Release Plan

Product Name: Saavy Shopper

Team Name: Saavy Developers

Release Name: 1.0

Release Date: Jul 25, 2023

**Revision Number: 1** 

Revision Date: Jul 11, 2023

## Release Plan

# **High Level Goals**

At SaavyShopper, our primary objective is to direct the development of a cutting-edge aggregate platform that empowers users to effortlessly monitor and manage the fluctuating prices of their desired products. In order to successfully accomplish this task, we have identified a number of overarching objectives that encompass both the technical and logistical dimensions of executing the development and deployment of said product. Our strategic initiatives encompass the following key areas: spearheading the development of a streamlined and impactful front-end user interface, harnessing the power of application programming interfaces and cutting-edge webscraping tools to efficiently procure item prices, implementing a robust cron job system to ensure timely updates of our stored prices, leveraging the scalability and flexibility of cloud infrastructure to securely store and facilitate user modifications of our invaluable data.

Firstly, developing a clean user interface is integral to the success of our application.

This is because the goal of our project is to streamline a process, and as such we need to

ensure that the user is able to quickly and easily add items. Secondly, we need to be able to actually extract the data from the link that the user pastes into the front end portion into the website. Thirdly, we need to be able to implement a cron job in order to run our webscraping script daily to update our stored price. We also need a way to store our price data, to which we will leverage cloud services to minimize operational costs. Finally, if we have time, we would like to also develop an extension for web browsers to truly streamline the item tracking experience for our users.

## User Stories Defining the Scope of the Project:

We define story points using the fibonacci sequence. 21 points to represent the entirety of the sprint (1 week long).

#### User Story 1:

- As a user, I want to be able to paste in amazon link and view relevant information in our app
  - 1.1: Created web scraping service that takes in an amazon link and returns the name and price of the item [13 story points]
  - 1.2: Created general event-scheduling service that acts as the api gateway [21 story points]

#### User Story 2:

- As a user, I want to be notified when my item changes in price
  - 2.1: As a user, I want the software to automatically fetch and update the current prices of tracked items from e-commerce sites. [5 story points]
  - 2.2: As a user, I want to receive notifications when the current price of a tracked item falls below the desired price. [3 story points]

#### User Story 3:

 As a user, I want to be able to visualize price trends of my item as well as change the notification threshold for my item

### User Story 4:

- As a user, I want to know all of my tracked items regardless of which device I'm on.
  - 4.1: Authentication and loading user-related data in firebase (13 story points)
    - Authenticate a user and store information about the items they are tracking to a database and associate it with them
  - 4.2 Connect tracked item to specific user and verify user by jwt token (13 story points)
    - Connect firebase to our python flask server
    - Add items to database and connect to the user by user id
    - Verify correct user by creating middleware to check if jwt\_token is valid

#### Sanity Check

I firmly believe that the planned workload given to each team member is certainly within the team's executable capacity. I am very familiar and experienced with working with the team members, as I am able to accurately gauge my team member's thirst for a challenge and their ability to complete the tasks given. Approximately the total number of points throughout all three sprints is 21 points. These points will be distributed based on each team member's expertise, experience, and skill level appropriately. The workload distribution across all sprints was reasonable, as the distribution of work was predated by a conference call in which all team members were present. Everyone discussed, considered, and eventually agreed upon their

respective workloads, allowing us to avoid running into issues involving infrastructure tasks, spikes, holidays, and midterms.

#### Product Backlog

Some of the product backlogs that were included implementing a working Email sending service like EmailJS. This was initially desired as the most optimal way of notifying the individual that an item they wanted was on sale, however it was soon discovered that there exists a Python library that already supported email sending capabilities. The Python library was not only more cost effective, as no subscription was needed, but also more convenient, as there was no need to learn Java Script or to familiarize ourselves with yet another programming application. Another instance of a product backlog that occurred was our quest to obtain and evaluate the benefits and drawbacks of using API's as opposed to web scraping. We were initially going to pursue a method of only using Amazon API's, as that would provide the most stable and sustainable way to achieve our website's desired functionality. However, we soon ran into great trouble trying to get the Amazon API to work properly, as well as our revised sprints have led us to expanding our website's supported ecommerce platforms to encompass many other major sellers like Ebay, Bestbuy, and Newegg. This led us to change focus and dedicate our efforts into web scraping, which after some trial and error, we were able to get working consistently and properly.