

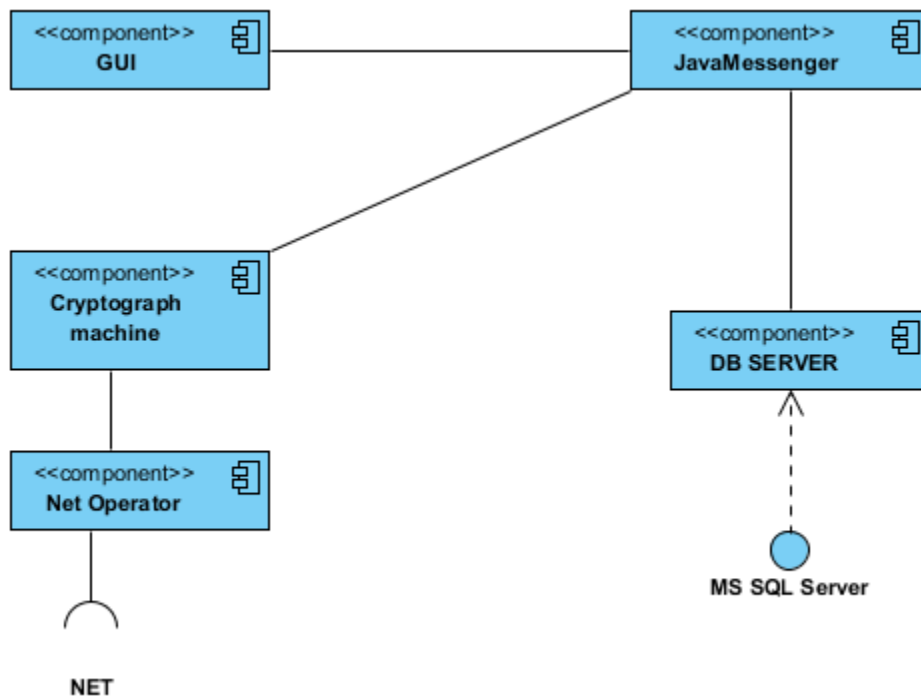
## System komunikacji i wideokonferencji

168080 Piotr Zuzak,  
168070 Norbert Zieliński,  
168010 Piotr Zakrzewski

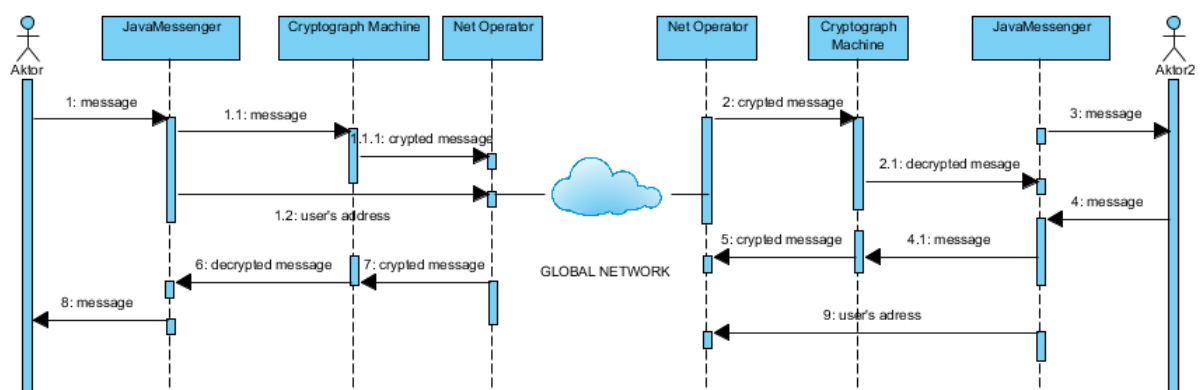
**OPIS:** Temat projektu dotyczy systemu informatycznego, który umożliwia szyfrowaną komunikację za pomocą stworzonego od podstaw protokołu komunikacyjnego wykorzystującego sieć Internet. System zostanie zaprojektowany w taki sposób, by umożliwić komunikację pomiędzy dwoma lub większą liczbą użytkowników poprzez wideokonferencję realizowaną za pomocą streamingu jak i tradycyjną komunikację za pomocą wiadomości tekstowych. System ten będzie samodzielną aplikacją okienkową, wymagająca połączenia z siecią Internet. Program będzie umożliwiał przesyłanie plików dowolnego typu. Przesłany plik muzyczny/video będzie można odtworzyć za pomocą streamingu a plik graficzny wyświetlić jako obrazek. Użytkownik systemu będzie mógł zapisać odebrane pliki w dowolnie wybranej lokalizacji dyskowej (domyślnie aplikacja zapisze w folderze pobrane). Każdy użytkownik systemu będzie posiadał swój unikalny profil. By zalogować się do systemu będzie musiał podać hasło, wpisane przy dodawaniu profilu. Podstawową funkcjonalnością systemu komunikacji jest zapewnienie wideokonferencji. Wideokonferencję rozpoczyna użytkownik zwany moderatorem. Zaczyna on od zaproszenia użytkowników do konferencji. W czasie trwania konferencji może on zaprosić nowych użytkowników lub wyrzucić tych, którzy biorą w niej aktualnie udział. Każdy z użytkowników biorących w niej udział widzi pozostałe osoby biorące udział w rozmowie. Dźwięk jest „wspólny”, tzn. że każdy z użytkowników słyszy wszystkich pozostałych. Wymagania minimalne szybkości połączenia do wideokonferencji to około 1Mbps dla każdego użytkownika połączonego. Użytkownik powinien mieć zainstalowane środowisko Java minimum w ver.1.6. Szyfrowane połączenie komunikacyjne w systemie to szyfrowanie SSL oraz szyfrowanie PGP/GPG, dzięki któremu zostanie wygenerowany klucz dla kont w systemie po stronie komunikatora. By wiadomości były szyfrowane trzeba wymienić oba klucze. Do przesyłania wiadomości tekstowych oraz opisów w statusach zastosowane będzie kodowanie UTF-8, dzięki czemu będzie można używać pełnego zestawu znaków Unicode. Aplikacja zostanie zaimplementowana z wykorzystaniem platformy Java. Protokół komunikacyjny będzie oparty o IM (instant messaging), wykorzystujący unicasting dla wideokonferencji (duża liczba bajtów).

Planowane użycie technologii:

- Java SDK v1.6, JDBC 4.0
- Netbeans 7.1
- Server dla DB: <http://www.freemysql.net/>
- Java Media Framework 2.1.1e
- szyfrowanie PGP/GPG, SSL



Rysunek 1 Diagram komponentów - Architektura aplikacji



Rysunek 2 Diagram sekwencji - Architektura komunikacji

## Harmonogram pracy zespołu:

1. **Zestawienie środowiska potrzebnego do implementacji projektu** - 1 tyg., 23.02.2012
  - Ustalenie wersji JAVA SDK, technologii
  - Założenie repozytorium systemu kontroli wersji na github.com
  - Założenie konta na serwerze <http://www.freemysql.net/>
2. **Przygotowanie graficznego interfejsu użytkownika** - 2 tyg., 8.03.2012

Prace nad implementacją okienkowej aplikacji po stronie użytkownika

  - Profil użytkownika
  - Lista kontaktów
  - Message box
  - Opcje
3. **Implementacja zdalnej bazy danych profili użytkowników** - 1 tyg., 15.03.2012
  - stworzenie schematu DB
  - implementacja skryptu bazy DB
  - stworzenie skryptu zapelnienia rekordami DB
  - CRUD w aplikacji, scaffolding
4. **Implementacja mechanizmu konferencji** - 1 tyg., 22.03.2012.
  - implementacja algorytmu wyboru uczestników konferencji video
5. **Implementacja protokołu dla komunikacji tekstowej** - 3 tyg., 12.04.2012
  - implementacja mechanizmu szyfrującego, deszyfrującego (Cryptograph machine)
  - implementacja obiektu sterującego i odbiornika (Net operator)
6. **Implementacja transferu plików** - 2 tyg., 26.04.2012
  - implementacja interfejsu do odbioru plików
  - implementacja funkcji przesyłu plików multimedialnych w JavaMessenger
7. **Implementacja streamingu audio i wideo** - 3 tyg., 17.05.2012
  - integracja biblioteki Java Media Framework z aplikacją JavaMessenger
  - implementacja modułu streamingu dla obiektu NetOperator