



Wyższa Szkoła Przedsiębiorczości i Administracji w Lublinie

Dokumentacja Projektu: Aplikacja Mobilna "ToDoManager"

Rafał Skowroński

Michał Łuczyński

Informatyka PUW

Semestr: 5

Rok akademicki: 24/25

Lublin 2025

Cel projektu

Celem aplikacji jest stworzenie nowoczesnego narzędzia do zarządzania zadaniami ("to-do") za pomocą aplikacji mobilnej. System umożliwi przechowywanie danych w lokalnej bazie danych (SQLite) lub w zewnętrznej bazie serwerowej (MySQL). Aplikacja pozwoli użytkownikowi dodawać, usuwać oraz oznaczać zadania jako wykonane.

Wymagania

Wymagania funkcjonalne

1. Użytkownik może wybrać, czy dane będą przechowywane lokalnie, czy zdalnie.
2. Aplikacja umożliwia dodawanie nowych zadań.
3. Aplikacja umożliwia oznaczanie zadań jako wykonane.
4. Aplikacja umożliwia usuwanie zadań.
5. Aplikacja wyświetla listę zadań w przejrzysty i nowoczesny sposób.
6. Dane lokalne są zarządzane za pomocą SQLite, a dane zdalne za pomocą MySQL.

Wymagania niefunkcjonalne

1. Aplikacja wykorzystuje nowoczesne wzorce projektowe (Room ORM, JDBC).
 2. Interfejs jest responsywny i intuicyjny.
 3. System wspiera SQLite i MySQL jako backend.
 4. Dane użytkownika są przechowywane w bezpieczny sposób.
-

Przypadki użycia

Dodanie nowego zadania

- **Aktor:** Użytkownik aplikacji
- **Opis:** Użytkownik wprowadza nazwę zadania i opcjonalny opis, a następnie zapisuje je w wybranej bazie danych.
- **Rezultat:** Zadanie zostaje dodane do listy.

Oznaczenie zadania jako wykonane

- **Aktor:** Użytkownik aplikacji
- **Opis:** Użytkownik oznacza wybrane zadanie jako wykonane.
- **Rezultat:** Zadanie zmienia status na wykonane i jest odpowiednio sformatowane (np. przekreślone).

Usunięcie zadania

- **Aktor:** Użytkownik aplikacji
- **Opis:** Użytkownik usuwa wybrane zadanie z listy.
- **Rezultat:** Zadanie zostaje usunięte z bazy danych.

Wyświetlenie listy zadań

- **Aktor:** Użytkownik aplikacji
 - **Opis:** Aplikacja wyświetla wszystkie zadania zapisane w wybranej bazie danych.
 - **Rezultat:** Lista zadań jest widoczna na ekranie w przejrzystym układzie.
-

Kluczowe obiekty

Todo

- **Atrybuty:**
 - id: Unikalny identyfikator (INTEGER, klucz główny).
 - title: Nazwa zadania (TEXT, nie może być NULL).
 - is_done: Status zadania (INTEGER, domyślnie 0).
 - **Opis:** Reprezentuje pojedyncze zadanie w bazie danych.
-

Struktura bazy danych

SQLite (lokalna baza danych)

```
CREATE TABLE Todo (  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  title TEXT NOT NULL,  
  is_done INTEGER DEFAULT 0  
);
```

MySQL (serwerowa baza danych)

```
CREATE TABLE todos (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  title VARCHAR(255) NOT NULL,  
  is_done BOOLEAN DEFAULT FALSE  
);
```

Przykładowe dane

SQLite

```
INSERT INTO Todo (title, is_done) VALUES ('Kup mleko', 0);
```

```
INSERT INTO Todo (title, is_done) VALUES ('Zakończ dokumentację', 1);
INSERT INTO Todo (title, is_done) VALUES ('Przypomnij o spotkaniu', 0);
```

MySQL

```
INSERT INTO todos (title, is_done) VALUES ('Przeczytaj książkę', 0);
INSERT INTO todos (title, is_done) VALUES ('Napraw samochód', 1);
INSERT INTO todos (title, is_done) VALUES ('Zapłać rachunki', 0);
```

Zapytania do baz danych

Zapytania wybierające:

1. Wszystkie zadania:
`SELECT * FROM Todo;`
2. Zadania oznaczone jako wykonane:
`SELECT * FROM Todo WHERE is_done = 1;`
3. Zadania niewykonane:
`SELECT * FROM Todo WHERE is_done = 0;`
4. Liczba wszystkich zadań:
`SELECT COUNT(*) FROM Todo;`
5. Zadania zawierające określony tekst:
`SELECT * FROM Todo WHERE title LIKE '%mleko%';`

Zapytania manipulujące danymi:

1. Dodanie nowego zadania:
`INSERT INTO Todo (title, is_done) VALUES ('Naucz się Javy', 0);`
2. Aktualizacja statusu zadania:
`UPDATE Todo SET is_done = 1 WHERE id = 1;`
3. Usunięcie zadania:
`DELETE FROM Todo WHERE id = 2;`

Zapytania manipulujące strukturą bazy:

1. Dodanie nowej kolumny "priority":
`ALTER TABLE Todo ADD COLUMN priority INTEGER DEFAULT 0;`
 2. Zmiana nazwy tabeli:
`ALTER TABLE Todo RENAME TO Tasks;`
-

Implementacja bazy danych w aplikacji mobilnej

Aplikacja wykorzystuje Room ORM do zarządzania SQLite:

Przykład zapisu danych:

```
Todo newTodo = new Todo("Zakończ projekt", false);  
database.todoDao().insert(newTodo);
```

Przykład odczytu danych:

```
List<Todo> todos = database.todoDao().getAll();
```

Implementacja połączenia z bazą MySQL

Aplikacja wykorzystuje JDBC do połączenia z bazą MySQL:

Przykład połączenia:

```
Connection connection = DriverManager.getConnection(  
    "jdbc:mysql://host:port/database_name", "username", "password");
```

Przykład zapisu danych:

```
String query = "INSERT INTO todos (title, is_done) VALUES (?, ?)";  
PreparedStatement statement = connection.prepareStatement(query);  
statement.setString(1, "Zaprojektuj aplikację");  
statement.setBoolean(2, false);  
statement.executeUpdate();
```