Test Strategy Document

Avani Gupta (2019121004), Parv Joshi(2018101062), SIddharth Jain(2018101038), Apoorva Tirupathi(2019121012)

Scrapshut

Team#24

**Scope**

To be Reviewed & Approved By :

The Client : The CEO, CoFounder of ScrapShut, Mounikesh

Our Mentor : Head TA Mohit Chandra sir

The Professor : Ramesh Loganathan sir

* This file contains a detailed writeup of our timeline of testing phases throughout the AGILE sprint cycle.

**Test Approach**

We made a detailed plan for testing our webapp, as we didn’t want to miss out on anything under the sun, and cross-checked our work inorder to take a second look at any edge cases one would’ve overlooked. Testing the ScrapShut web app was divided into 4 major components

* + Frontend
  + Backend
  + Deep Learning Models
  + The Customised Scrapper

Each testing was further divided into 2 levels of abstraction - The site admin functionalities & the user functionalities

We did testing at various stages as we’ve had multiple refactorings done in each consecutive sprint, for which we needed to ensure our changes haven’t affected the application’s behavior unexpectedly. It was a complex task as Scrapshut is made of several layers of logic, from HTTP-level request handling, to form validation and processing, to rendering of all the templates.

**Test Environment**

* We didnt use any new additonal environment apart from the existing environment.
* Our testing modules & frameworks were also in python, so we continued to use the same virtual environment to install them, and for selenium used the IDE

**Testing Tools**

* We used Django’s “unittest” module to simulate requests, insert test data into the user forms, inspect the application’s output for article rating requests and generally to verify all that the code should be doing.
* For automation of a simulated user we used Selenium, to moniter a complete cycle of user activities from login to exit.

**Test Use Cases**

* + We performed load testing to check site performance under normal & peak conditions
  + We did unit testing of all frontend & backend components, including our deployed ML models.
  + We did security testing for user authentication & verification
  + We did localization testing by trying out the site on various browsers & platforms
  + We did interface testing of the entire UI and have tried to keep things as easy to understand as possible
  + We did integration testing after connecting the 3 different components of our project - the UI, database & the on-the-fly ML models