Denys Matsuiev

Wrocławska Wyższa Szkoła Informatyki Stosowanej

Kierunek: “Aplikacje Biznesowe Java EE”

Semestr: III

nr. albumu: 6169

“Spending Tracker”

Praca zaliczeniowa z przedmiotu “JAVA Enterprise Edition”

Projekt MVC

dla dr Łukasz Jeleń

Wrocław, 23 stycznia 2020 r.

Abstract

Modern web applications vary features to meet with user's expectations and constantly add more and more capabilities into new or current using applications. It's good side for consumers. However, complexities of development such applications grow rapidly as well.

The achitecture of modern application appears at the scene as one of the most important part of development. Model-View-Controller (MVC) serves to order application structure and functionality. Different implementations of this approach allow it widely use in web application engineering.

My application is based on MVC achitectural style and written with Spring Boot and Hibernate frameworks using MySQL database as a persistant data storage. Application serves for tracking everyday expenses and incomes of a certain household. It allows to add amounts of expense or income at the time they had made or later as well. Keeping data of household expenses and incomes into persistant database allows to analyse information and display it in readable graphic way due to be inform about current financial situation this houshold.

Also, the application may use different accounts. It results in more comfortable way keeping information of expenses and incomes made by each member of family.

Keywords: MVC, Spring Boot, Hibernate, MySQL, Java, Web Application

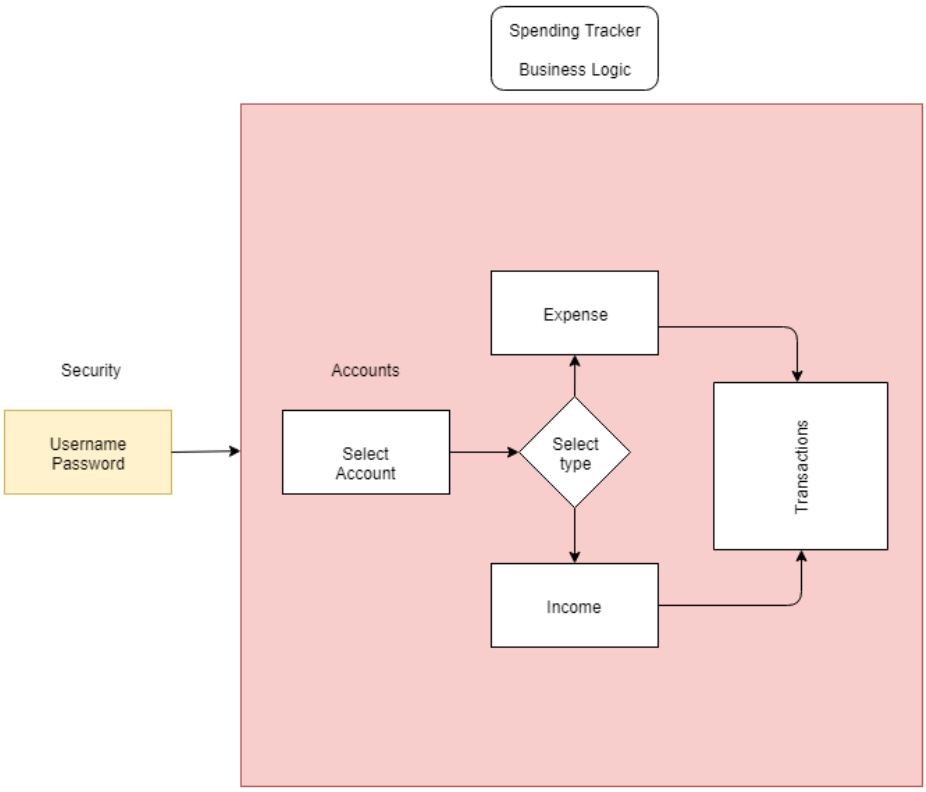
1. **Introduction**

The application “Spending Tracker” is written on architectural style MVC. It was the main requirement. The app serve to track everyday household spendings. It serves adding expense or income belongs to a certain account and keep this data into persistant database. Addings get provided manually by user owned this account at the moment when expense occured or later any time. Keeping household data spendings may be used in different analyses later.

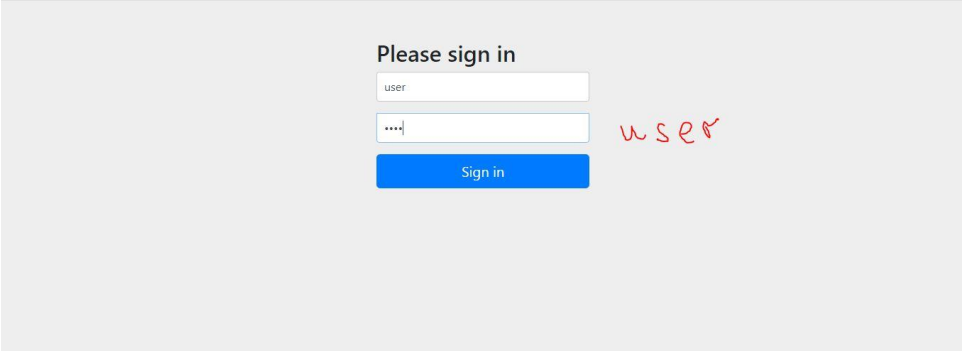
2. Proper content

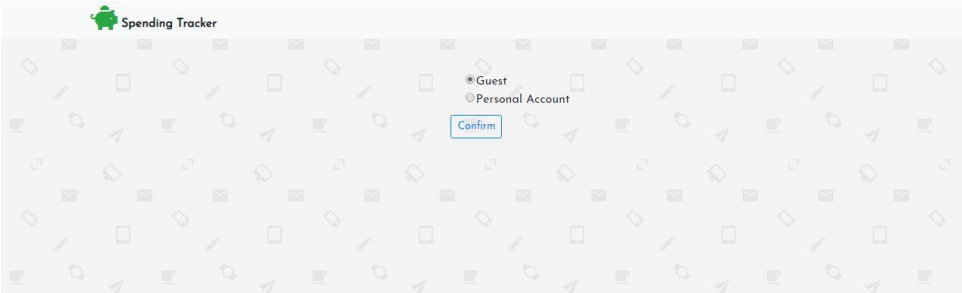
The application is simply secure by demanding rightful user name and password before start working. For study purpose, the user name is ‘user’ and the password ‘user’.

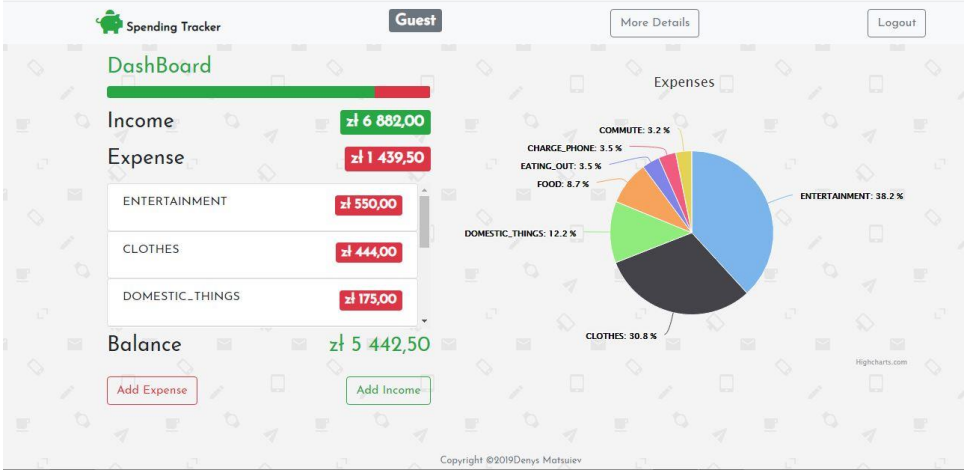
Here is a scheme of business logic for this application.

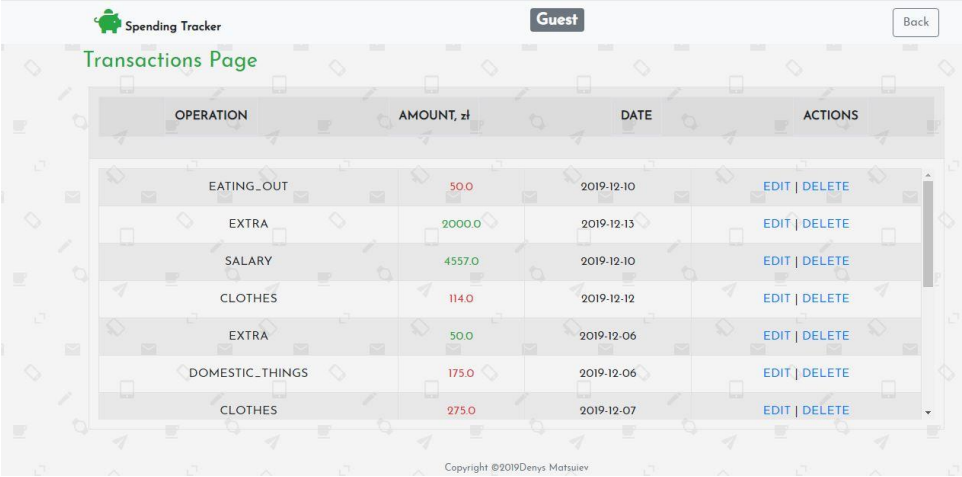


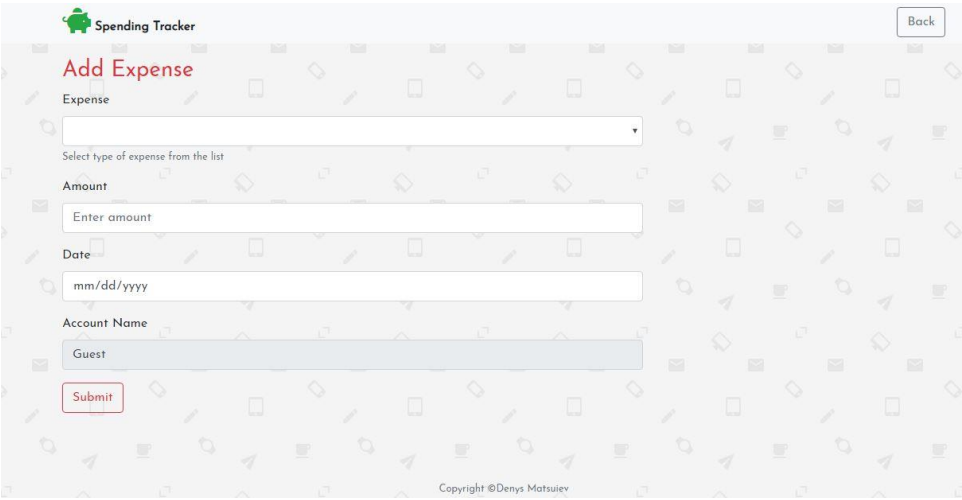
Description of the application via pictures:









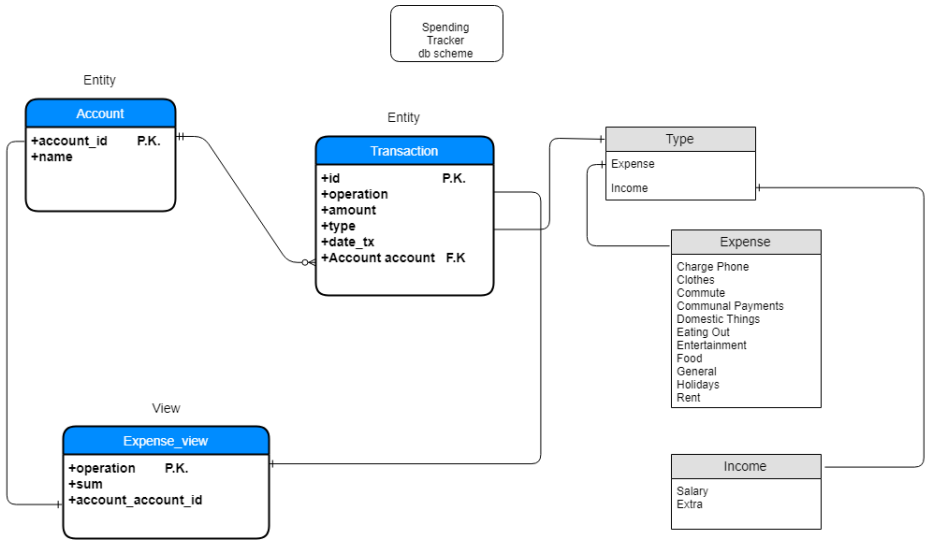


Here are several milestones of the project:

- MVC structure implemented into this app looks in that way:

  The app structure composes of packages. Each package includes classes with fields and methods coordingly their meaning. There are next packages ‘controller’, ‘dao’, ‘model’, ‘service’. The package ‘controller’ includes two classes ‘HomeController’ and ‘TransactionController’ which process requests from the user (browser) and redirect them to the services. ‘Service’ package constitute of several ‘services’. The ‘service’ package classes execute requests and return answer to ‘controller’ package. Package ‘model’ includes classes that represents tables in the mysql database. Package ‘dao’ with classes inside serves for setting connection with database. Answers come back to user displaing as html-pages. View’s technology using in this project is jsp.

- Database scheme is in the project



- Tables' structure

```
mysql> show tables;
+------------------+
| Tables_in_test3 |
+------------------+
| accounts         |
| expense_view     |
| transactions     |
| user             |
+------------------+
4 rows in set (0.31 sec)
```

```
mysql> desc transactions;
+-------------------+-------------+------+-----+---------+----------------+
| Field             | Type        | Null | Key | Default | Extra          |
+-------------------+-------------+------+-----+---------+----------------+
| id                | int(11)     | NO   | PRI | NULL    | auto_increment |
| amount            | double      | NO   |     | NULL    |                |
| date_tx           | varchar(255)| YES  |     | NULL    |                |
| operation         | varchar(255)| YES  |     | NULL    |                |
| type              | varchar(255)| YES  |     | NULL    |                |
| account_account_id| int(11)     | YES  | MUL | NULL    |                |
+-------------------+-------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

```
mysql> desc expense_view;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| operation         | varchar(255)| NO   | PRI | NULL    |       |
| account_account_id| int(11)     | YES  |     | NULL    |       |
| sum               | double      | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

```
mysql> desc user;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | int(11)      | NO   | PRI | NULL    | auto_increment |
| password | varchar(255) | YES  |     | NULL    |                |
| username | varchar(255) | YES  |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

- Creating tables is done by Hibernate. First data inject occurs from data.sql file.
- The graphic user interface represents by pie chart modeling by javascript framework ‘highchart’. https://www.highcharts.com/
- User is allowed using proper account to add expense or income, delete record or update recorded data.

3. Conclusions

The project done. During development I have been learnig a lot. However, some sorting function that was suposed to be done in the project haven’t done. I consider the main feature in the project is binding different accounts with providing data apart each account. All processes of creating entities into database and providing starting data into tables executes by Hibernate. All ties between tables is taking under Hibernate control.

I used jsp to display results in the browser. I thought, It is a simple and convenient way to done this job. But the difficulties occured, when you need to change or add additional features in the project. Combining bootsrap’s classes with html tags and jsp led to complicated and total unreadable logic of application.

My view after accomplished this project is to use something different like rest api to develop logic of application. And some frontend framework to represent data in the client’s side.