**1. Introduction**

**1.1 Purpose**

The goal of this project is to develop a scheduling application for Clemson University students. The app will allow students to create profiles with their enrolled courses, find classmates in those courses, and schedule study sessions. The system will simplify collaboration, improve peer-to-peer learning, and help students manage their time better.

**1.2 Scope**

The system will be a command-line. It will support:

* Student’s course enrollment.
* The ability to add and remove availability.
* The ability to search for classmates enrolled in the same courses.
* Suggesting study matches based on overlapping availability.
* Confirming meetings between two or more students.

The system will not include:

* An integration with official Clemson course databases.
* Mobile app deployment.
* Advanced calendar syncing (e.g., Google Calendar).

**1.3 Definitions, Acronyms, and Abbreviations**

* **App**: Study Buddy Scheduling App
* **User**: Clemson student using the system
* **Availability**: Timeslots when a student is available to study
* **Match**: Suggested study partner with overlapping availability

**2. Overall Description**

**2.1 Product Perspective**

This is a standalone software application developed as a course project. Users will manually enter their course lists and availability.

**2.2 User Characteristics**

* All users are Clemson students.
* Basic computer literacy is assumed.
* No prior technical knowledge required.

**2.3 Constraints**

* Must be completed within the course timeline.
* Implemented using a single programming language (Python).
* Will run on a standard PC or laptop with an internet connection.

**2.4 Assumptions and Dependencies**

* Students are truthful when entering their courses and availability.
* Students are willing to schedule outside of official university systems.

**3. Specific Requirements**

**3.1 Functional Requirements**

1. **User Profile Management**
   * FR1: The system shall allow users to create a profile with their name and courses.
   * FR2: The system allows users to update or delete their profiles.
2. **Availability Management**
   * FR3: The system shall allow users to add available study times.
   * FR4: The system shall allow users to remove available times.
3. **Matching and Scheduling**
   * FR5: The system shall allow users to search for classmates enrolled in the same course.
   * FR6: The system shall suggest matches based on overlapping availability.
   * FR7: The system shall allow users to send/accept/decline study session requests.
4. **Meeting Confirmation**
   * FR8: The system shall confirm a study session when both students agree.
   * FR9: The system shall display scheduled study sessions to users.

**3.2 Non-Functional Requirements**

* NFR1: The system shall store user data securely.
* NFR2: The system will be easy to use with clear prompts and instructions.
* NFR3: The system shall respond to user inputs within 2 seconds.

**4. System Models (to be expanded in Design Phase)**

* Use case diagram (profiles, availability, matching, scheduling).
* Sequence diagram (requesting and confirming a study session).
* Class diagram (User, Course, Availability, Session).

**5. Acceptance Criteria**

* A student can create a profile with at least one course.
* A student can add and delete availability.
* A student can search for at least one classmate.
* The system suggests at least one match if an overlap exists.
* A student can confirm a session and see it listed.