

AKADEMIA NAUK STOSOWANYCH
W NOWYM SĄCZU

Wydział Nauk Inżynierijnych
Katedra Informatyki

DOKUMENTACJA PROJEKTOWA
PROGRAMOWANIE URZĄDZEŃ MOBILNYCH

Ttest

Autor:
Paulina Hudzik
Magdalena Krzyszowska

Prowadzący:
mgr inż. Dawid Kotlarski

Nowy Sącz 2022

Spis treści

1. Ogólne określenie wymagań	4
2. Określenie wymagań szczegółowych	6
3. Projektowanie	9
3.1. Środowisko Android Studio oraz język programowania Java.	9
3.2. Środowisko programisty/pisanie dokumentów - Latex.	10
3.3. Czym jest Git oraz do czego służy?	11
4. Implementacja	12
5. Testowanie	21
5.1. Testowanie latarki	21
5.2. Testowanie trybu ciemnego	22
5.3. Testowanie czujnika zbliżeniowego	23
5.4. Testowanie lokalizacji	24
5.5. Testowanie wifi	25
5.6. Testowanie dźwięku	26
5.7. Testowanie mikrofonu	27
5.8. Testowanie aparatu	28
5.9. Testowanie menu	29
5.10. Testowanie podsumowania	30
6. Podręcznik użytkownika	31
6.1. Wyszukiwanie i uruchomienie aplikacji Ttest	31
6.2. Test latarki	33
6.3. Test trybu ciemnego	35
6.4. Test czujnika zbliżeniowego	37
6.5. Test GPS	39
6.6. Test Wifi	41
6.7. Test dźwięku	45
6.8. Test mikrofonu	47
6.9. Test aparatu	51

6.10. Podsumowanie wszystkich testów	55
Literatura	58
Spis rysunków	60
Spis tabel	61
Spis listingów	62

1. Ogólne określenie wymagań

Nowoczesne smartfony są naszpikowane różnego rodzaju czujnikami. To one sprawiają, że codzienne używanie telefonów komórkowych jest łatwe i przyjemne. Dzięki nim urządzenia znajdują naszą lokalizację, w którym kierunku się poruszamy oraz sposób w jaki je trzymamy. Wygaszanie ekranu, gdy zbliżymy słuchawkę do policzka, automatyczna zmiana orientacji wyświetlanego obrazu, kiedy obrócimy telefon czy w końcu sterowanie zaawansowanymi grami - to wszystko nie byłoby możliwe bez czujników. Ogólnym zamysłem naszego projektu jest stworzenie aplikacji która ma służyć do testowania czujników w telefonach, co ułatwiłoby i bardzo przyśpieszyło pracę w naszym serwisie komórkowym.

Zależy nam na możliwości sprawdzenia czujników takich jak:

- latarka,
- czujnik zbliżeniowy,
- gps,
- wifi,
- dźwięk,
- mikrofon,
- aparat.



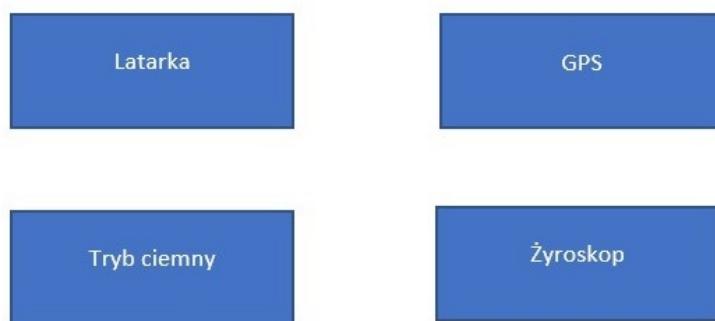
Rys. 1.1. Logo firmy TTest

Nasza firma powstała w 2011 roku; przez ponad 10 lat wcześniej pracujemy oraz staramy się zadbać nie tylko o naszych klientów lecz i o naszych pracowników, stąd też pojawił się pomysł na wdrożenie nowej aplikacji, która pozwoliłaby na jeszcze większe usprawnienie pracy w naszym zespole. Co roku przybywa nam tysiące nowych klientów którzy liczą na fachową pomoc z naszej strony, z związku z tym zatrudniamy coraz więcej nowych pracowników i to właśnie do nich w dużej mierze miałaby trafić nowa aplikacja testująca.

Nasza firma posiadała w przeszłości już jedną aplikację, niestety przez upływ czasu stała się przestarzała i nie spełnia naszych obecnych wymagań, np. brak testu lokalizacji, który obecnie jest jednym z podstawowych czujników wykorzystywanym w wielu aplikacjach. Co więcej wygląd aplikacji jest przestarzały na obecne standardy, test trzeba było wybrać z rozwijanej listy, która stała się niepraktyczna oraz mocno spowalniała pracę, dlatego zależy nam aby layout nowej aplikacji był zachowany w intuicyjny i minimalistyczny sposób, przez co nawet osoby które dopiero odbywają szkolenia w naszej firmie mogły wspomóc pracowników i same przeprowadzać testy poszczególnych czujników. Również zależy nam aby w aplikacji można było w jakiś sposób podsumować przeprowadzone testy. Kolorem przewodnim naszej firmy jest niebieski, dlatego zależy nam na tym aby akcenty właśnie w tym kolorze pojawiły się w aplikacji; mogą to być na przykład przyciski czy chociażby kolor tła aplikacji. Na rysunku 1.1 znajduję się logo naszej firmy, zależy nam aby pojawiło się jako ikonka aplikacji oraz w samej aplikacji.

Myślimy o prostej i przejrzystej budowie menu, składającej się z dużych kafelków które będą odnośnikiem do poszczególnych testów. Zobrazowanie naszego pomysłu znajduje się na rysunku 1.2.

Jeśli chodzi o przyciski to zależało nam na tym aby pojawiły się ikonki przedstawiające daną funkcję, na przykład zamiast samego napisu "Latarka" pojawił się przycisk z wizerunkiem przedstawiającym latarkę, co sprawi to że aplikacja będzie wyglądać przejrzystie i w szybki sposób będzie można wzrokowo znaleźć interesujący nas test.



Rys. 1.2. Przykładowy layout

2. Określenie wymagań szczegółowych

Firma serwisowa Ttest pragnie aby stworzyć aplikację mobilną do testowania czujników, która byłaby zachowana w sposób minimalistyczny i intuicyjny. Przejrzysta i prosta budowa menu, miałaby na celu usprawnić szukanie odpowiednich komponentów. Firma posiada również logo, które powinno zawierać się w projekcie, jak i niebieskie akcenty (np. przyciski) który powinny towarzyszyć aplikacji ponieważ jest to kolor kojarzony z firmą.

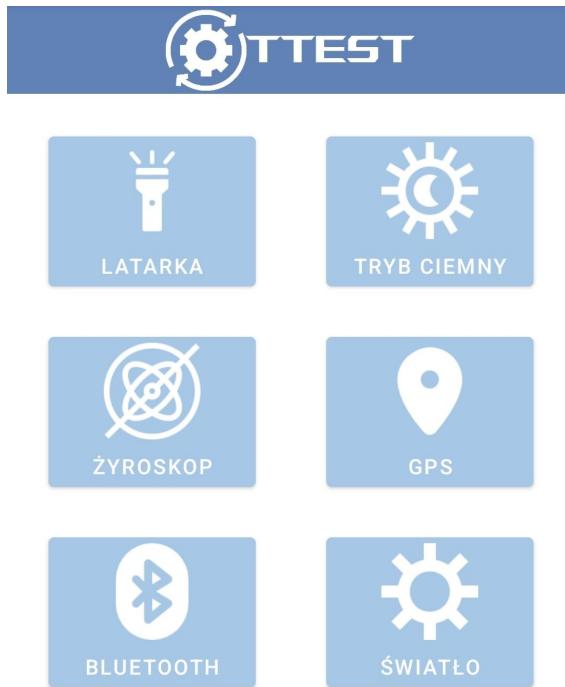
Jednym z kluczowych zadań jest stworzenie aplikacji w sposób jak najbardziej przejrzysty, do tego celu użyjemy ikonek, które w prosty i szybki sposób nakierują nas na odpowiedni komponent. Przykładem może być latarka; chcąc przetestować jej działanie wystarczy wybrać ikonkę która będzie przedstawiała latarkę. Po jej naciśnięciu zostaniemy przekierowani do strony z testem. Oprócz testowania czujników, zapewniamy moduł odpowiedzialny za podsumowanie wszystkich przeprowadzonych testów. Zapewniamy również zawarcie logotypu oraz zachowania layoutu aplikacji w niebieskich kolorach, sprawimy by layout w całości był jak najbardziej intuicyjny i zachowany w minimalistyczny sposób (brak jaskrawych kolorów, duże przejrzyste przyciski, okienka pop-up z wiadomościami o przebiegu testu). Postaramy się aby komponenty testujące takie jak: latarka, test dźwięku, tryb nocny i tym podobne, będą poprawnie spełniać swoje zadanie. Aplikacja zostanie napisana w programie Android Studio, natomiast język którym będziemy się posługiwać to Java.

Na rysunku 2.1 znajduje się logo, które będzie użyte jako ikonka aplikacji. Została ona stworzona przez zespół naszych grafików, wraz z zakończeniem prac nad aplikacją, firma Ttest dostanie pełne prawa do posługiwania się tym logotypem.



Rys. 2.1. Ikona aplikacji

Poglądowy zarys wyglądu menu znajduje się na rysunku 2.2. Jest to jedynie projekt layoutu, dlatego nie wszystkie z ukazanych na nim testów będą znajdować się w finalnej aplikacji.



Rys. 2.2. Wygląd menu głównego

Tak jak już wspomnieliśmy powyżej postaramy się by każdy z czujników działał w należyty sposób; to znaczy by latarka poprzez kliknięcie przycisku włączała się oraz wyłączała. Test dźwięku pozwoli nam poprzez kliknięcie przycisku usłyszeć melodię wydobywającą się z głośników. Tryb nocny, który jest bardzo przydatną funkcją w telefonie szczególnie wieczorami gdy korzystamy z urządzeń mobilnych, zmieni kolorystykę z jasnej na ciemną. Przez tą funkcję jaka jest tryb ciemny nasze oczy nie są narażane na tak mocne światło, co sprawia, że czujemy większy komfort użytkowania telefonów komórkowych. Kolejny czujnik jakim będzie GPS pozwoli nam szybko określić położenie w którym się znajdujemy. Test wifi umożliwi sprawdzenie połączenia z siecią, natomiast test mikrofonu sprawdzi się w sytuacji gdy na przykład pojawi się problem podczas rozmów telefonicznych. Test aparatu pozwoli wykonać zdjęcie oraz przetestować jakość wykonanego zdjęcia.

Do podsumowania wyników będzie prowadził nas jednej z przycisków znajdujących się w menu głównym. Podglądowy wygląd strony z wynikami znajduje się na rysunku 2.3. W miejscu "Marka i model telefonu", powinny wyświetlać się informacje o telefonie na którym obecnie są przeprowadzane testy. Do podsumowania testów zostaną użyte przyciski typu radio, które pozwalają na wybór tylko jednego przycisku z grupy: Tak lub Nie. Kiedy zostanie naciśnięty przycisk "PODSUMUJ", poniżej zostanie wyświetlona lista podsumowująca wszystkie przeprowadzone testy.

Marka i model telefonu

Podsumowanie testów

Czy test przebiegł pomyślnie?

Latarka: Tak Nie

Tryb Ciemny: Tak Nie

Zbliżeniowy: Tak Nie

Gps: Tak Nie

Wifi: Tak Nie

Dźwięk: Tak Nie

Mikrofon: Tak Nie

Aparat: Tak Nie

PODSUMUJ

Rys. 2.3. Podsumowanie testów

3. Projektowanie

3.1. Środowisko Android Studio oraz język programowania Java.

Android Studio to 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. Oprogramowanie oferuje podobne możliwości co środowisko Eclipse z zainstalowaną wtyczką ADT, jednak jest ono znacznie prostsze i bardziej intuicyjne w szczególności dla początkujących programistów. Przejęło ono wszystkie najlepsze rozwiązania znane użytkownikom IntelliJ, oferując narzędzie zoptymalizowane w dużym stopniu do wygodnej pracy z kodem źródłowym. Decydując się na korzystanie z Android Studio użytkownik otrzymuje środowisko programistyczne z przejrzystym i konfigurowalnym interfejsem graficznym, nie wspominając o funkcji kolorowania składni czy mechanizmie zakładek, pozwalającym na pracę z wieloma plikami jednocześnie.

Java¹ to wysokopoziomowy język programowania najczęściej wykorzystywany do tworzenia backendu aplikacji internetowych. Język ten jest łatwo przenośny, dzięki interpretowaniu przez wieloplatformową maszynę wirtualną Java Virtual Machine. Można stwierdzić, że Java jest językiem preferowanym przez korporacje i duże firmy. W Javie napisano m.in. takie aplikacje jak Gmail, OpenOffice czy Minecraft, ale także LinkedIn, Netflix czy Amazon.

¹Język programowania Java - więcej informacji [1].

3.2. Środowisko programisty/pisanie dokumentów - Latex.

Latex² służy do wytwarzania przejrzyście wyglądających dokumentów tekstowych takich jak książki, artykuły, czy nawet prezentacje. Docelowym formatem jest wydruk, czy też pliki w różnych formatach, takich jak PDF, Postscript, czy też HTML. Szczególnie wygodne jest tworzenie dokumentów technicznych, matematycznych, ale z powodzeniem może też być stosowany do pisania dokumentacji programów albo zbioru opowiadań.

Latex, podobnie jak języki programowania, ma swój własny język, w którym pisze się treść dokumentu oraz posiada narzędzia (można by powiedzieć "kompiatory"), które przetwarzają pliki źródłowe i generują pliki docelowe. W językach programowania zazwyczaj jedną z istotnych rzeczy jest zbiór bibliotek z gotowymi implementacjami różnych typowych czynności. Również w Latexu jest dużo gotowych pakietów pozwalających w szybki sposób tworzyć najróżniejsze elementy oraz rodzaje dokumentów.

Filozofia Latexa jest taka, aby skupiać się na tym co merytorycznie ma zawierać dany dokument, a jak najmniej poświęcać uwagi na to, jak ma to wyglądać. Innymi słowy wprowadzamy tylko strukturę i zawartość dokumentu, a latex za nas robi resztę roboty, aby wyjściowy dokument wyglądał jak należy. Oczywiście mamy dużą możliwość ingerencji w wygląd, ale zazwyczaj jest to tylko dobieranie jakiegoś szablonu lub potrzeba uzyskania niestandardowego efektu. Jest to zupełnie inna filozofia, niż w wielu innych edytorach tekstowych, szczególnie w różnych aplikacjach biurowych, gdzie prawie na każdym kroku musimy od decydować, jaki ma być wygląd, wielkość liter, czcionka, odstępy, sposób wyświetlania tytułów itp.

Podstawą możliwości cieszenia się twórczością w Latexu jest posiadanie wszystkich narzędzi, pakietów, czcionek, itp. Gotowe zbiory są dostępne w różnych dystrybucjach. Oprócz tych narzędzi, poczynając od użytkownicy mogą skorzystać z gotowych środowisk do obrabiania dokumentów Latexu.

Podstawową dystrybucją jest TeX Live. Jest ona dostępna pod wiele różnych platform. Jest łatwą w instalacji kompletną paczką narzędzi, programów, czcionek.

²Latex - więcej informacji [2].

3.3. Czym jest Git oraz do czego służy?

Co to jest Git³ i dlaczego cieszy się tak dużą popularnością? Ten system kontroli wersji znacznie usprawnia, a jednocześnie zabezpiecza codzienną pracę przy kodzie. Dzięki swojej prostocie i elastyczności może być wykorzystywany zarówno przy drobnych, jak i ogromnych projektach. Dlatego też jest używany przez programistów oraz grafików na całym świecie. Odpowiadając w skrócie na pytanie, co to jest Git, należy powiedzieć, że to system kontroli wersji. Służy on więc do zarządzania historią kodu źródłowego. Jego funkcjonalność ma kilka podłoży. Między innymi sprawdza się tak dobrze, ponieważ:

- pozwala na jednoczesną pracę na tym samym kodzie przez kilka osób,
- umożliwia transferowanie i łączenie zmian z różnych branchy w jednym projekcie,
- pozwala na pracę offline we własnym repozytorium,
- jest szybki i wydajny.

Cechy te sprawiły, że Git szybko został doceniony w całej branży. Przechowywanie wersji, a także możliwość rozgałęziania kodu to niewątpliwie jego ogromne zalety.

GitHub z kolei to firma, która hostuje repozytoria Git i dostarcza oprogramowanie do korzystania z niego. Jednym z przykładów jest tytułowy GitHub Desktop na systemy Windows 10 i 11. GitHub jest obecnie najpopularniejszym hostem projektów open source pod względem zarówno liczby projektów, jak i użytkowników. Choć GitHub koncentruje się głównie na kodzie źródłowym, to inne projekty coraz częściej wykorzystują systemy kontrolowania wersji do zarządzania przepływem pracy związanym z publikowaniem czasopism, artykułów, podręczników itp.

³Git - więcej informacji [3].

4. Implementacja

Listing 1 (s. 12) przedstawia implementacje jednego z przycisków którego zadaniem jest otworzenie nowej strony oznaczonej w 5 linijce kodu po kliknięciu.

```
1 button_1 = (Button) findViewById(R.id.button_1);
2 button_1.setOnClickListener(new View.OnClickListener() {
3     @Override
4     public void onClick(View view) {
5         Intent intent = new Intent(MainActivity.this, latarka.class);
6         startActivity(intent);
7     }
8 }) ;
```

Listing 1. Menu - Działanie Przycisków

Listing 2 (s. 12) przedstawia włączenie latarki. Jeśli telefon ma latarkę i mamy do niej dostęp, latarka zostanie uruchomiona i wyświetli się nam komunikat o włączeniu. Jeżeli nie, to dostaniemy odpowiednie powiadomienie o niepowodzeniu operacji. W podobny sposób wygląda metoda z wyłączeniem latarki. W 6 linijce trzeba zmienić wartość na false oraz w 7 wierszu nadpisać wyświetlany komunikat na odpowiedni.

```
1 private void flashLightOn(){
2     CameraManager cameraManager = (CameraManager) getSystemService(
3             Context.CAMERA_SERVICE);
4     try{
5         assert cameraManager != null;
6         String cameraId = cameraManager.getCameraIdList()[0];
7         cameraManager.setTorchMode(cameraId, true);
8         Toast.makeText(latarka.this, "Latarka wlaczona", Toast.
9             LENGTH_SHORT).show();
10    }
11    catch(CameraAccessException e){
12        Log.e("Camera Problem", "Nie mozna uruchomic latarki");
13    }
14 }
```

Listing 2. Latarka - Włączenie/wyłączenie latarki

Aby zaimplementować tryb ciemny potrzebujemy dostępu do modyfikowania interfejsu. Listing 3 (s. 13) pokazuje jak go uzyskać.

```
1  SharedPreferences sharedPreferences = getSharedPreferences("sharedPrefs", MODE_PRIVATE);
2  final SharedPreferences.Editor editor = sharedPreferences.edit();
3  final boolean isDarkModeOn = sharedPreferences.getBoolean("isDarkModeOn", false);
```

Listing 3. Tryb Ciemny - Modyfikowanie interfejsu

Listing 4 (s. 13) przedstawia metodę odpowiedzialną za działanie trybu ciemnego. Domyślnie aplikacja korzysta z trybu jasnego, co pokazuje kod od 5 do 10 linijki. Jednak po naciśnięciu na przycisku, zmieniamy kolorystykę aplikacji na ciemną (13-15 linijka), dodatkowo od 16 linijki zmieniamy przycisk na czerwony, aktualne logo na ciemne oraz wyświetlamy stosowny komunikat.

```
1  toggle_tryb_ciemny.setOnClickListener(new View.OnClickListener() {
2      @Override
3      public void onClick(View view) {
4          if(isDarkModeOn){
5              AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_NO);
6              editor.putBoolean("isDarkModeOn", false);
7              editor.apply();
8              toggle_tryb_ciemny.setImageResource(R.drawable.tryb_ciemny_off);
9              logo.setImageResource(R.drawable.logo_white);
10             Toast.makeText(toggle_tryb_ciemny.this, "Tryb ciemny włączony",
11                             Toast.LENGTH_SHORT).show();
12         }
13         else {
14             AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_YES);
15             editor.putBoolean("isDarkModeOn", true);
16             editor.apply();
17             toggle_tryb_ciemny.setImageResource(R.drawable.tryb_ciemny_on);
18             logo.setImageResource(R.drawable.logo_black);
19             Toast.makeText(toggle_tryb_ciemny.this, "Tryb ciemny włączony",
20                             Toast.LENGTH_SHORT).show();
21         }
22     });
23 };
```

Listing 4. Tryb Ciemny - Włączenie/wyłączenie trybu ciemnego

Listing 5 (s. 14) prezentuje uzyskanie dostępu do sensora, za pomocą którego sprawdzimy działanie czujnika zbliżeniowego.

```
1 SensorManager sensorManager;
2 sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
3 if(sensorManager!=null) {
4     Sensor proximitySensor = sensorManager.getDefaultSensor(Sensor.
5         TYPE_PROXIMITY);
6     if(proximitySensor!=null) {
7         sensorManager.registerListener(this, proximitySensor,
8             sensorManager.SENSOR_DELAY_NORMAL);
9     }
10 }
```

Listing 5. Czujnik Zbliżeniowy - Dostęp do czujnika

Metoda onSensorChanged ukazana na listingu 6 (s. 14) sprawdza czy przy czujniku jest obiekt, zależnie od wyniku zostanie wyświetlony odpowiedni komunikat.

```
1 @Override
2 public void onSensorChanged(SensorEvent sensorEvent) {
3     if(sensorEvent.sensor.getType()==Sensor.TYPE_PROXIMITY) {
4         if(sensorEvent.values[0]==0) {
5             ((TextView)findViewById(R.id.sensor)).setText("Przy
6                 czujniku jest obiekt");
7         } else {
8             ((TextView)findViewById(R.id.sensor)).setText("Przyloz
9                 obiekt do czujnika");
10        }
11    }
12 }
```

Listing 6. Czujnik Zbliżeniowy - Działanie

Aby wdrożyć test aplikacji trzeba sprawdzić pozwolenie, w tym wypadku interesuje nas uprawnienie dostępu do lokalizacji, co obrazuje listing 7 (s. 14).

```
1 if(ContextCompat.checkSelfPermission(gps.this, Manifest.
2     permission.ACCESS_FINE_LOCATION) != PackageManager.
3     PERMISSION_GRANTED) {
4     ActivityCompat.requestPermissions(gps.this, new String[]{
5         Manifest.permission.ACCESS_FINE_LOCATION
6     }, 1000);
7 }
```

Listing 7. GPS - Dostęp do lokalizacji

Metoda onLocationChanged przedstawiona jako listing 8 (s. 15) odpowiada za wyświetlenie powiadomienia pop-up wyświetlającego długość oraz szerokość geograficzną obecnej lokalizacji oraz pobiera adres tej lokalizacji na podstawie długości i szerokości geograficznej, który wyświetla jako tekst.

```
1  @Override
2  public void onLocationChanged(Location location) {
3      Toast.makeText(this, ""+location.getLatitude()+" , "+location.
4          getLongitude(), Toast.LENGTH_SHORT).show();
5      try {
6          Geocoder geocoder = new Geocoder(gps.this, Locale.getDefault());
7          List<Address> addresses = geocoder.getFromLocation(location.
8              getLatitude(), location.getLongitude(), 1);
9          String address = addresses.get(0).getAddressLine(0);
10         text_location.setText(address);
11     } catch (Exception e) {
12         e.printStackTrace();
13     }
14 }
```

Listing 8. GPS - Wyświetlanie lokalizacji

Listing 9 (s. 15) przedstawia implementację MediaPlayer (linijka 1) do którego podłączamy dźwięk. Teraz w prosty sposób możemy wywołać dźwięk poprzez metodę OnClick.

```
1  final MediaPlayer sound = MediaPlayer.create(this, R.raw.
2      dzwieki_audio);
3  Button btn_dzwiek = findViewById(R.id.btn_dzwiek);
4  btn_dzwiek.setOnClickListener(new View.OnClickListener() {
5      @Override
6      public void onClick(View view) {
7          sound.start();
8          Toast.makeText(dzwiek.this, "Odtwarzanie nagrania", Toast.
9              LENGTH_SHORT).show();
10     }
11 });
12 );
```

Listing 9. Dźwięk - Działanie z wykorzystaniem MediaPlayer

Aby test mikrofonu spełniał swoje zadanie trzeba sprawdzić autoryzację aplikacji. W tym wypadku uzyskujemy pozwolenie na nagrywanie audio, co przedstawia listing 10 (s. 16).

```
1 private void getMicrophonePermission() {  
2     if(ContextCompat.checkSelfPermission(this, Manifest.permission.  
RECORD_AUDIO) == PackageManager.PERMISSION_DENIED) {  
3         ActivityCompat.requestPermissions(this, new String[] {  
Manifest.permission.RECORD_AUDIO}, MICROPHONE_PERMISSION_CODE);  
4     }  
5 }
```

Listing 10. Mikrofon - Dostęp do nagrywania audio

Listing 11 (s. 16) przedstawia metodę z wykorzystaniem MediaRecorder który służy do nagrywania dźwięku lub obrazu. W tym wypadku MediaRecorder wykorzystany zostanie do nagrania dźwięku przez mikrofon w telefonie. Od 4 do 7 linijki określamy źródło dźwięku, format wyjściowy, zaznaczenie deskryptora pliku, definiowanie kodowania dźwięku.

```
1 public void btnRecordPressed(View v) {  
2     try {  
3         mediaRecorder = new MediaRecorder();  
4         mediaRecorder.set AudioSource(MediaRecorder.AudioSource.MIC);  
5         mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.  
THREE_GPP);  
6         mediaRecorder.setOutputFile(getRecordingFilePath());  
7         mediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.  
AMR_NB);  
8         mediaRecorder.prepare();  
9         mediaRecorder.start();  
10  
11         Toast.makeText(this, "Nagrywanie rozpoczęte", Toast.  
LENGTH_LONG).show();  
12     }  
13     catch (Exception e) {  
14         e.printStackTrace();  
15     }
```

Listing 11. Mikrofon - Włączenie nagrywania

Po naciśnięciu przycisku "Stop" nagranie zostaje zakończone, co obrazuje listing 12 (s. 17). MediaRecorder kończy nagrywanie, a opcja release użyta w 3 linijce nie pozwoli na ponowne uruchomienie pliku dźwiękowego w celu dogrania do niego dźwięku.

```
1 public void btnStopPressed(View v) {  
2     mediaRecorder.stop();  
3     mediaRecorder.release();  
4     mediaRecorder = null;  
5  
6     Toast.makeText(this, "Nagrywanie zakończone", Toast.LENGTH_LONG)  
7         .show();  
}
```

Listing 12. Mikrofon - Przerwanie nagrywania

Obiekt MediaPlayer wykorzystany w listingu 13 (s. 17) służy do obsługi odtwarzania plików audio i wideo. W 4 linijce kodu określamy skąd ma zostać pobrany plik, a następnie plik z dźwiękiem zostanie włączony.

```
1 public void btnPlayPressed(View v) {  
2     try {  
3         mediaPlayer = new MediaPlayer();  
4         mediaPlayer.setDataSource(getRecordingFilePath());  
5         mediaPlayer.prepare();  
6         mediaPlayer.start();  
7  
8         Toast.makeText(this, "Odtwarzanie nagrania", Toast.  
9             LENGTH_LONG).show();  
10    }  
11    catch(Exception e){  
12        e.printStackTrace();  
13    }  
}
```

Listing 13. Mikrofon - Odtworzenie nagrania

Metoda przedstawiona na listingu 14 (s. 18), odpowiada za uruchomienie aparatu w telefonie po naciśnięciu przycisku o id btnCam. Od 6 do 8 linijki znajdują się kod odpowiedzialny za uruchomienie aplikacji aparatu.

```
1  btnCam = findViewById(R.id.btnCam);
2  btnCam.setOnClickListener(new View.OnClickListener() {
3      @Override
4      public void onClick(View view) {
5          try {
6              Intent intent = new Intent();
7              intent.setAction(MediaStore.ACTION_IMAGE_CAPTURE);
8              startActivityForResult(intent);
9          } catch (Exception e) {
10              e.printStackTrace();
11          }
12      }
13  });
```

Listing 14. Aparat - Włączenie aparatu

Pętla switch (początek w linijce 6) przedstawiona w listingu 15 (s. 18) odpowiada za informowanie użytkownika o aktualnym statusie wifi. W linijce 7 sprawdzamy czy wifi jest włączone, jeśli jest włączone - wyświetlamy odpowiedni tekst, w 10 linijce sprawdzamy czy wifi jest wyłączone na danym urządzeniu, jeśli tak zostaje wyświetlony adekwatny tekst.

```
1  private BroadcastReceiver wifiStateReceiver = new
2  BroadcastReceiver() {
3      @Override
4      public void onReceive(Context context, Intent intent) {
5          int wifiStateExtra = intent.getIntExtra(WifiManager.
6          EXTRA_WIFI_STATE,
7          WifiManager.WIFI_STATE_UNKNOWN);
8          switch (wifiStateExtra) {
9              case WifiManager.WIFI_STATE_ENABLED:
10                  wifiSwitch.setText("WiFi jest włączone, można pobrać
11                  informacje");
12                  break;
13              case WifiManager.WIFI_STATE_DISABLED:
14                  wifiSwitch.setText("WiFi jest wyłączone, włącz aby pobrać
15                  informacje");
16                  break;
17          }
18      }
19  };
```

Listing 15. Wifi - Sprawdzenie statusu Wifi

Metoda przedstawiona na listingu 16 (s. 19) odpowiada za pobranie informacji o sieci wifi do której urządzenie jest obecnie podłączone oraz wyświetlenia tych informacji. Od 5 do 9 linijki mamy kolejno pobranie informacji o adresie IP telefonu, sformatowanie pobranego IP ma tekst, pobranie adresu MAC routera, SSID oraz wskaźnika mocy naszego połączenia. Pobrane informacje implementujemy jako string (od 11 do 15 linijki) i wyświetlamy jako tekst (linijka 17).

```
1  public void getWifiInformation(View view) {  
2      WifiManager wifiManager = (WifiManager) getSystemService(WIFI_SERVICE);  
3      WifiInfo wifiInfo = wifiManager.getConnectionInfo();  
4  
5      int ip = wifiInfo.getIpAddress();  
6      String ipAddress = Formatter.formatIpAddress(ip);  
7      String bssid = wifiInfo.getBSSID();  
8      String ssid = wifiInfo.getSSID();  
9      int rssi = wifiInfo.getRssi();  
10  
11     String info =  
12         "\n Adres IP: " + ipAddress +  
13         "\n Adres MAC Routera: " + bssid +  
14         "\n SSID: " + ssid +  
15         "\n Wskaznik mocy: " + rssi;  
16     txtWifiInfo.setText(info);  
17 }
```

Listing 16. Wifi - Pobieranie informacji o Wifi

Kod ukazany w listingu 17 (s. 19) jest odpowiedzialny za pobranie marki i modelu urządzenia z którego obecnie korzystamy i wyświetlenie tej informacji jako tekst.

```
1  model = (TextView) findViewById(R.id.model);  
2  String stringBuildModel = "Marka i model: " + Build.MANUFACTURER  
   + " " + Build.MODEL;  
3  model.setText(stringBuildModel);
```

Listing 17. Wyniki - Pobranie marki i modelu urządzenia

Listing 18 (s. 20) ukazuje kawałek layoutu strony z wynikami, użyto tutaj Radio Group⁴. Grupując przyciski, dostajemy gwarancję że w danej chwili będzie możliwa wybrać tylko jeden przycisk radio.

```
1 <RadioGroup  
2     android:id="@+id/test1"  
3     android:layout_width="0dp"  
4     android:layout_height="30dp"  
5     android:layout_marginTop="18dp"  
6     android:orientation="horizontal"  
7     app:layout_constraintEnd_toEndOf="parent"  
8     app:layout_constraintStart_toEndOf="@+id/test_latarka"  
9     app:layout_constraintTop_toBottomOf="@+id/text_pytanie">  
10  
11     <RadioButton  
12         android:id="@+id/tak_latarka"  
13         android:layout_width="60dp"  
14         android:layout_height="30dp"  
15         android:text="Tak" />  
16  
17     <RadioButton  
18         android:id="@+id/nie_latarka"  
19         android:layout_width="60dp"  
20         android:layout_height="30dp"  
21         android:text="Nie" />  
22 </RadioGroup>
```

Listing 18. Wyniki - Radio Group

⁴Android Developers - przykład użycia Radio Group[4].

5. Testowanie

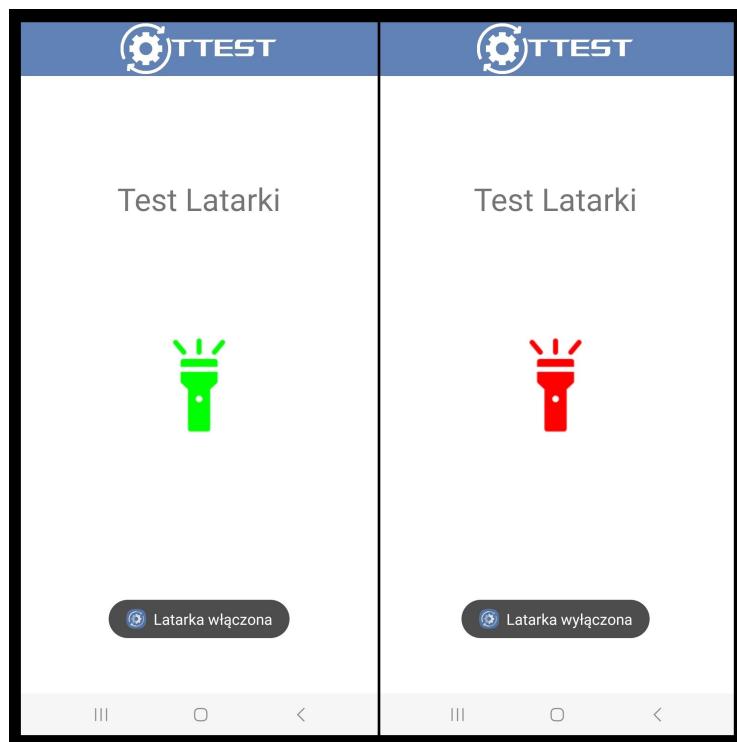
5.1. Testowanie latarki

Zadania jakie ma spełniać test latarki przedstawione są w tabeli 5.1. Jako "X" w kolumnie "Tak" lub "Nie", oznaczamy pomyślny lub niepomyślny przebieg poszczególnych zadań. Całość testu podsumowana jest rzutami ekranu 5.1 wykonanymi podczas testowania jako potwierdzenie wykonanego testu.

Tab. 5.1. Testowanie latarki

lp	Zadania do przetestowania	Tak	Nie
1	Latarka po naciśnięciu na guzik włączyła się	X	
2	Po włączeniu latarki, guzik zmienia kolor na zielony	X	
3	Wyświetlenie komunikatu o włączeniu latarki	X	
4	Latarka po naciśnięciu na guzik wyłączyła się	X	
5	Po wyłączeniu latarki, guzik zmienia kolor na czerwony	X	
6	Wyświetlenie komunikatu o wyłączeniu latarki	X	

Rysunek 5.1 przedstawia zrzuty ekranu potwierdzające pomyślny przebieg testu.



Rys. 5.1. Przebieg testowania latarki

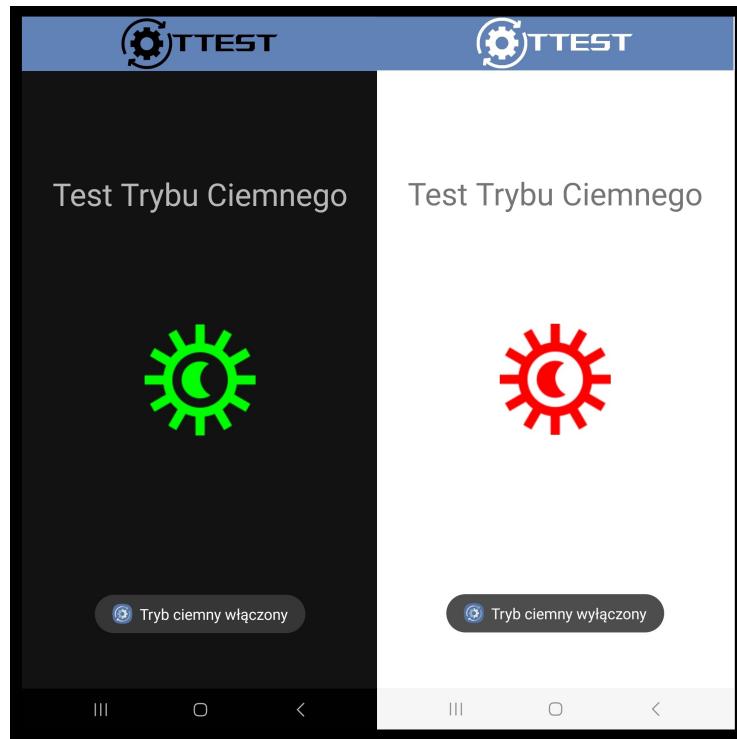
5.2. Testowanie trybu ciemnego

Zadania jakie ma spełniać test trybu ciemnego przedstawione są w tabeli 5.2. Jako "X" w kolumnie "Tak" lub "Nie", oznaczamy pomyślny lub niepomyślny przebieg poszczególnych zadań. Całość testu podsumowana jest rzutami ekranu 5.2 wykonanymi podczas testowania jako potwierdzenie wykonanego testu.

Tab. 5.2. Testowanie trybu ciemnego

lp	Zadania do przetestowania	Tak	Nie
1	Tryb ciemny po naciśnięciu na guzik włączył się	X	
2	Po włączeniu trybu ciemnego, guzik zmienia kolor na zielony	X	
3	Wyświetlenie komunikatu o włączeniu trybu ciemnego	X	
4	Tryb ciemny po naciśnięciu na guzik wyłączył się	X	
5	Po wyłączeniu trybu ciemnego, guzik zmienia kolor na czerwony	X	
6	Wyświetlenie komunikatu o wyłączeniu trybu ciemnego	X	

Rysunek 5.2 przedstawia zrzuty ekranu potwierdzające pomyślny przebieg testu.



Rys. 5.2. Przebieg testowania trybu ciemnego

5.3. Testowanie czujnika zbliżeniowego

Zadania jakie ma spełniać test czujnika zbliżeniowego przedstawione są w tabeli 5.3. Jako "X" w kolumnie "Tak" lub "Nie", oznaczamy pomyślny lub niepomyślny przebieg poszczególnych zadań. Całość testu podsumowana jest rzutami ekranu 5.3 wykonanymi podczas testowania jako potwierdzenie wykonanego testu.

Tab. 5.3. Testowanie czujnika zbliżeniowego

lp	Zadania do przetestowania	Tak	Nie
1	Po zbliżeniu obiektu do czujnika zmienia się tekst informacyjny	X	
2	Po oddaleniu obiektu od czujnika zmienia się tekst informacyjny	X	

Rysunek 5.3 przedstawia zrzuty ekranu potwierdzające pomyślny przebieg testu.



Rys. 5.3. Przebieg testowania czujnika zbliżeniowego

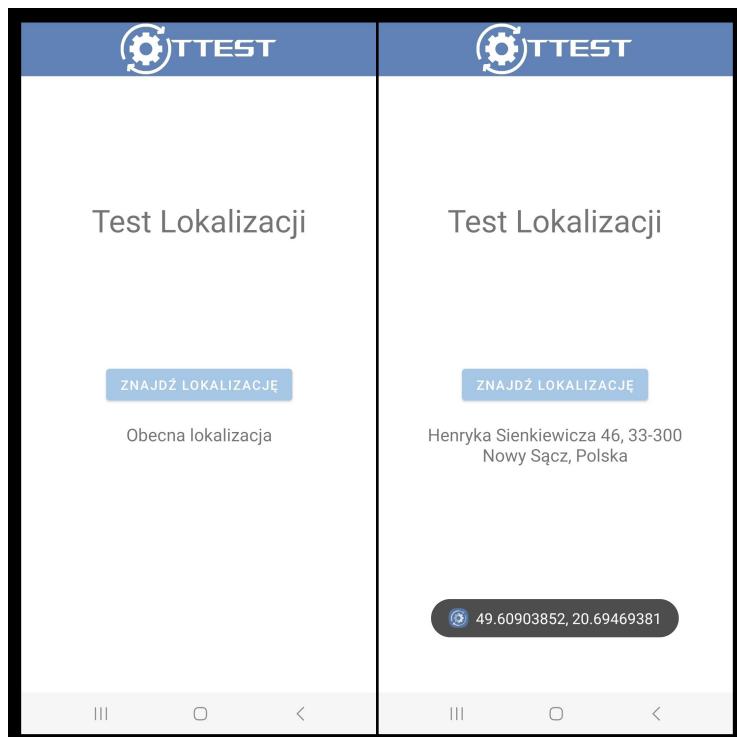
5.4. Testowanie lokalizacji

Zadania jakie ma spełniać test lokalizacji przedstawione są w tabeli 5.4. Jako "X" w kolumnie "Tak" lub "Nie", oznaczamy pomyślny lub niepomyślny przebieg poszczególnych zadań. Całość testu podsumowana jest rzutami ekranu 5.4 wykonanymi podczas testowania jako potwierdzenie wykonanego testu.

Tab. 5.4. Testowanie lokalizacji

lp	Zadania do przetestowania	Tak	Nie
1	Po naciśnięciu na guzik wyświetla się adres obecnej lokalizacji	X	
2	Po naciśnięciu na guzik wyświetla się komunikat z współrzędnymi	X	

Rysunek 5.4 przedstawia zrzuty ekranu potwierdzające pomyślny przebieg testu.



Rys. 5.4. Przebieg testowania lokalizacji

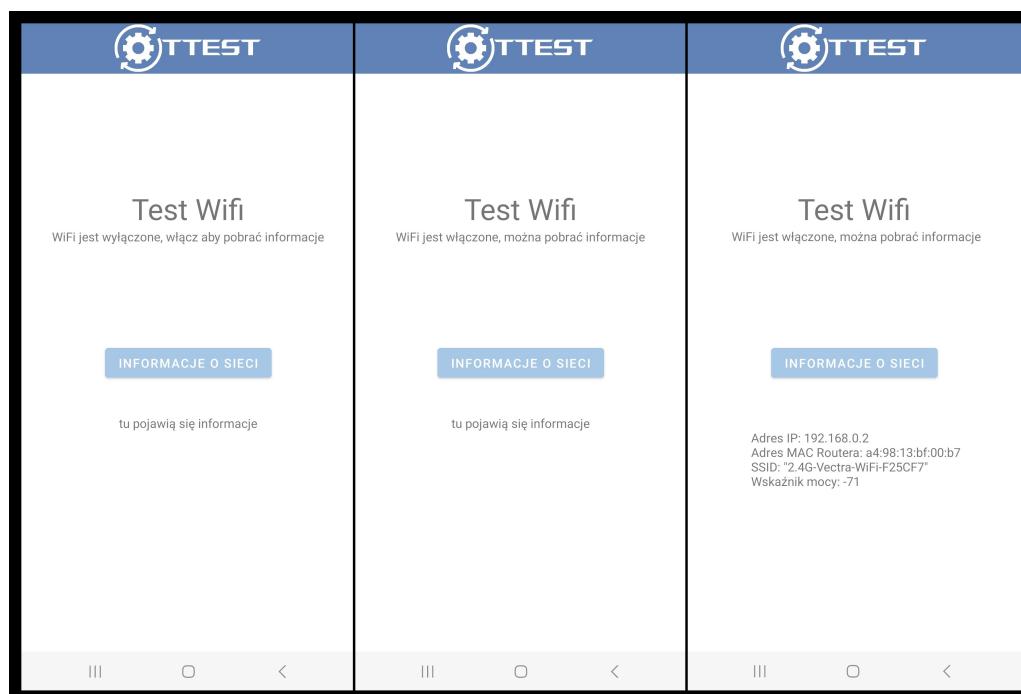
5.5. Testowanie wifi

Zadania jakie ma spełniać test wifi przedstawione są w tabeli 5.5. Jako "X" w kolumnie "Tak" lub "Nie", oznaczamy pomyślny lub niepomyślny przebieg poszczególnych zadań. Całość testu podsumowana jest rzutami ekranu 5.5 wykonanymi podczas testowania jako potwierdzenie wykonanego testu.

Tab. 5.5. Testowanie wifi

lp	Zadania do przetestowania	Tak	Nie
1	Pojawia się informacja o statusie wifi	X	
2	Po naciśnięciu na guzik pojawia się adres IP telefonu	X	
3	Po naciśnięciu na guzik pojawia się adres MAC routera	X	
4	Po naciśnięciu na guzik pojawia się SSID sieci	X	
5	Po naciśnięciu na guzik pojawia się wskaźnik mocy	X	

Rysunek 5.5 przedstawia zrzuty ekranu potwierdzające pomyślny przebieg testu.



Rys. 5.5. Przebieg testowania wifi

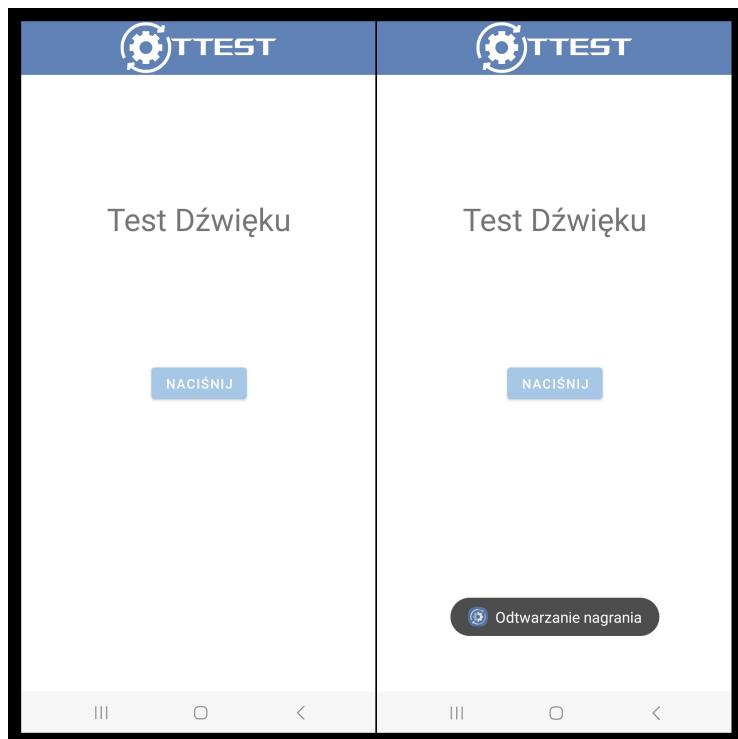
5.6. Testowanie dźwięku

Zadania jakie ma spełniać test dźwięku przedstawione są w tabeli 5.6. Jako "X" w kolumnie "Tak" lub "Nie", oznaczamy pomyślny lub niepomyślny przebieg poszczególnych zadań. Całość testu podsumowana jest rzutami ekranu 5.6 wykonanymi podczas testowania jako potwierdzenie wykonanego testu.

Tab. 5.6. Testowanie dźwięku

lp	Zadania do przetestowania	Tak	Nie
1	Po naciśnięciu na guzik z głośników słyszać śpiew ptaków	X	
2	Wyświetlenie komunikatu o rozpoczęciu odtwarzania nagrania	X	

Rysunek 5.6 przedstawia zrzuty ekranu potwierdzające pomyślny przebieg testu.



Rys. 5.6. Przebieg testowania dźwięku

5.7. Testowanie mikrofonu

Zadania jakie ma spełniać test mikrofonu przedstawione są w tabeli 5.7. Jako "X" w kolumnie "Tak" lub "Nie", oznaczamy pomyślny lub niepomyślny przebieg poszczególnych zadań. Całość testu podsumowana jest rzutami ekranu 5.7 wykonanymi podczas testowania jako potwierdzenie wykonanego testu.

Tab. 5.7. Testowanie mikrofonu

lp	Zadania do przetestowania	Tak	Nie
1	Naciśnięcie na guzik "START" zaczyna nagrywanie	X	
2	Po naciśnięciu na guzik "START" wyświetla się adekwatny komunikat	X	
3	Naciśnięcie na guzik "STOP" kończy nagrywanie	X	
4	Po naciśnięciu na guzik "STOP" wyświetla się adekwatny komunikat	X	
5	Naciśnięcie na guzik "PLAY" odtwarza nagranie	X	
6	Po naciśnięciu na guzik "PLAY" wyświetla się adekwatny komunikat	X	

Rysunek 5.7 przedstawia zrzuty ekranu potwierdzające pomyślny przebieg testu.



Rys. 5.7. Przebieg testowania mikrofonu

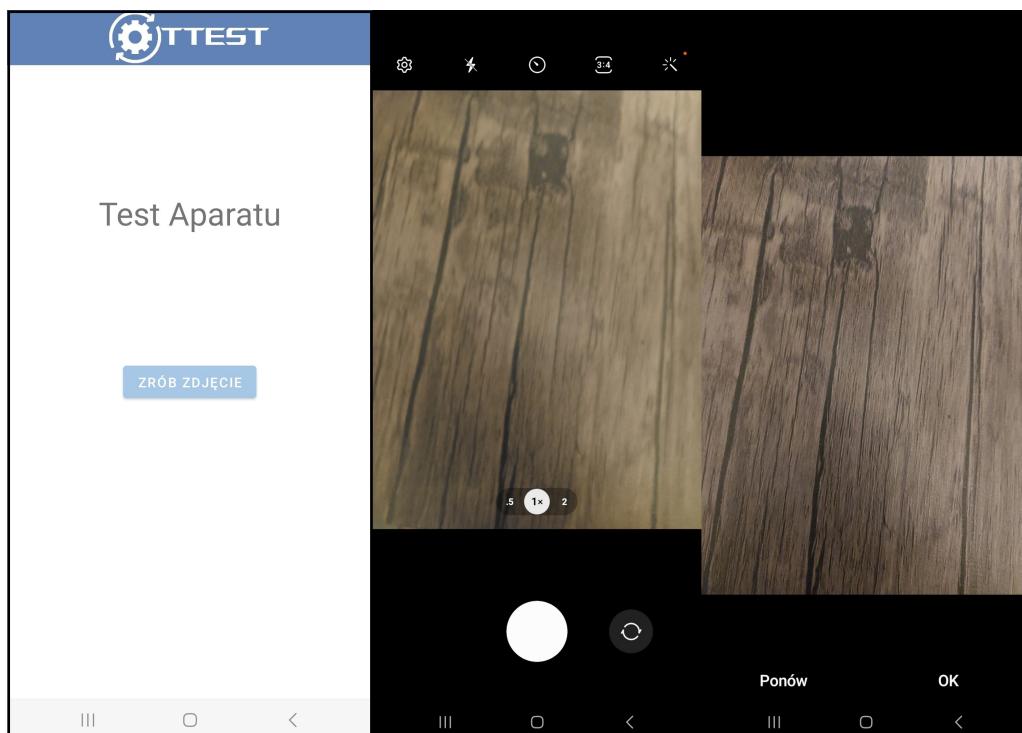
5.8. Testowanie aparatu

Zadania jakie ma spełniać test mikrofonu przedstawione są w tabeli 5.8. Jako "X" w kolumnie "Tak" lub "Nie", oznaczamy pomyślny lub niepomyślny przebieg poszczególnych zadań. Całość testu podsumowana jest rzutami ekranu 5.8 wykonanymi podczas testowania jako potwierdzenie wykonanego testu.

Tab. 5.8. Testowanie aparatu

lp	Zadania do przetestowania	Tak	Nie
1	Po naciśnięciu na guzik włącza się aplikacja aparatu	X	
2	Można wykonać zdjęcie poprzez naciśnięcie białego guzika	X	
3	Jako potwierdzenie wykonania zdjęcia, wyświetla się ono	X	

Rysunek 5.8 przedstawia zrzuty ekranu potwierdzające pomyślny przebieg testu.



Rys. 5.8. Przebieg testowania aparatu

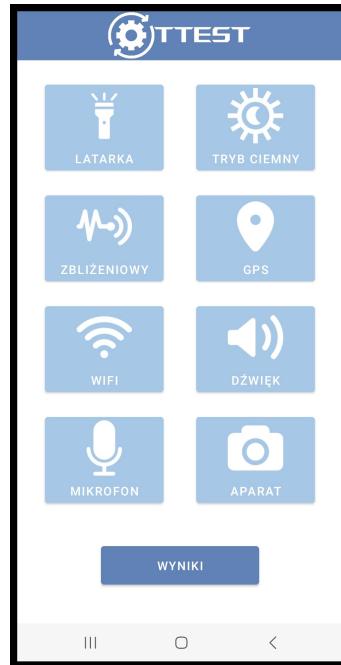
5.9. Testowanie menu

Zadania jakie ma spełniać menu przedstawione są w tabeli 5.9. Jako "X" w kolumnie "Tak" lub "Nie", oznaczamy pomyślny lub niepomyślny przebieg poszczególnych zadań. Całość testu podsumowana jest rzutami ekranu 5.9 wykonanymi podczas testowania jako potwierdzenie wykonanego testu.

Tab. 5.9. Testowanie menu

lp	Zadania do przetestowania	Tak	Nie
1	Guzik "Latarka" przekierowuje do testu latarki	X	
2	Guzik "Tryb ciemny" przekierowuje do testu trybu ciemnego	X	
3	Guzik "Zbliżeniowy" przekierowuje do testu czujnika zbliżeniowego	X	
4	Guzik "GPS" przekierowuje do testu lokalizacji	X	
5	Guzik "Wifi" przekierowuje do testu wifi	X	
6	Guzik "Dźwięk" przekierowuje do testu dźwięku	X	
7	Guzik "Mikrofon" przekierowuje do testu mikrofonu	X	
8	Guzik "Aparat" przekierowuje do testu aparatu	X	
9	Guzik "Wyniki" przekierowuje do podsumowania testów	X	

Rysunek 5.9 przedstawia zrzut ekranu potwierdzający pomyślny przebieg testu.



Rys. 5.9. Przebieg testowania menu

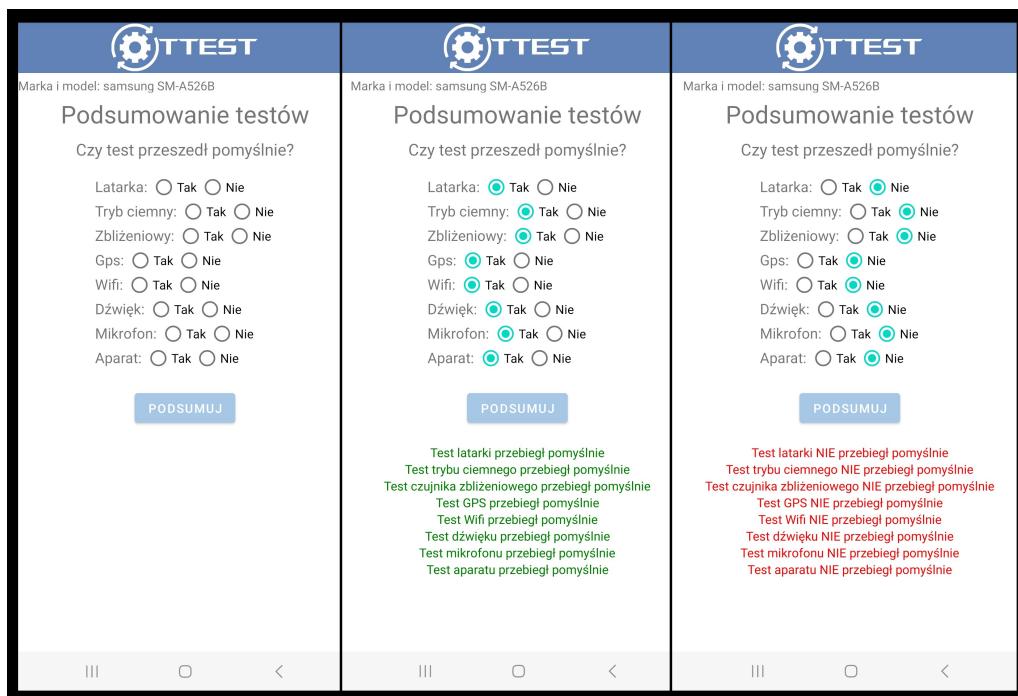
5.10. Testowanie podsumowania

Zadania jakie ma spełniać podsumowanie przedstawione są w tabeli 5.10. Jako "X" w kolumnie "Tak" lub "Nie", oznaczamy pomyślny lub niepomyślny przebieg poszczególnych zadań. Całość testu podsumowana jest rzutami ekranu 5.10 wykonanymi podczas testowania jako potwierdzenie wykonanego testu.

Tab. 5.10. Testowanie podsumowania

lp	Zadania do przetestowania	Tak	Nie
1	Wyświetla się informacja o marce i modelu telefonu	X	
2	W każdej grupie można wybrać tylko jeden przycisk typu radio	X	
3	Po naciśnięciu na guzik pojawia się podsumowanie zaznaczonych opcji	X	
4	Pozytywne wyniki testów są wyświetlane na zielono	X	
5	Negatywne wyniki testów są wyświetlane na czerwono	X	

Rysunek 5.10 przedstawia zrzuty ekranu potwierdzające pomyślny przebieg testu.

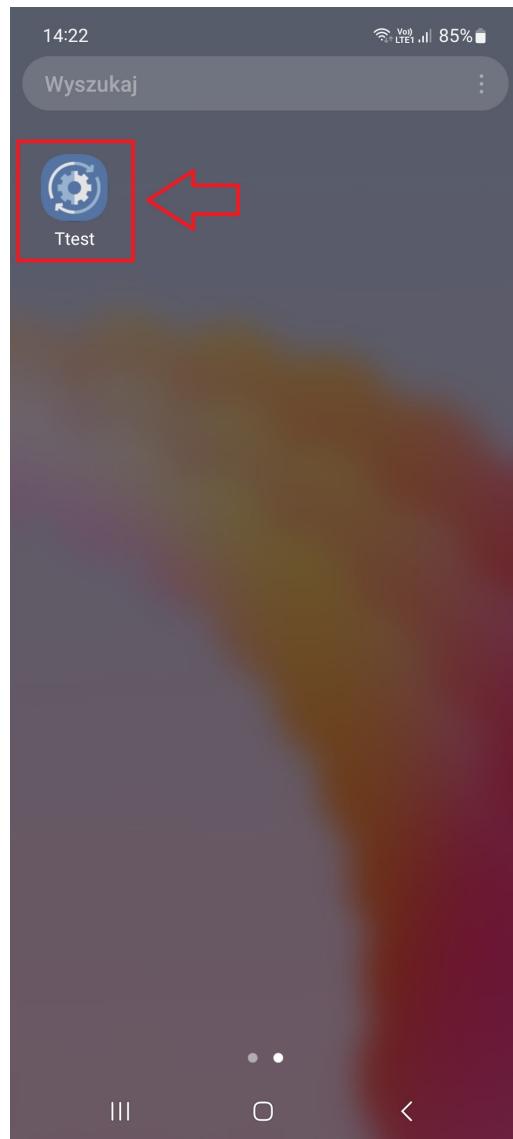


Rys. 5.10. Przebieg testowania podsumowania

6. Podręcznik użytkownika

6.1. Wyszukiwanie i uruchomienie aplikacji Ttest

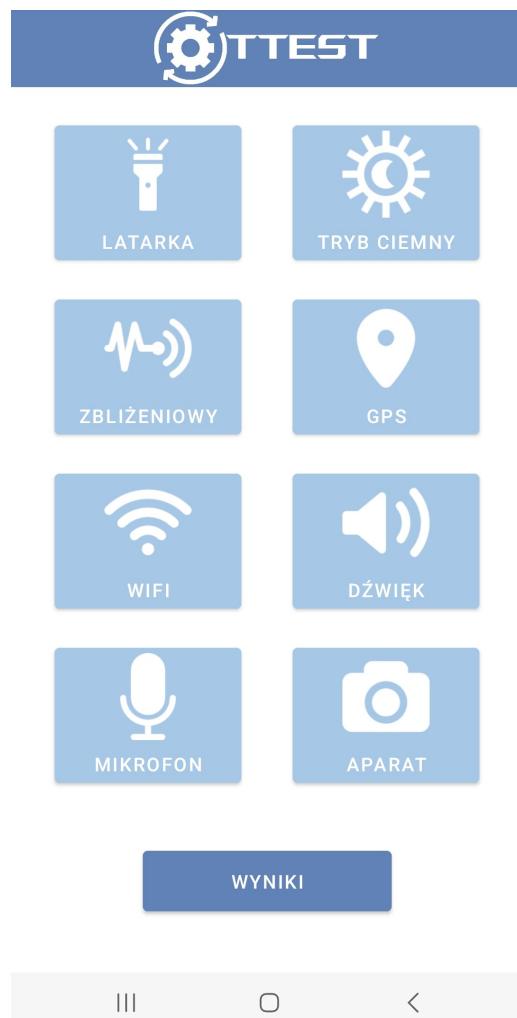
Pierwszym krokiem jaki musimy podjąć, aby uruchomić aplikację, jest włączenie telefonu a następnie wyszukujemy po nazwie aplikację: Ttest. Powinno nam wykoczyć okienko z nazwą i ikonką. Rysunek 6.1 przedstawia efekt po wyszukaniu aplikacji Ttest.



Rys. 6.1. Wyszukiwanie aplikacji

Gdy już odnaleźliśmy aplikację Ttest, klikamy w nią a program automatycznie przekierowuje nas do menu głównego aplikacji.

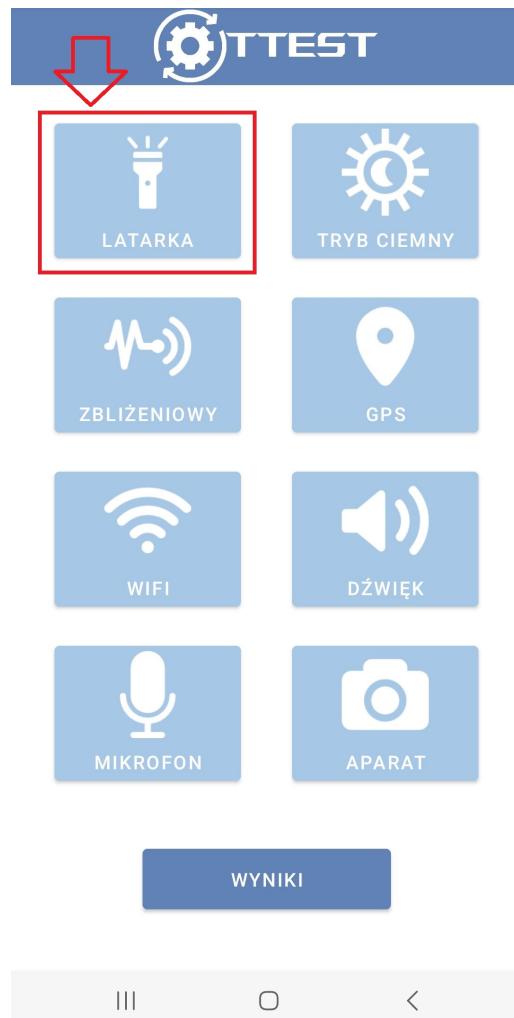
Menu składa się z ośmiu różnych testów. Są to testy: latarki, trybu ciemnego, czujnika zbliżeniowego, lokalizacji, wifi, dźwięku, mikrofonu oraz aparatu. Każdy jeden z testów kryje się pod przyciskiem, który został ówcześnie oznaczony odpowiednią grafiką odpowiadającą danemu testowi oraz podpisem znajdującym się u dołu obrazka. Na samym dole pod testami znajduje się przycisk z napisem "Wyniki", jest to swego rodzaju podsumowanie działalności wszystkich testów znajdujących się w aplikacji. Rysunek 6.2 prezentuje widok menu głównego.



Rys. 6.2. Menu główne

6.2. Test latarki

Aby uruchomić latarkę w menu głównego wybieramy i klikamy na ikonkę latarki, która znajduje się na samej górze po lewej stronie. Rysunek 6.3 ukazuje kafelek który należy wybrać.



Rys. 6.3. Wybór latarki

Następnie aplikacja przekierowuje nas do testu latarki. Pojawia się okienko a na środku znajduje się grafika przedstawiająca latarkę. Gdy naciśniemy na latarkę jej kolor zmieni się z szarego na zielony, oznacza to że test przeszedł pomyślnie i latarka włączyła się. Oprócz zielonego koloru pod latarką wyświetla się komunikat informujący nas o tym że latarka została włączona. Rysunek 6.4 przedstawia zrzut ekranu potwierdzający pomyślny przebieg testu.



Test Latarki



Rys. 6.4. Włączenie latarki

Analogicznie jak w przypadku włączenia gdy naciśniemy na latarkę kolejny raz, test zostanie przerwany, latarka wyłączy się i zmieni kolor z zielonego na czerwony. Rysunek 6.5 przedstawia wyłączoną latarkę.



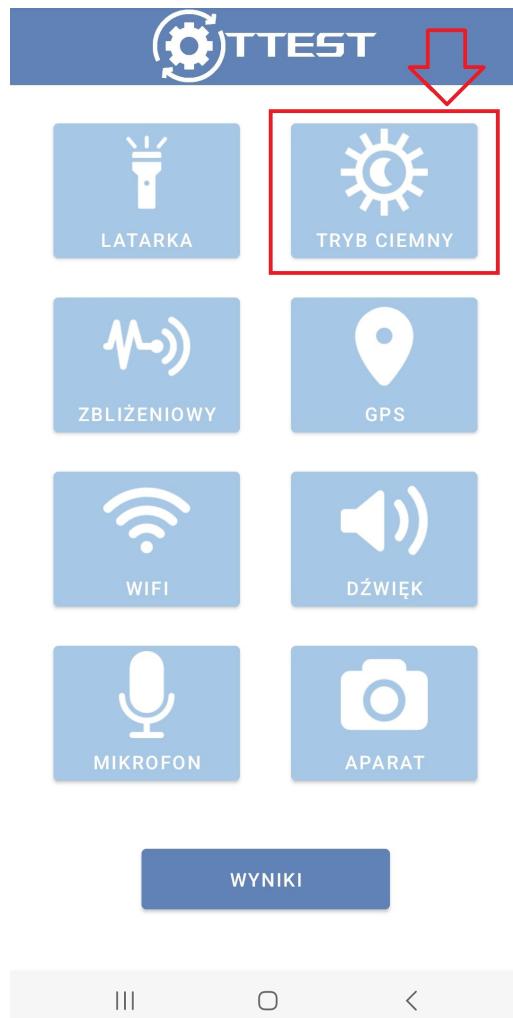
Test Latarki



Rys. 6.5. Wyłączenie latarki

6.3. Test trybu ciemnego

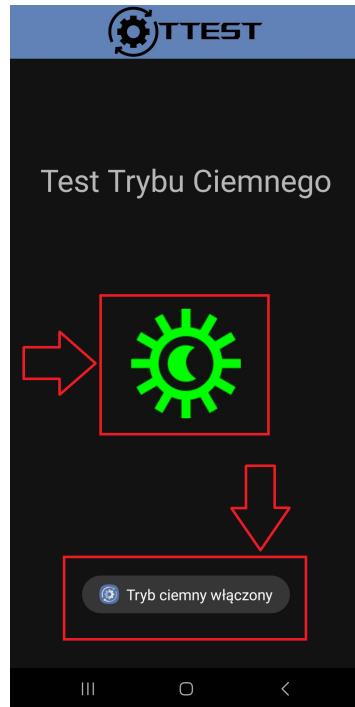
Aby uruchomić tryb ciemny musimy na panelu głównym wybrać i kliknąć w ikonkę który znajduje się na samej górze po prawej stronie. Rysunek 6.6 obrazuje kafelek który musimy wybrać i kliknąć aby przejść do testu.



Rys. 6.6. Wybór trybu ciemnego

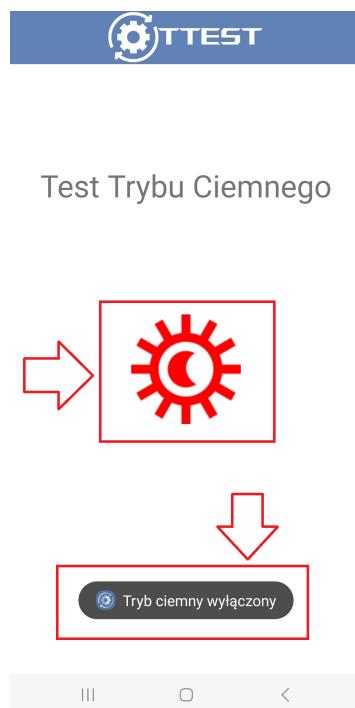
Następnie aplikacja przekierowuje nas do okienka testowego. Na środku znajduje się grafika, która podobnie jak w przypadku latarki zmienia kolory z zielonego na czerwony, lub odwrotnie w zależności czy tryb zostanie włączony lub wyłączony. Jeżeli klikniemy na ikonkę a jej kolor zmieni się na zielony oznacza to że test przeszedł pomyślnie, co więcej ekran zmieni kolor z białego na czarny a pod grafiką dodatkowo pojawia się informacja o tym że tryb ciemny został uruchomiony.

Rysunek 6.7 przedstawia uruchomienie trybu ciemnego.



Rys. 6.7. Włączenie trybu ciemnego

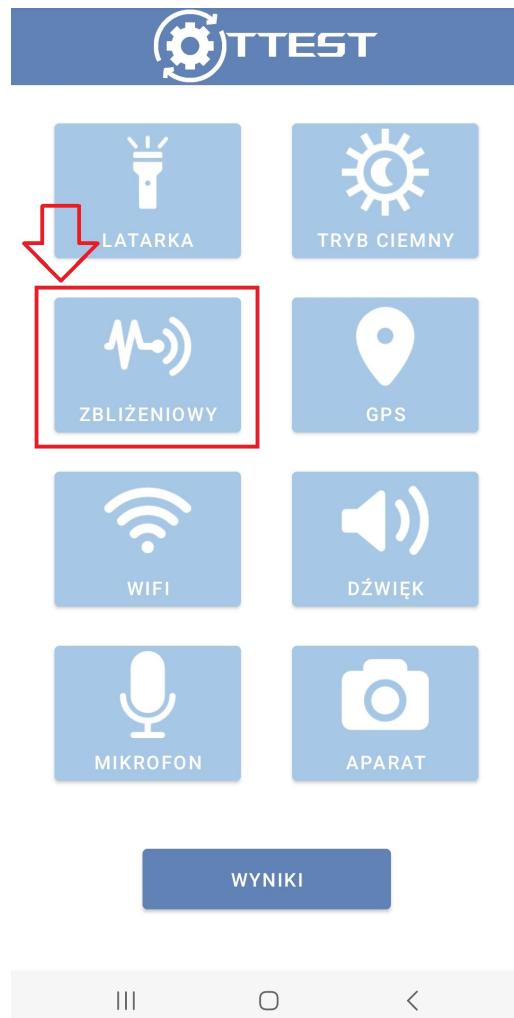
Po kliknięciu ponownie na środkowy przycisk tryb ciemny zostaje wyłączony, kolor ikonki zmienia się na czerwony, a komunikat u dołu informuje nas o wyłączeniu trybu ciemnego. Rysunek 6.8 przedstawia wyłączenie trybu ciemnego.



Rys. 6.8. Wyłączenie trybu ciemnego

6.4. Test czujnika zbliżeniowego

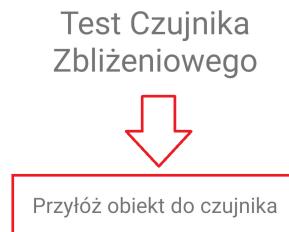
Aby uruchomić czujnik zbliżeniowy musimy w menu wybrać i kliknąć w ikonkę który znajduje się na pod ikonką latarki. Rysunek 6.9 obrazuje kafelek, który musimy wybrać i kliknąć aby przejść do testu.



Rys. 6.9. Lokalizacja czujnika zbliżeniowego

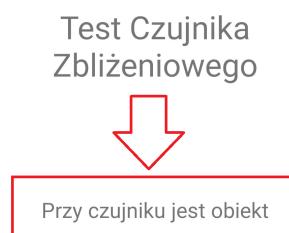
Po przejściu do strony z testem wyświetla nam się informacja aby przyłożyć obiekt do czujnika. Rysunek 6.10 prezentuje stronę z testem czujnika.

Czujnik znajduje się przy przedniej kamerce telefonu wystarczy przyłożyć palec a informacja zmienia się i zaczyna wyświetlać komunikat o treści : Przy czujniku jest obiekt. Jeżeli odsuniemy palec od czujnika informacja ponownie zacznie wyświetlać komunikat o treści: Przyłóż obiekt do czujnika.



Rys. 6.10. Strona testowa czujnika zbliżeniowego

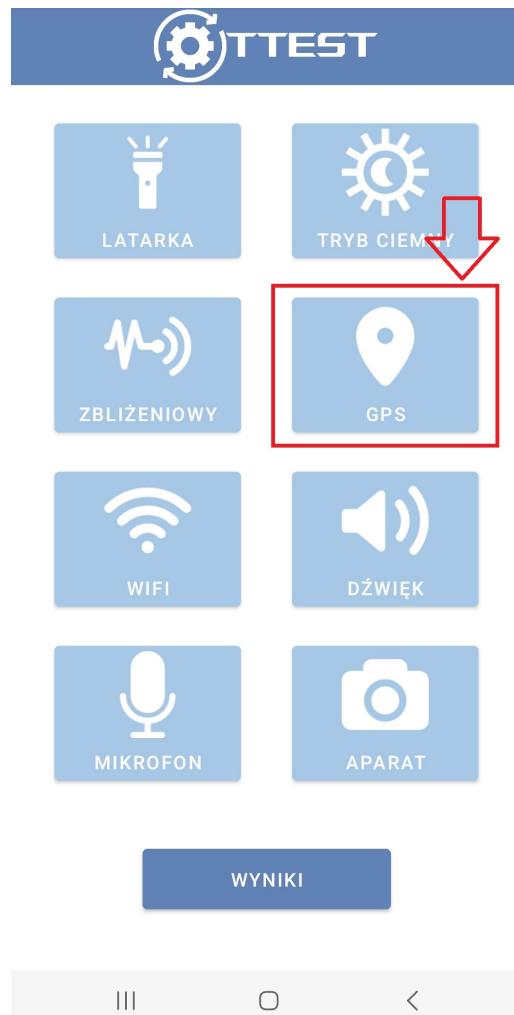
Rysunek 6.11 przedstawia zrzut ekranu potwierdzający pomyslny przebieg testu.



Rys. 6.11. Działanie czujnika zbliżeniowego

6.5. Test GPS

Aby uruchomić test lokalizacji musimy w menu wybrać i kliknąć w przycisk który znajduje się na pod ikoną trybu ciemnego. Rysunek 6.12 obrazuje kafelek, który musimy wybrać i kliknąć aby przejść do testu.



Rys. 6.12. Wybór testu lokalizacji

Po przejściu do strony z testem na środku ekranu wyświetla nam się przycisk, którego celem jest wyszukanie naszej aktualnej lokalizacji. Rysunek 6.13 prezentuje stronę z testem lokalizacji.

Gdy klikniemy na przycisk po krótkiej chwili wyświetli nam się dokładna lokalizacja - adres, kod pocztowy, miasto oraz kraj w którym obecnie się znajdujemy. Oprócz wyżej wymienionych informacji u dołu pod lokalizacją zostaje wyświetlona w formie pop-up'u szerokość i długość geograficzna.



Test Lokalizacji



Rys. 6.13. Stan testowa lokalizacji

Rysunek 6.14 przedstawia zrzut ekranu potwierdzający pomyślny przebieg testu.



Test Lokalizacji



Rys. 6.14. Działanie lokalizacji

6.6. Test Wifi

Aby uruchomić test wifi, w menu głównego wybieramy i klikamy na ikonkę, która znajduje się tuż pod testem czujnika zbliżeniowego. Rysunek 6.15 ukazuje kafelek który należy wybrać aby przejść do testu wifi.



Rys. 6.15. Wybór wifi

Po przejściu do strony z testem wifi otrzymujemy informację aby uruchomić wifi na naszym telefonie i połączyć się z dostępną siecią. W innym przypadku nie będzie możliwe wykonanie testu oraz nie otrzymamy informacji o sieci.

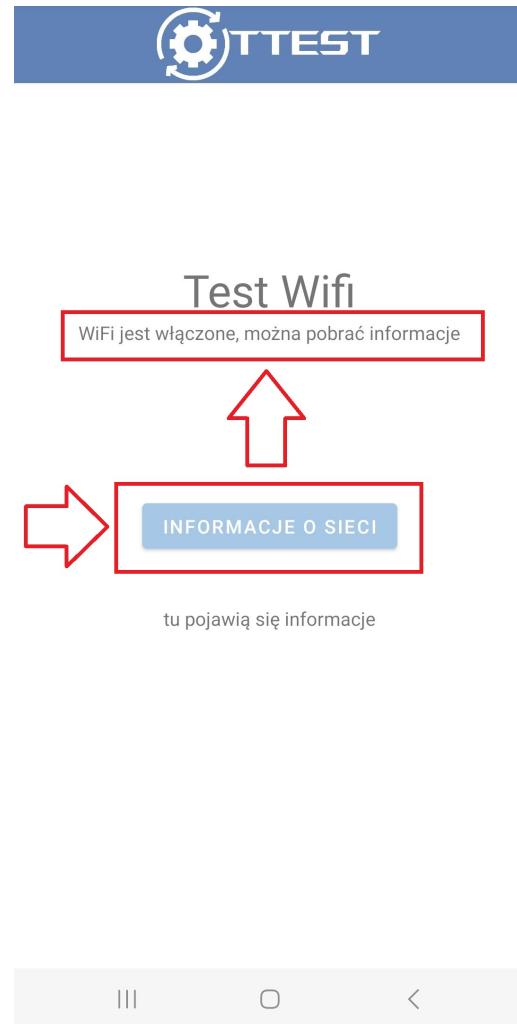
Rysunek 6.16 prezentuje stronę z testem wifi.



Rys. 6.16. Strona testowa wifi

Po połączeniu się z siecią za pomocą wifi, otrzymujemy komunikat który mówi, że aktualnie jesteśmy połączeni z siecią i możemy pobrać dostępne informacje o niej. Abytrzymać informacje o sieci należy kliknąć na przycisk znajdujący się na środku "Informacje o sieci".

Rysunek 6.17 prezentuje stronę z testem wifi po podłączeniu się do sieci.



Rys. 6.17. Test wifi

Po naciśnięciu na przycisk "Informacje o sieci" pojawią nam się wszystkie dostępne informacje o naszej sieci między innymi: Adres IP, Adres MAC Routera, SSID sieci do której jesteśmy podłączeni oraz wskaźnik mocy.

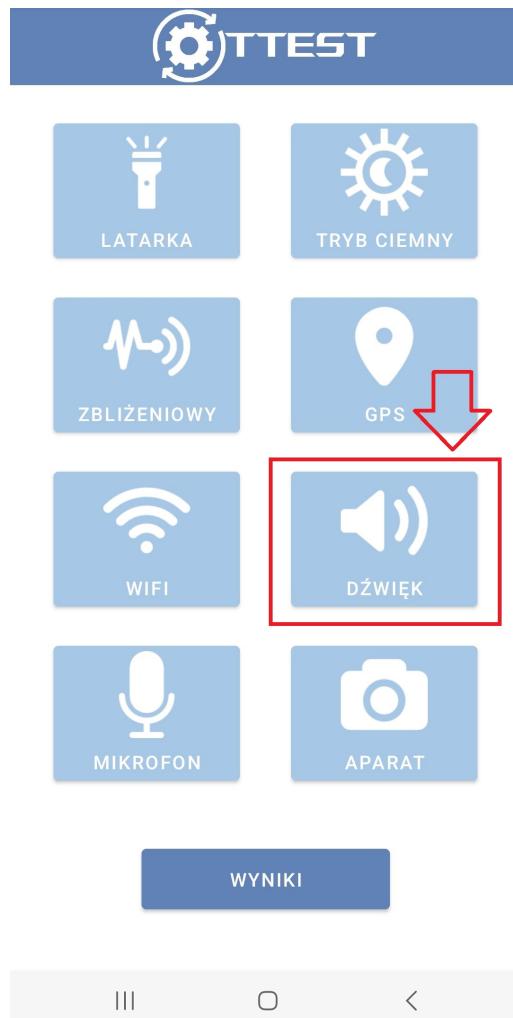
Rysunek 6.18 przedstawia zrzut ekranu potwierdzający pomyślny przebieg testu.



Rys. 6.18. Działanie wifi

6.7. Test dźwięku

Aby przejść do testu dźwięku należy kliknąć przycisk znajdujący się po prawej stronie tuż pod testem lokalizacji. Rysunek 6.19 przedstawia kafelek który należy wybrać aby przejść do testu dźwięku.



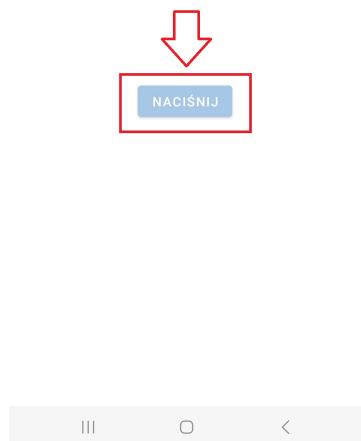
Rys. 6.19. Wybór testu dźwięku

Po przekierowaniu na stronę z testem dostrzegamy że na środku znajduje się przycisk z napisem - "Naciśnij". Rysunek 6.20 prezentuję przycisk uruchamiający test dźwięku.

Gdy naciśniemy przycisk z głośników telefonu usłyszmy dźwięk ćwierkających ptaków. Co więcej pod przyciskiem wyświetli nam się komunikat, który informuje nas o tym, że nagranie jest odtwarzane.



Test Dźwięku



Rys. 6.20. Strona testowa dźwięku

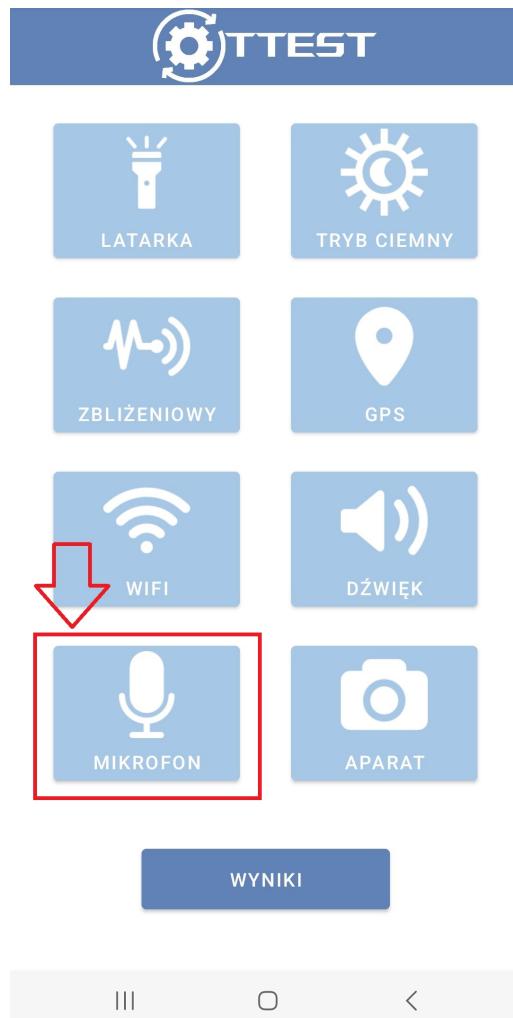
Rysunek 6.21 przedstawia zrzut ekranu potwierdzający pomyslny przebieg testu.



Rys. 6.21. Test dźwięku

6.8. Test mikrofonu

Aby przejść do testu mikrofonu należy kliknąć przycisk znajdujący się po lewej stronie tuż pod testem wifi. Rysunek 6.22 przedstawia kafelek który należy wybrać aby przejść do testu mikrofonu.



Rys. 6.22. Wybór testu mikrofonu

Po przekierowaniu na stronę z testem dostrzegamy że na środku znajduje się trzy przyciski kolejno: Start, Stop oraz Play.

Przycisk z napisem - "Start" rozpoczyna test oraz nagrywa wszystko to co mówimy. Ponadto u dołu ekranu pojawia się komunikat, który informuje nas o rozpoczęciu nagrywania. Rysunek 6.23 prezentuje przycisk startu w teście mikrofonu.



Test Mikrofonu

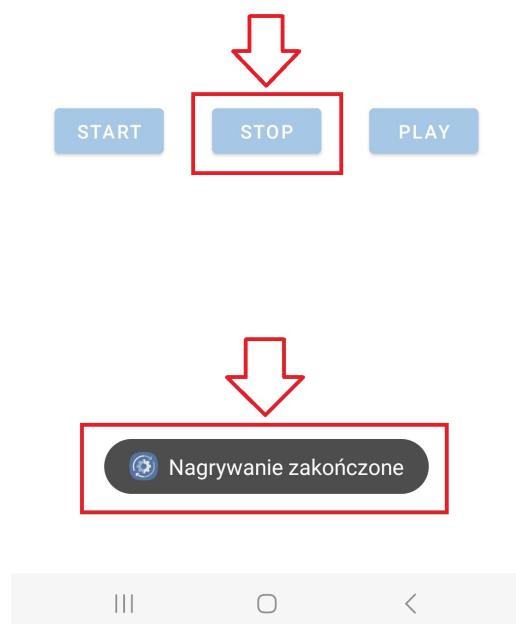


Rys. 6.23. Działanie przycisku Start - Mikrofon

Aby zatrzymać nagrywanie wystarczy przycisnąć drugi przycisk - "Stop". Podobnie jak w przypadku przycisku "Start", u dołu ekranu pojawi się informacja o tym, że nagrywanie zostało zakończone. Rysunek 6.24 prezentuje przycisk stopu w teście mikrofonu.



Test Mikrofonu



Rys. 6.24. Działanie przycisku Stop - Mikrofon

Ostatni z przycisków - "Play" pozwala na odtworzenie nagranego wcześniej nagrania. Pod przyciskami zarówno jak przy przycisku start jak i stop, pojawia się informacja o tym, że nagranie jest odtwarzane. Rysunek 6.25 prezentuję przycisk play w teście mikrofonu.



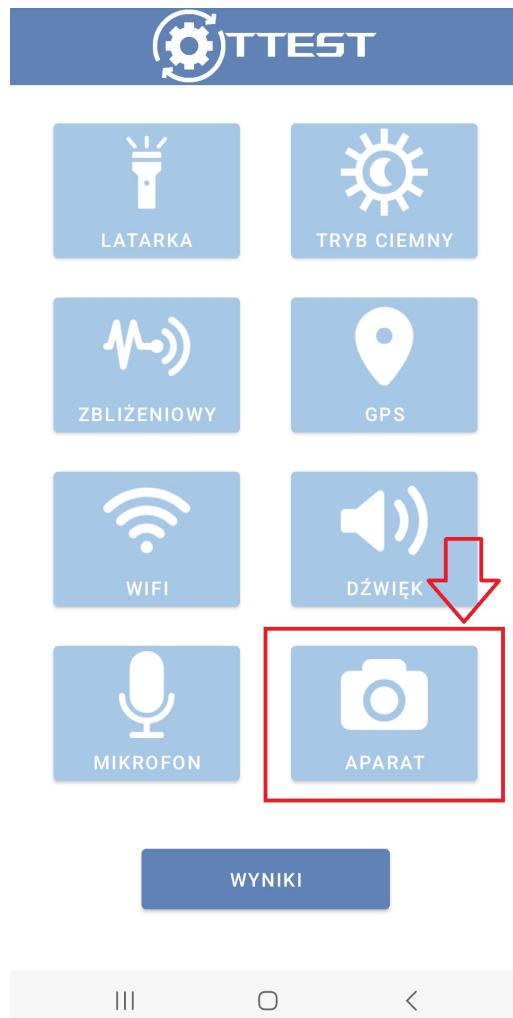
Test Mikrofonu



Rys. 6.25. Działanie przycisku Play - Mikrofon

6.9. Test aparatu

Aby przejść do testu aparatu należy w menu głównym wybrać i kliknąć przycisk znajdujący się po prawej stronie tuż pod testem dźwięku. Rysunek 6.26 przedstawia kafelek który należy wybrać aby przejść do testu aparatu.



Rys. 6.26. Wybór testu aparatu

Po przejściu na stronę z testem dostrzegamy że na środku znajduje się przycisk z napisem - "Zrób zdjęcie". Rysunek 6.27 prezentuję przycisk uruchamiający test aparatu.



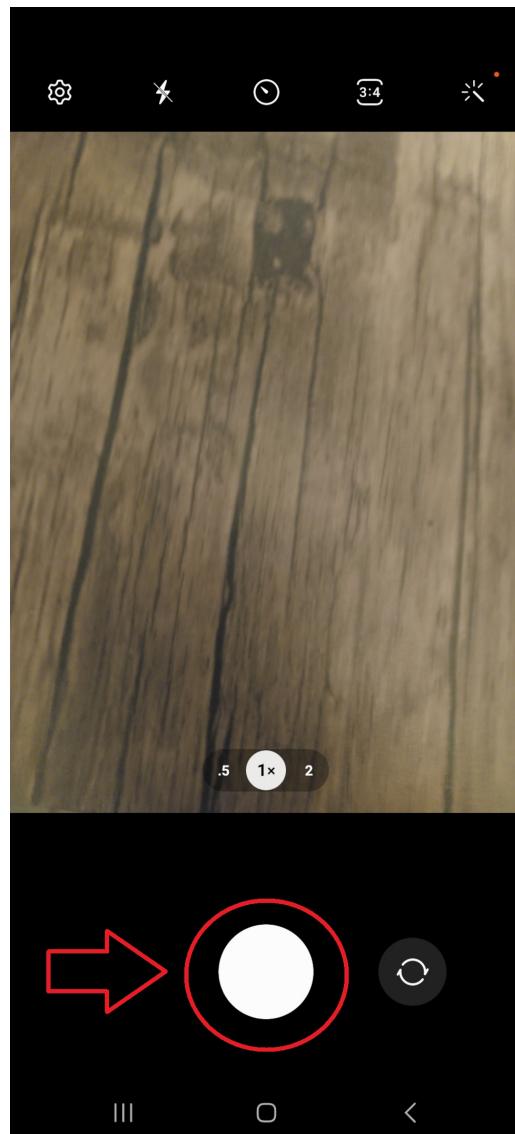
Test Aparatu



Rys. 6.27. Strona testowa aparatu

Gdy naciśniemy przycisk zostaje uruchomiony aparat i możemy wykonać zdjęcie poprzez kliknięcie białego kółka znajdującego się u dołu ekranu.

Rysunek 6.28 prezentuję test aparatu.



Rys. 6.28. Wykonanie zdjęcia- test aparatu

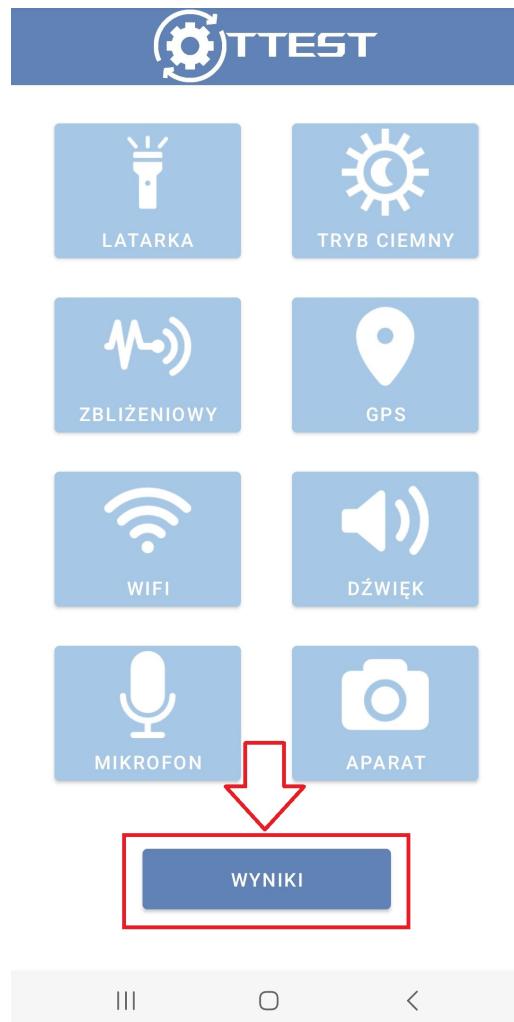
Po wykonaniu zdjęcia, możemy ponowić próbę lub zakończyć test klikając przycisk: "OK". Rysunek 6.29 przedstawia zrzut ekranu potwierdzający pomyślny przebieg testu.



Rys. 6.29. Wykonanie kolejnego zdjęcia lub koniec testu - test aparatu

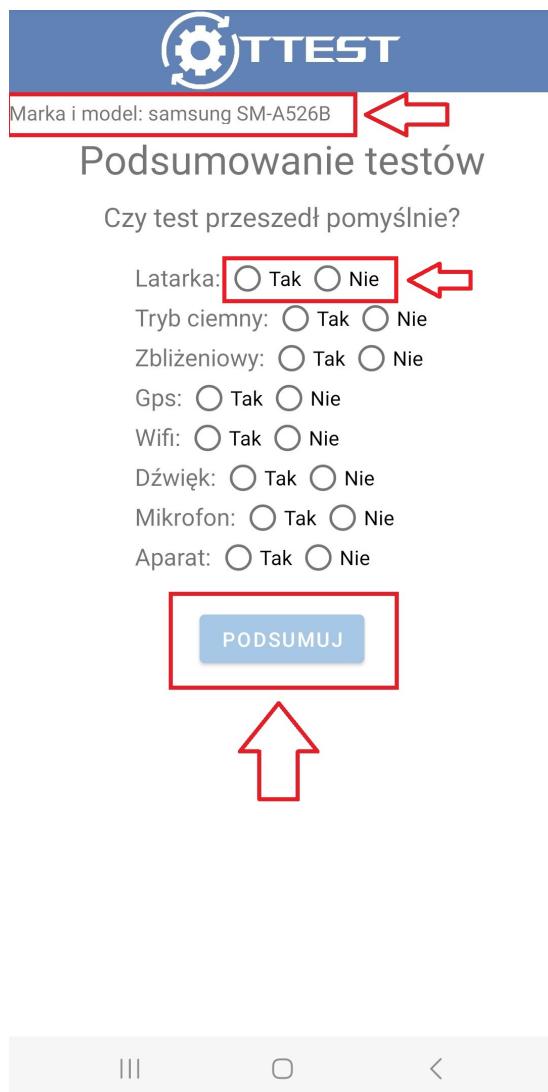
6.10. Podsumowanie wszystkich testów

Aby przejść do wykonania podsumowania wszystkich testów, należy w menu głównym wybrać i kliknąć przycisk: "Wyniki" znajdujący się na samym dole pod wszystkimi testami. Rysunek 6.30 przedstawia kafelek który należy wybrać aby przejść do wyników.



Rys. 6.30. Przejście do sekcji "Wyniki"

Po przejściu na stronę z wynikami dostrzegamy że na samej górze po lewej stronie podany jest marka oraz model naszego telefonu. Co więcej na środku wymienione zostały wszystkie testy, które aplikacja posiada. Naszym zadaniem jest zaznaczyć odpowiednią odpowiedź w zależności czy test odbył się pomyślnie - odpowiedź "Tak" lub w przypadku gdy test nie przeszedł pomyślnie - odpowiedź "Nie". Następnym krokiem jest naciśnięcie przycisku "Podsumuj", który zsumuje wszystkie testy. Rysunek 6.31 prezentuję podsumowanie testów.



Rys. 6.31. Strona z podsumowaniem testów.

W przypadku gdy wszystkie testy przebiegły pomyślnie zostaną one wypisane zieloną czcionką, u dołu ekranu pod przyciskiem podsumowującym.

Rysunek 6.32 prezentuję podsumowanie wszystkich testów, które przebiegły pomyślnie.



Podsumowanie testów

Czy test przeszedł pomyślnie?

Latarka: Tak Nie

Tryb ciemny: Tak Nie

Zbliżeniowy: Tak Nie

Gps: Tak Nie

Wifi: Tak Nie

Dźwięk: Tak Nie

Mikrofon: Tak Nie

Aparat: Tak Nie

PODSUMUJ

Test latarki przebiegły pomyślnie
 Test trybu ciemnego przebiegły pomyślnie
 Test czujnika zbliżeniowego przebiegły pomyślnie
 Test GPS przebiegły pomyślnie
 Test Wifi przebiegły pomyślnie
 Test dźwięku przebiegły pomyślnie
 Test mikrofonu przebiegły pomyślnie
 Test aparatu przebiegły pomyślnie



III ○ <

Rys. 6.32. Podsumowanie wszystkich testów, które przebiegły pomyślnie

Jeżeli wszystkie testy nie przebiegły pomyślnie zostaną one wypisane czerwoną czcionką, u dołu ekranu pod przyciskiem podsumowującym. Rysunek 6.33 prezentuje podsumowanie wszystkich testów, które nie przebiegły pomyślnie.



Podsumowanie testów

Czy test przeszedł pomyślnie?

Latarka: Tak Nie

Tryb ciemny: Tak Nie

Zbliżeniowy: Tak Nie

Gps: Tak Nie

Wifi: Tak Nie

Dźwięk: Tak Nie

Mikrofon: Tak Nie

Aparat: Tak Nie

PODSUMUJ

Test latarki **NIE** przebiegły pomyślnie
 Test trybu ciemnego **NIE** przebiegły pomyślnie
 Test czujnika zbliżeniowego **NIE** przebiegły pomyślnie
 Test GPS **NIE** przebiegły pomyślnie
 Test Wifi **NIE** przebiegły pomyślnie
 Test dźwięku **NIE** przebiegły pomyślnie
 Test mikrofonu **NIE** przebiegły pomyślnie
 Test aparatu **NIE** przebiegły pomyślnie



III ○ <

Rys. 6.33. Podsumowanie wszystkich testów, które nie przebiegły pomyślnie

Bibliografia

- [1] *Język programowania Java - więcej informacji.* URL: <https://geek.justjoin.it/wszystko-co-musicie-wiedziec-o-języku-java-co-to-dla-kogo-i-ile-zarobimy/> (term. wiz. 09.11.2022).
- [2] *Latex - więcej informacji.* URL: <http://smurf.mimuw.edu.pl/node/1306> (term. wiz. 09.11.2022).
- [3] *Git - więcej informacji.* URL: <https://expose.pl/czym-jest-git-i-dlaczego-warto-znac-systemy-kontroli-wersji-kodu/> (term. wiz. 09.11.2022).
- [4] *Strona internetowa Android Developers - przykład użycia Radio Group.* URL: <https://developer.android.com/develop/ui/views/components/radiobutton> (term. wiz. 25.11.2022).

Spis rysunków

1.1.	Logo firmy TTest	4
1.2.	Przykładowy layout	5
2.1.	Ikona aplikacji	6
2.2.	Wygląd menu głównego	7
2.3.	Podsumowanie testów	8
5.1.	Przebieg testowania latarki	21
5.2.	Przebieg testowania trybu ciemnego	22
5.3.	Przebieg testowania czujnika zbliżeniowego	23
5.4.	Przebieg testowania lokalizacji	24
5.5.	Przebieg testowania wifi	25
5.6.	Przebieg testowania dźwięku	26
5.7.	Przebieg testowania mikrofonu	27
5.8.	Przebieg testowania aparatu	28
5.9.	Przebieg testowania menu	29
5.10.	Przebieg testowania podsumowania	30
6.1.	Wyszukiwanie aplikacji	31
6.2.	Menu główne	32
6.3.	Wybór latarki	33
6.4.	Włączenie latarki	34
6.5.	Wyłączenie latarki	34
6.6.	Wybór trybu ciemnego	35
6.7.	Włączenie trybu ciemnego	36
6.8.	Wyłączenie trybu ciemnego	36
6.9.	Lokalizacja czujnika zbliżeniowego	37
6.10.	Strona testowa czujnika zbliżeniowego	38
6.11.	Działanie czujnika zbliżeniowego	38
6.12.	Wybór testu lokalizacji	39
6.13.	Strona testowa lokalizacji	40
6.14.	Działanie lokalizacji	40
6.15.	Wybór wifi	41
6.16.	Strona testowa wifi	42

6.17. Test wifi	43
6.18. Działanie wifi	44
6.19. Wybór testu dźwięku	45
6.20. Strona testowa dźwięku	46
6.21. Test dźwięku	46
6.22. Wybór testu mikrofonu	47
6.23. Działanie przycisku Start - Mikrofon	48
6.24. Działanie przycisku Stop - Mikrofon	49
6.25. Działanie przycisku Play - Mikrofon	50
6.26. Wybór testu aparatu	51
6.27. Strona testowa aparatu	52
6.28. Wykonanie zdjęcia- test aparatu	53
6.29. Wykonanie kolejnego zdjęcia lub koniec testu - test aparatu	54
6.30. Przejście do sekcji "Wyniki"	55
6.31. Strona z podsumowaniem testów.	56
6.32. Podsumowanie wszystkich testów, które przebiegły pomyślnie	57
6.33. Podsumowanie wszystkich testów, które nie przebiegły pomyślnie	57

Spis tabel

5.1. Testowanie latarki	21
5.2. Testowanie trybu ciemnego	22
5.3. Testowanie czujnika zbliżeniowego	23
5.4. Testowanie lokalizacji	24
5.5. Testowanie wifi	25
5.6. Testowanie dźwięku	26
5.7. Testowanie mikrofonu	27
5.8. Testowanie aparatu	28
5.9. Testowanie menu	29
5.10. Testowanie podsumowania	30

Spis listingów

1.	Menu - Działanie Przycisków	12
2.	Latarka - Włączenie/wyłączenie latarki	12
3.	Tryb Ciemny - Modyfikowanie interfejsu	13
4.	Tryb Ciemny - Włączenie/wyłączenie trybu ciemnego	13
5.	Czujnik Zbliżeniowy - Dostęp do czujnika	14
6.	Czujnik Zbliżeniowy - Działanie	14
7.	GPS - Dostęp do lokalizacji	14
8.	GPS - Wyświetlanie lokalizacji	15
9.	Dźwięk - Działanie z wykorzystaniem MediaPlayer	15
10.	Mikrofon - Dostęp do nagrywania audio	16
11.	Mikrofon - Włączenie nagrywania	16
12.	Mikrofon - Przerwanie nagrywania	17
13.	Mikrofon - Odtworzenie nagrania	17
14.	Aparat - Włączenie aparatu	18
15.	Wifi - Sprawdzenie statusu Wifi	18
16.	Wifi - Pobieranie informacji o Wifi	19
17.	Wyniki - Pobranie marki i modelu urządzenia	19
18.	Wyniki - Radio Group	20