|  |
| --- |
| Chat Application |

Gianinetti Lucas

Avenue de la gare 2

1450 Ste-Croix

Table des matières

[1 Analyse préliminaire 3](#_Toc511900269)

[1.1 Introduction 3](#_Toc511900270)

[1.2 Objectifs 3](#_Toc511900271)

[1.3 Planification initiale 4](#_Toc511900272)

[2 Analyse / Conception 6](#_Toc511900273)

[2.1 Concept 6](#_Toc511900274)

[2.2 Stratégie de test 6](#_Toc511900275)

[2.3 Risques techniques 6](#_Toc511900276)

[2.4 Planification 6](#_Toc511900277)

[2.5 Dossier de conception 7](#_Toc511900278)

[2.5.1 Hardware : 7](#_Toc511900279)

[2.5.2 Logiciels : 7](#_Toc511900280)

[2.5.3 Base de données : 7](#_Toc511900281)

[2.5.4 Cas d’utilisation : 8](#_Toc511900282)

[2.5.5 Maquettes : 8](#_Toc511900283)

[2.5.6 Scénarios : 19](#_Toc511900284)

[2.5.7 Architecture : 28](#_Toc511900285)

[3 Réalisation 28](#_Toc511900286)

[3.1 Dossier de réalisation 28](#_Toc511900287)

[3.2 Description des tests effectués 29](#_Toc511900288)

[3.3 Erreurs restantes 29](#_Toc511900289)

[3.4 Liste des documents fournis 29](#_Toc511900290)

[4 Conclusions 29](#_Toc511900291)

[5 Annexes 30](#_Toc511900292)

[5.1 Résumé du rapport du TPI / version succincte de la documentation 30](#_Toc511900293)

[5.2 Sources – Bibliographie 30](#_Toc511900294)

[5.3 Journal de travail 30](#_Toc511900295)

[5.4 Manuel d'Installation 30](#_Toc511900296)

[5.5 Manuel d'Utilisation 30](#_Toc511900297)

[5.6 Archives du projet 30](#_Toc511900298)

# Analyse préliminaire

## Introduction

Le projet consiste à réaliser une application d’un chat en C# (Windows Form) pour le module programmation/web.

J’ai choisi ce sujet parce que c’est un sujet qui m’intéresse et que je pourrai reprendre pendant mon temps libre afin de le perfectionner et l’améliorer.

## Objectifs

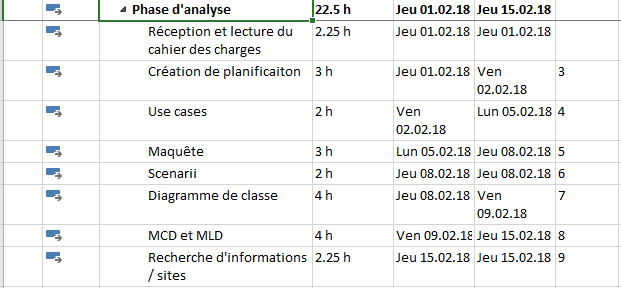
Les différents objectifs sont les suivants :

* Créer des groupes ou des chats de deux personnes seulement.
* Rejoindre une discussion sur un sujet : discussion ouverte à tous, il suffit de taper le nom du sujet et de rejoindre la room.
* Enregistrement des discussions : les discussions non supprimées apparaissent à la réouverture de l’application.
* Archivage des discussions : les discussions archivées n’apparaissent pas directement mais sont disponibles d’accès grâce à un dossier.
* Contacts : Création/Suppression/Modification (prénom, nom, possibilité d’ajoute une note sur le contact).
* Images/Smileys : possibilité d’ajouter des smileys et images dans les conversations.
* Messages vocaux.

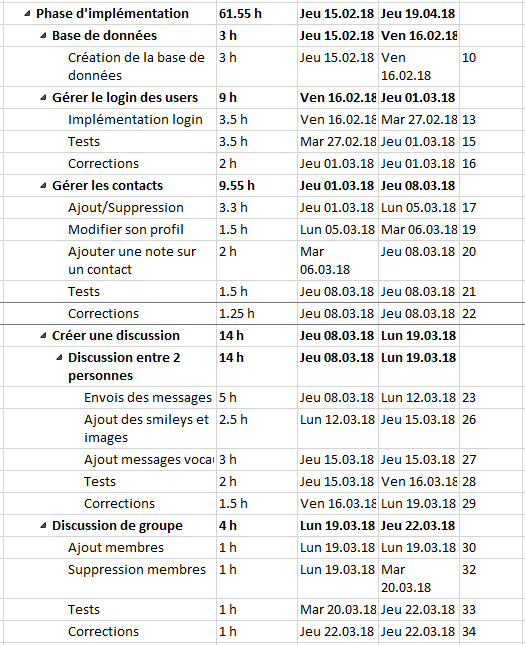
## Planification initiale

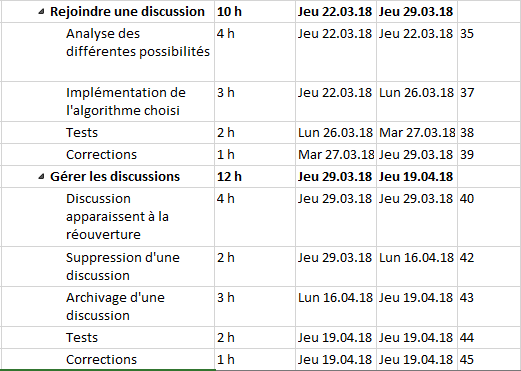
Ma planification initiale est séparée en trois grandes étapes :

* La phase d’analyse :

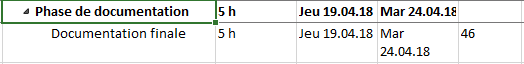


* La phase de conception :





* La phase de documentation :



# Analyse / Conception

## Concept

Un utilisateur peut s’enregistrer afin de pouvoir communiquer via un chat écrit avec d’autres utilisateurs dans des discussions privées ou publiques. Pour ce faire l’utilisateur peut ajouter d’autres utilisateurs en tant que contact.

Toutes les informations concernant les utilisateurs, les discussions et les messages contenus dans ses discussions sont stockés dans une base de données locale afin d’y avoir encore accès lors de la prochaine utilisation de l’application.

## Stratégie de test

Lors du développement du projet, j’ai testé les fonctionnalités au fur et à mesure qu’elles ont été développées. J’ai pris compte des différents bugs et particularités pour chacune des fonctions.

A la fin du développement, le projet a été testé par moi-même sur une machine regroupant le client et le serveur. Puis, sur une machine cliente et une machine serveur séparée.

Les tests sont réalisés sur des machine Windows 7 (64x) avec la base de données MySQL et l’application installées.

## Risques techniques

Le principal risque est que je ne connaissais pas comment utiliser les sockets avant de commencer le projet. Et que mon ambition a été trop grande pour pouvoir réaliser toutes les fonctions que j’ai soumis dans mon cahier des charges. C’est pourquoi je vais malheureusement abandonner le traitement d’image (image de profil, smileys, images) et les messages vocaux.

## Dossier de conception

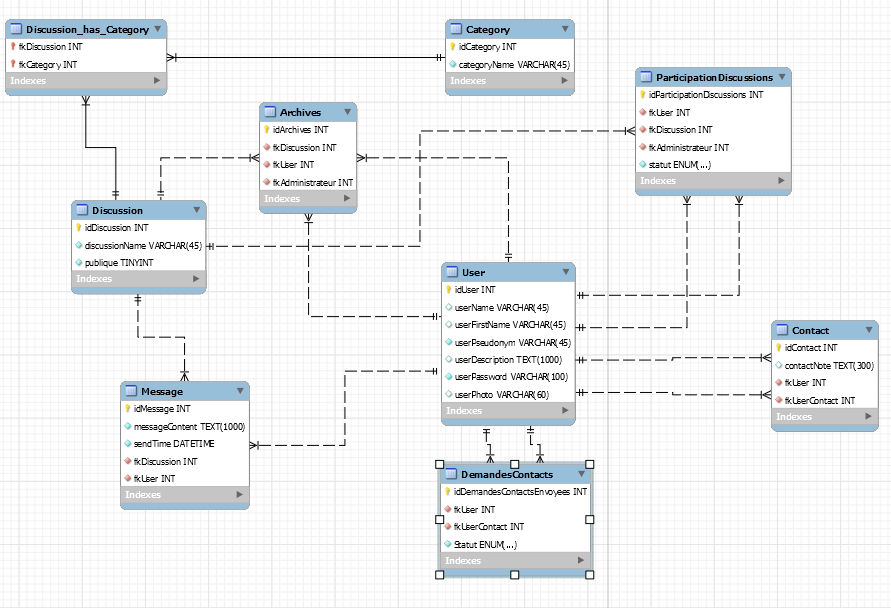
### Hardware :

Une partie du développement se déroulera sur un ordinateur Dell Optiplex 990 sous Windows 7(64x), l’autre partie se déroulera sur un ordinateur personnel sous Windows 10 (64x)

### Logiciels :

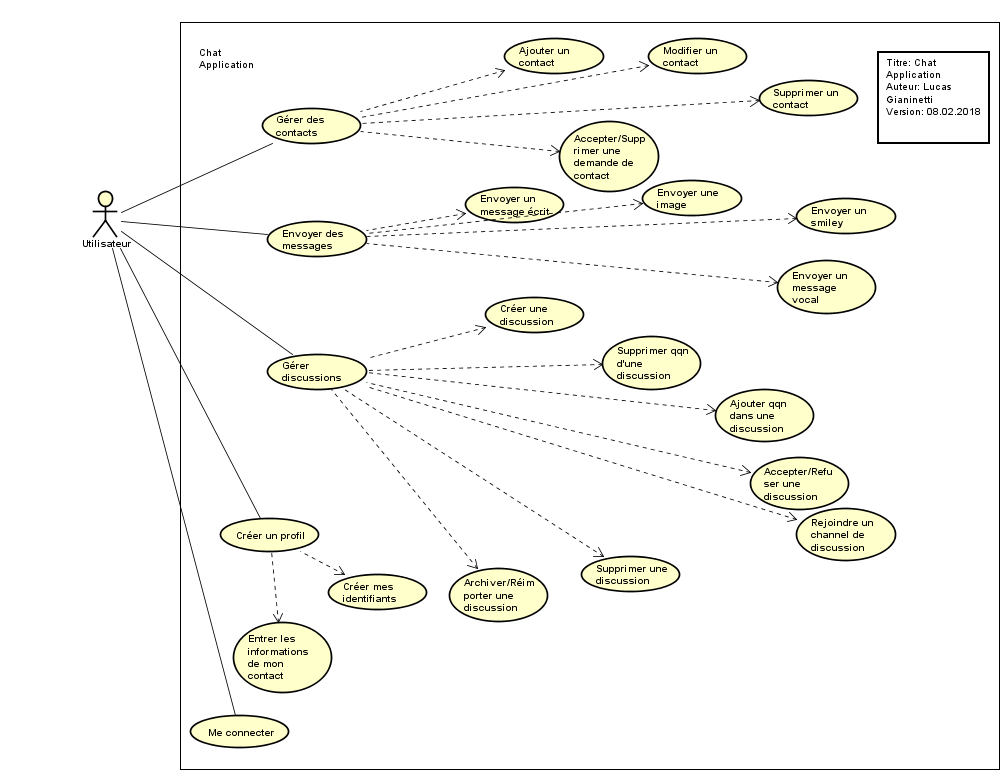
L’application est réalisée avec Visual Studio 2015 en C# avec des « Windows Form ». Elle communique avec une base de données MySQL. Wamp a été utilisé pour fournir le serveur de base de données

### Base de données :

Un script créant cette base de données est fourni, il y insère aussi les 10 catégories par défaut.

* ParticipationDiscussions : Le champ « statut » est une énumération (« En attente », « Participe »), il permet de savoir si un utilisateur participe à une discussion ou s’il a seulement reçu une invitation pour participer à une discussion.
* DemandesContacts : Le champ « statut » est une énumération («Recue», «Envoyee»), il permet de savoir si l’utilisateur a reçu ou envoyé une demande de contact.
* Discussion : Le champ « publique » est un boolean permettant de savoir si une discussion est privée ou publique.

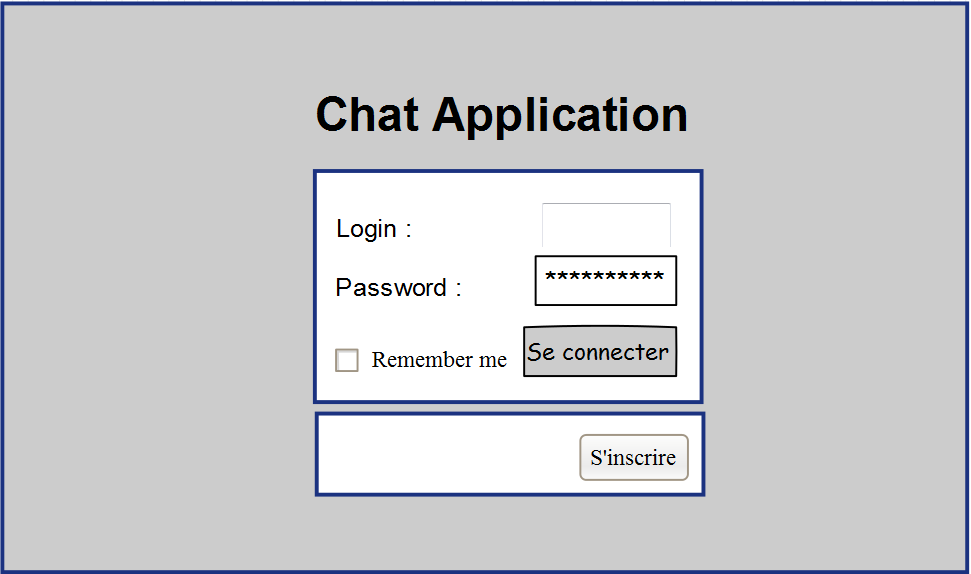
### Cas d’utilisation :



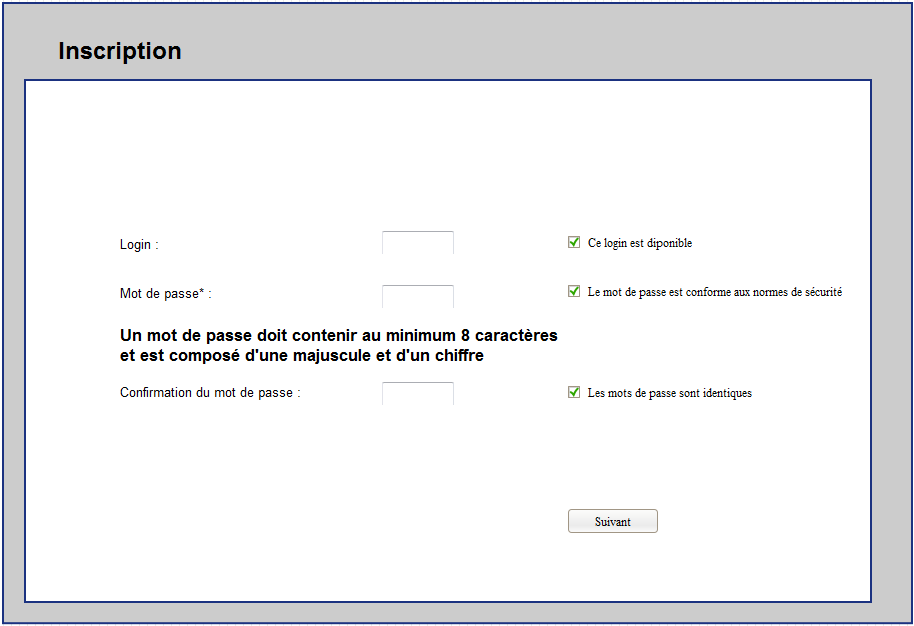
### Maquettes :

Il y a quelques différences entre les maquettes et l’aspect final de l’application. En effet, j’ai dû abandonner quelques fonctionnalités. Les boutons/éléments qui étaient prévu pour ses fonctionnalités ont été supprimé de l’application.

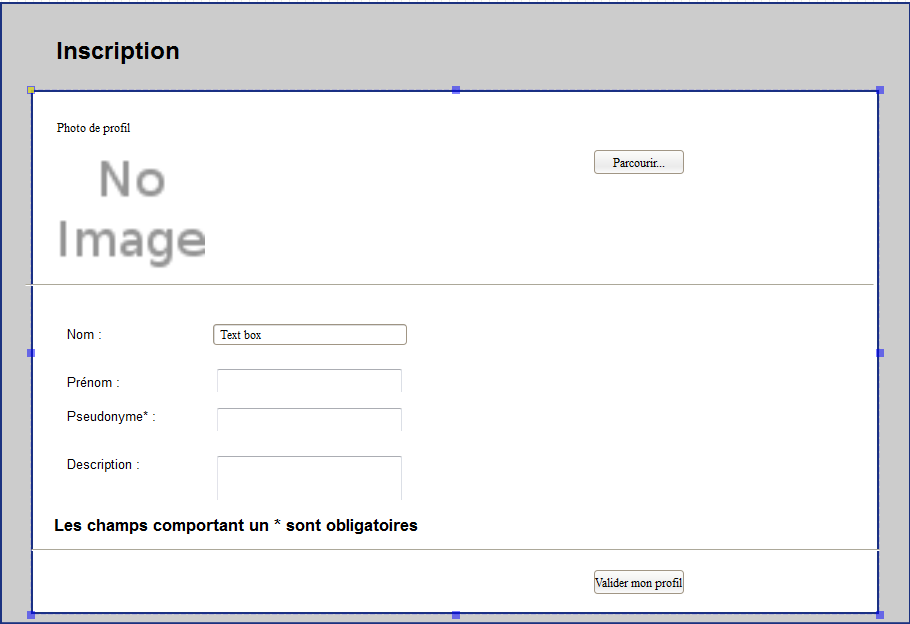
Connexion :



Enregistrement1 :



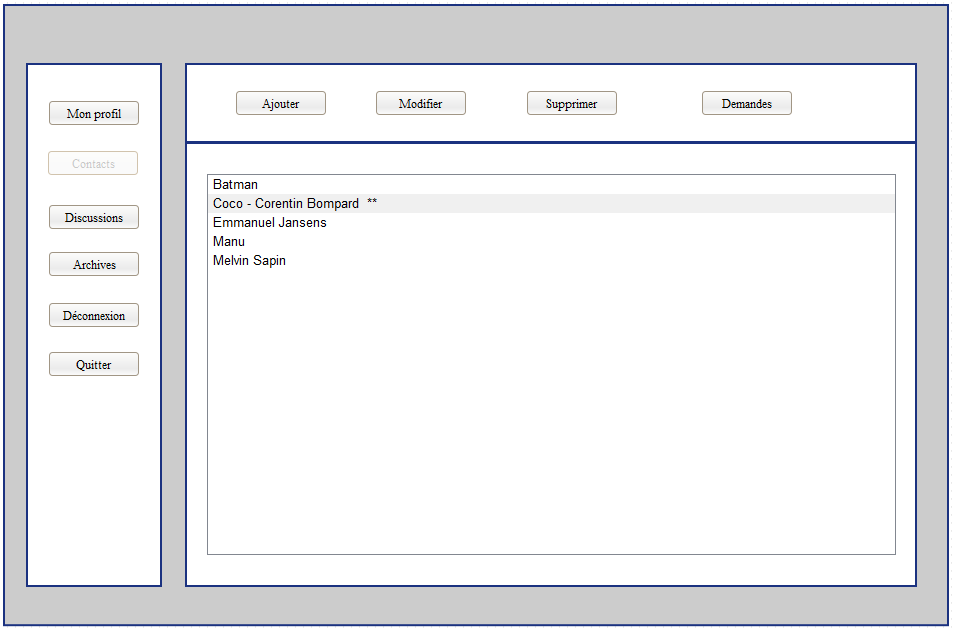
Enregistrement2 :

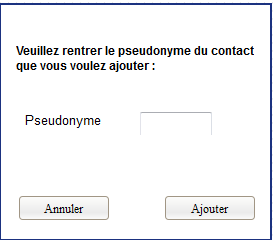


Mon profil :

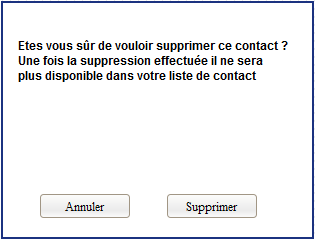


Contacts :

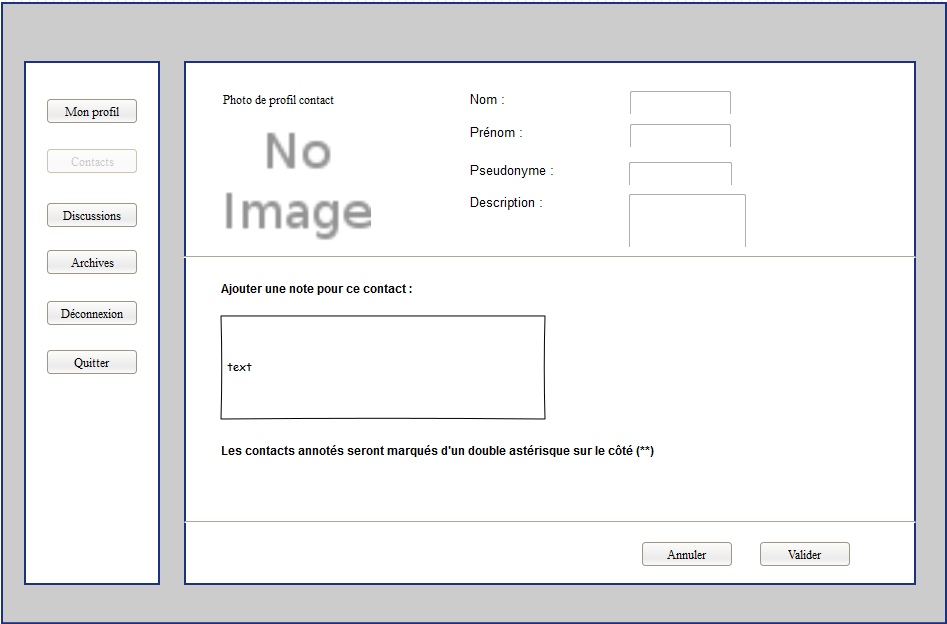


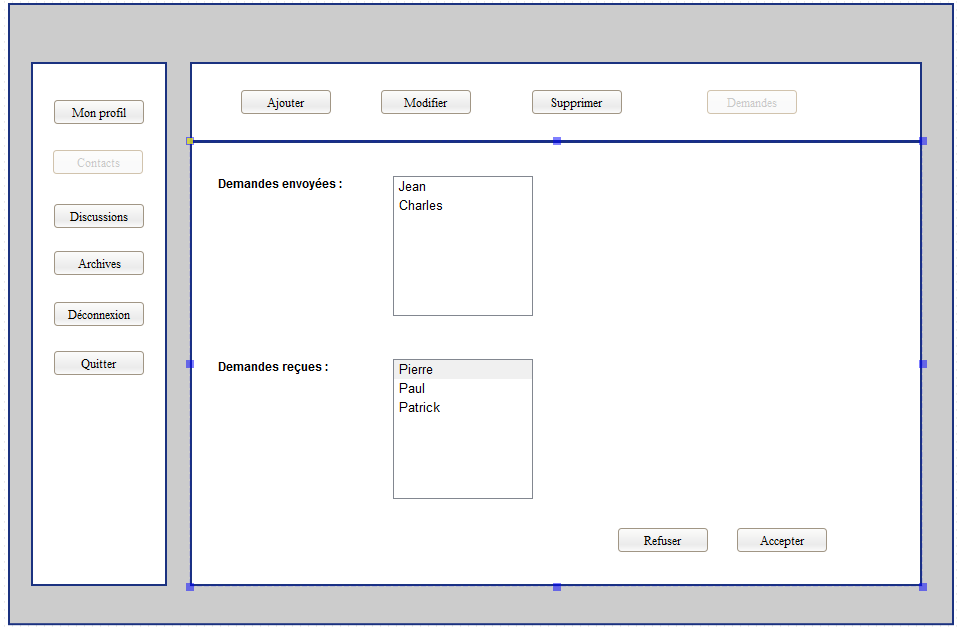
ContactAjouter :  


ContactSupprimer :

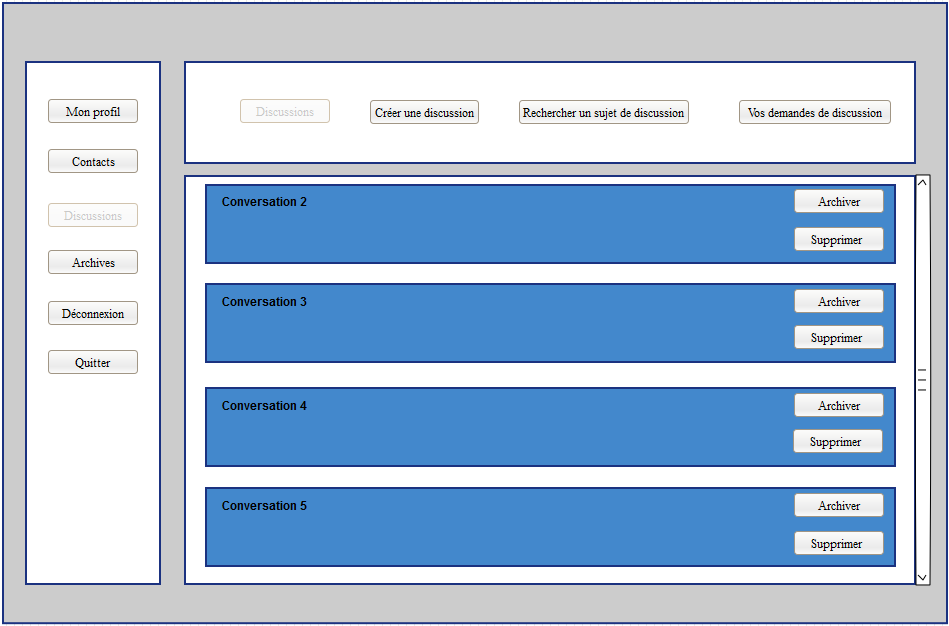


ContactModifier

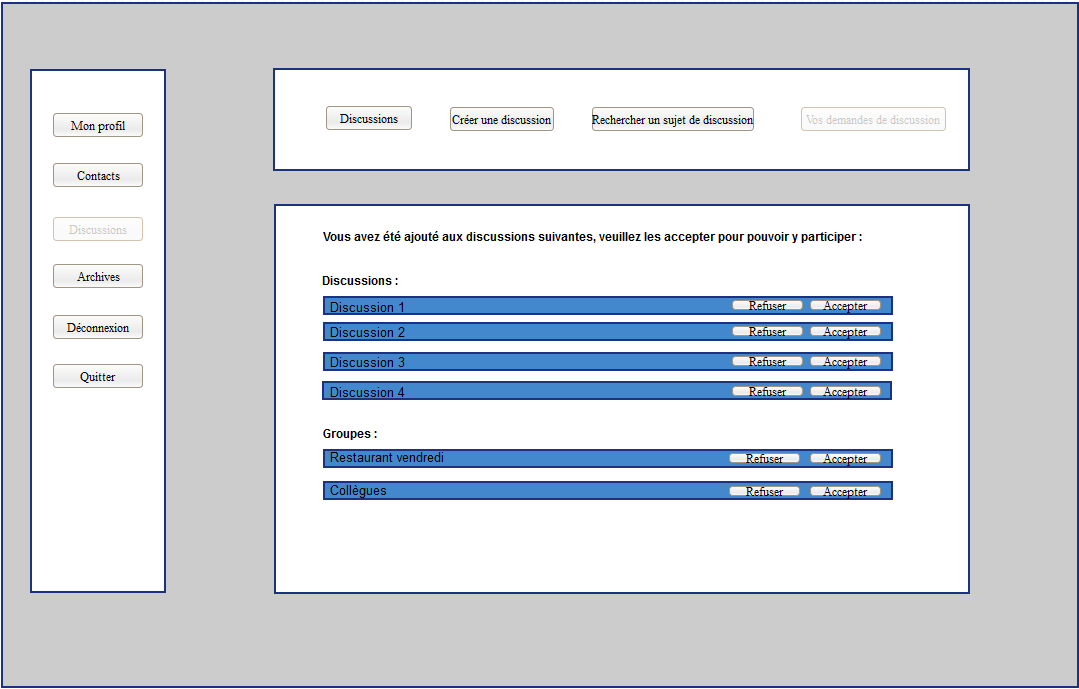


ContactDemandes :  


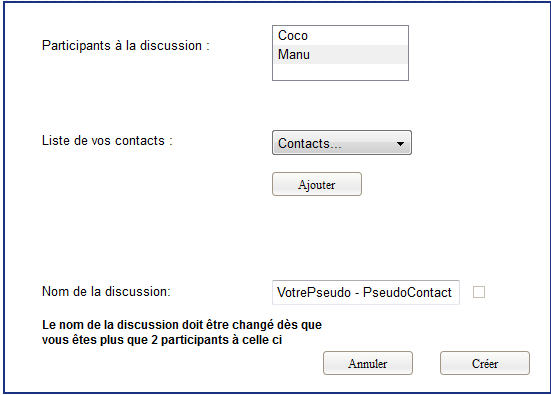
Discussions :



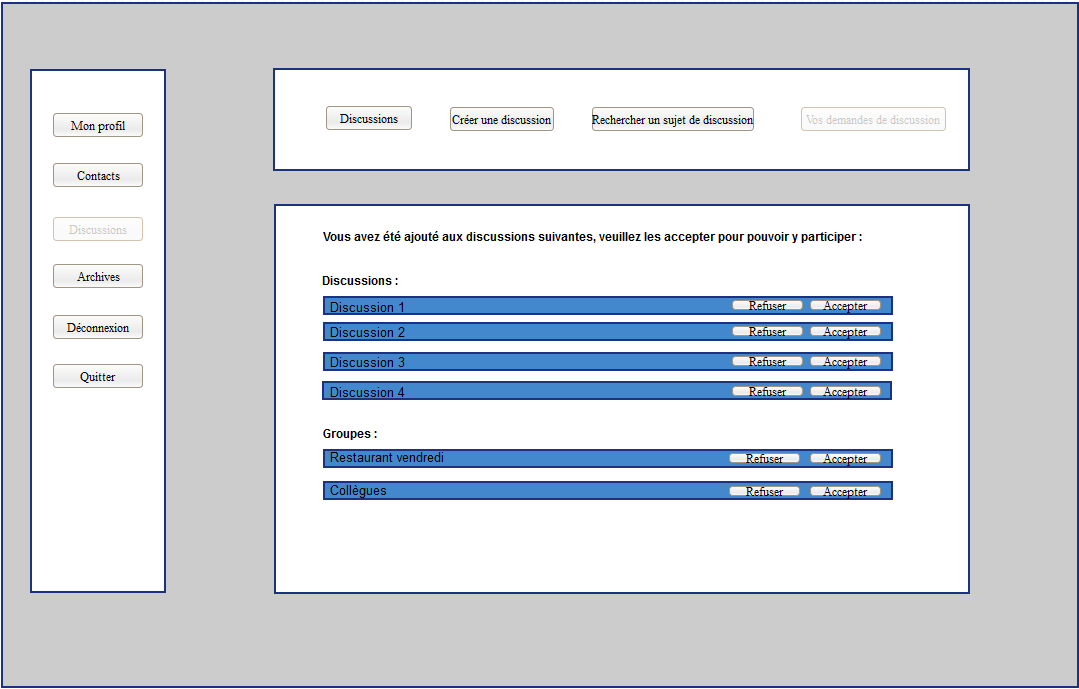
DiscussionDemande :



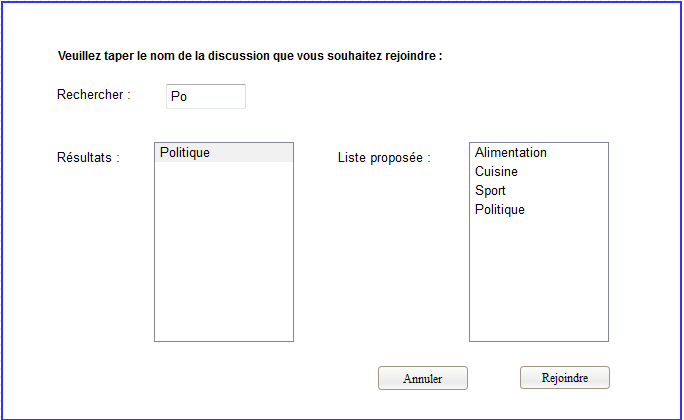
DiscussionCreer :



DiscussionDemande :



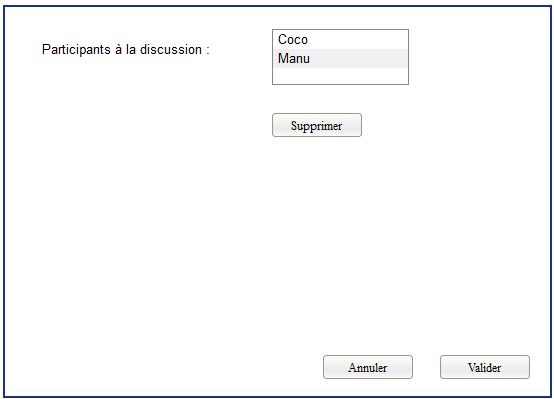
DiscussionRechercher :



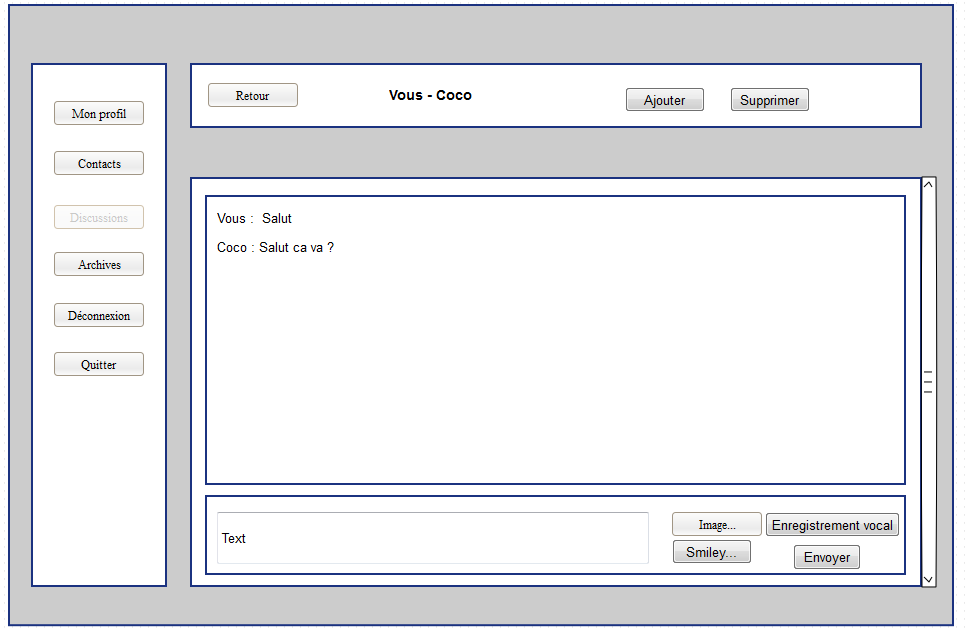
DiscussionAjoutContact :



DiscussionSuppressionContact :



EnvoisMessage :



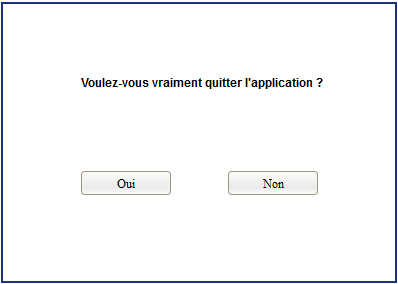
Archives :



MessageErreur :



Quitter :



### Scénarios :

|  |  |
| --- | --- |
| Identifiant | 1 |
| En tant que | Utilisateur |
| Je veux | Créer un profil |
| Pour | Discuter avec des contacts |
| Charge estimée | 9h |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Je lance le programme | Le programme lance une fenêtre et ouvre la page de connexion |
| Je clique sur s’inscrire | La page inscription apparait |
| Je remplis les champs correctement | Les checkbox correspondantes sont validées |
| Je clique sur suivant | La page inscription2 apparait |
| Je remplis les champs obligatoires | Le bouton « valider » est enable |
| Je clique sur le bouton « valider » | Mon profil est créé et retour à la page de connexion |

|  |  |
| --- | --- |
| Identifiant | 2 |
| En tant que | Utilisateur |
| Je veux | Me connecter |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Je rentre le login est le password | Le bouton se connecter est enable |
| Je coche la checkbox « Remember me » | Mon login sera enregistré pour la prochaine connexion |
| Je clique sur me connecter avec le mdp correspondant au login | J’arrive sur la fenêtre discussion |
| Je clique sur me connecter avec des faux identifiants | Un message d’erreur apparait |

|  |  |
| --- | --- |
| Identifiant | 3 |
| En tant que | Utilisateur |
| Je veux | Ajouter un contact |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Dans l’onglet contact je clique sur ajouter | La fenêtre ContactAjouter apparait |
| Je rentre le pseudo de mon contact | Le bouton « Ajouter » est enable |
| Je clique sur « Ajouter » | Une fenêtre apparait pour me dire si la demande de contact a été envoyée ou non en fonction de si mon contact existe vraiment. Une fois que le contact a accepté la demande il apparaîtra dans la liste des contacts |

|  |  |
| --- | --- |
| Identifiant | 4 |
| En tant que | Utilisateur |
| Je veux | Modifier un contact |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Je sélectionne un contact puis je clique sur modifier | La page Contact-Modifier apparait avec les informations du contact |
| Je remplis une note pour ce contact | Le bouton « valider » est enable |
| Je clique sur « valider » | La modification est apportée au contact |

|  |  |
| --- | --- |
| Identifiant | 5 |
| En tant que | Utilisateur |
| Je veux | Supprimer un contact |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Je sélectionne un contact et je clique sur « supprimer » | La fenêtre ContactSupprimer apparait |
| Je clique sur « annuler » | Retour à l’onglet contact |
| Je clique sur « supprimer » | Le contact est supprimé |

|  |  |
| --- | --- |
| Identifiant | 6 |
| En tant que | Utilisateur |
| Je veux | Accepter/Supprimer une demande de contact |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Je clique sur « Demandes » dans l’onglet contact | La fenêtre contact demande apparait |
| Je sélectionne le contact que je veux accepter | Le bouton accepter et refuser son enable |
| Je clique sur « Accepter » ou « Supprimer » | Le contact est ajouté à la liste de contact et disparait de la liste des demandes reçues ou le contact disparait de la liste des demandes reçues |

|  |  |
| --- | --- |
| Identifiant | 7 |
| En tant que | Utilisateur |
| Je veux | Envoyer un message écrit |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Je sélectionne une discussion | La fenêtre envoisMessage correspondante à la discussion sélectionnée apparait |
| Je rentre du texte et je clique sur « envoyer » | Le texte tapé est envoyé |

|  |  |
| --- | --- |
| Identifiant | 8 |
| En tant que | Utilisateur |
| Je veux | Envoyer une image |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Je sélectionne une discussion | La fenêtre envoisMessage correspondante à la discussion sélectionnée apparait |
| Je clique sur le bouton « image » | Je peux sélectionner une image sur mon pc |
| J’ai sélectionné l’image souhaitée et je clique sur « envoyer » | L’image est envoyée |

|  |  |
| --- | --- |
| Identifiant | 9 |
| En tant que | Utilisateur |
| Je veux | Envoyer un smiley |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Je sélectionne une discussion | La fenêtre envoisMessage correspondante à la discussion sélectionnée apparait |
| Je clique sur le bouton « smiley » | Je peux sélectionner un des smileys disponibles |
| J’ai sélectionné le smiley souhaitée et je clique sur « envoyer » | Le smiley est envoyé |

|  |  |
| --- | --- |
| Identifiant | 10 |
| En tant que | Utilisateur |
| Je veux | Envoyer un message vocal |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Je sélectionne une discussion | La fenêtre envoisMessage correspondante à la discussion sélectionnée apparait |
| Je clique sur le bouton « Enregristrement vocal » | Ce qui va être dit à travers le micro sera enregistré. |
| Je clique une deuxième fois sur le bouton « Enregsitrement vocal » | L’enregistrement s’arrête. |
| Je clique sur « Envoyer » | Le message vocal est envoyé. |

|  |  |
| --- | --- |
| Identifiant | 11 |
| En tant que | Utilisateur |
| Je veux | Créer une discussion |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Dans l’onglet discussion je clique sur « Créer une discussion » | La fenêtre « Créer une discussion » apparait |
| J’ajoute les différents contacts que je souhaite dans la discussion | Ils apparaissent dans la liste « participants à la discussion » |
| Si la discussion compte plus de 2 personnes, je change le nom de la discussion | La checkbox correspondante est validée |
| Je clique sur « créer » | La discussion est créée est apparait maintenant dans mes discussions |

|  |  |
| --- | --- |
| Identifiant | 12 |
| En tant que | Utilisateur |
| Je veux | Ajouter un contact à une discussion |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Dans l’onglet discussion je sélectionne une discussion | La fenêtre de la discussion correspondante apparait |
| Je clique sur « ajouter » | La fenêtre pour ajouter un contact à la discussion apparait |
| Je sélectionne les différents contacts que je veux ajouter et je clique sur « ajouter » | Les différents contacts apparaissent dans la liste des « participants à la discussion » |
| Je clique sur « valider » | Les contacts ont été ajouté à la discussion. |

|  |  |
| --- | --- |
| Identifiant | 13 |
| En tant que | Utilisateur |
| Je veux | Supprimer un contact d’une discussion |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Dans l’onglet de discussion je sélectionne une discussion | La fenêtre de la discussion correspondante apparait |
| Je clique sur « supprimer » | La fenêtre pour supprimer un contact à la discussion apparait |
| Je sélectionne les différents contacts que je veux supprimer et je clique sur « supprimer » | Les différents contacts disparaissent dans la liste des « participants à la discussion » |
| Je clique sur « valider » | Les contacts ont été supprimé de la discussion. |

|  |  |
| --- | --- |
| Identifiant | 14 |
| En tant que | Utilisateur |
| Je veux | Accepter/Refuser une discussion |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Dans l’onglet de discussion je sélectionne « vos demandes de discussions » | La fenêtre des demandes de discussions apparait. |
| Je clique sur « accepter » à côté de la discussion qui m’intéresse | La demande de discussion disparait de la liste et apparait dans mes discussions. |
| Je clique sur « refuser » à côté de la discussion qui m’intéresse | La demande de discussion disparait. |

|  |  |
| --- | --- |
| Identifiant | 15 |
| En tant que | Utilisateur |
| Je veux | Rejoindre un channel de discussion |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Je clique sur « Rechercher un sujet de discussion » dans l’onglet discussion | La fenêtre rechercher une discussion apparait |
| Je tape un mot dans la barre de recherche | Les discussions associées à ce mot sont listées dans la liste « Résultats » |
| Je sélectionne le résultat de ma recherche ou une thématique proposée puis je clique sur « rejoindre » | La discussion concernant la thématique sélectionnée apparait dans mes discussions |

|  |  |
| --- | --- |
| Identifiant | 16 |
| En tant que | Utilisateur |
| Je veux | Supprimer une discussion |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Dans l’onglet discussion je clique sur le bouton « supprimer » correspondant à une discussion | La discussion correspondante est supprimée définitivement. |
| Dans l’onglet « archives » je clique sur le bouton « supprimer » correspondant à une discussion | La discussion correspondante est supprimée définitivement. |

|  |  |
| --- | --- |
| Identifiant | 17 |
| En tant que | Utilisateur |
| Je veux | Archiver/Desarchiver une discussion |
| Priorité |  |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| Dans l’onglet discussion je clique sur le bouton « archiver » correspondant à une discussion | La discussion correspondante est supprimée des discussions et apparait maintenant dans les archives. |
| Dans l’onglet archives je clique sur le bout « réimporter » correspondant à une discussion | La discussion correspondante est supprimée des archives et apparait maintenant dans les discussions. |

### Architecture :

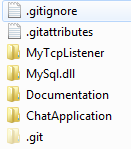
Le client est composé de formulaires qui gèrent l’aspect graphique, et d’une classe permettant d’envoyer les données au serveur.

Le serveur récupère ses données, et suivant le type de données il effectue diverse opérations avec la base de données.

# Réalisation

## Dossier de réalisation

Le dossier principal s’appelle « Chat » et se présente ainsi :

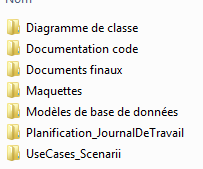
Les fichiers « .gitignore » et « .gitattributes » ainsi que le dossier « .git » sont des fichiers/dossier utilisés par le gestionnaire de version GitHub.

Le dossier « MyTcpListener » contient la solution Visual Studio du serveur.

Le dossier « ChatApplication » contient la solution Visual Studio du client.

Le dossier « MySql.dll » contient la librairie « MySql.Data.dll » qui permet de communiquer avec la base de données.

Le dossier « Documentation » se présente ainsi :



Le dossier « Diagramme de classe » contient des images représentant les différents diagrammes de classe.

Le dossier « Documents finaux » contient la documentation du projet, un fichier Excell contenant les tests effectués, le document d’installation et une vidéo expliquant l’utilisation de l’application. Un setup « ChatApplicationServeurSetup.exe » qui permet d’installer l’exe du serveur. Un setup « ChatApplicationClientSetup.exe » qui permet d’installer l’exe du client.

Le dossier « Maquettes » contient des images représentant les différentes maquettes.

Le dossier « Modèles de base de données » contient le MCD, le MLD et le script « DBChatApplication.sql ».

Le dossier « Planification\_JournalDeTravail » contient le journal de travail et la planification initiale.

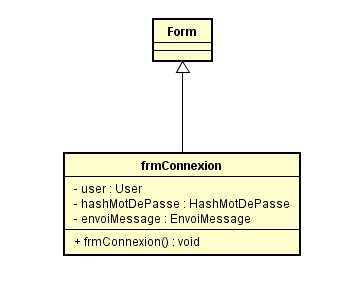
Le dossier « UseCases\_Scenarri » contient les uses cases et les scenarri.

Le dossier « Documentation code » contient la documentation du code du client et du serveur.

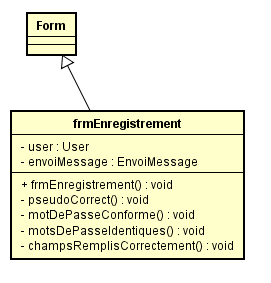
## Diagramme de classe

**Client :**

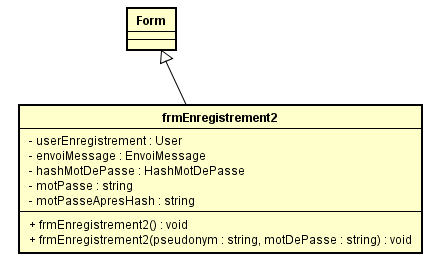
Formulaire connexion :



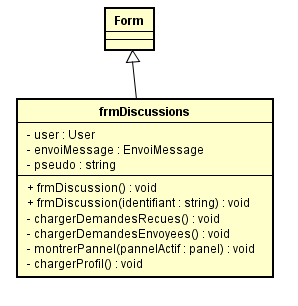
Formulaire d’enregistrement 1 :



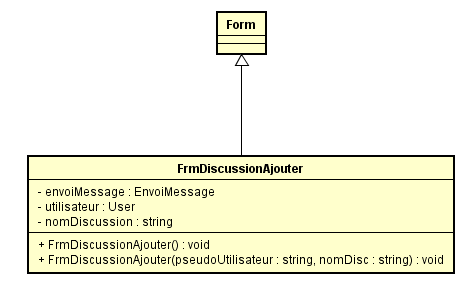
Formulaire d’enregistrement 2 :



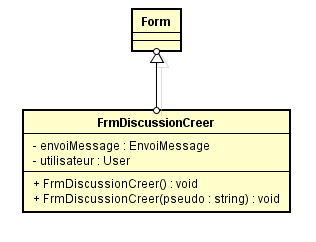
Formulaire général :



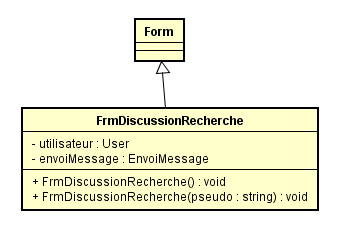
Formulaire DiscussionAjouter :



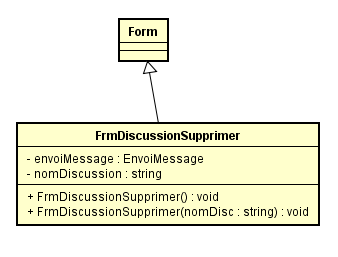
Formulaire DiscussionCreer :



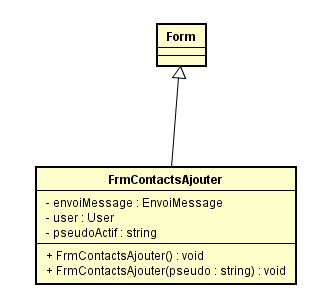
Formulaire DiscussionRecherche :



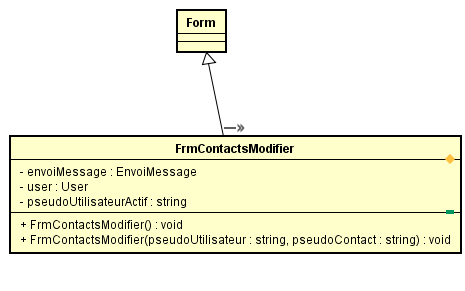
Formulaire DiscussionSupprimer :



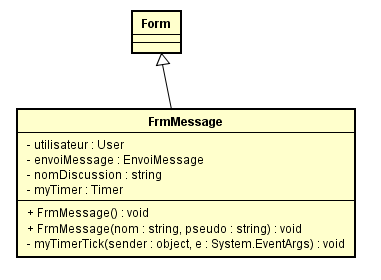
Formulaire ContactsAjouter :



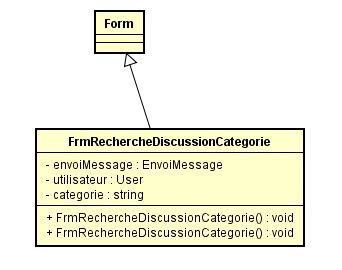
Formulaire ContactsModifier :



Formulaire Message :



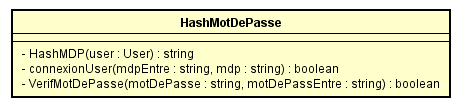
Formulaire RechercheDiscussionCategorie :



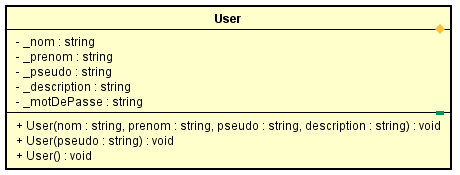
EnvoiMessage :



HashMotDePasse :

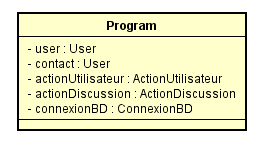


User :

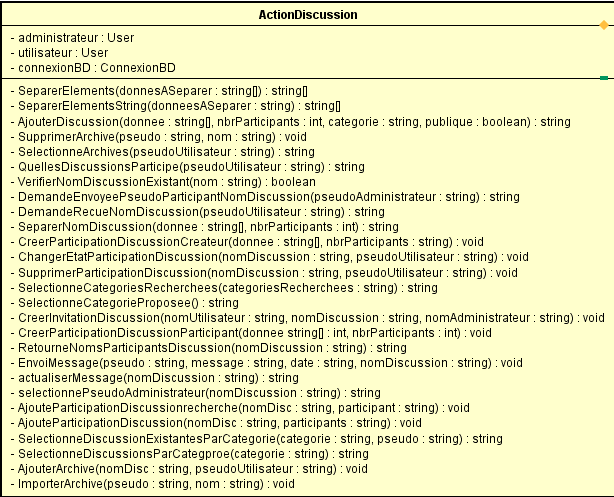


**Serveur :**

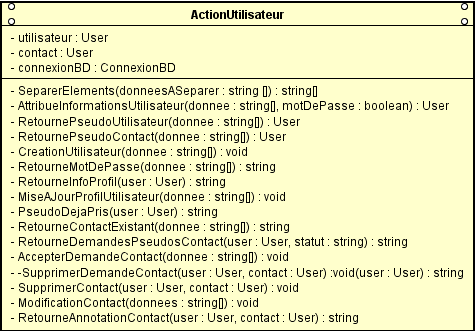
Program :



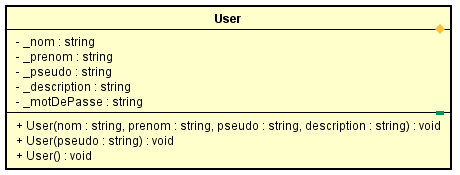
ActionDiscussion :



ActionUtilisateur :



User :



ConnexionBD :



## Erreurs restantes

* Sur le formulaire de connexion il existe une case à cocher « Se souvenir de moi ». Lorsqu’elle est cochée, l’identifiant doit être sauvegardé par l’application. Et lors de la prochaine utilisation de l’application, le champ « Identifiant » doit se remplir automatiquement.
  + Conséquence : L’utilisateur doit retaper son identifiant à chaque utilisation de l’application.
  + Actions possibles :
    - Les default settings de visual studio sont utilisés actuellement pour sauvegarder l’identifiant de l’utilisateur. Il faudrait que je travaille encore sur ma fonction pour comprendre pourquoi celle-ci ne se comporte pas comme je le souhaite (manque de temps).
    - Utiliser un fichier texte dans lequel le programme écrirait l’identifiant de l’utilisateur et irait lire ce fichier texte à l’ouverture de l’application pour récupérer cet identifiant.
* Si l’utilisateur saisit des données avec les caractères « spéciaux » (‘,’,’°’,’§’,’\*’, accents) le programme peut planter directement ou ultérieurement lorsqu’une autre action faisant appel à un de ses champs est utilisée. (Dans les messages l’utilisateur peut utiliser les virgules).
  + Conséquence : L’utilisateur n’a pas la possibilité de s’exprimer librement lorsqu’il rentre une annotation d’un contact, un nom de discussion, une catégorie de discussion, sa propre description. Le fait que les virgules posent problèmes dans certains cas est un gros problème parce qu’elles sont beaucoup utilisées dans la langue française (par exemple lorsque l’utilisateur fait une annotation sur un contact).
  + Actions possibles :
    - Il faudrait que l’insertion des données de l’utilisateur ne se fassent pas si certains caractères ont été rentrés. Et qu’un message d’erreur avertisse l’utilisateur de l’impossibilité d’utiliser ses caractères.
    - Actuellement le client envoie les données au serveur sous forme de string. Les différentes données dans un string sont séparées par des virgules ou d’autres caractères spéciaux. Il faudrait donc changer une grande partie du programme afin de ne plus utiliser du tout de caractères spéciaux pour séparer ses informations. Les utilisateurs auraient ainsi la possibilité d’utiliser les caractères qu’ils désirent.

## Liste des documents fournis

Les documents fournis se trouvent dans le dossier « Documents finaux » dans « Documentation » on y trouve un exécutable du client, un manuel d’installation, un manuel d’utilisation, un fichier contenant les tests effectués et ce document.

# Conclusions

A la fin de ce projet, 70% des objectifs ont été atteints :

* L’utilisateur peut créer des discussions de deux personnes ou plus.
* L’utilisateur a aussi la possibilité de rejoindre des discussions publiques sans avoir à recevoir une invitation pour la rejoindre.
* L’utilisateur retrouvera à chaque lancement du programme ses discussions précédentes.
* L’utilisateur peut archiver ses discussions, celles-ci n’apparaitront plus directement mais seront toujours disponible dans la section « Archive ».
* L’utilisateur peut ajouter/supprimer/modifier des contacts (possibilité d’ajouté une annotation sur le contact).

Il reste deux objectifs qui n’ont pas été du tout atteints :

* L’utilisateur ne peut pas utiliser des smileys ou des images dans les discussions.
* L’utilisateur ne peut pas utiliser de messages vocaux dans les discussions.

Le point positif et que l’utilisateur peut communiquer avec d’autres utilisateurs via un chat écrit et qu’il pourra retrouver ses communications lors de sa prochaine utilisation du programme.

Le point négatif est que l’utilisateur est limité dans ses moyens de communiquer avec d’autres utilisateurs, il ne peut utiliser qu’un moyen de communication écrit sans images. Et il doit faire attention à ne pas utiliser de caractères spéciaux dans certaines situations sans quoi le programme ne fonctionnera pas.

Il y a deux grosses difficultés principales dans ce projet : mon manque de connaissances sur la communication entre le client et le serveur et le fait que ce soit mon premier gros projet personnel.

En effet, ayant eu peur de ne pas réussir à finir tous les objectifs à temps, je n’ai pas choisi une solution très optimale dans la manière d’envoyer les informations entre le client et le serveur. Je pense que la méthode que j’ai utilisé (qui était la première que j’ai trouvé) m’a permis d’avancer rapidement dans le développement de mes premières fonctions mais m’a fait perdre beaucoup de temps par la suite.

Pour l’avenir de ce projet, il faudrait repenser à la manière d’envoyer les données entre le client et le serveur car c’est actuellement un gros handicap. Il faudrait aussi implémenter les smileys/images et les messages vocaux. Il faudrait rajouter sur le formulaire de connexion un champ dans lequel on rentre l’adresse IP du serveur afin de ne pas avoir à le changer dans le code à chaque fois que l’on change le serveur d’ordinateur. Et afficher quelques messages supplémentaires aux utilisateurs lorsqu’ils entreprennent certaines actions afin de leur faciliter l’utilisation du programme.

# Annexes

## Sources – Bibliographie

Pour la communication entre un client et un serveur :

* <https://msdn.microsoft.com/fr-fr/library/system.net.sockets.tcplistener(v=vs.110).aspx>
* <https://msdn.microsoft.com/fr-fr/library/system.net.sockets.tcpclient(v=vs.110).aspx>

Pour le hashage du mot de passe :

* <https://stackoverflow.com/questions/4181198/how-to-hash-a-password/10402129#10402129>

## Journal de travail

Il se trouve dans le dossier « Planification\_JournalDeTravail ».

## Manuel d'Installation

Il se trouve dans le dossier « Documents finaux » dans « Documentation ».

## Manuel d'Utilisation

Il se trouve dans le dossier « Documents finaux » dans « Documentation ».

## Archives du projet

Le projet est disponible sur GitHub à l’adresse suivante : <https://github.com/L-Gianinetti/Chat-Application>