

## **1. Purpose**

**This document describes the functional and non-functional requirements for the Course Registration System. The system allows students to register for courses, view their schedules, and manage their enrollment. Administrators can add, edit, and delete course information.**

**All requirements are written in a clear and measurable form.**

---

## **2. Functional Requirements**

---

### **a. User Registration & Login**

#### **Description:**

- **The system shall allow students to create an account and log in securely.**

#### **Requirements:**

- **The system shall allow students to register using name, email and password.**
  - **Registration and login shall not exceed three steps.**
  - **User data shall be stored using C++ classes.**
  - **The registration/login module shall be completed before any course-related functionality becomes available.**
- 

### **b. View Available Courses**

#### **Description:**

- **Students shall be able to view a list of all available courses.**

#### **Requirements:**

- **The system shall display the course code, course name, capacity and credit hours.**
- **The system shall show 100% of the stored courses.**
- **Course data shall be retrieved from a collection of Course objects.**
- **This feature shall be accessible immediately after the log in.**

---

### **c. Course Registration**

#### **Description:**

- The system shall allow students to enroll in available courses.

#### **Requirements:**

- A student may register for a course only if the course capacity is not exceeded.
- A student shall be allowed to register for a maximum of six (6) courses per semester.
- Enrollment shall link Student and Course objects.
- The course registration module shall be part of the main student interface.

---

### **d. Course Dropping**

#### **Description:**

- Students shall be able to drop courses they have previously registered for.

#### **Requirements:**

- A student shall only be allowed to drop a course that exists in their current schedule.
- Dropping a course shall remove it from the student's schedule.
- This feature shall be available from the same menu as course registration.

---

### **e. View Student Schedule**

#### **Description:**

- Students shall be able to view the list of courses they are registered for.

#### **Requirements:**

- The schedule shall list the course code, course name and credit hours.
- Data shall be retrieved from the student's stored course list.

- This feature shall be available in the same menu as registration and dropping functionality.
- 

#### **f. Admin Course Management**

##### **Description:**

- Administrators shall manage course information.

##### **Requirements:**

- The admin shall be able to add new courses.
  - The admin shall be able to edit existing course data.
  - The admin shall be able to delete courses.
  - All admin modifications shall be reflected immediately in the stored course data.
- 

#### **g. Enrollment Capacity Enforcement**

##### **Description:**

- The system shall enforce maximum course capacity.

##### **Requirements:**

- Each course shall have a defined maximum capacity.
  - The system shall prevent registration if the course is full.
  - Capacity checks shall be performed every time a student attempts to register.
- 

### **3. Non-Functional Requirements**

#### **a. Performance**

- The system shall load all course data within 2 seconds of startup.
- Performance shall be verified during testing.

#### **b. Usability**

- The console interface shall provide clear menus and intuitive steps.

- **Users shall complete main tasks in fewer than 5 steps.**

**e. Portability**

- **The system shall run on Windows and Linux.**
- **Portability shall be tested before final submission.**