CAT THINGS

Test Plan

Nicholas McHale, Andrew McCann, Susmita Awasthi, Manpreet Bahl, Austen Ruzicka, Luke Kazmierowicz, Hillman Chen

Introduction

The purpose of this document is to detail the testing for the development of the inventory system. This includes test plans for the front-end and the API. Ensuring that the API functions critically is top priority as it obtains the information from the database for the front-end to display to the user. The main constraint for testing is the setup of the database and the schema, tables, and views it will have which can be seen in the Database design document.

Features to be Tested:

The features listed below will be tested extensively to ensure that they work properly:

1) View Inventory
2) Checkout Item
3) Checkin Item
4) Request Item
5) View Transaction History
6) View Statistics
7) Add New Item
8) Generate Shopping List

Approach

The front-end will be tested manually as pages are developed, make API calls, and whether it displays the information from the API correctly and thus will utilize the Console in popular browsers such as Google Chrome. Another form of testing that will be used extensively on all aspects of the project will be code reviews. This will be done through pull requests when features are completed and ready to be merged into the master branch. The API will be tested utilizing Postman where all routes will be stored in a collection and share an environment for global variables. The reason for environment is to allow all routes to pass in appropriate authentication tokens as well as store the item ID of the test item that is created so that checkout and checkin can be performed on them.

Item Pass/Fail Criteria

The criteria for an item passing or failing tests are straightforward. The tests for the API routes contain a simple test to check whether the response code received from the API is 200 if the test is expecting the request to pass or an error code (404, 500 etc.) if the test is expecting the request to fail. The environment for testing the API also allows you to view the information that was received from the API in addition to the status of test to help with debugging in the event that a test fails. The testing of the API also tests the SQL queries that are being used on the database and whether the current schema is suited to the design.

With the API being tested using Postman, it ensures that the data being received by the front-end will be correct, giving a starting point to look at the front-end should an error occur while development. It’s also important to note that the tests run on the API in Postman will also consist of passing in bad data such as wrong credentials for authentication or negative values when checking out items.

Test Deliverables

In addition to this document, there will be a Postman collection and environment file that will be available to test the API routes. The tests will display the information that the API retrieved from the database as well as header information and a test to check the response code. This should be sufficient for testing the API. As mentioned before, testing for the front-end will be doing manually and through code reviews and thus there are no scripts to be delivered for testing the front-end.