

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

**DETAILED DESIGN SPECIFICATION
CSE 4317: SENIOR DESIGN II
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	10.17.2021	MT,LH,SS,SS,SD,RJ	document creation
0.2	10.24.2021	MT,LH,SS,SS,SD,RJ	complete draft

CONTENTS

REVISION HISTORY

CONTENTS

1 INTRODUCTION

2 SYSTEM OVERVIEW

2.1 USER-END LAYER

2.2 BACK-END LAYER

2.3 DATABASE LAYER

3 USER-END LAYER SUBSYSTEMS

3.1 USER-END LAYER HARDWARE

3.2 USER-END LAYER OPERATING SYSTEM

3.3 USER-END LAYER SOFTWARE DEPENDENCIES

3.4 USER INPUT SUBSYSTEM

3.4.1 SUBSYSTEM HARDWARE

3.4.2 SUBSYSTEM OPERATING SYSTEM

3.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

3.4.4 SUBSYSTEM PROGRAMMING LANGUAGES

3.4.5 SUBSYSTEM DATA STRUCTURES

3.4.6 SUBSYSTEM DATA PROCESSING

4 BACK-END LAYERS SUBSYSTEMS

4.1 LAYER HARDWARE

4.2 LAYER OPERATING SYSTEM

4.3 LAYER SOFTWARE DEPENDENCIES

4.4 USER ACCOUNT MANAGER SUBSYSTEM 1

4.4.1 SUBSYSTEM HARDWARE

4.4.2 SUBSYSTEM OPERATING SYSTEM

4.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

- 4.4.4 SUBSYSTEM PROGRAMMING LANGUAGES
- 4.4.5 SUBSYSTEM DATA STRUCTURES
- 4.4.6 SUBSYSTEM DATA PROCESSING
- 4.5 CHATS SUBSYSTEM 1
 - 4.5.1 SUBSYSTEM HARDWARE
 - 4.5.2 SUBSYSTEM OPERATING SYSTEM
 - 4.5.3 SUBSYSTEM SOFTWARE DEPENDENCIES
 - 4.5.4 SUBSYSTEM PROGRAMMING LANGUAGES
 - 4.5.5 SUBSYSTEM DATA STRUCTURES
 - 4.5.6 SUBSYSTEM DATA PROCESSING
- 4.6 STREAMING ROOM SUBSYSTEM 1
 - 4.6.1 SUBSYSTEM HARDWARE
 - 4.6.2 SUBSYSTEM OPERATING SYSTEM
 - 4.6.3 SUBSYSTEM SOFTWARE DEPENDENCIES
 - 4.6.4 SUBSYSTEM PROGRAMMING LANGUAGES
 - 4.6.5 SUBSYSTEM DATA STRUCTURES
 - 4.6.6 SUBSYSTEM DATA PROCESSING
- 4.7 VIDEO STREAM MANAGER SUBSYSTEM 1
 - 4.7.1 SUBSYSTEM HARDWARE
 - 4.7.2 SUBSYSTEM OPERATING SYSTEM
 - 4.7.3 SUBSYSTEM SOFTWARE DEPENDENCIES
 - 4.7.4 SUBSYSTEM PROGRAMMING LANGUAGES
 - 4.7.5 SUBSYSTEM DATA STRUCTURES
 - 4.7.6 SUBSYSTEM DATA PROCESSING
- 4.8 GROUP MANAGER SUBSYSTEM 1
 - 4.8.1 SUBSYSTEM HARDWARE
 - 4.8.2 SUBSYSTEM OPERATING SYSTEM
 - 4.8.3 SUBSYSTEM SOFTWARE DEPENDENCIES

4.8.4 SUBSYSTEM PROGRAMMING LANGUAGES

4.8.5 SUBSYSTEM DATA STRUCTURES

4.8.6 SUBSYSTEM DATA PROCESSING

5 DATABASE LAYER

5.1 LAYER HARDWARE

5.2 LAYER OPERATING SYSTEM

5.3 LAYER SOFTWARE DEPENDENCIES

5.4 USER SUBSYSTEM 1

5.4.1 SUBSYSTEM HARDWARE

5.4.2 SUBSYSTEM OPERATING SYSTEM

5.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

5.4.4 SUBSYSTEM PROGRAMMING LANGUAGES

5.4.5 SUBSYSTEM DATA STRUCTURES

5.4.6 SUBSYSTEM DATA PROCESSING

5.5 VIDEOS SUBSYSTEM 1

5.5.1 SUBSYSTEM HARDWARE

5.5.2 SUBSYSTEM OPERATING SYSTEM

5.5.3 SUBSYSTEM SOFTWARE DEPENDENCIES

5.5.4 SUBSYSTEM PROGRAMMING LANGUAGES

5.5.5 SUBSYSTEM DATA STRUCTURES

5.5.6 SUBSYSTEM DATA PROCESSING

5.6 CHAT SUBSYSTEM 1

5.6.1 SUBSYSTEM HARDWARE

5.6.2 SUBSYSTEM OPERATING SYSTEM

5.6.3 SUBSYSTEM SOFTWARE DEPENDENCIES

5.6.4 SUBSYSTEM PROGRAMMING LANGUAGES

5.6.5 SUBSYSTEM DATA STRUCTURES

5.6.6 SUBSYSTEM DATA PROCESSING

6 APPENDIX A

SUBSYSTEM DEFINITIONS & DATA FLOW

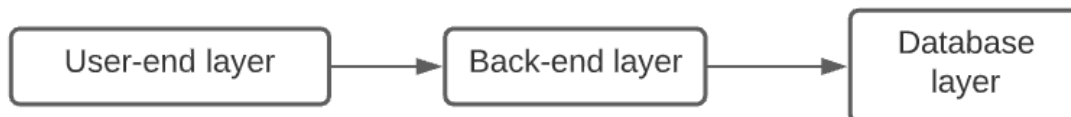
1 INTRODUCTION

The S wap (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 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. This is a software based project and we have been programming in JAVA language. It is an Android application developed using the Android studio platform. With this app we will enjoy our video streaming, chatting, and spending time with our family.

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.



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 USER-END LAYER SUBSYSTEMS

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.

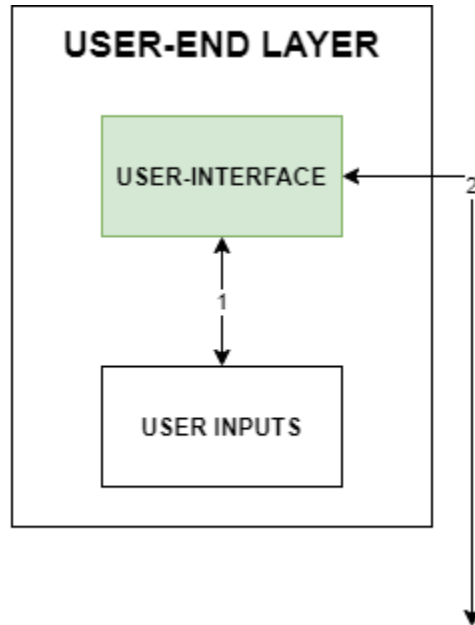


Figure: Example subsystem description diagram

3.1 USER-END LAYER HARDWARE

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.

3.2 USER-END LAYER OPERATING SYSTEM

This subsystem requires Google' Android operating system built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. The subsystem was created and compiled in Windows computer version 10 (64-bit). The Android studio version 4.2 was used.

3.3 USER-END LAYER SOFTWARE DEPENDENCIES

For the user-end layer, the team has implemented Java libraries and constraint layout libraries in android studio.

3.4 USER INPUT SUBSYSTEM

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.

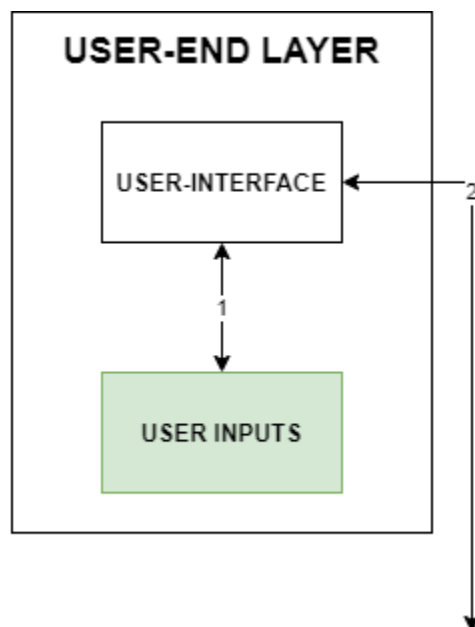


Figure 2: Example subsystem description diagram

3.4.1 SUBSYSTEM HARDWARE

Any android smartphone in which the user will be able to access the SWaP application.

3.4.2 SUBSYSTEM OPERATING SYSTEM

This subsystem requires Google' Android operating system built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. The subsystem was created and compiled in Windows computer version 10 (64-bit). The Android studio version 4.2 was used.

3.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

Similar to 3.3, the application uses Java libraries.

3.4.4 SUBSYSTEM PROGRAMMING LANGUAGES

This subsystem uses Java programming language and XML files.

3.4.5 SUBSYSTEM DATA STRUCTURES

N/A

3.4.6 SUBSYSTEM DATA PROCESSING

N/A

4 BACK-END LAYERS SUBSYSTEMS

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

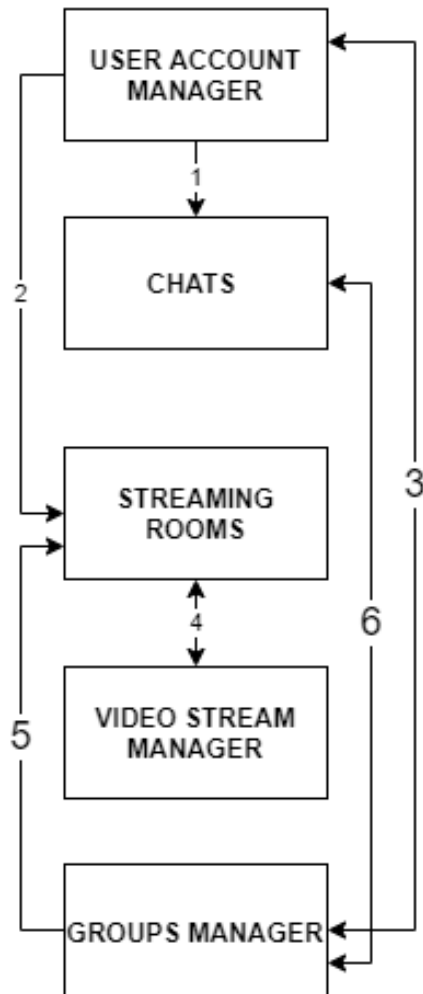


Figure 5:Example subsystem description diagram

4.1 LAYER HARDWARE

It does not require any hardware.

4.2 LAYER OPERATING SYSTEM

It is developed under Google's Android operating system.

4.3 LAYER SOFTWARE DEPENDENCIES

It depends on JAVA code and libraries which are compiled in the Android studio platform .

4.4 USER ACCOUNT MANAGER SUBSYSTEM 1

This subsystem lets the user login to the app with the username and password. It is responsible to manage the user account and keep the app active while the user is active.

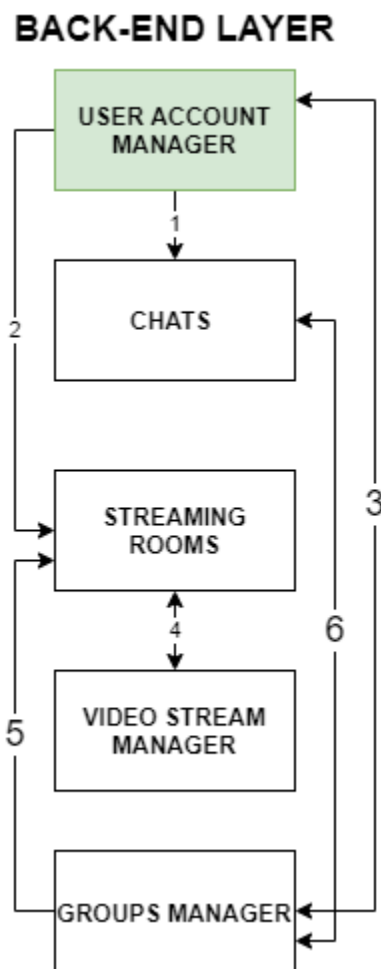


Figure 3: Example subsystem description diagram

4.4.1 SUBSYSTEM HARDWARE

It does not require any hardware.

4.4.2 SUBSYSTEM OPERATING SYSTEM

This subsystem requires Google' Android operating system built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. The subsystem was created and compiled in Windows computer version 10 (64-bit). The Android studio version 4.2 was used.

4.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

This subsystem software depends on JAVA library and Android Studio system software.

4.4.4 SUBSYSTEM PROGRAMMING LANGUAGES

The subsystem is coded using the JAVA programming language.

4.4.5 SUBSYSTEM DATA STRUCTURES

N/A

4.4.6 SUBSYSTEM DATA PROCESSING

N/A

4.5 CHATS SUBSYSTEM 1

This subsystem is responsible for keeping track of the friend list and the chat history.

BACK-END LAYER

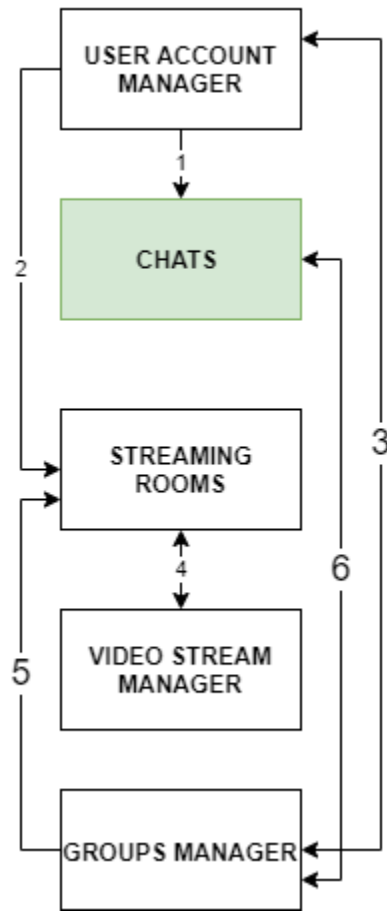


Figure 3: Example subsystem description diagram

4.5.1 SUBSYSTEM HARDWARE

It does not require any hardware.

4.5.2 SUBSYSTEM OPERATING SYSTEM

This subsystem requires Google' Android operating system built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. The subsystem was created and compiled in Windows computer version 10 (64-bit). The Android studio version 4.2 was used.

4.5.3 SUBSYSTEM SOFTWARE DEPENDENCIES

This subsystem software depends on JAVA library and Android Studio system software.

4.5.4 SUBSYSTEM PROGRAMMING LANGUAGES

The subsystem is coded using the JAVA programming language.

4.5.5 SUBSYSTEM DATA STRUCTURES

N/A

4.5.6 SUBSYSTEM DATA PROCESSING

N/A

4.6 STREAMING ROOM SUBSYSTEM 1

This subsystem is responsible for letting users add new friends or remove friends from the streaming room. It also controls the video streaming like pause, play, etc.

BACK-END LAYER

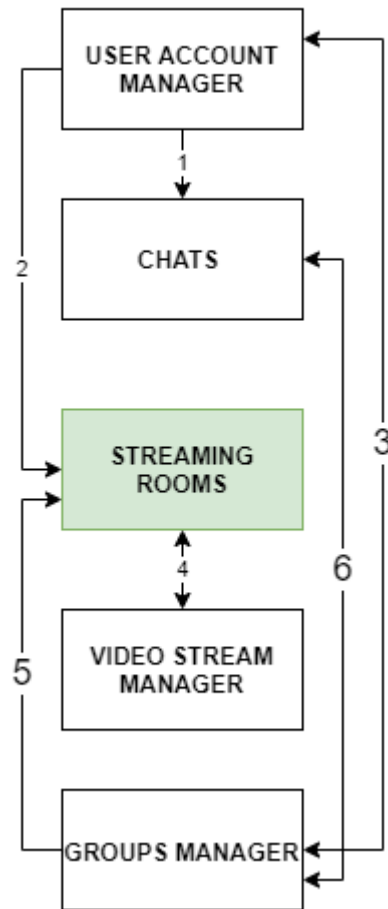


Figure 3: Example subsystem description diagram

4.6.1 SUBSYSTEM HARDWARE

It does not require any hardware.

4.6.2 SUBSYSTEM OPERATING SYSTEM

This subsystem requires Google' Android operating system built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. The subsystem was created and compiled in Windows computer version 10 (64-bit). The Android studio version 4.2 was used.

4.6.3 SUBSYSTEM SOFTWARE DEPENDENCIES

This subsystem software depends on JAVA library and Android Studio system software.

4.6.4 SUBSYSTEM PROGRAMMING LANGUAGES

The subsystem is coded using the JAVA programming language.

4.6.5 SUBSYSTEM DATA STRUCTURES

N/A

4.6.6 SUBSYSTEM DATA PROCESSING

N/A

4.7 VIDEO STREAM MANAGER SUBSYSTEM 1

This subsystem is responsible for arranging the database of video requested and streamed. It also recommends the video that is preferred by the user.

BACK-END LAYER

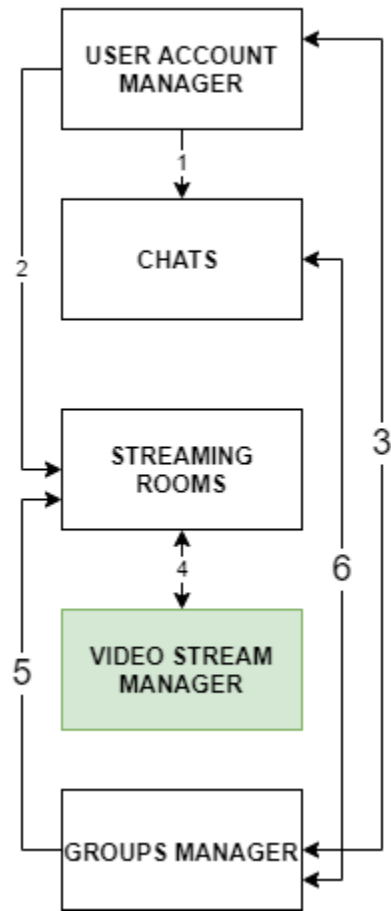


Figure 3: Example subsystem description diagram

4.7.1 SUBSYSTEM HARDWARE

It does not require any hardware.

4.7.2 SUBSYSTEM OPERATING SYSTEM

This subsystem requires Google' Android operating system built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. The subsystem was created and compiled in Windows computer version 10 (64-bit). The Android studio version 4.2 was used.

4.7.3 SUBSYSTEM SOFTWARE DEPENDENCIES

This subsystem software depends on JAVA library and Android Studio system software.

4.7.4 SUBSYSTEM PROGRAMMING LANGUAGES

The subsystem is coded using the JAVA programming language.

4.7.5 SUBSYSTEM DATA STRUCTURES

N/A

4.7.6 SUBSYSTEM DATA PROCESSING

N/A

4.8 GROUP MANAGER SUBSYSTEM 1

Group manager is responsible for recording the list of users in the group in the database. It also records the chat and videos that have been watched in that group to the database.

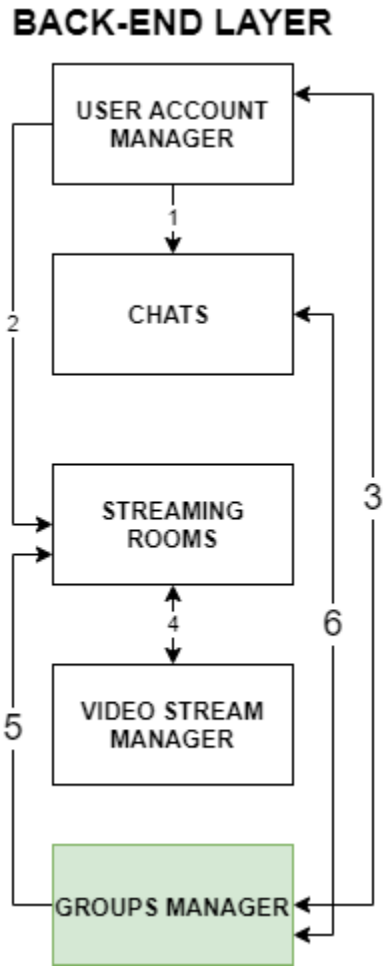


Figure 3: Example subsystem description diagram

4.8.1 SUBSYSTEM HARDWARE

It does not require any hardware.

4.8.2 SUBSYSTEM OPERATING SYSTEM

This subsystem requires Google' Android operating system built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. The subsystem was created and compiled in Windows computer version 10 (64-bit). The Android studio version 4.2 was used.

4.8.3 SUBSYSTEM SOFTWARE DEPENDENCIES

This subsystem software depends on JAVA library and Android Studio system software.

4.8.4 SUBSYSTEM PROGRAMMING LANGUAGES

The subsystem is coded using the JAVA programming language.

4.8.5 SUBSYSTEM DATA STRUCTURES

N/A

4.8.6 SUBSYSTEM DATA PROCESSING

N/A

5 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

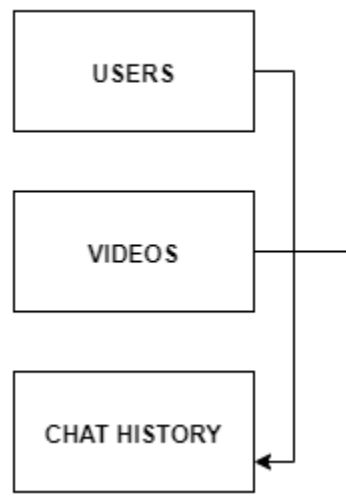


Figure:Example subsystem description diagram

5.1 LAYER HARDWARE

It does not require any hardware.

5.2 LAYER OPERATING SYSTEM

It is developed under Google's Android operating system.

5.3 LAYER SOFTWARE DEPENDENCIES

It depends on JAVA code and libraries which are compiled in the Android studio platform .

5.4 USER SUBSYSTEM 1

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

DATABASE LAYER

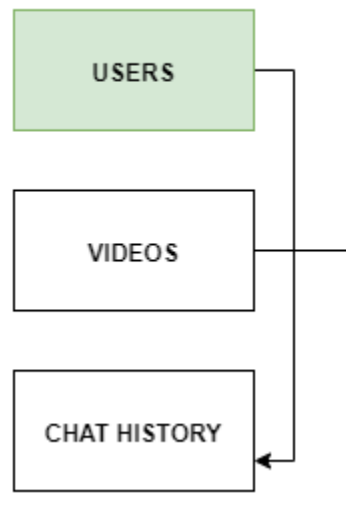


Figure 4: Example subsystem description diagram

5.4.1 SUBSYSTEM HARDWARE

This subsystem does not involve any hardware.

5.4.2 SUBSYSTEM OPERATING SYSTEM

This subsystem requires Google' Android operating system built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. The subsystem was created and compiled in Windows computer version 10 (64-bit). The Android studio version 4.2 was used.

5.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

This subsystem software depends on JAVA library and Android Studio system software.

5.4.4 SUBSYSTEM PROGRAMMING LANGUAGES

The subsystem is coded using the JAVA programming language.

5.4.5 SUBSYSTEM DATA STRUCTURES

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

5.4.6 SUBSYSTEM DATA PROCESSING

The user information or credentials are used as input data. Those data are recorded as user information which helps during the authenticating process of the user. The output is produced in the form of a Boolean i.e. True or false. If the data is verified during input then output is true or else false.

5.5 VIDEOS SUBSYSTEM 1

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

DATABASE LAYER

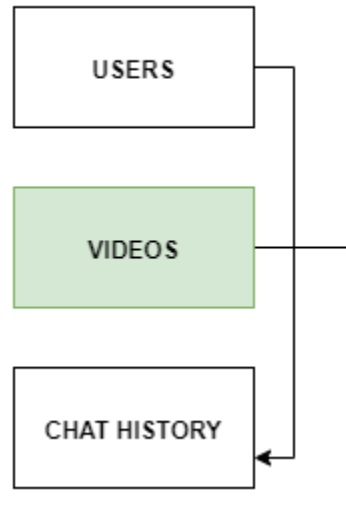


Figure 4: Example subsystem description diagram

5.5.1 SUBSYSTEM HARDWARE

This subsystem does not involve any hardware.

5.5.2 SUBSYSTEM OPERATING SYSTEM

This subsystem requires Google' Android operating system built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. The subsystem was created and compiled in Windows computer version 10 (64-bit). The Android studio version 4.2 was used.

5.5.3 SUBSYSTEM SOFTWARE DEPENDENCIES

This subsystem software depends on JAVA library and Android Studio system software.

5.5.4 SUBSYSTEM PROGRAMMING LANGUAGES

The subsystem is coded using the JAVA programming language.

5.5.5 SUBSYSTEM DATA STRUCTURES

This subsystem is structured to keep all the video streamed by the user and the new and popular video which are recently updated in the app. Its sole purpose is to keep track of the user video stream.

5.5.6 SUBSYSTEM DATA PROCESSING

This subsystem processes the video section. This creates a reference video section and recommends the videos that are most recently watched, new videos, and most popular videos. It manages all the videos streamed by the user. This section stores the videos and corresponds to the user demand of the video request.

5.6 CHAT SUBSYSTEM 1

This subsystem is created to connect the users through chat. Users will be able to chat while they are streaming the video together. The chat done during the stream is stored in this subsystem.

DATABASE LAYER

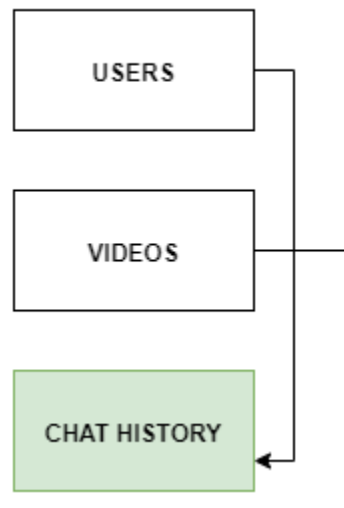


Figure 4: Example subsystem description diagram

5.6.1 SUBSYSTEM HARDWARE

This subsystem does not involve any hardware.

5.6.2 SUBSYSTEM OPERATING SYSTEM

This subsystem requires Google' Android operating system built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. The subsystem was created and compiled in Windows computer version 10 (64-bit). The Android studio version 4.2 was used.

5.6.3 SUBSYSTEM SOFTWARE DEPENDENCIES

This subsystem software depends on JAVA library and Android Studio system software.

5.6.4 SUBSYSTEM PROGRAMMING LANGUAGES

The subsystem is coded using the JAVA programming language.

5.6.5 SUBSYSTEM DATA STRUCTURES

This subsystem is structured to control the flow of the messages between users while the video is streamed continuously. It is connected directly with the user subsystem. It records and stores the chat and messages.

5.6.6 SUBSYSTEM DATA PROCESSING

This system is responsible for storing, retrieving and transferring the messages between one user to another. This is based on a boolean logic, where the messages are stored and transferred when logic is true and vice-versa.

SUBSYSTEM DEFINITIONS & DATA FLOW

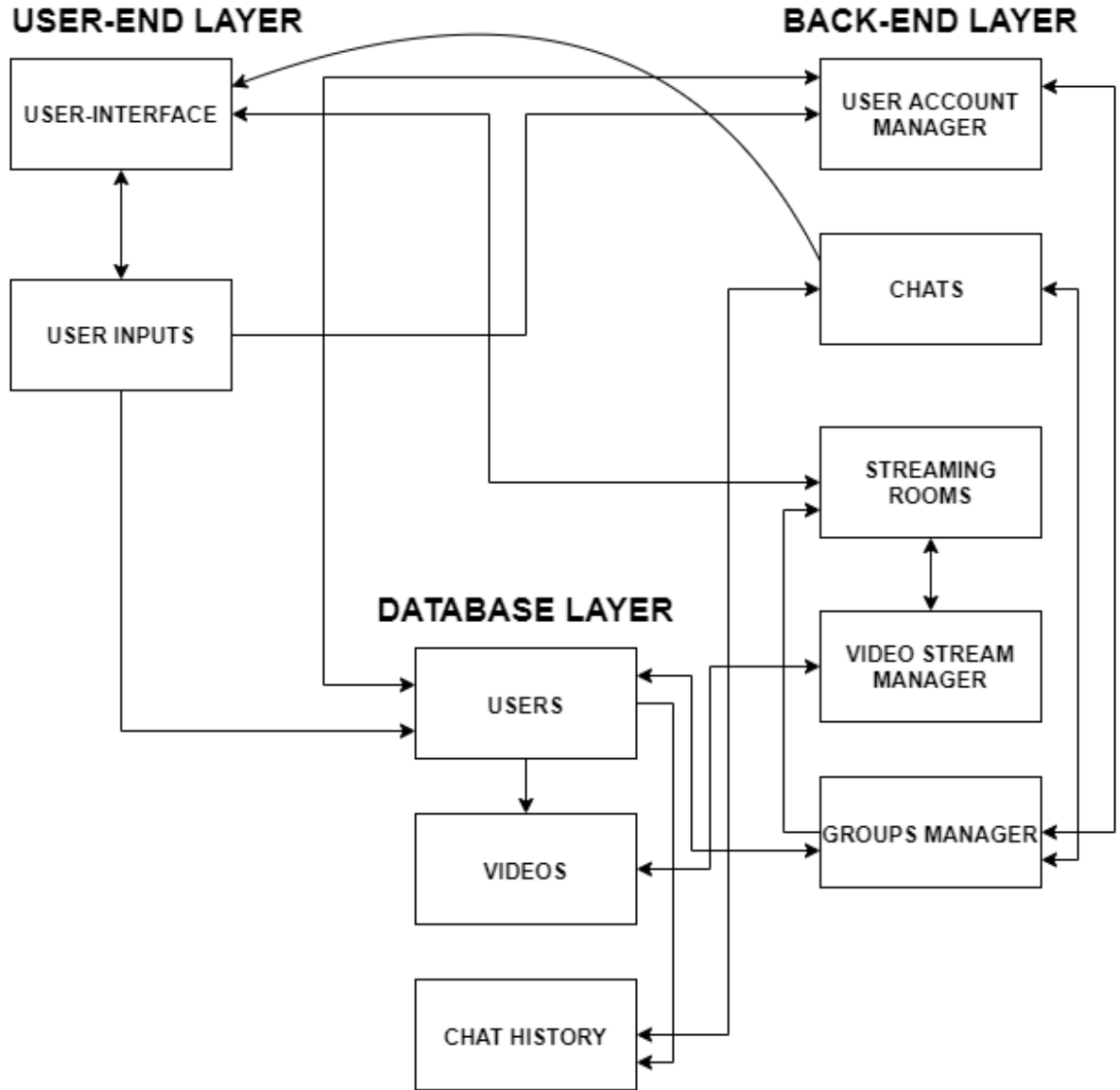


Figure : A simple data flow diagram