**x1. Introduction**

* **1.1 Purpose**: This document specifies the complete requirements for the "Study Buddy" command-line application. It is intended for project stakeholders, designers, and developers.
* **1.2 Scope**: The application will allow Clemson students to create profiles, list their courses, manage their availability, and schedule study sessions with classmates. The system will be a standalone command-line interface (CLI) application and will not require external network connectivity.
* **1.3 Definitions**:
  + **User**: A Clemson student using the application.
  + **Session**: A scheduled study meeting between two or more users.
  + **Availability Slot**: A specific time block (e.g., Monday 2-3 PM) when a user is free to study.
  + **CLI**: Command-Line Interface.

**2. Overall Description**

* **2.1 Product Perspective**: This is a new, self-contained software product. Data will be managed in memory during runtime and will not persist between application restarts.
* **2.2 Product Functions**:
  + User profile and course management.
  + Availability scheduling.
  + Automated matching of study partners based on course and availability.
  + Session creation and confirmation.
* **2.3 User Characteristics**: Users are Clemson University students who are comfortable using command-line applications.
* **2.4 Constraints**:
  + The application must be developed as a Python command-line application.
  + The system shall not require a persistent database; all data is session-based.
  + Development must strictly follow the Waterfall model.

**3. Specific Requirements**

* **3.1 Functional Requirements**
  + **FR-1: User Profile Creation**
    - **Description**: The system shall allow a new user to create a profile. The profile must include a unique username, a password, and the user's name.
    - **Inputs**: Username, password, full name.
    - **Processing**: The system shall validate that the username is not already taken.
    - **Outputs**: A confirmation message of successful profile creation or an error message if the username exists.
  + **FR-2: User Login**
    - **Description**: A registered user must be able to log in to the system.
    - **Inputs**: Username, password.
    - **Processing**: The system shall authenticate the user's credentials against stored records.
    - **Outputs**: Access to the main menu upon success; an error message upon failure.
  + **FR-3: Course Management**
    - **Description**: A logged-in user shall be able to add and remove courses from their profile. Courses are identified by a code (e.g., "CPSC 3720").
    - **Inputs**: Course code.
    - **Processing**: The system will add/remove the specified course from the user's list.
    - **Outputs**: Confirmation of the action.
  + **FR-4: Availability Management**
    - **Description**: A logged-in user shall be able to add and remove availability slots for studying. A slot is defined by a day of the week and a 1-hour time block (e.g., Wednesday, 15:00-16:00).
    - **Inputs**: Day, start time.
    - **Processing**: The system will add/remove the time slot from the user's availability schedule.
    - **Outputs**: Confirmation of the updated availability.
  + **FR-5: Suggest Study Matches**
    - **Description**: A logged-in user shall be able to request a list of suggested study partners.
    - **Inputs**: A course code for which to find partners.
    - **Processing**: The system shall identify all other users enrolled in the same course who have at least one overlapping availability slot.
    - **Outputs**: A list of usernames of potential study partners.
  + **FR-6: Schedule a Study Session**
    - **Description**: A user shall be able to schedule a study session with a suggested partner for a specific course at a mutually available time.
    - **Inputs**: Partner's username, course code, day, and time.
    - **Processing**: The system will verify that both users are free at the specified time and are enrolled in the course. It will then create a pending session invitation.
    - **Outputs**: Confirmation that the session invitation has been sent.
  + **FR-7: View and Confirm Meetings**
    - **Description**: A user shall be able to view their pending and confirmed study sessions. They must be able to confirm or decline pending invitations sent to them.
    - **Inputs**: User selection to view sessions, and selection to confirm/decline an invitation.
    - **Processing**: The system will update the status of the session from 'Pending' to 'Confirmed' or 'Declined'.
    - **Outputs**: A list of sessions and their statuses.
* **3.2 Non-Functional Requirements**
  + **NFR-1: Usability**: The CLI must be menu-driven and provide clear, unambiguous instructions to the user.
  + **NFR-2: Performance**: System responses to user inputs must be virtually instantaneous (under 1 second).
  + **NFR-3: Reliability**: The application should not crash due to invalid user inputs. Error handling must be implemented for all user entry points.
  + **NFR-4: Security**: User passwords should not be displayed in plain text on the screen during input.
* **3.3 Interface Requirements**
  + **3.3.1 User Interface**: All interaction will be through a text-based command-line interface. The system will present numbered menus for navigation.