Software Requirements Specification

For

System Performance Tracker

Prepared by
Srijan Badhya (PES1UG21CS616)
Sudeep Dhotre (PES1UG21CS631)
Srikrishna Sripathi Nayak (PES1UG21CS620)
Surya Kumarak Kannan (PES1UG21CS649)

17 October 2023

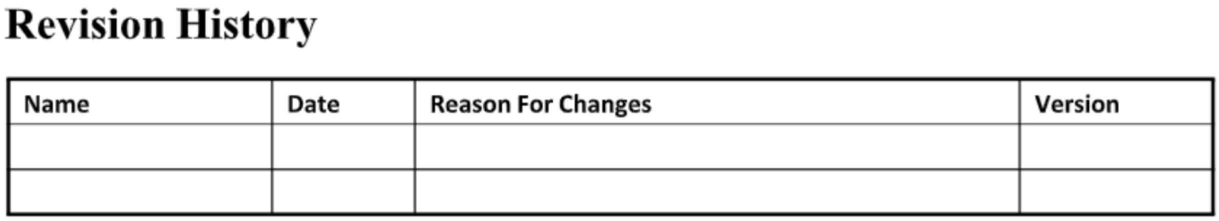
Table of Contents

6.3 Security Requirements

6.4 Software Quality Attributes

6.5 Business Rules

7. Other Requirements



Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The system performance tracker is designed to be used by computers with a Windows system. The Windows version is expected to be 10 and above. The software will act upon a device with these specifications, and do its job.

1.2 Intended Audience

The Intended Audience is for users who want all their computer's details in one place. Anybody with a Windows computer will be able to make use of this software because even the common man needs to be prepared for low battery or high memory usage before the catastrophe strikes and the battery runs out before the user starts searching for a charging port. People at the individual level and the company level will find this software useful.

1.3 Product Scope

The user will be able to track aspects of their system like battery, RAM usage, and memory usage, and will be given alerts to remedy any potential problems. Like a human fitness tracker, the system performance tracker will help the user keep their computers in good health. We aim to provide the user with accurate information and suggestions so that they can make better decisions to preserve their computer and use its abilities efficiently.

2. Overall Description

2.1 Product Perspective

This is a new software, the first of our group, which consolidates existing Windows features. Although the user can hunt and find their system details, the details are stored in separate places and they must know how to navigate settings. Our product offers a much more convenient process, where everything is displayed on the screen and accessible to them. All the Windows data will be brought together and presented in a clean readable format.

2.2 Product Functions

CPU usage, memory usage, battery status, list of processes running, and system information.

2.3 User Classes and Characteristics

Users must be of an age where they can safely operate a computer. Users who spend more time using the computer would find this product more useful. Tech experts and repairmen will be able to use the data of this product to judge the performance of the computer and see what upgrades would benefit the computer the most. Users who have a hard time keeping track of their battery will now be able to do it better. At both the individual level and the company level, this product will be relevant.

2.4 Operating Environment

This product will peacefully coexist with most other software. It requires permissions to access system data, and hence needs to establish its presence with antiviruses and confirm that it is not malware. Any computer which can support Windows 10, will also be able to support this product. This is run on Python code, which should be recognized by the system. Operations will be parallelly executed.

2.5 Design and Implementation Constraints

For the product to run successfully on the user's system, around 10 MB of free space will be required at the maximum. Permissions must be acquired to access system data and device state. Python code may be slow and have a large runtime, and this should also be recognized by the system.

2.6 Assumptions and Dependencies

We assume that the Python code will run without a hitch. A small amount of RAM must be devoted to this software which will routinely run. If the computer is in low power mode, this software may shut down.

3. External Interface Requirements

3.1 User Interfaces

There will be a dashboard for the user to view. This dashboard will display certain details to the user. We plan to implement using Python's GUI, tkinter. Buttons that the user can click, will lead to more information. Tkinter will create a new window for this software.

3.2 Software Interfaces

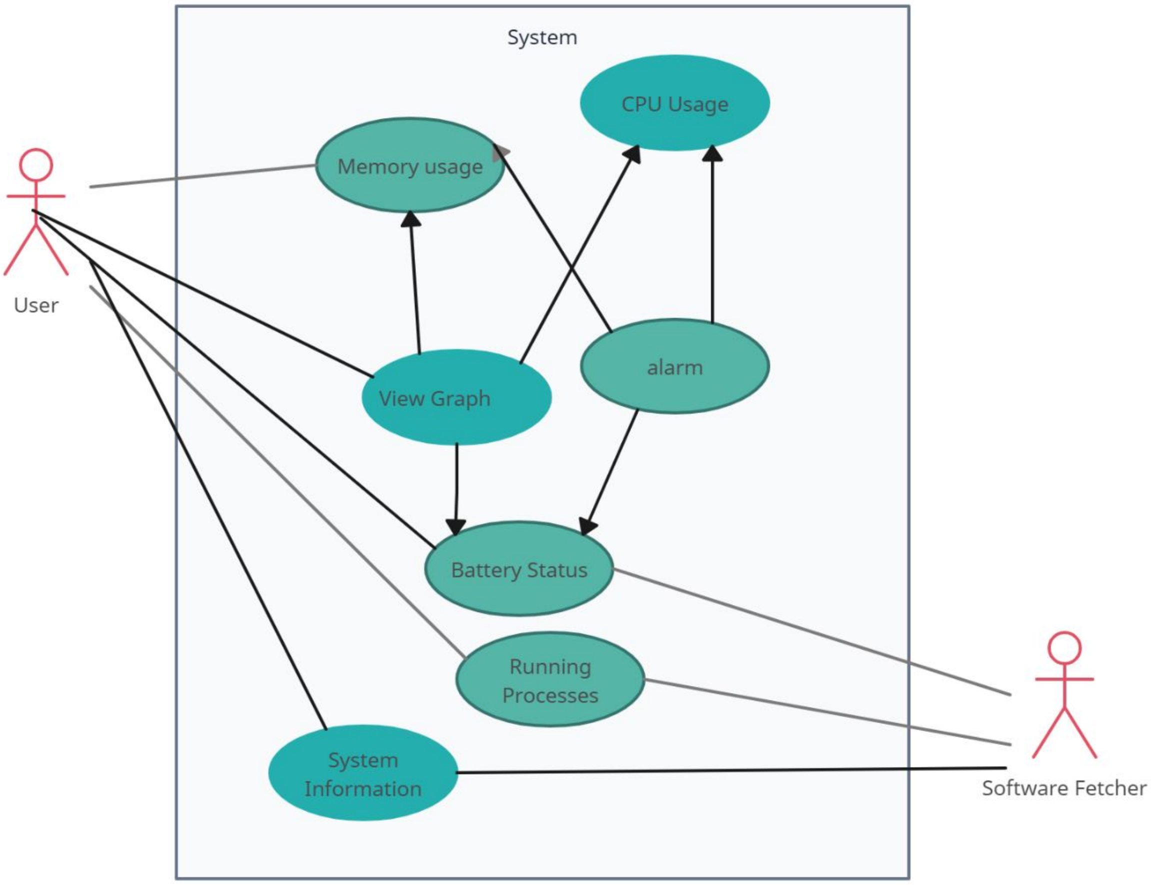
This product will use the standard systems library to read the information of the device on the operating system Windows 10. The computer's data will frequently be read by the product, and the product will make sense of this information in a user-readable format. Some information will be read at the first start, like the device specifications, which are unlikely to change.

3.3 Communication Interfaces

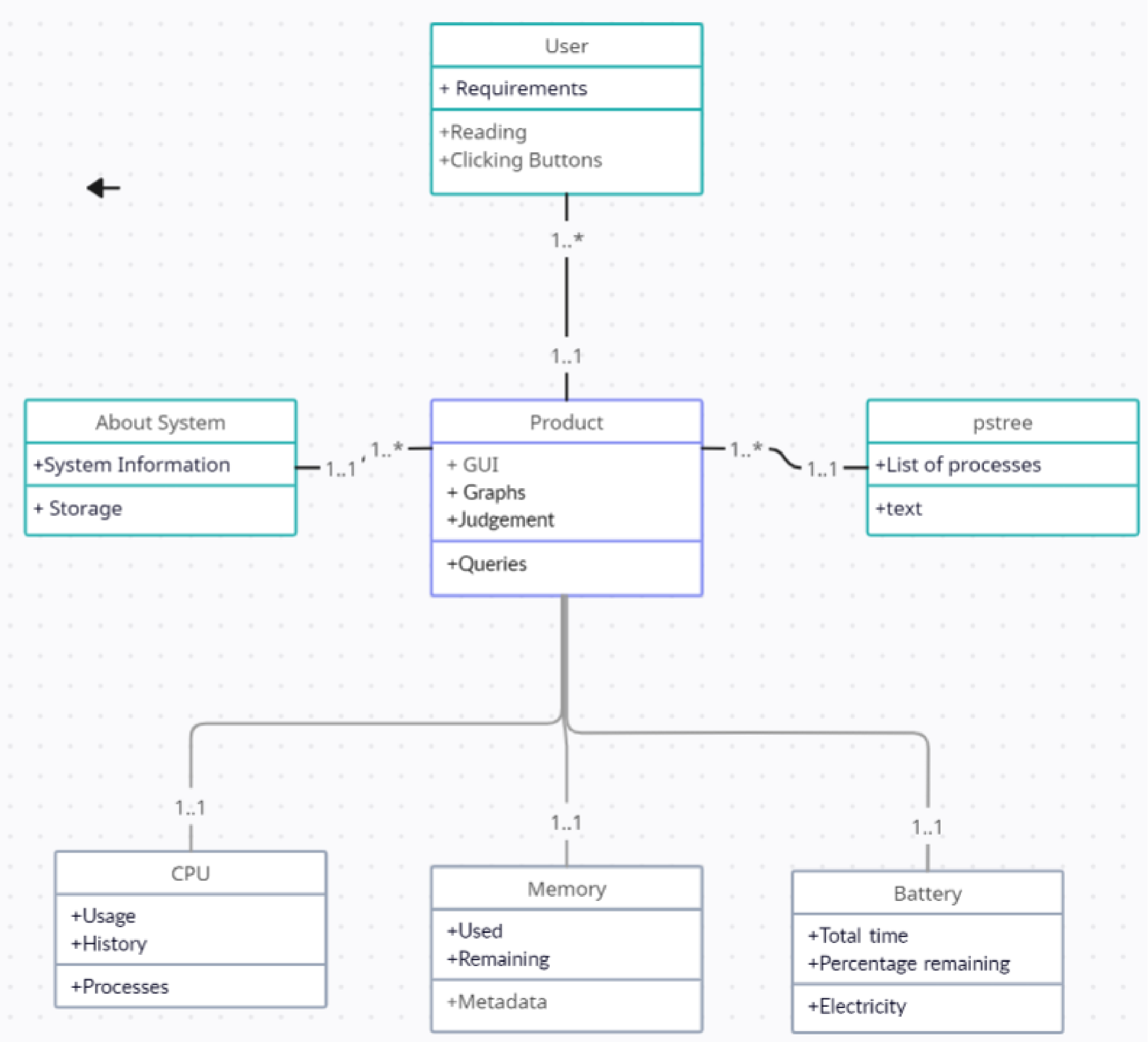
This device does not require to transmit data across the internet, because it will be self-sufficient. In future versions it can be expanded to consolidate data of different devices, but there are no plans as of now.

4. Analysis Models

Use Case Diagram:



Class Diagram:



5. System Features

5.1 CPU Usage

Low priority. The product will query the CPU and read the response. Measuring the CPU usage will show the user if they are taxing the CPU more than it can handle. High CPU usage shows a need for better optimization, or an upgrade of the CPU to handle all the tasks if they are necessary. Functionally the CPU data is required.

5.2 Memory Usage

High Priority. The product will search and measure all the total data of the user's system. After knowing the size of the regular memory, it will be able to calculate how much space is left. If the remaining space is extremely low, it will issue a warning to the user to delete unnecessary files. The size of the data of the user is functionally required.

5.3 Battery Status

High Priority. The product will access the battery information and calculate how much time is left. It is very important for the user to instantly give attention to this warning because otherwise the computer may lose power and the software including this product will be forced to shut down. The battery information is functionally required.

5.4 List of processes running

Medium Priority The list of processes running will show the foreground processes which the user can see, and the background process which the user may not be aware of. These tasks would be performing in parallel. Some processes which the user is not aware of, will be unnecessarily running in the background even when the user has assumed that they closed the application. Closing these applications properly will lengthen battery life since unnecessary tasks will not be running. Functionally the pstree will be accessed which has all the required information.

5.5 System information

Low priority. This information is not expected to change between most times of access, since computer system specifications remain the same unless the user intentionally changes it. This information would help to calculate all the previous information. Functionally the device specifications would be required.

6. Other Non-Functional Requirements

6.1 Performance Requirements

Under standard conditions, the software is expected to execute quickly with negligible latency. Even if many other processes are running on the system, due to the low processing power needed for this software, it is expected to still function smoothly with quick response.

6.2 Safety Requirements

Overheating of the system due to this product is not likely to occur due to the low burden that the system experiences. There are no disastrous consequences expected from the usage of this product.

6.3 Security Requirements

The software requires permissions to read the machine state and extract device data. These are not for nefarious purposes, so it must be communicated to the anti-virus software about the functioning of this product. Privacy of the user will also be maintained.

6.4 Software Quality Attributes

This product focuses on accessibility of the information to the user. It does also adhere to a high standard of affordability, reliability, portability, adaptability, and usability.

6.5 Business Rules

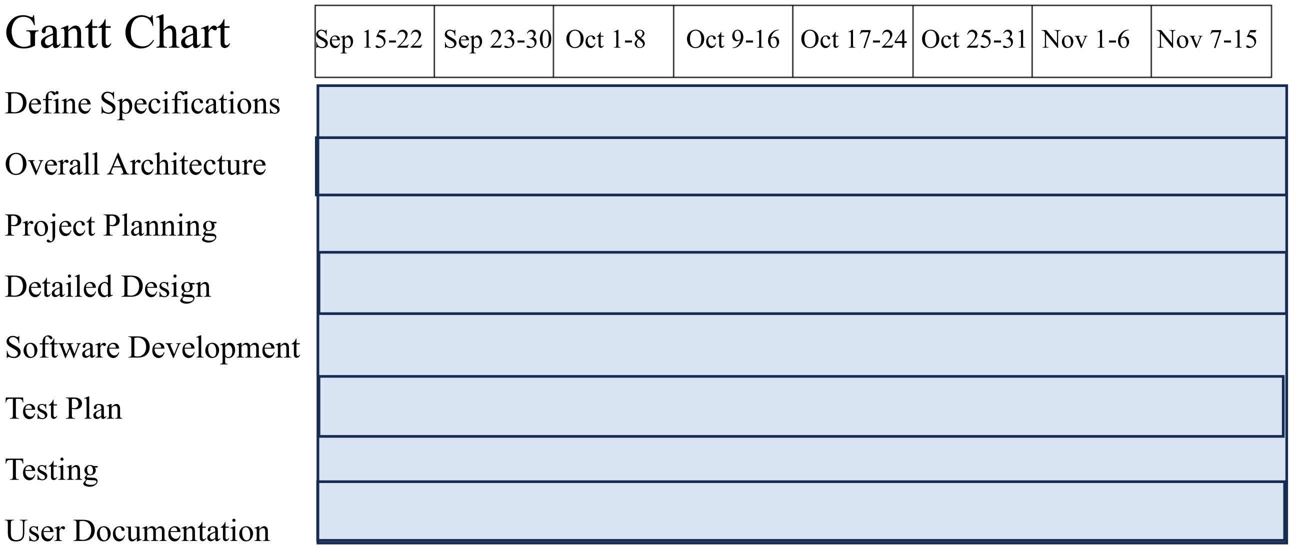
The full version is intended for use by individuals and companies. This is a non-commercial product which is not for sale. All users have access to all features.

7. Other Requirements

Requirement Traceability Matrix

Feature	Date of Testing	Test Case	Status	Comments
CPU Usage				
Memory Usage				
Battery status				
System Information				
List of Processes				

Gantt Chart


Sep 15-22
Sep 23-30
Oct 1-8
Oct 9-16
Oct 17-24
Oct 25-31
Nov 1-6
Nov 7-15
Define Specifications
Overall Architecture
Project Planning
Detailed Design
Software Development
Test Plan
Testing
User Documentation