TransactionManager

Курсов проект

Дисциплина: Проектиране и интегриране на софтуерни системи

Реализация на системата

Васил Андреев

Август, 2024

Съдържание

1 Въведение

1.1 Цел

Настоящият документ има за цел да опише имплементацията на платформата TransactionManager. Идеята на проекта е да се разработи банкова система, която да предоставя функционалност за управление на финансови транзакции, сметки, потребители и техните роли. Целта е да се предостави сигурна и ефективна платформа за извършване на финансови операции, като осигури лесен достъп до информацията и интуитивно управление на потребителските права. Крайният продукт е функционална и надеждна банкова система, която да отговаря на нуждите на потребителите и да осигури гладко и безопасно използване на финансовите услуги.

1.2 Резюме

Документът съдържа следните точки:

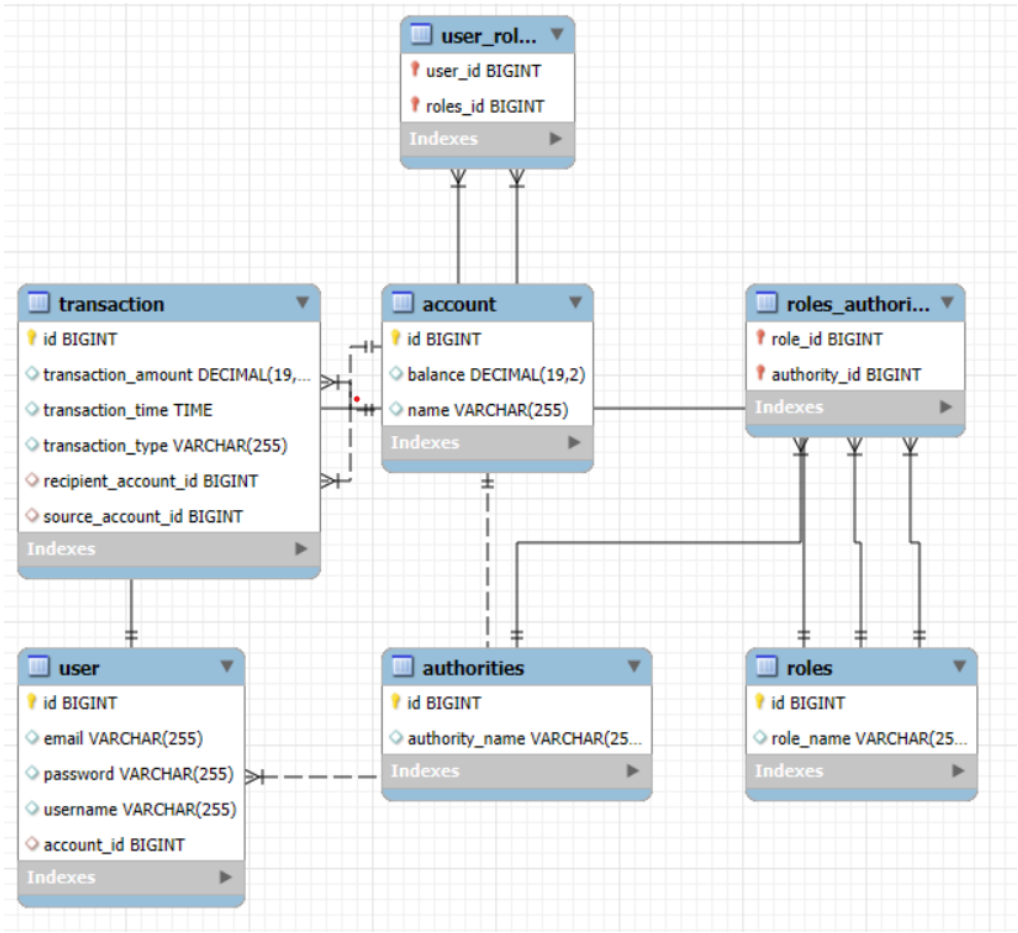
- Описание на технологиите, които са използвани за реализирането на системата. В основата на реализацията е Java с Spring boot и Angular.
- Реализация на базата данни - какви таблици съдържа, техните атрибути и за какво служат.
- Реализацията на бизнес логиката и обмена на информация между компонентите. Използваме REST заявки за комуникацията.
- Реализацията на потребителския интерфейс.

2 Използвани технологии

Графичният интерфейс е изграден с помощта на **CSS**, **HTML** и **JavaScript.** Ползва се модерна версия на рамка за **JavaScript –** Angular. За изграждането на бизнес логика е използван езикът **JAVA с Spring**. Информацията за потребителите както и транзакциите и сметките се съхраняват в базата от данни, достъпна чрез **MySQL.** Кодът е написан през **IntelliJ++** и **Visual Code**. За да се инсталира системата и да се ползва е нужен node пакетът и глобално инсталиран Angular Cli. Чрез тях може да се пусне **графичния интефейс,** а за сървъра ще има нужда ползването на IntelliJ, за стартиране на сървър и **SQL** база от данни. Освен това може да се ползва и Swagger. Приложението се стартира използвайки localhost адреса в браузъра с порт 420, освен ако изрично не е посочен друг порт в конфигурацията. Посредством REST се осъществява комуникацията между клиент-сървър.

3 Реализация на базата от данни

Базата данни има следната схема:

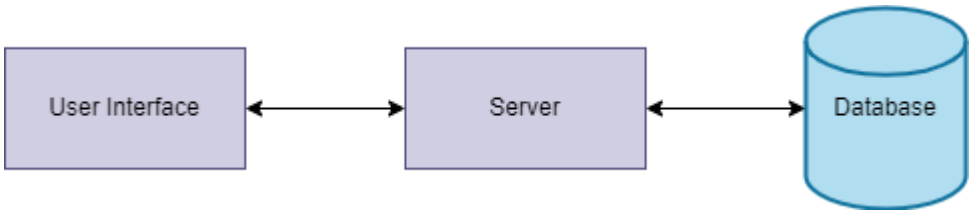


Фиг. 1 База данни

Базата от данни се състои от таблици, подходящо именувани, като в тях се съхраняват данни за сметките, транзакциите и данни за регистрираните потребители. На Фиг. 1 са представени връзките между отделните таблици, като всяка таблица има уникален идентификационен номер. В таблицата account се съдържа информация за сметките - уникален идентификационен номер, име и баланс. Всяка транзакция съдържа в себе си информация за сумата, типа на транзакцията, времето на извършване и сметките, върху които се извършва трансферирането на средствата. Имаме 2 възможни роли за потребителите – admin и regularUser. Естествено, администраторите имат достъп до всички данни на системата, когато правят справки, докато нормалните потребители имат ограничения да виждат само лични справки. В users държим потребителите на системата. Всеки потребител е с потребителско име, имейл адрес и парола, която пазим в хеширан вид.

4 РЕАЛИЗАЦИЯ НА БИЗНЕС ЛОГИКАТА

В сървър частта е имплементира същинската бизнес логика на системата - взаимодействието и обмена на информация между отделните обекти спрямо функционалните изисквания за системата. Бизнес логиката е реализирана с помощта на Java, базирана на Spring, като отделните задачи в нея са разделени в отделни модули – Контролери (Controllers) и Услуги (Services). Контролерите отговарят за обработката на HTTP заявки. Те приемат входящи данни от потребителите и извикват съответните услуги. В Услугите са реализирани основните операции, свързани с банковите функции. Използваме REST заявки за комуникация между клиент и сървър - свързват се HTTP заявките изпратени от клиентската част (браузъра) със съответния endpoint на сървърната страна, да предаде получените данни към бизнес логиката, те да бъдат правилно обработени(или запазени в Базата данни) и в последствие да върне подходящ отговор към клиента (данни в JSON формат или подходящо съобщение за грешка).



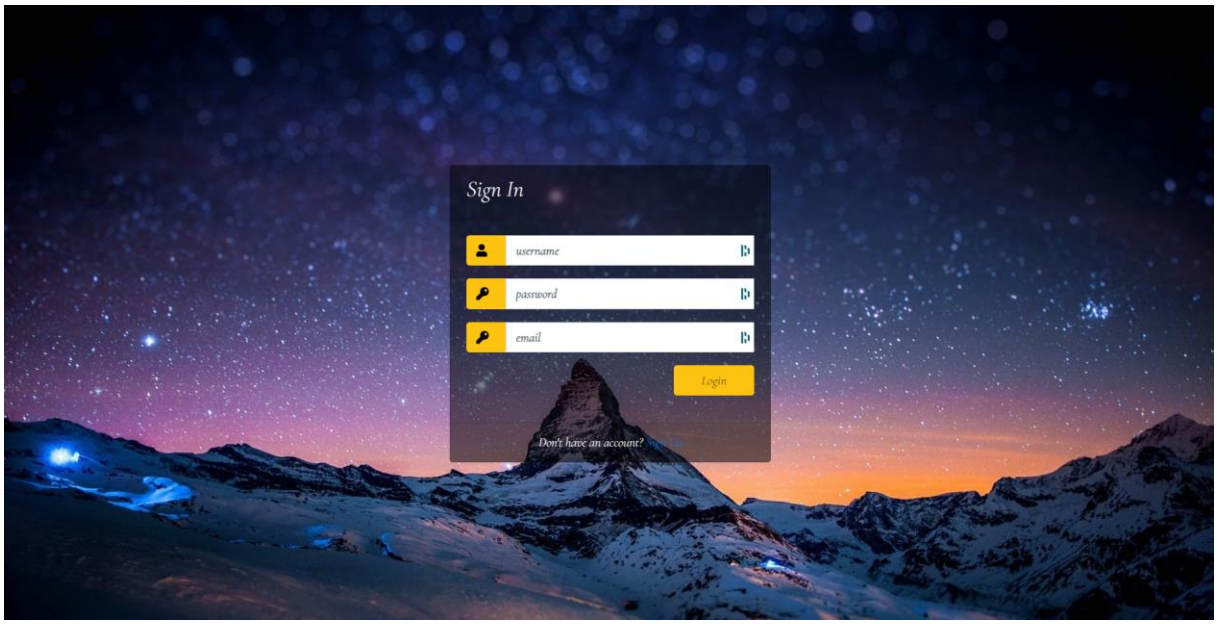
Фиг. 2 Бизнес логика

5 Реализация на потребителския интерфейс

В системата има три вида обекти – потребители (users), сметки (accounts) и транзакции (transactions). Чрез формата за регистрация могат да се създават нови потребители. Акаунт в сайта се прави като се регистрира с потребителско име, имейл и парола. След регистрация потребителите могат директно да влязат в системата. Оттам имат 4 възможности: да проверяват сметки през имена, да проверяват сметки през идентификатор, да създават нови транзакции и да разглеждат стари транзакции. Има функционалност, която позволява експортиране на информация към excel или pdf файл (архивиран или не). Сметки могат да се отварят само в базата, понеже целта на продукта е да управлява сметките на предварително зададените такива, с цел проектът да не излезе извън рамките на курса.

5.1 Вход

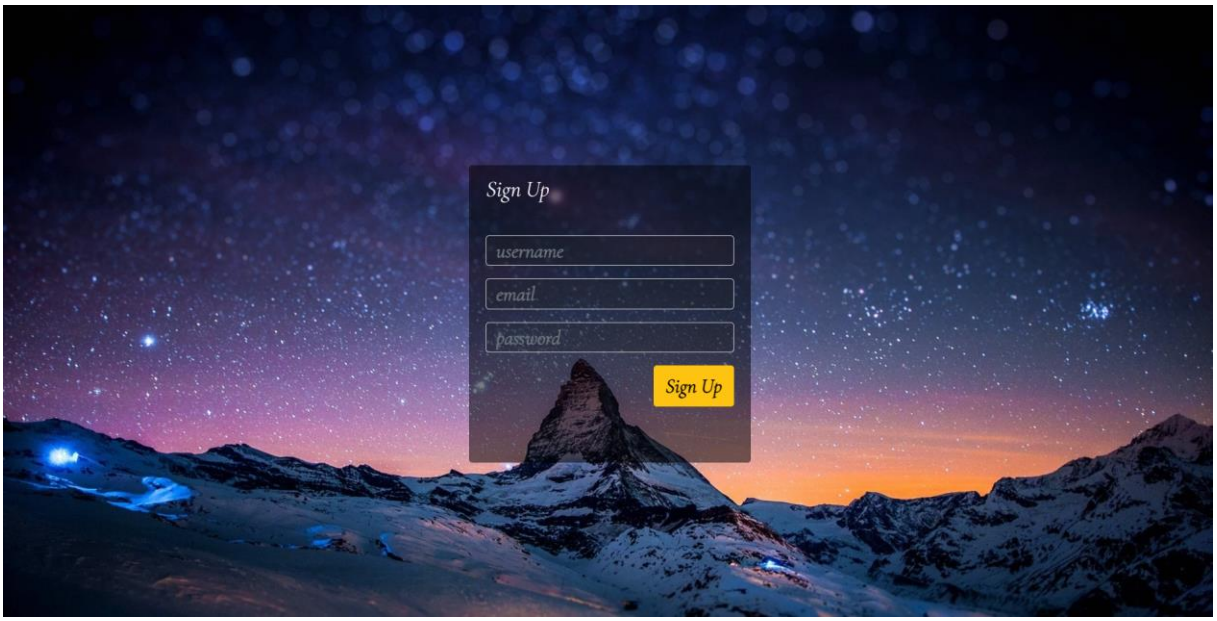
При отваряне на уеб сайта посетителите се препращат към страницата за вход. Имат възможност да въведат своето потребителско име, имейл и парола, в случай, че нямат акаунт се препращат към страницата за регистрация, като за улеснение има добавен бутон към нея.



Фиг. 3 Вход

5.2 Регистрация

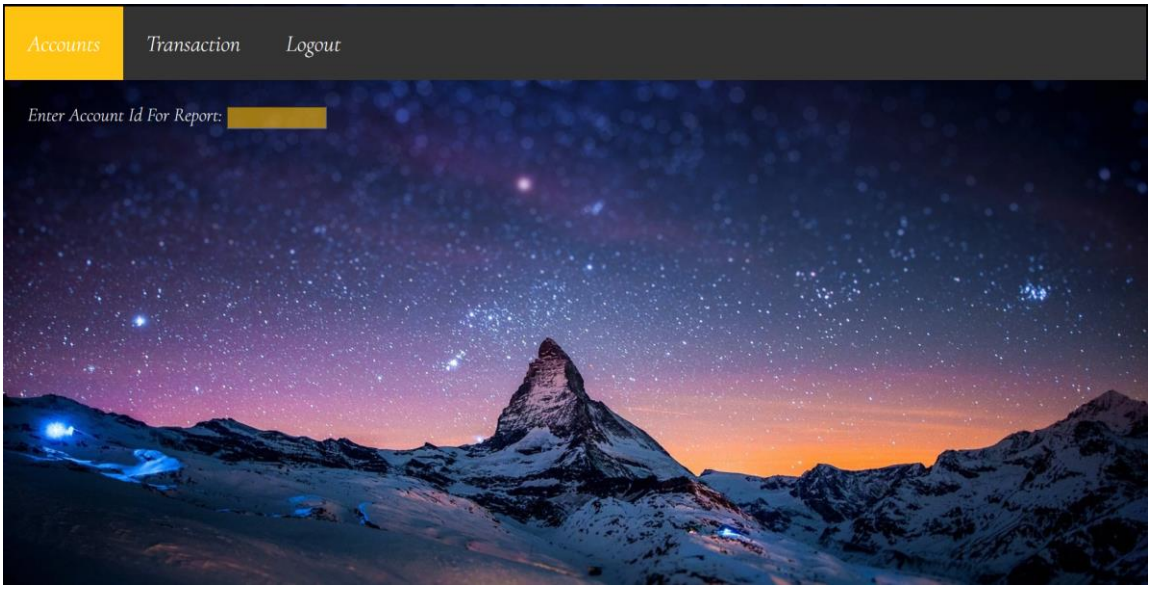
Страницата за регистрация предоставя възможност на гостите на системата да си създадат акаунт. Това значително ги улеснява, защото така ще се съхранява информация за техните записи в базата с данни. Формата за регистрация изисква от гостите да въведат валидна информация.



Фиг. 4 Регистрация

5.3 Начален екран

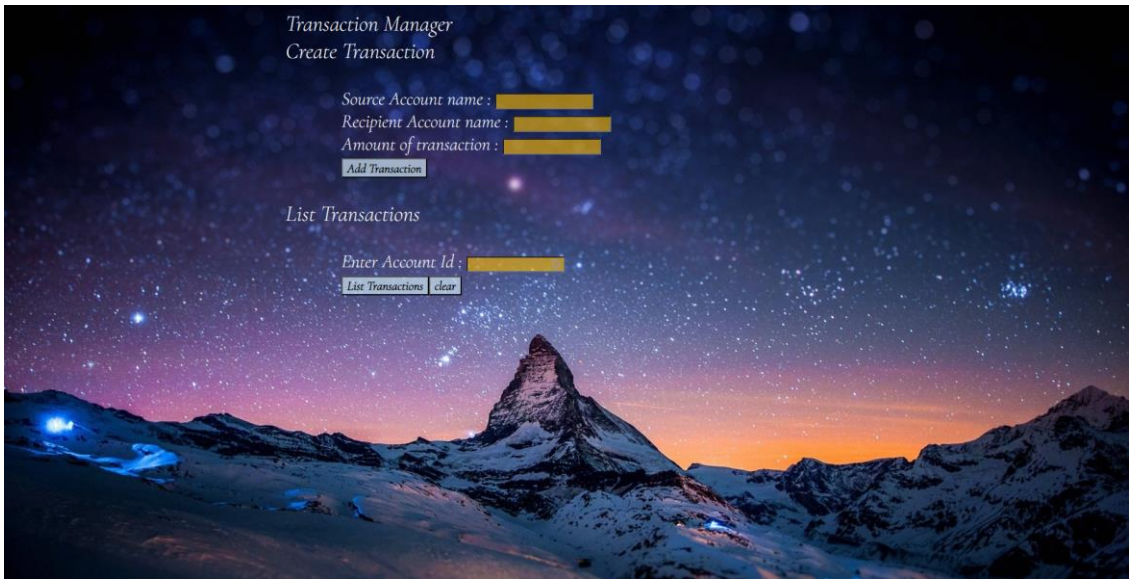
След влизане в системата, на началния екран се показва форма, която приема идентификационен номер за сметка. Потребителите могат да направят справка като въведат желания номер и след това изтеглят справката в избрания формат. Опциите са екселски файл или пдф, като за всяка има вариант да е архивиран.



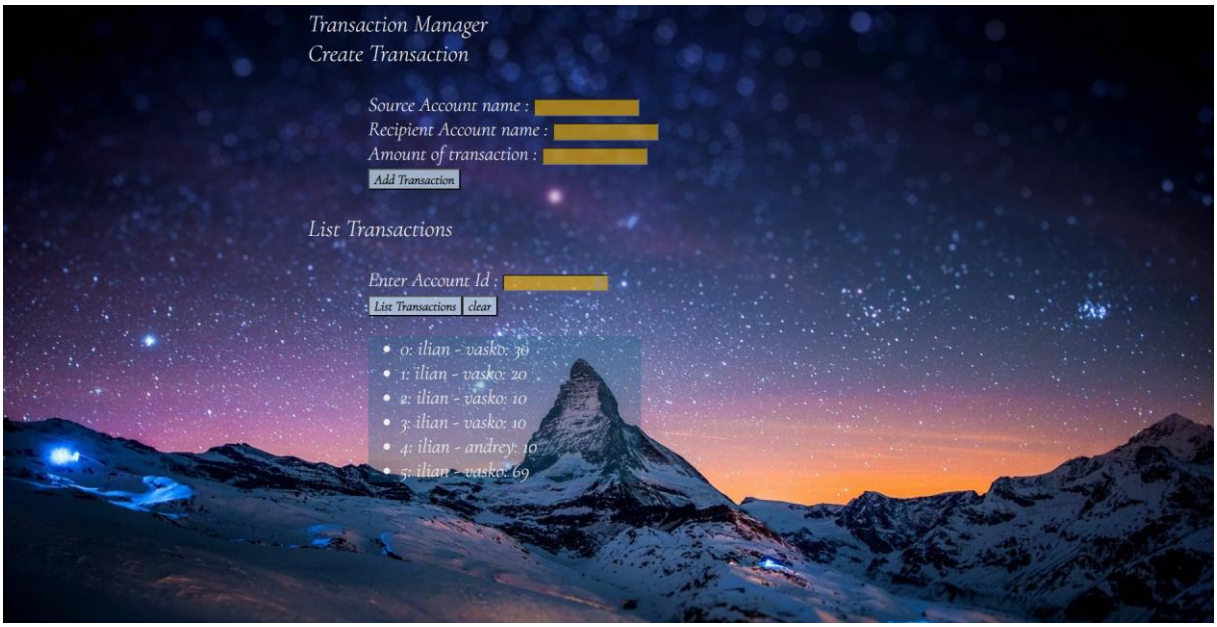
Фиг. 5 Екран за справки на сметки

5.4 Транзакции

Потребителите могат да преглеждат всички извършени транзакции, включващи като получател или източник потребителя с избрания идентификационен номер.



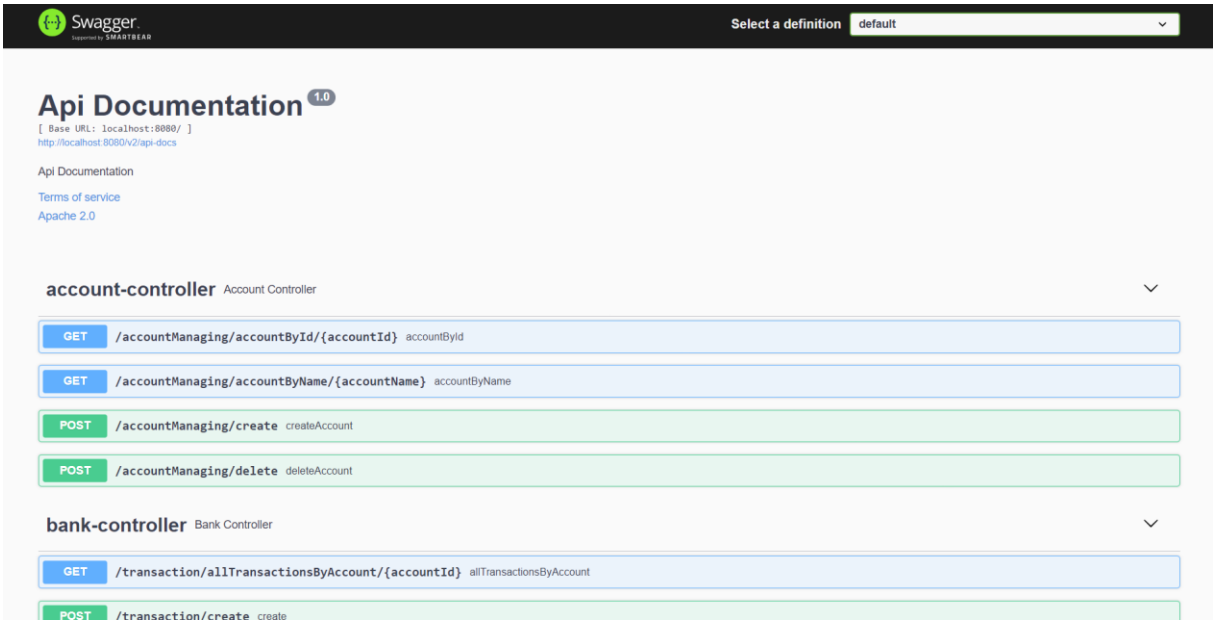
Фиг. 6 Екран за справки на транзакции



Фиг. 7 Преглед на транзакции

6 SWAGGER

Swagger е набор от инструменти и спецификации, които се използват за документиране и проектиране на API функционалностите. Служи за описание на функционалността на различните REST api фунцкии по структуриран начин, което улеснява взаимодействието с него. Swagger предоставя интерфейс за взаимодействие с API-то, което позволява на разработчиците да тестват различни функции директно от документацията.



Фиг. 8 Преглед на Swagger

7 ВНЕДРЯВАНЕ НА СИСТЕМАТА

- Сървъра е локално конфигуриран.
- Интерфейса е локално конфигуриран.
- База данни - MySQL.