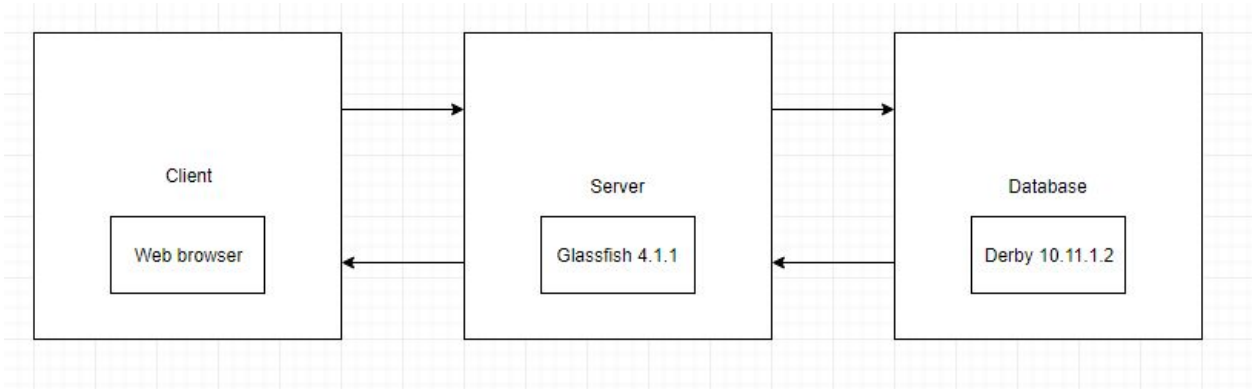
Final project for dat076

Project

For the final project of the course we chose to do a simple trading platform. The idea is to have a simple platform where users can trade different currencies.

Too accomplish this we first wanted to try Node.js but after a while chose to use JEE (Java Platform, Enterprise Edition). We changes after we had alot of problems with Node. With JEE, we first based the project of the server examples from workshop 3, after we tried, we decided we wanted to try a different technique from what we learned in the course. We used java servlets to communicate between the server and the client. This allowed us to use simple HTML and to generate some elements for the different pages. One example is that we generated the trading page depending on what is in the database. For every currency that is in the database, a trading module is generated and added to the trading page. Each module has a graph, facts, information, input field and buttons.

We developed everything in netbeans, using Chrome as the client, Glassfish 4.1.1 as the server and Derby 10.11.1.2 as the database.



Starting the application

Before using the application the following database tables have to be created:

ADMINS
- Key: USERNAME AS varchar(30)
- PW AS varchar(30)

BUGS
- Key: EMAIL AS varchar(30)
- MESSAGE AS varchar(100)

PORTFOLIO:
- Keys: EMAIL AS varchar(30) AND CURRENCY AS varchar(30)
- AMOUNT AS int
- PRICEWHENBOUGHT AS double

STOCKS
- Keys: CURRENCY AS varchar(30) AND TIME AS int
- VALUE AS int

USERS
- Key: EMAIL AS varchar(30)
- PW AS varchar(30)
- ID AS varchar(30) and unique
- Balance AS double

Use cases

We first decided on 10 use cases then added 2 more later, here are the final use cases in the project:

1. Create account
2. Sign in
3. Sign out
4. View my portfolio
5. Currency overview (Trader page)
6. Search for Currencies
7. Buy currency
8. Sell currency
9. Login to admin page
10. See admin overview
11. Add currency
12. Report bugs

In general the responsibilities were the following:
We helped each other on the different parts but the the majority of each use case was done by the person listed:

Fredrik:

- Login to admin page
- See admin overview
- Search for currency
- Add currency
- (Was the main front end programmer)

Arvid:

- Currency overview (Trader page)
- Buy currency
- Sell currency
- Search for Currencies

Johannes

- Create account
- Sign in
- Sign out
- Report bugs

Our git repo:

https://github.com/thefli0222/stocktrader

Results from gitinspector:

Author	Rows	Stability	Age	% in comments
ArvidHast	709	6.3	0.0	11.28
Johannes	442	8.3	0.0	13.35
Johannes Edenholm	121	53.8	0.0	16.53
johede3	51	70.8	0.0	17.65
thefli0222	671	6.1	0.0	13.41

(johede3, Johannes and Johannes Edenholm are all the same person and the rows add up to 614 rows)

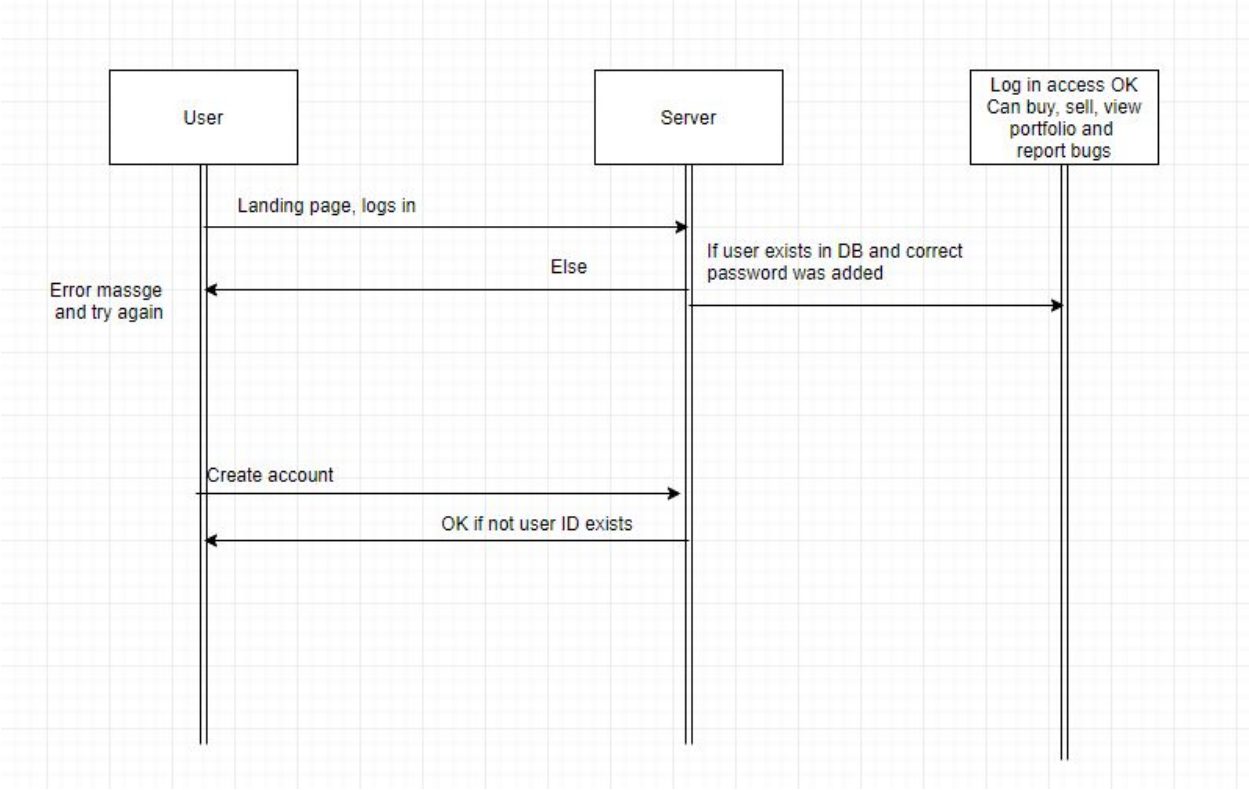
We initially tried to use node.js without success. The project was also hard because our forth team member quit chalmers during the project.

This is the surviving number of rows, the total commits is harder to calculate. In the git repo we have many thousands of rows that can't be counted as contributions. But we most definitely have written more than 800 rows per person during the course of the project.

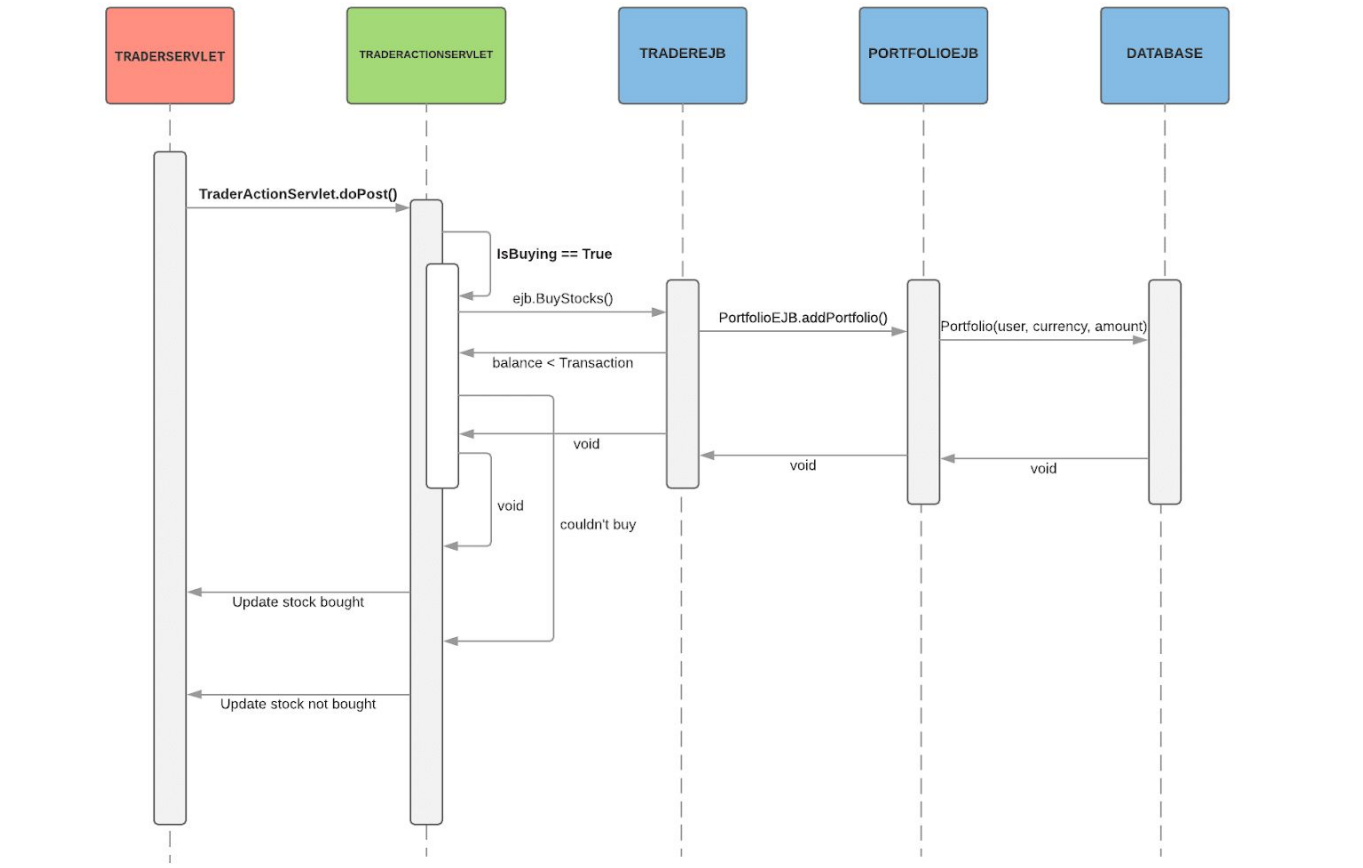
We all have extremely low stability (surviving rows) because we change everything so many times. The project had what you could maybe call an "identity crisis" also build rows and css/scripts that we didn't write was added. Therefore we cleaned the code up and made a git inspect so we could see rows that had survived.

Sequence diagrams:

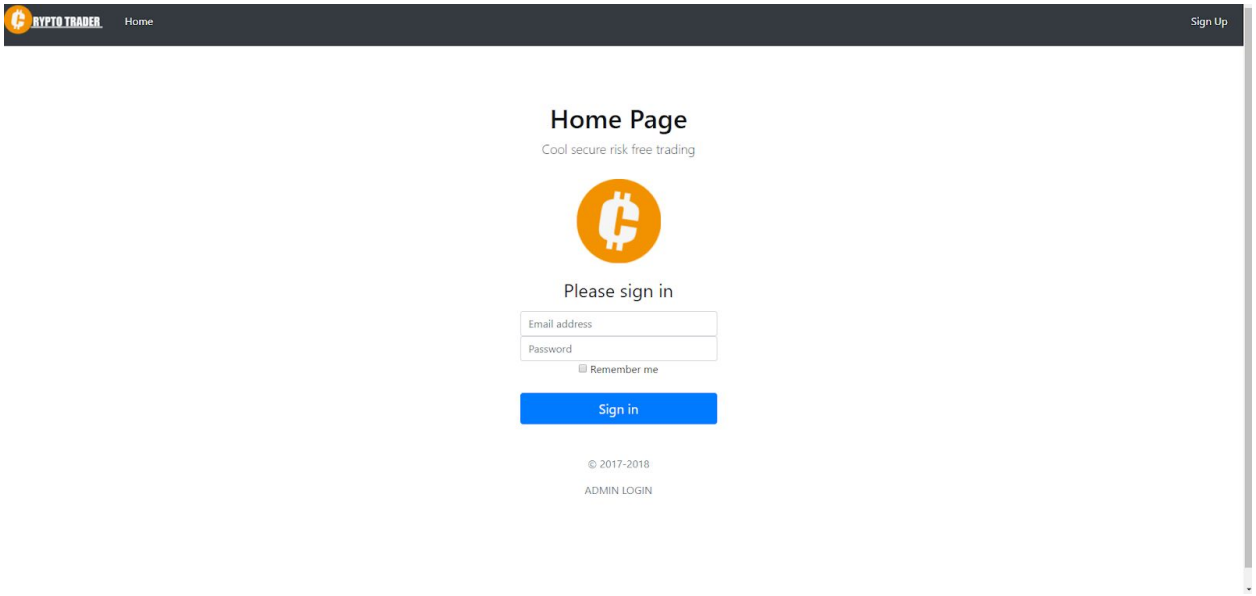
Basic abstraction of the buy stock sequenze



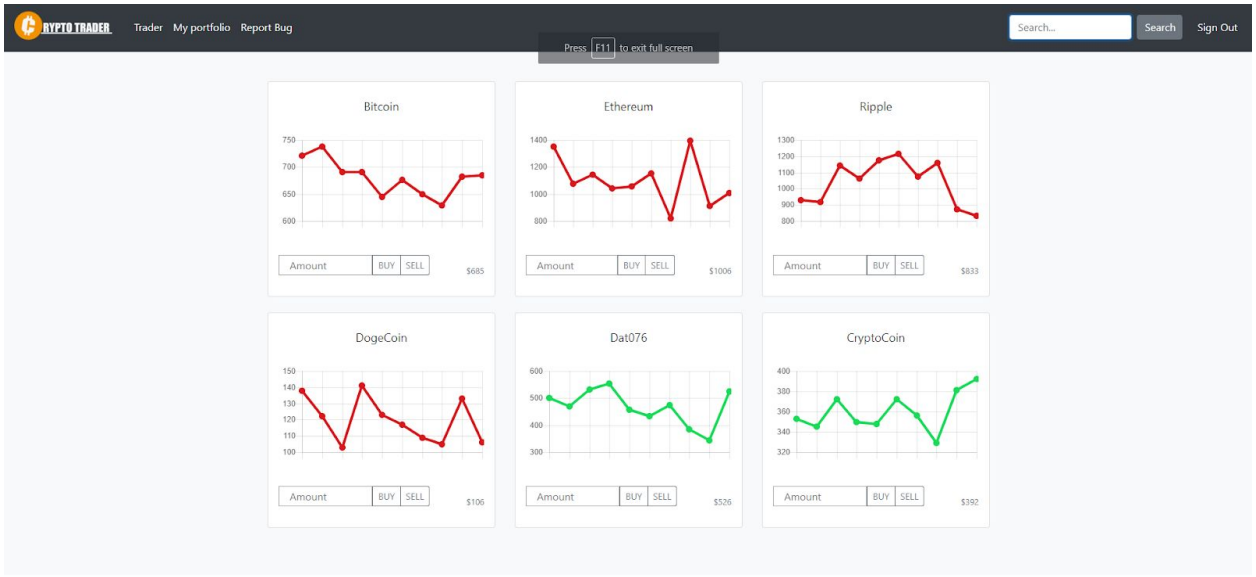
Sequence of the buy use case:



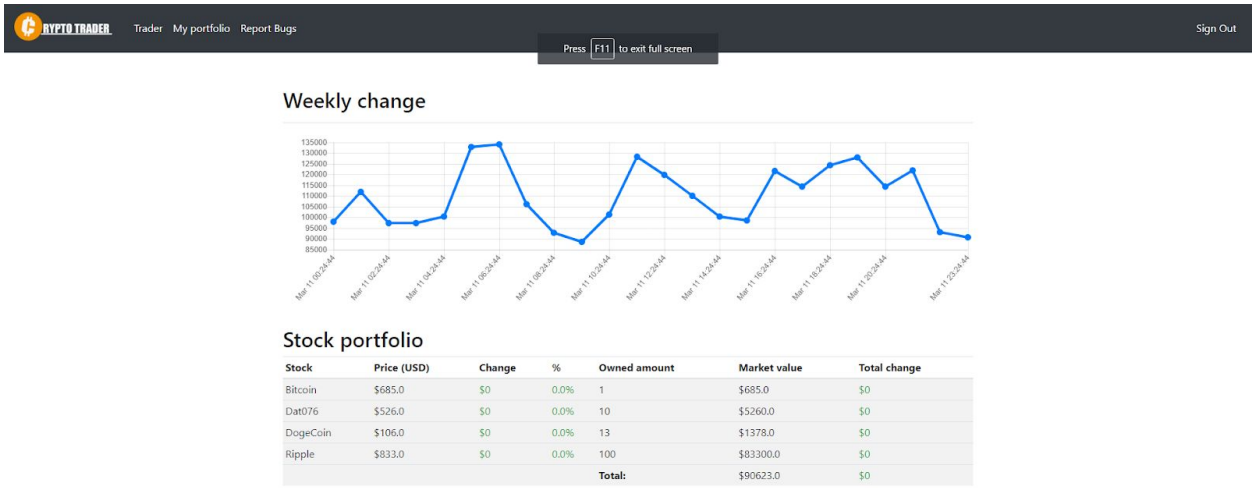
Pictures of some of the pages:



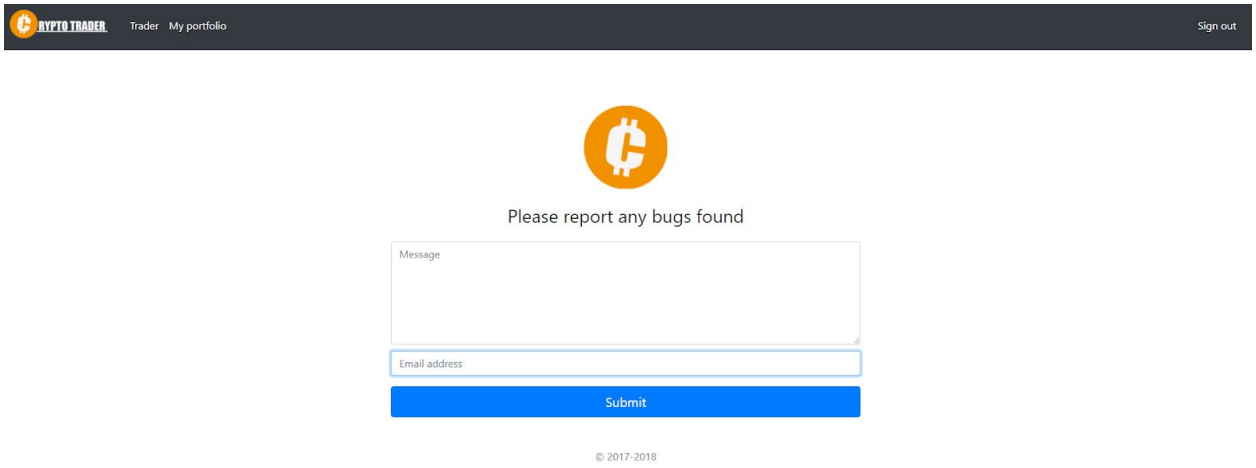
Trading view



Portfolio view

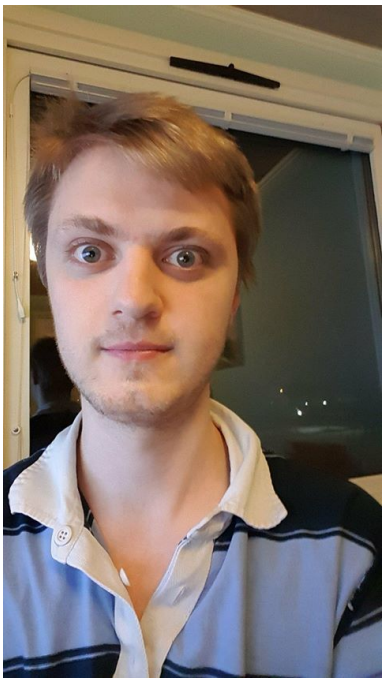


Report Bugs view

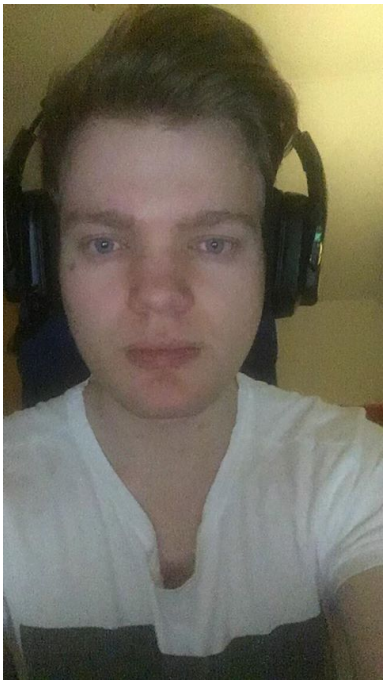


We don’t know if this is what you mean by “picture of the group in the Documentation.” but here it is:

Fredrik Lindevall



Arvid Hast



Johannes Edenholm