

Dokumentacja ZPOiF

Biblioteczka domowa

Zuzanna Glinka, Joanna Kajka, Jakub Kasprzak

Styczeń 2022

1 Cel projektu

W ramach projektu z przedmiotu „Zaawansowane Programowanie Obiektowe i Funkcyjne” postanowiliśmy stworzyć aplikację webową ułatwiającą zarządzanie biblioteczką domową. Dzięki niej z łatwością będziemy mogli zorientować się na temat m.in. ilości książek w naszym domowym zbiorze, preferowanych przez nas gatunków, przeczytanych przez nas książek, czy kto pożyczył od nas daną książkę. Naszym założeniem było, by aplikacja była jak najbardziej przyjazna dla każdego użytkownika. W związku z tym naszym głównym celem stało się stworzenie jak najbardziej intuicyjnego interfejsu. Wraz z powstawaniem aplikacji, modyfikowaliśmy wszystkie funkcjonalności, tak aby sprostać początkowym założeniom.

2 Charakterystyka projektu

Nasza aplikacja pozwala stworzyć jedną bibliotekę dla każdego użytkownika, zawierającą wprowadzony księgozbiór. Ów księgozbiór zawierać będzie informacje takie jak: tytuł książki, autor, rok wydania, wydawnictwo oraz liczba stron. Dodatkowo do każdej książki możemy dodać tagi ułatwiające późniejsze wyszukiwanie, informację czy książka jest obecnie czytana lub komuś wypożyczona.

W każdej chwili użytkownik może dodać do swojej biblioteki nową książkę, usunąć ją, zmodyfikować przypisane do niej tagi oraz zapisać, komu została wypożyczona.

Można też przeszukiwać bibliotekę wpisując wybrane słowa kluczowe, które następnie będą filtrować księgozbiór po każdej z informacji o książce (tytuł, autor, tagi etc.). Oprócz tego istnieje funkcja sortowania wyświetlanych książek po dowolnym parametrze (na przykład alfabetycznie po tytule lub autorze).

3 Wykorzystana technologia

Naszym początkowym założeniem było stworzenie aplikacji okienkowej przy wykorzystaniu JavaFX, jednak wraz z powstawaniem naszego projektu posta-

nowiliśmy zrezygnować z naszych początkowych założeń. Ostatecznie nasza aplikacja została napisana w języku Java z wykorzystaniem frameworka Spring do tworzenia aplikacji webowej oraz z wykorzystaniem frameworka Vaadin do stworzenia widoku tej aplikacji. Komunikację z bazą danych zrealizowaliśmy za pomocą Spring Data JPA.

Spring Data JPA pozwala na automatyczne wykorzystanie odpowiednio zapisanych komend, co pozwoliło nam uniknąć ręcznej implementacji poszczególnych zapytań do bazy danych. Vaadin natomiast zagwarantował spójny i przejrzysty interfejs graficzny przyjazny każdemu, nawet dopiero rozpoczynającemu pracę z naszą aplikacją, użytkownikowi.

Do przechowywania danych zastosowaliśmy relacyjną bazę danych H2, która jest wykorzystywana głównie podczas programowania, nie w aplikacjach produkcyjnych. Jest lekka, ale jej wadą jest przechowywanie danych w pamięci podręcznej komputera lub w pliku. Taka baza danych była wystarczająca na nasze potrzeby i do użytku domowego, do którego nasza aplikacja jest przede wszystkim przystosowana, ale jeśli ktoś chciałby używać jej na większą skalę lub mieć łatwiejszy dostęp do bazy z różnych urządzeń, należałoby wykorzystać inne narzędzie.

Dzięki wykorzystaniu tych narzędzi nasz program będzie bardziej intuicyjne dla zwykłego użytkownika, który nie jest przyzwyczajony do aplikacji okienkowych.

4 Opis biznesowy

Naszym celem jest przyciągnięcie użytkowników, którzy chcieliby lepiej zarządzać domowym księgozbiorem, np: znać listę książek danego autora czy listę książek danego gatunku, tak by w przyszłości nie zakupić ponownie takiej samej książki. Stworzenie aplikacji webowej ma też na celu zachęcenie bardziej przeciętnego użytkownika, pozbawionego jakiegokolwiek doświadczenia z językiem Java, do wypróbowania jej.

Dodatkowo często zdarza się, że wraz z upływem czasu zapominamy komu wypożyczyliśmy daną książkę. Dzięki stworzonej aplikacji będziemy mogli z łatwością uzupełnić nasze braki pamięciowe.

Ponadto, nasza aplikacja skierowana jest do osób, które mają problemy z wybieraniem książek do czytania ze swojego księgozbioru. Dzięki niej, wyszukując kluczowe słowo, na przykład nazwę gatunku, mogą ograniczyć przeszukiwany zbiór i ułatwić sobie ten wybór.

5 Założenia

Naszym założeniem było, aby aplikacja była dostępna na komputerze, oraz aby każdemu księgozbiornikowi odpowiadało jedno konto. Dodatkowo, przy tworzeniu aplikacji staraliśmy się, aby przeciętny użytkownik mógł modyfikować swój księgozbiór wykonując minimalną liczbę operacji.

Ponieważ baza danych jest lokalna, nie wymaga synchronizacji ani hostowania jej w sieci, co obniża potencjalny koszt eksploatacji aplikacji.

6 Funkcje

W naszej aplikacji wyróżnić można obiekty typu Książka i Użytkownik. Ostatecznie postanowiliśmy zrezygnować z klasy Pula tagów i zastąpić ją polem w Książce w celu uproszczenia projektu, jednocześnie bez utraty funkcjonalności. Nie zaimplementowaliśmy też klasy Biblioteka, a zamiast tego powiązaliśmy pojedynczą listę książek z użytkownikiem, z uwzględnieniem, że „Admin” ma dostęp do księgozbioru każdego z użytkowników, ale poszczególni użytkownicy mogą przeglądać jedynie własne listy.

Użytkownik, gdy chce rozpocząć korzystanie z biblioteki, musi początkowo założyć konto. W tym celu podaje swoje imię, nick oraz hasło. Gdy już założy konto, ma następujące możliwości modyfikacji swojego księgozbioru:

1. dodawanie książki do biblioteki;
2. usuwanie książki z biblioteki;
3. modyfikacja informacji o osobie, która pożyczyła książkę;
4. modyfikacja listy tagów dotyczących książki;
5. modyfikacja numeru czytanej strony;
6. wyszukiwanie słów kluczowych;
7. wyszukiwanie pożyczonych książek.

6.1 Dodawanie książki do biblioteki

Dodając książkę do biblioteki użytkownik ma możliwość podania następujących parametrów:

- obrazek w formacie .jpg, który będzie się wyświetlał obok książki na liście;
- tytuł książki;
- autor książki;
- data publikacji;
- liczba stron;
- numer ISBN (międzynarodowy numer identyfikujący książkę);
- tagi.

Należy podać wszystkie parametry, z wyjątkiem być może obrazka i tagów, żeby wszystkie funkcje aplikacji działały poprawnie. Po dodaniu książki, spośród tych parametrów, można modyfikować jedynie tagi.

6.2 Usuwanie książki z biblioteki

Analogicznie jeśli użytkownik może dodać książkę, może ją również usunąć. Czynność ta nie wymaga wykonywania żadnych skomplikowanych operacji.

6.3 Modyfikacja informacji o osobie, która pożyczyła książkę

Jednym z dwóch modyfikowalnych parametrów utworzonej już książki jest informacja o osobie, której dana książka została pożyczona. Możemy wpisać w tym polu dowolny ciąg znaków, najprawdopodobniej imię i nazwisko lub pseudonim, dzięki któremu później będziemy mogli zidentyfikować daną osobę.

6.4 Modyfikacja listy tagów dotyczących książki

W naszej bibliotece możemy korzystać z wyszukiwania książek za pomocą przypisanym do nich tagów. Aby nasza aplikacja była jak najbardziej przyjazna dla użytkownika, każdy może na bieżąco modyfikować listę tagów, tak aby spełniały jego kryteria, a w późniejszym czasie wyszukiwanie było pozbawione wszelkich komplikacji czy problemów. W tym celu wystarczy wejść w możliwość edytowania tagów i w odpowiednim polu zmodyfikować tagi.

6.5 Modyfikacja numeru czytanej strony

Ostatnim modyfikowalnym po utworzeniu książki parametrem jest numer strony, na której ostatnio skończyliśmy czytać. Nie jest to szczególnie przydatna funkcja jeśli planujemy dokończyć czytanie książki w jednym ciągu, ale jeśli chcemy odłożyć książkę i wrócić do niej po dłuższym czasie lub pożyczyć ją komuś nie skończywszy jej, w tym miejscu możemy sobie zapisać, od którego momentu możemy wrócić do czytania.

6.6 Wyszukiwanie słów kluczowych

Na dole strony w przeznaczonym do tego polu użytkownik może wpisać interesujące go słowa kluczowe, które następnie zostaną wykorzystane do filtrowania listy książek. Warto zaznaczyć, że nie będą filtrowane jedynie po liście tagów, ale po dowolnym parametrze, więc w ten sposób można przeprowadzić takie działania jak odnalezienie wszystkich posiadanych przez nas książek danego autora, odnalezienie książek wypożyczonych konkretnej osobie, wydanych w konkretnym roku albo należących do danego gatunku, który podaliśmy jako tag.

6.7 Wyszukiwanie pożyczonych książek

Tak samo jak w przypadku wyszukiwaniu słów kluczowych, możemy wyszukiwać pożyczone książki. Mianowicie na dole strony mamy opcję wyświetlenia wszystkich książek, książek pożyczonych oraz tych, które dalej powinny znajdować się w naszym domowym księgozbiore.

7 Napotkane problemy

Początkowo planowaliśmy zrealizować naszą aplikację samodzielnie tworząc odpowiednik bazy danych i implementując instrukcje które by się z nią komunikowały lub tworząc bazę danych i wykorzystując instrukcje w języku SQL, a interfejs graficzny wykorzystując JavaFX do stworzenia aplikacji okienkowej. Analizując jednak takie rozwiązanie, postanowiliśmy zamiast tego wykorzystać wspomniane przez nas framework Spring, Vaadin oraz Spring Data JPA i bazę H2. Decyzję o zmianie podejścia podjęliśmy jeszcze przed rozpoczęciem pracy nad projektem, więc nie straciliśmy kodu, musieliśmy jedynie zmienić założenia projektowe takie jak liczba i funkcje klas.

Aby rozpocząć pracę nad projektem, musieliśmy zrozumieć działanie Vaadina i JPA, co, zważywszy na to, że były to dla nas nowe technologie, okazało się pewnym problemem. Są one bowiem dość skomplikowane i zaznajomienie się z nimi w stopniu umożliwiającym realizację naszego pomysłu zajęło nam trochę czasu. Niemniej jednak ostatecznie udało nam się wykorzystać te technologie i zrealizować założony pomysł.

Podczas pracy nad projektem problematyczne okazało się zrealizowanie funkcji wyszukiwania po słowach kluczowych, która sprawiała problemy w połączeniu z funkcją zapisu nowo wprowadzonych informacji do bazy danych gdy nie wszystkie parametry książki były uzupełnione. Ostatecznie udało nam się doprowadzić do tego, żeby funkcja działała poprawnie.