README3

Integration testing was manually done in a bottom up fashion. Each individual class was manually tested by unit testing before being used for integration testing. Much of our system is used in a linear fashion made for a very simple approach to integration testing. However, much in the same way as our unit testing, manual testing was done due to the challenges of dealing with web forms. The general flow of our integration testing will be as follows assuming each class has passed its unit testing; connection, then layout, then manage review, then list review, then compare review or view review in either order. This will follow all of the major use cases in our system and ensure that all classes can successfully interact with each other as specified in our interaction diagrams. This form of integration testing worked well for our system. These tests have changed very little since Demo 1, our initial batch of tests seemed to be sufficient for our system.

Add A Review

Test Covered Use Case: UC-1 Submit Review
Test Covered Classes: Layout, ManageReview, Connection

Assumption: There exists a database for the review to go into, the user is on the site

Integration Testing

Steps:

- Click the “Add Review” button
- Input information into the form
- Submit the review

Expected: The review is added to the database

Fails if:

- Form to add review does not appear
- Submitting review does not work
- Review does not appear in the database

View Reviews

Test Covered Use Case: UC-2 View Review
Test Covered Classes: Layout,List, View, Connection

Assumption: There exists a database the reviews are stored in, the user is on the site

Integration Testing

Steps:

- The user clicks the “View Review” button
    - If the webpage appears, the user clicks on a category to sort by it

Expected: All the reviews from the database are displayed. When the user clicks the “details” button, more details for a specific review appear.\

Fails if:

- The site is unable to access the database
- There is no information in the database
- The sorting functionality does not work or sorts improperly
- The reviews do not appear on screen

Compare Reviews

Test Covered Use Case: UC-3 Compare Review
Test Covered Classes: Layout, ManageReview, Connection

Assumption: There exists reviews in the database, the user is on the site.

Integration Testing

Steps:

- The user clicks the “Compare Reviews” button
- The user selects the complex to see reviews from

Expected: All the reviews for a specific housing complex are shown

Fails if:

- The reviews are not shown
- There are not reviews in the database for the complex requested
- The site is unable to access the database

View Aggregate

Test Covered Use Case: UC-4 View Aggregate
Test Covered Classes: Layout, List, View, Connection

Assumption: There exists data for a housing complex

Integration Testing

Steps:

- The user clicks the “Compare Reviews” button
- The user selects the complex to see reviews from
- The user selects the option to see aggregated data

Expected: Aggregated data for a specific complex is shown

Fails if:

- Data is not shown
- There are not reviews for the chosen complex
- The database is unable to be accessed

Compare Aggregate

Test Covered Use Case: UC-5: Compare Aggregate
Test Covered Classes: Layout, List, Compare, Connection

Assumption: There exist reviews for multiple housing complexes

Integration Testing

Steps:

- The user clicks the “Compare Housing” button
- The user selects the complexes to compare
- The user selects the option to view aggregate data

Expected: Aggregate data for the chosen complexes is displayed

Fails if:

- There are not multiple complexes to compare
- There are not reviews for each complex
- The site is unable to access the database

Login

Test Covered Use Case: UC-6 Login
Test Covered Classes: Layout, Login, Connection

Assumption: There exists a database of users, the user has an account

Integration Testing

Steps:

- The user clicks the “Login” button
- The user inputs their credentials
- The user clicks the submit button

Expected: The user is now logged in

Fails if:

- The user’s login credentials are wrong
- The database does not have record of the user
- The site is unable to access the database

Edit Review

Test Covered Use Case: UC-7 Edit Review
Test Covered Classes: Layout, ManageReview, List, Connection

Assumption: The user is logged in and has made a review

Integration Testing

Steps:

- The user clicks on their profile
- The user selects a review they have made from their profile page
- The user clicks the edit button
- The user edits the form that appears with the review’s information
- The user submits the form

Expected: The review in the database is updated

Fails if:

- The review is unchanged
- The user is not logged in
- The database is inaccessible