# UCZELNIA TECHNICZNO-HANDLOWA

**im. Heleny Chodkowskiej**

**Wydział Inżynieryjny**

**Kierunek Informatyka**

**Egor Lezov**

Nr albumu 31099

**Projekt aplikacji mobilnej do pomocy osobom niewidomym „Głos”**

Praca projektowa wykonana pod kierunkiem

inż.

Warszawa 2023

**Spis treści**

[1. Wstęp 4](#_Toc155726025)

[1.1. Grupa docelowa 4](#_Toc155726026)

[1.2. Potrzeby grupy docelowej 5](#_Toc155726027)

[1.2.1. Potrzeby osób niewidomych 5](#_Toc155726028)

[1.2.2. Potrzeby wolontariuszy 5](#_Toc155726029)

[1.3. Oczekiwania grupy docelowej 6](#_Toc155726030)

[1.3.1. Oczekiwania osób niewidomych 6](#_Toc155726031)

[1.3.2. Oczekiwania wolontariuszy 7](#_Toc155726032)

[1.4. Cel projektu 7](#_Toc155726033)

[2. Wprowadzenie teoretyczne do narzędzi oraz zagadnień 8](#_Toc155726034)

[3. Projekt aplikacji 12](#_Toc155726035)

[3.1. Zakres funkcjonalności 12](#_Toc155726038)

[3.2. Diagram przypadków użycia 14](#_Toc155726039)

[3.3. Diagram przypływu 16](#_Toc155726040)

[3.4. Ekrany aplikacji 19](#_Toc155726041)

[3.4.1. Cechy projektowania ekranów dla osób niewidomych 20](#_Toc155726042)

[3.4.2. Ekran startowy 21](#_Toc155726043)

[3.4.3. Ekran wyboru grupy 22](#_Toc155726053)

[3.4.4. Ekran grupy 23](#_Toc155726064)

[3.4.5. Ekran połączenia 31](#_Toc155726065)

[3.5. Usługi i technologie wykorzystane w projekcie 34](#_Toc155726066)

[3.5.1. Język programowania 34](#_Toc155726067)

[3.5.2. Baza danych 34](#_Toc155726068)

[3.5.3. Mapa 37](#_Toc155726069)

[3.5.4. Komunikacja w czasie rzeczywistym 38](#_Toc155726070)

[3.6. Bezpieczeństwo aplikacji 40](#_Toc155726071)

[4. Spis rysunków 42](#_Toc155726072)

[5. Spis tabel 43](#_Toc155726073)

# Wstęp

W świecie ciągłego rozwoju technologii aplikacji mobilnych stały się integralną częścią naszego codziennego życia, ułatwiając wiele aspektów naszej komunikacji i interakcji z otaczającym nas światem. Jednak w tym dynamicznie rozwijającym się krajobrazie cyfrowym konieczne jest zwrócenie uwagi na tych, którzy często pozostają poza zasięgiem nowych możliwości technologicznych - osoby niewidome. Projekt "Głos" stanowczo zwraca uwagę na ten aspekt i ma na celu stworzenie mobilnej aplikacji przeznaczonej do pomocy osobom niewidomym poprzez wolontariuszom.

## Grupa docelowa

Wybór grupy docelowej - osób niewidomych i wolontariuszy - nie jest przypadkowy. Niewidomi często napotykają na ograniczenia w dostępie do informacji i usług z powodu barier w dziedzinie technologii. Potrzebują specjalistycznych rozwiązań, które uczynią ich codzienne zadania bardziej dostępnymi i efektywnymi. Wolontariusze z kolei odgrywają kluczową rolę w wsparciu osób niewidomych, ich dobrowolna praca stanowi niezbędne połączenie między światem niewidomych a rozwiązaniami technologicznymi. Aplikacja "Głos" dąży do stworzenia platformy, która ułatwi interakcję między tymi dwiema grupami, sprzyjając wzajemnemu zrozumieniu i pomocy.

## Potrzeby grupy docelowej

Grupa docelowa projektu, która obejmuje osoby niewidome i wolontariuszy, ma różnorodne potrzeby, które mogą być zaspokojone poprzez rozwijanie aplikacji mobilnej. Oto główne potrzeby obu grup:

## Potrzeby osób niewidomych

* Dostęp do informacji: osoby niewidome potrzebują dostępu do różnorodnych informacji, takich jak wiadomości, harmonogramy i inne, aby być na bieżąco.
* Nawigacja i orientacja: orientacja w nieznanym otoczeniu i nawigacja w pomieszczeniach mogą być trudne dla osób niewidomych, więc potrzebują narzędzi, które ułatwią im poruszanie się.
* Wsparcie w codziennych zadaniach: osoby niewidome mogą potrzebować pomocy w czytaniu etykiet na produktach, wypełnianiu dokumentów, wyborze odzieży i innych codziennych zadaniach.
* Aktywność społeczna: potrzebują również możliwości utrzymywania kontaktu z ludźmi i dostępu do wymiany informacji.

## Potrzeby wolontariuszy

* Środki komunikacji: wolontariusze muszą mieć dostęp do środków komunikacji z osobami niewidomymi, aby świadczyć pomoc i wsparcie w czasie rzeczywistym.
* Dostęp do informacji: mogą potrzebować dostępu do informacji o lokalizacji osób niewidomych i ich aktualnych zadaniach.
* Geolokalizacja i nawigacja: wolontariusze mogą potrzebować wsparcia w nawigacji, aby szybko dotrzeć do użytkowników i pomóc im.

Rozwój mobilnej aplikacji "Głos" pozwoli zaspokoić te potrzeby i zapewnić wsparcie osobom niewidomym oraz umożliwić wolontariuszom efektywną pomoc tej grupie użytkowników

## Oczekiwania grupy docelowej

Oczekiwania grupy docelowej przy korzystaniu z aplikacji mogą być zróżnicowane, ale ogólnie dotyczą one dostępności i poprawy codziennego życia. Oto niektóre z oczekiwań obu grup:

## Oczekiwania osób niewidomych

* Dostęp do informacji: oczekuje się, że aplikacja zapewni możliwość, z pomocą wolontariuszy, uzyskania dostępu do różnych informacji, takich jak wiadomości, książki i inne zasoby, dzięki czemu będą lepiej poinformowani.
* Nawigacja i orientacja: osoby niewidome oczekują, że aplikacja pomoże im łatwiej poruszać się po świecie dzięki pomocy wolontariuszy.
* Aktywność społeczna: oczekuje się, że aplikacja pomoże im pozostać w kontakcie z wolontariuszami i aktywnie się z nimi komunikować.

## Oczekiwania wolontariuszy

* Środki komunikacji: wolontariusze oczekują, że aplikacja zapewni im niezawodne narzędzia komunikacji z osobami niewidomymi, umożliwiając pomoc w czasie rzeczywistym.
* Dostęp do informacji: oczekuje się, że będą mieli dostęp do informacji o lokalizacji i bieżących zadaniach użytkowników niewidomych, co poprawi ich zdolność do udzielania pomocy.
* Geolokacja i nawigacja: wolontariusze oczekują, że aplikacja ułatwi im znalezienie niewidomego użytkownika i dostarczenie niezbędnej pomocy.

Te oczekiwania są związane z tym, że aplikacja ma być narzędziem, które poprawi jakość życia i umożliwi bardziej efektywną interakcję dla obu grup użytkowników.

## Cel projektu

Celem projektu jest stworzenie aplikacji mobilnej, która umożliwi osobom niewidomym dostęp do informacji i codziennych zadań poprzez komunikację z wolontariuszami. Aplikacja zostanie zaprojektowana z myślą o użytkownikach niewidomych, umożliwiając im korzystanie z funkcji głosu tekstowego TalkBack, komunikacji wideo oraz udostępniania swojej lokalizacji. Jednocześnie aplikacja dostarczy wolontariuszom możliwość pomocy, oferując dodatkowe funkcje, takie jak dostęp do komunikacji wideo i map, aby w przypadku nagłej potrzeby wolontariusz mógł zlokalizować osobę niewidomą i pomóc zarówno w sytuacjach awaryjnych, jak i w codziennych zadaniach.

# Wprowadzenie teoretyczne do narzędzi oraz zagadnień

Aplikacja - to jest program komputerowy, który wykonuje określone zadania na urządzeniach mobilnych lub komputerach.

API - to jest Interfejs Programowania Aplikacji, który umożliwia komunikację między różnymi aplikacjami lub serwisami.

Android - to jest system operacyjny mobilny stworzony przez Google, używany na wielu smartfonach i tabletach.

Baza danych - to jest struktura danych, która umożliwia przechowywanie, organizowanie i zarządzanie informacjami.

Biblioteka programistyczna - to jest zestaw gotowych funkcji i narzędzi, które pomagają programistom w tworzeniu aplikacji.

FCM - to jest Firebase Cloud Messaging, usługa umożliwiająca przesyłanie wiadomości do urządzeń mobilnych.

FireBase - to jest platforma do budowy aplikacji mobilnych i webowych, oferująca różne usługi, w tym bazę danych w czasie rzeczywistym.

Framework - to jest zestaw narzędzi i bibliotek programistycznych, które ułatwiają rozwijanie oprogramowania.

Google - to jest znana firma technologiczna oferująca różne usługi.

GPS - to jest Globalny System Pozycjonowania, który umożliwia określenie geograficznego położenia urządzenia.

GUI - to jest Graficzny Interfejs Użytkownika, który umożliwia interakcję z komputerem za pomocą elementów graficznych, takich jak przyciski i menu.

HTTP - to jest Hypertext Transfer Protocol, protokół używany do przesyłania danych w sieci internet.

IP - to jest Internet Protocol, który jest protokołem komunikacyjnym używanym w sieciach komputerowych.

Java - to jest popularny język programowania używany do tworzenia różnych aplikacji.

JSON - to jest format danych, który jest lekki i czytelny dla człowieka, często używany w wymianie danych w aplikacjach internetowych.

NAT - to jest Network Address Translation, która umożliwia wiele urządzeń w sieci domowej korzystać z jednego publicznego adresu IP.

NoSQL - to jest rodzaj bazy danych, który nie jest oparty na modelu relacyjnym, co pozwala na elastyczne przechowywanie danych.

P2P - to jest Peer-to-Peer, model komunikacji, w którym urządzenia komunikują się bez konieczności korzystania z centralnego serwera.

RTP - to jest Real-time Transport Protocol, używane do przesyłania strumieni multimediów w czasie rzeczywistym.

SDK - to jest zestaw narzędzi deweloperskich (Software Development Kit), który pomaga programistom w tworzeniu aplikacji na danej platformie.

Smartfon - to jest bardziej zaawansowane urządzenie, które łączy w sobie funkcje telefonu oraz komputera.

SQL - to jest Structured Query Language, używane do zarządzania bazami danych relacyjnych.

STUN Serwer - to jest Serwer oznaczenia translacji, używany w komunikacji w czasie rzeczywistym, aby określić publiczny adres IP urządzenia.

Serwer - to jest komputer lub urządzenie, które dostarcza usługi lub dane innym urządzeniom, np. klientom.

TalkBack - to jest funkcja stworzona przez Google przeznaczona dla osób niewidomych i niedowidzących. Jest czytnikiem ekranowym działającym na systemie Android, który przekształca treści wyświetlane na ekranie urządzenia w mowę lub wibracje, umożliwiając użytkownikom niewidomym lub niedowidzącym korzystanie z urządzenia i interakcję z aplikacjami.

Telefon - to jest urządzenie komunikacyjne służące do nawiązywania rozmów telefonicznych i przesyłania wiadomości głosowych.

TCP - to jest Transmission Control Protocol, protokół warstwy transportu używany w sieciach komputerowych.

TURN Serwer - to jest Serwer oznaczenia translacji, który pozwala urządzeniom w sieciach prywatnych komunikować się z urządzeniami w sieciach publicznych.

UDP - to jest User Datagram Protocol, inny protokół warstwy transportu, który jest bardziej "bezpołączeniowy" niż TCP i jest często używany w aplikacjach wymagających szybkiego przesyłania danych.

WebRTC - to jest Real-Time Communication na platformie webowej, który umożliwia przesyłanie strumieni audio i wideo w czasie rzeczywistym przez przeglądarki internetowe.

WEB - to jest skrót od "World Wide Web", czyli globalna sieć internetowa.

WCAG – to są międzynarodowe wytyczne określające standardy i zasady, które mają na celu zapewnienie dostępności i użyteczności treści internetowych dla osób z różnymi rodzajami niepełnosprawności, takich jak niewidomość, niedowidzenie i trudności w dostępie.

# Projekt aplikacji

Ten Rozdział jest poświęcony projektowaniu i planowaniu rozwoju aplikacji mobilnej oraz przedstawieniu ogólnego przeglądu i strategii rozwoju aplikacji.



## Zakres funkcjonalności

Funkcjonalność aplikacji została zaprojektowana z myślą o niewidomych użytkownikach i wolontariuszach i obejmuje następujące główne cechy:

* Funkcja odczytu tekstu: aplikacja zapewnia możliwość odczytu tekstu na ekranie za pomocą funkcji TalkBack, która pozwala użytkownikom słuchać tekstu, takiego jak wiadomości, insturkcje i inne informacje na ekranie spełniające normę WCAG.
* Geolokalizacja: użytkownicy mogą udostępniać swoją lokalizację wolontariuszom, co ułatwia szybkie znalezienie i udzielenie pomocy w razie potrzeby.
* Komunikacja w czasie rzeczywistym: aplikacja zapewnia środki komunikacji w czasie rzeczywistym między użytkownikami niewidomymi a wolontariuszami. Obejmuje to rozmowy głosowe i wideo.
* Szybka rejestracja: aplikacja pozwala uniknąć długich rejestracji za pomocą identyfikatora urządzenia.
* Możliwość wyboru grupy: aplikacja pozwala użytkownikom wybrać grupę do późniejszego wykorzystania aplikacji.
* Prośby o pomoc od niewidomych: niewidomi użytkownicy mają możliwość składania prośby o pomoc poprzez aplikację. Mogą określić rodzaj wsparcia, jakiego potrzebują, takie jak odczytanie tekstu, zakupy czy asystowanie w nawigacji.
* Automatyczne tworzenie połączenia wideo: aplikacja automatycznie tworzą połączenie wideo między niewidomym użytkownikiem a wolontariuszem. Dostęp do kamery niewidomego pozwala wolontariuszowi na dokładniejsze zrozumienie sytuacji i dostarczenie bardziej adekwatnej pomocy.
* Akceptowanie lub odrzucanie prośby o pomoc przez wolontariuszy: wolontariusze mają możliwość akceptowania lub odrzucania prośby o pomoc w zależności od swojej dostępności i umiejętności. To daje im kontrolę nad tym, kiedy i w jaki sposób chcą pomagać.
* Lokalizowanie osoby niewidomej przy akceptacji prośby o pomoc: po akceptacji prośby o pomoc przez wolontariusza, aplikacja umożliwia im zlokalizowanie niewidomej osoby, co ułatwia znalezienie jej w realnym środowisku i zapewnienie wsparcia.

Ten zakres funkcjonalności aplikacji został zaprojektowany w celu zaspokojenia potrzeb i oczekiwań docelowych odbiorców, zapewniając im narzędzia do poprawy ich interakcji ze światem zewnętrznym i wolontariuszami.

## Diagram przypadków użycia

Stosujemy diagram przypadków użycia w celu wizualizacji i modelowania interakcji między systemem a użytkownikami. To efektywny sposób przedstawiania funkcjonalnych wymagań i scenariuszy użycia, co pomaga lepiej zrozumieć i opisać funkcjonalność aplikacji.



Rysunek 1. Diagram przypadków użycia.

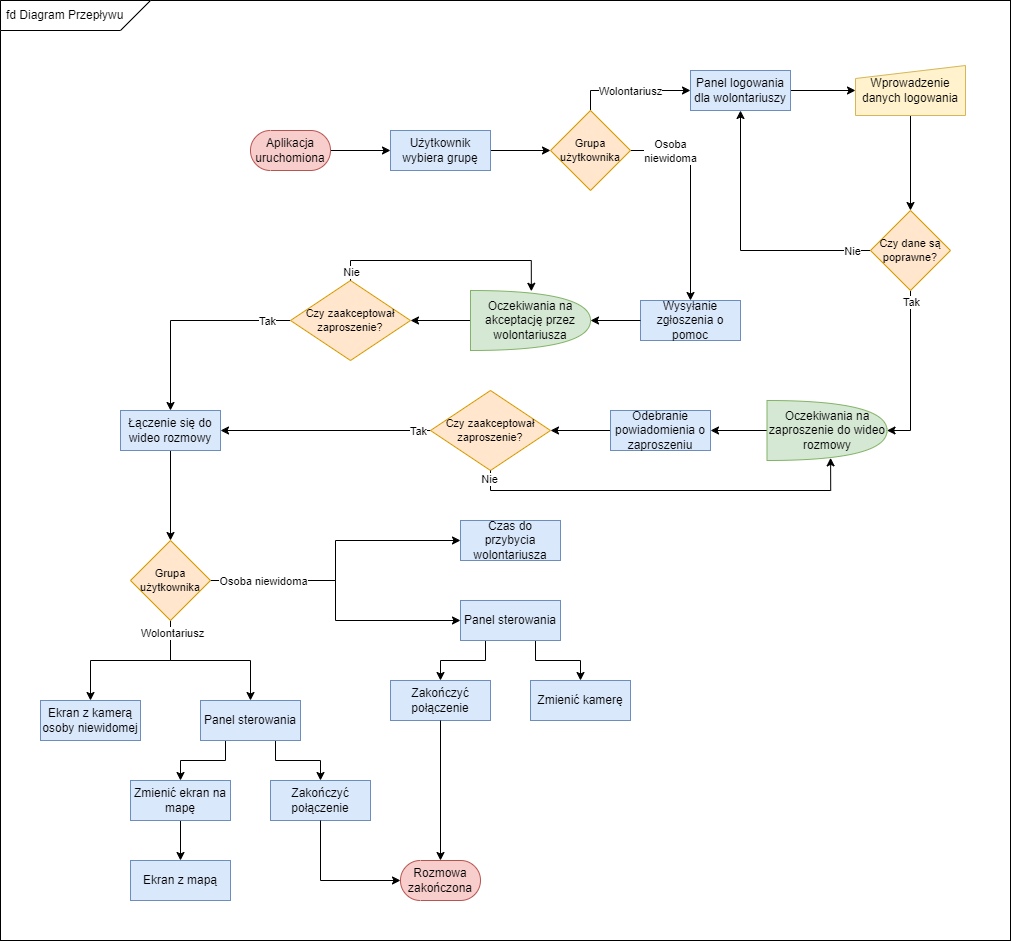
Źródło: Opracowanie własne.

Ten diagram przypadków użycia przedstawia interakcję między osobą niewidomą a wolontariuszem.

* Aktorzy:
  + Osoba niewidoma
  + Wolontariusz
  + System
* Przypadki użycia:
  + Wybór grupy – Użytkownik wybiera grupę w zależności od tego, czy jest wolontariuszem czy osobą niewidomą. To jest punkt początkowy, gdzie użytkownik definiuje swoją rolę.
  + Wezwanie pomocy - osoba niewidoma wysyła zgłoszenie o pomoc do najbliższych wolontariuszów.
  + Oczekiwanie na akceptację połączenia - osoba niewidoma czeka na odpowiedź wolontariusza, który może zaakceptować lub odrzucić prośbę o pomoc.
  + Udostępnienie lokalizacji - Osoba Niewidoma udostępnia swoją lokalizację wolontariuszowi, co umożliwia skuteczniejszą pomoc w nawigacj.
  + Oczekiwanie na zaproszenie połączenia z prośbą o pomoc – wolontariusz po wybraniu grupy czeka aż osoba niewidoma wyśle zgłoszenie z prośba o pomoc.
  + Łączenie się z rozmową wideo - Po akceptacji połączenia, osoba niewidoma i wolontariusz łączą się w rozmowie wideo, korzystając z kamery i mikrofonu oraz udostępnionej lokalizacją.
  + Przegląd wyznaczonej trasy - wolontariusz przegląda wyznaczoną trasę do osoby niewidomej, udzielając wskazówek dotyczących nawigacji i orientacji.
  + Zakończenie połączenia - Po zakończeniu interakcji, zarówno osoba niewidoma, jak i wolontariusz, mogą zakończyć połączenie, kończąc tym samym sesję pomocy.

## Diagram przypływu

Stosujemy diagram przypływu do wizualizacji i modelowania sekwencji działań w systemie. To narzędzie pomaga klarownie przedstawić przepływy kontroli i aktywności w procesach, co poprawia zrozumienie i projektowanie różnych aspektów funkcjonalności aplikacji.



Rysunek 2. Diagram przepływu.

Źródło: Opracowanie własne.

Opis tego diagramu:

* Wybór Grupy: użytkownik rozpoczyna proces, wybierając grupę, do której chce należeć. To może być punkt początkowy, gdzie definiuje swoje potrzeby i preferencje.
* Zgłoszenie Pomocy (dla Osób Niewidomych): jeśli użytkownik jest osobą niewidomą, ma opcję zgłoszenia potrzeby pomocy. Ta opcja jest dostępna po dokonaniu wyboru grupy. Zgłoszenie to może obejmować różne kategorie wsparcia, takie jak nawigacja, wyszukiwanie informacji czy rozwiązywanie problemów domowych.
* Oczekiwanie Wolontariusza: wolontariusz, zarejestrowany w systemie, oczekuje na zgłoszenia od osób niewidomych. Ten krok odzwierciedla gotowość wolontariuszy do udzielenia pomocy.
* Powiadomienie Wolontariusza: gdy zgłoszenie osoby niewidomej zostaje wysłane, system automatycznie powiadamia dostępnych wolontariuszy o potrzebie pomocy. To sprawia, że wolontariusz jest świadomy istnienia zgłoszenia.
* Akceptacja/Odrzucenie Zapytania: wolontariusz ma możliwość akceptacji lub odrzucenia zgłoszenia. Jeśli wolontariusz akceptuje, przechodzi do następnego etapu, czyli wideo rozmowy. Jeśli odrzuca, zgłoszenie pozostaje aktywne, czekając na akceptację innego wolontariusza.
* Wideo Rozmowa: w przypadku akceptacji zapytania, inicjowana jest wideo rozmowa między wolontariuszem a osobą niewidomą. W trakcie tej rozmowy osoba niewidoma może korzystać z kamery, mikrofonu i lokalizacji, aby skutecznie komunikować się z wolontariuszem.
* Sterowanie rozmową: W trakcie rozmowy użytkownicy mają możliwość sterować rozmową poprzez panel sterowania który znajduję się w dolnej części ekranu. Wolontariusz ma możliwość przełączenia swojego ekranu na mapę, żeby dostać informacje o lokalizacji osoby niewidomej. Niewidomy może wybierac z której kamery przekazywać obraz (tylny/przedni aparat).
* Udzielanie Pomocy: wolontariusz może udzielać pomocy w różnych sytuacjach, dostosowując się do potrzeb osoby niewidomej. To może obejmować pomoc w nawigacji, wyszukiwaniu informacji czy rozwiązywaniu codziennych problemów.
* Zakończenie Rozmowy: każdy z użytkowników aplikacji może w dowolnym momencie zakończyć połączenie.

## Ekrany aplikacji

Ten podrozdział będzie poświęcony opisowi interfejsu mobilnej aplikacji. W nim zostaną omówione struktura ekranów oraz ich wizualne uformowanie.

## Cechy projektowania ekranów dla osób niewidomych

Projektowanie ekranów aplikacji dla osób niewidomych to istotny i wrażliwy proces. W centrum tego projektowania leży zapewnienie maksymalnej dostępności i wygody użytkowania dla osób z ograniczonym lub całkowicie brakującym wzrokiem. Oto główne zasady przestrzegane podczas projektowania aplikacji:

* Cechy interfejsu: interfejs aplikacji dla osób niedowidzących powinien być "niewidoczny" i służyć tylko do wykonania zadania.
* Kontrast kolorów: zgodnie z międzynarodowym standardem WCAG kontrast kolorów i tła w interfejsach dla osób niewidomych powinien wynosić co najmniej 4.5:1, A kolor czcionki 3:1.
* Rozmiar czcionki: rozmiar czcionki powinien wynosić co najmniej 14 pt pogrubiony i od 18 pt normalny, a lepiej użyć skalowalnego punktu (SP).
* Elementy dekoracyjne: elementy dekoracyjne i ilustracje nie muszą spełniać wymagań dotyczących kontrastu, ich głównym zadaniem jest być prostymi i zrozumiałymi.

Oto nasze technologiczne rozwiązania, do których doszliśmy w wyniku projektowania ekranów aplikacji:

* Optymalizacja tekstów: z uwagi na korzystanie przez osoby niewidome z funkcji TalkBack, nasze teksty muszą być krótkie, zrozumiałe i przydatne. Głosowa obsługa interfejsu powinna ułatwiać nawigację, a nie sprawiać trudności.
* Opisy tekstowe przycisków i ikon: każdy przycisk i klikalna ikona wymaga opisu tekstowego, aby programy dostępu do ekranu mogły wyjaśnić użytkownikowi, co robią.
* Hierarchia w interfejsie: ekrany aplikacji dzielimy na 3 części, aby korzystanie z nich było zrozumiałe dla osób niewidomych.
  1. informacje o ekranie.
  2. obszar funkcjonalny.
  3. obszar odpowiedzialny za zarządzanie tym ekranem.
* Odrzucenie zbędnych dźwięków: niewidomy użytkownik rzadko patrzy na aplikację, on ją słucha. Dlatego unikamy zbędnych dźwięków i powiadomień dla tego użytkownika."

W sumie, przestrzeganie tych zasad i wprowadzenie rozwiązań projektowych z myślą o specyficznych potrzebach niewidomych w ramach kompleksowego podejścia doprowadzi do stworzenia prostych i intuicyjnie zrozumiałych aplikacji dedykowanych tej grupie użytkowników.

## Ekran startowy

Ten ekran aplikacji zawiera:

* Tytuł ekranu: krótki opis tej strony.
* Obraz: minimalistyczny obraz smartfona.
* Przycisk akcji: ten przycisk jest odpowiedzialny za przejście do następnego ekranu aplikacji.



Rysunek 3. Ekran startowy.

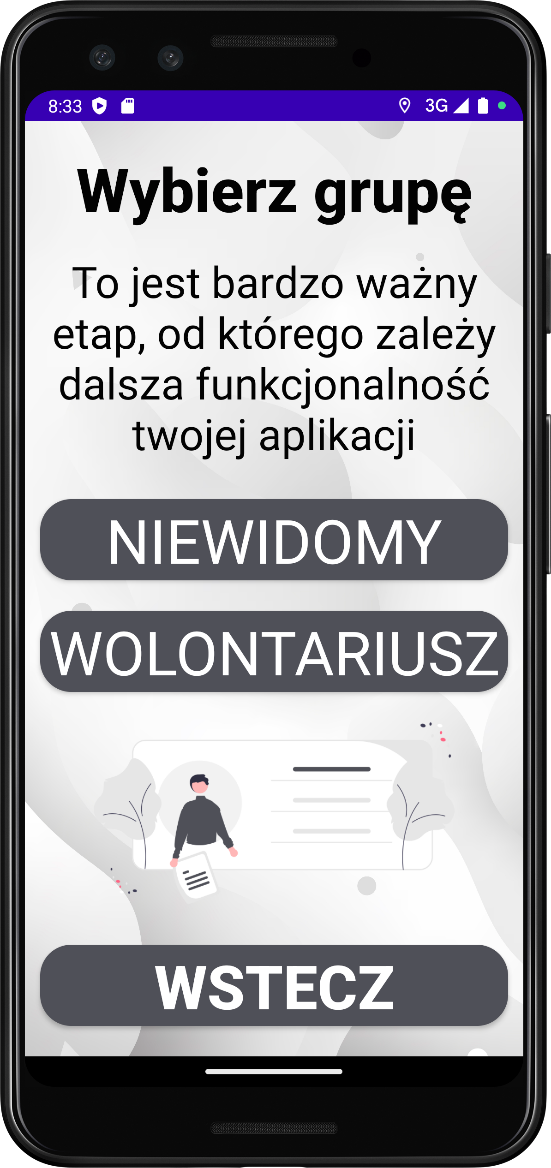
Źródło: Opracowanie własne.



## Ekran wyboru grupy

Ten ekran aplikacji zawiera:

* Tytuł ekranu: krótki opis tej strony.
* 2 przyciski akcji: te przyciski są odpowiedzialne za wybór grupy użytkownika.
* Przycisk “Wstecz”: odpowiedzialny za powrót do poprzedniego ekranu.



Rysunek 4. Ekran wyboru grupy.

Źródło: Opracowanie własne.



## Ekran grupy

Ten ekran aplikacji zmienia się w zależności od wybranej grupy.

* Grupa"Wolontariusz":
  + Tytuł ekranu: krótki opis ekranu.
  + Formularz wprowadzenia danych logowania.
  + Przycisk “Zaloguj się”: odpowiedzialny za przejście do następnego ekranu w razie wprowadzenia poprawnych danych logowania.
  + Przycisk “Wstecz”: odpowiedzialny za powrót do poprzedniego ekranu.



Rysunek 5. Ekran grupy "Wolontariusz", Panel logowania.

Źródło: Opracowanie własne.

Po zalogowaniu Wolontariusz zostaje aktywnym i zaczyna dostawać wezwania pomocy od osób niewidomych:

* + Tytuł ekranu: krótki opis ekranu.
  + Przycisk “Wstecz”: odpowiedzialny za powrót do poprzedniego ekranu.

Obraz zawierający tekst, Telefon komórkowy, zrzut ekranu, Urządzenie przenośne

Opis wygenerowany automatycznie

Rysunek 6. Ekran grupy "Wolontariusz", Ekran Oczekiwania.

Źródło: Opracowanie własne.

Jeśli wolontariusz otrzyma wezwanie pomocy, ukryte elementy, które zawiera, zostaną aktywowane na tym ekranie:

* + Tytuł wyzwania: krótki opis tego, z jakiego powodu powstał ten blok.
  + 2 przyciski akcji: przyciski te są odpowiedzialne za akceptację i odmowę wezwania pomocy.



Rysunek 7. Ekran grupy "Wolontariusz", prośba o pomoc.

Źródło: Opracowanie własne.

Jednocześnie z tym wolontariusz dostaje PUSH powiadomienie, które informuje go o otrzymaniu nowej prośby o pomoc.

Obraz zawierający tekst, Telefon komórkowy, zrzut ekranu, gadżet

Opis wygenerowany automatycznie

Rysunek 8. PUSH powiadomienie.

Źródło: Opracowanie własne

Po zaakceptowaniu prośby o pomoc wolontariusz dostaje Toast powiadomienie, które informuje go, że trwa proces nawiązywanie połączenia się do rozmowy.

Obraz zawierający tekst, Telefon komórkowy, Urządzenie przenośne, zrzut ekranu

Opis wygenerowany automatycznie

Rysunek 9. Toast powiadomienie.

Źródło: Opracowanie własne

* Grupa "Osoba niewidoma":
  + Tytuł ekranu: krótki opis tej strony.
  + Przycisk akcji "Znajdź pomoc": ten przycisk jest odpowiedzialny za rozpoczęcie poszukiwania aktywnego wolontariusza do pomocy.
  + Przycisk “Wstecz”: odpowiedzialny za powrót do poprzedniego ekranu.



Rysunek 10. Ekran grupy "Osoba niewidoma".

Źródło: Opracowanie własne.

Jeżeli niewidomy już nie będzie chciał pomocy to może odwołać swoją prośbę:

* + Tytuł ekranu: krótki opis tej ekranu.
  + Przycisk akcji "Anuluj": ten przycisk jest odpowiedzialny za anulowanie poszukiwanie aktywnego wolontariusza.
  + Przycisk “Wstecz”: odpowiedzialny za powrót do poprzedniego ekranu.

Obraz zawierający tekst, Telefon komórkowy, zrzut ekranu, multimedia

Opis wygenerowany automatycznie

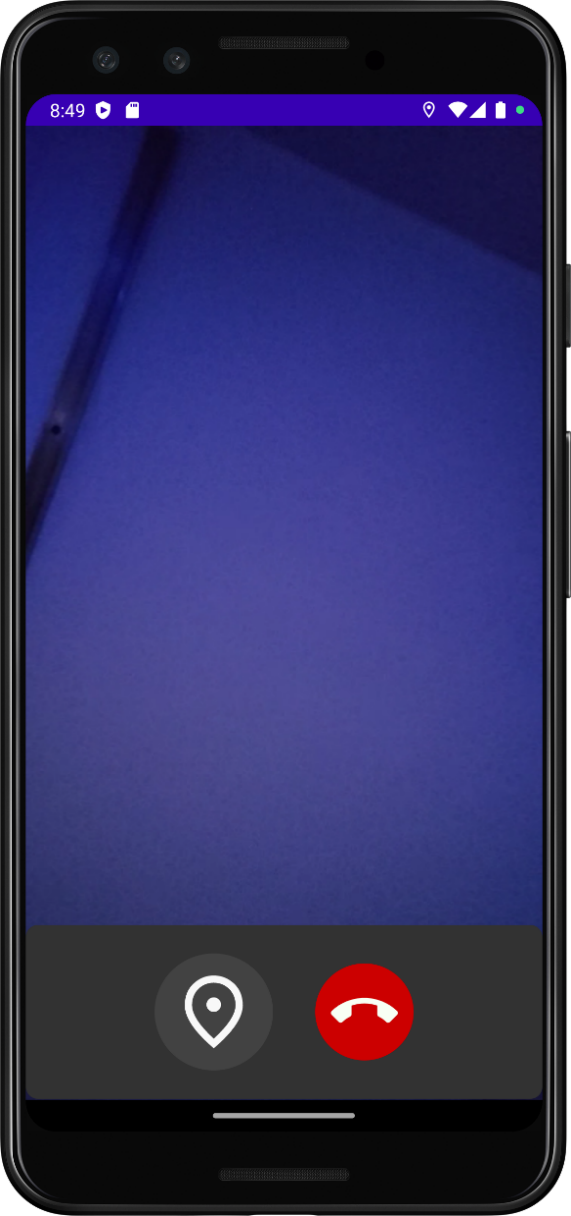
Rysunek 11. Ekran grupy "Osoba niewidoma", anulowanie.

Źródło: Opracowanie własne.

## Ekran połączenia

Ten ekran aplikacji zmienia się w zależności od wybranej grupy.

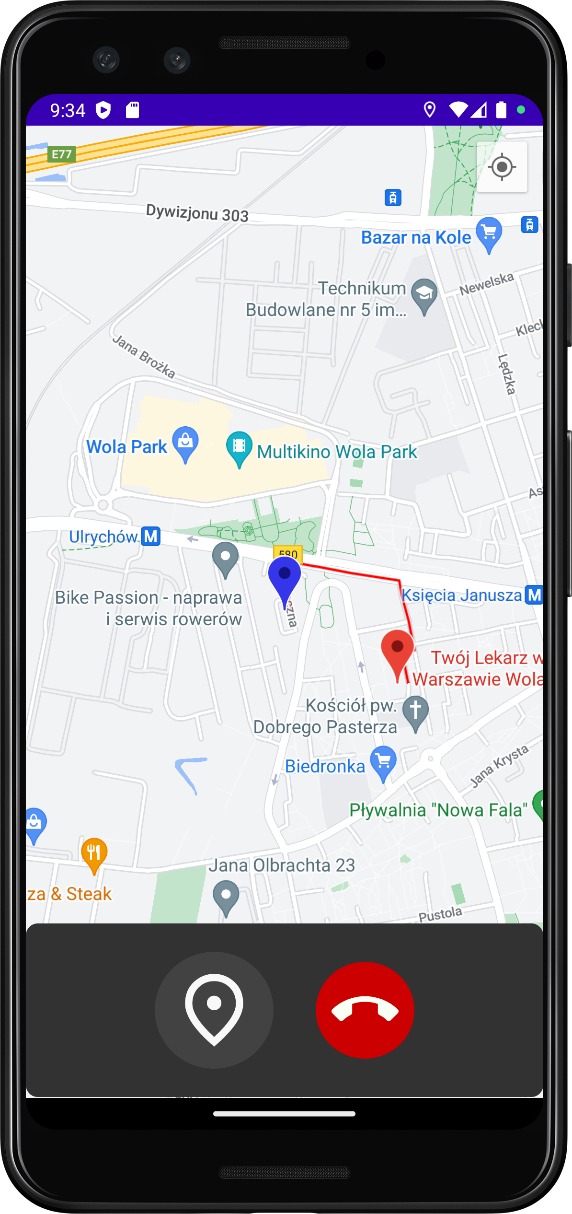
* Grupa"Wolontariusz":
  + Duży ekran: ekran z obrazem z kamery osoby niewidomej.
  + Panel sterowania z dwoma przyciskami: które są odpowiedzialne za mapę i koniec połączenia.



Rysunek 12. Ekran połączenia "Wolontariusz".

Źródło: Opracowanie własne.

Jeśli wolontariusz naciśnie przycisk mapy, otworzy mu się ekran z mapą, na której będzie widoczna jego pozycja i ścieżka do osoby niewidomej.



Rysunek 13. Ekran połączenia "Wolontariusz", mapa.

Źródło: Opracowanie własne.

* Grupa "Osoba niewidoma":
  + Tytuł ekranu: krótki opis tego ekranu.
  + Informacja: Informacja wskazująca, ile wolontariusz potrzebuje czasu, żeby dotrzeć do osoby niewidomej.
  + 2 aktywne przyciski: które są odpowiedzialne za odwrócenie kamery i koniec połączenia.



Rysunek 14. Ekran połączenia "Osoba niewidoma”

Źródło: Opracowanie własne.

## Usługi i technologie wykorzystane w projekcie

Niniejszy podrozdział będzie poświęcony opisowi wyboru konkretnych technologii i powodów ich wyboru.

## Język programowania

Wybór języka programowania Java do tworzenia mobilnej aplikacji "Głos" jest uzasadniony kilkoma czynnikami. Po pierwsze, Java zapewnia platformową niezależność, co umożliwia aplikacji funkcjonowanie na różnych mobilnych urządzeniach pod kontrolą systemu Android. Po drugie, bogaty zestaw standardowych bibliotek w języku Java ułatwia proces programowania, zwiększając efektywność kodowania.

Odrzucenie użycia zewnętrznych frameworków wynika z dążenia do osiągnięcia maksymalnej elastyczności i optymalizacji w procesie tworzenia mobilnej aplikacji "Głos", a także potrzeby pełniejszej kontroli nad aplikacją.

## Baza danych

Wybraliśmy Firebase Realtime Database z wielu powodów:

* Rzeczywisty czas**:** Firebase Realtime Database zapewnia synchronizację danych w czasie rzeczywistym, co jest kluczowe dla aplikacji skierowanej na dostarczanie aktualnych informacji dla niewidomych użytkowników i wolontariuszy.
* Prosta integracja**:** Firebase oferuje wygodny i łatwy w użyciu interfejs do pracy z bazą danych, co ułatwia proces programowania i pozwala skupić się na kluczowych funkcjach aplikacji.
* Rozległy stos usług**:** Firebase oferuje dodatkowe usługi, takie jak przechowywanie plików w chmurze lub autoryzacja.

Również porównanie Firebase Realtime Database z konkurentami, wyjaśniające wybór:

|  |  |  |  |
| --- | --- | --- | --- |
| Funkcjonalność | Firebase Realtime Database | Firestore (Google Cloud) | MongoDB Atlas |
| Synchronizacja w czasie rzeczywistym | Tak | Tak | Tak |
| Łatwość użytkowania | Wysoka | Średnia | Wysoka |
| Ogólne możliwości bazy danych | Tak | Tak | Tak |
| Integracja z innymi usługami Firebase | Tak | Nie | Nie |
| Koszty | Bezpłatne (ograniczone) | Płatne | Płatne |

Tabela 1. Porównanie Firebase Realtime Database z konkurencyjnymi usługami.

Źródło: Opracowanie własne.

W bazie danych przechowujemy dane o dacie i godzinie udzielenia pomocy, ID wolontariusza oraz ID niewidomego. Tych danych można użyć do tworzenia statystyk lub sprawdzenia, który wolontariusz udzielał pomocy osobie niewidomej w wybranym okresie.

Obraz zawierający tekst, paragon, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

Rysunek 15. Dane przechowywane w Firebase Realtime Database.

Źródło: Opracowanie własne.

Baza danych jest również niezbędna do przesyłania lokalizacji użytkowników aplikacji podczas połączenia w rozmowie wideo.

Obraz zawierający tekst, zrzut ekranu, Czcionka, dokument

Opis wygenerowany automatycznie

Rysunek 16. Dane tymczasowe w Firebase Realtime Database.

Źródło: Opracowanie własne.

## Mapa

Wybór API Google Maps umożliwia korzystanie z technologii w dziedzinie kartografii i nawigacji, co zapewnia wysoką funkcjonalność i wygodę użytkowania dla docelowej grupy odbiorców. Oto kilka czynników uzasadniających ten wybór:

* Rozległy zakres funkcji: API Google Maps oferuje szeroki zakres możliwości pracy z mapami, obejmujący wyświetlanie lokalizacji, tras, miejsc i innych istotnych elementów.
* Wiarygodność i stabilność: API Google Maps jest powszechnie używane na całym świecie i znane ze swojej stabilności i niezawodności. Jest to istotne dla zapewnienia ciągłej pracy aplikacji i dokładnego wyświetlania danych na mapie.
* Prosta integracja: API Google Maps dostarcza prosty i wygodny interfejs do integracji z aplikacją. Ułatwia to proces programowania i umożliwia szybką integrację funkcji map w aplikacji.
* Aktualizacje i wsparcie: API Google Maps regularnie jest aktualizowane i utrzymywane.

## Komunikacja w czasie rzeczywistym

Wybór technologii WebRTC dla projektu "Głos" jest uzasadniony kilkoma istotnymi czynnikami. Przede wszystkim WebRTC zapewnia niezawodne i efektywne narzędzia do realizacji komunikacji audio-wideo w czasie rzeczywistym przez internet. Jest to istotne dla zapewnienia płynnej komunikacji między niewidomymi użytkownikami a wolontariuszami, co przyczynia się do bardziej dostępnego i inkluzywnego doświadczenia.

WebRTC wyróżnia się także wysoką zgodnością z platformą Android, zapewniając efektywną integrację z mobilną aplikacją. Jego otwarte standardy i wygodne API sprawiają, że jest preferowanym wyborem do implementacji funkcji komunikacyjnych w projekcie.

Poniżej przedstawiono tabelę porównawczą wyboru WebRTC z rzeczywistymi konkurentami:

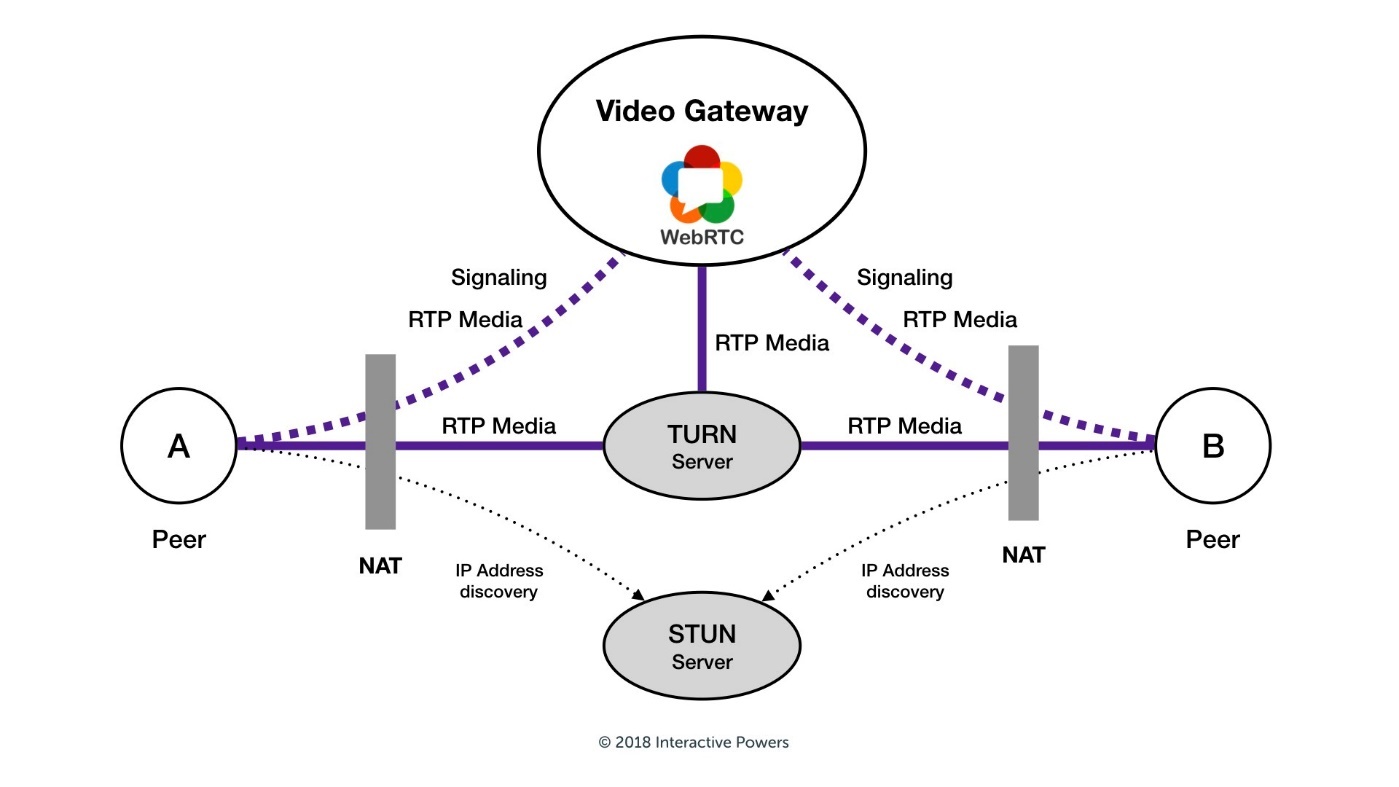
|  |  |  |  |
| --- | --- | --- | --- |
| Kryterium | WebRTC | Daily | Agora |
| Niezawodność | Wysoka | Średnia | Wysoka |
| Łatwość integracji | Wygodna | Wygodna | Wygodna |
| Jakość audio/wideo | Tak | Nie | Nie |
| Koszty | Bezpłatne | Płatne | Płatne |

Tabela 2. Porównanie WebRTC z konkurencyjnymi usługami.

Źródło: Opracowanie własne.

Wybór WebRTC podkreśla jego doskonałość w zapewnianiu wysokiej jakości komunikacji i odpowiada wymaganiom projektu "Głos".

Zapewniamy również schemat działania WebRTC dla lepszego zrozumienia:



Rysunek 17. Jak nawiązuje się połączenie WebRTC.

Źródło: <https://docs.freedomrobotics.ai/docs/how-webrtc-works>.

## Bezpieczeństwo aplikacji

Ten podrozdział jest poświęcony bezpieczeństwu aplikacji. Rozważmy kluczowe aspekty, mechanizmy i procedury zabezpieczające, które zostały wdrożone w celu ochrony prywatności i integralności danych użytkowników.

Google Maps API: Google Maps umożliwiają korzystanie z wewnętrznego protokołu HTTPS do blokowania za pomocą protokołu HTTP, który umożliwia szyfrowanie, które umożliwia szybkie przejście do serwera internetowego w celu zapewnienia bezpiecznego przesyłania informacji.

Firebase Realtime Database: wszystkie dane przesyłane między aplikacją a bazą danych Firebase Realtime Database są zabezpieczone protokołem HTTPS, co zapewnia szyfrowanie danych w trakcie transmisji i chroni je przed przechwyceniem.

WebRTC: WebRTC zapewnia szyfrowanie danych poprzez protokoły DTLS (Datagram Transport Layer Security) i SRTP (Secure Real-time Transport Protocol), co chroni integralność i poufność przesyłanych informacji.

Uwierzytelnianie w aplikacji: do uwierzytelnienia użytkownika używamy go Build.ID pozwala to na anonimizację użytkownika i szybkie logowanie do aplikacji, co jest kluczową cechą dla grupy docelowej.

# Spis rysunków

[Rysunek 1. Diagram przypadków użycia. 14](#_Toc155648478)

[Rysunek 2. Diagram przepływu. 17](#_Toc155648479)

[Rysunek 3. Ekran startowy. 22](#_Toc155648480)

[Rysunek 4. Ekran wyboru grupy. 23](#_Toc155648481)

[Rysunek 5. Ekran grupy "Wolontariusz", Panel logowania. 24](#_Toc155648482)

[Rysunek 6. Ekran grupy "Wolontariusz", Ekran Oczekiwania. 25](#_Toc155648483)

[Rysunek 7. Ekran grupy "Wolontariusz", prośba o pomoc. 26](#_Toc155648484)

[Rysunek 8. PUSH powiadomienie. 27](#_Toc155648485)

[Rysunek 9. Toast powiadomienie. 28](#_Toc155648486)

[Rysunek 10. Ekran grupy "Osoba niewidoma". 29](#_Toc155648487)

[Rysunek 11. Ekran grupy "Osoba niewidoma", anulowanie. 30](#_Toc155648488)

[Rysunek 12. Ekran połączenia "Wolontariusz". 31](#_Toc155648489)

[Rysunek 13. Ekran połączenia "Wolontariusz", mapa. 32](#_Toc155648490)

[Rysunek 14. Ekran połączenia "Osoba niewidoma” 33](#_Toc155648491)

[Rysunek 15. Dane przechowywane w Firebase Realtime Database. 36](#_Toc155648492)

[Rysunek 16. Dane tymczasowe w Firebase Realtime Database. 37](#_Toc155648493)

[Rysunek 17. Jak nawiązuje się połączenie WebRTC. 40](#_Toc155648494)

# Spis tabel

[Tabela 1. Porównanie Firebase Realtime Database z konkurencyjnymi usługami 35](#_Toc154696496)

[Tabela 2. Porównanie WebRTC z konkurencyjnymi usługami. 39](#_Toc154696497)