Sprint 7-8: Domain Analysis Documentation

SyncInCorp Crèche App Development

Team: SyncInCorp
Project: Crèche Management Application
Sprint: 3 - Domain Analysis
Sprint Duration:

Sprint Goal

Complete domain analysis, finish user journey diagrams, agree on architecture and technology choices with documentation, and build mockups for client presentation.

Sprint Objectives

1. **Domain Analysis:** Understand the crèche management industry and business processes
2. **User Journey Completion:** Create comprehensive user journey diagrams
3. **Architecture & Technology Agreement:** Finish and document technology stack decisions
4. **Mockup Development:** Build visual representations of the application interface
5. **Client Presentation:** Present mockups and gather feedback for validation

---

Team Availability

Sprint 3 Team Members

Team Member	Role	Availability	Sprint Focus Areas
Minenhle Dladla]	Project Manager	100%	Client liaison, sprint coordination
Minenhle Dladla and Shaldon Sindraj	UX Designer	85%	User journey mapping, mockup creation
Msizi Lamula]	Software Architect	100	Architecture design, technology selection
Amahle Gcumisa	Frontend Developer	60%	UI mockups, frontend architecture

Msizi Lamula and Amahle Gcumisa	Backend Developer	100%	Database design, API architecture
Msizi Lamula and Amhle Gcumisa	DevOps Engineer	100%	Infrastructure planning, deployment strategy
Nosipho kubheka nd Darren Danashar	Documentation	60%	Take notes, keep track of meetings and update documentation as app progresses.

---

1. Domain Analysis

1.1 Industry Overview: Crèche Management Sector

Market Context

- **Industry:** Early Childhood Care and Education
- **Target Market:** Private and institutional crèches/daycare centers
- **Market Size:** Growing demand for digital solutions in childcare management
- **Key Trends:** Digital transformation, parent engagement, regulatory compliance

Business Domain Understanding

Core Business Processes:

1. **Enrollment Management**
    - Application processing
    - Capacity planning
    - Waiting list management
    - Fee structure administration
2. **Daily Operations**
    - Attendance tracking
    - Activity planning and documentation
    - Meal planning and dietary management
    - Health and safety monitoring
3. **Communication Management**
    - Parent-staff communication
    - Emergency notifications
    - Progress reporting
    - Event announcements
4. **Administrative Functions**
    - Staff management
    - Financial reporting
    - Regulatory compliance
    - Quality assurance

1.2 Stakeholder Analysis

Primary Stakeholders

Stakeholder	Role	Key Interests	System Interaction Level
Parents/Guardians	End Users	Child safety, development tracking, communication	High - Daily usage
Crèche Staff	End Users	Efficient documentation, parent communication	High - Daily usage
Crèche Administrators	End Users	Operational oversight, reporting, compliance	Medium - Weekly usage
Children	Indirect Users	Safe, nurturing environment	Indirect - through documentation

Secondary Stakeholders

- **Regulatory Bodies:** Compliance monitoring and reporting
- **IT Support:** System maintenance and technical support
- **Management Company:** ROI and operational efficiency
- **Emergency Services:** Access to emergency contact information

1.3 Existing Systems Analysis

Current State Assessment

- **Manual Processes:** Paper-based attendance, handwritten daily reports
- **Communication:** Phone calls, WhatsApp groups, physical notice boards
- **Documentation:** Physical files, spreadsheets for basic tracking
- **Pain Points:** Information silos, delayed communication, inefficient reporting

Integration Requirements

- **Financial Systems:** Fee management and billing integration
- **HR Systems:** Staff management and scheduling
- **Regulatory Systems:** Compliance reporting automation
- **Emergency Services:** Quick access protocols

1.4 Regulatory and Compliance Requirements

Data Protection

- **POPIA Compliance** (South African context)
- **Child Data Protection:** Enhanced security for minors' information
- **GDPR Considerations:** International data handling standards

Health and Safety Regulations

- **Incident Reporting:** Mandatory documentation requirements
- **Medical Information:** Secure handling of health records
- **Emergency Procedures:** Quick access to emergency contacts and procedures

---

2. Complete User Journey Diagrams

2.1 Parent User Journey: Morning Drop-off to Evening Pickup

```
journey
    title Parent Daily Journey
    section Morning Preparation
      Check app for updates: 5: Parent
      Review yesterday's activities: 4: Parent
      Add special instructions: 3: Parent
    section Drop-off
      Confirm attendance via app: 5: Parent, Staff
      Receive confirmation notification: 5: Parent
    section During Day
      Receive activity updates: 5: Parent
      View photos/videos: 5: Parent
      Respond to staff messages: 4: Parent
    section Pick-up
      Get end-of-day summary: 5: Parent
      Review daily report: 4: Parent
      Confirm pickup: 5: Parent, Staff
```

2.2 Staff User Journey: Daily Care Documentation

```
journey
    title Staff Daily Journey
    section Morning Setup
      Login to system: 5: Staff
      Review children's special needs: 4: Staff
      Mark attendance: 5: Staff
    section Activity Management
      Document activities: 4: Staff
      Take photos/videos: 5: Staff
      Record mealtimes: 4: Staff
      Log nap schedules: 3: Staff
    section Parent Communication
      Send activity updates: 4: Staff
      Respond to parent queries: 4: Staff
      Handle emergencies: 5: Staff
    section End of Day
      Complete daily summaries: 3: Staff
      Prepare pickup notifications: 4: Staff
      Update child profiles: 3: Staff
```

2.3 Administrator User Journey: Weekly Management Overview

```
journey
    title Administrator Weekly Journey
    section Planning
      Review enrolment status: 4: Admin
      Check staff schedules: 3: Admin
      Monitor capacity: 4: Admin
    section Monitoring
      Review activity reports: 3: Admin
      Check parent feedback: 4: Admin
```

```
      Monitor compliance: 5: Admin
    section Management
      Generate reports: 3: Admin
      Handle escalations: 4: Admin
      Update system settings: 2: Admin
```

Resources Used:

-
- (Do mapping on canva and share link)

---

3. Architecture and Technology Choices

3.1 System Architecture Decision

Chosen Architecture: Micro-services with API Gateway

Rationale:

- **Scalability:** Independent scaling of different services
- **Maintainability:** Separate development and deployment cycles
- **Technology Flexibility:** Different services can use optimal technologies
- **Fault Isolation:** Issues in one service don't affect others

Architecture Diagram

```
graph TB
    subgraph "Client Layer"
        A[Web App]
        B[Mobile App]
        C[Admin Portal]
    end

    subgraph "API Gateway"
        D[Load Balancer]
        E[Authentication Service]
        F[Rate Limiting]
    end

    subgraph "Microservices"
        G[User Management Service]
        H[Child Management Service]
        I[Activity Service]
        J[Communication Service]
        K[Notification Service]
        L[File Storage Service]
    end

    subgraph "Data Layer"
        M[User Database]
        N[Activity Database]
        O[File Storage]
        P[Cache Layer]
```

```
end

A --> D
B --> D
C --> D
D --> E
D --> F
E --> G
E --> H
E --> I
E --> J
E --> K
E --> L
G --> M
H --> M
I --> N
J --> N
K --> P
L --> O
```

3.2 Technology Stack Selection

Frontend Technologies

Component	Chosen Technology	Rationale
Web Framework	React.js 18+	Large community, component reusability, PWA capabilities
Mobile App	React Native	Code sharing with web, faster development
State Management	Redux Toolkit	Predictable state management, DevTools integration
UI Framework	Material-UI (MUI)	Professional look, accessibility features
Build Tool	Vite	Fast development builds, modern tooling

Backend Technologies

Component	Chosen Technology	Rationale
Runtime	Node.js 18+	JavaScript ecosystem consistency, fast I/O
Framework	Express.js + NestJS	Express simplicity + NestJS structure for larger services
Database	PostgreSQL	ACID compliance, relational data integrity
Cache	Redis	Fast session storage, pub/sub for real-time features
File Storage	AWS S3 / Cloudinary	Secure image/video storage, CDN integration
Authentication	JWT + Refresh Tokens	Stateless authentication, mobile-friendly

DevOps and Infrastructure

Component	Chosen Technology	Rationale

Container Platform	Docker	Environment consistency, easy deployment
Orchestration	Docker Compose (dev) / Kubernetes (prod)	Scalability and orchestration
CI/CD	GitHub Actions	Integrated with repository, free for public repos
Cloud Provider	AWS / Google Cloud	Reliability, global reach, managed services
Monitoring	Sentry + CloudWatch	Error tracking and performance monitoring

3.3 Database Design

Entity Relationship Overview

```
erDiagram
    Users ||--o{ Children : "has"
    Users ||--o{ Messages : "sends"
    Children ||--o{ Activities : "participates"
    Children ||--o{ Attendance : "marked"
    Users ||--o{ Staff_Assignments : "assigned"
    Activities ||--o{ Activity_Media : "contains"

    Users {
        uuid id PK
        string email UK
        string password_hash
        enum role
        jsonb profile_data
        timestamp created_at
        timestamp updated_at
    }

    Children {
        uuid id PK
        uuid parent_id FK
        string first_name
        string last_name
        date date_of_birth
        jsonb medical_info
        jsonb emergency_contacts
        timestamp created_at
        timestamp updated_at
    }

    Activities {
        uuid id PK
        uuid child_id FK
        uuid staff_id FK
        string activity_type
        text description
        timestamp activity_time
        jsonb metadata
    }
```

3.4 Security Architecture

Authentication Flow

```
sequenceDiagram
    participant C as Client
    participant AG as API Gateway
    participant AS as Auth Service
    participant MS as Micro-service
    participant DB as Database

    C->>AG: Login Request
    AG->>AS: Confirm Credentials
    AS->>DB: Check User
    DB-->>AS: User Data
    AS-->>AG: JWT + Refresh Token
    AG-->>C: Authentication Response

    C->>AG: API Request + JWT
    AG->>AS: Validate JWT
    AS-->>AG: Token Valid
    AG->>MS: Forward Request
    MS-->>AG: Response
    AG-->>C: API Response
```

Data Protection Measures

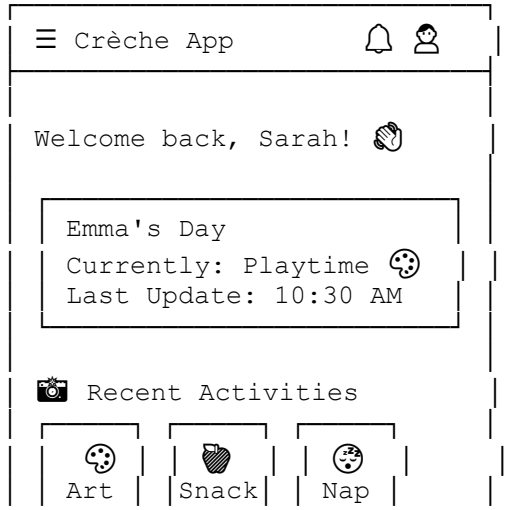
- **Encryption at Rest:** AES-256 for database
- **Encryption in Transit:** TLS 1.3 for all communications
- **PII Protection:** Field-level encryption for sensitive data
- **Access Control:** Role-based permissions with principle of least privilege
- **Audit Logging:** Comprehensive activity logging for compliance

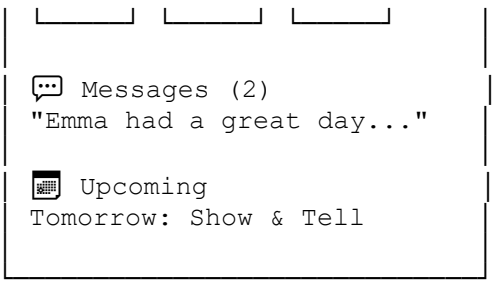
---

4. Mock-up Development

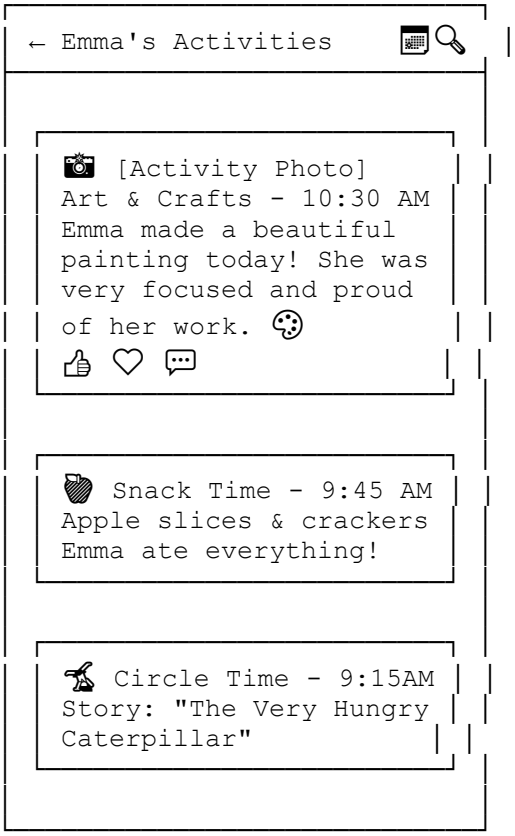
4.1 Mobile App Mockups

Parent Dashboard



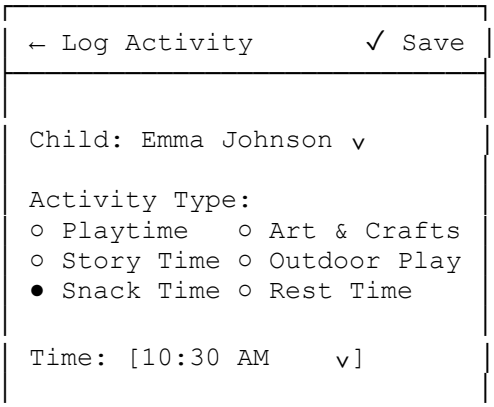


Activity Feed



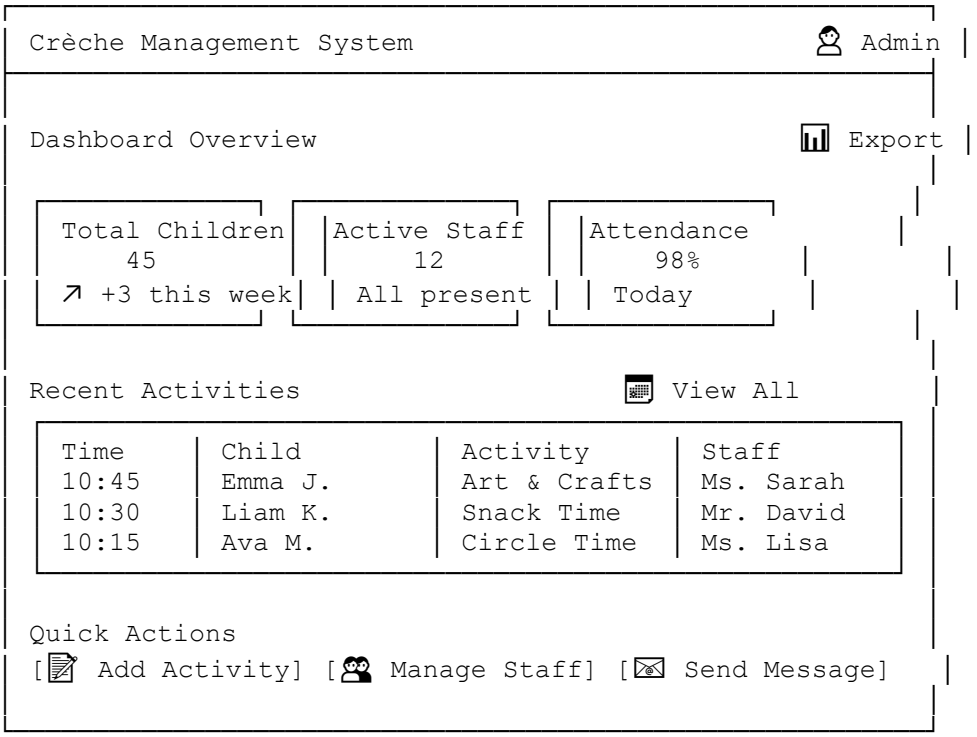
4.2 Staff Interface Mockups

Activity Logger



4.3 Web Admin Portal Mockups

Dashboard Overview



4.4 Responsive Design Considerations

Breakpoint Strategy

- **Mobile:** 320px - 768px (Primary focus)
- **Tablet:** 768px - 1024px
- **Desktop:** 1024px+ (Admin portal focus)

Accessibility Features

- **Color Contrast:** WCAG 2.1 AA compliance
- **Font Sizes:** Minimum 16px for mobile
- **Touch Targets:** Minimum 44px tap targets
- **Screen Reader Support:** Semantic HTML and ARIA labels
- **Keyboard Navigation:** Full keyboard accessibility

---

5. Client Presentation Plan

5.1 Presentation Structure

Agenda (60 minutes)

1. **Welcome & Recap** (5 minutes)
    - Sprint 2 achievements summary
    - Today's objectives
2. **Domain Analysis Findings** (15 minutes)
    - Industry insights
    - Stakeholder analysis
    - Current state vs future state
3. **User Journey Walkthrough** (15 minutes)
    - Parent journey demonstration
    - Staff workflow explanation
    - Administrator overview
4. **Technical Architecture** (10 minutes)
    - High-level architecture overview
    - Technology choices rationale
    - Security and scalability considerations
5. **Mockup Demonstration** (10 minutes)
    - Mobile app walkthrough
    - Web portal demonstration
    - Responsive design showcase
6. **Feedback & Q&A** (15 minutes)
    - Client feedback collection
    - Feature prioritisation discussion
    - Next steps alignment

5.2 Presentation Materials

Required Deliverables

- [ ] Domain analysis document
- [ ] Complete user journey diagrams
- [ ] Architecture documentation
- [ ] Interactive mockups (Figma)
- [ ] Technology stack justification
- [ ] Security and compliance overview
- [ ] Project timeline update

- [ ] Sprint 4 planning preview

5.3 Success Criteria

Client Approval Metrics

- **Domain Understanding:** Client confirms accurate business process mapping
- **User Experience:** Positive feedback on user journey flows
- **Technical Approach:** Approval of architecture and technology choices
- **Visual Design:** Acceptance of mockup designs and user interface
- **Project Direction:** Alignment on scope and timeline for development phases

Feedback Collection Methods

- **Digital Survey:** Post-presentation satisfaction survey
- **Priority Matrix:** Feature importance ranking exercise
- **Change Requests:** Documented modification requests
- **Approval Sign-off:** Formal approval for Sprint 4 commencement

---

Sprint 3 Deliverables Checklist

Completed Deliverables

- [ ] Comprehensive domain analysis report
- [ ] Stakeholder mapping and analysis
- [ ] Complete user journey diagrams for all user types
- [ ] System architecture documentation
- [ ] Technology stack selection with rationale
- [ ] Database design and ERD
- [ ] Security architecture plan
- [ ] Mobile app mockups (iOS/Android)
- [ ] Web admin portal mockups
- [ ] Responsive design specifications
- [ ] Client presentation materials
- [ ] Sprint 4 backlog preparation

Quality Assurance

- [ ] Peer review of all deliverables
- [ ] Client stakeholder validation
- [ ] Technical feasibility confirmation
- [ ] Accessibility compliance check
- [ ] Security review completion

Next Steps: Sprint 8-9 Preview

Anticipated Sprint 4 Focus Areas

1. **Development Environment Setup**
    - CI/CD pipeline implementation
    - Development and staging environments
    - Code quality standards establishment
2. **MVP Development Kickoff**
    - Core authentication system
    - Basic user management
    - Foundational database structure
3. **UI/UX Refinement**
    - Detailed wireframes based on client feedback
    - Design system establishment
    - Component library development

Risk Mitigation Planning

- **Technical Risks:** Proof of concept development for complex features
- **Resource Risks:** Team capacity planning and skill gap analysis
- **Timeline Risks:** Buffer time allocation for critical path items
- **Client Risks:** Regular feedback loops and change management processes

---

References: