SW Engineering CSC 648/848

Specta Coin

Section 04, Team 04

Joshua Boelman - Team Lead
Jack Deremiah - Backend
Kenneth Galang - Github Master
Madeleine De Mesa - Scrum Master
Elizabeth Kirwan - Frontend
Tony Tran - Frontend

5/2/2022
Revision Table

Date	Revision	Additional Notes
04/31/22	Version 1.0	No test cases.
05/2/22	Version 1.5	5 test cases. Applied Feedback to pages Quick stats working on nav bar.

CSC 648/848 Milestone 4: Testing

Write this in style targeted for wide audience: executives, marketing, customers, beta testers.

The list of final functions will be checked on your final delivery for functionality and correct operation as you specified them. Failure to deliver complete list of these committed functions may result in reduced grade.

Product: SpecataCoin

Features

- Searching for specific cryptocurrencies
    - Displaying the current value of said currencies
- Creating/logging in accounts
    - Saved accounts onto our database
    - Log in using magic links sent to their email
        - Clicking a link will send them to the site in a logged-in state
    - Searching and tracking currencies
        - Card-Based
        - Amount owned
        - current value
        - Approximate total value
        - Management of their current tracked currencies
    - Portfolio Data (QuickStats)
        - Measures total value of portfolio
        - Majority currency
        - Total Coins/number of currency
        - Unique currency counts
- Mobile Friendly Display
    - Navbar and page layout changes to fit to any given screen size and scale

Unique Product Features

Our platform targets beginners within the cryptocurrency space, with our simplified display data via a card-based container that holds immediately relevant data. This is to avoid excess “noise” to ensure the user isn’t overwhelmed and/or confused by what they see.

The application allows for the creation of accounts and subsequent logins via Magic Link, a link that is sent to the users email that allows them to login to their account without having to create a password for their account.

Url: http://ec2-3-83-190-71.compute-1.amazonaws.com/about

2) QA testing- max 2.5 pages

For the functions, write a QA unit & integration test plan and results (check class slides). You have to write a report to contain the following information :

I. Unit Test

Select 4~5 major user stories to be tested with your focus. For example, you can select 1) list of refrigerator inventory, 2) recipe recommendation, 3) generation of nutrition consumption report and 4) create a shopping list. For units (modules/classes/methods) in source files which covers those major user stories, write your unit test cases in the source and perform the unit test.

- Define, develop and write unit test cases in your chosen unit testing framework.
- Build unit test cases for units in your choice. Run the test cases by the Unit Test tool such as Junit, Jest and unittest.
- Analyze statement test coverage (refer to slides) and describe how much statements are covered by your unit testing.

II. Integration Test

For every P1 feature, perform the below:

- Select template for integration testing QA (refer to slides):
  - The format should include test case id; test case description; dates tested, Test scenario with steps to follow, Prerequisites, Test data and Test results (PASS or FAIL for each tested browser).
- Write the integration testing cases per the above format.
- Perform testing and write down the testing results.
  - You can write down the test results in template, or in the system of your choice like Jira, or Github Issues.
  - If your choice is the system, please submit the URL of the system which has the testing results.
- Analyze of your test coverage for your P1 features and describe it.
  - Out of all your P1 features, how many are tested?

Test #1:
description:
Loads crypto names from portfolio into a set. expect the size of the set to be greater than zero.
dates:
April 30
data:

```
{
        "crypto": "test1",
```

```
        "amount": "9000"
    },
    {
        "crypto": "test2",
        "amount": "9000"
    },
    {
        "crypto": "test3",
        "amount": "9000"
    }
```

results:
Passed successfully

Test #2:
description:
Loads crypto names from set into a map as keys with the units the user owns as values. Expect the map to not be undefined.
dates:
04/31/22
data:

```
{
        "crypto": "test1",
        "amount": "9000"
    },
    {
        "crypto": "test2",
        "amount": "9000"
    },
    {
        "crypto": "test3",
        "amount": "9000"
    }
```

results: Passed successfully

Test #3:
description:
Fetch portfolio from database and return the portfolio as an array. Expect the array to not be undefined.
dates:
04/31/22

data:

```
const sample_portfolio = [

{
"amount": 1,
"crypto": "ETH",
"id": 221,
"user_id": "test_user123"

},
{
   "amount": 1,
   "crypto": "AVAX",
   "id": 221,
   "user_id": "test_user123"

},
{
       "amount": 1,
       "crypto": "BTC",
       "id": 221,
       "user_id": "test_user123"

}
```

results: Passed successfully, map is defined

Test #4:
description:
Calculate total portfolio value. If the user has at least 1 coin, expect the value to be at least $0.
dates:
04/31/22
data:
portfolio data:

```
{
"amount": 1,
"crypto": "ETH",
"id": 221,
"user_id": "test_user123"
```

```
},
{
   "amount": 1,
   "crypto": "AVAX",
   "id": 221,
   "user_id": "test_user123"

},
{
       "amount": 1,
       "crypto": "BTC",
       "id": 221,
       "user_id": "test_user123"

}
```


results :
Passed successfully, $40,000

Test #5:
description:
Calculate total portfolio value. If the user has at least 1 coin, expect the value to be defined.
dates:
05/01/22
data:
portfolio data:


```
{
"amount": 1,
"crypto": "ETH",
"id": 221,
"user_id": "test_user123"

},
{
   "amount": 1,
   "crypto": "AVAX",
   "id": 221,
   "user_id": "test_user123"

},
{
```

```
        "amount": 1,
        "crypto": "BTC",
        "id": 221,
        "user_id": "test_user123"

}
```

results :
Passed, total value is not undefined.

```
PASS __tests__/loadCryptoData.test.js
  ✓ loads tickers into set  (1 ms)
  ✓ populating map of crypto   (1 ms)
  ✓  fetching portfolio
  ✓  total value > 0   (1072 ms)
  ✓  total value not undefined   (821 ms)

Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        4.478 s, estimated 6 s
Ran all test suites related to changed files.
```

3) Code Review:

a) A coding style. In the report, please describe what coding style you chose, and how to enforce it (refer to slides).
    a. Our team installed ESlint on our local machines and used that to keep a level of consistency between our coding styles. We also are going through the codebase as a team to unify our coding style.

b) Chose the code (substantial portion of it) related to the P1 features you used for the above QA. You need to have a proper code review practices for these P1 features which shows the below :
    1. One team member submits code to github.
    2. Peer review should be performed by other group member(s) (1 reviewer is OK).
    3. Meaningful comments are included by the reviewer, instead of saying “looks good”.
    4. Based on the comments, update and submit the code again (optional)
    5. Final code is merged to the github.

For this we have been improving our comments on all pull requests. Our github should display all of these.

Please submit ~10 Github pull request urls to show the above.

Important: It is critical that code reviews are friendly and helpful, intended to help and education, and not to criticize. It is strongly suggested that you use peer review in the development of the whole system. Reviewers should check if the functionality of the code is correct and is well-optimized (better than other possible alternatives) (refer to lecture slides).

5) Self-check: Adherence to original Non-functional specs

Copy all original non-functional specs as in high level application document published at the very beginning of the class and then for each say DONE if it is done (which is expected and required); ON TRACK if it is in the process of being done and you are sure it will be completed on time; or ISSUE meaning you have some problems and then explain it.

ID	Title	Description	Priority	Progress
1	Account Creation	Users can create a new account using an email address. Our website won't keep track of names or passwords.	High	DONE
2	Login	If a user already has an account, they can log in by entering their email. Supabase will authenticate the user.	High	DONE
3	Dash-board	The dashboard is where users view their chosen crypto currency. This is personal to every user, and they get to choose which currencies to add and how many they want to keep track of. If a user is not logged in, they won't have a dashboard feature.	High	DONE
4	Search	Users can use the search function to look up specific crypto currencies. If the user is logged in, they have the option of adding cryptos to their dashboard.	High	DONE
5	Filter	Users can add a filter when they are using the "search" feature or when they are browsing. Some ways you can filter include: price, change %, or whether the price is	Medium	ISSUE - Not necessary for searching, results based on popularity.

		rising or falling.		
6	Trending	All users have a “trending” feature, whether or not they’re currently logged in. This section displays the most popular crypto currencies.	High	DONE
7	Browse All	Users can view all crypto currencies listed in alphabetical order. They can also filter the results using the “filter” feature.	High	DONE
8	Forum	Users can also browse a forum by clicking the “Forum” button at the top of the screen or in the drop down menu on the left-hand side. In the forum, users can create posts, respond to others, or just read through discussions on different topics. It will let users talk to each other about anything crypto related, whether it be tips for investing, past experiences, or predictions for the future of crypto currency.	Medium	ISSUE - Not necessary for our site. Tends to take away from the simplicity of the dashboard.
9	Accessibility	Users can choose between a light mode and a dark mode. This is helpful for people with poor vision or who want to view the web application in dim lighting.	Low	ON TRACK
10	Remove Account	Users can delete their account from our website.	Low	ON TRACK
11	Email Notifications	Users can sign up to get notifications about cryptos listed in their dashboard.	Low	ISSUE - Not neccessary

Submission of Milestone 4 document for review

The whole student team submits one milestone document for Milestones 4, as follows:
Team leads will send e-mail to the instructor and TA, with subject line as specified below.

e-mail subject line: Must be “CSC648-848 Milestone4 Section X Team N” in the subject line.

- **File name of the M4 document MUST be**: CSC648-848 Section X Milestone4 Team*N*.PDF

Instructor's feedback on Milestone 4 document

Upon submission of Milestone 4 for review you may get feedback from instructors and TA by e-mails. This feedback will be used to revise your development practices. If you have any issue with the feedback, contact the instructor.

Creating Milestone 4 document for * final delivery in M5 folder ****

Final Millstone 4 document is submitted after the original one is revised or updated. At the end of the semester, the revised M4 will be regarded as a part of M5.

Evaluation and grading

We will grade each milestone after its submission. The revised doc will also be re-graded as a part of M5 at the end of the class.