EduReg: A Centralized Student Registration System

Phase 2: Requirements Engineering  
By:

Olira Tesgera (1093915)

Yisak Demelash (1093891)

Instructor: Dr. Murad Al-Rajab

Abu Dhabi, United Arab Emirates

October 19, 2025

Contents

[1. System Context Diagram 3](#_Toc210091585)

[2. Requirements Engineering 3](#_Toc210091586)

[3. System Users 4](#_Toc210091587)

[4. Functional Requirements 5](#_Toc210091588)

[5. Non-Functional Requirements 8](#_Toc210091589)

[6. Use Case Modeling 8](#_Toc210091590)

[7. Requirements Specifications 12](#_Toc210091591)

[8. Traceability Matrix 14](#_Toc210091592)

# System Context Diagram

The System Context Diagram provides a high-level view of the EduReg system, illustrating its boundaries, the external entities (actors) that interact with it, and the data flows between them.

**Description:** The **EduReg System** is at the center of the process. It interacts with the following external actors:

* **Administrator (Director/Registrar):** Manages the entire system. They initiate all user registrations, manage permissions, set policies, approve requests, create schedules, and generate reports.
* **Student/Parent:** Completes registration, submits assignments, views grades and attendance, and requests leave or profile changes.
* **Teacher/Staff:** Completes registration, manages assignments and grades, takes attendance, requests resources or leave, and may approve applications if delegated.
* **Finance Department:** This department is an external entity that uses the reports generated by the EduReg system to reconcile their internal financial records.

# 2. Requirements Engineering

To collect the requirements for the EduReg system, a combination of elicitation techniques was chosen to effectively gather information from the diverse group of stakeholders.

**Elicitation Techniques Used:**

1. **Interviews:**
   * **Target Audience:** School Administrators (Director, Registrar) and senior Teachers.
   * **Justification:** Direct interviews are essential for understanding the complex workflows for annual registration, grading policies, attendance, leave management, and the rules for delegating permissions.
   * **Outcomes:** A detailed understanding of all administrative processes was achieved, leading to a comprehensive set of functional requirements for all system modules.
2. **Questionnaires/Surveys:**
   * **Target Audience:** Parents, Students, and general Staff.
   * **Justification:** With a large user base, questionnaires provide a scalable way to gather data on their needs, such as the clarity of the assignment submission process and the usability of the grade viewing feature.
   * **Outcomes:** The surveys highlighted the need for a simple, mobile-friendly user interface, clear status indicators for all requests, and a centralized dashboard to view assignments, grades, and attendance.
3. **Document Analysis:**
   * **Analyzed Artifacts:** Existing paper-based application forms, re-registration forms, student report cards, assignment sheets, and manual attendance logbooks.
   * **Justification:** Analyzing current documents provides a concrete baseline for the data the new system must capture for all users and processes.
   * **Outcomes:** This analysis directly led to the data dictionary for all user profiles (including Fayda ID), the structure of the grading and assignment modules, and the required fields for leave applications.

# 3. System Users

The EduReg system is designed to be used by several distinct types of users, each with specific roles and permissions.

* **Administrator (Director/Registrar):**
  + **Role:** The superuser of the system with full control over the academic year.
  + **Responsibilities:** Initiating the annual registration for all users. Managing all user applications. Assigning students to sections. Delegating specific approval permissions to teachers. Setting academic, attendance, and grading policies. Viewing all student and staff data, including historical performance. Approving high-level requests.
* **Student/Parent:**
  + **Role:** The end-user who completes registration and interacts with academic modules.
  + **Responsibilities:** Completing their annual registration form. Viewing assignments and submitting their work. Tracking application status. Viewing current grades, feedback, attendance records, and complete academic history. Applying for leave.
* **Teacher:**
  + **Role:** A user with both academic and delegated administrative duties.
  + **Responsibilities:** Completing their annual registration. Creating and managing assignments for their classes. Entering, updating, and sorting grades. Taking daily student attendance. Requesting special resources (labs, tutorials). Potentially approving specific applications (e.g., leave requests) if granted permission by the Director. Requesting access to view student historical data.
* **Other Staff:**
  + **Role:** A non-teaching employee of the school.
  + **Responsibilities:** Completing their annual registration, managing their personal profile, and recording their attendance.

# 4. Functional Requirements

The functional requirements are grouped by feature for clarity.

**User & Permissions Management**

* **FR-001:** The system shall enforce distinct roles (Administrator, Parent/Student, Teacher, Other Staff).
* **FR-002:** An Administrator shall be able to grant specific, limited approval permissions to users with the "Teacher" role (e.g., approve leave requests).
* **FR-003:** All users must log in to access the system's functionalities.
* **FR-004:** The system shall provide a password reset mechanism via email.

**Annual Registration & Profile Management**

* **FR-005:** An Administrator (Director) must initiate the registration process for every user (new or existing) for each academic year.
* **FR-006:** The system shall provide a form for the Administrator to register a **new** user, which creates their account.
* **FR-007:** The system shall allow an Administrator to re-register **existing** users for the new academic year.
* **FR-008:** Once registration is initiated, the respective user must log in to complete/update a detailed information form, which includes a mandatory field for the National ID (Fayda ID) for students.
* **FR-009:** Users shall be able to upload required documents (e.g., transcripts, photos).
* **FR-010:** An Administrator must review and approve each completed registration to finalize the user's active status.
* **FR-011:** Users shall be able to submit formal applications to change their legal name or personal details, which must be approved by an Administrator.

**Academic & Resource Scheduling**

* **FR-012:** An Administrator shall define and manage the Academic Year (semesters, holidays, exam weeks).
* **FR-013:** An Administrator shall create and publish detailed exam schedules (date, time, subject, room).
* **FR-014:** Teachers shall be able to request special resources (e.g., computer labs, tutorial rooms) for available time slots.
* **FR-015:** An Administrator must approve or reject teacher resource requests. Approved requests will update the relevant student and teacher timetables.

**Class & Section Management**

* **FR-016:** Administrators shall be able to define classes and sections for each grade level.
* **FR-017:** Administrators shall have multiple options to assign students to sections (manual, random, balanced by grade/gender).
* **FR-018:** Students and Parents shall be able to view their assigned class and full timetable.

**Assignment Management**

* **FR-019:** Teachers shall be able to create assignments for their classes with a title, description, due date, and optional file attachments.
* **FR-020:** Students shall have a dashboard to view all upcoming and past assignments.
* **FR-021:** Students shall be able to upload one or more files as their submission for an assignment before the due date.
* **FR-022:** Teachers shall be able to view the submission status (Submitted, Late, Not Submitted) and access the uploaded files for each student in their class.

**Grading & Performance Management**

* **FR-023:** Teachers shall have an interface to enter and update grades for assignments and exams.
* **FR-024:** The grading interface for teachers shall allow sorting of students by name or grade.
* **FR-025:** Teachers shall be able to add textual feedback along with a grade.
* **FR-026:** Students and Parents shall have a dedicated section to view current grades and teacher feedback.
* **FR-027:** The system shall securely store all previous years' performance data for each student.
* **FR-028:** Students, Parents, and Administrators shall be able to view a student's complete historical performance data.
* **FR-029:** Teachers must submit a request to an Administrator to gain temporary access to a student's historical performance data.

**Attendance & Leave Management**

* **FR-030:** An Administrator shall set school-wide attendance policies, including the maximum number of permissible absences.
* **FR-031:** A designated "Head of Class" Teacher shall take and record daily student attendance.
* **FR-032:** All staff (Teachers and Other Staff) shall record their own attendance.
* **FR-033:** All user types shall be able to apply for leave with supporting documents.
* **FR-034:** Leave applications shall be routed to the appropriate authority for approval (Director or a delegated Teacher).
* **FR-035:** The system shall track absences and automatically flag users who exceed the policy limit.

# 5. Non-Functional Requirements

**Usability**

* **NFR-001:** The user interface shall be intuitive and require minimal training for parents and administrators. A user manual will be provided.
* **NFR-002:** The system shall be responsive and fully functional on common devices, including desktops, tablets, and mobile phones.
* **NFR-003**: The system shall support three languages: English, Amharic, and Oromic.  
  **Performance**
* **NFR-004:** Web pages shall load in under 3 seconds on a standard internet connection.
* **NFR-005:** The system must support at least 200 concurrent users during peak registration periods without significant degradation in performance.

**Security**

* **NFR-006:** All user passwords must be hashed and salted before being stored in the database.
* **NFR-007:** The system shall use role-based access control to ensure users can only access data and functions relevant to their role.
* **NFR-008:** All data transmitted between the client and the server must be encrypted using HTTPS.

**Reliability**

* **NFR-009:** The system shall have an uptime of 99.5%.
* **NFR-010:** The database shall be backed up automatically on a daily basis to prevent data loss.

**Scalability**

* **NFR-0011:** The system's architecture must be capable of handling a 20% year-over-year increase in the number of students without requiring a major redesign.

# 6. Use Case Modeling

**UML Use Case Diagram**

*(This is a textual description of the diagram)*

**Actors:** Administrator, Student/Parent

**System:** EduReg System

**Use Cases:**

* **Manage Account:** (Parent/Student) -> (Includes Login, Logout, Reset Password)
* **Submit Application:** (Parent/Student) -> (Includes Fill Form, Upload Documents)
* **Upload Payment Receipt:** (Parent/Student)
* **View Application Status:** (Parent/Student)
* **View Timetable:** (Parent/Student)
* **Manage System Users:** (Administrator) -> (Includes Create User Account)
* **Manage Registrations:** (Administrator) -> (Includes View Applications, Approve Application, Reject Application)
* **Verify Documents:** (Administrator)
* **Verify Fee Payment:** (Administrator)
* **Assign Student to Section:** (Administrator)
* **Generate Reports:** (Administrator)

**Detailed Use Case Descriptions**

**Use Case 1: Submit New Student Application**

* **ID:** UC-001
* **Name:** Submit New Student Application
* **Actors:** Student/Parent
* **Description:** This use case allows a parent or student to complete and submit a new registration application for a student.
* **Pre-conditions:** The user must have a registered account created by an administrator and be logged into the system.
* **Post-conditions:** A new application is created in the system with a "Submitted" status. All provided data and documents are saved to the database. The administrator is notified of a new submission.
* **Main Flow:**
  1. User logs into the system with credentials provided by the school.
  2. User navigates to the "New Application" page.
  3. User fills in all required fields in the student and guardian information forms.
  4. User proceeds to the document upload section.
  5. User uploads all required documents (e.g., previous transcript, photo).
  6. User reviews all entered information on a summary page.
  7. User confirms and submits the application.
  8. The system displays a success message with the unique application ID.

**Use Case 2: Verify Fee Payment**

* **ID:** UC-002
* **Name:** Verify Fee Payment
* **Actors:** Administrator
* **Description:** This use case allows an administrator to review the proof of payment uploaded by a parent and update the student's fee status accordingly.
* **Pre-conditions:** The administrator must be logged in. The student must have an application with a "Pending Payment Verification" status and must have uploaded a payment receipt.
* **Post-conditions:** The student's fee status is updated to "Verified" or remains "Pending." If verified, the student moves to the next stage of registration.
* **Main Flow:**
  1. Administrator logs into the dashboard.
  2. Administrator navigates to the list of applications pending payment verification.
  3. Administrator selects a student from the list to view their application details.
  4. The system displays the student's information and the uploaded payment receipt.
  5. Administrator views the receipt and cross-references it with the school's financial records (externally).
  6. If the payment is valid, the administrator clicks the "Verify Payment" button.
  7. The system updates the student's fee status to "Verified" and logs the action.
  8. The system sends a notification to the parent that the payment has been confirmed.

**Use Case 3: Assign Student to Section**

* **ID:** UC-003
* **Name:** Assign Student to Section
* **Actors:** Administrator
* **Description:** This allows an administrator to assign a fully approved student to a specific class section for their grade level.
* **Pre-conditions:** The administrator must be logged in. The student's application status must be "Approved" (meaning documents and payment are verified). Class sections must be pre-configured in the system.
* **Post-conditions:** The student is assigned to a section. The section's occupancy count is updated. The student's registration is marked as "Completed."
* **Main Flow:**
  1. Administrator logs into the dashboard.
  2. Administrator navigates to the "Class Assignment" module.
  3. Administrator selects a grade level to view the list of approved, unassigned students.
  4. The system displays the list of students and the available sections with their current occupancy.
  5. Administrator selects a student.
  6. Administrator selects an available section from a dropdown list.
  7. Administrator clicks the "Assign" button.
  8. The system assigns the student to the section, updates their status to "Completed," and generates their timetable.
  9. The system sends a notification to the parent that registration is complete and the timetable is available.

**Use Case 4: Create New User Account**

* **ID:** UC-004
* **Name:** Create New User Account
* **Actors:** Administrator
* **Description:** This use case allows an administrator to create a new user account for a parent, student, or teacher, which then sends the login credentials to the user.
* **Pre-conditions:** The administrator is logged into the system. The administrator has the necessary personal details of the new user (full name, email, phone number, role).
* **Post-conditions:** A new user account is created in the system with the specified role. An email or SMS containing the initial username and a temporary password is sent to the user.
* **Main Flow:**
  1. Administrator navigates to the "User Management" section of the dashboard.
  2. Administrator clicks "Create New User".
  3. Administrator fills in the user's details: full name, email address, phone number, and selects the user role (Parent/Student, Teacher).
  4. Administrator confirms the creation of the account.
  5. The system creates the new user account in the database.
  6. The system generates a temporary password.
  7. The system sends an email or SMS to the user with their username (email) and the temporary password, along with a link to log in.
  8. The system displays a success message to the Administrator.

# 7. Requirements Specifications

Here are three requirements specified using Structured Natural Language.

**Specification 1**

* **ID:** FR-012
* **Function:** Approve or Reject a Document
* **Source:** Administrator Interview
* **Description:** The system shall allow an **Administrator** to review an uploaded **Document** associated with a **Student Application** and mark it as either "Verified" or "Rejected."
* **Inputs:** Student Application ID, Document File.
* **Action:** If the document is rejected, the system shall require the Administrator to provide a brief reason for rejection.
* **Outputs:** The document's status is updated. A notification with the reason is sent to the **Parent/Student** if the document is rejected.

**Specification 2**

* **ID:** FR-020
* **Function:** View Final Timetable
* **Source:** Parent/Student Questionnaire
* **Description:** The system shall allow a **Parent/Student** to view the **Student's** final timetable only after their application status is marked as "Completed."
* **Pre-condition:** The **Administrator** must have assigned the student to a section.
* **Action:** The system presents the timetable in a clear, weekly-view grid format.
* **Outputs:** A display of the student's class schedule, including subjects, teacher names, and room numbers for each period.

**Specification 3**

* **ID:** FR-023
* **Function:** Generate Class Roster
* **Source:** Teacher Interview
* **Description:** The system shall allow an **Administrator** or **Teacher** to generate a **Class Roster** report for any given class section.
* **Inputs:** Grade Level, Section Name.
* **Action:** The system will gather the list of all students assigned to the specified section.
* **Outputs:** A report in PDF format containing the student's full name, student ID, and guardian contact information for every student in that section.

# 8. Traceability Matrix

This matrix maps the functional requirements (FR) to the use cases (UC) they support, ensuring all requirements are addressed in the system's design.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement ID** | **Description** | **UC-001 (Submit App)** | **UC-002 (Verify Pay)** | **UC-003 (Assign to Section)** | **UC-004 (Create User)** |
| **FR-001** | Admin creates user accounts |  |  |  | X |
| **FR-005** | Online application form | X |  |  |  |
| **FR-008** | Assign unique application ID | X |  |  |  |
| **FR-010** | Upload documents | X |  |  |  |
| **FR-014** | Upload proof of payment | X |  |  |  |
| **FR-015** | Admin views proof of payment |  | X |  |  |
| **FR-016** | Admin updates payment status |  | X |  |  |
| **FR-019** | Admin assigns student to section |  |  | X |  |
| **FR-020** | Student views timetable |  |  | X |  |