PAŃSTWOWA WYŻSZA SZKOŁA ZAWODOWA
W NOWYM SĄCZU

Instytut Techniczny
Informatyka Stosowana

DOKUMENTACJA PROJEKTOWA

PROGRAMOWANIE URZĄDZEŃ MOBILNYCH

Zdrowy Spacer

Autorzy:
Kamil Pociecha
Nicolas Świątnik

Prowadzący:
mgr inż. Dawid Kotlarski

Nowy Sącz 2021

Spis treści

1. Ogólne określenie wymagań

Aplikacja ma się nazywać Zdrowy Spacer a jej grupą docelową mają być osoby aktywne fizycznie, które chcą kontrolować swoje wyniki.

1.1. Podstawowe wymagania

Aplikacja powinna ona spełniać podstawowe funkcjonalności takie jak: pomiary przebytych odległości, liczbę przebytych kroków, liczbę spalonych kalorii.

1.2. Dodatkowe wymagania

Dodatkowo chcielibyśmy aby aplikacja rzutowała całą trasę na mapy, miała chociażby zapisywaną historię poprzednich tras do wglądu dla użytkownika, oczywiście umożliwiała ustalenie celu czy to podróży lub przebytych kroków/spalonych kalorii. Jako dodatkową funkcję przewidujemy też możliwość robienia zdjęć osiągniętego już celu, czyli np. zrobienie zdjęcia szczytu góry jako potwierdzenie osiągnięcia swojego celu. Chcielibyśmy również umożliwić naszym klientom logowanie się poprzez konta typu Facebook czy Google, co mogło by zautomatyzować proces logowania. Kolejną opcja którą przewidujemy by znalazła się w zamawianej przez nas aplikacji jest motyw typu jasny/ciemny dla uprzyjemnienia samego doświadczenia z korzystania z niej. Oczywiście aplikacja powinna mieć możliwość bezproblemowego działania w tle jak i również ewentualnego wykrycia ruchu bez jej inicjacji co przełoży się na to że sama zacznie rejestrować nasz spacer oraz aktywności.

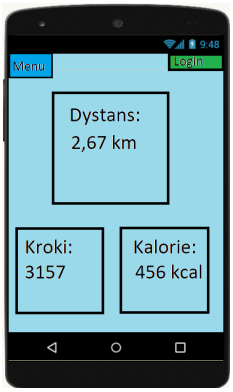
2. Określenie wymagań szczegółowych

2.1. Opis wymagań

Aplikacja korzystać będzie z czujnika GPS, który umożliwi pomiar pokonanej odległości oraz rzutowanie trasy na mapę - w tym celu użyte zostaną mapy Google. Na podstawie przebytej odległości będzie można określić także liczbę kroków oraz ilość spalonych kalorii. Odległości przebyte w poprzednich dniach będą zapisywane w aplikacji co pozwoli użytkownikowi na porównywanie swoich wyników z poszczególnych dni. W przypadku osiągnięcia wyniku poniżej wybranej normy użytkownik zostanie powiadomiony o tym fakcie przez aplikację. W celu zmiany motywu aplikacji użyty zostanie czujnik światła, dzięki któremu w trakcie dnia będzie funkcjonował tryb ciemny a w nocy tryb jasny. Użytkownik będzie miał możliwość zaznaczenia na mapie swojego celu, do którego chce danego dnia dotrzeć. Następnie będzie miał możliwość by przy użyciu aparatu w telefonie wykonać zdjęcie osiągniętego miejsca docelowego. Zdjęcie to będzie zapisywanie w folderze aparatu a użytkownik będzie miał wgląd w zdjęcia swoich poprzednich celów. Aby ułatwić proces logowania Użytkownik będzie miał możliwość logowania do aplikacji za pomocą Facebook'a lub konta Google. Aplikacja będzie działała cały czas bez potrzeby inicjowania. Możliwe będzie jej działanie w tle a także zwijanie do paska.

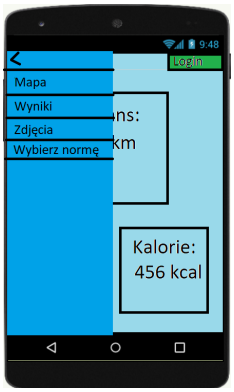
2.2. Layout aplikacji

Na stronie głównej aplikacja będzie wyświetlać przebytą odległość, liczbę wykonanych kroków oraz liczbę spalonych kalorii. W prawym górnym rogu znajduje się przycisk, który po jego naciśnięciu umożliwi zalogowanie się do aplikacji. W lewym górnym rogu znajduje się przycisk otwierający menu.



Rys. 2.1. podstawowy layout aplikacji

Po otwarciu menu pojawi się lista w której znajdują się elementy takie jak: mapa - pokazuje przebytą trasę oraz umożliwia ustalenie swojego celu, wyniki - wyświetla wyniki osiągnięte w poprzednich dniach, Zdjęcia - otwiera galerię ze zdjęciami miejsc docelowych, wybierz normę - pozwala ustalić użytkownikowi jaki wynik chciałby osiągnąć.



Rys. 2.2. layout aplikacji po otwarciu menu

3. Projektowanie

3.1. Przygotowanie narzędzi Git oraz Visual Studio

W celu korzystania z narzędzia Git należy utworzyć na swoim komputerze repozytorium lokalne oraz dodać do niego pliki projektu. Na platformie GitHub utworzyć repozytorium zdalne, które umożliwi wszystkim autorom projektu współprace przy tworzeniu aplikacji.
W Visual Studio należy zainstalować dodatek opracowywanie aplikacji mobilnych za pomocą środowiska Xamarin oraz zestaw Android SDK, który umożliwia korzystanie z emulatora. Do tworzenia projeku wybraliśmy szablo aplikacji mobilnej Xamarin.Forms.

3.2. Biblioteki i klasy

Biblioteka Xamarin.Essentials udostępnia międzyplatformowy interfejs programowania aplikacji mobilnych (API). Jest ona dostępna jako pakiet NuGet i jest uwzględniana w każdym nowym projekcie w programie Visual Studio. Biblioteka ta oferuje klasy, które zostaną przez nas użyte podczas tworzenia projektu.
Geolokalizacja:
Klasa Geolocation udostępnia interfejsy API do pobierania bieżących współrzędnych geolokalizacji urządzenia.
Motyw aplikacji:
Interfejs API RequestedTheme jest częścią AppInfo klasy i zawiera informacje dotyczące tematu żądanego dla uruchomionej aplikacji przez system.
Akcelerometr:
Klasa Accelerometer umożliwia monitorowanie czujnika przyspieszeniomierza urządzenia, który wskazuje przyspieszenie urządzenia w trójwymiarowej przestrzeni.
Klasa GoogleMap:
Firma Google oferuje natywny interfejs API mapowania dla systemu Android. Pozwala on na zmienianie punktu widzenia mapy, dodawanie i dostosowywanie znaczników, oznaczanie mapy za pomocą nakładek.
Wymagania wstępne Mapy API usługi Google: uzyskanie klucza Mapy API, zainstalowanie pakietu Xamarin.GooglePlayServices i Mapy pakietu z NuGet, określenie wymaganych uprawnień.
Klasa GoogleMap poprzez aplikację platformy Xamarin.Android będzie współdziałała z aplikacją Google Maps.

Biblioteka Microsoft Authentication Library (MSAL) pozwala na dodawanie uwierzytelniania do aplikacja. Umożliwi to logowanie do aplikacji przy użyciu Google lub Facebook.
Klasa WebAuthenticator umożliwia inicjowanie przepływów opartych na przeglądarce, które nasłuchują wywołania zwrotnego do określonego adresu URL zarejestrowanego w aplikacji.

4. Implementacja

Tworznie Menu:
W activity_main_drawer.xml:
- andriod:id - nadanie nazwy elementu
- android:icon - wybranie jednej z domyślnych ikon do menu
- android:title - dodaje napis, który będzie wyświetlany na elemencie
Przykład:
android:id="@+id/nav_mapmode"
android:icon="@android:drawable/ic menu_mapmode"
android:title="Mapa"
W MainActivity.cs:
W funkcji public boolOnNavigationItemSelected(IMenuItem item) warunek id == Resource.ID.[nazwa elementu] określa dla którego elementu będzie wykonywana dana instrukcja.
Przykład:

```
if (id == Resource.Id.nav_mapmode)
{

}
```

5. Testowanie

6. Podręcznik użytkownika

Bibliografia

[1] Tadeusz Legierski i in. *Programowanie Sterowników PLC*. Gliwice: Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, 1998.

[2] *Strona internetowa firmy SELS*. URL: http://www.sels.com.pl/index.php?cPath=1 (term. wiz. 29. 10. 2012).

Spis rysunków

Spis tabel