VidCall 

Video Conferencing App

BY:

Sridhar Sodhi 2021104

Vinayak Goel 2021113

Vinayak Arora 2021112

Saksham Pandey 2021486

Introduction

Video Calls in this day and age are an essential form of communication. They are used for a variety of purposes, such as conducting classes in schools or colleges, conferences and meetings conducted online through video calls and many more.VidCall is a video conferencing app designed to bring people together, whether for personal chats or professional meetings. It allows users to easily create, join, and manage video calls with just a few taps.

Objectives

The Objective of this Project is to create an android application using Kotlin which utilizes the Jitsi meet sdk, database in the form of ROOM database and consist of additional features such as accessibility,location,sensor etc.

Technologies Used

The Application is built on Kotlin, which is one of the most popular languages in creating android apps in the industry. We utilize its different features such as Compose and Semantics to create components, and add accessibility features to our application. We use the Rooms database to create a local database in our applications and store the data.

The application also utilizes Jitsi meet SDK, which provides a library in kotlin to add video conference functionality to our application. It allows us to create a new room, join an existing room and an entire set of video conferencing facilities just with a few lines of code.

Activities

HomeScreen Activity :-

HomeScreen Activity is the initial activity of the app i.e. the starting point. of the app. To display the HomeScreen Activity, a column composable was used and 2 button composable in the column scope. The onclick methods of button starts a new activity called user login. On clicking either of the 2 buttons,UserLogin activity is started and the action is passed as a parameter through the intent. If Existing User is clicked, then action = “login” is passed through intent. If New User is clicked, then action = “newuser” is passed through intent and a new activity is started.

UserLogin Activity :-

User Login Activity is the second activity in the flow which is started by the HomeScreen Activity.To display the User Login Activity, a column composable was used. TextFields and 2 button composable were created in the column scope. The onclick methods of Login / SignIn button starts a new activity i.e. the MainActivity. Sign In :- The UserName, password taken as user inputs. The values are then stored in user database. Repeated usernames or when passwords do not match, throws an error. Log In :- The username and password are verified from the user database and then if correct details are found user is able to log in.The username is sent as parameter through intent to start the MainActivity.

Main Activity :-

This is the screen which contains the core functionality of the app, that is providing the user with the ability to create video call rooms or join already

existing rooms. This functionality is built using Jitsi meet SDK for kotlin.To display the MainActivity of the app, a column was used inside a composable function. There are 4 composable buttons, which allow the user to Create a new room, join an existing room, Check details of past meetings and Sign out respectively.The default options of a Jitsi meet are set for every user like setting up a server, and other relevant flags. The user can click on “Create new room” and enter a roomname to join a new room, or click on “Join Existing room” to join a room which already exists with same name. “Meeting Details” give user the information about past meetings and “Sign out” takes the user back to home screen.The User can create a new room using Jitsi meet SDK, i.e enter a room name and create a fresh video meet over the server. Other users can join the meet by using the same room name. The User can join an already existing meet created in (1) or over the Jitsi website by entering the room name.The User can check the details of his past meetings such as meeting name, the number of participants.

Accessibility

To ensure the app is accessible to users with disabilities, several key features have been implemented, focusing on enhancing usability across various forms of impairments, including visual, auditory, and motor disabilities. These enhancements are in line with modern accessibility standards and best practices.

Implementation of Accessibility Features:

Screen Reader Compatibility:

Description: All interactive elements within our app, including buttons and input fields, are annotated with descriptive labels that are recognized by

screen readers. This allows visually impaired users to navigate and interact with the app using auditory feedback.

Technical Details: Utilizes the contentDescription attribute in Android to provide text descriptions that screen readers can convey to users.

Impact: Enables blind and visually impaired users to understand the functionality of each UI component without seeing the screen.

High Contrast Themes:

Description: Our app supports high-contrast colour themes, which are designed to offer better visibility and reduce eye strain, particularly beneficial for users with low vision.

Technical Details: Implementation adheres to the Web Content Accessibility Guidelines (WCAG) for colour contrast ratios, ensuring that text stands out against background colours sufficiently.

Impact: Makes the app's interface easier to read and navigate, enhancing visual clarity for users with visual impairments.

Sensor feature

The MainActivity includes a critical feature that leverages the device's built-in sensors to enhance user experience during video calls. This feature is primarily focused on monitoring ambient light levels to ensure optimal lighting for video communication.

Functionality:

Initialization: Upon launching the MainActivity, the application initializes the SensorManager and registers the light sensor to start receiving updates on ambient light levels.

Light Level Monitoring: The application continuously checks the light sensor's data. When the sensor detects light levels that are too low for a good video call experience, it triggers a response.

User Alert: If low light levels are detected, the application displays a modal warning dialog to the user. This dialog advises the user to increase their environment's lighting. The dialog contains a message: "Your screen brightness is quite low, which might make it hard to see what's on the screen. Try turning it up for a clearer view."

The sensor feature is integrated in such a way that it complements the core video call functionality provided by the Jitsi Meet SDK. While users engage in activities like creating or joining a room, the light sensor continuously monitors the environment to provide real-time feedback on the lighting conditions.

Room Database

USERS Database :-
The Database entity consists of the following columns :-
i) Username
ii) Password
iii) Meetings.
The meetings which the user was previously a part of were stored in the form of a string separated by space.

MEETS Database :-

The Database entity consists of the following columns :-

i) Id

ii) Participants

The participants in a meeting were stored in the form of a string separated by space.

A singleton instance of both databases was created and used throughout the app by passing the specific application context.