**TEAM -37**

**Eco Habits Tracker - Software Design Document (SDD)**

**1. Introduction:**

**1.1 Purpose:**

This document presents the detailed software design for the *Eco Habits Tracker*, a cross-platform mobile application designed to encourage users to build environmentally friendly habits through tracking, motivation, and personalized insights.

**1.2 Scope:**

The app enables users to record their eco-friendly actions, monitor their daily progress, maintain habit streaks, unlock badges, and receive useful green living tips. Built with Flutter for Android and iOS compatibility, the app is powered by Firebase, leveraging services like Firestore, Authentication, Cloud Functions, and Messaging for a seamless user experience.

**1.3 Target Audience:**

This document is intended for:

* Software Engineers
* UI/UX Designers
* Product Owners
* Project Stakeholders

**2. System Overview:**

The application consists of:

* **Frontend**: Flutter-based mobile interface
* **Backend**: Firebase (Firestore for database, Auth for user management, Functions for logic)
* **Push Notifications**: Firebase Cloud Messaging (FCM)
* **Analytics**: Google Analytics for Firebase

**3. Architecture Design**

**3.1 Architectural Notes**

* User login and registration are managed via Firebase Authentication.
* All user data such as habits, progress, and rewards are stored in Firestore.
* Cloud Functions handle automation for tasks like streak tracking and notifications.
* Firebase Cloud Messaging pushes daily reminders and alerts.
* Analytics tools track user engagement and in-app behaviour.

**4. Component Breakdown:**

**4.1 Mobile Application (Flutter):**

* **UI Layer**: Eco-friendly design system using nature-inspired colors and effects (neumorphism, glassmorphism).
* **State Handling**: Using **Provider** or **GetX** for managing UI state and screen transitions.
* **Primary Screens**:
  + Intro / Onboarding
  + Login and Sign-Up
  + Home (Dashboard with streak tracker)
  + Habit Log
  + Achievements
  + User Settings

**4.2 Firebase Backend:**

* **Authentication**: Supports Email and Google-based sign-in.
* **Firestore Data Structure**:
* users/
* └── {userId}/
* ├── habits/
* ├── streaks/
* ├── achievements/
* **Cloud Functions**:
  + Calculate and update habit streaks daily
  + Trigger badge unlocks events
  + Schedule reminders for unlogged habits

**4.3 Notification System:**

* Daily notifications are sent through **Firebase Cloud Messaging** based on user preferences and missed habits.

**4.4 Analytics:**

* Tracks:
  + Number of successful habit logs
  + Habit streak maintenance
  + Time spent on each screen
  + User drop-off points during onboarding

**5. Data Flow Description:**

**5.1 Habit Completion Flow:**

1. User selects and completes a habit.
2. Input is validated on the client side.
3. Firestore updates the corresponding habit\_logs entry.
4. Cloud Function verifies if the habit streak should be extended.
5. If applicable, a badge or milestone is unlocked.
6. If a habit is missed, a reminder notification is sent.

**6. User Interface Design:**

* **Visual Theme**: Earth tones, leafy greens, subtle browns, and calming whites.
* **Design Style**: Combines modern neumorphism with translucent glassmorphism for an elegant eco-look.
* **Core UI Elements**:
  + Streak progress bars
  + Motivational animations
  + Quick log buttons
  + Achievement preview cards

**7. Security Measures:**

* Firebase Auth secures user identity and access control.
* Firestore security rules restrict data visibility to authorized users only.
* All communication is encrypted via HTTPS.
* Data integrity is ensured through both client-side and server-side validation.

**8. Non-Functional Requirements:**

* **Performance**: Optimized UI with quick loading and smooth transitions.
* **Reliability**: Firebase ensures minimal downtime with real-time updates.
* **Scalability**: Supports large user bases by using scalable cloud infrastructure.
* **Maintainability**: Modular codebase with reusable widgets and clean architecture.

**9. Technologies Used:**

| **Component** | **Technology Stack** |
| --- | --- |
| Mobile Frontend | Flutter (Dart) |
| State Management | Provider or GetX |
| Authentication | Firebase Authentication |
| Database | Firestore |
| Server Logic | Firebase Cloud Functions |
| Notifications | Firebase Cloud Messaging (FCM) |
| Analytics | Google Analytics for Firebase |
| Diagram Tools | PlantUML, Draw.io |

**10. Planned Enhancements:**

* **Leaderboard & Social Features**: Compete and share eco-progress with friends
* **AI-Powered Suggestions**: Personalized eco-habit recommendations
* **Dark Mode**: User preference toggle for night-friendly UI
* **Quick Widgets**: Home screen widgets for fast logging