**Project Documentation Report   
Student Management System**

**First Term (Final Project 1)**

Eng. Abdallah Shabaan Ghazy

email:[abdallah.shabaan.ghazy@gmail.com](mailto:abdallah.shabaan.ghazy@gmail.com)

**My Profile:**

<https://www.learn-in-depth-store.com/certificate/abdallah.shabaan.ghazy%40gmail.com>

**Version 1.0 approved**

**Prepared by:** Abdallah Shabaan Ghazy

**Organization:** learn in depth

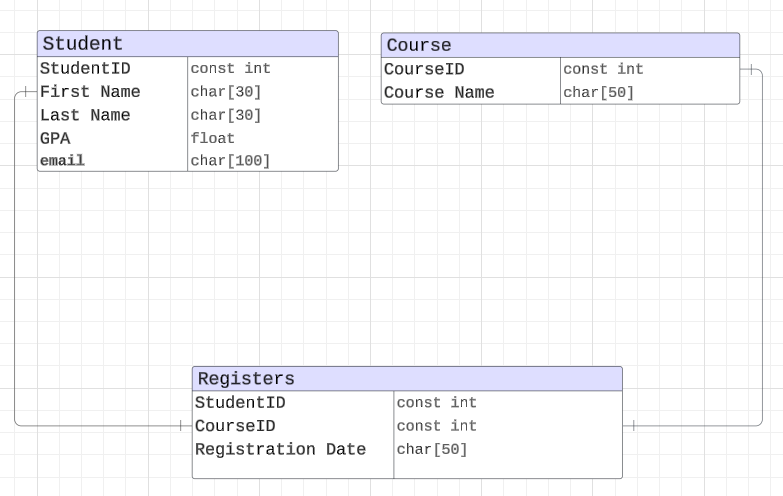
**Date Created:** 8/14/2024

## **1. Introduction**

The "Student Management System" project is a Command Line Interface (CLI) application designed to manage student data efficiently. The system supports various operations to add, update, delete, and retrieve student information. This documentation provides a comprehensive overview of the system's functionalities, implementation details, and usage instructions.

## **2. Objectives**

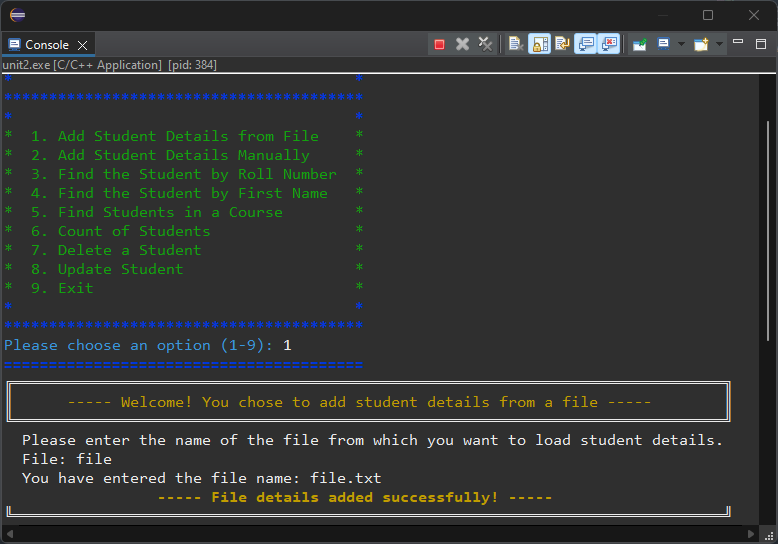
1. **Streamline Student Data Management**: Provide an efficient CLI for managing student records.
2. **Enable Easy Data Retrieval**: Allow users to search and retrieve student information based on different criteria.
3. **Facilitate Data Manipulation**: Offer functionalities to update, delete, and analyze student data.



## **3. Core System Functions**

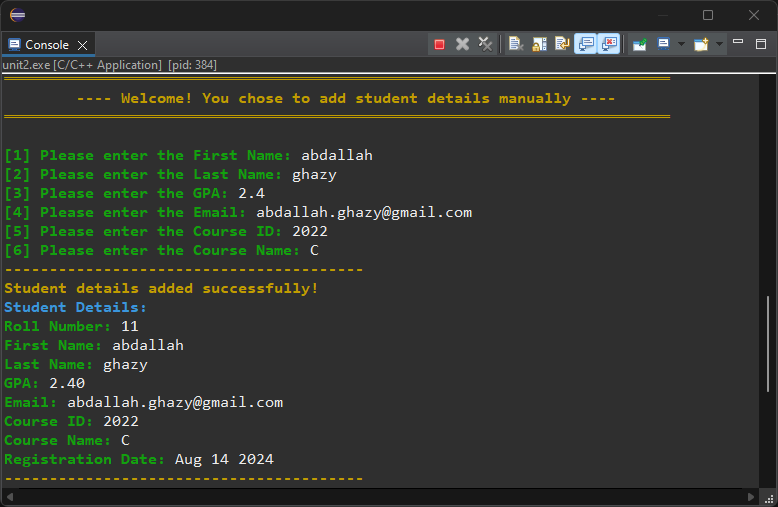
### ****3.1. Add Student Details from File****

* **Description**: Imports student data from a specified text file.
* **Inputs**: File path.
* **Actions**:
  1. Read student data from the file.
  2. Validate and parse the data.
  3. Add records to the system.

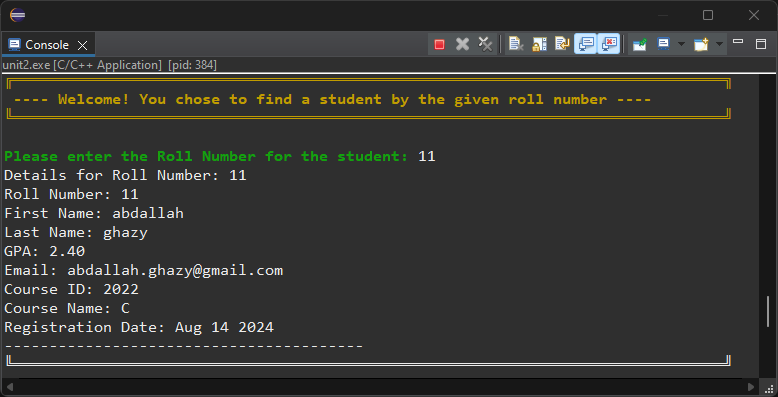


### ****3.2. Add Student Details Manually****

* **Description**: Allows users to input student details manually through the CLI.
* **Inputs**: Roll Number, First Name, Last Name, GPA, Email, Course ID, Course Name, Registration Date.
* **Actions**:
  1. Prompt for and collect user inputs.
  2. Validate the entered data.
  3. Save the student record to the system.

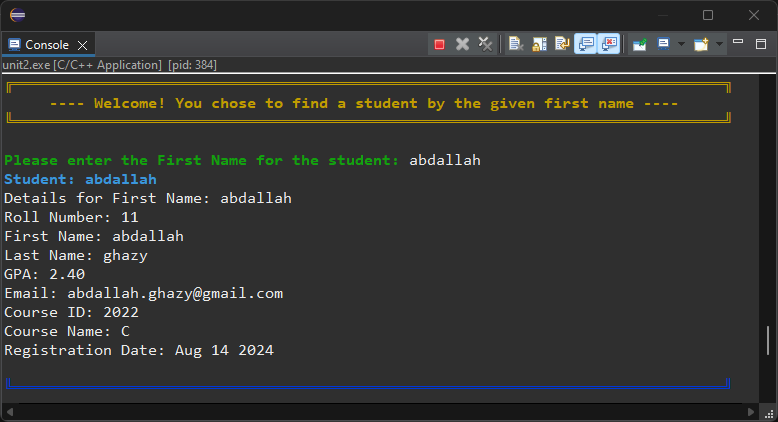


### ****3.3. Find Student by Roll Number****

* **Description**: Searches for and displays a student's details based on their roll number.
* **Inputs**: Roll Number.
* **Actions**:
  1. Search the database for the student using the roll number.
  2. Display the student's information if found.

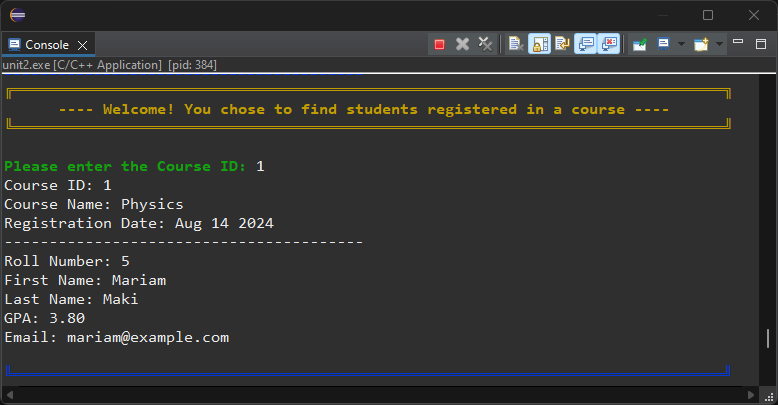
### ****3.4. Find Student by First Name****

* **Description**: Searches for students with a given first name.
* **Inputs**: First Name.
* **Actions**:
  1. Search the database for students matching the first name.
  2. Display a list of matching student records.



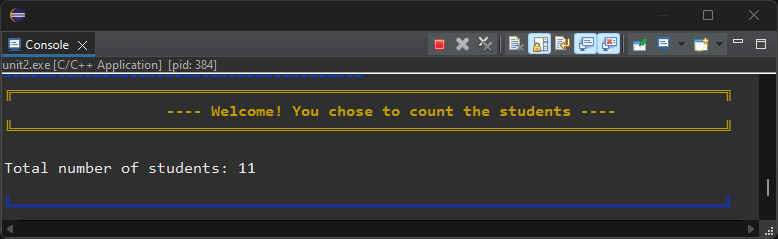
### ****3.5. Find Students in a Course****

* **Description**: Lists all students enrolled in a specific course.
* **Inputs**: Course ID.
* **Actions**:
  1. Search for students registered in the specified course.
  2. Display the details of students in that course.

****

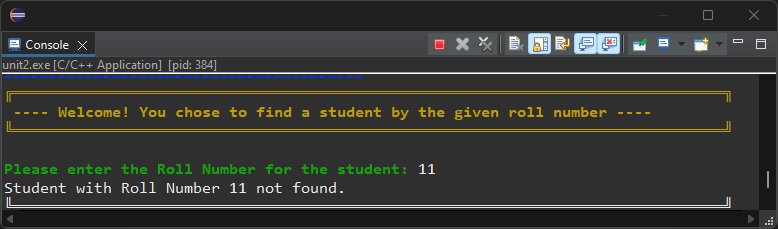
### ****3.6. Count of Students****

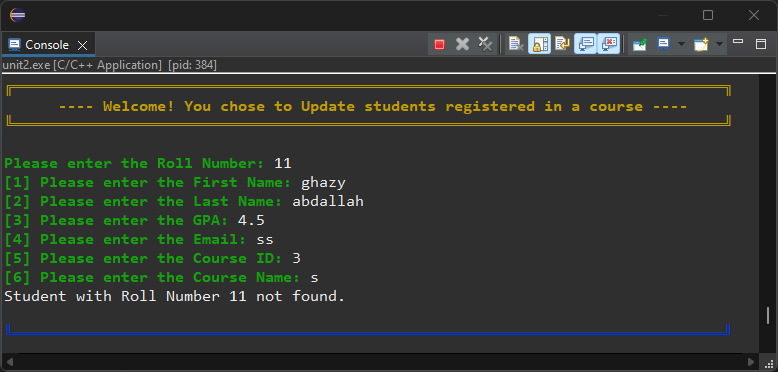
* **Description**: Provides the total number of students in the system.
* **Actions**:
  1. Count the total number of student records.
  2. Display the count to the user.

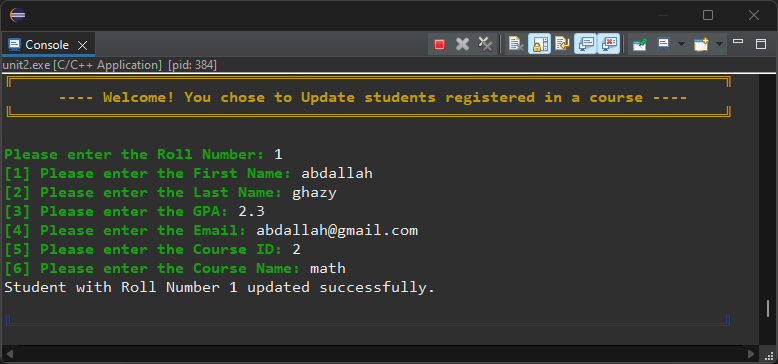


### ****3.7. Update Student****

* **Description**: Updates a student's details based on their roll number.
* **Inputs**: Roll Number, new First Name, new Last Name, new GPA, new Email, new Course ID, new Course Name.
* **Actions**:
  1. Find the student using the roll number.
  2. Update the student's details with new information.

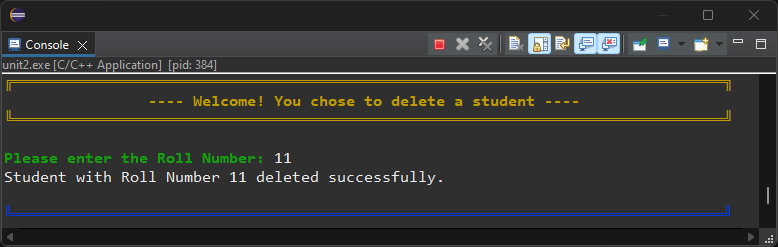






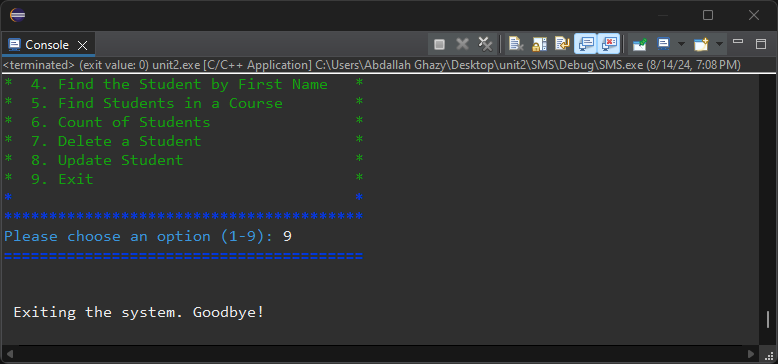
### ****3.8. Delete a student****

* **Description**: Deletes a student's record based on their roll number.
* **Inputs**: Roll Number.
* **Actions**:
  1. Locate the student using the roll number.
  2. Remove the student's record from the system.



### ****3.9. Exit****

* **Description**: Exits the application.
* **Actions**:
  1. Save any unsaved changes.
  2. Terminate the application.



## **4. Technical Design**

* **Interface**: Command Line Interface (CLI).
* **Data Storage**: Text files used for storing student and course data.
* **Validation**: Ensures input data is accurate and adheres to expected formats.