System and Unit Test Report

NBAStat
Justin^2
12/2/19

User Stories:

Sprint #1:

1. “As a beginning developer, I want a backend set up so that I can host the website.”
    - We used Django to handle integration between python data and html pages.
    - We tested the Django endpoints using a wide range of values created from python scripts to ensure that we understood how Django would handle different values. Such as integers, dictionaries, lists, strings, and even None values.

2. “As a beginning front end developer, I want to understand basic HTML/CSS so that we can implement an easy to use website.”
    - We used templates and edited them to fit our needs in the website design.
    - We tested our HTML/CSS design by running the website and finding the design of the website to be to our expectations. As we developed the website, we tweaked and modified the templates until the design fit our needs and wants.

3. “As a beginning graphic designer, I want to create a color palette to use for our pages.”
    - No testing necessary for creating a color palette that was used for our website.

Scenario #1:

- Design page layouts using photo editing software
- Use page layout designs as a reference
- Searched for HTML template pages
- Adapted relevant templates for our pages
- Created a python file to test relationship between datasets and Django’s creation of HTML pages using that data.
- Using host localhost and port 8000
- Checked the locally hosted webpage
- Verified that the data uploaded is the data provided by the example python3
- Example of data passed into Django: Lists, Dictionaries, ints, strings, and jsons

Sprint #2:

1. “As a backend developer, I want to set up Django to host our webpage, so that I can access it.”
    - We set up Django and compiled the website and found that it was working.

2. “As a front end developer, I want to realize our concept art, so that I have accessible and pleasing design.”
    - No testing was necessary for realizing the creation of html templates based on concept art for web pages.

3. “As a data analyst, I want to expose and prepare the data for database usage, so that our data look up are streamlined.”
    - We used the database for ID lookups of players to cut down on one API call.
    - We verified this by looking at the database in the Django admin page and could physically see the player names and IDs .

4. “As a designer, I want to create a examples of data visualization, so that the web developers have a reference.”
    - Testing for this story included creating web visualization examples using different datasets using FusionCharts.
    - We tested how FusionCharts expected data to be passed in, and fixed data formatting error as we went on, until we had a solid understanding of FusionCharts interactions.

Scenario #2:

- Installed the NBAstat api using pip
- Started the NBAStat app with Django runserver
- Went to admin page for Django which contains the database info
- Django Admin database info contains playerID’s and Player Names, and whether the player is active
- Edited the admin page for different players to verify that the changes were going through
- Used FusionChart program created in previous sprint as a reference for implementation on NBAstat application.

Sprint #3:

1. “As a front end developer, I would like to start listing the obtained data on the corresponding webpages.”
    - We used fusioncharts and html tables to display statistical graphs and year-by-year stats in a user-friendly and interactive way.
    - We verified this by running the website and finding that the website meet the standards we wanted it to.

2. “As a back end developer, I would like to begin implementing the data visualization aspect of each corresponding webpage.”
    - We used fusioncharts and html tables to display statistical graphs and year-by-year stats in a user-friendly and interactive way.
    - We verified this by running the website and finding that the website meet the standards we wanted it to.
3. “As a database engineer, I would like to be able to use the database for player name and id lookup.”
    - We used SQLite to store player names and IDs in a database for easy retrieval to make API calls.

Scenario #3:

- Started NBAStat app
- Typed a player name into the search bar
- Printed out database query results which gave us player id
- Website took us to the player page where we could see data visualization
- Verify that the data being presented on the page matched the data being provided by the API

Sprint #4:

1. “As a developer, I want to create a redirection page if the user types in an incorrect player name which will show similar player names.”
    - We typed a wrong name into the search bar and the website would redirect us to the results page with a dropdown menu of similar players

2. “As a database engineer, I would like to at the very least store persistent data into a database. And be able to access dynamic data easily and quickly.”
    - Player name and IDs were stored into the database, but the related data was not stored alongside it.
    - This was tested by checking whether using the IDs to fetch data would return the correct relevant data.

3. “As a frontend engineer, I would like to expose more spots on our HTML page to pull data from our backend as well as host basic data visualization with tables and graphs.”
    - We made multiple fusioncharts with data we thought to be the most important statistics in basketball which PPG and FG, per season.

Scenario #4:

- We started the NBAStat app
- We typed an incorrect name in the search bar
- We picked a player in the dropdown menu and were redirected to the player page
- On the player page we saw multiple fusion charts with the data we expected