

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies

Project documentation

Team no. 5 - YSTAD

Team members:

Bc. Tomáš Gábrš
Bc. Martina Halajová
Bc. Martin Hauskrecht
Bc. Lukáš Meňhert
Bc. Filip Mrocek
Bc. Michal Staškovan
Bc. Filip Súkeník

Team leader:

Ing. Karol Rástočný, PhD.

Product owners:

Ing. Slavomír Bača
Ing. Slavomír Maťašovský

Table of Content

- 1 Introduction** **3**

- 2 Global Goals** **3**

- 3 Overview of Investment Portal** **4**
 - 3.1 Architecture of Investment Portal 4
 - 3.1.1 Back-end Architecture 5
 - 3.1.2 Front-end Architecture 6
 - 3.2 Data Model 7
 - 3.3 Modules of the System 7
 - 3.3.1 Module for User Authentication 7
 - 3.3.2 Module for Portfolios Management 7
 - 3.3.3 Module for Notification 8
 - 3.3.4 Module for Logging 8

- 4 Summary** **8**

1 Introduction

Investing in financial products has become widely spread between ordinary people. Gaining profit from investments requires evaluation of large amount of available data. Investment Portal provides a simple solution for potential stock traders, who desire to keep track of their portfolios and get systematic help with their trading strategies.

This document contains goals for winter semester and overview of the designed system architecture.

2 Global Goals

Goals for winter semester 2017/2018 are:

- Design suitable and functional architecture of the system according to our future needs. We'll need databases to store our data. Other layers to access the data and make calculations or some operations. Other layers to expose them to the front end and of course the client side of the system.
- Create a dashboard with portfolios overview and an actual state of users' desired products.
- Design section for administration and authentication of users.

Goals for summer semester 2017/2018 are:

- Create a detailed view of the portfolio with a possibility to see information about positions and trades and manage it e.g. add/edit trades and edit positions.
- Create the notifications module for user if there are new calculation for his portfolio ready.
- Create the logging module for the system.

3 Overview of Investment Portal

In the overview of Investment Portal there is described main architecture of the system including example of the generic REST API, front-end component and designed data model.

3.1 Architecture of Investment Portal

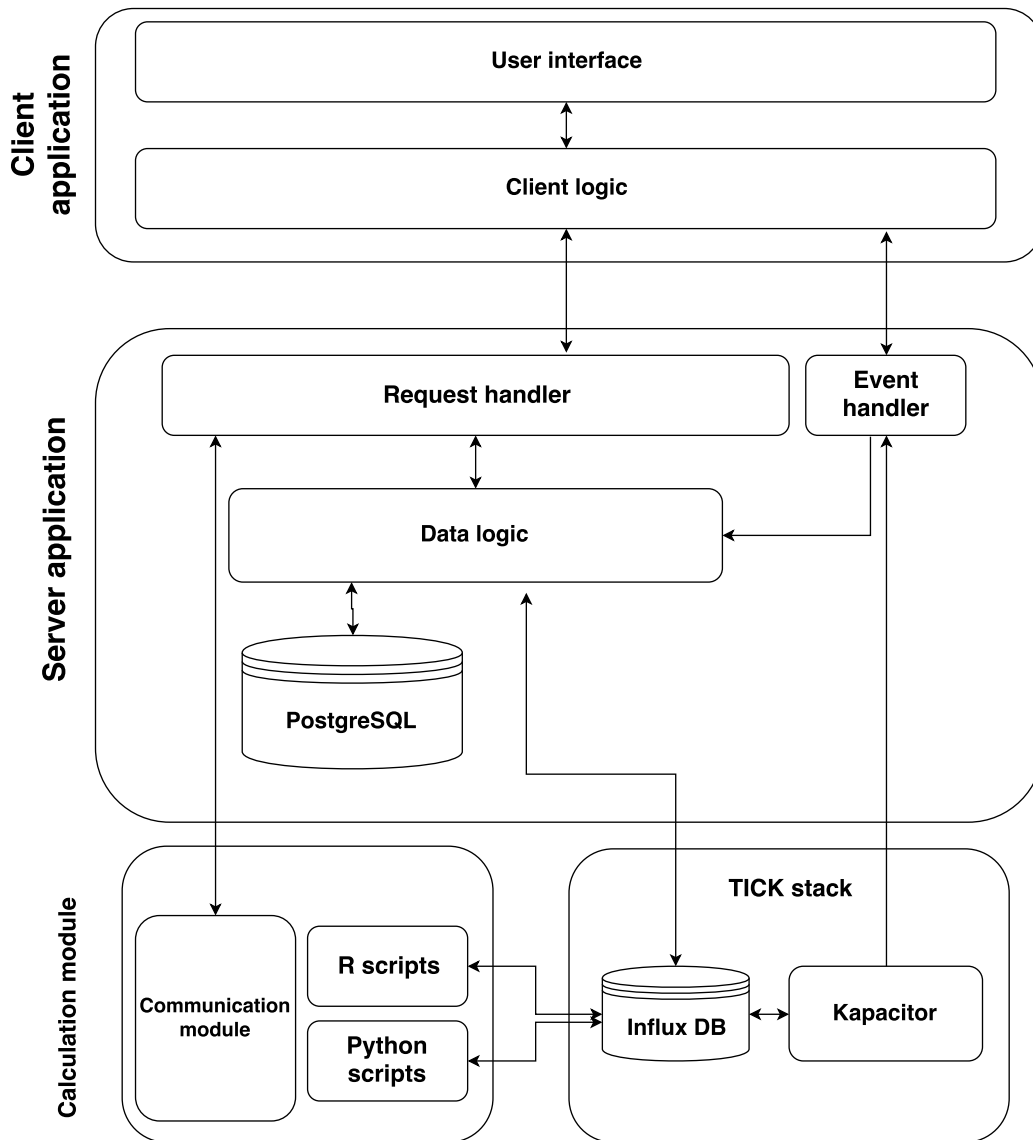


Fig. 1: Overview of the Investment Portal architecture.

Investment Portal is application which uses modified client-server architecture shown in the picture no. 1. The client side provides graphic user interface that is implemented as a web application.

The user can access GUI via web browser and after a successful authentication is able to access the data regarding their portfolios, manage used investment strategies and balance their portfolios.

The server side handles all required calculation and application logic over given databases, and provides the information through REST API (see technical documentation of REST API at <https://investment-portal.duckdns.org/portal/api/swagger-ui.html>). The Investment Portal provides the service of notifying the user about changing market trends which is implemented as separated module.

3.1.1 Back-end Architecture

Generic calling REST service:

- After calling method `findById` of `RestController` (rest api call) object with specified ID is found in database.
- Then `Builder` object is created and method `alterObjectIntoNeededFormat(DataObject)` is called which changes given object into required format.
- The method returns the formatted object and Spring handles transformation of the object into JSON format as output for REST API.
- `RestController` has field of type `CrudRepository`, whose task is to find object in database (if it exists).
- To alter given object `RestController` uses `Builder` which is nested in `DataObject` (each object has its own builder that formats specified object into required format). The feature is implemented according to builder pattern.

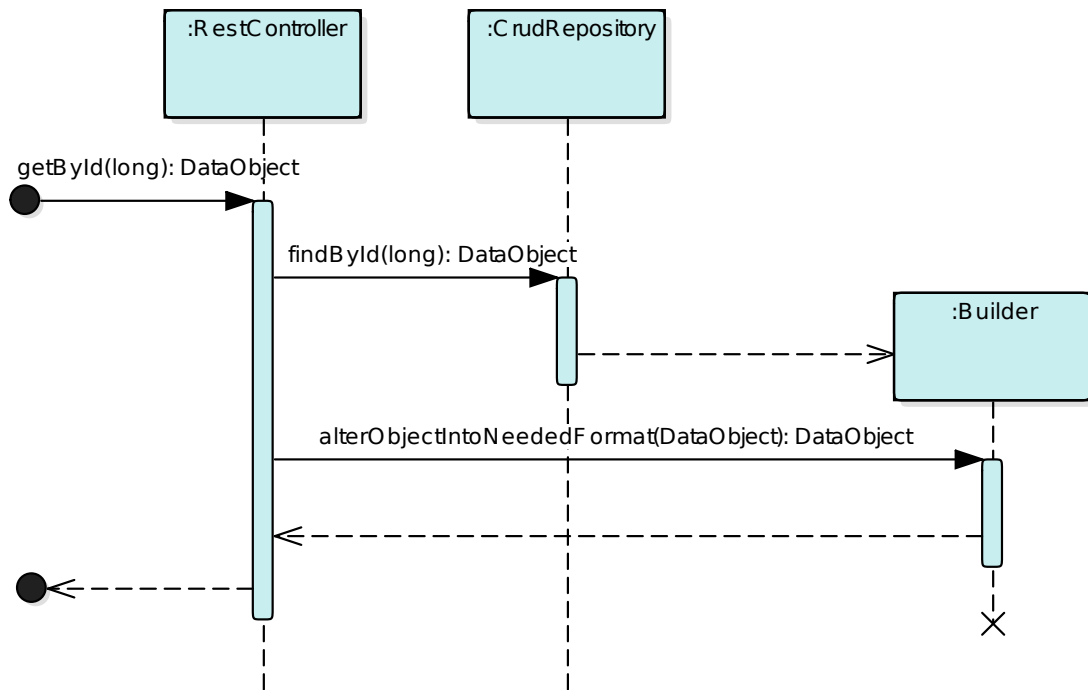


Fig. 2: Sequence diagram of the generic calling REST service.

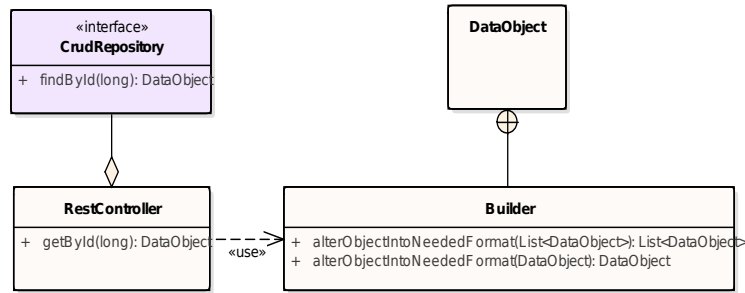


Fig. 3: Class diagram of the generic REST service.

3.1.2 Front-end Architecture

Investment Portal is built on Fuse template. It's template for application based on Angular 5 but using Google's Material Design. Angular 5 is a framework for building applications in HTML with TypeScript compiled to JavaScript.

Angular 5 builds the application from HTML templates that are managed by classes components. Application logic is handled by services. Components with services are merged into modules that group related functionality.

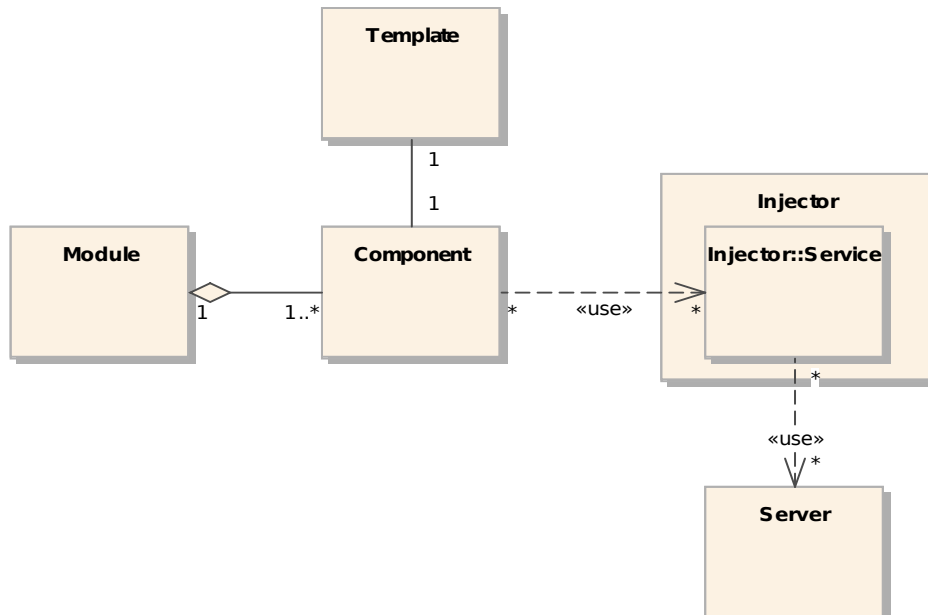


Fig. 4: Component diagram of the generic component group at front-end.

3.2 Data Model

Data model of the system is based on available knowledge and current state of the system. Model will be further updated as the development continues. Design of the data model was consulted with product owners and is based on features that system will include.

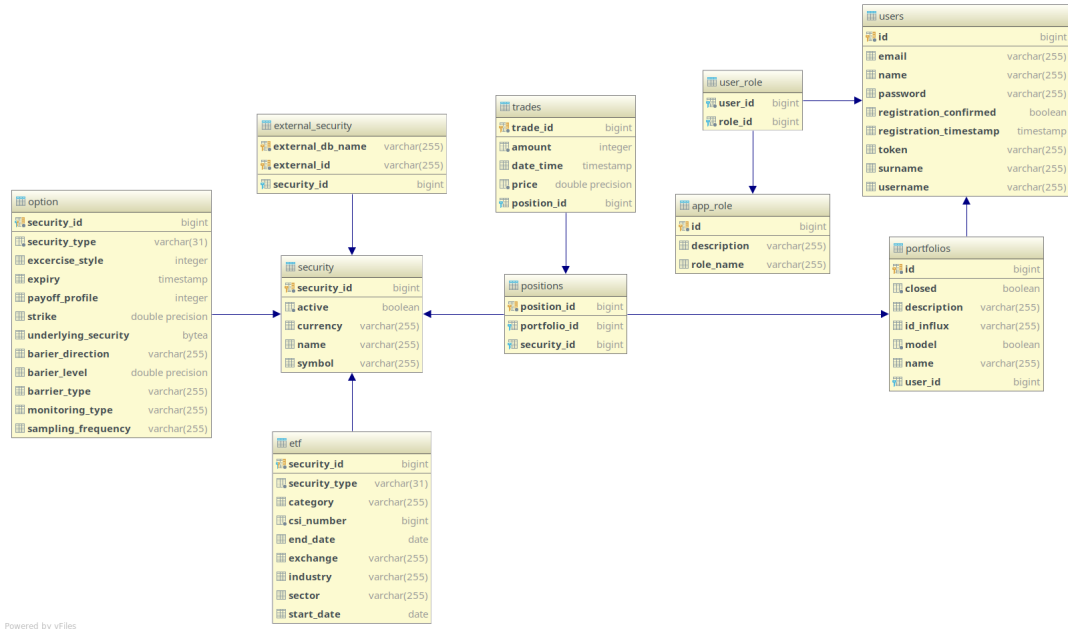


Fig. 5: Overview of the data model.

3.3 Modules of the System

The following section describes each of the five modules that the system contains or will contain in the future.

3.3.1 Module for User Authentication

The Authentication Module serves to authenticate the user's identity with the assigned data in the form of a email and password, and then authorize the user to perform the actions and authorize access to the data. Our system utilizes the Spring Security authentication tools for managing user credentials and authorization.

The module also covers two-factor authentication during user registration with a time-limited validation email on top of the basic Spring Security authentication tools. Validation email contains an unique 32bit token, which is generated upon user registration. The token is valid for 12 hours, during which the user can access the provided URL in the email. This method ensures that the user provided correct email. After accessing the URL, user's account is activated and user can log in into the system.

The module provides scheduled task for deleting inactivated users. The task runs every 6 hours, deleting every user which token has expired and wasn't activated. Deleted user is forced to create a new account on the next attempt to register.

To ensure the confidentiality of user's data, the password is hashed.

Technical description of provided REST services is located at <https://investment-portal.duckdns.org/portal/api/swagger-ui.html> and technical documentation involving server methods is located at <https://investment-portal.duckdns.org/files/docs/technical/>

3.3.2 Module for Portfolios Management

Portfolio management is important part of Investment portal. We want users to be able to manage their portfolios. Managing of portfolios includes:

- creating new portfolio

- editing existing portfolio
- displaying details of existing portfolio
- searching and filtering of portfolios

Portfolios Management module will provide access to important information about portfolios, such as:

- Portfolio Market Value
- Portfolio Last Day Return/Loss
- Portfolio Positions and their Weights

Overview of the user's portfolios is provided at page Dashboard, where are displayed several graphs representing the current state of the portfolios, such as cumulative chart of portfolios returns, gain and loss of each portfolio etc.

User is able to display details of each portfolio, where there are gathered information about portfolio's positions and their weights and performance, and list of each transaction (trades) made.

Technical description of provided REST services and technical documentation involving server methods are located at <https://investment-portal.duckdns.org/portal/api/swagger-ui.html> and <https://investment-portal.duckdns.org/files/docs/technical>.

3.3.3 Module for Notification

Purpose of the notification module is to notify registered users about finished calculations regarding user's portfolio. Notifications is handled using websocket communication.

During user's log in the socket is register to the server and socket is identified by access token, which is unique for every user. Server is constantly listening on changes written in Influx database, where are stored calculations for given portfolio. Monitoring of Influx is assured by tool Kapacitor, which in case of new calculation written into database PORTFOLIO_RETURNS send message containing portfolio ID (`influx_id`) to server. Server identify user's portfolio and then send message to client via websocket. When client application is notified about new calculations, then client requests new data from server through REST API.

3.3.4 Module for Logging

Logging is available in every layer of our system and it is used Sentry service. In Angular it is used the remote logging services to log all errors and important information. On the server side, server uses the logging framework Log4j. For each type of logs, there is appropriate logging level. The set of available levels includes:

- trace
- debug
- info
- warn
- error
- fatal

4 Summary

This document provides an overview of the system architecture and used technologies.

The product is based on client-server architecture, Angular 4 providing the front-end part and Java Spring Framework handling the back-end. The system provides five modules, each handling separate concerns - notifications, authentication, logging, investment strategies, portfolios management.

This document will be further updated based on implemented features and changes.