Topicus bank transaction analysis tool

The goal of our project was to create an application which analyzes bank transactions in order to give a better financial insight in the performance of a company, thus helping banks calculate the possible risk of lending a loan to a certain company. We achieved this by creating an application which digests the content of MT940 files and outputs it as structured graphs and tables on the main page of the app .

Banks always risked when they lent loans to businesses. Some businesses are safer to lend money to than others and there are many ways to check whether it is worth lending a loan or not. The intent of our web application is to help banks analyse their clients' bank account history.

By presenting the information about a businesses transaction history in an informative way, our project will help risk managers see nuances of the business that they may not have seen otherwise. Seeing a bigger picture will allow Banks to avoid risky loans and create less risky ones, thus saving thousands of euros per year.

We approached the project in different phases. Starting with the initialization phase. Here our main focus was to get to know the team and each other's qualities and setting our objectives while we identified our key stakeholders.

The second phase was the planning of our set objectives. We divided tasks in such a way that they were achievable for each individual group member, while still making sure there was a more or less fair division of labor. Also in this phase a deadline for all the mandatory requirements was set to ensure we would not experience any time shortage.

In the development phase we worked on all tasks and stimulated each other to work on the project. This was the longest and the most productive period of the project development. At this point we developed the two most important parts: the front end and the back end. The front end, which will be the visible interface for the user and the back-end, which contains all functional aspects of the application and makes sure the analysis tool is running smoothly.

Front-end: This was mostly implemented with the use of HTML, CSS and JavaScript with the help of Bootstrap. We managed to create a table where the user can see the uploaded MT940 files. Uploading other file types will result in an alert message. This also happens when you try to reupload the exact same file or the file with the same account ID. When a file is uploaded it will display the content in a table form. Here can view the account id, initial amount, initial Date, final amount and final Date. Furthermore, you have the option to press the analytics button, which will lead you to the dashboard page where you can see graphs which visualize the information. The biggest graph shows the amount of money the account had over

time. The pie chart shows the ratio between risky and safe transactions. We consider a risky transaction by counting the amount of transactions that leaves an amount of money more than (e.g.10000) and divide it by the total number of transactions (if the amount is between e.g(5000 and 10000) we count it as half(1/2))

The other, probably more important part of our project is the back-end, as without it, our project is just a shell without purpose. The backend consists of the parser, the security, the database and finally the bridge between the database and the frontend.

The parser was implemented by us with the help of an API which we found on our beloved internet. It scans every line of the MT940 file and converts those strings into database compatible chunks of information. After we parse the file we make use of the SQL queries in order to insert the data to the database. In our database we keep track of virtually almost every piece of information we could extract from the MT940 file, for example the date, time, amountand even the receiver of a transaction. Database consists of 4 tables. Balance, Transaction, Money, and Users. We omitted field86 because we decided that it does not give any additional value for the analysis.

The bridge between our database and frontend is the piece of code that takes the information from the database and displays it onto the front end. Here is where our graphs get life and utility. In this part of the project we thought about what would be useful for a risk manager to know in order to avoid losing Bank's money.

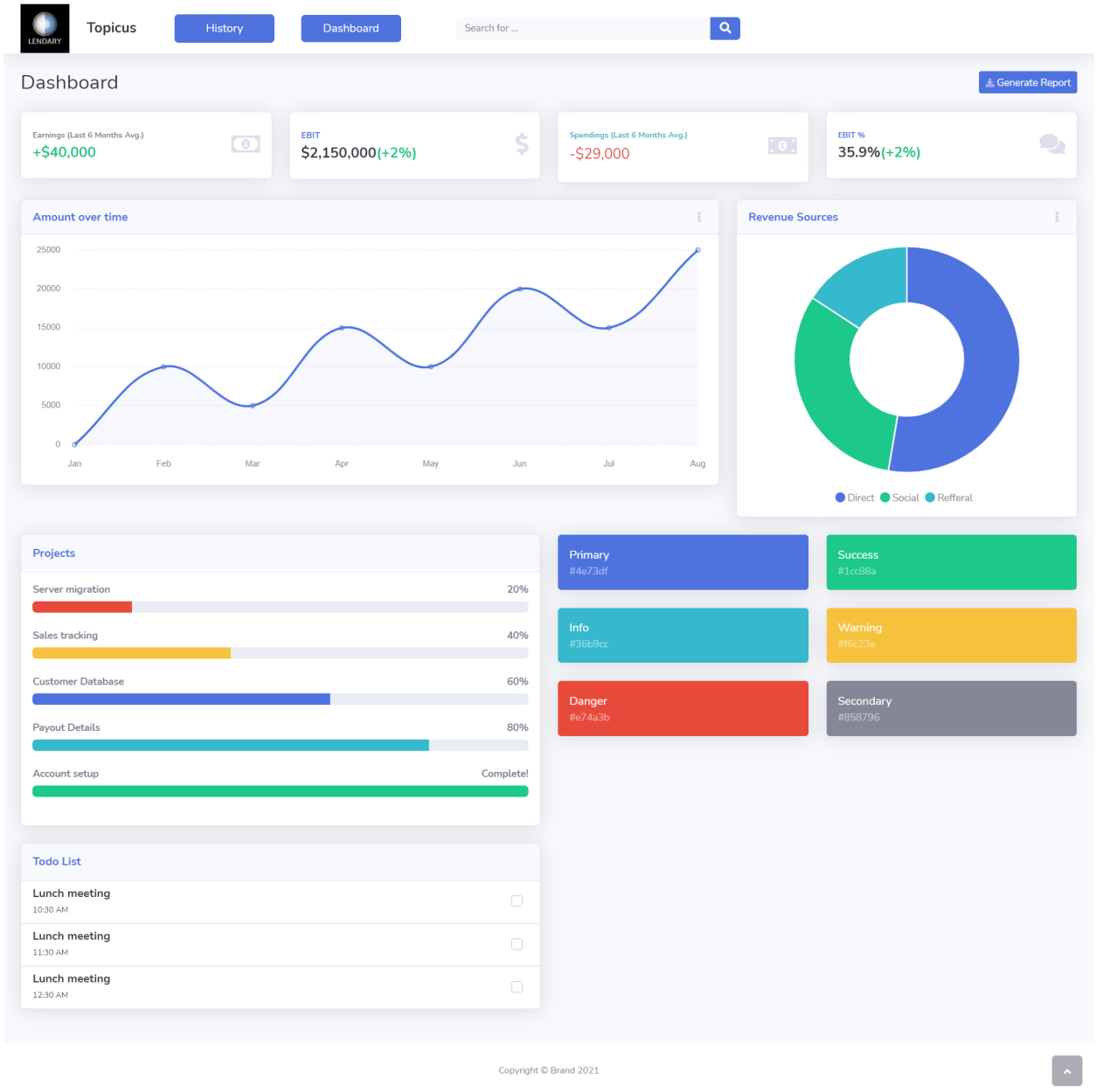
Finally we have the security part where we tried our best to create useful exceptions. When it comes to our app we make use of prepared statements, so we utilize the input and make sure that malicious script from the file cannot be performed. Moreover, we have created a log-in page where users have to log in with their credentials. Here we use the SHA-256 hash function to store hashed passwords in our database. Therefore, in the case the data leaks from the database, it will be almost impossible for the attacker to find out what the password is.

In the closing phase we looked at what still had to be implemented and if we forgot anything to successfully close the project. In this period we finished and deployed the project.

What really helped us during the planning phase is the planning app trello. Here we put in the tasks that had to be done and who was doing it. This gave a clear overview of the progress we made.

In the end, the result was as desired. We completed all our goals and worked effectively throughout the project period. There were some obstacles. Successfully working with gitlab was quite a challenge for us, but in the end everything worked out really well. Also with the help of code together. This is the reason most commits are done by one team member. We are satisfied with the end product and really think that it could be useful for real risk managers in real banks.

Below we posted some screenshots of the initial mock-up.



Topicus

History | Dashboard

Search for ...

Upload | Export Zip. | Querry 1

Show 10

Search

Name	IBAN	Date	Month In	Month Out
Airi Satou	NL34 ABN0 1267 2147	02.06.2021	$11111111	$162,700
Angelica Ramos	NL34 ABN0 1267 2147	02.06.2021	$11111111	$1,200,000
Ashton Cox	NL34 ABN0 1267 2147	02.06.2021	$11111111	$1,244,444
Bradley Greer	NL34 ABN0 1267 2147	02.06.2021	$11111111	$132,000
Brenden Wagner	NL34 ABN0 1267 2147	02.06.2021	$11111111	$206,850
Brielle Williamson	NL34 ABN0 1267 2147	02.06.2021	$11111111	$372,000
Bruno Nash	NL34 ABN0 1267 2147	02.06.2021	$11111111	$163,500
Caesar Vance	NL34 ABN0 1267 2147	02.06.2021	$11111111	$106,450
Cara Stevens	NL34 ABN0 1267 2147	02.06.2021	$11111111	$145,600
Cedric Kelly	NL34 ABN0 1267 2147	02.06.2021	$11111111	$433,060
Name	**Position**	**Office**	**Start date**	**Salary**

Showing 1 to 10 of 27

« 1 2 3 »

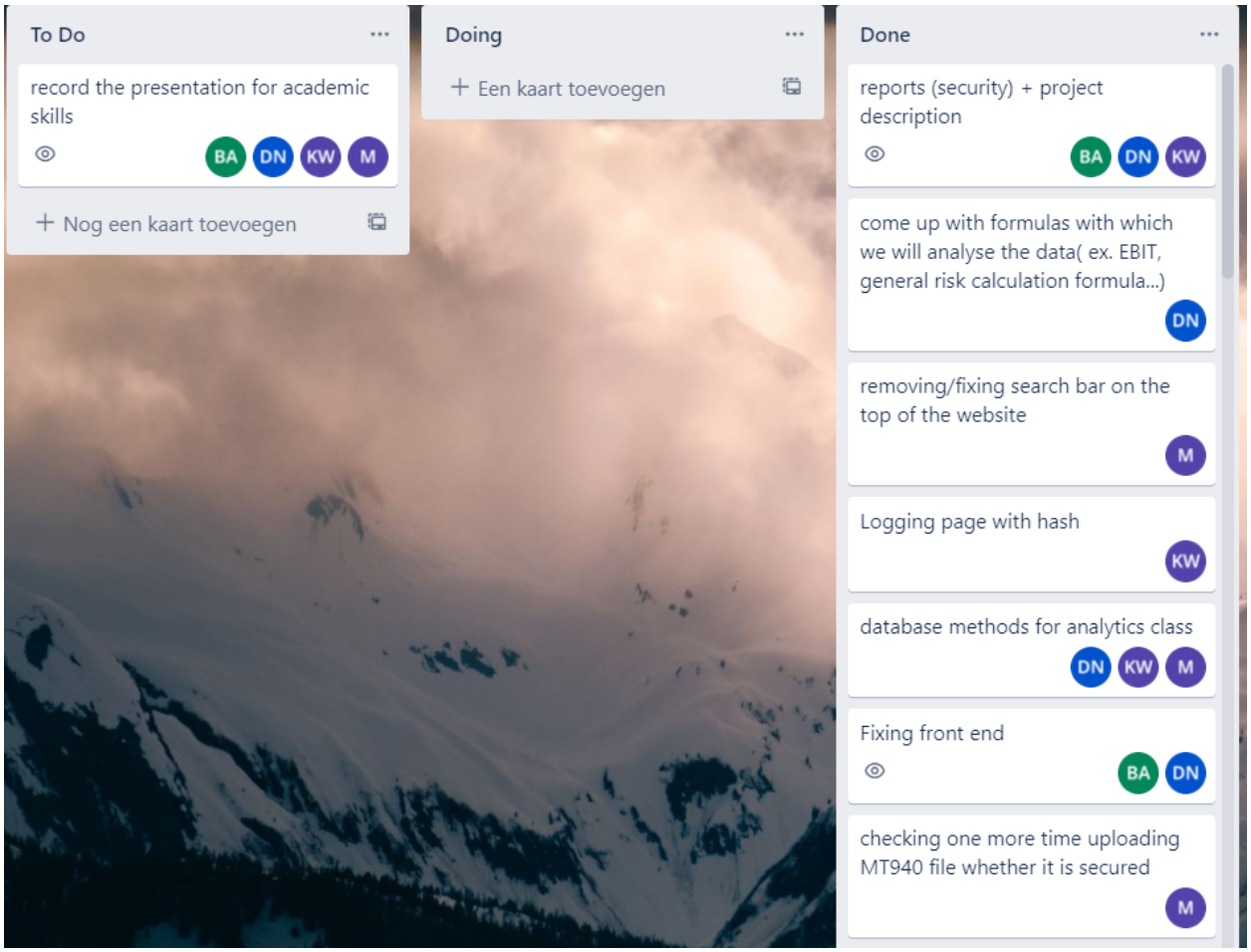
Welcome Back!

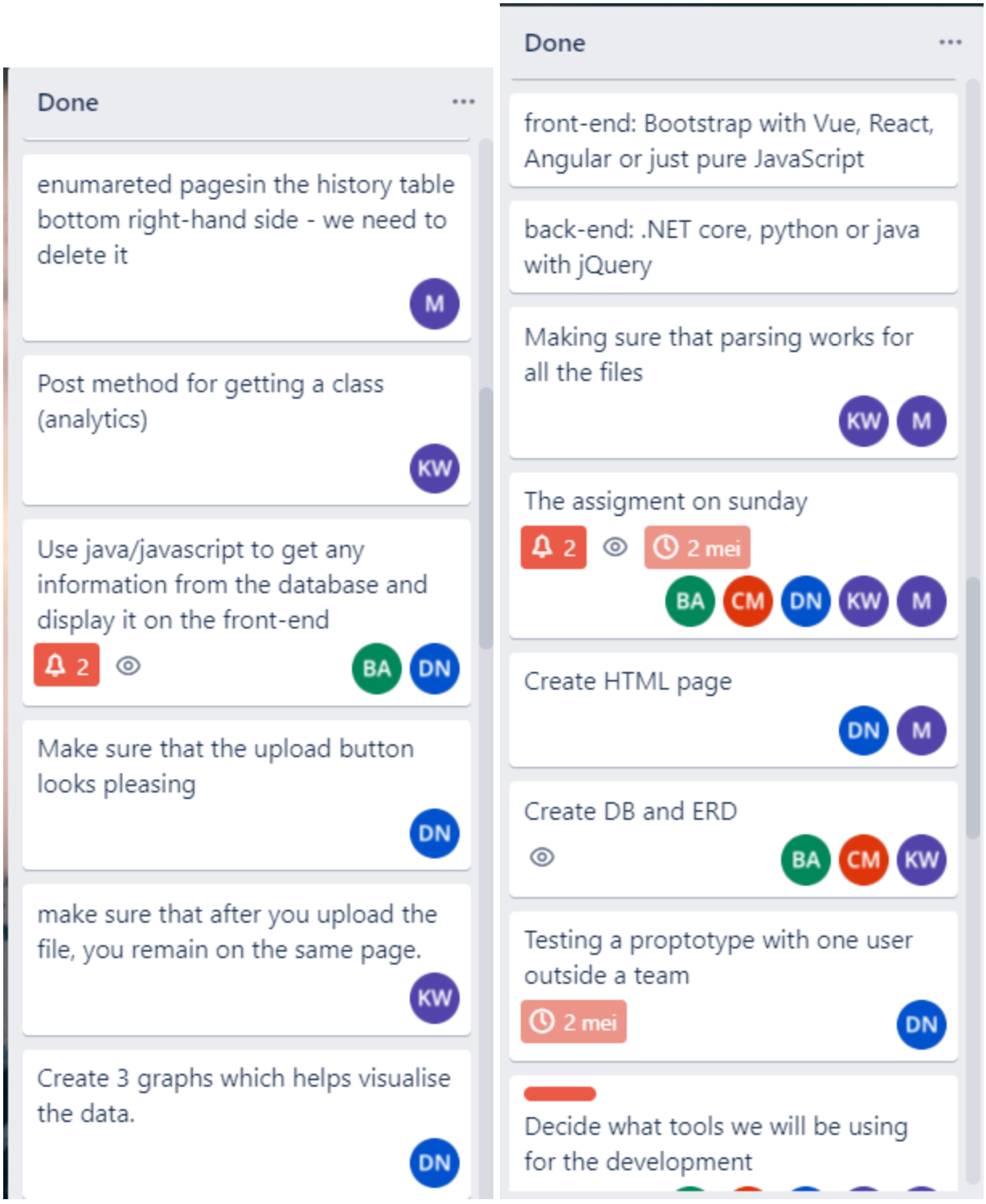
Enter Email Address...

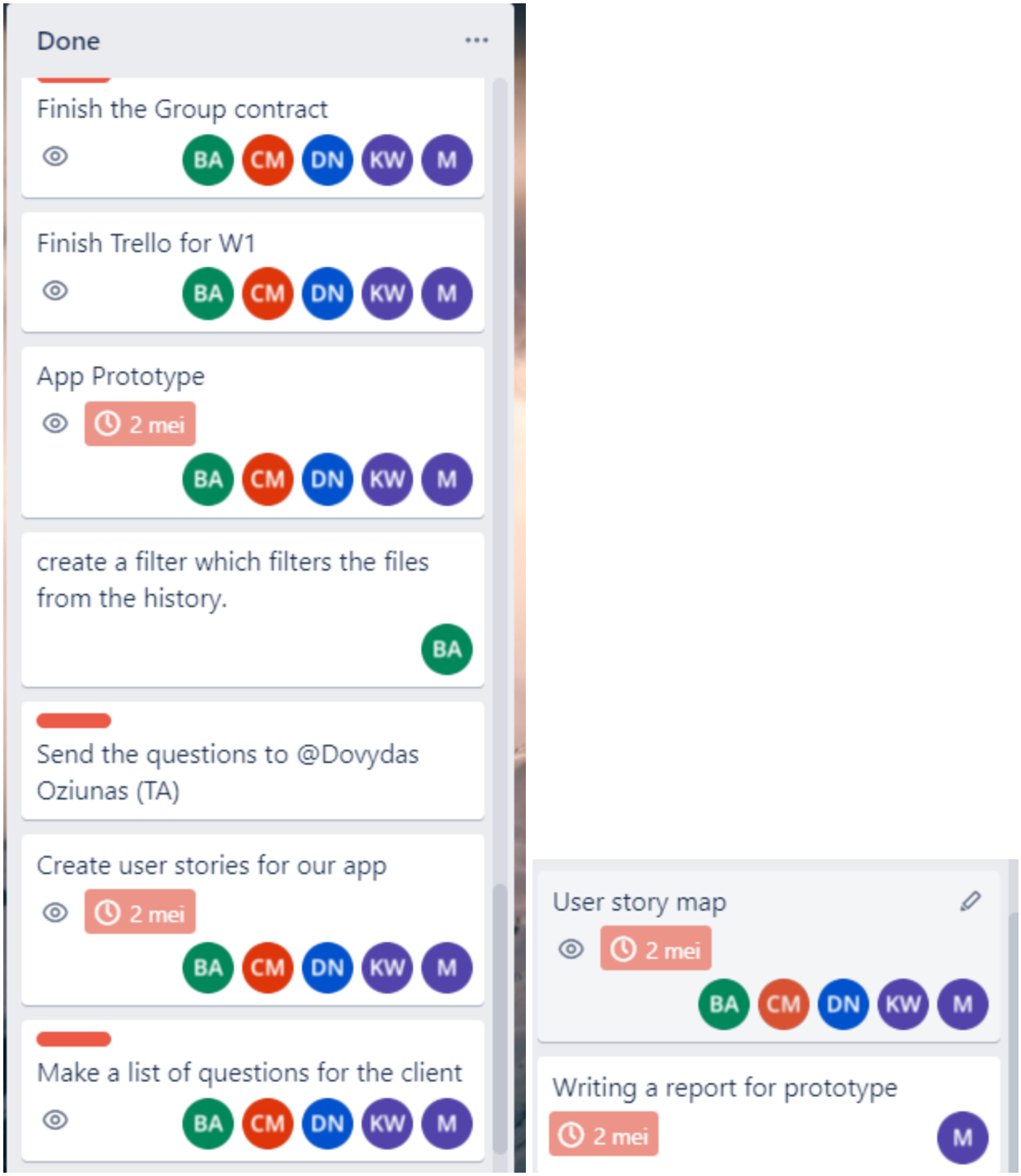
Password

Login

Below here we have screenshots of the trello board we made use of.




Done
enumareted pagesin the history table bottom right-hand side - we need to delete it
M
Post method for getting a class (analytics)
KW
Use java/javascript to get any information from the database and display it on the front-end
2
BA
DN
Make sure that the upload button looks pleasing
DN
make sure that after you upload the file, you remain on the same page.
KW
Create 3 graphs which helps visualise the data.
DN
Done
front-end: Bootstrap with Vue, React, Angular or just pure JavaScript
back-end: .NET core, python or java with jQuery
Making sure that parsing works for all the files
KW
M
The assigment on sunday
2
2 mei
BA
CM
DN
KW
M
Create HTML page
DN
M
Create DB and ERD
BA
CM
KW
Testing a proptotype with one user outside a team
2 mei
DN
Decide what tools we will be using for the development


Done
Finish the Group contract
BA CM DN KW M
Finish Trello for W1
BA CM DN KW M
App Prototype
2 mei
BA CM DN KW M
create a filter which filters the files from the history.
BA
Send the questions to @Dovydas Oziunas (TA)
Create user stories for our app
2 mei
BA CM DN KW M
Make a list of questions for the client
BA CM DN KW M
User story map
2 mei
BA CM DN KW M
Writing a report for prototype
2 mei
M