Quack Time – Productivity Assistant
Software Requirements Specifications

Neal Kimchi (nk), Nikhar Ramlakhan (nr), Areyan Rastawan (ar), Abie Safdie (as)

6-04-2024 - v1.06

Table of Contents

1. SRS Revision History

Date	Author	Description
5-18-2024	nr	Created the initial document.
5-18-2024	as	Completed the initial document.
5-19-2024	nr	Formatted the initial document.
6-2-2024	as	Edited the document
6-3-2024	as	Edited the document
6-4-2024	team	Finalized the document

2. The Concept of Operations (ConOps)

This document is the Software Requirements Specification (SRS) for the Quack Time software system. Quack Time is a computer web application that will aid in work productivity by promoting the pomodoro time-management method.

Quack Time will guide students in completing their assigned tasks by applying the pomodoro method. The pomodoro method, developed by Francesco Cirillo in the 1980s, is a time-management system that states users are most productive for 25 minutes followed by short (5 or 10 minute) breaks (Henry, 2022). For a more complete reference on what the Pomodoro method is see: https://todoist.com/productivity-methods/pomodoro-technique

The goal of the Pomodoro method is to ensure users are most productive with their time according to scientific studies. By completing 25 minutes of deep work, followed by short breaks, the user maximizes their focus and productive time based on science (To-do-ist, n.d.).

The basic concept of the Quack Time system is that it will provide a timer to run the Pomodoro method as well as a user-interface to visualize/edit user tasks, the time remaining on those tasks, and the productivity associated with each specific pomodoro session. Productivity will be self-inputted by the user. There will be a complete "proDUCKtivity" log, so the user can further visualize their productivity and form their own conclusions on how efficient they have been in their work. Furthermore, Quack Time will be geared towards college students by allowing the user to create "task boards" that are associated with their courses. Each task board will be further decomposed into the "tasks" that represent specific coursework to complete.

Task boards are designed as general categories that tasks will belong to. For example, "CS 422" or "ENG 208". Tasks are designed as general work that needs to be completed. When creating a task, a user will define the name of the task and how much time they believe the task will take to be completed. For example, "Write Essay – 25 Minutes" or "Code Project – 100 Minutes". These tasks are then bound to the task board that they associate with. Meaning, the "Write Essay" task will be bound to the "ENG 208" task board.

Furthermore, once a pomodoro session is complete (the timer expires), the user will be prompted to input their productivity. For example, if a user inputs a productivity ranking of "3" on their "Write Essay" task, the following will be displayed in the log (example in parentheses):

- Time and Data Pomodoro Session completed (6/1/2024, 3:00 PM)
- The Task Board and Task that was worked on (ENG 208 – Write Essay)
- The time spent on the task (25 minutes)
- The user-inputted Productivity ranking (3)

Subsequently, the time remaining on the "Write Essay" task will be updated and computed using the following algorithm:

- Time Remaining on task = Time Remaining on task - ((Productivity Ranking / 5) * Amount of time the timer was set to)

For example, in our previous example, it would look like the following:

- Time Remaining on task = 25 - ((3 / 5) * 25) = 10

Quack Time will store user task boards, tasks, and proDUCKtivity logs on a server using a MySQL database. This will permit the user to use multiple different machines to apply the pomodoro method to their work.

2.1. Current System or Situation

There are multiple papers stating the value of the pomodoro method for aiding in productivity. For example, The Pomodoro® Technique (Cirillo, n.d.) and Productivity 101: An Introduction to the Pomodoro Technique (Henry, 2022). Currently, there is one software system that visualizes task completion using pomodoro. It is called Pomofocus (Pomofocus.io), but it does not show the time remaining on your task, nor is it geared specifically for college students, as it does not provide a way to create "task boards".

2.2. Justification for a New System

As stated previously, the current system does not provide a way to easily visualize the time remaining on a task, create "task boards".

A new system geared towards college students is specifically needed because, according to the NIH, there is a significant increase in the number of students suffering from ADHD and other time-related productivity disorders (Hotez, 2022). Furthermore, students suffer from unstructured study and work habits. Having a way to aid and visualize productivity will help students manage their productivity.

Other systems, specifically the Pomofocus software system, have a majority of its features behind a paywall. This is troubling for college students due to their financial constraints. Furthermore, it does not provide a streamlined way to structure tasks into "course" categories. Meaning, a student would not be able to make a category called "CS 422", for example, and have a bunch of tasks bound to that category. This specific functionality is what Quack Time provides.

2.3. Operational Features of the Proposed System

The key operational features of Quack Time include: (a) creating task boards to denote courses/categories (b) creating tasks associated with each task board, (c) setting timers that follow the pomodoro method, (d) productivity ranking and visualization, (e) storage of user-inputted data on a server.

This system will specifically follow the Pomodoro method, which is described by many scientific papers as an efficient productivity management system.

2.4. User Classes

There is one user class:

1) A user who is attempting to use the Pomodoro method to aid in their productivity. The user can expect to find a system that guides them through Pomodoro to complete tasks. The user will know what tasks they neeed to complete, and an estimation of the time needed to complete that task. The user is expected to know how to navigate a website.

2.5. Modes of Operation

The system has one primary mode of operation, in which the server is running, and a user acceses Quack Time and gains client access to the server.

2.6. Operation Scenarios (Also Known as "Use Cases")

1. A UO student signs up for Quack Time to better manage their ADHD. They create a Task board for their CS 422 class. They create a task entitled "Complete SRS." They denote that this task will take 3 hours to complete. They begin their first pomodoro timer for 25 minutes and begin working on the assignment. Once the timer is complete, they rank their productivity and take a 5-minute break. This repeats until the SRS is completed.

2. A university student wants to learn a new card trick to impress their friends. They create a Task board entitled "Magic Card Tricks." They create a task entitled "Did I pick your card?" They believe the trick will take 50 minutes to learn. They begin their first pomodoro timer for 25 minutes and begin learning the card trick. Once the timer is complete, they rank their productivity and take a 5-minute break. This repeats until the user learns the new card trick.

3. An adult wants to decompress by doing Yoga but struggles to manage their time and gets easily distracted. They create a task board entitled "Yoga Poses/Practice." They create tasks called "learn new poses" and "practice old poses." They want to spend 30 minutes learning

new poses and 30 minutes practicing old ones. They start their pomodoro timers for 15 minutes and take 5-minute breaks in between. They repeat this until they have completed their yoga.

While these use cases show how the software can be adapted for various individuals and scenarios, the current intention is for the software to be used exclusively by University of Oregon students. This is because the software relies on the DuckID system for user authentication, ensuring that only UO students can sign up and access the features. The DuckID integration helps maintain a secure and exclusive environment tailored specifically for the UO community.

Thus, even though the software's functionality can be broadly applicable, it is designed with UO students in mind, leveraging DuckID for secure and streamlined user identification.

3. Specific Requirements

The basic functionality must include the following:

Functional Requirements | Absolutely Required

Externally Visible User Interactions

1. A visual field for logging in
2. A visual field for creating an account.
3. A visual field for inputting a Task board and Tasks.
4. A visual field for deleting Task boards and Tasks
5. A visual field for editing the time left on Tasks
6. A visual timer following the Pomodoro method.
7. A visual field for inputting productivity associated with each Pomodoro timer session.
8. A visual field showing productivity logs associated with each task.
9. A visual field explaining how to use the system and the pomodoro technique

Pomodoro Method

10. The program will have a “how to use” section that states how to use the software and apply the pomodoro technique
    a. This prompt will be able to be turned on and off by the user.

Login

11. There will be a need to create an account with a username and password to gain access to Quack Time.

Data Storage

12. Data will be stored on a server using MySQL

13. The initial version will have the server running on ix-dev and the client on a separate machine.

14. The system will save all user data and will never delete user data without a warning.

Non-Functional Requirements | Absolutely Required

Target Platform

15. The software will run on a laptop or desktop machine's web browser. The system will run on macOS 12.

Build-Related Constraints

All build-related constraints are considered non-functional requirements:

Absolutely Required

16. Installation Constraints

a. Instructions will be provided on how to run the code and get the system running.
b. There will not be more than 20 commands required to run the program.
c. An experienced programmer will not require more than 20 minutes working alone to set up everything necessary.
d. All code, except for approved library imports, will be submitted.
e. No server connections will be required except for mysql on ix.
f. No virtual environments, game engines, or machine learning will be used.

17. Performance Constraints

a. The timer will run effectively to the user-inputted time while the window is tabbed in and open.
b. The system will accept a maximum of 10 Task Boards and tasks. The user log will maintain an infinite number of pomodoro sessions.
c. The productivity ranking window will provide ranking from 1 to 5 to rank user productivity.

Not Absolutely Required

18. Programming Constraints

a. The following languages will be used:
    - HTML
    - JavaScript
    - CSS
    - Python
b. MySQL will be used for database storage.

4. Future Changes for a Final Version

Given more time and resources, our initial Quack Time version could be built upon with the following improvements:

- Maintain a correct timer when the window is tabbed out/closed. This would involve handling local cache requirements per OS and local Javascript runtime caches. As currently, our entire system gets put in the background and given less resources to execute by the OS when tabbed out, therefore the timer does not run to the correct precision.
- Expand the system to provide functionality to non UO students. Remove the DuckID authentication system and allow users from any university to use our productivity tool. We could use e-mail addresses to provide unique authentication instead.
- Create a more comprehensive way to manage and visualize completed tasks, so the user can receive the gratification of task completion. This could be done with a separate log showing completed tasks and information about such tasks. Such as, total time spent and average productivity.
- Create a way for students to see other students progress on tasks. Meaning, if two users share the same board and task, for example, CS 422 – Complete SRS, students will be able to see how much time other students have spent on this task and the average productivity ranking. This can be done through server communication – to pull the time spent and productivity data -- and a simple visual field showing statistics.

5. References

Cirillo, Francesco. "The Pomodoro® Technique." *The Pomodoro® Technique | Cirillo ConsultingGmbH*,web.archive.org/web/20230331051358/https://francescocirillo.com/products/the-pomodoro-technique. Accessed 18 May 2024.

Henry, Alan. "Productivity 101: An Introduction to the Pomodoro Technique." *Lifehacker*, 3 Mar. 2022, lifehacker.com/productivity-101-a-primer-to-the-pomodoro-technique-1598992730.

Hotez, Emily, et al. "A National Cross-Sectional Study of the Characteristics, Strengths, and Challenges of College Students with Attention Deficit Hyperactivity Disorder." *Cureus*, U.S. National Library of Medicine, 23 Jan. 2022, www.ncbi.nlm.nih.gov/pmc/articles/PMC8865363/.

Pomofocus. (n.d.). https://pomofocus.io/

The pomodoro technique - why it works & how to do it. Todoist. (n.d.). https://todoist.com/productivity-methods/pomodoro-technique

6. Acknowledgements

This SRS builds on the template from https://classes.cs.uoregon.edu/24S/cs422/Templates.html