



## **Google Summer of Code 2025 Proposal**



**Ccextractor**

**GSoC 2025 Proposal: Ultimate Alarm Clock  
Watch Companion**

## Personal Information

**Name:** Utkarsh Chandrakant Kadu

**Email:** [kaduutkarsh5@gmail.com](mailto:kaduutkarsh5@gmail.com)

**GitHub Profile:** [Utkarsh8867 \(Utkarsh Kadu\)](#)

**LinkedIn Profile:** [Utkarsh Kadu | LinkedIn](#)

**Location :** India

**University :** Savitribai Phule Pune University(SPPU)

**Degree Program:** Bachelor of Engineering in Computer Engineering

**Graduation Year:** 2026

# Background

I am currently pursuing a Bachelor of Engineering degree in Computer Engineering at Parvatibai Genba Moze College of Engineering, Pune. I have a strong passion for open-source development, particularly in areas of software engineering, data structures, and NLP.

I previously contributed to open-source projects, refining my skills in backend and API development. My experience with frameworks such as React.js, Next.js, and database management systems like Firebase and PostgreSQL enables me to design and develop scalable applications.

---

## Skills

- **Languages:** Dart, Java, Python, JavaScript, C++
- **Mobile Development:** Flutter (for cross-platform Android/Wear OS app development)
- **Frameworks & Libraries:** React.js, Next.js, Node.js, Django
- **Databases:** Firebase, PostgreSQL, MongoDB
- **Tools & Technologies:** Git, Docker, Cloudflare, Redis

---

## Why Am I Interested in CCExtractor?

In the world of video accessibility, CCExtractor plays a crucial role in providing accurate subtitle extraction solutions for different media formats. Unlike many automated subtitle generation tools, CCExtractor focuses on precise extraction without relying entirely on machine learning, making it an invaluable tool for low-latency environments.

Its open-source nature, community-driven development, and potential to enhance accessibility across various platforms make it an exciting project to contribute to. By improving and optimizing CCExtractor, I aim to contribute to a more inclusive and user-friendly experience for all users.

## Synopsis

This proposal focused on extending the open-source **Ultimate Alarm Clock (UAC)** project by designing and developing a **minimalistic WearOS watch companion** application. While UAC offered powerful features and a clean experience on Android phones, it lacked integration with wearable devices. The companion app aimed to bridge this gap by providing essential alarm management features on users' wrists, with a focus on usability, simplicity, and sync with the main app.

---

## Benefits to the Community

- **Extends Accessibility:** Brings the UAC experience to wearable devices, helping users manage alarms without reaching for their phones.
- **Open-Source Alternative:** Offers a free and open-source alarm companion for users who want control over their data and app behavior.
- **Enhanced Ecosystem:** Strengthens the UAC ecosystem by supporting multi-device usage.

---

## Deliverables

Period	Deliverable
Community Bonding	Understood UAC project structure, created PoC, interacted with mentors
Week 1–4	Built UI wireframes for the watch app, integrated basic alarm features
Week 5–8	Implemented alarm sync with the main UAC app using background services
Week 9–12	Added advanced features like snooze, dismiss, alarm preview with vibration
Week 13+	Testing on Wear OS emulator and real devices, bug fixing, final polish

---

## UI/UX Design

The companion app UI followed a **minimal and touch-friendly layout** with the following screens:

### 1. Home Screen:

- Next Alarm Time
- Add New Alarm button
- Sync Status indicator

### 2. Alarm List Screen:

- Scrollable list of alarms
- Toggle switch to enable/disable

- Tap to edit time

### 3. Alarm Ringing Screen:

- Large dismiss and snooze buttons
- Optional haptic feedback on ring

---

## Technical Approach

- **Tech Stack:** Flutter (for cross-platform support), Wear OS SDK, Dart, and Android Services.
- **Sync Mechanism:** Used background messaging (e.g., Firebase or Bluetooth channel) to communicate with the UAC Android app.
- **Storage:** Leveraged local SQLite for alarm persistence on the watch.
- **Alarm Handling:** Used Android AlarmManager for triggering alarms and showing watch UI.

---

## Timeline (175 hours)

Timeline	Task
Community Bonding	Explored UAC project, completed PoC app
Week 1–2	Designed UI mockups, implemented base screens
Week 3–4	Connected watch app with Android alarms (one-way sync)
Week 5–6	Implemented alarm creation/editing via watch
Week 7–8	Bi-directional sync, testing, haptic alert integration
Week 9+	Bug fixes, documentation, community feedback

---

## About Me

I am a third-year Computer Engineering student passionate about full-stack development and open-source contributions. I have experience building mobile apps using **Flutter**, creating RESTful APIs, and integrating third-party services. I previously developed a **Budget Tracker app**, a Wear OS Stopwatch, a **To-Do app in Flutter**, and I enjoy working on real-world projects that improve user productivity and convenience.

---

## Why Me

- Built multiple Flutter apps with focus on performance and design.
  - Self-motivated and familiar with exploring large codebases and community-led development.
  - Quick learner with dedication to open-source values and user privacy.
- 

## Contributions Before Proposal

- **Developed a PoC Watch App** with basic alarm triggering functionality.
  - Explored the Ultimate Alarm Clock Android codebase and replicated essential data models.
  - Shared initial UI mockups and architecture with mentors for feedback.
- 

## Mentors

- **Primary Mentor:** Rijuth Menon R (@Rijuth Menon R on Zulip)
  - **Secondary Mentor:** Akshat Tripathi (@Akshat Tripathi on Slack)
-

## References

1. **Ultimate Alarm Clock GitHub Repository**  
<https://github.com/Catrobat/Ultimate-Alarm-Clock>
2. **Catrobat Organization on GitHub**  
<https://github.com/Catrobat>
3. **Wear OS Developers Documentation**  
<https://developer.android.com/wear>
4. **Flutter Official Documentation**  
<https://docs.flutter.dev/>
5. **Google Summer of Code Official Website**  
<https://summerofcode.withgoogle.com/>
6. **AlarmManager - Android Developer Guide**  
<https://developer.android.com/reference/android/app/AlarmManager>

## Conclusion

The *Ultimate Alarm Clock Watch Companion* project aligned perfectly with my interest in mobile development and building meaningful, user-centric tools. By bringing core alarm functionalities to wearable devices, the project aimed to enhance the accessibility and convenience of the Ultimate Alarm Clock ecosystem. With my experience in Flutter, mobile UI design, and backend integration, I was well-prepared to take on this challenge.

I was excited by the opportunity to contribute to a community-driven, open-source project that values privacy, simplicity, and functionality. With the guidance of experienced mentors and a clear development roadmap, I was confident in my ability to deliver a robust, lightweight, and intuitive watch companion app that would make a lasting impact on the project and its users.