**A.1 Overview**

The **Advanced Student Management System** is a comprehensive web-based application designed to streamline and manage student-related data efficiently. This project provides essential features to handle various operations related to student records. The system is user-friendly and supports real-time data manipulation through a dynamic and interactive interface.

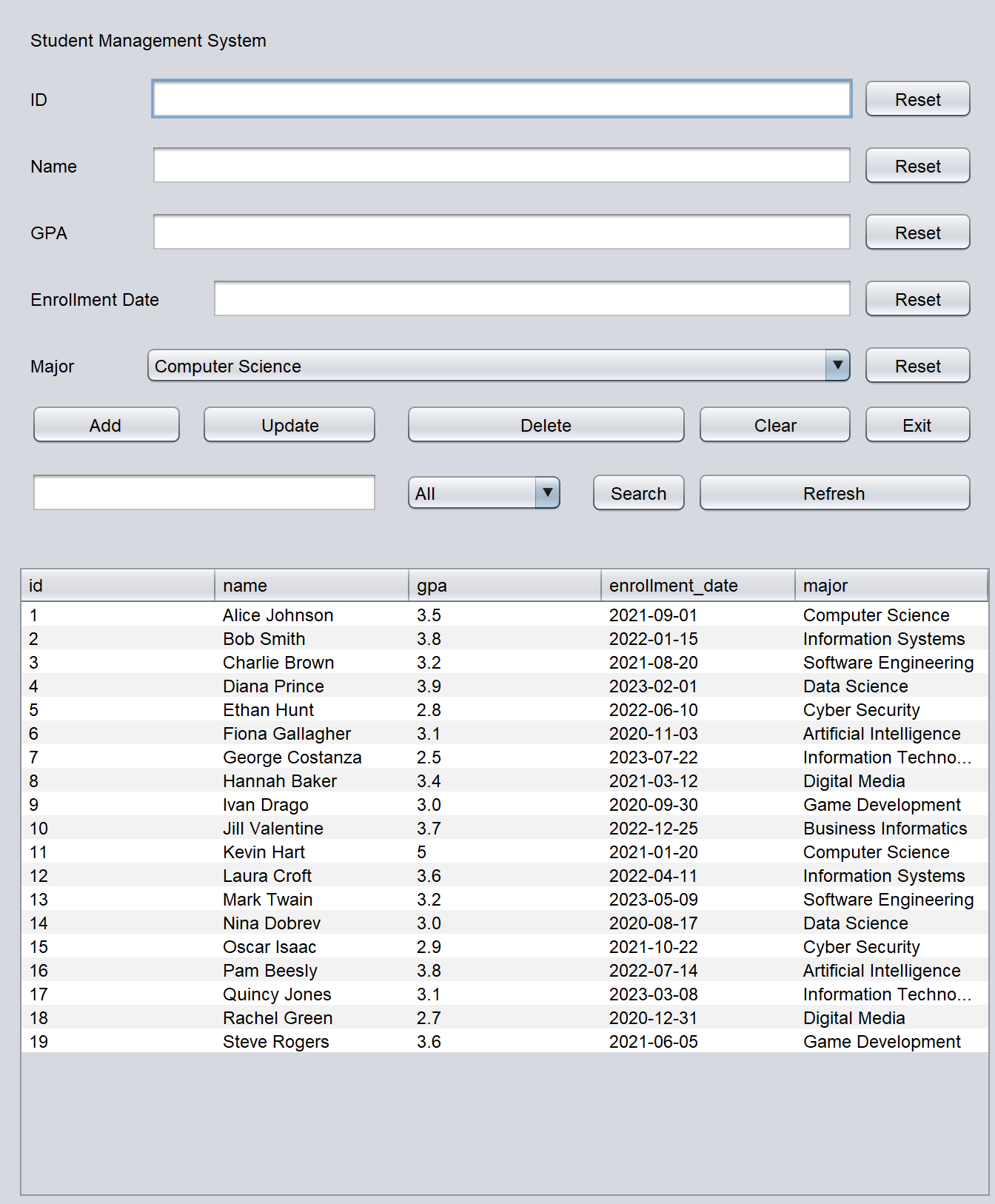
**Key Features:**

1. **Add Student** – Create new student entries with relevant information.
2. **Update Student** – Modify existing student records.
3. **Delete Student** – Remove student records using a unique identifier.
4. **Search Student** – Filter student records by name and/or major.
5. **Auto Display Students** – Automatically load and display all students when the application starts.
6. **Select Student from Table** – Click on a student row to auto-fill the form with their data for easy updates or deletion.

**A.2 Application View**

Click the [link](https://github.com/RickWijaya/OOVP_Semester_2_Project) for the full source code

The application features a simple, intuitive layout consisting of:

* A form section for inputting and editing student data
* A data table that lists all student records
* Search functionality at the top of the table
* Interactive buttons for add, update, delete, clear, exit and search actions

**A.3 Instructions**

**1. Adding a Student**

To add a new student to the system, follow these steps:

* Fill out all the required fields in the form.
* Ensure all values are valid and correctly formatted.
* Click the **"Add"** button to submit the new record.

**Required Fields:**

* **Name:** The full name of the student (cannot be empty).
* **ID:** A unique student identifier (e.g., numeric or alphanumeric).
* **GPA:** Grade Point Average (must be a valid number between 0.0 and 4.0).
* **Enrollment Date:** The date the student joined (format: YYYY-MM-DD).
* **Major:** The student's field of study. If no major is selected, it defaults to *Computer Science*.

**2. Updating a Student**

To update an existing student's information:

* Select the student from the data table or manually fill in the existing student’s ID and details.
* Modify the desired fields.
* Click the **"Update"** button to apply the changes.

**Required Fields:**

Same as the *Add Student* section. The ID must match an existing record in the system.

Note: If no new major is selected during the update, the student's current major will be retained.

**3. Deleting a Student**

To remove a student from the system:

* Enter the **ID** of the student to be deleted.
* Click the **"Delete"** button.

Make sure the ID is correct. Deletion is permanent and cannot be undone.

**4. Searching for Students**

To search for a specific student or group of students:

* Enter the **student's name** in the search input field.
* Optionally, select a **major** to narrow down the results.
* Click the **"Search"** button.

The table will display all students that match the search criteria. If no match is found, a message will appear indicating that no records were found.

**A.4 Additional Notes**

* All data is validated before submission to ensure accuracy and prevent duplication.
* The system is optimized for responsiveness and user-friendly interactions.
* Each action (Add, Update, Delete, Search) provides feedback or alert messages to guide the user.
* Selecting a row from the student table will auto-fill the form with that student’s data, simplifying updates and deletions.

**A.5 TechStack**

* Front-End Java GUI Builder
* BackEnd Java MySQLI Connector