Stock Prediction App

Week 2 Progress

CS361 HW 7

Open	1,504.58	Div yield	-
High	1,510.94	Prev close	1,516.99
Low	1,478.49	52-wk high	1,530.74
Mkt cap	1.02T	52-wk low	1,027.03
P/E ratio	30.17		

Instructions and Program available at:
https://github.com/cs361-stock-prediction/stock-predictor

Team 16:
Felix Brucker
Robert Detjens
Remi Kendig
Dominykas Zobakas
Lyell Read

Table of Contents

Table of Contents **2**

Week 2 User Stories **4**

Leftover from Last Week: 4

[✓]Account Creation 4

[✓]Account Login 4

[⚫]Edit Account 4

[✓]Search by Stock Name 5

Completion by 3/14/2020: 5

[X]Save Favorites 5

[X]Search History in Account Info 5

[X]Similar Stock Recommendations 5

[X]Search by Category 6

[⚫]Change Site Theme 6

[X]Quick Account Creation 6

Spike Reflections **8**

Burndown Diagram **9**

Refactoring **9**

Client 10

Server 10

CI Workflow 10

Schedule for Coming Week **11**

[⚫]Edit Account 11

[X]Save Favorites 11

[X]Search History in Account Info 11

[X]Similar Stock Recommendations 11

[X]Search by Category 11

[⚫]Change Site Theme 12

[X]Quick Account Creation 12

Customer Questions **13**

Integration Tests **14**

Linters 14

Python 14

JavaScript 14

CSS 14

HTML 14

Contributions **15**

Customer 15

HW7 15

Team Members 15

HW7 Document 15

HW7 Coding 15

Appendix A: Historical Contribution Log 16

Customer 16

HW1 16

HW2 16

HW3 16

HW4 16

HW5 16

HW6 16

Team Members 17

HW1 17

HW2 17

HW3 17

HW4 17

HW5 17

HW6 Document 18

HW6 Coding 18

Week 2 User Stories

Note: **[✓]**: Complete; **[⚫]**: Partially Complete; **[X]**: Not Complete
ε is 2 hours

Note: as a result of (a) sickness, (b) changes in course schedules as a result of Coronavirus (COVID-19) (c) Windows compatibility and other late-arisen difficulties that were more challenging than expected, we completed fewer tasks than we had expected to or (d) some team members either being highly limited in availability to work (Remi Kendig) or not contactable at all (Dominykas Zobakas).

Leftover from Last Week:

[✓]Account Creation

Due Date: 3/10/2020
Prerequisites: Basic Site Framework
Workers: (Robert, Felix)
Unit Tests: Code Linting (JS, Py, CSS, HTML)
Problems: None
Left to be Completed: None
Tasks:

- Client:
  - Password Hashing
- Server:
  - Account credential store
  - Creating user entry in account database

Projected time: 3ε

[✓]Account Login

Due Date: 3/10/2020
Prerequisites: Basic Site Framework, Account Creation
Workers: (Robert, Lyell)
Unit Tests: Code Linting (JS, Py, CSS, HTML)
Problems: None
Left to be Completed: None
Tasks: None

- Authentication Flow
  - Client: Login form, hashing function
  - Server: credential store, authentication matcher
- Session Management
  - Client: store session cookies to keep login across pages
  - Server: associate session activity to account

Projected time: 3ε

[⚫]Edit Account

Due Date: 3/10/2020
Prerequisites: Basic Site Framework, Account Creation, Account Login
Workers: (Lyell, Robert)
Unit Tests: Code Linting (JS, Py, CSS, HTML)

Problems: None
Left to be Completed: Full account db integration
Tasks:
- Creating account settings page (Done)
- Updating user info in account db

Projected time: 2ε

[✓]Search by Stock Name

Due Date: 3/14/2020
Prerequisites: Basic Site Framework
Workers: (Robert, Dom)
Unit Tests: Code Linting (JS, Py, CSS, HTML)
Problems: None
Left to be Completed: None
Tasks:
- Search functionality:
  - Client: search entry field, results page and query to server.
  - Server: Performing lookup in name database, generating results page

Projected time: 3ε

Completion by 3/14/2020:

[X]Save Favorites

Due Date: 3/10/2020
Prerequisites: Basic Site Framework, Search by Ticker Symbol, Account Login
Workers: ()
Unit Tests: Code Linting (JS, Py, CSS, HTML)
Problems: None
Left to be Completed: Button integration and DB link
Tasks:
- Client: Create button on stock visible when logged in
- Client: Send event to server when button is clicked
- Server: Adds/removes stock ticker to account db entry when event received
- Server: Checks if requested stock is saved to an account

Projected time: 3ε

[X]Search History in Account Info

Due Date: 3/12/2020
Prerequisites: Basic Site Framework, Search by Ticker Symbol, Account Login
Workers: ()
Unit Tests: Code Linting (JS, Py, CSS, HTML)
Problems: None
Left to be Completed: Event integration and DB link
Tasks:
- Client: Sends event to server when stock is viewed
- Server: Adds/removes stock ticker to account db entry

Projected time: 1ε

[X]Similar Stock Recommendations

Due Date: 3/14/2020
Prerequisites: Basic Site Framework, Search by Ticker Symbol, Account Login
Workers: ()
Unit Tests: Code Linting (JS, Py, CSS, HTML)
Problems: None
Left to be Completed: Elements' integration and DB link
Tasks:

- Client:
  - Button on stock visible when logged in
  - Click to toggle save or not save stock
  - Changes to show if saved or not
- Server:
  - Adds/removes stock ticker to account db entry
  - Checks if requested stock is saved to an account

Projected time: 3ε

[X]Search by Category

Due Date: 3/14/2020
Prerequisites: Basic Site Framework
Workers: ()
Unit Tests: Code Linting (JS, Py, CSS, HTML)
Problems: None
Left to be Completed: Data entry integration and DB link
Tasks:

- Search functionality:
  - Client: search entry field, results page and query to server.
  - Server: Performing lookup in category database, generating results page

Projected time: 3ε

[⚫]Change Site Theme

Due Date: 3/14/2020
Prerequisites: Basic Site Framework, Account Login
Workers: (Remi)
Unit Tests: Code Linting (JS, Py, CSS, HTML)
Problems: None
Left to be Completed: Color integration and DB link
Tasks:

- Client:
  - Field on account page to change theme
- Server:
  - Store accounts' themes
  - Data containing site themes

Projected time: 3ε

[X]Quick Account Creation

Due Date: 3/14/2020
Prerequisites: Basic Site Framework, Account Creation

Workers: ()
Unit Tests: Code Linting (JS, Py, CSS, HTML)
Problems: None
Left to be Completed: Google integration and DB link
Tasks:

- Client:
  - Button to create account via social media or Google account on "Create Account" page
- Server:
  - Google login service connection
  - Creating user entry in account database

Projected time: 3ε

Spike Reflections

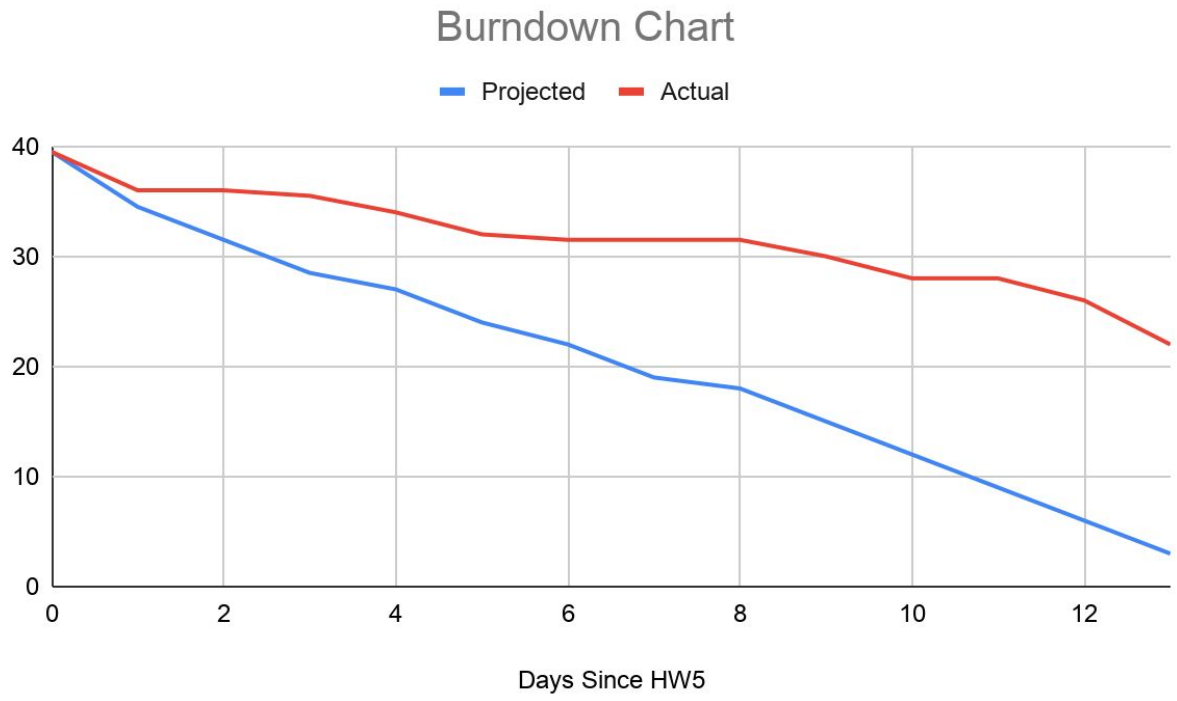
No diagrams were assigned for the second week, in the fifth homework assignment, so there are no diagrams to look back on this week. However there were some spikes that were informative.

On the server side, graphs were attempted as a data manipulation tool to see if they would be effective in handling stock data. Ultimately they were discarded but the spike let us know that trying to generate the graphs server-side was not an efficient avenue to pursue. It also helped inform future handling of the prediction algorithm.

An attempt at making the now more complicated server startup execute on a Windows machine was made for ease of accessibility (most of the development had previously been done in a Linux environment). Separate attempts were made in Git Bash and Windows Subsystem for Linux. Git Bash lacked the scope, and WSL hit difficulties due to installed resource finding, possibly due to drive configuration. The spike let us know to push forward with a linux configuration, but also made us aware of several incompatibilities in the setup script that had not been caught by the native shells.

Burndown Diagram

Note: This burndown diagram does not line up perfectly, as there were a couple tasks that were not complete at the end of Week 2. These tasks total the difference between Actual and Forecasted. This was a result of many unforeseen circumstances including sickness, irreconcilable compatibility difficulties with the new backend design, and large final projects in other classes.



Refactoring

Client

No refactoring was done to the client.

Server

This week, our server went through a major refactor. We previously had the server in just one file, which worked fine for the start of the project as we did not have many routes and supporting functions. However, with the addition of the account and login forms and user database functionality the file was quite large. We split up the server file into multiple separate ones, one for each function. An init file was created which includes all the other files and the external modules we needed. Routes, form setup, and the user model/database were put in their own file. Finally, a runner file was created at the root directory of the repository that includes our server as a Python module and starts the webserver.

Our requirements script also went through another refactor, this time separating out the list of dependencies into a separate requirements.txt file that Pip looks at.

CI Workflow

No refactoring was done to our CI workflow.

Schedule for Coming Week

[⚫]Edit Account

Due Date: 3/16/2020
Prerequisites: Basic Site Framework, Account Creation, Account Login
Tasks:

- Creating account settings page (Done)
- Updating user info in account db

Assignees: Lyell Read, Felix Bruker
Projected time: 2ε

[X]Save Favorites

Due Date: 3/17/2020
Prerequisites: Basic Site Framework, Search by Ticker Symbol, Account Login
Tasks:

- Client: Create button on stock visible when logged in
- Client: Send event to server when button is clicked
- Server: Adds/removes stock ticker to account db entry when event received
- Server: Checks if requested stock is saved to an account

Assignees: Remi Kendig, Lyell Read, Dominykas Zobakas
Projected time: 3ε

[X]Search History in Account Info

Due Date: 3/18/2020
Prerequisites: Basic Site Framework, Search by Ticker Symbol, Account Login
Tasks:

- Client: Sends event to server when stock is viewed
- Server: Adds/removes stock ticker to account db entry

Assignees: Felix Brucker, Remi Kendig
Projected time: 1ε

[X]Similar Stock Recommendations

Due Date: 3/17/2020
Prerequisites: Basic Site Framework, Search by Ticker Symbol, Account Login
Tasks:

- Client:
  - Button on stock visible when logged in
  - Click to toggle save or not save stock
  - Changes to show if saved or not
- Server:
  - Adds/removes stock ticker to account db entry
  - Checks if requested stock is saved to an account

Assignees: Robert Detjens, Lyell Read, Remi Kendig
Projected time: 3ε

[X]Search by Category

Due Date: 3/18/2020
Prerequisites: Basic Site Framework

Tasks:

- Search functionality:
  - Client: search entry field, results page and query to server.
  - Server: Performing lookup in category database, generating results page

Assignees: Dominykas Zobakas, Lyell Read, Robert Detjens
Projected time: 3ε

[⚫]Change Site Theme

Due Date: 3/19/2020
Prerequisites: Basic Site Framework, Account Login
Tasks:

- Client:
  - Field on account page to change theme
- Server:
  - Store accounts' themes
  - Data containing site themes

Remi Kendig, Dominykas Zobakas
Projected time: 3ε

[X]Quick Account Creation

Due Date: 3/17/2020
Prerequisites: Basic Site Framework, Account Creation
Tasks:

- Client:
  - Button to create account via social media or Google account on "Create Account" page
- Server:
  - Google login service connection
  - Creating user entry in account database

Lyell Read, Felix Brucker
Projected time: 3ε

Customer Questions

Our customer continued to be very responsive as we kept him updated on advancements toward the completion of major milestones in the project. We had clarified a lot of things back in week 1, so there were less questions this time. However the customer clearly expressed approval of what he liked and was both flexible and straightforward in any clarifications.

Integration Tests

As far as CI tests, we only used integration workflows consisting of Linters that check all the code files we push to make sure they follow certain style guidelines, i.e. enforcing indentation, CSS rule order, etc. We have set these linters to automatically fix simple errors, such as indentation and whitespace. These workflows are run every time a pull request (Github's term for the merging of two branches) is made. All checks must pass for the branch to be allowed to merge.

Linters

Python

Our server is written in python. This server is checked using Integration workflows that run both Flake8 (https://flake8.pycqa.org/en/latest/) and Black (https://github.com/psf/black). These two linters check the code against a set of style rules every time we make a pull (merge two branches) request. We later abandoned black in favor of Flake8 as they provided redundant outputs, and for simplicity.

JavaScript

Robert has previously used ESLint (https://eslint.org/) in personal projects to lint his JS code, so he reused his existing config to lint our Javascript code. We currently do not have any JS code on our website, but this will enforce and fix errors in our programming.

CSS

We set up Stylelint (https://github.com/stylelint/stylelint) to lint our CSS files. We went with the standard configuration, but set the indentation level from 2 spaces to use tabs instead.

HTML

We attempted to get a HTML linter to work, but we could not find one that integrated into Github Actions. As such, we decided not to use an HTML linter.

Contributions

Customer

HW7

- The customer was active and easy to communicate with on multiple occasions.

Team Members

HW7 Document

- Felix Brucker: Spike Reflections, Customer Questions, Burndown
- Robert Detjens: Refactoring, Editing
- Remi Kendig: --
- Dominykas Zobakas: --
- Lyell Read: User Stories, Integration Tests, Completion next week

HW7 Coding

- Felix Brucker: Spikes
- Robert Detjens: User Avatars, Prediction Graph
- Remi Kendig: Initial Prediction Work
- Dominykas Zobakas: --
- Lyell Read: Startup Script and Dependency Install Script Updates, Prediction Algorithm Design, Issue management, Server Start Dependency CI Check.

Appendix A: Historical Contribution Log

Customer

HW1

- We met with our customer Ghaith Shan after class on 1/23/2020.
- We have maintained communication with our customer on Discord since 1/16/2020.

HW2

- We met with our customer Ghaith Shan after class on 1/30/2020 to present the paper mockups.
- We discussed requirement revisions and went over the mockups with our customer over Discord on 1/31/2020.

HW3

- We did not meet with our customer as we had already verified that we share a common understanding of the goals and requirements for this app so far.

HW4

- We did not meet with our customer as we had already verified that we share a common understanding of the goals and requirements for this app so far.

HW5

- We met with our customer on Discord on 2/17/2020 to start the discussion of User Stories.
- We met with our customer in person on 2/18/2020 and discussed the User Stories further.
- We contacted our customer on Discord on 2/29/2020 to verify that we were interpreting the requirements of each user story correctly.
- In our meeting with our customer, he was very reasonable about what was needed.

HW6

- Talked with our customer on Discord on 3/6/2020 to discuss the potential failure to complete one user story in the coming week.

Team Members

HW1

- Felix Brucker: Document Creation, Data Flow Diagram, Proofreading
- Robert Detjens: Functional Specifications, Non-functional Specifications, ERD Revision, Use Case 2, Use Case 2 MSD
- Remi Kendig: Use Case 1, Data Flow Diagram
- Dominykas Zobakas: ERD Revision, Use Case 1 MSD
- Lyell Read: ERD First Draft, Document Formatting, Customer Contributions, Functional Definitions, Non-Functional Definitions, Use Case 3, Use Case 3 MSD

HW2

- Felix Brucker: Mockup proofing, Use Case 1 Revision
- Robert Detjens: Paper mockups, Mockup meeting w/ customer, Definition revision
- Remi Kendig: Changes made since HW1
- Dominykas Zobakas: MSD Revisions, ERD revision
- Lyell Read: Mockup proofing, Digital Mockup Adaption, Appendix, Corrections to UC1 & UC3 MSDs.

HW3

- Felix Brucker: Implications
- Robert Detjens: Architecture 2 Data Flow Diagram, Fault Trees, Document Formatting
- Remi Kendig: Quality Attribute Assessment, Architecture Validation
- Dominykas Zobakas: Decomposition
- Lyell Read: Architecture 1 Data Flow Diagram, Team Member Contributions, Interfaces Revision

HW4

- Felix Brucker: Design Patterns, Interfaces
- Robert Detjens: UML Class Diagram, Interface Revision
- Remi Kendig: Incremental Development
- Dominykas Zobakas: Class Sequence Diagram
- Lyell Read: Document Formatting, Exception Handling, UML Class Diagram

HW5

- Felix Brucker: User Stories Implementation Plan
- Robert Detjens: User Story Descriptions, User Story Sequence
- Remi Kendig: UML Diagrams
- Dominykas Zobakas: User Story Diagrams
- Lyell Read: Customer Contact on Discord, Plan Organization by Group, User Story Sequence, User Stories Implementation Plan, Document Formatting.

HW6 Document

- Felix Brucker: Refactoring, Diagram Reflection
- Robert Detjens: Refactoring, Integration Tests
- Remi Kendig: Customer Questions
- Dominykas Zobakas: --
- Lyell Read: Customer Contact, Document Setup, Week 2 Schedule, Burndown Diagram, Week 1 Tasks, Integration Tests.

HW6 Coding

- Felix Brucker: Login and Account Creation pages, Settings Page
- Robert Detjens: Basic Site, Search by Ticker, Account login forms, CI Setup, API Documentation
- Remi Kendig: Basic Site
- Dominykas Zobakas: Search by Ticker
- Lyell Read: Basic Site, Site Framework Server, Repository Setup, CI Linters for Python.