Echos - Clubhouse Kotlin App with Agora Integration

# MainActivity.kt

package com.example.echos  
  
import android.content.Intent  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.material3.\*  
import com.example.echos.ui.theme.EchosTheme  
  
class MainActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContent {  
 EchosTheme {  
 Surface(modifier = Modifier.fillMaxSize(), color = MaterialTheme.colorScheme.background) {  
 Button(onClick = {  
 startActivity(Intent(this@MainActivity, RoomActivity::class.java))  
 }) {  
 Text("Join Room")  
 }  
 }  
 }  
 }  
 }  
}

# RoomActivity.kt

package com.example.echos  
  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import com.example.echos.ui.RoomScreen  
import com.example.echos.ui.theme.EchosTheme  
  
class RoomActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContent {  
 EchosTheme {  
 RoomScreen()  
 }  
 }  
 }  
}

# AgoraManager.kt

package com.example.echos  
  
import android.content.Context  
import io.agora.rtc2.RtcEngine  
import io.agora.rtc2.RtcEngineConfig  
  
object AgoraManager {  
 private var rtcEngine: RtcEngine? = null  
  
 fun initialize(context: Context, appId: String, eventHandler: io.agora.rtc2.IRtcEngineEventHandler) {  
 val config = RtcEngineConfig()  
 config.mContext = context  
 config.mAppId = appId  
 config.mEventHandler = eventHandler  
 rtcEngine = RtcEngine.create(config)  
 }  
  
 fun getEngine(): RtcEngine? = rtcEngine  
  
 fun joinChannel(token: String, channelName: String, uid: Int) {  
 rtcEngine?.joinChannel(token, channelName, "", uid)  
 }  
  
 fun leaveChannel() {  
 rtcEngine?.leaveChannel()  
 }  
  
 fun muteLocalAudioStream(mute: Boolean) {  
 rtcEngine?.muteLocalAudioStream(mute)  
 }  
  
 fun destroy() {  
 RtcEngine.destroy()  
 }  
}

# RoomScreen.kt

package com.example.echos.ui  
  
import android.app.Application  
import androidx.compose.foundation.layout.\*  
import androidx.compose.material3.\*  
import androidx.compose.runtime.\*  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.platform.LocalContext  
import androidx.compose.ui.unit.dp  
import com.example.echos.AgoraManager  
import io.agora.rtc2.IRtcEngineEventHandler  
  
@Composable  
fun RoomScreen() {  
 val context = LocalContext.current  
 val appId = "YOUR\_AGORA\_APP\_ID"  
 val token = "YOUR\_AGORA\_TEMP\_TOKEN"  
 val channelName = "testchannel"  
  
 val isMuted = remember { mutableStateOf(false) }  
  
 LaunchedEffect(Unit) {  
 AgoraManager.initialize(context, appId, object : IRtcEngineEventHandler() {})  
 AgoraManager.joinChannel(token, channelName, 0)  
 }  
  
 DisposableEffect(Unit) {  
 onDispose {  
 AgoraManager.leaveChannel()  
 AgoraManager.destroy()  
 }  
 }  
  
 Column(modifier = Modifier.fillMaxSize().padding(16.dp)) {  
 Text("Live Room: $channelName", style = MaterialTheme.typography.headlineMedium)  
 Spacer(modifier = Modifier.height(20.dp))  
 Button(onClick = {  
 isMuted.value = !isMuted.value  
 AgoraManager.muteLocalAudioStream(isMuted.value)  
 }) {  
 Text(if (isMuted.value) "Unmute" else "Mute")  
 }  
 }  
}

# AndroidManifest.xml

<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.echos">  
  
 <uses-permission android:name="android.permission.RECORD\_AUDIO"/>  
 <uses-permission android:name="android.permission.INTERNET"/>  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Echos">  
 <activity android:name=".RoomActivity" />  
 <activity android:name=".MainActivity">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN"/>  
 <category android:name="android.intent.category.LAUNCHER"/>  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

# CreateRoomActivity.kt

package com.example.echos  
  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.foundation.layout.\*  
import androidx.compose.material3.\*  
import androidx.compose.runtime.\*  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.unit.dp  
import androidx.lifecycle.viewmodel.compose.viewModel  
import com.example.echos.ui.theme.EchosTheme  
  
class CreateRoomActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContent {  
 EchosTheme {  
 val viewModel: RoomListViewModel = viewModel()  
 var roomName by remember { mutableStateOf("") }  
  
 Column(modifier = Modifier.fillMaxSize().padding(16.dp)) {  
 Text("Create a Room", style = MaterialTheme.typography.headlineMedium)  
 Spacer(modifier = Modifier.height(16.dp))  
 OutlinedTextField(  
 value = roomName,  
 onValueChange = { roomName = it },  
 label = { Text("Room Name") },  
 modifier = Modifier.fillMaxWidth()  
 )  
 Spacer(modifier = Modifier.height(16.dp))  
 Button(onClick = {  
 viewModel.addRoom(roomName)  
 finish()  
 }, modifier = Modifier.fillMaxWidth()) {  
 Text("Create")  
 }  
 }  
 }  
 }  
 }  
}

# RoomListViewModel.kt

package com.example.echos  
  
import androidx.lifecycle.ViewModel  
import kotlinx.coroutines.flow.MutableStateFlow  
import kotlinx.coroutines.flow.StateFlow  
import com.example.echos.model.Room  
  
class RoomListViewModel : ViewModel() {  
 private val \_roomList = MutableStateFlow<List<Room>>(listOf(  
 Room("Android Devs"),  
 Room("Kotlin Learners")  
 ))  
 val roomList: StateFlow<List<Room>> = \_roomList  
  
 fun addRoom(name: String) {  
 \_roomList.value = \_roomList.value + Room(name)  
 }  
}

# Room.kt

package com.example.echos.model  
  
data class Room(val name: String)

# RoomActivity.kt

package com.example.echos  
  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import com.example.echos.ui.RoomScreen  
import com.example.echos.ui.theme.EchosTheme  
  
class RoomActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 val roomName = intent.getStringExtra("ROOM\_NAME") ?: "Unnamed Room"  
 setContent {  
 EchosTheme {  
 RoomScreen(roomName)  
 }  
 }  
 }  
}

# RoomScreen.kt

package com.example.echos.ui  
  
import androidx.compose.foundation.layout.\*  
import androidx.compose.material3.\*  
import androidx.compose.runtime.\*  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.unit.dp  
import androidx.compose.ui.platform.LocalContext  
import com.example.echos.AgoraManager  
import io.agora.rtc2.IRtcEngineEventHandler  
  
@Composable  
fun RoomScreen(roomName: String) {  
 val context = LocalContext.current  
 val appId = "YOUR\_AGORA\_APP\_ID"  
 val token = "YOUR\_TEMP\_TOKEN"  
 val isMuted = remember { mutableStateOf(false) }  
  
 LaunchedEffect(Unit) {  
 AgoraManager.initialize(context, appId, object : IRtcEngineEventHandler() {})  
 AgoraManager.joinChannel(token, roomName, 0)  
 }  
  
 DisposableEffect(Unit) {  
 onDispose {  
 AgoraManager.leaveChannel()  
 AgoraManager.destroy()  
 }  
 }  
  
 Column(modifier = Modifier.fillMaxSize().padding(16.dp)) {  
 Text("Room: $roomName", style = MaterialTheme.typography.headlineMedium)  
 Spacer(modifier = Modifier.height(16.dp))  
 Button(onClick = {  
 isMuted.value = !isMuted.value  
 AgoraManager.muteLocalAudioStream(isMuted.value)  
 }) {  
 Text(if (isMuted.value) "Unmute" else "Mute")  
 }  
 }  
}

# AgoraManager.kt

package com.example.echos  
  
import android.content.Context  
import io.agora.rtc2.RtcEngine  
import io.agora.rtc2.RtcEngineConfig  
  
object AgoraManager {  
 private var rtcEngine: RtcEngine? = null  
  
 fun initialize(context: Context, appId: String, eventHandler: io.agora.rtc2.IRtcEngineEventHandler) {  
 val config = RtcEngineConfig()  
 config.mContext = context  
 config.mAppId = appId  
 config.mEventHandler = eventHandler  
 rtcEngine = RtcEngine.create(config)  
 }  
  
 fun joinChannel(token: String, channelName: String, uid: Int) {  
 rtcEngine?.joinChannel(token, channelName, "", uid)  
 }  
  
 fun muteLocalAudioStream(mute: Boolean) {  
 rtcEngine?.muteLocalAudioStream(mute)  
 }  
  
 fun leaveChannel() {  
 rtcEngine?.leaveChannel()  
 }  
  
 fun destroy() {  
 RtcEngine.destroy()  
 }  
}

# AndroidManifest.xml

<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.echos">  
  
 <uses-permission android:name="android.permission.RECORD\_AUDIO"/>  
 <uses-permission android:name="android.permission.INTERNET"/>  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Echos">  
 <activity android:name=".CreateRoomActivity" />  
 <activity android:name=".RoomActivity" />  
 <activity android:name=".MainActivity">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN"/>  
 <category android:name="android.intent.category.LAUNCHER"/>  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>