International Journal of Software & Hardware Research in Engineering (IJSHRE)
ISSN:2347-4890, ijournals.in/ijshre
Volume 14 Issue 9, September 2026

# System Design and Architecture of a Cross-Platform Mobile Health and Habit-Tracking Application for Nutrition, Exercise, and Growth Monitoring in Shelter-Home Care

**Authors: Siddharth Narang[1], Vinay Vishwakarma[2], Shreya Narang[3]**

Affiliation: Student, Dhirubhai Ambani International School, Mumbai[1]; Mentor, On My Own Technology[2]; University of Pennsylvania[3]

**http://doi.org/10.26821/ijshre.14.09.2026.140903**

## ABSTRACT

Institutional care settings such as shelter homes face persistent difficulty in systematically tracking the nutrition, physical activity, growth, and screen-time behaviour of the children under their supervision, because such tracking is still largely performed on paper registers that are slow to update, difficult to aggregate, and impossible to analyse at scale. This paper presents AthLead, a cross-platform mobile health and habit-tracking application engineered with React Native, Expo Router, Redux Toolkit, and a Firebase Realtime Database backend, designed specifically for Shelter Home Nutrition and Fitness Programs. Unlike a purely feature-level description, this paper foregrounds system design: it derives functional and non-functional requirements from the target use case, enumerates actors and use cases, presents a three-tier architecture, documents the meal-logging data flow as a sequence diagram, specifies the Firebase Realtime Database schema as a data-model diagram, and identifies the software design patterns (Flux/Redux unidirectional state flow, a thin repository-style data-access layer, and file-based routing) that structure the codebase. These design decisions are grounded in a structured review of fifteen peer-reviewed and indexed sources spanning mHealth behaviour change, pediatric digital nutrition interventions, screen-time health effects, gamified exercise adherence, cross-platform and Backend-as-a-Service (BaaS) architecture, and mHealth privacy and security. The paper closes with a discussion of how the realised design aligns with this literature, its limitations, and concrete, literature-motivated directions for future system evolution.

*Keywords: system design, software architecture, mobile health (mHealth), habit tracking, React Native, Expo, Firebase Realtime Database, Redux Toolkit, data modelling, child nutrition monitoring, shelter habit tracking - home care.*

## INTRODUCTION

The proliferation of smartphones and cloud-backed mobile platforms has repositioned mobile health (mHealth) applications as a primary channel for behaviour-change interventions in nutrition, physical activity, and self-monitoring [1]. While a large share of this literature addresses adult, self-directed users, a smaller but growing body of work addresses caregiver-mediated health tracking for children in institutional settings such as orphanages, shelter homes, and residential care facilities, where nutrition and growth outcomes are shaped less by individual choice than by the consistency of caregiver record-keeping [2]. Children living in institution-based care face documented, compounding vulnerabilities to malnutrition and inconsistent monitoring, and comprehensive assessment and monitoring systems have been identified as one of the few interventions with demonstrated positive effect on feeding outcomes in such settings.

AthLead is a mobile application designed to address this specific gap. Built for Shelter Home Nutrition and Fitness Programs, it allows caregivers, nutrition staff, and health coordinators to onboard children (aged 8-16 years) with individual health profiles, assign them to region-specific weekly meal plans (currently Mumbai and Delhi cuisines), track meal completion against age-wise portion guidelines, log structured exercise sessions with associated calorie burn, monitor BMI trends, apply screen-time limits, and export PDF summaries for auditing and reporting.

This paper is deliberately structured around system design rather than feature description. Section III does not merely list what AthLead does; it derives the requirements that motivate the design, enumerates the actors and their use cases, presents the layered architecture as a diagram, traces a representative operation (marking a meal complete) end-to-end as a sequence diagram, specifies the database schema as a data-model diagram, and names the software design patterns used and why. Section II situates these design choices within fifteen indexed academic sources. Section IV documents the concrete technology stack realising the design. Sections V-VII discuss the design against the literature, its limitations, and directions for future evolution.

## RELATED WORK AND LITERATURE REVIEW

This section reviews fifteen sources across six thematic clusters that ground the design decisions elaborated in Section III: (A) mobile health and habit-formation applications in general; (B) digital nutrition interventions for children and institutional care; (C) screen-time monitoring and its health consequences; (D) gamification and exercise adherence; (E) cross-platform mobile development and layered/BaaS system architecture; and (F) privacy and security in mHealth system design.

### *A. Mobile Health and Habit-Formation Applications*

Milne-Ives et al. [1] conducted a systematic review of 52 studies evaluating mobile applications for health behaviour change across physical activity, diet, and other domains, finding that effective apps consistently incorporated self-monitoring, goal-setting, and feedback as core behaviour-change techniques — the same triad that AthLead's use-case analysis (Section III-B) formalises as the primary caregiver use cases. Basto and Ferreira [15] found, from interviews with adult app users, that the ability to view historical, self-referential progress data is a primary engagement driver, motivating the inclusion of the dashboard and progress-view use cases in AthLead's requirements set.

### *B. Digital Nutrition Interventions for Children and Institutional Care*

Prowse and Carsley [2] performed an umbrella review of digital nutrition interventions for children under 18, concluding that theory-informed interventions using explicit tailoring and goal-setting show the most promise, and calling for common, validated outcome measures — a requirement AthLead's design addresses through structured, auditable age-wise portion tables (Section III-A, functional requirement FR-3) rather than free-text logging. Lee, Su, and Chen [3] analysed 43 commercial children's health apps and found that caregiver-mediated, dual-use workflows suited to institutional contexts remain an underserved design space, directly motivating AthLead's multi-student, caregiver-centred actor model (Section III-B). Nossair and El Housni [13] describe a React Native dietary-assistant application using on-device food recognition to log meals; while AthLead's current design uses caregiver-marked completion, this represents a concrete future extension to the data-access layer discussed in Section VI.

### *C. Screen-Time Monitoring and Pediatric Digital Well-Being*

Piszka et al. [4] reviewed the short- and long-term health consequences of excessive pediatric screen time, documenting associations with obesity, sleep disruption, and myopia — the evidentiary basis for the screen-time-limit attribute in AthLead's student data model (Section III-E). Kristensen et al. [12] validated a purpose-built screen-time tracking application against native OS tools, finding near-perfect agreement on Android but a systematic bias on iOS, a design caution relevant should AthLead's schema evolve from a caregiver-set limit to automated usage capture.

### *D. Gamification and Exercise Adherence*

Rehman et al. [6], applying the GAMEFULQUEST instrument to 240 fitness-app users, found that challenge, immersion, and social interaction significantly predicted engagement, while purely competitive elements did not — evidence favouring AthLead's current, non-competitive, completion-based exercise use case over leaderboard mechanics. Korkmaz and Çoruh [7] ran a four-week cluster-randomised trial of a gamified exercise-tracking application in secondary-school students and reported substantially higher goal completion (65-68% versus 15-40% in the control arm), providing age-matched, design-relevant evidence for extending AthLead's exercise module architecture (Section VI).

### *E. Cross-Platform Development and Layered / Backend-as-a-Service System Architecture*

AthLead's presentation tier is built on React Native with Expo Router rather than as two separate native codebases, and its data tier is built on Firebase rather than a self-hosted relational server. Saraf et al. [8] reviewed Firebase as a Backend-as-a-Service (BaaS) platform, characterising the Realtime Database as a JSON-tree, cloud-hosted NoSQL store that synchronises data to all connected clients with millisecond-level latency and provides built-in offline caching, the mechanism underlying the data-flow design in Section III-D. Efendi et al. [9], in a fielded IoT-based elderly-health-monitoring system, adopt an explicit three-layer architecture (physical, network, and application layers) around an Android front end and a Firebase Realtime Database backend, and report that this layering directly analogous to AthLead's presentation/state / data-access tiers (Section III-C) supported real-time data collection with low error rates (mean absolute percentage error of 0.90% across monitored parameters) in a comparable caregiver-facing health-tracking context, reinforcing layered separation of concerns as an appropriate architectural style for this class of system.

### *F. Privacy and Security in mHealth System Design*

Because AthLead's data model stores minors' names, ages, heights, weights, and daily nutrition/exercise records, its security design is grounded directly in the mHealth privacy literature. Kotz et al. [10], in a foundational IEEE Computer research agenda, argue that mHealth system design must treat authentication, data-at-rest encryption, and access control as first-class architectural requirements rather than afterthoughts. Iwaya, Ahmad, and Babar [11], in a systematic mapping study spanning 365 primary studies, similarly find that authentication and secure transmission are the two most frequently studied — and most frequently deficient — control categories in fielded mHealth system designs. AthLead's architecture addresses the authentication dimension through Firebase Authentication and identity-scoped Realtime Database security rules, consistent with the access-control requirements identified in [10] and [11] and formalised as non-functional requirement NFR-4 in Section III-A.

## III. SYSTEM DESIGN

This section presents Athlead's system design in the conventional order of a software design specification: design objectives and requirements; actors and use cases; high-level architecture; a representative data-flow / sequence; the data-model / database schema; module-level component design; the software design patterns employed; and the non-functional design considerations that shaped implementation choices.

### *A. Design Objectives and Requirements*

The design objective is to replace paper-based nutrition and fitness registers in shelter-home care with an auditable, real-time, multi-caregiver digital system, without requiring reliable, continuous internet connectivity at the point of use. This objective was decomposed into the functional and non-functional requirements in Table II.

**Table II. Functional and non-functional requirements**

| ID | Requirement | Type |
|---|---|---|
| FR-1 | Caregivers shall authenticate (sign up, log in, reset password) before accessing any student record. | Functional |
| FR-2 | Caregivers shall create and maintain student profiles (name, age, height, weight, screen-time limit, region). | Functional |
| FR-3 | The system shall present a region- and age-appropriate structured meal plan and record per-slot completion (breakfast/lunch/snack/dinner) per student per day. | Functional |
| FR-4 | The system shall present a structured daily exercise routine and record per-exercise completion, duration, and calories burned. | Functional |
| FR-5 | The system shall compute BMI from recorded height and weight per student. | Functional |
| FR-6 | The system shall present a dashboard summarising meal-completion percentage, exercise progress, and calorie totals across all students managed by a caregiver. | Functional |
| FR-7 | The system shall generate and share a PDF report summarising a student's weekly meal and exercise record. | Functional |
| NFR-1 | Data entered while offline shall be cached locally and synchronised automatically once connectivity is restored. | Non-functional (Availability) |
| NFR-2 | The application shall run on both Android and iOS from a single codebase. | Non-functional (Portability) |
| NFR-3 | State updates shall propagate from the data tier to the UI within the same session without a manual refresh. | Non-functional (Real-time responsiveness) |
| NFR-4 | Access to a student's records shall be restricted to the authenticated caregiver who owns that record. | Non-functional (Security) |
| NFR-5 | The UI shall follow a consistent Material Design language across all screens. | Non-functional (Usability) |

### *B. Actors and Use-Case Analysis*

Three actor roles were identified from the target deployment context: the Caregiver / Nutrition Staff (the primary actor, who onboards students and performs day-to-day logging), the Shelter-Home Administrator (who reviews aggregated reports), and the System itself (which performs BMI computation and PDF generation as automated, non-human actors in use-case terms). Table III enumerates the primary use cases realised in the current implementation.

**Table III. Primary use cases and traceability to requirements**

| Use Case | Primary Actor | Description | Trace |
|---|---|---|---|
| UC-1: Register / Authenticate | Caregiver | Create an account or log in via Firebase Authentication; reset password if needed. | FR-1 |
| UC-2: Manage Student Profile | Caregiver | Add/edit a student's age, height, weight, screen-time limit, and region assignment. | FR-2 |

| Use Case | Primary Actor | Description | Trace |
|---|---|---|---|
| UC-3: Track Meal Completion | Caregiver | View the day's region-specific, age-appropriate meal plan and mark each slot complete. | FR-3 |
| UC-4: Track Exercise Completion | Caregiver | View the day's exercise routine and mark each exercise complete, logging duration/calories. | FR-4 |
| UC-5: View Dashboard | Caregiver | View real-time aggregated meal/exercise/calorie progress across all managed students. | FR-5, FR-6 |
| UC-6: Generate PDF Report | Caregiver | Export a weekly nutrition/exercise summary as a shareable PDF. | FR-7 |
| UC-7: Compute BMI (system) | System | Automatically compute and classify BMI whenever height/weight is updated. | FR-5 |

### *C. High-Level System Architecture*

AthLead follows a three-tier architecture (Fig. 1): a presentation tier of React Native screens organised by Expo Router's file-based (auth) and (main) route groups; a state and business-logic tier consisting of two Redux Toolkit slices (authSlice and trackerSlice) kept in sync with the backend via a useTrackerSync hook; a thin data-access tier of four Firebase API modules (authApi.ts, mealPlanApi.ts, exerciseApi.ts, trackerApi.ts) that isolate all Firebase SDK calls from UI and state code; and a cloud data tier comprising Firebase Authentication and the Firebase Realtime Database. This layering follows the same physical/network/application-style separation of concerns validated for a comparable Firebase-backed caregiver health-monitoring system in [9], and keeps each tier independently testable and replaceable — for example, the data-access tier could later be re-pointed at a different backend without changes to the presentation or state tiers.

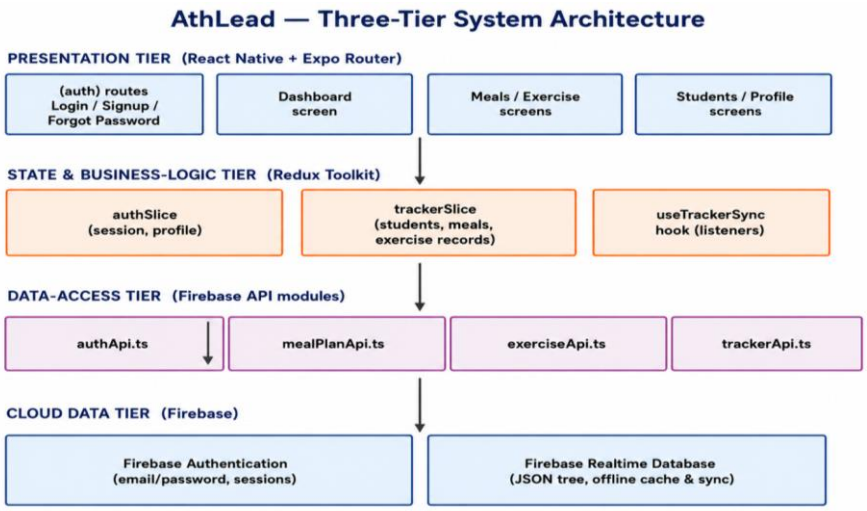


***Fig. 1. AthLead three-tier system architecture: presentation, state/business-logic, data-access, and cloud data tiers.***

### *D. Data Flow: Meal-Logging Sequence*

Fig. 2 traces UC-3 (Track Meal Completion) end-to-end as a sequence of eight steps: a caregiver's tap on the Meals screen dispatches a Redux action, which is handled by an async thunk in mealPlanApi.ts that writes to the completedMeals/{date}/{slot} path in the Firebase Realtime Database. Firebase's onValue listener, registered by useTrackerSync, fires on that same write (and on any other client's write to the same path), pushing the updated snapshot back into trackerSlice, which triggers a re-render of the Meals screen showing the slot as complete. Because the Realtime Database SDK maintains a local cache, steps 1-3 complete instantly even offline, with steps

4-8 deferred until connectivity resumes — realising NFR-1 without any offline-specific code in the UI or state tiers.

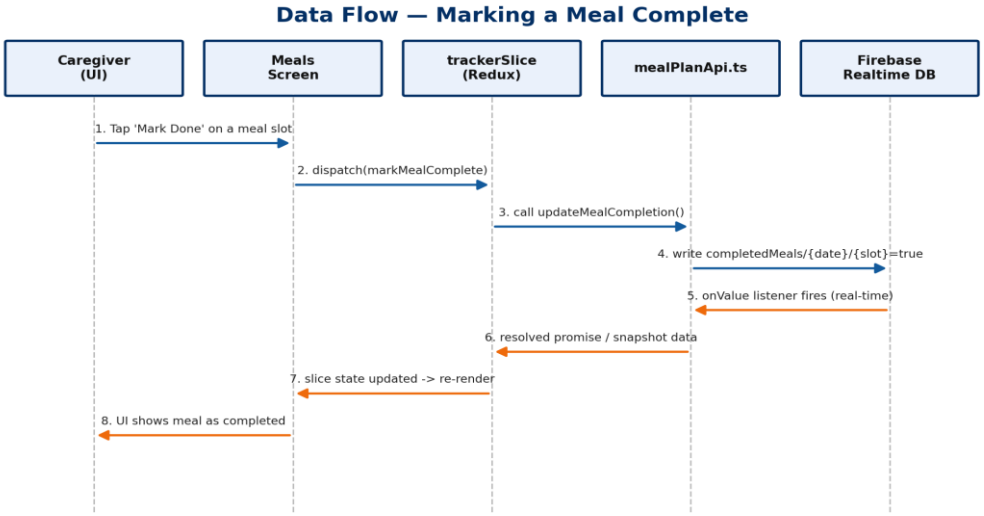


***Fig. 2. Sequence diagram for marking a meal complete, showing the eight-step round trip from UI tap to synchronised state.***

### *E. Data Model and Database Schema Design*

The Firebase Realtime Database schema (Fig. 3) is rooted at users/{userId} and organised as a denormalised JSON tree with four child nodes: profile (caregiver identity), students/{studentId} (per-child attributes, including the screenTimeLimitMin field motivated by [4]), mealPlans/{planId} (a weeklyMenu reference plus a completedMeals map keyed by ISO date), and exercisePlans/{planId} (an exercises map keyed by ISO date, each entry carrying name, duration, calories, and a completed flag). This design deliberately favours read/write locality over relational normalisation: a caregiver's dashboard query for 'today's status across all students' resolves against a small, co-located subtree rather than requiring joins across separate collections, which is the data-modelling pattern recommended for the Firebase Realtime Database in [8].

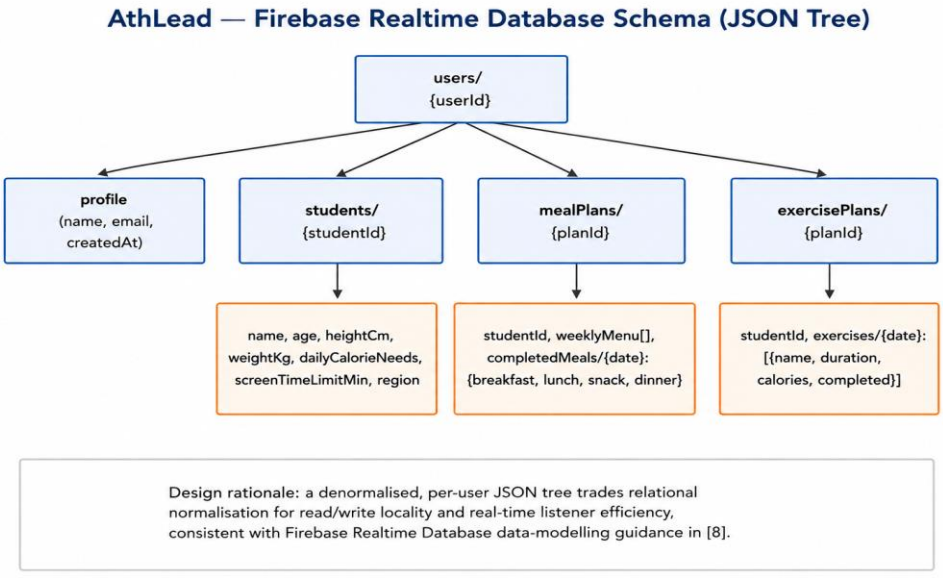


***Fig. 3. Firebase Realtime Database schema for AthLead, rooted at users/{userId}.***

### *F. Module-Level Component Design*

**Table IV. Module-level component design**

| Module / File | Responsibility |
|---|---|
| app/(auth)/*.tsx | Login, signup, and forgot-password screens; unauthenticated route group |
| app/(main)/dashboard.tsx | Aggregated real-time progress dashboard (UC-5) |
| app/(main)/meals/index.tsx | Daily meal plan display and completion toggling (UC-3) |
| app/(main)/exercise/index.tsx | Daily exercise routine display and completion toggling (UC-4) |
| app/(main)/children/index.tsx | Student profile list and management (UC-2) |
| src/store/authSlice.ts | Redux slice: caregiver session and profile state |
| src/store/trackerSlice.ts | Redux slice: students, meal-completion, exercise-completion state |
| src/hooks/useTrackerSync.ts | Subscribes Redux store to relevant Firebase Realtime Database listeners |
| src/firebase/authApi.ts | Firebase Authentication calls (sign-up, login, reset) |
| src/firebase/mealPlanApi.ts | Reads/writes meal-plan and completedMeals data |
| src/firebase/exerciseApi.ts | Reads/writes exercise-plan and completion data |
| src/utils/bmi.ts | BMI computation and classification (UC-7) |
| src/constants/shelterHomeMealPlan.ts | Regional weekly menus (KOLKATA_WEEKLY_MENU, DELHI_WEEKLY_MENU) and portion tables |

### *G. Design Patterns Employed*

Three design patterns structure the codebase. First, a Flux/Redux-style unidirectional data flow (Section III-D) — UI events dispatch actions, actions are handled by a single reducer per slice, and components subscribe to derived state — gives predictable, traceable state transitions and avoids the prop-drilling and inconsistent-state problems associated with ad-hoc component state in an application with cross-screen shared data (student lists, completion records). Second, a repository-style data-access layer (the four *Api.ts modules in Table IV) isolates Firebase SDK calls behind plain async functions, so UI and state code depend only on a stable internal interface rather than directly on the Firebase SDK, bounding the impact of any future backend migration to a single tier. Third, file-based routing (Expo Router's (auth) and (main) directory groups) encodes the authentication boundary directly in the project's file structure, so the set of screens reachable without a session is enforced by directory placement rather than by scattered conditional checks.

### *H. Non-Functional Design Considerations*

- Availability / offline-first (NFR-1): realised by the Firebase Realtime Database client SDK's local cache, requiring no additional offline-sync code in Athlead itself (Section III-D).
- Portability (NFR-2): realised by targeting React Native/Expo rather than separate native codebases, at the performance and native-integration trade-offs documented in the cross-platform literature.
- Real-time responsiveness (NFR-3): realised by Firebase's onValue listeners pushing updates directly into Redux state rather than relying on manual polling or pull-to-refresh.

- Security / access control (NFR-4): realised by Firebase Authentication plus identity-scoped Realtime Database security rules restricting each caregiver to their own users/{userId} subtree, consistent with [10] and [11].
- Usability (NFR-5): realised by building all screens from React Native Paper, a Material Design component library, giving visual consistency without bespoke styling per screen.

## IV. TECHNOLOGY STACK AND DEVELOPMENT METHODOLOGY

Table V summarises the concrete technology stack that realises the design specified in Section III.

**Table V. Athlead technology stack**

| Layer | Technology | Design Role |
|---|---|---|
| UI Framework | React Native 0.81.5 + Expo SDK 54 | Presentation tier; cross-platform rendering (NFR-2) |
| Navigation | Expo Router 6.0.21 | File-based routing enforcing the auth boundary (Sec. III-G) |
| State Management | Redux Toolkit 2.11.2 + React-Redux | State/business-logic tier; Flux-style unidirectional flow |
| Backend / Database | Firebase Realtime Database | Cloud data tier; offline-capable JSON store (NFR-1, NFR-3) |
| Authentication | Firebase Authentication | Identity and access control (NFR-4) |
| UI Component Library | React Native Paper 5.14.5 | Material Design consistency (NFR-5) |
| Data Visualisation | React Native Chart Kit | Dashboard charts (UC-5) |
| Reporting | Expo Print + Expo Sharing | PDF report generation and sharing (UC-6) |
| Language / Typing | TypeScript 5.9.2 | Static typing of data-model interfaces (Sec. III-E) |

Development followed an iterative, design-first sequence consistent with Section III: (i) the data model (Fig. 3) was fixed first, as TypeScript interfaces in src/types/tracker.ts; (ii) the authentication boundary and route groups were scaffolded next; (iii) the Redux slices and the four Firebase API modules were implemented as the data-access tier; (iv) screen-level components were built against that state layer; and (v) regional meal-plan content, PDF export, and screen-time configuration were added as later, additive features, without requiring changes to the underlying architecture — evidence, in itself, that the layering in Fig. 1 achieved its intended separation-of-concerns objective.

## V. DISCUSSION

Read against the literature reviewed in Section II, AthLead's system design is well aligned with established mHealth design principles: its use-case set operationalises self-monitoring and feedback [1] directly as UC-3 through UC-6, its actor model addresses the caregiver-mediated, dual-use design gap identified for children's health apps [3], and its layered, Firebase-backed architecture mirrors the physical/network/application layering validated for a comparable caregiver health-monitoring system in [9]. The offline-first data flow in Fig. 2 — a direct consequence of choosing the Firebase Realtime Database (Section III-C, III-D) — matters concretely in shelter-home settings where connectivity may be intermittent, and is supported by the store-and-synchronise characterisation of the Realtime Database in [8].

At the same time, three design gaps emerge from the comparison. First, the gamification literature [6], [7] indicates that lightweight, non-competitive reinforcement (streaks, milestone badges) measurably raises exercise-goal completion in school-age children, yet UC-4 as currently designed exposes only binary completion tracking with no such layer in the trackerSlice state shape. Second, the screen-time element of the data model (Section III-E) currently stores only a caregiver-set limit rather than measured usage; extending it to automated capture would, per [12], require platform-specific bias correction rather than a single cross-platform assumption. Third, while the access-control design (NFR-4) follows [10] directly, it has not yet been evaluated against the specific control gaps — transport security, at-rest encryption, session handling — that [11] identifies as most frequently deficient in fielded mHealth system designs, and would benefit from a structured security-design review before wider institutional rollout.

## VI. LIMITATIONS AND FUTURE WORK

- Gamification layer: extend the trackerSlice state shape and UC-4 with non-competitive streak/milestone fields, informed by the design evidence in [6] and [7].
- Automated dietary logging: introduce an on-device food-recognition step ahead of the existing mealPlanApi.ts write path, in the spirit of [13], while retaining manual override for accuracy.
- Security-design review: commission a structured assessment of the data-access and cloud tiers against the control gaps identified in [10] and [11] — encryption at rest, granular security-rule auditing, and optional multi-factor authentication for administrator accounts.
- Validated screen-time capture: if the screenTimeLimitMin field is extended to automated usage tracking, apply platform-specific bias correction consistent with the Android/iOS divergence reported in [12].
- Multi-tenant administrator tier: extend the three-tier architecture (Fig. 1) with a facility-level reporting role above the caregiver actor, aggregating nutrition and growth trends across all caregivers and students, addressing the engagement/feedback quality gap noted for growth-monitoring apps in [14].
- Growth-screening logic: add population-referenced growth-flagging computation alongside the existing BMI module (UC-7), in the spirit of [5], so abnormal height/weight trajectories are proactively surfaced rather than only point-in-time BMI.

## VII. CONCLUSION

This paper has presented the system design of AthLead, a cross-platform React Native and Firebase-based mobile application for tracking the nutrition, exercise, growth, and screen-time habits of children in shelter-home care. Starting from derived functional and non-functional requirements and an explicit actor/use-case model, the paper documented a three-tier architecture, a representative data-flow sequence, a Firebase Realtime Database schema, module-level component responsibilities, and the software design patterns — Flux/Redux state flow, a repository-style data-access layer, and file-based routing — that structure the implementation. These design decisions were grounded throughout in a structured review of fifteen indexed academic sources spanning mHealth behaviour change, pediatric digital nutrition, screen-time health effects, gamified exercise adherence, layered/BaaS system architecture, and mHealth security. The review indicates that Athlead's design is consistent with current best practice while surfacing concrete, literature-motivated opportunities for its next iteration in gamification, automated logging, and security hardening.

## REFERENCES

[1] M. Milne-Ives, C. Lam, C. De Cock, M. H. Van Velthoven, and E. Meinert, "Mobile Apps for Health Behavior Change in Physical Activity, Diet, Drug and Alcohol Use, and Mental Health: Systematic Review," JMIR mHealth and uHealth, vol. 8, no. 3, e17046, Mar. 2020, doi: 10.2196/17046.

[2] R. Prowse and S. Carsley, "Digital Interventions to Promote Healthy Eating in Children: Umbrella Review," JMIR Pediatrics and Parenting, vol. 4, no. 4, e30160, Nov. 2021, doi: 10.2196/30160.

[3] J. Lee, Z. Su, and Y. Chen, "Mobile Apps for Children's Health and Wellbeing: Design Features and Future Opportunities," AMIA Annual Symposium Proceedings, vol. 2023, pp. 1027-1036, Jan. 2024.

[4] M. Piszka, E. Kwapien, P. Brasse, K. Staszkiewicz, J. Zerdka, K. K. Staszkiewicz, J. Bartkowski, F. Czarnecki, and M. Kubicka, "The Impact of Screen Time on the Health of the Pediatric Population: Short- and Long-Term Consequences for Lifestyle, Ophthalmology, and Mental Health," Cureus, vol. 17, no. 11, e96944, Nov. 2025, doi: 10.7759/cureus.96944.

[5] T. Thaventhiran, J. Orr, J. K. Morris, A. Hsu, L. Martin, K. M. Davies, V. Harding, P. Chapple, L. Dunkel, and H. L. Storr, "A Digital Health Solution for Child Growth Monitoring at Home: Testing the Accuracy of a Novel 'GrowthMonitor' Smartphone Application to Detect Abnormal Height and Body Mass Indices," Mayo Clinic Proceedings: Digital Health, vol. 1, no. 4, pp. 498-509, Dec. 2023, doi: 10.1016/j.mcpdig.2023.08.001.

[6] U. Rehman, A. Z. Abbasi, D. H. Ting, M. Hassan, and N. Khair, "Exploring the Impact of Gamified Experiences on User Engagement in Fitness Apps: A GAMEFULQUEST Perspective," IEEE Transactions on Engineering Management, vol. 71, pp. 3613-3628, 2024, doi: 10.1109/TEM.2023.3299385.

[7] N. H. Korkmaz and H. Çoruh, "Time to Move: A 4-Week Gamified Mobile Application Intervention to Promote Physical Activity in Secondary School Students," European Journal of Sport Science, 2025, doi: 10.1002/ejsc.70027.

[8] P. R. Saraf, S. M. Jadhao, S. J. Wanjari, S. G. Kolwate, and A. D. Patil, "A Review on Firebase (Backend as a Service) for Mobile Application Development," International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 10, no. 1, Jan. 2022, doi: 10.22214/ijraset.2022.39958.

[9] A. Efendi, M. I. Ammarullah, I. G. T. Isa, M. P. Sari, J. N. Izza, Y. S. Nugroho, H. Nasrullah, and D. Alfian, "IoT-Based Elderly Health Monitoring System Using Firebase Cloud Computing," Health Science Reports, vol. 8, no. 3, e70498, Mar. 2025, doi: 10.1002/hsr2.70498.

[10] D. Kotz, C. A. Gunter, S. Kumar, and J. P. Weiner, "Privacy and Security in Mobile Health: A Research Agenda," Computer, vol. 49, no. 6, pp. 22-30, Jun. 2016, doi: 10.1109/MC.2016.185.

[11] L. H. Iwaya, A. Ahmad, and M. A. Babar, "Security and Privacy for mHealth and uHealth Systems: A Systematic Mapping Study," IEEE Access, vol. 8, pp. 150081-150112, 2020, doi: 10.1109/ACCESS.2020.3015962.

[12] P. L. Kristensen, L. G. Olesen, H. K. Egebæk, J. Pedersen, M. G. Rasmussen, and A. Grøntved, "Criterion Validity of a Research-Based Application for Tracking Screen Time on Android and iOS Smartphones and Tablets," Computers in Human Behavior Reports, vol. 5, 100164, Mar. 2022, doi: 10.1016/j.chbr.2021.100164.

[13] A. Nossair and H. El Housni, "Eating Smart: Advancing Health Informatics with the Grounding DINO-Based Dietary Assistant App," International Journal of Scientific Research and Innovative Studies, vol. 3, no. 3, pp. 26-34, Jun. 2024, doi: 10.5281/zenodo.11243881.

[14] A. S. Irawan, A. D. Alristina, R. D. Laili, N. Amalia, A. P. Muharram, A. V. Miranda, B. Döbrössy, and E. Girasek, "Beyond the Interface: Benchmarking Pediatric Mobile Health Applications for Monitoring Child Growth Using the Mobile App Rating Scale," Frontiers in Digital Health, vol. 7, 1621293, Jun. 2025, doi: 10.3389/fdgth.2025.1621293.

[15] P. S. Basto and P. Ferreira, "Mobile Applications, Physical Activity, and Health Promotion," BMC Health Services Research, vol. 25, art. 344, Mar. 2025, doi: 10.1186/s12913-025-12489-z.

## APPENDIX: LINKS TO REFERENCED PAPERS

[1] M. Milne-Ives et al., JMIR mHealth uHealth (2020) — https://mhealth.jmir.org/2020/3/e17046/

[2] R. Prowse and S. Carsley, JMIR Pediatr. Parent. (2021) — https://pediatrics.jmir.org/2021/4/e30160

[3] J. Lee, Z. Su, and Y. Chen, AMIA Annu. Symp. Proc. (2023) — https://pmc.ncbi.nlm.nih.gov/articles/PMC10785842/

[4] M. Piszka et al., Cureus (2025) — https://www.ncbi.nlm.nih.gov/pmc/articles/PMC12706514/

[5] T. Thaventhiran et al., Mayo Clin. Proc. Digit. Health (2023) — https://www.mcpdigitalhealth.org/article/S2949-7612(23)00069-X/fulltext

[6] U. Rehman et al., IEEE Trans. Eng. Manage. (2024) — https://ieeexplore.ieee.org/document/10374284/

[7] N. H. Korkmaz and H. Çoruh, Eur. J. Sport Sci. (2025) — https://www.ncbi.nlm.nih.gov/pmc/articles/PMC12294549/

[8] P. R. Saraf et al., IJRASET (2022) — https://www.ijraset.com/research-paper/firebase-backend-as-a-service-for-mobile-application-development

[9] A. Efendi et al., Health Science Reports (2025) — https://pmc.ncbi.nlm.nih.gov/articles/PMC11873372/

[10] D. Kotz et al., Computer / IEEE (2016) — https://pmc.ncbi.nlm.nih.gov/articles/PMC5362254/

[11] L. H. Iwaya, A. Ahmad, and M. A. Babar, IEEE Access (2020) — https://arxiv.org/pdf/2006.12069

[12] P. L. Kristensen et al., Comput. Hum. Behav. Rep. (2022) — https://www.sciencedirect.com/science/article/pii/S2451958821001123

[13] A. Nossair and H. El Housni, IJSRIS (2024) — https://arxiv.org/pdf/2406.00848

[14] A. S. Irawan et al., Front. Digit. Health (2025) — https://www.frontiersin.org/journals/digital-health/articles/10.3389/fdgth.2025.1621293/full

[15] P. S. Basto and P. Ferreira, BMC Health Serv. Res. (2025) — https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11892298/.