**HIGH LEVEL DESIGN DOCUMENT**

**Video Conferencing Application**

Contents

Abstract

1. Introduction
   1. Why is High Level Design Document?
   2. Scope
2. General Description
   1. Products Perspective
   2. Problem Statement
   3. Proposed Solution
   4. Further Improvements
   5. Tools Used
   6. Constraints
3. Design Details
   1. Process Flow
   2. Error Handling
   3. Performance
   4. Reusability
   5. Resource Utilization
   6. Deployment

**Abstract**

Video conferencing is an online technology that allows users in different locations to hold face-to-face meetings without having to move to a single location together. We have seen during pandemic that only source we had to conduct business meets, work from home, impart education and so much was only possible because of tech and apps like zoom, Google meet, teams and much more. So as a techie What can be better than making your own video conferencing application and interact with your friends.

# 1. Introduction

## Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

* Present all of the design aspects and define them in detail
* Describe the user interface being implemented
* Describe the hardware and software interfaces
* Describe the performance requirements
* Include design features and the architecture of the project

• List and describe the non-functional attributes like:

* + - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application
    - Compatibility
    - Serviceability

## 1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

2. **General Description**

2.1 **Product Perspective & Problem Statement**

The Video Conferencing is a web application which will help user to connect to your friends and colleagues, and share screen chat etc.

2.2 **Problem Statement**

Create a Video Conferencing Application:

Approach: Implement the below feature in your application.

1. Enter the Room

2. Audio and Video on off feature

3. Share Screen

4. Chat

**2.3 Proposed Solution**

The solution proposed here is in the form web application that provide an ability to video call and share screen with chat functionalities by adding n number of users in same room

(multipeer) application

Further Improvements

1. Make it a multi room application with more consistency and better features for file sharing etc.

**2.4 Tools Used**

Python Programming language and Django as a framework is used to build the application

* + PyCharm and Atom are used as IDE
  + Database SQLite is used to maintain data of users
  + Python Django and JavaScript is used for backend development
  + Frontend development is done using -: HTML/CSS
  + Git/Github desktop is used as version Control
  + Github
  + AWS/Heroku are used for hosting the application
  + Libraries Used – WebRTC, Django Channels, WebSockets etc

**2.5 Constraints**

The application should be highly user friendly and interactive so that even children can use users should not be required to know any of the workings.

**3. Design Details**

**3.1 Process Flow**

**3.2 Error Handling**

Should errors be encountered, an explanation will be displayed as to what went wrong. An error will be defined as anything that falls outside the normal and intended usage.

**3.3 Performance**

Performance is going to be very important for this project. For everything to run smoothly the SDPS should be connected to the peers and then websocket and django channels should be run properly

**3.4 Reusability**

The code written and components used should have ability to be reused with no problems i.e., use all concepts of OOPs programming and modules will be defined for every part of code

**3.5 Resource Utilization**

When any task is performed, it will likely use all the processing power available until that function is finished.

**3.6 Deployment**

