**SyncInCorp Crèche Management Mobile Application**

**Sprint 2: Conceptualize and Plan**

**Project:** Crèche Management Mobile Application  
**Team:** SyncInCorp  
**Module:** INSY7315 - Work Integrated Learning  
**Institution:** Varsity College  
**Facilitator:** Mr D S Govender  
**Period:** March 31, 2025 - October 31, 2025

### **Session 3-4( SPRINT 2)**

**By Nosipho kubheka and Minenhle Dladla**

## **1. Sprint Objective**

Sprint 2 focuses on **eliciting, organizing, and prioritizing requirements** for the Crèche Management Mobile App. This includes defining **functional** and **non-functional** requirements, developing **user stories** for backlog creation, and designing **entity structures** for system components such as users, children, events, and notifications.

(Atlassian, 2024; ProjectManager.com, 2025)

**2. Sprint Backlog Items (User Stories)**

| **No.** | **User Story** | **Responsibility** | **Acceptance Criteria** |
| --- | --- | --- | --- |
| 1 | As a developer, I need a **prioritized backlog of requirements from the client** so that I can develop an application that meets their needs. | Requirements Responsibility | Requirements are collected, validated, and approved by the client. |
| 2 | As a developer, I need to understand how the **screens and user roles** will work together, so that I can build a UI that meets the client’s needs. | UX Responsibility | A first **user journey and flow diagram** is created and reviewed by stakeholders. |
| 3 | As a frontend developer, I need to know which **frameworks and standards** to use so that I can develop high-quality, consistent code. | Frontend Responsibility | Technology stack finalized and coding standards documented. |
| 4 | As a DevOps member, I need to understand how **GitHub Actions** work so that I can build a CI/CD pipeline to ensure code quality. | DevOps Responsibility | GitHub Actions tested and integrated for code commits. |
| 5 | As a backend developer, I need to know which **SDK versions and libraries** to use so that my environment matches the rest of my team. | Backend Responsibility | Local environments configured with consistent versions across all developers. |

(Adapted from ProjectManager.com, 2025; Govender, 2025)

3. Functional Requirements

| **Module** | **Requirement Description** | **Priority** |
| --- | --- | --- |
| **1. User Management** | Admin, Staff, and Parents can create and manage profiles with secure login (email/OTP/password). | High |
| **2. Parent & Child Registration** | Admin captures parent and child details and assigns children to teachers. | High |
| **3. Attendance Tracking (Wow Factor)** | Teachers/Admin mark attendance; parents receive real-time notifications. | High |
| **4. Event Notifications & Calendar Integration** | Admins create events; parents receive notifications and can sync to phone calendars. | Medium |
| **5. Payment Gateway** | Secure payment for tuition/meals; downloadable receipts and reminders. | High |
| **6. Messaging & Communication** | Admin sends announcements; parents receive urgent alerts. | Medium |
| **7. Media Sharing (Wow Factor)** | Teachers upload photos/videos; parents can view/download with consent. | Medium |
| **8. Lunch Ordering System (Wow Factor)** | Parents pre-order meals; view meal plans; make payments. | Low |

(SyncInCorp, 2025)

4. Non-Functional Requirements

| **Category** | **Description** |
| --- | --- |
| **Performance** | The system must handle up to 500 concurrent users without performance degradation. |
| **Security** | All sensitive data must be encrypted using AES-256; authentication through OTP or password. |
| **Usability** | Interface should be intuitive, mobile-responsive, and accessible for non-technical users. |
| **Availability** | The system should achieve 99% uptime through reliable cloud hosting. |
| **Scalability** | Architecture should allow easy addition of new modules. |
| **Maintainability** | Code should follow modular principles with documentation for all major functions. |
| **Compliance** | Must comply with POPIA (Protection of Personal Information Act) standards. |

(Atlassian, 2024; SyncInCorp, 2025)

5. Entity Schema (Data Model)

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| UserID | PK, UUID | Unique identifier |
| FirstName | String | Required |
| LastName | String | Required |
| Email | String | Unique; used for login |
| PhoneNumber | String | Optional / OTP |
| PasswordHash | String | Stored securely |
| Role | Enum (Admin, Staff, Parent) | Defines access level |
| CreatedAt | DateTime | Auto-generated |
| IsActive | Boolean | For soft deletion |

Children

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| ChildID | PK, UUID | Unique identifier |
| FirstName | String | Required |
| LastName | String | Required |
| DOB | Date | Date of Birth |
| Allergies | Text | Optional |
| MedicalInfo | Text | Optional |
| ParentID | FK → Users | Parent (Role=Parent) |
| AssignedStaffID | FK → Users | Assigned Teacher (Role=Staff) |
| IsActive | Boolean | Child’s enrollment status |

EmergencyContacts

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| ContactID | PK, UUID | Unique identifier |
| ChildID | FK → Children | Linked child |
| Name | String | Required |
| Phone | String | Required |
| Relationship | String | e.g., Aunt, Grandfather |

Events

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| EventID | PK, UUID | Unique identifier |
| Title | String | Event name |
| Description | Text | Details |
| EventDateTime | DateTime | Event time |
| CreatedByID | FK → Users | Admin user only |
| CreatedAt | DateTime | Timestamp |

EventNotifications

| **.Field** | **Type** | **Notes** |
| --- | --- | --- |
| NotificationID | PK, UUID | Unique identifier |
| EventID | FK → Events | Associated event |
| RecipientID | FK → Users | Notification recipient |
| SentAt | DateTime | Timestamp |
| Status | Enum (Sent, Delivered, Read) | Notification status |

(Govender, 2025)

## **8. Summary**

Sprint 2 successfully establishes the foundation for development through structured requirements gathering, prioritization, and documentation. Functional and non-functional requirements ensure alignment with client expectations and system quality standards

(Govender, 2025; Atlassian, 2024)

# **References**

Atlassian. (2024). *Agile Software Development: Definition of Ready and Definition of Done.* Retrieved from https://www.atlassian.com/agile

Govender, D. S. (2025). *INSY7315 Work Integrated Learning Guidelines.* Varsity College.

ProjectManager.com. (2025). *Agile Sprint Planning and Project Charter Templates.* Retrieved from [https://www.projectmanager.com/templates/project-charter-template](https://www.projectmanager.com/templates/project-charter-template?utm_source=chatgpt.com)

SyncInCorp. (2025). *Internal Requirement Documentation and Meeting Notes.* Varsity College, INSY7315.