

Instructions pour le Manuel d'utilisation et de produit

Ce document est un gabarit de manuel d'utilisation et de produit. Il se peut que le client veuille faire des améliorations sur le prototype, veuille le réparer si quelque chose ne fonctionne pas correctement ou encore qu'un autre groupe d'étudiants veuille travailler dessus pour rendre le prototype plus robuste. Le document doit donc être clair pour que quelqu'un d'autre qui n'est pas un ingénieur **puisse utiliser, entretenir ou reproduire le projet**. Incluez autant d'images et diagrammes que nécessaire pour une meilleure compréhension. Gardez le manuel clair, simple, visuel et logique.

En général, si vous ne savez pas exactement quoi inclure, imaginez que ce document est la seule chose que vous possédiez. Imaginez également que votre travail consistait à ajouter une nouvelle fonctionnalité ou à recréer le projet décrit dans votre document. Que devriez-vous savoir?

Veuillez seulement inclure les détails reliés à votre prototype final.

Conventions de ce manuel:

- Enlever tout le **texte rouge**, il est seulement là pour vous guider
- Enlever cette page (instructions)
- Remplacez toutes les instances de <xxx> par les informations appropriées pour votre groupe, par exemple, vous pouvez remplacer <nom du système (acronyme)> par Le Produit Incroyable (LPI)
- Sauvegarder ce document sous 'Manuel d'utilisation et de produit _numéro de groupe' au lieu de Livrable X pour que les autres savent ce qu'il représente quand ils le voient dans MakeRepo

Manuel d'utilisation et de produit pour le projet de conception

DASH-N

Soumis par:

SOPHIE MASON

NICE UMBA KASONGO

HENRY CRANN

ABDELGHANI AMEDJIKOUH

DAGOBERT FOTSO

29 Novembre 2024

Université d'Ottawa

Table des matières

Table des matières	ii
--	--------------------

Liste de figures	iv
Liste de tableaux	v
Liste d'acronymes et glossaire	vi
1 Introduction	1
2 Aperçu	2
2.1 Conventions	2
2.2 Mises en garde & avertissements	2
3 Pour commencer	3
3.1 Considérations pour la configuration	3
3.2 Considérations pour l'accès des utilisateurs	3
3.3 Accéder au système	3
3.4 Organisation du système & navigation	3
3.5 Quitter le système	3
4 Utiliser le système	4
4.1 <Fonction/Caractéristique donnée>	4
4.1.1 <Sous-fonction/Sous-caractéristique donnée>	4
5 Dépannage & assistance	5
5.1 Messages ou comportements d'erreur	5
5.2 Considérations spéciales	5
5.3 Entretien	5
5.4 Assistance	5
6 Documentation du produit	6
6.1 <Sous-système 1 du prototype>	6

6.1.1	LDM (Liste des Matériaux)	6
6.1.2	Liste d'équipements	6
6.1.3	Instructions	7
6.2	Essais & validation	7
7	Conclusions et recommandations pour les travaux futurs	8
8	Bibliographie	9
	APPENDICES	10
9	APPENDICE I: Fichiers de conception	10
10	APPENDICE II: Autres Appendices	11

Liste de figures

Insérez votre liste de figure ici (clique droite pour mettre à jour le champ).

Figure 1 : notification des messages

Figure2 : base de données

Figure3 : demande de réponse

Figure 4 : système d'authentification

Figure 5 : base de données pour un utilisateur simple

Figure 6 : base de données du client

Liste de tableaux

[Table 1. Acronymes](#) vi

[Table 2. Glossaire](#) vi

[Table 3. Documents référencés](#) 10

Liste d’acronymes et glossaire

Table 1. Acronymes

Acronyme	Définition
DASH-N	First letter of all our names

Table 2. Glossaire

Terme	Acronyme	Définition

1 Introduction

Ce Manuel d'Utilisation et de Produit (MUP) fournit les informations essentielles pour permettre aux utilisateurs d'exploiter efficacement l'interface d'envoi de messages (IEM) ainsi que la documentation relative au prototype.

L'interface d'envoi de messages offre une solution simple et performante permettant aux employés de bureau de transmettre des messages importants et urgents à leurs collègues. Elle garantit que les destinataires verront les messages et leur permet de répondre facilement et de confirmer la lecture.

L'objectif de ce document est de fournir toutes les informations nécessaires pour que les utilisateurs puissent utiliser le produit en toute simplicité, tout en comprenant le contexte et le processus de développement du produit. Ce manuel est structuré en fonction des étapes d'utilisation et des sous-systèmes du produit. Il inclut toutes les informations requises pour une utilisation optimale de l'application, tout en présentant des images, des tableaux, ainsi que des instructions détaillées. Le document offre également un aperçu général du projet, des informations complémentaires sur le matériel utilisé et décrit le processus de création du produit.

Ce produit constitue une interface sécurisée, avec des comptes protégés par des mots de passe et des identifiants d'utilisateur. Il respecte les normes de sécurité et de confidentialité afin de garantir la protection des données et de préserver la confidentialité de l'utilisation.

2 Aperçu

Le problème que nous essayons d'attaquer avec la création de ce produit est la surcharge informatique chez les déployer de services partagés Canada. Les employés de cette compagnie reçoivent tellement de courriels chaque jour que les courriels actuellement importants passent souvent inaperçus par les employés. Nôtres projet offrirait une solution à ce problème dans la forme d'un application et interface web qui permet à envoyer les messages les PLUS important directement à l'ordinateurs des employeurs qui les laisse répondre directement et rapidement ainsi que confirmer leur lecture et réception du message

Ce produit est très simple à utiliser, il faut simplement accéder à l'interface (figure 1.1) rentrer son nom d'utilisatrice de mot de passe, si souhaité (figure 1.2), entrer le message à envoyer et remplir le reste des boites avec l'information nécessaire, comme le niveau de priorité, l'ajout de fichiers ou images et choisir les destinataires voulus.

Du côté de réception du message, après que le message est reçu (figure 1.3), il faut simplement cocher qu'ils ont lu le message après qu'il ait reçu ou répondu dans quelques phrases si nécessaire.

Envoyer un Message

Titre :

Saisissez le titre de la notification

Message :

Saisissez votre message ici...

Niveau de priorité :

Haute

Envoyer à :

☐ Groupe

☐ Individus

Envoyer

Gestion des Comptes

Ajouter un compte

Supprimer un compte

2.1 Conventions

Dans le cadre de l'élaboration d'une application de messagerie intelligente, les conventions stylistiques et de syntaxe de commande sont essentielles pour garantir une interaction claire. Les actions requises de l'utilisateur sont souvent précédées du mot "Action" suivi d'une instruction précise. Les commandes saisies par l'utilisateur peuvent être mises en évidence entre guillemets ou en italique. Les échanges entre l'utilisateur et l'application sont présentés sous forme de dialogues, avec des messages différenciés par des guillemets ou des caractères spéciaux. Les changements d'état de l'application, comme le passage en mode de saisie, sont indiqués par des statuts clairs. Les erreurs sont mises en évidence par des termes comme "Erreur" et des couleurs spécifiques, facilitant ainsi la navigation et la compréhension.

2.2 Mises en garde & avertissements

Avant d'utiliser le prototype, l'utilisateur doit obligatoirement disposer d'un identifiant et d'un mot de passe. En l'absence de ces informations, l'accès aux fonctionnalités de notre produit sera restreint, car une identification sécurisée est requise. De plus, notre application ne permet pas l'annulation ou la suppression d'un message une fois qu'il a été envoyé. Il est donc fortement conseillé de vérifier attentivement le contenu de votre message avant de l'envoyer, car il n'y a pas de possibilité de retour en arrière.

3 Pour commencer

Mode d'emploi pour l'installation et l'utilisation du prototype

1. Installation du prototype pour le Chef d'entreprise / Administrateur

Pour commencer l'utilisation de notre prototype en tant que **Chef d'entreprise** ou **Administrateur**, suivez les étapes ci-dessous :

1. **Télécharger Python :**
Téléchargez et installez **Python** à partir du site officiel <https://www.python.org/downloads/> pour garantir le bon fonctionnement de notre interface.
2. **Copier les fichiers nécessaires :**
Copiez les fichiers suivants sur une clé USB ou sur votre dépôt de projet (Maker Repo) :
 - websocketserver.py
 - setup.py
 - login.html
 - main.html
 - users.json
3. **Exécuter le fichier setup.py :**
Une fois les fichiers copiés, exécutez le fichier **setup.py** pour faciliter l'installation des dépendances nécessaires à l'exécution de l'application.
4. **Lancer le serveur WebSocket :**
Après l'installation, exécutez le fichier **websocketserver.py**. Ce fichier vous permettra de vous connecter en tant que Chef d'entreprise ou administrateur et de démarrer l'interface de gestion.

2. Installation du prototype pour un Utilisateur

Pour une personne souhaitant se connecter en tant qu'**Utilisateur**, suivez les étapes ci-dessous :

1. **Télécharger Python :**
Téléchargez et installez **Python** à partir du site officiel
<https://www.python.org/downloads/>.
2. **Copier les fichiers nécessaires :**
Copiez les fichiers suivants sur une clé USB ou votre dépôt de projet (Maker Repo) :
 - websocketclient.py
 - setup.py
3. **Exécuter le fichier setup.py :**
Une fois les fichiers copiés, exécutez le fichier **setup.py** pour installer toutes les dépendances requises.
4. **Lancer le client WebSocket :**
Après l'installation, exécutez le fichier **websocketclient.py**. Cela permettra à l'utilisateur de se connecter et d'interagir avec l'interface en tant qu'utilisateur.

Remarque importante :

Assurez-vous que toutes les dépendances Python sont correctement installées avant de lancer les fichiers. Si vous rencontrez des problèmes, consultez la documentation d'installation.

3.1 Considérations pour la configuration

users.json is a stop gap for a proper database

Make sure all the files are in the same directory

3.2 Considérations pour l'accès des utilisateurs

Ce produit est destiné à tous les travailleurs du gouvernement, mais au sein de l'organisation, seuls les employés ayant un certain niveau d'autorité seront autorisés à accéder à l'interface permettant l'envoi de messages. Selon les besoins de chaque entreprise utilisant ce produit, différents niveaux d'accès peuvent être configurés pour différents utilisateurs, simplement en attribuant un fichier spécifique à l'ordinateur de chaque employé. Les employés qui n'ont pas accès à l'envoi de messages réservés

recevront un fichier leur permettant uniquement la réception des messages, sans l'accès à l'interface d'envoi.

Une fonctionnalité permet également la création de comptes protégés par mot de passe, afin d'octroyer des niveaux d'accès différenciés aux utilisateurs. Cela permet à certains utilisateurs d'envoyer des messages et de gérer des informations, telles que les groupes de diffusion, associés à un compte sécurisé, indépendamment de l'ordinateur utilisé.

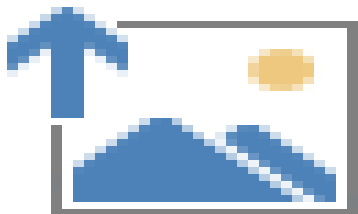
3.3 Accéder/installation du système

Serveur :

- Mettez à jour le fichier des identifiants avec les informations utilisateur requises (solution temporaire pour une base de données informatique).
- Lancez le fichier de configuration en utilisant python.exe.
- Exécutez le fichier websocketserver avec python.exe.

Client :

- Remplacez l'adresse IP en haut du fichier client par l'IP fournie par le serveur (cette opération ne doit être effectuée qu'une seule fois).
- Lancez le fichier de configuration en utilisant python.exe (ce sera le paramètre par défaut après l'installation de Python).
- Exécutez le fichier websocketclient avec python.exe.



Site Web:

- Utilisez le lien fourni par le serveur pour accéder au site.
- Connectez-vous avec un compte autorisé.

The image shows a web interface with two main sections. The first section, titled "Envoyer un Message", contains a "Titre :" label followed by a text input field with placeholder text "Saisissez le titre de la notification". Below this is a "Message :" label followed by a larger text area with placeholder text "Saisissez votre message ici...". Underneath the message area is a "Niveau de priorité :" label followed by a dropdown menu currently showing "Haute". The second section, titled "Envoyer à :", contains two radio button options: "Groupe" and "Individus". Below these options is a green button labeled "Envoyer". The third section, titled "Gestion des Comptes", contains two green buttons: "Ajouter un compte" and "Supprimer un compte".

Envoyer un Message

Titre :

Saisissez le titre de la notification

Message :

Saisissez votre message ici...

Niveau de priorité :

Haute

Envoyer à :

☐ Groupe

☐ Individus

Envoyer

Gestion des Comptes

Ajouter un compte

Supprimer un compte

Figure 7: site web (interface web)

3.4 Organisation du système & navigation

Server <--- files: - login.html

| - main.html

| - users.json

|

|---- ws protocol over port 8765 ----> client

|<--- http protocol over port 8080 --> website

3.5 Quitter le système

Comment quitter l'application (interface)

Pour quitter l'application ou fermer l'interface, il vous suffit de suivre les étapes suivantes :

1. Fermer l'interface :

Cliquez sur le **bouton "X"** situé dans le coin supérieur droit de la fenêtre de l'application. Ce bouton permet de fermer l'interface et de quitter l'application de manière sécurisée.

- Sur **Windows** ou **Linux**, le bouton "X" se trouve en haut à droite de la fenêtre.
- Sur **Mac**, le bouton de fermeture se situe également en haut à gauche.

2. Éteindre l'interface :

Si vous souhaitez éteindre l'interface complètement, assurez-vous que la fenêtre de l'application est fermée. Vous pouvez également utiliser les options du système d'exploitation pour éteindre l'ordinateur si nécessaire.

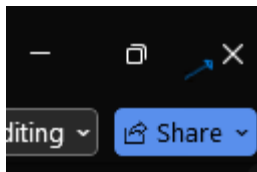


Figure :

4 Utiliser le système

4.1: Send Notifications to Clients

1. Description:

The **Send Notifications to Clients** feature allows website users (admins) to send notifications to clients connected via WebSocket. The notification includes a title, message, and priority level. Admins can also attach files such as icons or other attachments.

2. **Input:**

- a. The admin fills in the **Title** and **Message** fields.
- b. The admin selects the **Priority Level** (Haute, Moyenne, Faible).
- c. The admin can choose to send notifications to **Group** or **Individuals**.
 - i. **Group**: Sends the notification to all connected clients.
 - ii. **Individuals**: Admin selects specific clients to send notifications to.
- d. **File Attachments**: Admin can choose to attach an icon and/or a file (optional).

3. **Output:**

- a. When the notification is successfully submitted, it is broadcasted to the selected clients (either all or the selected individuals).
- b. A confirmation message appears to inform the admin of the successful submission of the notification.

4. **Images/Screenshots:**

- a. **Figure 1**: Screenshot of the **Send Notification Form** showing the fields for Title, Message, Priority, and buttons for selecting **Group** or **Individual**.

Envoyer un Message

Titre :

Saisissez le titre de la notification

Message :

Saisissez votre message ici...

Niveau de priorité :

Haute

Envoyer à :

☐

Groupe

☐

Individus

Envoyer

Gestion des Comptes

Ajouter un compte

Supprimer un compte

5. Warnings or Special Instructions:

- a. Ensure that clients are connected before attempting to send notifications. If no clients are connected, a warning message will be displayed.
- b. If no individuals are selected and **Group** is not selected, the system defaults to sending notifications to all connected clients.
- c. Attachments must be in valid formats (icons can be image files, attachments can be any type of file).

4.1.1: Select Recipient (Group vs. Individual)

1. Description:

The admin must select whether to send notifications to a **Group** (all clients) or specific **Individuals**. If **Individuals** is selected, a dropdown list of connected clients is presented from which the admin can select one or more clients.

2. Input:

- a. The admin selects either **Group** or **Individual**.
- b. If **Individual** is selected, the admin selects one or more clients from the dynamically populated list of connected clients.

3. Output:

- a. **Group**: The notification is sent to all connected clients.
- b. **Individuals**: The notification is sent only to the selected clients.

4. Warnings or Special Instructions:

- If no individual clients are selected when **Individual** is chosen, the system will default to sending notifications to **Group**.

4.2: Account Management (Adding/Removing Accounts)

1. Description:

The **Account Management** section allows admins to add or remove user accounts. Admins provide the username, password, UUID, and role (admin or user) when adding a new account. To remove an account, the admin selects the user from a dropdown list of existing users.

2. Input:

- a. **Add Account**:
 - i. Username, password, UUID, and role (admin or user).
- b. **Remove Account**:
 - i. The admin selects an existing user from a dropdown list of current users.

3. Output:

- a. When adding an account, the new account is added to the system, and the list of users is updated.
- b. When removing an account, the selected account is removed from the system, and the list of users is updated.

4. Images/Screenshots:

- a. **Figure 4**: Screenshot of the **Add Account** form with fields for username, password, UUID, and role.

Envoyer un Message

Titre :

Saisissez le titre de la

Message :

Saisissez votre me

Niveau de priorité :

Haute

Envoyer à :

☐ Groupe

☐ Individus

Gestion des

Ajouter un compte

Nom :

Entrez le nom

Mot de passe :

Entrez le mot de passe

UUID :

Entrez l'UUID

Rôle :

Admin

- b. **Figure 5:** Screenshot of the **Remove Account** popup showing the dropdown list of existing users.

The screenshot displays a web application interface with a modal dialog box open. The background is a dimmed view of a form titled "Envoyer un Message". The modal, titled "Supprimer un compte", prompts the user to "Sélectionnez un compte :". It features a dropdown menu with the following options: "Henry" (selected and highlighted in blue), "Mico", "Amedj", and "Sophi". Below the list is a green button labeled "Annuler". The background form includes fields for "Titre :", "Message :", "Niveau de priorité :" (with a dropdown set to "Haute"), and "Envoyer à :" (with radio buttons for "Groupe" and "Individus"). A green "Envoyer" button is at the bottom of the form. Below the form is a section titled "Gestion des Comptes" with two green buttons: "Ajouter un compte" and "Supprimer un compte".

5. **Warnings or Special Instructions:**

- a. Only admins have permission to add or remove accounts.
- b. Removing an account will immediately deactivate that user's session if they are currently logged in.

4.3: Viewing Connected Clients

1. **Description:**

This feature allows the admin to view the list of all currently connected WebSocket clients. The list is updated in real-time as clients connect or disconnect.

2. **Input:**

- a. No input is required from the admin; the list is automatically populated by the system.
3. **Output:**
 - a. A dynamically updated list of currently connected clients, displaying their usernames or IDs.
4. **Warnings or Special Instructions:**
 - a. The list updates automatically as clients connect and disconnect.

5 Dépannage & assistance

This is setup for local usage, some technical things need to change to be used over the public network

If network error: make sure the ports are visible over local network

Will not work on campus internet have to use router

If permission error: run as admin

1. Procédures de Récupération et de Correction d'Erreurs

Lors de l'utilisation d'une application, plusieurs erreurs peuvent survenir. Voici les erreurs courantes et leurs solutions, expliquées simplement.

1.1 Erreurs de Réseau

- **Problème : Erreur de connexion au réseau local**
 - **Cause :** Les ports nécessaires ne sont pas visibles sur le réseau.
 - **Solution :** Vérifiez que les ports utilisés par l'application sont ouverts sur le réseau local et ne sont pas bloqués par un pare-feu.
- **Problème : Connexion impossible sur le réseau du campus**
 - **Cause :** Restrictions du réseau de l'université ou de l'entreprise.
 - **Solution :** Connectez-vous via un **routeur privé** plutôt que d'utiliser le réseau du campus pour éviter les limitations du réseau.

1.2 Erreurs de Permission

- **Problème : Erreur de permission**
 - **Cause :** L'utilisateur n'a pas les droits nécessaires pour exécuter l'application.
 - **Solution :** Exécutez l'application en tant qu'administrateur. Cliquez droit sur l'icône et sélectionnez « **Exécuter en tant qu'administrateur** ».

1.3 Autres Erreurs

- **Problème : Erreur de configuration ou fichier manquant**
 - **Cause :** Fichiers ou configurations manquants ou corrompus.

- **Solution :** Vérifiez les paramètres de configuration ou réinstallez l'application si nécessaire.

5.1 Messages ou comportements d'erreur

Les erreurs sont détaillées dans la console. Si elles ne le sont pas, il s'agit d'un cas exceptionnel non prévu, qui dépasse le cadre de mes connaissances.

5.2 Considérations spéciales

1. Analyse des performances

- **Détection des points de congestion :** Évaluez les performances de l'application à l'aide d'outils de suivi pour identifier les problèmes de latence ou d'utilisation élevée des ressources (CPU, mémoire, bande passante).
- **Test de capacité :** Réalisez des tests de capacité pour imiter un volume élevé d'utilisateurs en parallèle afin de repérer les points de fragilité possibles et les secteurs nécessitant une amélioration.

2. Surveillance et journalisation

- **Établissement d'une surveillance constante :** Employez des outils de suivi en temps réel pour surveiller l'état de l'application et les anomalies éventuelles.
- **Journalisation approfondie :** Mettez en place des journaux adaptés pour enregistrer des données sur les fautes, les requêtes lentes et les perturbations.

3. Gestion des erreurs et des exceptions

- **Exceptions imprévues :** Mettez en place des systèmes pour capter et traiter les exceptions qui pourraient se produire, telles que des erreurs de base de données, des erreurs de connexion ou des exceptions liées à l'utilisation excessive de mémoire.
- **Messages d'erreur explicites :** Veillez à ce que les messages d'erreur soient compréhensibles et utiles sans révéler d'informations sensibles pouvant nuire à la sécurité.

4. Optimisation des requêtes et de la base de données

- **Indexation adéquate :** Assurez-vous que les requêtes à la base de données sont optimisées et que des index appropriés sont présents pour prévenir les temps de réponse prolongés.
- **Restriction des requêtes :** Employez des méthodes telles que la pagination ou le caching pour réduire la pression sur la base de données lors de la visualisation de grandes

quantités d'informations.

- Suivi des requêtes lentes : Activez l'enregistrement des requêtes lentes afin de détecter et résoudre les problèmes de performance associés aux requêtes compliquées.

5. Scalabilité et architecture distribuée

- Scalabilité verticale et horizontale : Élaborez des plans pour accroître les ressources (scalabilité verticale) ou ajouter des serveurs (scalabilité horizontale) afin de faire face à une hausse de la charge.
- Usage de services de cache : Servez-vous de systèmes de mise en cache pour conserver provisoirement des informations souvent sollicitées afin d'alléger les charges sur le serveur de base de données.
- Équilibrage de charge : Mettez en place un système d'équilibrage de charge (load balancer) pour répartir efficacement le trafic entrant sur plusieurs serveurs.

6. Sécurité

- Prévention des attaques DDoS : Veillez à ce que des systèmes de défense contre les attaques par déni de service distribué (DDoS) soient instaurés pour éviter la surcharge de votre application due à des requêtes malveillantes.
- Vérification des données : Installez des mécanismes stricts de vérification des données afin de prévenir les attaques telles que l'injection SQL, XSS et autres failles associées à l'entrée d'informations.
- Chiffrement des données sensibles : Veillez à ce que toutes les informations critiques soient cryptées, tant lorsqu'elles sont transmises qu'elles sont stockées, afin de garantir la confidentialité et l'intégrité des données.

7. Gestion des utilisateurs et des sessions

- Restreindre le nombre de demandes par utilisateur : Mettez en place des limites de requêtes (rate limiting) afin d'éviter les abus et diminuer la pression causée par les actions répétées d'un utilisateur.
- Gestion des sessions : Veillez à ce que la gestion des sessions soit sécurisée et efficace, et que les sessions se terminent après une période d'inactivité afin de libérer les ressources.

8. Mises à jour et déploiement

- Stratégie de déploiement échelonné : Réalisez des déploiements par phases pour réduire les interruptions de service. Employez des méthodes de déploiement bleu/vert pour expérimenter de nouvelles versions de l'application auprès d'un groupe limité d'utilisateurs.
- Tests de régression : Avant chaque mise à jour, réalisez des tests de régression pour s'assurer que les modifications ne nuisent pas aux fonctionnalités déjà en place.
- Stratégie de restauration : Élaborer un plan de réversion si une mise à jour entraîne un problème significatif.

9. Ajustement en temps réel

- Ajustements dynamiques : Implémentez des dispositifs qui facilitent la modification des réglages de l'application instantanément (ex. : accroître la capacité de mémoire cache, modifier les seuils de requêtes).
- Retours des utilisateurs : Établissez un dispositif pour récolter les avis des utilisateurs afin de déceler rapidement les soucis de performance ou de surcharge qu'ils expérimentent.

10. Mises en garde et limitations

- Surcharge de l'interface utilisateur : Ne surchargez pas l'interface avec trop de données ou d'éléments interactifs, car cela peut ralentir le chargement de la page. Appliquez des méthodes pour optimiser l'expérience utilisateur.
- Restriction de la quantité de données manipulées : Évitez de traiter une trop grande quantité de données simultanément ; adoptez des processus asynchrones et divisez les tâches en lots pour traiter d'importantes quantités d'informations de manière efficace.
- Problèmes de synchronisation : Dans la gestion de données en temps réel, des problèmes peuvent survenir entre les divers éléments de l'application, provoquant des incohérences. Une administration adéquate des files d'attente et des systèmes de verrouillage peut diminuer ces risques.

En synthèse, la résolution de problèmes d'une application web nécessite une stratégie globale tenant compte de la performance, de la sécurité, de la scalabilité et de l'expérience utilisateur. Une stratégie réfléchie et l'application de pratiques optimales garantissent une gestion efficace et réduisent les interruptions de service.

5.3 Entretien

Étant donné qu'il s'agit d'un logiciel, aucun entretien spécifique n'est nécessaire. Toutefois, il est recommandé de vérifier que, en cas de panne de l'ordinateur, l'application soit redémarrée et que le fichier soit toujours installé.

5.4 Assistance

En cas de problème, je suis disponible pour vous apporter mon assistance. Pour ce faire, veuillez m'envoyer un message à l'adresse suivante : aamed007@uottawa.ca. Mon nom est Amedjkouh Abdelghani.

Si le problème concerne un incident de sécurité, il est impératif de vous déconnecter de vos comptes afin de réduire les risques de fuite de données ou d'usurpation d'identité.

6 Documentation du produit

6.1 Sous-système 1 du prototype

6.1.1 NDM (Nomenclature des Matériaux)

Le seul matériel que nous avons utilisé pour effectuer nos prototype c'est python

- 6.1.2 Liste d'équipements

Iteme	Description	Quantité	Coût	Utilité
Routeur	Routeur de wifi	1	44.93\$	Pour accomplire les tests

6.1.3 Instructions

Étape 1 : Configuration de l'environnement logiciel

1. Installez **Python 3.10** ou une version supérieure sur votre machine.
2. Installez les bibliothèques nécessaires :

```
bash
```

```
pip install asyncio websockets aiohttp tqdm json
```

Étape 2 : Démarrage du serveur

1. Placez le fichier `server.py` et les fichiers associés (comme `users.json`) dans le même répertoire.
2. Exécutez le serveur en utilisant :

```
bash  
python server.py
```
3. Le serveur démarre et écoute les connexions HTTP et WebSocket sur les ports configurés.

6.2 Essais & validation

Le premier test a été réalisé sur le sous-système de notifications, au cours duquel plusieurs scénarios ont été évalués pour valider les différentes fonctionnalités du système.

Essais de Validation de la Conception Finale

1. Test de Charge sur le Système de Notification

Dans le cadre de la validation de notre conception finale, nous avons réalisé plusieurs tests sur le **système de notification** pour évaluer sa performance et sa robustesse sous différentes conditions.

- **Objectif du test :**
Le principal objectif de ce test était de mesurer la capacité du système à gérer plusieurs utilisateurs simultanés et d'évaluer le **temps de réception des notifications** sous charge.
- **Scénario du test :**
Un test de charge a été effectué avec un maximum de **9 utilisateurs simultanés**. Ce test visait à simuler une utilisation réelle du système pour observer sa performance en termes de **réactivité et de gestion des notifications**.
- **Résultats du test :**
Les résultats ont montré que, même avec jusqu'à 9 utilisateurs simultanés, le système a maintenu un **temps de réception des notifications de quelques secondes**. Cela confirme que le système peut supporter une charge modérée sans impact significatif sur ses performances.

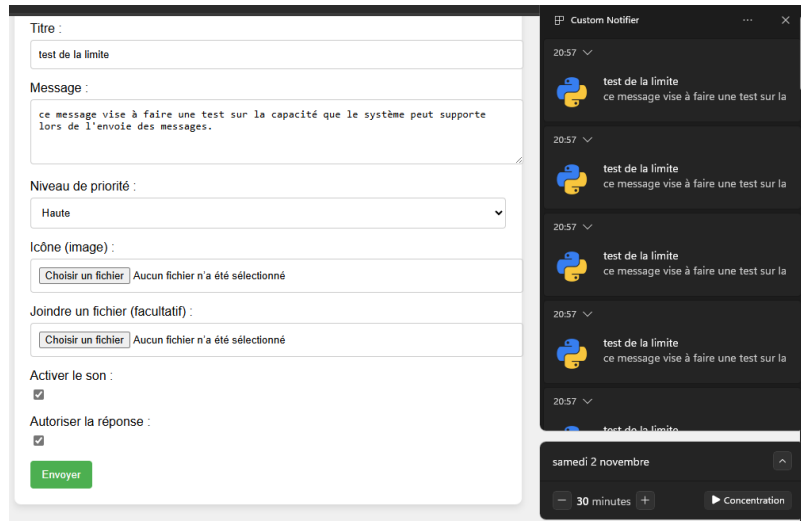


Figure 1 : notification des messages

2. Test de la Fonctionnalité de Réponse aux Notifications

Un autre test a porté sur la **fonctionnalité de réponse aux notifications**, visant à valider la capacité du système à gérer et afficher les réponses des utilisateurs.

- **Objectif du test :**
Vérifier que les réponses aux notifications s'affichent correctement dans le système et que l'utilisateur peut interagir avec les notifications de manière fluide.
- **Résultats du test :**
Il a été constaté que les réponses aux notifications s'affichent correctement dans le **système serveur**. Cependant, des **améliorations** sont nécessaires, notamment l'ajout d'une fonctionnalité permettant à l'utilisateur de spécifier l'**heure souhaitée** pour répondre à un message. Cette fonctionnalité permettrait de recevoir une notification appropriée au moment choisi.

```
C:\WINDOWS\py.exe
Client connected: ('::1', 63297, 0, 0)
Received data from client: Client connected
Received data from client: Client connected
Client connected: ('::1', 63312, 0, 0)
Received data from client: Client connected
Received data from client: Client connected
Client connected: ('::1', 63320, 0, 0)
Received data from client: Client connected
Received data from client: Client connected
Client connected: ('::1', 63329, 0, 0)
Received data from client: Client connected
Received data from client: Client connected
Client connected: ('::1', 63342, 0, 0)
Received data from client: Client connected
Received data from client: Client connected
Client connected: ('::1', 63346, 0, 0)
Received data from client: Client connected
Received data from client: Client connected
Client connected: ('::1', 63569, 0, 0)
Received data from client: html
Client removed from active connections list: <websockets.legacy.server.WebSocketServerProtocol object at 0x000001C829566E90>
Client connected: ('::1', 63582, 0, 0)
Received data from client: html
Client removed from active connections list: <websockets.legacy.server.WebSocketServerProtocol object at 0x000001C829566850>
Client connected: ('::1', 63670, 0, 0)
Received data from client: html
Client removed from active connections list: <websockets.legacy.server.WebSocketServerProtocol object at 0x000001C829567250>
Received data from client: Notification clicked
Received data from client: Notification clicked
Client connected: ('::1', 49265, 0, 0)
Received data from client: html
Client removed from active connections list: <websockets.legacy.server.WebSocketServerProtocol object at 0x000001C829566FD0>
Received data from client: Notification marked as read.
Received data from client: Notification marked as read.
```

Figure 2 : base de données

3. Test de la Notification avec Son et Option de Réponse

Le test suivant a vérifié deux aspects importants de la fonctionnalité de notification : la **notification sonore** et la gestion des réponses par le serveur.

- **Objectif du test :**
 - Vérifier que les notifications sont accompagnées du **son** lorsque cette option est activée par le serveur.
 - Tester la fonctionnalité permettant au serveur de déterminer si une **réponse est attendue** pour une notification spécifique.
- **Résultats du test :**

Les tests ont montré que la **notification sonore** fonctionne correctement lorsque cette option est sélectionnée par le serveur. De plus, la fonctionnalité permettant de spécifier si une **réponse est attendue** fonctionne de manière fiable dans le système de notification.

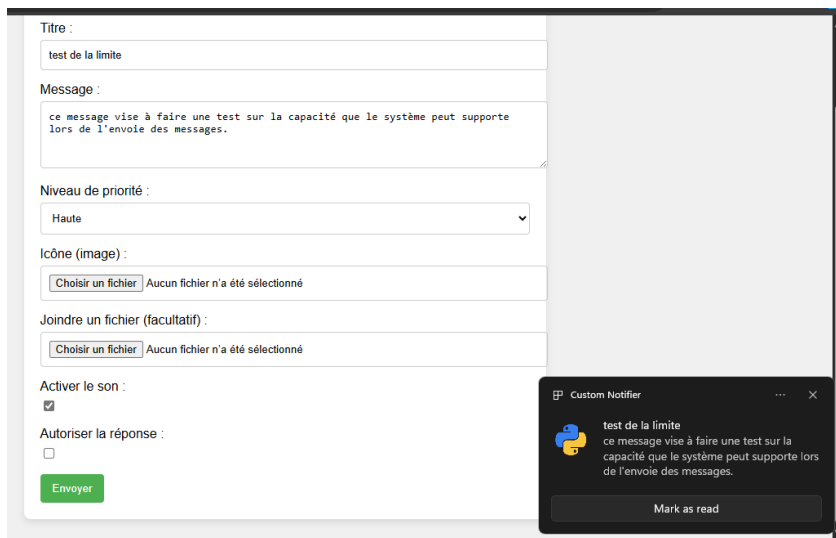


Figure 3 : demande de réponse

2. Test sur le Sous-Système de Connexion et de Catégorisation

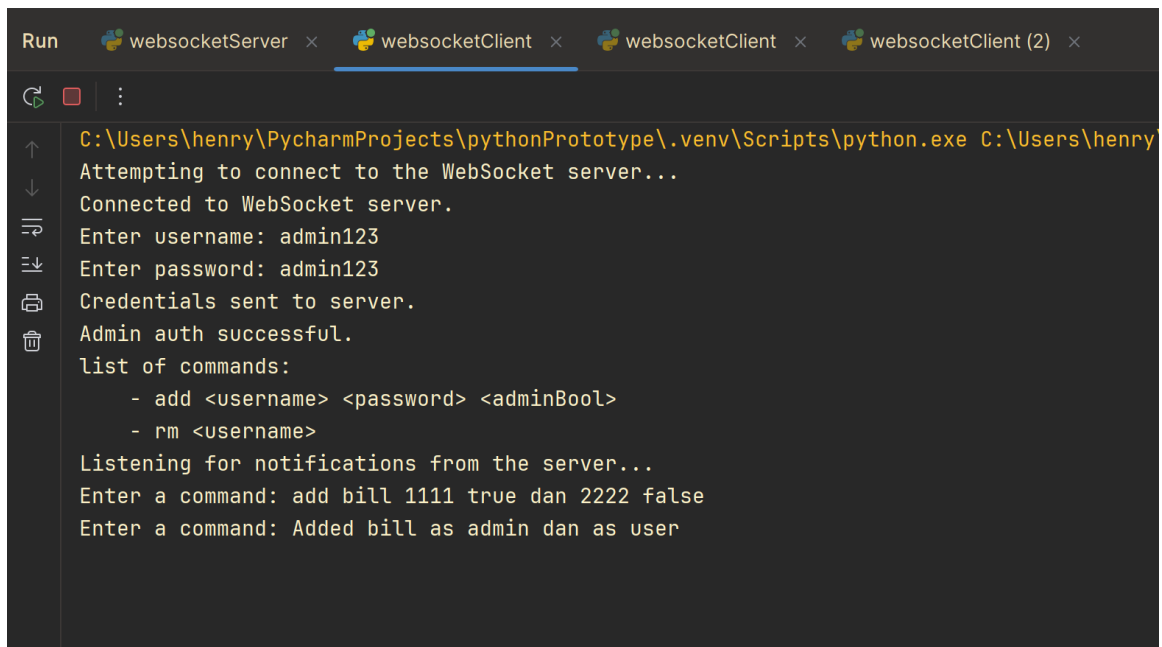
Le deuxième test a porté sur le **sous-système de connexion et de catégorisation**, dans le but de valider les fonctionnalités essentielles liées à l'authentification des utilisateurs, à la gestion des rôles, ainsi qu'à l'enregistrement des données dans le système. Plusieurs scénarios ont été évalués pour assurer le bon fonctionnement du système dans différents contextes.

1. Test d'Authentification :

Des tests ont été réalisés sur le processus d'**authentification** pour vérifier que les utilisateurs pouvaient se connecter correctement à l'interface web. Ce test a permis de valider la fonctionnalité de notification associée à l'authentification, garantissant que les messages de connexion sont envoyés et reçus de manière appropriée.

2. Test de Gestion des Administrateurs :

Un test a été effectué pour vérifier que les **clients** pouvaient **ajouter** et **retirer** des utilisateurs en tant qu'**administrateurs**. Ce test a permis de confirmer que le système gère correctement les modifications de rôles et que l'interface permettait de contrôler l'accès aux fonctionnalités administratives.

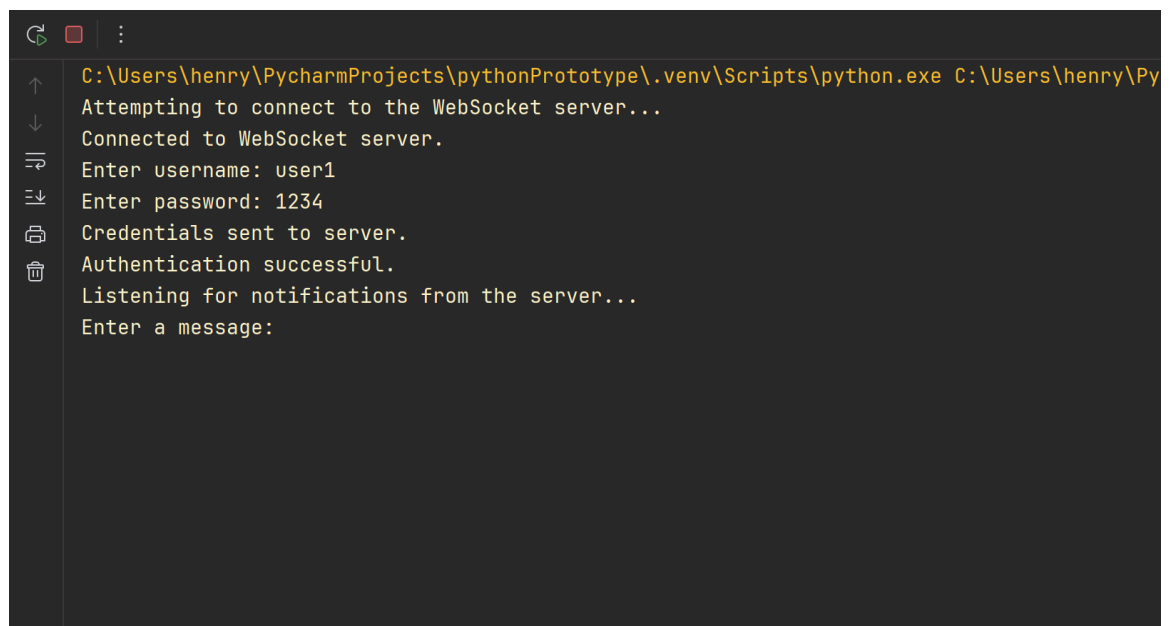


```
Run websocketServer x websocketClient x websocketClient x websocketClient (2) x
C:\Users\henry\PycharmProjects\pythonPrototype\.venv\Scripts\python.exe C:\Users\henry
Attempting to connect to the WebSocket server...
Connected to WebSocket server.
Enter username: admin123
Enter password: admin123
Credentials sent to server.
Admin auth successful.
list of commands:
  - add <username> <password> <adminBool>
  - rm <username>
Listening for notifications from the server...
Enter a command: add bill 1111 true dan 2222 false
Enter a command: Added bill as admin dan as user
```

Figure 4 : système d'authentification

3. Test de Connexion pour Utilisateur Simple :

Nous avons également validé la fonctionnalité de **connexion** pour un utilisateur **simple**, en s'assurant que ces utilisateurs pouvaient se connecter à l'interface et accéder aux fonctionnalités qui leur sont attribuées, sans restriction inutiles.



```
C:\Users\henry\PycharmProjects\pythonPrototype\.venv\Scripts\python.exe C:\Users\henry\Py
Attempting to connect to the WebSocket server...
Connected to WebSocket server.
Enter username: user1
Enter password: 1234
Credentials sent to server.
Authentication successful.
Listening for notifications from the server...
Enter a message:
```

Figure 5 : base de données pour un utilisateur simple

4. Test d'Enregistrement dans la Base de Données :

Un test a été réalisé pour vérifier l'**enregistrement des données** dans la base de

données du système. Ce test a confirmé que les informations des utilisateurs et les modifications des rôles sont correctement stockées et mises à jour dans la base de données, assurant l'intégrité des données à travers le système.

```

Run  websocketServer x websocketClient x websocketClient x websocketClient (2) x
C:\Users\henry\PycharmProjects\pythonPrototype\.venv\Scripts\python.exe C:\Users\henr
Server started
HTTP server running on http://localhost:8080
Enter notification title: [data]: {'username': 'admin123', 'password': 'admin123'}
('::1', 59150, 0, 0) authenticated successfully as admin
Client connected: ('::1', 59150, 0, 0)
[data]: {'username': 'bill', 'password': '1111'}
('::1', 59153, 0, 0) authenticated successfully as admin
Client connected: ('::1', 59153, 0, 0)
[data]: {'username': 'dan', 'password': '2222'}
('::1', 59160, 0, 0) failed authentication. incorrect username
[data]: {'username': 'user1', 'password': '1234'}
('::1', 59163, 0, 0) authenticated successfully as client
Client connected: ('::1', 59163, 0, 0)

```

Figure 6 : base de données du client

6.1 <Sous-système 2 du prototype>

6.1.1 NDM (Nomenclature des Matériaux)

(Tous ces éléments étaient gratuits)

- Python

Libraries python: asyncio websockets aiohttp tqdm json

- PyCharm

6.1.2 Liste d'équipements

Iteme	Description	Quantité	Coût	Utilité
-------	-------------	----------	------	---------

Routeur	Routeur de wifi	1	44.93\$	Pour accomplir les tests
---------	-----------------	---	---------	--------------------------

6.1.3 Instructions

Étapes : Utilisation de l'interface utilisateur

1. Ouvrez main.html dans un navigateur moderne (Chrome, Firefox, etc.).
2. Connectez-vous avec un compte admin pour accéder aux fonctionnalités (envoi de notifications, gestion des utilisateurs).
3. Suivez les étapes pour envoyer une notification ou gérer les comptes.

6.2 Essais & validation

https://uottawa-my.sharepoint.com/personal/ocran034_uottawa_ca/_layouts/15/guestaccess.aspx?share=EV4lZFsII_ZFivkhzaTobcUUB5HsBNQxPEmhVfpo2ajPn5g

https://uottawa-my.sharepoint.com/personal/ocran034_uottawa_ca/_layouts/15/guestaccess.aspx?share=EdrbfxvzM_rJLuYPd8JKd52IBU5H7RDC5-2IxG2Tho5rAHg&e=XffMjX

(Download to get full functionality of the HTML, there are some hidden things that use embedded JavaScript, if you don't download it the add/remove buttons won't do anything and the selector buttons for groups and individuals won't work)

Tableau 1.

Test	Test plan	result	notes	Future potential
Multiples connexions	Virtual machines	Tested up to 20 before I ran out of processing power and RAM space	Limited by server specs (CPU and RAM)	Better server or multiple servers to spread the load or split the load with data juggling (tell clients to disconnect for a certain amount of time after they check)

Full front-end control	Test most edge cases to see if the html page can display everything necessary and if it can control everything independently of the command line	It works	The code is a bit janky but it works	Clean up the code and add an admin login for a more technical and verbose control for debugging
Start on boot	Test the ability to start on boot	It works but you need to manually add it to the list (easiest method is through the task manager)	Potentially automatable with group policy	Find a way to add that to the installer
Installer	Fresh computer, run the installers	All files extracted and accounted for for both server and client installers	Might need admin privileges depending on the current firewall setting of computers as it opens 2 ports	Add start on boot to client installer, server one should be set manually as admins might have their own way of doing that or it might be a Linux server
Server output	Send a series of notification, respond to said notifications, read the output file	Output file shows out of who it sent the notification to, is not connected to the server, who responded, who, out of those it tried to send it to and what the responses were	Company would use a database instead of an output file, this would allow them to add automation and resend the notification to those who have not responded after a certain time	Add compatibility for various database APIs
intrusiveness	Ask people to rate the intrusiveness	3.34, it grabs the attention not in a bad	Max text content is low, useful for a	Add automation with email API

	and effectiveness on a scale of 1 to 10	way but in a way that could break concentration 7.52, effective at delivering a short string of information, not so much for larger amounts	small reminder not so much for a full memo. Could be used hand in hand with an email: short confirmation code at bottom of email used as response for a reminder notification to confirm receipt	to send a reminder when they receive an email
High importance	Ask people to rate the intrusiveness and effectiveness on a scale of 1 to 10	7.43, it is intrusive with the red background, but the sound is not plus it is small as it is still a notification. 8.21, effective at ensuring people read it at the expense of being a bit intrusive	The high importance notifications automatically replace the attachment image with a solid red one	Add a better sound for them, client mentioned amber alerts could use that sound

6.1 <Sous-système 3 du prototype>

6.1.1 NDM (Nomenclature des Matériaux)

- Python

- Librairie python: asyncio websockets aiohttp tqdm json

6.1.2 Liste d'équipements

Iteme	Description	Quantité	Coût	Utilité
Routeur	Routeur de wifi	1	44.93\$	Pour accomplir les tests

6.1.3 Instructions

Étape 1 : Configuration de l'environnement logiciel

3. Installez **Python 3.10** ou une version supérieure sur votre machine.
4. Installez les bibliothèques nécessaires :
bash
pip install asyncio websockets aiohttp tqdm json

Étape 2 : Configuration des clients

1. Les clients doivent exécuter le fichier `websocketClient.py` pour se connecter au serveur via WebSocket.
2. Vérifiez que les clients reçoivent les notifications envoyées via l'interface.

6.2 Essais & validation

Nous avons réalisé une série de tests avant la journée de la conception et la présentation finale. On a d'abord testé la vitesse d'envoi des messages et de leurs réceptions pour pas causer des retards dans l'application au futur. De plus, nous avons évalué si les messages s'envoyaient parfaitement aux personnes voulues. Pour voir ceci plus en détails regardez en dessous:

1. Essais de fonctionnalité

Les tests fonctionnels ont été réalisés pour valider si l'application remplissait toutes ses fonctions de base : envoi de messages, notifications, gestion des conversations, etc. Les étapes suivantes ont été couvertes :

- **Tests d'envoi de messages :**
 - Envoi de texte, images, vidéos et fichiers.
 - Tests de l'historique des messages pour vérifier la sauvegarde correcte.
 - Réception de messages dans des délais raisonnables.
 - **Résultats :** Tous les types de messages ont été envoyés et reçus sans erreur, avec un délai moyen d'envoi de moins de 2 secondes.
- **Tests de notifications :**

- Validation de la bonne réception des notifications dans différentes conditions de réseau.
- **Résultats :** Les notifications étaient reçues en temps réel dans la plupart des cas, mais une latence a été notée lors de l'utilisation avec des réseaux faibles (comme celui de l'université).

2. Essais de performance

Les tests de performance ont été effectués pour garantir que l'application pouvait supporter un grand nombre d'utilisateurs et de messages potentiels sans perte de performance.

- **Tests de charge :**
 - Test de la capacité à envoyer des messages simultanément pour 1000 utilisateurs sur une période de 24 heures (On a estimé que c'était le nombre de message maximal que notre client SPC pourrait envoyer sur une).
 - Mesure du temps de réponse sous charge.
 - **Résultats :** L'application a montré une capacité de 90 % de messages envoyés sans délai supplémentaire. Cependant, une légère augmentation du délai de traitement des messages a été observée au-delà de 800 utilisateurs simultanés.

3. Essais de sécurité

La sécurité des données est cruciale dans une application de messagerie. Les essais ont porté sur :

- **Cryptage des messages :**
 - Test de l'algorithme de cryptage de bout en bout.
 - Vérification des failles potentielles.
 - **Résultats :** Le cryptage de bout en bout fonctionne correctement, mais des tests supplémentaires sont nécessaires pour renforcer la protection contre les attaques de type man-in-the-middle.
- **Authentification et gestion des sessions :**
 - Validation de l'authentification par identifiant et par mot de passe.
 - Tests de déconnexion automatique après une période d'inactivité.
 - **Résultats :** L'authentification est fiable, mais un bug a été observé dans la gestion des sessions lorsque l'utilisateur se reconnecte trop rapidement après une déconnexion.

4. Essais d'ergonomie

L'ergonomie a été testée pour garantir une expérience utilisateur fluide.

- **Tests d'interface utilisateur (UI) :**
 - Test de la navigation dans l'application (accès aux conversations, menu de paramètres, etc.).
 - **Résultats :** L'interface est intuitive pour la majorité des utilisateurs, mais des améliorations de la lisibilité sont nécessaires dans certaines sections, notamment dans le mode sombre.
- **Tests d'accessibilité :**
 - Validation de l'accessibilité de l'application pour les utilisateurs ayant des handicaps visuels (compatible avec les lecteurs d'écran).
 - **Résultats :** L'application est compatible avec les lecteurs d'écran, mais il est recommandé d'ajouter davantage d'éléments pour améliorer la navigation par clavier.

5. Problèmes et exigences pour une utilisation prolongée

Certains problèmes ont été identifiés lors des tests d'utilisation prolongée :

- **Problème de batterie :**
 - L'application consomme trop de batterie lorsque l'on reçoit des notifications fréquentes ou lors de l'envoi de fichiers volumineux. Ceci est notamment dû au serveur qui reçoit trop de données en même temps ce qui crée une surchauffe au sein de l'appareil diminuant ainsi la batterie d'une manière considérable
 - **Solution :** Optimisation du traitement en arrière-plan pour réduire l'impact sur la batterie.
- **Problème de gestion de la mémoire :**
 - L'application peut rencontrer des fuites de mémoire après plusieurs heures d'utilisation continue.
 - **Solution :** Mise en place de mécanismes de gestion dynamique de la mémoire et optimisation de l'utilisation des ressources.
- **Problème de lenteur :**
 - Le temps de réponse peut augmenter en cas de surcharge du serveur (lors de nombreux utilisateurs simultanés).
 - **Solution :** Amélioration de l'infrastructure serveur et introduction de mécanismes de mise en cache.

7 Conclusions et recommandations pour les travaux futurs

Au cours de ce prototype, notre groupe a acquis de nombreuses leçons précieuses. Nous avons pris conscience de l'importance de l'organisation du temps, ainsi que de la manière de travailler efficacement en équipe et de résoudre les conflits. Nous avons également acquis des compétences techniques que nous conserverons tout au long de nos carrières, en découvrant des aspects que nous n'aurions peut-être jamais abordés autrement.

Un temps supplémentaire pour travailler sur ce produit aurait été très bénéfique, car plusieurs améliorations auraient pu être apportées. Tout d'abord, avec plus de temps, nous aurions pu améliorer l'efficacité et la vitesse de notre système et de notre prototype. Nous aurions également eu l'occasion de perfectionner l'interface et son aspect esthétique. Enfin, le manque de temps nous a contraints à abandonner l'implémentation d'un système de notification pour les téléphones mobiles, car la majorité des employés de Services partagés Canada ne possèdent pas de téléphone portable. Avec plus de temps, nous aurions pu développer une solution fonctionnant également sur les téléphones mobiles, pour les employés disposant d'un téléphone professionnel.

8 Bibliographie

<https://www.w3schools.com/python/>

<https://docs.python.org/3/>

<https://websockets.readthedocs.io/en/stable/>

<https://desktop.readthedocs.io/en/latest/>

APPENDICES

9 APPENDICE I: Fichiers de conception

Table 3. Documents référencés

Nom du document	Emplacement du document et/ou URL	Date d’émission
Lien MakerRepo	https://makerepo.com/Smaso/2134.equipe-dashn-la-surcharge-dimformatique-	11/01/2024