

# Case Study for Frontend Developers

## Goal

New customers of Savedo have access to an online bank account. The bank account allows the usual banking functionality:

- transfer money
- check balance
- add beneficiaries/payees

The website needs to work both on mobile (phone, tablet) as well as desktop browsers. Thus, it should adhere to UI/UX best practices. Assume that the UI requires multiple different inputs, some radio buttons, checkboxes and drop-downs with explanatory text elements, headlines etc.

Build a small single page web application using frontend technologies only. The choice of frameworks, preprocessors etc. is up to you – try to impress us and please explain your choices of technology very briefly in your README file.

Both normal web users using desktop browsers and mobile users (tablet, phone) shall be able to use this web app.

The user stories are as follows:

1) As a customer, I want to be able to view my balance so that I can see how much money I have in my account.

Given a customer opening the web app,

When they click on Balances

Then they see their account balance.

2) As a customer, I want to be able to create a payee so that I can send money to someone.

Given a customer opening the web app,

When they click on Create Payee

Then they see a form to create a new payee and can enter the Payee name, Bank and IBAN

3) As a customer, I want to be able to transfer money to a payee, so that I can pay my bills.

Given a customer opening the web app,

When they click on Make Payment,

Then they see a form that allows them to choose a Payee, enter an amount and a date.

Try to be creative in terms of UI/UX.

## Core requirements

- Feel free to use any framework but try to avoid the “Bootstrap”-look.
- Your implementation should be DRY
- Your implementation should be tested
- Keep your code minimal and clean
- Upload your source code to GitHub  
(please create an account if don't already have one, you'll need it eventually)
- Deploy your application to GitHub
- Make sure there is validation and that errors are displayed should validation fail
- Don't worry about the user logging in or saving data to a database.

## Data storage

Please use the local browser data storage engine so that the new payee is available as an option when making payments and the account balance goes down when making a payment.

## Goals/Priorities

- Functionality as described here (very high priority)
- Usability (very high priority)
- Error free application (high priority)
- Visual appeal, look & feel (high priority)
- Tests (high priority)

Please return your code by sending us the Github link within two days to us.  
Your concept will be reviewed both from a design and a mobile HCI perspective.

**Good luck!**