****

**PROGmasters Fundraiser App**

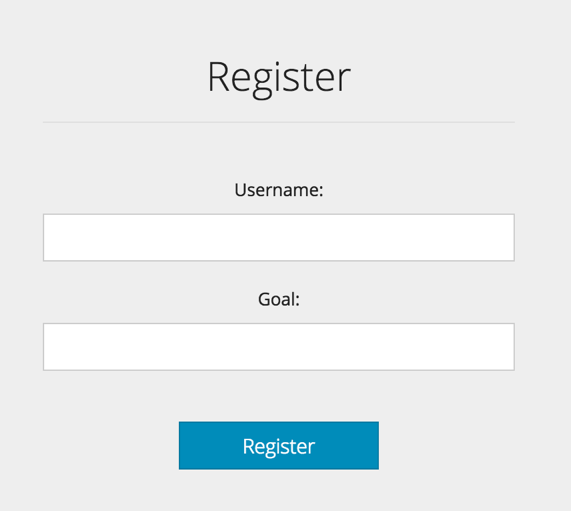
**(Spring REST + ReactJS)**

The task is to create a fundraising application that handles transfers from bank accounts to special caritative goals to support them.

After registration, a user is logged in automatically and has an account with a predefined amount (5000) of credits (or dollars).

The login is bound to an IP address, it means that from the same client/same IP address there’s no possibility to log in twice.

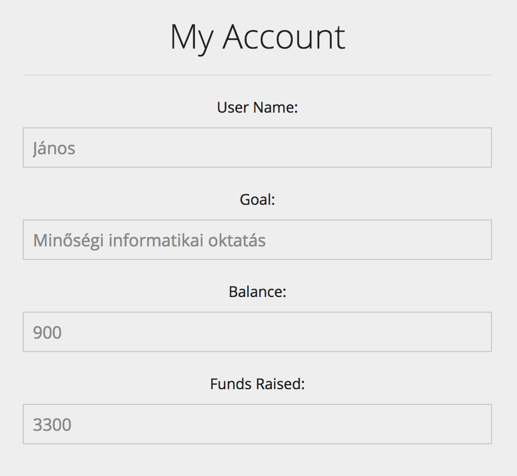
1. **Registration** screen



The first screen is the Registration screen, the user has to enter his/her user name and define a goal. This goal is the noble cause that the user wants to support.

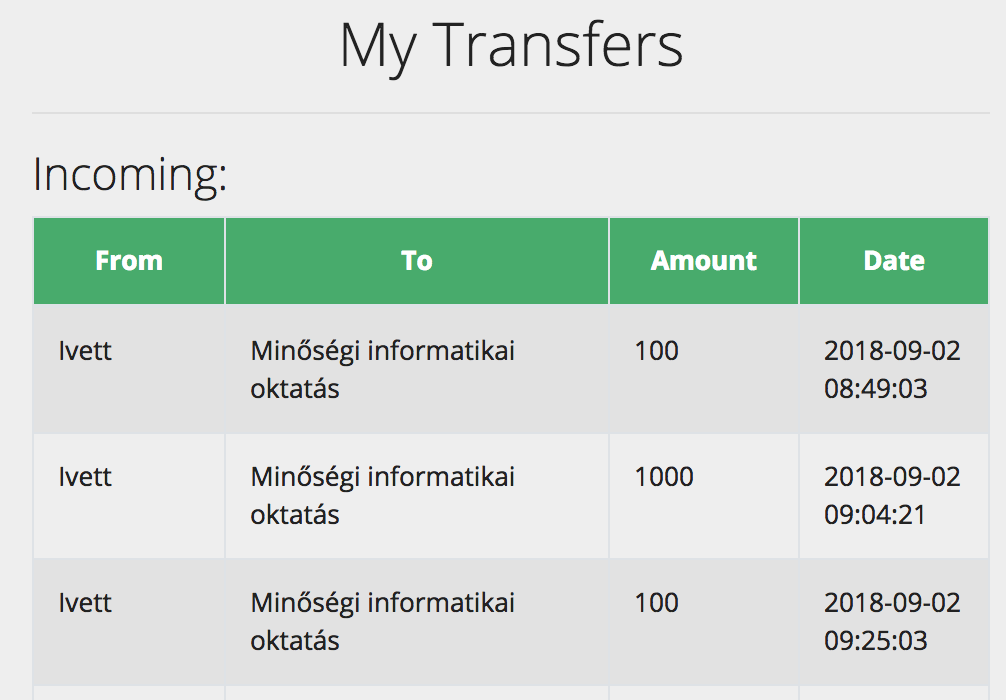
1. **My account** screen

The home screen (My account) shows the details of the user’s account: user name, goal, balance (amount the user can spend), funds (amount that is received from other accounts). It lists all the transfers this account was involved in (either as source or target).

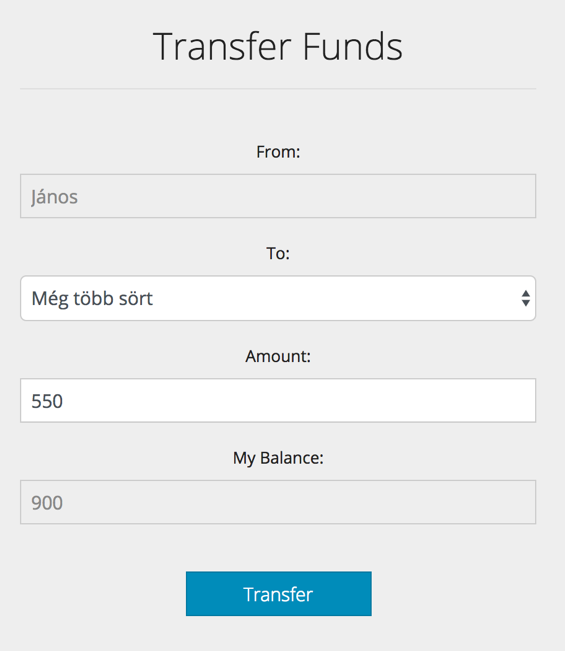


Every account has a subaccount for this noble cause that acts like a fund. In general, it means that incoming transfers to the user’s account will be added to this fund. In the same way, when the user transfers money to someone else’s account, the amount of the transfer will be added to the target fund.

This screen also displays the transfers related to this account:



1. **Transfer funds** screen



The “Transfer funds” page is for making the transfers from one account to another. The source account is fixed (Transfer from) to the actual user’s account. The target can be selected from a drop-down list, and it’s the list of the other logged in users. The logged in user cannot make a transfer to him/herself. The amount can be between 50 and 1000, and cannot go below 0. With hitting the “Transfer” button a new transfer will be saved in the system that contains the source and target accounts, the amount and a timestamp.

1. **Accounts** screen



The “Accounts” screen shows all the accounts sorted by funds in descending order, and all transfers that happened so far in the system sorted by creation date (latest first). This page refreshes itself every 10 seconds to show the changes.

The “Accounts” screen also shows all the transfers sorted by date in descending order.

