Project name: ICE CHAIN
Team members:
Romano Potechin,RomanoPotechin@gmail.com
Dan Kalinichenko, zaoaura@yandex.ru
Thomas Luong, luongt@me.com

DEVELOPER´S TASK PAPER

ICE CHAIN complete working scheme (beta)

	Call for contract =>	Approval =>	Loading =>	Transportation =>	Unloading
User	Salesman, Company A	Purchasing agent, Company B	Warehouse worker, Company A	Truck driver	Warehouse worker, Company B
What does user do	Creates a new smart-contrac; sets all the terms	Confirms all the terms or makes some corrections	Puts TDLs into the cargo boxes; hooks TDLs to the smart-contract by inputting their IDs	Connects to the TDLs via Bluetooth and checks	Comes to the cargo and launches the scanning mode in the mobile app
Application	Web	Web	Web/Mobile	Mobile	Mobile
Variables	Contract ID Buyer ID Type of cargo Temperature range Cargo value Depositor party Deposit rate (% of cargo value) Deadline	Contract ID Type of cargo Temperature range Cargo value Depositor party Deposit rate (% of cargo value) Deadline	TDLs' IDs	Temperature value (by minutes)	Temperature value (by minutes)
What happens	Purchasing agent, Company B gets notification and the task to confirm the terms	If any corrections have been made, the contract goes back to the Salesman, Company A If approved then Warehouse worker, Company A gets notification and the task to input IDs of TDLs assigned to the current smart-contract.	Smart-contract has been created in Blockchain. Deposit has been tied up Cargoholder has been set to Company A	TDLs send temperature log to the driver's smartphone via Bluetooth; mobile app makes a diagram and signalizes on deviations	The smartphone detects each TDL in the batch and gets the temperature logs. Mobile app consolidates the temperature data to the report and sends it to the server. Smart-contract searches for breaches and finally makes a decision on deposit transfer. Cargoholder has been changed from

					Company A to Company B

The system should consist of 5 separate modules, each for one stage of delivery process.

For the Hackathon, we have to build up the mobile demo app with simplified architecture.

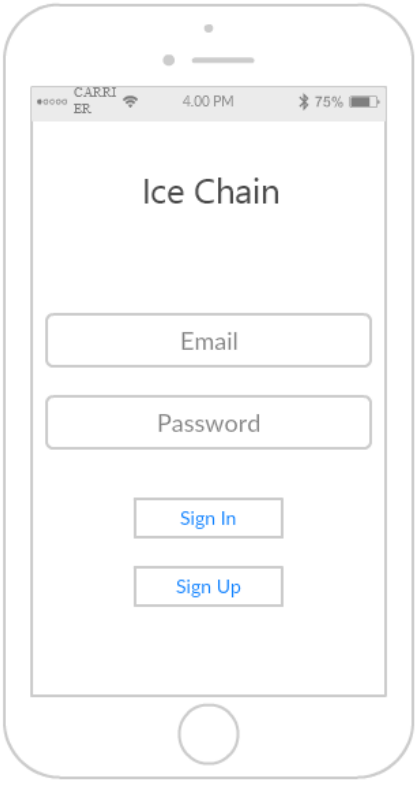
'**Loading**' and '**Transportation**' modules do not exist as the judges do not have access to TDLs (Temperature Data Loggers). All other functions are realized in the mobile demo app.

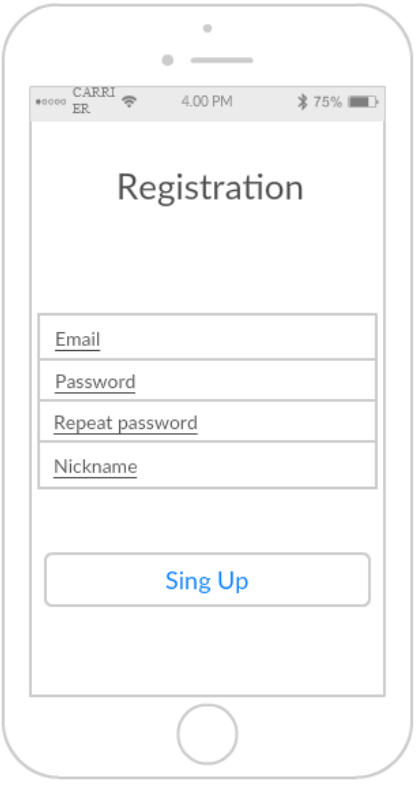
1. **Authentication screen.**

User can 'Sign In' by using email and password or 'Sign Up'. If user presses 'Sign Up' button, he goes to the registration screen with the following fields to fill:

- Email
- Password
- Repeat password
- Nickname

After registration a new user gets an account with QTUM testnet coins.





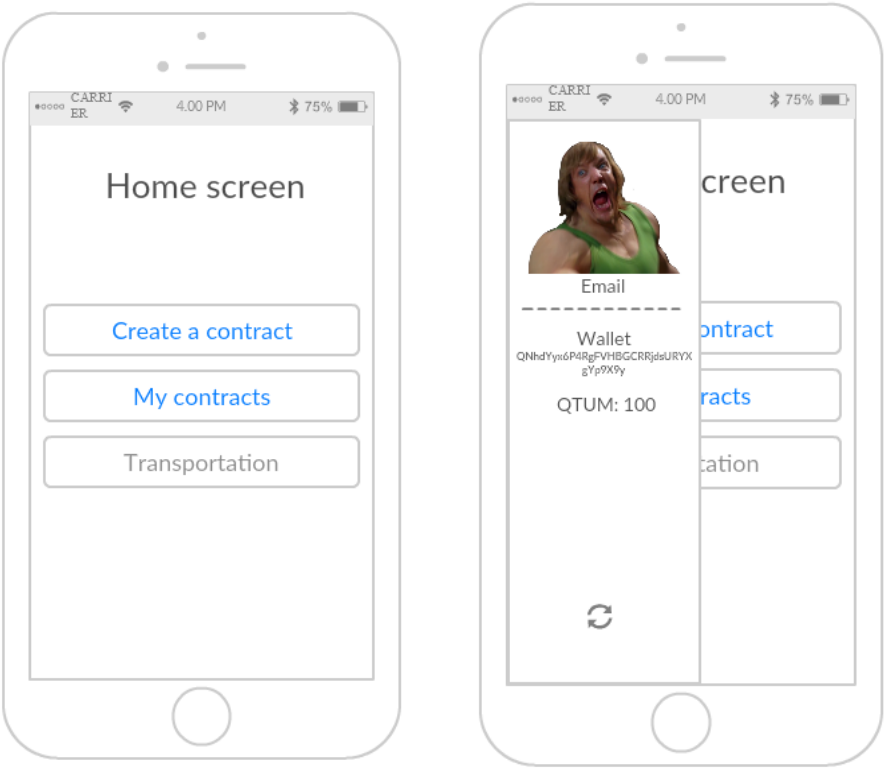
2. **Homescreen**

A registered user can choose the following options:

- Create a contract

- My contracts
- Transportation (UNWORKING BUTTON FOR DEMO).

In the left side of the screen he may see his login and check his current balance.

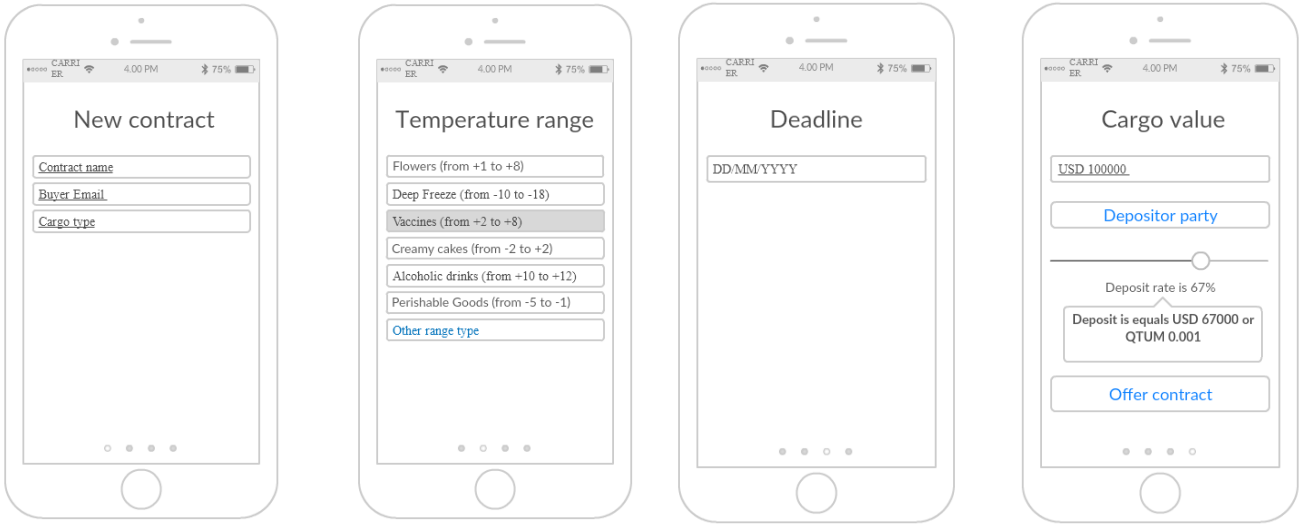


3. Call for contract

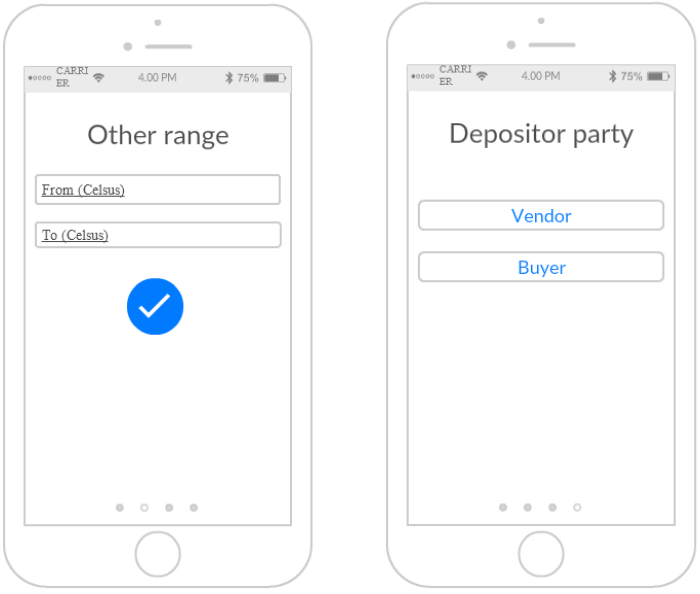
In demo version of the app for this contest user can create a smart contract. He creates it as Vendor. Vendor should fill in the following fields:

- Contract ID - the name of smart contract
- Buyer ID - Vendor can choose one of the registered users (Vendor can choose himself to make the process of demo presentation more easier)
- Cargo type - a short description of a cargo (for example, bananas, ice-cream or vaccines names)
- Temperature range - vendor chooses one type of range from the following list:
  - Flowers (from +1 to +8 ℃)
  - Deep Freeze (from -10 to -18 ℃)
  - Vaccines (from +2 to +8 ℃)
  - Creamy cakes (from -2 to +2 ℃)
  - Alcoholic drinks (from +10 to +12 ℃)
  - Perishable Goods (from -5 to -1 ℃)
  - Other range type (vendor should be able to make a new range type for his cargo)
- Cargo value - cargo’s fiat value

- Depositor party - Vendor can choose from who´s account deposit is made: ‘Vendor’ or ‘Buyer’
- Rate of deposit - Vendor can choose deposit amount: it should be % of the cargo cost (from 0% to 100%)
- Deadline - deadline of the cargo delivery (date and time), this field can be left empty



Also when user presses ‘Other range type’ and ‘Depositor party’ the following screens appear:



When setting the rate of deposit, some calculations should be made. The program makes notification:
‘Deposit rate is X of cargo value, which equals Y or Z’
where X is the rate of deposit chosen by user (%),
Y equals Cargo_value * X (United States Dollars),
Z equals Y / Qtum Testnet Coins’ rate (Qtum).

Qtum Testnet Coins' rate equals current official USD/Qtum rate or some imitative rate made up by developer.

For example: 'Deposit rate is 20% of cargo value, which equals USD 20000 or QTUM 371,65'

After filling all the fields user (Vendor) presses the button 'Offer contract'. Other user (Buyer) gets a notification '*the vendor %name%* suggested a contract'.

If the first user (the Vendor) chose himself as a buyer, he gets a notification too.

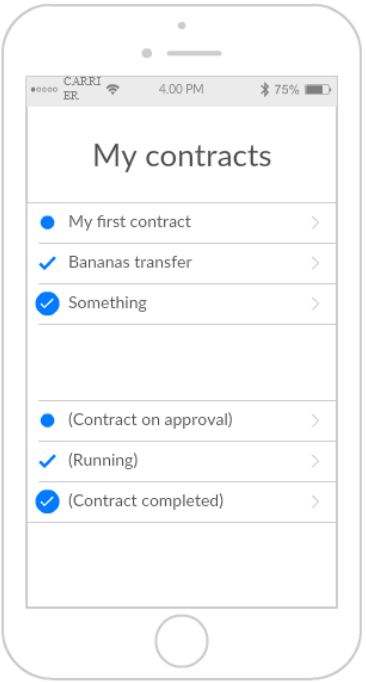
After getting this notification, user can open the screen or window with all conditions of the contract. There are two buttons at the bottom of the screen: 'Approve' and 'Edit'.

After clicking 'Edit', user can change any field, and then press 'OK' button. After that the contract returns to the previous user, who can also approve or edit it. This continues indefinitely until someoпщте clicks 'Approve'.

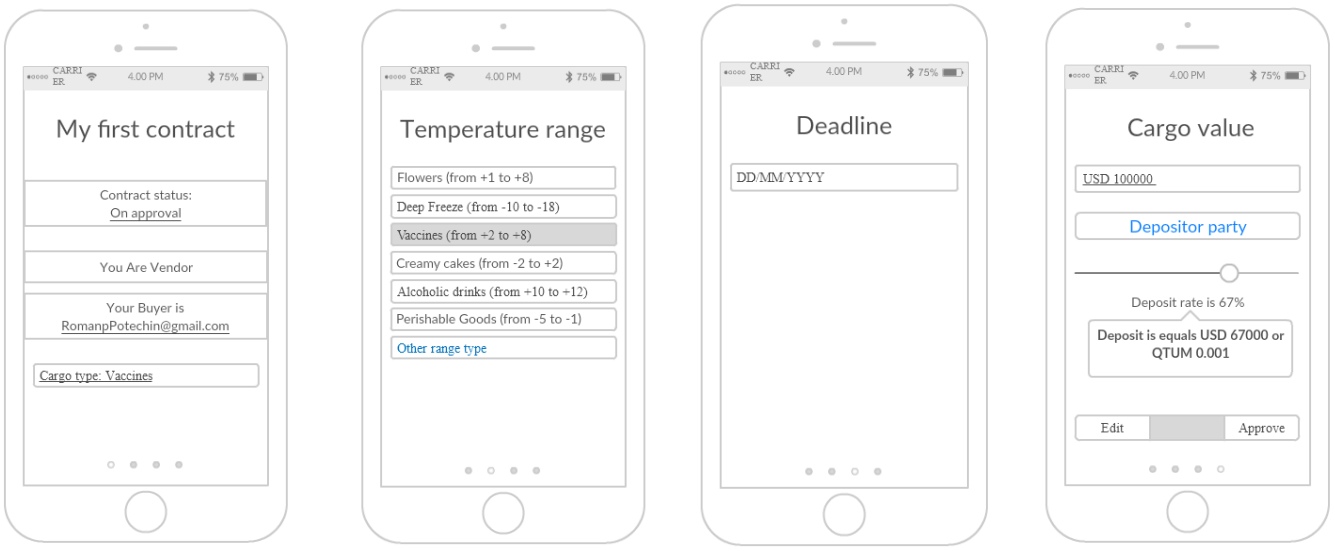
After clicking 'Approve', the deposit funds become frozen in the required amount, and a smart contract is created.

4. My contracts

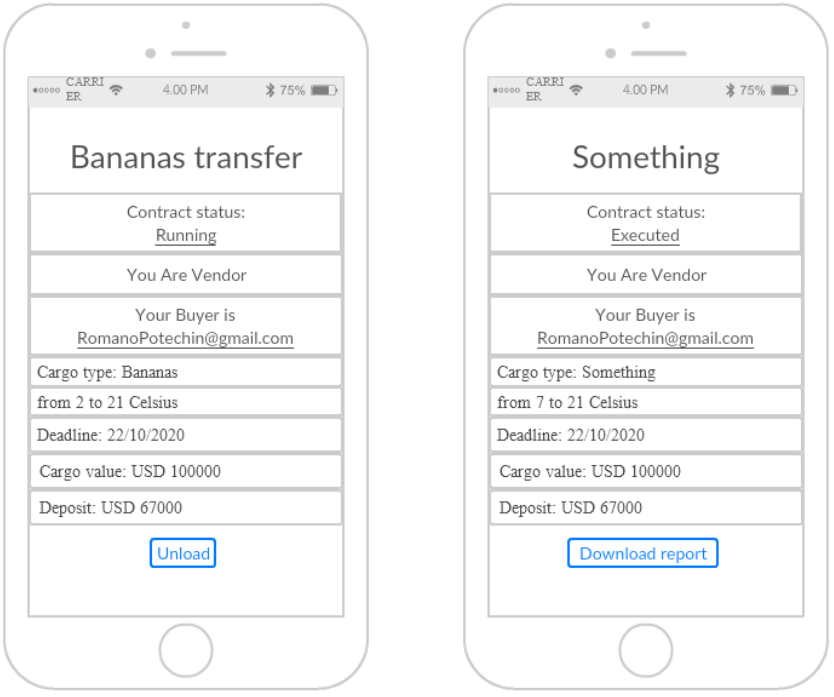
There is a list of all contracts in which user participates as a Vendor or as a Buyer. On this screen you see list of your contracts. There are some differences between 'on approval', 'in process' and 'completed' contracts.



Editing Smart Contract (on approval):



Viewing Smart Contract (running and executed):



If user presses on any contract, he gets the whole information (as in 'Call for contract'), plus

- Contract status - 'On approval','Running' or 'Executed'

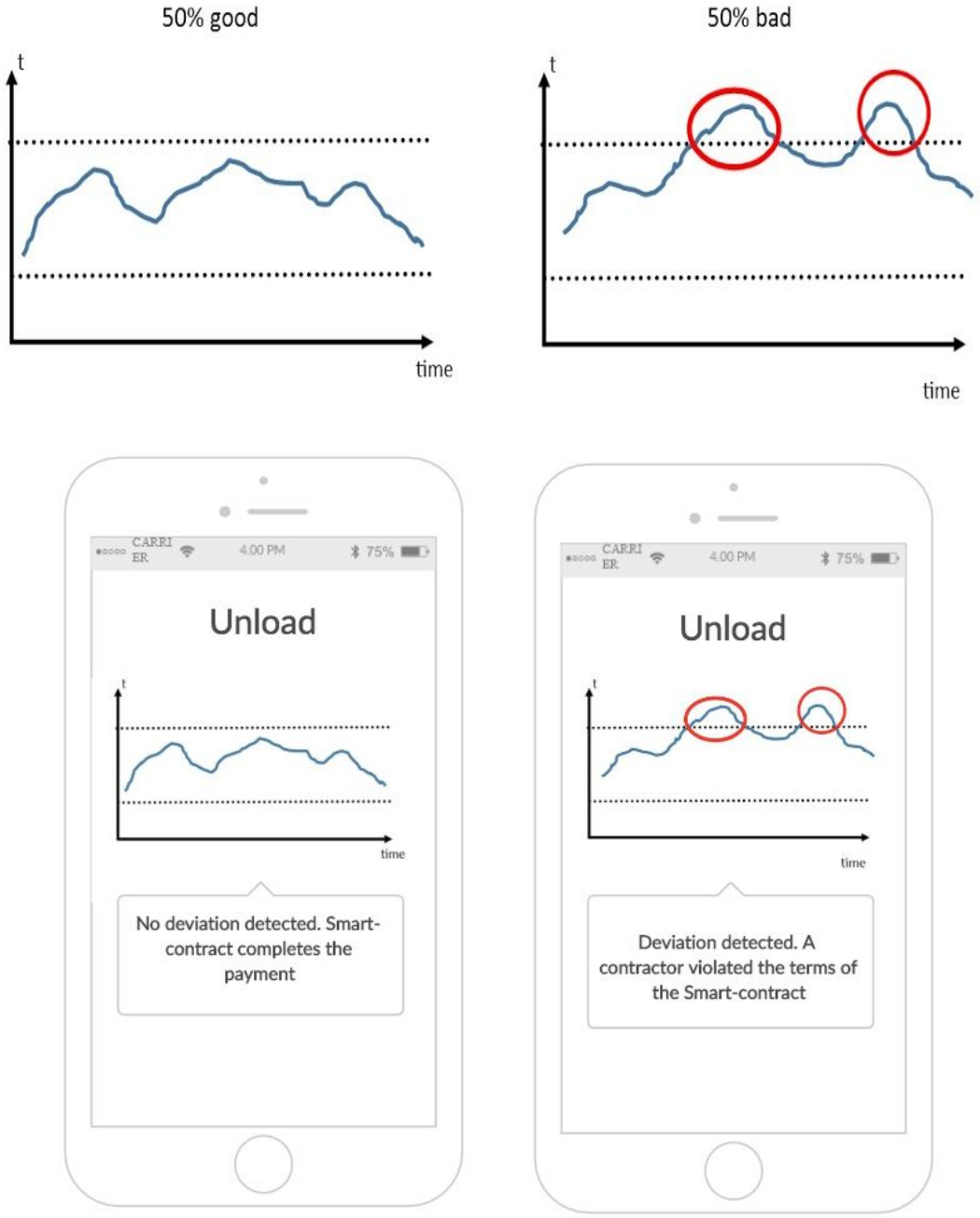
If user is a Buyer, after pressing on contract the additional button 'Unload" should appear.

If a contract is Executed, there is a button 'Download Report'. It provides downloading a short PDF-report with the temperature chart.

5. Unload

If user is a Buyer, after pressing on contract the additional button ‘Unload’ should appear.

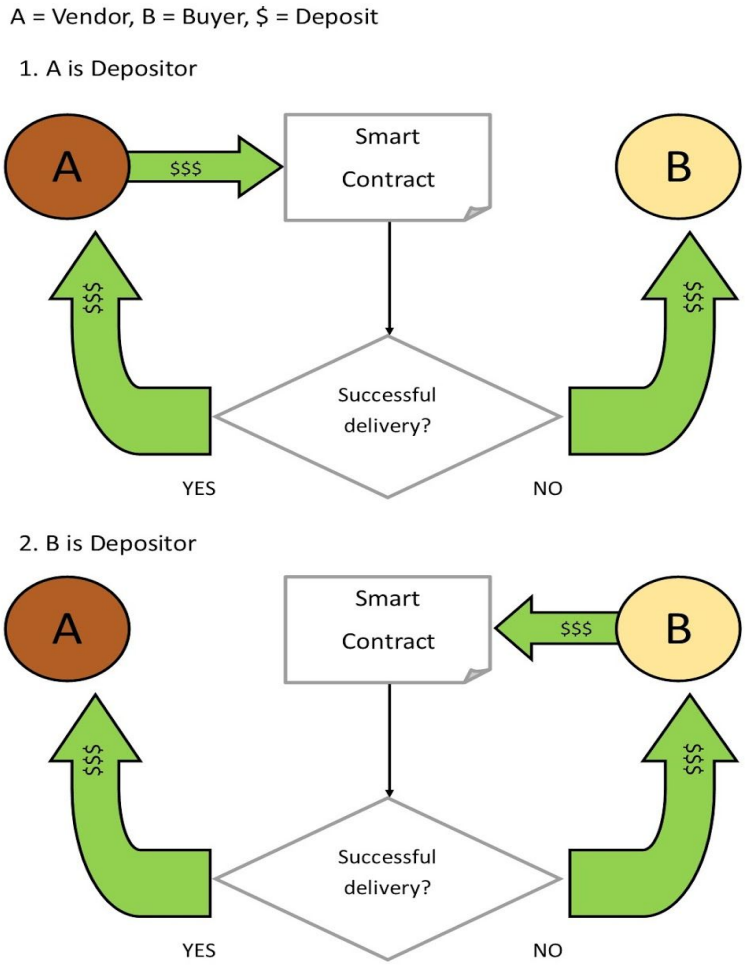
The simulating process of TDL connection begins (as if all the temperature data was uploaded to the app). A random temperature report with a time-temperature chart is created. The report has 50% chance to be ‘good’ (with no deviations) or ‘bad’ (with some deviations from the preset temperature range).



If there is no any deviation, then message ‘No deviation detected. Smart-contract completes the payment’ pops up.

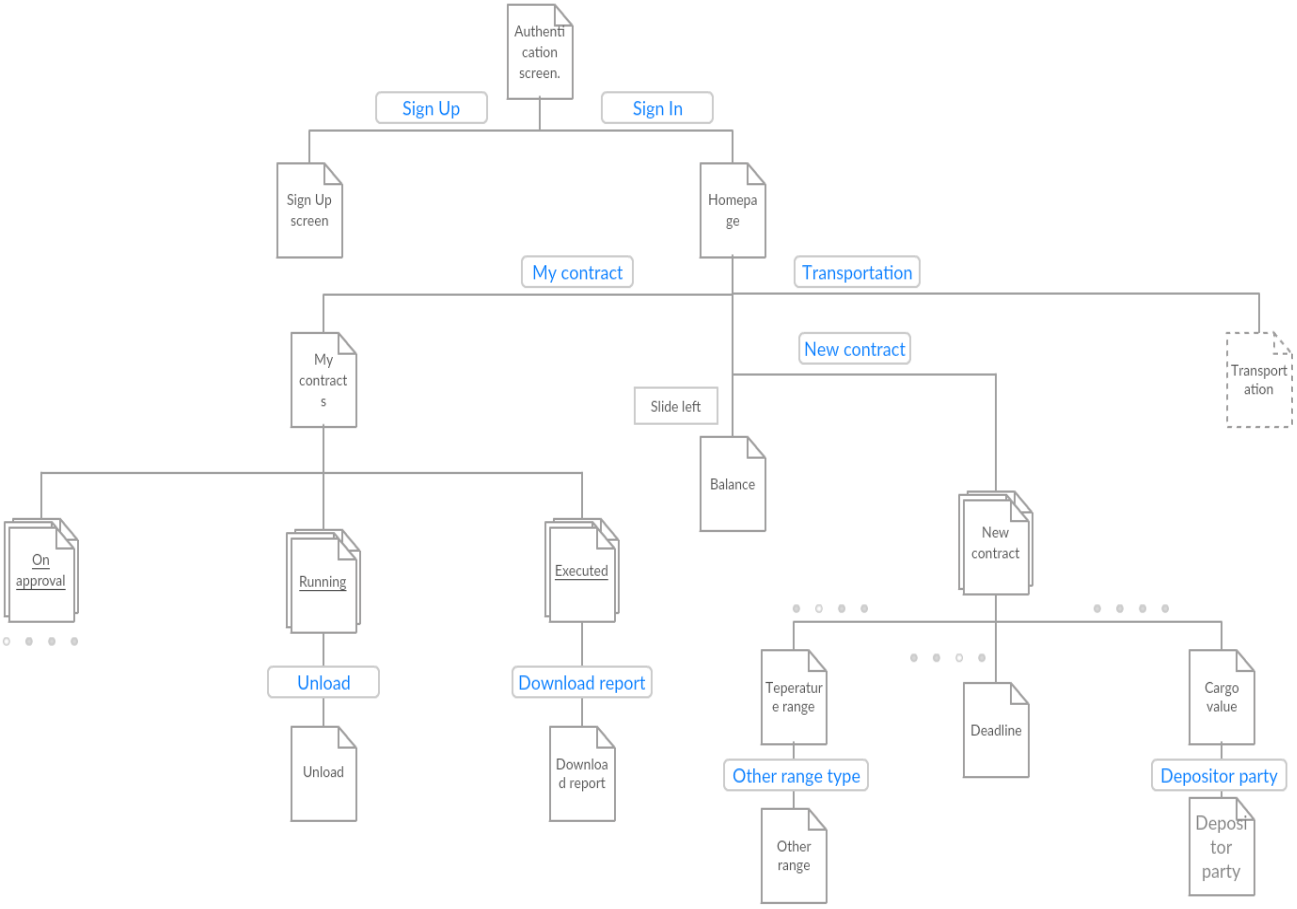
If there is a deviation of temperature conditions, then message ‘Deviation detected. A contractor violated the terms of the Smart-contract.’ pops up. User may see some information on the deviation (date, time, temperature).

In both case the deposit goes to the vendor or the buyer in accordance to the scheme:



Important note: If the contract wasn’t uploaded **before deadline**, it becomes unsuccessful, and deposit goes to the buyer.

The block-scheme of the screens:


Authenti cation screen.
Sign Up
Sign In
Sign Up screen
Homepa ge
My contract
Transportation
My contract s
New contract
Transport ation
Slide left
Balance
On approval
Running
Executed
New contract
Unload
Download report
Teperatur e range
Deadline
Cargo value
Unload
Downloa d report
Other range type
Depositor party
Other range
Deposi tor party