Software Requirements Specification (SRS)

Project: YZone Module (NiFo Platform)

---

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a complete and detailed description of the functional and non-functional requirements of the **YZone module**, which operates as part of the **NiFo platform**. The document is intended for developers, testers, project managers, stakeholders, and academic evaluators.

1.2 Scope

YZone is a cohort-based experiential learning and execution module designed to manage student programs, industry-aligned activities, assessments, and engagement tracking. It enables structured onboarding, learning delivery, progress tracking, reviews, and certification under the supervision of facilitators and administrators.

1.3 Definitions, Acronyms, and Abbreviations

- **NiFo** – Parent platform hosting multiple modules
- **YZone** – Program execution module
- **TYN** – The Yellow Network
- **Tenant** – Organization or institution using the system
- **Cohort** – Group of enrolled students
- **Facilitator** – Program supervisor
- **TYN Executive** – Platform administrator

1.4 References

- YZone Business Requirements Document (BRD)
- NiFo Platform Architecture Document
- IEEE 830 SRS Standard

---

2. Overall Description

2.1 Product Perspective

YZone is a sub-module of the NiFo platform and relies on shared authentication, role-based access control, and infrastructure services. It follows a multi-tenant architecture with strict data isolation.

2.2 Product Functions

- Tenant onboarding and MoU management
- Student application and selection
- Cohort and team management
- Curriculum and session scheduling
- Attendance and daily activity tracking
- Reviews, assessments, and scoring
- Badge allocation and leaderboard generation
- Certification and reporting

2.3 User Classes and Characteristics

User Class	Description
TYN Executive	System administrator
Facilitator	Manages cohorts and reviews
Faculty / Principal	Institutional observer
Industry Mentor	External reviewer
Student	Program participant

2.4 Operating Environment

- Frontend: React.js
- Backend: Node.js (TypeScript)
- Database: PostgreSQL (Azure)
- Storage: Azure Blob Storage
- Notifications: WhatsApp Business API

2.5 Design and Implementation Constraints

- Tenant-level data isolation
- Role-based access enforcement
- WhatsApp-only notifications (v1)
- Timezone locked to IST

3. Functional Requirements

3.1 Tenant & MoU Management

- **FR-TEN-01**: The system shall allow TYN Executives to onboard tenants.
- **FR-TEN-02**: The system shall store MoU validity and constraints per tenant.

3.2 Student Application & Selection

- **FR-APP-01**: The system shall allow students to submit applications.
- **FR-APP-02**: The system shall support approval, rejection, and waitlisting.

3.3 Cohort & Team Management

- **FR-COH-01**: The system shall allow facilitators to create cohorts.
- **FR-COH-02**: The system shall assign students to teams.

3.4 Curriculum & Sessions

- **FR-CUR-01**: The system shall define learning modules.
- **FR-CUR-02**: The system shall schedule sessions per cohort.

3.5 Attendance Tracking

- **FR-ATT-01**: Only facilitators shall mark attendance.
- **FR-ATT-02**: Attendance records shall be immutable after 48 hours.

3.6 Daily Tracker & Weekly Reflection

- **FR-TRK-01**: Students shall submit daily progress updates.
- **FR-TRK-02**: The system shall enforce one submission per day.

3.7 Reviews & Assessments

- **FR-REV-01**: Facilitators shall open review windows.
- **FR-REV-02**: Review scores shall be locked after submission.

3.8 Badges & Leaderboard

- **FR-BAD-01**: The system shall award badges based on performance.
- **FR-BAD-02**: The system shall generate leaderboards per cohort.

3.9 Notifications

- **FR-NOT-01**: The system shall send WhatsApp notifications for key events.

3.10 Certification

- **FR-CER-01**: The system shall generate completion certificates.

---

4. External Interface Requirements

4.1 User Interfaces

- Student Dashboard
- Facilitator Dashboard
- Executive Dashboard

4.2 Software Interfaces

- WhatsApp Business API
- Azure Services

4.3 API Interfaces

- REST-based JSON APIs

5. Data Requirements

- Student, Cohort, Attendance, Tracker, Review, Badge, Certificate entities
- Relational integrity and unique constraints

6. Non-Functional Requirements

6.1 Performance

- System shall support 10,000 concurrent users.

6.2 Security

- OAuth-based authentication
- Encrypted data storage

6.3 Reliability

- 99.5% uptime annually

6.4 Scalability

- Horizontal scaling supported

6.5 Compliance

- GDPR-aligned data handling

7. Assumptions and Dependencies

- Availability of WhatsApp API
- Stability of NiFo core services

8. Acceptance Criteria

- Successful completion of MVP features
- All functional requirements validated

9. Risks

- API downtime
- High user concurrency

10. Appendices

- API contracts
- Database schema
- Notification templates