Team Working Agreement

Project: sano
Class: CSE 115A - Winter 2025
Version: 1.0

---

1. Logistics

Work Room & Location

- **Primary Work Location:** A table on the third floor of SnE.
- **Secondary Location:** Discord Voice Chat.

Meeting Times

- **Tri-Weekly Standup:**
  - **Monday:** End of 12 PM TA Meeting + 15 mins.
  - **Wednesday:** 4:00 PM - 4:15 PM.
  - **Friday:** 12:00 PM - 12:15 PM.
- **Sprint Planning/Review:** Initiated during Monday Standup, completed Tuesday.

Project Repository

- **Location:** `https://github.com/kdelmo1/sano.git`
- **Organization:**
  - The `main` branch is protected (no direct pushes).
  - Feature branches must follow the naming convention: `feature/short-description` (e.g., `feature/chat`).

Communication

- **Urgent:** SMS Text.
- **Project-related:** Dedicated Discord server.

---

2. Development Environment

Platform

- **Primary Development:** Mobile (iOS/Android) via **Expo Go**.
- **Simulators:** Xcode (iOS) or Android Studio (Android).

IDE & Tools

- **Editor:** VS Code.
- **Framework:** React Native (Expo) + TypeScript.
- **Backend:** Supabase (Auth, Storage, Realtime).
- **Linter:** ESLint with Google Config.

---

3. Coding Style & Standards

Language

- **TypeScript:** Must be used for all new components (interfaces required for props).

Linting

- Code must pass `npm run lint` before pushing.
- Use `npm run lint:fix` to auto-correct formatting issues.

Styling

- Use `SharedStyles` and `Colors` from `src/styles/sharedStyles.ts` to maintain the consistent yellow/white/black theme/color scheme.
- Use `ResponsiveUtils.moderateScale` for dimensions to ensure compatibility across different screen sizes.

Naming Conventions

- **Files:** PascalCase for Components (e.g., `ProfileScreen.tsx`), camelCase for utilities (e.g., `getFromDB.tsx`).
- **Assets:** kebab-case for images (e.g., `icon-thumbs-up.png`).

---

4. Work (Process) Patterns

Definition of Done (DoD)

- ☐ Feature is fully implemented in TypeScript.
- ☐ Linter throws no errors (`npm run lint`).
- ☐ Tested on at least one physical device or simulator.
- ☐ Pull Request (PR) created and reviewed by at least one teammate.
- ☐ Branch handling feature is pushed to `main`.

Collaboration

- **Code Review:** Reviewers check for adherence to `SharedStyles` and proper Supabase error handling.
- **Hand-off:** If backend schema changes (e.g., Supabase table updates), notify the team immediately in Discord or text, depending on urgency.

Areas of Responsibility (SMEs)

- **Frontend/Animations:** Kyle, Elton, Nora.
- **Backend/Data:** Gianmarco, Andy, Alonso.
- **Scrum Masters (in-order):** Alonso, Elton, Gianmarco, Nora.

---

5. Product Design Patterns

UI/UX Look and Feel

- **Navigation:** Use the custom `NavBar` with floating animations for main screens.
- **Interactions:** Use "Optimistic Updates" for user actions like ratings (update UI immediately, then sync DB) to ensure the app feels fast.

Error Handling

- **User Facing:** Use `Alert.alert()` for critical blocks (e.g., "Reservation required").
- **System:** Log detailed errors to console but show graceful fallbacks (e.g., "No posts to display") to the user.

Common Approaches

- **Data Fetching:** Centralize logic in `GetFromDB.tsx` where possible to keep screens clean.
- **Realtime:** Use Supabase channels (`supabase.channel`) for features requiring live updates like Chat and Notifications.

---

6. Conflict Resolution

- **Disagreement Handling:** Majority vote + optional consultation with TA if difficult to resolve.

Signatures:

_ Kyle Delmo, Date: 10/19/25

_ Gianmarco Sassoli, Date: 10/19/25

_ Andy Lau, Date: 10/19/25

_ Alonso Isauro, Date: 10/19/25

_ Elton Su, Date: 10/19/25

_ Nora Fotoohi, Date: 10/19/25