# Fashion Trend Analytics

**TEST PLAN**
**Team JAZZ MEN**

Anagh Goswami 1217426
Meet Pandya 1214306
Jasman Gill 1211554
Jesse Truong 1222722
Jia Xu 1213268

November 2, 2016

# Contents



# Revision History

| Date | Version | Comments |
|---|---|---|
| November 2, 2016 | 1.0 | First draft |

# 1 Overview

The purpose of this document is to provide an initial plan for the testing of our project and application. The following gives out a brief outline and an overview of what is discussed in this document:

- A proof of concept is described in Section 3
- The different variations of test cases to be tested are described in section 5
- Each test case is categorized based on whether it is for a functional or non-functional requirement in sections 5.1 and 5.2

## 1.1 Test Case Format

Each test case is formatted as shown below:

| | |
|---|---|
| **Test:** | **Test name** |
| **Requirement:** | **Requirement Type** |
| **Description:** | A description of the test being conducted |
| **Input:** | The input to the system that will change the system's state |
| **Output:** | The relevant output that is observed |
| **Pass:** | The pass criteria describes what is acceptable as a successful test based on the requirements of the system |

# 2 Software Specifications

## 2.1 Functional Requirements

- Data Scraping
  - Download HTML/XML from specified fashion blogs and websites
  - Parse HTML data for relevant fashion trend information
  - Find product pages from retailer websites based on analysis of fashion trends
- Analyzing
  - Identify top industry trends mentioned by bloggers and industry people in data scraped from fashion sites
  - Persist analysis results in database
- Reporting
  - Display top trends in website
  - Provide link for user to directly purchase product from retailer

## 2.2 Non-Functional Requirements

- Look and Feel
  - Trends should be easily identifiable for users in results web page
- Usability
  - Web site should be usable by anyone of the age 15+ with no prior training
- Performance
  - Scraping should complete in reasonable amount of time, without timing out
  - Scraping and analysis tasks should not be too resource-intensive
  - Products and trends presented should be accurate and relevant
  - Website should be available 24 hours, 365 days
- Operational
  - Server should be available to perform scraping, analysis and web serving tasks
  - Server should output log files with console output and errors
- Security

  - Website should not be vulnerable to any injection attacks

- Legal
  - Should only be scraping data from public domains
  - Should not violate website rules in regards to scraping, or API usage

# 3 Proof of Concept Testing

Datasets (cover typical application scenario from real world to widely used)
Performance Metrics (latency, throughput, accuracy)

- Accuracy in terms of how well the website works
- Maximum occupancy
  - Response time
  - How fast the web page loads for the users

Risks in your project (think this together with the proof of concept demo)

- We can't develop an accurate algorithm
  - Solution: put more effort into researching of already working algorithms and work from there
- Web scraping
  - Finding the needed information on a website
  - Can be hard to differentiate between useful information and extra jumble
  - Information will be organized in a different manner across all the websites. This causes a problem with how to do unified web scraping of the data with one generic scraping algorithm.
  - The keywords that we are using to scrape data from sites might not be the most optimal syntax, since there are many ways for a website to describe their
- If there isn't enough information available for us to find a reliable trend
  - Solution: switch industries and try a different topic
- What type of scraping we will be doing?
  - Whether it is graphical for the images of clothes? Or scraping the data-text, for ex: scraping through the text on a page and then interpret the data in an understandable manner.
  - We run in the risk of having difficulty with data scraping to begin with and how to scrape through multiple websites designed in their own unique way.
  - Potential risk is the web scraping algorithm only works on a particular website.
    - Solution: make an algorithm that works on multiple websites efficiently. Algorithm which can also be applied to fashion websites that were never tested during this project.

# 4 Testing

## 4.1 Automated Testing

All unit testing for this project will be automated to save time. Unit tests will be created and tested as the project is developed. These unit tests will be separated by different modules including Python unit testing and Javascript unit testing. We will be using Unittest for Python and QUnit for Javascripts. Uni ttest is a built-in unit testing module / library in Python2 and Python3. This will serve as the testing module we will use to perform our unit testing on all Python code. QUnit is a Javascript unit testing framework that can be used to test any generic Javascript code.

## 4.2 Functional Requirements Testing

| | |
|---|---|
| **Test:** | **Verify Data Scraping(Manual)** |
| **Requirement:** | **Functional (Data Scraping)** |
| **Description:** | Run scraping task for a page then compare result with that page's source code |
| **Input:** | A webpage of any clothing website |
| **Output:** | HTML data |
| **Pass:** | The output HTML data matches the page's source code |

| | |
|---|---|
| **Test:** | **Verify Data Scraping Part 2 (Manual)** |
| **Requirement:** | **Functional (Data Scraping)** |
| **Description:** | Check scraped data for relevance |
| **Input:** | HTML data from webpage |
| **Output:** | Text |
| **Pass:** | The output text should be fashion trends mentioned in the webpages |

| | |
|---|---|
| **Test:** | **Relevant Product Page (Manual)** |
| **Requirement:** | **Functional (Data Scraping)** |
| **Description:** | Product pages pertain to trend fashion data |
| **Input:** | Trends Data |
| **Output:** | Product Page |
| **Pass:** | Product page is relevant to that particular trend |

| | |
|---|---|
| **Test:** | **Verify Trends (Manual)** |
| **Requirement:** | **Functional (Analyzing)** |
| **Description:** | Check scraped data for mentioned top trends |
| **Input:** | Data scraped from fashion website |
| **Output:** | Trends Data |
| **Pass:** | Data should contain trends mentioned by bloggers |

| | |
|---|---|
| **Test:** | **Database Test (Automated)** |
| **Requirement:** | **Functional (Analyzing)** |
| **Description:** | Ensure data is encapsulated in database |
| **Input:** | Scraped Data |
| **Output:** | Database |
| **Pass:** | Data is stored in database |

| | |
|---|---|
| **Test:** | **Verify Top Trends (Manual)** |
| **Requirement:** | **Functional (Reporting)** |
| **Description:** | Check system website's data to ensure top trends |
| **Input:** | Top Trends |
| **Output:** | Webpage displaying top trends |
| **Pass:** | Webpage pertains top trends |

| | |
|---|---|
| **Test:** | **Product link connects user to product page (Automated)** |
| **Requirement:** | **Functional (Reporting)** |
| **Description:** | When link is clicked, it directs user to the purchase page of that product |
| **Input:** | Click link associated to the product |
| **Output:** | Product Page |
| **Pass:** | Takes user to accurate product page |

### 4.3 Non-Functional Requirements

| | |
|---|---|
| **Test:** | **Results Page Visual Testing (Automated/Manual)** |
| **Requirement:** | **Non-Functional (Look and Feel)** |
| **Description:** | Check if the results shows trends in a way that user can understand it |
| **Input:** | A Web page inputted in the system |
| **Output:** | Accurate results appear showing trends and easy to read |
| **Pass:** | The resulted data looks understandable and useful for the user. Its shows the user what are the current popular trends in the particular clothing category. |

| | |
|---|---|
| **Test:** | **Usability Testing (Manual/Automated)** |
| **Requirement:** | **Non-Functional (Usability)** |
| **Description:** | Check the features of the system to conclude overall how easy it is to operate the system |
| **Input:** | webpage submitted by users |
| **Output:** | Successful utilization from user |
| **Pass:** | User is able to successfully operate all the features of the system |

| | |
|---|---|
| **Test:** | **Scraping Operation Time (Automated)** |
| **Requirement:** | **Non-Functional (Performance)** |
| **Description:** | Verify that scraping data takes a reasonable amount of time |
| **Input:** | A webpage |
| **Output:** | HTML data |
| **Pass:** | Scraping operation does not timeout and completes in a reasonable time |

| | |
|---|---|
| **Test:** | **Scraping Operation Resources (Automated)** |
| **Requirement:** | **Non-Functional (Performance)** |
| **Description:** | Website should display actual trends accurately that is relevant to user |
| **Input:** | A web page |
| **Output:** | HTML data |
| **Pass:** | Scraping operation does not take a surplus of resources |

| | |
|---|---|
| **Test:** | **Website information is accurate and relevant (Automated)** |
| **Requirement:** | **Non-Functional (Performance)** |
| **Description:** | Website should display actual trends accurately that is relevant to user |
| **Input:** | Click on web page |
| **Output:** | Relevant data shown on web page |
| **Pass:** | Data shown on web page is relevant and accurate |

| | |
|---|---|
| **Test:** | **Website is functional at all times (Automated)** |
| **Requirement:** | **Non-Functional (Performance)** |
| **Description:** | Website should always be available to users 24 hours 365 days of the year |
| **Input:** | Click on web page |
| **Output:** | Web page |
| **Pass:** | Web page loads |

| | |
|---|---|
| **Test:** | **Server Operations are functional (Automated)** |
| **Requirement:** | **Non-Functional (Operational)** |
| **Description:** | Servers should be available to perform scraping, analysis and web serving tasks |
| **Input:** | Instructions to the server |
| **Output:** | Tasks delivered accordingly |
| **Pass:** | Tasks are fulfilled and satisfactory |

| | |
|---|---|
| **Test:** | **Server operations reports status to the console (Automated)** |
| **Requirement:** | **Non-Functional (Operational)** |
| **Description:** | Servers should output log files with console output and errors |
| **Input:** | Instructions to the server |
| **Output:** | Log in console |
| **Pass:** | Server successfully displays information in the console output |

| | |
|---|---|
| **Test:** | **Prevent Injection Attacks (Automated)** |
| **Requirement:** | **Non-Functional (Security)** |
| **Description:** | HTML commands hidden as search queries on website should not be executed |
| **Input:** | String from webpage with inline HTML commands |
| **Output:** | Items related to search string |
| **Pass:** | Search string does not affect HTML data |

| | |
|---|---|
| **Test:** | **Prevent Injection Attacks (Automated)** |
| **Requirement:** | **Non-Functional (Security)** |
| **Description:** | SQL commands hidden as search queries on website should not be executed |
| **Input:** | String from webpage with inline SQL commands |
| **Output:** | Items related to search string |
| **Pass:** | Search string does not affect SQL data |

| | |
|---|---|
| **Test:** | **Prevent Injection Attacks (Automated)** |
| **Requirement:** | **Non-Functional (Security)** |
| **Description:** | Javascript commands hidden as search queries on website should not be executed |
| **Input:** | String from webpage with inline Javascript commands |
| **Output:** | Items related to search string |
| **Pass:** | Search string does not affect Javascript data |

| | |
|---|---|
| **Test:** | **Scraping Data from Public Domains (Automated/Manual)** |
| **Requirement:** | **Non-Functional (Legal)** |
| **Description:** | Ensuring all websites that are being scraped are public websites that are available on the Web |
| **Input:** | Search keywords on Google to find public domain websites within project interest |
| **Output:** | List of websites to scrape from |
| **Pass:** | All websites on project's scrape list are available on the Web |

| | |
|---|---|
| **Test:** | **Website links are functional (HTML/CSS Testing) (Automated/Manual)** |
| **Requirement:** | **Non-Functional (Look and Feel)** |
| **Description:** | Ensure that the HTML/CSS code used in building the website is validated and functional |
| **Input:** | HTML and CSS code used in creating the website |
| **Output:** | Debugging errors spotted by Validator.nu |
| **Pass:** | Code is validated i.e no errors are found |

| | |
|---|---|
| **Test:** | **Javascript code used is functional (Automated/Manual)** |
| **Requirement:** | **Non-Functional (Look and Feel)** |
| **Description:** | Ensure that Javascript functions are valid |
| **Input:** | HTML and CSS code used in creating the website |
| **Output:** | Debugging errors spotted by Validator.nu |
| **Pass:** | Code is validated i.e no errors are found |

# 5 Timeline

This document is designed in a way to follow the timeline posted below. The timeline is to assist the progression of testing the system and it is an estimated completion time which could or could not be met on time due to running into errors.

| **Expected Completion Date** | **Task to be completed** |
|---|---|
| November 10, 2016 | Verify data scraping scripts |
| November 22, 2016 | Proof of Concept Demonstration completed |