Develop Your Project with Vue

Authors:

Guilherme Carra

Yulia Belyakova

Christian Callau

Jorge García

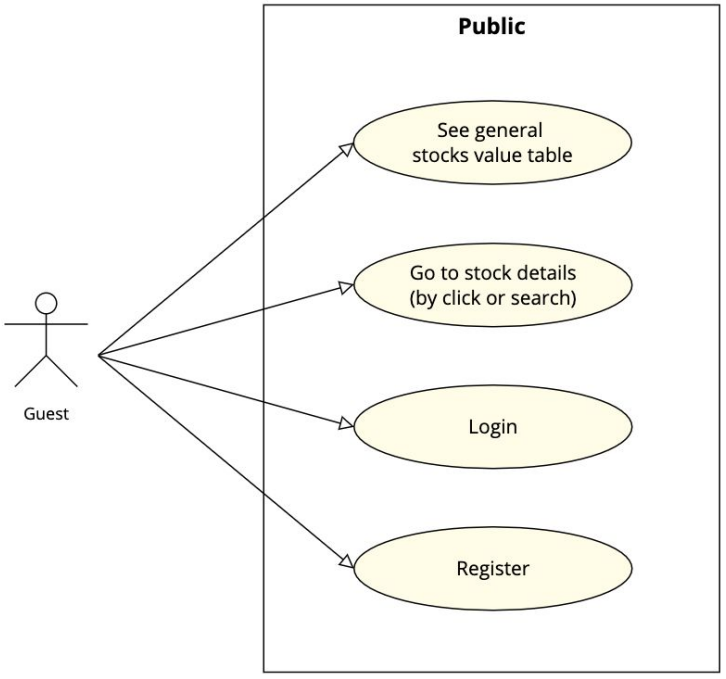
Table of content

1. Use case diagrams

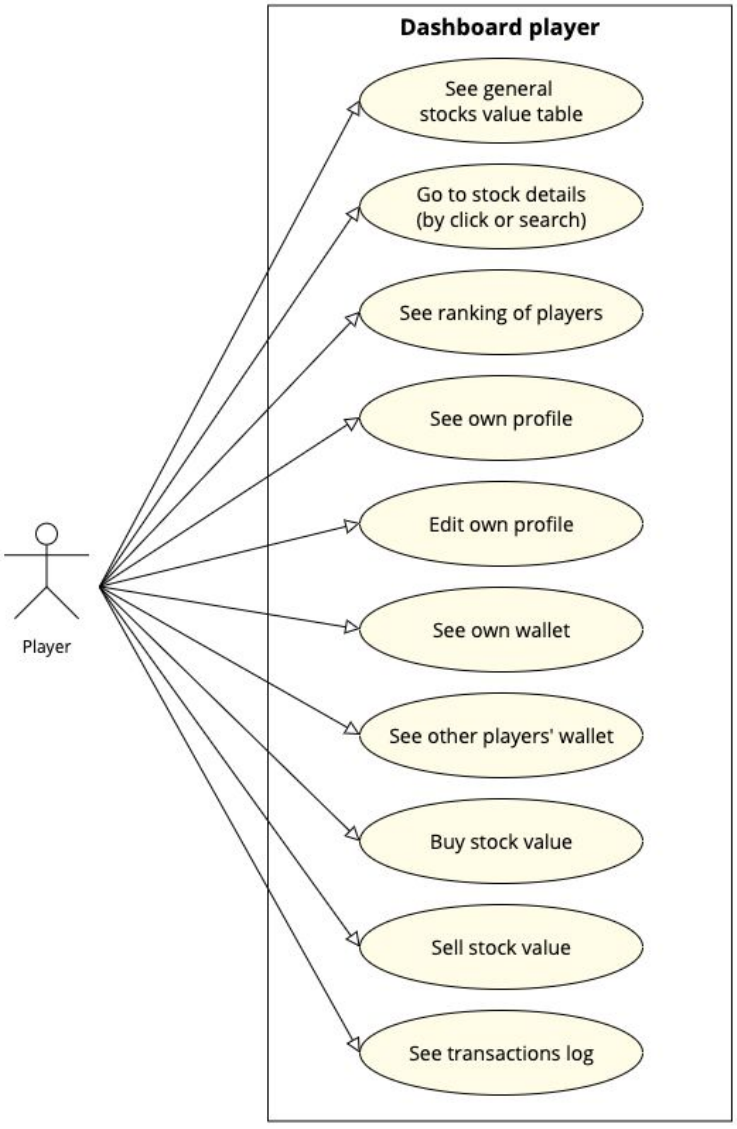
The use case diagram should represent in a graphic, simplified manner the interaction possibilities of the different participants using a given platform (like a webpage) or program.

In this project, we could identify the following:

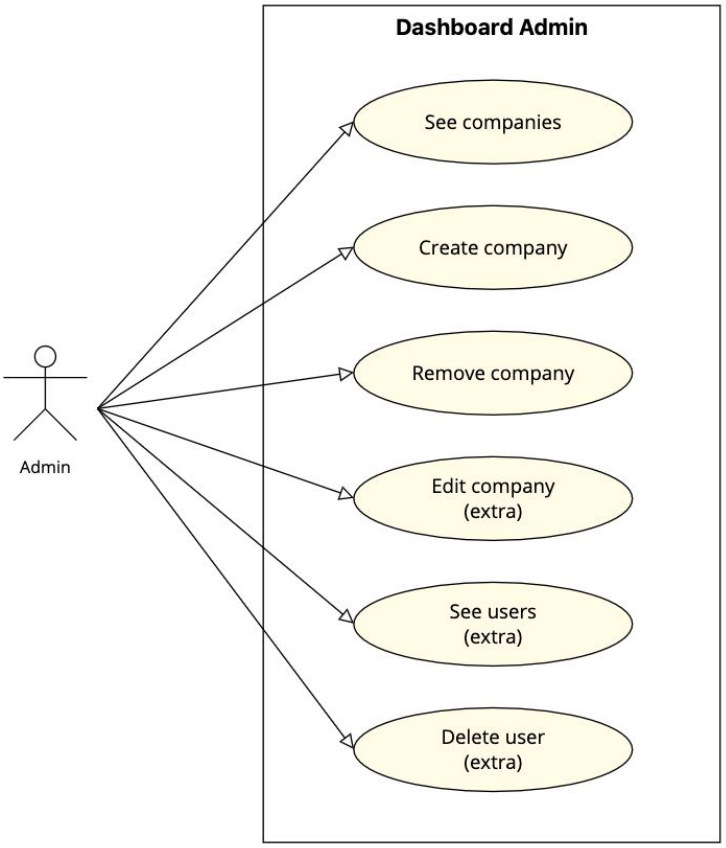
1.1. Public access pages



1.2. Player dashboard

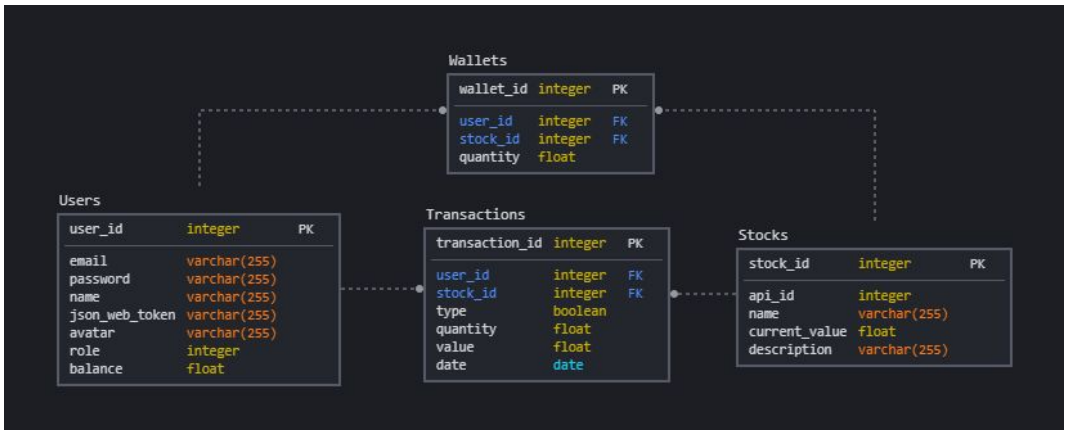


1.3. Admin dashboard



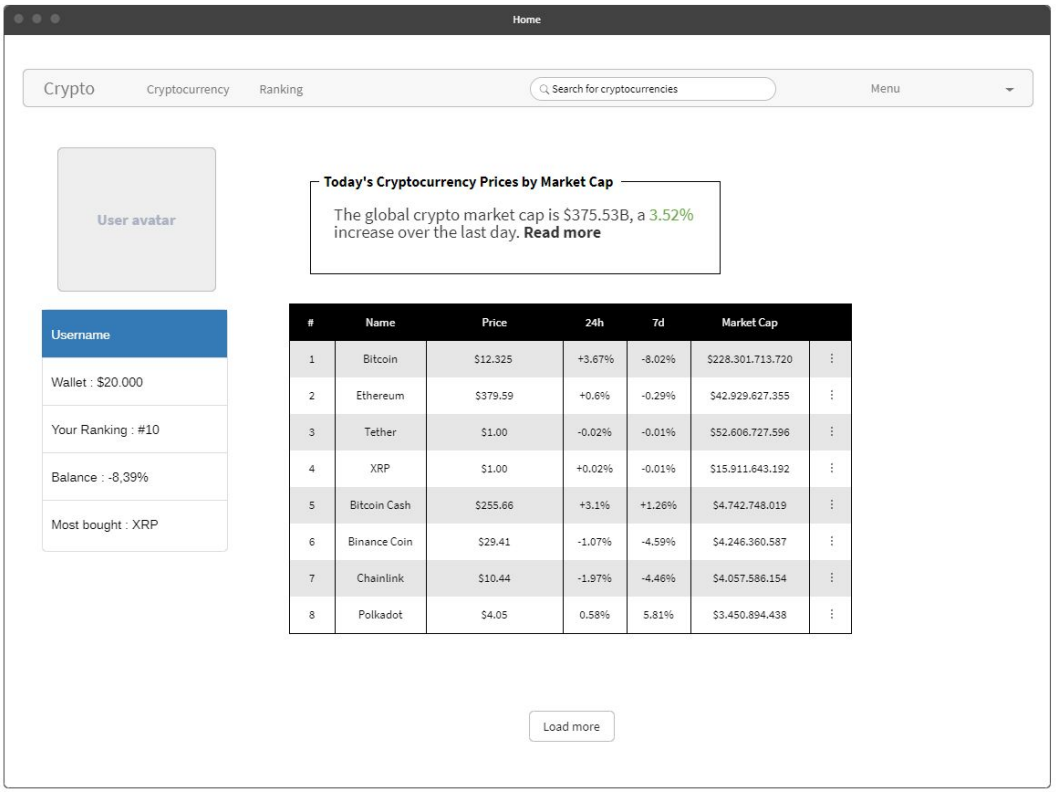
2. Database design diagram

We used the following database design in our project, consisting basically in the four tables and their relations as per the screenshot below:

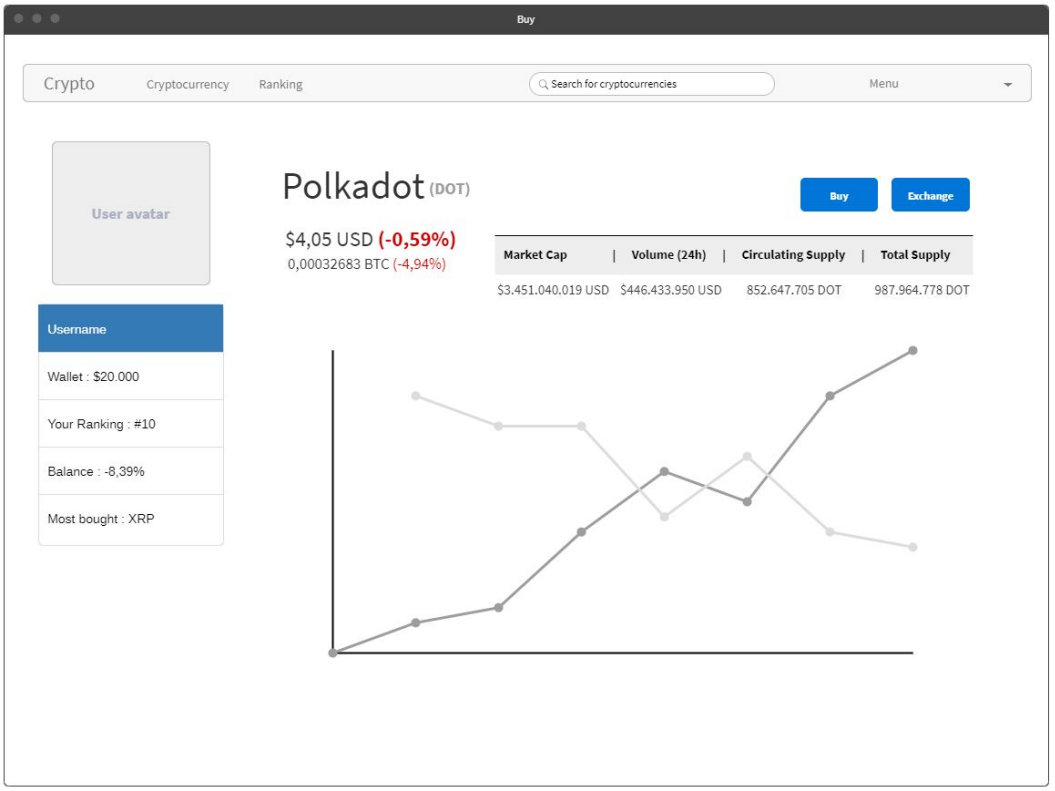


3. Basic wireframe design

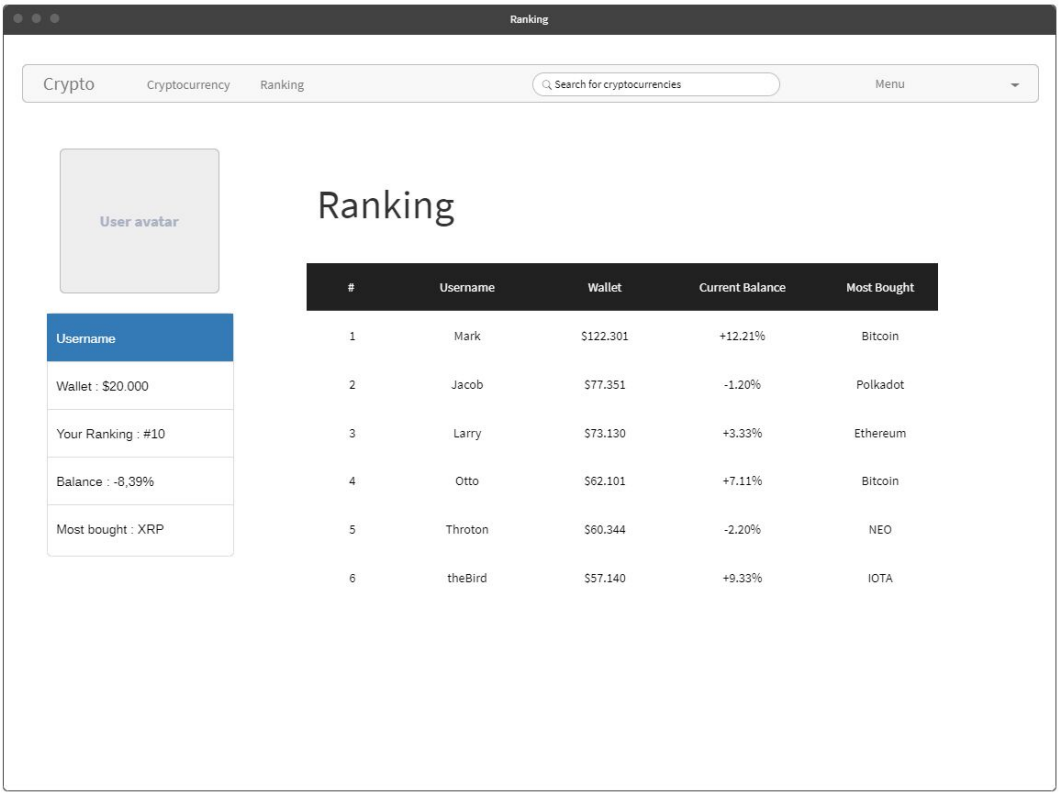
3.1. Homepage



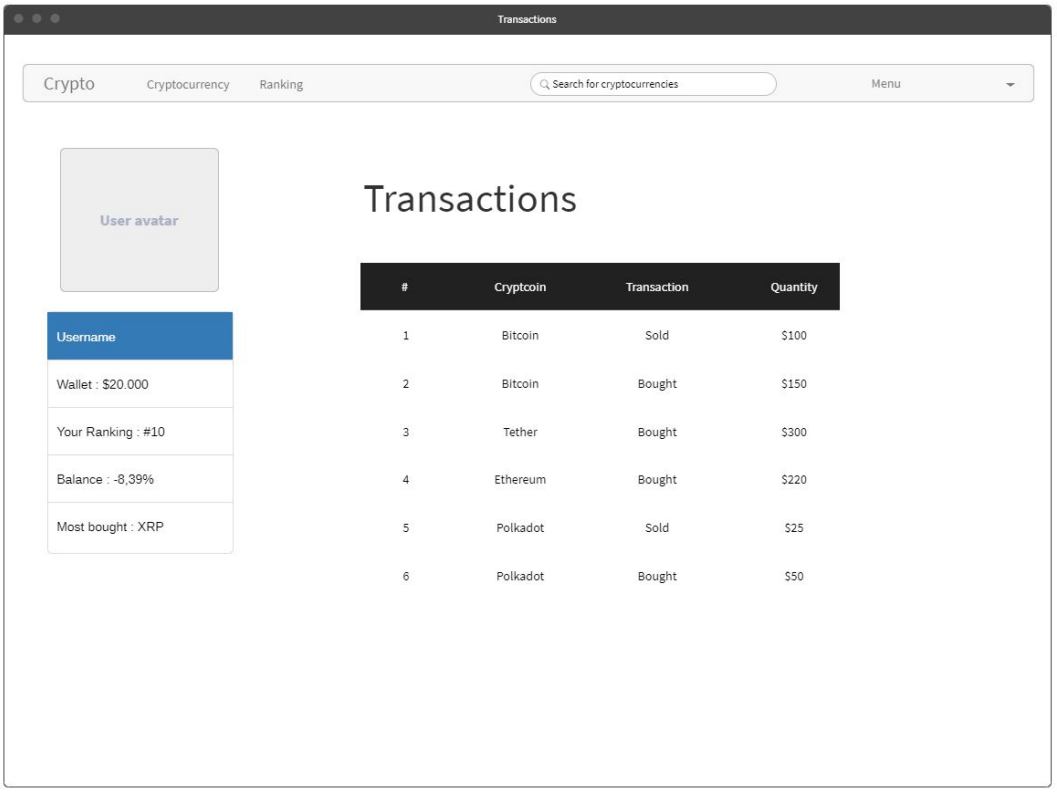
3.2. Details


Buy
Crypto
Cryptocurrency
Ranking
Search for cryptocurrencies
Menu
User avatar
Polkadot (DOT)
Buy
Exchange
$4,05 USD (-0,59%)
0,00032683 BTC (-4,94%)
Market Cap | Volume (24h) | Circulating Supply | Total Supply
$3.451.040.019 USD $446.433.950 USD 852.647.705 DOT 987.964.778 DOT
Username
Wallet : $20.000
Your Ranking : #10
Balance : -8,39%
Most bought : XRP

3.3. Ranking


Ranking
Crypto
Cryptocurrency
Ranking
Search for cryptocurrencies
Menu
User avatar
Username
Wallet : $20.000
Your Ranking : #10
Balance : -8,39%
Most bought : XRP
Ranking
Username Wallet Current Balance Most Bought
1 Mark $122.301 +12.21% Bitcoin
2 Jacob $77.351 -1.20% Polkadot
3 Larry $73.130 +3.33% Ethereum
4 Otto $62.101 +7.11% Bitcoin
5 Throton $60.344 -2.20% NEO
6 theBird $57.140 +9.33% IOTA

3.4. Transactions



4. Technologies used

4.1. Front-end

- Vue.js
- Chart.js
- Bootstrap
- jQuery
- Axios/Fetch
- SCSS/SASS

4.2. Back-end

- node.js (with common.js)
- Express
- Bcrypt
- JSON Web Tokens
- SQL ORM Sequelize
- External API (https://www.coingecko.com/en/api)
- Axios/Fetch

5. Roles and Responsibilities

5.1. Front-end

	Guilherme	*Jorge*
Selected role	Full Stack	Full Stack
Main Tasks	• SPA design with Vue.js based on single file components • Routing • CRUD-views authentication using Vue routing • Pagination • Ajax request to back-end service	• Project lead • Documentation • Tasks follow-up • Project communication and coordination
Supports with	-	• SPA design/components • Implementation of Chart.js

5.2. Back-end

	Yulia	*Christian*
Selected role	Back-end	Back-end
Main Tasks	• Database design and creation • MVC pattern: Controllers • Endpoints creation • MVC pattern: Models • ORM Sequelize • Ajax request to third-party API • README file	• Database design and creation • MVC pattern: Controllers • Endpoints creation • MVC pattern: Routers • Authentication with JWT • Route protection middleware • Ajax request to third-party API • README file
Supports with	• Front-end	-

6. Initial time estimation

6.1. Day 0 - October 21st

Front-end		Back-end	
Task	*Est. time*	*Task*	*Est. time*
Prepare presentation with project proposal	4h	Prepare presentation with project proposal	4h

6.2. Day 1 - October 22nd

Front-end		**Back-end**	
Task	*Est. time*	*Task*	*Est. time*
Project presentation (all)	1h	Project presentation (all)	1h
Create project structure with CLI	0.5h	Create project structure with Express	0.5h
Define basic routing	1h	Define basic routing	1h
Create .vue files for all necessary views	0.5h	Research on ORM Sequelize	3h
Create components for view Home	3h	Research on login with JSON Web Tokens	3h
Team meeting on daily progress (all)	0.5h	Team meeting on daily progress (all)	0.5h

6.3. Day 2 - October 23rd

Front-end		**Back-end**	
Task	*Est. time*	*Task*	*Est. time*
Design Details view	6.5h	Implement login with JWT	6.5h
Design Login + Register views	6.5h	Implement DB Models with Sequelize	6.5h
Test Chart.js in Homepage	3h		
Team meeting on daily progress (all)	0.5h	Team meeting on daily progress (all)	0.5h

6.4. Day 3 - October 26th

Front-end		Back-end	
Task	*Est. time*	*Task*	*Est. time*
Design admin Dashboard view	6.5h	Implement route protection and finish-up routing	3.5h
Design user Dashboard	6.5h	Define controllers and methods signature	3.5h
Team meeting on daily progress (all)	0.5h	Team meeting on daily progress (all)	0.5h

6.5. Day 4 - October 27th

Front-end		Back-end	
Task	*Est. time*	*Task*	*Est. time*
Implement ajax requests to back-end for corresponding endpoints	3.5h	Implement ajax requests to third-party API to obtain raw data	3.5h
Design error messages when using forms and ajax	3.5h	Implement controller methods to return data in JSON format at defined endpoints	3.5h
Implement CRUD-views authentication with Vue routing	3.5h		
Team meeting on daily progress (all)	0.5h	Team meeting on daily progress (all)	0.5h

6.6. Day 5 - October 28th

Front-end		Back-end	
Task	*Est. time*	*Task*	*Est. time*
Overall review and minor bug fixing	3h	Overall review and minor bug fixing	3h
Front + Back converge tests	3h	Front + Back converge tests	3h
Wrap-up project documentation and README files	2h	Wrap-up project documentation and README files	2h

7. Changes and unforeseen events

- An additional view “Reset password” might be necessary and is added to the design. If the functionality is finally not implemented we can hide or remove the resource.
- APIs for Cryptocurrency handle big amounts of data and are difficult to understand. Additional research time required to pick one and get used to working with it.
- The chart we wanted to implement consisted of a 24-hour historical evolution of the market price for a given cryptocurrency, therefore it included dealing with time data. Chart.js is a little more difficult to implement with time data and it does it through the dependency momentjs, which we didn’t consider initially in the estimation of the development of the feature.
- 404 error page and route not considered in initial project estimation.
- Requests from frontend to backend were not working because CORS was not enabled with Express. Took us some time to figure out what the problem was as the request was getting hanged with status “pending”.
- Use of webSocket instead of HTTP as a protocol to get updated information from API with market prices
- The purchase process of a product which price is updated every second can cause problems when updating the database (maybe the total price is suddenly higher than your available money). We had to figure out a way to handle that error when it happened.

8. Daily log on project progress

Day	Task	Time	Priority	Responsible
1	Create project structure front-end	0.5h	Low	Guilherme
1	Create project structure back-end	0.5h	Low	Christian
1	Create basic routing front-end	1h	Low	Guilherme
1	Create basic routing back-end	1h	Low	Christian
1	Research on usage of Sequelize	7h	Medium	Yulia
1	Create .vue files for all necessary views	0.5h	Low	Guilherme
1	Create components for view Home	3h	Medium	Guilherme
1	Create Login/Register/Reset views	3h	Low	Jorge
1	Create Database tables	2h	Medium	Christian
1	Protect API routes with JWT	3h	Medium	Christian
1	Implement Login and Register endpoints	4h	High	Christian
2	Create Models using Sequelize	7h	Medium	Yulia
2	Create component for error messages and validation logic	2h	Low	Jorge
2	Research on API and required data to render chart	3h	Medium	Jorge
2	Complete design of view Home including rendered data	7h	Medium	Guilherme
2	Create chart with Chart.js based on data response from API (www.coingecko.com/es/api)	4h	Medium	Jorge
2	Start design of Details view including rendered data	4h	Medium	Jorge
3	Connect frontend with backend through API (including CORS)	3h	High	All

3	Implement route protection “middleware” (global guards) with Vue.js	3h	Medium	Jorge
3	Complete CRUD over users and companies	6h	Medium	Yulia
3	Complete design of Details view (buy/sell interaction)	3h	High	Guilherme
3	Bug fixing and adaptation of first endpoints to frontend needs	5h	High	Christian
3	Implement authorization middleware in backend with JWT	2h	Medium	Christian
4	Replace mocked data in views with responses from API	6h	High	Guilherme
4	Complete CRUD over coins in Model	6h	High	Yulia
5	Create admin view to perform CRUD over coins	5h	Medium	Jorge
5	Implement frontend logic for buy/sell functionality	6h	High	Guilherme
5	Implement pagination in view transactions	2h	Medium	Guilherme
5	Implement frontend logic for profile update functionality	3h	Medium	Guilherme
5	Correct endpoints response data to match frontend needs	4h	High	Christian
5	Research on email sendout from node.js to implement password reset page	3h	Low	Yulia
5	Research on historical data storing (24h) to implement endpoint internally instead of using additional third-party API	3h	Medium	Christian
6	Implement pagination in main page (coins)	3h	Medium	Guilherme/Jorge

6	Implement pagination in coins Model	1h	Medium	Yulia
6	General bug fixed and convergence tests	6h	High	All

9. Lessons learned

- Use of ORM Sequelize with a database
- Implement authentication system with JWT
- Implement line chart with time axis in Chart.js
- Creation of a SPA with Vue.js based on single components, including data exchange between views with “props”
- CORS in Express is mandatory when connecting frontend with backend from different ports
- WebSocket as al alternative protocol to HTTP
- Implement route protection with Vue router