0.2	08.16.2021	MT,LH,SS,SS,SD,RJ	complete draft

CONTENTS

1	Introduction	4
2	System Overview	5
2.1	User end Layer Description	5
2.2	Back end Layer Description	5
2.3	Database layer Description	5
3	Subsystem Definitions & Data Flow	6
4	User end Layer Subsystems	6
4.1	User Interface	7
4.2	User Input	8
5	Back End Layer Subsystems	9
5.1	User Account Manager	11
5.2	Chats	12
5.3	Streaming Rooms	13
5.4	Video Streaming	14
5.5	Group Manager	16
6	Database Layer Subsystems	17
6.1	User	18
6.2	Videos	19
6.3	Chat History	20

LIST OF FIGURES

1	A simple architectural layer diagram	5
2	A simple data flow diagram	6
3	Example subsystem description diagram	7
4	Example subsystem description diagram	10
5	Example subsystem description diagram	11

LIST OF TABLES

2	Subsystem interfaces	13
3	Subsystem interfaces	14
4	Subsystem interfaces	15
4	Subsystem interfaces	15

1 INTRODUCTION

The SWrap (Social Watch Party) is a social video watching app designed to connect people online in order to stream online videos from platforms like netflix, youtube and more, together with the connected friends. In this app, a user logs in into their profile and starts streaming videos from netflix or youtube (may add more platforms later) and they can also add or invite their friends using email address, once they are able to add a friend then they can stream the videos together including chatting at the same time. It is for entertainment purposes. The purpose of the project is to provide a platform for people to entertain and socialize digitally through streaming social videos online together with friends or strangers from anywhere in the world with connection to the internet.

Video chat is future features that could be added when all the other functions are completed successfully.

2 SYSTEM OVERVIEW

SWaP is a mobile application that is compatible with android smartphones. SWaP will consist of 3 layers which will include User-End Layer, Back-End Layer, and the Database Layer. In the User-End layer, the user interacts directly with on-click components, which directs to the back end layers. Back- end layer acts as a mediator between the user layer and the database layer. Back-end gets the necessary information from the database and displays it to the user.

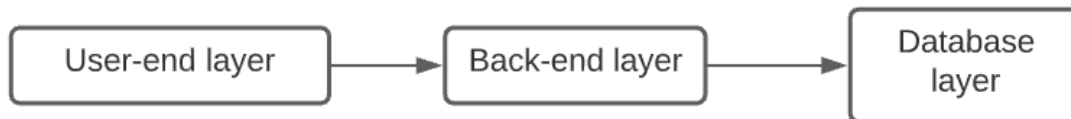


Figure 1: A simple architectural layer diagram

2.1 User-end Layer

User-end layer is displayed to the user in the app. The user-end layer interacts directly with the user. User-end layer or front-end layer is what users can view and interact with. Users will be able to provide the necessary inputs and interact with the user interface in this layer, which will be redirected to the back-end layers and database to provide necessary output to the screen.

2.2 Back-end Layer

Back-end layer is where all the processes take place but are not displayed to the user. This layer gets inputs from the user-end layer, and processes the data and requests from the database if necessary and then displays the output to the user-end layer.

2.3 Database Layer

The database layer stores the user information and returns it back to the user through the back-end layer when requested. This layer is also responsible for keeping record of and returning it back to the back end layer such as; groups of friends, their chats either private or a group chat, list of videos.

3 SUBSYSTEM DEFINITIONS & DATA FLOW

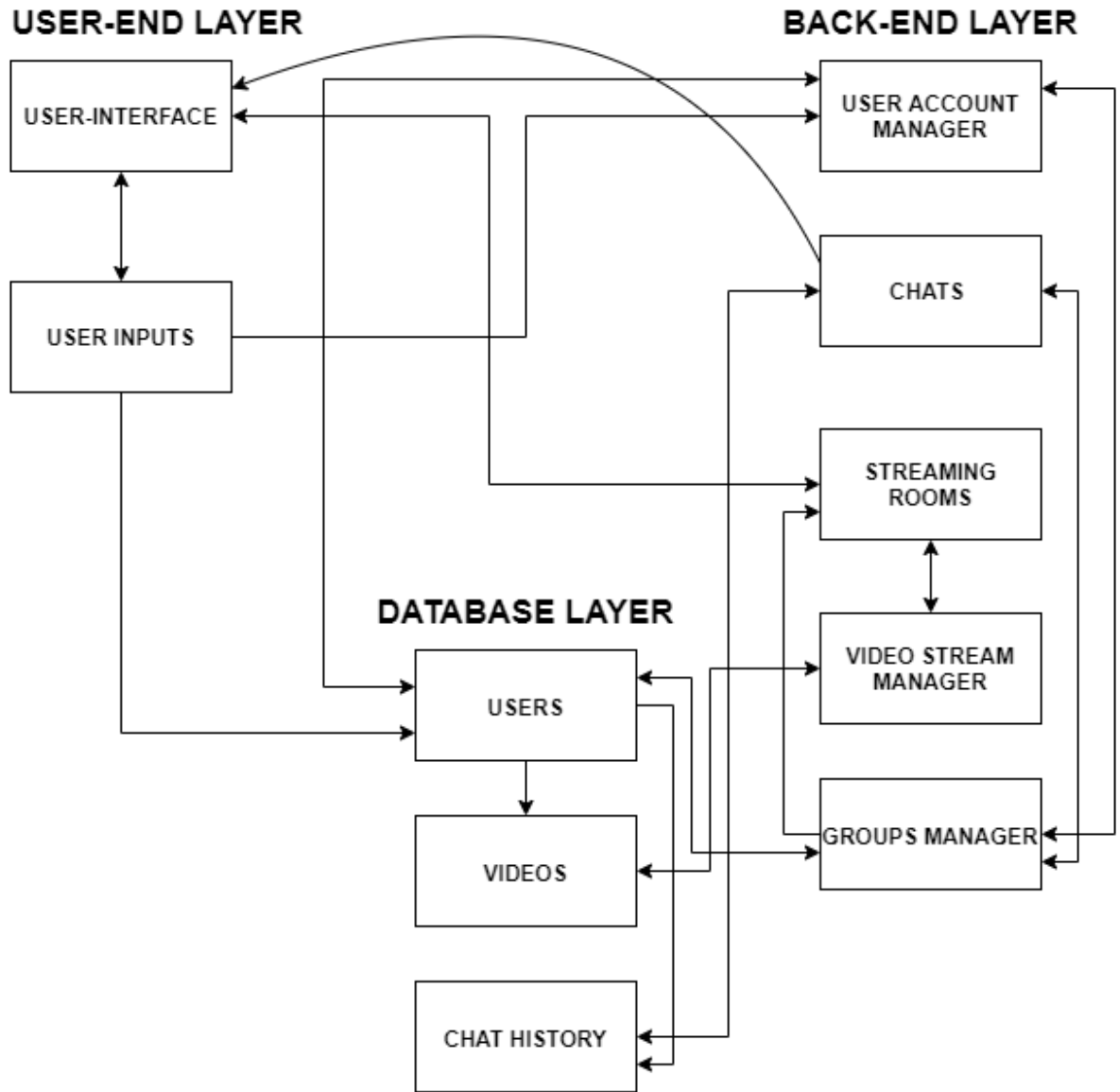


Figure 2: A simple data flow diagram

4 USER-END LAYER

The user end layer deals with the inputs from the user and processes them to the back-end layer and database layer and is responsible for displaying appropriate output. The user inputs can be information typed by the user or any click events.

4.1 USER INTERFACE

This is a GUI for the android application that allows users to interact with the content displayed. Any input or click event is captured and sent to the back-end layer for further processing and the result received is displayed on the screen. For instance, on the login page that is displayed at the very first, users are able to enter information and click a button which will be captured and sent to the backend and database for verification and if valid, as a result the user interface opens a new page.

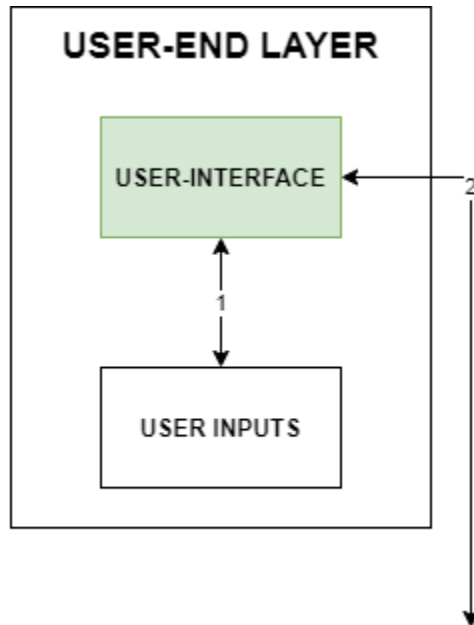


Figure 3: Example subsystem description diagram

4.1.1 ASSUMPTIONS

The user's click is recorded only after clicking on a clickable object.
"Enter" can be used as a click whenever the object is on focus.

4.1.2 RESPONSIBILITIES

This subsystem is responsible for allowing users to interact within the displayed page, send the requests of the user to the back-end and provide response to every user interaction.

4.1.3 SUBSYSTEM INTERFACES

Table 1: Subsystem interfaces

ID	Descriptions	Inputs	Outputs
1	Sends the inputs from a user to back-end and displays response to the user according to the result received from backend	Valid inputs from user after they perform Touch/Click/Enter	Display message or a interactive page

4.2 USER INPUT

Any information typed by the user for logging in or registering are user inputs and the event of a button click is also an user input which triggers the interface to send the provided input to the back-end for further process.

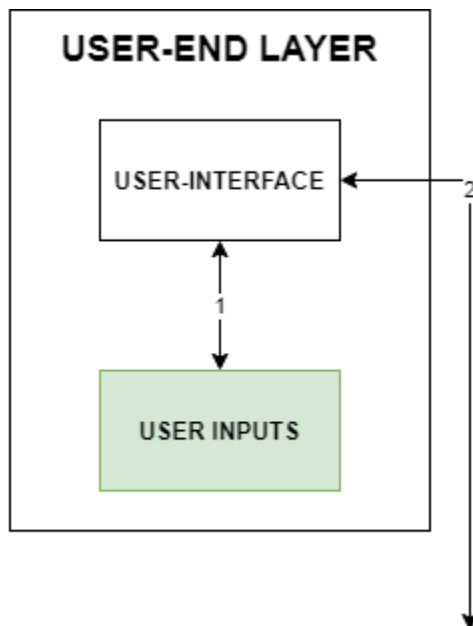


Figure 4: Example subsystem description diagram

4.2.1 ASSUMPTIONS

All the input data provided is valid.

User clicks/touches/enters a valid clickable object.

4.2.2 RESPONSIBILITIES

This subsystem is responsible for the next event to occur by providing an input from the user to the user interface which sends that information to the back end to receive an appropriate response.

4.2.3 SUBSYSTEM INTERFACES

Table 2: Subsystem interfaces

ID	Descriptions	Inputs	Outputs
1	Allows for different event to occur within the application	Touch/Click/Enter	information sent to back end layer

5 Back-End Layer

In this layer, the inputs received from the user-end layer will be processed and necessary information will be fetched from the database and displayed to the user interface. It is composed of five layers, which are User Account Manager, Chats, Streaming Rooms, Video Stream Manager, Groups Manager. When the user login with their username and password, the back-end checks with the database if the information is valid or not and then if the information is correct, it will login the user to the home page.

BACK-END LAYER

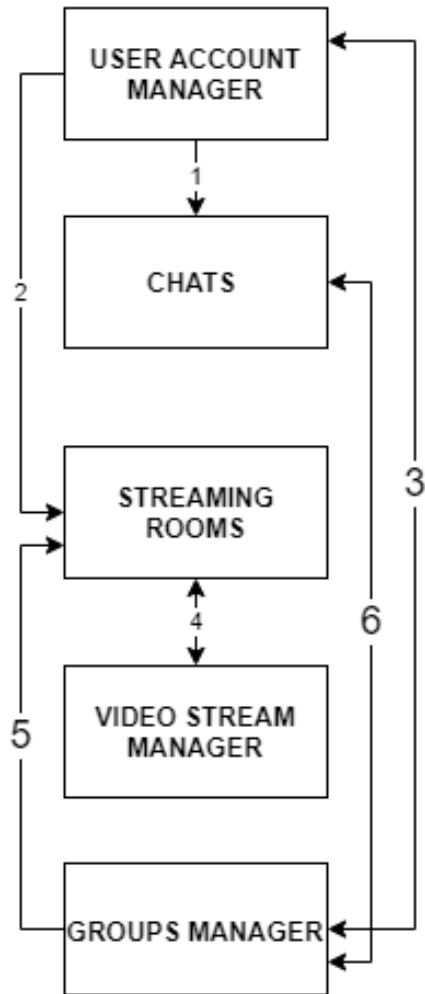


Figure 5: Example subsystem description diagram

5.1 User Account Manager

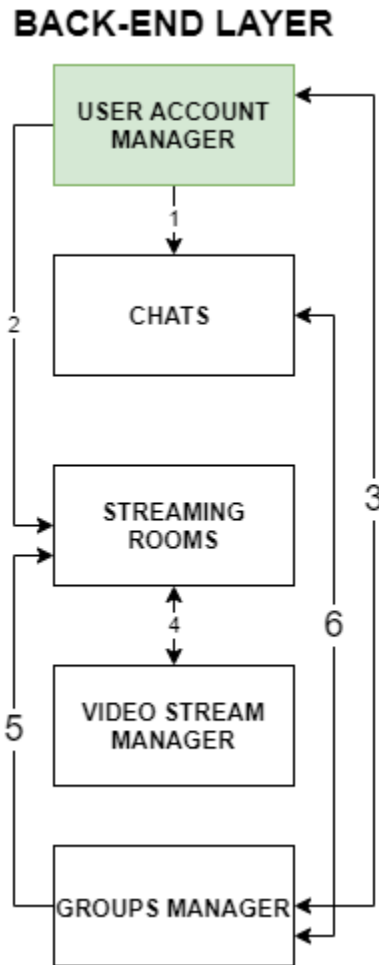


Figure 6: Example subsystem description diagram

5.1.1 ASSUMPTIONS

The user has logged in to the app using their username and password.

5.1.2 RESPONSIBILITIES

1. The user account manager validates if the user information is accurate.
2. Keeps the user logged in and active within the app.

5.1.3 SYSTEM INTERFACES

Table 3: Subsystem interfaces

ID	Descriptions	Inputs	Outputs
1	Authentication of sign-up and login	User inputs their email, username and password	Verifies the information and logs in the user into

			the app.
--	--	--	----------

5.2 CHATS

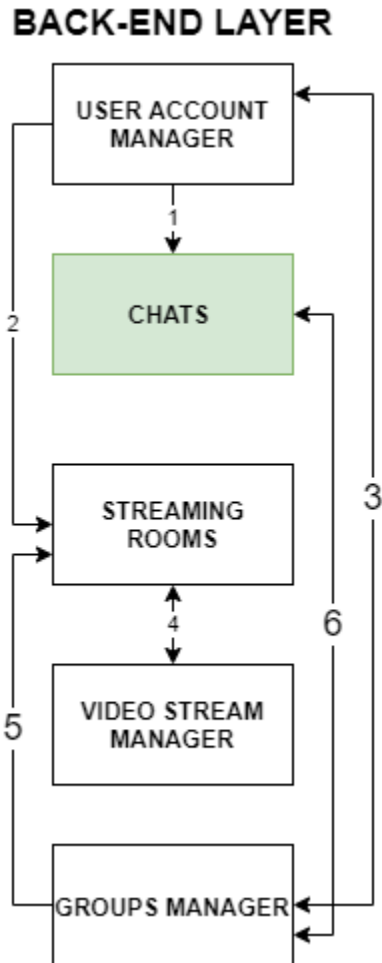


Figure 7:Example subsystem description diagram

5.2.1 ASSUMPTIONS

The users are friends to each-other or in the same watch party.

5.2.2 RESPONSIBILITIES

None

5.2.3 SYSTEM INTERFACES

Table 4: Subsystem interfaces

ID	Descriptions	Inputs	Outputs
1	Users are allowed to chat with friends and everyone in the same watch party.	text messages, symbols, emojis	Visible to everyone in the group.

5.3 STREAMING ROOMS

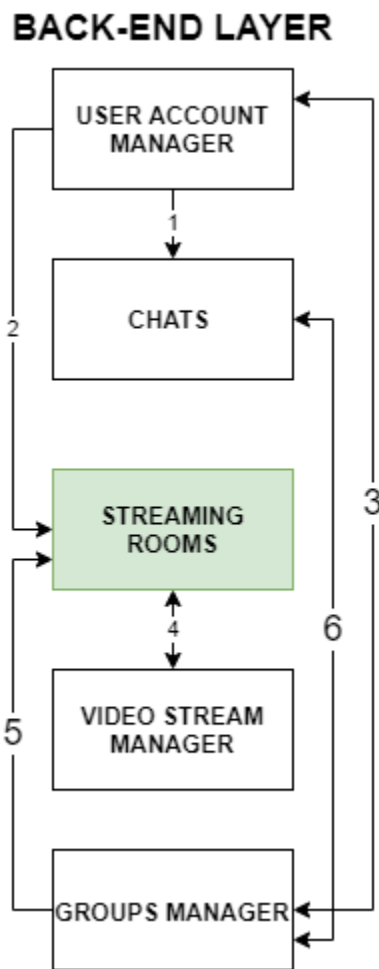


Figure 7: Example subsystem description diagram

5.3.1 ASSUMPTIONS

The users are logged into the app.

5.3.2 RESPONSIBILITIES

1. Let the users add new friends or remove friends in the streaming room.
2. Play/pause the video.
3. Restart the video.

5.3.3 SYSTEM INTERFACES

Table 5: Subsystem interfaces

ID	Descriptions	Inputs	Outputs
1	Add/ remove friends	username of the users logged in to the app	Users will be added to/removed from the video streaming group.

5.4 VIDEO STREAM MANAGER

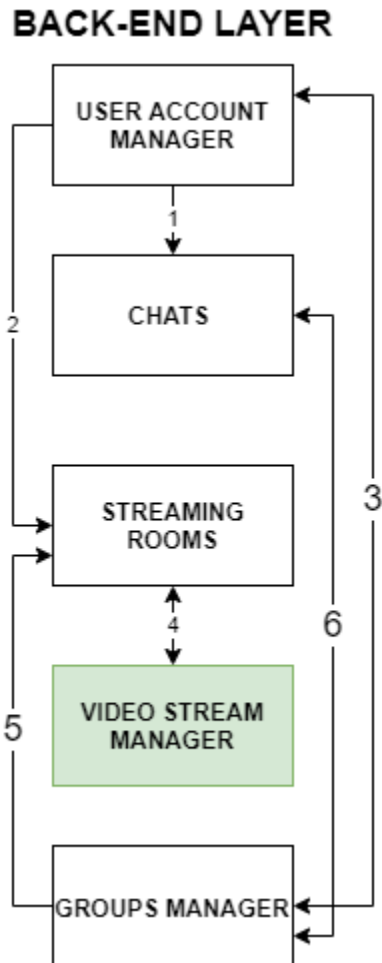


Figure 8:Example subsystem description diagram

5.4.1 ASSUMPTIONS

1. The users are logged into the app.
2. Users are in the room to stream the video.
3. Users either choose to play video from videos stored in the database or use a search engine such as youtube or netflix.

5.4.2 RESPONSIBILITIES

1. Searches the database if the requested video has been watched before by other users.
2. Uses a search engine to search the videos requested from the user-end layer.
3. Update database with the video for the video history.

5.4.3 SYSTEM INTERFACES

Table 6: Subsystem interfaces

ID	Descriptions	Inputs	Outputs
1	Requested video will be displayed	video requests from the user-end layer	displays the list of similar videos watched before

5.5 GROUP MANAGER

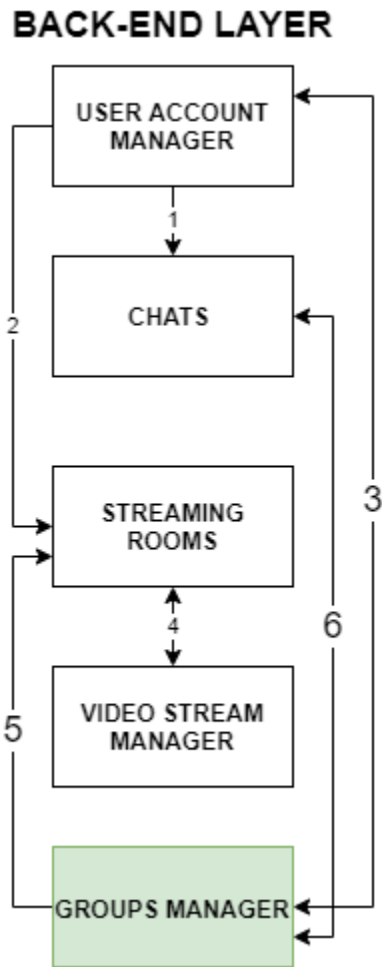


Figure 9:Example subsystem description diagram

5.5.1 ASSUMPTIONS

1. Users are in the groups.

5.5.2 RESPONSIBILITIES

1. Records the list of users of the group in the database.
2. Records the chat and videos watched in the database.

5.5.3 SYSTEM INTERFACES

Table 7: Subsystem interfaces

ID	Descriptions	Inputs	Outputs
1	Users accepted in the room	Username of the users for the group	Users of the same group are stored in the database.

6 DATABASE LAYER

This layer validates the user info and returns information from the database such as user info, chat history and videos when requested.

DATABASE LAYER



Figure 10: Example subsystem description diagram

6.1 USER

This layer keeps a record of all the registered users. It is used for login authentication.

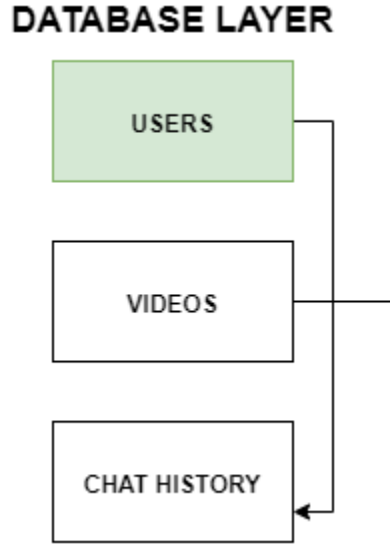


Figure 11: Example subsystem description diagram

6.1.1 ASSUMPTIONS

The information in the database will be stored as soon as a user registers.

Any changes in the information will be updated in real time.

6.1.2 RESPONSIBILITIES

The user subsystem is responsible for storing users complete information securely. Updating any changes in real time. Authenticating a new login with the provided credentials.

6.1.3 SUBSYSTEM INTERFACE

Table 8: Subsystem interfaces

ID	Descriptions	Inputs	Outputs
1	keeps record of user information and helps in authenticating a user .	user information user credentials	store: user information Boolean: true or false

6.2 VIDEOS

This subsystem stores a list of videos and provides the list of videos when requested by the video manager.

DATABASE LAYER

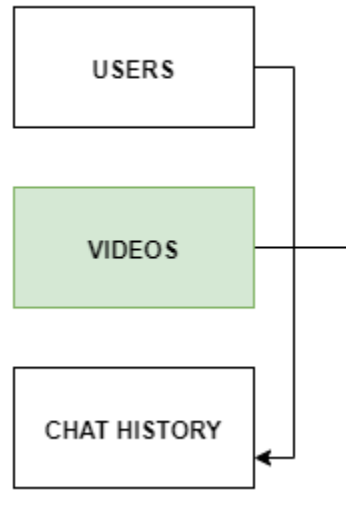


Figure 12: Example subsystem description diagram

6.2.1 ASSUMPTIONS

A reference to the video is provided.

6.2.2 RESPONSIBILITIES

It is responsible for providing video to the video manager which can be streamed by the user.

6.1.3 SUBSYSTEM INTERFACE

Table 9: Subsystem interfaces

ID	Descriptions	Inputs	Outputs
1	Provides video to the video manager	reference to the video(phrase can be used)	Video is available to stream

6.3 CHAT HISTORY

DATABASE LAYER

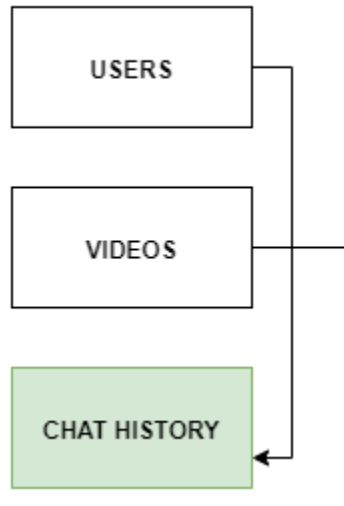


Figure 13:Example subsystem description diagram

6.3.1 ASSUMPTIONS

chatting has occurred between users.

6.3.2 RESPONSIBILITIES

This subsystem is responsible for saving and retrieving messages between the users.

6.3.3 SUBSYSTEM INTERFACE

Table 10: Subsystem interfaces

ID	Descriptions	Inputs	Outputs
1	saving and retrieving messages	messages	save/retrieve (boolean: true or false)