FocusForge
Manvitha P. K. and Hrehaan M.

PROJECT PROBLEM STATEMENT

Many high school students struggle with maintaining focus due to distractions and poor time management habits. While there are many productivity apps available, most are cluttered or not suited to student needs. **FocusForge** is a simple, lightweight, offline Pomodoro-style timer built in Java that helps students stay focused through structured work sessions and breaks.

CONCISE DESIGN OVERVIEW

- **Pomodoro Timer:** Console-based timer that runs 25-minute focus sessions followed by 5-minute short breaks and a 15-minute long break after every four sessions.
- **Session Counter:** Tracks the number of completed Pomodoro cycles in a day.
- **Basic Menu System:** Start session, reset session counter, exit.

SCOPE STATEMENT

INCLUDES:

- Console-based application (no GUI)
- Basic Pomodoro timer (25/5/15 model)
- Manual start/stop/reset functionality
- Simple session tracking (in-memory only)
- Fully offline use

EXCLUDES:

- GUI, mobile, or web version
- Data saving or file persistence
- Notifications or alert sounds
- Task management features

SMART Goals:

- The initial goal is to **build the basic Pomodoro timer functionality** by **May 22**. This includes implementing a countdown for focus sessions and breaks using standard Java features.
- By **May 30**, the project should include additional features such as a **session counter** that tracks completed Pomodoros, along with a **menu system** that allows users to start a session, reset the counter, or exit the program.
- The goal for **June 10** is to **fully test and debug** the application. This involves testing all menu options, timer transitions, and handling edge cases like invalid inputs.
- Finally, the completed and tested application will be **presented and demonstrated live** by either **June 17 (for 11th graders)** or **June 3 (for 12th graders)**, depending on the class schedule.

TIMELINE

Date	Task	Description
5/13	Set up project files	Create main class and timer structure
5/15	Implement timer logic	Countdown timer for work/break periods
5/17	Add session tracking	Count completed Pomodoros, track break type
5/20	Build menu interface	Options to start, reset, or quit
5/22	MVP complete	Pomodoro timer fully working
5/24	Manual testing	Test different edge cases and timing logic
5/30	Polish + error handling	Add input validation, clean up output
6/10	Full testing + dry run	Simulate demo and check all features
6/17 or 6/3	Final presentation	Live demo of working app

TEST PLAN

Manual Testing (No automation)

- **Positive Tests**:
  - Start a Pomodoro session and verify the timer runs to 0.
  - Verify short break triggers after 1 session.
  - Verify long break after 4 sessions.
  - Reset session counter and confirm reset.
- **Negative Tests**:
  - User inputs invalid menu options (e.g., letters instead of numbers).
  - Try starting a timer while one is already running.

RISKS AND CONTINGENCY PLANS

Risk	Probability	Mitigation
Bugs in timer countdown	Medium	Use Thread.sleep() carefully and isolate timer logic for testing

Incorrect session tracking	Low	Create and test logic in small chunks; print current state after each step
Time constraints or schedule conflicts	Medium	Divide tasks evenly between team members and meet regularly for updates
Unexpected errors (e.g., input crash)	Medium	Add input validation and catch exceptions during input handling