Dokumentacja projektu z przedmiotu Programowanie urządzeń mobilnych pt.: „Plan zajęć UR”

Adrian Jonarski
106445

18 stycznia 2021

Uniwersytet Rzeszowski
Kolegium Nauk Przyrodniczych
Informatyka (I st.), semestr 5

1 Opis projektu

Projekt z przedmiotu Programowanie urządzeń mobilnych miał na celu stworzenie aplikacji mobilnej do prezentacji planu zajęć uczelni wyższej.

2 Wykorzystane technologie

Aplikacja „Plan zajęć UR" została wykonana w technologii Flutter z użyciem języka Dart. W/w aplikacja do przechowywania danych korzysta z bazy danych SQLite.

3 Implementacja

3.1 main.dart

Głównym plikiem aplikacji jest plik *main.dart*. Tworzona jest tam nawigacja aplikacji oraz deklarowany jest jej wygląd w trybie *light* i *dark*.

Fragment kodu (nawigacja po aplikacji):

```
initialRoute: '/',
  routes: {
    '/': (context) => MyHomePage(),
    '/college': (context) => College(),
    '/course': (context) => Course(),
    '/timetable': (context) => Timetable(),
  },
```

3.2 home.dart

W pliku *home.dart* znajduje się odwołanie do klasy *GridDasboard*, która znajduje się w pliku *griddashboard.dart*.

3.3 griddashboard.dart

W/w plik odpowiada za wyświetlanie kolegiów z listy. Po kliknięciu w wybrany „kafelek" wyświetlana jest lista kierunków studiów z danego kolegium (plik: *college.dart*).

3.4 college.dart

W pliku tym wywoływana jest m.in. funkcja *fetchCourses*, która odpowiada za przypisywanie do listy kierunków studiów z obiektów typu *future*.

Fragment kodu (funkcja *fetchCourses*):

```
void fetchCourses(String collegeLetter) async {
    setState(() => isLoading = true);
    final tmpList = await DatabaseProvider.db.
        getAllCourses(collegeLetter);
    setState(() {
      isLoading = false;
      coursesList = tmpList;
    });
  }
```

Po kliknięciu w kierunek na liście, wyświetlana jest lista roczników dla danego kierunku (plik: *course.dart*).

3.5 course.dart

W tym pliku m.in. wywoływana jest funkcja asynchroniczna przekazująca listę roczników z obiektów typu Future. Po kliknięciu w odpowiedni rocznik następuje przekierowanie do pliku *timetable.dart*

3.6 timetable.dart

Plik *timetable.dart* odpowiada za wyświetlanie planu zajęć.

3.6.1 ToggleButtons

Widget ToggleButtons odpowiada za przyciski do wyboru tygodnia A lub B.

Fragment kodu (widget *ToggleButtons*):

```
\child:  ToggleButtons(
              isSelected: isSelected,
              selectedColor: Colors.white,
              fillColor: Colors.blue[900],
              borderColor: Colors.blue[900],
              selectedBorderColor: Colors.blue[900],
              children: <Widget>[
                Padding(
                  padding: const EdgeInsets.symmetric(
                      horizontal: 60),
                  child Text('Tydzien A'),
                ),
                Padding(
                  padding: const EdgeInsets.symmetric(
                      horizontal: 60),
                  child: Text("Tydzien B"),
                ),
              ],
```

```
            onPressed: (int newIndex) {
              setState(() {
                if (isSelected[newIndex] == false){
                for (int index = 0; index < isSelected.
                    length; index++) {
                  if (index == newIndex) {
                    isSelected[index] = true;
                  } else {
                    isSelected[index] = false;
                  }
                }
                String tmp = week1;
                week1 = week2;
                week2 = tmp;
                fetchLessons(idC, year, week1);
              }
            });
            },
          ),
```

3.6.2 StickyGroupedListView

Widget *StickyGroupedListView* odpowiada za wyświetlanie planu zajęć pogrupowanego dniami wg odpowiedniej kolejności. Wykorzystywany jest pakiet *sticky_grouped_list 1.3.0*

Fragment kodu (widget *StickyGroupedListView*):

```
return (StickyGroupedListView<dynamic, String>(
      elements: lessonsList,
      groupBy: (lessons) {
        return lessons.id_dnia.toString();
      },
      groupSeparatorBuilder: (lessons) => Text(
        lessons.dzien_tygodnia,
        textAlign: TextAlign.center,
        style: TextStyle(fontSize: 18, fontWeight:
            FontWeight.bold),
      ),
      floatingHeader: true,
...
```

3.7 database_provider.dart

Klasa modeli *database_provider.dart* jest to klasa, która odpowiada danym otrzymywanym z bazy danych SQLite. Posiada parsery, które konwertują dane na

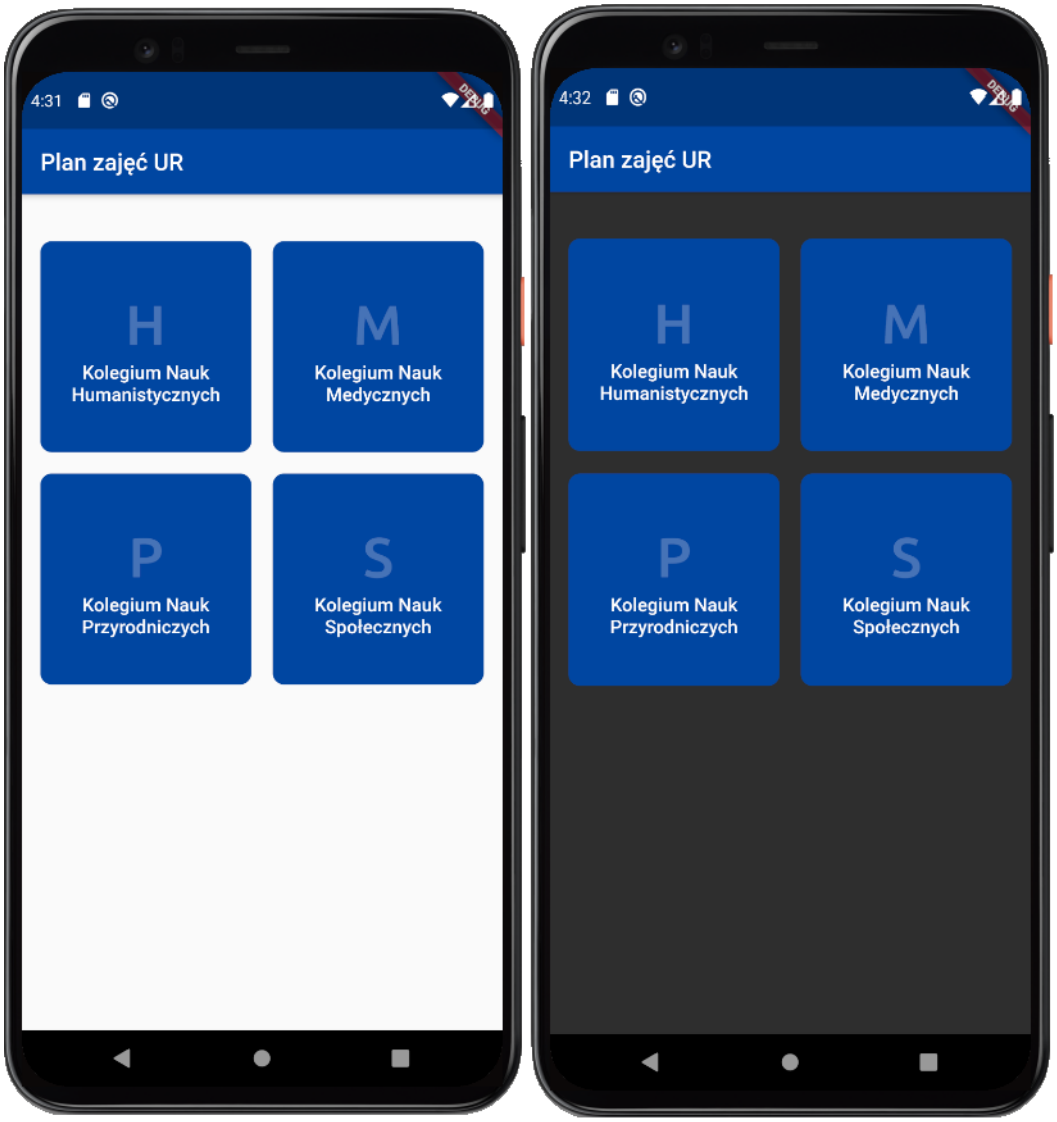
odpowiadające im pola w języku Dart. W/w plik zawiera odpowiednie funkcje do pobierania danych z bazy danych, takie jak: pobieranie wszystkich kierunków z danego kolegium (*getAllCourses*), pobieranie roczników z danyego kierunku (*getYears*), pobieranie zajęć dla danego rocznika (*getLessons*) oraz pobieranie dni (*getDays*).

Fragment kodu (funkcja *getAllCourses*):

```
Future<List<Courses>> getAllCourses(college) async {
    final db = await database;
    var response = await db.rawQuery('SELECT * FROM
        kierunki WHERE id_kierunki LIKE ?', ['%$college
        %']);
    List<Courses> list = response.map(
            (s) => Courses.fromMap(s)
    ).toList();
    db.close();
    return list;
  }
```

4 Podsumowanie

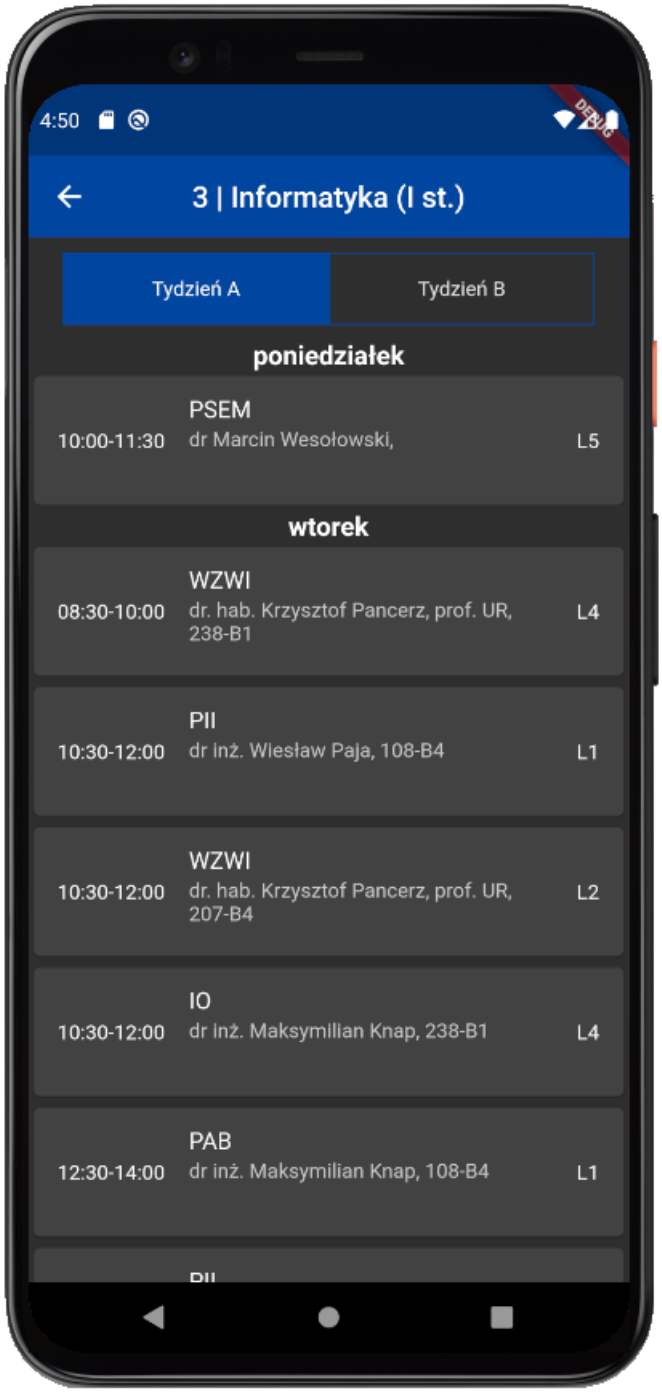
Użytkownik aplikacji po uruchomieniu zobaczy listę kolegiów w formie czterech „kafelków". Po kliknięciu w kolegium, użytkownik zobaczy listę kierunków. Następnie po wybraniu kierunku, wyświetlana jest lista roczników. Po wybraniu rocznika, wyświetlany jest plan zajęć, domyślnie zawsze dla tygodnia A (1/3). Aby zobaczyć pełną nazwę przedmiotu, należy kliknąć w dane zajęcia i zostanie wyświetlony *SnackBar*.



Rysunek 1: Widok główny w trybie jasnym i ciemnym.

Rysunek 2: Widok listy kierunków.

Rysunek 3: Widok listy roczników.



Rysunek 4: Widok zajęć.