# Functional Requirements:

* **Profile Management**
  + Students must be able to create and update their profiles
  + Profiles must store the student name, Clemson email, and a list of enrolled courses

# Course Management

* + Students must be able to add courses from their profile.

# Availability Management

* + Students must be able to add weekly availability slots (e.g., Mondays 3–5 PM)

# Study Buddy Matching

* + The system must suggest potential study partners who share at least one course and have overlapping availability
  + Suggestions must display student names, shared courses, and available times

# Non-Functional Requirements:

* **Usability**
  + The interface must be simple and intuitive, requiring no prior technical knowledge

# Reliability

* + Data must persist across sessions using a local database or storage file.
  + System failures (e.g., crashes) must not cause permanent data loss.

# Security

* + Each student should only be able to modify their own profile.
  + Email addresses must be stored securely and not visible to non- matched students.

# Portability

* + The app must run on Windows, macOS, and Linux

# Constraints:

* The system must be developed strictly using the Waterfall model (requirements frozen after SRS).
* Must be implemented as either a command-line or simple web-based app in a language chosen by the group (e.g., Python, Java, or JavaScript).
* The system will not integrate with Clemson’s official student systems (iRoar, Canvas).
* The system is intended for small-scale use (classroom projects), not enterprise deployment.

# Scope and context:

* The Study Buddy app provides Clemson students with a platform to manage availability and schedule study sessions. It supports profile creation, course tracking, availability input, match suggestion, and session confirmation.
* The app operates as a standalone scheduling tool for students. It assumes manual data entry. The primary interaction is between the student user and the system, with the system generating matches and session confirmations.

**Users and use cases:**

* **Primary User: Clemson Student**

1. **UC1: Create Profile**
   * Actor: Student
   * Goal: Create a new profile with personal info and course list.

# UC2: Manage Availability

* + Actor: Student
  + Goal: Add availability slots.

# UC3: Find Matches

* + Actor: Student
  + Goal: View suggested study buddies with overlapping courses and times.

# UC4: Add Courses

* + Actor: Student
  + Goal: Add courses to student profile so that you can find other students in the same class