

Tomasz SZYMCZYK, Mateusz ŚWIETLICKI, Paweł ZIELIŃSKI
Politechnika Lubelska, Instytut Informatyki

WYKORZYSTANIE TECHNOLOGII GPS ORAZ ROZSZERZONEJ RZECZYWISTOŚCI W INTELIGENTNYM SYSTEMIE PREZENTACJI ROZKŁADU JAZDY AUTOBUSÓW

Streszczenie. W artykule zaprezentowano nowatorski system informatyczny. Jest to aplikacja prezentująca na telefonie komórkowym rozkład jazdy autobusów. Aplikacja łączy w sobie kilka współczesnych technologii. Dane w systemie zapisane są w relacyjnej bazie danych, która współpracuje z warstwą wspomagającą zarządzanie informacją. Strona klienta to inteligentna aplikacja mobilna wykorzystująca pozycjonowanie GPS. Dla lepszego komfortu użytkownika aplikacja mobilna, współpracuje z Google Maps oraz aplikacją wytyczającą drogę do przystanku lub celu na podstawie rozszerzonej rzeczywistości. Aplikacja jest uniwersalna i skalowalna, dzięki czemu umożliwia implementacje zarówno w miastach, jak i na dużych obszarach, np. całych państwach.

Słowa kluczowe: bazy danych, aplikacje mobilne, pozycjonowanie GPS, systemy ekspertowe, rozszerzona rzeczywistość

USING GPS TECHNOLOGY AND AUGMENTED REALITY IN AN INTELLIGENT PRESENTATION SYSTEM BUS SCHEDULES

Summary. The article shows the innovative information system. This is the application showing on your mobile phone buses schedules. The application combines several of today's technology. The data in the system are saved in a relational database, which works with a layer that helps you manage information. The client side is an intelligent mobile application that uses GPS positioning. For better comfort of use, your mobile collaborates with Google Maps and a plane delineating the way to the bus stop or other destination based on the augmented reality. The application is versatile and scalable so that allows implementations in both cities and in large areas, for example whole countries.

Keywords: database, mobile applications, positioning GPS, experts systems, augmented reality

1. Wstęp

Na ostatnie dekady ubiegłego wieku przypada dynamiczny rozwój technologii informatycznych. Powstały komputery osobiste, które, dzięki swojej niskiej cenie, trafiły do szerokiego grona odbiorców instytucjonalnych, a także prywatnych. Na uczelniach rozwinął się Internet, który szybko przekształcił się w ogólnie dostępną sieć służącą wymianie poglądów, komunikacji i rozrywce. Nieustanna miniaturyzacja i ciągle rosnąca moc obliczeniowa procesorów [1, 7] sprawiły, że komputery zaczęto wykorzystywać niemalże w każdej dziedzinie życia człowieka. Zaawansowane systemy mikroprocesorowe znalazły zastosowanie we wszystkich dziedzinach życia i działalności ludzkiej. Wraz z rozwojem elektronicznych mediów zaczął pojawiać się problem „szumu informacji”, „natłoku informacji”. Obecne systemy informatyczne tworzone są z wykorzystaniem tzw. sztucznej inteligencji, w celu wybiórczego prezentowania wyników zapytania klienta. Pojawiły się systemy ekspertowe wspomagające podejmowanie skomplikowanych decyzji na podstawie analizy danych.

W artykule przedstawiono program służący do wyszukiwania rozkładów jazdy na przystankach autobusowych. Jego zadaniem nie jest jednak powielenie na telefonie użytkownika portalu WWW, z którego może on odczytać godzinę odjazdu interesującego go autobusu czy też innego środka transportu. Aplikacja ma za zadanie w sposób łatwy i trafny przedstawić najbliższe godziny odjazdu autobusu dla danego, konkretnego użytkownika. Dodatkowo powinna pomagać w odnalezieniu najbliższego przystanku oraz wskazać do niego drogę.

Duża część prezentowanej aplikacji powstała jako praca inżynierska panów: Mateusza Świetlickiego i Pawła Zielińskiego, których autor referatu był promotorem [8].

2. Projekt systemu

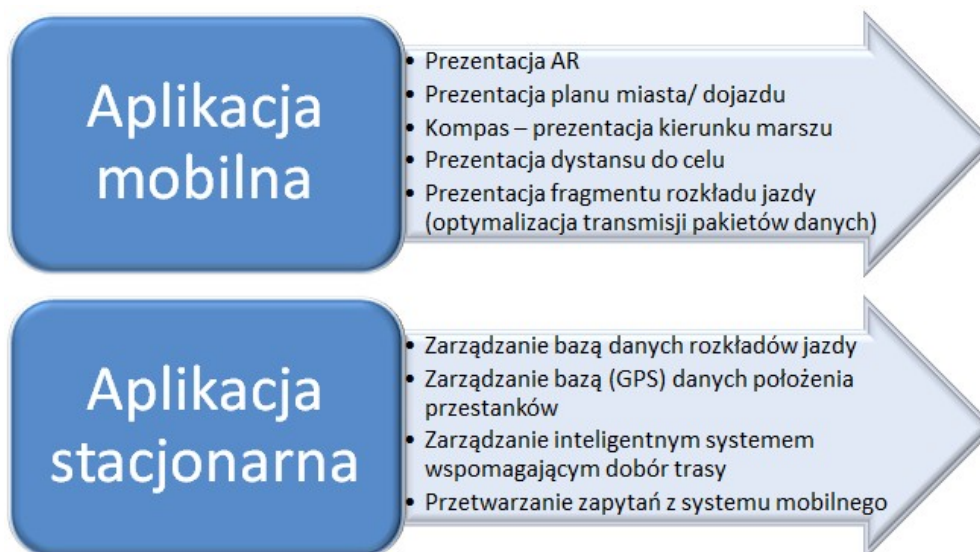
Prezentowana aplikacja składa się z dwóch zasadniczych modułów: części mobilnej oraz z części działającej na komputerze stacjonarnym. Podział z uwzględnieniem najważniejszej funkcjonalności tych modułów przedstawiono na rysunku 1.

Aplikacja mobilna [8], napisana jest dla systemu operacyjnego Windows Mobile 6.x, testowana była z wykorzystaniem telefonu komórkowego HTC HD2.

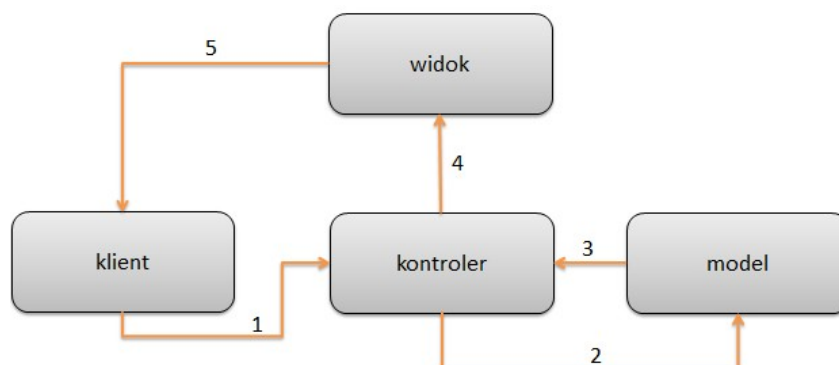
Architektura aplikacji serwerowej jest oparta na wzorcu MVC (ang. *Model-View-Controller*), który jest obsługiwany przez Framework CakePHP. Dzięki zastosowaniu tej konwencji uzyskano ułatwienie w procesie komunikacji pomiędzy poszczególnymi warstwami systemu (aplikacji). Model View Controller zakłada podział aplikacji na trzy główne części [2, 5]:

1. Model – reprezentuje dane aplikacji.

2. Widok – wyświetla dane z modelu.
3. Kontroler – zarządza zadaniami wywołanymi przez klienta.



Rys. 1. Najważniejsze funkcje systemu
Fig. 1. The most significant function of system



Rys. 2. Diagram MVC
Fig. 2. Model-View-Controller diagram

Sposób działania wzorca MVC można prześledzić na powyższym diagramie (rys. 2).

Ogólnie zasadę działania można przedstawić następująco:

1. Klient wysyła zapytanie do aplikacji przez adres URL, a wbudowana funkcja odpowiada za przekazanie zapytania do odpowiedniego kontrolera.
2. Kontroler wykonuje określone zadanie, na przykład poszukuje informacji w bazie danych.
3. Model umożliwia dostęp do danych.
4. Kontroler przetwarza dane i w odpowiedniej formie wysyła je do widoku.
5. Widok odpowiada za graficzną prezentację danych prezentowanych klientowi.

2.1. Projekt systemu mobilnego

Dla mobilnej części zdefiniowano, a następnie oprogramowano następujące wymagania funkcjonalne i нефункционалне.

Wybrane (najważniejsze) wymagania funkcjonalne aplikacji „komórkowej”:

- wyszukiwanie przystanków po ich nazwie lub po nazwie ulicy, na której się znajdują,
- wyszukiwanie przystanków po nazwie (numerze) linii autobusowej/trolejbusowej,
- wyświetlanie linii dostępnych na wybranym przystanku oraz rozkładu ich jazdy,
- wyświetlenie przystanków na danej linii,
- znalezienie i wyświetlenie przystanków w okolicy klienta oraz podawanie odległości w metrach do nich,
- określenie pozycji geograficznej klienta,
- określenie azymutu między pozycją klienta a określonym przystankiem,
- rejestrowanie klienta na serwerze (z uwzględnieniem wieku i sprawności motorycznej klienta oraz nazwy sieci komórkowej, w której się znajduje),
- szyfrowanie hasła klienta podczas transmisji,
- dodanie, edycja i wyświetlanie linii na wybranym przystanku na liście „ulubionych”,
- implementacja obsługi wbudowanego odbiornika GPS oraz elektronicznego kompasu,
- interfejs w języku polskim,
- wykorzystanie protokołu http do komunikacji z serwerem,
- wykorzystanie kodowania ISO-8859-2 podczas komunikacji z serwerem,
- lokalne przechowywanie loginu i hasła klienta.

Wymagania нефункционалне aplikacji mobilnej:

- implementacja ustalonego kontraktu komunikacji z serwerem,
- kompatybilność z ekranem o rozdzielczości do 480x800 pikseli,
- smartphоne z systemem Microsoft Windows Mobile 6.0.0 lub nowszym,
- procesor taktowany częstotliwością co najmniej 400 Mhz,
- wbudowany lub zewnętrzny odbiornik GPS oraz kompas elektroniczny,
- komunikacja z Internetem za pośrednictwem GPRS, HSDPA lub WiFi.

Wszystkie wyszczególnione powyżej wymagania zostały zaimplementowane i funkcjonują prawidłowo w prezentowanej aplikacji.

2.2. Projekt systemu stacjonarnego

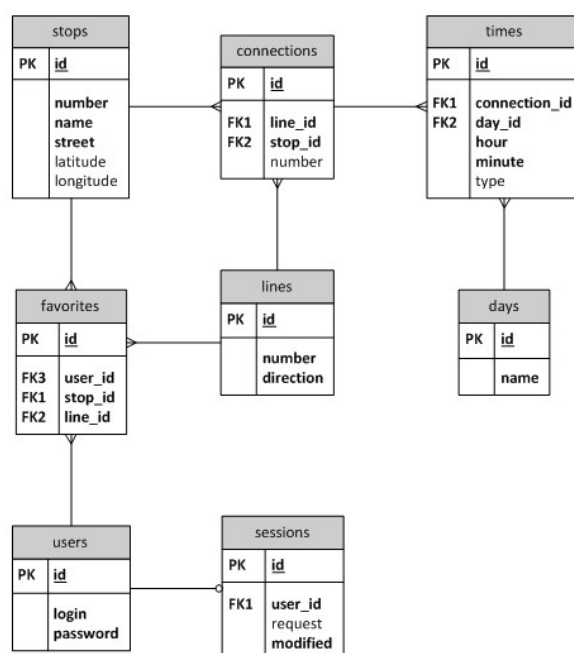
Dla serwerowej części zdefiniowano następujące wymagania funkcjonalne i нефункционалне.

Wymagania funkcjonalne (najważniejsze):

- pełna obsługa i rejestracja konta użytkownika (login, hasło, data urodzenia, poziom sprawności fizycznej, nazwa sieci komórkowej),
- dodawanie i edycja ulubionych rozkładów do bazy danych na serwerze,
- pobieranie ulubionych rozkładów z serwera,
- dodawanie i edycja geolokalizacji przystanków, ich nazw, nazw (numerów) linii, które przejeżdżają przez przystanek oraz rozkładów jazdy autobusów/trolejbusów,
- obliczenie odległości między klientem a przystankiem,
- pobranie położenia przystanku,
- wyszukanie najbliższych przystanków,
- wyszukanie najbliższych przystanków, na których zatrzymuje się autobus interesującej użytkownika linii,
- pobranie danych (rozkładów jazdy) ze strony WWW przewoźnika,
- obliczenie świąt ruchomych,
- przeliczenie czasu potrzebnego na dojście do przystanku indywidualnie dla każdego użytkownika (w zależności od wieku i kondycji fizycznej),
- inteligentne wyświetlenie rozkładu jazdy.

Dane potrzebne do funkcjonowania systemu zostały zapisane w ośmiu tabelach (rys. 3). Nazwy tabeli, zgodnie z ogólnymi konwencjami, jednoznacznie nawiązują do rodzaju danych w nich przechowywanych. Tabela przystanki (stops) zawiera czterocyfrowy numer przystanku, który widnieje na każdym przystanku i nadany jest przez MPK Lublin. Numer jest wymagany i unikalny. W przypadku wdrożenia systemu jedynie w obrębie jednego miasta mógłby z powodzeniem posłużyć za klucz główny, jednak uniwersalność aplikacji wymusiła nadanie dodatkowego pola pełniącego tę funkcję. Tabela stops zawiera także potoczne nazwy przystanków (pole: name) np. – „Real 02”, nazwę ulicy (pole: street), przy której znajduje się ten przystanek, np. „Szelińskiego”, oraz w polach: latitude i longitude – współrzędne położenia geograficznego tego przystanku. Tabela stops składa się blisko z 760 rekordów.

Z uwagi na fakt, iż przez konkretny przystanek prowadzi wiele linii autobusowych lub trolejbusowych, tabela przystanki połączona jest z tabelą linie (lines) przez tabelę pośredniczącą – połączenia (connections). Tabela linie (lines) zawiera informacje o numerze linii (pole numer) w formacie umożliwiającym zapis zarówno liczb, jak i liczby z dodatkową literą. W Lublinie istnieje konwencja numerowania linii nocnych z literą N, natomiast linii trolejbusowych jako numery powyżej 150. Ostatnie pole w tej tabeli – kierunek (direction) – to potoczna nazwa końcowego przystanku danej linii, np. „Dworzec PKP”. Zestawienie numeru linii i kierunku w sposób jednoznaczny ułatwia orientację w sieci połączeń.



Rys. 3. Uproszczony widok ERD

Fig. 3. Entity Relationship Diagram (a simplified view)

Tabela dni (days) jest zbiorem typów rozkładu, które zależą od dnia. Rozkład jazdy komunikacji miejskiej jest inny w dni: robocze (poniedziałek – piątek), dni wolne od pracy (sobota, niedziela) oraz świąteczne. Rozróżnia się także okres wakacyjny (lipiec, sierpień), w którym komunikacja funkcjonuje odmiennie niż w pozostałe części roku.

Dzięki tej tabeli aplikacja ma możliwość znalezienia odpowiedniego rozkładu na konkretny dzień. Tabela days połączona jest relacją jeden do wielu z następną opisywaną tabelą – times.

Tabela times jest zbiorem godzin odjazdów autobusów z przystanków, czyli rozkładem jazdy, której każdy rekord składa się z:

- Klucza obcego – połączenia (pole connection_id) – identyfikuje połączenie linii z przystankiem. Jest on tożsamy z kluczem głównym z tabeli connections.
- Klucza obcego – dotyczącego dnia (pole day_id) – określa typ rozkładu.
- Godzina i minuta odjazdu.
- Typ odjazdu (pole type) – dodatkowo specyfikuje, czy jest to przewoźnik MPK Lublin czy inny, prywatny. Zawiera także informację, czy jest to autobus niskopodłogowy przystosowany do przewozu osób niepełnosprawnych.

Tabela (times) jest najobszerniejszą tabelą, zawierającą ponad 400 tysięcy rekordów.

Tabela users jest zbiorem danych o użytkownikach, którzy zalogowali się do systemu. Składa się z nazwy użytkownika oraz hasła. Podczas logowania aplikacja tworzy wpis w tabeli sessions, która składa się z unikalnego uuid oraz numeru identyfikacyjnego użytkownika. Dodatkowo tabela sessions przechowuje informację o czasie ostatniej akcji użytkownika.

Tabela favorites jest miejscem, na którym użytkownicy przechowują ulubione rozkłady (odjazdy linii na przystanku).

3. System mobilny

Mobilna część systemu to aplikacja umożliwiająca odnalezienie informacji na temat rozkładów jazdy autobusów, z wykorzystaniem telefonu komórkowego wyposażonego w GPS oraz kompas elektroniczny [6]. Aplikacja „komórkowa” pełni bardzo ważną funkcję w systemie. Z uwagi na fakt, iż współczesne smartfony posiadają jeszcze zdecydowanie wolniejsze procesory oraz dysponują znacznie mniejszymi pojemnościami pamięci niż odpowiadające im komputery stacjonarne czy też dedykowane serwery, całą część „obliczeniową” przeniesiono na serwerową część prezentowanego systemu.

Pracę z systemie użytkownik rozpoczyna od podania danych potrzebnych do rejestracji. Pobierane są tradycyjne dane jak login i hasło. Pojawiają się także pytania o wiek (data urodzin), a także o „sprawność fizyczną” (sprawny, otyły, poruszanie się o kulach, inwalida, wózek inwalidzki itp.). Podczas rejestracji użytkownik proszony jest także o wybór operatora GSM, za pośrednictwem którego łączy się z Internetem, jeżeli jest to połączenie dodatkowo płatne.

W celu ułatwienia pracy z systemem, w smartfonie zapamiętywane są login i hasło, tak aby podłączenie do systemu mogło odbywać się w sposób automatyczny, niewymagający pamięci i interwencji użytkownika. Wykorzystując fakt, iż każdy telefon posiada unikalny kod IMEI, ten numer domyślnie jest loginem użytkownika. Istnieje także możliwość zmiany na inny, dowolny ciąg znaków. Nad unikalnością nazwy loginu czuwa serwerowa aplikacja.

Po zakończeniu procesu rejestracji użytkownik może już przejść do wyboru przystanku w celu wyszukania interesującego go rozkładu jazdy.

Wyszukiwanie odbywa się wedle kilku kryteriów:

- numeru autobusu, którym użytkownik zamierza jechać,
- nazwy ulicy, na której znajduje się przystanek,
- najbliższych (dowolnych) przystanków w okolicy,
- numeru autobusu, którym użytkownik zamierza jechać (w pobliżu użytkownika),
- nazwy ulicy, na której znajduje się przystanek (w pobliżu użytkownika).

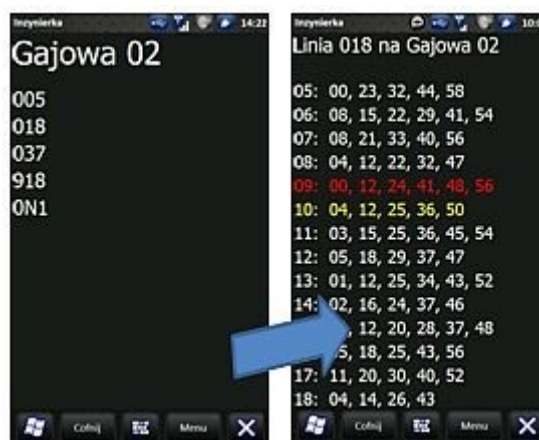
Przykład wyszukiwania po kryteriom nazwy pokazano na rys. 4. W celu ułatwienia pracy z ekranem dotykowym (który w przypadku złych warunków atmosferycznych czy też niskiej temperatury jest utrudniony) zaimplementowano możliwość wyszukiwania rozkładu jazdy po fragmencie nazwy ulicy czy też numeru linii. Należy wspomnieć, że cała aplikacja napisana jest w konwencji „finger friendly”. W przypadku wyszukiwania po fragmencie, czy to nazwy

czy numeru linii, pojawia się podpowiedź w postaci wyświetlenia wszystkich numerów linii zawierających wprowadzony ciąg znaków.



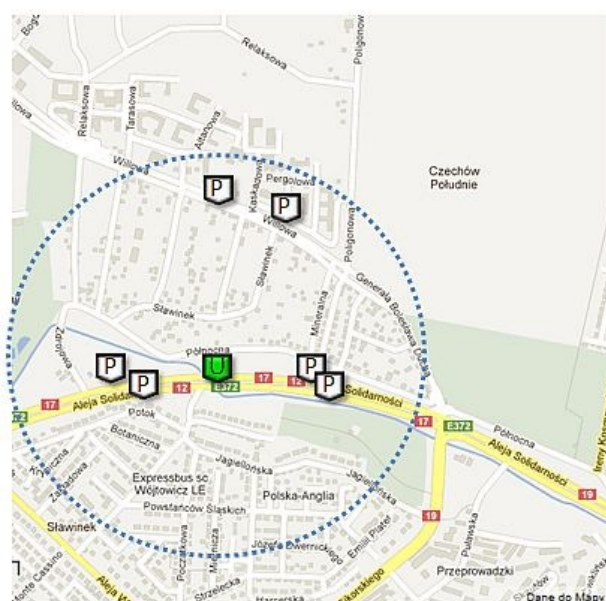
Rys. 4. Przykładowy widok wyszukiwania informacji
Fig. 4. Searching for information (the example view)

Po wyborze konkretnego przystanku otrzymuje się informację o rozkładzie jazdy z tego przystanku (rys. 5).



Rys. 5. Widok rozkładu jazdy autobusów
Fig. 5. A buses schedule view

Istnieje również możliwość wyboru rozkładu jazdy na kilku „najbliższych” przystankach (rys. 6). Po wybraniu takiej opcji system pobiera z telefonu klienta jego aktualną pozycję GPS, ustala położenie klienta (marker U) i następnie wyświetla zadaną liczbę przystanków w określonym promieniu odległości, np. 500 m, a obliczając odległość w linii prostej, wybiera najbliższe znajdujące się przystanki (markery P). Przedstawiono to na rys. 6.



Rys. 6. Wybór najbliższych przystanków
Fig. 6. Selection of the nearest bus stops

W celu wyszukania informacji o rozkładzie jazdy na wybranym przystanku, konkretnej linii autobusowej lub trolejbusowej, należy podać nazwę ulicy, na której szukamy najbliższego przystanku lub numer przystanku i numer linii, którą chcemy pojechać.

Odpowiedzią systemu jest seria następujących czynności:

1. Określenie dokładnej geolokalizacji użytkownika.
2. Nawiązanie połączenia z serwerem (autentykacja i autoryzacja).
3. Przesłanie na serwer informacji o geolokalizacji klienta oraz zapytania o najbliższy przystanek w okolicy spełniający kryteria: ulica i numer linii.
4. Oczekiwanie i przyjęcie odpowiedzi z serwera (współrzędne GPS przystanków spełniających kryteria, ich nazwy oraz rozkłady jazdy szukanej linii na tych przystankach).
5. Inteligentne przeliczenie (na podstawie estymacji czasu dojścia do przystanku według wieku, zdolności motorycznych oraz odległości do przystanku) godzin odjazdu najbliższych „osiągalnych” autobusów lub trolejbusów.
6. Wyświetlenie informacji na ekranie smartfonu, prezentującej uporządkowane (aktualne) godziny odjazdów autobusów.

Dodatkowo istnieje możliwość wskazania (wykorzystując kompas, a także informację o położeniu użytkownika i przystanku, na który kieruje się klient) kierunku, w którym powinien się udać klient na konkretny przystanek. Jednocześnie, wraz z marszem, zmienia się na bieżąco odległość do przystanku oraz kierunek, w którym znajduje się przystanek. Zaprezentowano to na rys. 7. Wyznaczenie kierunku jest dodatkową wskazówką pokazującą, czy użytkownik udaje się w dobrym kierunku.



Rys. 7. Przykładowy widok prezentujący kierunek i odległość do wybranego przystanku
 Fig. 7. The sample view showing the direction and distance to selected bus stop

Transmisja danych może odbywać się za pośrednictwem dowolnej technologii: WiFi, GPRS, HSDPA. Wielkość transmitowanych danych jest optymalizowana, tak aby był to jeden pakiet rozliczeniowy danego operatora GSM.

4. System stacjonarny

System stacjonarny napisany jest z wykorzystaniem technologii CakePHP [3]. Pełni funkcję bazy danych rozkładów jazdy, przystanków i użytkowników. Dodatkowo jest w nim zaimplementowana większość logiki aplikacji. Uproszczony schemat relacyjnej bazy danych przedstawiono na rys. 3. Aplikacja stacjonarna umożliwia zarówno ręczne wprowadzenie danych o przystankach: geolokalizacja, numery linii i rozkłady jazdy, jak również automatyczne pobranie informacji o rozkładach jazdy na konkretnych przystankach ze strony WWW dowolnego przewoźnika. Aplikacja automatycznie pobiera dane, rozpoznaje je, parsuje i zapisuje do odpowiednich tabeli.

Aplikacja serwerowa, przed odesłaniem odpowiedzi dotyczącej rozkładu jazdy, sprawdza, która jest godzina, jaki jest obecnie dzień tygodnia (pracujący czy wolny od pracy), czy jest to dzień świąteczny (również wylicza datę świąt ruchomych – Święta Wielkanocne). Następnie analizuje dane identyfikacyjne konkretnego użytkownika wywołującego zapytanie i dopiero po tym przesyła aktualny rozkład jazdy. Istnieje wariant odpowiedzi serwera pokazujący tylko wybrane godziny, a nie cały dzienny rozkład jazdy. Prezentowane są wtedy informacje o ostatnim autobusie, który odjechał oraz o kilku najbliższych. W celu obniżenia kosztów opłaty za transmisję danych operatora GSM, liczba przesyłanych danych jest automatycznie skalkulowana na taką, aby była to pojedyncza rozliczeniowa paczka danych. Możliwe jest także obliczenie przybliżonego czasu dojścia do przystanku. W przypadku gdy w czasie rejestracji podany został wiek lub stan sprawności fizycznej. Aplikacja serwerowa uwzględ-

nia ten fakt przy wyświetlaniu najbliższego rozkładu jazdy. Eliminowany jest więc dzięki temu przypadek, gdy schorowanej, starszej osobie wyświetla się czas odjazdu najbliższego autobusu, do którego nie ma szans zdążyć dojść. Jest to istotne zwłaszcza w przypadku „rzadkich”, np. co pół godziny, odjazdów danego autobusu. Ma to zapobiec niepotrzebnemu, często nawet kilkudziesięciominutowemu oczekiwaniu na przystanku na następny środek transportu.

Komfort pracy z aplikacją zapewnia funkcja „ulubione przystanki”. Ekran „ulubione” wyświetla zapamiętane na serwerze wybrane wcześniej linie na konkretnych przystankach. Dodatkowo w widoku tym prezentowana jest godzina, o której odjechał już ostatni autobus, oraz trzy najbliższe godziny odjazdów autobusów danej linii na wybranym przystanku. Godzina przyjazdu może posiadać literę „n” lub „p”, która informuje o rodzaju autobusu odpowiednio niskopodłogowym i komunikacji prywatnej. System może zawierać także inne oznaczenia.

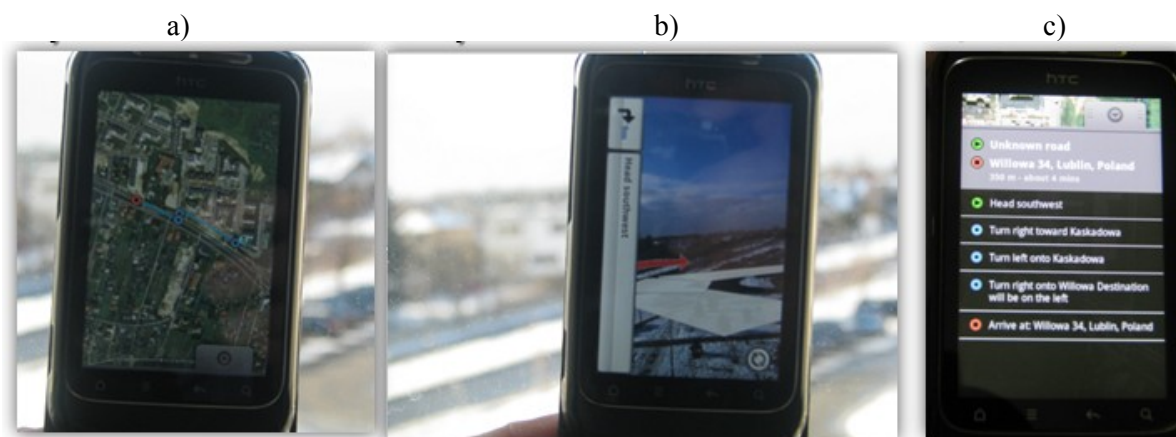
5. System Augmented Reality

Określenie kierunku i odległości do przystanku w skrajnych wypadkach może być niewystarczające, aby sprawnie i szybko dojść do przystanku. Często infrastruktura ulic, ich zawiłości, sprawiają, że sprawne, szybkie dojście do przystanku nie jest łatwe. Aplikacja pokazuje odległość liczoną w linii prostej. Bardzo rzadko długość fizycznej drogi, jaką trzeba pokonać jest z nią zgodna. W takim przypadku warto użyć dodatkowego „przewodnika”. Darmowa Aplikacja (*AugSatNav*) [9], opierając się na mapach Google, pokazuje, którądy należy się udać do wybranego celu. Istnieje także możliwość skorzystania z kamery w celu wykorzystania rozszerzonej rzeczywistości. AR (*Augmented Reality*) to nic innego jak naniesienie na normalny widok (widok z kamery) dodatkowych cyfrowych informacji. W opisywanym przypadku jest to linia pokazująca, którądy należy iść, aby szybko osiągnąć cel (rys. 8b). Widok z kamery oraz widok typowej mapy GPS (rys.8a) przedstawiono na rysunku poniżej (rys. 8).

Rysunek 8b, za pomocą „białej linii” w obrazie z kamery smartfonu, pokazuje, którądy należy iść do wybranego przystanku (punktu na mapie). Dzięki takiej „wskazówce” kluczenie pomiędzy blokami zasłaniającymi przystanek nie jest już żadnym problemem. Użytkownik jest prowadzony w sposób bardzo dokładny, a także zna odległość oraz czas dojścia do przystanku. Odległość, jaką ma do pokonania, obliczona jest na podstawie długości ścieżki pobranej z Google Maps i jest to rzeczywista odległość do przystanku. Czas dojścia liczony jest ze wzoru (1), przy czym prędkość, z jaką się porusza w momencie rozpoczęcia marszu, wynosi ok. 4 km/h (1,1 m/s), a następnie jest uśredniana na podstawie prawdziwych, chwilowych wartości prędkości odczytanych z GPS:

$$t = \frac{s}{\bar{v}}, \quad (1)$$

gdzie s jest odległością w metrach, a \bar{v} oznacza średnią, chwilową prędkość, aktualizowaną co kilka sekund, wyrażoną odpowiednio w m/s.



Rys. 8. Przykładowy widok prezentujący kierunek, w którym znajduje się przystanek, oraz odległość do niego

Fig. 8. The sample view showing the direction and distance to selected bus stop

Na rysunku rys. 8c pokazano podsumowanie, któredy (jakimi ulicami) użytkownik powinien udać się do celu. Pokazano tutaj także szacunkowy czas dojścia do celu oraz odległość do niego.

6. Podsumowanie

Prezentowana aplikacja może z powodzeniem konkurować z dostępnymi komercyjnymi aplikacjami. Do największych konkurentów należy zaliczyć produkty: mobileMPK (mMPK) oraz Transportoid [9]. W momencie rozpoczęcia prac nad opisywaną aplikacją w Lublinie nie funkcjonował żaden podobny system. Źródłem informacji o rozkładzie jazdy autobusów były jedynie papierowe rozpiski na przystankach oraz strona WWW MPK Lublin [10]. Portal przewoźnika nie był dostosowany do przeglądania go w telefonach komórkowych czy też w smartfonach. W przypadku użycia dodatkowej aplikacji, wykorzystującej technologię AR, należy wspomnieć, że dopiero od połowy stycznia 2012 roku (po aktualizacji map Google) w Lublinie można było w miarę poprawnie wytyczyć pieszą trasę. Do tego czasu aplikacja kierowała użytkownika jedynie „drogami samochodowymi”, często wprowadzając w poważny błąd: odległość w linii prostej do przystanku -100 m, natomiast po drodze wskazanej przez „białą linię” ponad 500 m. Należy także wspomnieć, że nawet obecny, bardziej poprawny, sposób wytyczania trasy do przystanku nie uwzględnia ogrodzeń czy też wydeptanych ście-

żek „na skróty”. Stanowi jednak nieocenioną pomoc, w sytuacji gdy nie widać przystanku, a użytkownik nie zna topologii dróg do niego prowadzących.

BIBLIOGRAFIA

1. Bigo Ł.: Prawo Moore’a – było, jest, nie będzie. PC World Komputer, Nr. 4, 2006.
2. Gajda W.: PHP. Praktyczne projekty. Helion, Gliwice 2009.
3. Golding D.: Beginning cakePHP: From Novice to Professional. Apress, 2008.
4. Filipowski P.: Sposoby projektowania systemów informatycznych we współczesnych firmach. Promotor Szymczyk T., Politechnika Lubelska, Lublin 2011.
5. Freeman E., Freeman E., Bates B., Sierra K.: Wzorce projektowe. Rusz głową! Helion, 2010.
6. McPherson-McGraw F.: How to do everything with Windows Mobile. Hill Osborne Media, 2006.
7. Lisowski P.: Intel wyprzedza prawo Moore’a, które sam ustanowił w 1965 roku. Chip, Nr. 5, 2011.
8. Świetlicki M., Zieliński P.: Internetowy rozkład jazdy autobusów w mieście. Projekt i wykonanie aplikacji serwerowej i mobilnej. Promotor Szymczyk T., Politechnika Lubelska, Lublin 2011.
9. <https://market.android.com>.
10. <http://mpk.lublin.pl>.
11. <http://store.ovi.com>.
12. <http://pl.wikipedia.org/wiki/>

Wpłynęło do Redakcji 22 stycznia 2012 r.

Abstract

This article presents a new working program to help the user in the presentation buses schedule for the selected stops. An application can be useful also for any other shuttle buses. The application uses GPS to locate the client. Then tells us where the nearest bus stops.

It showing a buses departure times the application takes into account the distance the between the customer to bus station, and chance to come in at right time. The results of the application were shown in Fig. 4, Fig. 5, Fig. 7.

The application works in client-server architecture. Client is a mobile software on your smartphone running Microsoft Windows Mobile 6.0.0 or later. Its principal task is to use geolocation, communication with the database (server side) visualization of query results. It is possible to designate the route to bus stop based on Google Maps using augmented reality technology (Fig. 8). Mobile application uses the Model View Controller architecture (Fig. 2).

Some server was written using CakePHP. Its primary task is to save information about buses schedules, records of users' data, along with authorization and authentication process. The application is designed as intelligent selection and transferring schedules. For this purpose automatically calculates whether the day is an ordinary working day, non-working or whether they are public holidays (always accruing to that day, or moveable).

At the time of start of work on the described application there didn't function any similar system in Lublin area. Source of information about bus schedules were only paper notes on the bus stations and website MPK Lublin [10]. Portal of transporter was not adapted to the viewing it on mobile phones or the smartphones. Currently there are two serious competitors: mobileMPK (mMPK) and Transportoid [9]. Their functionality, however, is much more smaller.

Adres

Tomasz SZYMCZYK: Politechnika Lubelska, Instytut Informatyki, ul. Nadbystrzycka 36 b, 20-618 Lublin, Polska, T.Szymczyk@pollub.pl.

Mateusz ŚWIETLICKI: Politechnika Lubelska, Instytut Informatyki, ul. Nadbystrzycka 36 b, 20-618 Lublin, Polska.

Paweł ZIELIŃSKI: Politechnika Lubelska, Instytut Informatyki, ul. Nadbystrzycka 36 b, 20-618 Lublin, Polska.