

# AKADEMIA NAUK STOSOWANYCH W NOWYM SĄCZU

Wydział Nauk Inżynierskich  
Katedra Informatyki

## DOKUMENTACJA PROJEKTOWA PROGRAMOWANIE URZĄDZEŃ MOBILNYCH

### **Aplikacja Survivalowa**

Autor:  
Bogusław Gruca  
Karol Kowalik

Prowadzący:  
mgr inż. Dawid Kotlarski

Nowy Sącz 2022

## **Spis treści**

<b>1. Ogólne określenie wymagań</b>	<b>3</b>
<b>2. Określenie wymagań szczegółowych</b>	<b>4</b>
<b>3. Projektowanie</b>	<b>6</b>
<b>4. Implementacja</b>	<b>9</b>
<b>5. Testowanie</b>	<b>11</b>
<b>6. Podręcznik użytkownika</b>	<b>12</b>
<b>Literatura</b>	<b>13</b>
<b>Spis rysunków</b>	<b>13</b>
<b>Spis tabel</b>	<b>14</b>
<b>Spis listingów</b>	<b>15</b>

## 1. Ogólne określenie wymagań

Celem programu jest utworzenie prostej w obsłudze oraz uniwersalnej w każdej sytuacji aplikacji, dzięki której będzie możliwe bezpieczne poruszanie się w terenie. Aplikacja powinna zawierać wszystkie najważniejsze narzędzia do posługiwania się poza domem. Odbiorcą ma być każdy, niezależnie od wieku liczy się prostota w obsłudze, aby nikt nie miał problemu z użytkowaniem. Aplikacja ma być obsługiwana przez system Android. Program musi posiadać dostęp do mapy, która będzie umożliwiała orientowanie się w terenie górzystym, zabudowanym, leśnym.

## 2. Określenie wymagań szczegółowych

Naszym celem w budowaniu aplikacji jest zaprogramowanie tak, aby posiadała następujące funkcje:

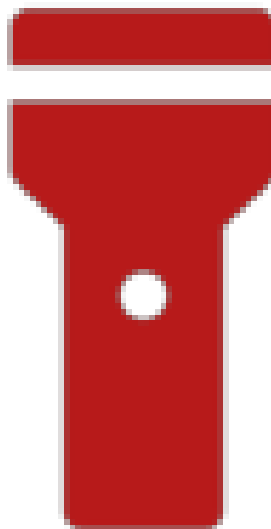
- mape
- latarkę
- kompas
- stoper który będzie odliczał czas do przerwy
- lokalizacja

Aplikacja będzie miała prosty interfejs. Brak internetu będzie skutkował utrudnieniami w korzystaniu z aplikacji. Zaczniemy budowę programu od zaprojektowania graficznego aplikacji i ułożenia danych narzędzi, przycisków. Po ustaleniu grafiki i przycisków funkcyjnych, przejdziemy do rozpoczęcia kodowania aplikacji i podłączać mapę z lokalizacją. Po zaprogramowaniu aplikacji będziemy sprawdzali jej prawidłowe działanie i poprawiali błędy. Program będziemy rozwijali w przyszłości o kolejne funkcje.

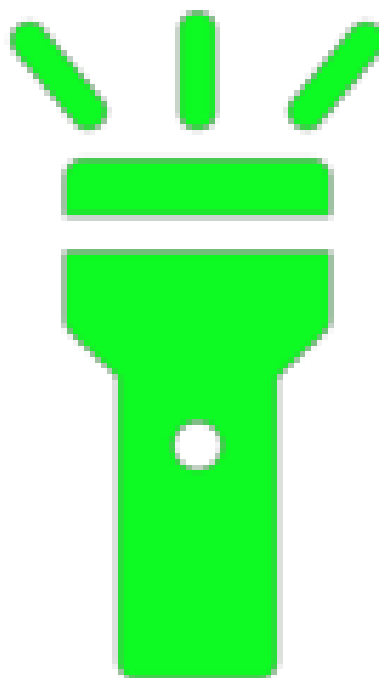
Przycisk latarki będącej w stanie off. (Rys.2.1)

Przycisk latarki będącej w stanie on. (Rys.2.2)

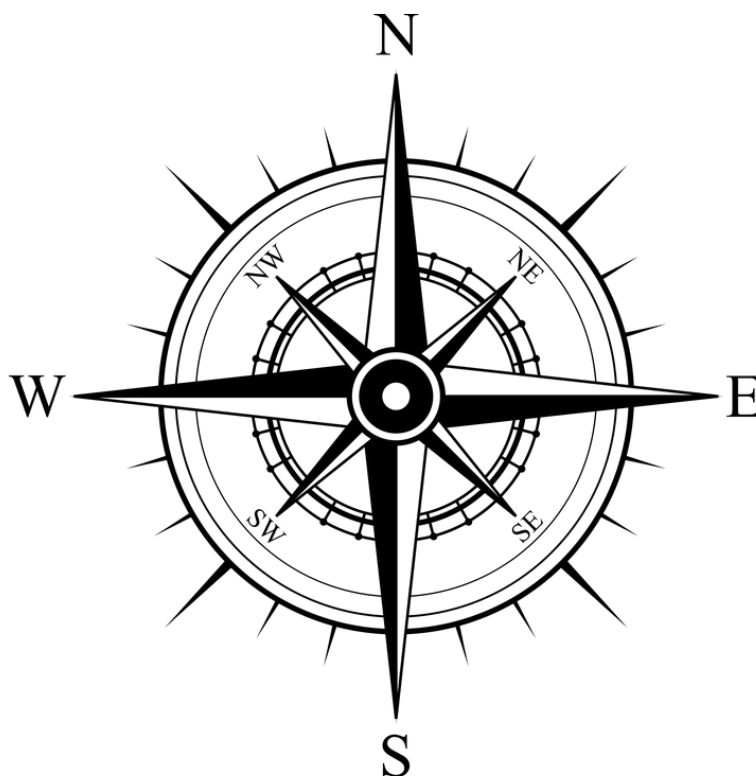
Wygląd kompasu. (Rys.2.3)



**Rys. 2.1.** Przycisk latarki będącej w stanie off.



**Rys. 2.2.** Przycisk latarki będącej w stanie on.



Rys. 2.3. Wygląd kompasu.

### 3. Projektowanie

Do projektu wykorzystaliśmy narzędzia takie jak GIT czyli rozproszony system kontroli wersji, oprogramowanie służące do śledzenia zmian głównie w kodzie źródłowym oraz pomocy programistom w łączeniu zmian dokonanych w plikach przez wiele osób w różnym czasie.

GitHub– hostingowy serwis internetowy przeznaczony do projektów programistycznych wykorzystujących system kontroli wersji Git. Stworzony został przy wykorzystaniu frameworka Ruby on Rails i języka Erlang. Serwis działa od kwietnia 2008 roku[1]. GitHub udostępnia darmowy hosting programów open source i prywatnych repozytoriów (część funkcji w ramach prywatnych repozytoriów jest płatna)(Rys.3.1).

Aby efektywnie pracować w grupie, utworzyliśmy repozytorium Aplikacja-Survivalowa, gdzie umieściliśmy kod do naszej aplikacji wraz z dokumentacją, dzięki temu możemy w prosty sposób wprowadzać zmiany oraz ulepszać program. Do zapisywania oraz wysyłania zmian używamy programu GitHub Desktop (Rys.3.2).

Do wykonania dokumentacji oraz raportów wykorzystaliśmy TeXstudio to wieloplatformowy edytor LaTeX typu open source. Jego funkcje obejmują interaktywne sprawdzanie pisowni, składanie kodu i podświetlanie składni. Nie zapewnia samego

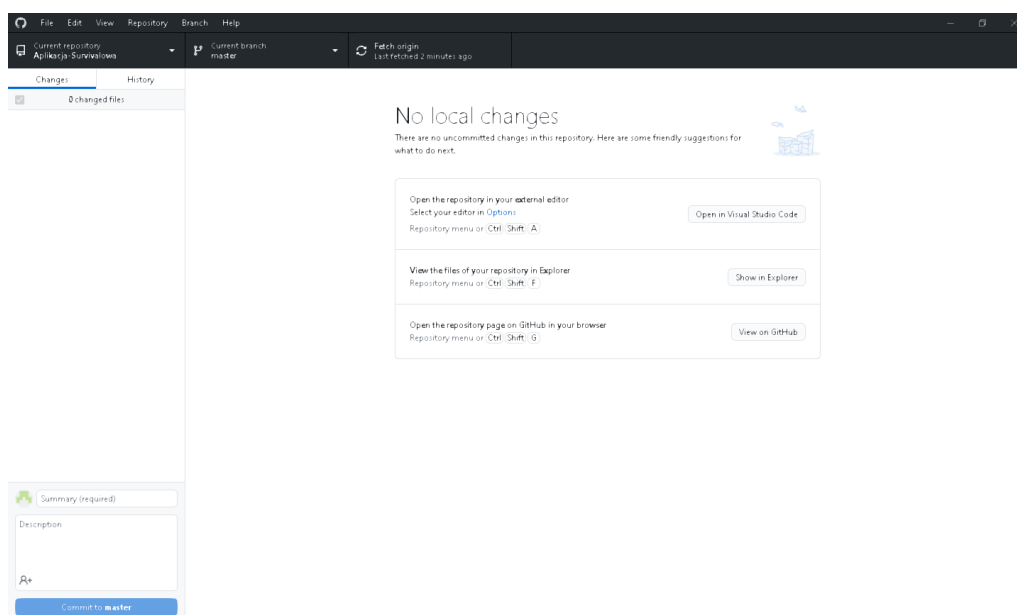
LaTeX-a – użytkownik musi najpierw wybrać dystrybucję TeX-a i ją zainstalować.

Pierwotnie nazywany TexMakerX, TeXstudio został uruchomiony jako rozwinięcie Texmakera, który próbował rozszerzyć go o dodatkowe funkcje, zachowując jednocześnie jego wygląd i styl

Środowiskiem programistycznym gdzie piszemy naszą aplikację jest Android Studio (Rys.3.3) czyli środowisko programistyczne (IDE) stworzone przez Google na bazie IntelliJ, które kierowane jest do developerów aplikacji na Androida. Pozwala ono wygodnie projektować, tworzyć i debugować własne programy na najpopularniejszą obecnie platformę systemową dla urządzeń mobilnych.

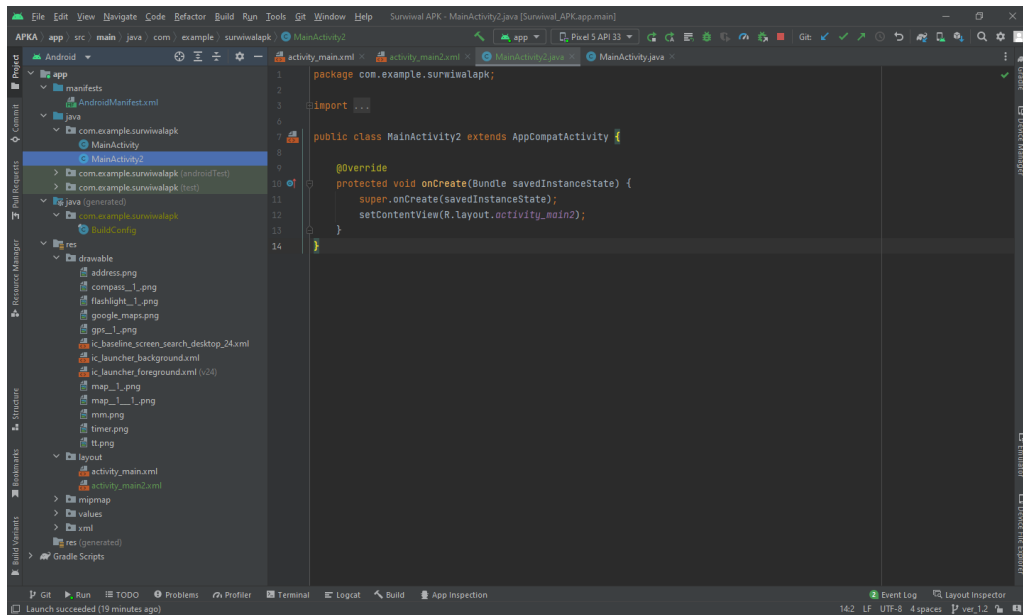


**Rys. 3.1.** GitHub



**Rys. 3.2.** GitHub Setup





Rys. 3.3. Android Studio

## 4. Implementacja

Na listingu nr. 1 przedstawiony jest kod utworzenia przycisku latarki oraz pozycjonowanie na ekranie. Przykładowo w linii 7 kodu `marginBottom` - oznacza zachować odległość od dołu widoku

```

1 <ImageButton
2     android:id="@+id/imageButton14"
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     android:layout_marginStart="203dp"
6     android:layout_marginEnd="16dp"
7     android:layout_marginBottom="260dp"
8     android:backgroundTint="#00FFFFFF"
9     app:layout_constraintBottom_toBottomOf="parent"
10    app:layout_constraintEnd_toEndOf="parent"
11    app:layout_constraintStart_toStartOf="@+id/imageButton12"
12    app:srcCompat="@drawable/flashlight__1_" />
13

```

Listing 1. Przykładowy kod Przycisku latarki

Na listingu nr. 2 przedstawiony jest kod przycisku wyłączonej latarki.

```

1 <ImageButton
2     android:id="@+id/imageButton"
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     android:layout_marginBottom="24dp"

```

```

6  android:backgroundTint="#00FFFFFF"
7  app:layout_constraintBottom_toTopOf="@+id/imageButton19"
8  app:layout_constraintEnd_toEndOf="parent"
9  app:layout_constraintHorizontal_bias="0.428"
10 app:layout_constraintStart_toEndOf="@+id/imageButton12"
11 app:srcCompat="@drawable/off" />

```

**Listing 2.** Przykładowy kod przycisku wyłączonej latarki

Na listingu nr. 3 przedstawiony jest kod definiowania zmiennych.

```

1 private ImageButton toggleButton;
2 boolean hasCameraFlash = false;
3 boolean flashOn = false;

```

**Listing 3.** Przykładowy kod definiowania zmiennych

Na listingu nr. 4 przedstawiony jest kod pozwalający na wyłączenie latarki.(kod poniżej)

```

1  public void onClick(View view) {
2  if (hasCameraFlash){
3      if (flashOn){
4          flashOn = false;
5          toggleButton.setImageResource(R.drawable.off);
6          try {
7              flashLightOff();
8          } catch (CameraAccessException e) {
9              e.printStackTrace();
10         }
11     }
12
13
14     private void flashLightOn() throws CameraAccessException {
15         CameraManager cameraManager = (CameraManager)
16         getSystemService(Context.CAMERA_SERVICE);
17         assert cameraManager != null;
18         String cameraId = cameraManager.getCameraIdList()[0];
19         cameraManager.setTorchMode(cameraId, true);
20         Toast.makeText(MainActivity.this, "FlashLight is ON",
21         Toast.LENGTH_SHORT).show();
22     }

```

**Listing 4.** Przykładowy kod ukazujący mechanizm wyłączenia latarki.

```

1

```

**Listing 5.** Przykładowy kod definiowania zmiennych

## 5. Testowanie

## 6. Podręcznik użytkownika

## Spis rysunków

2.1. Przycisk latarki będącej w stanie off. . . . .	5
2.2. Przycisk latarki będącej w stanie on. . . . .	5
2.3. Wygląd kompasu. . . . .	6
3.1. . . . .	8
3.2. GitHub Setup . . . . .	8
3.3. Android Studio . . . . .	9

## **Spis tabel**

## Spis listingów

1.	Przykładowy kod Przycisku latarki . . . . .	9
2.	Przykładowy kod przycisku wyłączonej latarki . . . . .	9
3.	Przykładowy kod definiowania zmiennych . . . . .	10
4.	Przykładowy kod ukazujący mechanizm wyłączenia latarki. . . . .	10
5.	Przykładowy kod definiowania zmiennych . . . . .	10