

Software Requirements Specification

for

Django\_Web\_App

Version <1.0>

Prepared by

|  |  |  |
| --- | --- | --- |
| Muniker Aragon | 11632391 | Muniker.aragon@wsu.edu |
| <name> | <student #> | <e-mail> |

|  |  |
| --- | --- |
|  |  |
| Date: | 10/25/2019 |
|  |  |
|  |  |
|  |  |

Content

Revisions iii

1 Introduction 1

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 Document Conventions 1

1.6 References and Acknowledgments 2

2 Overall Description 3

2.1 Product Perspective 3

2.2 Product Functionality 3

2.3 Users and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

3 Specific Requirements 5

3.1 External Interface Requirements 5

3.2 Functional Requirements 6

3.3 Behaviour Requirements 6

4 Other Non-functional Requirements 7

4.1 Performance Requirements 7

4.2 Safety and Security Requirements 7

4.3 Software Quality Attributes 7

5 Other Requirements 8

Appendix A – Data Dictionary 9

Appendix B - Group Log 10

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| Draft Type and Number | Full Name | Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded. | 00/00/00 |

# 

# Introduction

## Document Purpose

This document provides a descriptive view of the scope and specifications that will be implemented in the software application. This document describes the intended user interaction with the software in terms of accessibility, safety, and performance. It will also carefully explain the functionality and architecture of the software along with any present constraints. Some design components that will be analyzed in the following sections of this document will be the registration component, user profile, and the software's main feature. This document will also cover the interface requirements that will be implemented to provide an excellent user experience. Lastly, this document will cover the performance of the software in terms of speed, accessibility, and any precautions that must be followed.

## Product Scope

The objective for this software is to provide users an efficient manner of overseeing the price of any online product that they wish to purchase. Users will be able to create a personalized profile where they can store multiple products by simply entering the URL of the products site. This application will then save this URL and continuously monitor inform the user of a drop in the product’s price. The main benefits of this application is to provide users a place where they can save and monitor all the items that they wish to buy while eliminating the time that it would take to search each of these individually.

## Intended Audience and Document Overview

This document is mainly directed to the users or anyone who is interested in the design architecture of this application. The sections that will be covered in this document will provide information and diagrams to demonstrate the major components of the system and also describe how smaller subsystems  interact with one another. This document will also cover the underlying operating system of the application as well as the main functionality and purpose that this product will be able to provide. The sections of this document will examine the different levels of interfaces of the software from hardware to user interaction. Lastly, this document will dive into the non-functional requirements of the systems such as performance, quality, safety and  security.

## Definitions, Acronyms and Abbreviations

URL: Uniform Resource Locator

MVC: Model View Controller

AWS: Amazon Web Services

E2C: Elastic Compute Services

## Document Conventions

<In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1” margins found in this template. For Section and Subsection titles please follow the template.

TO DO: Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. Sometimes, it is useful to divide this section to several sections, e.g., Formatting Conventions, Naming Conventions, etc.>

## References and Acknowledgments

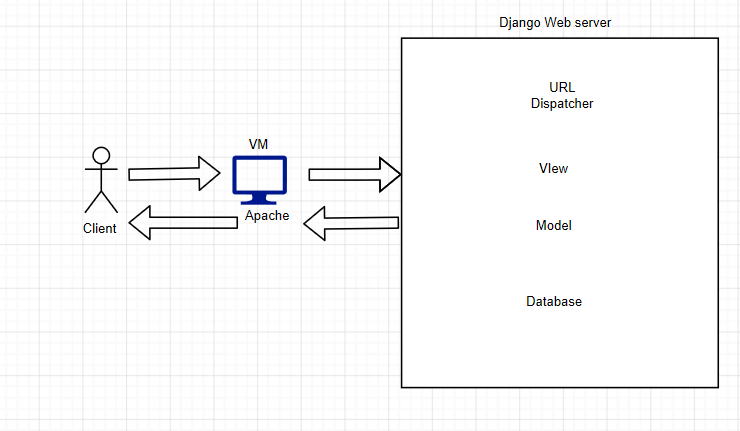
Django Documentation.Internet:https://www.djangoproject.com/, [10,25,2019].

Apache.Internet: <http://www.apache.org/>, [10,25,2019].

# Overall Description

## Product Perspective

This project implements an MVC architecture by implementing a Full-Stack Web application using the Django framework. This Model-View-Controller will separate the different responsibilities of our project such as databases, views and templates. An overview of the system interactions is represented in the diagram shown below. In this diagram a client will pass a request to the Web Server running an Ubuntu virtual machine.The Server will then utilized WSGI scripts to communicate with the URL dispatcher. The URL Dispatcher will decide  which views to execute and from here the views will then select which templates to render and if any models need to be accessed.



## Product Functionality

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, will be effective.

TO DO:

1. Provide a bulleted list of all the major functions of the system

2. **(Optional)** Provide a Data Flow Diagram of the system to show how these functions relate to each other. This is useful when there is a clear sequence for the functions being performed.>

Client access to the application shall be through a standard web browser, the chrome browser shall be the standard browser that is supported.

* The user interface to the app shall be an interactive form-based interface.
* All users shall have the ability to register an account using a unique email address not yet registered with the system, accounts shall be password protected.
* Users who have registered accounts shall be able to login to the web app using their credentials.
* Any user with a verified account shall have the ability to submit several online store pages for monitoring when logged into their account.
* The user shall have access to all data conected to their account. Including updates on the online store pages submitted for monitoring.
* The user shall have the ability to remove online store pages from their accounts monitoring list.
* Users shall have the ability to log out of their accounts.
* Users shall have the ability to delete their accounts.
* The system shall inform the user of what online stores are supported by the system.
* The system shall scrape any online store page that is submitted to it for the price of the relevant item. The system shall then store the relevant data on that item for later use.

## Users and Characteristics

<Identify the various users that you anticipate will use this product. Users may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience.

TO DO:

1. Describe the pertinent characteristics of each user. Certain requirements may pertain only to certain users.

3. Distinguish the most important users for this product from those who are less important to satisfy.>

All users shall come from the general consumer population. The most important users for our application will be those whose financial situation incentivizes them to seek discounted items. The least important user for our application will be those whose financial situation is not as severe and so would not require them to use our application as frequently. The characteristics of these users will vary drastically. Users may be disabled in many ways or have poor skills with technology these variables will need to be taken into account.

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist. In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface

TO DO: As stated above, in at least one paragraph, describe the environment your system will have to operate in. Make sure to include the minimum platform requirements for your system. >

The environment that will be implemented for this software is composed of virtual machine running an Apache Web Server. This software utilize Amazon Web Services to run an E2C instance whose operating system is Ubuntu 18.04. This instacance will run an Apache Web Server that will handle client requests from different users.Finally the web application utilizes the django web server to communicate with the Apache server via WSGI. The scripts contain in WSGI control the flow of communication between URLs, views, models, and database.

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).

TO DO: In this section you need to consider all of the information you gathered so far, analyze it and correctly identify relevant constraints.>

Issue of periodacally have to update the intems from each user which will require correct timing and threading to not back up the system.

2.5.1 The system will need to provide accurate information. Therefore updates on pricing data must be quick, about one update per hour when the user is not signed in and update within 10 seconds on user request.

2.5.2 All sensitive user data such as passwords and email addresses will need to be secure. This will require proper hashing and salting of relevant data. Additionally, all source code must comply with Open Web Application Security Project (OWASP) guideline and all data must be validated on the server-side.

## User Documentation

This application will provide a specific web page designed to teach users how to navigate and interact with the application. This page will contain images and text that will help instruct users on how they can add new items to their shelf, and how they can check for notifications on the drop of prices in their items.

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.

TO DO: Provide a short list of some major assumptions that might significantly affect your design. For example, you can assume that your client will have 1, 2 or at most 50 Automated Banking Machines. Every number has a significant effect on the design of your system. >

1. The system will assume that the prices displayed on online store listings are truthful.
2. The system will depend on the Django web application framework for much of its functionality.
3. The front end of the system will depend on bootstrap for its front end appearance and functionality.
4. For web scraping functionality the system will depend on the Beautiful Soup Python library.

# Specific Requirements

## External Interface Requirements

### User Interfaces

The different types of user interfaces that will be implemented in this application are a Homepage, Registration page,Login Page,How to use page, Add item page, My items page, and an analytics  page. A live preview of the basic layout of this page can be seen by searching the following link<http://3.91.252.175/>.

### Hardware Interfaces

This application will be contained in a virtual machine located in Amazon Web Services. This VM will provide the same components as normal computer in terms of CPU, memory, storage, and networking capacity.We will be using this virtual hardware to store all the contents of the website and also to run the Apache web server which will be in charge of  receiving and maintain connections from other client machines.

### Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems (Windows? Linux? Etc…), tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.

TO DO: The previous part illustrates some of the information you would usually include in this part of the SRS document. To make things simpler, you are only required to describe the specific interface with the operating system.>

The application shall use the Django web application framework for back end systems and for front end systems the application shall use bootstrap. For data storage, the system shall use MySQL and SQLite databases. For user data and any other private or sensitive information mySQL databases will be used for storage. For non-sensitive data such as pricing information, SQLlite servers will be implemented. The MySQL database will be hosted on a Linux server. All web scraping shall be accomplished using Beautiful Soup, a python library for pulling data from HTML and XML files.

### Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

TO DO: Do not go into too much detail, but provide 1-2 paragraphs were you will outline the major communication standards. For example, if you decide to use encryption there is no need to specify the exact encryption standards, but rather, specify the fact that the data will be encrypted and name what standards you consider using. >

## Functional Requirements

The functional components that will be implemented in this application will be a registration and login system that will allow users to create a personalized location to save Items that they wish to monitor. Another main component that will be implemented for this system is a model that creates a one to many relationship between a User and different products. The application will implement a web scraping component which will parse the html from a product and locate where the price is embedded. The last component that will be implemented is an Update component that will be in charge of periodically grabbing each User and updating all of their items.

## Behaviour Requirements

### Use Case View

<A use case defines a goal-oriented set of interactions between external actors and the system under consideration.

TO DO: Provide a use case diagram which shows the entire system and all possible actors. Do not include detailed use case descriptions (these will be needed when you will be working on the Test Plan), but make sure to include a short description of what every use-case is, who are the actors in your diagram.>

# Other Non-functional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

TODO: Provide relevant performance requirements based on the information you collected from the client. For example you can say “1. Any transaction will not take more than 10 seconds, etc…>

## Safety and Security Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.

TODO:

* Provide relevant safety requirements based on your interview with the client or, on your expectation for the product.
* Describe briefly what level of security is expected from this product by your client and provide a bulleted (or numbered) list of the major security requirements.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Portability, etc…) provide requirements related to the different software quality attributes. Base the information you include in these subsections on the material you have learned in the class. Make sure, that you do not just write “This software shall be maintainable…” Indicate how you plan to achieve it, etc.>

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist the Teaching Assistant to determine the effort put forth to produce this document>