Celem niniejszej pracy był projekt oraz implementacja aplikacji internetowej służącej do zarządzania finansami domowymi. Szczególny nacisk położono przy tym na pokazanie możliwości jakie oferują wykorzystane w aplikacji technologie.

Pierwsza część pracy zawiera analizę problematyki związanej z zarządzaniem domowymi finansami, a także analizę i porównanie możliwości dostępnych na rynku aplikacji wspomagających zarządzanie budżetem i majątkiem gospodarstwa domowego. Przeprowadzone w tej części analizy pozwoliły na ustalenie przyjętych do realizacji wymagań funkcjonalnych.

Część praktyczna zawiera projekt systemu, a także dokumentację oraz opisy implementacji i wdrożenia systemu. Zaimplementowana i wdrożona aplikacja składa się z aplikacji klienckiej napisanej w języku TypeScript z wykorzystaniem bibliotek Angular oraz warstwy serwerowej zbudowanej z wykorzystaniem architektury mikrousług, na którą składają się trzy niezależne mikroserwisy napisane w języku Java z wykorzystaniem bibliotek Spring Boot. Mikroserwisy komunikują się ze sobą na dwa sposoby: synchronicznie z wykorzystaniem interfejsu REST API oraz asynchronicznie z wykorzystaniem brokera wiadomości RabbitMQ. Aplikacja wdrożona została na platformie deweloperskiej Okteto oferującej możliwość uruchamiania aplikacji z wykorzystaniem konteneryzacji w systemie Kubernetes. Ponadto w wybranym mikroserwisie oraz we wspólnej dla wszystkich mikroserwisów bibliotece stworzonej na potrzeby projektowanego systemu zaimplementowano szereg testów jednostkowych oraz integracyjnych napisanych w języku Groovy z wykorzystaniem środowiska Spock. Praca obejmowała także przygotowanie dokumentacji dla wszystkich punktów końcowych REST API, z wykorzystaniem bibliotek do automatycznego generowania dokumentacji w formacie OpenAPI.

Ostatecznym efektem pracy jest działający, skalowalny system zbudowany w oparciu o architekturę mikrousług o podstawowej funkcjonalności, który dzięki modułowej budowie zapewnia możliwość łatwego rozszerzania.

Słowa kluczowe:

Angular, aplikacja internetowa, architektura oparta na wiadomościach, architektura oparta na zdarzeniach, Docker, Groovy, Java, Kubernetes, mikrousługi, MongoDB, nierelacyjna baza danych, programowanie funkcyjne, programowanie reaktywne, Spock, Spring Boot, testowanie sterowane danymi, TypeScript, zarządzanie finansami domowymi

The thesis aimed to design and implement a web application for managing household finances. Particular emphasis was placed on showing the possibilities offered by the technologies used in the application.

The first part of the work contains an analysis of questions related to household finances management and an analysis and comparison of the functionality of applications for budget and household asset management. The analysis carried out within this part allowed for determining the functional requirements adopted for implementation.

The practical part includes the system design, documentation, and descriptions of the implementation and system deployment. The implemented and deployed application consists of a browser application written in TypeScript with the use of the Angular framework and a server layer built using the microservices architecture, which consists of three independent microservices written in Java with the help of Spring Boot framework. Microservices communicate with each other in two ways: synchronously using the REST API and asynchronously using the RabbitMQ message broker. The application was implemented on the Okteto development platform, which offers the ability to run applications using containerization in the Kubernetes system. In addition, several unit and integration tests written in the Groovy language using the Spock test framework were implemented in selected microservice and in the common microservices library, which was created for the needs of the designed system. The work also included preparing documentation for all REST API endpoints, using libraries for the automatic documentation generation in the OpenAPI format.

The final result of the work is a working, scalable system built based on microservices architecture with basic functionality, which thanks to its modular structure allows an easy extension.

Key worlds:

Angular, Event-Driven Architecture, Data-Driven Testing, Docker, functional programming, Groovy, household finance management, Java, Kubernetes, Message-Driven Architecture, microservices, MongoDB, non-relational database, reactive programming, TypeScript, Spock, Spring Boot, web application