Software Requirements Specification

commit

PDG - HEIG-VD

Leonard Cseres   Tristan Gerber   Aladin Iseni   David Schildböck

August 18, 2025

Contents

1   Introduction

- **Purpose**:
  - Primary users: Private adult individuals who want to take action in their lives and seek stronger follow-through on commitments.
  - Core problem: Providing extra motivation to get things done by introducing a financial stake and social accountability.
  - Top objectives:
    1. Increase goal adherence by letting users stake money against personal
    2. Enable social accountability via groups and shared challenges with pooled stakes.
    3. Provide automated financial consequences when goals are missed.
  - Success metrics:
    * App Store/Play Store rating above 4.0 stars.
    * Weekly active users (WAU) growth.
    * Total amount of money staked/processed in the app.
- **Scope**:
- Platforms: iOS and Android via Expo; web landing page (marketing/information only).
- Geography at launch: Switzerland.
- User types: Standard users only (no admin/moderator roles for MVP).
- MVP goal types: Wake-up time, location-based, time-based, duration-based, and combinations of location/time/duration.
- Money flow: Real money via Stripe (payments and transfers).
- Group challenges: Invite-only, private groups.
- Out of scope (MVP): App Store/Play Store deployment, tablet support, advanced analytics, notifications, appeals/dispute handling.
- Languages: English only at launch.
- Accessibility & compliance: Out of scope for MVP.
- **Definitions & Acronyms**:

2   Project Description

- **Objective**:
  - Product vision: Commit turns intentions into actions by combining simple goal tracking with real financial stakes and social accountability.
  - Primary use cases:
    1. Daily wake-up challenge at a set time
    2. Location-based workout (arrive and stay at a gym/park)
    3. Focused study session for a specified duration
  - Differentiators: Ease of use, pooled stakes for group challenges.
  - Constraints/guiding principles: Instant money transfers between parties with minimal fees.
- **Non-Functional Requirements**:

3   System Overview

- High-level description of the app
    - Solo flow: A user creates a goal, configures recurrence or due date, selects a verification method (GPS/time/photo), sets a stake amount and a destination for funds. If the goal is completed and verified within the rules, no transfer occurs; if not, the staked amount is transferred to the configured destination.
    - Group flow: A user creates a private, invite-only group challenge with a defined goal and stake amount (and a fallback destination if all participants fail). Invitees accept the stake. Upon completion, successful participants receive the pooled stakes from members who failed; if all fail, funds are sent to the fallback destination.
- Target platforms (iOS, Android, etc.)
- Mobile: iOS and Android (phones only; no tablet support)
- Permissions/capabilities: Background location (GPS), camera, push notifications, device usage detection (for no-phone-use goals)
- Expo: SDK and EAS details TBD
- Web: Single static marketing landing page (Astro + React)
- Target audience
    - Adults 18+ seeking productivity, fitness improvements, and habit-building.

4   Functional Requirements

4.1   User Management

- Users can register with email/social login
- Users can manage profiles

4.1.1   Authentication

- Sign-in methods: Google, Apple (no email/password at MVP)
- Pre-stake verification: No email/phone verification required before staking
- KYC/identity checks: Not required at MVP for payouts via Stripe

4.1.2   Payments & Account Receivers (MVP decisions)

- Charging model: Stakes are authorized/held at creation but only captured and deducted if the goal is not achieved (on failure). If the goal is achieved, no funds are captured.
- Stake range: CHF 1 (min) to CHF 1000 (max) per goal.
- Currency: CHF only at launch.
- Recipients:
    - Solo challenge: at creation the user selects a recipient among: (a) a named person who is an existing app user, (b) a charity from a small predefined list, or (c) the developers (platform donation account).
    - Group challenge: on resolution, winners split the stakes evenly among all winners. If no participants succeed, the pooled funds go to the destination configured by the

creator for the group goal.

- Platform fees: No operational commission will be taken on stake transfers for the MVP; transfers to the developers' account are treated as donations. Stripe processing fee payer is TBD.
- Payout timing: instant payouts to winners are preferred; this requires connected payout accounts for recipients or platform-managed routing via Stripe. The team will use Stripe Connect patterns appropriate to this choice in implementation.

4.2   Core Features

- Goal lifecycle: create, edit, delete, view history
- Group lifecycle: create private group, invite by link, join/leave, view results
- Verification capture: GPS check, time check, in-app photo capture within window
- Money: create stake authorization, capture on failure, distribute to winners or fallback destination
- Activity/history: per-user list of past goals/challenges with outcomes
- Settings: change display name and profile photo

4.3   Goal Creation

- Required fields: title/name, description, goal type (wake-up/location/time/duration/combined), start date, and due date or schedule/recurrence.
- Recurrence: select days of the week with an end date.
- Verification window: allowed; default ± 10 minutes around the scheduled time (configurable per goal).
- Location goals: geofence with default and maximum radius (meters) and a must-stay duration — default radius 50 meters (maximum 500 meters) and minimum dwell time 5 minutes.
- Duration/focus goals: strictly continuous session; minimum and maximum duration values: default minimum 15 minutes, maximum 240 minutes (4 hours).
- Photo verification: photo must be captured within the verification window; front or back camera allowed; selfie not required.
- Failure definition: missing verification or verification outside the allowed window results in automatic failure.
- Grace/retries: none for MVP.

4.4   Verification & Goal Rules

- Verification methods supported in MVP: GPS (location), device time checks, and photo evidence.
- Photo verification: captured within the verification window; front/back camera allowed; initially verified manually by the project team before final settlement of funds. AI-assisted verification is planned for a future iteration.
- Combined verification: goals may require multiple verification methods (for example, a

hike may require both GPS route/arrival and a photo of the summit).

- GPS rules: behavior depends on goal type. Default geofence radius 50 meters (maximum 500 meters) and minimum dwell time 5 minutes; parameters are configurable per goal.
- No-phone-use goals: included in MVP but the precise detection mechanism is TBD. Options include in-app foreground session monitoring, OS usage APIs, or photographic proof workflows.
- Time windows: default ±10 minutes around scheduled time; configurable per goal.
- Failure handling: automatic failure if verification is missing or outside the allowed window.
- Grace/retries: no additional grace period or retries for MVP.
- Offline/technical failures: there is no automatic fallback; users may file an appeal if verification cannot be performed due to technical reasons.

4.5   Group Challenges

- Size limit: up to 100 participants per group challenge.
- Stake uniformity: same stake amount for all participants.
- Join flow: invite via link with expiration; joiners must register/sign in and have a valid payment method on file.
- Invite expiration: default 7 days.
- Schedule: group goals follow the creator's schedule. The creator may set a time interval window to allow flexibility for participants to perform within their availability.
- Distribution: on resolution, winners split the pooled stakes evenly. If no participants succeed, funds go to the destination selected by the creator for this group goal.
- Failure to verify: not providing required verification within the window is an automatic failure.
- Cancellation: if the creator cancels before the start, no stakes are captured (since capture happens only on failure at resolution).

4.6   Notifications

Out of scope for MVP.

4.7   Settings

- Update display name
- Update profile photo

5   Non-Functional Requirements

5.1   Performance

- App should load within 2 seconds
- Support at least 10,000 concurrent users
- Backend latency targets (p95 within CH/EU): $<= 500$ ms

5.2 Security

- Data encryption in transit and at rest
- Secure authentication (OAuth2, JWT, etc.)
- Stored data: user profile, account, groups, goals, integrations. Avoid storing unnecessary sensitive data.
- Photo storage: stored as objects in Supabase Storage (S3-compatible). Retention policy: Out of scope for MVP.
- Location data: not stored server-side; processed on-device for verification where possible.
- Access control: goals are private by default; group members only see minimal status (success/failure) and not each other's raw verification artifacts.

5.3 Privacy

- Photos are stored for verification purposes only; access is restricted to the account owner and authorized reviewers.
- Location traces are not persisted server-side; only ephemeral checks are performed for verification.
- Compliance posture: out of scope for MVP.

5.4 Reliability & Operations notes

- Manual verification SLA: initial target is to perform manual photo verifications within 24–48 hours of submission. The team will adjust this SLA based on capacity.

5.5 Usability

- Intuitive navigation
- Consistent UI across platforms

5.6 Reliability & Availability

- 99.9% uptime
- Graceful error handling

5.7 Compatibility

- Support iOS 14+ and Android 10+
- Responsive design for different screen sizes

5.8 Maintainability

5.9 Battery & Location Usage

- Use region/geofence monitoring only; avoid continuous GPS tracking

5.10 Offline Behavior

- Online-only MVP: goal creation and verification require connectivity
- Modular codebase following Clean Architecture
- Comprehensive documentation and unit tests

6 Preliminary Architecture Description

- Presentation layer (UI/UX)
- Application layer (state management, controllers)
- Domain layer (business logic, use cases)
- Data layer (APIs, local DB, repositories)
- Infrastructure (networking, analytics, logging)

7 Mockups / Landing Page

- Figma designs
- Paper sketches
- Landing page prototype

8 Technical Choices

- Programming languages & frameworks: Expo + React Native (TypeScript)
- Database: Postgres (Supabase)
- Backend/services: Supabase (Auth, DB, storage, edge functions)
- Payments: Stripe Connect Standard with TWINT enabled for Switzerland
- Third-party libraries & APIs: Stripe SDK, Expo Location/Camera/Notifications
- Hosting: Supabase (backend, DB, auth); Cloudflare Pages (Astro + React landing page)

9 Work Process

- Version control: GitHub with pull requests; trunk-based development
- DevOps: continuous delivery to production when changes pass CI and review
- Process: lightweight Kanban for MVP

10 Development Tools Setup

- Issue tracker: GitHub Issues
- Code review: GitHub pull requests
- Documentation: repository README and SRS in docs/
- Code style: Prettier + ESLint with TypeScript rules
- State/data libraries: TBD (e.g., Zustand, React Query)

- Testing approach: TBD (at minimum unit tests for core logic; consider 1 e2e flow with Detox)
- Secrets/config: Supabase and Stripe keys via env files with secure storage

11   Deployment Environment

- Target infrastructure (e.g., AWS, Azure, on-premise)
- Environment setup: development and production only

12   CI/CD Pipeline

- CI on PRs: lint, typecheck, build, and tests
- CD: auto-deploy to development on main merges; promote to production on tagged releases
- Monitoring & rollback: basic health checks; manual rollback by reverting deploy

13   Constraints & Assumptions

- Budget & time constraints: 70K CHF budget; 3-week MVP timeline
- Regulatory compliance: out of scope for MVP

14   Glossary

- Stake: the amount of money a user commits that may be captured if the goal is not achieved.
- Capture: charging the authorized stake when a goal is marked as failed.
- Goal window: the scheduled time period during which the user must complete and verify the goal.
- Verification window: the allowed buffer around the scheduled time for submitting verification (default ±10 minutes).
- Geofence: a virtual radius around a location used to verify presence (default 50 m; max 500 m).
- Dwell time: the minimum time a user must remain inside a geofence (default 5 minutes).
- Destination: the recipient configured to receive funds when a goal fails (person, charity, or platform donation).
- Winner pool: the set of participants in a group challenge who achieved the goal and split the captured stakes.
- Group challenge: a private, invite-only challenge with a uniform stake and shared rules created by a user.

15   Appendices

- Mockups/Wireframes

- API reference (if applicable)

16   Points to Refine Later

- Minimum OS versions for iOS and Android
- Expo SDK version and EAS build configuration
- Implementation/permission model for device usage detection (no-phone-use goals)
- Details on ensuring instant transfers with minimal fees via Stripe
- Stripe fee responsibility (payer model)
- Initial charity list for solo goal destinations