|  |
| --- |
| Projet TPI  Chat en C# |

Table des matières

[1 Analyse préliminaire 3](#_Toc104296192)

[1.1 Introduction 3](#_Toc104296193)

[1.2 Objectifs 3](#_Toc104296194)

[1.3 Planification initiale 4](#_Toc104296195)

[2 Analyse / Conception 11](#_Toc104296196)

[2.1 Concept 11](#_Toc104296197)

[2.1.1 Environnement de travail 11](#_Toc104296198)

[2.1.2 Maquettes 12](#_Toc104296199)

[2.1.3 Uses cases scenarios 16](#_Toc104296200)

[2.1.4 Diagramme de flux 18](#_Toc104296201)

[2.1.5 Liste employées / Comptes et types de comptes 19](#_Toc104296202)

[2.1.6 Nomenclature 21](#_Toc104296203)

[2.2 Stratégie de test 23](#_Toc104296204)

[2.2.1 Test concernant le login : 23](#_Toc104296205)

[2.2.2 Test concernant le chat : 23](#_Toc104296206)

[2.2.3 Test concernant le modérateur : 24](#_Toc104296207)

[2.2.4 Test concernant l’administrateur : 24](#_Toc104296208)

[2.3 Risques techniques 25](#_Toc104296209)

[2.4 Planification 25](#_Toc104296210)

[2.5 Dossier de conception 25](#_Toc104296211)

[3 Réalisation 26](#_Toc104296212)

[3.1 Dossier de réalisation 26](#_Toc104296213)

[3.2 Description des tests effectués 28](#_Toc104296214)

[3.3 Erreurs restantes 29](#_Toc104296215)

[3.4 Liste des documents fournis 29](#_Toc104296216)

[4 Conclusions 30](#_Toc104296217)

[5 Annexes 31](#_Toc104296218)

[5.1 Résumé du rapport du TPI / version succincte de la documentation 31](#_Toc104296219)

[5.2 Sources – Bibliographie 31](#_Toc104296220)

[5.3 Journal de travail 31](#_Toc104296221)

[5.4 Archives du projet 31](#_Toc104296222)

*NOTE L’INTENTION DES UTILISATEURS DE CE CANEVAS:  
Toutes les parties en italiques sont là pour aider à comprendre ce qu’il faut mettre dans cette partie du document. Elles n’ont donc aucune raison d’être dans le document final.*

*De plus, en fonction du type de projet, il est tout à fait possible que certains chapitres ou paragraphes n’aient aucun sens. Dans ce cas il est recommandé de les retirer du document pour éviter de l’alourdir inutilement.*

# Analyse préliminaire

## Introduction

Ce projet a pour but de réaliser une application de chat dans le domaine de l’entreprise.

L’application Chat sera développée en C# avec une base de données SQL. Cette application sera dédiée en live à la communication entre collaborateurs d’une entreprise.

Cette application a pour but de permettre au personnel d’une entreprise ou institution de pouvoir communiquer en direct au sein de leur réseau interne

Elle pourra faciliter la communication entre les différents collègues et ainsi faciliter leur mode de travail en équipe.

Afin de pouvoir commencer à utiliser l’application, il est impératif que l’utilisateur se connecte avec son nom d’utilisateur (son pseudonyme) et son mot de passe. Seul le personnel de l’entreprise a accès au Chat.

En 4ème année du CPNV, je réalise ce projet dans le cadre du TPI de fin d’année.

J’ai choisi ce projet pour le domaine dans lequel il va être réalisé : la programmation.

Ce projet va me permettre d’enrichir mes connaissances en développement C# ainsi que les différentes manipulations avec la base de données.

Mon chef de projet est M. Benzonana.

Le projet va être réalisé dans la méthode agile.

Les modules effectués dans le domaine du C# sont des prérequis pour pouvoir réaliser ce projet ainsi que les modules de base de données (SQL).

Un projet en entreprise lorsque j’étais en stage chez Valtronic a été réalisé dans le domaine du C# en travaillant avec une base de données. Elle était développée plus précisément pour faire du CRUD[[1]](#footnote-1). Cela va être d’une grande aide pour le TPI.

## Objectifs

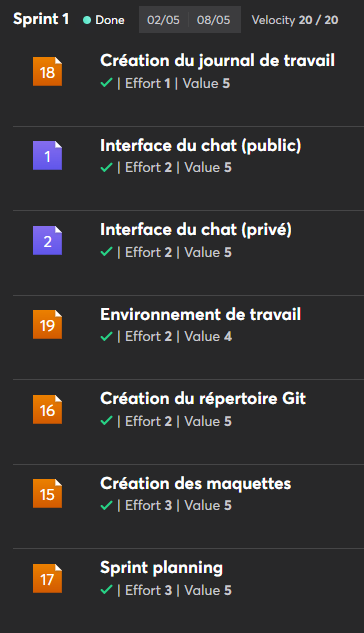
* Conversation publique
  + Toutes les personnes peuvent voir les conversations publiques
* Conversation privée
  + Seules les personnes ayant accès au canal privé peuvent voir la conversation de celle-ci
* Envoyer Message
  + Toutes les personnes peuvent envoyer des messages dans le canal public
  + Les personnes peuvent envoyer des messages dans le canal privé seulement dans celles dans lesquels elles ont accès
* Modifier son message
  + Toutes les personnes peuvent modifier leurs messages
* Supprimer son message
  + Toutes les personnes peuvent supprimer leurs messages
* Supprimer un message (Modérateur et Admin)
  + Seules les personnes possédant un compte de type « Modérateur » ou « Administrateur » peuvent supprimer n’importe quel message de n’importe qui
* Gérer les collaborateurs et les droits (Admin)
  + Les personnes dont leur compte est de type « Administrateur » possèdent un menu en plus qui est « Gestion des collaborateurs »
  + Les administrateurs peuvent ajouter, supprimer et modifier un collaborateur

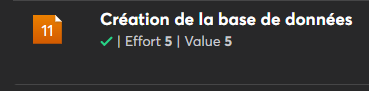
## Planification initiale

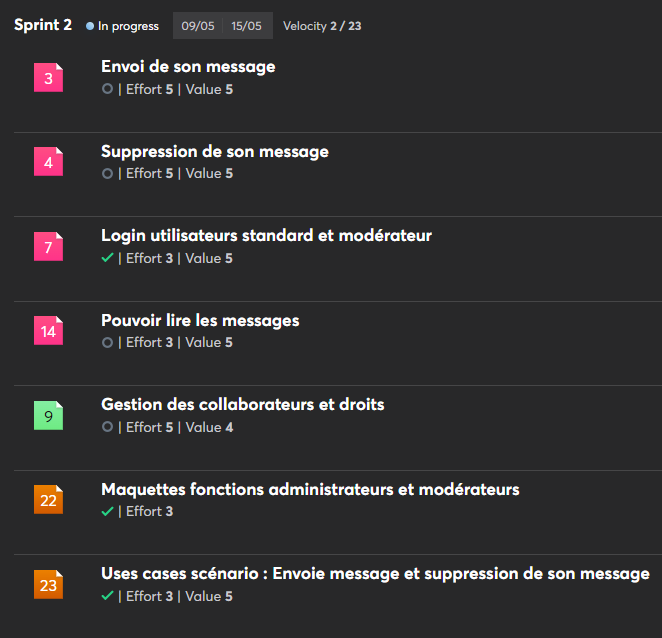
**Sprint Goal**

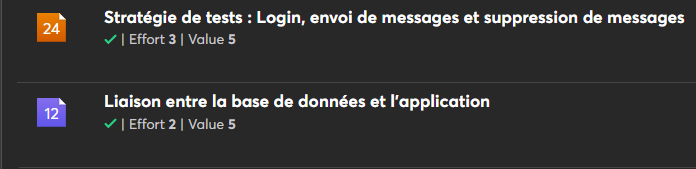
Les sprints se dérouleront comme au-dessus. Nous aurons les différentes tâches à faire.

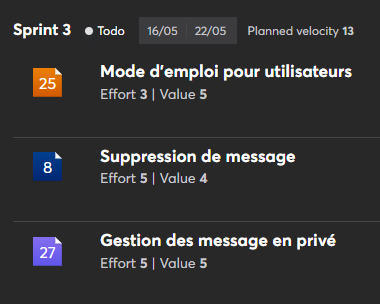
**Planification iceScrum :**

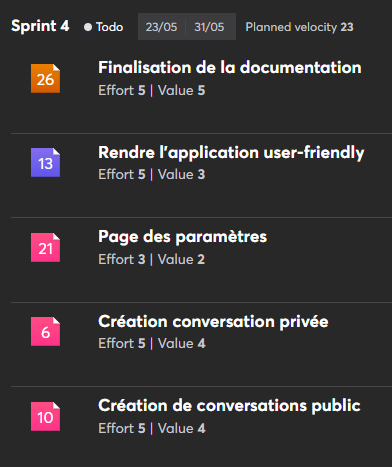












**Sprint 1 / 02.05.2022 – 06.05.2022 :**

Documentation :

* Introduction
* Objectifs
* Plannification initiale
* Maquette de l’application

IceScrum :

* Plannification initiale

Programmation :

* Interface du chat (public)
* Interface du chat (privé)
* Liaison entre la base de données et l’application

Base de données :

* Création de la base de données
* Création des tables
* Liaison entre les tables

Création des différents fichier / répertoires :

* Création du journal de travail (fichier modèle)
* Création du répertoire Git[[2]](#footnote-2)
* Création du iceScrum[[3]](#footnote-3)
* Création du projet Visual Studio[[4]](#footnote-4)

**Sprint 2 / 09.05.2022 – 13.05.2022 :**

Documentation :

* Uses cases scenarios
* Listes utilisateurs de l’application

Programmation :

* Login
* Pouvoir lire les messages
* Envoi de son message
* Suppression de son message
* Gestion des collaborateurs et droits

Base de données :

* Gestion des messages en public

**Sprint 3 / 16.05.2022 – 20.05.2022 :**

Documentation :

* Mode d’emploi pour utilisateurs

Programmation :

* Modification de son message
* Suppression de message

Base de données :

* Gestion des messages en privé

**Sprint 4 / 23.05.2022 – 30.05.2022 :**

Documentation :

* Finalisation de la documentation

Programmation :

* Création conversation privée
* Création de conversations public
* Rendre l’application user-friendly

# Analyse / Conception

## Concept

### Environnement de travail

L’environnement sous lequel le projet a été réalisé :

Système d’exploitation :

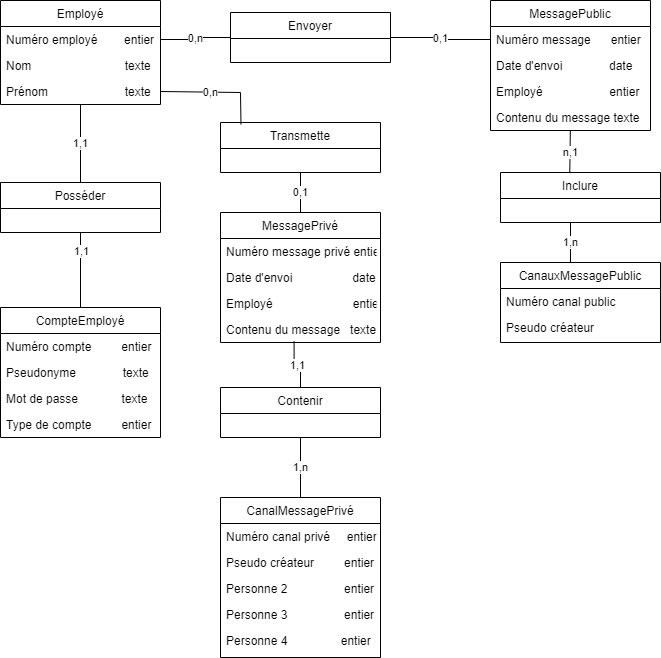
* Édition : Windows 10 Éducation
* Version : 21H2
* Build du système d’exploitation : 19044.1645

Les logiciels qui seront mis à dipositions :

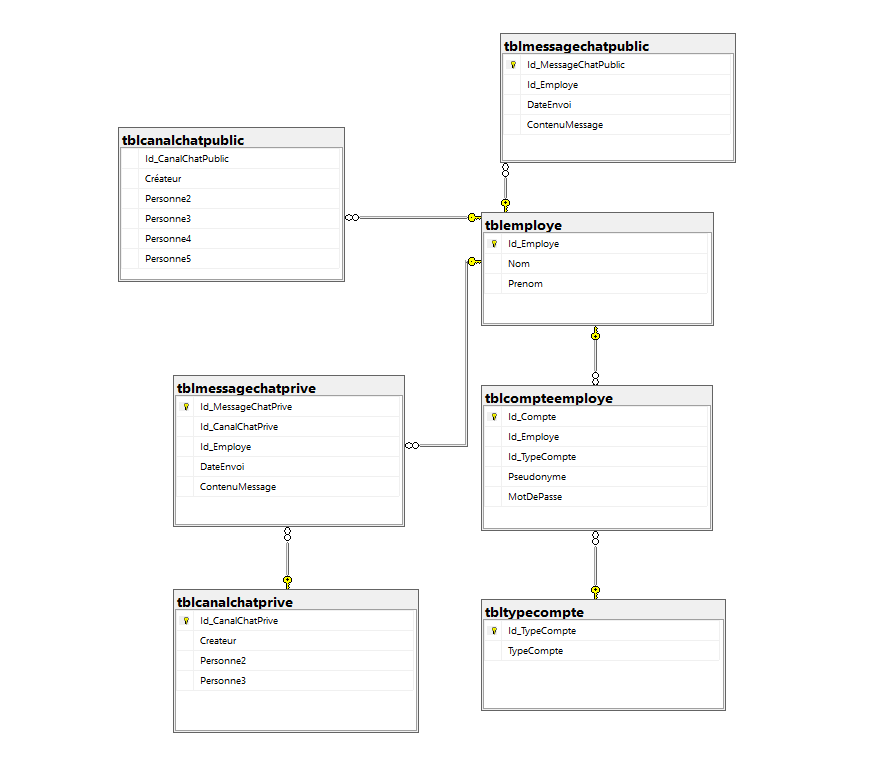
* Visual Studio Entreprise 19 / Version 16.4.2
* MockFlow (version internet)
* Suite office
* Microsoft SQL Server Management 18

### Modélisation de la base de données :

**MCD :**



**MLD :**



### Maquettes

Maquette pour le page du chat public :

**5**

**6**

**4**

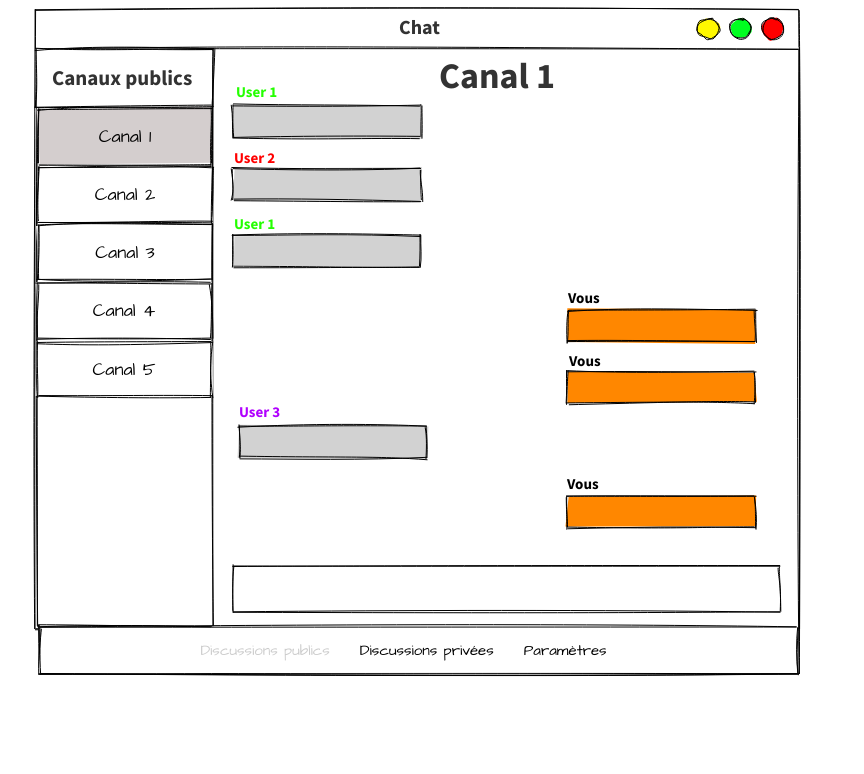
**8**

**7**

**2**

**3**

**1**



1 : La fenêtre du programme

2 : Cette section regroupe tous les canaux publics accessibles

3 : Cette section contient le nom d’utilisateur des personnes auxquelles vous parlez (la couleur varie selon les personnes auxquelles vous parlez).

4 : Cette section contient les messages des autres utilisateurs.

5 : Cette section contient votre nom d’utilisateur.

6 : Cette section contient vos messages.

7 : Cette section sert à écrire votre message.

8 : Cette section est le menu de l’application.

Maquette pour la page du chat privé :

**8**

**7**

**6**

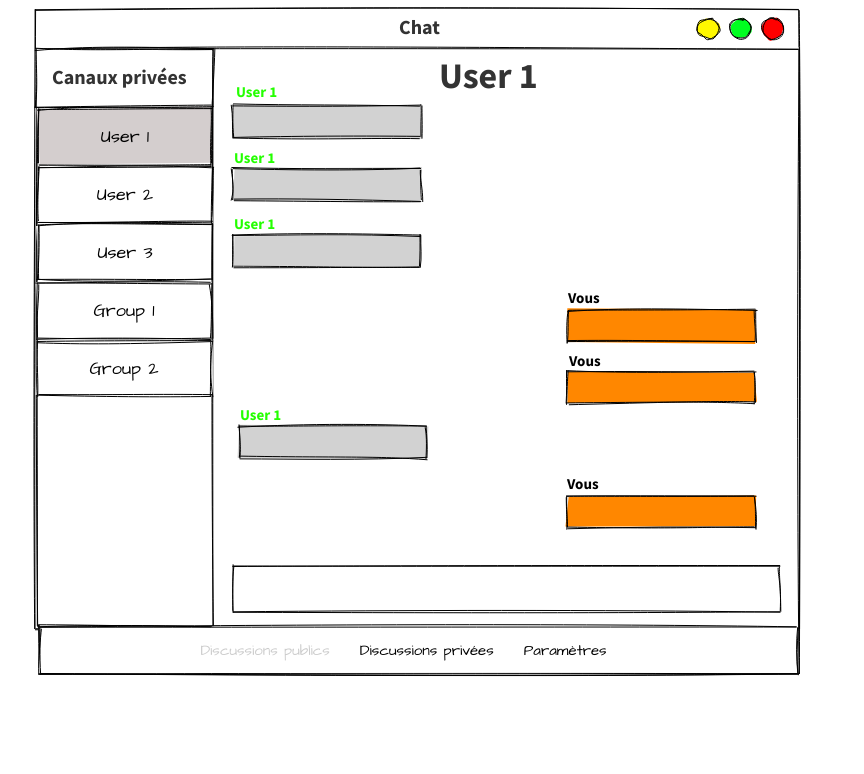
**3**

**2**

**5**

**4**

**1**



1 : La fenêtre du programme

2 : Cette section regroupe tous les canaux privés accessibles

3 : Cette section contient le nom d’utilisateur de la personne à laquelle vous parlez.

4 : Cette section contient les messages de la personne à laquelle vous parlez.

5 : Cette section contient votre nom d’utilisateur.

6 : Cette section contient vos messages.

7 : Cette section sert à écrire votre message.

8 : Cette section est le menu de l’application.

Maquette pour les fonctions modératrices :

**8**

**6**

**5**

**9**

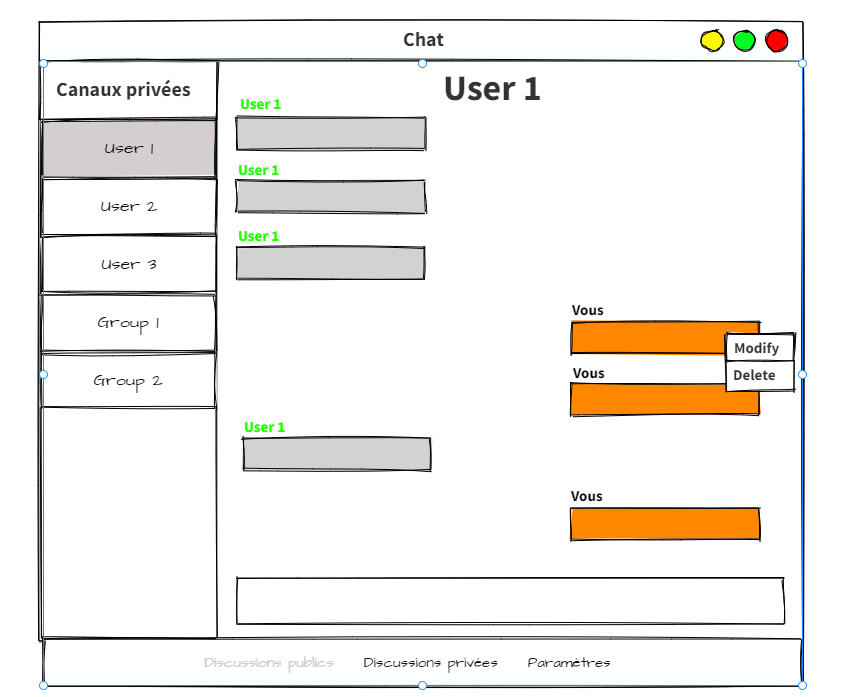
**7**

**4**

**2**

**3**

**1**



Ces fonctions des utilisateurs standard de base peuvent être utilisés dans tous les fils de discussions. Pour l’exemple ci-dessus, j’ai décidé de le faire en discussion privée.

1 : La fenêtre du programme

2 : Cette section regroupe tous les canaux privés accessibles

3 : Cette section contient le nom d’utilisateur de la personne à laquelle vous parlez.

4 : Cette section contient les messages de la personne à laquelle vous parlez.

5 : Cette section contient votre nom d’utilisateur.

6 : Cette section contient vos messages.

7 : Ce petit onglet survient lorsque l’on appuie avec le clic droit de la souris sur son propre message. On a la possibilité de le supprimer ou de le modifier.

8 : Cette section sert à écrire votre message.

9 : Cette section est le menu de l’application.

Maquette pour la gestion des compte des collaborateurs :

**8**

**7**

**6**

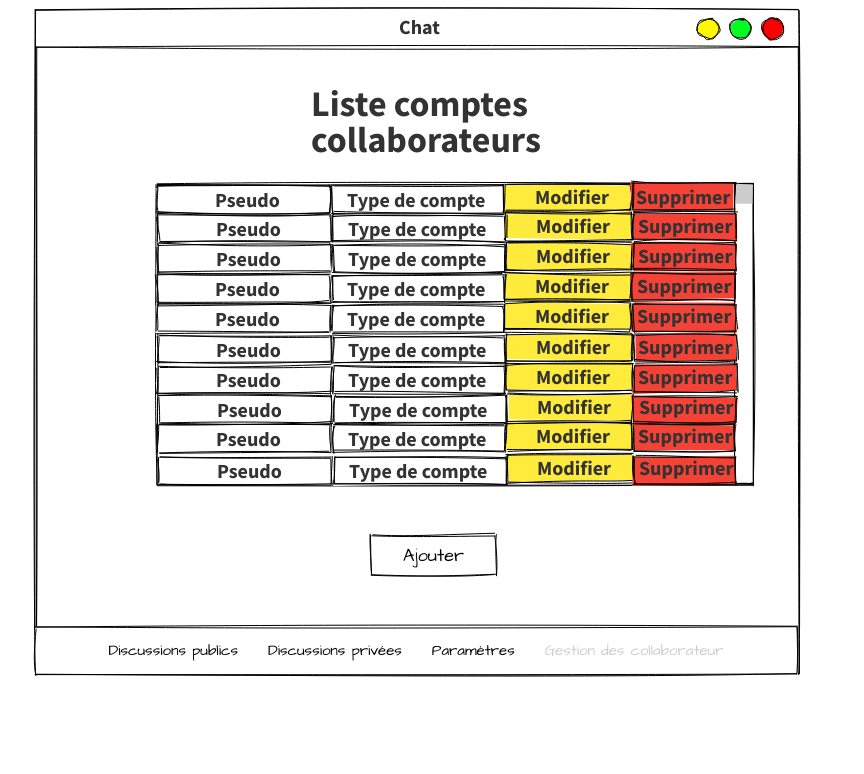
**5**

**4**

**3**

**2**

**1**



1 : La fenêtre du programme

2 : Cette section contient la liste des comptes des collaborateurs

3 : Cette section contient le nom d’utilisateur du collaborateur.

4 : Cette section contient le type de compte du collaborateur.

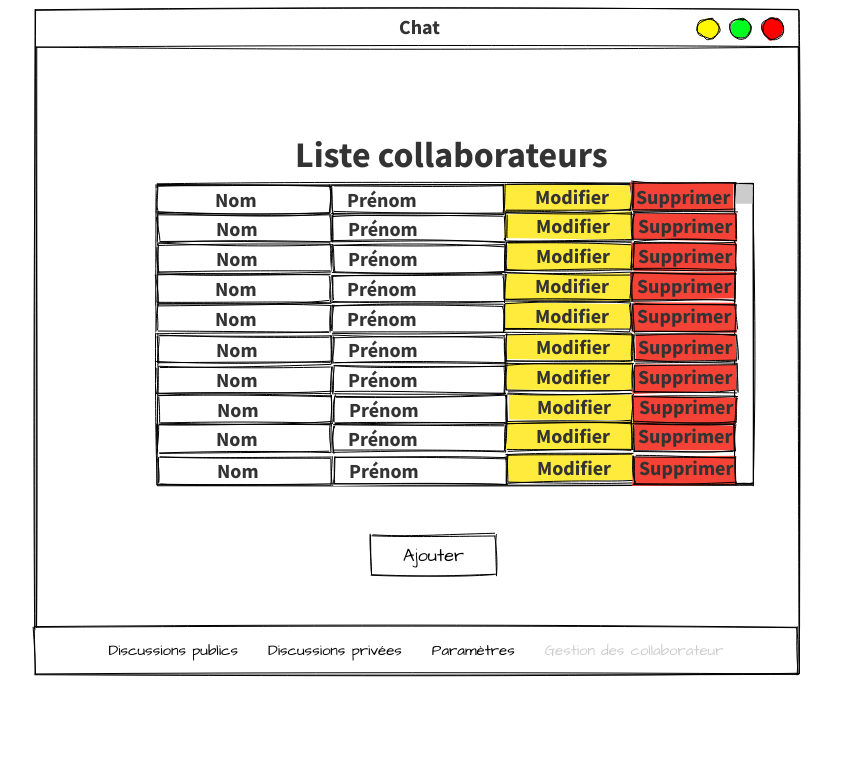
5 : Cette section est un bouton permettant de modifier le compte du collaborateur.

6 : Cette section est un bouton permettant de supprimer le compte du collaborateur.

7 : Cette section est un bouton permettant d’ajouter (de créer) un compte collaborateur.

8 : Cette section est le menu de l’application.

Maquette pour la gestion des collaborateurs :



**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

1 : La fenêtre du programme.

2 : Cette section contient la des collaborateurs.

3 : Cette section contient le nom du collaborateur.

4 : Cette section contient le prénom du collaborateur.

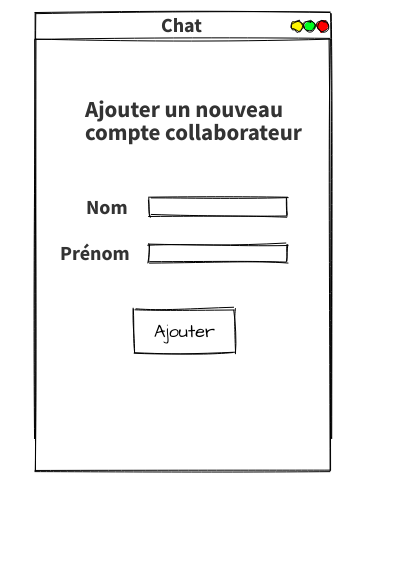
5 : Cette section est un bouton permettant de modifier le compte du collaborateur.

6 : Cette section est un bouton permettant de supprimer le compte du collaborateur.

7 : Cette section est un bouton permettant d’ajouter un nouveau collaborateur.

8 : Cette section est le menu de l’application.

Maquette pour l’ajout d’un nouveau collaborateur :



**4**

**3**

**2**

**1**

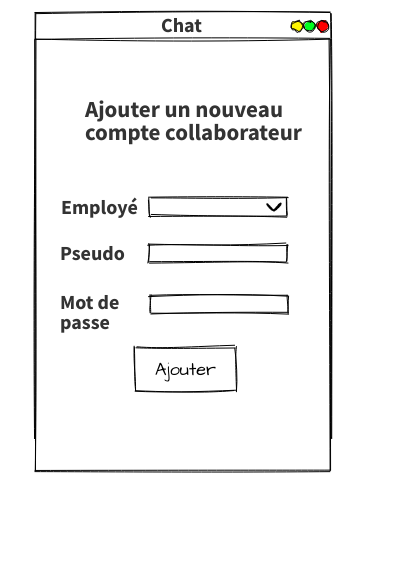
1 : La fenêtre du programme.

2 : Il faut remplir cette section en insérant le nom du nouvel(le) employé(e).

3 : Il faut remplir cette section en insérant le prénom du nouvel(le) employé(e).

4 : Cette section permet de valider tous les champs insérés puis ajouter le nouveau collaborateur dans la base de données.

Maquette pour l’ajout d’un nouveau compte collaborateur :



**5**

**4**

**3**

**2**

**1**

1 : La fenêtre du programme.

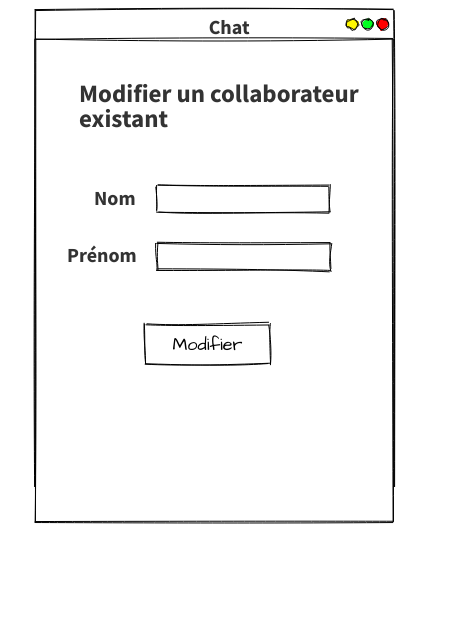
2 : Il faut remplir cette section en sélectionnant l’employé qui sera assigné au compte.

3 : Il faut remplir cette section en insérant le pseudonyme qui sera assigné au compte.

4 : Il faut cette section en insérant le mot de passe qui sera assigné au compte.

5 : Cette section permet de valider tous les champs insérés puis ajouter le nouveau collaborateur dans la base de données.

Maquette pour la modification d’un collaborateur existant :



**4**

**3**

**2**

**1**

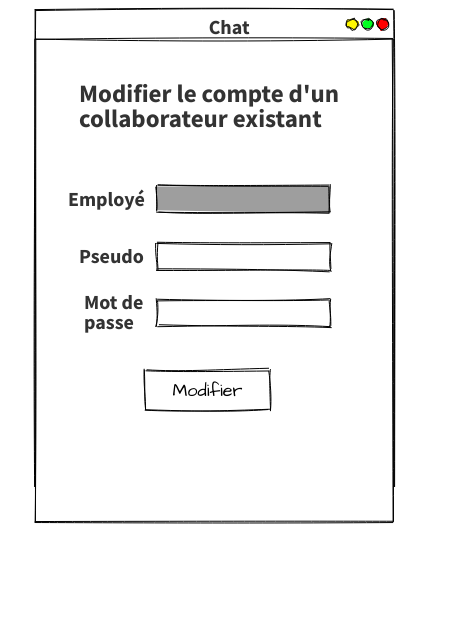
1 : La fenêtre du programme

2 : Cette section contiendra le nom de l’employé choisi

3 : Cette section contiendra le prénom de l’employé choisi

4 : Ce bouton permettra de modifier les données s’il y a eu du changement.

Maquette pour la modification du compte d’un collaborateur :



**5**

**4**

**3**

**2**

**1**

1 : La fenêtre du programme

2 : Cette section contiendra le nom et le prénom de l’employé choisi (non modifiable)

3 : Cette section contiendra le pseudo de l’employé choisi

4 : Cette section contiendra le mot de passe de l’employé choisi

4 : Ce bouton permettra de modifier les données s’il y a eu du changement.

### Uses cases scenarios

**Lancement du programme**

Uses cases du lancement de l’application :

|  |  |  |
| --- | --- | --- |
| Identifiant | Lancement du programme |  |
| En tant que | Utilisateur standard / Utilisateur modérateur |  |
|  |  |  |
| **Action** | **Conditions** | **Réaction** |
| Lancer le programme |  | Le programme se lance |
|  | La page du login s'affiche |
| Se connecter |  | La page du chat public avec 3 menus s'affiche |

|  |  |  |
| --- | --- | --- |
| Identifiant | Lancement du programme |  |
| En tant que | Utilisateur administrateur |  |
|  |  |  |
| **Action** | **Conditions** | **Réaction** |
| Lancer le programme |  | Le programme se lance |
|  | La page du login s'affiche |
| Se connecter |  | La page du chat public avec 4 menus s'affiche |

**Message**

Uses cases scénario de l’envoi d’un message dans le fil de discussion public :

|  |  |  |  |
| --- | --- | --- | --- |
| Identifiant | Envoi de message | |  |
| En tant que | Utilisateur standard / Utilisateur Modérateur / Utilisateur Administrateur | |  |
|  |  |  | |
| **Action** | **Condition** | **Réaction** | |
| Appuyer sur "Écrire un message ici" |  |  | |
| Écrire un message |  |  | |
| Appuyer sur "Envoyer" | Si le message est offensant | Un message apparaît est précise que le message est inadéquat | |
| Si le message respecte toutes les règles | Le message s'envoi dans la base de données | |
|  | Le message apparaît dans le fil de discussion | |

Uses case scénario suppression de son message :

|  |  |  |
| --- | --- | --- |
| Identifiant | Supprimer son message |  |
| En tant que | Utilisateur standard / Utilisateur modérateur / Utilisateur administrateur |  |
|  |  |  |
| **Action** | **Condition** | **Réaction** |
| Clic droit sur son message |  | Un petit onglet s'affiche |
| Cliquer sur supprimer le message |  | Le message se supprime |

Uses case scénario modification de son message :

|  |  |  |
| --- | --- | --- |
| Identifiant | Modifier son message |  |
| En tant que | Utilisateur standard / Utilisateur modérateur / Utilisateur administrateur |  |
|  |  |  |
| **Action** | **Condition** | **Réaction** |
| Clic droit sur son message |  | Un petit onglet s'affiche |
| Cliquer sur modifier le message |  | Le champ pour écrire un message se remplit avec le message sélectionné |
|  | Le bouton "Ajouter" sera remplacé par "Modifier" |
| Modifier son texte |  |  |
| Appuyer sur le bouton "Modifier" |  | La modification sera envoyé à la base de données |

**Gestion administrateur**

Uses cases scénario ajout d’un collaborateur :

|  |  |  |
| --- | --- | --- |
| Identifiant | Ajout d'un collaborateur |  |
| En tant que | Utilisateur administrateur |  |
|  |  |  |
| **Action** | **Condition** | **Réaction** |
| Aller dans la page des gestions de collaborateurs |  |  |
| Remplir le champ nom |  |  |
| Remplir le champ prénom |  | Le bouton "Ajouter" sera remplacé par "Modifier" |
| Cliquer sur le bouton "Ajouter" |  |  |
|  |  | La modification sera envoyée à la base de données |

Uses cases scénario modification d’un collaborateur :

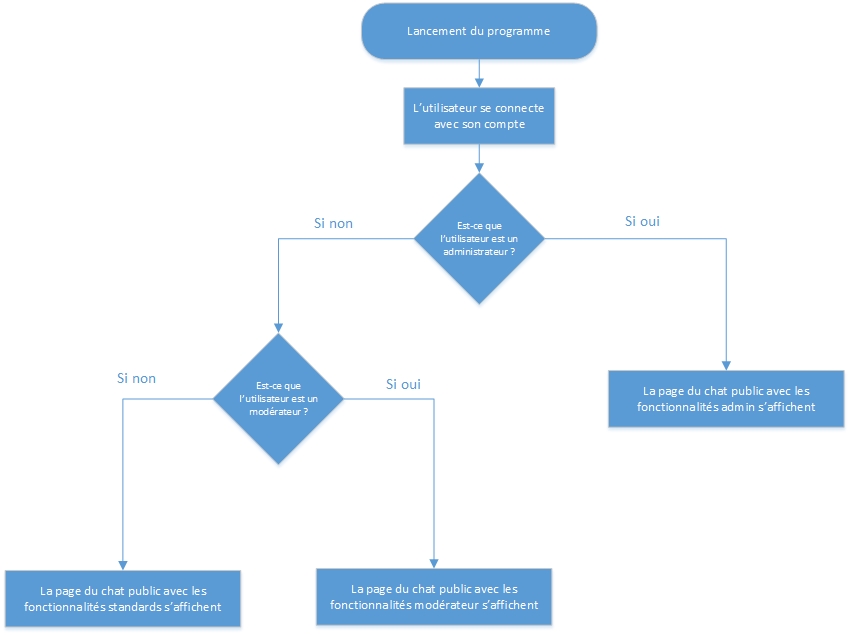
|  |  |  |
| --- | --- | --- |
| Identifiant | Modification d'un collaborateur |  |
| En tant que | Utilisateur administrateur |  |
|  |  |  |
| **Action** | **Condition** | **Réaction** |
| Double clic sur la ligne que l'on veut changer |  | Les champs "nom" et "prénom" se remplissent par la ligne sélectionnée |
| Modifier les champs |  |  |
| Cliquer sur le bouton "Modifier" |  | La modification s'effectue dans la base de données |
|  | Le tableau affiche la modification |

Uses cases scénario suppression d’un collaborateur :

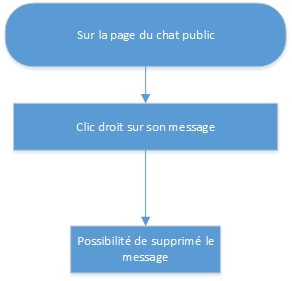
|  |  |  |
| --- | --- | --- |
| Identifiant | Suppression d'un collaborateur |  |
| En tant que | Utilisateur administrateur |  |
|  |  |  |
| **Action** | **Condition** | **Réaction** |
| Double clic sur la ligne que l'on veut changer |  | Les champs "nom" et "prénom" se remplissent par la ligne sélectionnée |
| Cliquer sur le bouton "Supprimer |  | La suppression s'effectue dans la base de données |
|  | Le tableau n'affiche plus la donnée choisie |

### Diagramme de flux

Pour le lancement du programme :



Le diagramme de flux pour la suppression de son message :



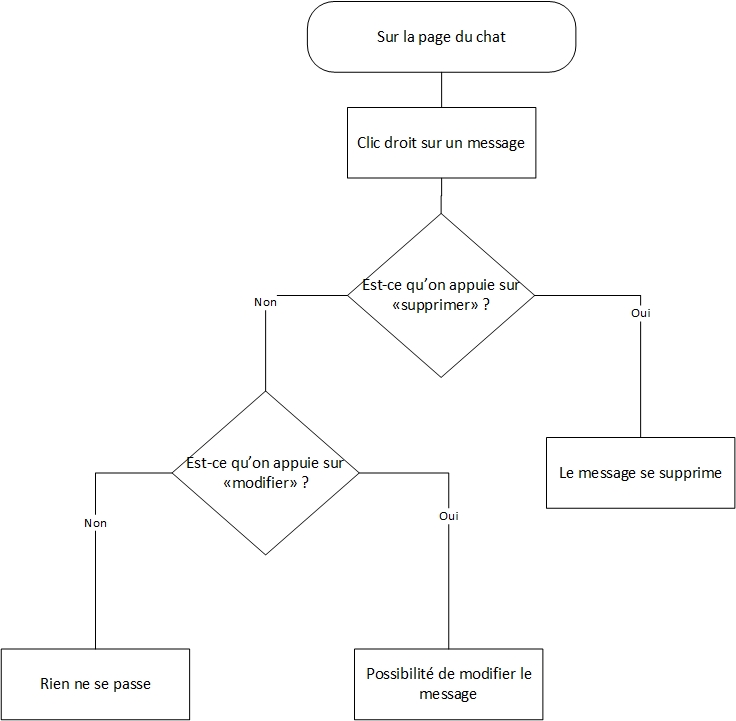
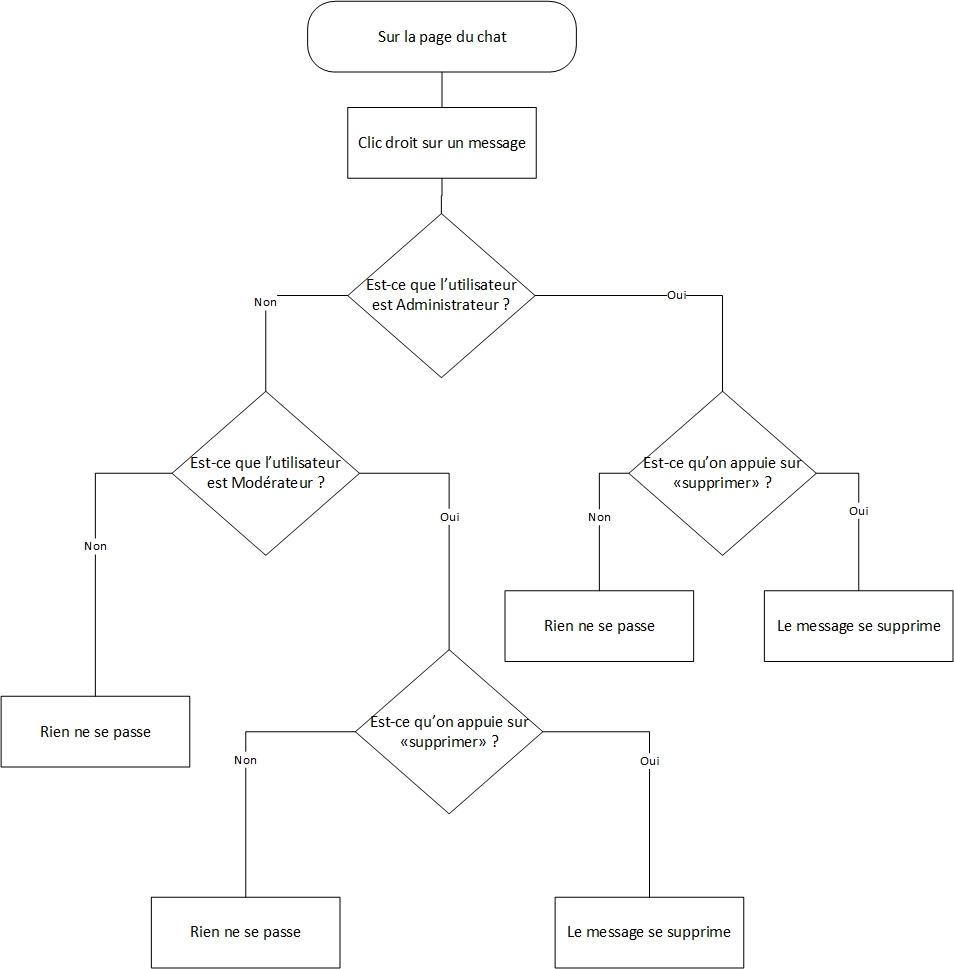
Le diagramme de flux pour la suppression de son message 

Diagramme de flux pour les fonctions modérateurs concernant les messages :



### Liste employées / Comptes et types de comptes

Ensuite, nous allons voir les différentes informations concernant les employées et leur compte.

Voici la liste des employées créés :

* Faustin Bergeron
* Eliot Primeau
* Eugène Alexandre
* Auriville Lebrun
* Senapus Brunelle
* Mercer Veilleux
* Charles Ferland
* Victorine Chabot
* Fabrice Blais
* Arianne Bizier
* Marine Gauvin
* Caroline Parmentier
* Brunella Poisson
* Gauthier Desnoyer
* Noelle Laboissonnière

Chacun d’eux a un compte et un type de compte.

Voici les types de comptes :

* Standard
* Modérateur
* Administrateur

Il y a 2 administrateurs et 2 modérateurs.

Le reste des utilisateurs sont standards.

Les administrateurs :

* Eugène Alexandre
* Auriville Lebrun

Les modérateurs :

* Faustin Bergeron
* Fabrice Blais

Les utilisateurs standards peuvent seulement envoyer des messages, supprimer leurs propres messages, voir les messages et créer des fils de discussions privées.

Les utilisateurs qui sont modérateurs ont toutes les fonctionnalités des utilisateurs standards et peuvent aussi supprimer les messages d’autrui. Cela leur permet de pouvoir modérer leurs discussions et ainsi avoir une bonne cohésion.

Les utilisateurs qui sont administrateurs ont toutes les fonctionnalités de utilisateurs standards et modérateurs et peuvent aussi gérer les collaborateurs. Ils peuvent ajouter, supprimer et modifier la liste des collaborateurs qui ont accès au chat. Ils peuvent aussi gérer les droits des utilisateurs.

Pour les comptes, les pseudonymes sont générés de 3 lettres. Ce sera l’acronyme de l’employé. Pour les acronymes, nous prenons la première lettre du prénom, puis la première et dernière lettre du nom de famille.

Voici les pseudonymes :

* Faustin Bergeron - FBN
* Eliot Primeau - EPU
* Eugène Alexandre - EAE
* Auriville Lebrun - ALN
* Senapus Brunelle - SBE
* Mercer Veilleux - MVX
* Charles Ferland - CFD
* Victorine Chabot - VCT
* Fabrice Blais - FBS
* Arianne Bizier - ABR
* Marine Gauvin - MGN
* Caroline Parmentier - CPR
* Brunella Poisson - BPN
* Gauthier Desnoyer - GDR
* Noelle Laboissonnière - NLE

### Ergonomie de l’application

L’application doit avoir un minimum d’ergonomie pour qu’il soit exploitable sans trop de perte de temps (perte de temps qui peuvent être provoquée par des bugs ou le choix de la mise en place de certaines fonctionnalités).

Le minimum requis serait :

* Le menu sera composé de trois boutons simples en bas de chaque fenêtre
* Le moins d’interaction possible avec la souris (c’est-à-dire pouvoir utiliser le clavier quand on le peut)
* Les champs textes doivent pouvoir être validés avec le bouton « enter » (cette condition est liée à celle d’au-dessus)
* Les boutons pour qui envoient une requête comme le bouton « se connecter » à la page de login ou « envoyer » pour envoyer un message doivent pouvoir être appuyés à l’aide du bouton « enter » du clavier.
* Chaque zone de chaque fenêtres doit être clairement définies et facilement compréhensibles pour l’utilisateur.

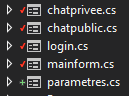
### Nomenclature

Différentes nomenclatures vont être utilisés au long de ces projets. Et cela dans plusieurs domaines.

Commençons par la programmation.

Tout d’abord nous avons les différents « formulaires ». C’est les pages nécessaires pour l’application.

Voici les différentes pages :



Donc ici, il n’y a pas de nomenclatures particulières. J’ai juste choisi que les noms seront toujours en minuscules.

Pour les items :

* Boutons : btnNomdubouton
* TextBox : txtNomdutextBox
* Labels : lblNomdulabel
* Listbox : listbox
* DataGridView : NomdelatableGridView
* ComboBox : NomdelalisteBox

Pour SQL :

* Tblnomdelatable
* Id\_Nomdeid

## Stratégie de test

### Test concernant le login :

Pour prouver que le login est fonctionnel :

* Test avec un compte standard
* Test avec un compte modérateur
* Test avec un compte administrateur
* Test avec un compte non existant

La page affichée selon le login :

* Test avec un compte standard
* Test avec un compte modérateur
* Test avec un compte administrateur
* Test avec un compte non existant

La page administrateur d’affiche pour la bonne personne :

* Test avec un compte administrateur

### Test concernant le chat :

Pour prouver que le message est bien envoyé :

* Le message doit être inséré dans la base de données.
* Le message s’affiche dans le canal avec la date de l’envoie et le nom de l’expéditeur.

Pour prouver que les utilisateurs peuvent bien envoyer un message :

* Test avec tous les types de comptes qu’on puisse bien envoyer un message
* Vérifier que l’envoi se fait bien et qu’on puisse le lire dans le fil de discussion

Pour prouver que chaque utilisateur peut supprimer son propre message :

* Clic droit et vérifier si l’option « supprimer ce message » apparaît
* Vérifier si le message se supprime en cliquant sur l’option
* Vérifier que le message se supprime également dans la base de données

Pour prouver que chaque utilisateur peut modifier son propre message :

* Clic droit et vérifier si l’option « modifier ce message » apparaît
* Vérifier si l’onglet de modification du message apparaît
* Vérifier si le message se modifie dans le fil de discussion
* Vérifier si le message se modifie bien dans la base de données

### Test concernant le modérateur :

Pour prouver le modérateur a les droits de gestions des messages :

* Tester s’il peut supprimer n’importe quel message de n’importe qui

### Test concernant l’administrateur :

Pour prouver que l’administrateur a les droits de gestion des messages :

* Tester s’il peut supprimer n’importe quel message de n’importe qui

Pour prouver que l’administrateur peut gérer les collaborateurs dans la liste de celles-ci :

* Tester s’il peut ajouter un collaborateur
* Tester s’il peut modifier un collaborateur
* Tester s’il peut supprimer un collaborateur

Pour prouver que l’administrateur peut gérer les comptes des collaborateurs dans la liste de celles-ci :

* Tester s’il peut ajouter un collaborateur
* Tester s’il peut modifier un collaborateur
* Tester s’il peut supprimer un collaborateur
* Tester s’il peut gérer les rôles d’un compte collaborateur

### Test avec deux applications ouvertes simultanément :

Pour prouver que le message s’affiche bien :

* Tester si les messages s’affichent bien

Pour prouver que le message s’envoie bien :

* Tester l’envoi du message
* S’assurer que le message est bien présent dans l’application depuis lequel il est envoyé
* S’assurer que l’on voit aussi dans la deuxième application

Pour prouver que le message se supprime bien :

* Tester que la suppression de message
* S’assurer que le message se supprime dans l’application depuis lequel il est envoyé
* S’assurer qu’on ne le voit plus dans la deuxième application

Pour prouver que le message se modifie bien :

* Tester que l’on peut modifier un message
* Tester que le message se modifie bien dans la base de données
* Tester que le message modifié soit bien affiché

## Risques techniques

* Risque au niveau de l’affichage des messages. Idée très floue de comment les afficher.
  + Solution appliquée : Recherche de comment utiliser les listbox en C#. Toute l’affichage est faite grâce à cela.
* Risque au niveau du création du canal privé. Faciliter les utilisateurs en mettant le nom des personnes ce qui est assez compliqué.
  + Il faudrait essayer de travailler avec les INNER JOIIN, LEFT JOIN, etc…

## Planification

*Révision de la planification initiale du projet :*

* *planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.*
* *partage des tâches en cas de travail à plusieurs.*

*Il s’agit en principe de la planification* ***définitive du projet****. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l’historique.*

## Dossier de conception

Pour tous les matériaux utilisés, le poste du travail du CPNV suffisait largement pour la réalisation de ce TPI.

Système d’exploitation :

* Édition : Windows 10 Éducation
* Version : 21H2
* Build du système d’exploitation : 19044.1645

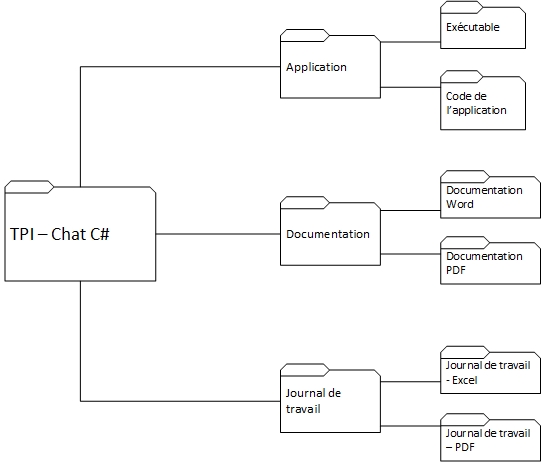
Logiciels utilisés :

* Microsoft SQL Server Management Studio 18
* Visio Professional 2021
* Visual Studio 19
* MockFlow (Site web pour créer des maquettes)

# Réalisation

## Dossier de réalisation

Voici les différents dossiers que l’on peut trouver sur GitHub :



## Description des tests effectués

Test concernant le login effectué le 16.05.2022 :

|  |  |  |
| --- | --- | --- |
| Identifiant | Login |  |
|  |  |  |
| **Action** | **Réussi ?** | **Remarques** |
| Se connecter avec un compte standard | Réussi |  |
| Se connecter avec un compte Modérateur | Réussi |  |
| Se connecter avec un compte Administrateur | Réussi |  |
| Lorsque l'admin se connecte, l'interface est différente des autres types de comptes | Échec | L'interface est la même que pour les autres types de comptes. L'administrateur doit avoir un menu en plus. |

Test concernant l’affichage des messages effectué le 24.05.2022 :

|  |  |  |
| --- | --- | --- |
| Identifiant | Affichage les messages (chat public) |  |
|  |  |  |
| **Action** | **Réussi ?** | **Remarques** |
| Les messages s'affichent avec un compte standard | Réussi |  |
| Les messages s'affichent avec un compte modérateur | Réussi |  |
| Les messages s'affichent avec un compte administrateur | Réussi |  |

Test concernant l’envoi de message effectué le 24.05.2022 :

|  |  |  |
| --- | --- | --- |
| Identifiant | Envoi message (chat public) |  |
|  |  |  |
| **Action** | **Réussi ?** | **Remarques** |
| Envoi d'un message avec un compte standard | Réussi |  |
| Envoi d'un message avec un compte modérateur | Réussi |  |
| Envoi d'un message avec un compte Administrateur | Réussi |  |

Test concernant la suppression de son propre message effectué le 24.05.2022 :

|  |  |  |
| --- | --- | --- |
| Identifiant | Suppression de son propre message (chat public) |  |
|  |  |  |
| **Action** | **Réussi ?** | **Remarques** |
| Suppression de son message avec un compte standard | Réussi |  |
| Suppression de son message avec un compte modérateur | Réussi |  |
| Suppression de son message avec un compte administrateur | Réussi |  |

Test concernant le modérateur effectué le 24.05.2022 :

|  |  |  |
| --- | --- | --- |
| Identifiant | Fonctions modérateurs |  |
|  |  |  |
| **Action** | **Réussi ?** | **Remarques** |
| Le modérateur peut supprimer tous les messages | Réussi |  |

Test concernant la modification de son propre message effectué le 24.05.2022 :

|  |  |  |
| --- | --- | --- |
| Identifiant | Modification de son message |  |
|  |  |  |
| **Action** | **Réussi ?** | **Remarques** |
| Modification avec un compte standard | Échec | Pas eu le temps d'implémenter |
| Modification avec un compte modérateur | Échec | Pas eu le temps d'implémenter |
| Modification avec un compte administrateur | Échec | Pas eu le temps d'implémenter |

Test concernant la gestion des collaborateurs effectué le 24.05.2022 :

|  |  |  |
| --- | --- | --- |
| Identifiant | Gestion de collaborateurs |  |
|  |  |  |
| **Action** | **Réussi ?** | **Remarques** |
| Les administrateurs peuvent ajouter un collaborateur | Réussi |  |
| Les administrateurs peuvent modifier un collaborateur | Réussi |  |
| Les administrateurs peuvent supprimer un collaborateur | Réussi |  |

Test concernant la création d’un canal privé effectué le 24.05.2022 :

|  |  |  |
| --- | --- | --- |
| Identifiant | Création de canal |  |
|  |  |  |
| **Action** | **Réussi ?** | **Remarques** |
| Les utilisateur peuvent créer un canal | Réussi |  |
| Le système ne permet pas à l'utilisateur d'ajouter deux fois la même personne | Réussi |  |
| L'ajout de personne dans le canal est facilité pour les utilisateurs | Échec | Les utilisateurs doivent entrer l'id des utilisateurs qu'ils veulent ajouter |

Test concernant l’utilisation d’un canal privé effectué le 24.05.2022 :

|  |  |  |
| --- | --- | --- |
| Identifiant | Utilisation d'un canal privé |  |
|  |  |  |
| **Action** | **Réussi ?** | **Remarques** |
| Les utilisateurs peuvent accèder à un canal en cliquant deux fois dessus | Réussi |  |
| Le chat varie selon le canal choisie | Réussi |  |
| Les utilisateurs inscrits au canal sont visible | Échec | On peut bien voir leurs messages, cependant nous ne pouvons pas voir qui est inscrit au canal. |

## Erreurs restantes

Il reste des erreurs au niveau de l’ergonomie de l’application.

* L’affichage qui varie selon la résolution mise.

L’option « Modifier » du menu qui apparaît lorsque on fait clic droit sur un message n’est pas fonctionnel.

Au niveau de la création d’un canal privé, l’ajout des personnes dans ce canal est difficile pour les utilisateurs. Elle n’est qu’ajoutable par id et donc non ajoutable par nom et prénom de la personne.

* Possibilité de mettre des ComboBox (liste déroulantes) afin de pouvoir directement mettre tous les employés dans celle-ci. Ceci permettrait aux utilisateurs de pouvoir tout simplement sélectionner les personnes qu’ils veulent en toute simplicité.

## Liste des documents fournis

Le répertoire github avec tous les fichiers du projet sera fournis aux experts ainsi qu’au chef de projet.

* Une version papier de la documentation sera fournis aux deux experts ainsi qu’au chef de projet.
* Un exécutable sera fourni aux deux experts ainsi qu’au chef de projet.
* Le journal de travail final sera aussi fourni aux deux experts ainsi qu’au chef de projet.

*Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions*

* *le rapport de projet*
* *le manuel d'Installation (en annexe)*
* *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
* *autres…*

# Conclusions

*Développez en tous cas les points suivants:*

* *Objectifs atteints / non-atteints*
* *Points positifs / négatifs*
* *Difficultés particulières*
* *Suites possibles pour le projet (évolutions & améliorations)*

# Annexes

## Résumé du rapport du TPI / version succincte de la documentation

## Sources – Bibliographie

*Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)… Et de toutes les aides externes (noms)*

## Journal de travail

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Durée** | **Activité** | **Remarques** |
|  |  |  |  |
|  |  |  |  |

## Archives du projet

*Media, … dans une fourre en plastique*

1. CRUD : Create (Créer des données), Read (Lire des données), Update (Mettre à jour (modifier) des données), Delete (Supprimer des données) [↑](#footnote-ref-1)
2. Logiciel de gestion de versions qui suit l’évolution des fichiers sources et garde les anciennes versions de chacun d’eux sans rien écraser. [↑](#footnote-ref-2)
3. iceScrum est un outil de gestion de projets basé sur les principes Agiles et la méthode Scrum. [↑](#footnote-ref-3)
4. Visual Studio est une suite de logiciels de développement [↑](#footnote-ref-4)