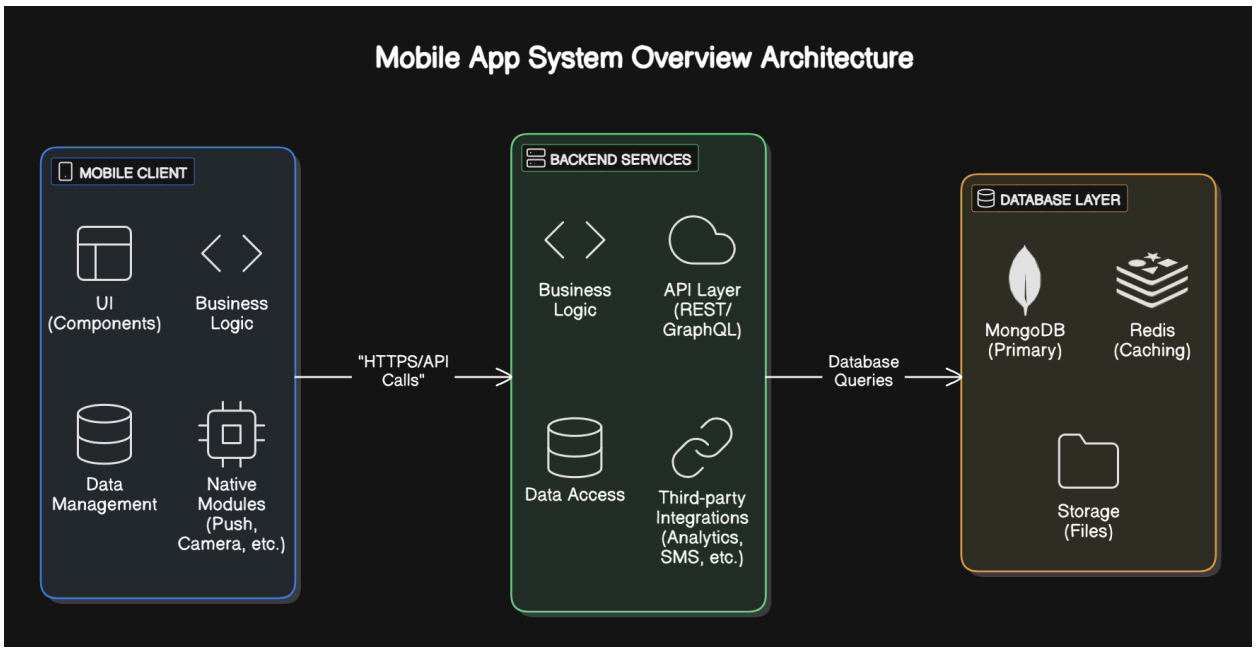
System Requirements Document for "Healthy Start" App

1. Introduction

The "Healthy Start" app is a digital health intervention designed to educate parents of under-five children about proper antibiotic use, home management of minor illnesses, and when to seek medical care. The app aims to combat antimicrobial resistance (AMR) by promoting antibiotic stewardship through behavior change.

2. System Architecture

2.1 High-Level Architecture



2.2 Design Patterns

MVC (Model-View-Controller) Pattern:

- Separates UI (View), business logic (Controller), and data (Model)
- Ideal for React Native (View) and Node.js (Controller/Model) architecture

Repository Pattern:

- Abstracts database operations from business logic
- Makes it easier to switch databases if needed

Strategy Pattern:

- For implementing different notification strategies (push, SMS, email)
- For different content delivery methods (text, audio, video)

Observer Pattern:

- For real-time features like push notifications
- For tracking user engagement metrics

Factory Pattern:

- For creating different types of content (quizzes, symptom checkers, etc.)
- For generating different report formats

3. Technical Stack

3.1 Frontend (React Native)

- Framework: React Native (with TypeScript)
- Navigation: React Navigation
- UI Components: NativeBase or React Native Paper
- State Management: Redux Toolkit or Context API
- Offline Support: Redux Persist + WatermelonDB
- Multimedia: React Native Video for video content
- Audio: React Native Sound for voice-assisted modules
- Localization: i18next for multi-language support (English, Hindi, Nepali)
- Push Notifications: Firebase Cloud Messaging (FCM)
- Analytics: Firebase Analytics or Mixpanel

3.2 Backend (Node.js)
Framework: Express.js or NestJS

- API: RESTful API (with potential GraphQL endpoints for complex queries)
- Authentication: JWT (optional for future features)
- Database:
    - Primary: MongoDB (document store for flexible schema)
    - Cache: Redis (for session management and caching)
- File Storage: AWS S3 or Firebase Storage for multimedia
- Real-time Features: Socket.io (for future live help)
- Background Jobs: Bull (Redis-based queue for notifications)
- Email/SMS: Twilio API (for future SMS features)
- Monitoring: Winston for logging + Prometheus + Grafana

3.3 DevOps

- Version Control: Git (GitHub/GitLab)
- CI/CD: GitHub Actions or GitLab CI
- Containerization: Docker
- Orchestration: Kubernetes (for scale) or Docker Compose (for development)
- Cloud Provider: AWS or GCP (with focus on Indian region for better latency)
- Monitoring: Sentry for error tracking, New Relic for performance

4. Functional Requirements

4.1 Core Features

Risk Assessment Tool

- Interactive quiz with branching logic
- Immediate feedback on whether to seek medical care
- Color-coded results (green=home care, yellow=monitor, red=seek help)

Symptom Checker

- Home remedy suggestions for minor issues
- Clear indicators for when to see a doctor

Myth-Buster Section

- Gamified content (swipe left/right for true/false)
- Simple animations explaining AMR concepts
- Shareable fact cards

Audio/Video Education

- Short, culturally appropriate videos
- Audio transcripts for low-bandwidth situations
- Downloadable content for offline access

Push Notifications

- Scheduled health tips (e.g., "Antibiotics don't work for colds")
- Reminders for medication schedules (if prescribed)
- Seasonal illness alerts

Bookmarking System

- Save favorite tips/remedies
- Personalized "My Health Guide" section
- Share functionality for community education

User

4.2 Admin Features

Content Management

- Update educational materials without app updates
- Schedule notification campaigns
- Manage languages and translations

Analytics Dashboard

- User engagement metrics
- Feature usage statistics
- Pre/post-intervention comparisons

User Feedback System

- Simple thumbs up/down for content
- Optional text feedback
- FAQ builder based on common questions

5. Non-Functional Requirements

Performance

- App launch time < 2 seconds
- API response time < 1 second for 90% of requests
- Support for low-end Android devices

Scalability

- Handle 10,000+ concurrent users
- Database scaling strategies
- CDN for static content

Security

- Data encryption at rest and in transit
- Regular security audits
- Compliance with Indian data protection laws

Accessibility

- WCAG 2.1 AA compliance
- High contrast mode
- Screen reader support

Offline Capability

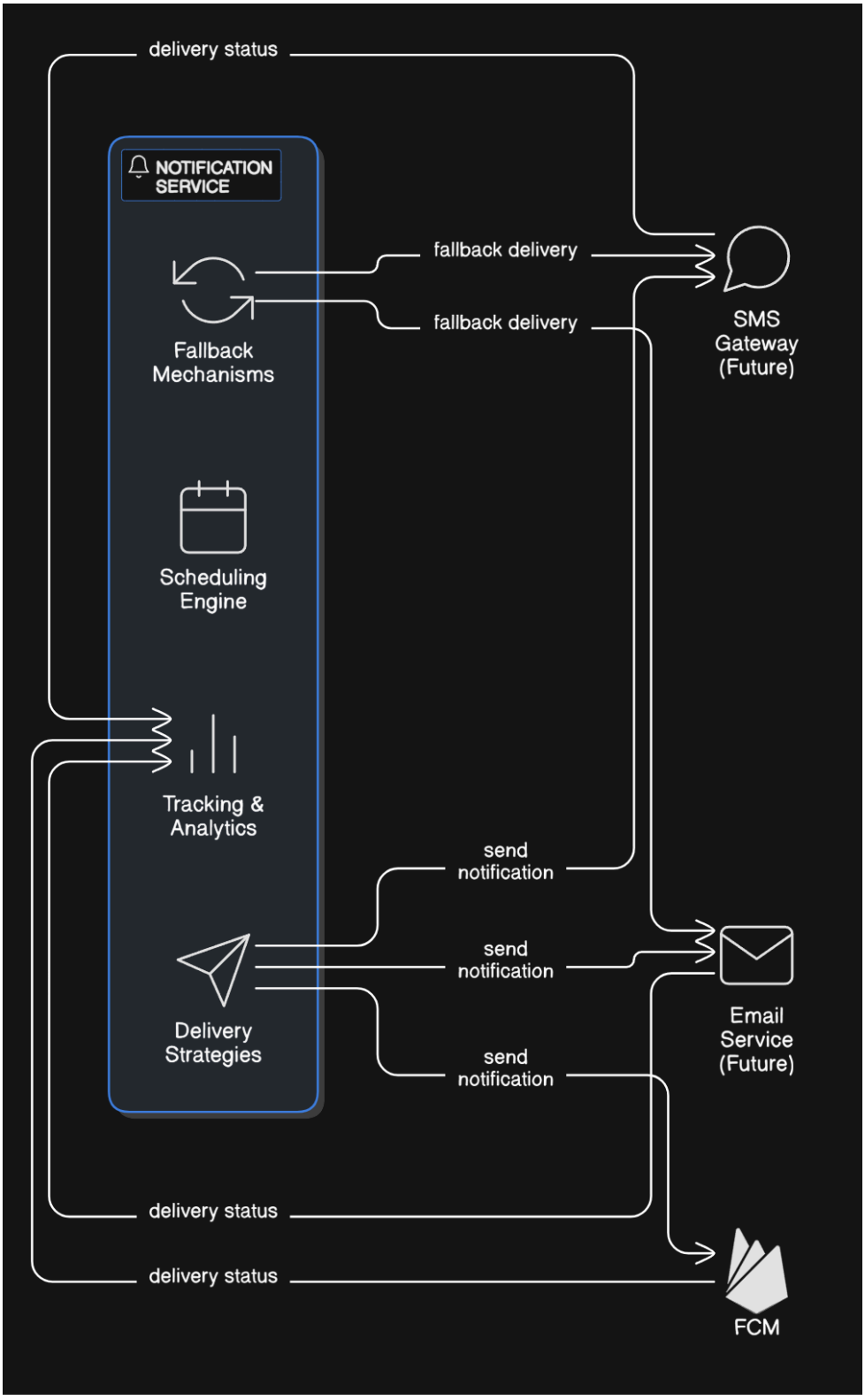
- Core content available offline
- Queue actions when connection is restored
- Sync conflicts resolution

6. Data Model (Key Entities)

```
User {
  _id: ObjectId
  deviceId: String (anonymous users)
  languagePreference: String
  savedItems: [ContentItem]
  usageStats: {
    lastLogin: Date
    quizCompletions: Number
  }
}

ContentItem {
  _id: ObjectId
  type: Enum['video','article','quiz','symptomChecker']
  title: String
  body: String
  mediaUrl: String
  tags: [String]
  translations: [{
    language: String
    title: String
```

```
    body: String
  }]
}

QuizResult {
  _id: ObjectId
  userId: ObjectId
  quizId: ObjectId
  answers: [{
    questionId: ObjectId
    response: Mixed
  }]
  recommendation: String
  timestamp: Date
}

NotificationLog {
  _id: ObjectId
  userId: ObjectId
  type: String
  content: String
  sentAt: Date
  opened: Boolean
}
```

7. Notification System Design



7.2 Implementation Details

Primary Technology: Firebase Cloud Messaging (FCM)

- Reliable push notifications for Android
- Supports rich notifications with images
- Built-in analytics for notification open rates

Notification Types:

- Scheduled Notifications: Daily/weekly health tips
- Trigger-based: After completing a quiz section
- Emergency Alerts: Disease outbreak warnings

Content Strategy:

- Localized content based on user preference
- Timezone-aware delivery
- Frequency capping to avoid spam

Fallback Mechanism:

- If push notifications are disabled, show in-app messages
- For critical alerts, consider SMS fallback (future)

Tracking:

- Open rates tracking
- Conversion tracking (did user take action after notification?)
- A/B testing for message effectiveness

8. Deployment Strategy

8.1 Staging Plan

Phase 1 (Pilot):

- 2 PHCs in Sikkim
- 100-200 users
- Basic features only

Phase 2 (Expansion):

- 5-10 PHCs
- 500-1000 users
- Add advanced features based on feedback

Phase 3 (State-wide):

- All PHCs in Sikkim
- 5000+ users
- Full feature set with optimizations

8.2 Rollout Strategy

- Gradual rollout (5% → 25% → 50% → 100%)
- Feature flags for experimental features
- Canary releases with monitoring

9. Monitoring and Maintenance

- Error Tracking: Sentry for real-time error monitoring
- Performance Monitoring: New Relic for backend, Firebase Performance for mobile
- Usage Analytics: Custom dashboard + Firebase Analytics
- Regular Updates:
    - Monthly content updates
    - Quarterly feature updates
    - Security patches as needed

10. Success Metrics

Engagement Metrics:

- Daily Active Users (DAU)
- Session length
- Feature usage frequency

Behavior Change Metrics:

- Pre/post quiz scores
- Self-reported behavior changes

- Reduction in unnecessary doctor visits

System Metrics:

- App crash rate
- API response times
- Notification open rates

Third-Party Services (Annual) - Android

Service	Provider	Estimated Cost (₹)	Notes
Google Play Developer Account	Google	**₹2,500 (one-time)**	Lifetime fee for Android app publishing
Push Notifications	Firebase (FCM)	**₹0 – ₹5,000**	Free tier (10K messages/month), then pay-per-use
SMS Notifications	MSG91 (India)	**₹6,000 – ₹18,000**	~5K SMS/month (₹1–2/SMS)
Analytics	Firebase Analytics	**₹0**	Free for basic usage
Crash Reporting	Firebase Crashlytics	**₹0**	Free for Android
In-App Updates	Google Play Core	**₹0**	Built into Play Store
Dynamic Links	Firebase	**₹0**	Free for app install tracking
Email Service	SendGrid	**₹0 – ₹12,000**	Free tier (100 emails/day)
Content Delivery Network (CDN)	Cloudflare	**₹0 – ₹6,000**	Free tier + optional paid plans
Security (SSL, Monitoring)	Let's Encrypt + Grafana	**₹0**	Open-source alternatives
Localization (Hindi/Nepali)	Freelancers (Upwork)	**₹15,000 – ₹30,000**	One-time translation cost

Total Estimated Annual Cost (Android-Only)

- Low-End (Free tiers + minimal SMS): ₹8,500 – ₹20,000
- Mid-Range (Scaled usage): ₹25,000 – ₹50,000
- High-End (Premium SMS + heavy traffic): ₹60,000+