****

**Project Report**

**Video-Conferencing Application**

|  |
| --- |
| **Name: Abdullah Mohsin**  **ROLL-NO:23fa-048-st**  **Semester:3rd**  **INSTRUCTOR: Shagufta Aftab (SA1)**  **COURSE CODE: SET-214**  **DATE OF SUBMISSION:02-JAN-2025** |

# ACKNOWLEDGEMENT

I am writing this acknowledgement to tell the effort done my teacher in teaching SDLC (Software Development Life Cycle). She taught us about how to do project management. We have learned different kind of models such as Agile, V-Model, Incremental Model, Waterfall, Spiral Model and Iterative Model. We have assigned the project of SDLC from our teacher miss Shagufta Aftab. I have chosen video conferencing application. First Milestone is to give proposal and in second Milestone, we to present our project and give presentation and third Milestone to finally give full project. Our teacher gives us inspiration and idea for developing the application efficiently. We have used Scrum as Software Development Model. These three milestone gives us idea of how to do project step by step without any hurdle. We understand the cycle of SDLC, what are the steps involved in the development of any software. Planning, Requirement gathering, designing, coding, testing, deployment and maintenance. I wanted to acknowledge the whole our done by our teacher Miss: Shagufta Aftab.

# ABSTRACT

I have chosen a conferencing app, Conveyor O-R-C, for my SDLC project. It has features like Remote Desktop, a To-Do list, and real-time translation message, just like Zoom and Google Meet. The real-time translation in message helps people talk easily, while Remote Desktop makes it simple to share content and engage with others. In addition to these features, Conveyor O-R-C aims to bridge communication gaps and enhance productivity for users across different industries. The To-Do List feature helps users stay organized by allowing them to track tasks and deadlines directly within the app, making it a comprehensive tool for both individual and team productivity.

One of the standout features, real-time translation, ensures that language barriers are no longer an obstacle, fostering seamless communication among users from diverse linguistic backgrounds. This feature is particularly beneficial for global teams, educational institutions, and multicultural events.

The Remote Desktop functionality further distinguishes Conveyor O-R-C by enabling efficient collaboration during virtual meetings, presentations, or technical support sessions. Users can easily share their screens or access another device remotely, reducing the need for additional software and streamlining workflows.

Conveyor O-R-C follows a structured approach within the Software Development Life Cycle (SDLC), focusing on user-centric design and iterative improvements. The project emphasizes innovation and usability, ensuring the app remains competitive in the ever-growing market of conferencing tools. By integrating advanced features and addressing common pain points in virtual communication, Conveyor O-R-C aims to redefine how people connect and collaborate online.

# TABLE OF CONTENTS

Contents

[ACKNOWLEDGEMENT 2](#_Toc186655141)

[ABSTRACT 3](#_Toc186655142)

[TABLE OF CONTENTS 4](#_Toc186655143)

[CHAPTER 1 6](#_Toc186655144)

[INTRODUCTION 6](#_Toc186655145)

[1.1 Motivation and Project Overview 6](#_Toc186655146)

[1.2 Background 6](#_Toc186655147)

[1.3 Problem Statement 6](#_Toc186655148)

[1.4 Aim and Objectives 6](#_Toc186655149)

[1.5 Scope of the Project 6](#_Toc186655150)

[1.6 Contributions of the thesis 7](#_Toc186655151)

[1.7 Structure of the thesis 7](#_Toc186655152)

[CHAPTER 2 8](#_Toc186655153)

[LITERATURE REVIEW/BACKGROUND 8](#_Toc186655154)

[2.1 Background 8](#_Toc186655155)

[The purpose of this project is to develop a user-friendly video conferencing application that meets system requirements and ensures smooth communication. The application should provide efficient performance, without any hurdles in video or audio streaming, even under varying network conditions. The focus is on creating a reliable platform that offers a flexible user experience for individuals and teams. 8](#_Toc186655156)

[2.2 Related Work/Literature Review 8](#_Toc186655157)

[2.3 Comparative Study 8](#_Toc186655158)

[CHAPTER 3 9](#_Toc186655159)

[Research/System Design 9](#_Toc186655160)

[3.1 Research /System Methodology 9](#_Toc186655161)

[3.1.1 Software Requirements 9](#_Toc186655162)

[3.1.2 Hardware Requirement 9](#_Toc186655163)

[3.1.3 System Constraints 9](#_Toc186655164)

[3.1.4 External Interface Requirements 10](#_Toc186655165)

[3.2.1 Use Cases: 10](#_Toc186655166)

[3.2.2 Activity/flow Chart Diagram: 11](#_Toc186655167)

[3.2.3 Sequence Diagram/Working Mechanism : 11](#_Toc186655168)

[3.2.4 ER-DIAGRAM: 12](#_Toc186655169)

[3.3 Non-functional Requirements 12](#_Toc186655170)

[3.3.1 Performance Requirements 12](#_Toc186655171)

[3.3.2 Security Requirements 12](#_Toc186655172)

[CHAPTER 4 13](#_Toc186655173)

[System Implementation 13](#_Toc186655174)

[4.1 System Development Process 13](#_Toc186655175)

[4.2 System Simulation 13](#_Toc186655176)

[4.4 Algorithms/ API/ Packages Used 14](#_Toc186655177)

[4.4 Mathematical Model 14](#_Toc186655178)

[CHAPTER 5 15](#_Toc186655179)

[RESULTS AND DISSCUSSION 15](#_Toc186655180)

[4.5 Summary 15](#_Toc186655181)

[CHAPTER 6 15](#_Toc186655182)

[CONCLUSION AND FUTURE DIRECTIONS 15](#_Toc186655183)

[6.1 Conclusion: 15](#_Toc186655184)

[6.2 Future Directions 15](#_Toc186655185)

# CHAPTER 1

# INTRODUCTION

Here, I have worked on a very interesting project video conferencing to meet the course software development lifecycle

## Motivation and Project Overview

I have inspired by zoom, TeamViewer and google-meet platform, so I wonder to make similar application. Video conferencing is the best choice to understand practical implementation of SDLC model. I have used scrum as model to make a productive application.

## Background

Overall background is to create a functional, secure and scalable software. Video-Conferencing application serve as a path for conferencing meeting. Planning, requirement, design, development and deployment are the phases involved in overall development.

## Problem Statement

The problem statement is to make communication easily, remove the physical barrier and from this you can communicate easily. This is basically developed for developers, remote desktop connection become easy as if client have bug so developer fix it by remotely without any hurdle.

## Aim and Objectives

The main aim and objective is to create a user friendly application. Which is friendly to use and end user does not feel any difficulty or trouble in using this application.

## Scope of the Project

The scope of the project is that face to face communication, to-do listing of task, group message box is different and screen sharing. These all are in in scope and out scope includes translation, video filter and alarm clock etc.

## Contributions of the thesis

This thesis contributes to the development of a video conferencing application by:

1. **Design and Implementation**: Developing a user-friendly interface and robust backend to support seamless video and audio communication.
2. **Innovative Features**:
   * Real-time video face to face communication.
   * Integration of chat and screen sharing functionalities.
   * Blocked user and present user automatically view real-time visualize.
3. **Performance Optimization**: Implementing algorithms for efficient bandwidth usage.
4. **Comparative Analysis**: Providing insights through a detailed study of existing platforms like Zoom, Google Meet, and Microsoft Teams, highlighting areas for improvement.
5. **User-Centric Design**: Focusing on simplicity, security, and scalability to fulfil various user needs.

## 1.7 Structure of the thesis

 **Chapter 1: Introduction**  
Provides an overview of the project, its objectives, and the scope of the work.

 **Chapter 2: Background and Literature Review**  
Explains the foundational knowledge, related works, and a comparative analysis of existing platforms.

 **Chapter 3: System Requirements and Methodology**  
Details the software and hardware requirements, system constraints, and functional and non-functional requirements, supported by diagrams.

 **Chapter 4: System Design and Implementation**  
Describes the development process, architecture, algorithms, and database design used in building the application.

 **Chapter 5: Results and Discussion**  
Presents the outcomes of the system, testing results, and a discussion of performance and challenges.

 **Chapter 6: Conclusion and Future Work**  
Summarizes the key contributions, evaluates the system’s effectiveness, and suggests potential enhancements for future work.

# CHAPTER 2

# LITERATURE REVIEW/BACKGROUND

## 2.1 Background

## The purpose of this project is to develop a user-friendly video conferencing application that meets system requirements and ensures smooth communication. The application should provide efficient performance, without any hurdles in video or audio streaming, even under varying network conditions. The focus is on creating a reliable platform that offers a flexible user experience for individuals and teams.

## 2.2 Related Work/Literature Review

To gain knowledge and inspiration for this project, I studied:

1. **Networking Concepts**:
   * Explored foundational topics like network protocols, data transmission, and real-time communication from course books.
   * Understood the importance of technologies such as WebRTC for peer-to-peer communication.
2. **Existing Applications**:
   * Researched the features, limitations, and user experiences of video conferencing platforms.
   * Gained insights into critical factors like video quality, and scalability.  
     This research provided the necessary knowledge to design and develop a video conferencing application.

## 2.3 Comparative Study

A comparative analysis was conducted to evaluate popular video conferencing platforms, such as:

1. **Google Meet**:
   * Strengths: Simple interface, integrates well with Google Workspace, supports up to 100 participants for free.
   * Weaknesses: Limited advanced features in the free version.
2. **Zoom**:
   * Strengths: Widely used, excellent scalability, breakout rooms, and virtual backgrounds.
   * Weaknesses: Concerns about security in early versions, time limits on free calls.
3. **Microsoft Teams**:
   * Strengths: Robust collaboration tools, integrates well with Office 365, strong security features.
   * Weaknesses: Interface may feel cluttered for some users.
4. **Team Viewer:**

* Strengths: Team-viewer allow remote desktop connection means user can access remotely and connection with Client.
* Weaknesses: It does not provide more flexible experience; it is not efficient for huge traffic.

# CHAPTER 3

# Research/System Design

## 3.1 Research /System Methodology

The methodology used for the development of the video conferencing application includes:

1. **Research**: Understanding the requirements and existing solutions like Zoom and Google Meet.
2. **Development Approach**: Agile methodology was used to deliver small functional increments and incorporate user feedback.

### 3.1.1 Software Requirements

The requirement includes video conferencing application, create room, join room, to-do listing and remote desktop for remote access. User have control for viewing camera, audio and share screen and chat message to team member

### 3.1.2 Hardware Requirement

 **Client-Side**:

* Minimum: Dual-core processor, 4GB RAM, webcam, microphone, and stable internet connection.
* Recommended: Quad-core processor, 8GB RAM, HD webcam, and high-speed internet

 **Server-Side**:

* Multi-core processor, 16GB RAM, SSD storage for quick data access, and a high-bandwidth internet connection.

### 3.1.3 System Constraints

* **Network Dependency**: Requires a stable internet connection for real-time communication.
* **Device Compatibility**: The app must work across various devices and operating systems.
* **Latency**: Low latency is critical for smooth video and audio communication.

### 3.1.4 External Interface Requirements

* **User Interface**:

It should have a responsive design with clear options for video, audio, and chat.

* **Hardware Interfaces**:

Supports webcams, microphones, and speakers.

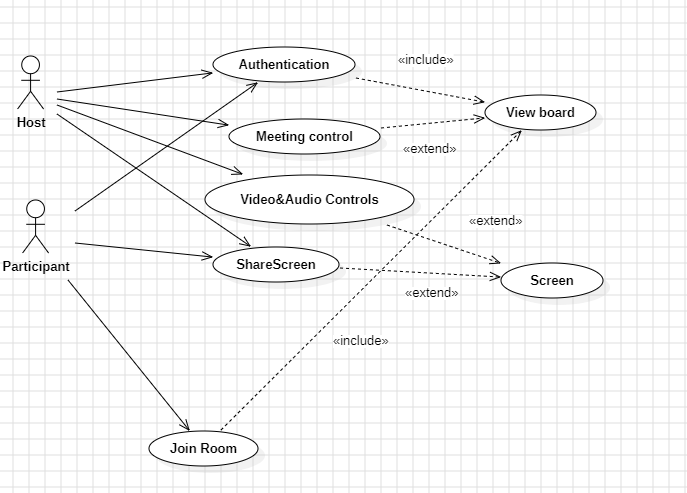
* **Software Interfaces**:

Integration with third-party tools like calendars or cloud storage for meeting scheduling and file sharing.

**3.2 Functional Requirements**

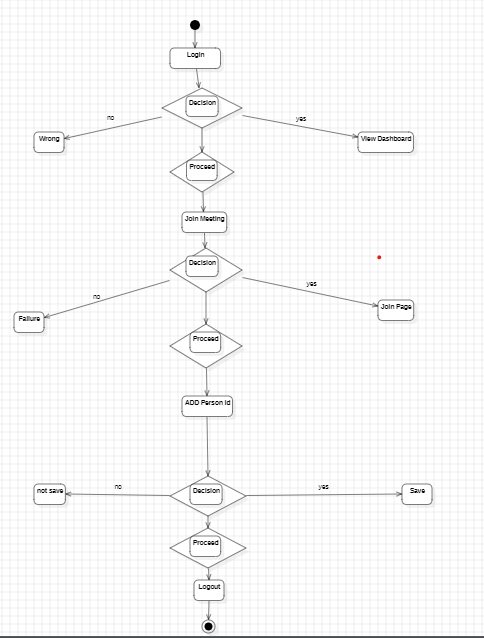
The functional requirement includes face to face communication, remote desktop meeting, login authentication with google and video&audio controls. Screen sharing, create room and join room are considered as functional requirement. To do listing, contact of person id and name adding, removing history of chat messages to clean storage, logout and message box where you can text and upload files

### 3.2.1 Use Cases:



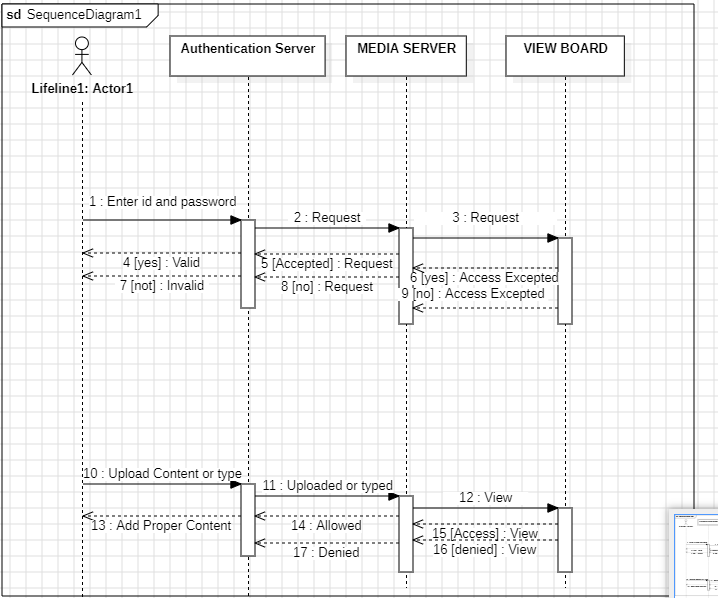
###### **Fig1.1:UseCase**

### 3.2.2 Activity/flow Chart Diagram:



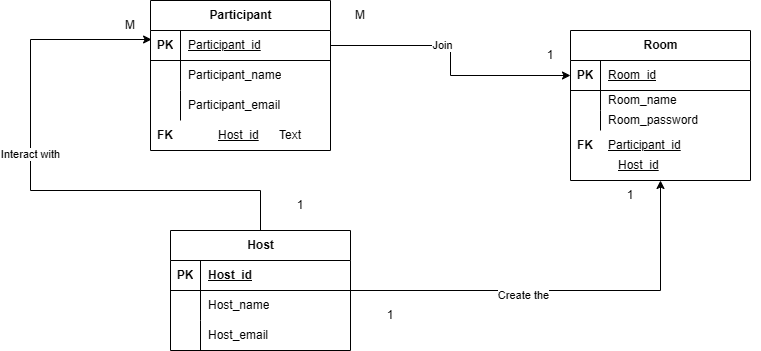
###### **Fig1.2: ACTIVITY DIAGRAM**

### 3.2.3 Sequence Diagram/Working Mechanism :



###### **Fig1.3: SEQUENCE DIAGRAM**

### 3.2.4 ER-DIAGRAM:



###### **Fig1.4: ENTITY RELATIONSHIP DIAGRAM**

## 3.3 Non-functional Requirements

The following are the non-functional requirements of our application:

* The system should support up to 100 participants per meeting.
* Video resolution should adjust automatically based on bandwidth.
* Response time for user actions (like joining a meeting) should be under 2 seconds.
* Security and scalability in this app means it can handle huge amount of traffic and data should be hidden.

### 3.3.1 Performance Requirements

### 3.3.2 Security Requirements

* **Data Encryption**: End-to-end encryption for audio, video, and messages.
* **Authentication**: Secure login with two-factor authentication.
* **Data Protection**: User data should comply with GDPR and other privacy regulations.

# CHAPTER 4

# System Implementation

## 4.1 System Development Process

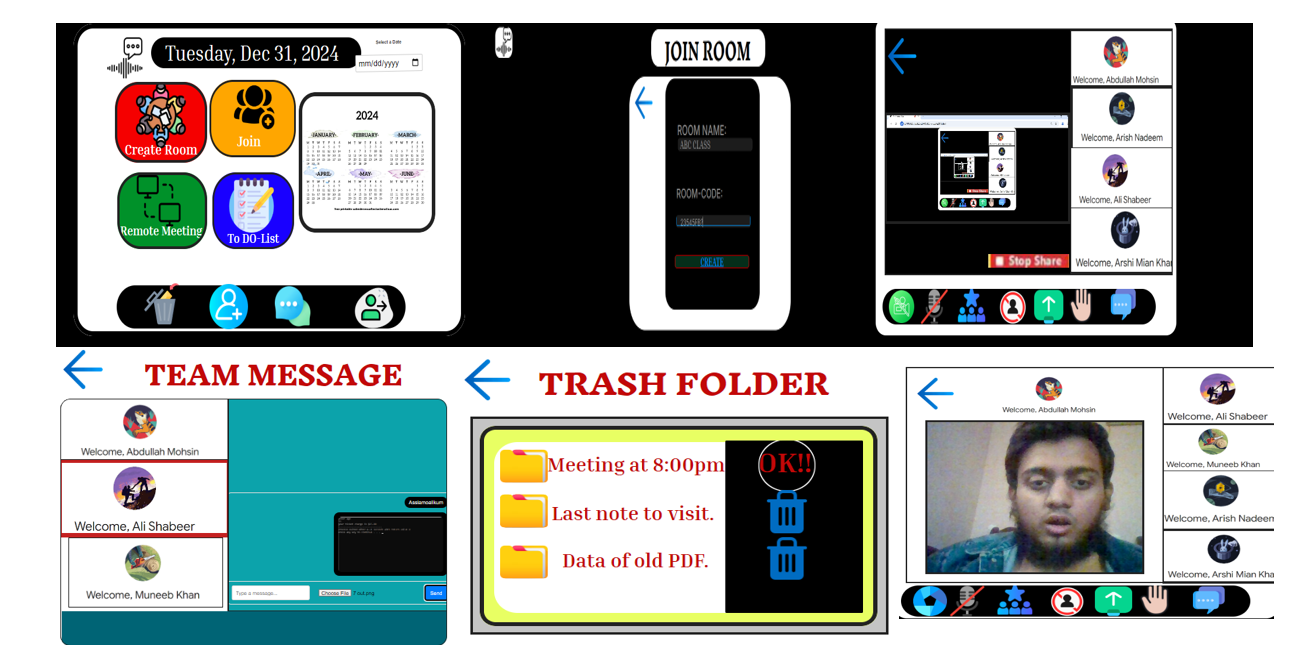
The development process followed a structured approach to ensure the successful implementation of the video conferencing application:

1. **Requirement Analysis**: Determining the key features such as video/audio calls, chat, and screen sharing.
2. **System Design**: Creating a blueprint for the user interface, backend services, and database.
3. **Implementation**: Developing the frontend and backend components using modern web technologies and APIs like calendar, socket connection, google account. We used React Js Framer tool for developing application.
4. **Testing**: Conducting extensive tests for audio-video synchronization and system scalability.
5. **Deployment**: Hosting the application on a secure server and making it accessible to users.

## 4.2 System Simulation

System simulation was conducted to test functionality in various scenarios:

* **Single-User Testing**: Verified that video and audio streams work seamlessly for one-to-one calls.
* **Multi-User Testing**: Simulated group calls with varying network speeds to check performance and stability.



###### **Fig1.5: Simulation Results**

**4.3. System Framework/System Architecture**

1. **Frontend**: The frontend of the system will be built using HTML, CSS, and JavaScript frameworks like React to create a responsive user interface. React will help in building dynamic, interactive, and fast user interfaces that adapt to different screen sizes, ensuring an optimal experience on desktops, tablets, and mobile devices.
2. **Backend**: The backend will be implemented using React.js to handle requests from users, manage user data, and establish communication sessions. This backend system will process tasks such as user authentication, real-time messaging, and video call management, ensuring smooth and secure communication between users.

## 4.4 Algorithms/ API/ Packages Used

* Noise reduction and echo cancellation for audio clarity.
* Video compression algorithms to reduce bandwidth usage.

 **APIs**:

* REST APIs for user authentication and data retrieval.
* Socket for connection with other person (Person to person)
* Google account API for interaction.

## 4.4 Mathematical Model

The mathematical foundation includes:

1. **Bandwidth Calculation**: B=R×TB = R \times TB=R×T where BBB is the required bandwidth, RRR is the data rate, and TTT is the duration.
2. **Video Compression**:  
   Compression ratios are applied to reduce video size without significant quality loss.

# CHAPTER 5

# RESULTS AND DISSCUSSION

## 4.5 Summary

The video conferencing application is designed to provide a seamless user experience for real-time communication. It uses WebRTC for efficient video and audio calls, a secure backend for data management, and advanced algorithms for performance optimization. Extensive testing ensures reliability under various network conditions and user loads.

# CHAPTER 6

# CONCLUSION AND FUTURE DIRECTIONS

## 6.1 Conclusion:

The conclusion is that SDLC cycle give idea of designing and implementing software without software failure. Different model understood by SDLC which includes waterfall, scrum, agile and Kanban. These model make the project easily to deploy. I have used scrum as model for my application. Choosing the model according to the requirement. Best requirement effects all over performance

## 6.2 Future Directions

We will add face to face real time translation feature, more real time experience. Augmented reality as for education purpose and enhance the icon and more updatation according to end user needs