SRS – Behavior Driven Development (Behave)

1. Introduction
   1. Purpose – The purpose of this document is to provide an overview of the software requirements, functional and non-functional requirements and other specifications related to the Behave Lab section of the Fruit Stand Application.
   2. Scope – This document outlines the functional and non-functional requirements of the software, including the features and its constraints.
2. System Overview
   1. Objectives of Behave Lab
      1. Familiarize the user with the basics of the Behavior Driven Development Framework Behave in Python.
      2. Expand Users knowledge of testing suites and methodologies within Python.
      3. Create ‘Features’ that correlate with tasks within the Fruit Stand application.
   2. Technologies Used
      1. Behave
      2. IDE of choice. (PyCharm, VS Code, etc.)
      3. Python
      4. Flask
3. Functional Requirements
   1. Directory Creation
      1. Directories will be created to house ‘Feature files’ to be used in housing the created tests.
   2. Gherkin Testing Language familiarization
      1. Behave uses Feature files contained within a directory and are written in The Gherkin testing language.
   3. Automated Testing
      1. Behavior Driven Development testing uses the Gherkin Language to create test scenarios to verify functions on the Fruit Stand application such as adding items to the cart, removing items from the cart, applying a discount to the total purchase price.
   4. Reports
      1. After the Feature files and Fixtures have been run a summary of the tests should be available for review.
4. Non-Functional Requirements
   1. Reliability – Feature tests should run consistently and produce reliable results.
   2. Performance – Feature tests should run within acceptable timeframes and when complete start the next test within the feature file.
   3. Scalability – Feature tests should be designed to cover specific elements of the Fruit Stand application. Tests should be adaptable and easily allow testing of features that could be added at a later date.
5. Setup and Environments Required
   1. Supported web browser for Fruit Stand application launch.
   2. Behave Ver 1.2.7.dev5 or newer.
   3. Python version 3.8 or newer.
   4. IDE software
6. Deliverables
   1. Feature File structures that contain test cases covering Fruit Stand Application elements.
   2. Familiarization of ‘Features’ directory and a basic understanding of The Gherkin Language.
   3. Generate a test report detailing feature test passes and failures.
7. Conclusion
   1. This Lab will be a walkthrough on the basics of using the Behave framework in the Python language interacting with a Fruit Stand Application in order to breakdown the basic concepts of creating tests utilizing the Behavior Driven Development method. Additionally, this lab will give an overview of the Gherkin Language used in creating Feature files used in Behave.
8. Appendices
   1. Reference Materials Used
      1. Python Documentation - https://www.python.org/doc/
      2. Behave Documentation - https://behave.readthedocs.io/en/latest/#
      3. Flask Documentation - https://flask.palletsprojects.com/en/3.0.x/