TalkEntreprise

Table des matières

[Introduction 3](#_Toc452888929)

[Description du projet 3](#_Toc452888930)

[Motivation 3](#_Toc452888931)

[Cahier des charges 3](#_Toc452888932)

[Particularité du projet 3](#_Toc452888933)

[Matériel Utilisé 3](#_Toc452888934)

[Logiciels utilissés 3](#_Toc452888935)

[Analyse fonctionnelle 3](#_Toc452888936)

[Préambule 4](#_Toc452888937)

[Point de vu du client/Administrateur 5](#_Toc452888938)

[fonctionnalité spécifique aux administrateur 6](#_Toc452888939)

[Choix des composants 6](#_Toc452888940)

[Schémat de l'application 6](#_Toc452888941)

[Analyse organique 6](#_Toc452888942)

[Choix du type de communication 6](#_Toc452888943)

[Diagramme de classes 6](#_Toc452888944)

[Fichier utile pour l'application 6](#_Toc452888945)

[Du coter Client 6](#_Toc452888946)

[Du coter Serveur 6](#_Toc452888947)

[Méthode d'envoie des messages 6](#_Toc452888948)

[Descritpion des méthodes principal 6](#_Toc452888949)

[Protocole de tests 6](#_Toc452888950)

[Conclusion 6](#_Toc452888951)

[Difficultés rencontrées 7](#_Toc452888952)

[Amélioration future 7](#_Toc452888953)

# Introduction

## Description du projet

Le but de ce projet est de créer une messagerie instantanée pour que les employés d'un même service, puissent communiquer entre eux.

## Motivation

Durant ma formation j'ai appris à utiliser différents langages de programmation, et j'ai choisi de réaliser mon projet en "C Sharp", car j'ai plus de facilité à l'utiliser qu'un autre langage.

Mon choix de créer une messagerie instantanée m'a été inspiré par mes camarades de classe.

Cela remonte à ma première année au CFPT : mes camarades et moi avions parlé de créer un site internet avec lequel nous aurions pu nous envoyer des messages durant les cours.

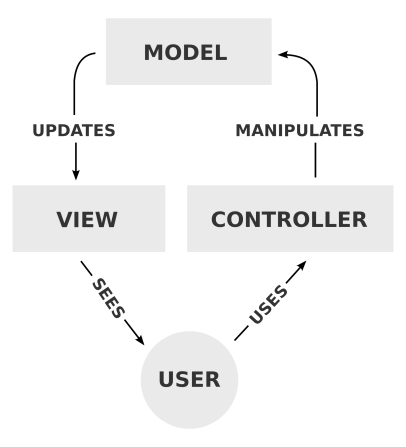
Nous avons essayé, mais nous nous sommes vite rendu compte que nous n'avions pas encore toutes les compétences nécessaires à sa réalisation.

En début d'année scolaire, nous en avons parlé à nouveau. J'ai alors décidé de me lancer ce défi pour mon TPI.

# Cahier des charges

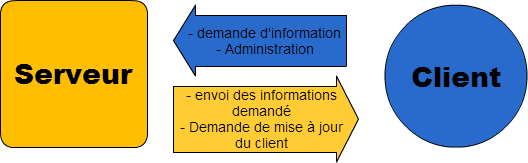
## Particularité du projet

Ce projet va être créé en C Sharp en Windows Form.

* Le programme va être réalisé en MVC (Modèle Vue Contrôleur).

Il y a 2 applications distinctes. Le côté serveur et le côté client

1 Concept du MVC



2 explication des 2 applications

Cette application va permettre d’améliorer la communication des informations dans un même département.

## Matériel Utilisé

* PC windows 7
* Disque dur externe
* Journal de bord (papier)

## Logiciels/fichiers utilissés

* Visual studio 2013
* Easyphp 14.1 VC9
* Suite office 2010
* Github desktop
* Cacoo.com
* mysql-connector-net-6.9.7
* WindowsBase.dll
* MysqlData.dll

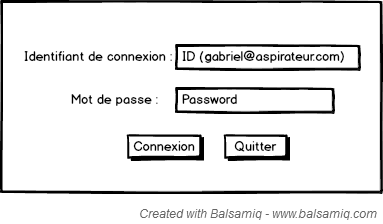
# Analyse fonctionnelle

## Préambule

Pour pouvoir utiliser cette application dans de bonnes conditions, plusieurs interfaces ont dû être créées.

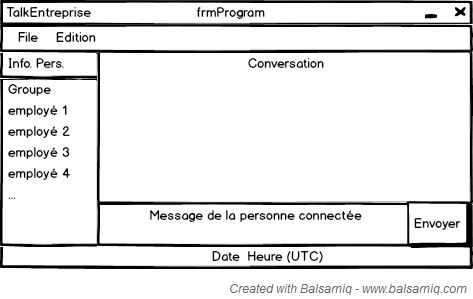
* Application "client"
  + Interface de connexion
  + Interface du programme
  + Interface de déconnexion
  + Interface "à propos"
  + Interface permettant de changer le mot de passe
* Application" serveur"
  + Interface de connexion
  + Interface du programme
  + Interface de déconnexion
  + Interface "à propos"
  + Interface permettant de changer le mot de passe
  + Interface permettant de modifier les informations des utilisateurs

## Point de vue du client/Administrateur



3 Interface de connexion

Cette interface permet à n’importe quel utilisateur de se connecter.



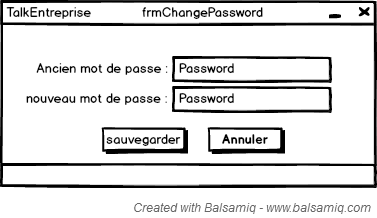
4 Interface du programme

Cette interface permet à l'utilisateur :

* d'envoyer un message
* d'accéder à la fenêtre de modification du

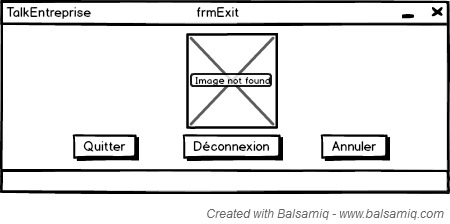
mot de passe

* de consulter la fenêtre "à propos"
* d'accéder à la fenêtre de déconnection.



5 Interface de modification du mot de passe

Cette interface permet à l'utilisateur la modification de son mot de passe.

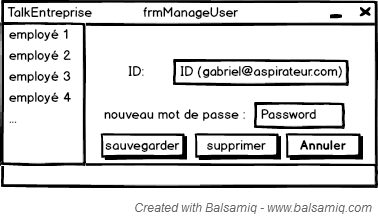


6 Interface de déconnexion

Cette interface permet à l’utilisateur :

* de quitter le programme
* de se déconnecter
* d’annuler la fermeture du programme.

## fonctionnalité spécifique aux administrateur



7 Interface permettant la modification des utilisateurs ainsi que leur suppression

Cette interface permet à l'administrateur de modifier/supprimer un utilisateur. Elle est accessible depuis l'interface du programme (voir image Interface du programme)

## Choix des composants

Mon programme est principalement composé de "textbox". J'ai choisi ce composant car grâce à lui je peux récupérer les informations tapé par l'utilisateur. Et le plus important est l'affichage des différentes conversations. Je peux décider de modifier à tout moment l'affichage des messages et le texte s'adapte automatiquement à la taille du "textbox".

Pour ce qui est de l'affichage des employés, j'ai décidé d'utiliser une "listbox" car elle me permet d' afficher plusieurs informations. Dans mon cas c'est si l'employé est connecté, le nom de celui-ci et le nombre de message en retard.

## Schémat de l'application

# Analyse organique

## Choix du type de communication

Lors de la conception du projet je pensais que l'application client allait chercher les données automatiquement sur la base de données. Mais après avoir discuté longuement avec Monsieur Garcia de la manière de récupérer les données, il m'a conseillé de creuser l'envoie de données via le réseau. En pesant le pour et le contre de nos méthodes, j'en ai conclus que l'envoie de donnés à travers le réseau était la meilleurs solution.

## Diagramme de classes de l'application CLient

## Diagramme de classes de l'application SERVEUR

## Fichier utile pour l'application

### Du coter Client

#### Les vues :

#### Les classes :

### Du coter Serveur

#### Les vues :

#### Les classes :

## Méthode d'envoi des messages

## Descritpion des méthodes principal

# Protocole de tests

# Conclusion

## Difficultés rencontrées

Au cours de ce projet, j'ai rencontré un bon nombre de problèmes qui sont du principalement à l'utilisation de certaines fonctionnalités que nous n'avons pas vu durant ma formation.

Lors de ma première Entrevu avec Monsieur Garcia, nous avons discuté sur comment l'application client allait pouvoir chercher les informations dans la base de données. Au début j'avais pensé que les utilisateurs se mettent à jour automatiquement en allant chercher les informations directement dans la base de données. Mais après cette discussion, j'ai pris le pari de me lancer dans l'envoi de données via le réseau.

J'ai alors commencé a cherché un site qui me permettent d'apprendre comment envoyer des informations sur le réseau. En cherchant des exemples de messageries instantanées, je suis tombé sur un site qui expliquait le code et qui donnait une démonstration. En utilisant le mode pas à pas sur Visual Studio j'ai pu comprendre comment cela fonctionnait.

Mon second problème a été de lancer la fenêtre du programme dans un nouveau processus. J'ai essayé de lancer cette fenêtre avec l'objet "Thread", mais cela ne fonctionnait pas. En essayant certain exemple, j'ai remarqué que beaucoup de forum parlait d'utiliser l'objet "Dispatcher". Mon souci a été que Visual studio n'acceptait pas le code pris sur le net. J'ai alors pensé que c'est du à l'absence d'une référence[[1]](#footnote-1) comme pour l'utilisation de MYSQL dans Visual Studio. En recherchant j'ai compris qu'il fallait belle et bien ajouté un une référence. Le dll manquant était "WindowsBase.dll"

Ma troisième difficulté est due à l'utilisation de l'objet "Thread". Lors de la création de ma méthode d'affichage des employées dans ma base de données, je récupérais les informations émis par le serveur pour les donner à mon objet "ListBox". Quand j'ai lancé le programme, j'ai eu une erreur qui me disait qu'il n'est pas possible de modifier un composant depuis un autre processus. En tapant sur le net mon problème, j'ai vu que un des moyens de donner ma liste était de passer par l'objet "Invoke". Il permet d'encapsuler (de sécuriser) des méthodes et de pouvoir modifier les informations de composant.

Ma quatrième difficulté a été de récupérer la conversation des groupes. Car lors de l'insertion du message pour chaque utilisateur présent dans le groupe, il fallait attendre quelques millisecondes. Dans ma requête de récupération de conversation de groupe, je supprime les doublons qui ont le même identifiant, le même message et la même date. Sachant que la date est stockée avec les secondes, il suffit que lors de l'enregistrement du même message on change de seconde pour que le message ne devienne pas un doublon. Pour pallier à cette difficulté, j'ai décidé de créer une autre méthode qui permet d'enregistrer le message à tous les personnes instantanément.

Mon cinquième et sixième problèmes ont été l'affichage de la fenêtre indiquant que le serveur a été éteint lors de la reconnexion et un message d'erreur que j'ai vu en lançant l'application depuis le dossier "Debug".

Ces erreurs sont liés car lors de la fermeture de la fenêtre principal, le processus est censé ce fermer. Mais vu que j'utilise l'objet "Dispatcher",

## Amélioration future

1. Librairie de fichiers ".dll" qui permet l'utilisation de différent type d'objet [↑](#footnote-ref-1)