SpectaCoin Milestone 2

CSC 648 Section 04
Team 4
Joshua Boelman - Team Lead
Jack Deremiah - Backend
Kenneth Galang - Github Master
Madeleine De Mesa - Scrum Master
Elizabeth Kirwan - Frontend
Tony Tran - Frontend

1. Data Definitions V2

This should be reasonably consistent with Milestone 1 but should be expanded as needed and refined as per instructors' feedback. Major data items that comprise of sub-data items have to be defined in full (list all its sub-data items, and for images/video list formats, max size, etc.). You must use all the data definitions and names consistently in all documents and SW, including UI text, naming for main variables, classes and database elements, etc. Focus on data items unique and important to your implementation. Be sure to cover ALL items critical to your project and especially those providing a competitive advantage. At this stage data describing user privileges, and main info (raw data, metadata, supporting data) have to be fully defined (as much as it is possible at this stage)

Backend Definitions

- User
    - User id
    - email
- Portfolio
    - Array of crypto assets
    - Created at (date)

Frontend Definitions

Name	Meaning
Account ID	Unique identifier for User
Username	Login
Password	Login
Email	Authentication
Tracking Cryptocurrency ID	Identifier
Cryptocurrency Name	Identifier (ticker)
Cryptocurrency Money Value	Currency (in USD)
User Portfolio	Container of user's favorite crypto assets (Array)

User Status	(Admin, Moderator, Standard)
Admin	Control of App
Moderator	Assistance in Control
Standard User	Default
Cryptocurrency Search Filter	Filters type of Crypto

2. Functional Requirements V2

Expand functional requirements from Milestone 1 into Milestone 2, with more details as necessary. Keep the same reference numbers with respect to Milestone 1 (i.e. if the high-level requirement was number R.3. in Milestone 1, then Milestone 2 more detailed requirements of requirement R.3. are R.3.1., R.3.2. etc.).

Prioritize each requirement/spec with 1, 2, 3. (1-must have; 2 – desired; 3 – opportunistic as defined in the class). To develop these priorities on behalf of the user, and make your application complete from usability, marketing, and business aspects. The priorities you set later in Milestone 3 and 4 will constitute your commitment (especially priorities of 1).

ID		Title	Description	Priority
1.1	User Feature	Account - Creation	Users can create a new account by entering an email address.	1
1.2	User Feature	Account - Supabase	Supabase stores newly created accounts into a database.	1
2.1	User Feature	Login - enter email	If a user already has an account, they can log in by entering their email address into a form on our “Login” page.	1
2.2	User Feature	Login - Supabase Authentication	Supabase searches for the user in its database. When it finds them, the user is sent a link to log in.	1
2.3	User Feature	Login - check email	Users go to their email and look for a message from supabase. This message should have a link which logs them into our web application.	1
3.1	Portfolio	Dashboard -	The dashboard is where users view their	1

	Feature	View	portfolio of crypto currencies. If a user is not logged in, they don't have this feature.	
3.2	Portfolio Feature	Dashboard - Add/Remove	Users get to add or remove cryptos to their dashboard/portfolio.	1
3.3	Portfolio Feature	Dashboard - Supabase	Supabase stores the portfolios of each user.	1
4.1	User Feature	Search	Users can use the search function to look up specific crypto currencies.	1
4.2	Portfolio Feature	Search - add to Portfolio	If the user is logged in, they have the option of adding cryptos to their portfolio.	1
5.1	User Feature	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, whether the price is rising or falling.	2
6.1	User Feature	Homepage	All users have a Homepage, whether or not they're currently logged in.	1
6.2	User Feature	Trending	This section displays the most popular crypto currencies.	2
7.1	User Feature	Browse All	Users can view all crypto currencies listed in alphabetical order.	2
7.2	User Feature	Browse All - Filter	Users can also filter the "Browse All" results using the same filter as the "Search" feature.	3
8.1	User Feature	Forum - List Posts	The "Forum" will display a list of user-created posts that people can browse through.	3
8.2	User Feature	Forum - Create Posts	The "Forum" feature will also allow users to create posts. They can ask questions or discuss anything crypto currency related, whether it be tips for investing, past experiences, or predictions for the future of crypto currency.related.	3
8.3	User Feature	Forum - Responses	Users can also read and respond to already created posts.	3
9.1	User Feature	Accessibility - Light Mode	This is the default setting of the web application.	3
9.2	User Feature	Accessibility - Dark Mode	Users can also switch to dark mode. They may want to do this when viewing the web application in a dim setting.	3

10.1	User Feature	Remove Account	Users can delete their account from our website.	3
11.1	User Feature	Email Notifications	When creating an account, users can sign up to get notifications about cryptos listed in their dashboard.	3

3. UI Mockups and Storyboards (high level only)

Create storyboards for 5~6 major functional requirements (e.g., Priority 1 requirements). Start with black and white wire diagrams focusing on basic UX flows for 5~6 major functional requirements. Create simple “storyboards” (sequence of mockups) which represent the functional requirements.

The format for UI mockups is very flexible. Do not use graphics or colors yet (unless absolutely necessary), it draws attention from basic UI concepts (functions, behaviors, layouts, flow…).

Once the storyboard is developed, perform the UX validation meeting so that UX principles are ensured. Include the summary of your UX validation meetings.

Design GUI design for the developed storyboard.

We recommend front-end team be assigned to this task.

Account Creation
—
home -> login -> register -> register confirmed

Login
—
home -> login

Dashboard
—
home -> login -> User dashboard

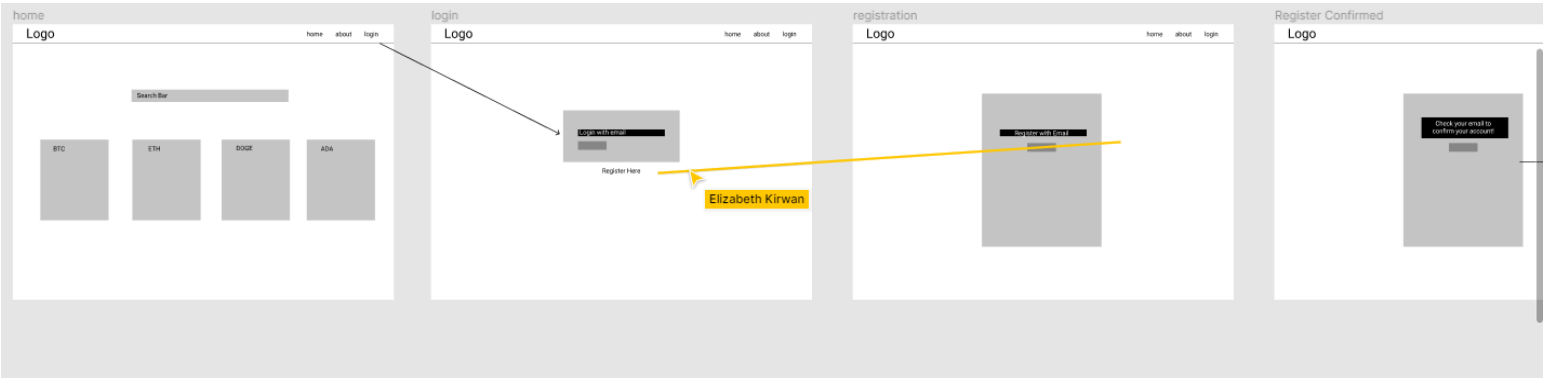
Search
—
home -> dashboard -> Search Results

Filter
—
home -> dashboard -> Search/Filter

Trending
—
home -> Trending

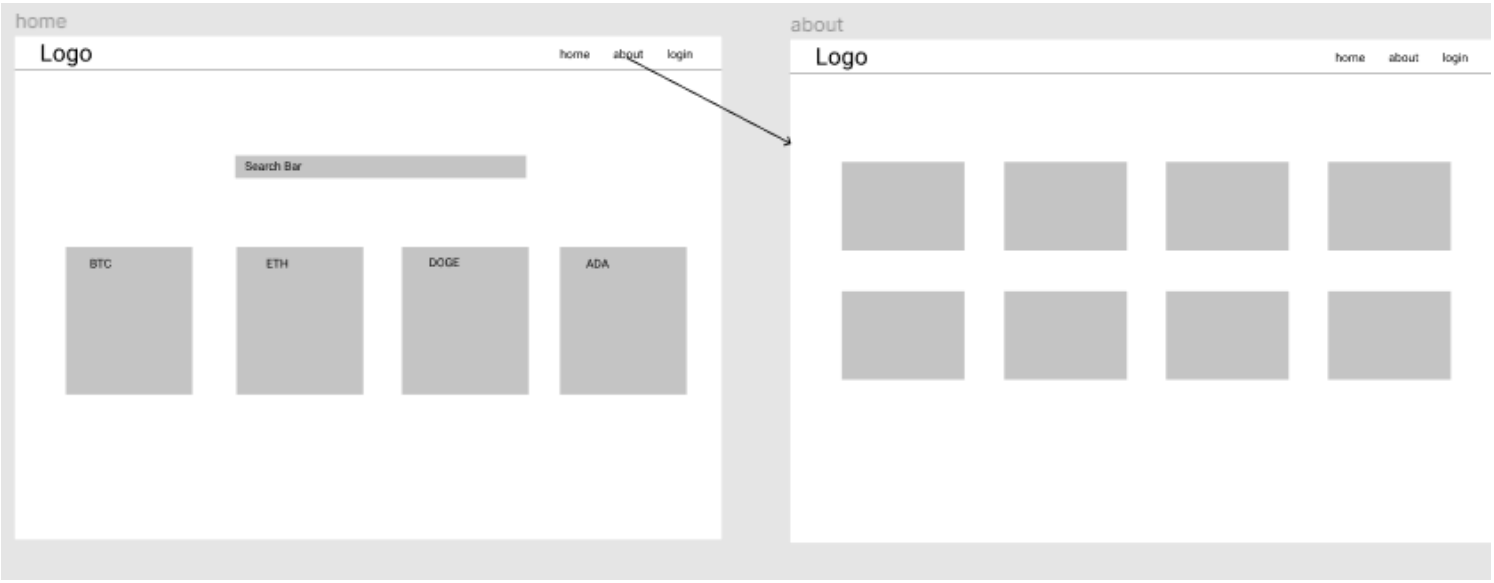
Browse all
—
home -> browse all button

Home -> Login -> Registration -> Registration Confirmation



Home -> Login -> Dashboard

Home -> About



Home -> Search from Main page -> Search from Search Results page

4. High-level Architecture, Database Organization

DB organization: Describe the main database schema/organization (high-level), e.g., list main DB tables (e.g. their titles) and items in each DB table. Make sure the titles and

variable names are easy to understand in plain English and consistent with data definitions in Section 1 above.

Profiles/User Table
We would store User ID and their emails here. Their account will be added once they confirm their email.

User ID (unique user id)	Emails (email)

Portfolio Table
id - Unique number to keep track of the user's portfolio for certain coins.
crypto- Stores the specified crypto, so that we can use it to fetch from crypto api.
amount- Stores how much the user owns of that crypto.
user_id- Relates to profile table and gets the user id.

id (int8)	crypto (varchar)	amount (int 8)	user_id (connects to profile User ID)

Add/Delete/Search architecture:
In regard to the functional requirements, please specify which DB operations are permitted on DB (e.g., what DB entries will be added, searched, deleted, and displayed) Describe the technical feasibility of those DB operations in detail.

For Profiles/User Table, users can add once they confirm their email.

For the Portfolio Table, users can add, delete, and search their data. They can add their portfolio/wallet and the amount they own. If they no longer decide to watch that wallet, they can delete it from their portfolio. User's portfolios will be displayed only to themselves.

APIs

Your own APIs: Describe and define at high-level major APIs that you will create among your modules. Here the APIs are class name and method name.

For example, list the name of major route functions and controller functions.

If you are using the 3rd party API, please describe them in your architecture.
If you are using open-source components, please describe them in your architecture.

- Coinbase api's
  - Endpoint -> https://api.coinbase.com/v2/prices/${symbol}-usd/buy
  - ex request (client calls function getPrice(eth) * eth is the symbol
  - Response
    - {"data":{"base":"ETH","currency":"USD","amount":"2702.66"}}

- Supabase api's
  - ```
    await supabase
    ```
  - .from("portfolio")
    - ```
      .insert([{ user_id: user.id, crypto: ${symbol}-usd, amount: 1 }]);
      ```

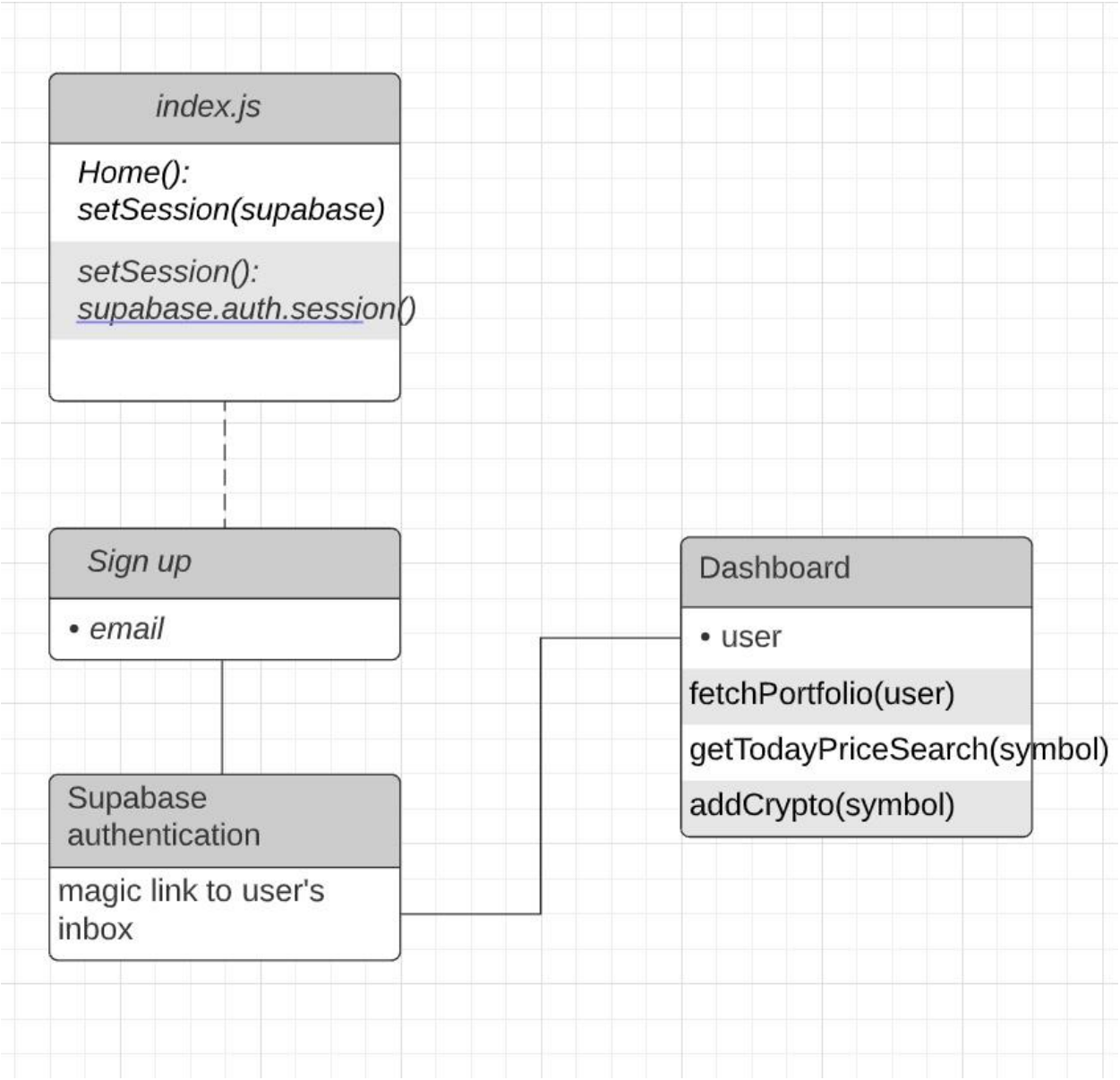
If you have changed SW tools and frameworks or added any new ones, please describe them.

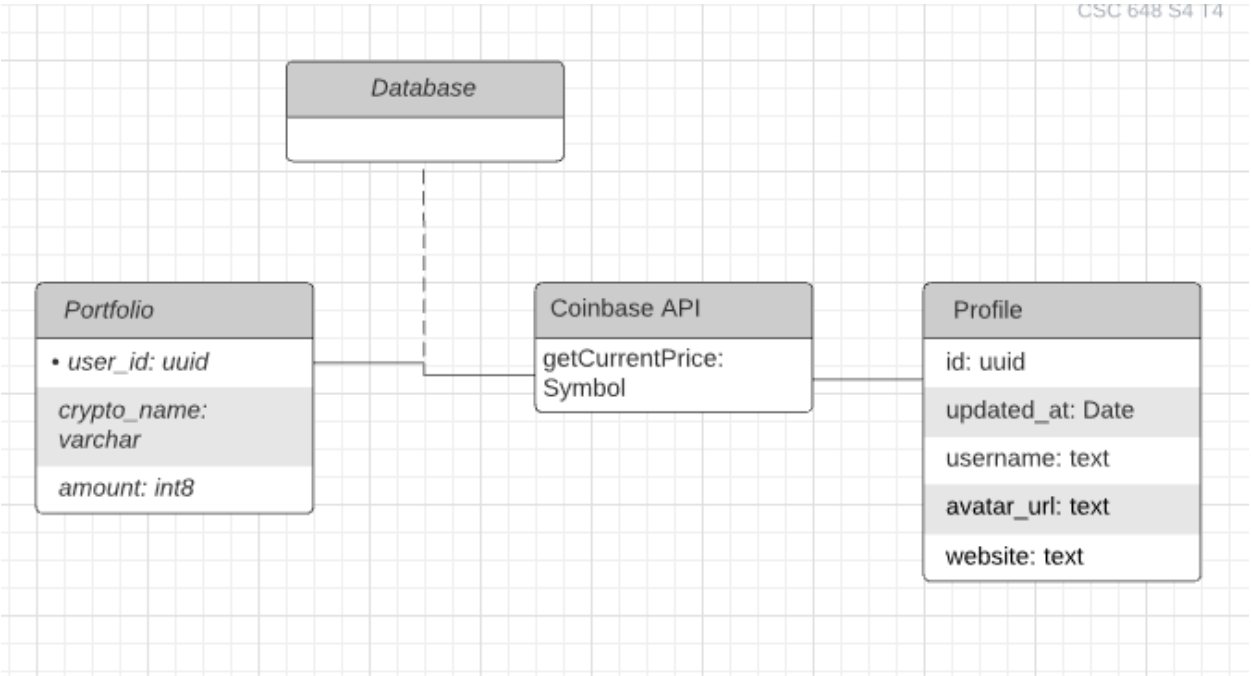
5. High-Level UML Diagrams

Familiarize yourself with Unified Modeling Language (UML). Find your favorite UML tutorials from the Internet.

For Milestone 2, provide only:
High-level UML class diagrams for implementation classes of core functionality, i.e. functionality with provided interfaces. Focus on main high-level classes only (one or at most two levels deep). This must reflect an OO approach to implementing your website. For UML, you could find many references including http://edn.embarcadero.com/article/31863.

High-level sequence diagrams: for ~5 major functional requirements, please develop a UML sequence diagram.


index.js
Home():
setSession(supabase)
setSession():
supabase.auth.session()
Sign up
• email
Supabase
authentication
magic link to user's
inbox
Dashboard
• user
fetchPortfolio(user)
getTodayPriceSearch(symbol)
addCrypto(symbol)



6 . Identify actual key risks for your project at this time

One of the skill risks in our team is that we don't have too much experience on Supabase. In order to mitigate that issue, we all are trying to identify pages in the documentation that could help with how we fetch data for our app. We have a proper study plan where we report what technologies that are beneficial to the development of our app for every meeting. In every meeting, the scrum master creates a Notion page (team management application) that demonstrates what went on in the meeting and what each role member should finish by the next meeting. For new developments of our app and pull requests, our team uses the instant messaging app of Discord, where the team can get notified when a new development feature has been made. Everyone meets at the meeting regularly, and if someone can't make it, it's up to the team member to ask the scrum master what they have missed. Everyone in the team is trying to keep their pace, and if an event occurs where they can't pull themselves, it is up to one of the team members to educate them to perform their duties.
There are no legal nor content risks since we are using coinbase APIs.

7. Project management

Each team member's progress is shared during team meetings. Outside the scrum meeting, our team uses the team management application, Notion, to define what tasks are finished and the ones that need to be done. It also displays meeting minutes, meeting notes and tasks to all members and it is updated bi-weekly. To manage M2 tasks, we decided to split each section off the Milestone 2 doc to each team

member. Some sections were split into two people to collaborate and their respective roles.