**University Registration System Documentation**

**Group 5**

**Overview**

The University Registration System is a command-line application designed for Zed University to manage student registration, course enrollment, and scheduling. The system allows students to register for accounts, view available courses, enroll in courses, and manage their academic schedules while ensuring there are no timing conflicts or credit limit violations.

**Team Member Contributions**

**Grace Ngoma (G00450740)** : Input Validation & Error Handling

- *Primary Responsibilities*: Implemented comprehensive input validation and error handling throughout the system

- *Specific Tasks:*

- Created validation checks for all user inputs (student IDs, passwords, menu selections)

- Implemented try/except blocks to catch and handle exceptions gracefully

- Designed informative error messages to guide users

- Assisted with the course conflict detection algorithm

- Helped with enrollment verification logic

**Chilawo Nchimunya Munene (G00450721)**: User Interface & Navigation

- *Primary Responsibilities*: Designed and implemented the console-based user interface and navigation flow

- *Specific Tasks:*

- Created the main menu, login, and student dashboard interfaces

- Implemented screen clearing and formatting for better readability

- Designed user-friendly prompts and information displays

- Contributed to course data presentation formatting

- Assisted with student registration flow

**Shalom Donga (G00450733)**: Enrollment Logic & Academic Rules

- *Primary Responsibilities*: Implemented the core enrollment logic and academic rule verification

- *Specific Tasks:*

- Developed schedule conflict detection algorithm

- Implemented credit limit verification

- Created course capacity management

- Designed the course drop functionality

- Contributed to the Student class implementation

- Assisted with data loading for enrollment relationships

**Dingani Freddie Kandiwo (G00450731)**: Data Storage & System Architecture

- *Primary Responsibilities:* Designed the system architecture and implemented data persistence

- *Specific Tasks:*

- Created the CSV-based storage system for students, courses, and enrollments

- Implemented data loading and saving functionality

- Developed sample data generation for testing

- Contributed to the course filtering logic

- Assisted with the authentication system

**Challenges & Solutions**

*Challenge 1:* Data Consistency & Integrity

Ensuring data remains consistent across different operations and between program sessions.

*Solution:* Implemented save operations and comprehensive validation before data modifications. Added fallback mechanisms for header variations in CSV files.

*Challenge 2*: User Experience in Console Interface

Creating an intuitive experience despite the limitations of a text-based interface.

*Solution:* Designed clear menu structures with consistent formatting, helpful prompts, and informative messages. Implemented screen clearing to reduce visual clutter.

*Challenge 3:* Schedule Conflict Detection

Developing an algorithm to accurately detect time conflicts between courses.

*Solution:* Created a two-level check that first identifies day overlaps, then checks for time intersections only on overlapping days, improving both accuracy and performance.

*Challenge 5:* Modular System Design

Creating a system that separates concerns while maintaining coherent functionality.

*Solution*: Implemented a clear separation between data management (EnrollmentSystem) and user interface (UniversitySystem), allowing each component to focus on specific responsibilities.

**Code Structure**

The system is built around four main classes:

1. Student: Manages student information and enrolled courses

2. Course: Handles course details, scheduling, and student enrollment

3. EnrollmentSystem: Controls data persistence and enrollment business logic

4. UniversitySystem: Provides the user interface and control flow

**Future Enhancements**

1. Graphical user interface implementation

2. Database integration to replace CSV files

3. Administrator functionality for course and student management