AI-based Drop-out Prediction and Counseling System Software Requirements Specification (SRS)

Prepared By: [Your Team Name] Date: 07-Sep-2025 Version: 1.0

---

1. Introduction

1.1 Purpose The purpose of this system is to provide educational institutions with a low-cost, easy-to-configure platform that consolidates student attendance, assessment scores, and fee-payment data to predict at-risk students, visualize risk levels, and notify mentors or guardians. The system helps reduce dropout rates by enabling early intervention.

1.2 Scope The system will: - Automatically ingest CSV/Excel files containing attendance, assessment, and fee/payment data. - Merge and process student-level data. - Compute risk scores using rule-based thresholds and optionally simple explainable ML models. - Visualize risk levels in an intuitive dashboard (color-coded: green/yellow/red). - Send automated notifications to mentors/guardians based on risk scores. - Allow configuration of thresholds, weights, and notification schedules.

Out of Scope: - Predictive analytics requiring complex deep learning models. - Integration with external LMS platforms (initial MVP). - Automated intervention execution (system only alerts humans).

1.3 Definitions, Acronyms, and Abbreviations - SRS: Software Requirements Specification - CSV: Comma-Separated Values - ML: Machine Learning - KPI: Key Performance Indicator

---

1. Overall Description

2.1 Product Perspective Standalone web-based dashboard with backend services for data ingestion, processing, scoring, and notifications. Can be hosted on local server, VM, or cloud.

2.2 Product Functions 1. Data Ingestion: Upload multiple spreadsheets (attendance, assessments, fees). 2. Data Processing: Clean, standardize, and merge student data. 3. Risk Scoring: Compute risk scores using configurable rule-based weights; optional ML model. 4. Dashboard Visualization: Show risk levels per student, with charts for trends and explanations. 5. Notifications: Automated alerts via email/SMS/WhatsApp to mentors/guardians. 6. Configuration: Admin interface for thresholds, weights, and notification scheduling.

2.3 User Classes and Characteristics - Mentor: Reviews flagged students, takes action (Low–medium technical skill) - Admin: Configures thresholds, weights, notifications (Medium technical skill) - Guardian: Receives alerts about their child (Low technical skill)

2.4 Operating Environment - Web browser (Chrome/Edge/Firefox) - Backend: Python (Flask/FastAPI) + SQLite/PostgreSQL - Server: Local VM or cloud host

2.5 Design Constraints - Supports Excel/CSV input. - Risk calculations must be explainable and auditable. - Minimal setup required (no paid analytics platforms).

1. Specific Requirements

3.1 Functional Requirements | ID | Requirement | Description | Priority | |----|------------|------------|----------| | FR1 | Data Upload | Upload attendance, assessment, and fee/payment files | High | | FR2 | Data Cleaning & Merging | Standardize columns, handle missing data, merge student records | High | | FR3 | Risk Scoring | Apply rule-based thresholds; optional ML | High | | FR4 | Risk Visualization | Dashboard displays color-coded risk tiers | High | | FR5 | Notification | Send weekly notifications to mentors/guardians | High | | FR6 | Configuration | Admin can set thresholds, weights, notification schedules | Medium | | FR7 | Reporting | Export at-risk student list to CSV/PDF | Medium |

3.2 Non-Functional Requirements - Performance: Process up to 10,000 student records within 5 minutes. - Usability: Dashboard must be intuitive; no technical knowledge required. - Security: Protect sensitive student information; only authorized users can access. - Reliability: Scheduled notifications run reliably; downtime <2%. - Portability: Deployable on Windows/Linux server or Docker container.

3.3 System Interfaces - Input: CSV/Excel spreadsheets - Output: Dashboard, email/SMS notifications, downloadable CSV/PDF reports - APIs: Optional REST APIs for data ingestion and notification

---

1. External Interface Requirements

4.1 User Interfaces - Dashboard Overview: Total students, risk tiers, trend charts - Student Detail: Attendance trend, test scores, payment status, risk explanation - Admin Panel: Configure thresholds, weights, notification schedule

4.2 Hardware Interfaces - Standard PC/Server with network access

4.3 Software Interfaces - SMTP server (email notifications) - Optional SMS/WhatsApp APIs (Twilio/MSG91)

4.4 Communications Interfaces - HTTPS/HTTP for web dashboard

---

1. System Features
2. At-risk student detection: Combines multiple signals
3. Trend analysis: Shows changes in attendance, scores, and fee payments over time
4. Explainable risk scoring: Shows top reasons for each flagged student
5. Alerts & notifications: Mentors and guardians get timely updates
6. Configurable system: Admins adjust thresholds, weights, schedules

---

1. Performance Requirements
2. Dashboard load time <5 seconds for 10,000 students
3. Data upload and processing <5 minutes for 10,000 records
4. Notification delivery success rate ≥95%

1. Other Requirements

7.1 Security - User authentication and authorization - Encrypt sensitive data at rest and in transit

7.2 Legal & Regulatory - Compliance with local student data privacy laws (GDPR/India data protection rules)

7.3 Maintainability - Modular code structure to allow adding new risk factors or ML models

---

1. Appendices
2. Sample CSV schemas for attendance, assessment, fees
3. Example rule-based scoring algorithm
4. References: IEEE SRS standards, hackathon guidelines