**Chord Viewer - Android Application**

**1. Introduction**

**Chord Viewer** is an Android application developed using **Kotlin** and **Jetpack Compose** in Android Studio. This app was created as a **personal solution** to a problem I faced in my **daily life at church** while playing the keyboard. I needed a convenient way to **store and retrieve song notes in image format**, but manually searching for images was time-consuming.

To address this, **Chord Viewer** allows users to:

* **Upload an image** from their device.
* **Rename the image** for easier identification.
* **Search for stored images** using the assigned name.
* **Display the searched image** on a new screen for quick reference.

The application enhances usability by providing an **intuitive and interactive UI** for efficient image management, making it easier to organize and access important visual references during live performances

**2. Features**

* Upload an image from the device's file storage.
* Rename and save the image with a user-defined name.
* Search for an image by entering the saved name.
* Display the retrieved image in a new screen with smooth navigation.
* Minimalistic and user-friendly UI using Jetpack Compose.

**3. Technologies Used**

* **Programming Language:** Kotlin
* **Framework:** Jetpack Compose
* **IDE:** Android Studio
* **Gradle Dependencies:**
  + AndroidX Libraries
  + Jetpack Compose UI Toolkit
  + AppCompat for backward compatibility
  + Kotlin Coroutines for background processing

**4. Project Structure**

ChordViewer/

|--app/

| |--- src/main/

| | |--- AndroidManifest.xml

| | |---java/com/example/chordviewer/

| | | |--- MainActivity.kt

| | | |--- UploadScreen.kt

| | | |--- SearchScreen.kt

| | | |--- DisplayScreen.kt

| | |---res/

| | | |--- layout/

| | | |--- drawable/

| | | |--- values/

| |--- build.gradle.kts

|--- settings.gradle.kts

|--- gradle.properties

**5. Implementation Details**

**5.1 Upload Image Feature**

* The user selects an image from the device’s file picker.
* The selected image is displayed in a preview.
* The user enters a name to save the image.
* The image is stored in the app’s local storage with the given name.

**5.2 Search and Display Image Feature**

* The user types a previously saved image name.
* The app searches the local storage for a matching file.
* If found, the image is displayed in a new screen.
* If not found, an error message is shown.

**6. Dependencies and Build Configuration**

**6.1 App-Level Gradle (build.gradle.kts)**

plugins {

alias(libs.plugins.android.application)

alias(libs.plugins.kotlin.android)

alias(libs.plugins.kotlin.compose)

}

android {

namespace = "com.example.chordviewer"

compileSdk = 35

defaultConfig {

applicationId = "com.example.chordviewer"

minSdk = 21

targetSdk = 35

versionCode = 1

versionName = "1.0"

testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"

}

}

dependencies {

implementation("androidx.appcompat:appcompat:1.7.0")

implementation("org.jetbrains.kotlinx:kotlinx-coroutines-android:1.6.0")

implementation(libs.androidx.core.ktx)

implementation(libs.androidx.lifecycle.runtime.ktx)

implementation(libs.androidx.activity.compose)

implementation(libs.androidx.material3)

}

**7. User Interface Design**

* **Upload Screen:** Allows users to select and rename an image.
* **Search Screen:** Provides a text field to enter the image name for searching.
* **Display Screen:** Shows the retrieved image upon a successful search.



**8. Application Screenshots**

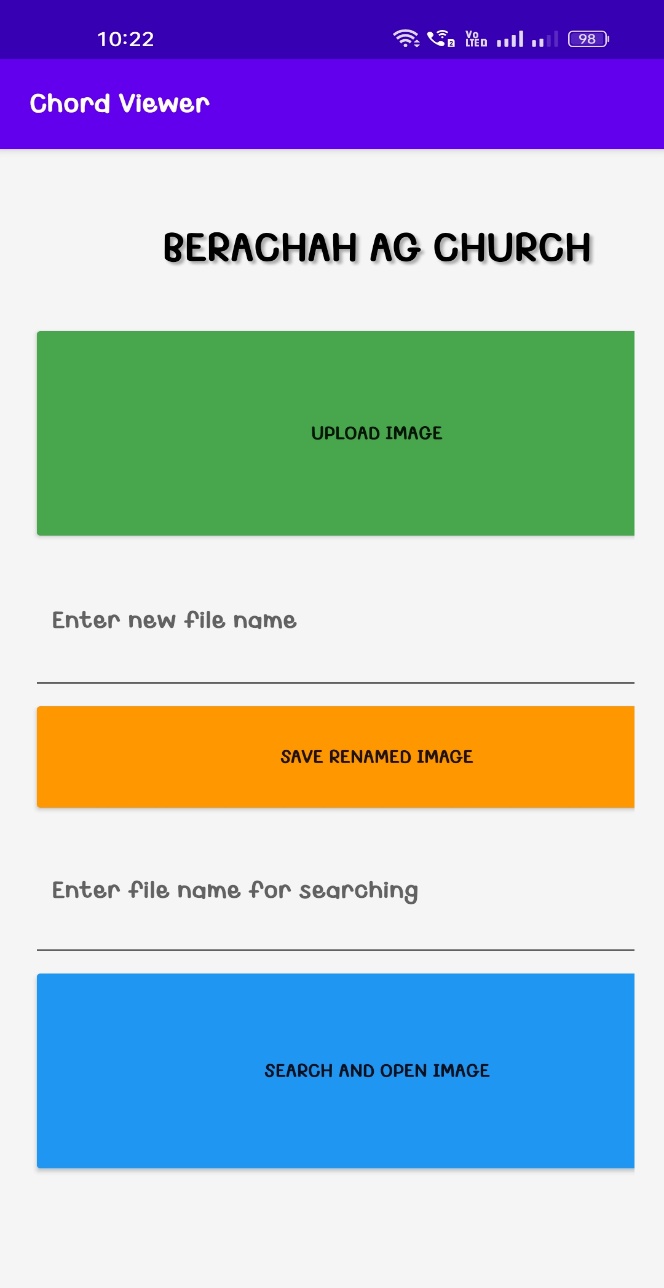
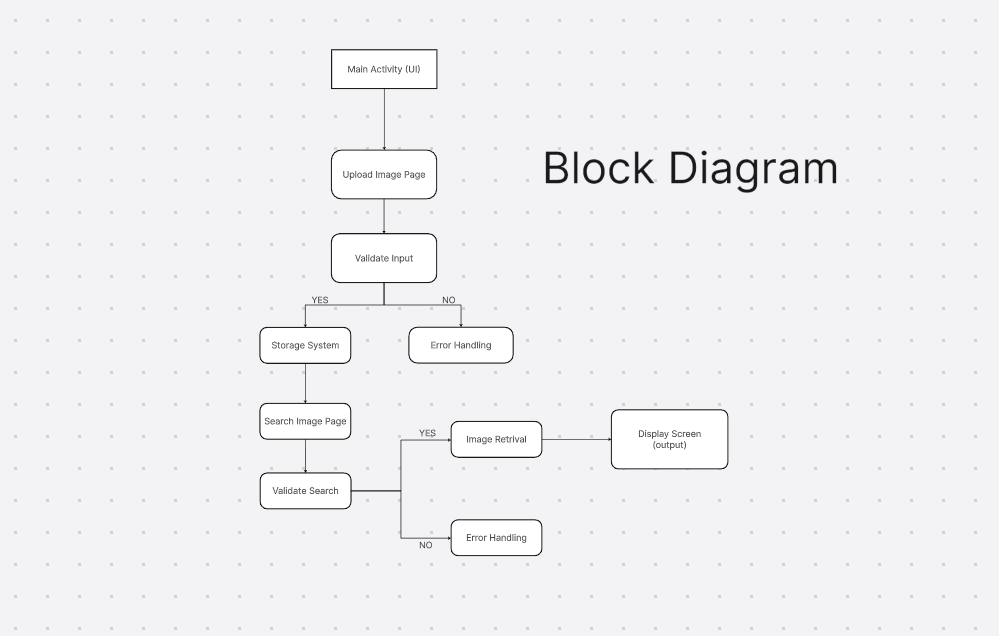
****

Fig 1 **Upload & Search Screen**

Fig 2 **Display Screen**

**9. Block Diagram**

****

**9. Conclusion**

**Chord Viewer** is a lightweight and efficient application designed to simplify **image management with custom naming**, addressing a real-world need I encountered while playing the keyboard at church. By leveraging **Jetpack Compose** for a modern UI and **Kotlin Coroutines** for efficient background processing, the app delivers a **smooth and responsive user experience**.

**Prepared by:** Jenishlin Brisho M  
**Date:** January 16, 2025