Contents

Specification of Student Management System

Group 1

Version 1.2

October 24, 2025

I. Document Change Log

Date	Version	Description	Author
10/19/2025	1.0	Created document	Truong Do Vuong
10/21/2025	1.1	Added Functional/NonFunctional Requirements	Truong Do Vuong
10/24/2025	1.2	Rewrite Requirements, added FDD	Truong Do Vuong

II. Student Management system

The **Student Management system (SMS)** is designed to administrative activities within a school or university. The **system** provides essential functions such as **managing students, grades, and user accounts for students and Administrators.**

1 Main Functional Requirements

1.1 Student Registration and Profile Management

- **Administrators** can register new **students** by entering information such as full name and citizen ID.
- During registration the student's **StudentID** is set exactly to their citizen ID (i.e., **StudentID = CitizenID**).
- The system issues a default temporary password equal to the student's date of birth (stored and applied in MMDDYYYY format). The student is required to change this default password at first login.
- Newly registered students can log in using their StudentID (citizen ID) and the default password, then change their password and update additional profile information including date of birth, gender, address, contact number, and email, portrait.
- **The system** allows viewing and (where permitted by role) updating of student information.
- Each student has a unique ID for identification within the **system** (the citizen ID).

1.2 Fee and Financial Management

- Financials page
  - **Quick access from profile:** From the student's main profile page a single link ("Financials") opens a consolidated financial view for that student.

  - **Profile-level summary:** The top of the financial view shows: **StudentID** (which equals CitizenID), Date of Birth (for verification), Full Name, Major/Program, current Enrollment Status, total outstanding balance, and next due date.
  - **Line-item fee list:** A compact table lists every charge related to the student (course tuition, lab fees, insurance, hostel, fines, etc.) with columns: Fee Code, Description, Period, Amount, Due Date, and Status (paid/partial/overdue).
  - **Transaction summary:** Under the line-items show: Subtotal, Discounts/Scholarships, Taxes (if any), Amount Paid, and **Total Due** (final balance to pay).

- **Payment page:**
  - Displays StudentID (CitizenID), DOB, Name at the top for confirmation.
  - Shows the same line-item list with checkboxes allowing full or partial selection for payment.
  - Display a note on where to go (Room's number, Open-Closing time)

1.3 User Authentication and Role Management

- **The system** supports two types of users: Administrator and Student.
- Each user logs in with a username and password.
- Password recovery is available via registered email.
- Permissions and access levels differ by role:
  - **Administrator:** Full system control and data management
  - **Student:** View profile

1.4 ADMIN Management

- **Administrator landing page:** When an **Administrator** logs in the system displays an **Admin Dashboard** instead of the student home page. The dashboard's primary purpose is administrative data management and quick navigation to operational tools.
- **Table selector (primary control):** A prominent control (dropdown or left-side list) lets the admin choose which database table to open. table choices include: `Students`, `Payments`.
- **Data grid / table viewer:**

  - Displays the selected table in a sortable and filterable grid with column.
  - Every entries can be **Edit**,
  - Every rows can be **Delete** and **Export row**.
  - Every change are "soft" meaning that **Admin** have to "press" **comfirm change** button to save.

- Deletions are soft by default (mark as inactive) with an option for permanent removal restricted to super-admins.

1.5 Notifications and Announcements

- **Administrators** have the option to post announcements in **Admin Dashboard**.
- **students** receive notifications for new announcements in the login page and after login.

2 Functional Decomposition Diagram

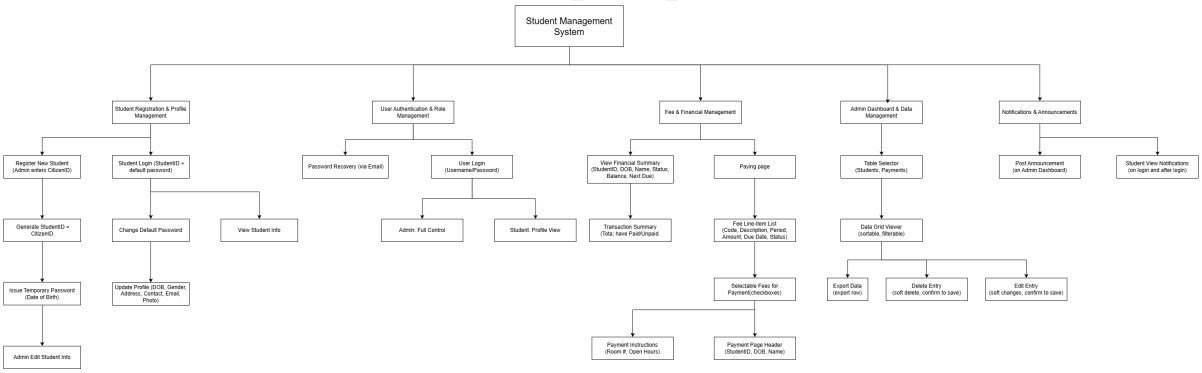

Figure II.1: FDD for Student Management System

3 Non-Functional Requirements

3.1 Usability

- Simple and intuitive user interface for basic testing and demonstrations.
- Layout is partially responsive, suitable for desktop browsers.
- Minimal visual design focused on clarity and ease of understanding rather than aesthetics.

3.2 Reliability

- Intended for local testing; continuous 24/7 uptime is not required.
- Occasional restarts or maintenance are acceptable in a learning environment.
- Data loss is not critical; database resets may occur during experiments.

3.3 Performance

- Optimized for small-scale testing (up to 10 simultaneous users).
- Response time within 1–3 seconds under normal test load.
- Performance tuning (e.g., caching, load balancing) is not a priority for this learning project.

3.4 Supportability

- Codebase is modular and commented for easy understanding by students.
- Documentation provided for setup, execution, and testing steps.

3.5 Security

- Authentication for users (student, admin roles).
- Input validation to prevent common mistakes, simple injection attempts, DOS/DDOS.

3.6 Design Constraints

- Platform: Web-based (HTML5, CSS3, Python, MongoDB).
- Operating System: Works on Windows, Linux, or macOS.
- Framework: May use Flask or Django for backend logic.
- Deployment: Localhost or free-tier hosting (e.g., Render, Replit, or Vercel).

III. Functional Decomposition Diagram