

Programowanie Zespołowe 2016

Wojciech Polak, Klaudia Głocka, Rafał Ziemiński, Konrad Chojnecki

10 listopada 2016

Spis treści

1	Opis aplikacji	2
1.1	Użyte technologie	2
1.2	Zarys architektury aplikacji	2
1.2.1	Moduł Menadżera Bluetooth	2
1.2.2	Asynchroniczność aplikacji	2
1.2.3	Moduł Handler	3
1.2.4	Modele	3
1.2.5	Kontrolery	4
1.2.6	Widoki	4
2	Podział prac	4
2.1	Rafał Ziemiński - moduł BluetoothManager dla systemu Android	4
2.2	Klaudia Głocka - moduł ConnectionHandler	4
2.3	Konrad Chojnecki - moduł MessageHandler	4
2.4	Wojciech Polak - moduł BluetoothManager dla systemu iOS	4
3	Stan prac	4
3.1	Na dzień 01-11-2016	4
3.2	Na dzień 10-11-2016	4

1 Opis aplikacji

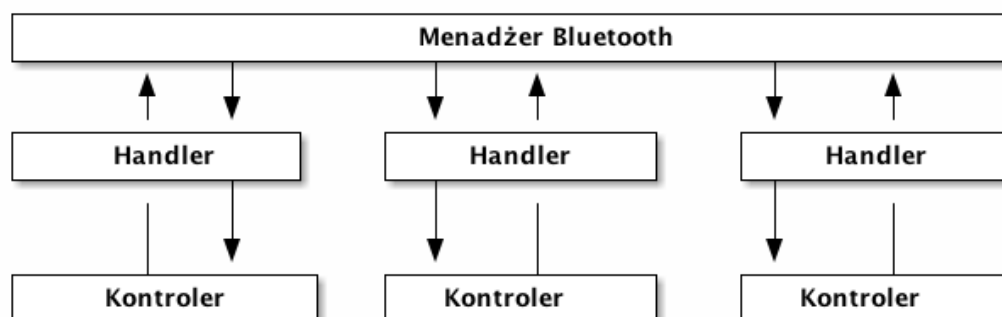
Aplikacja ma za zadanie połączenie dwóch lub więcej urządzeń, poprzez protokół komunikacyjny *Bluetooth*, w celu zapewnienia podstawowej komunikacji w postaci wiadomości tekstowych.

1.1 Użyte technologie

Do stworzenia aplikacji użyto zestawu narzędzi firmy *Xamarin*. Kod źródłowy aplikacji został napisany głównie w języku C#. Użyte biblioteki i oprogramowanie pozwoliło współdzielić kod dla wielu platform, tym samym umożliwiając szybki rozwój aplikacji zarówno na system **iOS** jak i **Android**.

1.2 Zarys architektury aplikacji

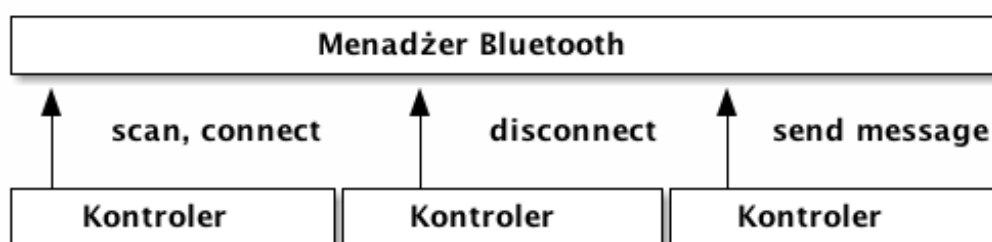
Aplikacja została stworzona w oparciu o wzorec projektowy *Model - View - Controller*.



1.2.1 Moduł Menadżera Bluetooth

Menadżer Bluetooth odpowiada za najniższą warstwę aplikacji. Każdy z systemów - odpowiednio **iOS** jak i **Android** posiadają inne Programistyczne Interfejsy Aplikacji (w skrócie *API*). Najniższa warstwa będąca *najbliższą sprzętu* została rozdzielona pomiędzy systemy. Dlatego ten menadżer został zaimplementowany jako dwa moduły.

Ponieważ wymagane jest aby aplikacja w przyszłości była łatwo rozszerzalna, została wprowadzona warstwa abstrakcji - **IBluetoothManager** która służy jako punkt wyjścia i wejścia dla innych modułów.



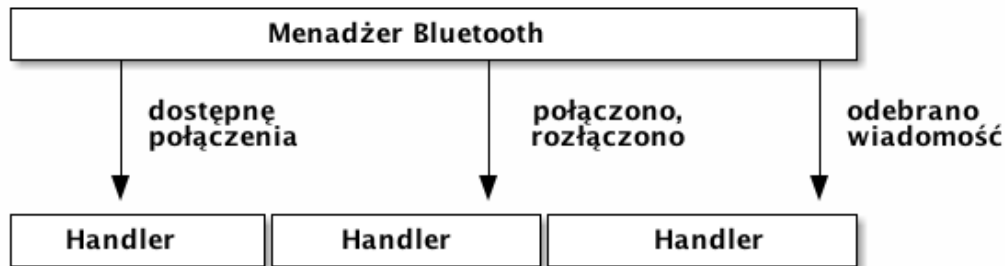
1.2.2 Asynchroniczność aplikacji

Aplikacja jest zależna od zasobów zewnętrznych takich jak sieć *Bluetooth*. W momencie gdy system czeka na nawiązanie połączenia z drugą komórką jest wymagane aby użytkownik cały czas miał aplikację interaktywną i responsywną.

Aby aplikacja nie blokowała interfejsu użytkownika wszystkie akcje wykonywane przez warstwy niższe i pośrednie muszą być **asynchroniczne**. Rozwiązaniem są moduły **Handlera** których zadaniem jest reakcja na wydarzenia.

1.2.3 Moduł Handler

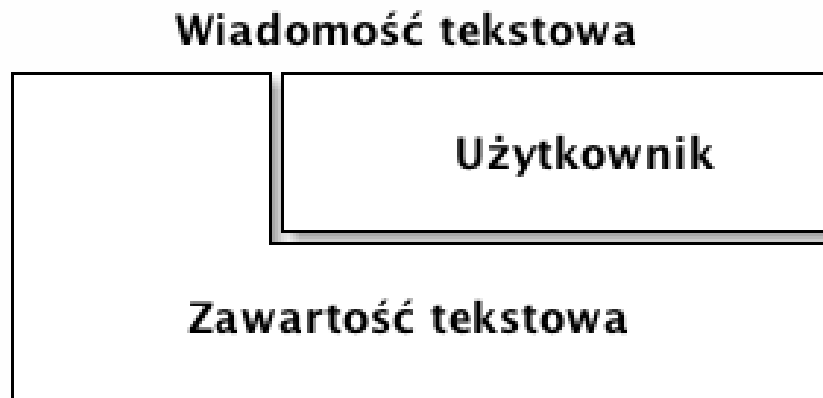
W momencie gdy zadanie zostanie wykonane (np. użytkownik zostanie połączony z innym), wiadomość zostaje wysłana do odpowiedniego **Handler** który reaguje w odpowiedni sposób do danych wejściowych.



W projekcie znajdują się dwa moduły typu **Handler**

1. Moduł **Message Handler** odpowiada za łączenie z innymi użytkownikami. Do zadań tego modułu należą:
 - Reakcja na pobraną listę dostępnych w pobliżu użytkowników - pseudonimy użytkowników są wyświetlane na ekranie.
 - Reakcja na połączenie się z danym użytkownikiem - następuje zmiana widoku na widok wysłanych i odebranych wiadomości.
2. Moduł **Connection Handler** odpowiada za reakcję na wiadomości odebrane od innego użytkownika. Wiadomości takie są wyświetlane w czytelnej formie na ekranie telefonu.

1.2.4 Modele



1. Użytkownik Aplikacja przechowuje informacje o użytkowniku takie jak:
 - pseudonim
 - unikalny identyfikator oparty o technologię **GUID4**
2. Wiadomość Wysyłane i odbierane wiadomości mają format:
 - Użytkownik
 - Wiadomość tekstowa

1.2.5 Kontrolery

Kontrolery odpowiadają za zarządzanie danymi które zostały odebrane przez moduły **Handler**. Często wymagane jest aby dane te zostały odpowiednio spreparowane zanim zostaną wyświetlone na ekranie. Dobrą praktyką jest, aby w dalszych widokach **nie było żadnej logiki biznesowej**. Dlatego każda operacja na danych musi się odbyć w kontrolerze.

Kontrolery są modułami które odbierają wydarzenia (np. naciśnięcie przycisku, wpisanie tekstu, gesty czy ruch zarejestrowany przez akcelerator) które zostały wykonane w odpowiednich widokach. Kontrolery reagują wydarzeniami i na podstawie zawartości wydarzeń przesyłają odpowiednie komendy do pozostałych modułów, najczęściej do Menadżera Bluetooth.

1.2.6 Widoki

Aplikacja składa się z dwóch widoków.

1. Widok z możliwymi połączeniami. W tym widoku użytkownik może zobaczyć wszystkich innych użytkowników, którzy są w zasięgu. Urządzenie skanuje obszar w określonym interwale czasowym. Użytkownik może nawiązać bezpośrednie połączenie z jednym użytkownikiem tym samym przechodząc do widoku drugiego.
2. Widok wymiany wiadomości. W tym widoku użytkownik wysyła i odbiera wiadomości nadane przez drugiego użytkownika. Na raz możliwa jest rozmowa tylko z jednym użytkownikiem. Użytkownik oprócz wysyłania wiadomości może także zakończyć rozmowę tym samym wracając do widoku pierwszego.

2 Podział prac

2.1 Rafał Ziemiński - moduł BluetoothManager dla systemu Android

2.2 Klaudia Głocka - moduł ConnectionHandler

2.3 Konrad Chojnecki - moduł MessageHandler

2.4 Wojciech Polak - moduł BluetoothManager dla systemu iOS

3 Stan prac

3.1 Na dzień 01-11-2016

1. Przerwa w pracy w wyniku dni wolnych od pracy
2. Poprawa dokumentu opisującego projekt. Wykorzystanie w tym celu \LaTeX .
3. Konfiguracja środowisk programistycznych:
 - Próby instalacji IDE, wymaganych bibliotek i narzędzi pracy
 - Konfiguracja maszyn wirtualnych oraz urządzeń natywnych

3.2 Na dzień 10-11-2016

1. Reinstalacja systemu operacyjnego Microsoft Windows na jednym stanowisku pracy, konfiguracja wszystkich potrzebnych bibliotek, narzędzi i edytorów.
2. Usunięcie Visual Studio 2013 na drugim stanowisku pracy. Konfiguracja Visual Studio 2015.
3. Aktualizacja dokumentacji o grafy i wykresy połączeń pomiędzy modułami