SENG 426

Software Quality Engineering

Summer 2021

University of Victoria

Delivery 3: Automated Testing & Continuous Integration

June 30th, 2021

David Bishop	V00884250
Amaan Makhani	V00883520
Ben Austin	V00892013
Nolan Kurylo	V00893175
Liam Franke	V00887604
Evan Roubekas	V00891470

Table of Contents

CI Benefits and Analysis

Implementing CI into our project has provided both benefits and hindrances as the project development progresses throughout the term. Doing CI has allowed our team to coordinate and view the projects' progression as features are implemented and committed while using the git flow model. During this deliverable, we have seen the benefit of implementing a growing test suite into our application, ensuring that features and fixes are not only verified at the time of development, but are continually monitored during each build, with appropriate notifications set up from Jenkins to inform us of build errors along the way.

While there are ultimately more benefits than hindrances from implementing continuous integration into our application, there are a handful of inconveniences it causes for our team. Firstly, our group has found it harder to have certain members work on features while others work on tests, as to follow the git flow model correctly, the tests need to be written in advance of a merge request to the master branch. Additionally, we have found that integrating more tests into our CI process has been tedious, although the value of having these tests outweighs this temporary hindrance.

When compared to a professional environment, our group is not as experienced when it comes to doing CI on such an application. In a professional environment, the CI process would likely be more complicated, but would likely have a higher quality setup and process to follow when compared to the efforts of our group.

Software Quality Characteristics

During this deliverable of the project, the quality characteristics of the application have evolved as our team fixed issues, added features, and improved our CI and automated testing framework.

In terms of functionality, the application has gained functionality through both newly implemented features such as the foreign exchange calculator and the cryptocurrency feature, and has additionally gained functionality due to bug fixes that allow pre-existing features to function properly as intended. Reliability has increased throughout this part of the assignment as a direct result of fixing pre-existing issues, and implementing proper automated test suites for both existing and newly implemented features.

We believe that usability of the application has also increased as a result of implementing fixes for features that may not have been intuitive due to lack of feedback to users, such as field validation and informative error messages. Unfortunately due to new features, especially ones that require API calls to external resources such as the new cryptocurrency feature, I believe the efficiency of the application has decreased,

although this decrease is confined to the use of the new features. Additionally, maintainability has slightly decreased due to added complexity after adding functionality and increasing the amount of automated testing suites.

Overall, we believe the quality of the application as a whole throughout this development & assignment phase has increased and we are pleased with the trajectory of the applications overall quality and functionality as we progress forward during this term.

Description of New Feature Implementation

For this phase of the project, two main new features, the foreign exchange calculator and the cryptocurrency feature were implemented into the Neptune Bank application.

The foreign exchange calculator feature was split into three tickets. Firstly, it was required that a table was implemented that displayed the latest exchange rates for at least 6 main currencies. Fetching from a third party API (Exchange Rates API), we were able to get the exchange rates for USD,CAD,JPY,EUR,GBP,AUD,CHF,CNY,SEK,NZD, using CAD as the base rate. Using the fetched data, we output the required data into a table.

Secondly, we were required to set up the conversion between CAD and any other currency. This was done through parsing the Amount of Money a User Entered, and multiplying by the exchange rate of another currency.

Lastly, we were required to set up the conversion between any two of the currencies we selected. Using CAD as the intermediary, we divided the entered amount by the source currency exchange rate, and multiplied the result by the target currency exchange rate.

The functionality of the feature will be outlined in our video, however included below are screenshots that show the Forex page before and after a currency conversion has taken place.

Foreign Exchange Calculator

Neptune Bank's Foreign Exchange Calculator feature allows you to convert currency between 10 currencies used around the world.

Currency	Exchange Rate
USD	0.813402
CAD	1
JPY	90.104657
EUR	0.681384
GBP	0.585814
AUD	1.0721
CHF	0.745752
CNY	5.25149
SEK	6.905289
NZD	1.15107

From: USD

To: CAD

Amount: 110.55

Convert

Figure X - Foreign Exchange Calculator - Pre Conversion

Foreign Exchange Calculator

Neptune Bank's Foreign Exchange Calculator feature allows you to convert currency between 10 currencies used around the world.

Currency	Exchange Rate
USD	0.813402
CAD	1
JPY	90.104657
EUR	0.681384
GBP	0.585814
AUD	1.0721
CHF	0.745752
CNY	5.25149
SEK	6.905289
NZD	1.15107

From: USD

To: CAD

Amount: 110.55

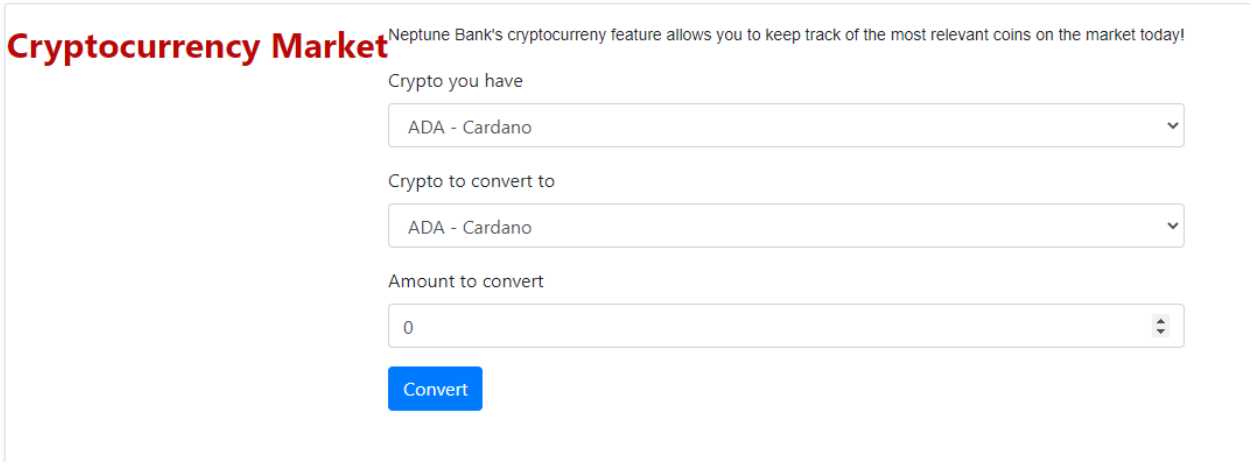
Convert

Converted: 135.91

Foreign Exchange Calculator -- Post Conversion

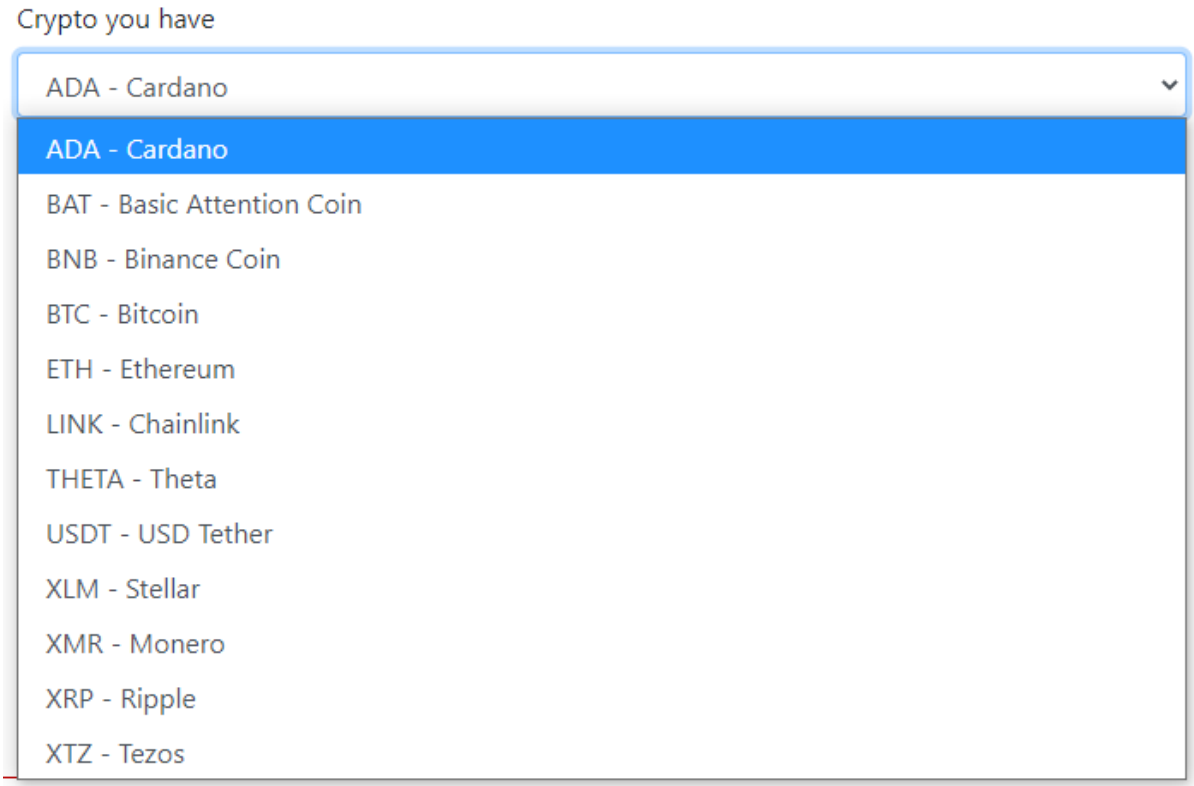
Secondly the cryptocurrency feature, which was implemented in a similar fashion to the currency converter. We first grab exchange data from a free API found online (https://coinlayer.com/) using reacts `useEffect()` hook that fetches the data. Upon

visiting the crypto converter screen the user is confronted with this simple UI (see figure below).



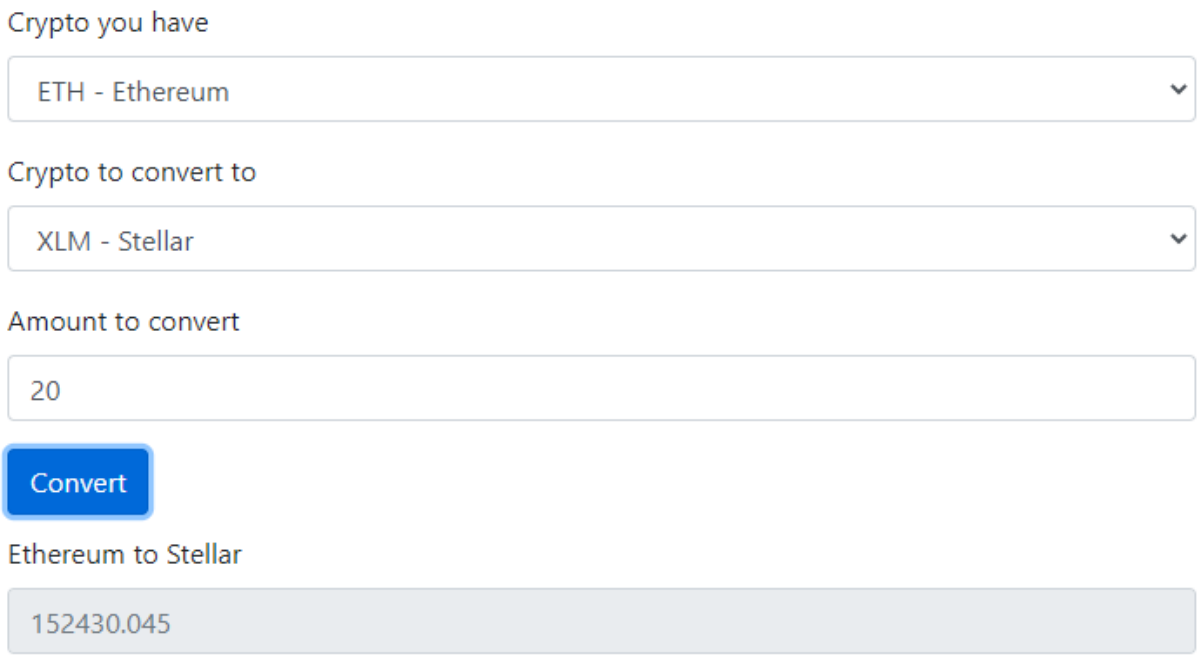
Base crypto converter page when a user visits it

The user can then change the selected cryptos into a multitude of options (see figure below).



The crypto currencies available to users for converting

When the user has filled out the form and pressed the “Convert” button a new box will show with the successful conversion between the two currencies (see figure below).



The resulting view when a user converts between crypto currencies

This conversion is done simply by taking the exchange rates (with the base currency as USD) of the two currencies and dividing the crypto you have by the crypto you want. Using state in react this converted number is then persisted as the components of the website re render and display the converted number.

Bugs Discovered

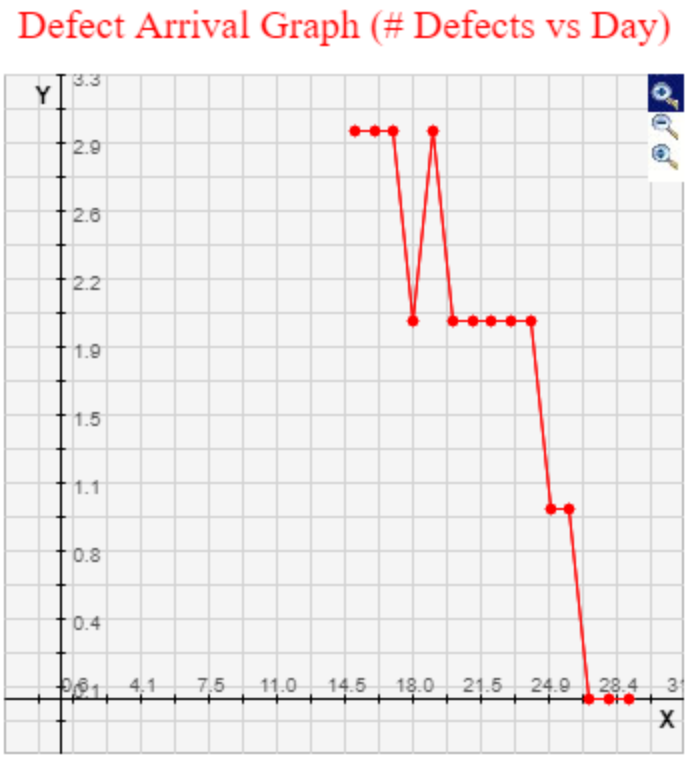
During testing of account management we found a bug that prevents a customer from being removed. This bug prevents some customers from being deleted but some are able to be deleted. Unfortunately, we did not have time to take on this bug this sprint as it was discovered too close to the deadline. We also took on a lot of other bug fixes so this bug was unable to be worked on.

Defect Arrival Graph

Before this task was performed, many defects were sought out and identified via various sections of the application. At first glance, many defects were found across the application and were added as bug fix tickets in the group's Gitlab repository. These defects outlined unfulfilled quality characteristics in terms of the application's user interface, user experience and application functionality. By performing the Defect Arrival Graph on found defects, around 3 defects were found a day at the beginning of the project deliverable period. The number of

defects began to taper off as the deliverable went on. It can be seen in the Defect Arrival graph that the number of defects identified each day increased. This was due to fixing the bugs and discovering new bugs along the way. By the end of the Defect Arrival Graph duration, no new bugs were identified which implies that the underachieving quality characteristics have increased in their value. Furthermore, the quality attributes improved from the bug fixes and new features that were implemented. The following table and subsequent graph outline the group's identified defects throughout the duration of the deliverable.

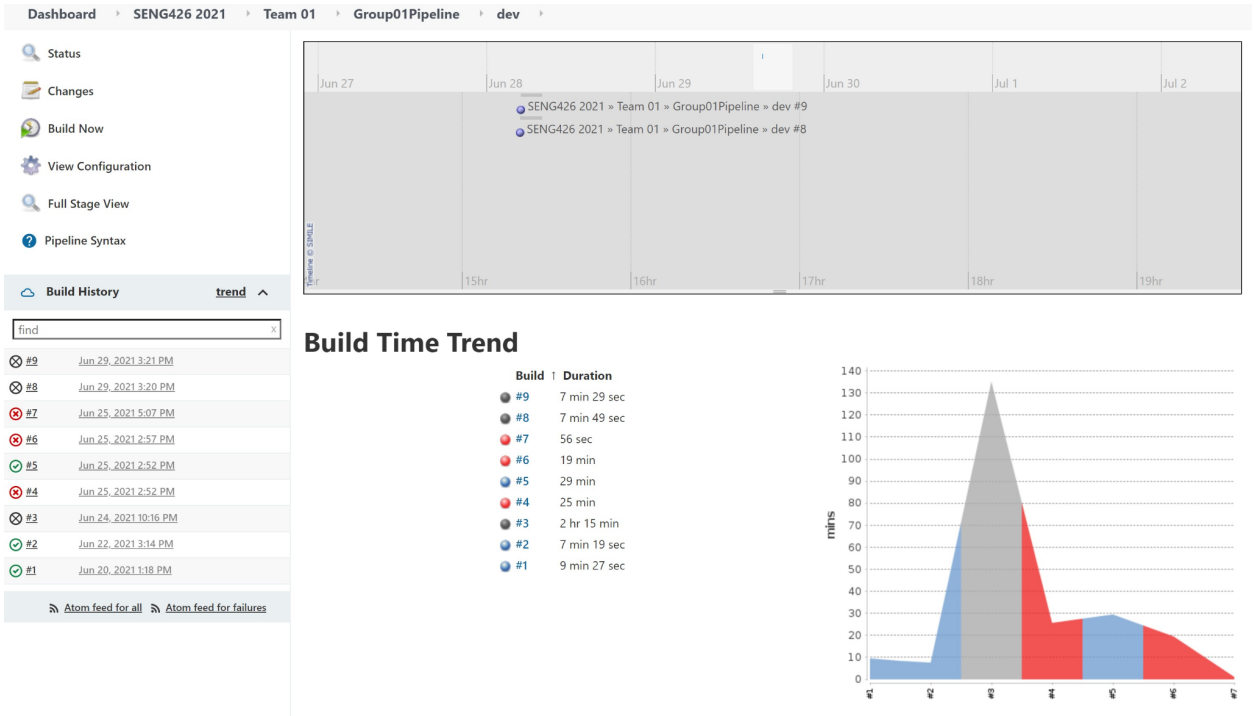
X - Day of June	Y - Number of Defects
15	3
16	3
17	3
18	2
19	3
20	2
21	2
22	2
23	2
24	2
25	1
26	1
27	0
28	0
29	0



Work Analysis

The total hours spent on the tickets was 20.25. We expected the tasks to take 89 hours. Our velocity calculations are shown in a section below. As mentioned below when the code is not known bug fixes are harder to estimate. To track this time and maintain the information we created an excel sheet. This sheet was then updated by our team mates as tasks were completed.

Jenkins History and Trends



Above the Jenkins builds history and the trends are shown. Jenkins had no more space so the latest builds had to be aborted. These reports are shown as best as possible but it is difficult due to the lack of resources we were provided.

Estimation Accuracy Calculation

The accuracy of our estimations can be calculated by the following formula:

$$
\begin{aligned}
Accuracy &= \frac{Total\ Actual\ Time\ Spent}{Total\ Estiamted\ Time} \times 100 \\
&= \frac{20.25\ Hours}{89\ Hours} \times 100 \\
&= 22.27\%
\end{aligned}
$$

From the above accuracy calculation, our time spent a significantly smaller time working on feature implementation and bug fixes than originally estimated. We can attribute this to the fact that we were unfamiliar with the codebase and made generous estimations when creating tickets for features and bugs. As time goes on and we become more familiar with the codebase and our own skills, we believe that the overall estimation accuracy will increase, as the estimated times will be closer to the actual times.

Ticket Summary

Ticket Title	Original Estimation (HRS)	Actual Time (HRS)
Transaction list date range filter functions incorrectly.	8	.5
Can not sort branches by Branch ID in "Our Branches" Tab	4	.5
Navigating back from Document Upload results in authorization error	4	.25
Can not sort Transactions by Tran ID	4	.5
Can not sort accounts by Branch	4	.5
Sort by payeeID	1	.5
Sorting by Account ID in "Accounts" tab results in error	1	.5
"End date" can be before "start date" when sorting by date in Transactions.	4	1
Email ID of a Payee does not need to be in Email Format	2	1
Staff/manager Account Type unable to access settings/password pages	4	.5
Username Error Message displays improperly on First Click on Hyperlinks on Sign in Modal	8	.5
No input validation on fields while requesting a new customer account	1	.5
View the Current Buy/Sell exchange rates between foreign currencies and CAD as a table	8	4
Calculate the currency conversion between a main currency and CAD	8	2
Calculate the currency conversion between any pair of main currencies using CAD as intermediary	8	2
Admin cannot post to news/updates page	2	.5
Forgot Password Email does not send to email address entered	2	2
Implement cryptocurrency page	8	1
Implement cryptocurrency data display	8	2
Totals	89	20.25