**1. System Requirements Specification (SRS) Document**

The SRS document describes the system's intended purpose, functionalities, and behaviors. Below is a basic structure for your Inventory Tracking System.

**System Requirements Specification (SRS) for Inventory Tracking System**

**1. Introduction**

* **1.1 Purpose:**  
  The purpose of this document is to provide a detailed description of the Inventory Tracking System (ITS) for managing product inventories. This system allows users to add, update, delete, search, and display inventory items, and it persists the data in a CSV file.
* **1.2 Scope:**  
  This software application manages a product catalog. It will support operations such as adding new products, updating existing products, deleting products, searching products, and displaying the inventory.
* **1.3 Definitions, Acronyms, and Abbreviations:**
  + **ITS:** Inventory Tracking System
  + **CSV:** Comma-Separated Values
  + **UI:** User Interface
* **1.4 References:**
  + "Tkinter Documentation" - Python Tkinter GUI Library
  + "CSV File Format" - CSV Documentation

**2. System Overview**

The Inventory Tracking System (ITS) is a desktop-based application developed using Python with Tkinter and CSV for storing product details. It allows the user to interact with the system through a graphical user interface (GUI) and perform CRUD operations on product data.

**3. Functional Requirements**

**3.1 Add Product**

* The system allows users to input product details, including name, price, and quantity.
* The system validates the input and stores the product in the inventory.

**3.2 Edit Product**

* Users can modify the details of an existing product, including its name, price, and quantity.
* The system updates the product in the inventory.

**3.3 Delete Product**

* Users can select and delete a product from the inventory.
* The system removes the product from the inventory and updates the CSV file.

**3.4 Search Product**

* Users can search for products by name.
* The system filters the inventory based on the search query and displays matching products.

**3.5 Show Inventory**

* The system displays the entire inventory, including the product name, price, and quantity.

**3.6 Save Inventory**

* The system saves the current state of the inventory to a CSV file.

**3.7 Load Inventory**

* The system loads inventory data from the CSV file when the application is launched.

**4. Non-Functional Requirements**

**4.1 Usability**

* The system should be easy to use with a simple and intuitive interface.

**4.2 Reliability**

* The system should ensure that product data is properly saved, updated, and deleted.

**4.3 Performance**

* The application should be responsive even with a large number of products.

**4.4 Compatibility**

* The system should work on all major platforms where Python and Tkinter are supported.

**5. System Architecture**

The system architecture consists of the following components:

* **User Interface (UI):** Tkinter is used to create the graphical user interface.
* **Backend Logic:** The application logic is handled by Python, including product data management.
* **Data Storage:** Product data is stored in a CSV file on the disk.

**Sequence Diagram**

The sequence diagram describes how objects interact in a particular scenario, such as adding a product. Below is the flow for adding a product.

A screenshot of a computer program

Description automatically generated

**Data Flow Diagram (DFD)**

The DFD shows how data flows within the system and how inputs are transformed into outputs.

A computer screen shot of white text

Description automatically generated

A screenshot of a computer program

Description automatically generated