



Membre du groupe

Amedjkouh Abdelghani
 Dagobert, Fotso Ouambo
 Henry Crann
 Umba Kasongo Nice
 Sophie Mason

Date le 3 novembre 2024	
Groupe DASHN FB24	Université d'Ottawa

Table de matière

Résumer.....	2
Introduction	3
Rétroaction du Client	3
Analyse des Système Critiques	4
Sommaire des résultats des prototypes et des tests	4
Prototype 1	5
Résultats des tests et commentaires ainsi que le spécification cible	8
Rétroaction et commentaires des utilisateurs potentiel	8
Conclusion	9

Résumé

Dans ce livrable, nous développerons un premier prototype que nous soumettrons à une série de tests approfondis afin d'améliorer notre produit en fonction des retours des clients. Nous documenterons nos observations, validerons les idées clés, optimiserons les spécifications et nous préparerons à la phase de prototypage. Pour recueillir des commentaires supplémentaires, nous organiserons également une présentation succincte destinée aux clients. Cette présentation introduira notre démarche et les étapes à venir.

I. Introduction

Dans ce livrable, on va aborder le développement d'un premier prototype de notre application de notifications. L'objectif est de tester ce prototype à travers une série d'analyses approfondies, tout en intégrant les retours des clients pour optimiser notre produit. On va également documenter les observations recueillies, valider les idées principales et préparer le terrain pour les prochaines étapes de prototypage. Pour s'assurer d'une bonne adéquation avec les besoins des utilisateurs, une présentation rapide sera faite aux clients afin de recueillir leurs impressions. En suivant cette démarche, on aspire à garantir que notre solution soit non seulement fonctionnelle, mais également conforme aux attentes des utilisateurs.

II. Rétroaction du client

En date du 25 Octobre 2024 a eu lieu notre deuxième rencontre client. Nous avons fait

Une présentation de notre projet DASH-N pendant une quinzaine de minutes. La rencontre a été très satisfaisante et a beaucoup ému le client. Les rétroactions à la fin ont été très enrichissantes et édifiante pour la suite de notre conception.

Notre client a été particulièrement attentifs sur la présentation de nos trois sous-systèmes et ont appréciés l'Investissement qui a été fait. Le client a demandé à Comprendre le premier sous-système notamment dans le fonctionnement, la connexion au serveur. Ensuite leurs réserves ont été porte sur gestion et la maintenance de notre application web en nous recommandant d'être beaucoup explicite dans le manuel d'utilisation qui accompagnera le produit final. Ils n'ont pas manqué de féliciter l'idées de priorisation et de catégorisation des messages. L'accent a été mis sur le système d'alerte car pour notre client les courriels importants doivent être lu par les destinataires. Au sortir de notre rencontre client, le client a apprécié les idées et réitère la qualité sécuritaire que doit avoir l'application, nous propose d'inclure une fonctionnalité qui demanderait aux employés de choisir une langue de communication entre le Français et l'anglais car l'entreprise est bilingue. Souhaite que l'on y ajoute à des "wes" (Tips) comme ce qui est fait dans leur agence. Notre solution ne devrait pas être désinstallé ou supprimer par employé de son téléphone et autres supports de communication octroyé par l'agence. Ne Pas y inclure des traducteurs automatiques qui pourraient d'après notre client ne pas respecter la fidélité du message initial de l'expéditeur.

Les réserves du client restent au niveau du support qui abritera le concept final pour sa mise en fonction et aussi le processus d'implémentation de l'application qui peut être long pour la validation du prototype mais trouves que les idées des sous-systèmes sont excellentes

III. Prototype 1

Dans cette section, vous trouverez trois liens qui vous donneront accès à notre prototype 1.

- a. [notification_interface.html](#)
- b. [websocketClient.py](#)
- c. [websocketServer.py](#)

IV. Prototype 2

Pour cette section, nous procéderons aux tests du système de notification, qui sera désormais basé sur Apple. En effet, le premier prototype de notre système était exclusivement développé pour Windows. Voici le code du deuxième prototype :

[Tasks.txt](#)

Tasks:

Network Configuration

Run `ipconfig` on Windows to determine server IP

Document the WebSocket server port

Test basic network connectivity between devices:

bash

On iPhone

ping [windows-ip]

Document any eduroam port restrictions or firewall rules

Pythonista Environment

Verify Pythonista installation

Install required packages:

websockets

asyncio

notifications

Test notification permissions

Create a test script to verify notifications work:

python

import notifications

notifications.schedule('Test', 'Testing notifications')

WebSocket Server Testing

Create simple test client on Windows

Verify server is accessible within local network

Document successful connection parameters

Initial Client Implementation

Implement basic WebSocket connection

Add connection status logging

Test basic message reception

Implement simple error handling

Message Handling

Define message format specification

JSON

```
{  
  "title": "string",  
  "message": "string",  
  "priority": "number?",  
  ...  
}
```

Implement message parsing

Add validation for received messages

Create error responses for malformed messages

Notification System

Implement notification creation

Add notification scheduling

Handle notification permissions

Test different notification types

Connection Management

Implement automatic reconnection

Add connection status monitoring

Implement graceful shutdown

Add connection timeout handling

Network Error Handling

Handle common network errors:

Connection timeouts

Network drops

Server disconnects

Implement exponential backoff for reconnection

Add network status monitoring

Message Error Handling

Handle malformed messages

Implement message queuing

Add message delivery confirmation

Create error logging system

Testing Plan

Unit tests for message parsing

Integration tests for WebSocket connection

Network failure scenario tests

Notification delivery tests

Documentation

Setup instructions

Network configuration guide

Message format specification

Troubleshooting guide

Technical Considerations:

Network

Eduroam Restrictions:

Port limitations

Firewall rules

P2P connectivity

Fallback options:

Alternative ports

VPN solutions

HTTP fallback

Security

Message encryption

Connection authentication

Notification permissions

Data privacy

Performance

Connection timeout values

Reconnection intervals

Message queue size

Battery impact

Error Scenarios

Network Failures:

Server unreachable

Network disconnect

Port blocked

Message Failures:

Invalid format

Missing data

Oversized messages

System Failures:

Permission denied

App background state

Memory constraints

Success Metrics

Connection reliability > 95%

Message delivery < 1s

Reconnection time < 5s

Battery impact < 5%

Error recovery > 90%

V. Analyse simple du systèmes critique

Nous allons utiliser Python comme le logiciel principal de programmation pour nos prototypes ainsi que pour la version finale. Elle va nous aider à faire la plupart du code et de la programmation pour être capables d'envoyer et de recevoir les notifications à partir d'un ordinateur Windows, ainsi que pour faire l'interface web. Nous avons choisi Python car il est assez facile à utiliser et à lire, et il est capable de faire toutes les choses dont nous avons besoin. De plus, il fonctionne le mieux avec les connaissances et les expériences des membres de notre groupe.

Un serveur HTML va aussi être utilisé pour servir le site HTML aux clients qui y accèdent via [HTTPS://](https://). Cela permettra aux utilisateurs de se connecter via [HTTPS://](https://) et de recevoir une interface de type site web qui facilitera l'utilisation.

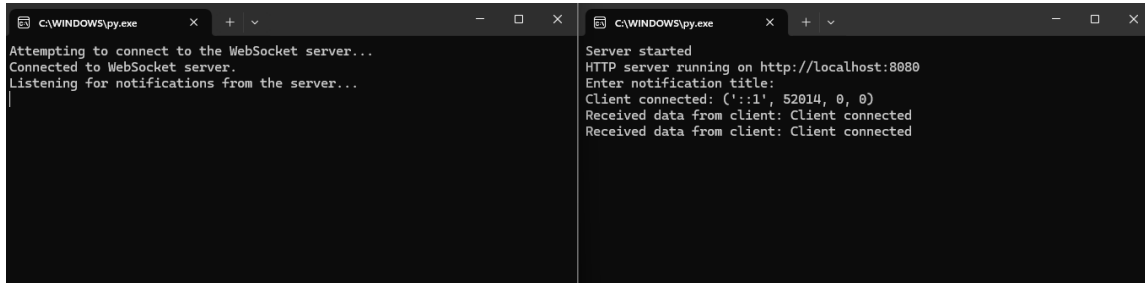
Android PN va aussi agir comme serveur et client afin d'être capable d'envoyer et de recevoir des notifications sur les téléphones Android. * Nous utilisons aussi Desktop Notifier. Pi, qui prend l'information et la transforme en notification sur l'ordinateur. C'est une plateforme gratuite qui est facile à utiliser et qui est compatible avec les autres applications que nous utilisons. *

VI. Sommaire des résultats des prototypes et des tests

Prototype					Test		
N	Pourquoi	Quoi	Fiabilité	Quand	Objectif	Résultat	Durée
1	<p>Le prototype 1 nous permettra d'évaluer comment le système de notification répond aux attentes des clients, tout en effectuant des tests sur son fonctionnement et ses performances. Les éléments clés à analyser incluent :</p> <ul style="list-style-type: none"> Le temps de réception des notifications Le son associé aux notifications La réactivité du système lors de la réception des notifications La possibilité d'ajouter des images aux messages <p>Ces tests seront essentiels pour garantir que le système répond efficacement aux besoins des utilisateurs</p>	<p>Pour développer ce premier prototype, nous avons utilisé l'application Python pour créer un code dédié au système de notification. Ce code inclut des paramètres spécifiques que nous avons configurés afin d'évaluer la réception des notifications.</p>	Faible	Le 01-11-2024	L'objectif du test était de vérifier le bon fonctionnement du système de notification en intégrant tous les éléments essentiels, afin d'acquérir une meilleure compréhension de son fonctionnement global.	À la suite de nos tests, nous avons constaté que le système de notification fonctionne correctement, avec l'envoi des notifications, l'activation du son, et l'affichage des réponses lors de la réception des notifications.	2heures (01-11-2024)

- **Essai du prototype1 :**

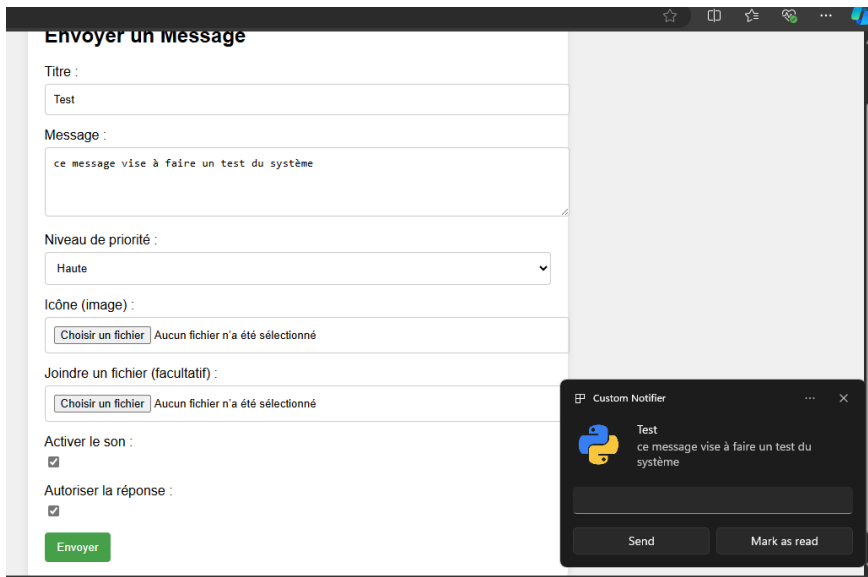
L'objectif du test était de vérifier le fonctionnement du système de notification en intégrant tous les éléments essentiels. Pour ce faire, nous avons commencé par installer le serveur développé par un membre de notre équipe. Nous avons téléchargé deux serveurs : l'un conçu pour le serveur principal et l'autre destiné aux clients. Cette étape est cruciale pour assurer une communication efficace entre les différents composants du système.



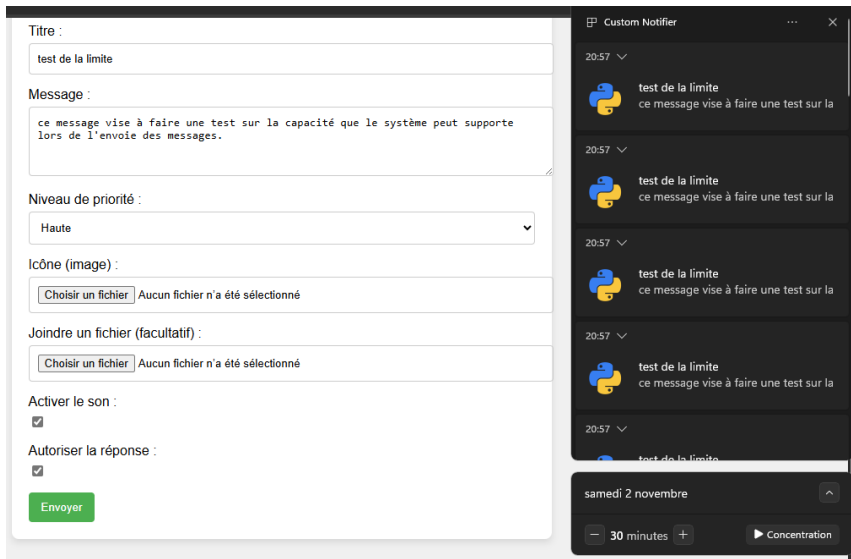
```
C:\WINDOWS\py.exe x + -
Attempting to connect to the WebSocket server...
Connected to WebSocket server.
Listening for notifications from the server...

C:\WINDOWS\py.exe x + -
Server started
HTTP server running on http://localhost:8080
Enter notification title:
Client connected: ('::1', 52014, 0, 0)
Received data from client: Client connected
Received data from client: Client connected
```

Les deux interfaces communiquent entre eux ceci nous permet d'envoyer des messages et de recevoir la notification comme ceci :



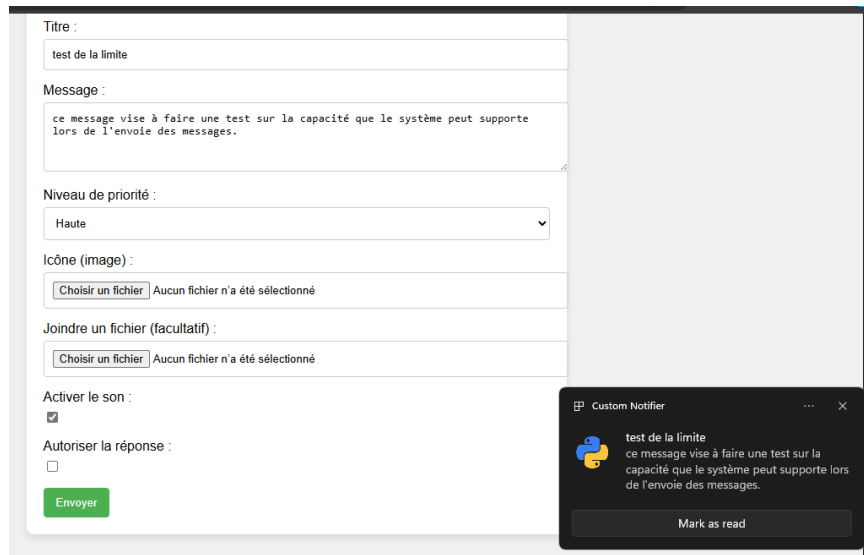
- À l'issue de ce test, notre système a satisfait une grande partie de nos attentes. Toutefois, des améliorations restent à envisager, notamment l'ajout de couleurs pour la priorisation des messages.
- Nous avons réalisé un test de charge afin d'évaluer la capacité du système à gérer un nombre d'utilisateurs simultanés. Ce test a inclus jusqu'à 9 utilisateurs, permettant de mesurer le temps nécessaire pour la réception des notifications et de déterminer si le système peut supporter une charge importante. Les résultats ont montré que le temps de réception des messages est de quelques secondes.



- Nous avons effectué un test concernant la fonctionnalité de réponse aux notifications. Il a été constaté que les réponses s'affichent correctement dans le système serveur. Cependant, des améliorations sont nécessaires, notamment l'ajout d'une fonctionnalité permettant au client de spécifier l'heure à laquelle il souhaite répondre aux messages afin de recevoir une notification appropriée.

```
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', 63328, 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', 63678, 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.
```

- Nous avons également vérifié que les notifications s'affichent avec le son lorsque cette option est sélectionnée par le serveur. De plus, nous avons testé la fonctionnalité permettant de déterminer si le serveur souhaite recevoir une réponse ou non, et cela fonctionne correctement dans notre système de notification.



VII. Résultats des tests et commentaires ainsi que le spécification cible

Prototype 1	Critère fonctionnel et spécification	Valeur mesurée	Valeurs ciblée	Observation/commentaire
Critère fonctionnelle				
	Confirmation de lecture	Disponible	Disponible	Le système de notification permettra aux clients d'exiger une confirmation lorsque certaines personnes ont lu les messages. En effet, notre système offre aux clients la possibilité de choisir s'ils souhaitent recevoir une réponse lors de la réception d'un message par les destinataires.
	Facilite de création			La création du système n'est pas simple, mais l'essentiel est de savoir par où commencer.
Critère non-fonctionnelle				
	Animations dans le message	= 2 images max	Limitée, Emojis, gifts	Notre système permet aux utilisateurs de joindre des images, ce qui ajoute une dimension ludique et engageante pour animer les

				échanges avec les destinataires.
Contrainte				
	Facilite d'utilisation	Élevée	Élevée	Le système est relativement simple à utiliser et ne nécessite pas beaucoup de pratique pour son utilisation. Cependant, la principale difficulté réside dans l'installation de l'interface, qui nécessite un peu de pratique et des guides. Nous prévoyons d'améliorer cet aspect au fur et à mesure de l'avancement de nos prototypes.
	Pas perturber l'ouvrier	Personnalisable	Personnalisable	Le système est non perturbateur pour les utilisateurs, car les messages sont notifiés par un son uniquement si l'expéditeur le souhaite. Cette fonctionnalité permet de maintenir une communication efficace sans causer de distractions inutiles.

VIII. Plan du deuxième essai

Prototype 2				Test 2				
Type	Objectif	Fidélité	Rétroaction	Objectif	Quoi	Comment	Quand	Mesure de performance attendues
Ciblé Analytique	Test le fonctionnement de la connexion et d'envoi de	Test le fonctionnement de la connexion et d'envoi de message entre l'employeur	Aucune du client ou d'utilisateur	Faire le test fonctionnel en vérifiant que les utilisateurs peuvent accéder avec leur identifiants valides à	La connexion et l'envoi de message	Une fois que l'interface web sera en place (7-11-2024)	L'interface de la connexion et l'envoi de messages	Lorsque l'interface ne répond pas à nos attentes, nous procéderons à des modifications jusqu'à obtenir un niveau de performance satisfaisant.

	message entre l'employeur et les employés	et les employées (moyen fidélité)		l'interface sans problème, et vérifier aussi les différents types de contenu que les utilisateurs peuvent envoyer (texte, image, gifs, etc..).			doivent être compatibles avec tous les types d'appareils, y compris les téléphones, ordinateurs et tablettes. De plus, l'application doit être capable de gérer une charge élevée, permettant à de multiples utilisateurs d'envoyer des messages simultanément sans rencontrer de problèmes de performance.	
--	---	-----------------------------------	--	--	--	--	---	--

IX. Rétroaction du prototype 1 par les clients potentiels

Client/ Utilisateur	Rétroaction	
	Commentaires positifs	Commentaires négative ou constructifs
Pascale	J'aime	Mais il faut que l'icone image soit aussi facultatifs, comme celle de joindre un fichier
Prisca	Waouh	Comment on fait pour envoyer les messages aux personnes souhaitées car il n'y a pas un endroit qui nous permet de choisir à qui on veut envoyer les messages.
Julie	Très utile	Devrait avoir une section pour les messages confidentielles. Il devrait y avoir une autre option pour le paramètre Marquer comme lu.
Adèle	Trouver que cette interface est bien faite	Les polices ne sont pas coordonnées et pas trop professionnelle. Les niveaux de priorités devraient être associer avec des couleurs.

X. Mise à jour de la nomenclature des matériaux

Item	Description	Unité de mesure	Quantité	Cout unitaire	Cout étendue	Sources
Mongo DB	Base de donne	Heures	~ 60	0\$	Inconnue	Inter web
Cout totale:	0\$					

Nous n'allons plus utilise le mongo DB dans nos prototypes

XI. Conclusion

En somme, Nous avons présenté les rétroactions de la 2^e rencontre avec client dans lequel nous faisons part de ses satisfactions sur les idées développées autour du concept, aussi de ses réserves quant à l'implémentation, l'entretien et l'utilisation de notre produit finale. On a également

procédé à une analyse critique des sous-systèmes de notre prototype avant de montrer à l'aide d'un tableau les résultats de nos tests de prototypes. Il était aussi question dans ce livrable de la mise à jour des spécifications cibles, des commentaires, et aussi des rétroactions de quelques utilisateurs. Beaucoup de critiques et de suggestions ont été formulées et dont nous en tiendront compte dans la suite de notre travail de conception du prototype final.