BookTracker - Personal Library Management System

14 czerwca 2025

COLLEGIUM WITELONA
Uczelnia Państwowa

Autorzy

Przemysław Typa
Damian Domański
Daniel Mosakowski
Mateusz Rejman

Sprawozdanie projektowe

Zaawansowane metody programowania
Kierunek - Informatyka S3PAM2(1)u

Prowadzący

mgr inż. Krzysztof Rewak

Spis treści

1 Wstęp

1.1 Opis

BookTracker miał być wieloplatformowym systemem stworzonym z myślą o miłośnikach literatury, umożliwiającym zarządzanie kolekcją książek, ocenianie ich oraz śledzenie postępów czytelniczych. Celem była budowa narzędzia, które w intuicyjny i przyjazny sposób pomogłoby organizować czytelnicze doświadczenia — niezależnie od urządzenia, z jakiego korzysta użytkownik.

Projekt był planowany jako aplikacja wielowarstwowa - użytkownicy mieli posiadać dostęp do swoich danych poprzez aplikację mobilną (Android), aplikację desktopową (Windows) oraz przeglądarkę internetową. Komunikacja pomiędzy poszczególnymi komponentami miała się odbywać poprzez API oparte o framework Laravel, a dane miał przechowywać centralny serwer baz danych, umożliwiając pełną synchronizację informacji.

Finalnie jednak założenia te uległy zmianie, gdyż zespół miał znaczne trudności organizacyjne, a wielu członków nie wywiązało się ze swoich zadań. W efekcie aplikacja mobilna przechowuje dane w Firebase, aplikacja desktopowa korzysta z odrębnej bazy, a API zostało wdrożone, ale tak naprawdę nikt z niego ostatecznie nie korzysta. Pomimo takiej sytuacji, praca stanowi cenne doświadczenie dotyczące zarządania projektem, współpracy w zespole oraz tworzenia aplikacji od podstaw.

1.2 Zespół i opis technologiczny

Skład zespołu oraz przypisane komponenty:

- Przemysław Typa - odpowiedzialny za aplikację mobilną (Android) napisaną w języku Kotlin z użyciem nowoczesnej biblioteki Jetpack Compose.
- Mateusz Rejman - odpowiedzialny za aplikację webową stworzoną w technologii Vue 3 z wykorzystaniem TypeScript oraz frameworka Bootstrap.
- Damian Domański - autor aplikacji desktopowej rozwijanej w języku Python, dostosowanej do działania na systemach operacyjnych Windows.
- Daniel Mosakowski - odpowiedzialny za backend oraz API systemu, stworzonego w technologii PHP z użyciem frameworka Laravel oraz relacyjnej bazy danych MariaDB.

1.3 Zastosowane technologie :

- Aplikacja mobilna: Kotlin + Jetpack Compose
- Aplikacja webowa: Vue 3, TypeScript, Bootstrap
- Aplikacja desktopowa: Python
- Backend/API: PHP + Laravel

- Baza danych: MariaDB
- Autoryzacja i synchronizacja: Firebase Authentication + Firestore (dla aplikacji mobilnej)

1.4 Opis funkcjonalny

Functionalities/Features

OPZ	Feature	API	Web	Mobile	Desktop
BKT-01	The first administrator is automatically added to the system.	✓			
BKT-02	The administrator can log in to the system.	✓		✅	✅
BKT-03	The administrator can add books to the database.	✓			
BKT-04	The administrator can manage users (add, remove, edit).	✓		✅	✅
BKT-05	The user can add books to their collection.	*	✓	✅	✅
BKT-06	The user can edit the details of books in their collection.	*	✓	✅	
BKT-07	The user can rate books and add reviews.	*	✓	✅	✅
BKT-08	The user can view a list of books they want to read, are reading, or have already read.	*	✓	✅	✅
BKT-09	The user can filter books by genre, author, or rating.	*	✓	⏳	✅
BKT-10	The user can track their reading progress.	*	✓	✅	
BKT-11	The user can add books to their "want to read" list.	*	✓	✅	✅
BKT-12	The user can create an account / log in.	*	✓	✅	✅
BKT-13	The user can rate other users based on their reviews and books.	*	✓		
BKT-14	The user can browse reviews of other users.	*	✓	✅	✅
BKT-15	The mobile and web applications support offline reading progress tracking.	*	✓	✅	
BKT-16	The application supports languages: Polish, English.	*	✓	⏳	✅
BKT-17	The user can reset their password.	*	✓	✅	✅
BKT-18	The user can add books by ISBN number.	*	✓		✅
BKT-19	The user can enable 2FA for security.	*	✓	✅	✅
BKT-20	The user can change the app theme.	*	✓		✅
BKT-21	The user can log out.	*	✓	✅	✅

1.5 Repozytoria GIT

Przemysław Typa - https://github.com/PrzemekTypa/BookTracker-mobile
Damian Domański - https://github.com/fvalz/BookTracker-Deskop
Daniel Mosakowski - https://github.com/danielmosakowski/BookTracker-api

1.6 Cele pracy

Głównym celem projektu BookTracker było zaprojektowanie i stworzenie kompletnego, wieloplatformowego systemu do zarządzania książkami, który pozwala użytkownikowi na katalogowanie, ocenianie i śledzenie postępów w czytaniu. W dobie cyfrowej konsumpcji treści, wiele osób gubi się w liczbie rozpoczętych, niedokończonych lub planowanych
książek. Brakuje im narzędzia, które w prosty, zintegrowany i dostępny sposób umożliwiałoby organizację prywatnej biblioteki i motywowało do regularnego czytania.

Aplikacja ma służyć nie tylko jako cyfrowa półka z książkami, ale także jako osobisty asystent czytelniczy, który wspiera użytkownika w jego własnym tempie i stylu czytania. Dla użytkownika oznacza to:

- możliwość łatwego dodawania książek, zarówno ręcznie, jak i poprzez automatyczne wyszukiwanie po numerze ISBN, zarządzanie statusem lektury: „chcę przeczytać", „czytam", „przeczytane",
- możliwość oceniania i recenzowania przeczytanych pozycji,
- funkcjonalność przypomnień i wyzwań czytelniczych dla zwiększenia motywacji,
- synchronizację danych między urządzeniami, by móc kontynuować czytanie bez względu na platformę.

Z perspektywy zespołu projektowego, celem pracy było także praktyczne przećwiczenie umiejętności tworzenia systemów rozproszonych, z podziałem obowiązków na frontend i backend, a także wdrożenie współczesnych metod zabezpieczania danych i autoryzacji użytkowników. Projekt stanowił również szansę na wykorzystanie nowoczesnych technologii, takich jak Jetpack Compose, Vue 3, czy Laravel, w środowisku bliskim rzeczywistym warunkom pracy zespołowej w branży IT.

2 Aplikacja mobilna

Aplikacja mobilna BookTracker-mobile na system Android służy do zarządzania osobistą biblioteką użytkownika. Pozwala w prosty sposób dodawać książki do kolekcji, oceniać je, pisać recenzje oraz śledzić postęp czytania. Użytkownik może oznaczać książki jako „chcę przeczytać", „czytam"lub „przeczytane", co ułatwia kontrolowanie własnego procesu czytelniczego. Dzięki intuicyjnemu interfejsowi, aplikacja zapewnia wygodne i szybkie zarządzanie listą lektur z poziomu urządzenia mobilnego.

2.1 Przegląd technologii i rozwiązań

Aplikacja mobilna BookTracker została stworzona z wykorzystaniem języka Kotlin i nowoczesnej biblioteki Jetpack Compose, co pozwala na budowanie deklaratywnego i responsywnego interfejsu użytkownika w sposób szybki i wygodny. Jetpack Compose jest obecnie rekomendowanym standardem tworzenia UI przez Google i znacznie upraszcza proces projektowania interfejsów.

Jako backend zastosowano Firebase, który zapewnia szeroki zestaw usług, takich jak:

- Firebase Authentication - obsługa logowania przez e-mail oraz Google
- Firebase Firestore - baza danych NoSQL do przechowywania danych użytkownika
- Firebase Console - do zarządzania użytkownikami i monitorowania zdarzeń

Dzięki integracji z Firebase możliwe było szybkie wdrożenie mechanizmów logowania, synchronizacji danych oraz podstawowego bezpieczeństwa - bez konieczności implementowania własnego API po stronie mobilnej. Firebase działa w oparciu o protokół HTTPS, co zapewnia domyślną ochronę przesyłanych danych.

2.2 Struktura i części składowe aplikacji

Struktura aplikacji opiera się na kilku podstawowych ekranach (tzw. screens), które odpowiadają za poszczególne funkcjonalności. Każdy ekran to oddzielny komponent Jetpack Compose, co pozwala na przejrzysty podział kodu.

```
com.example.booktrackermobile/
│
├── model/          # Modele danych (np. Book, User) – klasy reprezentujące strukturę danych
│
├── navigation/     # Nawigacja w aplikacji – definicje tras, NavGraph, kontrola przepływu ekranów
│
├── network/        # Warstwa sieciowa – API, Retrofit, zapytania HTTP
│
├── repository/     # Repozytoria – logika dostępu do danych z różnych źródeł (API, lokalne DB)
│
├── screens/        # Ekrany UI – pliki z interfejsem użytkownika (Compose), poszczególne widoki
│
├── storage/        # Lokalna pamięć – obsługa bazy danych (np. Room), SharedPreferences itp.
│
├── ui.theme/       # Motywy – kolory, typografia, style Compose
│
├── viewmodel/      # ViewModel – logika biznesowa i stan dla ekranów, zgodnie z architekturą MVVM
│
└── MainActivity    # Główna aktywność aplikacji – punkt startowy aplikacji (Compose setup)
```

Główne komponenty aplikacji to:

- LoginScreen – ekran logowania użytkownika (Google lub e-mail)
- RegisterScreen – ekran rejestracji konta
- ResetPasswordScreen – ekran umożliwiający zresetowanie hasła
- MainScreen – ekran główny, zawierający zakładki
  - AllBooksTab – wyszukiwarka książek z wykorzystaniem OpenLibrary API
  - BookDetailsScreen – szczegóły książki, możliwość dodania do biblioteczki
  - BookItem – pojedynczy komponent książki w liście
- SettingsTab – ustawienia użytkownika (wylogowanie, nazwa/email konta)

Każdy ekran zarządza swoją własną logiką i stanem za pomocą wzorca MVVM (Model-View-ViewModel), co pozwala na lepsze oddzielenie warstwy interfejsu od logiki biznesowej.

2.3 System Android i jego komponenty

BookTracker-mobile działa w środowisku systemu operacyjnego Android, który jest dominującą platformą mobilną na rynku. Aplikacja wykorzystuje podstawowe komponenty systemu Android, takie jak:

- Activity - kontenery dla interfejsu użytkownika
- Composable Functions - nowy model UI w Compose
- ViewModel - architektura zarządzająca danymi UI
- LiveData / State - do śledzenia i aktualizowania stanu interfejsu

Dodatkowo wykorzystano system nawigacji w Compose, co pozwala użytkownikowi przechodzić między ekranami w sposób płynny i intuicyjny, z zachowaniem historii nawigacji.

2.4 Jetpack Compose - nowoczesne podejście do UI

Jetpack Compose to nowa biblioteka do tworzenia interfejsów użytkownika na Androida, która znacząco upraszcza budowanie dynamicznych i atrakcyjnych graficznie aplikacji. Zamiast korzystać z plików XML, UI budowany jest w całości w kodzie Kotlin.

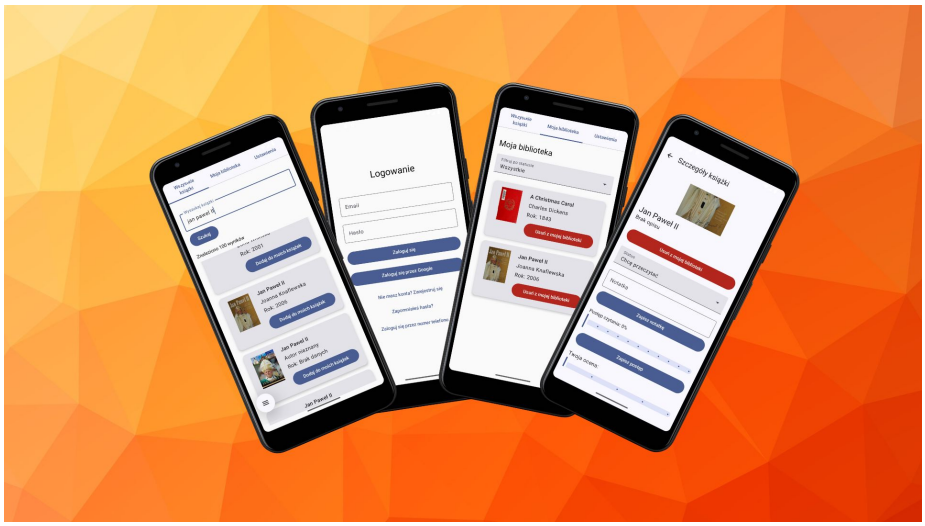
Zalety Compose, które wykorzystano w projekcie:

- Deklaratywny styl kodu: interfejs „wynika"ze stanu aplikacji
- Łatwe zarządzanie stanem komponentów
- Przejrzysty i modularny kod
- Lepsza integracja z narzędziami testującymi i animacjami

Dzięki Compose możliwe było stworzenie przejrzystego i nowoczesnego interfejsu, który można łatwo rozbudować lub dostosować w przyszłości.

3 Aplikacja mobilna - Interfejs użytkownika

W aplikacji BookTracker-mobile interfejs użytkownika został zaprojektowany z myślą o prostocie, przejrzystości i intuicyjnej obsłudze. Użytkownik może w łatwy sposób poruszać się między ekranami, dodawać książki, zmieniać ich status oraz zarządzać swoim kontem. Poniżej opisano główne widoki aplikacji.



3.1 Ekran logowania

Ekran logowania umożliwia zalogowanie się do systemu za pomocą trzech metod:

- konta Google (OAuth 2.0),
- adresu e-mail i hasła (Firebase Authentication).
- Numeru telefonu potwierdzając kodem.

Na ekranie znajdują się pola do wpisania danych logowania oraz przycisk do logowania przez Google. W przypadku błędnych danych użytkownik otrzymuje stosowny komunikat.

3.2 Ekran rejestracji

Ekran rejestracji umożliwia założenie konta w systemie przy użyciu adresu e-mail i hasła. Formularz rejestracyjny zawiera walidację poprawności danych (np. długość hasła, format e-maila). Po pomyślnym utworzeniu konta, użytkownik automatycznie zostaje zalogowany i przeniesiony do głównego ekranu aplikacji.

3.3 Zmiana hasła

Ekran umożliwiający resetowanie hasła w przypadku jego zapomnienia. Użytkownik wprowadza swój adres e-mail, a system (Firebase) automatycznie wysyła wiadomość z linkiem do zresetowania hasła. Funkcjonalność ta podnosi poziom bezpieczeństwa i pozwala użytkownikowi na szybkie odzyskanie dostępu do konta.

3.4 Ustawienia

Ekran ustawień pozwala użytkownikowi:

- wyświetlić nazwę lub adres e-mail konta, z którego aktualnie korzysta,
- wylogować się z aplikacji.

W przyszłości ten ekran może być rozszerzony o dodatkowe funkcje, takie jak zmiana hasła, wybór motywu, synchronizacja ustawień itp.

4 Aplikacja mobilna - wnioski projektowe oraz możliwości rozwoju

4.1 Wnioski

Praca nad aplikacją BookTracker była dla mnie cennym doświadczeniem, które pomogło mi znacznie poszerzyć swoje umiejętności w zakresie tworzenia aplikacji mobilnych.
Podczas realizacji tego zadania miałem okazję wdrożyć szereg funkcjonalności od podstaw, jak zarządzanie kolekcją książek, po te bardziej zaawansowane, jak autoryzacja użytkowników czy przechowywanie danych w chmurze.
Pełna integracja z Firebase była dla mnie sporym wyzwaniem — wymagała odpowiedniego skonfigurowania, zarządania uprawnieniami oraz uzupełnienia o dodatkowe warstwy kontroli dostępu — ale ostatecznie znacznie ułatwiła tworzenie aplikacji od strony backendu.
Pracując nad aplikacją, miałem okazję zapoznać się z architekturą MVVM, ViewModel, LiveData, Repository oraz Jetpack Compose.
Nie wdrożyłem ich w pełni, ale zrozumiałem podstawowe założenia i zasady, które się za nimi kryją, co na pewno ułatwi mi tworzenie aplikacji w przyszłości. Pomogło mi to lepiej zorganizować kod, zarządzać stanem aplikacji i sprawić, by był on czytelny oraz bardziej przejrzysty. Poznałem też szereg dobrych praktyk dotyczących tworzenia aplikacji na platformę Android, które zamierzam stosować w kolejnych projektach.
Mimo wielu napotkanych trudności, uważam, że praca nad BookTracker była dla mnie cennym doświadczeniem. Pomogła zrozumieć proces tworzenia aplikacji od podstaw, pokazała, jak skutecznie zarządzać danymi, jak dbać o bezpieczeństwo informacji, a także jak organizować kod w większym projekcie.

4.2 Zdobyte kompetencje i możliwości rozwoju aplikacji

W trakcie realizacji projektu zdobyłem szereg kompetencji praktycznych, w tym:

- Tworzenie aplikacji w architekturze **MVVM** z wykorzystaniem **Jetpack Compose**, co znacząco ułatwia rozdzielenie logiki od widoku i poprawia przejrzystość kodu.

- Integrację z **Firebase** (Authentication, Firestore, Storage) oraz obsługę logowania, sesji i danych użytkownika w sposób bezpieczny i wydajny.
- Zagadnienia związane z **bezpieczeństwem**, takie jak szyfrowanie połączeń, przechowywanie tokenów JWT, monitoring logowań oraz implementacja uwierzytelniania dwuskładnikowego (2FA).

BookTracker w wersji mobilnej stanowi w pełni funkcjonalny komponent, który może zostać użyty jako część większego, wieloplatformowego systemu do zarządzania książkami. Aplikacja może być dalej rozwijana, a jej otwarta architektura umożliwia łatwe wdrażanie nowych funkcji.

4.3 Możliwości rozwoju aplikacji

Chociaż aktualna wersja aplikacji oferuje komplet podstawowych funkcji, istnieje wiele kierunków dalszego rozwoju, zarówno pod względem technicznym, jak i funkcjonalnym. Do potencjalnych usprawnień i rozszerzeń można zaliczyć:

- **Lokalna baza danych (Room)** – integracja z lokalną bazą danych w celu lepszej wydajności i przechowywania danych tymczasowych.
- **Rozbudowany system powiadomień** – przypomnienia o celach czytelniczych, nowych recenzjach czy rekomendacjach książkowych.
- **Integracja z zewnętrznymi serwisami** (np. Goodreads, Google Books) – automatyczne pobieranie recenzji i synchronizacja list.
- **Tryb społecznościowy** – dodanie opcji znajomych, obserwowania profili i wymiany opinii w ramach zamkniętej społeczności użytkowników.
- **Rozszerzona analityka czytania** – statystyki czasu czytania, najczęściej czytane gatunki, wykresy postępów itp.
- **Motywacja i gamifikacja** – punkty, odznaki, wyzwania miesięczne i ranking użytkowników.

Takie kierunki rozwoju nie tylko zwiększyłyby atrakcyjność aplikacji dla użytkowników, ale również pozwoliłyby na szersze wykorzystanie jej w różnych grupach wiekowych i środowiskach edukacyjnych.

5 Aplikacja mobilna – uruchamianie

5.1 Uruchamianie lokalnie

Aby uruchomić projekt lokalnie za pomocą Android Studio:

1. **Sklonuj repozytorium:**

```
git clone https://github.com/twoja-nazwa-uzytkownika/BookTracker-mobile.git
```

2. **Otwórz projekt w Android Studio:**
   (a) Uruchom Android Studio.
   (b) Wybierz *„Open an existing project"* i wskaż folder `BookTracker-mobile`.
3. **Zbuduj i uruchom aplikację:**
   (a) Upewnij się, że masz uruchomiony emulator lub podłączone urządzenie fizyczne.
   (b) Kliknij przycisk *Run ()* lub użyj skrótu `Shift + F10`.
4. **Zależności:**
   (a) Android Studio automatycznie pobierze wszystkie zależności Gradle.
   (b) W razie potrzeby zsynchronizuj projekt z Gradle i doinstaluj brakujące SDK.

5.2 Instalacja na urządzeniu

Jeśli chcesz zainstalować aplikację bezpośrednio na telefonie (bez użycia Android Studio):

1. **Zlokalizuj plik APK:**
   Ścieżka: `BookTracker-mobile/app/release/app-release.apk`
2. **Przenieś plik APK na urządzenie:** Możesz użyć kabla USB, Bluetooth, Dysku Google lub innej metody przesyłania plików.
3. **Zezwól na instalowanie z nieznanych źródeł:**
   (a) Na telefonie przejdź do: `Ustawienia > Zabezpieczenia > Instalowanie aplikacji z nieznanych źródeł`
   (b) Zezwól menedżerowi plików lub aplikacji, z której będziesz instalować, na instalację APK.
4. **Zainstaluj aplikację:**
   (a) Otwórz menedżer plików na telefonie.
   (b) Znajdź `app-release.apk` i kliknij, aby rozpocząć instalację.
   (c) Potwierdź chęć instalacji i poczekaj na zakończenie.
5. **Uruchom aplikację** z ekranu głównego lub listy aplikacji.