Pomodoro App Requirements Document

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to outline the requirements, design, and implementation plan for a Pomodoro productivity tracking application. The application will help users track their work sessions and breaks, providing statistics to improve productivity.

## 1.2 Scope

This application will consist of a command-line interface (CLI) for initiating Pomodoro sessions, a web interface for monitoring and statistics, and deployment in a Docker container for ease of deployment and scaling.

# 2. User Personas

## 2.1 John Doe - The Busy Professional

Age: 35  
Occupation: Marketing Manager  
Needs: Track productivity, improve work habits, visualize time spent on tasks  
Tech Skill: Intermediate

## 2.2 Jane Smith - The Student

Age: 22  
Occupation: University Student  
Needs: Manage study sessions, balance study and break time, analyze study patterns  
Tech Skill: Beginner

# 3. User Stories

1. As a user, I want to start a Pomodoro session from the CLI, so that I can begin working on a task.  
2. As a user, I want to be notified when a Pomodoro session ends, so that I can take a break.  
3. As a user, I want to view my productivity statistics on a web page, so that I can analyze my work patterns.  
4. As a user, I want to track my completed Pomodoro sessions, so that I can see my progress over time.

# 4. Functional Requirements

1. Start a Pomodoro Session (CLI)  
 - Input: Task description, session duration  
 - Output: Timer starts, notification when session ends  
2. End a Pomodoro Session (CLI)  
 - Input: Session end command  
 - Output: Session stops, log entry created  
3. View Statistics (Web)  
 - Input: User login  
 - Output: Display charts and graphs of completed sessions, break times, and productivity trends  
4. User Authentication (Web)  
 - Input: Username, password  
 - Output: User login, session management  
5. Docker Deployment  
 - Input: Dockerfile and Docker Compose configuration  
 - Output: Deployable Docker containers for CLI and web interface

# 5. Non-Functional Requirements

1. Usability  
 - Simple and intuitive CLI commands  
 - User-friendly web interface  
2. Performance  
 - Minimal latency in command execution  
 - Quick loading of web pages  
3. Reliability  
 - Robust error handling in CLI and web interface  
 - Consistent uptime for Docker deployment  
4. Scalability  
 - Ability to handle multiple users concurrently  
 - Efficient resource usage in Docker containers

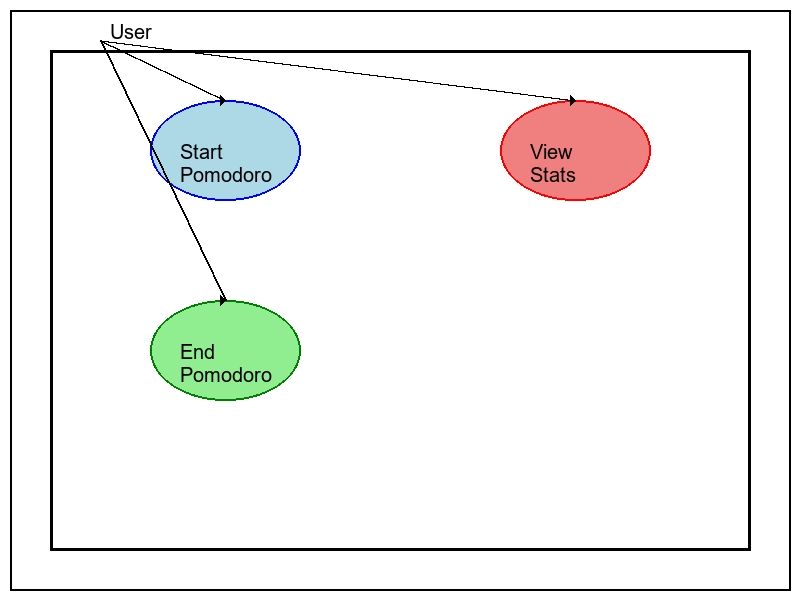
# 6. System Architecture

## 6.1 System Components

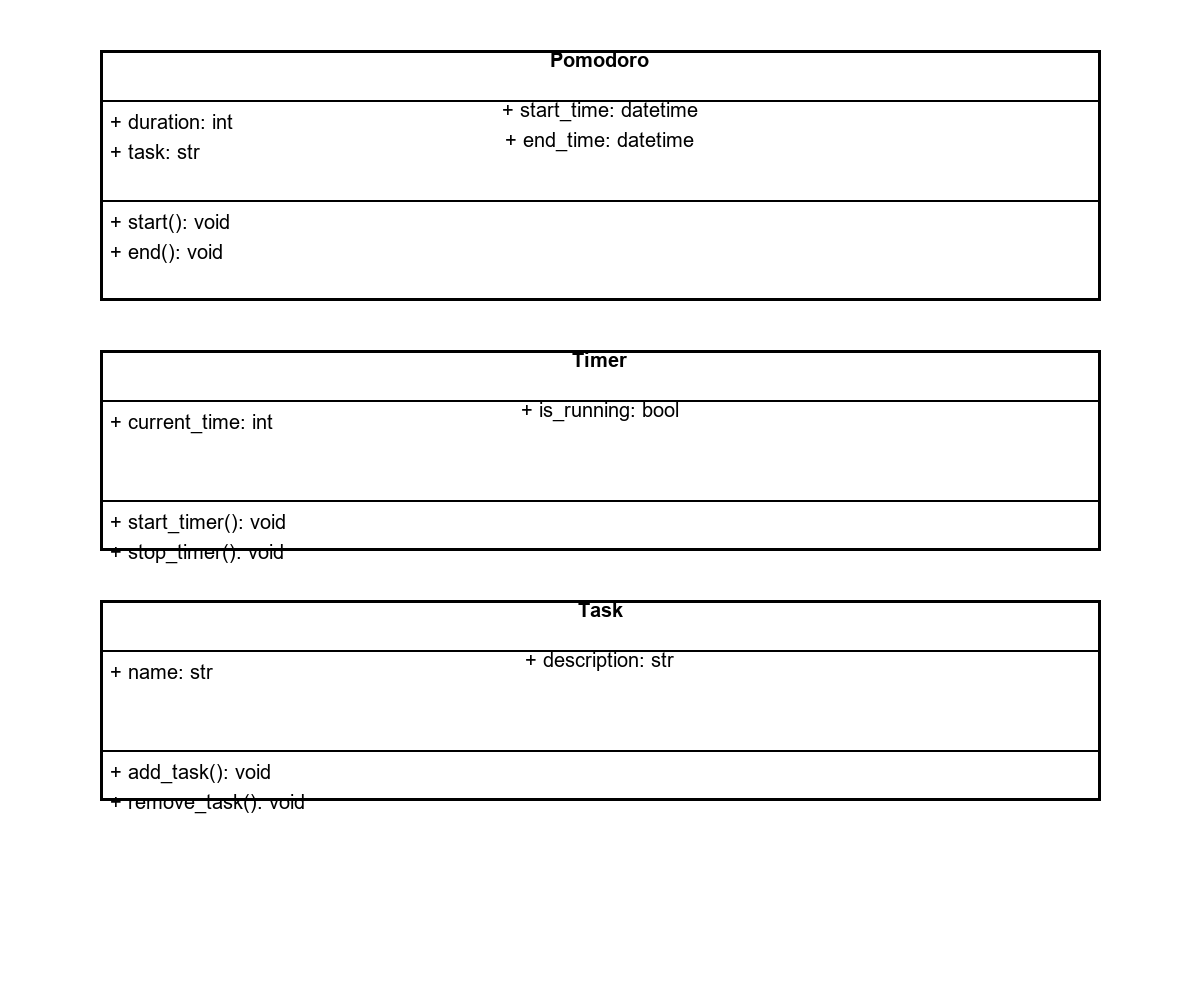
CLI Application: Python-based command-line interface for managing Pomodoro sessions.  
Web Application: Flask-based web interface for displaying statistics.  
Database: SQLite for storing user data and session logs.  
Docker: Containerization of both CLI and web application for deployment.

## 6.2 UML Diagrams

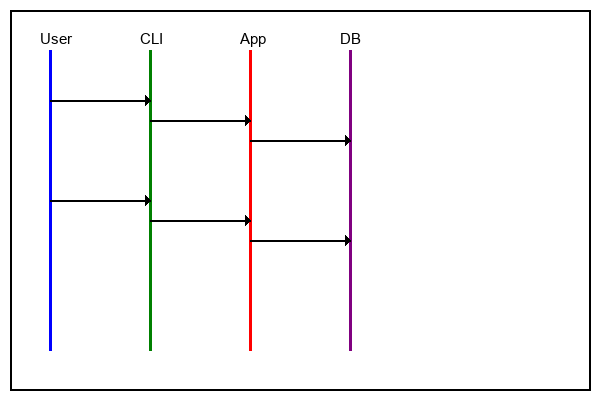
### Use Case Diagram



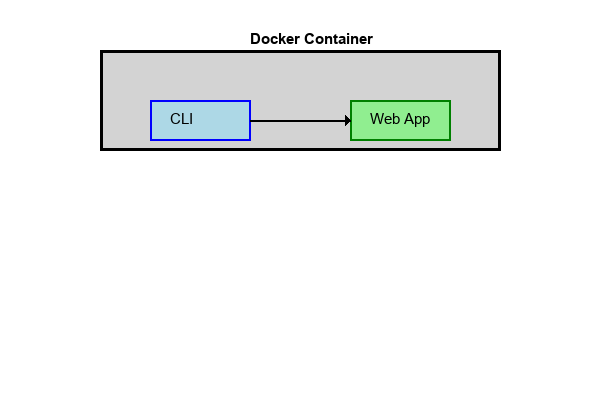
### Class Diagram



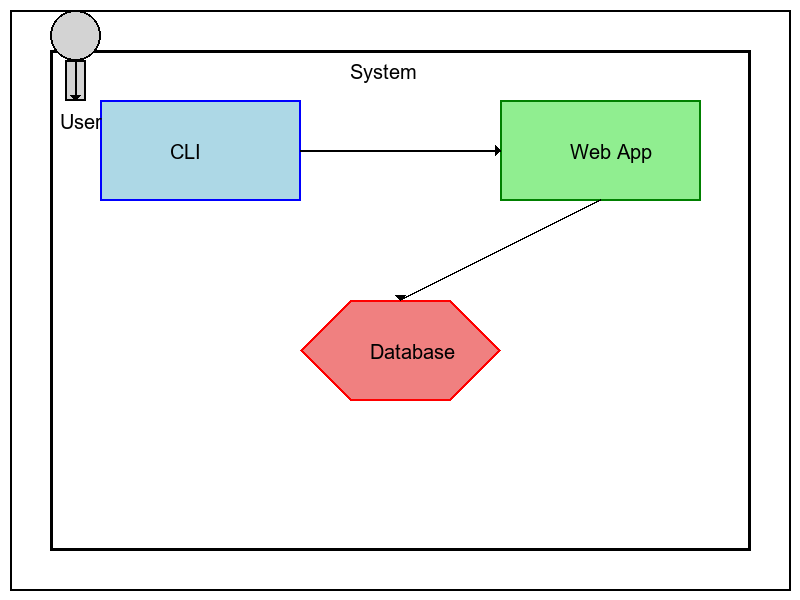
### Sequence Diagram



### Deployment Diagram



### System Diagram



# 7. Deployment Diagram

Deployment diagram showing the system architecture and components.

# 8. System Diagram

System diagram illustrating the relationship between the CLI, web application, and database.

# 9. To-Do List and Priority Order

1. Define User Stories and Personas  
 - Priority: High  
 - Status: Completed  
2. Gather Functional and Non-Functional Requirements  
 - Priority: High  
 - Status: Completed  
3. Create UML Diagrams  
 - Priority: High  
 - Status: In Progress  
4. Develop CLI Application  
 - Priority: High  
 - Status: Pending  
5. Develop Web Application  
 - Priority: Medium  
 - Status: Pending  
6. Set Up Database  
 - Priority: Medium  
 - Status: Pending  
7. Implement User Authentication  
 - Priority: Medium  
 - Status: Pending  
8. Containerize Applications with Docker  
 - Priority: Medium  
 - Status: Pending  
9. Write Tests for CLI and Web Application  
 - Priority: Medium  
 - Status: Pending  
10. Deploy Applications  
 - Priority: Medium  
 - Status: Pending  
11. Debug and Test Applications  
 - Priority: High  
 - Status: Pending  
12. Launch and Monitor Application  
 - Priority: High  
 - Status: Pending