Functional Requirements:

- **Profile Management**
  - Students must be able to create, update, and delete their profiles
  - Profiles must store the student name, Clemson email, and a list of enrolled courses
- **Course Management**
  - Students must be able to add and remove courses from their profile.
- **Availability Management**
  - Students must be able to add weekly availability slots (e.g., Mondays 3–5 PM)
  - Students must be able to remove or modify previously set availability
- **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
- **Session Scheduling**
  - Students must be able to request a study session with another student
  - The other student must be able to confirm or decline the request
  - Confirmed sessions must be saved and viewable
- **Session Management**
  - Students must be able to view all upcoming confirmed study sessions
  - Students must be able to cancel sessions if needed

Non-Functional Requirements:

- **Usability**
  - The interface must be simple and intuitive, requiring no prior technical knowledge
- **Performance**
  - Match suggestions must be generated within 2 seconds
  - The system must support at least 100 student profiles without noticeable slowdown
- **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 and relies on local storage (e.g., SQLite database or JSON file). 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.
2. **UC2: Manage Availability**
    - Actor: Student
    - Goal: Add or remove availability slots.
3. **UC3: Find Matches**
    - Actor: Student
    - Goal: View suggested study buddies with overlapping courses and times.
4. **UC4: Schedule Session**
    - Actor: Student
    - Goal: Send a request to meet; another student confirms.
5. **UC5: Manage Sessions**
    - Actor: Student
    - Goal: View or cancel confirmed study sessions.