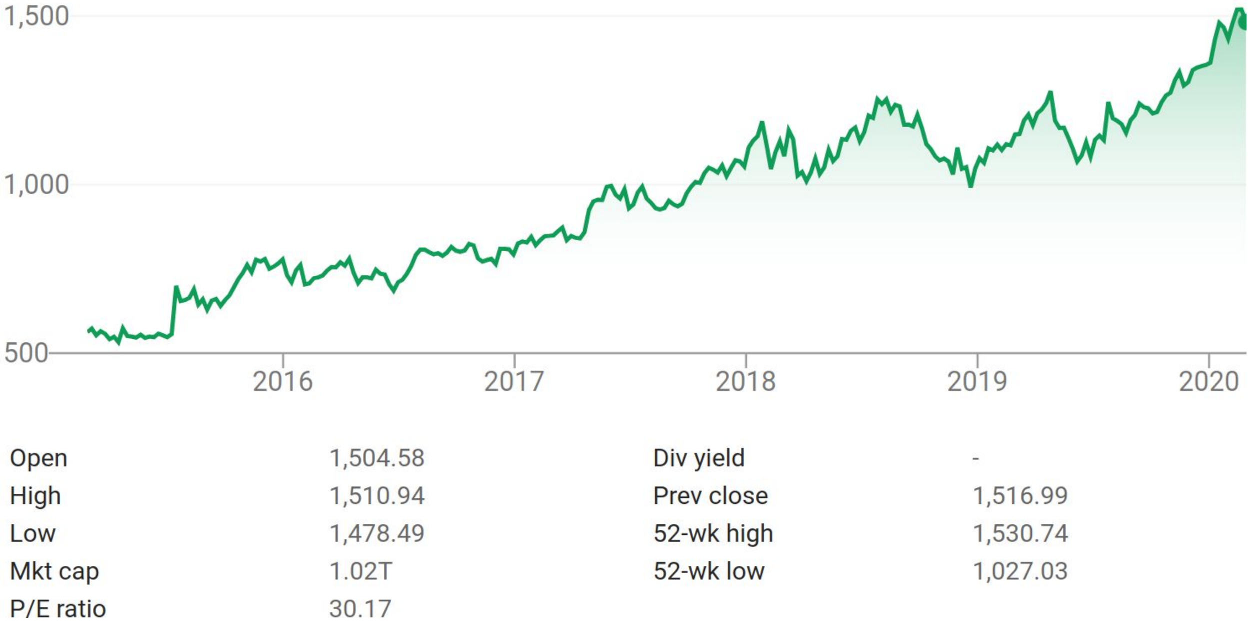
Stock Prediction App

User Stories & Set Up

CS361 HW 5



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		

Team 16:

Felix Brucker
Robert Detjens
Remi Kendig
Dominykas Zobakas
Lyell Read

Table of Contents

User Stories

Account Login

As a busy worker, I want to quickly be able to save my favorite stocks to an account for quick access.

The customer wants an account that they can save their stock to. This requires:

- A login/authentication handler
- Remembering the logged-in account across pages (e.g. cookies)
- A secure credential store on the server
- Marking stocks as associated with an account
- Retrieving that association on each stock page to mark if it is saved

Account Creation

As a user, I want to be able to create an account to save stocks to an account.

The customer wants to have an account service on the site. This requires:

- A account creation page
- A secure account credential database
- Creating a new entry in the database for new accounts

Edit Account

As a person who likes to change things, I want the option to be able to edit my profile and change user icons later on.

The customer wants to be able to edit their account information. This requires:

- An account/profile settings page
- Storing a user’s avatar
- A means to set a new avatar, e.g. upload a new picture

Search History in Account Info

As someone who researches many types of stocks, I want to be able to view what I have searched in the past for future reference.

The customer wants to see the history of what predictions they have viewed. This requires:

- A account history page, or recent searches in a search bar dropdown
- Saving a stock to an account’s history when viewed
- Retrieving the view history to be displayed when searching

Search by Ticker Symbol

As a user interested in seeing stock predictions, I want to be able to quickly search for a stock given its ticker symbol to save time.

The customer wants to be able to select a stock for prediction by their ticker symbol. This requires:

- A search field for ticker entry
- Autocompletion / suggestions as ticker is entered
- Checking if the ticker is valid when entered

Save Favorites

As a possible repeat user, I want to be able to save favorites to my account for future quick access.

The customer wants to be able to favorite a stock and see their favorited stocks on their account page. This requires:

- An account page to view the saved stocks
- Keeping a list of stocks associated with an account
- Retrieving that association on each stock page to mark if it is saved

Quick Account creation

As a user, I want to be able to create an account quickly by using social media or google account so that I may not waste time creating an account.

The customer wants to have the option to log in to the site with an existing third party account. This requires:

- “Sign up with Google” integration with the login flow
- Using OAuth2 as the authorization protocol
- Creating an account with our site aliased to the Google login
- Allowing users to either login with us or use the linked Google account to sign in

Personalize Account

As a creative person, I want to be able to personalize my account by creating a custom profile with user icons to differentiate myself from others.

The customer wants to be able to set a custom profile picture on account creation. This requires:

- Being able to upload a custom image during account creation
    - OR, pulling the Google account’s avatar if signing in with Google
- Storing the account’s avatar
- Displaying the account’s avatar on the account page and the account modal

Search History in Browser Cookies

As someone without a lot of free time in the day, I want to be able to view search history without having to login to my account.

The customer wants to be able to see local history, even when not logged in. This requires:

- Detecting if an account is logged in
- Storing view history in cookies instead of to the account if not logged in
- Reading history data from cookies to be displayed when searching

Similar Stock Recommendations

As someone interested in stocks, I want to be recommended stocks based on what kind of stocks I have saved in my account so that I may view additional stocks.

The customer wants to see related stocks based on their view history This requires:

- Grouping stocks into categories, whether manual or from the API
- Determining the categories each user/account views
- Getting other stocks from that category
- Displaying those recommendations to the user

Search by Stock Name

> *As a user looking for stocks, I want to be able to search for a stock by the company's name so that I can find a stock for a company for which I do not know the ticker.*

The customer wants to be able to search by the stock’s name in addition to ticker. This requires:

- Translating the entered stock name into the ticker
- Performing the search actions, etc on the stock as normal

Search by Category

> *As a user interested in a category of stocks, I want to be able to sort stocks by the type of stock so that I can understand what stocks are considered to be part of that category.*

The customer wants to be able to see the category a stock falls under as well as other stocks in that category. This requires:

- Grouping stocks into categories, whether manual or from the API
- Listing all stocks in a requested category

Change Site Theme

> *As a frequent and customizability-liking user, I want to be able to have a custom background to change how the site looks as a change of scenery.*

The customer wants to be able to change the look of the site. This requires:

- Using variables in the site design for color
- A dropdown in account settings to set the preferred accent color for the site
- Storing the preferred color for each account
- Retrieving the stored color when a user is logged in
- Setting the site’s color to that preferred color

User Story Details and Sequence

The effort associated with completing any one of these tasks will be represented in ε, a unit that is equivalent to 1 work hour for two members of our group (undergraduate students) completing this task as pair programmers. These tasks are ordered in the order they need to be completed, if applicable.

Completion by 3/7/2020

Basic Site Framework (not a user story)

Due Date: 3/4/2020

Prerequisites: none

Tasks:

- Client:
  - Basic site layout & design (including HTML, CSS and JS files that make this up).
  - Development of statically served / generated pages
- Server:
  - Serves static pages
- CI tests that check code consistency of function.

Estimated time: 5ε

Search by Ticker Symbol

Due Date: 3/4/2020

Prerequisites: Basic Site Framework

Tasks:

- Set-up of database
- Script to fill database
- Search functionality
  - Client: search entry field, results page
  - Server: Performing lookup in database, generating results page
- Development of Prediction Algorithm

Projected time: 3ε

Account Creation

Due Date: 3/4/2020

Prerequisites: Basic Site Framework

Tasks:

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

Projected time: 3ε

Personalize Account

Due Date: 3/7/2020

Prerequisites: Basic Site Framework, Account Creation

Tasks:

- Client: custom picture field during account creation
- Server: storing picture with account details

Projected time: 1.5ε

Account Login

Due Date: 3/5/2020

Prerequisites: Basic Site Framework, Account Creation

Tasks:

- 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: 3/6/2020

Prerequisites: Basic Site Framework, Account Creation, Account Login

Tasks:

- Creating account settings page
- Updating user info in account db

Projected time: 2ε

Completion by 3/14/2020

Save Favorites

Due Date: 3/10/2020

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

Projected time: 3ε

Search History in Account Info

Due Date: 3/12/2020

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

Projected time: 1ε

Search History in Browser Cookies

Due Date: 3/14/2020

Prerequisites: Basic Site Framework, Search by Ticker Symbol

Tasks:

- Client:
  - Button on stock visible when logged in
  - Click to toggle saved
  - 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

Similar Stock Recommendations

Due Date: 3/14/2020

Prerequisites: Basic Site Framework, Search by Ticker Symbol, Account Login

Tasks:

- Client:
  - Button on stock visible when logged in
  - Click to toggle saved
  - 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

Search by Stock Name

Due Date: 3/14/2020

Prerequisites: Basic Site Framework

Tasks:

- Search functionality:
  - Client: search entry field, results page
  - Server: Performing lookup in database, generating results page

Projected time: 3ε

Search by Category

Due Date: 3/14/2020

Prerequisites: Basic Site Framework

Tasks:

- Search functionality:
  - Client: search entry field, results page
  - Server: Performing lookup in database, generating results page

Projected time: 3ε

Change Site Theme

Due Date: 3/14/2020

Prerequisites: Basic Site Framework, Account Login

Tasks:

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

Projected time: 3ε

Quick Account Creation

Due Date: 3/14/2020

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

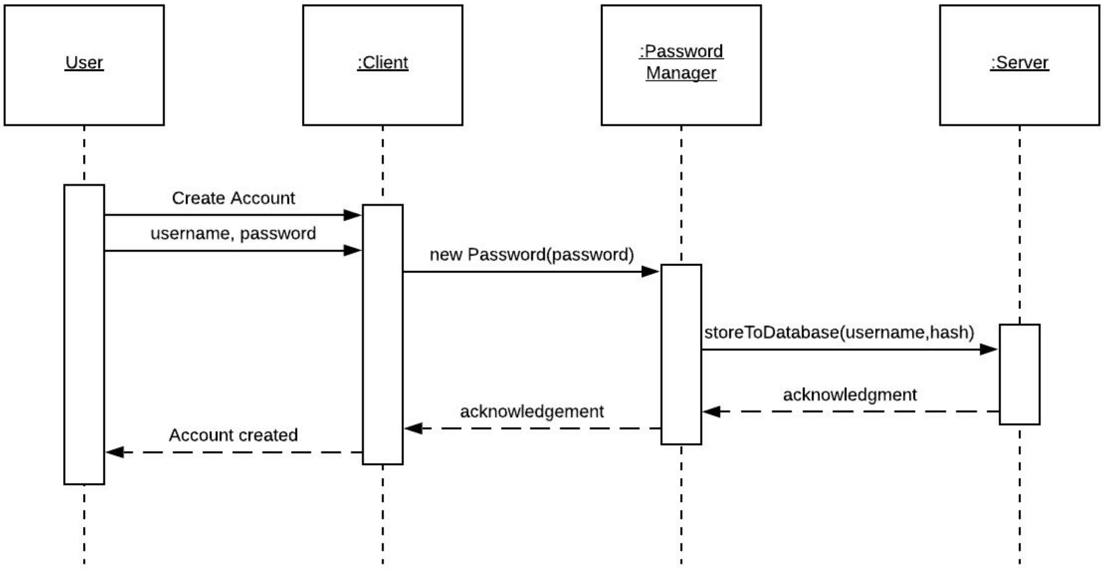
Projected time: 3ε

Will Not Be Completed

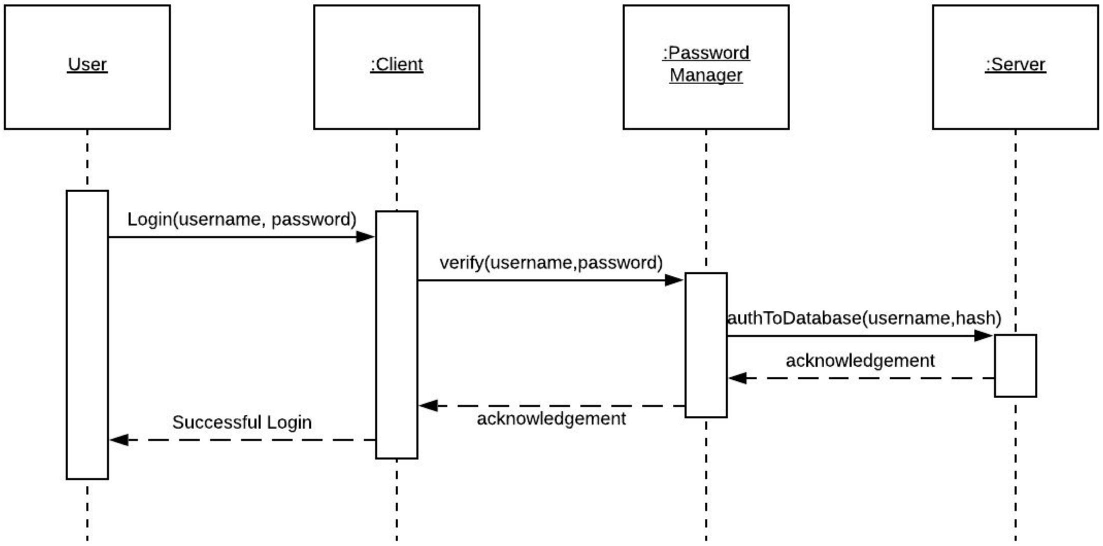
None of the tasks proposed by our customer will be impossible to complete. Therefore, there are no tasks that we foresee ourselves not being able to complete in time.

User Story Diagrams

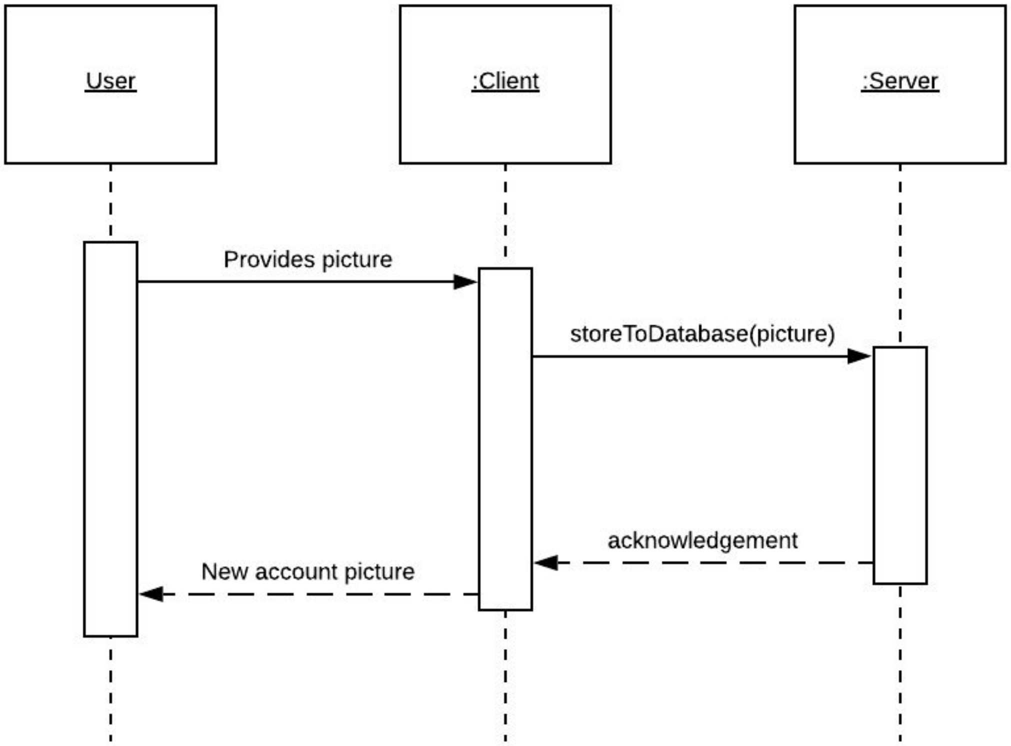
Account creation



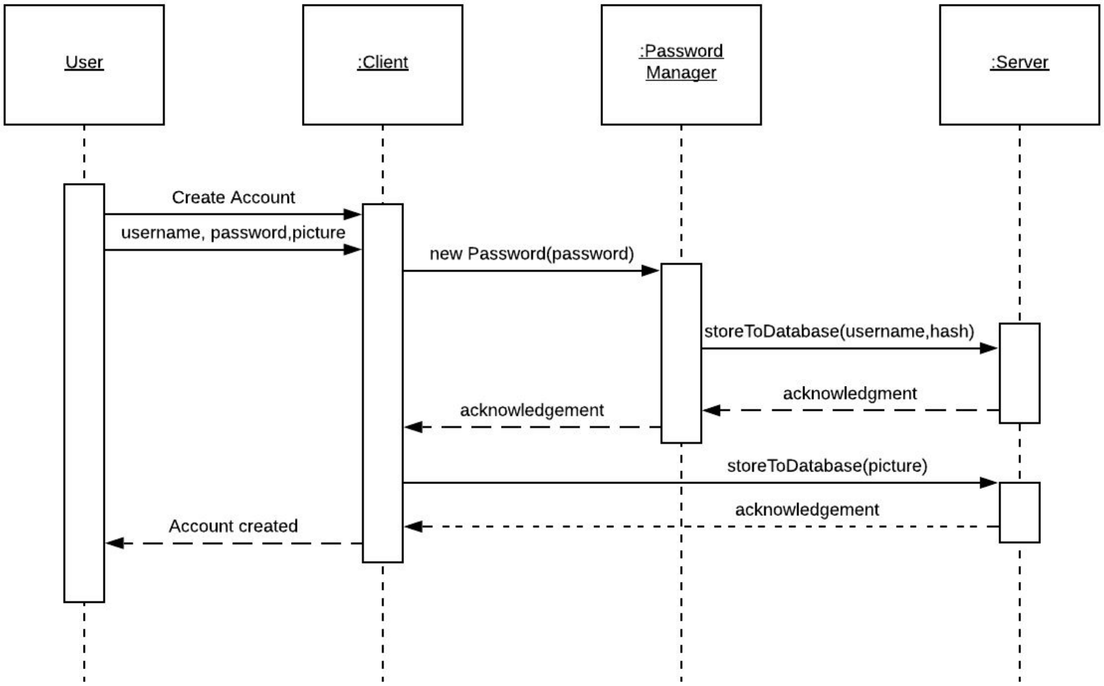
Account Login



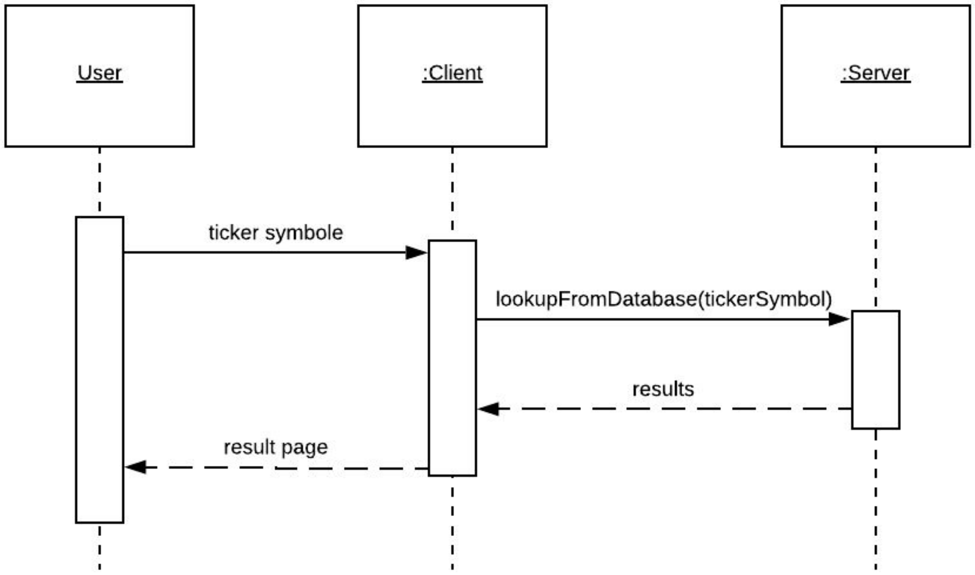
Edit Account



Personalize Account



Search by Ticker Symbol



User Story Implementation Plan

The effort associated with completing any one of these tasks will be represented in ε, a unit that is equivalent to 1 work hour for two members of our group (undergraduate students) completing this task as pair programmers.

Completion by 3/7/2020

The completion of tasks for this week will follow this order. We will use issues in GitHub (or projects, we are unsure as of yet) to manage the timely completion and accountability of these user stories and milestones.

Basic Site Framework (not a user story)

Due Date: 3/3/2020
Prerequisites: none
Tasks:

- Client:
  - Basic site layout & design (including HTML, CSS and JS files that make this up). 2ε
  - Development of statically served / generated pages 1ε
- Server:
  - Serves static pages 1ε
- CI tests that check code consistency of function. 1ε

Estimated time: 5ε
Assignee(s):

- Felix Brucker
- Robert Detjens
- Remi Kendig
- Dominykas Zobakas
- Lyell Read

Explanation: As this is the largest task and supports the functioning of all of our user stories, we will all complete this, pair programming as efficient and helpful. For example, Robert Detjens will write most of the JavaScript for the client side, pair programming with Felix Brucker. This task has no prerequisites, so we will start with it. One or two group members will also be in charge of setting up a CI environment that checks code function.

Search by Ticker Symbol

Due Date: 3/4/2020
Prerequisites: Basic Site Framework
Tasks:

- Set-up of database .5ε
- Script to fill database .5ε
- Search functionality 1ε
  - Client: search entry field, results page
  - Server: Performing lookup in database, generating results page
- Development of Prediction Algorithm 1ε

Projected time: 3ε
Assignee(s):

- Lyell Read
- Dominykas Zobakas
- Robert Detjens
- Remi Kendig
- Felix Brucker

Explanation: Lyell Read and Dominykas Zobakas will complete the server side modifications required to enable this functioning using pair programming, while Robert Detjens and Remi Kendig will complete the client-side modifications required to facilitate searching by ticker symbol, also using pair programming principles. Felix Brucker will be involved in ensuring that our CI tests are up to date and are checking both past code and new code added. This step is to be completed only once the Basic Site Framework is done, but relies on nothing else.

Account Creation

Due Date: 3/4/2020

Prerequisites: Basic Site Framework

Tasks:

- Client:
  - Create Account page .5ε
  - Password Hashing 1ε
- Server:
  - Account credential store .5ε
  - Creating user entry in account database 1ε

Projected time: 3ε

Assignee(s):

- Felix Brucker
- Robert Detjens
- Remi Kendig
- Lyell Read
- Dominykas Zobakas

Explanation: Felix Brucker and Lyell Read will handle modifications to the server-side of our implementation to facilitate account creation. Robert Detjens and Remi Kendig will be implementing the client-side changes that are required to enable users logging in. These pairs will both approach this using pair programming principles. Dominykas Zobakas will ensure that our CI tests are functioning for this step, as well as develop specific new steps to handle the new code written, to ensure its consistency of function over future steps. This step is to be completed only once the Basic Site Framework is done, but relies on nothing else.

Account Login

Due Date: 3/5/2020

Prerequisites: Basic Site Framework, Account Creation

Tasks:

- Authentication Flow
  - Client: Login form, hashing function .5ε
  - Server: credential store, authentication matcher .5ε
- Session Management
  - Client: store session cookies to keep login across pages 1ε

    - Server: associate session activity to account 1ε

Projected time: 3ε

Assignee(s):

- Robert Detjens
- Remi Kendig
- Lyell Read

Explanation: Lyell Read and Robert Detjens will pair program both the server side and the client side of this operation. Remi Kendig will design the CI tests to ensure this is functioning properly. This step can only be done after the account creation step is complete.

Edit Account

Due Date: 3/6/2020

Prerequisites: Basic Site Framework, Account Creation, Account Login

Tasks:

- Creating account settings page .5ε
- Updating user info in account db 1.5ε

Projected time: 2ε

Assignee(s):

- Felix Brucker
- Dominykas Zobakas
- Lyell Read

Explanation: Felix Brucker and Dominykas Zobakas will pair program any changes that are necessary to implement user account editing. Then, Lyell Read will ensure that any CI checks currently in place cover these new additions, or add new ones. This step must be completed here, as it hinges on many other tasks.

Personalize Account

Due Date: 3/7/2020

Prerequisites: Basic Site Framework, Account Creation

Tasks:

- Client: custom picture field during account creation .5ε
- Server: storing picture with account details 1ε

Projected time: 1.5ε

Assignee(s):

- Felix Brucker
- Robert Detjens
- Dominykas Zobakas

Explanation: Robert Detjens and Felix Brucker will work as a pair programming team to implement account personalization ability. Dominykas Zobakas will ensure that the CI checks are appropriate to test these newly added features. This step belongs here as this step will be greatly facilitated by the completion of the steps above (before) it.

Contributions

Customer

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.

Team Members

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.

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.

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