

Aplikacja mobilna dla biegacza - GitFitScrub

Michał Ptak, Katarzyna Poręba, Jan Michalik

Zespołowe przedsięwzięcie inżynierskie

Informatyka

Rok. akad. 2017/2018, sem. I

Prowadzący: dr hab. Marcin Mazur

Spis treści

1	Opis projektu	4
1.1	Członkowie zespołu	4
1.2	Cel projektu (produkt)	4
1.3	Potencjalny odbiorca produktu (klient)	4
1.4	Metodyka	4
2	Wymagania użytkownika	4
2.1	User story 1	4
2.2	User story 2	4
2.3	User story 3	4
2.4	User story 4	4
2.5	User story 5	5
2.6	User story 6	5
2.7	User story 7	5
2.8	User story 8	5
2.9	User story 9 (opcjonalne)	5
2.10	User story 10 (opcjonalne)	5
2.11	User story 11 (opcjonalne)	5
2.12	User story 12 (opcjonalne)	5
2.13	User story 13 (opcjonalne)	5
2.14	User story 14 (opcjonalne)	5
2.15	User story 15 (opcjonalne)	6
2.16	User story 16 (opcjonalne)	6
2.17	User story 17 (opcjonalne)	6
2.18	User story 18 (opcjonalne)	6
2.19	User story 19 (opcjonalne)	6
2.20	User story 20 (opcjonalne)	6
3	Harmonogram	6
3.1	Rejestr zadań (Product Backlog)	6
3.2	Sprint 1	6
3.3	Sprint 2	7
3.4	Sprint 3	7
3.5	Sprint 4	7
3.6	Sprint 5	7
3.7	Sprint 6	7
4	Product Backlog	8
4.1	Backlog Item 1	8
4.2	Backlog Item 2	8
4.3	Backlog Item 3	8
4.4	Backlog Item 4	8
4.5	Backlog Item 5	9
4.6	Backlog Item 6	9
4.7	Backlog Item 7	9
4.8	Backlog Item 8	9
4.9	Backlog Item 9	10
4.10	Backlog Item 10	10

4.11	Backlog Item 11	10
4.12	Backlog Item 12	11
4.13	Backlog Item 13	11
4.14	Backlog Item 14 (opcjonalne)	11
4.15	Backlog Item 15 (opcjonalne)	11
4.16	Backlog Item 16 (opcjonalne)	12
4.17	Backlog Item 17 (opcjonalne)	12
4.18	Backlog Item 18 (opcjonalne)	12
4.19	Backlog Item 19 (opcjonalne)	12
4.20	Backlog Item 20 (opcjonalne)	13
4.21	Backlog Item 21 (opcjonalne)	13
4.22	Backlog Item 22 (opcjonalne)	13
4.23	Backlog Item 23 (opcjonalne)	13
4.24	Backlog Item 24 (opcjonalne)	14
4.25	Backlog Item 25 (opcjonalne)	14
4.26	Backlog Item 26 (opcjonalne)	14
4.27	Backlog Item 27 (opcjonalne)	14
4.28	Backlog Item 28 (opcjonalne)	15
5	Sprint 1	15
5.1	Cel	15
5.2	Sprint Planning/Backlog	15
5.3	Realizacja	15
5.4	Sprint Review/Demo	18
6	Sprint 2	19
6.1	Cel	19
6.2	Sprint Planning/Backlog	19
6.3	Realizacja	19
6.4	Sprint Review/Demo	26
7	Sprint 3	26
7.1	Cel	26
7.2	Sprint Planning/Backlog	27
7.3	Realizacja	27
7.4	Sprint Review/Demo	31
8	Sprint 4	31
8.1	Cel	31
8.2	Sprint Planning/Backlog	31
8.3	Realizacja	31
8.4	Sprint Review/Demo	37
9	Sprint 5	38
9.1	Cel	38
9.2	Sprint Planning/Backlog	38
9.3	Realizacja	38
9.4	Sprint Review/Demo	50

10 Sprint 6	50
10.1 Cel	50
10.2 Sprint Planning/Backlog	50
10.3 Realizacja	51
10.4 Sprint Review/Demo	51

1 Opis projektu

1.1 Członkowie zespołu

1. Michał Ptak (kierownik projektu).
2. Katarzyna Poręba.
3. Jan Michalik.

1.2 Cel projektu (produkt)

Naszym celem jest zbudowanie aplikacji mobilnej na system operacyjny Android przy użyciu map Google, która pobiera informacje poruszającej się osoby z modułu GPS i gromadzi statystyki dotyczące długości trasy, prędkości czy spalanych kalorii, a także zapisuje poczynania w bazie danych.

1.3 Potencjalny odbiorca produktu (klient)

Klient jest biegaczem oczekującym w pełni darmowej aplikacji bez reklam, która będzie śledzić trasę, po której się porusza oraz wyświetlać aktualne statystyki. Aktualnie żadna dostępna na rynku aplikacja nie spełnia odbiorcy produktu.

1.4 Metodyka

Projekt będzie realizowany przy użyciu (zaadaptowanej do istniejących warunków) metodyki *Scrum*.

2 Wymagania użytkownika

2.1 User story 1

Jako użytkownik, chcę mieć dostęp do mapy podczas treningu, aby wiedzieć jaką trasą się poruszam.

2.2 User story 2

Jako użytkownik, chcę mieć możliwość wciśnięcia przycisku "Start", aby rozpocząć pomiar.

2.3 User story 3

Jako użytkownik, chcę mieć możliwość wciśnięcia przycisku "Stop", aby zakończyć pomiar.

2.4 User story 4

Jako użytkownik, chcę wiedzieć czas, który upłynął od rozpoczęcia pomiaru biegu, aby kontrolować jego upływ.

2.5 User story 5

Jako użytkownik, chcę widzieć przebyty dystans, aby kontrolować jego długość.

2.6 User story 6

Jako użytkownik, chcę widzieć prędkość chwilową, z jaką się poruszam, aby zachować określone tempo.

2.7 User story 7

Jako użytkownik, chcę zobaczyć podsumowanie odbytego treningu, aby ocenić jego rezultat.

2.8 User story 8

Jako użytkownik, chcę mieć możliwość włączenia trybu nocnego, aby zmniejszyć zmęczenie wzroku po zmroku.

2.9 User story 9 (opcjonalne)

Jako użytkownik niezarejestrowany, chcę mieć możliwość rejestracji, aby móc się zalogować.

2.10 User story 10 (opcjonalne)

Jako użytkownik zarejestrowany, chcę mieć możliwość zalogowania się, aby móc korzystać ze wszystkich funkcji aplikacji.

2.11 User story 11 (opcjonalne)

Jako użytkownik zalogowany, chcę mieć możliwość wylogowania się, aby móc zalogować się na inne konto.

2.12 User story 12 (opcjonalne)

Jako użytkownik zalogowany, chcę mieć możliwość ustawienia swojego pseudonimu, aby wiedzieć czy jestem zalogowany na swoim koncie.

2.13 User story 13 (opcjonalne)

Jako użytkownik, chcę aby statystyki z odbytych treningów zostały zapisane w historii, celem ich późniejszej analizy.

2.14 User story 14 (opcjonalne)

Jako użytkownik, chcę widzieć w statystykach spalone kalorie, by móc dostosować swoją dietę.

2.15 User story 15 (opcjonalne)

Jako użytkownik zalogowany, chcę mieć możliwość usunięcia (całej lub części) historii treningów, aby rozpocząć nowy cykl treningowy.

2.16 User story 16 (opcjonalne)

Jako użytkownik zalogowany, chcę by aplikacja pamiętała ostatnio użyty login, aby nie wpisywać go ręcznie przy kolejnym logowaniu.

2.17 User story 17 (opcjonalne)

Jako użytkownik zalogowany, chcę mieć możliwość ręcznego zatrzymania pomiaru czasu oraz dystansu, aby napotkane przeszkody (np. czerwone światło) nie fałszowały statystyk.

2.18 User story 18 (opcjonalne)

Jako użytkownik zalogowany, chcę mieć możliwość podjęcia wyzwania pokonania określonych odległości (np. z Ziemi do Księżyca), aby mieć motywację do kontynuowania treningu.

2.19 User story 19 (opcjonalne)

Jako użytkownik zalogowany, chcę mieć możliwość udostępnienia swoich osiągnięć na portalach społecznościowych, aby podzielić się rezultatami ze znajomymi.

2.20 User story 20 (opcjonalne)

Jako użytkownik, chcę mieć możliwość ustawienia powiadomień po określonym przez siebie odcinku trasy, by nie być zmuszonym do ciągłego sprawdzania aplikacji.

3 Harmonogram

3.1 Rejestr zadań (Product Backlog)

- Data rozpoczęcia: 24.10.2017.
- Data zakończenia: 31.10.2017.

3.2 Sprint 1

- Data rozpoczęcia: 31.10.2017.
- Data zakończenia: 14.11.2017.
- Scrum Master: Katarzyna Poręba.
- Product Owner: Michał Ptak.
- Development Team: Michał Ptak, Katarzyna Poręba, Jan Michalik.

3.3 Sprint 2

- Data rozpoczęcia: 14.11.2017.
- Data zakończenia: 28.11.2017.
- Scrum Master: Michał Ptak.
- Product Owner: Jan Michalik.
- Development Team: Michał Ptak, Katarzyna Poręba, Jan Michalik.

3.4 Sprint 3

- Data rozpoczęcia: 28.11.2017.
- Data zakończenia: 12.12.2017.
- Scrum Master: Jan Michalik.
- Product Owner: Katarzyna Poręba.
- Development Team: Michał Ptak, Katarzyna Poręba, Jan Michalik.

3.5 Sprint 4

- Data rozpoczęcia: 12.12.2017.
- Data zakończenia: 19.12.2017.
- Scrum Master: Katarzyna Poręba.
- Product Owner: Michał Ptak.
- Development Team: Michał Ptak, Katarzyna Poręba, Jan Michalik.

3.6 Sprint 5

- Data rozpoczęcia: 19.12.2017.
- Data zakończenia: 09.01.2018.
- Scrum Master: Michał Ptak.
- Product Owner: Jan Michalik.
- Development Team: Michał Ptak, Katarzyna Poręba, Jan Michalik.

3.7 Sprint 6

- Data rozpoczęcia: 09.01.2018.
- Data zakończenia: 16.01.2018.
- Scrum Master: Michał Ptak.
- Product Owner: Katarzyna Poręba.
- Development Team: Michał Ptak, Katarzyna Poręba, Jan Michalik.

4 Product Backlog

4.1 Backlog Item 1

Tytuł zadania. Przygotowanie zestawu narzędzi SDK.

Opis zadania. Dodanie do zestawu SDK: Google Services, nową dystrybucję Androida (wersja Oreo), debugowanie poprzez USB. Wykonanie prostego programu testującego działanie kompilatora oraz zainstalowanej dystrybucji.

Priorytet. 10.

Definition of Done. Poprawne zainstalowanie wybranych narzędzi SDK na stacjach roboczych.

4.2 Backlog Item 2

Tytuł zadania. Stworzenie wstępnej wersji aplikacji.

Opis zadania. Utworzenie aplikacji zawierającej puste menu główne.

Priorytet. 10.

Definition of Done. Aplikacja po uruchomieniu wyświetla menu główne.

4.3 Backlog Item 3

Tytuł zadania. Dodanie uprawnień aplikacji.

Opis zadania. Dodanie uprawnień aplikacji do korzystania z Internetu i modułu GPS, co weryfikowane jest w 4.4.

Priorytet. 10.

Definition of Done. Aplikacja posiada możliwość korzystania z Internetu i nawigacji.

4.4 Backlog Item 4

Tytuł zadania. Stworzenie przycisku "Rozpocznij" w menu głównym oraz nadanie mu funkcjonalności.

Opis zadania. Utworzenie nowej aktywności uruchamiającej mapę Google i nanoszącej na nią znacznika wskazującego geolokalizację użytkownika. Utworzenie przycisku i nadanie mu akcji uruchomienia tejże aktywności. Zaprojektowanie szaty graficznej menu głównego i stworzenie logo oraz ikony aplikacji.

Priorytet. 8.

Definition of Done. Po naciśnięciu przycisku "Rozpocznij" uruchomi moduł GPS i pokaże znacznik wskazujący aktualną pozycję użytkownika na mapie.

4.5 Backlog Item 5

Tytuł zadania. Stworzenie alertu o braku aktywacji modułu GPS.

Opis zadania. Utworzenie metody w klasie MapsActivity sprawdzającej stan aktywności modułu GPS, oraz powiadamiającej użytkownika o wyłączonym GPS-ie. Po zatwierdzeniu alertu aplikacja przenosi użytkownika do ustawień systemowych Android™.

Priorytet. 6.

Definition of Done.

4.6 Backlog Item 6

Tytuł zadania. Pobranie informacji z modułu GPS.

Opis zadania. Utworzenie nowej aktywności pobierającej surowe dane geograficzne oraz gromadzenie ich w pamięci operacyjnej. Weryfikacja poprawności uzyskanych danych poprzez wydruk w konsoli dewelopera.

Priorytet. 9.

Definition of Done. Surowe dane geograficzne znajdują się w pamięci operacyjnej systemu.

4.7 Backlog Item 7

Tytuł zadania. Utworzenie pustego panelu statystyk i przycisku "Start" oraz nadanie mu funkcjonalności.

Opis zadania. Utworzenie nowego obiektu button wywołującego metody rozpoczynające pomiary po jego wciśnięciu. Wydzielenie obszaru, na którym będą wyświetlane statystyki.

Priorytet. 9.

Definition of Done. U dołu mapy, wyświetlającej naszą aktualną pozycję, widnieje przycisk rozpoczynający trening, a pod nim panel ze statystykami.

4.8 Backlog Item 8

Tytuł zadania. Przetworzenie zgromadzonych informacji.

Opis zadania. Utworzenie funkcji kalkulujących dystans, prędkość, ilość spalonych kcal i upływ czasu, które zostają wywołane po naciśnięciu przycisku "Rozpocznij nowy trening". Przechowywanie wartości zwracanych przez funkcje w pamięci operacyjnej systemu.

Priorytet. 9.

Definition of Done. Dane zwracane przez funkcję mają odwzorowanie w rzeczywistości.

4.9 Backlog Item 9

Tytuł zadania. Modyfikacja panelu zawierającego statystyki aktualnego treningu.

Opis zadania. Utworzenie nowej aktywności pobierającej przetworzone dane geograficzne. Określenie pozycji wyświetlanych parametrów typu: czas, dystans, prędkość. Sprzężenie tejże aktywności z pozycjami statystyk celem ich wyświetlenia użytkownikowi.

Priorytet. 7.

Definition of Done. Wyświetlenie przetworzonych informacji pobranych z modułu GPS.

4.10 Backlog Item 10

Tytuł zadania. Rysowanie polilinii na mapie.

Opis zadania. Utworzenie metody tworzącej ślad na mapie przemieszczającego się użytkownika, na podstawie informacji z modułu GPS.

Priorytet. 8.

Definition of Done.

4.11 Backlog Item 11

Tytuł zadania. Stworzenie przycisku "Zakończ trening" oraz nadanie mu funkcjonalności.

Opis zadania. Utworzenie nowej aktywności wywołującej metodę odpowiadającą za zakończenie pomiarów i rysowanie polilinii na mapie, a także otwierającą panel z podsumowaniem treningu — wszystkich wyznaczonych statystyk. Wciśnięcie przycisku będzie powodować również zapis statystyk w pamięci aplikacji.

Priorytet. 9.

Definition of Done. Po naciśnięciu "Zakończ trening" aplikacja zatrzymuje pomiary i wyświetla podsumowanie treningu.

4.12 Backlog Item 12

Tytuł zadania. Stworzenie przycisku "Ustawienia" w menu głównym oraz nadanie mu funkcjonalności.

Opis zadania. Stworzenie nowej aktywności uruchamiającej listę dostępnych ustawień.

Priorytet. 5.

Definition of Done. Po naciśnięciu przycisku "Ustawienia" w menu głównym otwiera się lista ustawień.

4.13 Backlog Item 13

Tytuł zadania. Dodanie funkcji trybu nocnego do panelu ustawień.

Opis zadania. Dodanie przełącznika do panelu ustawień zmieniającego styl mapy oraz panelu statystyk na "nocny" lub "dzienny".

Priorytet. 5.

Definition of Done. Po naciśnięciu przełącznika mapa oraz panel statystyk zmieniają styl wyglądu na "dzienny" lub "nocny".

4.14 Backlog Item 14 (opcjonalne)

Tytuł zadania. Dodanie aktywności logowania użytkownika.

Opis zadania. Dodanie możliwości logowania za pośrednictwem konta Google.

Priorytet. 6.

Definition of Done. Konto Google użytkownika zostaje połączone z aplikacją GitFitScrub. Poprawność połączenia weryfikowana na podstawie strony "Aplikacje połączone z Twoim kontem" <https://myaccount.google.com/permissions>.

4.15 Backlog Item 15 (opcjonalne)

Tytuł zadania. Dodanie aktywności wylogowania użytkownika.

Opis zadania. Dodanie możliwości wylogowania się z konta Google.

Priorytet. 6.

Definition of Done. Konto Google użytkownika zostaje rozłączone z aplikacji GitFitScrub.

4.16 Backlog Item 16 (opcjonalne)

Tytuł zadania. Stworzenie przycisku "Zaloguj" w ustawieniach oraz nadanie mu funkcjonalności.

Opis zadania. Wywołanie metody połączenia się z kontem Google.

Priorytet. 5.

Definition of Done. Po wprowadzeniu poprawnych danych użytkownik zostaje pomyślnie uwierzytelniony.

4.17 Backlog Item 17 (opcjonalne)

Tytuł zadania. Stworzenie przycisku "Wyloguj" w ustawieniach oraz nadanie mu funkcjonalności.

Opis zadania. Wywołanie metody rozłączenia się z kontem Google.

Priorytet. 5.

Definition of Done. Po naciśnięciu przycisku użytkownik zostaje pomyślnie wylogowany.

4.18 Backlog Item 18 (opcjonalne)

Tytuł zadania. Utworzenie miejsca, w którym archiwizowane będą statystyki użytkownika zalogowanego.

Opis zadania. Wygenerowanie certyfikatu do autoryzacji komunikacji aplikacji z chmurą. Sprawdzenie komunikacji baza - aplikacja poprzez wysłanie testowych informacji na chmurę.

Priorytet. 5.

Definition of Done. Aplikacja bez przeszkód komunikuje się z bazą danych w chmurze Google Drive.

4.19 Backlog Item 19 (opcjonalne)

Tytuł zadania. Ustawienie pseudonimu użytkownika zalogowanego.

Opis zadania. Utworzenie nowego pola w ustawieniach, którego celem będzie ustawienie aktualnego pseudonimu.

Priorytet. 3.

Definition of Done. Użytkownik zalogowany ma możliwość ustawienia pseudonimu.

4.20 Backlog Item 20 (opcjonalne)

Tytuł zadania. Stworzenie przycisku "Historia" w menu głównym oraz nadanie mu funkcjonalności.

Opis zadania. Utworzenie nowej aktywności do wyświetlania historii treningów.

Priorytet. 5.

Definition of Done. Użytkownik po naciśnięciu przycisku historia widzi listę odbytych treningów.

4.21 Backlog Item 21 (opcjonalne)

Tytuł zadania. Gromadzenie danych dotyczących treningów w chmurze Google Drive.

Opis zadania. Po nawiązaniu połączenia z chmurą następuje przesył zgromadzonych informacji do bazy danych. Weryfikacja danych zgromadzonych w bazie poprzez sprawdzenie zawartości za pomocą interfejsu administratora.

Priorytet. 5.

Definition of Done. Dane są poprawnie gromadzone w bazie danych.

4.22 Backlog Item 22 (opcjonalne)

Tytuł zadania. Prezentowanie danych w historii.

Opis zadania. Stworzenie nowej aktywności prezentującej użytkownikowi statystyki w panelu Historia, dostępnego z menu głównego.

Priorytet. 5.

Definition of Done. Dane są poprawnie wyświetlane.

4.23 Backlog Item 23 (opcjonalne)

Tytuł zadania. Usuwanie danych z historii.

Opis zadania. Utworzenie metody wysyłające polecenie kasujące dane zapisane w historii.

Priorytet. 4.

Definition of Done. Dane są poprawnie usuwane z bazy danych i nie znajdują się już w panelu historia.

4.24 Backlog Item 24 (opcjonalne)

Tytuł zadania. Zapamiętanie loginu ostatniego użytkownika.

Opis zadania. Wykorzystanie klasy SharedPreferences w celu zapamiętania loginu użytkownika. Sprawdzenie poprawności funkcjonalności zapisanego loginu za pomocą metody get wywołanej na obiekcie SharedPreferences.

Priorytet. 4.

Definition of Done. Po wylogowaniu się w polu login widnieje ostatnio użyta nazwa użytkownika.

4.25 Backlog Item 25 (opcjonalne)

Tytuł zadania. Zatrzymanie pomiarów czasu i dystansu.

Opis zadania. Stworzenie przycisku pauza. Dodanie metody wstrzymującej pomiary czasu i dystansu do klasy odpowiadającej za pomiary oraz ich powiązanie.

Priorytet. 5.

Definition of Done. Po wciśnięciu przycisku pauza pomiary czasu i dystansu są wstrzymane. Po ponownym naciśnięciu przycisku pomiary są wznowiane.

4.26 Backlog Item 26 (opcjonalne)

Tytuł zadania. Utworzenie wyzwań.

Opis zadania. Stworzenie listy wyzwań. Dodanie do ustawień możliwości pojęcia własnego wyzwania polegającego na pokonaniu określonego dystansu w zadany przedział czasowy.

Priorytet. 3.

Definition of Done. Po aktywowaniu wyzwania odległości z historii (od momentu rozpoczęcia wyzwania) są dodawane do progresji aktywnego wyzwania.

4.27 Backlog Item 27 (opcjonalne)

Tytuł zadania. Udostępnianie wyniku podsumowania treningu na portalu społecznościom.

Opis zadania. Dodanie przycisku udostępnij w panelu podsumowanie, który wywołuje metodę wysyłającą dane o treningu na portal społecznościowy.

Priorytet. 3.

Definition of Done. Po udostępnieniu informacji dane są widoczne na profilu portalu społecznościowego użytkownika.

4.28 Backlog Item 28 (opcjonalne)

Tytuł zadania. Powiadomienia dźwiękowe.

Opis zadania. Dodanie metody odtwarzającej alert dźwiękowy po przebiegnięciu określonego czasu/dystansu. Dodanie do panelu ustawień pozycji powiadomienia, w której będzie można wybrać plik dźwiękowy oraz kryterium(czas/dystans) wywołania tejże metody.

Priorytet. 3.

Definition of Done. Po przebiegnięciu określonego czasu/dystansu użytkownik usłyszy alert dźwiękowy.

5 Sprint 1

5.1 Cel

Utworzenie wersji bazowej aplikacji zawierającej ekran menu głównego i przycisk Rozpocznij wyświetlający mapę Google i aktualną pozycję użytkownika.

5.2 Sprint Planning/Backlog

Tytuł zadania. Przygotowanie zestawu narzędzi SDK.

- Estymata: M.

Tytuł zadania. Stworzenie wstępnej wersji aplikacji.

- Estymata: L.

Tytuł zadania. Dodanie uprawnień aplikacji.

- Estymata: S.

Tytuł zadania. Stworzenie przycisku "Rozpocznij" w menu głównym oraz nadanie mu funkcjonalności.

- Estymata: L.

5.3 Realizacja

Tytuł zadania. Przygotowanie zestawu narzędzi SDK.

Wykonawca. Jan Michalik, Michał Ptak, Katarzyna Poręba.

Realizacja. Uruchamiamy SDK Manager, następnie instalujemy składniki: Android SDK Tools, Platform Tools, Build Tools, Android 8.1.0 (API 27), Google Play services, Google USB Driver. Czasochłonność zadania pokryła estymatę.

Tytuł zadania. Stworzenie wstępnej wersji aplikacji.

Wykonawca. Katarzyna Poręba.

Realizacja. Utworzenie nowego pustego projektu Android Studio. Dodanie nowej aktywności - MainMenu. Ustawienie jej jako domyślny ekran po uruchomieniu aplikacji.

```
<activity
    android:name=".MainMenu"
    android:label="GitFitScrub"
    android:theme="@style/AppTheme">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER"
        />
    </intent-filter>
</activity>
```

Tytuł zadania. Dodanie uprawnień aplikacji.

Wykonawca. Michał Ptak.

Realizacja. Uzupełnienie pliku AndroidManifest.xml o polecenia zezwalające na dostęp do Internetu i modułu GPS. Kompilacja kodu i wygenerowanie pliku .apk, następnie instalacja na urządzeniu z systemem Android. Podczas instalacji urządzenie poprawnie wyświetla informacje o wykorzystywanych funkcjonalnościach. Plik AndroidManifest.xml:

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.
    ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.
    ACCESS_COARSE_LOCATION" />
```

Tytuł zadania. Stworzenie przycisku "Rozpocznij" w menu głównym oraz nadanie mu funkcjonalności.

Wykonawca. Jan Michalik.

Realizacja. Uzupełnienie pliku activity_main_menu.xml o polecenia tworzące przycisk "Rozpocznij", oraz nadanie mu określonego wyglądu.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res
    /android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#dddddd"
    tools:context="pl.ppm.gitfitscrub.MainMenu">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_above="@+id/button2"
        android:layout_alignParentStart="true"
        android:src="@drawable/capture" />

    <Button
        android:id="@+id/button2"
        style="@style/Widget.AppCompat.Button.Colored"
        android:layout_width="400px"
        android:layout_height="wrap_content"
        android:layout_alignStart="@+id/button3"
        android:layout_centerVertical="true"
        android:layout_marginBottom="20dp"
        android:onClick="run"
        android:text="Rozpocznij"
        android:textSize="24sp" />
</RelativeLayout>
```

Nadanie funkcjonalności przyciskowi. Dodanie kodu do pliku MainMenu.java, który spowoduje, że po naciśnięciu przycisku otworzy się mapa z aktualną pozycją użytkownika.

Plik MainMenu.java - menu główne:

```
public class MainMenu extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_menu);
    }
    public void run(View view) {
        Intent intent = new Intent(getApplicationContext(),
        MapsActivity.class);
        startActivity(intent);
    }
};
```

Plik MapsActivity.java - ekran mapy:

```

public class MapsActivity extends FragmentActivity implements
    OnMapReadyCallback {

    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate ( savedInstanceState );
        setContentView ( R.layout.activity_maps );
        SupportMapFragment mapFragment = (SupportMapFragment)
        getSupportFragmentManager ()
            .findFragmentById ( R.id.map );
        mapFragment.getMapAsync ( this );
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        LatLng ns = new LatLng ( posY, posX );
        if (ActivityCompat.checkSelfPermission ( this, Manifest.
        permission.ACCESS_FINE_LOCATION ) == PackageManager.
        PERMISSION_GRANTED || ActivityCompat.checkSelfPermission (
        this, Manifest.permission.ACCESS_COARSE_LOCATION ) ==
        PackageManager.PERMISSION_GRANTED) {
            mMap.setMyLocationEnabled ( true );
        }
    }
}

```

Plik AndroidManifest.xml - ikona aplikacji:

```

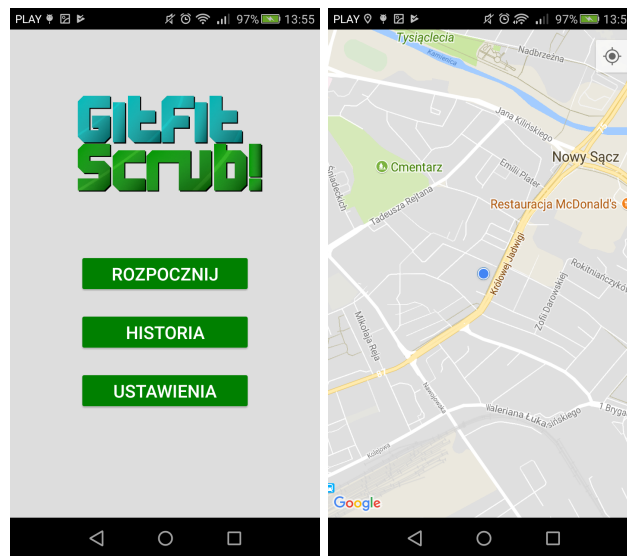
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportRtl="true"
android:theme="@android:style/Theme.Black.NoTitleBar.Fullscreen">

```

5.4 Sprint Review/Demo

Podczas przebiegu sprintu wykonane zostały wszystkie zaplanowane zadania oraz pokryły one estymaty.

Demonstracja przyrostu produktu:



Ekran menu głównego (po lewej) oraz widok mapy, po wciśnięciu przycisku "Rozpocznij", z oznaczoną pozycją urządzenia.

6 Sprint 2

6.1 Cel

Pobranie i przetworzenie danych na podstawie geolokalizacji użytkownika oraz utworzenie przycisku wysuwającego panel wyświetlający wybrane informacje (aktualną prędkość, czas i dystans).

6.2 Sprint Planning/Backlog

Tytuł zadania. Stworzenie alertu o braku aktywacji modułu GPS.

- Estymata: M.

Tytuł zadania. Pobranie informacji z modułu GPS.

- Estymata: L.

Tytuł zadania. Stworzenie przycisku "Rozpocznij nowy trening" oraz nadanie mu funkcjonalności.

- Estymata: L.

Tytuł zadania. Przetworzenie zgromadzonych informacji.

- Estymata: XL.

6.3 Realizacja

Tytuł zadania. Stworzenie alertu o braku aktywacji modułu GPS.

Wykonawca. Katarzyna Poręba.

Realizacja. Utworzenie alertu, powiadamiającego użytkownika o wyłączonym GPS-ie, przy pomocy biblioteki `android.support.v7.app.AlertDialog`.
Utworzenie funkcji wyświetlającej alert:

```
private void buildAlertMessageNoGps() {
    final AlertDialog.Builder builder = new AlertDialog.
        Builder(this);
    builder.setMessage("Twój GPS jest wyłączony. Czy chcesz
        go włączyć?")
        .setCancelable(false)
        .setPositiveButton("Tak", new DialogInterface.
            OnClickListener() {
                public void onClick(final DialogInterface
                    dialog, @SuppressWarnings("unused") final int id) {
                    startActivity(new Intent(android.provider
                        .Settings.ACTION_LOCATION_SOURCE_SETTINGS));
                }
            })
        .setNegativeButton("Nie", new DialogInterface.
            OnClickListener() {
                public void onClick(final DialogInterface
                    dialog, @SuppressWarnings("unused") final int id) {
                    dialog.cancel();
                }
            });
    final AlertDialog alert = builder.create();
    alert.show();
}
```

Utworzenie warunku sprawdzającego, czy GPS jest wyłączony:

```
if (!locationManager.isProviderEnabled(LocationManager.
    GPS_PROVIDER)) {buildAlertMessageNoGps();}
```

Tytuł zadania. Pobranie informacji z modułu GPS.

Wykonawca. Michał Ptak.

Realizacja. Utworzenie metod nasłuchujących zmian w informacjach dostarczanych przez moduł GPS w oparciu o Google Play Services.
Plik `MapsActivity.java`:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);

    mFusedLocationClient = LocationServices.
        getFusedLocationProviderClient(this);
}
```

```

        mapFrag = (SupportMapFragment) getSupportFragmentManager
().findFragmentById(R.id.map);
        mapFrag.getMapAsync(this);

        (...)
    }

    @Override
    protected void onPause() {
        super.onPause();

        if (mFusedLocationClient != null) {
            mFusedLocationClient.removeLocationUpdates(
mLocationCallback);
        }
    }

    @Override
    public void onMapReady(GoogleMap googleMap)
    {
        mGoogleMap=googleMap;
        mGoogleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);

        if (android.os.Build.VERSION.SDK_INT >= Build.
VERSION_CODES.M) {
            if (ContextCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_FINE_LOCATION)
                == PackageManager.PERMISSION_GRANTED) {
                buildGoogleApiClient();
                mGoogleMap.setMyLocationEnabled(true);
            } else {
                checkLocationPermission();
            }
        }
        else {
            buildGoogleApiClient();
            mGoogleMap.setMyLocationEnabled(true);
        }
    }

    protected synchronized void buildGoogleApiClient() {
        mGoogleApiClient = new GoogleApiClient.Builder(this)
            .addConnectionCallbacks(this)
            .addOnConnectionFailedListener(this)
            .addApi(LocationServices.API)
            .build();
        mGoogleApiClient.connect();
    }
}

```

```

@Override
public void onConnected(Bundle bundle) {
    mLocationRequest = new LocationRequest();
    mLocationRequest.setInterval(1000);
    mLocationRequest.setFastestInterval(100);
    mLocationRequest.setPriority(LocationRequest.
PRIORITY_BALANCED_POWER_ACCURACY);
    if (ContextCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION)
        == PackageManager.PERMISSION_GRANTED) {
        mFusedLocationClient.requestLocationUpdates(
mLocationRequest, mLocationCallback, Looper.myLooper());
    }
}

LocationCallback mLocationCallback = new LocationCallback(){
    @Override
    public void onLocationResult(LocationResult
locationResult) {
        for (Location location : locationResult.getLocations
()) {
            Log.i("MapsActivity", "Location: " + location.
getLatitude() + " " + location.getLongitude());
            mLastLocation = location;
            if (mCurrLocationMarker !=null) {
                mCurrLocationMarker.remove();
            }

            LatLng latLng = new LatLng(location.getLatitude()
, location.getLongitude());
            MarkerOptions markerOptions = new MarkerOptions()
;
            markerOptions.position(latLng);
            markerOptions.title("Tu jestem!");
            markerOptions.icon(BitmapDescriptorFactory.
defaultMarker(BitmapDescriptorFactory.HUE_MAGENTA));
            mCurrLocationMarker = mGoogleMap.addMarker(
markerOptions);

            mGoogleMap.moveCamera(CameraUpdateFactory.
newLatLngZoom(latLng, 18));

        }
    }
};

@Override
public void onConnectionSuspended(int i) {}

```

```
@Override
public void onConnectionFailed(ConnectionResult
connectionResult) {}
```

Tytuł zadania. Stworzenie przycisku "Rozpocznij nowy trening" oraz nadanie mu funkcjonalności.

Wykonawca. Katarzyna Poręba, Jan Michalik.

Realizacja. Początkowym założeniem było, że po naciśnięciu przycisku panel będzie się otwierał i będą rozpoczęte pomiary. Panel został utworzony, jednak nie jest otwierany przez przycisk. Został on umieszczony bezpośrednio w aktywności mapy. Nazwę przycisku zmieniono na "Start", rozpoczyna on pomiary.

Plik activity_maps.xml:

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/
    android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <fragment xmlns:android="http://schemas.android.com/apk/res/
        android"
        xmlns:map="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.
        SupportMapFragment"
        android:layout_width="match_parent"
        android:layout_height="450dp"
        tools:context="pl.ppm.gitfitscrub.MapsActivity" />

    <Button
        android:id="@+id/start_button"
        style="@style/Widget.AppCompat.Button.Colored"
        android:layout_width="match_parent"
        android:layout_height="30dp"
        android:background="@color/colorAccent"
        android:padding="5dp"
        android:text="start"
        android:textColor="#ffffff" />

    <Chronometer
        android:id="@+id/chronometer"
```



```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textColor="#000000"
        android:textSize="20dp" />

<TextView
    android:id="@+id/speedText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Predkosc: "
    android:textColor="#000000"
    android:textSize="20dp"/>

<TextView
    android:id="@+id/distanceText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Dystans: "
    android:textColor="#000000"
    android:textSize="20dp"/>

</LinearLayout>

Plik MapsActivity.java:

@Override
protected void onCreate(Bundle savedInstanceState) {
    (...)
    mStartButton = (Button) findViewById(R.id.start_button);
    mChronometer = (Chronometer) findViewById(R.id.
chronometer);

    distanceText = (TextView) findViewById(R.id.distanceText)
;
    speedText = (TextView) findViewById(R.id.speedText);

    mStartButton.setOnClickListener(new View.OnClickListener
() {
        @Override
        public void onClick(View view) {
            mChronometer.setBase(SystemClock.elapsedRealtime
());
            mChronometer.start();
        }
    });
}

```

```

LocationCallback mLocationCallback = new LocationCallback(){
    @Override
    public void onLocationResult(LocationResult
locationResult) {
        for (Location location : locationResult.getLocations
()) {
            (...)
            updateUI();
        }
    }
};

private void updateUI() {
    distance = distance + (lStart.distanceTo(lEnd) / 1000.00)
;
    if (speed > 0.0)
        speedText.setText("predkosc: " + new DecimalFormat("#.##").format(speed) + " km/h");
    else
        speedText.setText("obliczam predkosc...");

    distanceText.setText("dystans: " + new DecimalFormat("#.###").format(distance) + " km");

    lStart = lEnd;
}

```

Tytuł zadania. Przetworzenie zgromadzonych informacji.

Wykonawca. Michał Ptak.

Realizacja. Dystans w kilometrach liczony jest poprzez dodawanie długości kolejnych pokonywanych odcinków (droga pokonana pomiędzy dwoma aktualizacjami lokalizacji) do dystansu całkowitego, przy pomocy metody `distanceTo()`. Do pobierania prędkości wykorzystana została metoda `getSpeed()`, zwracająca prędkość w m/s, która została przeliczona na km/h.

Plik `MapsActivity.java`:

```

@Override
protected void onPause() {
    (...)
    distance = 0;
}

LocationCallback mLocationCallback = new LocationCallback(){
    @Override
    public void onLocationResult(LocationResult
locationResult) {

```

```

        for (Location location : locationResult.getLocations
    ()) {
        (...)
        if (lStart == null) {
            lStart = mLastLocation;
            lEnd = mLastLocation;
        } else {
            lEnd = mLastLocation;
        }

        speed = location.getSpeed() * 3.6;

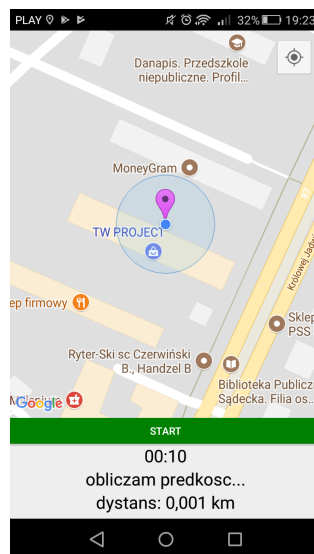
    }
}
};

```

6.4 Sprint Review/Demo

Założony cel sprintu został osiągnięty. Estymowana czasochłonność dla zadań 2-4 była błędna - zadania 2 i 3 wymagały większego nakładu pracy niż założono, natomiast zadanie ostatnie - przeciwnie.

Demonstracja przyrostu produktu:



Widok mapy oraz panelu ze statystykami po wciśnięciu przycisku "START", z oznaczoną znacznikiem pozycją urządzenia.

7 Sprint 3

7.1 Cel

Uaktualnienie panelu statystyk, rysowanie śladu na mapie oraz utworzenie aktywności "Ustawienia".

7.2 Sprint Planning/Backlog

Tytuł zadania. Modyfikacja panelu zawierającego statystyki aktualnego treningu.

- Estymata: XL

Tytuł zadania. Rysowanie polilinii na mapie.

- Estymata: XL

Tytuł zadania. Stworzenie przycisku "Ustawienia" w menu głównym oraz nadanie mu funkcjonalności.

- Estymata: M

7.3 Realizacja

Tytuł zadania. Modyfikacja panelu zawierającego statystyki aktualnego treningu.

Wykonawca. Jan Michalik, Michał Ptak

Realizacja. Zaplanowanie struktury interfejsu panelu statystyk. Wprowadzenie zmian do pliku activity_maps.xml.

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="bottom"
    android:orientation="vertical">

    <TextView
        android:id="@+id/speedText"
        android:layout_width="match_parent"
        android:layout_height="70dp"
        android:gravity="center"
        android:textColor="#000000"
        android:textSize="42sp" />

    <TextView
        android:id="@+id/textView6"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="PREDKOSC"
        android:textAlignment="center" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
```

```

<Chronometer
    android:id="@+id/chronometer"
    android:layout_width="match_parent"
    android:layout_height="36dp"
    android:layout_gravity="center"
    android:layout_marginLeft="50dp"
    android:layout_weight="1"
    android:textColor="#000000"
    android:textSize="24sp" />

<TextView
    android:id="@+id/distanceText"
    android:layout_width="match_parent"
    android:layout_height="36dp"
    android:layout_weight="1"
    android:gravity="center"
    android:textColor="#000000"
    android:textSize="24sp" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:orientation="horizontal">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/textView8"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="CZAS"
            android:textAlignment="center" />

        <TextView
            android:id="@+id/textView7"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="DYSTANS"
            android:textAlignment="center" />
    </LinearLayout>
</LinearLayout>

```

```

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="20dp"
    android:gravity="bottom"
    android:orientation="horizontal">

    <Space
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1" />

    <Button
        android:id="@+id/start_button"
        style="@style/Widget.AppCompat.Button.Colored"
        android:layout_width="80dp"
        android:layout_height="40dp"
        android:layout_weight="1"
        android:background="@color/colorAccent"
        android:padding="5dp"
        android:text="start"
        android:textColor="#ffffff"
        android:textSize="20dp"
        android:textStyle="bold" />

    <Space
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1" />
</LinearLayout>

```

Tytuł zadania. Rysowanie polilinii na mapie.

Wykonawca. Katarzyna Poręba

Realizacja. Utworzenie polilinii przy użyciu metody mapy google *addPolyline*. Ustalenie koloru i szerokości linii. Przekazanie zmiennych zawierających współrzędne geograficzne do funkcji *add* rysującej polilinie.

```

Polyline line = mGoogleMap.addPolyline(new PolylineOptions()
    .add(new LatLng(lStart.getLatitude(), lStart.getLongitude()),
        new LatLng(lEnd.getLatitude(), lEnd.getLongitude()))
    .width(8)
    .color(Color.GREEN));

```

Tytuł zadania. Stworzenie przycisku "Ustawienia" w menu głównym oraz nadanie mu funkcjonalności.

Wykonawca. Jan Michalik

Realizacja. Stworzenie nowej aktywności "Ustawienia" (pustej), do której przechodzimy po wciśnięciu przycisku "Ustawienia" w menu głównym.
Plik SettingsActivity.java

```
package pl.ppm.gitfitscrub;

import android.os.Bundle;
import android.app.Activity;

public class SettingsActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate ( savedInstanceState );
        setContentView ( R.layout.activity_settings );
    }

}
```

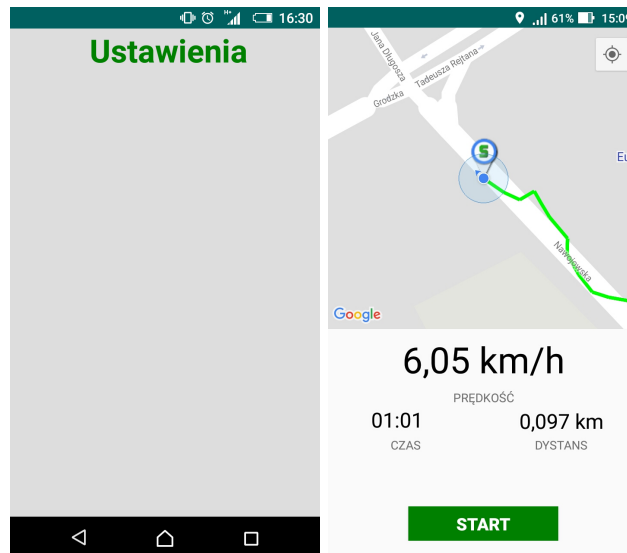
Plik activity_settings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#dddddd"
    tools:context="pl.ppm.gitfitscrub.SettingsActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="368dp"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:text="Ustawienia"
        android:textAlignment="center"
        android:textColor="#008000"
        android:textSize="32sp"
        android:textStyle="bold"
        tools:layout_editor_absoluteX="8dp" />
</android.support.constraint.ConstraintLayout>
```

7.4 Sprint Review/Demo

Założony cel sprintu został w pełni zrealizowany. Estymaty czasochłonności poszczególnych zadań pokryły się z rzeczywistym nakładem pracy.
Demonstracja przyrostu produktu:



Pusta aktywność "Ustawienia" (po lewej) oraz aktywność mapy z narysowaną polilinią i zmodyfikowanym panelem.

8 Sprint 4

8.1 Cel

Utworzenie przełącznika w ustawieniach, odpowiadającego za aktywację trybu nocnego.

8.2 Sprint Planning/Backlog

Tytuł zadania. Dodanie funkcji trybu nocnego do panelu ustawień.

- Estymata: L

8.3 Realizacja

Tytuł zadania. Dodanie funkcji trybu nocnego do panelu ustawień.

Wykonawca. Jan Michalik, Katarzyna Poręba, Michał Ptak.

Realizacja. W aktywności ustawienia został dodany przełącznik umożliwiający zmianę trybu na nocny. Stworzony został skrypt zmieniający wygląd mapy. Zmienione zostały kolory panelu.
Plik activity_settings.xml


```

<Switch
    android:id="@+id/nightmode"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Tryb nocny"
    android:textSize="24sp"
    android:checked="false"
    tools:ignore="MissingConstraints"
    tools:layout_editor_absoluteX="111dp"
    tools:layout_editor_absoluteY="44dp" />

```

Plik SettingsActivity.java

```

import android.widget.Switch;
import android.widget.CompoundButton;
import android.widget.Toast;
import android.widget.CompoundButton.OnCheckedChangeListener;
import android.content.SharedPreferences;

public class SettingsActivity extends Activity {
    int night = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_settings);
        Switch switch1 = findViewById(R.id.nightmode);
        SharedPreferences prefs = getSharedPreferences("database"
, MODE_PRIVATE);
        night = prefs.getInt("night", 1);
        if (night == 1) {
            switch1.setChecked ( false );
        } else {
            switch1.setChecked ( true );
        }
        switch1.setOnCheckedChangeListener(new
OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton
buttonView, boolean isChecked) {
                if(isChecked){
                    Toast.makeText(getApplicationContext(), "
Włączono tryb nocny", Toast.LENGTH_LONG).show();
                    SharedPreferences prefs =
getSharedPreferences("database", Activity.MODE_PRIVATE);
                    SharedPreferences.Editor editor = prefs.edit
();
                    editor.putInt("night", 2);
                    editor.commit();
                }
            }
        });
    }
}

```

```

        } else {
            Toast.makeText(getApplicationContext(), "
Wylaczono tryb nocny", Toast.LENGTH_LONG).show();
            SharedPreferences prefs =
getSharedPreferences("database", Activity.MODE_PRIVATE);
            SharedPreferences.Editor editor = prefs.edit
();
            editor.putInt("night", 1);
            editor.commit();
        }
    }
});
}
}

```

Plik MapsActivity.java

```

public void deepChangeTextColor(ViewGroup parentLayout){
    for (int count=0; count < parentLayout.getChildCount();
count++){
        View view = parentLayout.getChildAt(count);
        if(view instanceof TextView){
            ((TextView)view).setTextColor(Color.parseColor("#
dddddd"));
        } else if(view instanceof ViewGroup){
            deepChangeTextColor((ViewGroup)view);
        }
    }
}

private void setSelectedStyle() {

    MapStyleOptions style;
    if (night == 2) {
        style = MapStyleOptions.loadRawResourceStyle ( this,
R.raw.night_style );
        LinearLayout root = findViewById(R.id.main);
        root.setBackgroundColor(Color.parseColor("#222222"));
        deepChangeTextColor ( root );
    } else {
        style = null;
    }
    mGoogleMap.setMapStyle(style);
}

```

Plik night_style.json

```

[
{
    "elementType": "geometry",
    "stylers": [

```

```

        {
            "color": "#242f3e"
        }
    ],
},
{
    "elementType": "labels.text.fill",
    "stylers": [
        {
            "color": "#2f6031"
        }
    ]
},
{
    "elementType": "labels.text.stroke",
    "stylers": [
        {
            "color": "#242f3e"
        }
    ]
},
{
    "featureType": "administrative.locality",
    "elementType": "labels.text.fill",
    "stylers": [
        {
            "color": "#76c772"
        }
    ]
},
{
    "featureType": "poi",
    "elementType": "labels.text.fill",
    "stylers": [
        {
            "color": "#76c772"
        }
    ]
},
{
    "featureType": "poi.park",
    "elementType": "geometry",
    "stylers": [
        {
            "color": "#263c3f"
        }
    ]
},
{
    "featureType": "poi.park",

```

```

        "elementType": "labels.text.fill",
        "stylers": [
            {
                "color": "#6b9a76"
            }
        ]
    },
    {
        "featureType": "road",
        "elementType": "geometry",
        "stylers": [
            {
                "color": "#38414e"
            }
        ]
    },
    {
        "featureType": "road",
        "elementType": "geometry.stroke",
        "stylers": [
            {
                "color": "#212a37"
            }
        ]
    },
    {
        "featureType": "road",
        "elementType": "labels.text.fill",
        "stylers": [
            {
                "color": "#9ca5b3"
            }
        ]
    },
    {
        "featureType": "road.arterial",
        "elementType": "labels.icon",
        "stylers": [
            {
                "visibility": "off"
            }
        ]
    },
    {
        "featureType": "road.highway",
        "elementType": "geometry",
        "stylers": [
            {
                "color": "#005050"
            }
        ]
    }

```

```

    ]
  },
  {
    "featureType": "road.highway",
    "elementType": "geometry.stroke",
    "stylers": [
      {
        "color": "#1f2835"
      }
    ]
  },
  {
    "featureType": "road.highway",
    "elementType": "labels",
    "stylers": [
      {
        "visibility": "off"
      }
    ]
  },
  {
    "featureType": "road.highway",
    "elementType": "labels.text.fill",
    "stylers": [
      {
        "color": "#f3d19c"
      }
    ]
  },
  {
    "featureType": "transit",
    "stylers": [
      {
        "visibility": "off"
      }
    ]
  },
  {
    "featureType": "transit",
    "elementType": "geometry",
    "stylers": [
      {
        "color": "#2f3948"
      }
    ]
  },
  {
    "featureType": "transit.station",
    "stylers": [
      {

```

```

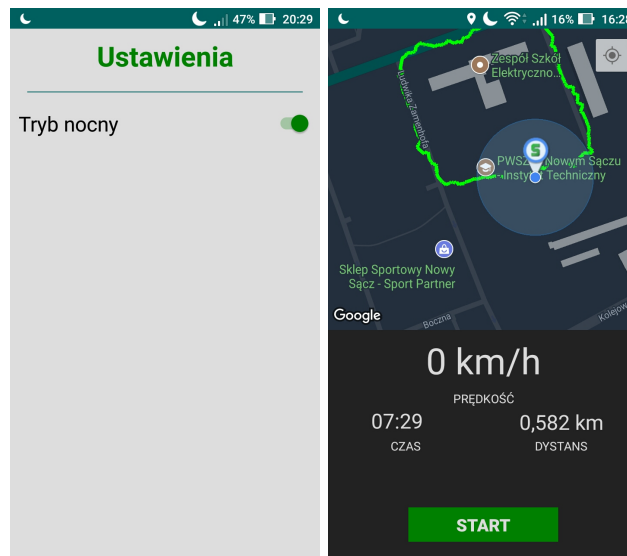
        "visibility": "off"
      }
    ]
  },
  {
    "featureType": "transit.station",
    "elementType": "labels.text.fill",
    "stylers": [
      {
        "color": "#d59563"
      }
    ]
  },
  {
    "featureType": "water",
    "elementType": "geometry",
    "stylers": [
      {
        "color": "#17263c"
      }
    ]
  },
  {
    "featureType": "water",
    "elementType": "labels.text.fill",
    "stylers": [
      {
        "color": "#515c6d"
      }
    ]
  },
  {
    "featureType": "water",
    "elementType": "labels.text.stroke",
    "stylers": [
      {
        "color": "#17263c"
      }
    ]
  }
]

```

8.4 Sprint Review/Demo

Założony cel został osiągnięty, a czas wykonania zadania był zgodny z estymatą. Stan ustawień trybu nocnego jest przechowywany w pamięci telefonu, nawet po wyłączeniu aplikacji. Dodatkowo usprawniona została praca modułu GPS - działa płynniej oraz przy zablokowanym ekranie.

Demonstracja przyrostu produktu:



Aktywność "Ustawienia" (po lewej) z opcją trybu nocnego aktywowanego przełącznikiem oraz aktywność mapy z narysowaną polinią w trybie nocnym.

9 Sprint 5

9.1 Cel

Utworzenie przycisku "Zakończ trening", po wciśnięciu którego statystyki z aktualnego treningu zostaną wyświetlone oraz zapisane w historii.

9.2 Sprint Planning/Backlog

Tytuł zadania. Stworzenie przycisku "Zakończ trening" oraz nadanie mu funkcjonalności.

- Estymata: XXL

Tytuł zadania. Stworzenie przycisku "Historia" w menu głównym oraz nadanie mu funkcjonalności.

- Estymata: S

Tytuł zadania. Prezentowanie danych w historii.

- Estymata: XL

9.3 Realizacja

Tytuł zadania. Stworzenie przycisku "Zakończ trening" oraz nadanie mu funkcjonalności.

Wykonawca. Jan Michalik, Katarzyna Poręba.

Realizacja. Stworzenie przycisku "Stop" kończącego pomiary i rysowanie polilinii oraz wyświetlającego statystyki treningu. Po zamknięciu podsumowania treningu użytkownik przenoszony jest do historii. Plik MapsActivity.java:

```
private Button mStopButton;
    static TextView distanceText, speedText;
    boolean StartB = false;
    int night = 1;
protected void onCreate(Bundle savedInstanceState) {
    mapFrag.getMapAsync(this);

    mStartButton = (Button) findViewById(R.id.start_button);
    mStopButton = (Button) findViewById (R.id.stop_button);
    mChronometer = (Chronometer) findViewById(R.id.
chronometer);

    distanceText = (TextView) findViewById(R.id.distanceText
);
protected void onCreate(Bundle savedInstanceState) {
    @Override
    public void onClick(View view) {
        StartB = true;
        mStartButton.setVisibility(View.GONE);
        mStopButton.setVisibility(View.VISIBLE);
        mStopButton.setBackgroundColor ( Color.parseColor(
"#800000") );
        mChronometer.setBase(SystemClock.elapsedRealtime
());
        mChronometer.start();
    }
}
```

Plik activity_maps.xml:

```
<Button
    android:id="@+id/stop_button"
    style="@style/Widget.AppCompat.Button.Colored"
    android:visibility="gone"
    android:layout_width="80dp"
    android:layout_height="40dp"
    android:layout_weight="1"
    android:background="@color/colorAccent"
    android:padding="5dp"
    android:text="stop"
    android:textColor="#ffffff"
    android:textSize="20dp"
    android:textStyle="bold" />

<Button
    android:id="@+id/start_button"
    style="@style/Widget.AppCompat.Button.Colored"
    android:visibility="visible"
```



```

        android:layout_width="80dp"
        android:layout_height="40dp"
        android:layout_weight="1"

        android:textSize="20dp"
        android:textStyle="bold" />

```

Plik MapsActivity.java:

```

long sec;
int min;
double maxspeed = 0.0;
double averagespeed = 0.0;
int numberspeed = 0;

@Override
protected void onCreate(Bundle savedInstanceState) {
    protected void onCreate(Bundle savedInstanceState) {
        distanceText = (TextView) findViewById(R.id.distanceText
    );
        speedText = (TextView) findViewById(R.id.speedText);

        mStartButton.setOnClickListener(new View.OnClickListener
    () {
        @Override
        public void onClick(View view) {
        public void onClick(View view) {
            mChronometer.start();
        }
        });
        mStopButton.setOnClickListener(new View.OnClickListener()
    {
        @Override
        public void onClick(View view) {
            StartB = false;
            mStopButton.setBackgroundColor ( Color.parseColor
    ("#800000") );
            averagespeed = averagespeed/(double)numberspeed;

            mChronometer.stop();
            showElapsedTime();
            buildAlertPodsumowanie();
        }
    });

        SharedPreferences prefs = getSharedPreferences("database
    ", MODE_PRIVATE);
        night = prefs.getInt("night", 1);
        public void onClick(final DialogInterface dialog,

```

```

@SuppressWarnings("unused") fi
    alert.show();
}

private void showElapsedTime() {
    sec = SystemClock.elapsedRealtime() - mChronometer.
getBase();
    min = 0;
    sec = sec/1000;
    if ( sec > 60 ) {
        min = (int)sec/60;
        sec = sec-min*60;
    }
}

}

private void buildAlertPodsumowanie() {
    final android.app.AlertDialog.Builder builder = new
android.app.AlertDialog.Builder(this);

    builder .setTitle("Podsumowanie treningu")
        .setMessage("dystans: " + new DecimalFormat ( "
###" ).format ( distance ) + " km"
            + "\n" + "czas: " + String.format("%02d",
min) + ":"
            + String.format("%02d", sec) + " min" + "
\n"
            "maksymalna predkosc: " + new
DecimalFormat ( "###" ).format (maxspeed) + " km/h"
            + "\n" + "srednia predkosc: " + new
DecimalFormat ( "###" ).format (averagespeed) + " km/h")
        .setCancelable(false)
        .setPositiveButton("Ok", new DialogInterface.
OnClickListener() {
            public void onClick(final DialogInterface
dialog, @SuppressWarnings("unused") final int id) {
                distance = 0.0;
                maxspeed = 0.0;
                averagespeed = 0.0;
                numberspeed = 0;
                Intent intent = new Intent(
getApplicationContext(), MainMenu.class);
                startActivity(intent);
            }
        });

    final android.app.AlertDialog alert = builder.create();
    alert.show();
}

```

```

@Override
public void onMapReady(GoogleMap googleMap)
{
public void onLocationResult(LocationResult locationResult) {
    .color(Color.GREEN));

        speed = location.getSpeed() * 3.6;
        if(maxspeed < speed) maxspeed = speed;
        averagespeed = averagespeed + speed;
        numberspeed = numberspeed + 1;
    }
}

```

Tytuł zadania. Stworzenie przycisku "Historia" w menu głównym oraz nadanie mu funkcjonalności.

Wykonawca. Jan Michalik, Michał Ptak.

Realizacja. Stworzenie przycisku, który przenosi użytkownika do panelu historia. Plik activity_main_manu.xml:

```

<Button
    android:id="@+id/button3"
    style="@style/Widget.AppCompat.Button.Colored"
    android:layout_width="400px"
    android:layout_height="wrap_content"
    android:layout_below="@+id/button2"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="20dp"
    android:onClick="runHistory"
    android:text="Historia"
    android:textSize="24sp" />

```

Plik MainMenu.java:

```

public void runHistory(View view) {
    Intent intent = new Intent(getApplicationContext(),
        HistoryActivity.class);
    startActivity(intent);
}

```

Tytuł zadania. Prezentowanie danych w historii.

Wykonawca. Michał Ptak

Realizacja. Utworzenie szablonu bloku prezentującego statystyki. Stworzenie widoku RecyclerView oraz CardView i powiązanie ich z adapterem odpowiedzialnym za dostarczenie szablonu do widoku. Wykonanie funkcji pozwalającej na przechowywanie statystyk po zmianie aktywności lub wyłączeniu aplikacji. Zrealizowanie funkcjonalności pozwalającej na usunięcie wybranych pozycji z historii. Plik build.gradle:

```

implementation 'com.android.support:recyclerview-v7:26.1.0'
implementation 'com.android.support:cardview-v7:26.1.0'
implementation 'com.google.code.gson:gson:2.8.2'

maven {
    url "https://maven.google.com"
}

```

Plik HistoryActivity.java:

```

import android.content.SharedPreferences;
import android.os.Bundle;
import android.app.Activity;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import java.lang.reflect.Type;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

public class HistoryActivity extends Activity {
    private ArrayList<ItemTemplate> mExampleList;

    private RecyclerView mRecyclerView;
    private RecyclerView.Adapter mAdapter;
    private RecyclerView.LayoutManager mLayoutManager;
    private static final DateFormat sdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");

    private Button buttonInsert;
    private Button buttonRemove;
    private EditText editTextInsert;
    private EditText editTextRemove;

    Date date = new Date();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_history);

        loadData();
        buildRecyclerView();
    }
}

```

```

        buildHistoryButtons();
    }

    private void saveData() {
        SharedPreferences sharedPreferences =
getSharedPreferences("historyDB", MODE_PRIVATE);
        SharedPreferences.Editor editor = sharedPreferences.edit
        ();
        Gson gson = new Gson();
        String json = gson.toJson(mExampleList);
        editor.putString("history list", json);
        editor.apply();
    }

    private void loadData() {
        SharedPreferences sharedPreferences =
getSharedPreferences("historyDB", MODE_PRIVATE);
        Gson gson = new Gson();
        String json = sharedPreferences.getString("history list",
        null);
        Type type = new TypeToken<ArrayList<ItemTemplate>>() {}.
        getType();
        mExampleList = gson.fromJson(json, type);

        if (mExampleList == null) {
            mExampleList = new ArrayList<>();
        }
    }

    public void insertItem(int position) {
        mExampleList.add(position, new ItemTemplate(R.drawable.
ic_run, sdf.format(date),"stats"));
        mAdapter.notifyItemInserted(position);
        saveData();
    }

    public void removeItem(int position) {
        mExampleList.remove(position);
        mAdapter.notifyItemRemoved(position);
        saveData();
    }

    public void buildRecyclerView() {
mRecyclerView = findViewById(R.id.recyclerView);
        mRecyclerView.setHasFixedSize(true);
        layoutManager = new LinearLayoutManager(this);
        mAdapter = new HistoryAdapter(mExampleList);

mRecyclerView.setLayoutManager(layoutManager);
        mRecyclerView.setAdapter(mAdapter);
    }

```

```

    }

    private void buildHistoryButtons() {
        buttonInsert = findViewById(R.id.button_insert);
        buttonRemove = findViewById(R.id.button_remove);
        editTextInsert = findViewById(R.id.editttext_insert);
        editTextRemove = findViewById(R.id.editttext_remove);

        buttonInsert.setOnClickListener(new View.OnClickListener
        () {
            @Override
            public void onClick(View view) {
                int position = Integer.parseInt(editTextInsert.
                getText().toString());
                insertItem(position);
            }
        });

        buttonRemove.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                int position = Integer.parseInt(editTextRemove.
                getText().toString());
                removeItem(position);
            }
        });
    }
}

```

Plik HistoryAdapter.java

```

package pl.ppm.gitfitscrub;

import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import java.util.ArrayList;

public class HistoryAdapter extends RecyclerView.Adapter<
    HistoryAdapter.HistoryViewHolder> {
    private ArrayList<ItemTemplate> mExampleList;

    public static class HistoryViewHolder extends RecyclerView.
    ViewHolder {
        public ImageView mImageView;
        public TextView mTextView1;
    }
}

```

```

public TextView mTextView2;

    public HistoryViewHolder(View itemView) {
        super(itemView);
        mImageView = itemView.findViewById(R.id.imageView);
        mTextView1 = itemView.findViewById(R.id.textView1);
        mTextView2 = itemView.findViewById(R.id.textView2);
    }
}

public HistoryAdapter(ArrayList<ItemTemplate> exampleList) {
    mExampleList = exampleList;
}

@Override
public HistoryViewHolder onCreateViewHolder(ViewGroup parent,
    int viewType) {
    View v = LayoutInflater.from(parent.getContext()).inflate
(R.layout.example_item, parent, false);
    HistoryViewHolder evh = new HistoryViewHolder(v);
    return evh;
}

@Override
public void onBindViewHolder(HistoryViewHolder holder, int
position) {
    ItemTemplate currentItem = mExampleList.get(position);

    holder.mImageView.setImageResource(currentItem.
getImageResource());
    holder.mTextView1.setText(currentItem.getText1());
    holder.mTextView2.setText(currentItem.getText2());
}

@Override
public int getItemCount() {
    return mExampleList.size();
}
}

```

Plik ItemTemplate.java:

```

package pl.ppm.gitfitscrub;

public class ItemTemplate {
    private int mImageResource;
    private String mText1;
    private String mText2;
    public ItemTemplate(int imageResource, String text1, String
text2) {
        mImageResource = imageResource;
        mText1 = text1;
    }
}

```

```

        mText2 = text2;
    }

    public int getImageResource() {
        return mImageResource;
    }

    public String getText1() {
        return mText1;
    }

    public String getText2() {
        return mText2;
    }
}

```

Plik ic_run:

```

<vector xmlns:android="http://schemas.android.com/apk/res/android"
    "
        android:width="24dp"
        android:height="24dp"
        android:viewportWidth="24.0"
        android:viewportHeight="24.0">
    <path
        android:fillColor="#FF000000"
        android:pathData="M13.49,5.48c1.1,0 2,-0.9 2,-2s-0.9,-2
-2,-2 -2,0.9 -2,2 2,2zM9.89,19.38l1,-4.4 2.1,2v6h2v-7.5
1-2.1,-2 0.6,-3c1.3,1.5 3.3,2.5 5.5,2.5v-2c-1.9,0 -3.5,-1
-4.3,-2.4l-1,-1.6c-0.4,-0.6 -1,-1 -1.7,-1 -0.3,0 -0.5,0.1
-0.8,0.1l-5.2,2.2v4.7h2v-3.4l1.8,-0.7 -1.6,8.1 -4.9,-1 -0.4,2
7,1.4z"/>
</vector>

```

Plik activity_history.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#dddddd"
    android:orientation="vertical">

    <TextView
        android:id="@+id/textViewHistory"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:text="@string/historia"
        android:textAlignment="center"

```



```

        android:textColor="#008000"
        android:textSize="32sp"
        android:textStyle="bold"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="8dp" />

<View
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:layout_margin="20dp"
    android:background="@color/colorPrimaryDark" />

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="50dp"
        android:padding="4dp"
        android:scrollbars="vertical" />

    <EditText
        android:id="@+id/edittext_insert"
        android:layout_width="40dp"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentStart="true"
        android:layout_marginStart="15dp"

    <Button
        android:id="@+id/button_insert"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_toEndOf="@+id/edittext_insert"
        android:text="dodaj" />

    <EditText
        android:id="@+id/edittext_remove"
        android:layout_width="40dp"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_toStartOf="@+id/button_remove"
        android:inputType="number"/>

    <Button
        android:id="@+id/button_remove"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="15dp"
        android:text="usun" />

</RelativeLayout>

</LinearLayout>

Plik example_item.xml:

<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardCornerRadius="4dp"
    android:orientation="vertical"
    android:id="@+id/cv"
    android:layout_marginBottom="4dp"
    android:layout_marginHorizontal="2dp">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="4dp">

        <ImageView
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:padding="2dp"
            android:id="@+id/imageView" />

        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:layout_toEndOf="@+id/imageView"
            android:text="Line 1"
            android:textColor="@android:color/black"
            android:textSize="20sp"
            android:textStyle="bold"/>

        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

```

```

        android:layout_below="@+id/textView1"
        android:layout_toEndOf="@+id/imageView"
        android:text="Line 2"
        android:textSize="15sp"
        android:layout_marginStart="8dp"/>

```

```

</RelativeLayout>

```

```

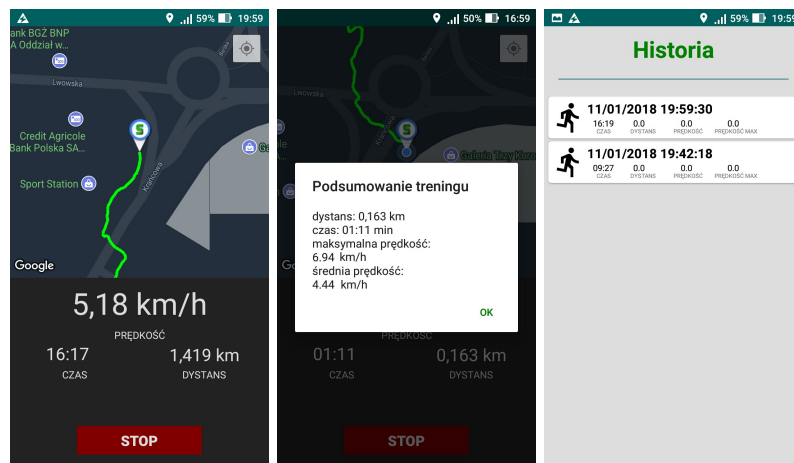
</android.support.v7.widget.CardView>

```

9.4 Sprint Review/Demo

Założony cel przyrostu został osiągnięty z wyjątkiem przekazania wartości statystyk (dystans, prędkość, prędkość maksymalna) do obiektu prezentującego dane w historii. Estymaty zostały poprawnie oszacowane, oprócz zadania prezentowania danych w historii. Zadanie zostanie przeniesione do kolejnego sprintu.

Demonstracja przyrostu produktu:



Od lewej: przycisk stop, podsumowanie treningu, aktywność "Historia"

10 Sprint 6

10.1 Cel

Umożliwienie przeglądania statystyk z wcześniejszych treningów.

10.2 Sprint Planning/Backlog

Tytuł zadania. Prezentowanie danych w historii.

- Estymata: XL

10.3 Realizacja

Tytuł zadania. Prezentowanie danych w historii.

Wykonawca. Jan Michalik, Katarzyna Poręba, Michał Ptak.

Realizacja. «Sprawozdanie z realizacji zadania (w tym ocena zgodności z estymatą). Kod programu (środowisko `verbatim`):

```
for (i=1; i<10; i++)  
...  
».
```

10.4 Sprint Review/Demo

«Sprawozdanie z przeglądu Sprint'u – czy założony cel (przyrost) został osiągnięty oraz czy wszystkie zaplanowane Backlog Item'y zostały zrealizowane? Demonstracja przyrostu produktu».

Literatura

- [1] S. R. Covey, *7 nawyków skutecznego działania*, Rebis, Poznań, 2007.
- [2] Tobias Oetiker i wsp., Nie za krótkie wprowadzenie do systemu L^AT_EX 2_ε, <ftp://ftp.gust.org.pl/TeX/info/lshort/polish/lshort2e.pdf>
- [3] K. Schwaber, J. Sutherland, *Scrum Guide*, <http://www.scrumguides.org/>, 2016.
- [4] <https://agilepainrelief.com/notesfromatooluser/tag/scrum-by-example>
- [5] https://www.tutorialspoint.com/scrum/scrum_user_stories.htm
- [6] <https://developer.android.com/index.html>
- [7] <https://docs.oracle.com/javase/8/docs/api/>
- [8] <https://stackoverflow.com/questions/tagged/android>
- [9] Dawn Griffiths, David Griffiths, *Android. Programowanie aplikacji. Rusz głową!*, Helion, 2016.