Software Requirements Specification

for

Callify

**Version 1.0 approved**

**Prepared by Team Exceptional Pointers**

**Soham Shinde, Shubham Pandey and Harsh Gupta**

**CSE Junior Year Undergraduates at IIIT Kottayam**

**26-09-2021**

**Table of Content**

**Table of Content 2**

**Revision History 3**

**-------------------------------------------------------------------------------------------------------------------------------**

1. **Introduction [1](#_heading=h.1fob9te)**
   1. Purpose [1](#_heading=h.3znysh7)
   2. Document Conventions [1](#_heading=h.2et92p0)
   3. Intended Audience and Reading Suggestions [1](#_heading=h.tyjcwt)
   4. Product Scope [1](#_heading=h.3dy6vkm)
   5. References [1](#_heading=h.1t3h5sf)
2. **Overall Description 2**
   1. Product Perspective 2
   2. Product Functions [2](#_heading=h.17dp8vu)
   3. User Classes and Characteristics [2](#_heading=h.3rdcrjn)
   4. Operating Environment [2](#_heading=h.26in1rg)
   5. Design and Implementation Constraints [2](#_heading=h.lnxbz9)
   6. User Documentation [2](#_heading=h.35nkun2)
   7. Assumptions and Dependencies [2](#_heading=h.35nkun2)
3. **External Interface Requirements [3](#_heading=h.1ksv4uv)**
   1. User Interfaces [3](#_heading=h.44sinio)
   2. Hardware Interfaces [3](#_heading=h.2jxsxqh)
   3. Software Interfaces [3](#_heading=h.z337ya)
   4. Communications Interfaces [3](#_heading=h.3j2qqm3)
4. **System Features 4**
   1. User Authentication 4
   2. VideoConferencing ……………....…………………………………………………………….5
   3. Group messaging during video conferencing ………………………………………………...
   4. Privacy Lock for the User stream……………………………………………………………...
   5. Raise Hand …………………………………………………………………………………..
   6. White Board …………………………………………………………………………………...
   7. Recording…………………………………………………………………………………....
   8. Screen Share 5
5. **Other Nonfunctional Requirements 6**
   1. Performance Requirements 6
   2. Safety Requirements 6
   3. Security Requirements 6
   4. Software Quality Attributes 6
   5. Business Rules [5](#_heading=h.3as4poj)
6. **Other Requirements 7**

**Appendix A: Glossary 7**

**Appendix B: Analysis Models 8**

**Appendix C: To Be Determined List 8**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Software Requirements Specification for Callify | 26-09-2021 | Initial Copy | 1.0 |
| Revision of SRS with updated Features | 17-10-2021 | Updated it with new features | 2.0 |

# Introduction

## Purpose

This is the Software Requirements Specification (SRS) report (Version 2.0) for the Callify (Conferencing with WebRTC). The reason for this record is to impart the necessities (useful, nonfunctional and others) of Callify to the pursuer.

This report gives:

a) A general depiction of the item.

b) A meaning of the outer interface necessities.

c) A determination of the framework highlights (practical prerequisites), nonfunctional necessities

furthermore, different necessities

## Document Conventions

## *We underlined some of the salient features of our platform.*

* In this document User refers to both, the Host and the person who joins as well as the one who attends the meeting.
* Every Requirement has its own importance on the stage.
* Every detail has been taken care of while explaining.
* Requirements which are yet to be determined are represented with the Abbreviation TBD in brackets

## Intended Audience and Reading Suggestions

The different types of reader that the document is intended for are developers, project managers, marketing staff, end-users, testers, and documentation writers. This archive was made by holding the perspective on various partners of this stage in various areas guaranteeing the world's top tier experience for end-clients*.*

## Product Scope

Callify is designed in order to give quality conferencing to people in urban and rural areas with a variety of scopes. It helps people connect with others. The possible use cases for it could be having an online lecture over it, having a corporate meeting over it, meeting family and friends from the other part of the globe, different types of collaboration for different purposes, having job interviews, legal meetings and conferences and even remote Diagnostics.

The product uses peer to peer connectivity. This archive was made by holding the perspective on various partners of this stage at various areas guaranteeing the world's top tier experience for end-clients.

## References

1. This Software Requirements Specification is prepared by taking the references of Library Management System (LMS) prepared **by Dr. Divya Sindhu Lekha** at Indian Institute of Information Technology Kottayam.

# Overall Description

## Product Perspective

The product Callify is a video conferencing app. It can have multiple use cases, it can be used for Educational, Office purposes as well as home use. Especially in today’s times where the pandemic has changed the way people even breathe it is necessary for the world to get equipped with video conferencing service. This can be used to educate in remote locations without physically being present. Even office meetings can be conducted on the internet without having to be present in the meeting room practically. It can be used to make video calls to family members and help incase on homesickness to one.

The best part of the app is that it uses P2P connection making it efficient over a network.

## Product Functions

1. Easy to manage and schedule web calls
2. Generic platform to fit for all scenarios like group discussions, webinars, e classrooms, interviews, presentations etc.
3. Session and policy management.
4. To be marketed as (SaaS) and (PaaS)
5. Developer friendly API.
6. Recording of sessions.

## User Classes and Characteristics

1. Host: The person who makes the Room for conference
2. Viewer: The person who joins the already made room.

## Operating Environment

Callify is a web-based application. It works in Chrome Version 94.0.4606.81 (Official Build) (64-bit). Users will be able to use the platform using desktop and laptop computers, and mobile devices having the given browser installed.

## Design and Implementation Constraints

1. Anti WebRTC corporate or regulatory policies.
2. Hardware limitations i.e., no Mic or Camera
3. Software limitations i.e., no or unsuitable browser.
4. RAM and memory requirements
5. Organization has blocked access to sockets or TURN servers.

## User Documentation

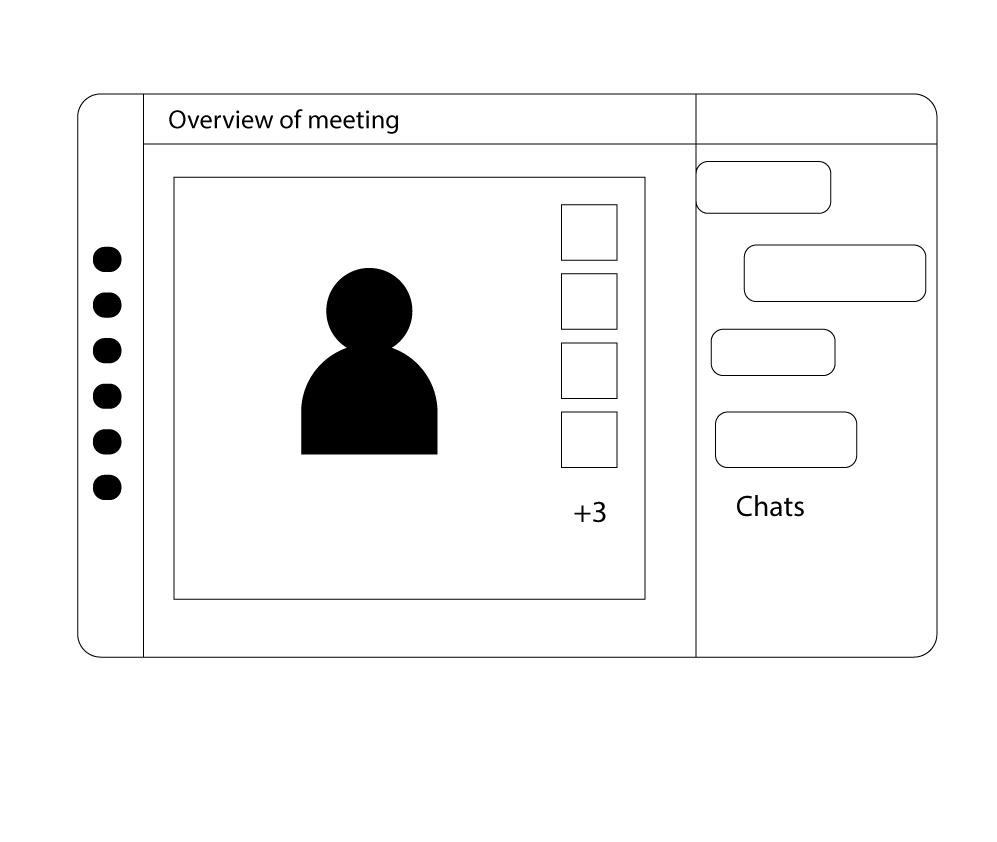
1. User manuals
2. WebRTC documentation
3. Socket.IO documentation
4. PeerJS documentation
5. Other documentations

## Assumptions and Dependencies

1. The users should have WebRTC supported browser Google Chrome Version 94.0.4606.81 (Official Build) (64-bit).
2. User must give media permission for the camera and microphone access for the video session
3. Permission should be given to the firewall if needed.

# External Interface Requirements

## User Interfaces



## Hardware Interfaces

1. Microphone access to the device.
2. Web camera access to the device.
3. Active and fast internet connection preferably ethernet.

## Software Interfaces

1. The application would be hosted on Heroku.
2. The STUN servers involved while initializing the connection would be free google STUN servers ([list here](https://gist.github.com/zziuni/3741933) ).
3. For databases it would be using MongoDB preferably hosted on AWS.
4. Google Chrome web browser (preferred)

## Communications Interfaces

1. The communication would be happening over the HTTP/TCP protocol.
2. Data from the input of the user will be transferred using the HTTP post method and webpages would be served using HTTP Get method.
3. Further communication of Feeds would take place over TCP.
4. The signaling would also involve the use of socket programming internally handled by Socket.IO.

# System Features

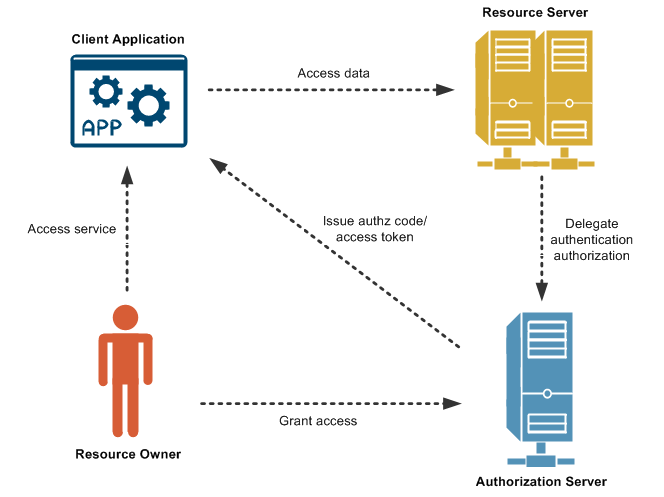
## User Authentication

### Description and Priority

Users must register into the platform to use the services offered. This is a ***high*** priority feature. The mode of registration would be through Google Authentication. There it is necessary for the user to have a Google account/ Gmail Account for the same.

### Stimulus/Response Sequences

User registration doesn’t happen at our system end, we use the credentials provided by the user while making the Google account itself. The user would be asked for Authenticating his Google/ Gmail account to the service.



Depending upon the type of security the User has enabled in his/ her account the user will proceed that way to authorize.

### Functional Requirements

* + - 1. **User authentication**

| **Callify/SRS/4.1 User Registration/ 1.1.3.1 User Registration** | |
| --- | --- |
| Function | Registers User as a customer |
| Description | Allows user to register himself as a customer and joins in  the platform as a ***member*** |
| Inputs | Gmail email address |
| Source | Inputs from field in customer registration form |
| Output | User Account created with unique identity in the database. |
| Destination | Google Oauth 2.0 page where user would be asked to allowing account access to the app. |
| Action | All Input fields are validated. If the user inputs are valid, create user account. If anything goes wrong it wont proceed. |
| Requirements | No Requirements. |
| Pre-Condition | Customer must possess age above 18 years. Gmail Email ID and they must be entered correctly. |
| Post-Condition | New User account with a unique id. |
| Side Effects | None |

## Video Conferencing

### Description and Priority

Users would be able to do video conferencing with each other. This would be ***high*** and the main feature of the product. All the users would be authorized ones.

### Stimulus/Response Sequences

The users have an option of either making a new room for the conference or joining an already created one. In case the user makes a new room he has to click on making new room button and a new room would be made. In case the user wants to join an already created room he must have the room id of the Room the user wants to join.

### Functional Requirements

* + - 1. **Video conferencing**

| **Callify/SRS/4.2 User Registration/ 4.2.3.1 Customer Registration** | |
| --- | --- |
| Function | Group video conferencing |
| Description | Allows user to have live video conferencing. |
| Inputs | Room ID is the user intends to join an already created room. |
| Source | Inputs from field in Room ID input. |
| Output | A gird of video elements on the page. |
| Destination | Video conference room |
| Action | User can start interacting with other users in the room there after. |
| Requirements | No Requirements. |
| Pre-Condition | No pre-conditions |
| Post-Condition | User entering a video conferencing room |
| Side Effects | None |

## Group Messaging during video conferencing

### Description and Priority

Users in the same conferencing room would be able to have a text-based chat. This test-based chat would be based on the chat rooms. Users who are connected at a later time won’t be able to see the earlier messages. This is an ***medium*** priority Feature.

### Stimulus/Response Sequences

The messages in the chat windows would be arranged in a fashion such that the name of the sender would be displayed along with the message. To send an message the user needs to type in the message input textbox and click on send. The message would be sent immediately to all the users in the same room.

### Functional Requirements

* + - 1. **Group Chat**

| **Callify/SRS/ 4.3 User Registration/ 4.3.1.1 Group Chat** | |
| --- | --- |
| Function | User group chat while video conferencing. |
| Description | Allows connected people in the conference room to interact via text messages without interrupting the speaker. |
| Inputs | Message in the text message input box. |
| Source | Inputs from field in text message input box. |
| Output | Message is sent to all the users in the room. |
| Destination | No other destination, the changes would be reflected on the same page. |
| Action | The message is sent instantly to all the users in the same room. |
| Requirements | The user must be online connected. |
| Pre-Condition | No preconditions. Its a user intended function i.e., the user can message whenever he wants. |
| Post-Condition | Message is received by all. |
| Side Effects | None |

## Privacy Lock for user stream

### Description and Priority

Each user is given the option to stop his Camera feed at any point of time. As soon as the user chooses to stop his Video and Audio stream the same is reflected at the end of all users. This is a ***high*** priority Feature.

### Stimulus/Response Sequences

The user is given an option to join the meeting with stream enabled or disabled.

### Functional Requirements

## Privacy Lock for user stream

| **Callify/SRS/ 4.3 / 4.3.1.1 Group Chat** | |
| --- | --- |
| Function | User group chat while video conferencing. |
| Description | Allows connected people in the conference room to stop the stream and have their privacy ensured. |
| Inputs | Toggle button |
| Source | Video stream from camera and mic. |
| Output | As soon as the user chooses to stop his Video and Audio stream the same is reflected at the end of all users |
| Destination | User screen of all users. |
| Action | The message is sent instantly to all the users in the same room. |
| Requirements | None |
| Pre-Condition | No preconditions. It's a user intended function i.e The user can toggle his stream whenever he/she wants. |
| Post-Condition | The stream of the user is stopped at the end of all users. |
| Side Effects | None |

## Privacy Lock for user stream

### Description and Priority

Each user is given the option to stop his Camera feed at any point of time. As soon as the user chooses to stop his Video and Audio stream the same is reflected at the end of all users. This is a ***high*** priority Feature.

### Stimulus/Response Sequences

The user is given an option to join the meeting with stream enabled or disabled.

### Functional Requirements

## Privacy Lock for user stream

| **Callify/SRS/ 4.3 / 4.3.1.1 Group Chat** | |
| --- | --- |
| Function | User group chat while video conferencing. |
| Description | Allows connected people in the conference room to stop the stream and have their privacy ensured. |
| Inputs | Toggle button |
| Source | Video stream from camera and mic. |
| Output | As soon as the user chooses to stop his Video and Audio stream the same is reflected at the end of all users |
| Destination | User screen of all users. |
| Action | The message is sent instantly to all the users in the same room. |
| Requirements | None |
| Pre-Condition | No preconditions. It's a user intended function i.e The user can toggle his stream whenever he/she wants. |
| Post-Condition | The stream of the user is stopped at the end of all users. |
| Side Effects | None |

# Other Nonfunctional Requirements

## Performance Requirements

Since it’s a video conferencing application, although its web based it need users to be connected on a high-speed internet connection. The app made during the project would be using the open source freely available servers for P2P connection. They would be already filled with other project developers and hence the users need to make sure they don’t provide lag in the connection. The application would be transferring the video stream over the network hence it would be highly bandwidth occupying service. Also, its needed that the browser which the user uses supports WebRTC API, without which running the application would be impossible. Suggested browser for the same would-be Google chrome. Also your browser must be allowing socket connections, as it is used integrally by the webapp.

## Safety Requirements

The data of the P2P connection must be made sure it is sent securely over the network. Unintended injection of packets is a big concern over P2P connections. Also, the user shouldn’t share the room details with the person unintended to join it.

## Security Requirements

User details stored in the server must be encrypted and in case the server gets hacked the data won’t be usable to the hackers. Using powerful one-way encryption algorithms is advised.

## Software Quality Attributes

Since it is a Web App it can be accessed from a browser without any limitation on the operating system it's being used on. The suggested web browser is Google chrome. Since the Google Chrome is available on a wide range of operating systems the webapp too can be accessed across multiple operating systems. The device should have a working Camera and microphone for communication. The robustness can be controlled on the backend. Since the app we are making free open-source servers their robustness would be limited to the bandwidth offered under the free tier.

## Business Rules

The room creator would be allowed to control the accessibility of the room to other users.

Also the developers have the authority to stop the server at any time.

# Other Requirements

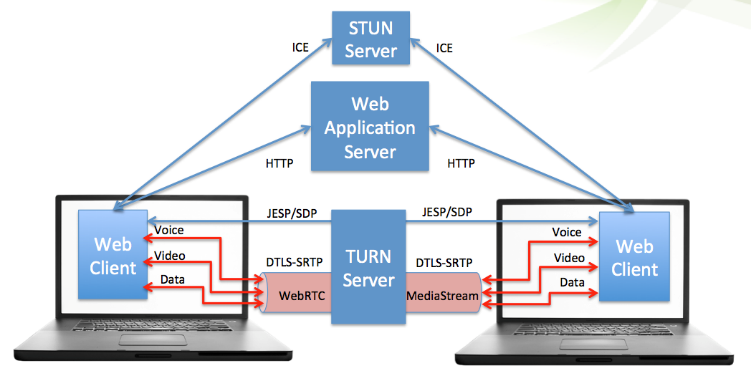
*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

**Appendix A: Glossary**

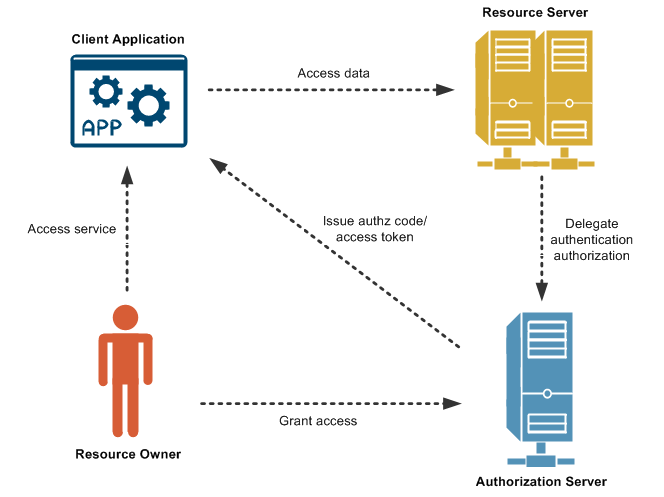
1. WebRTC : *Web Real-Time Communication is a P2P communication API.*
2. Room ID : a string of characters which identifies the room uniquely.
3. OAuth 2.0 : *OAuth 2.0 is the industry-standard protocol for authorization.*
4. P2P: Peer to Peer. It’s the topology of devices connected over the network.
5. API: Application Programming Interface
6. Google ID: A Gmail email ID
7. STUN server: server by a device for getting is network details
8. TURN server : Server used to facilitate the P2P connection in case of some firewall blocking it.
9. PeerJS: The library used for implementing WebRTC.
10. Heroku: Online web app hosting service, student friendly
11. AWS : Amazon Web Services

**Appendix B: Analysis Models**

1. Working model of WebRTC:



1. Working model of OAuth 2.0:

****

**Appendix C: To Be Determined List**

1. Special functionality depending upon the use to be finalized.
2. UX/UI to be more improved over the time depending upon the requirement.