Segfaulters - Software Requirements Specification Document

Isha Devgan, Michael Kopczynski, Stuti Shah, Anushka Trivedi

December 2024

1 Introduction

This Software Requirements Specification document is intended for the development team, project managers,and software engineers (end users) involved in the PomBot project. It will serve as a guide to ensure alignment across all phases of development. For developers, it provides detailed requirements and constraints to guide design of feature implementation. Project managers and product owners will use it to plan project timelines, tasks, and budget.

In this software requirement specification, we will discuss the scope and value of our product, highlighting the productivity benefits of using PomBot. We also dive into the broad intended audience and intended use of the PomBot.

2 Product Scope

The PomBot is a Pomodoro timer extension tool that addresses the universal problem of staying focused when working. Utilizing the Pomodoro technique within a user-friendly Chrome extension, our product is an accessible and applicable tool for anyone looking to improve their productivity. This project aims to reduce mental fatigue and prevent burnout, allowing users (like software engineers) to work with sustained focus and improve task completion. With these goals in mind, this project will ultimately change how software engineering teams manage their time, and help more teams meet their targets. It will

3 Product Value

In a field as demanding as software engineering, task productivity is incredibly important. Software engineers are expected to constantly produce good work in rapid development cycles. However, with both the demand and the amount of work, burnout is extremely common among those in the software engineering field. The PomBot aims to aide software engineers in effective task management that forces mental breaks and breaks tasks into manageable chunks. The tool

encourages accountability and offers value to organizations focused on employee well-being and productivity.

4 Intended Audience

Our intended audience is primarily software engineers, but is open-source and can be used by any and everybody who works at a computer whether that be software engineers needing a structured work/break cycle, students who want a proven time management technique, or remote workers who need encouragement to balance productivity and rest. Since our product integrates seamlessly with Google Chrome, there is no limit the the number of users who would benefit from using it.

5 Intended Use

The main purpose of PomBot is to facilitate personal time management for software Engineers. Using our product, it is easy to break work down into focused intervals with regular breaks improving productivity and reducing burnout. On top of the timer function, our product allows users to stay organized and plot out their workday, facilitating an organized approach to study sessions, work tasks, or any activity they want to focus on.

The PomBot should be used not as a tracking tool by management, but as a team wide tool to ensure productivity and meeting team goals. For example, a developer can use a personal PomBot to stay on track for their task for the day, or a manager can set a team-wide PomBot for the purpose of a Sprint planning session. Our PomBot can be used in any setting with the goal of enhancing productivity.

6 General Description

PomBot is a Chrome Extension that is an all in one Pomodoro timer, task scheduler, and break enforcer. Like many popular programs, PomBot will live in the extension hub of the Google Chrome window, and with one click the user will be brought to their Pomodoro session hub. Here, the user is able to create or manage their Pomodoro session.

When creating a Pomodoro session, the user has the option to give their session a name, select a custom Pomodoro length, and select either a team-wide (input team member emails) or individual Pomodoro session. Once the user selects this, they will be prompted to write a list of tasks for each Pomodoro session, encouraging an intentional and achievable schedule. Once this step is completed, they can begin their Pomodoro session. A team-wide Pomodoro session will send a summary of the tasks/breaks completed when the Pomodoro Session is over to each team member by email.

There will also be the option to edit notification/timing settings in a settings sidebar or request a recommended interval based on previous session times. This functionality allows the user some flexibility in how they want to interpret/apply the Pomodoro technique.

Once the user completes one Pomodoro stretch, PomBot will send a notification that it is time to take a break. Before the next session starts, it will remind the user of the tasks they have set to complete during the next Pomodoro. Managing the Pomodoro timer is as easy as clicking the PomBot extension. Once clicked, the user can view how much time is left in the current session, and is are able to pause and resume if an impromptu break is necessary. The user can additionally edit the list of tasks as they work on each deliverable.

7 Constraints

7.1 High-Level Design Constraints

High-level constraints for the PomBot focus on the foundational features that define its functionality: the timer, session manager, and notification system. These components must integrate seamlessly to ensure a cohesive user experience. For example, the timer service must trigger the notification service to remind users of breaks, while the session manager dictates the timer service and how long sessions last. Additionally, PomBot must adhere to Chrome API requirements as a Chrome Extension. This dictates permissions on how it manages the chrome browsing experience and storage.

7.2 Low-Level Design Constraints

As a Google Chrome extension, PomBot must be implemented with JavaScript, HTML, and CSS, the standard for web applications.

PomBot's session history storage requires efficient database or local storage implementations, allowing for users to parse through previous sessions. local storage may be used for offline capabilities, while cloud-based storage would be used for online storage and, with limited free storage for users.

The team Pomodoro session functionality will depend on internet services and an email API, such as Gmail API, to ensure synchronized use between multiple users as well as delivering team summaries.

7.3 UI Design Constraints

The main goal of the PomBot is to enhance focus and productivity. This requires that the UI of PomBot be simple and subtle while communicating the necessary information needed for use.

Many Pomodoro timers use distracting timers or require users to stop working on their task in order to check the time remaining. To remedy this, our timer will be a discreet circular timer that replaces the PomBot logo in the extension

hub while in session. This way, the timer is non-obtrusive and 'glanceable' allowing users to keep the focus on their task, not the timer.

When clicking the PomBot extension, users are taken to the session manager, designed to integrate into the workday seamlessly. The interface must be simple and allow for sessions to start with a single click. However, there must be customization for the session to set your desired session time and task lists, presets which would be saved for quick-starting future sessions.

8 Test Plan

8.1 Overview

This test plan details the priorities, strategies, and methodologies for testing the PomBot application. The purpose of this section is to ensure that all functionality is properly and thoroughly tested via the requirements outlined below, ensuring that the application achieves its intended purpose.

8.2 Objectives

To achieve the goals outlined for PomBot in this document, the following testing objectives have been identified:

- PomBot must be compatible with the newest version of Chrome and various webpages
- PomBot's primary Pomodoro functionality must operate as intended, including timers and notifications.
- PomBot's UI must maintain a discreet and non-distracting profile through minimization and seamless integration.
- PomBot must integrate with external services such as email notifications and multi-user functionality.

8.3 Strategy

To achieve the objectives of the test plans, there will be three types of tests to evaluate: functional tests, UI tests, and integration tests.

8.3.1 Functional Testing

This section focuses on testing the core functions that PomBot offers, including Pomodoro timers, session management, and notifications. Without these functions, PomBot would not be usable and thus thorough testing is imperative.

Unit testing will be used to verify these functions perform as expected, ensuring accuracy and consistency in their outputs. These tests cover different scenarios such as:

- Verifying timer accuracy for different session lengths.
- Ensuring tasks can be created, updated, and deleted within the task manager.
- Testing notification triggers for both session completion and break reminders.

8.3.2 UI Testing

This section focuses on the UI design of the PomBot application, including the visual timer, session management screen, and notifications. These portions of the program ensure that user experience is effective and non-distracting as discussed in our UI Design Constraints.

Manual Testing will be used to verify these sections appear and respond as intended. These tests cover different scenarios such as:

- The visual timer icon, if enabled, appears upon starting a work session.
- The visual timer counts down in accordance to the actual time left in the session.
- Notifications and the session manager are visible across different web pages.
- Ensuring UI appears as intended across different window sizes.

8.3.3 Integration Testing

This section focuses on the integration aspects PomBot has to offer, including the multi-user Pomodoro sessions and email functionality. These portions of the program take PomBot to the next level of convenience and gives reason to use the application over alternatives.

Hybrid Testing, a mix of different testing strategies, will be employed to verify these functionalities. Unit tests, workflow tests, and manual tests would ensure that these integration works as intended. These tests cover scenarios such as:

- Users are able to join a public team session
- Emails are sent to each user upon a team session completing
- Ensure synchronization across team members