AKADEMIA NAUK STOSOWANYCH
W NOWYM SĄCZU

Wydział Nauk Inżynieryjnych
Katedra Informatyki

DOKUMENTACJA PROJEKTOWA

PROGRAMOWANIE URZĄDZEŃ MOBILNYCH

Wirtualny Dziekanat

Autor:

Marcin Dudek
Mateusz Basiaga

Prowadzący:
mgr inż. Dawid Kotlarski

Nowy Sącz 2024

Spis treści

1. Ogólne określenie wymagań

1.1. Główne funkcje aplikacji

- **Logowanie i autoryzacja:**
  Użytkownicy powinni móc logować się za pomocą uczelnianego adresu e-mail oraz hasła. Po piewrszym logowaniu użytkownik dostanie opcje logowania za się pomocą odcisku palaca albo skanu twarzy. Możliwość logowania tyczy się zarówno dla studentów, jak i pracowników (np. wykładowcy, administracja).

- **Podgląd ocen i zaliczeń:**
  Studenci powinni mieć możliwość przeglądania swoich ocen z egzaminów, kolokwiów oraz innych zaliczeń. Możliwość filtrowania wyników na podstawie przedmiotu, semestru czy wykładowcy.

- **Plan zajęć:**
  Podgląd bieżącego planu zajęć z opcją aktualizacji na żywo (jeśli np. zajęcia zostaną odwołane czy przeniesione).

- **Harmonogram egzaminów i sesji:**
  Informacje o nadchodzących egzaminach, sesjach poprawkowych i innych ważnych wydarzeniach związanych z uczelnią. Możliwość zapisywania się na egzaminy, jeżeli to wymagane.

- **Powiadomienia:**
  Push notifications o nowych ocenach, nadchodzących zajęciach, zmianach w harmonogramie lub ważnych ogłoszeniach.

- **Informacje ogólne:**
  Tablica ogłoszeń z najważniejszymi informacjami od uczelni, np. nowe zarządzenia rektora, wydarzenia na kampusie itp.

- **Profile użytkowników:**
  Każdy użytkownik powinien mieć profil z podstawowymi danymi (imię, nazwisko, nr indeksu, rocznik, itd.). Możliwość aktualizacji niektórych danych kontaktowych.

- **Responsywność i UX:**
  Chcemy, żeby aplikacja była prosta i szybka w obsłudze.

- **Bezpieczeństwo danych:**
  Chcemy, żeby dane były dobrze chronione, bo będą tu przechowywane prywatne informacje studentów. Może jakieś szyfrowanie?

- **Offline mode:**
  Dobrze by było, gdyby część funkcji działała offline (np. podgląd planu zajęć lub ocen).

2. Określenie wymagań szczegółowych

2.1. Cel aplikacji

Celem aplikacji jest ułatwienie studentom oraz pracownikom uczelni dostępu do kluczowych funkcji administracyjnych i informacyjnych związanych z edukacją. Aplikacja ma zastąpić tradycyjne interakcje z dziekanatem, umożliwiając szybki dostęp do ocen, planu zajęć, harmonogramu egzaminów, e-dokumentów oraz ułatwiając komunikację z administracją uczelni.

- **Zakres aplikacji:**
  Studenci: dostęp do ocen, planu zajęć, harmonogramu egzaminów, komunikacja z dziekanatem.
  Wykładowcy/Pracownicy: dostęp do harmonogramu zajęć, oceny studentów, komunikacja z dziekanatem.
  Administracja: zarządzanie danymi, kontakt ze studentami.

- **Platformy:**
  Systemy operacyjne:
  Android,
  iOS.

- **Dane wejściowe: (Logowanie użytkowników)**
  Dane studentów: nr indeksu, rocznik, oceny, plan zajęć, egzaminów, zgłoszenia do dziekanatu.
  Zdarzenia dziekanatu: zmiany w planie, ogłoszenia, nowe dokumenty, powiadomienia o ocenach.
  Formularze zgłoszeń: wnioski o zaświadczenia, prośby do dziekanatu.

- **Opis interfejsu użytkownika i elementów interaktywnych:**
  Ekran logowania:
  Pola tekstowe: „Email" oraz „Hasło".
  Przyciski: „Zaloguj" – po kliknięciu, autoryzacja danych w tle i przejście do ekranu głównego. W przypadku błędnych danych, komunikat „Niepoprawne dane logowania". „Zapomniałem hasła" – przekierowanie do formularza resetowania hasła.

- **Ekran główny (Dashboard):** Wyświetlane informacje: skrót do ocen, nadchodzących zajęć, powiadomienia o ważnych wydarzeniach.
  Przyciski:
  „Oceny" – po kliknięciu, przejście do ekranu z listą ocen.
  „Plan zajęć" – po kliknięciu, podgląd planu zajęć (interaktywny kalendarz).
  „Harmonogram egzaminów" – otwiera harmonogram egzaminów z możliwością filtrowania według przedmiotu/semestru.
  Automatyczne zdarzenia: wyświetlanie powiadomień push o nadchodzących zajęciach, ocenach, ważnych wydarzeniach.

- **Ekran ocen:**
  Tabela ocen: kolumny „Przedmiot", „Ocena", „Komentarz wykładowcy", „Data".
  Opcje sortowania: sortowanie ocen po przedmiocie, dacie.
  Zachowanie: po kliknięciu w przedmiot, otwarcie szczegółów przedmiotu (np. opis, prowadzący, historia ocen). Plan zajęć:

- **Widok kalendarza**: wyświetlanie zajęć na dany tydzień/dzień.
  Funkcje interaktywne:
  Zmiana tygodnia za pomocą przesuwania palcem (swipe left/right).
  Kliknięcie na zajęcia otwiera szczegóły, np. nazwisko wykładowcy, sala, godziny.
  Powiadomienia push: automatyczne przypomnienia o nadchodzących zajęciach z możliwością wyłączenia.

- **Harmonogram egzaminów**:
  Lista egzaminów: możliwość filtrowania według przedmiotu, prowadzącego, daty.
  Opcja zapisu: jeśli wymagana rejestracja na egzamin, po kliknięciu „Zapisz się" użytkownik zapisuje się na egzamin.
  Powiadomienia push: przypomnienia o zbliżających się egzaminach. Komunikacja z dziekanatem:

- **Zdarzenia automatyczne**:
  Aplikacja będzie automatycznie wysyłać powiadomienia push o zmianach w planie zajęć, wynikach egzaminów, nowo dodanych dokumentach i ogłoszeniach.
  Automatyczne aktualizacje planu zajęć oraz harmonogramu egzaminów po synchronizacji z serwerem (np. co 30 minut).

- **Działanie offline**:
  Gdy brak połączenia z internetem, użytkownik ma dostęp do zapisanych wcześniej danych (plan zajęć, oceny).

- **Zachowanie aplikacji w niepożądanych sytuacjach**:
  Brak połączenia z internetem:
  Aplikacja wyświetla komunikat „Brak połączenia. Sprawdź połączenie z internetem" przy próbie wykonania operacji wymagającej synchronizacji z serwerem (np. wysłanie wiadomości do dziekanatu). W trybie offline: dostęp do zapisanych danych, brak możliwości interakcji z funkcjami wymagającymi połączenia.
  Błędne dane logowania:
  Komunikat „Niepoprawny email lub hasło" oraz możliwość ponownego wpisania danych. Pole tekstowe zostaje podświetlone na czerwono.
  Serwer niedostępny:
  Wyświetlenie komunikatu „Serwer dziekanatu jest obecnie niedostępny. Spróbuj ponownie później".
  Niepoprawne działanie aplikacji:
  W przypadku błędu technicznego aplikacja wyświetli komunikat „Wystąpił błąd. Spróbuj ponownie" i zapisze logi błędów do późniejszej analizy przez programistów.

3. Projektowanie

W ramach przygotowania środowiska do implementacji aplikacji wirtualnego dziekanatu oraz zarządzania wersjami kodu, wybrano zestaw narzędzi wspierających proces tworzenia oraz zapewniających automatyzację wielu czynności. W poniższych punktach opisano każde z wykorzystanych narzędzi wraz z ich rolą oraz załączonym linkiem do dokumentacji.

3.1. Przygotowanie narzędzi

- **Git** – system kontroli wersji, umożliwiający śledzenie zmian w kodzie oraz współpracę w zespole. Dokumentacja narzędzia: https://gitscm.com/doc
- **VSCode** – edytor kodu źródłowego, który zapewnia wsparcie dla wielu języków programowania i umożliwia instalację rozszerzeń wspierających programowanie. Dokumentacja narzędzia: https://code.visualstudio.com/docs
- **Doxygen** – narzędzie do generowania dokumentacji automatycznej na podstawie komentarzy w kodzie źródłowym. Ułatwia utrzymywanie aktualnej dokumentacji technicznej. Dokumentacja narzędzia: https://www.doxygen.nl/
- **Doxygen Awesome** – motyw graficzny dostosowujący wygląd strony wygenerowanej przez Doxygen do współczesnych standardów. Motyw jest zainspirowany stroną Nuxt i pomaga poprawić czytelność dokumentacji. Więcej informacji: https://github.com/jothepro/doxygenawesomecss
- **Lefthook** – narzędzie do zarządzania hookami Git, które wspiera automatyczne formatowanie, walidację kodu, generowanie dokumentacji oraz zgodność wiadomości commitów z konwencją. Dokumentacja narzędzia: https://github.com/evilmartians/lefthook
- **Commitlint** – narzędzie do sprawdzania zgodności wiadomości commitów z konwencją *Conventional Commits*. Dokumentacja: https://commitlint.js.org/
- **GitHub Actions** – platforma do automatyzacji procesów CI/CD. Umożliwia między innymi automatyczną walidację commitów, generowanie dokumentacji oraz wersjonowanie wydań. Dokumentacja: https://docs.github.com/en/actions

3.2. Wybór technologii

3.3. Framework Flutter

Flutter to framework stworzony przez Google, umożliwiający tworzenie aplikacji wieloplatformowych z jednego kodu źródłowego, co znacząco skraca czas potrzebny na rozwój aplikacji. W projekcie wirtualnego dziekanatu Flutter został wybrany ze względu na jego elastyczność oraz bogaty ekosystem widgetów. Poniżej przedstawiono kluczowe aspekty wykorzystania Fluttera w projekcie.

- **Material Design** – Flutter dostarcza szeroką gamę komponentów zgodnych z wytycznymi Material Design, co pozwala na stworzenie nowoczesnego i spójnego interfejsu użytkownika, dostosowanego do standardów Google. Dokumentacja: https://flutter.dev/docs/development/ui/widgets/material
- **Hot Reload** – Flutter umożliwia szybkie testowanie zmian w kodzie dzięki funkcji Hot Reload, która natychmiast odświeża widoki aplikacji, co pozwala programistom na szybką iterację i oszczędność czasu w trakcie testowania.
- **Kompatybilność wieloplatformowa** – Aplikacja wirtualnego dziekanatu została zaprojektowana jako aplikacja mobilna, ale Flutter umożliwia łatwe rozszerzenie wsparcia na inne platformy, takie jak web, desktop (Windows, macOS, Linux) oraz urządzenia IoT.
- **State Management (Zarządzanie stanem)** – W projekcie zastosowano zarządzanie stanem z użyciem pakietu Provider, co umożliwia łatwe zarządzanie danymi oraz stanami ekranów, szczególnie w dynamicznych sekcjach, takich jak ekran wiadomości czy plan zajęć. Dokumentacja Provider: https://pub.dev/packages/provider
- **Bogaty ekosystem pakietów** – Flutter wspiera szeroką gamę pakietów dostępnych w repozytorium `pub.dev`, co pozwala na szybkie wdrożenie dodatkowych funkcji. Przykłady wykorzystanych pakietów to:
  - **firebase_auth** – zapewnia integrację z Firebase Authentication dla bezpiecznego logowania użytkowników. Dokumentacja: https://pub.dev/packages/firebase_auth
  - **cloud_firestore** – pozwala na połączenie z bazą danych Firestore i zarządzanie danymi w czasie rzeczywistym. Dokumentacja: https://pub.dev/packages/cloud_firestore

  - **firebase_messaging** – umożliwia wysyłanie powiadomień push do użytkowników. Dokumentacja: https://pub.dev/packages/firebase_messaging
  - **intl** – używany do formatowania dat i liczb, co wspiera różne lokalizacje i języki. Dokumentacja: https://pub.dev/packages/intl

- **Wysoka wydajność** – Flutter jest bezpośrednio kompilowany do natywnego kodu ARM, co zapewnia wydajność zbliżoną do natywnych aplikacji. Dla płynnego działania aplikacji kluczowe było odpowiednie zarządzanie wydajnością komponentów i optymalizacja ekranów, szczególnie dla list z dużą ilością danych, jak np. plan zajęć.

- **Testy jednostkowe i widgetowe** – Flutter oferuje rozbudowane wsparcie dla testów, co umożliwia testowanie logiki biznesowej aplikacji (testy jednostkowe) oraz interakcji użytkownika z komponentami UI (testy widgetowe). Dzięki temu można łatwo wykrywać błędy i sprawdzać działanie aplikacji w sposób automatyczny.

Funkcjonalności backendowe zapewnia platforma Firebase, dostęp do nich umożliwiają moduły:

- **Firebase Auth** – moduł autoryzacji użytkowników. Dokumentacja: https://firebase.google.com/docs/auth

- **Firestore** – baza danych czasu rzeczywistego, umożliwiająca skalowalne przechowywanie danych aplikacji. Dokumentacja: https://firebase.google.com/docs/firestore

- **Firebase Messaging** – moduł do wysyłania powiadomień push do użytkowników. Dokumentacja: https://firebase.google.com/docs/cloudmessaging

4. Implementacja

5. Testowanie

6. Podręcznik użytkownika

Spis rysunków

Spis tabel

Spis listingów