



BOGDAN KWIATKOWSKI¹, TOMASZ BINKOWSKI²

Projektowanie i zastosowanie aplikacji mobilnej MyCollection za pomocą dostępnych narzędzi programistycznych

Design and use of mobile application MyCollection using available development tools

¹ Doktor inżynier, Uniwersytet Rzeszowski, Wydział Matematyczno-Przyrodniczy, Katedra Inżynierii Komputerowej, Polska

² Doktor inżynier, Politechnika Rzeszowska, Katedra Energoelektroniki, Elektroenergetyki i Systemów Złożonych, Polska

Streszczenie

Celem artykułu jest przedstawienie procesu projektowania aplikacji mobilnych przeznaczonych dla systemu operacyjnego Android oraz stworzenie takiej aplikacji mobilnej mającej za zadanie łatwe gromadzenie informacji ważnych dla danego użytkownika, dzielenie ich na własne kategorie oraz udostępnianie ich znajomym. Projekt został zrealizowany za pomocą środowiska programistycznego Android Studio.

Słowa kluczowe: aplikacja mobilna, Android, MySql, PHP.

Abstract

Aim of this article was to create mobile application, which gives user a possibility to collect for example his favorite book titles, films or visited places. Application allows to add photos and contents that are connected with them. For this propose was used technologies like: Android Studio, PHP and MySQL.

Key words: Mobile application, Android, MySql, PHP.

Wstęp

Każdy z nas zauważa, jak w ciągu ostatnich lat wzrosła liczba użytkowników urządzeń mobilnych, takich jak smartfony czy tablety. Coraz więcej osób zamiast korzystać z laptopów czy komputerów stacjonarnych woli schować sobie do kieszeni telefon, z którego będzie mógł mieć dostęp do każdej interesującej go dziedziny życia. Telefony i tablety dawno przestały już być po prostu urządzeniami do wykonywania połączeń telefonicznych

bądź przeglądania stron internetowych. Wiele osób traktuje je jak narzędzie swojej pracy na równym poziomie, co kiedyś laptopy. Urządzenia te służą użytkownikom również w dużej mierze do przechowywania osobistych dokumentów czy zdjęć. Każdy z nas lubi mieć swoje ulubione czy osobiste rzeczy przy sobie. Szczególnie, jeśli uda się je wszystkie zamknąć w małym urządzeniu, jakim jest tablet czy smartfon, który może nam umilić długą podróż, nudne spotkanie biznesowe czy wręcz przeciwnie, byliśmy na wyśmienitym obiedzie i chcielibyśmy mieć numer telefonu, adres oraz zdjęcie restauracji, by móc ją polecić znajomym bez zbędnego przeszukiwania internetu. W tym celu właśnie powstała aplikacja MyCollection, która pozwala na przechowywanie i dodawanie swoich ulubionych informacji. Celem tej publikacji jest przedstawienie kolejnych etapów powstawania aplikacji mobilnej przygotowanej dla systemu operacyjnego Android.

MyCollection – główne zastosowanie aplikacji

Aplikacja MyCollection ma za zadanie umożliwić użytkownikom kolekcjonowanie dokumentów lub interesujących wiadomości w poszczególnych kategoriach stworzonych przez niego wedle uznania. Jedną z funkcji aplikacji jest tworzenie drzewa kategorii, dzięki którym będzie przejrzysty dostęp do interesujących nas treści. Ponadto do każdej wiadomości jest możliwość dodania zdjęcia, co wzbogaca daną treść. Kolejną z opcji jest dodawanie adnotacji do ulubionych, gdzie wszystkie oznaczone elementy pojawią się w zakładce „ulubione”. Za pomocą zakładki „szukaj” użytkownik po wpisaniu danego słowa będzie mógł wyszukać konkretny tytuł wiadomości. Wszystkie dodane treści, jak i budowa kategorii trafiają na zewnętrzny serwer, dzięki czemu każdy ma pewność, że cała zawartość nie zostanie utraczona, chociażby w przypadku uszkodzenia urządzenia.



Rys. 1. Logo aplikacji MyCollection, menu głównej aplikacji

Źródło: opracowanie własne.

Aplikacja została zaprojektowana w środowisku programistycznym Android Studio¹. Jest ono nowym IDE (zintegrowanym środowiskiem projektowym) udostępnionym przez Google dla developerów Androida. Jest wspierane i stale rozwijane przez giganta z branży internetowej. Dzięki temu mamy pewność, że zawsze korzystamy z najnowszych wersji API, bibliotek i narzędzi służących do tworzenia profesjonalnych aplikacji. Android Studio umożliwia wygodne programowanie aplikacji na urządzenia mobilne, takie jak: smartfony, tablety, zegarki, okulary czy też do samochodów i telewizorów³. W prosty sposób możemy zaprojektować, stworzyć oraz sprawdzić na specjalnym emulatorze, jak działa tworzony przez nas projekt. Środowisko programistyczne Android Studio zostało stworzone, aby móc tworzyć oprogramowanie na system operacyjny Android, który w ostatnich latach opanował rynek związany z urządzeniami mobilnymi oraz aplikacjami dla nich. Jego historia ma swój początek w 2003 r. na terenie Kalifornii, gdzie narodziła się mała spółka Android Inc. Wzbudziła zainteresowanie Google, które ostatecznie kupiło ją rok później. W 2007 r. założono Open Handset Alliance, w skład którego wchodzi m.in. takie potęgi, jak: Google, HTC, Intel, Motorola, T-mobile czy Nvidia. Do tej pory wyprodukowano 12 oficjalnych wersji Androida⁵. Najpopularniejszymi jego wersjami na rynku są: KitKat, Lollipop oraz Lollipop².

Technologie zastosowane przy tworzeniu aplikacji

W celu wykonania aplikacji MyCollection niezbędne było wykorzystanie technologii niebędących częścią środowiska programistycznego. Ponadto wykorzystano opcjonalny program w celu zobrazowania oraz usprawnienia pracy. W tym punkcie zostaną przedstawione technologie, jakie zostały wykorzystane przy tworzeniu aplikacji.

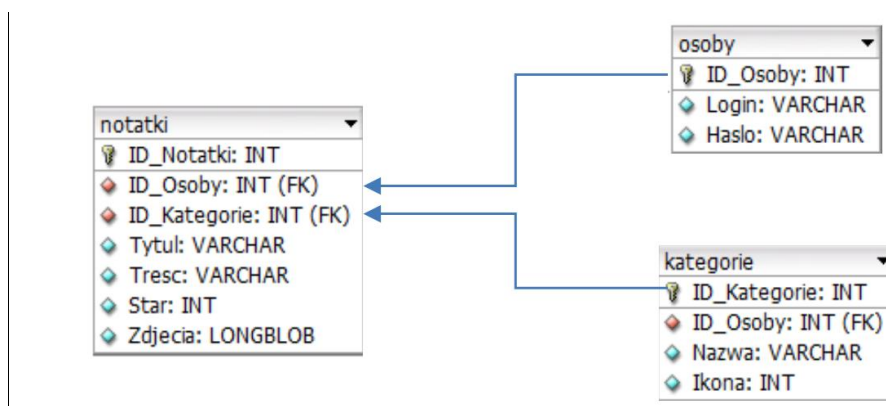
Pencil Project

Tworzenie aplikacji powinno rozpocząć się od przygotowania podstawowego schematu działania programu, jak również jego graficznego interfejsu, który powinien być maksymalnie funkcjonalny i przejrzysty dla użytkownika. W tym celu, skorzystano z darmowego narzędzia Pencil Project. Pencil służy do przygotowania interfejsu aplikacji według własnego uznania. Stwarza możliwość utworzenia prototypu GUI (z ang. *graphical user interface*), czyli interfejsu graficznego na popularne platformy mobilne, tj. Android, IOS. Program posiada bogatą bazę kształtów oraz elementów – przyciski, paski ładowania, statusy urządzeń umożliwiające tworzenie projektów. Znajdziemy tu również wysokiej jakości kształty wykorzystywane przez platformę Android. Dodatkowo na stronie programu jest wiele kształtów i elementów GUI przeznaczonych do tworzenia interfejsów innych platform. Korzystanie z tego narzędzia jest bardzo przyjemne i proste. W zakładce „kolekcja” znajdują się wszystkie elementy, jakie

można zamieścić na pustym polu. W ten oto sposób powstaje graficzny interfejs. Zastosowanie tego narzędzia dało możliwość efektywniejszego i znacznie prostszego tworzenia wstępnego interfejsu aplikacji.

Baza danych – MySQL

Bazy danych odgrywają istotną rolę we wszystkich sektorach gospodarczych. Są to zbiory danych zapisane zgodnie z zasadami przyjętymi dla poszczególnego programu komputerowego, istniejące w długim okresie, często nawet przez wiele lat. Bazy danych można podzielić na bazy proste: hierarchiczne i kartotekowe oraz bazy złożone: relacyjne, obiektowe, strumieniowe, temporalne, nierelacyjne⁴. Jednym z popularniejszych modeli baz danych jest relacyjny model danych. W modelu tym dane reprezentowane są w dwuwymiarowej tabeli, która nazywa się relacją. Każda tabela posiada swój klucz główny, dzięki któremu może tworzyć połączenia z innymi tabelami.



Rys. 2. Model bazy danych aplikacji MyCollection

Źródło: opracowanie własne.

Etapy tworzenie aplikacji

Wykorzystując wcześniej stworzony interfejs w programie Pencil Project oraz utworzywszy bazy danych przechowujące niezbędne dane, przystąpiono do tworzenia aplikacji. W tym punkcie zostaną przedstawione etapy implementowania poszczególnych skryptów i sposób połączenia programu z bazą danych. Biorąc pod uwagę wszystkie zalecenia sugerowane przez firmę Google, przygotowano wstępny i końcowy interfejs aplikacji. W aplikacji zastosowano szablon TabbedActivity, dostępny z poziomu Android Studio³. Dzięki niemu użytkownik ma możliwość zmiany widoków pomiędzy sąsiadującymi oknami za pomocą poziomego przesunięcia palca po ekranie. Poszczególne zakładki tego szablonu opierają swoje działania na osobno tworzonych fragmentach. Każdy fragment

w stworzonej aplikacji odpowiada za inne funkcjonalności. Wczytywanie poszczególnych fragmentów odbywa się za pomocą funkcji `setupViewPager`. Wewnątrz niej stworzony został obiekt `ViewPagerAdapter`, w którym zostały dodane kolejne fragmenty wraz z ich nazwami⁶.

```
...
private void setupViewPager(ViewPager viewPager) {
    ViewPagerAdapter adapter = new
    ViewPagerAdapter(getSupportFragmentManager());
    adapter.addFragment(new Fragment1(), "Menu");
    adapter.addFragment(new Fragment2(), "Ulubione");
    adapter.addFragment(new Fragment3(), "Szukaj");
    viewPager.setAdapter(adapter);
}
...
```

W domyślnym ustawieniu poszczególne zakładki `TabbedView` posiadają nazwy w formie tekstu. W celu uatrakcyjnienia wyglądu nazwy te zostały zastąpione ikonami. Ikony znajdują się w projekcie w folderze `drawable`. Aby móc się do nich odwołać, została stworzona tablica `tabIcons`. Do ustawienia ikon użyto funkcji `setupTabIcons`. Na każdej zakładce wywołano `setIcon` zawierającą parametr indeksu tablicy z ikonami.

```
private void setupTabIcons() {
    tabLayout.getTabAt(0).setIcon(tabIcons[0]);
    tabLayout.getTabAt(1).setIcon(tabIcons[1]);
    tabLayout.getTabAt(2).setIcon(tabIcons[2]);
}
```

Głównym elementem aplikacji jest rozwijana lista (z ang. *expandable list*). Różni się ona od zwykłej listy tym, że daje możliwość rozwijania jej obiektów. W każdym z nich znajdują się kolejne listy nierozwijane `ListView`. W celu uzupełnienia listy niezbędne było utworzenie adaptera `ExpandableListAdapter`. Składa się on z dwóch obiektów. Jednym z nich jest `parentItems`, który odpowiada za wyświetlanie nagłówków listy, natomiast drugi, o nazwie `childItems`, przechowuje ich zawartość w postaci listy.

```
...
public class ExpandableListAdapter extends BaseExpandableListAdapter {
    private final LayoutInflater inf; private final ArrayList<Object>
    childItems; private final ArrayList<String> parentItems; public
    ExpandableListAdapter(ArrayList<String> parents, ArrayList<Object>
    children) { this.parentItems = parents; this.childItems = children;
    inf = LayoutInflater.from(getActivity()); }
    ...
```

Kolejnym atutem `ExpandableListView` jest możliwość dodania ikony przy nagłówku. Każdy nagłówek posiada swoją własną ikonę, a informacja o tym jest przechowywana w bazie danych. Każdy użytkownik indywidualnie definiuje nagłówki wraz z ich ikonami. W poniżej umieszczonym kodzie przedstawione

jest ustawienie obrazka poprzez `setImageDrawable`. Dodatkowo `ListView` posiada opcję zaznaczenia elementów, jako ulubione poprzez użycie `checkbox` w formie gwiazdki. Po kliknięciu ikona zostaje podświetlona, a informacja o wprowadzeniu zmiany w aplikacji skutkuje aktualizacją w bazie danych. Poniższy kod odpowiada za przechwytywanie kliknięcia na `checkbox` oraz wykonanie zmiany poprzez funkcję `add_favorite` z odpowiednim parametrem. Po rozwinięciu nagłówka `ExpandableList` ukazuje się lista obiektów `ListView`. Podrzędna lista posiada dodatkową funkcjonalność. Jest nią możliwość szczegółowego podglądu danego elementu oraz jego edycji. Poniższy kod odpowiada za przechwytywanie kliknięć na `ListView` oraz przejścia do nowego activity wyświetlającego zawartość obiektu listy. Aby aplikacja miała możliwość połączenia z zewnętrzną bazą danych niezbędne było wykorzystanie odpowiednich skryptów oraz zezwoleń. W celu nawiązania połączenia pomiędzy aplikacją a bazą danych użyto skryptów PHP. Skrypty te mają za zadanie połączenia z bazą oraz wykonywanie zapytań. Uruchamianie skryptów jest możliwe z wykorzystaniem pakietów Apache `HttpComponents`. Niezbędne jest, aby każde zapytanie posiadało parametry przekazane z aplikacji. Do tego użyto listy obiektów `NameValuePair`⁷.

```
...
InputStream iss = null; @Override public void onClick(View v) {
String login = LoginUser.getText().toString(); String haslo =
LoginPass.getText().toString(); List<NameValuePair> nameValuePairs
= new ArrayList<NameValuePair>(1); nameValuePairs.add(new Basic-
NameValuePair("login", login)); nameValuePairs.add(new Basic-
NameValuePair("haslo", haslo)); try { HttpClient httpClient = new
DefaultHttpClient(); HttpPost httpPost1 = new Http-
Post("http://mycollections.net16.net/login.php"); http-
Post1.setEntity(new UrlEncodedFormEntity(nameValuePairs)); HttpRe-
sponse httpResponse = httpClient.execute(httpPost1); HttpEntity
httpEntity = httpResponse.getEntity(); iss = httpEnti-
ty.getContent();
...

```

Przedstawiony kod uruchamia skrypt PHP logowania z podaniem jako parametrów logina oraz hasła wprowadzonego uprzednio przez użytkownika.

Konstrukcja skryptu PHP składa się z kilku etapów. Pierwszym z nich jest nawiązanie połączenia z serwerem bazy danych z uwzględnieniem nazwy użytkownika i hasła. Kolejną czynnością jest wybór bazy danych poprzez funkcję `mysql_select_db`. Następnie zostały pobrane parametry przekazane z aplikacji. Parametrów tych użyto przy wykonaniu zapytania w funkcji `mysql_query`. Wynik zapytania zostaje przekonwertowany do formatu json i zwrócony do aplikacji. Jedną z funkcji, która została wdrożona do projektu, jest możliwość dodawania nowych kategorii z opcją wyboru nazwy oraz ikony. W tym celu utworzono odrębny layout activity `new_category` oraz klasę `NewCategory`.

Podsumowanie

Dzięki zastosowaniu takich technologii, jak: Android Studio, PHP oraz MySQL, udało się stworzyć aplikację mobilną polegającą na tworzeniu przez jej użytkownika dwupoziomowych bibliotek służących do kolekcjonowania informacji czy ciekawostek w jednym wybranym przez użytkownika miejscu. Dzięki aplikacji osoba posiadająca aplikację ma możliwość przechowywania informacji dla niej ważnych bez konieczności przeszukiwania internetu za każdym razem od nowa. Użytkownik nie musi martwić się np. o to, że zapomni nazwy restauracji, którą niedawno odwiedził i przez to nie będzie mógł znaleźć do niej numeru telefonu. Wszystkie informacje, które zamieści w aplikacji, będą dostępne na zewnętrznym serwerze i kiedy zaloguje się do MyCollection na jakimkolwiek urządzeniu mobilnym, będzie miał dostęp do zawartości. Ponadto aplikacja jest bardzo łatwa w obsłudze, dlatego nawet osoba najmniej zaawansowana w użytkowaniu aplikacji mobilnych będzie w stanie poradzić sobie, korzystając z niej.

Aplikacja ma możliwość dalszego rozwoju i poszerzenia jej funkcjonalności np. o bezpośrednie odsyłanie do linków w nich umieszczonych czy wyszukiwanie użytkowników MyCollection o podobnych zainteresowaniach, co umożliwi dzielenie się własnymi kolekcjami.

Literatura

- Cruz Zapata B. (2015), *Android Studio. Podstawy*, Gliwice.
- Delessio C., Darcey L., Conder S. (2015), *Android Studio w 24 godziny. Wygodne programowanie dla platformy Android*, Gliwice.
- Gerber A., Craig C. (2015), *Android Studio. Wygodne i efektywne tworzenie aplikacji*, Gliwice.
- <http://developer.android.com/sdk/index.html>.
- <http://www.oracle.com/technetwork/database/mysql/index.html>.
- https://www.android.com/intl/pl_pl/history/.
- Wilton P., Colby J., *SQL. Od podstaw*, Gliwice.