CCEXTRACTOR
DEVELOPMENT

Ultimate Alarm Clock II - Flutter App

About Me:

Name:	Deepanshu Singh
University:	ITM University, Gwalior, India
Github:	Deepanshuigtm (Deepanshu Singh) · GitHub
LinkedIn:	https://www.linkedin.com/in/deepanshu-singh-6a57b1235/
Resume:	https://drive.google.com/file/d/1IWSMyu32LJUFvA5__nAdFmPO8WQZyo-9/view
Location:	Gwalior, Madhya Pradesh, India
Email:	2207deepanshu@gmail.com
Phone no:	+91 8103702636
Timezone:	IST(GMT +5:30)
Project:	Ultimate Alarm Clock II I CCExtractor
Proposal:	https://github.com/Deepanshuigtm/ccextracter-gsoc-proposal

Abstract:

This project aims to build an Alarm Clock app that develops an innovative alarm clock with intelligent functionalities. This advanced clock is designed to intuitively dismiss alarms based on user activity on their phone, **real-time weather conditions**, and additional customizable parameters. It introduces various challenges to ensure users are fully awake, offering a unique approach to morning routines. Moreover, it incorporates a feature **allowing users to set shared alarms**, promoting collaboration and coordination among multiple individuals. This project **aims to redefine the traditional alarm clock experience, enhancing wake-up routines with modern technology and personalized settings**

Goals:

Here are some goals for the project:

- **Architectural and data flow** changes in the app to boost performance and maintainability. These optimizations will enhance **resource utilization and streamline user experience**.
- Provide **default alarm tones** catering to various preferences, ensuring users have diverse options to choose from.
- **Timer functionality** to the app, extending the utility of the "Ultimate Alarm Clock" beyond traditional alarms. Users can set countdown timers for various purposes, **enhancing their experience**

- Once a feature is completed, I would be adding the **unit tests** for the implemented feature.
- **If there are no violations of YouTube's Terms of Service**, then I want to create a new feature where **users can search for their favorite music** using the youtube_explode Flutter package. They can then set the searched music as their alarm ringtone
- Setup **automatic CI/CD** on Github
- Setup **issue and PR template** on Github

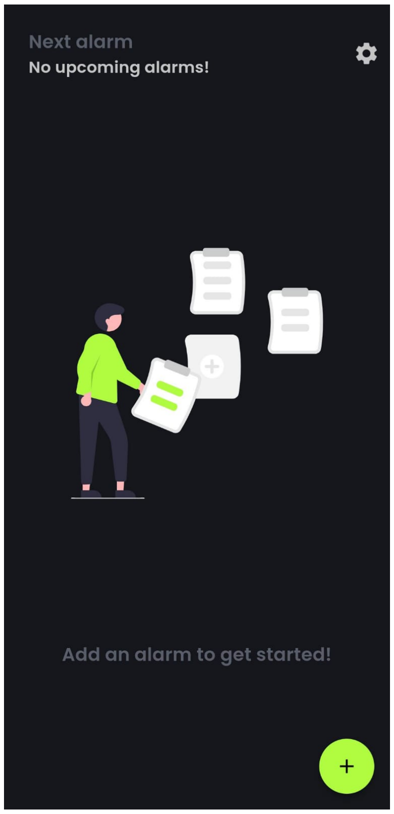
Implementation:

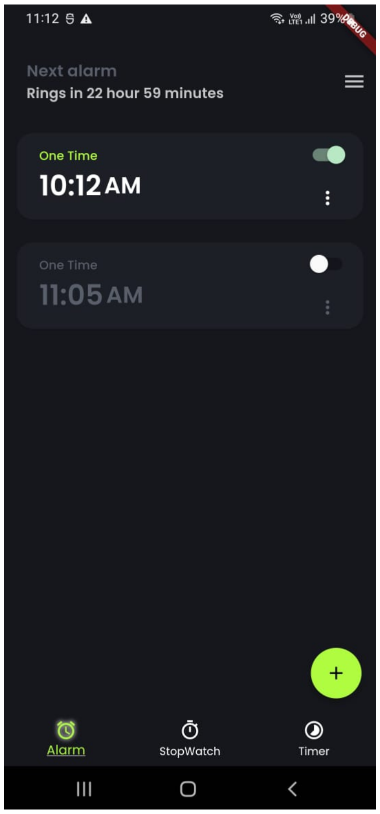
App implementation is divided into the following **Milestones**

(all dated mentioned below are of 2022)

- **Community bonding period (April 19 to May 20)**
    - Interact and engage in the community, get to know my mentors and fellow students. Also, I would try to learn about other ongoing projects in the organisation, especially those related to flutter.
    - I will be discussing the **app flow** in detail with mentors. I would go with UI that is intuitive and easy to use and app flow that is efficient as much as possible.
    - Since this app would be published on the **Google Playstore**, I would start preparing assets like icons.

- **MILESTONE 1 (June 13 to June 26) - Workflow of the app and setup**

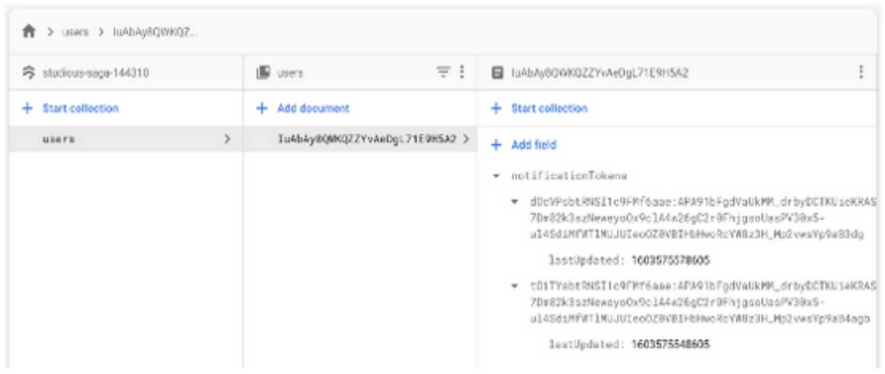




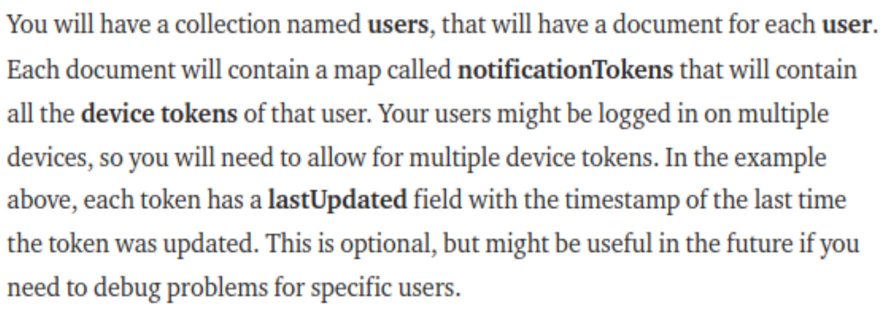
- The homepage displays all set alarms and allows users to update existing ones or create new alarms.
- The "Next Alarm" feature shows the time remaining until the upcoming alarm. Additionally, it includes functions like a stopwatch and timer.

● MILESTONE 2 (June 27 to July 12)- Firebase server setup

- Setup the **firebase project** and set up **Firebase Cloud Messaging** and **Firebase Firestore**.
- Also, learn about the firestore database and research its querying and scaling abilities.
- Set up a [mock server]() to test the features of the server.
- **Firebase Firestore** would be used for storing the data since its better **integration with android** and **improved querying** compared to the firebase's real-time DB.
- Firestore allows users to share alarms with one another in real-time. Users can collaborate on alarm settings, making it convenient for multiple individuals to set alarms for a common purpose.
- The schema for Firebase Firestore consists of collections and documents consist of **Users Collection , Alarms Collection**



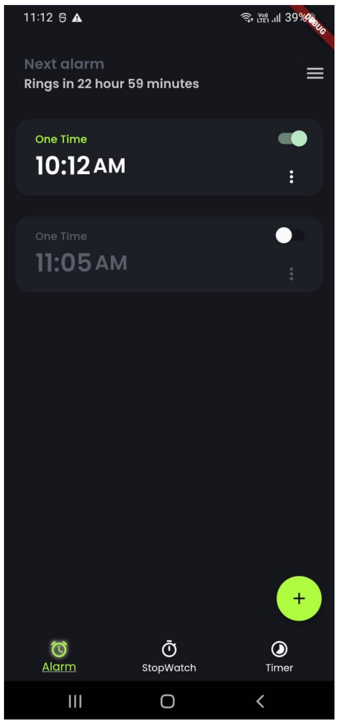
```
users                 IuAbAy8QWKQZZYvAeDgL71E9H5A2
▾ notificationTokens
    ▾ dOcVPsbtRNSI1c9FMf6aae:APA91bFgdVaUkMM_drbyDCTKUicKRAS7Dm82k3szNewayoOx9c1A4a26gC2r0FhjgsoUasPV30x5-ul4SdiMfWT1MUJUIeoOZ8VBIHbHwoRcYW8z3H_Mp2vwsYp9a83dg
        lastUpdated: 1603575570605
    ▾ tO1TYabtRNSI1c9FMf6aae:APA91bFgdVaUkMM_drbyDCTKUi4KRAS7Dm82k3szNewayoOx9c1A4a26gC2r0FhjgsoUasPV30x5-ul4SdiMfWT1MUJUIeoOZ8VBIHbHwoRcYW8z3H_Mp2vwsYp9a84agb
        lastUpdated: 1603575548605
```


You will have a collection named **users**, that will have a document for each **user**. Each document will contain a map called **notificationTokens** that will contain all the **device tokens** of that user. Your users might be logged in on multiple devices, so you will need to allow for multiple device tokens. In the example above, each token has a **lastUpdated** field with the timestamp of the last time the token was updated. This is optional, but might be useful in the future if you need to debug problems for specific users.

```
{
  "message":{
    "token":"bk3RNwTe3H0:CI2k_HHwgIpoDKCIZvvDMExUdFQ3P1...",
    "data":{
      "Phone No" : "+919876543210",
      "Message" : "Calling for getting maths assignment",
      "Urgency Rating" : "8/10"
    }
  }
}
```

- **MILESTONE 3 (July 12 to July 26)- Database and its Schema**
  - **ISAR is used to store for alarm-related data local storage**. It facilitates the efficient and structured management of alarm settings and preferences, ensuring that alarms function smoothly even in offline scenarios.
  - **Offline Functionality**: IsarDb ensures that alarms function seamlessly even in offline scenarios. Users can trust that their alarms will trigger as expected, regardless of their internet connection status
  - **Performance Optimization**: IsarDb optimizes data retrieval, allowing for quick access to triggered alarms and user-specific configurations. It enhances the overall performance and responsiveness of the application.
  - It storing various **settings and preferences**. This storage solution employs key-value pairs to manage and access data :
    - userModel : Stores user-related data in a JSON-encoded format, including user settings and preferences.
    - weather_state : Stores the current weather state information.
    - API keys : Several keys are used for storing API keys securely, allowing the application to access external services.
    - Haptic Feedback : Key-value pairs are used to store and manage user preferences related to haptic feedback settings.
    - Sorted Alarm List : Key-value pairs are used to store and manage user preferences for sorting the alarm list.
    - Theme_value : Stores the selected theme (e.g., dark or light) for the application.

- **MILESTONE 4 (July 26 to August 21)- Home, Settings and Add or Update Alarm View**
  - **Home View**



11:12 39%

Next alarm
Rings in 22 hour 59 minutes

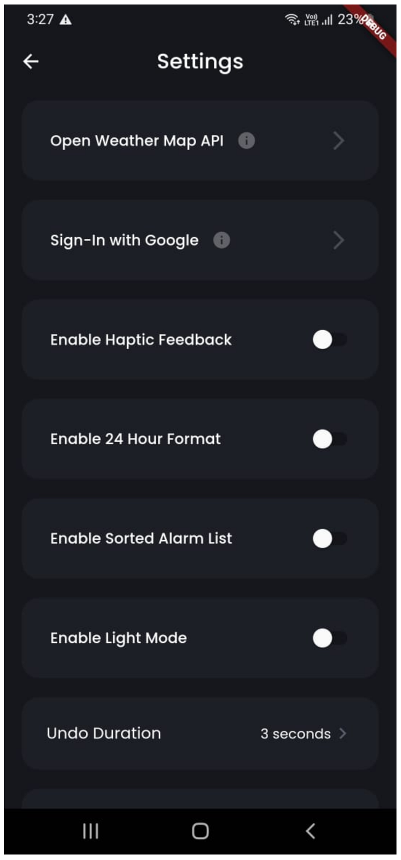
One Time
10:12 AM

One Time
11:05 AM

  - The home view displays all user-set alarms. If the shared alarm is on, it fetches data from Isar DB (local storage) otherwise, it fetches from Firestore.
  - Users can toggle alarms on/off, create new alarms using the floating action button, update, and delete them. Additionally, we offer an undo option on swipe to delete alarms.

  - **Settings View**

  - In Settings View, you can customize various aspects of the app.
  - Set or edit the weather API key for weather-related features.
  - Log in with your Google account.
  - Toggle haptic feedback on/off.
  - Toggle the sorting of the alarm list based on time.
  - Toggle between light and dark modes.
  - Enable 24-hour format to change the time display format to 24-hour notation
  - The undo duration sets the time for swipe-to-delete alarm undo action, defaulting to 3 seconds.



3:27 23%

Settings

Open Weather Map API

Sign-In with Google

Enable Haptic Feedback

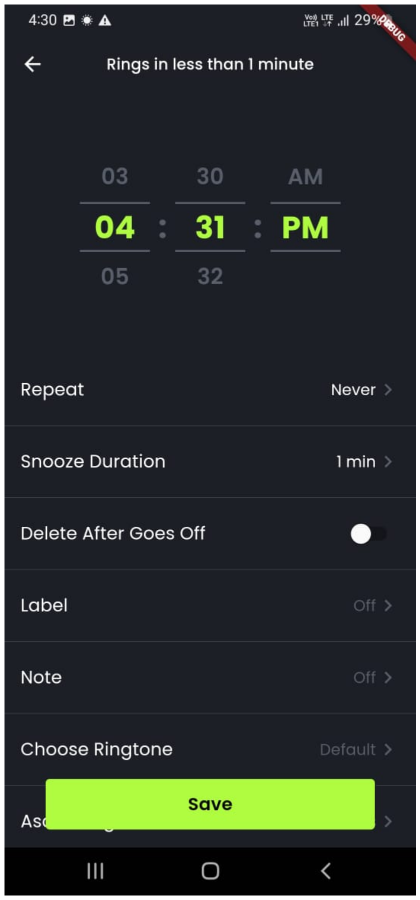
Enable 24 Hour Format

Enable Sorted Alarm List

Enable Light Mode

Undo Duration 3 seconds

- **Add or Update Alarm View**


4:30 29%

Rings in less than 1 minute

03 30 AM
04 : 31 : PM
05 32

Repeat Never

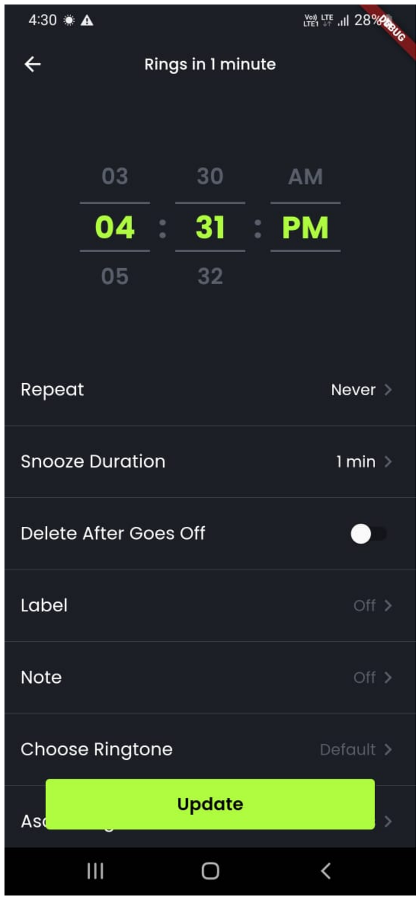
Snooze Duration 1 min

Delete After Goes Off

Label Off

Note Off

Choose Ringtone Default


4:30 28%

Rings in 1 minute

03 30 AM
04 : 31 : PM
05 32

Repeat Never

Snooze Duration 1 min

Delete After Goes Off

Label Off

Note Off

Choose Ringtone Default

- The floating action button in the Home View allows you to create alarms, directing you to the Add or Update Alarm View.
- Customize the ringtone by selecting the default option or uploading their own.
- Configure alarm repetition, set snooze duration, add labels, and define automatic cancellation conditions based on screen activity, weather, and location. Choose challenges like shake to dismiss, QR code, or math challenges. Also, manage shared alarms.

- **MILESTONE 5 (August 22 to September 10) - Feature Shared Alarms**
  - To create or join shared alarms, users **sign in with Google**. Creating one is done from Home View by tapping "Create Alarm" and selecting "shared." Joining involves entering a unique shared alarm ID or accepting an invitation from the owner.
  - The **addUserToAlarmSharedUsers** function allows users to add new collaborators to a shared alarm, verifying the provided alarm ID's existence in the database and returning false if not found.

- If the alarm ID is found, the function proceeds to add the user as a collaborator, with special handling to prevent the owner from being added, updating the shared user IDs and offset details for synchronization with all collaborators.
- **StreamAlarms**
  - The data utilized by the system is sourced from three primary repositories, namely Firestore, Shared Alarms, and IsarDB. Firestore serves as the central database housing alarms stored within its structure. Additionally, alarms shared with the user through collaborative efforts are accessed from the Shared Alarms repository. To ensure offline functionality, IsarDB stores alarms locally, providing accessibility even when internet connectivity is limited. These combined data sources contribute to the system's comprehensive alarm management capabilities, catering to both online and offline user needs
  - **Data Transformation**: The function processes data retrieved from **Firestore, Shared Alarms, and IsarDB**, converting raw document snapshots into AlarmModel objects—a data structure representing alarm records.
  - Sorting: Alarms are sorted based on user preferences, with options to enable or disable sorting. Two methods are utilized: Time-Based Sorting, where alarms are arranged in ascending order according to their scheduled time if sorting is enabled, and Priority Sorting, which organizes alarms by priority—enabled alarms precede disabled ones, and sorting considers upcoming time, repetition patterns, and immediate execution for one-time alarms if sorting is disabled
  - **Returned Stream**: After these processes, the function returns a stream that represents the list of alarms. This stream reflects the real-time status of alarms and is used to update the user interface whenever alarms are added, modified, or removed.

- **Optional MILESTONE**
  - **Automatic CI/CD for repo for releasing builds**

    If i get free time between i will also work on setting up ci/cd for the repo by **Github Actions** for automating many chore tasks like releasing builds, siging up apks, etc. There are various tools listed on flutter deployment doc like codemagic and fastlane that could be used for setting up the workflow. Appropriate tools and actions would be implemented after discussion with mentor
    -
  - **PR and issue template for bugs, features or feedback.**

    User can **report bugs, request and give feedback** by ready-made templates on Github. These templates auto-populate the issue/pull-request description field and provide a skeleton framework that contributors can fill out. They help provide a baseline standard of informational quality and organisational rigour.

My background / Technical skills

I am **Deepanshu Singh**, **3nd year computer science student at** [ITM University, Gwalior](), India. I've been enjoying web development for the first time ever since I built my **first Python project, "Automation"**. I have been deeply immersed in web development. Initially, I **began by exploring Flutter** and Python, with my early Flutter projects involving the creation of basic Android **applications in Android Studio**, such as a tic-tac-toe game, quiz app etc.

In my 10th grade, I built many small applications in Python, such as a Stock Portfolio Tracker and a Fitness Tracker. I used to post these **projects on social media** to show my friends and family.Then I became interested in **mobile application developmen**t and built many projects in Java. Additionally, I developed native applications using Flutter. After that, I learned Android development, web development, **Flutter, React, Docker, Django, Flask, and shell scripting**. I began participating in Capture **The Flag (CTF) contests** during one of my college events

Right now I am proficient in **app dev (both native and framework like react and flutter), and full-stack web dev (js, HTML, Django, react, flutter).** I use vscode, android studio and **sublime** as my text editors. Recently i tried learning server hosting, cloud and firebase.

I have gained experience **working with several startups as an intern**, where I contributed to various projects involving the development of mobile applications, web applications, and more. Recently, I have focused on a **Flutter project** where I delved into various state management techniques such as **Bloc and Riverpod** to streamline state management efficiently. Additionally, I have acquired expertise in cloud technologies like **Firebase and Firestore** during my recent projects.

I worked as **Python Automation** Intern in CHAKRABORTY LAKHAR INNOVATIONS PVT.LTD, used technologies like Web scraping using Python, **BeautifulSoup and Selenium** for automation . There I learned agile methodologies to design, **software testing** and debug software applications
Certificate of Internship - Link of certificate
Letter of recommendation - Link of letter of recommendation

I am **Microsoft Learn Student Ambassador**, helping students of my college. The MSP program enhances students' employability by offering training in skills not usually taught in academia, including knowledge of Microsoft technologies.
I am also a coordinator, responsible for managing, ideating and supervising most things tech in the college. I worked on various campus development initiatives.
My recent focus is on the **TecOS** initiative, **encouraging the open-source culture in our college**.
I am very active in open source. I try to participate in many of the open-source events happening. I participated in **KWOC, DWOC, and Hacktoberfest and this is my attempt in GSOC.** I am also actively involved in encouraging open-source culture in my college, I listed a few initiatives like org review, human of open source, etc under TeCOS.

AI Task Automation 2024

I have developed an **AI-based web application** that assists users in a wide range of tasks. Users can ask questions by providing context through various media formats such as video files, audio files, PDFs, or text documents. Additionally, the application offers numerous other features aimed at saving users' time and enhancing their productivity.

Containerize Next.js, Flask with Docker. Use Docker Compose for multi-service setup. Optimize images, manage networking. Simplify deployment and scaling

https://github.com/Deepanshuigtm/ml-app.git

Personal Projects -

https://github.com/Deepanshuigtm/internshala_job.git
https://github.com/Deepanshuigtm/flutter-chat.git
https://github.com/Deepanshuigtm/webscraping_amazon_bags.git
https://github.com/Deepanshuigtm/guess_My_Number.git

Contributions to open source:

CCExtractor Ultimate Alarm Clock

Some of my contribution to **Ultimate Alarm Clock** project are listed below :

- **Fixed a bug that was reported by someone else**. The issue can be found here: https://github.com/CCExtractor/ultimate_alarm_clock/issues/170
- **Feature Swipe-to-Delete alarm** - https://github.com/CCExtractor/ultimate_alarm_clock/issues/271
- **Undo option Swipe-to-Delete alarm -** https://github.com/CCExtractor/ultimate_alarm_clock/issues/314
- **Customizable Undo Duration -** https://github.com/CCExtractor/ultimate_alarm_clock/issues/378

Commitments:

After June, I'll begin my last year of college, the fourth year, and will dedicate full-time hours to the project. Occasionally, I'll need to attend college, but apart from those scheduled days, I'll be fully committed. In the event of any emergencies, I'll inform my mentor in advance.

Weekly Commitment: 40-50 hours

I am more than ready to work well past my committed time if needed.