**POLITECHNIKA POZNAŃSKA**

**Telefonia IP**

**Projekt komunikatora głosowego**

**Damian Lewandowski**

**Dariusz Bernat**

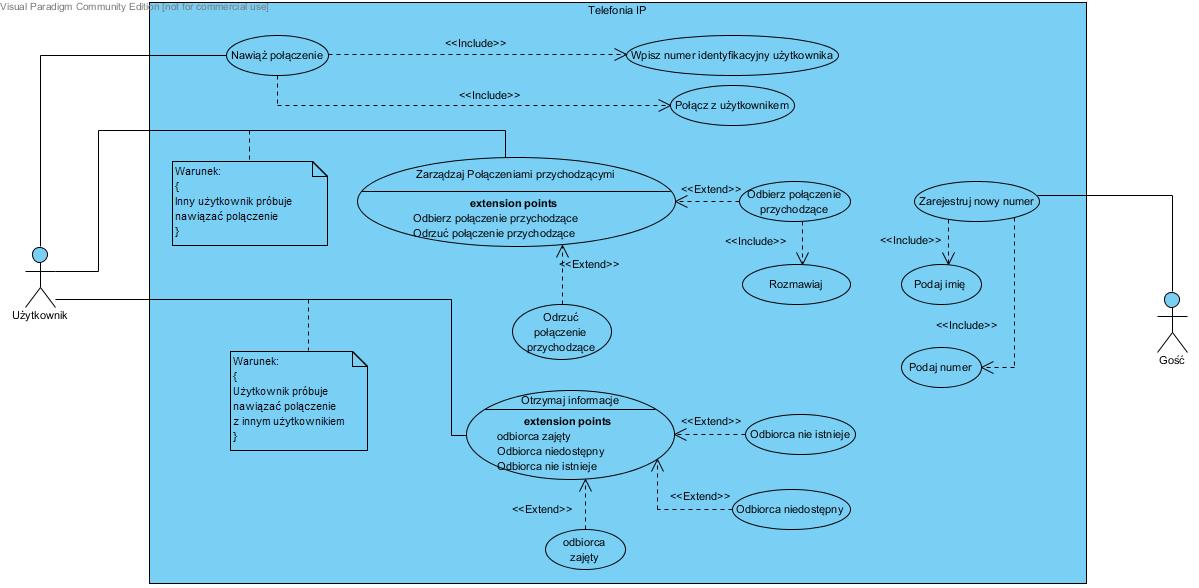
**1. Wstęp:**

Celem tego projektu jest zaprojektowanie oraz implementacja komunikatora głosowego służącego do komunikowania się między użytkownikami przy wykorzystaniu sieci komputerowych opartych o stos protokołów TCP/IP. Komunikator ten poza protokołem sterowania ma również przetwarzać, wysyłać , odtwarzać odbierać dźwięk między użytkownikami.  
  
**2. Zasada działania systemu:**

Aby możliwa była w miarę swobodna komunikacja między użytkownikami przyjęta została zasada działania tego systemu która jest bardzo podobna do tego jak działają obecne telefony komórkowe oraz telefony stacjonarne. W tym celu wydzielone zostały następujące sytuacje:

* Użytkownik A ma zamiar nawiązać połączenie głosowe z użytkownikiem B, jednak użytkownik B nie jest podłączony (zalogowany) do systemu przez co nie będzie możliwości nawiązania połączenia. Użytkownik A otrzymuje zwrotną informację o tym że nie ma możliwości połączenia się z użytkownikiem B z powodu tego że użytkownik ten nie jest podłączony.
* Użytkownik A ma zamiar nawiązać połączenie głosowe z użytkownikiem B, jednak użyty numer będący identyfikatorem tego użytkownika nie istnieje. Użytkownik A analogicznie jak było w poprzednim przykładzie otrzymuje informację zwrotną o tym że nie ma takiego numeru.
* Użytkownik A ma zamiar nawiązać połączenie głosowe z użytkownikiem B, po pewnym czasie oczekiwania połączenie zostaje nawiązane oraz następuje komunikacja między tymi użytkownikami. Po naciśnięciu przycisku zakończenia połączenia z strony użytkownika A lub użytkownika B następuje zakończenie połączenia.
* Użytkownik A ma zamiar nawiązać połączenie głosowe z użytkownikiem B, następuje oczekiwanie na połączenie jednak użytkownik B z jakiegoś powodu odrzuca połączenie. Użytkownik A otrzymuje informację że użytkownik B jest zajęty.
* Użytkownik A ma zamiar nawiązać połączenie głosowe z użytkownikiem B, jednak okazuje się że użytkownik B nawiązał połączenie głosowe z innym użytkownikiem przez co nie ma możliwości nawiązania połączenia. Użytkownik A otrzymuje komunikat o tym że użytkownik B jest zajęty.

Przyjęte założenia przedstawione zostały w postaci poniższego diagramu przypadków użycia:



*il. 1: Diagram przypadków użycia systemu telefonii IP*

Na powyższym diagramie przypadków użycia można zauważyć aktora o nazwie gość. W tym wypadku gościem można nazwać użytkownika który nie posiada jeszcze numeru który umożliwiałby komunikację z innym użytkownikiem. Aby gość stał się użytkownikiem musi zarejestrować nowy numer identyfikacyjny, numer ten jest niezbędny by móc używać aplikacji. Przypadek "Podaj numer" jest przypadkiem w którym użytkownik sam definiuje sobie numer identyfikacyjny.

**3. Architektura działającego oprogramowania:**

Aby możliwe było działanie tego systemu na zasadzie takiej jakiej opisane zostało w punkcie poprzednim, przyjęta została architektura klient - serwer - klient. Dzięki takiej przyjętej architekturze możliwe jest sterowanie sesją połączeniową w taki sposób by możliwe były przypadki typu:

* zajęty,
* niedostępny,
* niepodłączony

By możliwa była sesja między dwoma użytkownikami zdefiniowany zostanie protokół komunikacyjny warstwy zastosowań wykorzystujący protokół transportowy TCP ze względu na swoje właściwości: Pakiety te będą przesyłane w sytuacjach gdy użytkownik będzie próbował nawiązać połączenie z innym użytkownikiem oraz w momencie przerwania transmisji. Transmitowane będą dane w postaci tekstowej do serwera oraz pod pewnymi warunkami przekazywane do drugiego użytkownika przez serwer. Dane będą następujące rodzaje komunikatów:

* INVITE - wysyłany do odbiorcy za pośrednictwem serwera kiedy odbiorca ten jest podłączony w celu zaincjalizowania sesji połączeniowej,
* OK - wysyłane przez odbiorce oraz nadawce w celu akceptacji nawiązania połączenia oraz akceptacji zakończenia połączenia głosowego.
* NEX - (Not EXists) - komunikat zwrotny od serwera w wypadku gdy nadawca próbuje się połączyć z odbiorcą który nie istnieje,
* BUSY - komunikat zwrotny od serwera w wypadku gdy nadawca próbuje się połączyć z użytkownikiem który odrzucił połączenie bądź jest zajęty,
* HELLO - Komunikat wysyłany do serwera w celu poinformowania go o tym że użytkownik podłącza się do serwera,
* BYE - komunikat wysyłany do serwera w celu poinformowania go o tym że użytkownik odłącza się od serwera.

Komunikaty protokołu sterowania będą miały przyjętą postać prefiksową.

Przykłady komunikatów protokołu sterującego:

* INVITE 5550989 543432332 - Użytkownik o numerze 5550989 próbuje nawiązać połączenie z użytkownikiem 543432332,
* OK 543432332 - Użytkownik akceptuje połączenie nawiązane przez innego użytkownika będącego nadawcą,
* HELLO 554343234 - użytkownik o numerze identyfikacyjnym 554343234 informuje serwer o tym że chce się podłączyć do serwera.

Do transmisji głosu między dwoma użytkownikami wykorzystany będzie inny protokół który oparty będzie na protokole UDP ze względu na to by pakiety te mogły dojść do hostów możliwe w jak najkrótszym czasie. Transmisja głosu odbywać się będzie za pośrednictwem jednego portu.

**4. Środki techniczne wykorzystane w projekcie**

Do zaprojektowania oraz implementacji tego systemu wykorzystane zostaną następujące środki techniczne:

* **Visual Paradigm CE 12.0** - Aplikacja służąca do modelowania oraz projektowania w języku UML tego systemu. Wszystkie diagramy UML tworzone zostaną za pomocą tej aplikacji.
* **Github** - Repozytorium projektu w którym umieszczane zostaną prototypy aplikacji oraz cały projekt wraz z dokumentacją.
* **Serwer VPS** - Komputer klasy PC z zainstalowanym systemem operacyjnym Debian 7.6 Wheezy, posiadający publiczny adres IP oraz własną domenę. Na tym komputerze będzie działał serwer pełniący rolę centali oraz działała będzie Baza danych.
* **MySQL** - Baza danych działająca również na serwerze VPS, na której będzie przechowywana lista osób oraz przypisane do nich numery identyfikacyjne,
* **MS Visual Studio 2013** - Srodowisko programistyczne na którym zaimplementowana zostanie aplikacja klienta
* **Eclipse Luna** - Środowisko programistyczne na którym zaimplementowana zostanie aplikacja serwera. Ze względu na wieloplatformowość oraz na fakt że serwer będzie działał na systemie operacyjnym Linux preferowanym językiem programowania do zaimplementowania tego systemu jest JAVA.
* Biblioteka **BASS Audio** - Biblioteka służąca do przetwarzania, kodowania dźwięku w celu przetworzenia jego do strumienia cyfrowego.