

(Wetube)
(Team 1)
(Hunter Hults, Michael Capelotti, Egan Bower, Colin
Woodard)

Software Requirements Specification Document

Version: (1.0)

Date: (03/07/2019)

Table of Contents

| | |
|---|---|
| 1 Purpose. | 3 |
| 2 Scope. | 3 |
| 3 User characteristics. | 3 |
| 3.1 Key users. | 3 |
| 3.2 Secondary users. | 4 |
| 3.3 Unimportant users. | 4 |
| 4 Product perspective. | 4 |
| 4.1 System Context | 5 |
| 4.2 User interfaces. | 5 |
| 4.3 Software interfaces. | 6 |
| 4.4 Hardware interfaces and Memory constraints. | 6 |
| 4.5 Deployment requirements. | 7 |
| 5 Assumptions and Dependencies. | 7 |
| 6 Specific requirements. | 7 |
| 6.1 System Functional Requirements. | 8 |
| 6.2 Logical Database Requirements. | 8 |
| 6.3 Software System Attributes. | 8 |
| 6.3.1 Usability. | 8 |
| 6.3.2 Performance. | 8 |
| 6.3.3 Reliability/Dependability. | 8 |
| 6.3.4 Security. | 8 |
| 6.3.5 Maintainability. | 9 |

1 Purpose

YouTube is a great place to watch videos for all sorts of needs: entertainment, educational, news, etc. While YouTube has enabled the sharing of video content at a massive scale, the YouTube frontends (mobile and web applications) have been made with the idea of a single user viewing mode in mind. The use case for group viewing, multiple users watching a video or set of videos together from multiple devices, offers a unique feature on top of the already existing YouTube ecosystem. The desire for group watching offers several benefits for general and specific use cases. For example, many users like to watch videos with their friends who may be remote for entertainment purposes, but groupwatching is also extremely beneficial for remote educational purposes as well. Currently, competitors are entering the space with services like Facebook Watch and Oculus TV, both Facebook products that allow users to watch their

favorite content with each other. We don't see this as a deterrent, but rather we think it is proof of market viability. We feel WeTube offers a unique experience even in this space because YouTube is still primarily centered around single user viewing. In addition, the various Facebook products in this space focus more upon the social aspect and Oculus TV is aimed only at VR users, whereas WeTube will focus on a general group viewing experience with key features we believe users will find helpful for entertainment and education, while still providing a use for cases we may have not predicted. Thus, we believe the unique value of WeTube offering a group viewing experience for the ever popular YouTube content, in addition to our focus on generally applicable features, will make WeTube an attractive product to a wide audience.

2 Scope

WeTube would need to be able to play videos on both the web and mobile front ends that are planned for development. On top of this, we would need the experience to be synchronized between users. When a group of users is watching together inside of a video room, things like the current video, and playback position and speed, should be synchronized in real time between all members of the given room. With the notion of a room, there will need to be some way to authenticate users, even if at the basic level of identification. In addition to watching YouTube videos in a room, members should be able to communicate with each other about the content of the video, most likely through a chatlike functionality to begin, but possibly audio and video in the future. Finally, users will need the ability to change the currently playing video. Insofar, as the application needs map onto technical needs, the initial technical requirements would most likely be a realtime database, offered by Firebase, for storing room information.

This would include existing rooms, which would include the current state of a video (YouTube video ID, seektime, playback state, etc.), and communication information like chat messages. Additionally, we would need access to the YouTube API. Currently, access is permitted by creating a Google Cloud Platform project and enabling the API in Console. For the frontend, we would need a user interface on a platform supported by the YouTube and Firebase APIs along with the required components to allow for the basic flow through the app. This includes username selection, room selection, and group watching (with chat features). Both Firebase and YouTube support Android, IOS, and web on the frontend. Our Focus will be on Android to begin.

Service Breakdown

- Identity Basic service to distinguish users in the application. Upon application boot, user will choose a username to be used appwide
- Rooms A room is an abstract entity that has the subservices of Player, and Messaging, but also maintains metadata regarding things like Access Control (can

members change things like the video or the player controls?), and player information (what video is currently being watched, seek time, etc.)

- ❖ **Player** The group enabled player integrating with the YouTube API and Firebase to provide real time synchronization between members of the group

- ❖ **Messaging** A general purpose message interface that integrates with Firebase to provide real time messaging within a group

3 User characteristics

WeTube offers a service that is attractive to a wide range of users spanning multiple demographics. In general, we think the characteristics of a user attracted to WeTube will be anyone either already in or looking for some sort of social group, as WeTube is primarily targeted for a group audience. This could include students, friends, educators, employers, and many more. These users could span across multiple levels of education, experience, or expertise, so our goal is to make WeTube as accessible as possible for all these separate groups.

3.1 Key users

WeTube will be attractive to many different users, but we also assume that initial users of WeTube will be younger, and probably follow similar demographics and trends to that of YouTube.

3.2 Secondary users

Secondary users would be users that are invited onto the platform by the key users, and as such they could be of any demographic or group. **User role responsibilities:** They would be invited to join existing groups and maybe occasionally creating their own groups. **Subject matter experience:** They could have medium to low levels of experience with the YouTube platform. **Technological experience:** We would expect that they could have a lower level of experience with technology than the key users, so we would want to make it as easy as possible for these users to join a room and watch the video.

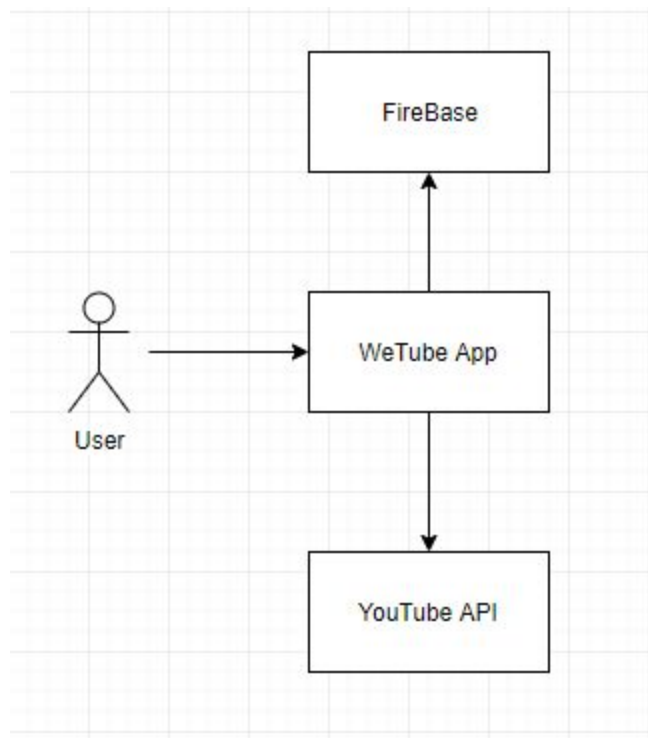
3.3 Unimportant users

Unimportant users would be users that only use the service very infrequently. User role responsibilities: Invited to watch videos with other users. Subject matter experience: Little or no experience with the YouTube platform. Technological experience: Little to no experience with technology can be expected for our unimportant users

4 Product perspective

4.1 System Context

The system will operate as an android app, which itself uses the YouTubePlayer API to include a video player in the room screen. The app will communicate directly with the YouTube API to retrieve the video feed. Synchronization, room management, and communication between clients will be handled by the Firebase Realtime Database, which will be directly implemented into Android Studio, which is where the app will be developed.



4.2 User interfaces

It is required of the system to use Youtube as the source of the videos that are watched on the application.

It is required of the system to display a error message if there is an error providing the video.

It is required of the system to work on all devices running on android os version KitKat or higher.

It is required of the system to restrict or give access to certain features depending on the permissions given by the admin of a room.

It is required of the system show the user's actions and sync them across to all users in the room.

4.3 Software interfaces

The system uses a database to hold room information.

The system will use youtube as the source for the videos.

The system will run on devices running an android os.

The system will run on newer versions of android.

The system will have a chat feature that works on mobile.

To summarize the system will hold room data to know what room is active and what video is being played. It is able to work together with mobile and watch videos with those who are on different devices. Users will be able to chat through message system to other users in the room.

4.5 Deployment requirements

The program runs mostly on the app the user just needs to connect to the firebase to comment to a room and update the video's state. For mobile they would download an app to view the videos but most of the data is on the user side and not the firebase side. The video is streamed through a youtube API.

5 Assumptions and Dependencies

WeTube is an Android application thus it will be distributed through the Google Play Store. The application must be compliant with the Google Play Store's policies. If there are any requirements in this SRS that come into conflict with Google's policies on sale or function the requirements are to be changed. These changed requirements will be changed such that there is no longer a conflict between WeTube and Google's policies.

6 Specific requirements

6.1 System Functional Requirements

R1.1.1 as an unauthorized user i want to be presented with a login screen with options to login.

R1.2.9 As an authenticated user, I want to be able to join or leave rooms that other users have created and watch the video that is currently being played.

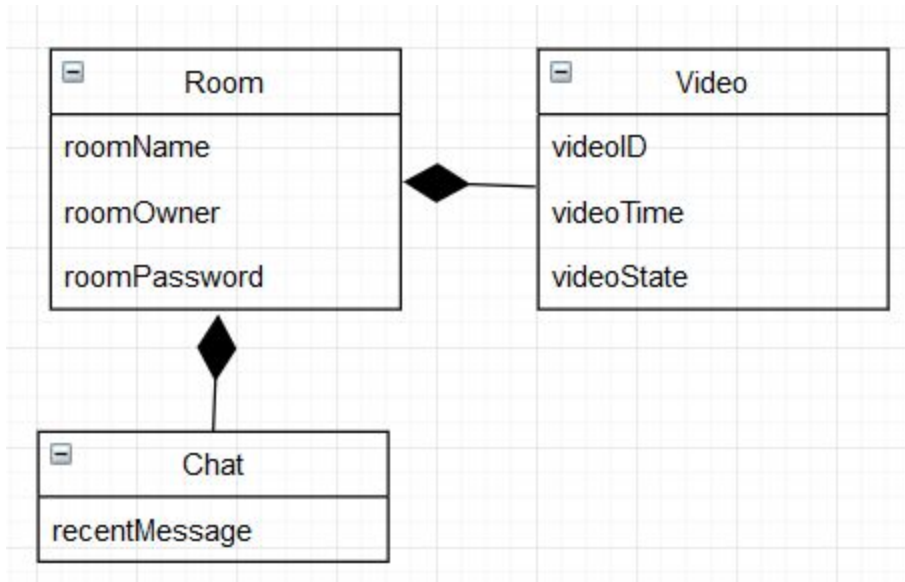
R1.2.10 As an authenticated user, I want to be able to create my own room and run it as an administrator.

R1.3.1 As a room administrator, I want to be able to choose what video is being played as well as when to pause, rewind, or fast forward during the video.

R1.3.2 As a room administrator, I want to be able to set the room's password.

R1.3.4 As a room administrator, I want to be able to close a room whenever I wish.

6.2 Logical Database Requirements



6.3 Software System Attributes

6.3.1 Usability

R3.1.1 The system does need to be active 24 hours a day

R3.1.3 The system should be easy to use so that anyone can use this service with little to no instruction.

R3.1.4 The system should show any video with ease.

6.3.2 Performance

R3.2.1 User should be able join video room in less than 5 seconds.

R3.2.2 Video rooms should be able to hold 100+ users at one time.

6.3.3 Reliability/Dependability

R3.3.1 The system will have a server access failure rate of less than 1%.

R3.3.2 The system will have a user action (messages) failure rate of less than 1%.

R3.3.3 The system will notify user when any user action fails.

R3.3.5 The system will not allow unauthorized access 100% of the time.

6.3.4 Security

R3.4.1 The system will distinguish between authorized and unauthorized users.

6.3.5 Maintainability

R3.5.1 The system will be designed in a way that it is easy to add new features in the future without significant modification to the current code.

R3.5.2 The system will maintain testability and be designed in a way that features and functions are easily testable at regular intervals.

R3.5.3 The system will maintain short, simple, and concise units of code for ease of future understanding and modification during the development process.

R3.5.4 The system will avoid duplication of code to maintain simplicity and extendibility.