System and Unit Test Report

Atiq S. Brian B. Carl R. Young M. Yuan J.

CUAH - UCSC Academic Helper

Unit Test

- Frontend
    - Essential extensibility of server-client and usage of request Ajax, it is necessary that the Frontend is able to send the specific requests needed and able to retrieve a JSON object, the Unit Testing for the Frontend side is similar to the System Test Scenario unless there was a prototype for the backend. The frontend side was able to Unit Test successfully in the scope of the database running and server-client interaction working from both the Backend and Database sides
    - Testing Frontend User-Interface
        - Open html Frontend
            - Verify that UI is up and connected to scripts.js/style.css
    - Testing Frontend Requests are able to send
        - Open html Frontend
        - Click on one of the three buttons listed (CS/CE/EE)
            - Verify that a table format is then created on the UI
    - Testing Frontend Request Headers
        - Open html Frontend
        - Click and verify checkboxes in different combinations
        - Click on one of the three buttons listed
            - Verify that a table is displayed X amount of classes with flags in accordance to the Schematic
                - eg: GE checkbox will display GE tags, List all will show all courses, combinations of it will stack in accordance to logic
    - Sprint 1-3, Manual testing: Open html Frontend
        - Open scripts.jss and set an alert('ok') in $('#CS').click(function(e) in the case-statement (error/success)
        - Verify that the .click is able to reach the branch
    - Functions in scripts.jss
        - $(document).ready(function()) – Encompasses the scope of frontend
        - var user – holds an array key/value to checkbox element and a booleancheck array indexed to previous array function
        - check_boxes(user, booleancheck) – compares checkboxes actives/lows and changes booleancheck values to match required Schematic request headers for the backend
        - function ajax_pdf(Major) – http GET request for checkbox pdf of specified Major (button clicked)
        - function ajax_request(Major, booleancheck) – Sets up ajax request in accordance to a request
        - .click() – calls check_boxes function, and if-else statement for sending either a http GET PDF request or http GET request with specific headers

- Backend
    - Due to the extensibility of server-client interaction and the usage of json, each user story is essentially a different combination of request headers. Hence testing procedure for each user story is universal, as long as the correct output is pre-determined for comparison.
    - Testing Prasing of incoming request from frontend for each User Story:
        - Start up backend instance
        - Using Curl utility, send a series of headers that would represent each user story's desired filtering effect to the backend
        - Print out parsed request conditions and compare with desired
    - Testing Response composing to frontend for each user story
        - Run and update the database instance in directory backend/db with "make update"
        - Follow procedure for Testing Parsing
        - Redirect http output to stdio/file
        - Compare with desired
    - Testing of total backend functionality
        - Follow procedure Testing Response composing
        - Reset
        - Manually go to localhost:5001, and input as normal
        - Observe response on webpage, compare with terminal output.

- Database
    - Connecting to database from a separate Python file, using psycopg2 library, and retrieve all tables within database
    - This was done continuously throughout all 3 Sprints, and with the same goal in each.
    - This mimics the interface of what the backend would do to access the database when it services the frontend requests.
    - Upon each update of the database, use script to establish access to the database and retrieve all entries. Verify with the original dataset.