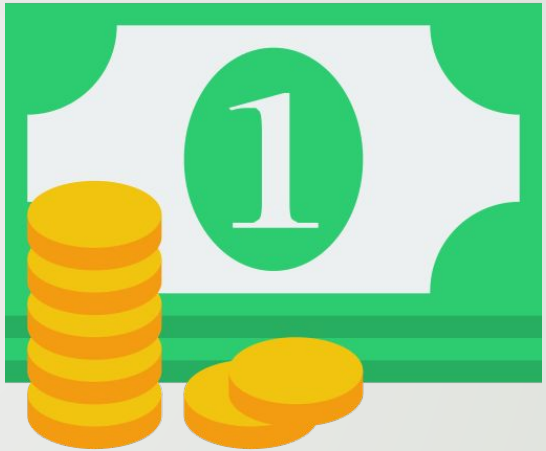
eBank web app

Development process



*All usernames, passwords or other information shown in this presentation is fictive and for illustrative purpose only

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

01

Implementing the web app initial structure

Skills used

N.O.

WHY WAS THIS STEP IMPORTANT

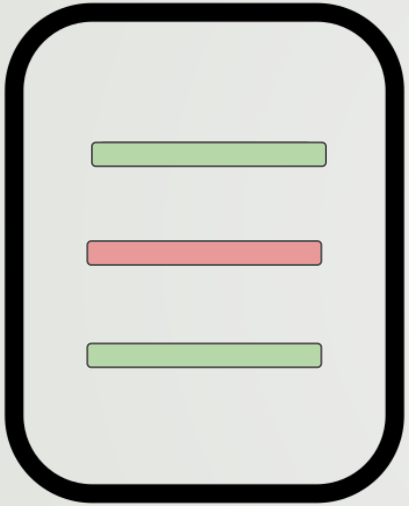
*HTML and CSS templates were made available / created by Jonas Schmedtmann.

This step was important to ensure that the initial structure of the web app was properly established (structure on which JavaScript would then be able to work with).

WHAT WAS THE GOAL

Ensuring that all static content for the web app is present in the HTML and CSS documents, with appropriate class names and ids to allow easy reference to it later using JavaScript.

02



Displaying banking transaction history

Skills used

Research
Problem solving
Code writing
Debugging

WHY WAS THIS STEP IMPORTANT

This step was important to allow users to see each transaction carried out in their account, and thus have a good overview of their financial activities.

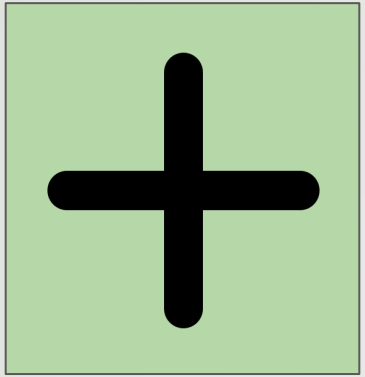
WHAT WAS THE GOAL

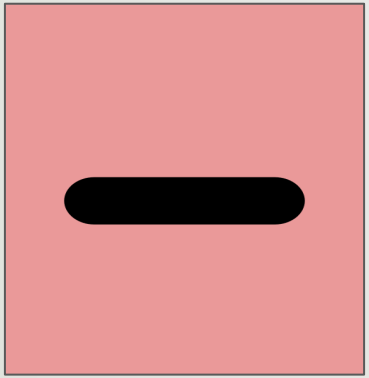
- Using mock user accounts and mock data, the objective was to display on the screen each operation carried out by a user.
    - To do this, a logic to identify whether each operation is a money inflow (deposit) or outflow (withdrawal) was created.
- Creating a template literal to dynamically select the right HTML class in the DOM, and the right CSS element associated with it to make the deposits appear in green, and the withdrawals in red.

FIRST VIEW OF THE **OPERATIONS HISTORY**

Deposit

Withdrawal





03

Displaying total balance

Skills used

Research
Problem solving
Code writing
Debugging

WHY WAS THIS STEP IMPORTANT

The total balance of a bank account is one of the main information displayed in any banking app, so it was important to calculate it and ensure that it is displayed correctly.

WHAT WAS THE GOAL

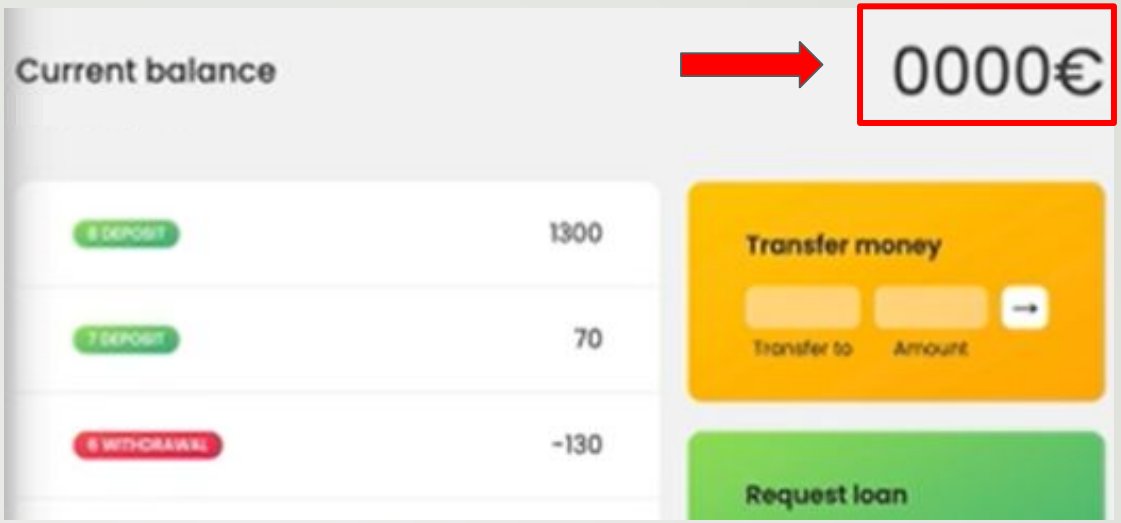
Calculating the balance of the users' bank account based on their banking history (deposits and withdrawals), and then display it.

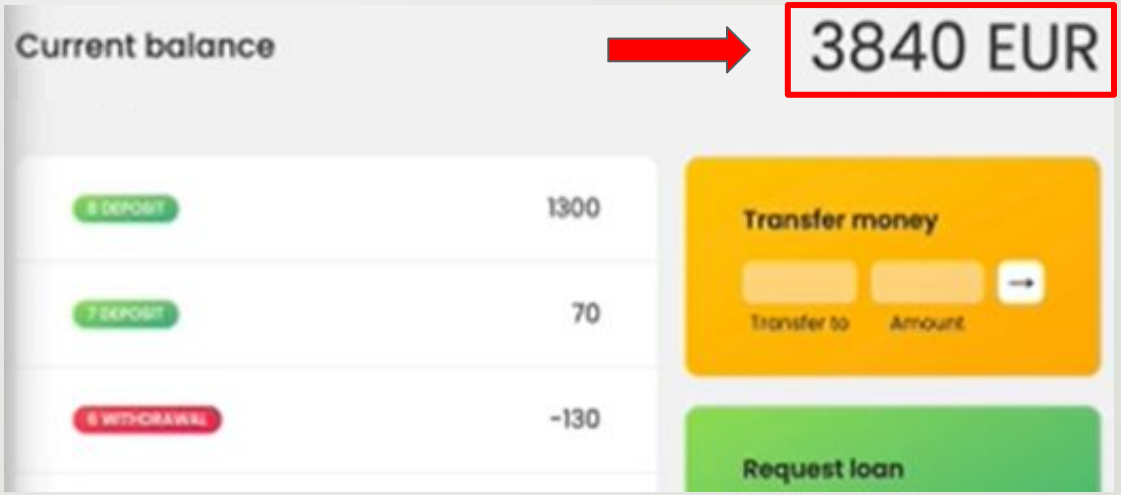
- To do so, the *reduce( )* method has been used to process each user banking operation array (array that contains all amounts of deposits and withdrawals for a user), in order to return a single balance value.
- For example, a user having 2 deposits of 100$ and 1 withdrawal of 25$ would have 175$ as a balance.

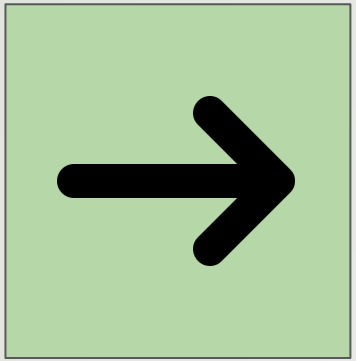
BALANCE

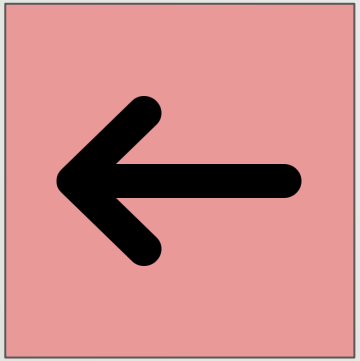
CALCULATED

AND DISPLAYED









04

Displaying total money in, total money out, and interest

Skills used

Research
Problem solving
Code writing
Debugging

WHY WAS THIS STEP IMPORTANT

This step was important to present in a simple and quick way a summary of users' money inflows and outflows.

WHAT WAS THE GOAL

From the users' transaction history, separating the deposits from the withdrawals (using the *filter( )* method), and then add respectively all the deposits together, and all the withdrawals together, to obtain a single value for each (by chainingthe *reduce( )* method to the *filter( )* method).

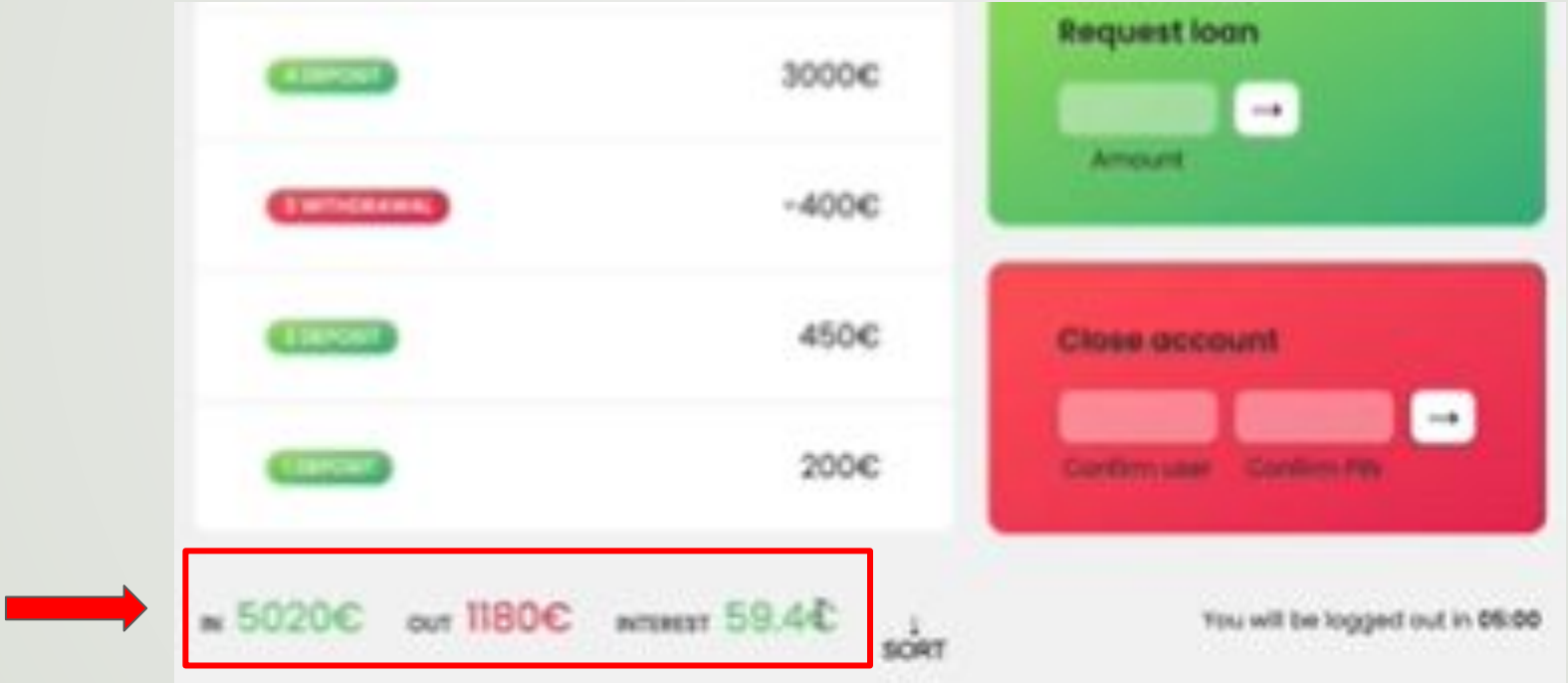
Interests were also calculated and displayed at this point.

FIRST VIEW

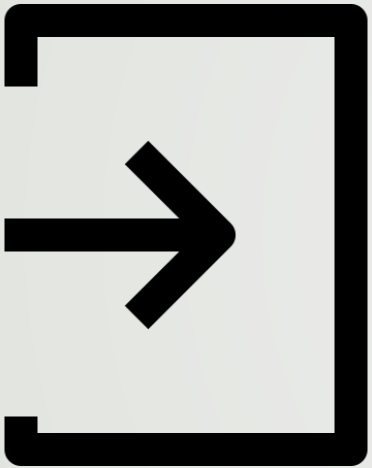
SUMMARY

UPDATED VIEW

SUMMARY



05



Implementing the login logic

Skills used

Research
Problem solving
Code writing
Debugging

WHY WAS THIS STEP IMPORTANT

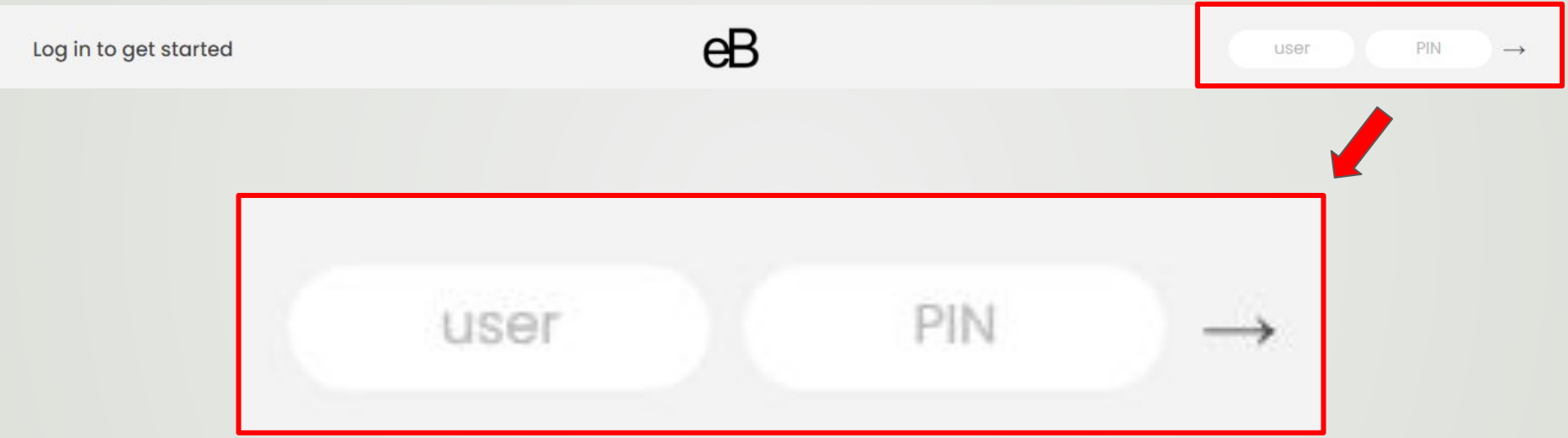
This step was important to ensure that each user has a personalized experience, and that the financial information displayed is unique and specific to each user.

WHAT WAS THE GOAL

Implementing a simple system allowing users to log in with a username and a PIN.

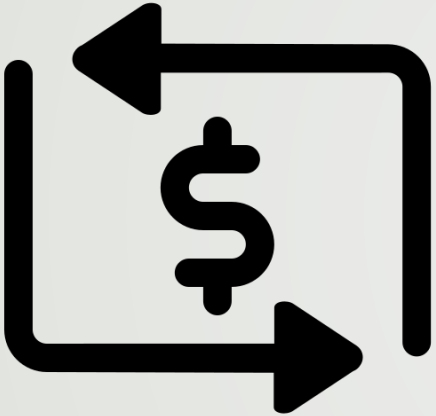
*Since the focus of this project was primarily on working with arrays, DOM manipulations and functions, the login system has not been developed as much as it could have been, on purpose. Since users information is also not stored in a database, but rather hard coded in the JS file for testing purpose, the login system remains quite basic, and several other features could have been added for a more complete and realistic login process.

LOGIN PAGE



SIMPLE SYSTEM LOOKING IF USERNAME AND PIN MATCH CORRESPONDING DATA IN THE USER OBJECT

06



Implementing money transfer between users

Skills used

Research
Problem solving
Code writing
Debugging

WHY WAS THIS STEP IMPORTANT

Transferring money between people is one of the most common operations. It was therefore important to allow users to do the same with eBank web app.

WHAT WAS THE GOAL

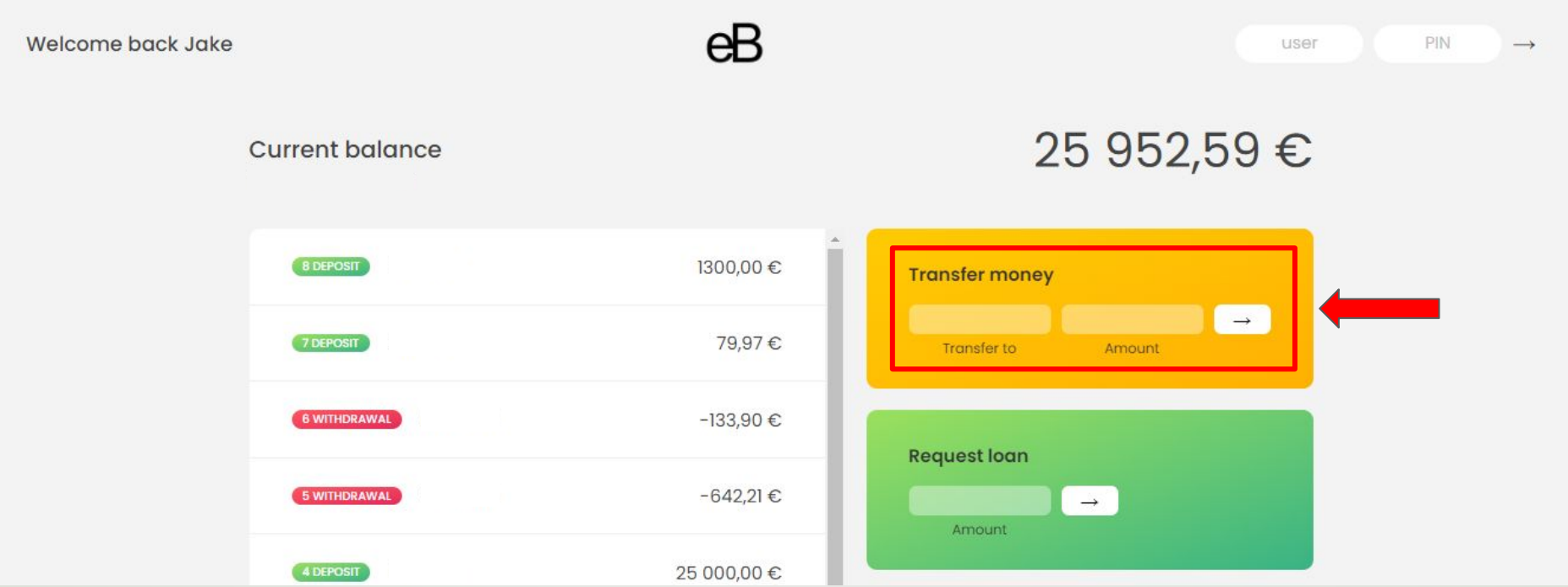
Ensuring that users can transfer (fictive) money to each other. To do this, two main parts were worked on:

- A part allowing users to identify the receiver account, and to transfer the money there correctly.
- A part allowing users to define the amount to be transferred.

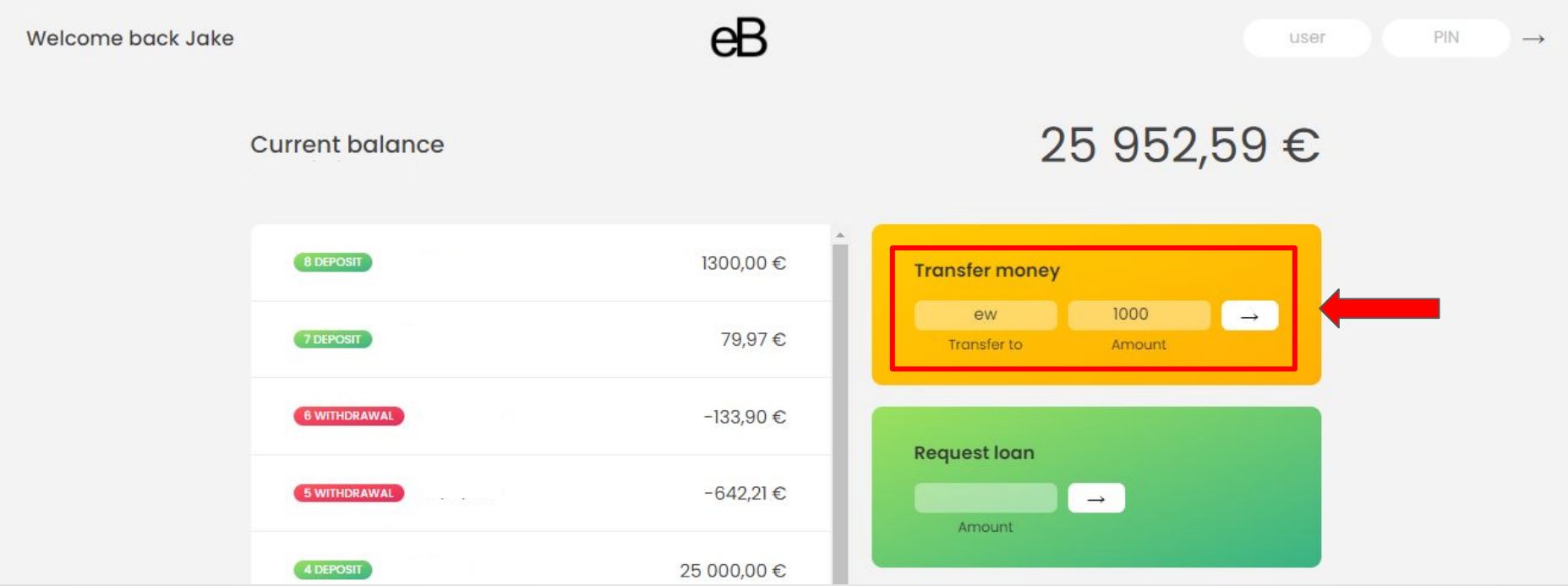
*In both parts, several checks and validations have been added to prevent users from doing incorrect operations (ex: sending more money than they have, or sending money to a receiver account that does not exist).

Updating in real time, after each transfer, users' UI (balance, operation history, and summary) for both the sender and the receiver.

MONEY TRANSFER BOX

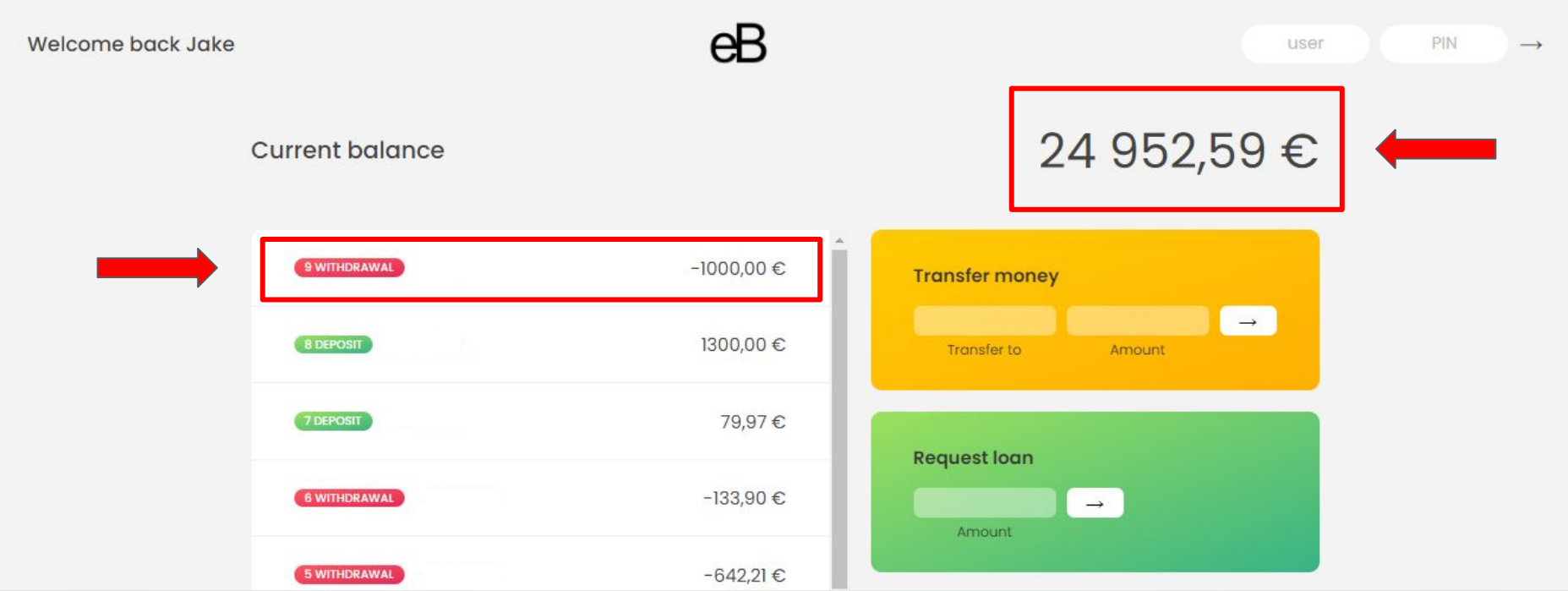


TRANSFERRING MONEY TO ANOTHER USER



TRANSFERRING MONEY TO ANOTHER USER

(*SAME UPDATE IN THE ***RECEIVER ACCOUNT*** TO SHOW THE DEPOSIT TRANSFER +1000)



07

Implementing the loan feature

Skills used

Research
Problem solving
Code writing
Debugging

WHY WAS THIS STEP IMPORTANT

Requesting a loan from a bank online is also a very common operation for people. It was therefore important to allow users to do the same with the eBank web app.

WHAT WAS THE GOAL

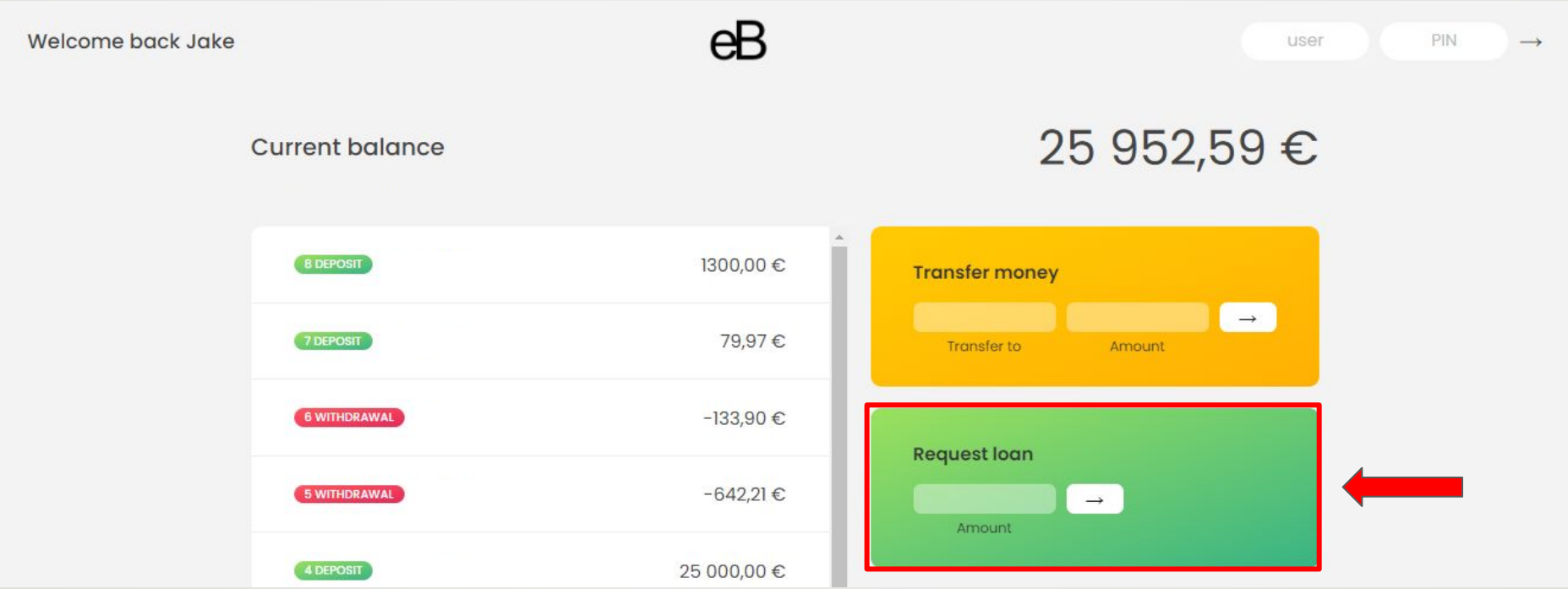
Ensuring that users can request a loan. To do this, two main parts were worked on:

- A part allowing users to define the requested loan amount.
  - Two checks/validations have been added to ensure that the users request an amount greater than 0, and that users have at least 1 deposit greater or equal to 10% of the requested loan.

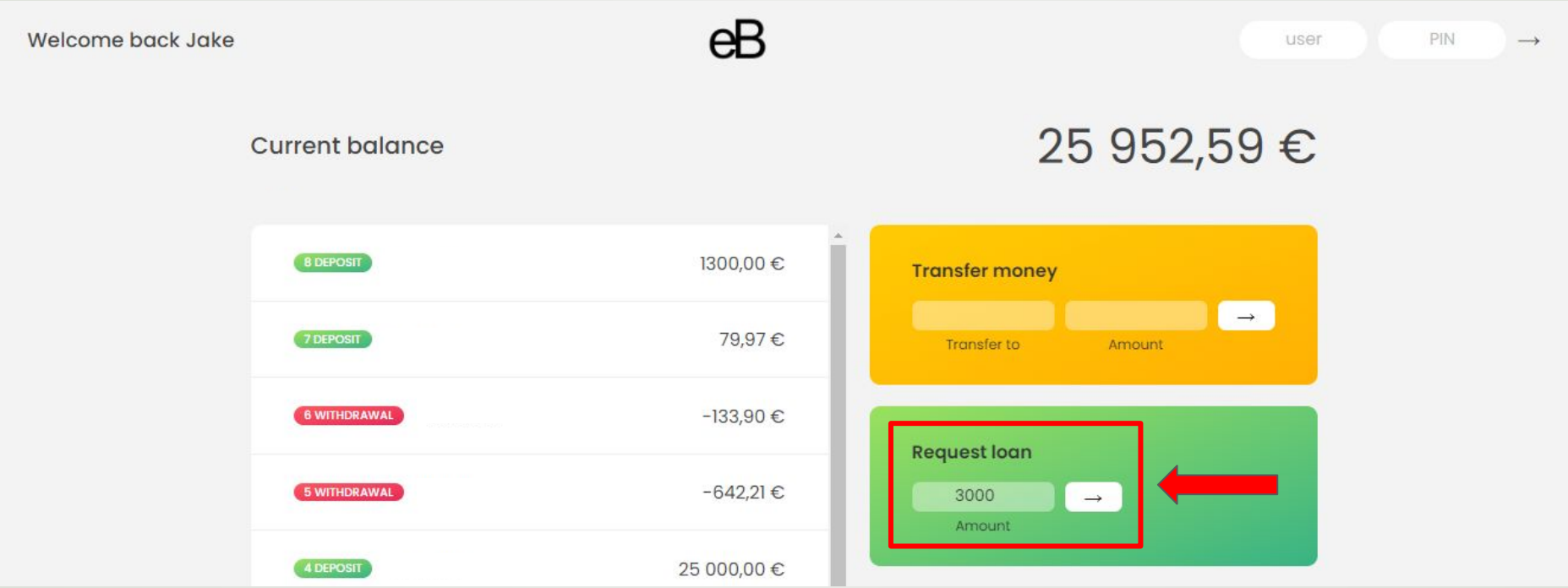
- A part in which a timer has been implemented, simulating the approval time required for the loan (delay of three seconds).

Updating users' UI automatically after loan approval with the new correct numbers (balance, operation history, and summary).

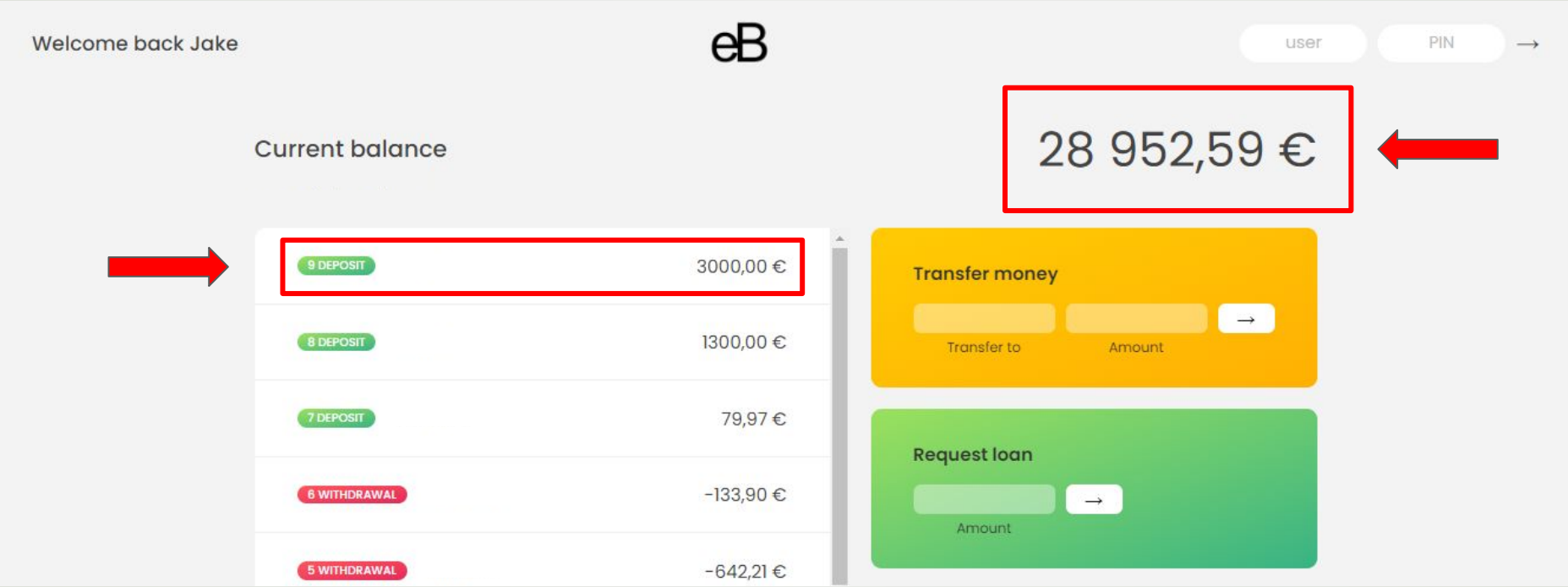
LOAN REQUEST BOX



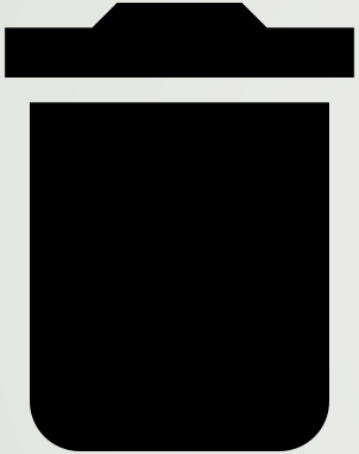
REQUESTING A LOAN



LOAN ACCEPTED AND NUMBERS UPADTED



08



Implementing the delete account box

Skills used

Code writing

WHY WAS THIS STEP IMPORTANT

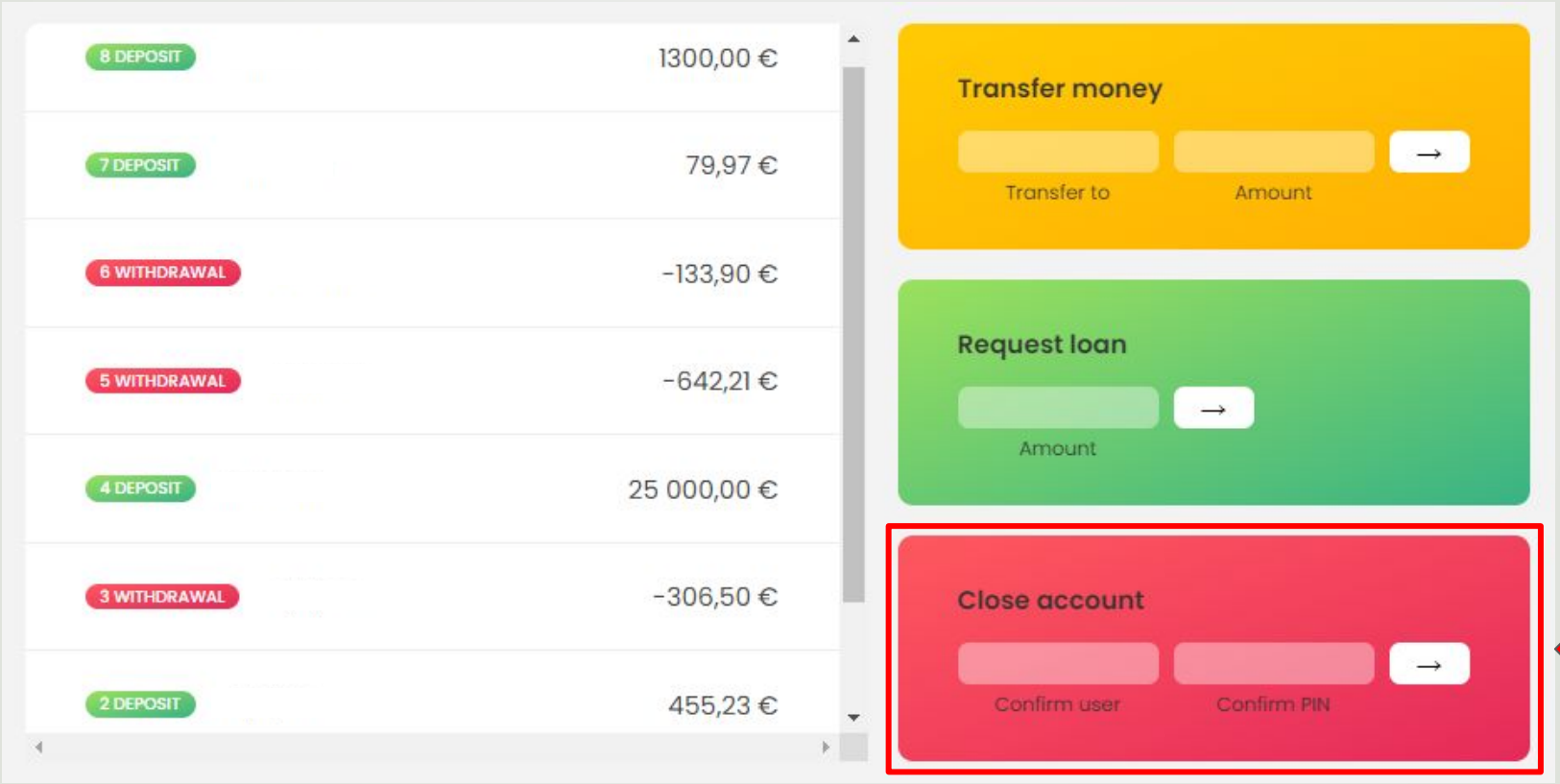
Allowing users to close their accounts is a basic operation that users generally expect to have. It was therefore important to allow users to do the same in the eBank web app.

WHAT WAS THE GOAL

Ensuring that users can delete their account by using their username and PIN.

- The *findIndex( )* and *splice( )* methods have been used for this.

CLOSE ACCOUNT BOX



09

Implementing the sorting option

Skills used

Research
Problem solving
Code writing
Debugging

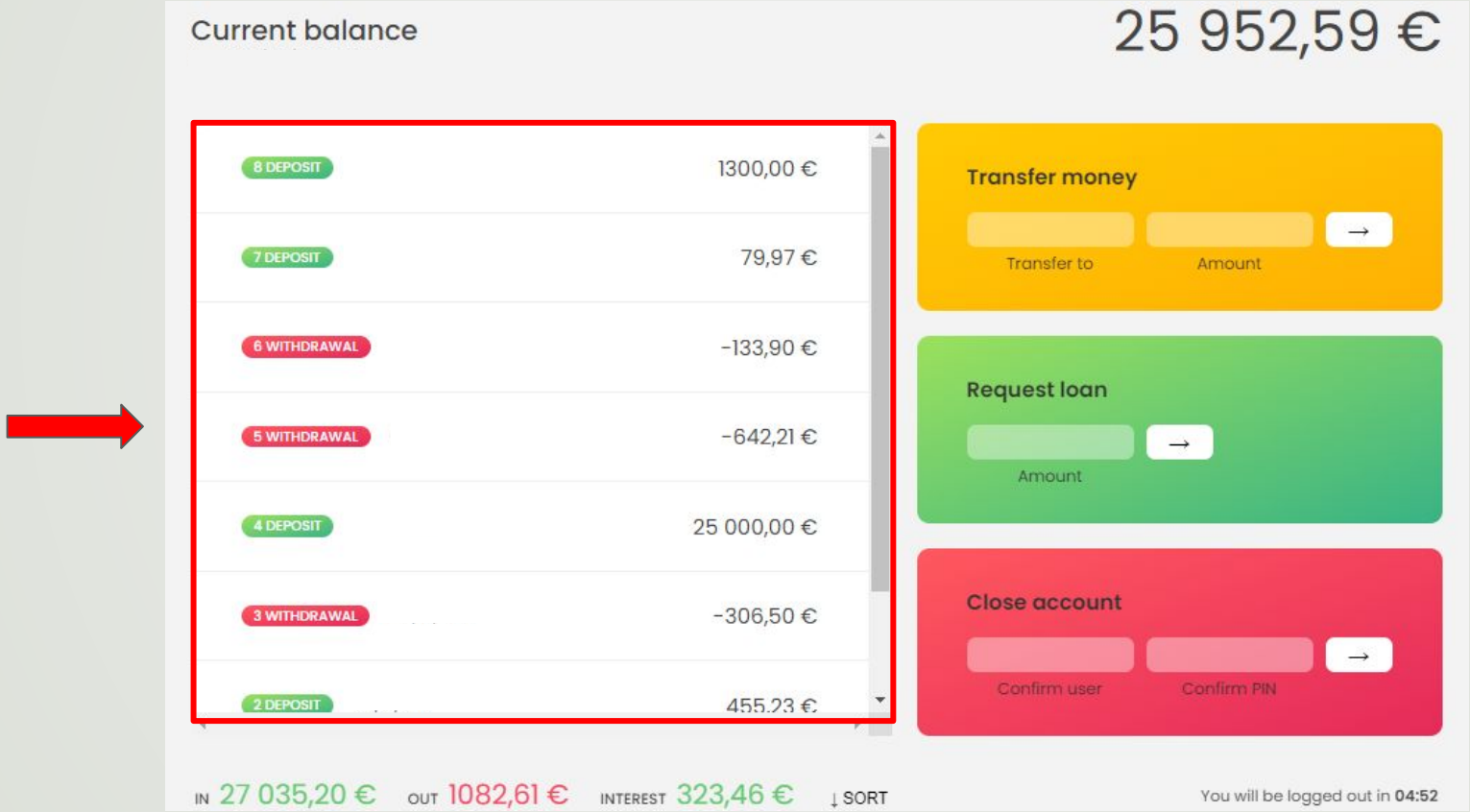
WHY WAS THIS STEP IMPORTANT

Sorting financial operations history in an orderly way helps to have a better overview, and improves user experience.

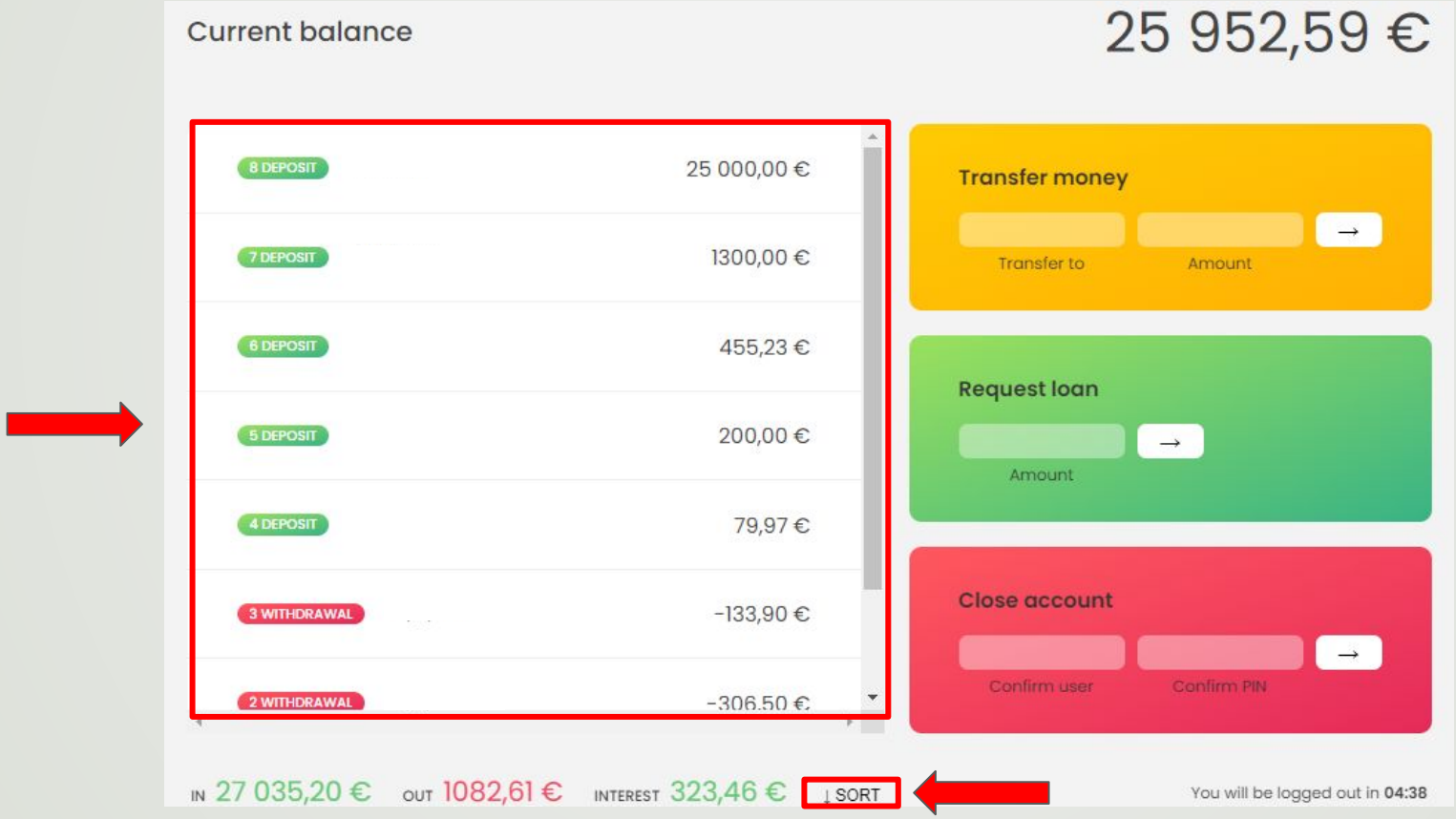
WHAT WAS THE GOAL

Implementing a feature (using the *sort( )* method) to allow users to display their financial transactions from the largest deposit to the largest withdrawal, and return to a chronological display when desired.

CHRONOLOGICAL DISPLAY



SORTED DISPLAY



10

€ ¥ $

Formatting dates and currencies

Skills used

Research
Problem solving
Code writing
Debugging

WHY WAS THIS STEP IMPORTANT

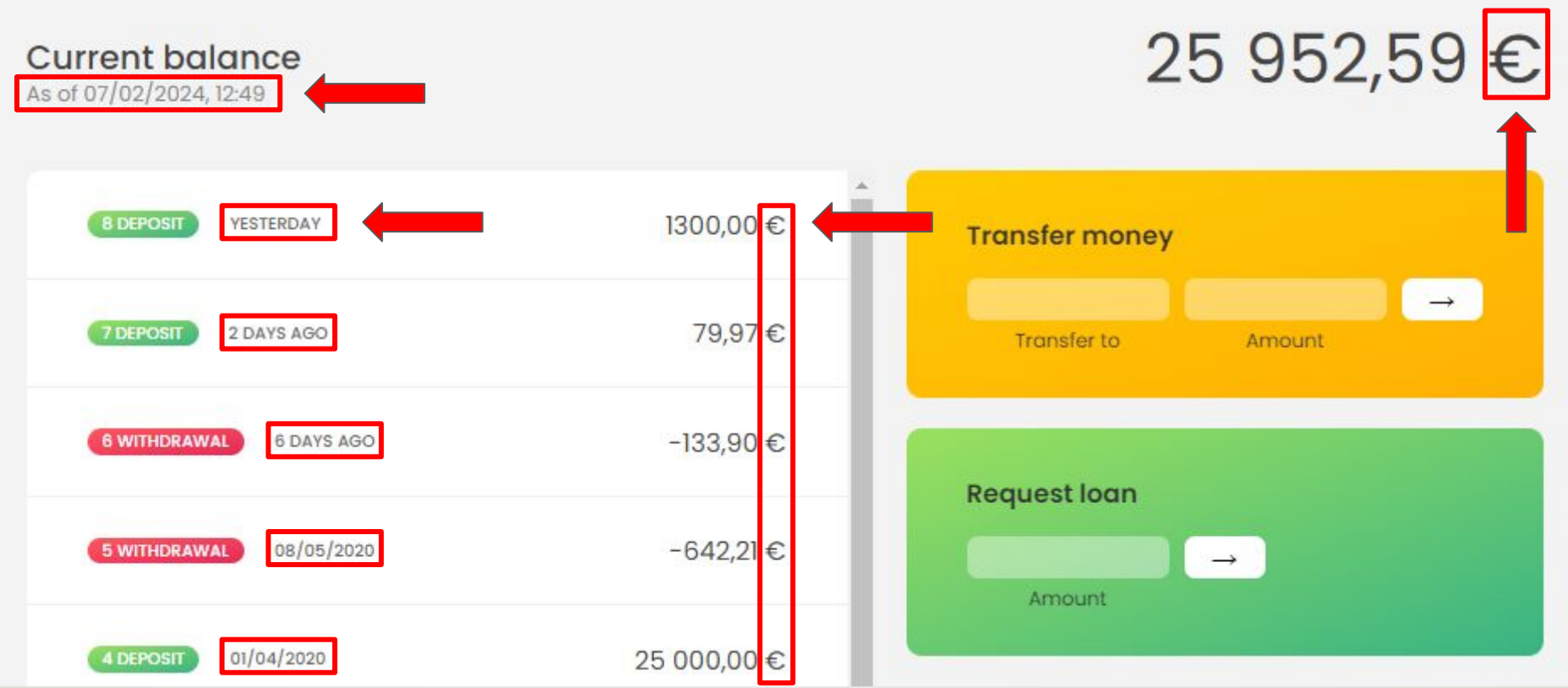
Considering that users may come from multiple regions around the world, it was important to adjust automatically the date format and the currency displayed accordingly, to avoid any possible confusions.

WHAT WAS THE GOAL

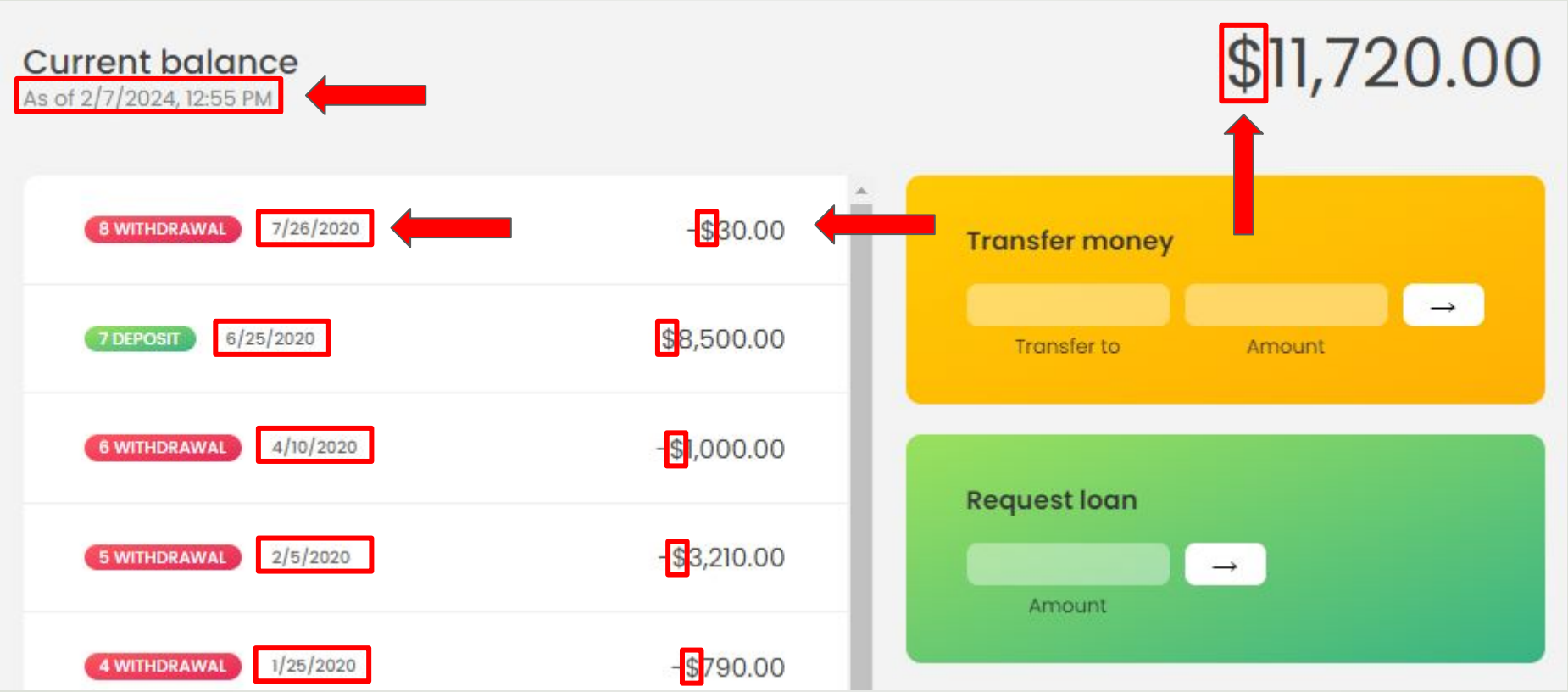
Displaying the correct date format and currency in each user's UI considering their location in the world / ensure users don’t have to configure themselves these parameters.

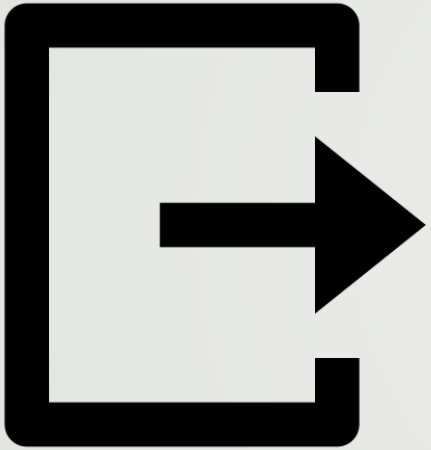
- To do so, the *Internationalization* API provided by JavaScript, and more precisely the *Intl.DateTimeFormat* and the *Intl.NumberFormat,* have been used.

DATE AND CURRENCY
EUROPE FORMAT



DATE AND CURRENCY
US FORMAT





Implementing security automatic logout

Skills used

Research
Problem solving
Code writing
Debugging

WHY WAS THIS STEP IMPORTANT

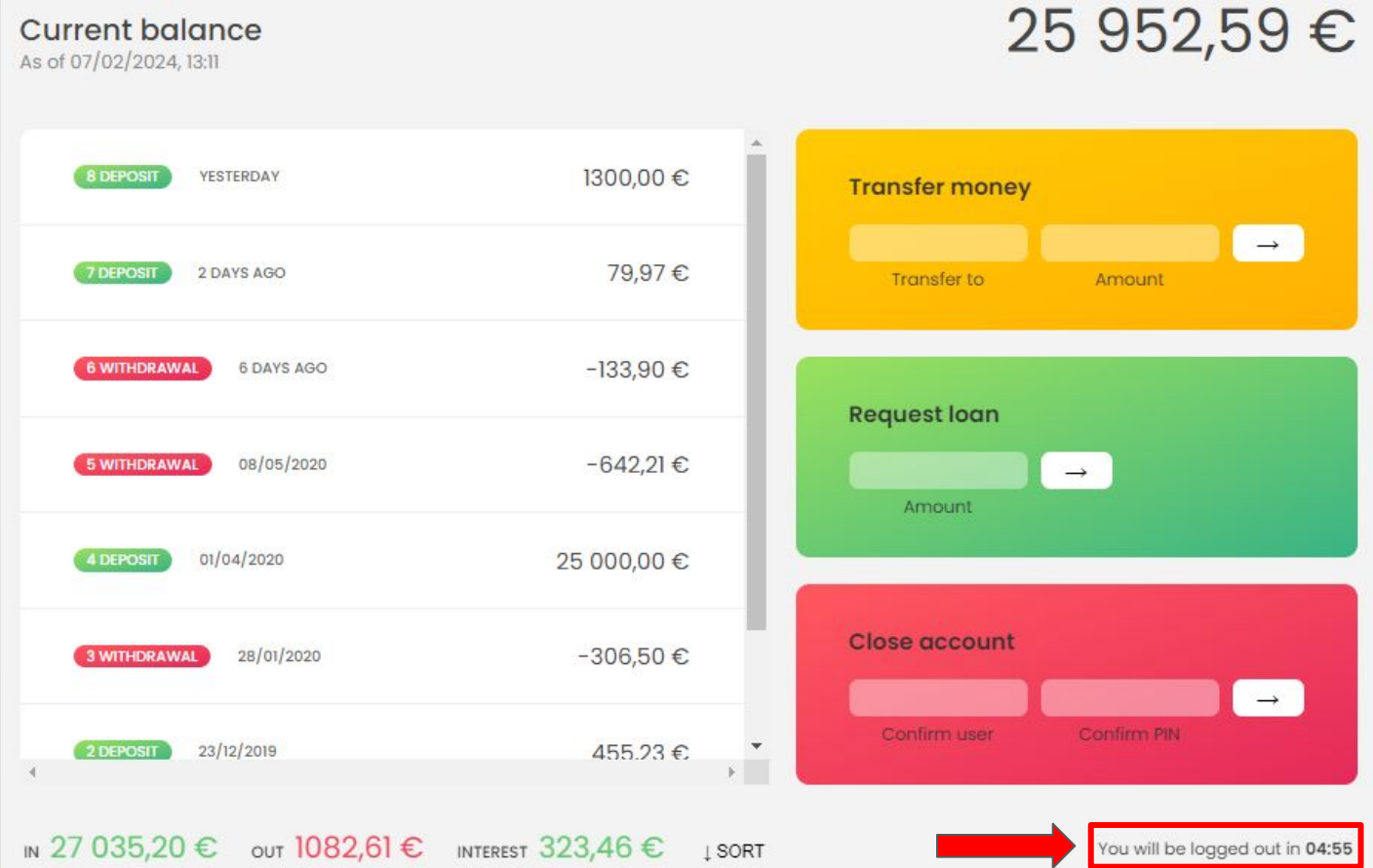
User security is always very important in a banking application context. Implementing this feature was therefore necessary.

WHAT WAS THE GOAL

Creating a timer to track each user activity. If a user does not perform any operations within 5 minutes, their account is automatically logged out for security reasons.

- For each operation carried out by users (money transfers, loan requests or sorting financial transactions), the timer is automatically set back to its starting time and re-begin to count down, thus avoiding unwanted logout.
- To do this, a function (*startLogOutTimer*) was created to display the timer on users' UI and track inactivity time.

INACTIVITY TIME CALCULATOR



Finalizing code refactoring and **running** final tests

Skills used

Code writing

WHY WAS THIS STEP IMPORTANT

When possible, it is good practice to refactor codes. When there are a lot of duplicate codes, and some functionalities need to be changed for example, multiple identical updates may be necessary across various locations and files, leading to a potentially lengthy and error-prone process. Code refactoring helps prevent this by making the code cleaner, more logical, and more concise.

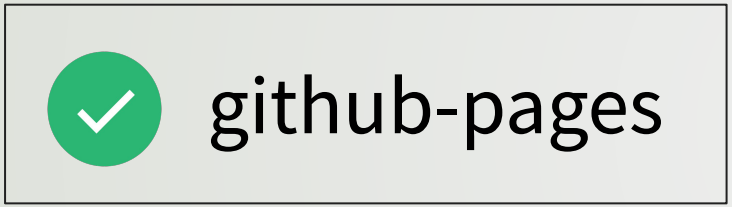
WHAT WAS THE GOAL

Replacing some previously duplicated lines of code to accomplish the same actions, but with fewer code lines, thus significantly reducing the length of the code. To do this, different functions were created.

Testing the web app with all possible scenarios that users might encounter to ensure everything works as expected.

Adding comments where necessary.

13



Deploying the web app online

Skills used

N.O.

WHY WAS THIS STEP IMPORTANT

This step was important to make the eBank web app publicly available by hosting it on GitHub Pages (gh-pages).

README.

md

Completing the README document

Skills used

Communication
Content writing

WHY WAS THIS STEP IMPORTANT

Ensure the project is well documented and easily accessible by anyone interested.

WHAT WAS THE GOAL

Updating and completing the README file located in the *eBank-web-app* Github repository. The goal was to ensure that all relevant information regarding this project is accessible under these three categories:

- Project description
- User interface
- Technical aspects

README SAMPLE - FULL VERSION ON GITHUB

README

eBAnk web app

Table of content

Project description

*All names and information are fictive and for demonstration purpose only.

This web app was created to simulate a banking application, allowing users to perform different real-life operations. Users can use eBank web app to carry out the following fictive operations:

- Log in
- View their total balance
- View their latest transactions and sort them
- View their total money inflows (deposits) and total money outflows (withdrawals)