Software Requirements Specification

for

Amazon Price Tracker

Version 0.9

Prepared by Parna Das and Sneha Chavan

10-08-2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this documentation is to provide a comprehensive guide for the development, deployment, and usage of the price tracking application for the website "Amazon." This application is designed to track the price of specific products on Amazon and notify users when the price of a product reaches a specified amount or lesser.

1.2 Intended Audience and Reading Suggestions

The purpose of this documentation is to provide a comprehensive guide for:

1. Developers

2. Testers

3. Product Owners.

1.3 Project Scope

1. Site-Specific Application: The application is specifically designed to track prices on the website "Amazon" only. It will not support price tracking for any other websites. However, the scope and the application should be flexible enough to onboard any other websites falling under project objective per view.

2. Single Product Tracking: The application will allow users to track the price of only one product at a time. Users will need to select the product they wish to monitor, and the application will focus solely on that product until the user opts to track a different one.

3. (Future scope point) Web-Based Application: The application will be web-based, accessible through a web browser. There will be no mobile or desktop application versions of this tool. Users will interact with the application via a dedicated website.

4. Limited Operational Time: The application will be designed to run for a maximum 24 hours per session. After 24 hours, the application will automatically terminate and will need to be restarted by the user for further use and the user will be notified of session termination.

5. Periodic Server Requests: The application will make requests to the Amazon website at regular intervals as configured by the end user. In the absence of input provision a default frequency of 10 minutes. The default metric for input is minutes as a real integer starting with 1.

6. No Data Collection: The application does not collect or store any inputs or intermediate data collected during the program execution.

1.4 References

- https://www.amazon.in/

2. Overall Description

2.1 Product Perspective

The Amazon Price Tracker is a new, self-contained software application designed to operate independently while offering potential integration with existing e-commerce platforms. It is developed to enhance the shopping experience by providing users with comprehensive price tracking and analysis tools. This introduces innovative capabilities to meet the growing demand for smart shopping solutions.

2.2 Product Features

- **Specific Product Tracking:**
- **Customizable Price Check Frequency:**
- **Instant Alert on Target Price:**
- **Realistic Tracking Window:**

2.3 Operating Environment

The following OS can be used:

- Windows
- Linux

2.4 Design and Implementation Constraints

2.4.1 Technical Constraints

Platform Compatibility:

- The application must be compatible with major operating systems, including Windows, and Linux. The CLI should function correctly in default terminal environments on these platforms.

Programming Language:

- The application will be developed using Python 3.10 and above due to its rich library support for web scraping, data processing, and CLI development.

Library Dependencies:

- The application will utilize libraries such as **requests** for HTTP requests, **BeautifulSoup** for web scraping, and **argparse** for command-line argument parsing. Versions of these libraries must be compatible with Python 3.10 and above.

Development Environment:

- The application will be developed on the following environment:
- Development IDE: Microsoft VSCode.
- Virtual Environment: Python 3.10 and above.

2.4.2 Performance Constraints

- **Response Time:** The application should fetch and display fetch price information with a peak of 60 times per 1 hr interval for a single product lookup under normal network conditions.
- **Resource Usage:** The application should run efficiently on systems with at least 2GB of RAM and dual-core processors, without significantly impacting system performance.

2.4.3 Security Constraints

- **Data Privacy:** User credentials and any sensitive information must not be stored in plaintext. Secure encryption mechanisms should be employed for storing any sensitive data.
- **Network Security:** All network communications should be conducted over HTTPS to ensure data integrity and confidentiality during transmission.

2.4.4 User Interface Constraints

- **CLI Usability:** The CLI must provide clear and concise help messages for all commands and options, ensuring ease of use for non-technical users.
- **Localization:** The application must support English initially, with the potential to add support for other languages in future versions.

2.4.5 Operational Constraints

- **Deployment Environment:** The application should be installable and runnable in user environments with Python 3.10 or later, without requiring administrative privileges.
- **External Services:** The application must be capable of accessing external APIs for fetching price data, and any third-party service dependencies must have a documented fallback or retry mechanism in case of failure.

2.4.6 Legal and Compliance Constraints

- **Licensing:** All third-party libraries and tools used in the application must comply with open-source licenses and any proprietary restrictions.
- **Data Compliance:** The application must comply with relevant data protection regulations, such as GDPR(General Data Protection Regulation), to ensure the ethical handling of user data.

This section should be adapted to reflect any specific constraints unique to your project or organizational requirements.

2.5 User Documentation

2.5.1 Introduction

The Amazon Price Tracker is a command-line application that allows users to track and compare product prices throughout 24 hrs.

This documentation provides installation instructions, usage guidelines, and troubleshooting tips.

2.5.2 Installation

- **System Requirements - *Operating System*:** Windows and Linux
- **Python Version:** Python 3.10 or later
- **Internet Connection:** Required for accessing online retailers

Installation Steps:

1. **Download Python:**

- Ensure that Python 3.10 or later is installed on your system. Download it from the [official Python website](https://www.python.org/downloads/).

2. **Clone the Repository:**

- Clone the Amazon Price Tracker repository from GitHub

```
git clone <yet to be added>
```

2.5.3. **Install Dependencies:**

- Using pyproject.toml for requirement management and installation.

2.5.4 Usage

Parameters:-

- **Help Command:**

  - Display help information for the CLI commands:

  ```
  python tracker.py --help
  ```

- **Remove a Product:**

  Remove a product from your tracking list

  ```
  python tracker.py remove --url <product_url>
  ```

2.5.5 Troubleshooting

- **Common Issues:**
  - **Problem:** Unable to connect to the internet.
  - **Solution**: Ensure your device is connected to the internet and try again

  - **Problem:** Invalid product URL
  - **Solution:** Verify that the URL is correct and points to a supported retailer.

2.5.6 Updates and Maintenance

- To keep the application up-to-date, pull the latest changes from the repository:
  ```
  git pull origin main
  ```
- Periodically check for updates to the dependencies:
  ```
  pip list --outdated
  ```

This section should be customized to match *the actual features and functionalities of the Amazon Price Tracker application.*

2.6 Assumptions and Dependencies

Assumptions

- **User Environment:**

  It is assumed that users have a basic understanding of how to use command-line interfaces and have access to a terminal or command prompt.

- **Python Installation:**

  Users have Python 3.10 or later installed on their systems and have added Python to their system PATH, allowing them to run Python scripts from the command line.

- **Internet Connectivity:**

  Users have a stable internet connection when using the application to fetch prices from online retailers.

- **Supported Retailers:**

  The product URLs provided by users belong to supported retailers. The list of supported retailers will be documented and maintained separately. (Optional)

- **Operating System Compatibility:**

  The application is assumed to work on Windows, and Linux environments with minimal platform-specific issues.

- **Data Accuracy:**

  It is assumed that the price data retrieved from online retailer is accurate and up-to-date at the time of retrieval.

Dependencies

- **Python Packages:**

  The application relies on several Python packages listed in the `pyproject.toml` file. Key packages include:

  1. **Requests:** For making HTTP requests to fetch product data.
  2. **BeautifulSoup:** For parsing HTML content from retailer websites.
  3. **Argparse:** For handling command-line arguments.

- **External APIs:**

  The application depends on external APIs for accessing additional data or services, if any. The functionality may be limited or impacted if these APIs are unavailable or changed.

- **Version Control System:**

  The source code is managed using Git. Users are expected to have Git installed if they are cloning the repository for installation or updates.

- **Third-Party Websites:**

  The application's functionality depends on the structure and availability of the websites of supported retailers. Any significant changes to these websites may require updates to the application.

- **Security and Compliance:**

  The application assumes compliance with relevant data protection regulations, such as GDPR, and relies on secure HTTPS connections for data transmission.

This section should be tailored to reflect any specific assumptions or dependencies that apply to your project and its environment.

3. System Features

- **Product Tracking**

  Description:
  Allows users to track the prices of product from Amazon by specifying product URL

  Functional Requirements:

  - **FR1.1:** The system shall allow users to add a product to their tracking list using a valid product URL.
  - **FR1.2:** The system shall validate the product URL to ensure it belongs to a supported retailer.
  - **FR1.3:** The system shall store the product information and tracking details in a local database.

- **Price Checking**

  Description:
  Enables users to check the current price of tracked product and receive notifications of any changes.

Functional Requirements:

- **FR2.1:** The system shall fetch the current price of tracked product from the respective retailer.
- **FR2.2:** The system shall allow users to manually check prices using a command-line option.

- **Product Management**

  Description:
  Facilitates management of the user's tracked product.

  Functional Requirements:

  - **FR4.1:** The system shall provide details about each tracked product, including the product name, URL, and recorded price.

- **User Notifications**

  Description:
  Notifies users of significant price changes or updates to tracked product.

  Functional Requirements:

  - **FR5.1:** The system shall provide notifications via the command line or by generating a log file.

3.0.1.1 *3.6 Configuration and Settings*

Description:
Allows users to configure application settings and preferences.

- **Functional Requirements:**

  - **FR6.1:** The system shall provide a CLI interface for the users to set application preferences such as notification thresholds and update intervals.
  - **FR6.2:** The system shall load user configurations at startup and apply them to the application settings.
  - **FR6.3:** The system shall provide default settings if the configuration file is not present.

4. External Interface Requirements

4.1 User Interfaces

4.1.2 Command-Line Interface (CLI)

- The Amazon Price Tracker shall provide a command-line interface that allows users to interact with the system through a terminal or command prompt.
- The CLI shall support commands for adding product url, timeout, frequency, and target price, confirmation of product.
- It shall display clear and concise command prompts and error messages to guide users in executing commands effectively.
- The interface shall include a help command that provides users with a list of available commands and their usage syntax.

4.2 Hardware Interfaces

4.2.3 User Device Requirements

- The application shall be compatible with personal computers and laptops equipped with a keyboard and monitor, which are essential for CLI interactions.
- Minimum hardware specifications required for running the application include:
  - Operating System: Windows 10,and compatible Linux distribution
  - RAM: 2 GB
  - Processor: Dual-core 1.8 GHz or higher
  - Disk Space: 100 MB of free storage

4.3 Software Interfaces

4.3.1 Amazon Platform API

- The Amazon Price Tracker shall interact with the Amazon platform through its web scrapped elements to retrieve real-time product pricing and details.
- The system must authenticate and handle web scrapped elements.
- Data obtained from the web scrapped elements must be processed to ensure accuracy and consistency in the application.

4.3.2 Notification Services

- The application shall integrate with email and SMS services to send notifications about price-hitting target and alerts to users.
- It must support compatibility with major email service providers, and SMS gateways.

- The system shall store user preferences for notification methods and ensure they are applied when sending alerts.

4.4 Communications Interfaces

4.4.1 Internet Connectivity

- An active internet connection is required for the Amazon Price Tracker to access the web scrapping the elements and send notifications.
- The system shall implement error handling for network interruptions and attempt to reconnect or retry operations as needed.

4.4.2 Security Protocols

- All communications with external services, such as notification services, shall be conducted over secure protocols (e.g. HTTPS).

5. Other Non-functional Requirements

- **Development Environment:**
  - VS Code Workspace: The project must be developed within a Visual Studio Code workspace to ensure consistency in the development environment.
  - Virtual Environment: The project must be set up and developed within a virtual environment to manage dependencies and avoid conflicts with global packages.
- **Code Quality and Static Code Analysis (SCA):**
  - SCA Pipeline: The project must incorporate a Static Code Analysis pipeline that includes tools like Flake8, MyPy, Pylint, and Vulture to ensure code quality, type checking, and adherence to coding standards.
  - Code Coverage: The project must include unit tests with a requirement of achieving 100% code coverage to ensure all parts of the code are tested.
- **Reliability:**
  - Application Stability: The application must be robust and should not fail or break down under any circumstances. Proper error handling and fault tolerance mechanisms must be implemented to ensure continuous operation.
- **Version Control and Hosting:**
  - GitLab Hosting: The project must be hosted on GitLab, utilizing GitLab's version control features for source code management and collaboration.
- **Documentation and Presentation:**
  - Detailed README: The project repository must include a detailed README file that provides comprehensive documentation, including setup instructions, usage guidelines, and an overview of the project.

- Demo Video: A demo video showcasing the application's features and functionality must be created and included in the project repository or linked within the README.

- **Showcase:**

  - LinkedIn Showcase: The project must be showcased on LinkedIn to highlight the work done, including a brief description, key features, and a link to the demo video or project repository.

These non-functional requirements ensure that the project adheres to quality standards, is developed in a controlled environment, is thoroughly documented, and is presented professionally.