**The Official Rulebook (Elaborated)**

This document outlines our contract for building the Pulse application. It defines **what** we are building and **how** we will build it, ensuring a high-quality, consistent, and predictable development process.

**Part 1: The Vibe and Specification (The "What")**

This section defines the end product. It ensures that what we build is a direct and faithful translation of the vision you've laid out.

• **Visual & UI Design**

• **Meaning:** We will treat your descriptive documents as a strict design system. Every visual detail mentioned is a requirement.

• **In Practice:** This means if the design calls for a Background (Dark) of #121212, that is the only background color we will use. If a button needs 8px rounded corners and an Accent - Calm (Teal) background, every primary button will be built to those exact specifications. Typography like "Inter Bold, 32px" will be implemented as a reusable style for all H1 headers.

• **Functional Flow**

• **Meaning:** The application's navigation and interactivity must behave exactly as described. We are building a precise user journey, not just a collection of screens.

• **In Practice:** When the user taps "Become a Verified Neighbor" on the EntryScreen, the code must initiate a navigation event that leads directly to the VerificationScreen. We will implement every described tap, swipe, and user action to trigger the specified outcome.

• **State Variations & Error Handling**

• **Meaning:** Our components and screens will be dynamic. They must respond to different data and user inputs by changing their appearance and behavior as specified.

• **In Practice:** The Home Screen isn't a single static design; it has "Calm," "Elevated," and "Urgent" variations. We will write the code to conditionally apply styles—like changing the accent color to Accent - Urgent (Crimson) and expanding an alert module—based on the state of the app's data. Error messages like "Please enter a valid phone number" will be implemented to appear only when the input is invalid.

• **UX & Atmosphere**

• **Meaning:** The feel of the app is a core requirement. It's not just about function; it's about the emotional response it evokes.

• **In Practice:** Words like "fluid," "calm," and "reassuring" will translate into technical decisions. We will implement smooth screen transitions, subtle micro-interactions (like a button pulsing when active), and maintain generous spacing and an uncluttered layout to create a calm user experience.

• **Accessibility (a11y)**

• **Meaning:** We will build the app for everyone, including users with disabilities, right from the start. Accessibility is a foundational requirement, not an afterthought.

• **In Practice:** All interactive elements without text (like icon buttons) will have descriptive aria-labels. We will use semantic HTML/native elements to ensure proper screen reader navigation (e.g., VoiceOver, TalkBack). We will verify that text color contrasts are sufficient against the dark background to be readable.

**Part 2: The Development Process (The "How")**

This section defines our workflow. It ensures the underlying code is as clean, robust, and well-organized as the final UI.

• **Atomic Code**

• **Meaning:** We will build the application out of small, reusable, single-purpose components. This makes the code easier to manage, test, and debug.

• **In Practice:** Instead of building a single giant file for the HomeScreen, we will break it down. We'll create separate components like <PulseCard>, <Header>, and <BentoGridItem>. Each component will do one thing and do it well.

• **Test Coverage**

• **Meaning:** We will write automated tests for our code to ensure it works as expected and to prevent future changes from breaking existing functionality.

• **In Practice:** For each component, we will write tests to verify it renders correctly given different inputs (props). For any non-UI logic (e.g., a function that validates a phone number), we will write unit tests to check its behavior against a range of possible inputs.

• **One Code at a Time**

• **Meaning:** Our workflow will be sequential and focused. We will complete one task fully (code, tests, styling) before moving to the next.

• **In Practice:** We will decide to build the <Button> component. I will write the code, the styles, and the tests for only that button. I will present it to you. Only after you approve it will we move on to the next component, like <PhoneInput>.

• **Confirm Before Coding**

• **Meaning:** This is the first step of any task. I will never assume what to do next. I will always get your explicit "go-ahead" before writing a single line of code.

• **In Practice:** Our interaction will look like this: Me: "I am now ready to build the OnboardingWalkthroughScreen component, which will be a swipeable carousel. Shall I proceed?" You: "Yes." Me: [writes code].

• **No Workarounds**

• **Meaning:** We will prioritize code quality over speed. The code must be clean, readable, and maintainable for the long term.

• **In Practice:** This means no "magic strings" or numbers; we will use constants. We will use clear variable names. We will not comment out old code; we will delete it. We will avoid shortcuts that compromise the structure or clarity of the code.

• **Consistency**

• **Meaning:** All the rules in this document apply to every single file and line of code we write for the entire project.

• **In Practice:** The way we structure our first component will be the same way we structure our last. The level of detail in our tests will not decline over time. This ensures the entire codebase feels unified and professional.