SYSTEM ARCHITECTURE DOCUMENT

FamNest — AI Wellness Coach for Busy Parents

Prepared by:

Virginia Mwega

Full-Stack Developer & Technical Author

virginiamwega2@gmail.com | virginiamwega-com.vercel.app

Document Reference	FN-SAD-2026-001
Version	1.2 — Updated Baseline
Status	For Review
Classification	CONFIDENTIAL
Date	June 2026
Target Markets	United States | Kingdom of Saudi Arabia

1. Document Control

1.1 Version History

Version	Date	Author	Description
0.1	April 2026	V. Mwega	Initial draft — architecture defined
0.9	May 2026	V. Mwega	Internal review — security and compliance sections added
1.0	May 2026	V. Mwega	Baseline release — approved for external review
1.1	June 2026	V. Mwega	Architecture description aligned to ISO/IEC/IEEE 42010:2011; AI inference layer corrected to Groq Llama 3.3 70B (OpenAI-swappable); multi-agent pipeline and AI safety controls documented from live code (lib/agent.ts, app/actions/checkin.ts)
1.2	June 2026	V. Mwega	Payment integration corrected from Stripe to Flutterwave (verif-hash validation + transaction re-verification before access grant); KES / multi-currency noted

1.2 Reviewers & Approvers

Name	Role	Action	Date
Virginia Mwega	Author / Architect	Authored	June 2026
[Reviewer Name]	Technical Lead	Reviewed	TBD
[Approver Name]	Project Sponsor / CTO	Approved	TBD

1.3 Distribution List

Recipient	Organisation	Classification
Virginia Mwega	FamNest	Author
Technical Reviewers	Client / Partner	CONFIDENTIAL
Project Stakeholders	Client / Partner	CONFIDENTIAL

2. Introduction

2.1 Purpose

This System Architecture Document (SAD) describes the architectural design of FamNest, an AI-powered wellness coaching application purpose-built for busy parents. The document establishes the technical baseline for all development, integration, and operational activities and serves as the primary reference for stakeholders, technical teams, and compliance reviewers.

This document conforms to **ISO/IEC/IEEE 42010:2011 (Systems and software engineering — Architecture description)** and incorporates principles aligned with HIPAA technical safeguards (US market) and the Saudi

Ministry of Health National E-Health Strategy digital health architecture guidelines (KSA market).

2.2 Scope

This document covers all architectural layers of the FamNest platform version 1.0, including:

- Web application frontend (Next.js 15 — App Router)
- Application logic via Next.js server actions and API routes (Node.js runtime)
- Relational data store (PostgreSQL via Supabase)
- AI inference layer — multi-agent pipeline on **Groq Llama 3.3 70B**; provider-swappable to OpenAI
- Authentication and session management (Supabase Auth)
- Transactional email infrastructure (Resend)
- Deployment and hosting infrastructure (Vercel, Supabase Cloud)

2.3 Intended Audience

Audience	Primary Interest
Technical Architects	Component design, integration patterns, technology decisions
Software Engineers	Component interfaces, data models, API contracts
Security / Compliance	Auth flows, data classification, HIPAA / MOH alignment
Project Sponsors / CTO	System capability, scalability, technology risk
Healthcare IT Procurement	Standards alignment, compliance posture, vendor readiness

2.4 Definitions & Acronyms

Term	Definition
AI	Artificial Intelligence
API	Application Programming Interface
CDN	Content Delivery Network
DHAD	Digital Health Architecture Document
EMR	Electronic Medical Record
HIPAA	Health Insurance Portability and Accountability Act (US)
JWT	JSON Web Token
LLM	Large Language Model
MOH	Ministry of Health (Kingdom of Saudi Arabia)
NDMO	National Data Management Office (KSA)
ORM	Object-Relational Mapper
PHI	Protected Health Information
PII	Personally Identifiable Information
RAG	Retrieval-Augmented Generation

Term	Definition
RLS	Row-Level Security (PostgreSQL)
SAD	System Architecture Document
SRS	Software Requirements Specification
SSR	Server-Side Rendering
TLS	Transport Layer Security

2.5 References

- ISO/IEC/IEEE 42010:2011 — Systems and software engineering — Architecture description
- HIPAA Technical Safeguards — 45 CFR §164.312
- Saudi MOH National E-Health Strategy — Digital Health Transformation Framework
- OWASP Application Security Verification Standard (ASVS) 4.0
- Next.js 15 Documentation — Vercel Inc.
- Supabase Architecture Documentation — Supabase Inc.
- Groq API Reference — Groq, Inc. (default inference provider)
- OpenAI API Reference — OpenAI (provider-swappable alternative)

3. System Overview

3.1 Product Description

FamNest is a web-based AI wellness coaching platform designed for parents experiencing burnout, time pressure, and reduced capacity for self-care. The platform delivers personalised, time-bounded micro-wellness interventions derived from a proprietary daily check-in assessment.

Unlike conventional wellness applications that require sustained engagement or lengthy onboarding, FamNest is explicitly designed around the fragmented, unpredictable schedules of parents with young children. Every core interaction is achievable within 90 seconds or fewer, and no feature imposes guilt mechanics such as streaks, missed-day penalties, or persistent notification pressure.

> **Research basis:** Survey of 74 parents (April 2026) confirmed that primary blockers to wellness-app adoption are time constraints, guilt mechanisms, and mismatch between app structure and lived parenting experience. FamNest architecture is a direct response to these findings.

3.2 Key Capabilities

Capability	Description	Priority
Daily Check-In	3-slider mood / energy / time assessment producing an AI-generated personalised plan	P0 — Core
AI Wellness Plan	LLM-generated, time-bounded action recommendations matched to user state	P0 — Core

Capability	Description	Priority
Trend Dashboard	Aggregated weekly / monthly mood and energy visualisation	P1 — High
Premium Subscription	Flutterwave-gated premium tier with extended plan depth and history	P1 — High
Transactional Email	Onboarding, check-in reminders, and weekly digest emails via Resend	P1 — High
Crisis Signposting	Persistent access to mental-health crisis resources, visible at all states	P0 — Core / Safety

3.3 User Personas

Persona	Profile	Primary Use Case
The Exhausted Parent	Parent of children under 6, <10 mins free time per day	Daily check-in, micro-action, mood trend
The Working Parent	Dual-income household, structured schedule, moderate burnout	Weekly digest, trend review, planning mode
The Premium User	Engaged user, 30+ days retention, outcome-oriented	Deep AI plan, historical trends, journalling

4. Architectural Goals & Constraints

4.1 Architectural Goals

Goal	Description	Rationale
Responsiveness	Sub-2s page load; sub-5s AI plan generation	Core UX — parent has 90 seconds
Privacy-First	Minimal PII collection; no third-party data selling	Trust is the product in wellness
Resilience	AI layer fails gracefully; deterministic fallback always served	Single point of failure unacceptable
Scalability	Architecture supports 0–100K users without redesign	Growth path to enterprise / B2B2C
Compliance-Ready	HIPAA alignment (US); MOH / NDMO alignment (KSA)	Required for healthcare IT market
Maintainability	Solo-developer-operable; minimal infrastructure overhead	Stage 1 operational reality

4.2 Constraints

Constraint	Type	Impact
Solo development team	Resource	Architecture must minimise operational complexity
Vercel + Supabase hosting	Technology	Vendor lock-in accepted for velocity; documented for exit
Groq / OpenAI API dependency (provider-swappable)	Technology	AI quality tied to third-party LLM providers; provider switchable via LLM_PROVIDER env config
No native mobile app (v1)	Scope	PWA approach taken; native considered for v2
HIPAA alignment required (US)	Regulatory	Auth, encryption, audit-log requirements enforced

Constraint	Type	Impact
MOH / NDMO alignment (KSA)	Regulatory	Data residency and classification standards applied
Flutterwave for payments (KES + card, v1)	Business	African payment rails; multi-currency (incl. KSA SAR) planned for v2

5. System Architecture

5.1 Architecture Pattern

FamNest adopts a layered, service-oriented architecture built on a modern full-stack JavaScript runtime. The system follows a three-tier pattern:

- **Presentation Tier** — Next.js 15 (App Router) with Tailwind CSS, deployed to Vercel Edge Network
- **Application Tier** — Next.js server actions / API routes; business logic, validation, and the AI orchestration pipeline
- **Data Tier** — PostgreSQL (Supabase), Row-Level Security enforced at the database layer

This architecture prioritises developer velocity, operational simplicity, and provable security boundaries — appropriate for a compliant healthcare-adjacent consumer product at v1 scale.

5.2 Component Architecture (Textual Representation)

A normal check-in performs two model calls (coach draft + safety review); a draft flagged *revise* performs a third. The *crisis* verdict and the deterministic keyword floor both short-circuit to support resources.

5.3 Technology Stack

Layer	Technology	Version	Justification
Frontend	Next.js (App Router)	15.x	SSR, file-based routing, server actions — unified codebase
Styling	Tailwind CSS	3.4.x	Utility-first; rapid iteration
State / Data	TanStack Query	5.x	Server-state caching; reduces redundant calls
Backend Runtime	Node.js	20 LTS	LTS stability; broad ecosystem
Validation	Zod	3.x	End-to-end type-safe schema validation
Database	PostgreSQL (Supabase)	15.x	ACID; RLS; proven at scale
Auth	Supabase Auth	2.x	JWT sessions; OAuth; PKCE flow
AI Inference	**Groq — Llama 3.3 70B** (OpenAI-swappable)	llama-3.3-70b-versatile	Low-latency inference; structured output; provider switchable via LLM_PROVIDER
AI SDK	Vercel AI SDK	3.x	Streaming; generateObject; Next.js native
Payments	Flutterwave	Latest	PCI-DSS Level 1; African payment rails; subscriptions
Email	Resend	Latest	Developer-first; high deliverability
Hosting	Vercel	Latest	Zero-config Next.js; global edge
DB Hosting	Supabase Cloud	Latest	Managed Postgres; RLS; backups

6. Data Architecture

6.1 Data Model — Core Entities

6.1.1 users

Column	Type	Constraints	Description
id	UUID	PK, NOT NULL	Supabase Auth UID — primary identifier
email	TEXT	UNIQUE, NOT NULL	User email — login credential
display_name	TEXT	NULLABLE	Optional display name
plan_tier	TEXT	DEFAULT 'free'	'free' | 'premium'
flw_ref	TEXT	NULLABLE	Flutterwave customer / transaction reference
created_at	TIMESTAMPTZ	DEFAULT NOW()	Account creation timestamp
updated_at	TIMESTAMPTZ	DEFAULT NOW()	Last profile update

6.1.2 check_ins

Column	Type	Constraints	Description
id	UUID	PK, gen_random_uuid()	Check-in record identifier

Column	Type	Constraints	Description
user_id	UUID	FK → users.id, NOT NULL	Owning user — RLS enforced
mood	SMALLINT	CHECK (1–10), NOT NULL	Self-reported mood score
energy	SMALLINT	CHECK (1–10), NOT NULL	Self-reported energy score
stress	SMALLINT	CHECK (1–10), NOT NULL	Self-reported stress score
time_available	SMALLINT	NOT NULL	Available minutes (1–60)
created_at	TIMESTAMPTZ	DEFAULT NOW()	Check-in timestamp

6.1.3 ai_plans

Column	Type	Constraints	Description
id	UUID	PK, gen_random_uuid()	Plan record identifier
check_in_id	UUID	FK → check_ins.id	Parent check-in
user_id	UUID	FK → users.id, NOT NULL	Owning user — RLS enforced
actions	JSONB	NOT NULL	Array of recommended micro-actions
tone_flag	TEXT	NOT NULL	'gentle' | 'encouraging' | 'calm'
tokens_used	INTEGER	NULLABLE	LLM token consumption — cost tracking
model_version	TEXT	NOT NULL	LLM identifier used (e.g. llama-3.3-70b-versatile)
created_at	TIMESTAMPTZ	DEFAULT NOW()	Plan generation timestamp

6.2 Row-Level Security (RLS) Policy

PostgreSQL Row-Level Security is enabled on all user-data tables. No application-layer bypass is permitted. The enforced policy pattern is:

```
CREATE POLICY "user_owns_data" ON check_ins
  USING (user_id = auth.uid());
```

This ensures that even in the event of application-layer misconfiguration, a user's data cannot be accessed by any other user.

6.3 Data Classification

Classification	Examples	Handling
PII — Identifiable	Email address, display name	Encrypted at rest (AES-256); TLS in transit; minimal collection
Sensitive Wellness	Mood, stress, energy scores	RLS enforced; not shared with third parties; user-deletable
Operational	Token usage, model version, timestamps	Internal use; retained for cost optimisation and audit
Public / Non-sensitive	Plan tone flags (aggregate)	May be used in anonymised product analytics

7. Security Architecture

7.1 Authentication & Session Management

FamNest uses Supabase Auth for all authentication flows. The implementation follows the PKCE (Proof Key for Code Exchange) pattern for OAuth flows and issues short-lived JWT access tokens (1-hour expiry) with refresh-token rotation.

Control	Implementation	Standard Alignment
Identity Provider	Supabase Auth (email/password + OAuth)	HIPAA §164.312(d)
Token Type	JWT — RS256 signed, 1-hour expiry	OWASP ASVS V3.3
Refresh Rotation	Refresh token rotated on each use	OWASP ASVS V3.3.3
Password Hashing	bcrypt, cost factor 12	NIST SP 800-63B
Session Termination	Explicit logout invalidates refresh token	HIPAA §164.312(a)(2)(iii)
MFA (Roadmap)	TOTP planned for premium tier (v2)	HIPAA §164.312(d)

7.2 Transport & Encryption

Control	Specification
TLS Version	TLS 1.3 enforced; TLS 1.2 minimum; SSLv3 / TLS 1.0 / 1.1 disabled
Certificate	Vercel-managed TLS certificates (auto-renewed)
HSTS	Strict-Transport-Security: max-age=31536000; includeSubDomains
Encryption at Rest	AES-256 — Supabase managed (database volumes)
API Key Storage	Environment variables only; never client-exposed; never in VCS

7.3 API Security Controls

- Input validation on all endpoints — Zod schema enforcement; malformed payloads rejected with 400
- Rate limiting — Edge rate limiting on the AI plan endpoint; 10 requests/minute per authenticated user
- CORS — restricted to application origin; wildcard (*) not permitted
- Content Security Policy — restrictive policy applied; inline scripts blocked
- Dependency scanning — automated via GitHub Dependabot; critical CVEs resolved within 48 hours

7.4 AI Safety Controls

The AI inference layer introduces risks not present in conventional CRUD applications. FamNest applies a multi-stage control model — a two-agent review, a deterministic crisis floor, and graceful degradation — across the plan-generation pipeline:

Risk	Control	Status
Unreviewed harmful output	Two-stage agent design: a dedicated **safety-reviewer agent** evaluates every coach draft and returns ok / revise / crisis; *revise* triggers one bounded revision pass before shipping	Implemented

Risk	Control	Status
Harmful output to distressed user	**Deterministic crisis floor**: a keyword guard escalates self-harm language to crisis resources independent of model output — enforced both online and in the offline fallback	Implemented
Provider / inference failure	**Graceful degradation** to a deterministic offline plan (mockRecommendation — zero LLM calls) carrying its own offline keyword crisis guard; users never hit a dead end	Implemented
Hallucinated medical advice	System prompt prohibits medical diagnosis; plans framed as wellness suggestions, not clinical guidance	Implemented
Prompt injection	User inputs passed as data parameters, not concatenated into the system prompt	Implemented
Cost runaway	max_tokens capped per request; per-user daily token usage logged; alerting configured (2 model calls / check-in, 3 on revision)	Implemented
PII leakage to LLM	No PII (email, name) included in prompts; only anonymised scores and retrieved snippets passed	Implemented

Compliance note: FamNest does not collect, store, or process Protected Health Information (PHI) as defined under HIPAA. Mood and wellness scores are self-reported user preferences, not clinical data. This classification should be reviewed by qualified legal counsel prior to any clinical or healthcare-system integration.

8. Integration Architecture

8.1 External Service Integrations

Service	Purpose	Integration Pattern	Data Shared
Groq API (Llama 3.3 70B)	AI plan generation — coach + safety agents	REST / Vercel AI SDK	Anonymised scores, retrieved snippets
OpenAI API	Provider-swappable inference alternative (via LLM_PROVIDER)	REST / Vercel AI SDK	Anonymised scores, retrieved snippets
Flutterwave	Subscription & payments	Webhooks (verif-hash) + REST API	Email, subscription status, transaction ID
Resend	Transactional email	REST API / React Email	Email address, plan summaries (user-controlled)
Supabase Auth	Identity management	SDK (PKCE OAuth flow)	Email, hashed password

8.2 Flutterwave Webhook Flow

Payment events are received and processed via Flutterwave webhooks. The handler first validates the **verif-hash** signature header, then re-verifies the transaction via verifyTransaction(event.data.id) against Flutterwave's API before granting any access — a valid signature alone is never trusted as proof of payment.

- charge.completed (transaction re-verified as successful, amount and currency matched) → activate subscription; update plan_tier to 'premium'

- charge.completed (re-verification fails or amount / currency mismatch) → reject; no access granted; logged for review
- failed / pending / unrecognised events → acknowledged with HTTP 200 and ignored (no state change)

8.3 AI Plan Generation Flow

The check-in to AI-plan pipeline is a single entry point, generateRecommendation() in lib/agent.ts, invoked by the authenticated check-in server action. It is a multi-agent loop — not a single model call:

1. User submits check-in (mood, energy, stress, time_available) via authenticated server action (app/actions/checkin.ts).
2. Zod validates inputs server-side; invalid inputs are rejected with 400.
3. Check-in record is written to check_ins (RLS enforced).
4. **RAG grounding** — retrieveKnowledge() pulls vetted wellness snippets to ground the draft.
5. **Coach agent** drafts a history-aware plan, conditioned on retrieved snippets and user memory (buildUserMemory). No PII is included.
6. **Safety agent** — a second model call reviews the draft and returns a verdict: ok | revise | crisis.
7. **Branch** — crisis ⇒ output replaced with crisis support resources; revise ⇒ coach performs one bounded revision pass; ok ⇒ plan ships.
8. **Deterministic crisis floor** — an independent keyword guard escalates self-harm language to crisis resources regardless of the model verdict.
9. Plan is written to ai_plans with token usage and model_version logged.
10. Structured plan returned to the client and rendered as action cards.
11. **Graceful degradation** — any failure (missing key, quota, network, malformed JSON) drops to mockRecommendation(): a deterministic offline plan with zero LLM calls and its own offline keyword crisis guard, so a parent never hits a dead end.

9. Deployment Architecture

9.1 Environments

Environment	Purpose	URL Pattern	Notes
Production	Live user traffic	famnest-iota.vercel.app	Vercel + Supabase production project
Preview	PR review & demo	famnest-[hash].vercel.app	Auto-generated per pull request
Development	Local development	localhost:3000	.env.local — separate Supabase dev project

9.2 CI/CD Pipeline

- Source control: GitHub — main branch production-protected; PRs required for all changes
- CI: GitHub Actions — runs Vitest test suite on every push and pull request
- Deployment: Vercel GitHub integration — automatic deployment on merge to main

- Database migrations: Supabase CLI — migration files version-controlled in /supabase/migrations
- Secrets: Vercel Environment Variables — never committed to VCS; rotated quarterly

9.3 Scalability Considerations

Component	Current Scale	Scale Path
Frontend / SSR	Vercel Edge — global CDN	Auto-scales; no action required
Server Actions / API	Vercel Serverless Functions	Auto-scales; cold start managed via Fluid Compute
PostgreSQL	Supabase Free / Pro tier	Vertical scaling; read replicas at 50K+ MAU
AI Inference	Groq API — standard tier (OpenAI-swappable)	Rate-limit increase request at 10K+ daily check-ins
Email	Resend — developer plan	Upgrade to production plan at 1K+ daily sends

10. Non-Functional Requirements

NFR ID	Category	Requirement	Target	Status
NFR-01	Performance	Page load time (LCP)	< 2.0s (P95)	Target
NFR-02	Performance	AI plan generation time	< 5.0s (P95)	Target
NFR-03	Availability	Platform uptime	> 99.5% monthly	Target
NFR-04	Security	TLS version	TLS 1.3 minimum	Met
NFR-05	Security	Auth token expiry	60 min maximum	Met
NFR-06	Privacy	PII in AI prompts	Zero PII transmitted	Met
NFR-07	Compliance	HIPAA technical safeguards	Aligned (not certified)	Partial
NFR-08	Compliance	MOH / NDMO data classification	Documented baseline	Partial
NFR-09	Usability	Core check-in interaction time	< 90 seconds	Met
NFR-10	Scalability	Architecture supports growth	0–100K users, no redesign	Met

11. Open Issues & Architecture Roadmap

11.1 Known Open Issues

ID	Issue	Priority	Owner
OI-01	Formal HIPAA BAA not in place with Supabase / Groq (and OpenAI) — required for any clinical integration	High	V. Mwega
OI-02	KSA data residency — confirm Supabase EU/ME region availability for MOH compliance	High	V. Mwega
OI-03	MFA not yet implemented — planned for v1.1 premium tier	Medium	V. Mwega

ID	Issue	Priority	Owner
OI-04	AI plan evals framework not yet formalised — manual testing only at v1	Medium	V. Mwega
OI-05	Audit logging — structured audit trail not yet implemented; required for HIPAA §164.312(b)	Medium	V. Mwega

11.2 Architecture Roadmap

Version	Feature	Architecture Impact
v1.1	Mobile PWA optimisation	Service Worker; offline check-in; push notifications
v1.1	MFA (TOTP)	Supabase Auth MFA API integration
v1.2	Native iOS / Android	React Native (Expo) — shared business logic; native shell
v2.0	B2B2C — Employer Wellness	Multi-tenancy; org-level RLS; admin dashboard; SAML SSO
v2.0	Clinical Integration (future)	FHIR R4 API layer; HIPAA BAA; formal healthcare IT compliance
v2.0	KSA Localisation	Arabic RTL UI; SAR currency; MOH DHAD formal certification

12. Appendix

12.1 Author Profile

This document was authored by Virginia Mwega, a full-stack developer and technical writer based in Nairobi, Kenya, building wellness and productivity tools for parents globally.

Portfolio (Writing)	virginiamwega-com.vercel.app
Portfolio (Dev)	virginia-mwega.vercel.app
FamNest (Live)	famnest-iota.vercel.app
Hashnode (Technical)	virginiamwegahashnodedev.hashnode.dev
Medium (Founder)	medium.com/@virginiamwega2
Email	virginiamwega2@gmail.com
LinkedIn	linkedin.com/in/virginia-mwega-196309313

12.2 Related Documents

Document	Reference	Status
Software Requirements Specification	FN-SRS-2026-001	Available
Technical Findings & Remediation Report	FN-TFR-2026-001	Available
API Reference	FN-API-2026-001	Available
Digital Health Architecture Document	FN-DHAD-2026-001	Planned (KSA)
Security Assessment Report	FN-SAR-2026-001	Planned
FamNest Architecture — Hashnode Public Blog Post	—	Published

> This document is a portfolio and demonstration artifact authored by Virginia Mwega. Architecture details reflect the actual FamNest v1.0 system as deployed. Compliance statements represent alignment intent, not formal certification. Legal review is recommended before any regulated healthcare deployment.