Test Plan Document

“AutoBuild” - Automatic PC Part Builder

October 24, 2020

Saturday Solution

Connor Kobel	(015036474)
Daniel Gulland	(018648771)
Ian Ho-Sing-Loy	(026339220)
Nickolaus Marshall-Eminger	(025567241)
Sirage El-Jawhari	(018359144)
Zeinab Farhat	(017990360)

Table of Contents

1 Document Revisions

Date	Version	Changes
10/24/2020	0.1.0	Initial Draft formatting.
10/28/2020	0.2.0	Update format, input data.
10/29/2020	0.3.0	Change format, update content, clean up.
10/30/2020	1.0.0	Added test policy, feature level scenarios, and tests to be performed.

2 Test Policy

Test Scope

In-Scope

- Unit Testing
- Integration Testing
- End to End Testing

Out-of-Scope

- Performance Testing
- Penetration Testing
- Regression Testing

General Test Policy

Unit Testing

All parts of a class (such as methods or constructors) must pass unit tests before further testing is to be performed. To pass unit testing each method, call, etc. must result in valid expected results, be they success or fail scenarios. If the test returns successful, for example in the case of a bad input correctly being caught, that test is passed. We aim for all tests to return their expected results.

Unit testing will be performed on each sprint with the completion of a new function, series of functions, or the completion of a class as a whole.

Each module undergoing unit testing must take no longer than 10 seconds to complete a test suite. This number is subject to change, but to accommodate expected usability for users, upon final release, code may need to be optimized to fall under this time. Ideally a module's unit test should take no more than 1 second to complete.

Integration Testing

Integration testing will be used to determine if components of our web app are compatible and work together as intended. This is not at a system-wide scale as end-to-end testing is, This is for testing done on features that require multiple components but not necessarily the entire system.

Integration testing will be performed on individual features and components that require it. This will make sure that components work well together on a feature level.

This is a test of compatibility between modules rather than a test of flow, which is meant for end to end.

End to End Testing

The entire web application will be tested from the front end to the back end. This process will ensure that all parts and features work together and work as intended.

End to end testing will work as a story. Testing the flow of a scenario from one end to the other end. Not necessarily testing the communication between two modules, but more of the flow from start to finish.

The feature that partakes in end to end testing will also be integrated with the UI feature. For this case UI will not be tested alone but implied with other feature level end to end tests.

Test Data

Test data will consist of using mock, fake or simulated data entries and user actions. All tests will be automated.

Results from testing, both successful and failed tests will be recorded and modified until all tests are proven successful. Failed testing data will be useful in determining the origin of the issues being faced.

3 Feature Level Scenarios

Logging

Test Creation: 18 hours
Test Execution: 2 hours
Total Test Time Estimate: 20 hours

Logging will be tested along with every other feature, therefore the amount of logging specific testing will be narrowed down to a basic end to end test of verifying that data has been logged to the database.

End to End Testing

- **Expected**:
    - An action or data is correctly logged to the database
- **Failure**:
    - An action or data is not logged to the database.
    - An action or data is not accurately logged to the database.

Archiving

Test Creation: 2 hours
Test Execution: 1 hours
Total Test Time Estimate: 3 hours

Just as logging, archiving is a backbone component of our web app, therefore, archive testing will be narrowed down to the simple action of logs being moved to archives after a certain time frame.

End to End Testing

- **Expected**:
    - Logs that have extended the lifespan of one day will be moved to archives.
- **Failure**:
    - Logs don't move to archives after one day.
    - Logs are not properly carried over to archives.

Datastore

Test Creation: 8
Test Execution: 1
Total Test Time Estimate: 9

Datastore will be tested integrally with other features as well as work in with other features' end to end tests. Therefore, verifying datastore integrity is the sole purpose of datastore testing.

End to End Testing

- **Expected:**
    - The datastore will store all data.
    - The datastore will store all data required for feature functionality.
    - The datastore will store all necessary data that is produced from certain features.

- **Failure:**
    - The datastore does not store data.
    - The datastore jumbles data and inaccurately stores data.
    - The datastore stores unnecessary data that is not needed for future querying.
    - Features that access data in the data store will not alter the data, causing loss of data integrity.

UI

UI will be tested with the system-wide End-to-End testing, there is no UI specific testing to be done.

User Access Controls

Test Creation: 15 hours
Test Execution: 4 hours
Total Test Time Estimate: 19 hours

End to End Testing

- A vendor account attempts to edit a different vendor account's page.
    - **Expected:**
        - The vendor account is prompted by an error message stating that they don't have permission to edit that page.
        - No changes are made to the data store.

    - **Failure:**
        - Any section on a vendor account's page is edited and those changes are successfully saved in the data store.

- A user attempts to make an action that they don't have permission for. Or the user attempts to use a functionality that they don't have permission to use
    - **Expected**:
        - The user is not given the option to access that action. Certain actions that are not offered to all users are not given, meaning they have a 'navigation' end that is unaccessible.
        - The user is not given access to view data that may result from an action that they are not permitted to perform.
        - The user is prompted with a message that they can't perform that action.
        - The user is prompted to sign up or sign in if they want to access a certain action.
    - **Failure:**
        - The user is able to perform an action that they are not permitted for.
        - The user is not prompted with a message that they are unable to perform an action.
        - The user is not prompted with a message asking them to sign in/sign up to gain additional access.

User Management

Test Creation: 25 hours
Test Execution: 3 hours
Total Test Time Estimate: 28 hours

Integration Testing

- An administrator attempts to add a user.
    - **Expected:**
        - A user account record is stored in the data store which saves the user's email and a hash value of their password.
        - A user account already existing in the datastore cannot be duplicated.
    - **Failure:**
        - A record isn't stored in the data store.
        - Duplicate user records are stored into the datastore

- An administrator attempts to delete an account.
    - **Expected:**
        - The selected user account record is deleted from the data store.
    - **Failure:**
        - The selected user record isn't deleted from the data store.
        - The incorrect user is deleted and not the one intended for deletion. i.e. username is not case-checked, therefore a user with a similar name but not identical is deleted instead.

- An administrator attempts to create a vendor account.
    - **Expected:**
        - A vendor account record is stored in the data store which saves all relevant vendor information.
        - All necessary permissions to the vendor account will be given to the new vendor account.
    - **Failure:**
        - A vendor account record is not stored in the data store.
        - A user account record is stored in the data store instead.
        - The incorrect account type is associated with the created vendor account upon creation. i.e. "basic user'.
        - Vendor functionalities are not offered/ accessible to the vendor account.

- An account holder attempts to delete their account.
    - **Expected:**
        - The account holder record is deleted from the data store and is simultaneously logged out.
    - **Failure:**
        - The account holder record remains in the data store.
        - The user remains logged in after successful account deletion

End to End Testing

- An administrator attempts to reset a user's (an account holder) password.
    - **Expected:**
        - A password reset link is sent to the preferred account holders email.
        - The password reset link sent should be a valid link.
        - The password reset link should expire within 15 minutes
        - The new password associated with the account should not be the same as the old password.
        - Upon completed password reset, the new password should be updated in the accounts holder records.
    - **Failure:**
        - No password reset link is sent to the account holders preferred email.
        - The password reset link is sent to an unintended account holder.
        - The new password is not stored in the datastore
        - The new password does not update the old password

Registration

Test Creation: 20 hours
Test Execution: 5 hours
Total Test Time Estimate: 25 hours

Integration Testing

- User attempts to register for an account and enters a valid email, valid username, and valid password.
    - **Expected:**
        - A user account record is stored in the data store which saves the user's email, username, and a hash value of their password.
        - The user is redirected to the home page.
        - A verification email is sent to the user's email.
    - **Failure:**
        - A user account record isn't stored in the data store.
        - Incorrect user account record is stored in the data store
        - The account holder never receives a verification email
        - An invalid username, password, or email is input and a user account record is created and stored in the data store.

- User attempts to register for an account and enters a duplicate username.
    - **Expected:**
        - The user is prompted by an error message telling them that the username is already in use. No account should be created.
        - The information entered should enforce a save state so that the user can apply fixes without re-entering all other correctly entered information once again.
    - **Failure:**
        - No error message is displayed.
        - An account is created and the record is stored in the data store.
        - Previously entered information is not saved.

- User attempts to register for an account and enters a duplicate email.
    - **Expected:**
        - The user is prompted by an error message telling them that the email is already in use. No account should be created
        - Save state should be applied so the user can apply fixes without re entering all other correctly entered information once again.
    - **Failure:**
        - No error message is displayed.
        - An account is created and stored in the data store resulting in two users with same email

- User attempts to register for an account and enters an invalid username.
    - **Expected:**
        - The user is prompted by an error message telling them that the username is invalid. No account should be created.
        - Save state should be applied to so users do not need to reenter the same information.
    - **Failure:**
        - No error message is displayed.
        - An account is created and stored in the data store resulting in two users with same usernames

- User attempts to register for an account and enters an invalid email.
    - **Expected:**
        - The user is prompted by an error message telling them that the email is invalid. No account should be created.
        - A save state should save information so that a user does not have to re-enter all information again.

    - **Failure:**
        - No error message is displayed.
        - An account is created and stored in the data store.

- User attempts to register for an account and enters a password that doesn't follow the password policy.
    - **Expected:**
        - A notification is prompted to the user stating that the password doesn't follow the password policy. Our password policy is displayed to the user.
    - **Failure:**
        - No error message is displayed.
        - An account is created and stored in the data store.

Login

Test Creation: 8 hours
Test Execution: 1 hour
Total Test Time Estimate: 9 hours

Integration Testing

- User attempts to login by entering a correct username and password combination.
    - **Expected:**
        - The system will verify that the username and password match with what is stored in the database and will log the user into the system and take them to the homepage.
    - **Failure:**
        - The username and/or password cannot be verified from the database.

End to End Testing

- User attempts to login by entering a correct username but incorrect password.
- User attempts to login by entering an incorrect username but a correct password.
    - **Expected:**
        - The system will give the user an error message stating that the username password combination is incorrect.
        - It will stay on the same page and allow the user to re-enter the username and password.
    - **Failure:**
        - No error message is displayed

- User fails to login 5 times in a row and attempts to login a 6th time.
    - **Expected:**
        - The system will lock out the user so they cannot try anymore login attempts for 30 minutes.
    - **Failure:**
        - The user is able to attempt a login before 30 minutes has passed since their last failed login attempt.

Logout

Test Creation: 8 hours
Test Execution: 1 hour
Total Test Time Estimate: 9 hours

Integration Testing

- If the user closes the application or walks away while logged in.
    - **Expected:**
        - After 1 hour of inactivity, the user is automatically logged out.
    - **Failure:**
        - The user does not automatically log out after 1 hour of inactivity.

- A user attempts to logout.
    - **Expected:**
        - When the user attempts to log out, it logs them out of the system and returns them to the homepage.
    - **Failure:**
        - The user is still logged in after an attempted log out.

- The user loses internet connection.
    - **Expected:**
        - If a user's connection is lost and is not reconnected within 1 hour then they are automatically logged out.
    - **Failure:**
        - If a user loses connection and is not automatically logged out after 1 hour.

User Analysis Dashboard

Test Creation: 35 hours
Test Execution: 3 hours
Total Test Time Estimate: 38 hours

Integration Testing

- Dashboard correctly outputs the calculation of metrics to be measured within our our system
    - **Expected:**
        - the calculation on the data retrieved is correctly done, For example: The percentage of usage for a particular component.
    - **Failure:**
        - The data calculations are not understandable
        - The data calculated has no output, just garbage
        - The data calculated gives negative results.

End to End Testing

- An admin selects a graph to view.
    - **Expected:**
        - The admin is shown the proper graph that they selected to view.
    - **Failure:**
        - The graph the admin selected is not shown.
        - The admin is not shown the correct graph.
        - The correct visualization is not shown given the calculated values (graph is inaccurate, showing wrong values or displaying a readable graph).
        - Account's sensitive information is logged or represented somehow on the graph or in the analysis.

- An admin updates or refreshes a visualization.
    - **Expected:**
        - The graph will be refreshed and updated with the current information. (Example: if a user creates a new registration then that means logging that information will alter the current number of account holders with autobuild)
    - **Failure:**
        - The graph is not updated. New information is not retrieved upon refresh of the user analysis dashboard by an admin.

- The data required for the User dashboard is stored/ present in the database for retrieval.
    - **Expected:**
        - The necessary data for the dashboard can be queried.
    - **Failure:**
        - The data needed is not being logged by the logging feature
        - The data that is meant to be queried for the dashboard is not present in the datastore or the information is insufficient.

Components Catalog

Test Creation: 45 hours
Test Execution: 5 hours
Total Test Time Estimate: 50 hours

Integration Testing

- Account holders tries to apply the PC upgrader to a product within components catalog
    - **Expected:**
        - The PC upgrader option is accessible from the components catalog for account
        - Account holders are able to apply the PC upgrader to the product.
    - **Failure:**
        - Account holders are unable to apply the PC upgrader to the product.
        - The PC upgrader calculations are producing non-optimal upgrades.
        - The PC upgrader is applied to the wrong product.

- Account holder tries to save the product to their user shelf which is located within “User's Garage”.
    - **Expected:**
        - Account holders are able to add a product to their shelf.
    - **Failure:**
        - Account holders are unable to place the product on their shelf.
        - The saved product is not shown in the account holders shelf.
        - The wrong product is added to the account holder shelf.
        - The product is added to the wrong account holder shelf.

- An unregistered user tries to save a product to the user shelf.
    - **Expected:**
        - The user is prompted to log in or register for an account. They may also choose to cancel the prompt and return to the previous page if they do not wish to sign in / register.
    - **Failure:**
        - The user is not prompted with a sign in, register or cancel dialog.

End to End Testing

- User selects search filter
    - **Expected:**
        - A new result set is generated using the specified filter.
        - Web page is updated to display the result set.
        - Search bar shows possible autocomplete results.
    - **Failure:**
        - The wrong filter is applied.
        - No filter is applied.
        - The result set is not generated.
        - The web page does not update to display the result set.
        - The web page updates with the wrong result set.

- User searches for a product
    - **Expected:**
        - A new result set that corresponds to the search string is generated.
        - Web page is updated to display the result set.
    - **Failure:**
        - The result set does not closely resemble the search string.
        - The search is not executed.
        - The result set is not generated.
        - The webpage does not update to display the result set.
        - The webpage updates with the wrong result set.

- User selects a product to view.
    - **Expected:**
        - A product details overlay screen is displayed.
    - **Failure:**
        - The overlay screen does not appear.
        - The overlay screen displays details of a different product.
        - The dimensions of the overlay screen do not allow the details to be displayed properly.

- Option to return to webpage
    - **Expected:**
        - The overlay screen is closed.
        - User is returned to the version of the webpage before the user selects the product.
    - **Failure:**
        - The user does not exit the overlay screen
        - The user is not redirected to the webpage he/she left

- Vendor List
    - **Expected:**
        - A list of vendors is displayed in the overlay screen.
        - Users can be redirected to the vendor's website if the product is available.
        - Users can sign up for email notifications if the product becomes available by the vendor.
    - **Failure:**
        - **User visits product on vendor website**
            - The vendor list does not appear in the overlay screen.
            - Users are unable to visit the product on the vendor's website after clicking on the redirecting "buy" button.
            - Users visit the product on the wrong vendor's website.
        - **User sign up for email notifications**
            - Users are unable to sign up for notifications.
            - Users sign up but receive no notifications.
            - The wrong user is notified.
            - The user is notified for the wrong product.
            - The user is notified for product availability from the wrong vendor.
            - The user is notified for product availability when it is not.

Vendor Linking

Test Creation: 40 hours
Test Execution: 4 hours
Total Test Time Estimate: 44 hours

Integration Testing

- A vendor account adds an item to their profile page with a valid discount and a valid price.
    - **Expected:**
        - An item record linked to the vendor is stored in the data store.
    - **Failure:**
        - No record is stored in the data store.

- A vendor adds an item to their account with an invalid discount or invalid price.
    - **Expected:**
        - An error message is prompted to the vendor stating that an invalid input was made. No item record is stored in the data store.
    - **Failure:**
        - No error message is prompted to the user.
        - An item record with the invalid input is added to the data store.

- A vendor edits an item on their profile page with a valid discount and a valid price.
    - **Expected:**
        - The selected item record is edited in the data store.
    - **Failure:**
        - The selected item record is not properly updated in the data store.
        - The updated item and the old item record co-exist in the datastore

- A vendor edits an item on their profile page with an invalid discount or an invalid price.
    - **Expected:**
        - An error message is prompted to the vendor stating that an invalid input was made. The selected item record is not updated in the data store.
    - **Failure:**
        - No error message is prompted to the user.
        - The selected item record is edited in the data store.
        - The updated item and the old item record co-exist in the datastore.

- A vendor deletes an item from their profile page.
    - **Expected:**
        - A confirmation message is displayed to the vendor to confirm they want to delete the selected item.
        - The selected item record is deleted from the data store.
    - **Failure:**
        - No confirmation message is shown.
        - The selected item record is not deleted from the data store.
        - The deletion results in another item to be deleted that was not intended.
- Account holders navigate to a vendor profile page.
    - **Expected:**
        - The correct vendor's information is displayed.
        - The account holders can only view the profile page.
        - The list of components that are associated with the vendor are properly retrieved from the data store for viewing.
    - **Failure:**
        - A different vendor's information is displayed.
        - The list of components fails to load or retrieves incorrect information.
- A vendor deletes their account.
    - **Expected:**
        - The vendor account record and all records related to the selected vendor are deleted from the data store.
        - The account is successfully logged out.
    - **Failure:**
        - The vendor account record and related records remain in the data store.
        - The vendor account remains logged in.

End to End Testing

- An API request is made.
    - **Expected:**
        - The correct information is displayed from the API request.
    - **Failure:**
        - Incorrect information is displayed from the API request.

Builder

Test Creation: 50 hours
Test Execution: 10 hours
Total Test Time Estimate: 60 hours

Integration Testing

- A user selects either Basic, Gaming, or Graphic Intensive for their build which changes the focus of the components formula for the builder.
    - **Expected:**
        - The system (builder feature) navigates to the peripherals selection page.
        - The selection is put into the build formula.
    - **Failure:**
        - Selection has no impact on the build criteria formula.
        - No options for Basic, Gaming, or Graphic Intensive are displayed.

- A user selects which peripherals they wish to include (the user does not have to select any peripherals if they wish).
    - **Expected:**
        - A warning message will be given to the user that they should keep in mind their chosen peripherals when they are selecting the price for their build
        - The selection is put into the build formula.
        - The system navigates to the PSU type selection.
    - **Failure:**
        - Selection does not affect the build formula.

- A user selects their PSU type.
    - **Expected:**
        - The system navigates to the hard drive selection.
        - The selection is put into the build formula.
    - **Failure:**
        - The selection does not affect the build formula.

- A user selects their hard drive type and quantity.
    - **Expected:**
        - The system navigates to the price selection page.
        - The selection is put into the build formula
    - **Failure:**
        - The selection does not affect the build formula.

- There are no results that fit the price from the user. (IE: The user only inputs $1)
    - **Expected:**
        - The user is given an error message that no builds are possible with the selected price.
    - **Failure:**
        - No error message is given.

End to End Testing

- A user enters a valid maximum price for the builder.
    - **Expected:**
        - Input is accepted and the build starts.
    - **Failure:**
        - The build is unsuccessful.

- A user enters an invalid maximum price value or leaves it blank.
    - **Expected:**
        - The user is given an error message and they will be allowed to re-enter their price.
    - **Failure:**
        - No error message is given.

Upgrader

Test Creation: 25 hours
Test Execution: 5 hours
Total Test Time Estimate: 30 hours

Integration Testing

- The user selects a component to upgrade that is already the most optimal or best component (ie: rtx 3090 for GPU)
    - **Expected:**
        - The user gets a message that tells them there are no components better than the one currently in their build.
    - **Failure**:
        - The upgrader shows components that are worse than or duplicate to the pre-existing component.

- A user enters a valid maximum price for the upgrader.
    - **Expected:**
        - Input is accepted and the build starts.
    - **Failure:**
        - The build is unsuccessful.
        - The build calculates based on the maximum price, but outputs a non-optimal build.

- A user enters an invalid maximum price value or leaves it blank.
    - **Expected:**
        - The user is given an error message and they will be allowed to re-enter their price.
    - **Failure:**
        - No error message is given.
        - The upgrader calculates a result given the entered invalid price value.

End to End Testing

- A user enters a too low maximum price for the upgrader feature.
    - **Expected:**
        - The user gets a message that tells them there are no components that fit the criteria that was input.
    - **Failure:**
        - The upgrader shows components that are more expensive than what the user entered. (performs invalid calculations)

Most Popular PC Builds

Test Creation: 30 hours
Test Execution: 5 hours
Total Test Time Estimate: 35 hours

Integration Testing

- Add build to user garage
    - **Expected:**
        - Account holders can add a build to their user garage.

    - **Failure:**
        - Published builds are added to the wrong user garage.
        - Published builds are not visible after adding it to the user garage.
        - Build is not saved to the garage. Garage or database might be full. Server might be shut down.

- Comment on Build
    - **Expected:**
        - Comments can be posted for a specific build.
    - **Failure:**
        - Account holders can post null comments.
        - Comments are posted to the wrong build.
        - Comments for a build are not visible.

- Publish builds
    - **Expected:**
        - Starting account holders are allowed to publish a maximum of three builds per day.
        - Reputable account holders can publish up to six builds per day when they reach a level of notoriety.
        - Each build is composed of a screenshot of the build and the build name
        - Published builds cannot be edited.
    - **Failure:**
        - Users cannot publish the build. Might be because the database is full or the server is down.
        - The wrong build is published.
        - Starting account holders publish more than three builds per day.
        - Reputable account holders publish more than six builds per day.
        - Build from another user garage is published.
        - Build name is null.
        - Screenshot is below 72 DPI.
        - The published build does not show up in the Most Popular Builds webpage.

End to End Testing

- User selects search filter
    - **Expected:**
        - A new result set is generated based on the selected filter.
        - Web page is updated to display the result set.
        - Filter by gaming, productivity, and animation.
    - **Failure:**
        - The wrong filter is applied.
        - No filter is applied even after being selected.
        - The result set is not generated.
        - The web page does not update to display the result set.
        - The web page updates with the wrong result set.

- Search for a build
    - **Expected:**
        - A new build result set that corresponds to the search string is generated.
        - Web page is updated to display the result set.
    - **Failure:**
        - The result set does not closely resemble the search string.
        - The result set is not derived from published builds.
        - The search is not executed.
        - The result set is not generated.
        - The webpage does not update to display the result set.
        - The webpage updates with the wrong result set.

- Upvoting and Downvoting
    - **Expected:**
        - Account holders can upvote or downvote a specific build.
    - **Failure:**
        - Account holders can upvote and downvote at the same time.
        - Account holders can upvote and downvote multiple times.

User's Garage

Test Creation: 41 hours
Test Execution: 4 hours
Total Test Time Estimate: 45 hours

Integration Testing

- An account holder creates a build manually in the garage and attempts to save their progress.
    - **Expected**:
        - The build is saved in the datastore under the account holders private data within 5 seconds and returns the user to their main garage page, showing the build in their garage.
    - **Failure**:
        - The build does not save within 5 seconds
        - The save is not placed into the users private data, but a public place.
        - The build does not save at all, this could be due to the garage being full, the database being full, or the server being down.

- An account holder attempts to duplicate a build inside the user garage when the user is not out of save spaces (current maximum is 50 builds).
    - **Expected**:
        - The user's selection is duplicated, the datastore is updated, the garage is populated with the new clone, and the user can interact with the clone freely without modifying the original.
        - Replication should take less than 5 seconds.
    - **Failure**:
        - The original is a reference instead of a new duplicate copy, making the original modifiable from the clone.
        - The build takes longer than 5 seconds to replicate.
        - The build is not replicated into the datastore.

- The account holder has the maximum number of builds allowed in their garage and attempts to create and save a new build.
    - **Expected**:
        - The account holder will be unable to save a newly created build.
        - The intended save feature will be disabled with a notification that the user is out of space.

    - **Failure**:
        - The user is not prompted that they have reached the maximum allowed simultaneous saves count.
        - The save option is available to the user despite the limitation being reached.
        - The user is able to save to their garage, beyond the save limit.

- A user attempts to view product specific details.
    - **Expected**:
        - The user should be able to view the product details in full, as well as listings of vendors with availability, in an overlay with redirect options.
        - Product details should load in under 5 seconds.
    - **Failure**:
        - Details are not related to the product being viewed, as in the incorrect product is shown.
        - The overlay is hidden and does not show or allow the user to view in the catalog.
        - Listings take longer than 5 seconds to show without informing the user an error has occurred.

- A user attempts to view a product in the catalog rather than in the current view.
    - **Expected**:
        - The user is redirected from their current view to the catalog page with the correct product details shown.
        - This redirect should happen in under 5 seconds.
    - **Failure**:
        - The user is redirected to an external website.
        - The user is redirected to the wrong product.
        - The redirect takes longer than 5 seconds.

- The account holder attempts to add products to a shelf.
    - **Expected**:
        - The desired item is added to the shelf, saving a reference to the item for future viewing by the user. This save addition should take no longer than 5 seconds.

    - **Failure**:
        - The wrong item is saved to the shelf.
        - The user cannot save an item to their shelf.
        - The item is not viewable after the addition occurs, but the reference is viewable in the datastore.
        - The user is not informed the item has been added when successful and to which shelf.
        - The addition takes more than 5 seconds.

End to End Testing

- An account holder attempts to manually organize their garage by moving builds from one location to another.
    - **Expected**:
        - The user's selection is placed where the user intends it to be, and the new location is saved in the datastore permanently.
        - Datastore updates should be done in under 5 seconds.
    - **Failure**:
        - User's selection is not correctly identified.
        - The location of the build is not correctly saved and resets to its previous location, either upon reload, or immediately.
        - Swaps do not occur and save in under 5 seconds.
        - The build ceases to exist upon movement, or placing in an unexpected area.

- The account holder has the maximum amount of builds in their garage and attempts to duplicate a build.
    - **Expected**:
        - The account holder is not given the option to duplicate a build, and instead is prompted that their garage is full on this account.
    - **Failure**:
        - The user is not prompted that they have reached the maximum allowed simultaneous saves count.
        - The duplicate option is available to the user despite the limitation being reached.
        - The user is able to duplicate a build in their garage, beyond the save limit.

- The account holder moves a build in the garage while at maximum capacity.
    - **Expected**:
        - The user's selection is placed where the user intends it to be, and the new location is saved in the datastore permanently.
        - Datastore updates should be done in under 5 seconds.
    - **Failure**:
        - User's selection is not correctly identified.
        - The location of the build is not correctly saved and resets to its previous location, either upon reload, or immediately.
        - Swaps do not occur and save in under 5 seconds.
        - The build ceases to exist upon movement, or placing in an unexpected area.
        - The build is moved beyond the 50 save locations.

- The account holder attempts to publish a fully completed build to the most popular build page.
    - **Expected**:
        - A copy of the build is made and posted to the most popular builds page for ratings and reviews, this copy cannot be edited any further by the publisher.
        - The process of publishing should take the system no longer than 5 seconds.
        - Upon completion the user should be able to view the published build in the popular builds section.
    - **Failure**:
        - The publisher can modify the original and accidentally modify the published build.
        - The published build does not show in the popular builds section.
        - The publication process takes longer than 5 seconds after submission.
        - The published build is removed from the user's saves.

- The account holder attempts to edit/modify an existing build inside their garage.
    - **Expected**:
        - The user can change the title of a build, or any such interactable content, without creating a duplicate, or a new build.
        - All changes should be saved upon user's request within 5 seconds of the request.

    - **Failure**:
        - Alterations made are saved without the user performing a save function.
        - Saves take longer than 5 seconds.
        - Changes made create duplicates.
        - Creates a duplicate build when temporary modifications are made.

- The account holder attempts to edit or delete a build they published.
    - **Expected**:
        - The user should have no ability/access to modify or delete the build, or any content associated with it (such as reviews and ratings).
    - **Failure**:
        - The user can modify the published content.
        - The user can delete published content.
        - The user can modify content associated with the build.

- A user wishes to be directed to a sales vendor for purchasing the item they are viewing.
    - **Expected**:
        - The user is redirected to the appropriate website, listing, and relative information on our website matched the listing and product details.
        - This redirect should not take more than 5 seconds, provided the other website is not down.
    - **Failure**:
        - The product is not the same one the user is attempting to view.
        - The product is not even related to the one the user is attempting to view.
        - The website they are redirected to is no longer active, or is currently down.
        - The website they are redirected to is not the expected website (Such as BestBuy.com but instead redirects to google.com)
        - The redirection takes longer than 5 seconds and the user is not informed as to the progress of the redirection.

- The account holder attempts to delete a build or builds from their garage.
    - **Expected**:
        - The user is able to delete the selected build, or builds, and it is/they are removed from the user's private storage, housed in the datastore, in under 5 seconds.
    - **Failure**:
        - The deletion fails to remove the build(s) from the user's garage view.
        - The deletion fails to remove the build(s) from the datastore, but remains in the garage view.
        - The build(s) are not actually present in the datastore.
        - The deletion removes a non-related build, even from another user.
        - In the case of a deletion failure, the user is not informed.
        - The deletion takes longer than 5 seconds but does not inform the user that the deletion was successful.

- The account holder attempts to delete a build with no builds in their garage.
    - **Expected**:
        - The user is not given the option to delete any build(s), and is informed there are no objects to delete, if the option becomes available.
    - **Failure**:
        - The user is not informed, in a delete attempt, there are no objects to delete.
        - The user is given the ability to delete with no active objects.

- The account holder attempts to create a new shelf for item storage.
    - **Expected**:
        - The user creates a new shelf in the user garage that creates a datastore object that can be used for storing single items. It should take no longer than 5 seconds after submission to create.
    - **Failure**:
        - The datastore object is not created, so no items can be placed on it.
        - The user is not informed of a successful creation.
        - The creation takes longer than 5 seconds.

- The account holder navigates through multiple shelves, browsing items.
    - **Expected**:
        - The user successfully views items on a shelf, navigates to a new shelf, and can see only items that should be on that shelf.
        - New shelves or item details should load in under 5 seconds.
    - **Failure**:
        - When a user switches from one shelf to another the items viewable are a mix of both shelves.
        - When a user selects an item for viewing it loads an item in a similar position from another shelf.
        - A shelf does not load items that are intended to be there.(Only a partial list is shown.)

- The account holder attempts to move items from one shelf to another, and vice versa.
    - **Expected**:
        - Items should be transferred from the current shelf to the desired shelf.
        - Removes the reference from one shelf and adds it to the other in the datastore.
        - The addition and removal should take less than 5 seconds after request to finish.
    - **Failure**:
        - The items are duplicated to another shelf rather than removed.
        - The item overwrites another item in a shelf instead.
        - The process takes more than 5 seconds, or does not move at all.

- The account holder attempts to delete an existing shelf, along with the products saved on it.
    - **Expected**:
        - When the user deletes a shelf the datastore section containing all the data for that shelf should also be cleaned out. Preventing a memory leak.
        - Shelf deletion should take no longer than 5 seconds.

    - **Failure**:
        - When the shelf is deleted all references on the account to any products that were on that shelf are also deleted. (Meaning not just the specific occurrence on that shelf.)
        - The shelf does not remove the items from the datastore, creating a memory leak.
        - The operation of deleting a shelf takes longer than 5 seconds and the user is not informed whether the operation is successful or not.

- The account holder browses all their builds in their garage, viewing details one by one.
    - **Expected**:
        - The user, upon selection, is redirected to the builds information page and can view the details of a build, that is correctly associated with the selected build.
        - Details load within 5 seconds.

    - **Failure**:
        - The selected build does not load but another, wrong build, is loaded.
        - A selected build's details do not load.
        - A selection takes longer than 5 seconds to load and does not inform the user.

- A non-admin user attempts to modify or delete another user's saved builds.
    - **Expected**:
        - The non-admin user cannot perform any actions outside of viewing another user's garage. Retaining all data integrity for the account holder's garage.
        - The personal tool-kit provided to users to modify and maintain their garage should not be viewable to external users.
    - **Failure**:
        - A non-admin user has access to the account holders garage management tools, such as delete and modify.
        - A non-admin user viewing the garage sorts the garage to their liking, and the sort is saved to the datastore as the account holder's preference.

- An account holder sorts another user's garage while viewing its content.
    - **Expected**:
        - The sort occurs on the viewers end but is not saved to the data store, retaining the owner's preference.
        - The sorts should complete in under 5 seconds.
    - **Failure**:
        - The sorting the viewer used is retained and modifies the owners preferences.
        - Sorts do not complete in under 5 seconds.

User Reviews and Ratings

Test Creation: 10 hours
Test Execution: 2 hours
Total Test Time Estimate: 12 hours

Integration Testing

- An account navigates away from the page after typing up a review.
    - **Expected**:
        - The draft is automatically saved.
    - **Failure:**
        - The draft is not saved.
        - A draft is saved but the words or characters are omitted from it.

End to End Testing

- An account selects a star rating of 1-5
    - **Expected**:
        - The account selected user rating is successfully submitted, logged and is accurately visualized on the UI.
    - **Failure**:
        - The rating is not properly registered.
        - The rating is not properly logged.
        - The rating has been mix-matched with other rating numbers and does not accurately represent the rating that the account selected.
        - The rating is not properly visualized at the UI level (2 stars show up as 3, 5 stars show up as 4, etc.)

- An account tries to submit a review that consisted of over 10,000 characters.
    - **Expected**:
        - The review is not submitted, the user is prompted to continue editing the review.

    - **Failure:**
        - The review is submitted.
        - Some characters are not identified by the character checking implementation.
        - When an account submits a review of over 10,000 characters, the review is deleted without prompting the user to continue editing.

- An account tries to post an empty review.
    - **Expected**:
        - The review is not posted as the review content insufficient content. The account is prompted with a message notifying the account that the review is insufficient.
    - **Failure:**
        - An empty review is posted.
        - The account is not prompted with a message that notifies them of insufficient content.

- An account tries to submit a rating or review on a build more than once.
    - **Expected**: The account is prompted with a message that says they are not able to submit a review or rating on a build more than once.
    - **Failure:**
        - The account is able to submit an additional review or rating.
        - The account is not prompted with the expected message.

Support Page

The support page has been excluded. See project Plan for more details.

References

https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing

https://www.clariontech.com/blog/20-scenarios-for-testing-login-pages-and-search-functionality-on-websites

https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html

https://www.guru99.com/test-case-vs-test-scenario.html

https://www.softwaretestinghelp.com/what-is-integration-testing/