Uniwersytet Jagielloński

Wydział Matematyki i Informatyki

INSTYTUT INFORMATYKI I MATEMATYKI KOMPUTEROWEJ

Studia dzienne.

Numer indeksu: 1091582

**Kamil Paszkiewicz**

**Platforma do nawiązywania kontaktów "Poznawajka Mobile". Projekt i implementacja warstwy bazodanowej oraz projekt algorytmów dopasowywania na podstawie preferencji.**

Opiekun pracy licencjackiej:

Dr Marcin Żelawski

Opracowano zgodnie z Ustawą o prawie autorskim i prawach pokrewnych z dnia 4 lutego

1994 r. (Dz.U. 1994 nr 24 poz. 83) wraz z nowelizacją z dnia 25 lipca 2003 r. (Dz.U. 2003 nr

166 poz. 1610) oraz z dnia1 kwietnia 2004 r. (Dz.U. 2004 nr 91 poz. 869)

**Kraków 2015**

**Abstrakt**

Celem pracy jest stworzenie projektu wraz z modelem warstwy bazodanowej,   
a także projektów algorytmów dopasowywania na podstawie preferencji   
w platformie do nawiązywania kontaktów „Poznawajka Mobile”. Opisywany projekt jest skierowany na urządzenia mobilne z systemem Android.

Target of my Bachelor's thesis is create project with model and implementation of database layer and projects of matching algorithms from preferences. This project is on Android mobile devices.

**Spis treści**

Rozdział I: Wstęp

Rozdział II : Charakterystyka funkcjonalna i wymagania systemu „Poznawajka Mobile”

Rozdział III : Warstwa bazodanowa, języki UML, SQL i algorytmika.

Rozdział IV : Diagramy UML oraz opis funkcjonalny warstwy bazodanowej w systemie „Poznawajka Mobile"

Rozdział V : Opis i prezentacja algorytmu dopasowywania względem preferencji

Rozdział VI : Specyfikacja techniczna

Rozdział VII : Podsumowanie

Literatura

**Rozdział I : Wstęp**

Celem mojej pracy będzie przeanalizowanie warstwy bazodanowej oraz jej projekt, a także projekt algorytmów dopasowywania znajomości na podstawie preferencji w platformie do nawiązywania kontaktów „Poznawajka Mobile”. Napiszę również o języku UML,SQL oraz krótko o bazach danych i algorytmach.

W swojej pracy pokażę jaką ważną rolę odgrywa baza danych w prawie każdej aplikacji mobilnej dostępnej na rynku. Postaram się krótko i bardzo ogólnie przedstawić wszystkie funkcjonalności jakie będzie umożliwiać tworzona przeze mnie wyżej wymieniona platforma. Ponadto określę sposób w jaki będzie się komunikować z bazą danych oraz przeanalizuję ją pod kątem wydajności.

Każda większa aplikacja korzysta z bazy danych. Jest to najlepszy sposób na przechowywanie różnego rodzaju informacji. Mowa tu zarówno o informacjach tajnych takich jak np. hasła czy inne ważne dla nas informacje, ale również o informacjach jawnych takich jak np. historia, posty, publikacje itp.

Stworzenie wydajnego modelu bazy danych, który będzie spełniał wszystkie założenia projektu może okazać się dużym wyzwaniem. Podczas procesu kreacji trzeba uwzględnić wielkość bazy, rodzaj informacji, jakie będzie przechowywać, częstotliwość jej używania oraz wiele innych ważnych komponentów.

Baza danych może być wbudowana w aplikację mobilną na telefonie, wtedy oczywiście dostęp do niej jest łatwiejszy i bezpieczniejszy. Jednak moja baza może znajdować się również na serwerze, z którym można się dowolnie komunikować. Taki sposób przechowywania informacji będzie realizowany w „PoznawajkaMobile”. Baza danych, która będzie znajdować się na serwerze będzie posiadała informacje o wszystkich użytkownikach. Dostęp do niej będzie przyznawany za pomocą uwierzytelnienia i autoryzacji, dlatego aplikacja będzie wymagała stałego dostępu do Internetu.

Na koniec wstępnie określę i opiszę treść każdego z rozdziałów zawartego w spisie treści. Rozdział pierwszy stanowi wstęp pracy. W rozdziale drugim opiszę dokładną specyfikację funkcjonalną „Poznawajka Mobile”, wszystkie możliwości jakie oferuje i czego można od niej wymagać. W części trzeciej zamieszczę ogólny rozdział o UML-u , SQL-u, bazach danych oraz o algorytmach. W rozdziale czwartym zaprezentuje stworzone przeze mnie diagramy UML, które dokładnie opiszą warstwę bazodanową w aplikacji.

**Rozdział II : Charakterystyka funkcjonalna**

„Poznawajka Mobile” to nowoczesna platforma służąca nawiązywaniu znajomości. Jej istotą będzie prostota oraz użyteczność. Wygodny interfejs graficzny ma wywierać na użytkowniku miłe odczucia. Aplikacja jest skierowana do osób samotnych w każdym wieku.

Zaraz po włączeniu aplikacji pojawi się ekran główny, który będzie wymuszał na konsumencie autoryzację. Dane każdego użytkownika będą chronione i prywatne. Nie będą one jednak przechowywane w telefonie, lecz na serwerze, dlatego też konieczna będzie autoryzacja, by serwer i aplikacja rozpoznawały eksploatatora i selekcjonowały wyświetlane treści. Loginem będzie e-mail. Z ekranu głównego można również przejść do formularza rejestracji oraz do formularza odzyskiwania hasła.

Po kliknięciu na odnośnik rejestracji nastąpi przeniesienie do formularza rejestracji. Wszystkie pola będą wymagane i walidowane przed wysłaniem do serwera. Użytkownik będzie musiał podać swoje imię, nazwisko, płeć, wiek, stan cywilny, preferencje, a także osobowość (przy czym preferencje i osobowość będą to pola wyboru z listy). Taki sposób znacznie ułatwi implementacje algorytmów dopasowywania w późniejszej fazie. Chodzi głównie o uniknięcie błędów leksykograficznych wpisanych przez użytkowników, itp.

Zaraz po popranym zalogowaniu konsument zostanie przeniesiony do ekranu zarządzania profilem, w którym będzie mógł kierować kontem, to znaczy edytować własną galerie oraz zdjęcie główne, zmieniać swoje dane, preferencje oraz kryteria wyglądu. Zostanie wyświetlona lista posiadanych znajomych oraz przynależne grupy. Znajduje się tam również przejście do wyszukiwarki, umożliwiającej odkrywanie nowych znajomości.

Wyszukiwanie znajomości będzie możliwe według określonych preferencjach selekcji. Mianowicie do wyboru będą 3 opcje:

* Szukanie przez preferencje podane w profilu. Odpowiedzialny będzie za to algorytm proponowania znajomości na podstawie preferencji. Osoby wyszukiwane będą sortowane według poziomu trafności określonego przez specjalny algorytm.
* Szukanie po lokalizacji. Będzie on dostępny tylko wtedy, gdy będzie włączona lokalizacja GPS. Wtedy kolejne wyświetlane osoby będą sortowane względem odległości od użytkownika.
* Totalnie losowo. Jest to sposób szukania, w którym nie będą wykorzystywane i brane pod uwagę żadne opcje. Przedstawione osoby jako wyniki będą całkowicie losowe.

Oprócz tego aplikacja będzie od czasu do czasu sugerować nowe znajomości i pokazywać okienko dialogowe, w którym będą znajdować się propozycje nowych kontaktów. Oprócz tego będzie istnieć możliwość przeglądania profili innych użytkowników. Na profilu innej osoby będzie dostęp do jej podstawowych informacji i zdjęcia głównego. Z poziomu jej profilu będziemy mieli możliwość:

* Zaproponowania spotkania użytkownikowi.
* Wystawienia oceny oraz komentarza.
* Propozycji wspólnej znajomości.

Podczas gdy inna osoba zaproponuje użytkownikowi spotkanie lub znajomość, aplikacja wyświetli okno dialogowe, w którym będzie dostęp do danych nadawcy oraz wyświetlony zostanie cel spotkania. Będzie ich kilka:

* Randka
* Pogawędka
* Spotkanie grupowe

Będzie można odrzucić to zaproszenie i powiadomić nadawcę o odmowie, albo zaakceptować propozycję i wymienić się danymi kontaktowymi oraz przekazać dowolną wiadomość dotyczącą spotkania.

Oprócz tego każdy użytkownik będzie mógł posiadać własną galerie zdjęć, którą będzie mógł udostępniać innym. Podczas udostępniania osoba, która dostała uprawnienia do przeglądu galerii zobaczy daną galerię na profilu właściciela.

Dodatkową funkcję będzie stanowić możliwość integracji profilu  
 z aplikacji z portalami społecznościowymi np. takimi jak Facebook.

**Rozdział III : Warstwa bazodanowa, języki UML, SQL i algorytmika.**

Rozdział ten rozpocznę od krótkiego wprowadzenia do wymienionych   
w tytule wyrażeń kluczowych. UML, czyli Unified Modeling Language to język wykorzystywany do tworzenia różnego rodzaju systemów informatycznych. Jest on używany wraz z jego graficzną reprezentacją. W języku UML wyróżnia się 14 różnych diagramów oraz 3 abstrakcyjne. Przedstawię niektóre z nich, które będę później wykorzystywał przy modelowaniu warstwy bazodanowej.

* **Diagram klas**. Ma na celu graficzne zaprezentowanie wszystkich klas stosowanych w projekcie. Pokazanie wszystkich zależności, hierarchii pomiędzy nimi oraz oddziaływania klas na siebie i współprace pomiędzy nimi. Klasy są przedstawiane wraz z atrybutami publicznymi/prywatnymi oraz metodami publicznymi/prywatnymi.
* **Diagram przypadków użycia**. Służy do modelowania funkcjonalności systemu. Przedstawia aktorów, przypadki użycia i zależność między nimi. Tworzy się go jeszcze przed powstaniem systemu informatycznego.
* **Diagram sekwencji.** Diagram sekwencji jest rodzajem diagramu interakcji, opisującym interakcje pomiędzy instancjami klasyfikatorów systemu w postaci sekwencji komunikatów wymienianych między nimi.
* **Diagram komunikacji.** Diagram komunikacji jest rodzajem diagramu interakcji, specyfikującym strukturalne związki pomiędzy instancjami klasyfikatorów biorącymi udział w interakcji oraz wymianę komunikatów pomiędzy tymi instancjami.
* **Diagram stanów.** Diagram używany przy analizie i projektowaniu oprogramowania. Pokazuje przede wszystkim możliwe stany obiektu oraz przejścia, które powodują zmianę tego stanu.
* **Diagram czynności.** Diagram czynności to graficzne przedstawienie sekwencyjnych i (lub) współbieżnych przepływów sterowania oraz danych pomiędzy uporządkowanymi ciągami czynności, akcji i obiektów.

Teraz przedstawię i wyjaśnię definicję bazy danych. Baza danych to zbiór danych przedstawiony zgodnie z ustawiony regułami. Pierwsze zmiany o bazie danych pojawiły się już w 1963 roku. W 1970 E. F. Codd zaproponował relacyjny model danych. Komercyjnie pierwszy raz został on wprowadzony w 1980r. Z upływem czasu dopiero po 1990 roku branża IT zaczęła wyrażać zainteresowanie takimi bazami danych jak PostgreSQL i MySQL.

Bazy danych dzielą się na 2 rodzaje:

1. Bazy proste

* Bazy kartotekowe
* Hierarchiczne bazy danych

1. Bazy złożone

* Bazy relacyjne
* Bazy obiektowe
* Bazy relacyjno-obiektowe

Moim zdaniem najbardziej powszechne są bazy relacyjne. W bazach relacyjnych główną ideą jest to, że tabele z bazy danych mogą ze sobą współpracować oraz, o ile są dobrze zaprojektowane, pozwalają uniknąć redundancji (powtarzania się) tych samych danych.

Najbardziej powszechną bazą danych łączoną z systemem Android jest SQLite. Jej największą zaletą jest to, że może pracować bez konieczności instalowania serwera oraz to, że jest na licencji Public Domain, czyli może być wykorzystywana za darmo nawet w komercyjnych aplikacjach.

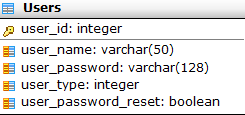
Teraz opiszę krótko język SQL, czyli Structured Query Language. Jest to strukturalny język zapytań, używany w bazach danych do tworzenia, modyfikacji, wstawiania i pobierania danych. Stał się standardem w komunikacji z serwerami relacyjnych baz danych. Jest bardzo powszechnie znany w bazach danych i ciągle sukcesywnie ulepszany oraz stosowany. Najnowszy standard SQL-a został opracowany i wprowadzony w 2003 roku i jest stosowany do tej pory.

Na końcu tego rozdziału wspomnę o algorytmach i algorytmice. Algorytm to inaczej jasno określony ciąg zdefiniowanych czynności, który ma przeprowadzić system z pewnego stanu początkowego do pożądanego stanu końcowego. Algorytmy są głównie zaimplementowane jako kod źródłowy programów komputerowych. Nauką zajmującą się algorytmami i ich badaniem jest algorytmika. Algorytmy mogą się różnić między sobą językiem programowania, w którym zostały zaimplementowane oraz innymi własnościami np. takimi jak złożoność, dane wejściowe, wyjściowe itp. W dalszej części mojej pracy przedstawię problem tworzenia od samego początku algorytmu przeznaczonego dla mojej aplikacji, który będzie szukał dla danej osoby najlepszych możliwych propozycji znajomości na podstawie wybranych wcześniej przez niego preferencji. Zatem danymi wejściowymi będą tutaj wcześniej określone upodobania. Natomiast danymi wyjściowymi będą id użytkowników posortowane malejąco według „dopasowania” na tle preferencji poznawczych użytkownika, które zostały zinterpretowane przez algorytm.

Rozdział IV : Diagramy UML warstwy bazodanowej w systemie „Poznawajka Mobile"

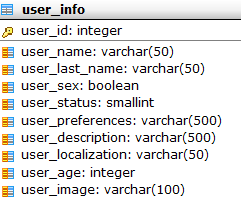
Zaprezentuję wstępną strukturę warstwy bazodanowej aplikacji. Tak jak wcześniej wspominałem baza danych będzie znajdować się po stronie serwera. Do korzystania z jej zasobów będzie potrzebna autoryzacja i uwierzytelnianie. Cała baza danych będzie posiadała mniej więcej 10 tabeli głównych, które będą ze sobą w różnego rodzaju relacjach.

1. Główna tabela Users wraz z hasłami(najprawdopodobniej hashem hasła).



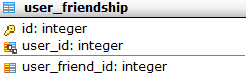
Rys. 1

1. Tabela users\_info przechowywująca dane o użytkownikach. Jest w relacji z tabelą Users.



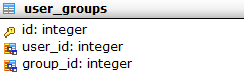
Rys. 2

1. Tabela user\_friendship określająca znajomości użytkowników. Każdy rekord w bazie odpowiadał jednej znajomości, która zostanie później wyświetlona na profilu aplikacji. Jest w relacji z tabela Users.



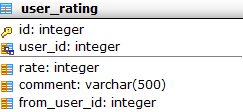
Rys. 3

1. Tabela user\_grups zawiera informacje do jakich grup użytkownik jest zapisany. Działa podobnie jak tabela powyżej. Jest w relacji z tabela Users i Groups.

****

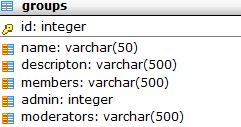
Rys 4

1. Tabela user\_rating zawiera informacje dotyczące ocen, komentarzy zamieszczonych do profilu danego użytkownika.



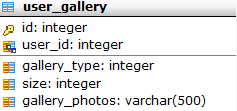
Rys. 5

1. Tabela groups zawiera wszystkie utworzone grupy w aplikacji. Każda grupa posiada swoją nazwę, opis, listę członków, administratorów i moderatorów.



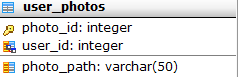
Rys. 6

1. Tabela user\_gallery zawiera informacje i wszystkich posiadanych przez użytkownika galerie zdjęć. Jest w relacji z tabela Users.



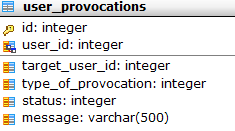
Rys 7.

1. Tabela user\_photos zawiera informacje o zdjęciach użytkowników. Jest w relacji z tabela Users.



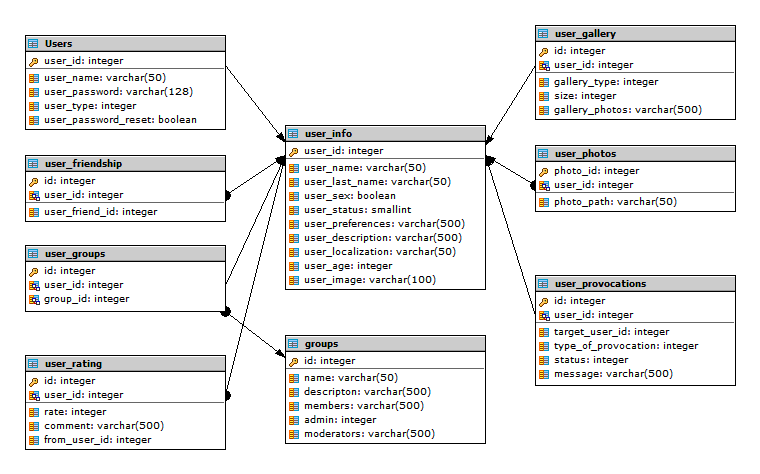
Rys 8.

1. Tabela users\_provocations to tabela, która zapisuje informacje o zaczepkach/propozycjach spotkań, znajomości. Jest w relacji z tabelą Users.



Rys. 9

Cała baza danych z wszystkimi tabelami i relacjami przedstawia się następująco:

Rys. 10

Aplikacja będzie się komunikowała z bazą danych za pomocą zapytań http POST i GET(głównie POST). Przy uwierzytelnianiu będzie generowany unikalny klucz sesji dla użytkownika, który umożliwi mu późniejszą komunikację z bazą bez każdorazowego potwierdzania tożsamości. Klucz będzie wygasał po jakimś czasie. Serwer będzie zwracał odpowiednie dane w formacie JSON, które później będą interpretowane przez aplikacje   
i wyświetlane użytkownikowi.

Poniżej przedstawiam przykładowe przypadki zastosowania, które pokazują w jaki sposób będzie wykorzystywana baza danych znajdująca się na serwerze podczas korzystania z podstawowych funkcjonalności aplikacji:

Rozdział V : Opis i prezentacja algorytmu dopasowywania względem preferencji

Algorytm będzie wykonywany po stronie serwera. Będzie on przyjmował user\_id (czyli unikalny nr dla każdego użytkownika w aplikacji za pomocą którego będzie się miało dostęp w bazie danych do wszystkich potrzebnych danych dla działania algorytmu) . A będzie zwracał user\_id które będą posegregowane względem pasowania. Przedstawię teraz przybliżony opis działania algorytmu.

Zaraz po tym jak użytkownik zasygnalizuje chęć poznania nowych ludzi po preferencjach. Aplikacja wyśle zapytanie do serwera w którym znajdzie się jego wyżej wspomniany user\_id. Gdy serwer otrzyma zapytanie od aplikacji.

1. Wczyta dane personalne użytkownika z tabeli users\_info. Zapisze wygląd użytkownika, oraz jego preferencje. Wszystkie wartości najprawdopodobniej będą indeksowane, będzie miało to na celu późniejsze usprawnienie działania algorytmu i aplikacji oraz uniknięcie nieścisłości.
2. Serwer wykona zapytanie do bazy danych o użytkowników których wygląd znajduje się (lub jest przybliżony) do preferencji przez niego określonych.
3. Następnie usunie jedną z określonych preferencji użytkownika i wykona zapytanie raz jeszcze. Wynik złączy z wcześniejszym i usunie duplikaty. Czynność tą powtórzy do momentu gdy w preferencjach użytkownika zostanie 3 preferowane cechy wyglądu
4. Następnie wykona dla każdego zwróconego użytkownika zapytanie które będzie sprawdzało ile cech wyglądu użytkownika znajduje się w preferencjach osoby zwróconej z zapytania SQL. Zwrócone wyniki zapisze i posegreguje po ilości danych pasujących (użytkownika oraz osoby z rekordu). Najprawdopodobniej będą sortowane po sumie ilości pasujących preferencji użytkownika i ilość pasujących postronnie użytkownika z rekordu z ilorazem ilości wszystkich preferencji użytkownika i użytkownika z rekordu.
5. Serwer sprawdzi dane, zobaczy które z user\_id znajdują się już w kontaktach użytkownika. I wtedy je odrzuci.
6. Dla każdego z user\_id wykona zapytanie po szczegółowe dane i zwróci je owe dano do klienta.
7. Aplikacja po otrzymaniu odpowiedzi. Wyświetli dane użytkowników na liście w odpowiedniej kolejności.

W/w algorytm jest wstępny. Będzie jeszcze dopracowywany podczas dokładnej implementacji klienta oraz serwera. Dlatego też zapisanie go w pseudokodzie pominąłem.

Literatura

[1] Stanisław Wrycza, Bartosz Marcinkowski, Krzysztof Wyrzykowski, Język UML 2.0 w modelowaniu systemów informatycznych, wyd. Helion 2006

[2] <https://pl.wikipedia.org/wiki/Unified_Modeling_Language> [10-01-2015]

[3] <https://pl.wikipedia.org/wiki/SQL> [10-01-2015]

[4] <http://pl.wikipedia.org/wiki/Baza_danych> [10-01-2015]

[5] <https://pl.wikipedia.org/wiki/Algorytm> [10-01-2015]

[6] Charlie Collins, Michael Galpin, Matthias Kaeppler, Android in Practice , tłumaczenie Tomasz Walczak, wyd. Helion 2012