

Programowanie aplikacji mobilnych

Informacje ogólne

Przedmiot: Programowanie aplikacji mobilnych

Forma zaliczenia: Projekt semestralny

Termin oddania: Ostatnie zajęcia w semestrze

Cel projektu

Celem projektu jest samodzielne zaprojektowanie, zaimplementowanie i przetestowanie działającej aplikacji mobilnej. Projekt ma na celu praktyczne zastosowanie wiedzy zdobytej podczas wykładów i ćwiczeń z zakresu programowania aplikacji mobilnych.

Wymagania projektowe

Wybór technologii

Projekt można zrealizować w **dowolnej technologii** tworzenia aplikacji mobilnych:

- **Android natywny** - Android Studio (Kotlin/Java)
- **iOS natywny** - Xcode (Swift)
- **Cross-platform** - React Native, Flutter, Kotlin Multiplatform, .NET MAUI, Ionic

Wymagania funkcjonalne

Aplikacja musi zawierać **minimum 5 z poniższych elementów**:

1. **Wiele ekranów/widoków** - co najmniej 3 różne ekrany z nawigacją między nimi
2. **Interakcja z użytkownikiem** - formularze, przyciski, gesty, obsługa zdarzeń
3. **Przechowywanie danych** - lokalna baza danych, SharedPreferences/UserDefaults lub pliki
4. **Komunikacja sieciowa** - pobieranie danych z API, wyświetlanie danych z internetu
5. **Lista/siatka elementów** - wyświetlanie kolekcji danych (RecyclerView, ListView, FlatList itp.)
6. **Multimedia** - obsługa obrazów, dźwięków, animacji lub kamery
7. **Powiadomienia** - lokalne lub push notifications
8. **Geolokalizacja** - wykorzystanie GPS, mapy
9. **Sensory urządzenia** - akcelerometr, żyroskop, czujnik światła itp.
10. **Autoryzacja użytkownika** - logowanie, rejestracja (lokalne lub przez zewnętrzny serwis)

Wymagania niefunkcjonalne

- **Responsywność** - aplikacja powinna poprawnie wyświetlać się na różnych rozmiarach ekranów
- **Obsługa orientacji** - działanie w orientacji pionowej i poziomej (lub uzasadnione ograniczenie do jednej)
- **Estetyka** - spójny styl wizualny, czytelny interfejs użytkownika
- **Stabilność** - aplikacja nie powinna się zawieszać ani crashować przy typowym użytkowaniu

Etapy pracy nad projektem

Etap 1 - Koncepcja (zajęcia 1-2)

- Wybór tematu aplikacji
- Wybór technologii
- Wstępny projekt funkcjonalności
- Szkice interfejsu użytkownika (mockupy)

Etap 2 - Implementacja podstawowa (zajęcia 3-5)

- Konfiguracja środowiska i projektu
- Implementacja struktury aplikacji i nawigacji
- Tworzenie podstawowych ekranów i layoutów

Etap 3 - Implementacja zaawansowana (zajęcia 6-7)

- Implementacja logiki biznesowej
- Integracja z bazą danych lub API
- Dodanie zaawansowanych funkcjonalności

Etap 4 - Finalizacja i testy (zajęcia 8)

- Testowanie aplikacji
- Poprawki i optymalizacja
- Przygotowanie dokumentacji
- **Prezentacja i oddanie projektu**

Dokumentacja projektu

Do projektu należy dołączyć dokumentację zawierającą:

1. **Opis aplikacji** - cel, grupa docelowa, główne funkcjonalności
2. **Wybrana technologia** - uzasadnienie wyboru
3. **Architektura aplikacji** - schemat struktury, użyte wzorce projektowe
4. **Opis zaimplementowanych funkcjonalności** - lista z krótkim opisem każdej
5. **Zrzuty ekranu** - pokazujące działanie aplikacji
6. **Instrukcja uruchomienia** - jak skompilować i uruchomić projekt
7. **Napotkane problemy** - opis trudności i sposoby ich rozwiązania
8. **Możliwości rozwoju** - co można by dodać w przyszłości

Kryteria oceny

Kryterium	Punkty
Funkcjonalność (spełnienie wymagań)	0-40
Jakość kodu i architektura	0-20
Interfejs użytkownika i UX	0-15
Dokumentacja	0-15
Innowacyjność i dodatkowe funkcje	0-10
Razem	100

Skala ocen

- 91-100 pkt bardzo dobry (5.0)
- 81-90 pkt dobry plus (4.5)
- 71-80 pkt dobry (4.0)
- 61-70 pkt dostateczny plus (3.5)
- 51-60 pkt dostateczny (3.0)
- 0-50 pkt niedostateczny (2.0)

Forma oddania projektu

1. **Kod źródłowy** - repozytorium Git (GitHub, lub inne)
2. **Dokumentacja** - plik PDF lub Markdown w repozytorium
3. **Prezentacja** - krótka demonstracja działającej aplikacji na ostatnich zajęciach (5–10 minut)

Przykładowe tematy aplikacji

Poniżej propozycje tematów - można realizować własny pomysł po konsultacji z prowadzącym:

- Aplikacja do zarządzania listą zadań (to-do list) z przypomnieniami
- Dziennik treningowy z wykresami postępów
- Aplikacja pogodowa z prognozą na podstawie lokalizacji
- Organizator przepisów kulinarnych z listą zakupów
- Tracker nawyków z statystykami
- Aplikacja do nauki słówek z fiszkami
- Prosty komunikator lub czat
- Gra mobilna (quiz, memory, prosta gra zręcznościowa)
- Portfolio / CV w formie aplikacji
- Aplikacja do skanowania i katalogowania kodów kreskowych/QR

Dodatkowe informacje

- W przypadku problemów technicznych - konsultacje podczas zajęć
- Zachęcam do korzystania z systemu kontroli wersji Git od początku projektu
- Dozwolone jest korzystanie z bibliotek zewnętrznych i frameworków
- Plagiat skutkuje oceną niedostateczną