

# SOFTWARE REQUIREMENTS SPECIFICATION

## FamNest — AI Wellness Coach for Busy Parents

*Conformant to IEEE Std 29148-2018*

Prepared by:

**Virginia Mwega**

*Full-Stack Developer & Technical Author*

[virginiamwega2@gmail.com](mailto:virginiamwega2@gmail.com) | [virginiamwega-com.vercel.app](https://virginiamwega-com.vercel.app)

<b>Document Reference</b>	FN-SRS-2026-001
<b>Version</b>	1.0
<b>Status</b>	BASELINE
<b>Classification</b>	CONFIDENTIAL
<b>Date</b>	May 2026
<b>Standard</b>	IEEE Std 29148-2018
<b>System</b>	FamNest v1.0 (Production)

# 1. Introduction

## 1.1 Purpose

This Software Requirements Specification (SRS) defines the functional and non-functional requirements of FamNest, an AI-powered wellness coaching application for busy parents. It establishes the agreed behaviour of the system as a basis for development, verification, and stakeholder review. The document is conformant to IEEE Std 29148-2018.

## 1.2 Scope

FamNest delivers personalised, time-bounded wellness recommendations generated from a brief daily self-assessment. This SRS covers authentication, the daily check-in, AI recommendation generation, the trend dashboard, subscription management, transactional email, crisis signposting, and data export, as implemented in version 1.0.

## 1.3 Definitions, Acronyms & Abbreviations

Term	Definition
Check-in	A user-submitted daily self-assessment (mood, energy, stress, available time)
Recommendation	An AI-generated wellness plan produced from a single check-in
RLS	Row-Level Security — PostgreSQL access control enforced at the database layer
LLM	Large Language Model — the AI inference provider
P0 / P1 / P2	Requirement priority: Mandatory / High / Desirable
SHALL	Denotes a mandatory requirement per IEEE convention

## 1.4 References

- IEEE Std 29148-2018 — Systems and software engineering — Life cycle processes — Requirements engineering
- FN-SAD-2026-001 — FamNest System Architecture Document
- FN-TFR-2026-001 — FamNest Technical Findings & Remediation Report
- HIPAA Technical Safeguards — 45 CFR §164.312 (US market reference)

## 2. Overall Description

### 2.1 Product Perspective

FamNest is a self-contained web application built on Next.js with a PostgreSQL data store (Supabase) and an external LLM inference layer. It operates as a standalone consumer product and does not, in version 1.0, integrate with external clinical or healthcare systems.

### 2.2 Product Functions (Summary)

- Account creation, authentication, and credential recovery
- Daily check-in capture and AI recommendation generation
- Editing and deletion of the user's own check-ins
- Trend visualisation across historical check-ins
- Premium subscription via Stripe
- Transactional and scheduled email (reminders, weekly report)
- Crisis resource signposting
- Export of the user's own data (CSV)

### 2.3 User Classes & Characteristics

User Class	Characteristics	Technical Skill
Free User	Parent using the core daily check-in and recommendation	Low — consumer
Premium User	Subscribed user with extended history and features	Low — consumer
System (Automated)	Scheduled jobs: reminders, weekly reports	N/A

### 2.4 Operating Environment

- Client: modern evergreen web browsers (desktop and mobile), responsive layout
- Server: Vercel serverless runtime (Node.js 20 LTS)
- Data: PostgreSQL 15 (Supabase) with Row-Level Security enabled
- External services: LLM inference provider, Stripe, Resend email

### 2.5 Constraints

- All user data access SHALL be governed by Row-Level Security at the database layer
- No Protected Health Information (PHI) is collected; wellness scores are self-reported preferences
- Payments are processed exclusively through Stripe (PCI-DSS compliant); USD only in v1

### 2.6 Assumptions & Dependencies

- The LLM inference provider is available; the system SHALL degrade gracefully when it is not
- Supabase RLS remains enabled in all environments (see SRS §4 NFR-SEC-02 and FN-TFR-2026-001 OBS-01)
- Users have a valid email address for authentication and transactional mail



## 3. Functional Requirements

### 3.1 Authentication & Account

FR-001 — Account Registration [P0 — Mandatory]	
Actor	Prospective User
Description	The system SHALL allow a user to create an account using an email address and password.
Precondition	User is not authenticated.
Input	email: string, password: string
Output	Account created; verification/session established via Supabase Auth.
Validation	Email SHALL be well-formed and unique; password SHALL meet minimum strength policy.
Error handling	Duplicate email SHALL return a clear error; weak password SHALL be rejected with guidance.

FR-002 — Authentication (Login) [P0 — Mandatory]	
Actor	Registered User
Description	The system SHALL authenticate a registered user via email and password and establish a session.
Precondition	User has a registered account.
Input	email: string, password: string
Output	Authenticated session (JWT) issued; user redirected to dashboard.
Validation	Credentials SHALL be validated against the identity provider.
Error handling	Invalid credentials SHALL return a non-specific error (no account enumeration).

FR-003 — Password Recovery [P0 — Mandatory]	
Actor	Registered User
Description	The system SHALL allow a user to request a password reset link and SHALL provide a functioning route to set a new password. [Ref: FN-TFR-2026-001 TFR-02]
Precondition	User has a registered account and access to their email.
Input	email: string; subsequently new password via reset route.
Output	Reset link issued; on completion, the user's password is updated and they can authenticate.
Validation	Reset link SHALL be single-use and time-limited; new password SHALL meet strength policy.
Error handling	Expired or used links SHALL return a clear message and allow re-request; the reset route SHALL exist and resolve (no dead-end).

FR-004 — Logout [P1 — High]	
Actor	Authenticated User
Description	The system SHALL allow an authenticated user to terminate their session.
Precondition	User is authenticated.

<b>Input</b>	Logout action.
<b>Output</b>	Session and refresh token invalidated; user returned to public state.
<b>Validation</b>	N/A
<b>Error handling</b>	N/A

### 3.2 Daily Check-In & Recommendation

FR-005 — Submit Check-In [P0 — Mandatory]	
<b>Actor</b>	Authenticated User
<b>Description</b>	The system SHALL allow an authenticated user to submit a daily check-in comprising mood, energy, stress, and available time.
<b>Precondition</b>	User is authenticated.
<b>Input</b>	mood: int(1–10), energy: int(1–10), stress: int(1–10), time_available: int(minutes)
<b>Output</b>	A check-in record is persisted to daily_checkins (scoped to the user); recommendation generation is triggered.
<b>Validation</b>	All scores SHALL be within 1–10; time_available SHALL be a positive integer within the supported range.
<b>Error handling</b>	Out-of-range or missing values SHALL be rejected with HTTP 400 and a descriptive message.

FR-006 — Generate AI Recommendation [P0 — Mandatory]	
<b>Actor</b>	System
<b>Description</b>	On successful check-in, the system SHALL generate exactly one wellness recommendation using the LLM inference layer, bounded by the user's stated available time.
<b>Precondition</b>	A valid check-in has been persisted.
<b>Input</b>	Anonymised check-in scores and available time (no PII).
<b>Output</b>	Exactly one recommendation row persisted, linked to the originating check-in; rendered to the user.
<b>Validation</b>	Output SHALL conform to the defined recommendation schema; recommendation SHALL NOT exceed stated time budget.
<b>Error handling</b>	On LLM failure, the system SHALL serve a graceful fallback and inform the user of temporary degradation.

FR-007 — View Recommendation [P0 — Mandatory]	
<b>Actor</b>	Authenticated User
<b>Description</b>	The system SHALL display the recommendation associated with a given check-in to its owner.
<b>Precondition</b>	A recommendation exists for the check-in and belongs to the requesting user.
<b>Input</b>	check-in / recommendation identifier
<b>Output</b>	Recommendation rendered. Retrieval SHALL return at most one row per check-in.
<b>Validation</b>	Exactly one recommendation SHALL exist per check-in (enforced by unique constraint).
<b>Error handling</b>	Absence of a recommendation SHALL show a clear state; multiple rows SHALL be prevented at the data layer.

FR-008 — Edit Check-In [P1 — High]	
Actor	Authenticated User
Description	The system SHALL allow a user to edit their own check-in, and edits SHALL persist. [Ref: FN-TFR-2026-001 TFR-01]
Precondition	Check-in exists and belongs to the user.
Input	Updated mood/energy/stress/time_available
Output	Check-in updated in place; associated recommendation regenerated without creating duplicate rows.
Validation	Same range rules as FR-005; UPDATE SHALL be permitted by RLS policy for the owner.
Error handling	Failed persistence SHALL surface an error rather than silently succeeding.

FR-009 — Delete Check-In [P1 — High]	
Actor	Authenticated User
Description	The system SHALL allow a user to delete their own check-in, and deletion SHALL be effective. [Ref: FN-TFR-2026-001 TFR-01]
Precondition	Check-in exists and belongs to the user.
Input	check-in identifier
Output	Check-in and its dependent recommendation removed; confirmation reflects the true result.
Validation	DELETE SHALL be permitted by RLS policy for the owner.
Error handling	If deletion affects zero rows, the system SHALL NOT report success.

### 3.3 Dashboard & Data

FR-010 — Trend Dashboard [P1 — High]	
Actor	Authenticated User
Description	The system SHALL display the user's mood, energy, and stress trends across their historical check-ins.
Precondition	User is authenticated; at least one check-in exists.
Input	N/A (reads user's own records)
Output	Aggregated trend visualisation scoped to the authenticated user.
Validation	Results SHALL be scoped to the owner via RLS and an explicit application-level filter (defense in depth).
Error handling	Empty history SHALL show a welcoming empty state, not an error.

FR-011 — Export Data (CSV) [P2 — Desirable]	
Actor	Authenticated User
Description	The system SHALL allow a user to export their own check-in history as CSV.
Precondition	User is authenticated.
Input	Export request
Output	CSV file containing only the requesting user's records.
Validation	Export SHALL be scoped to the owner via RLS and an explicit user_id filter. [Ref: FN-TFR-2026-001 OBS-01]
Error handling	Export of an empty history SHALL produce a valid file with headers only.

### 3.4 Subscription

FR-012 — Premium Subscription Purchase [P1 — High]	
Actor	Authenticated User
Description	The system SHALL allow a user to purchase a premium subscription via Stripe.
Precondition	User is authenticated.
Input	Stripe checkout session
Output	On checkout.session.completed, the user's plan_tier is set to premium.
Validation	Payment SHALL be processed by Stripe; webhook signature SHALL be verified before processing.
Error handling	Failed payment SHALL leave plan_tier unchanged and inform the user.

FR-013 — Subscription Lifecycle Sync [P1 — High]	
Actor	System
Description	The system SHALL keep the user's plan status synchronised with Stripe subscription events.
Precondition	An active or changing subscription exists.
Input	Stripe webhook events (updated, deleted, payment_failed)
Output	plan_tier reflects current subscription state; downgrade preserves historical data.
Validation	Only signature-verified webhook events SHALL be processed.
Error handling	payment_failed SHALL trigger a grace-period email sequence.

### 3.5 Email & Notifications

FR-014 — Daily Reminder [P1 — High]	
Actor	System
Description	The system SHALL send a daily check-in reminder to eligible users. Where a preferred reminder time is offered, delivery SHALL honour it. [Ref: FN-TFR-2026-001 TFR-03]
Precondition	User is eligible and has not opted out.
Input	User eligibility set; preferred_time where applicable
Output	Reminder email dispatched within the promised window of the user's preferred_time.
Validation	If per-user timing cannot be honoured, the product SHALL NOT promise per-user timing in copy.
Error handling	Email send failure SHALL be logged and retried per provider policy.

FR-015 — Weekly Report [P2 — Desirable]	
Actor	System
Description	The system SHALL send an eligible user a weekly summary of their wellness trends.
Precondition	User is eligible; sufficient history exists.
Input	User's weekly check-in data
Output	Weekly summary email dispatched.
Validation	Content SHALL reflect only the recipient's own data.

<b>Error handling</b>	Send failure SHALL be logged and retried per provider policy.
-----------------------	---------------------------------------------------------------

### 3.6 Safety & Measurement

<b>FR-016 — Crisis Signposting [P0 — Mandatory / Safety]</b>	
<b>Actor</b>	Authenticated User
<b>Description</b>	The system SHALL provide persistent, visible access to mental-health crisis resources at all application states.
<b>Precondition</b>	User is within the application.
<b>Input</b>	N/A
<b>Output</b>	Crisis resources accessible; prominently surfaced where low mood is indicated.
<b>Validation</b>	Resources SHALL be present regardless of check-in state or score.
<b>Error handling</b>	N/A — must never be suppressed.

<b>FR-017 — Conversion Analytics [P1 — High]</b>	
<b>Actor</b>	System
<b>Description</b>	The system SHALL record a checkin_submitted analytics event on each successful check-in submission. [Ref: FN-TFR-2026-001 TFR-04]
<b>Precondition</b>	A check-in has been successfully persisted.
<b>Input</b>	Successful submission signal
<b>Output</b>	A checkin_submitted event recorded in the analytics destination.
<b>Validation</b>	Event SHALL fire only after confirmed successful persistence.
<b>Error handling</b>	Analytics failure SHALL NOT block or degrade the user-facing flow.

<b>FR-018 — Data Deletion (Right to Erasure) [P1 — High]</b>	
<b>Actor</b>	Authenticated User
<b>Description</b>	The system SHALL allow a user to delete their account and associated personal data.
<b>Precondition</b>	User is authenticated.
<b>Input</b>	Account deletion request (confirmed)
<b>Output</b>	User account and owned records removed or irreversibly anonymised.
<b>Validation</b>	Deletion SHALL cascade to all tables owning the user's data.
<b>Error handling</b>	Partial failure SHALL be logged and surfaced; no silent partial deletion.

## 4. Non-Functional Requirements

ID	Category	Requirement	Target
NFR-PERF-01	Performance	Page load (Largest Contentful Paint)	< 2.0 s (P95)
NFR-PERF-02	Performance	Recommendation generation time	< 5.0 s (P95)
NFR-AVAIL-01	Availability	Monthly platform uptime	> 99.5%
NFR-SEC-01	Security	Transport encryption	TLS 1.3 minimum
NFR-SEC-02	Security	Data access governed by RLS + explicit code filter	Enforced on all user data paths
NFR-SEC-03	Security	Auth token lifetime	≤ 60 minutes; refresh rotation
NFR-PRIV-01	Privacy	PII transmitted to LLM	Zero
NFR-USE-01	Usability	Core check-in completion time	< 90 seconds
NFR-REL-01	Reliability	LLM failure handling	Graceful fallback; no hard failure
NFR-COMP-01	Compliance	HIPAA technical safeguards (US)	Aligned (not certified)
NFR-COMP-02	Compliance	Single recommendation per check-in	Enforced by unique constraint

## 5. External Interface Requirements

Interface	Purpose	Data Exchanged	Failure Behaviour
LLM Inference API	Recommendation generation	Anonymised scores out; structured plan in	Graceful fallback plan served
Stripe	Payments & subscription	Email, customer/subscription IDs	User informed; tier unchanged
Resend	Transactional & scheduled email	Email address, summaries	Logged and retried
Supabase Auth	Identity & sessions	Email, hashed password	Auth error surfaced to user

## 6. Data Requirements

### 6.1 Core Entities

Entity	Key Fields	Ownership / RLS
users	id (PK), email, plan_tier, preferred_time, stripe_id	Self; identified by auth.uid()
daily_checkins	id (PK), user_id (FK), mood, energy, stress, time_available, created_at	SELECT/INSERT/UPDATE/DELETE own (RLS)
recommendations	id (PK), checkin_id (FK, UNIQUE), user_id (FK), content, created_at	SELECT/INSERT/UPDATE/DELETE own (RLS)
subscriptions	id (PK), user_id (FK), status, stripe_subscription_id	Self; synced from Stripe

*Data integrity note: recommendations.checkin\_id SHALL carry a UNIQUE constraint to guarantee a single recommendation per check-in, satisfying FR-007 and NFR-COMP-02 and preventing the multiple-row condition identified in FN-TFR-2026-001 (TFR-01).*

### 6.2 Validation Rules

- mood, energy, stress: integer, 1–10 inclusive (CHECK constraint)
- time\_available: positive integer within supported range
- email: well-formed and unique
- All write operations SHALL be authorised by the owning user via RLS

## 7. Compliance & Regulatory Alignment

Requirement	Standard	Implementation in FamNest	Status
Encryption in transit	HIPAA §164.312(e)	TLS 1.3 enforced via host	Met
Encryption at rest	HIPAA §164.312(a)	AES-256 (managed Postgres)	Met
Access control	HIPAA §164.312(a)	RLS owner-scoping on all user tables	Met
Authentication	HIPAA §164.312(d)	Supabase Auth; functioning recovery (FR-003)	Partial
Audit controls	HIPAA §164.312(b)	Structured audit logging	Planned
Data minimisation	GDPR / NDMO	No PII to LLM; minimal collection	Met
Right to erasure	GDPR / NDMO	Account & data deletion (FR-018)	Partial
Data residency (KSA)	Saudi NDMO / MOH	Region confirmation required	Open

*Compliance statements represent alignment intent for a consumer wellness product and are not a certification. FamNest v1.0 does not process PHI as defined under HIPAA. Formal legal review is required before any regulated healthcare deployment or MOH integration.*

## 8. Open Issues

The following items are tracked against this specification. Items prefixed TFR are drawn from the companion Technical Findings & Remediation Report (FN-TFR-2026-001).

ID	Description	Affected Requirement	Status
TFR-01	UPDATE/DELETE RLS policies absent; edit/delete silently no-op	FR-008, FR-009	Open — HIGH
TFR-02	Password reset route missing (404 dead-end)	FR-003	Open — HIGH
TFR-03	Preferred reminder time not honoured by cron	FR-014	Open — MED
TFR-04	checkin_submitted analytics event never fires	FR-017	Open — MED
SRS-05	Audit logging not yet implemented	NFR-COMP / §7	Planned
SRS-06	KSA data residency region unconfirmed	§7 Compliance	Open

## 9. Appendix

### 9.1 Document Suite

Document	Reference	Lens
System Architecture Document	FN-SAD-2026-001	How the system is built
Software Requirements Specification	FN-SRS-2026-001	What the system must do (this document)
Technical Findings & Remediation Report	FN-TFR-2026-001	What is broken and how to fix it

### 9.2 Author

Authored by Virginia Mwega, full-stack developer and technical writer. This specification documents the requirements of a real, live production system (famnest-iota.vercel.app) and is cross-referenced with a companion architecture document and findings report.

Writing portfolio	viriniamwega-com.vercel.app
Live system	famnest-iota.vercel.app
Email	viriniamwega2@gmail.com

*This document is a portfolio and engineering artifact authored by Virginia Mwega. It demonstrates requirements engineering to IEEE Std 29148-2018, cross-referenced with companion architecture and findings documents to form a coherent documentation suite.*