

AKADEMIA NAUK STOSOWANYCH W NOWYM SĄCZU

Wydział Nauk Inżynierskich
Katedra Informatyki

DOKUMENTACJA PROJEKTOWA PROGRAMOWANIE URZĄDZEŃ MOBILNYCH

...Spacer...

Autor:
Kamil Gruca
Józef Czelusta

Prowadzący:
mgr inż. Dawid Kotlarski

Nowy Sącz 2024

Spis treści

1. Ogólne określenie wymagań	3
1.1. Przykład	3
1.2. Instalacja	5
2. Określenie wymagań szczegółowych	7
2.1. Język programowania	7
2.2. Podstawowe funkcje aplikacji	8
3. Projektowanie	10
4. Implementacja	11
5. Testowanie	12
6. Podręcznik użytkownika	13
Literatura	14
Spis rysunków	14
Spis tabel	15
Spis listingów	16

1. Ogólne określenie wymagań

Aplikacja mobilna będzie pełnić funkcje aplikacji która będzie skierowana do osób które uprawiają sporty biegowe. Aplikacja będzie monitorowała takie parametry jak: przebyty dystans, liczba kroków, obecną lokalizację, liczbę spalonych kalorii. Dodatkowo aplikacja będzie przechowywała historie aktywności, będzie można ustawić cele na przyszłość oraz dla lepszego komfortu użytkownika automatyczna regulacja jasności ekranu. .

1.1. Przykład

Tak zaczynamy pisanie pierwszego akapitu. Jeśli chcemy napisać przypis do bibliografii wykonujemy to w ten sposób¹.



Rys. 1.1. Logo

Tutaj może coś być wpisane.

Tutaj może coś być wpisane². Rysunek 1.1 (s. 3) pokazuje przykładową ilustrację.

Tab. 1.1. Tabelka przykładowa

U_n	I_{zw}
kV	%
7.2	100

Listing kodu

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4 using namespace std;
5
6 /*
7 liczby pseldolosowe
```

¹Przykład odnośnika do książki[legierski].

²Przykład odnośnika do strony www[www1].

```

8 */
9
10 int main(int argc, char** argv) {
11
12     int tab[10][10];
13
14     for(int i=0;i<10;i++)
15     for(int j=0;j<10;j++)
16     tab[i][j]=0;
17
18     srand(time(NULL));    //generowanie z czasu
19     int min=3;
20     int max=7;
21     for(int i=0;i<10;i++)
22     for(int j=0;j<10;j++)
23     tab[i][j]=(rand()%(max-min+1))+min;
24
25     for(int i=0;i<10;i++)
26     {
27         for(int j=0;j<10;j++)
28             cout<<tab[i][j]<<" ";
29         cout<<endl;
30     }
31
32     return 0;
33 }

```

Listing 1. Przykładowy kod 001

Tutaj może coś być wpisane. Tutaj może coś być wpisane. Tutaj może coś być wpisane. Tabela 1.1 (s. 3) pokazuje sposoby użycia trybu matematycznego.

Kod 1 (s. 3) przedstawia sposób generowania liczb pseudolosowych. Kod 2 (s. 4) przedstawia generowanie pliku HTML.

Alternatywna metoda wklejenia kodu:

```

1 #include <iostream>
2 #include <fstream>
3 using namespace std;
4
5 int main(int argc, char** argv) {
6
7     ofstream plik("strona.html");
8     if(!plik)
9         cout<<"blad zapisu pliku";
10    else

```

```
11 {
12     plik<<"<html>";
13     plik<<"<head><title>Moja pierwsza strona www</title></head>";
14     plik<<"<body>Strona WWW</body>";
15     plik<<"</html>";
16
17     cout<<"Wygenerowana strona";
18 }
19 plik.close();
20
21
22 return 0;
23 }
```

Listing 2. Przykładowy kod 002

1.2. Instalacja

Poniżej są opisane kroki potrzebne do instalacji L^AT_EX'a oraz do używania tego szablonu.

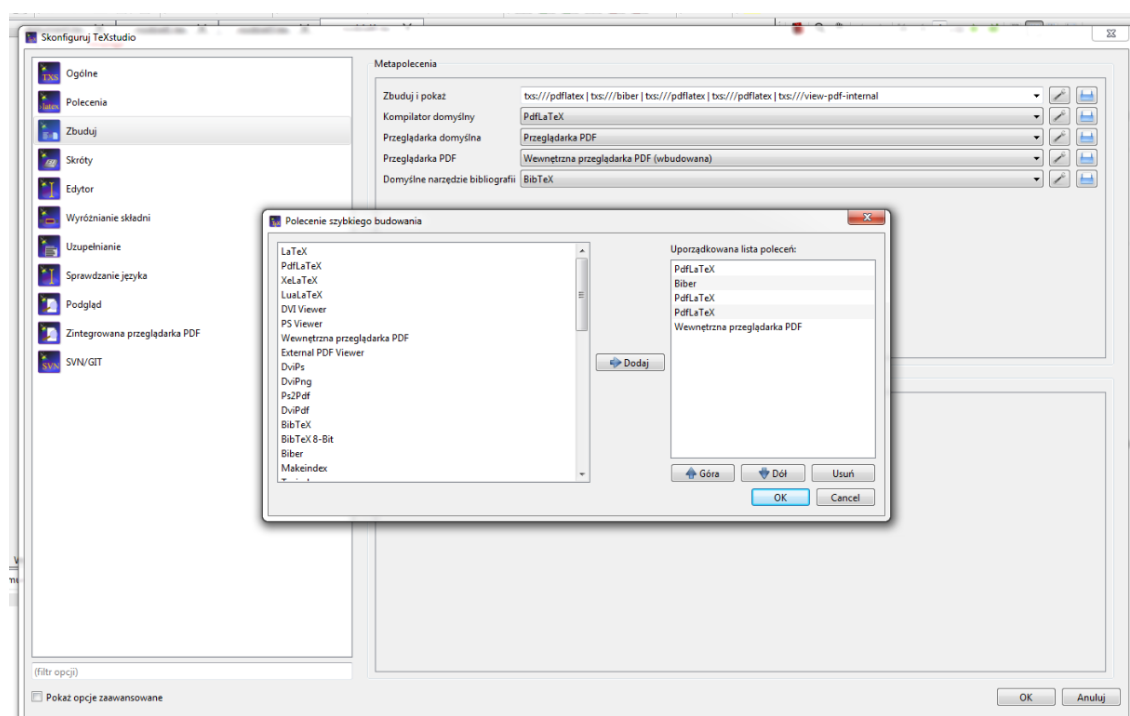
Na początku instalujemy T_EXLive³. Ściągamy plik instalacyjny, zajmuje około 25MB. Podczas instalacji można wybrać do zainstalowania różne kolekcje pakietów. Jeśli nie ma problemów z miejscem na dysku to można zainstalować wszystkie, wtedy nie będzie problemu z brakującymi pakietami i błędami. Po wybraniu kolekcji brakujące pliki są pobierane z internetu. Pełna instalacja programu zajmuje około 8GB. Najlepiej zostawić instalację na noc, ponieważ proces zabiera sporo czasu. Warto ustawić komputer tak, aby się nie wyłączył lub nie uśpił. Warto także przed instalacją zablokować antywirusa, ponieważ może blokować niektóre z komponentów.

Następnie instalujemy T_EXstudio⁴. Ściągamy plik instalacyjny zajmujący około 120MB. Instalacja przebiega standardowo.

Następnym krokiem jest ustawienie w T_EXStudio kolejności budowania projektu. Należy wybrać zakładkę: „Opcje/Konfiguruj T_EXstudio...”. W otwartym oknie przechodzimy na zakładkę „Zbuduj”. Na rysunku 1.2 (s. 6) pokazany jest zrzut ekranu z konfiguracją. W linijce „Zbuduj i pokaż” klikamy ikonę klucza, żeby przejść do konfiguracji polecenia. W otwartym oknie ustawić kolejność tak jak pokazano na rysunku.

³Instalka na stronie <https://www.tug.org/texlive/acquire-netinstall.html>[**www2**].

⁴Plik instalacyjny na stronie <https://www.texstudio.org>[**www3**].



Rys. 1.2. Ustawienie TeXstudio

2. Określenie wymagań szczegółowych

2.1. Język programowania

Język w którym będzie programowana aplikacja to Kotlin. Język ten jest nowoczesnym językiem ogólnego przeznaczenia w pełni kompatybilnym z Javą. Kotlin jest językiem programowania na platformę Android. Kotlin działa na JVM co oznacza że można go bez problemu używać razem z istniejącym kodem Javy. Zalety Kotlin: Nowoczesne funkcje języka, Wysoka kompatybilność z Javą, Lepsza wydajność i efektywność, Bezpieczeństwo i stabilność.

Podczas tworzenia projektu będziemy korzystali ze środowiska programistycznego Android studio. Środowisko to pozwala na projektowanie, pisanie, testowanie i debugowanie aplikacji mobilnych. Obsługuje różne języki programowania, w tym Kotlin.

2.2. Podstawowe funkcje aplikacji

- **Przebyty dystans** - Aplikacja ma za zadanie mierzyć dystans jaki użytkownik przebiegł podczas jednego treningu. Aby wykonać ten pomiar wymagane jest użycia czujnika GPS, który jest jednym z najdokładniejszych narzędzi do mierzenia przebytego dystansu.
- **Ilość kroków**- Aplikacja ma zliczać ilość kroków podczas jednego treningu. Do tego należy wykorzystać czujnik Akcelerometr. Czujnik ten mierzy przyspieszenie w trzech osiach (płaszczyznach): X, Y i Z, co oznacza, że rejestruje zmiany prędkości i kierunku ruchu telefonu w trzech wymiarach przestrzeni.
- **Zaznaczanie przebytej trasy na mapie** - Aplikacja ma za zadanie pokazywać na bieżąco na mapie przebytą trasę zaznaczając ją linią (Od punktu w którym trening został rozpoczęty, aż do punktu w którym trening zostanie zakończony). Żeby wykonać tą funkcję należy zintegrować aplikację z usługą dostarczającą mapy, w tym wypadku będzie to OpenStreetMap. Należy również ustawić interwał aktualizacji lokalizacji co 6-8 sekund w celu zmniejszenia zużycia baterii
- **Prędkość poruszania się** - Aplikacja wyświetla prędkość z jaką użytkownik porusza się w danym momencie, do tej funkcji należy również wykorzystać czujnik GPS,
- **Licznik spalonych kalorii** - Aplikacja po zakończonym trenningu ma pokazywać spalone kalorie podczas jednostki treningowej. Do tego będzie korzystała ze wzoru:

$$\text{Spalone kalorie} = \text{MET} \times \text{masa ciała [kg]} \times \text{czas}$$

Gdzie wartość MET jest zależna od rodzaju aktywności i jej intensywności dla przykładu, średnia prędkość biegu podczas treningu wynosiła 8km/h to wartość $\text{MET} = 9$. Użytkownik na wstępie będzie musiał podać swoje parametry takie jak: waga, wzrost, wiek, płeć.

- **Liczenie czasu wykonywanej aktywności** - Aplikacja liczy czas od momentu startu aktywności aż do jego zakończenia
- **Zapisywanie historii aktywności fizycznych** - Aplikacja zapisuje do bazy danych historię jednostek treningowych.

- **Podsumowanie** - Aplikacja w każdą sobotę wysyła tygodniowe podsumowanie aktywności fizycznych. Zlicza ile w danym tygodniu zostało przebiegniętych kilometrów, ile zostało zrobionych kroków, ile zostało spalonych Kalorii.
- **Funkcja podjaśniania aplikacji** - Aplikacja dostosowuje jasność aplikacji dla lepszego komfortu użytkownika w zależności od naświetlenia. W momencie kiedy czujnik wykryje że jest wysokie natężenie światła aplikacja stanie się jaśniejsza.

3. Projektowanie

4. Implementacja

5. Testowanie

6. Podręcznik użytkownika

Spis rysunków

1.1. Logo	3
1.2. Ustawienie TeXstudio	6

Spis tabel

1.1. Tabelka przykładowa	3
------------------------------------	---

Spis listingów

1.	Przykładowy kod 001	3
2.	Przykładowy kod 002	4