Project Documentation: Recallio

1. Introduction

Recallio is a cloud-synced note-taking Android application designed to provide a seamless and secure way for users to create, store, and organize notes across multiple devices. Unlike basic note-taking apps, Recallio integrates **Firebase Authentication** with **OTP verification**, **Firebase Realtime Database** for instant syncing, and additional features such as **labels, reminders, checklists, and image attachments** to make note management more powerful and user-friendly.

The application ensures that user data is always synchronized in real time, allowing access from multiple devices without loss of information. In addition, robust account management features such as **password change/reset** are included for security and reliability.

---

2. Objectives

- Provide a **simple yet powerful note-taking platform**.
- Enable **real-time synchronization** using Firebase Realtime Database.
- Ensure **multi-device login** with consistent user data.
- Enhance **note usability** through additional features:
  - Add **images**, **checklists**, **labels**, and **reminders**.
- Implement **secure authentication** with OTP verification.
- Allow users to **view, edit, and organize notes** efficiently.
- Ensure **account security** with password change/reset features.

---

3. System Requirements

3.1 Hardware Requirements

- **Minimum RAM:** 4 GB
- **Recommended RAM:** 8 GB or higher
- **Processor:** Intel i3 or higher (or equivalent AMD)

- **Storage:** At least 1 GB of free space for Android Studio, Gradle cache, and project files
- **Device:** Android 7.0 (Nougat) or above for testing

3.2 Software Requirements

- **Operating System:** Windows 10/11, macOS, or Linux
- **Development Environment:** Android Studio (latest stable version)
- **Programming Language:** Java (with some XML layouts)
- **Database:** Firebase Realtime Database
- **Authentication:** Firebase Authentication with OTP verification
- **Dependencies:**
    - `com.google.firebase:firebase-auth`
    - `com.google.firebase:firebase-database`
    - `com.google.firebase:firebase-storage`
    - `com.github.bumptech.glide:glide` (for image loading)
    - `com.google.android.material:material` (UI components)

---

4. System Architecture

4.1 Overview

The architecture of Recallio is based on **client-server communication** with Firebase as the backend.

- The **client (Android app)** handles UI, note creation, and local data handling.
- The **Firebase Realtime Database** acts as a cloud data store for storing user notes.
- **Firebase Authentication** manages secure login using OTP verification.
- **Firebase Storage** manages uploaded images.

4.2 Components

1. **Authentication Layer** – Handles user sign-up, login, OTP verification, and password changes.
2. **Database Layer** – Stores notes in structured JSON format in Firebase Realtime Database.
3. **Sync Layer** – Provides automatic syncing of notes between devices.
4. **UI Layer** – User-facing screens for creating, editing, and organizing notes.
5. **Notification Layer** – Reminder system using Android AlarmManager/WorkManager.

---

5. Features of Recallio

1. **User Authentication**
    - Secure signup/login using Firebase Authentication.
    - OTP verification for mobile/email-based authentication.
    - Password change and reset options.
2. **Notes Management**
    - Create, edit, and delete notes.
    - Add **labels** to categorize notes.
    - Attach **images** to notes using Firebase Storage.
    - Add **checklists** inside notes for tasks.
    - Add **reminders** with Android’s notification/AlarmManager.
3. **Real-Time Sync**
    - Notes automatically sync across multiple devices.
    - Offline support with Firebase local cache.
4. **Multi-Device Login**
    - Same user account accessible on multiple devices simultaneously.
5. **User Profile Management**
    - Change account password securely.
    - Persistent authentication using Firebase Auth tokens.
6. **Modern UI/UX**
    - Material Design components for an intuitive experience.
    - Dark mode compatibility (optional).

---

6. Database Design

6.1 Firebase Realtime Database Structure

```
users
  ├── userId1
  │     ├── profile
  │     │     ├── email: "example@mail.com"
  │     │     ├── phone: "+91XXXXXXXXXX"
  │     └── notes
  │           ├── noteId1
  │           │     ├── title: "Shopping List"
  │           │     ├── content: "Buy groceries"
  │           │     ├── imageUrl: "https://..."
  │           │     ├── labels: "personal"
  │           │     ├── reminder: "1692878400"
  │           │     ├── checklist:
  │           │     │     ├── item1: true
  │           │     │     ├── item2: false
  │           │     └── timestamp: 1692870000
  │           └── noteId2 ...
  └── userId2 ...
```

---

7. Implementation Details

7.1 Authentication Flow

1. User registers with **phone/email**.
2. Firebase sends OTP → user enters OTP → verified by Firebase.
3. Auth token is stored locally to maintain session.

7.2 Notes Operations

- Notes are saved under each `userId` node in Firebase.
- CRUD operations: Create, Read, Update, Delete via `DatabaseReference`.
- Images are uploaded to Firebase Storage, and URLs are stored in the note object.

7.3 Reminders

- Android AlarmManager/WorkManager schedules notifications.
- Notes with reminders trigger local notifications even offline.

7.4 Syncing

- Firebase automatically syncs data changes to all logged-in devices in real time.
- Uses Firebase offline persistence for temporary offline usage.

---

8. Testing

8.1 Testing Types

- **Unit Testing:**
  Tested utility functions (note creation, label management, date/time handling).

- **Integration Testing:**

  Verified that authentication, database sync, and reminder notifications work together.
- **UI Testing:**

  Ensured correct layout behavior across multiple devices using Android Emulator and real devices.

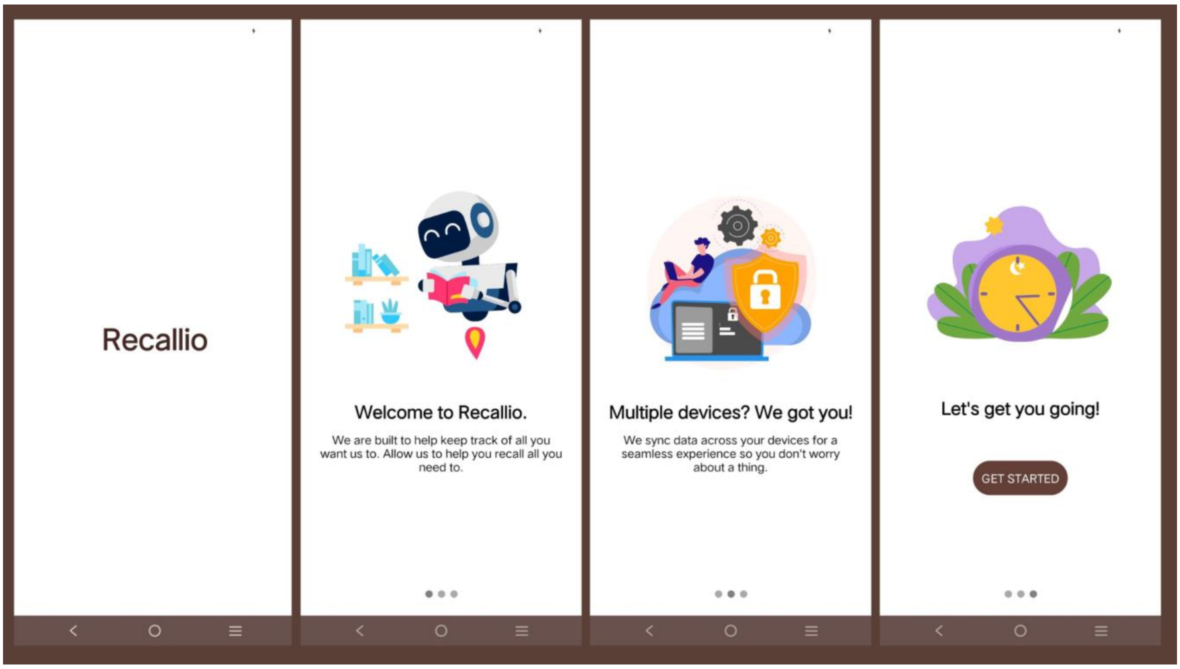
8.2 Tools Used

- Android Studio Emulator
- Firebase Test Lab (optional)
- JUnit for logic tests
- Espresso for UI automation

---

9. Results and Discussion

- The application successfully supports **multi-device sync**.
- Firebase OTP verification adds a **secure authentication layer**.
- CRUD operations for notes are **fast and reliable** due to Firebase’s real-time syncing.
- The additional features (labels, checklist, image attachments, reminders) make Recallio **more functional than basic note apps**.
- The **UI is responsive and lightweight**, meeting Android design standards.

- 

Recallio

Welcome to Recallio.

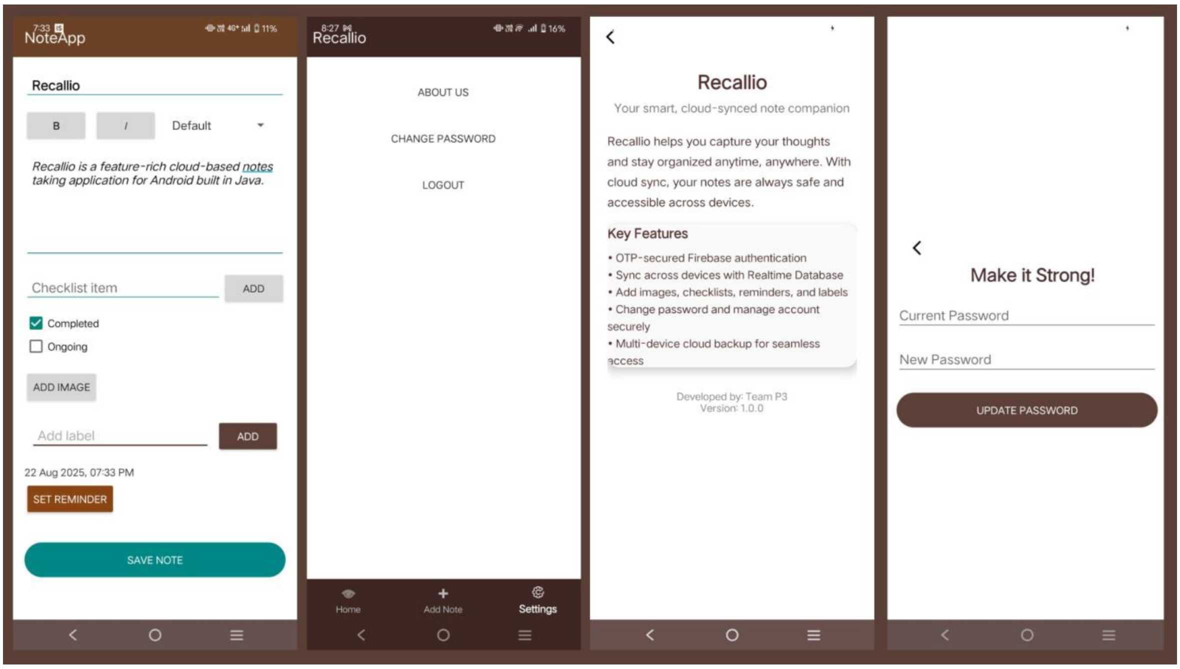
We are built to help keep track of all you want us to. Allow us to help you recall all you need to.

Multiple devices? We got you!

We sync data across your devices for a seamless experience so you don't worry about a thing.

Let's get you going!

GET STARTED



NoteApp

Recallio

B I Default

Recallio is a feature-rich cloud-based notes taking application for Android built in Java.

Checklist item ADD

☑ Completed

□ Ongoing

ADD IMAGE

Add label ADD

22 Aug 2025, 07:33 PM

SET REMINDER

SAVE NOTE

Recallio

ABOUT US

CHANGE PASSWORD

LOGOUT

Home Add Note Settings

Recallio

Your smart, cloud-synced note companion

Recallio helps you capture your thoughts and stay organized anytime, anywhere. With cloud sync, your notes are always safe and accessible across devices.

Key Features

- OTP-secured Firebase authentication
- Sync across devices with Realtime Database
- Add images, checklists, reminders, and labels
- Change password and manage account securely
- Multi-device cloud backup for seamless access

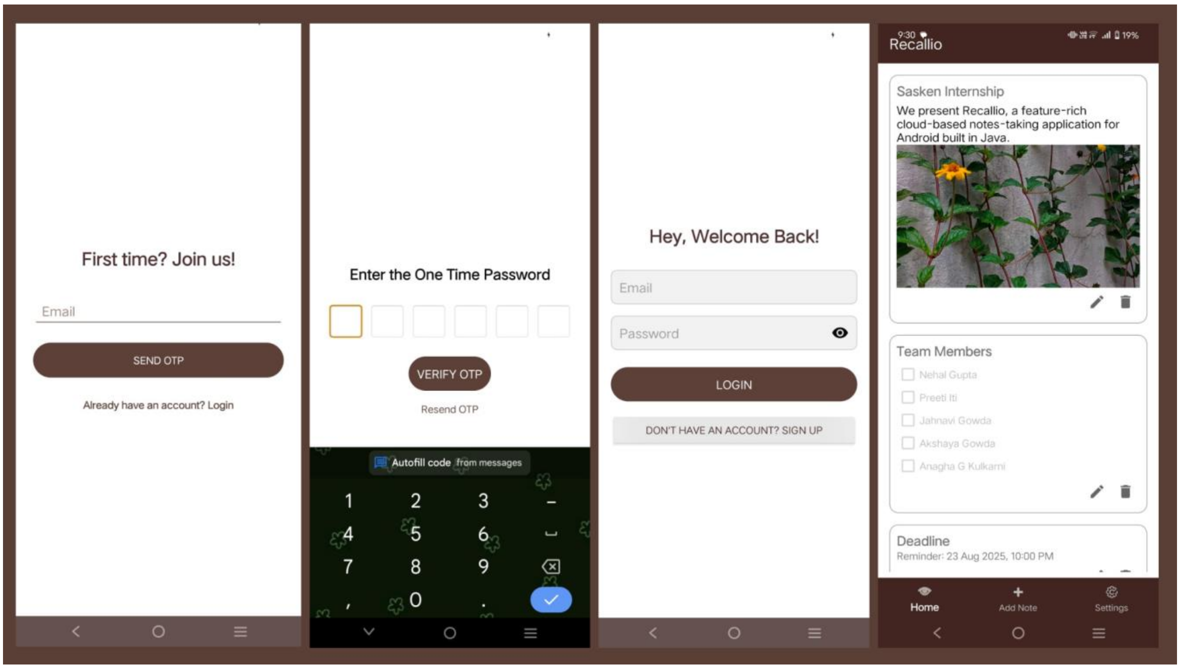
Developed by: Team P3
Version: 1.0.0

Make it Strong!

Current Password

New Password

UPDATE PASSWORD

- 

First time? Join us!

Email

SEND OTP

Already have an account? Login

Enter the One Time Password

VERIFY OTP

Resend OTP

Autofill code from messages

Hey, Welcome Back!

Email

Password

LOGIN

DON'T HAVE AN ACCOUNT? SIGN UP

Recallio

Sasken Internship

We present Recallio, a feature-rich cloud-based notes-taking application for Android built in Java.

Team Members

□ Nehal Gupta

□ Preeti Iti

□ Jahnavi Gowda

□ Akshaya Gowda

□ Anagha G Kulkarni

Deadline

Reminder: 23 Aug 2025, 10:00 PM

Home Add Note Settings

10. Conclusion and Future Scope

Recallio achieves its primary goal of providing a **cloud-synced, secure, and feature-rich note-taking app**. By using Firebase services, it eliminates the need to maintain a separate backend while providing real-time syncing and robust authentication.

Future Scope

- Add **voice notes and speech-to-text** functionality.
- Implement **note-sharing with other users**.
- Add **end-to-end encryption** for sensitive data.
- Include **search and filter by labels** with advanced indexing.
- Provide **web and iOS versions** of Recallio for cross-platform usage.