STUDENT INFORMATION SYSTEM (SIS)

USER MANUAL & DOCUMENTATION

Course: CSE102 - Computer Programming

Project: Term Project

Student Name: Yener Tuncer

Student ID: 230104004081

Date: January 16, 2026

📄 TABLE OF CONTENTS

4. PHASE 3: GRADING & REPORTING

- 4.1. Assigning Grades
- 4.2. Generating Transcripts
- 4.3. Graduation Eligibility Check (Bonus)

5. DATA MANAGEMENT & BACKUP

- 5.1. Saving Your Work
- 5.2. Restoring from Backup
- 5.3. File Structure

6. APPENDIX

- 6.1. Error Code Reference
- 6.2. Troubleshooting

---

1. GETTING STARTED

1.1. System Overview

The Student Information System (SIS) is a robust, command-line interface (CLI) application developed to facilitate the management of university academic records. It serves as a central database for:

- **Student Records: Personal information, contact details, and academic history.**
- **Course Catalog: Managing curriculum, credits, and prerequisites.**
- **Enrollment: Handling complex logic like capacity checks and prerequisite validation.**
- **Grading: Calculating GPAs and generating transcripts dynamically.**

The system relies on a Linked List data structure for runtime operations and CSV Files for persistent storage, ensuring speed and data reliability.

1.2. Installation & Launch

To set up the environment on your local machine, follow these steps:

Step 1: Extract Files Unzip the StudentInfoSystem_Tuncer.zip archive. You should see the following folders:

- src/: Contains source codes (.c files).
- data/: Contains database files (students.csv, courses.csv).
- Makefile: Configuration for compilation.

Step 2: Compile the Project Open your terminal in the project directory and run the compilation command:

Bash

```
make
```

Step 3: Launch Run the executable to start the system:

Bash

```
./StudentSystem
```

Note: Ensure the data/ folder exists before running the program to prevent "File Not Found" errors.

1.3. The "5-Minute" Setup Guide

Follow these quick steps to test the system immediately:

1. Go to **Course Management** and add a test course (Code: TEST101).
2. Go to **Student Management** and add a student (ID: 1001).
3. Go to **Enrollment Management** and enroll Student 1001 to TEST101.
4. Go to **System Options** and **Save**.

---

2. PHASE 1: ACADEMIC SETUP

Before students can enroll, the academic infrastructure (courses and professors) must be defined.

2.1. Defining the Curriculum (Courses)

- Path: Main Menu > Option 2

To open a new course, provide a unique Course Code and Capacity.

This module allows administrators to build the course catalog. Each course is defined by a unique code, name, credit value, and quota.

How to Add a New Course:

1. Select Option 1 (Add New Course).
2. **Course Code:** Enter a unique code (e.g., CSE102). This is used for prerequisites.

3. **Capacity:** Set the maximum number of students allowed.
4. **Prerequisites:** You can specify a required previous course. If none, enter -.

- Tip: You can set "Prerequisites" here to restrict future enrollments.

```
--- COURSE MANAGEMENT ---
1. Add New Course
2. Edit Course Info
3. Delete Course
4. Search & View Details
5. Display All Courses (with Prerequisites)
0. Back
Choice: 1
ID (0 to cancel): 8001
Code: CSE801
Name: Computer Analysis
Credits: 4
Capacity: 80
Dept: Computer Science
Prereqs (comma sep, 0 for none): 0
Success: Course 'Computer Analysis' added successfully!
```

2.2. Managing Faculty (Professors)

- Path: Main Menu > Option 3

Add professors to the system to assign them to courses later.

```
--- PROFESSOR MANAGEMENT ---
1. Add Professor
2. Edit Professor (Supports '0' to Cancel)
3. Delete Professor
4. Search & View Details
5. Assign Course to Professor
6. Display All Professors
0. Back
Choice: 1
ID (0 to cancel):
```

PHASE 2: STUDENT ADMISSION & ENROLLMENT

This section covers the daily operations of the registrar office.

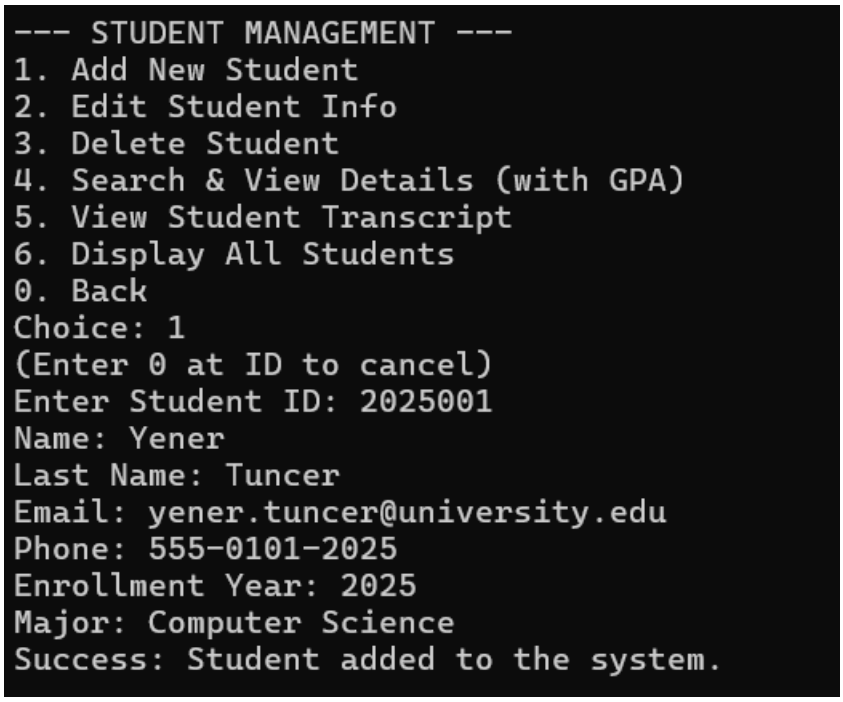
3.1. Registering New Students

- Path: Main Menu > Option 1 > Add Student

The system requires a unique numeric ID for each student. Duplicate IDs are automatically rejected for data integrity.

Data Entry Steps:

1. **Student ID:** Enter a unique number (e.g., 2025001).
2. **Name/Surname:** Enter the student's full name.
3. **Validation:** The system checks if the email contains @ and if the ID is unique.



3.2. Enrollment Process

- Path: Main Menu > Option 4 > Enroll

This is the core function where Students are linked to Courses.

The Logic Behind Enrollment: When you attempt to enroll a student, the system performs three critical checks in the background:

1. **Existence Check:** Do the Student and Course exist?
2. **Capacity Check:** Is the course full?
3. **Prerequisite Check:** Has the student passed the required prior course?

```
--- ENROLLMENT MANAGEMENT ---
1. Enroll Student
2. Drop Student
3. View Student Enrollments
4. View Course Roster
0. Back
Choice: 1
Student ID: 2025001
Course ID: 1001
Assign Professor ID: 5001
Semester: SPRING
Success: Enrollment added successfully!
```

- **Automatic Checks:** The system will block enrollment if:
    - The course capacity is full.
    - The student has not passed the required prerequisite.

```
--- ENROLLMENT MANAGEMENT ---
1. Enroll Student
2. Drop Student
3. View Student Enrollments
4. View Course Roster
0. Back
Choice: 1
Student ID: 2025001
Course ID: 1002
Error: Student has not completed prerequisite: CS101
Error: Prerequisites not met!
```

---

4. PHASE 3: GRADING & REPORTING

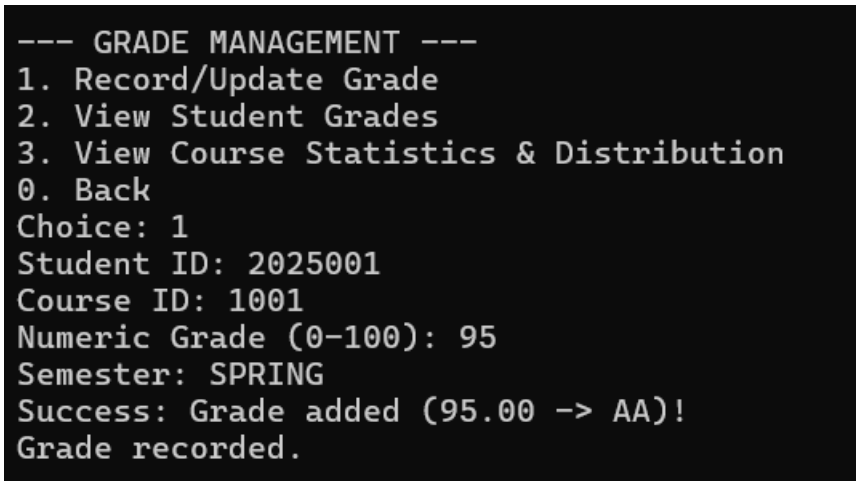
At the end of the semester, grades are entered and reports are generated.

4.1. Assigning Grades

- Path: Main Menu > Option 5

Grades are entered as numeric values (0-100). The system uses a built-in algorithm to convert these scores into Letter Grades (AA, BB, CC, etc.) automatically.

Grading Scale: | Numeric Score | Letter Grade | Status | | :--- | :--- | :--- | | 90 - 100 | AA | Passed | | 85 - 89 | BA | Passed | | ... | ... | ... | | 0 - 49 | FF | Failed |



4.2. Generating Transcripts

- Path: Main Menu > Option 6 > Student Transcript

The transcript module compiles the student's entire academic history. It lists every course taken, the semester, the grade received, and calculates the **Cumulative GPA** dynamically.

```
--- REPORTS ---
1. Student Transcript
2. Course Roster
3. Grade Statistics
4. Professor Course Load
0. Back
Choice: 1
Student ID: 2023001

================================================================================
                            TRANSCRIPT OF RECORDS
================================================================================
Student: Can Turan (ID: 2023001)
Major:   Computer Science
================================================================================
Code     Course Name                          Semester        Credits Grade    Letter
--------------------------------------------------------------------------------
CS101    Introduction to Programming          2023-FALL       4       92.00    A-
MATH101  Calculus I                           2023-FALL       4       88.50    B+
MATH202  Discrete Mathematics                 2023-FALL       3       90.50    A-
CS102    Data Structures                      2024-SPRING     4       93.50    A
MATH102  Calculus II                          2024-SPRING     4       87.50    B+
================================================================================
Total Credits: 19
Cumulative GPA: 3.79 / 4.00
================================================================================
```

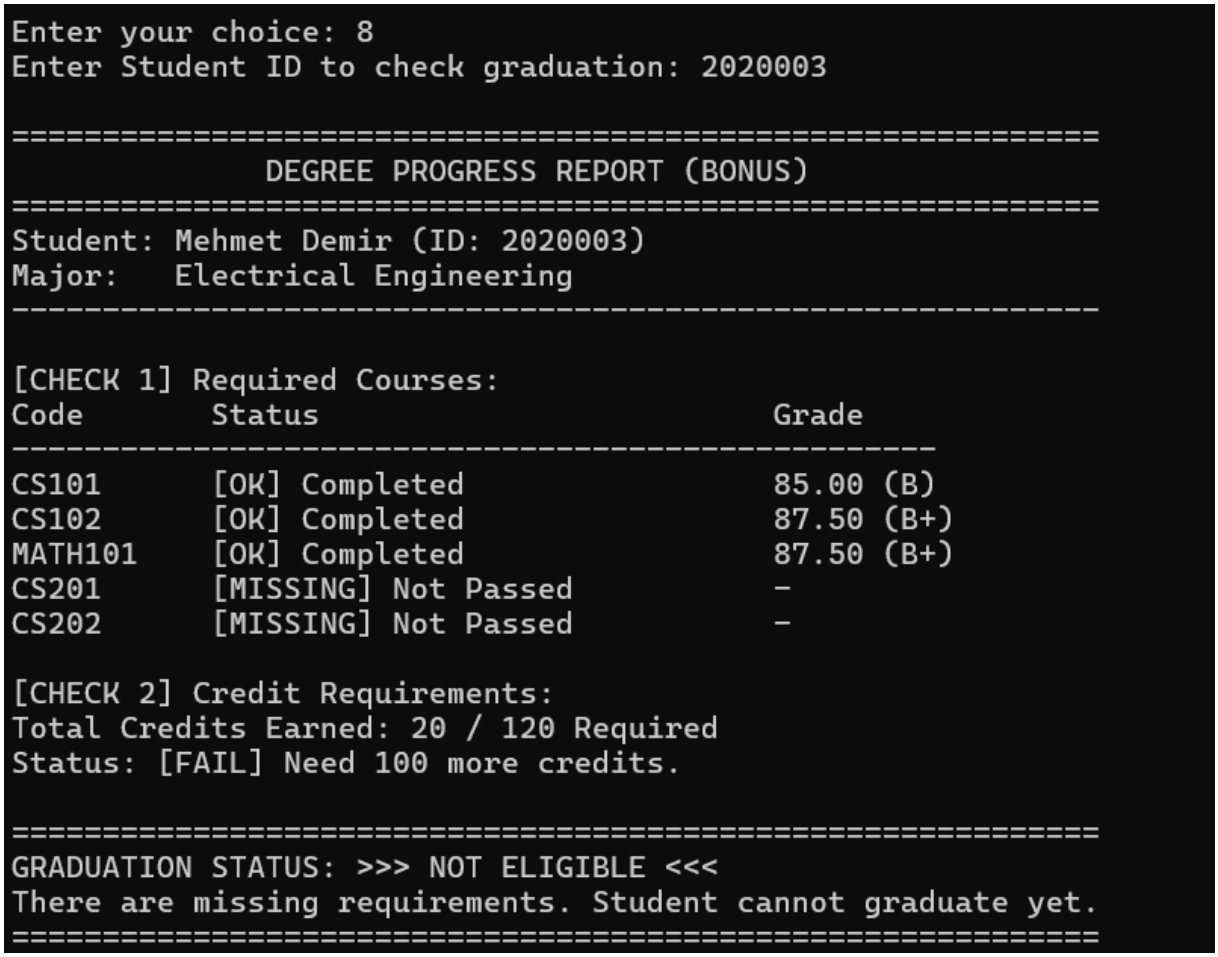
4.3. Graduation Eligibility Check (Bonus)

- Path: Main Menu > Option 8

This advanced feature analyzes the student's transcript against graduation requirements.

- **Requirement 1:** Total Credits must be >= 120.
- **Requirement 2:** All mandatory courses must be completed.

- **Requirement 3:** No active "FF" grades.



---

5. DATA MANAGEMENT & BACKUP

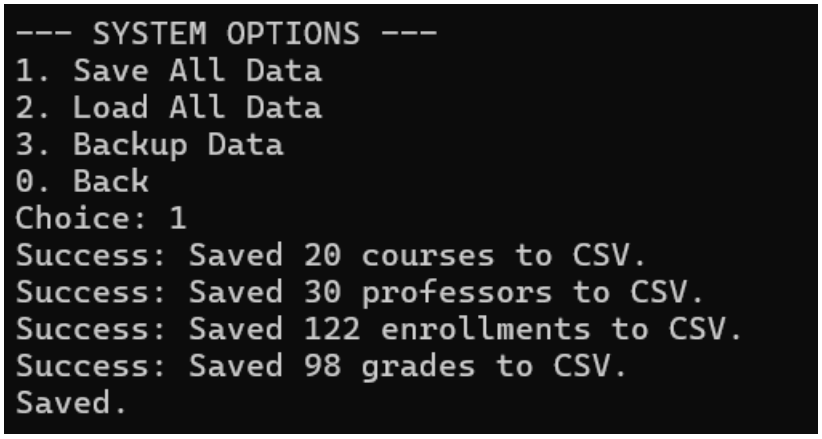
The system uses CSV files for persistent storage.

5.1. Saving Your Work

- Path: Main Menu > Option 7 > Save

Always save before exiting to write changes to the data/*.csv files.

CRITICAL WARNING: The system operates in the computer's memory (RAM). You must save your work before exiting the application. If you close the window without saving, any new entries (Students, Grades, etc.) will be lost.



5.2. File Structure

Do not manually edit CSV files while the program is running.

- students.csv: Student records.
- courses.csv: Course catalog.
- enrollments.csv: Relationship links.

5.3. 7- System Options Menu: This menu provides essential tools for maintaining your database:

- **1. Save All Data:** Manually saves the current state of the system to primary CSV files.
- **3. Create FULL System Backup:** Creates a safety copy of all records. It is highly recommended to perform this after major changes (e.g., end-of-semester grading).
- **4. RESTORE System From Backup:** Use this option if you encounter data corruption. It will wipe the current unsaved session and reload everything from your latest backup files.

6. APPENDIX

6.1. Error Code Reference

Message	Context	Solution
[DENIED] Duplicate ID	**Adding Student/Course**	**The ID is already in use. Try a different ID.**
[DENIED] Capacity Full	**Enrollment**	**The course quota is full. Increase capacity or drop a student.**
[DENIED] Prerequisite Missing	**Enrollment**	**The student must enroll in and pass the prerequisite course first.**
Invalid Input	**General Menus**	**You entered text where a number was expected. Please enter digits only.**

6.2. Troubleshooting

- **Q: Why are my characters (ş, ğ) broken?**

  A: This is a Windows Console limitation. The data is saved correctly; please view CSV files in Notepad++.

Q: How do I reset the system to factory settings?

- **A:** Simply delete all .csv files in the data/ folder. The system will detect missing files and start with empty lists (or you can replace them with your backup files).

Q: The program closes immediately after I run it.

- **A:** If you are double-clicking the .exe file, it might close after execution. It is recommended to run the program from a Command Prompt (cmd) window or PowerShell to see the output logs.

Q: Duplicate Entry Detected

A: Use "System Options" -> "Load Data" only once at startup or restart the program to clear memory.

Q: Makefile: No such file or directory.

A: You are trying to run make inside the src folder instead of the project root.

Q: Permission Denied (Linux)

A: The StudentSystem binary does not have execution permissions. Run chmod +x StudentSystem in the terminal to grant execution rights.