BABEL PROJECT

SKYPE-LIKE (VOIP)

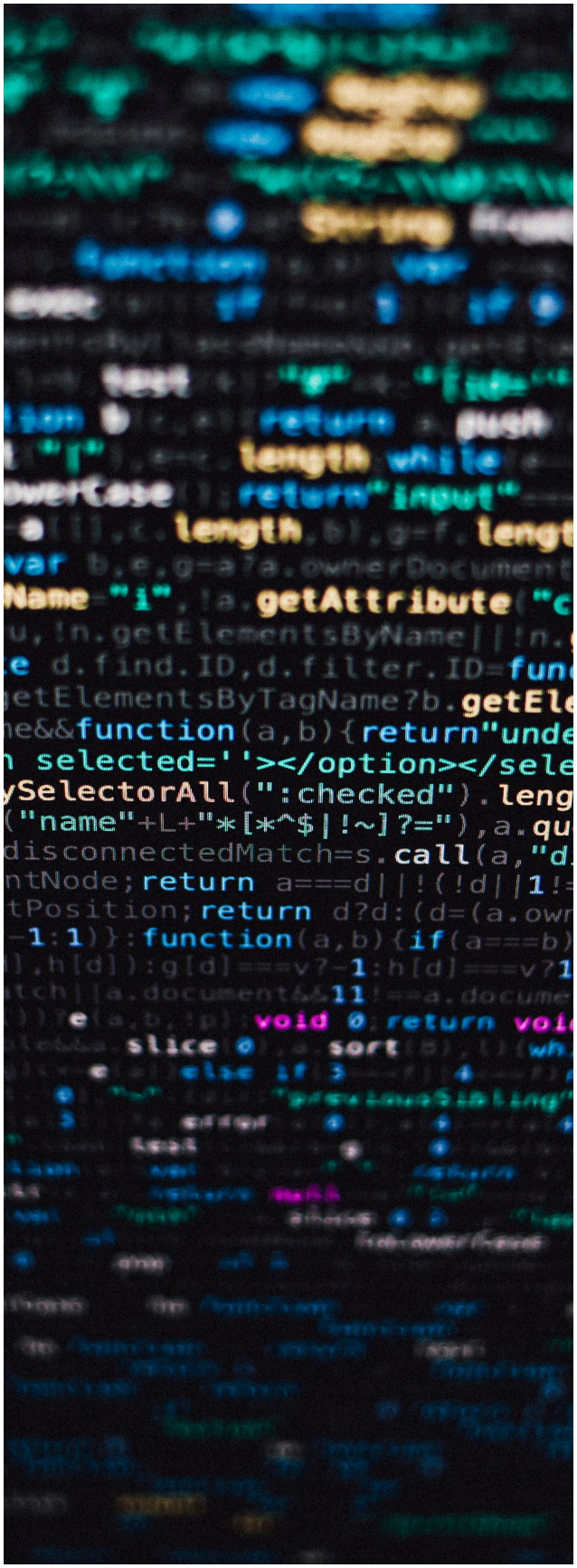
Create by:
Maxime Barbier, Guillaume Corbet, Alexandre Limongi, Damien Vachier

Contents

About

This document is a technical documentation about Babel. Babel is a software developed by a part of group Canap'gang, based in Marseille. This part of group developed a Skype-like product in a month for Epitech's project.

We firstly explain the project conception: it is divided in 2 parts. There is a Server and Client. The Server receives data form client and transfers in to the others. Client send a request to the server and get datas in order to call other clients



Client - Libraries Chosen

Qt

We use Qt for the Graphical User Interface and Qt Network for all the network part in client. We chose it because it is a good Cross-platform tools know for powerfull user interface. Otherthinks, It is a native C++ Library so we don't need to encapsulate it

Client - Libraries Chosen

Port Audio

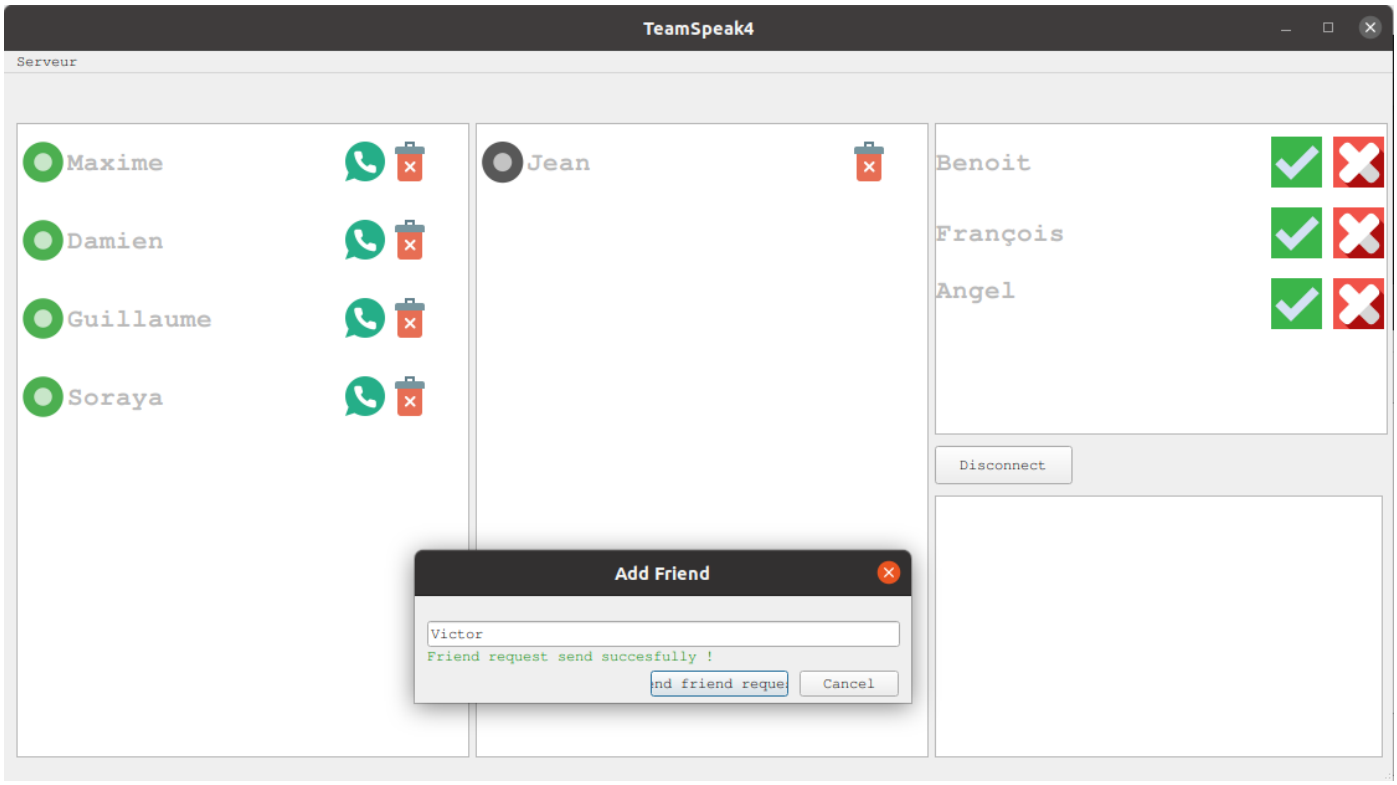
We use Port Audio for sound related things. We chose it because it s an Open-source Cross-platform Audio libraries recognized and used by many free audio editor like Audacity. It helps us to get sound from the computer's microphone and to play it througth it's speakers.

Client - Libraries Chosen

Opus

For the compression codec, we use Opus because is verry versatile with the two algorithm, and is totaly open source. One of the two algorithm is create by Skype, This is specialize in compression the human voice, and we use it after recode the voice using Port Audio for send to other client.

Client - User Interface



Here you can see the User Interface

Server – Libraries Chosen

Boost

The main library that we used for the server is Boost, Boost is a C++ Library and he is also a cross-platform, we use it mainly for network. It helps us to build a strong server with asynchronous multi client connections.

Server – Libraries Chosen

Sqlite

For all the server storage, we use Sqlite, in this we store all information about User

Server - Parsing

We see it, we use Sqlite for store all diverse things about User, and we use homemade parser for communicate, server and database

It show an exemple of the code for parsing :

```
#define LOGIN_SUCCES "10 LOGIN_SUCCES\n"
#define LOGIN_FAIL "11 LOGIN_FAIL\n"
#define LOGIN_SQL_ERROR "19 LOGIN_SQL_ERROR\n"
#define LOGOUT_SUCCES "20 LOGOUT_SUCCES\n"
#define LOGOUT_FAIL "21 LOGOUT_FAIL\n"
#define LOGOUT_SQL_ERROR "29 LOGOUT_SQL_ERROR\n"
#define CREATE_USER_SUCCES "30 CREATE_USER_SUCCES\n"
#define CREATE_USER_USER_ALREADY_EXIST "31 CREATE_USER_USER_ALREADY_EXIST\n"
#define CREATE_USER_SQL_ERROR "39 CREATE_USER_SQL_ERROR\n"
#define CREATE_CONTACT_SUCCES "40 CREATE_CONTACT_SUCCES\n"
#define CREATE_CONTACT_NO_REQUEST_CONTACT "41 CREATE_CONTACT_NO_REQUEST_CONTACT\n"
#define CREATE_CONTACT_ALREADY_EXIST "42 CREATE_CONTACT_ALREADY_EXIST\n"
#define CREATE_CONTACT_SQL_ERROR "49 CREATE_CONTACT_SQL_ERROR\n"
#define CREATE_REQUEST_CONTACT_SUCCES "50 CREATE_REQUEST_CONTACT_SUCCES\n"
#define CREATE_REQUEST_CONTACT_ALREADY_EXIST "51 CREATE_REQUEST_CONTACT_ALREADY_EXIST\n"
#define CREATE_REQUEST_CONTACT_CONTACT_ALREADY_EXIST "52 CREATE_REQUEST_CONTACT_CONTACT_ALREADY_EXIST\n"
#define CREATE_REQUEST_CONTACT_USER_NOT_EXIST "53 CREATE_REQUEST_CONTACT_USER_NOT_EXIST\n"
#define CREATE_REQUEST_CONTACT_SQL_ERROR "59 CREATE_REQUEST_CONTACT_SQL_ERROR\n"
#define DELETE_REQUEST_CONTACT_SUCCES "60 DELETE_REQUEST_CONTACT_SUCCES\n"
#define DELETE_REQUEST_CONTACT_NO_REQUEST_CONTACT "61 DELETE_REQUEST_CONTACT_NO_REQUEST_CONTACT\n"
#define DELETE_REQUEST_CONTACT_USER_NOT_EXIST "62 DELETE_REQUEST_CONTACT_USER_NOT_EXIST\n"
#define DELETE_REQUEST_CONTACT_SQL_ERROR "69 DELETE_REQUEST_CONTACT_SQL_ERROR\n"
#define DELETE_CONTACT_SUCCES "70 DELETE_CONTACT_SUCCES\n"
#define DELETE_CONTACT_NO_CONTACT "71 DELETE_CONTACT_NO_CONTACT\n"
#define DELETE_CONTACT_SQL_ERROR "79 DELETE_CONTACT_SQL_ERROR\n"
#define GET_FRIEND_CONNECTED_SUCCES "100 "
#define GET_FRIEND_CONNECTED_SQL_ERROR "109 GET_FRIEND_CONNECTED_SQL_ERROR\n"
#define GET_USERS_SUCCES "110 "
#define GET_USERS_SQL_ERROR "119 GET_USER_SQL_ERROR\n"
#define GET_ALL_FRIEND_SUCCES "120 "
#define GET_ALL_FRIEND_NO_CONTACT "120 \n"
#define GET_ALL_FRIEND_SQL_ERROR "129 GET_ALL_FRIEND_SQL_ERROR\n"
#define GET_FRIEND_REQUEST_SUCCES "130 "
#define GET_FRIEND_REQUEST_SQL_ERROR "139 GET_FRIEND_REQUEST_SQL_ERROR\n"
#define GET_STATUS "140 "
#define GET_STATUS_SQL_ERROR "149 GET_STATUS_SQL_ERROR\n"
#define GET_CALL_REQUEST "310 "
#define GET_CALL_REQUEST_NO_CALL_INCOMING "311 GET_CALL_REQUEST_NO_CALL_INCOMING\n"
#define GET_CALL_REQUEST_SQL_ERROR "319 GET_CALL_REQUEST_SQL_ERROR\n"

#define USER_TABLE_EMPTY "201 USER_TABLE_EMPTY\n"
#define CONTACT_TABLE_EMPTY "211 CONTACT_TABLE_EMPTY\n"
#define REQUESTCONTACTS_TABLE_EMPTY "221 REQUESTCONTACTS_TABLE_EMPTY\n"
#define DB_ERROR_IN_OPEN "301 DB_ERROR_IN_OPEN\n"
```