Minimum Viable Product (MVP) Draft for Cig++

---

1. Product Overview

Product Name: Cig++ - Automated Cigarette Counter
Tagline: "Every Cigarette Counts."

Cig++ is a state-of-the-art solution designed to assist smokers in monitoring their cigarette consumption through an innovative blend of hardware and software. The system includes a Bluetooth-enabled cigarette case that syncs with a mobile app, offering users real-time insights and tools to understand and manage their smoking habits. The MVP emphasizes delivering core features for data tracking, visualization, and user engagement, ensuring a streamlined and impactful experience.

---

2. MVP Goals

- 🚬 Automate cigarette consumption tracking with smart hardware integration.

- 📊 Provide clear and engaging data visualization via a user-friendly mobile application.

- 📅 Allow users to monitor their smoking habits using daily, monthly, and all-time counters.

- 🛠️ Establish a foundation for advanced features like group comparisons and detailed analytics.

---

3. Key Features in MVP

Hardware (Smart Case)

- 🧳 **Cigarette Storage:** A stylish, functional case for holding cigarettes or rolling tobacco.

- 🔗 **Bluetooth Integration:** An embedded Bluetooth module logs case openings and sends data to the app.

- 🔋 **Battery Efficiency:** Designed with low-power components to maximize battery life, minimizing recharging needs.

Mobile App

- 🔢 **Dynamic Counters:** Real-time tracking of daily, monthly, and all-time cigarette usage.

- 📈 **Data Visualization:** Clear and interactive charts to highlight usage trends and daily averages.

- 🔒 **User Authentication:** Secure account creation and login to ensure personalized data tracking.

Backend System

- 🔄 **Data Synchronization:** Ensure seamless communication between the hardware and the app for precise tracking.

- 💾 **Database:** Robust storage to securely manage user data and enable advanced analytics.

- ⚙️ **APIs:** Efficient endpoints to support seamless hardware-to-app integration.

---

4. MVP Technical Requirements

Hardware:

- 🔗 Bluetooth-enabled microcontroller to detect and record case openings.

- ⚙️ Integrated sensors for accurate logging.

- 🔋 Rechargeable battery with USB or equivalent charging capabilities.

Mobile App:

- 🔲 **Platforms:** Initial focus on Android or iOS to expedite development and testing.

- 🛠️ **Framework:** Use Flutter or React Native for cross-platform compatibility.

- 🔑 **Core Features:** Include authentication, real-time counters, and data visualization.

Backend:

- ⚙️ **Framework:** Implement using Spring Boot or Node.js for backend functionality.

- 💾 **Database:** Use PostgreSQL or MongoDB for scalable, secure data storage.

- ⚪ **Hosting:** Deploy on cloud services like AWS or Firebase for reliability and scalability.

---

5. Target Audience for MVP

- 👥 Smokers seeking tools to monitor and manage their cigarette consumption.

- 🗓 Support groups and rehabilitation centers requiring behavioral tracking tools.

- 🌍 Health-tech adopters looking for innovative ways to track habits and improve health outcomes.

---

6. Metrics for Success

- 📊 **User Engagement:** Measure daily active users (DAU) interacting with the app.

- 🔄 **Retention Rates:** Track weekly user return rates to assess long-term utility.

- ✅ **Hardware Accuracy:** Validate precise case opening detection to maintain data integrity.

- 💬 **User Feedback:** Gather user reviews to improve satisfaction and identify feature enhancements.

---

7. Development Timeline

- 📅 **Month 1:** Finalize product design and create a working hardware prototype.

- 🔢 **Month 2-3:** Develop core mobile app features and establish backend infrastructure.

- 🔄 **Month 4:** Integrate hardware with the app and conduct comprehensive testing.

- 🚀 **Month 5:** Launch MVP with beta testers and collect actionable feedback for improvements.

---

8. Future Enhancements (Post-MVP)

- 👥 **Group Comparisons:** Enable users to compare smoking habits within social or support groups.

- 🎮 **Gamification:** Add milestones, badges, and rewards to incentivize healthier behaviors.

- 📊 **Advanced Analytics:** Provide deeper insights into consumption patterns and triggers.

- 🛠️ **Rehabilitation Support:** Offer personalized reminders and tailored features for individuals in recovery programs.

- 🕐 **Wearable Integration:** Synchronize with smartwatches for holistic health tracking.

---

9. Roles and Responsibilities

Omer Shalev:

- 🛠️ **Hardware Development:** Design and assemble the smart cigarette case, integrating Bluetooth technology.

- 🧪 **Testing and Quality Assurance:** Conduct rigorous testing to ensure the hardware performs reliably under various conditions.

- 📈 **Feature Algorithms:** Develop algorithms to accurately track cigarette consumption and log data.

Tomer Geva:

- 🎇 **System Architecture:** Plan and implement the technical framework connecting hardware, app, and backend systems.

- 🔍 **Research:** Perform market and user research to align the product with consumer needs and preferences.

Shared Responsibilities:

- 🔄 **App Backend Development:** Collaborate on building and maintaining the server-side systems for data synchronization.

- 🎨 **App UX/UI Design:** Work together to design an intuitive, aesthetically pleasing user interface.

- 🔲 **App Front-End Development:** Jointly develop the client-facing features for smooth and responsive app functionality.

---

10. Project Plan

Phase 1: Research and Design (Month 1)

- 🔍 Conduct market research to validate user needs and refine the product concept.

- 🛠️ Finalize the hardware design, including the case dimensions, Bluetooth components, and battery configuration.

- 📝 Outline mobile app workflows, wireframes, and core features.

- 📋 Establish technical specifications for the backend infrastructure.

Phase 2: Prototype Development (Months 2-3)

- 🔧 Build a working prototype of the smart cigarette case, integrating Bluetooth and sensors.

- 📱 Develop the mobile app with core features: authentication, counters, and basic visualization.

- 🔄 Set up the backend system, including the database and APIs for hardware-app communication.

- 🧪 Begin initial testing of individual components (hardware, app, and backend).

Phase 3: Integration and Testing (Month 4)

- 🔗 Integrate the hardware prototype with the mobile app.

- 🛠️ Conduct comprehensive system testing to identify and resolve integration issues.

- 📡 Optimize Bluetooth communication for real-time data synchronization.

- 🧪 Perform user testing with a small group to gather feedback on usability and functionality.

Phase 4: Beta Launch and Refinement (Month 5)

- 🚀 Release the MVP to beta testers and collect feedback on performance, accuracy, and user experience.

- 🛠️ Address any identified bugs or design shortcomings.

- 🎨 Refine the app's UI/UX based on user input.

- 📈 Prepare for scaling and broader deployment, including marketing and distribution strategies.