

# Extranet de Velocity Distribution

Plan de Gestion

**Version 1.0**

# Sommaire

Version 1.0 – Plan de Gestion.....	1
Sommaire.....	2
1 Objectifs.....	3
2 Ressources.....	3
3 Parties prenantes.....	4
4 Calendrier.....	5
5 Livrables.....	6
6 Tableau des risques.....	6
7 Faisabilité.....	7

# 1 Objectifs

Le système en ligne existant ne fournit pas un support adéquat pour les clients les plus importants, les fournisseurs et les entreprises partenaires. Velocity Distribution souhaite améliorer la qualité de service, les délais de réponse, et simplifier la gestion des parties prenantes. Pour répondre à ces attentes, le développement d'un extranet est envisagé afin de permettre aux clients les plus importants d'avoir un accès plus direct à leurs représentants internes.

Les clients pourront accéder à un support unique via un site web pour communiquer avec leurs représentants de compte et d'autres personnes compétentes via un chat durant les heures ouvrées. En dehors de ces heures, le chat sera remplacé par un outil de messagerie qui permettra au client d'envoyer une demande avec une indication du niveau d'importance.

Les conseillers utiliseront le site web pour répondre aux demandes par chat durant les jours ouvrés, ils disposeront également d'une application mobile IOS pour réceptionner les différents messages des clients le reste du temps. Une indication sonore ou visuelle viendra communiquer au conseiller le niveau d'importance du message.

L'objectif est de développer et déployer le système extranet complet.

# 2 Ressources

Pour la réalisation de ce projet, nous avons besoin de plusieurs profils de compétences :

- **Architecte** pour la conception et la réalisation de l'architecture du projet.
- **Designer** pour la réalisation du visuel de l'interface de la partie web et mobile.
- **Développeur web** pour l'intégration et la réalisation de l'interface web des clients.
- **Développeur mobile** pour la réalisation de l'application destinées aux conseillers.
- **Développeur backend** pour la réalisation de l'API web et du chat.
- **Technicien devops** pour le déploiement de la base de données et la mise en production.

Pour la mise en production de l'application :

- **Serveur** pour le déploiement du Frontend Web et du Backend

### 3 Parties prenantes

Agence	Partie intéressée	Rôle	Interfaces
Velocity	Cary Jurgensen	Encadrement supérieur	Mail
Velocity	N/A	Client de Velocity	Mail
Velocity	Braham Griffin	Testeur du projet	Mail
Velocity	Elizabeth Abramowicz	Testeur du projet	Mail
Vigil	Kris Ito	Consultant	Mail
Erudite	Jan Saunders	Responsable du projet	Mail
Erudite	N/A	Développeur	Mail
Erudite	Geoffrey Menon	Architecte	Mail



## 4 Calendrier

La planification de ce projet se fera en sprint de deux semaines correspondant à une fonctionnalité du système.

Sprint	Début	Fin	Livrables
Base de données	2/10	13/10	Schéma Base de données, Fichiers SQL migration et création des tables
Mise en place du projet et architecture logicielle	16/10	27/10	Diagrammes techniques, base application frontend, base application mobile, base application backend, design
Système d'invitation/connexion	30/10	10/11	Page login/invitation, endpoint login/invitation, système de mailing, système d'authentification
Système de message	13/11	24/11	Fonctionnalité messagerie client, pages, et endpoint de messagerie
Système de chat instantané	27/12	08/12	Fonctionnalité chat client/conseiller, serveur de chat, page de chat
Test et déploiement	11/12	22/12	Mise en place de test, déploiement backend, mise en production application web et mobile, documentations

## 5 Livrables

- **Application web** permettant la mise en relation entre clients et conseillers.
- **Application mobile IOS** permettant aux conseillers de recevoir les notifications des clients.
- **API et Serveur de Chat** permettant de recevoir et d'envoyer les données de ce nouvel extranet.
- **Fichiers SQL** permettant de déployer toutes les données du nouveau système.
- **Documentation technique** des différentes parties du nouvel extranet.

## 6 Tableau des risques

Nature	Description	Gravité	Probabilité	Conséquence
Audit	Demande modification suite audit par l'entreprise Vigil	Peu Grave	Moyenne	Délais
Client	Demande modifications de Velocity en cours de développement	Peu Grave	Haute	Délais
Client	Mauvaise compréhension des besoins de Velocity	Grave	Basse	Délais
Développement	Complications liées au développement du serveur de chat	Grave	Basse	Délais
Développement	Recours à un intervenant extérieur pour régler un problème	Grave	Basse	Délais
Déploiement	Complications liées à l'enregistrement de l'application sur Apple Store	Grave	Basse	Budget
Déploiement	Complications sur la mise en production de l'application web	Modérément Grave	Basse	Délais
Déploiement	Complications sur la mise en place de la base de données	Grave	Basse	Délais
Fonctionnement	Incompatibilité des signaux sonores et visuels entre différents modèles de téléphone	Grave	Basse	Budget
Humain	Maladie ou désistement du projet par un développeur	Grave	Basse	Délais
Humain	Mauvaise coopération des développeurs	Grave	Basse	Délais
Humain	Mauvaise compréhension du projet par les intervenants	Grave	Basse	Délais & Budget
Base de données	Altération de la base de données principale lors de la migration	<u>Très Grave</u>	Basse	Juridique
Piratage	Faible de sécurité et mise en danger de l'intégrité des données clients	<u>Très Grave</u>	Basse	Juridique

## 7 Faisabilité

- **Général**

Suite à l'analyse des besoins et des contraintes du projet, il semble que le nouveau système puisse être développé et mis en service en respectant le délais de 3 mois correspondant aux 6 sprint de deux semaines. Erudite dispose des ressources humaines nécessaire à la réalisation du projet et l'implémentation des fonctionnalités demandées ne devraient pas poser de problèmes majeurs durant le développement.

En prenant compte des ressources nécessaires à la réalisation de cet extranet, le budget initial de 75 000€ devrait être respecté en incluant une marge de sécurité permettant de faire face à un problème inattendu.

- **Backend**

### 1. API

Le système sera axé autour d'une API web qui viendra délivrer les informations nécessaires au fonctionnement du système. Les différentes applications viendront interagir avec cette API via le standard HTTP pour obtenir des données au format JSON. Ces standards sont reconnus et représentent la norme actuelle pour concevoir ce type d'application.

Le développement de la partie backend nécessitera une migration de données de la base actuellement en production, vers la nouvelle base de données de l'application extranet, la réalisation de cette opération ne semble pas poser de problèmes majeurs si elle est effectuée avec sécurité.

Il sera nécessaire de disposer d'un système d'authentification pour garantir l'intégrité des données clients. Des permissions devront être définies afin de limiter les actions de chaque acteurs du système au strict nécessaire. Une Api tel que proposé pour la réalisation du projet convient pour s'assurer de la sécurité des données.

L'architecture proposé ne posera pas de problème de développement particulier, toutes les fonctionnalités demandées peuvent être développer et gérer par l'API directement à l'exception de l'application de chat en temps réel.

### 2. Chat

Le système de chat doit permettre la communication en temps réels d'un client avec son conseiller, un serveur de type websocket pourrait convenir pour ce type d'application, il permettra la création d'un canal privé entres les deux intervenants de façon à communiquer librement et de façon sécurisée.

Les messages devront être chiffrés et uniquement accessibles aux clients ou conseillers concernés.

- **Frontend**

1. Interface Web

L'interface web devra être dynamique afin de recevoir et de générer en temps réel les différentes données provenant du backend. Le standard HTTPS devra être utilisé pour garantir la protection des données.

L'utilisation d'un frontend de type JavaScript convient pour ce type d'application web, il permet d'afficher les données dynamiquement provenant de l'API ou du serveur de chat sans action de l'utilisateur.

Aucune donnée sensible ne devrait être stockés coté navigateur, l'espace de stockage local ne devrait pas être utilisé pour mettre en cache des données ou des clés d'authentification.

2. Interface Mobile

L'interface mobile devra réceptionner des messages clients et en notifier le conseillers par des signaux sonores ou lumineux, ces deux actions sont possibles avec les appareils IOS et ne posent pas de problème de développement particulier, à noter que des différences de compatibilité matérielle pourraient avoir lieu entre différents modèles de téléphone.

L'espace de stockage interne a l'appareil ne devrait pas servir de stockage de données de l'application ou de mise en cache de clés d'authentification.