1. Introduction

Avec l’augmentation des préocupations environnementales et la congestion des routes, le covoiturage a gagné beaucoup de popularité lorsqu’il s’agit de moyens de transport respectueux de l’environnement et bon marché. Le covoiturage est lorsque deux personnes ou plus partagent un trajet dans l’unes de leurs voitures personnelles. Le covoiturage réduit la pollution puisque nous avons moins de voitures sur la route. C’est aussi économique puisque les frais de déplacement sont partagés entre les coureurs.

Voyager seul peut etre stréssant, donc avoir d’autres personnes avec vous lors d’un voyage réduit le stress et aussi l’occasion de socialiser et de rendre le voyage plus amusant.

Le but de ce projet est de développer une application générique pour n’importe quel capooler dans n’importe quel pays. WEGO est le nom que nous avons choisi pour cette application; elle est aussi une application en temps réel : toute personne participant à un trajet peut signaler les autres participants qu’il ou elle est arrivé au point de rendez-vous.

1. Analyse des besoins

La première étape consistera à étudier de différentes applications de covoiturage pour resssortir les besoins avec améliorations. Dans le cadre de ce projet, les fonctionalités à implémenter sont :

* Payer à partir de l’application
* Map picker pour choisir et reperer les différentes localisations.

1. Besoins des utilisateurs et du système

* Description du projet

Cette application regroupera les covoitureurs, des passagers et des conducteurs. Les utilisateurs pourront rechercher ou partager un trajet. L’application sera divisée en deux parties. La première partie sera pour les trajets interurbains ouu les utilisateurs pourront publier leurs trajts et s’enregistreront pour des trajets créés par des autres. En qddition, un système d’enregistrement sera disponible pour notifier les utilisateurs quand le conducteur ou le passager arrivera au point de recontre.

L’autre partie sera consacrée aux trajets fréquents. L’accès à l’application sera accordé aux utilisateurs authentifiées.

* Dictionnaire

Pour éviter l’ambiguité et faciliter la comprehnsion, les termes fréqueement utiiisés sont:

|  |  |
| --- | --- |
| TERME | DEFINITION |
| CONDUCTEUR | Toute personne possédant une voiture et qui voudrait de déplacer d’un endroit à un autre et publie le trajet sur l’application |
| PASSAGER | Toute personne qui ne possède pas de voiture et voudrait rejoindre un conducteur |
| TRAJET REGULIER | Un voyage planifié de longue distnce entre deux points (d’habitude des villes) avec un prix et une heure de départ fixés |
| TRAJET FREQUENT | Une petite ou moyenne distance fréquente (quotidienne ou hebdomadaire) entre un quartier et un lieu de travail. |

* Dépendences

L’application dépendra lourdement sur la géolocalisation et le sytème de cratographie de google. Ceci sera utilisé pour montrer les étinéraires et cartes. Aussi, les donnés GPS seront traités à l’aide de Google Maps. Un système d’authentification sera utilisé pour que les utilisateurs aient accès à l’application. Dans ce cas.

1. Exigences fonctionnelles

* Exigences générales de l’application

1. Se connecter

Toutes les opérations sur l’application requierent que l’utilisateur soit connecté.

1. Modifier les informations du profile

Tous les utilisateurs pourront modifier leurs profiles. Ces informations seront : nom, téléphone, email, type ouc ouleur de voiture s’ils en possèdent une.

1. Noter des conducteurs ou passagers

Les conducteurs ou passagers pourront se noter pour gagner de la réputation. L’importance de noter est que cela encouragera les utilisateurs à bien se comporter durant un trajet. C’est aussi un moyen d’assurer si un utilisateur est digne de confiance ou pas.

* Trajets réguliers

1. Créer un trajet régulier

Le conducteur pourra créer un nouveau trajet qui serq vue par les passagers cherchant un trajet. L’application montrera les informations du trajet qui consiste de la destination, l’origine, le point de rencontre (qui pourra etre visible sur la carte), l’heure et la date de départ, une estimation du temps d’arrivée et les préférences pour le trajet (nombres de places libres, prix, taille de bagage, fumeurs, animaux, arrets). Après avoir procuré ces informations, l’utilisateur pourra les publier pour trouver des passagers.

1. Rechercher un trajet régulier

Un passager à la recherche d’un trajet régulier pourra utiliser un formulaire de recherche qui demandera des informations nécéssaires pour le trajet.

1. Notifier les trajets

Quand un conducteur ou passager est arrivé au point de rencontre, il pourra le notifier à l’autre utilisateur pour lui montrer sa ponctualité.

* Trajet fréquent

1. Créer un trajet fréquent

Un utilisateur pourra créer un trajet fréquent ouu il montrera l’origine et la destination, l’heure de départ et de retour et la fréquence ( journalière ou hebdomadaire ).

1. Rechercher un trajet fréquent

Un passager pourra chercher un trajet fréquent qu’il pourra rejoindre. Le passager devra spécifier le quartir de départ, la destination, l’heure de départ et la fréquence. L’application essayera de matcher sa recherche avec le trajet qui conviendra le mieux.

1. Exigences non-fonctionnelles
2. Performance

L’application devra avoir un temps de réponse très court puisse que la rencontre entre le conducteur et passager se fait grace aux notifications. En d’autres mots, le serveur doit etre capable de traiter et propager les notification instatanement.

1. Evolutivité

L’appliction doit s’adapter à une augmentation du nombr d’utilisateurs. Il devra etre capable de gérer entre 10 000 et 100 000 utilisateurs . ET aussi entre 100 000 et 1 million d’utilisateurs.

1. Extensibilité

L’application devra etre extensible de manière à pouvoir supporter plusieurs plateformes tel que IOS, windows.

1. Disponibilté

Puisse que beaucoup d’informations concernant les trajets et les enregistrement seront disponible via l’application, it devra etre très disponible.

5) Confidentialité et sécurité

L’application devra s’assurer de la confidentialité des utilisateurs et des trajets auxquels ils participent et leurs comptes.

6) Maintenabilité

Puisse que l’application pourrait etre developpée dans le futur pour y ajouter d’autres fonctionnalités, il doit etre facilement maintenable.

1. Technologies utilisés

Le choix des technologies pour cette application est essentiel pour son succès. Ces technologies devront remplir les besoins cités plus haut. A travers ce document, nous vous présenterons les différentes technologies choisies et les raisons du choix pour le coté serveur et le coté client.

1. Technologies coté serveur
2. Serveur d’application

Pour le serveur d’application, Node Js a été choisi parce qu’il d'adapte au nombre croissant d’utilisateurs. Des systèmes de notifications sont offerts comme un service web ce qui le rend multiplateforme. Il offre plus de performance comparé aux technologies comme JEE et .Net.



1. Stockage et persistance des données

Node Js et le NoSQL formattent les données de la meme manière en utilisant le format JSON, ce qui améliore la performence parceque le reformattage de donnés ne sera plus nécéssaire

Le NoSQL procure une manière flexible de stocker les données ouu les règles sont établies par les développeurs et non par le modèle relationnel comme avec le SQL. Dans le contexte de cette application, Nous utiliserons Mngo DB qui est une base de données NoSQL qui utilise le JSON comme format de stockage.



1. Technologies coté client