

codeur^{.com}

```
content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no"
<link rel="shortcut icon" href="/favicon.ico" type="image/x-icon">
<link rel="icon" href="/favicon.ico" type="image/x-icon">
<!-- CSS -->
<link type="text/css" rel="stylesheet" href="css/materialize.min.css" media="screen, projection">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.6.0/css/font-awesome.min.css">
<link rel="stylesheet" href="/css/animate.css">
<link rel="stylesheet" href="css/theme.css">
</head>
<body>
  <!-- banner -->
  <div class="banner">
    <nav class="nav">
      <div class="nav-wrapper">
        <div class="container">
          <a href="#" class="brand-logo hide-on-med-and-up"><span style="font-weight: bold; display: inline-block; width: 100px; height: 1.2em; vertical-align: middle; background: linear-gradient(to right, #009688, #00bcd4); background-size: 100% 100%; border: 1px solid #00bcd4; border-radius: 15px; margin-right: 10px;">
        </div>
      </div>
    </nav>
  </div>
</body>
</html>
```

Cahier des charges d'un logiciel

SOMMAIRE

- A. Développement du logiciel
 - A.1. Caractéristiques et fonctionnalités
 - A.2. Structure du logiciel
- B. Charte graphique du logiciel
- C. Les spécificités et les livrables
 - C.1. Les contraintes techniques
 - C.2. Le planning

Cahier des charges d'un logiciel

A. DÉVELOPPEMENT DU LOGICIEL

A.1. Caractéristiques et fonctionnalités

(Le X indique le choix)

Quel type de logiciel souhaitez-vous développer ?

Partie 1 (un choix maximum) :

- ☐ Logiciel de système d'exploitation
- ☒ X Logiciel d'application

Partie 2 (un choix maximum) :

- ☐ Logiciel propriétaire et payant
Ce type de logiciel est vendu à l'utilisateur
- ☒ X Logiciel gratuit
L'utilisateur peut se procurer le logiciel gratuitement par exemple téléchargement par exemple
- ☐

Précisez la catégorie de votre logiciel

- | | |
|---------------------------------------------------|-------------------------------------------------|
| <input type="checkbox"/> Antivirus | <input type="checkbox"/> Courrier électronique |
| <input type="checkbox"/> Navigateur Internet | <input type="checkbox"/> Messagerie instantanée |
| <input type="checkbox"/> Voix sur IP (type Skype) | <input type="checkbox"/> Photo |
| <input type="checkbox"/> Bureautique | <input type="checkbox"/> Lecture audio/vidéo |
| <input type="checkbox"/> Jeux | <input type="checkbox"/> Dessin / Graphisme |
| <input type="checkbox"/> Torrent | <input type="checkbox"/> Mixage / Montage |
| <input checked="" type="checkbox"/> X Autre | |

Si le choix « Autre » a été coché, écrire la catégorie ici :

Système de navigation GPS

Cahier des charges d'un logiciel

Fonctionnalités du logiciel :

Le programme assurera les fonctions de base d'un GPS, à savoir :

Tout d'abord, le tracé de l'itinéraire du point de départ au point de destination choisi. Ensuite, plusieurs choix d'itinéraires seront proposés, en fonction de paramètres choisis par l'utilisateur, comme le chemin le plus court, en nombre de kilomètres à parcourir, ou le plus rapide, si celui-ci doit éviter ou non les péages. Aussi, le programme spécifiera les villes intermédiaires situées sur l'itinéraire, dont les noms seront indiqués. Il sera ainsi indiqué s'il faudra y passer ou les éviter. Il sera également possible d'indiquer ou non le type de villes intermédiaires, par exemple s'il s'agit de grandes villes ou de villes touristiques. Le programme précisera également les caractéristiques des routes empruntées, que ce soient leur type (Exemple : Autoroute ou route départementale), si des radars sont situés dessus ou non, la vitesse limite des routes à suivre selon l'itinéraire.

Ensuite, le programme recalculera les itinéraires dans le cas où l'utilisateur se serait trompé, en se trompant de sortie par exemple.

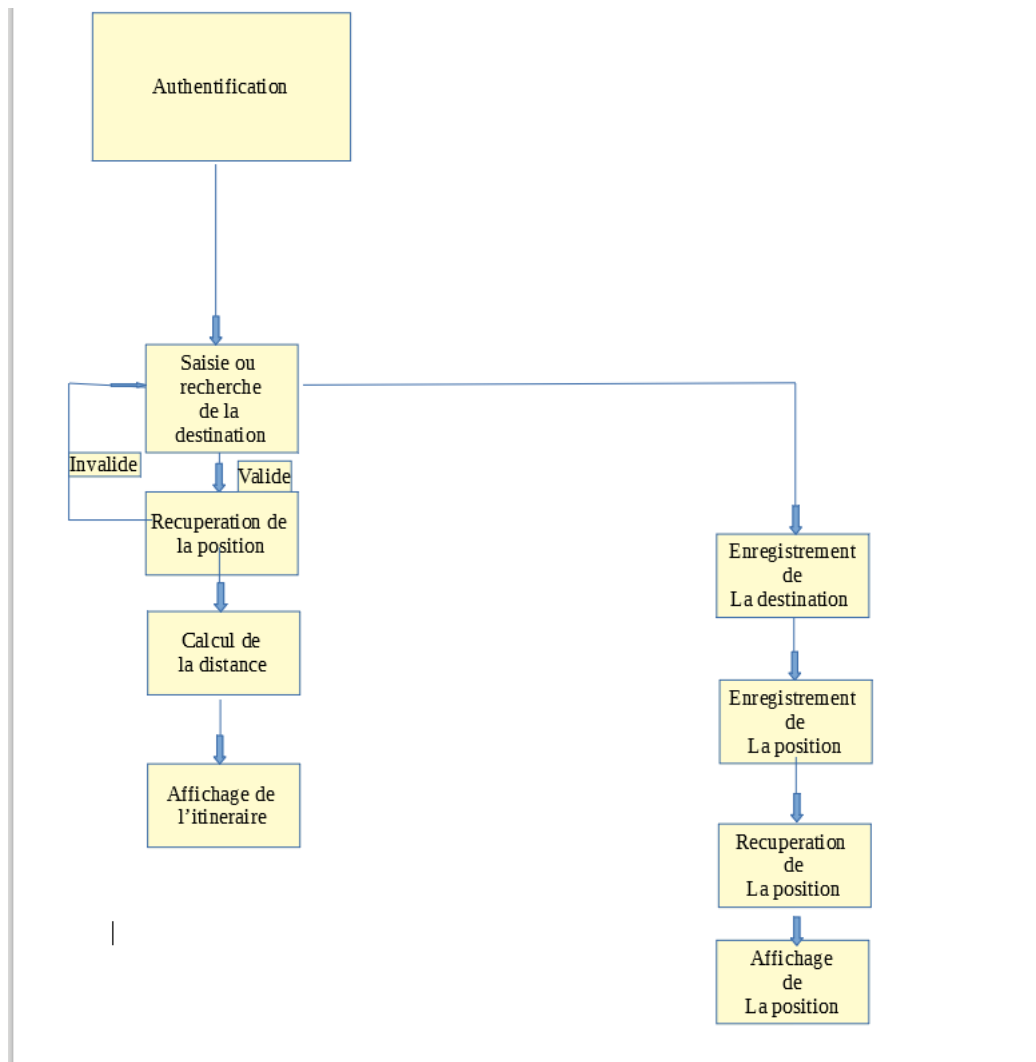
Cependant, une notification sera envoyée si jamais l'un des choix n'est plus respecté en cours de navigation. Il sera alors demandé s'il faudra conserver ou non les choix effectués. A noter que si jamais le programme ne peut pas prendre en compte les contraintes, alors une notification sera alors affichée pour prévenir l'utilisateur des changements de paramètres de navigation.

Liste des généralités du programme:

Le programme rendra service à tout utilisateur ayant installé le programme sur un appareil adapté. Ainsi, il agira sur l'itinéraire emprunté ou souhaité par l'utilisateur, dans le but d'éviter qu'il ne se perde. Pour se faire, un calcul de l'estimation du temps de trajet sera effectué en temps réel. Le programme déterminera également la position actuelle sur la latitude et la longitude en temps réel, tout comme quelle est la ville la plus proche en distance. Il en sera de même pour la consommation d'essence et du volume de CO2 expulsé, en fonction du type de véhicule. Pour que la formule de calcul de consommation d'essence et d'émission de CO2 soit fonctionnelle, il faudra rentrer le type de véhicule avant l'entrée de l'itinéraire à travers une interface dédiée. Ensuite, le programme déterminera la vitesse courante en temps réel, afin de vérifier que le conducteur ne dépasse pas les limitations de vitesse, auquel cas il en sera notifié. Le temps durant lequel l'utilisateur a roulé sera également calculé en temps réel, dans le but de prévenir le conducteur par un message si jamais celui-ci a déjà roulé pendant 2 heures. Ensuite, le programme doit prévenir l'utilisateur si la ville de départ et/ou de destination ne sont pas reconnues, pour qu'il puisse en saisir une autre à la place. De même, il sera prévenu si l'itinéraire voulu est inexistant, pour qu'il puisse chercher une autre ville.

Cahier des charges d'un logiciel

Enfin, le programme déterminera si l'on est bien arrivé à destination, afin de demander confirmation à l'utilisateur, qui doit répondre pour confirmer si c'est la bonne destination ou non.



Scénario de l'utilisation du programme

Cahier des charges d'un logiciel

A.2. Structure du logiciel

Définissez les différentes rubriques et sous-rubriques de votre logiciel qui constitueront le menu et le sous-menu

Tout d'abord, un menu s'affichera pour le choix des paramètres, comme le modèle du véhicule, le prix maximal à payer pour les péages. Il sera possible de retourner à ce menu.

Ensuite, le programme disposera d'une interface de GPS basique, à savoir le trajet, avec le chemin à suivre en couleur différente ainsi que les informations sur les villes. Enfin, le logiciel interpellera l'utilisateur à l'oral si le trajet doit être modifié ou si les contraintes ne sont plus respectées.

B. CHARTE GRAPHIQUE DU LOGICIEL

Définissez les polices de caractère (typographie, taille, aspect) et les couleurs

Utilisation d'une interface graphique en JAVA, non définie pour l'instant.
Peut-être JFX.

Cahier des charges d'un logiciel

C. LES SPECIFICITES ET LES LIVRABLES

C.1. Les contraintes techniques

Précisez vos attentes concernant les besoins connexes à ce projet que le prestataire devra fournir :

- Formation à l'utilisation du logiciel
- Assurer la maintenance
- Proposer des astreintes pour le dépannage
- Rédaction de la documentation
- ...

Contraintes de fonctionnement :

Tout d'abord, la connection à une base de données permettra d'effectuer le calcul des trajets. Ensuite, un système de login pourrait éventuellement être implémenté, pour assurer une compatibilité entre un utilisateur et plusieurs appareils compatibles. Cependant, l'utilisateur aura la possibilité d'utiliser le logiciel sans avoir de compte, comme un mode hors-ligne. Ensuite, la sauvegarde d'un historique d'adresses de destination, ainsi que des préférences de navigation pour faciliter les sélections de trajet.

Contraintes techniques :

Premièrement, le programme doit pouvoir être exécuté sur tout appareil disposant de la dernière version de JAVA. En conséquence, le programme sera développé en JAVA. Puis, les calculs doivent être rapides dans la détermination des itinéraires, pour assurer un confort lors de la navigation. Ensuite, pour des raisons évidentes, il est impossible de choisir une même ville en tant que point de départ et de destination en même temps.

Comme il s'agit d'un programme informatique, aucune contrainte industrielle ne sera demandée. Cependant, le programme respectera une contrainte environnementale dans le cas où l'utilisateur souhaiterait emprunter le chemin émettant le moins de CO2 ou étant le plus économique en volume d'essence consommé. Aussi, le programme assurera une contrainte sécuritaire en indiquant le chemin le plus sûr, si la limitation de vitesse est dépassée ou s'il a roulé au moins 2 heures, et lui suggérer ainsi de faire une pause dans la ville la plus proche.

Aussi, nous respecterons la contrainte économique, dans le cas où l'utilisateur souhaiterait emprunter le chemin avec le moins de péages possible, voire aucun.

Cahier des charges d'un logiciel

Critères de mesures des contraintes :

Tout d'abord, pour la contrainte économique, déterminer le prix à payer en péage sur tous les itinéraires possibles. En fonction du seuil choisi par l'utilisateur, proposer le chemin respectant la contrainte.

Ensuite, concernant les contraintes écologiques, les itinéraires détermineront le volume de CO2 dégagé, ainsi que le volume d'essence consommé, puis le programme choisira le trajet correspondant au mieux aux contraintes.

Enfin, pour les contraintes d'ergonomie, le programme aura une interface simple d'utilisation, afin d'éviter que l'utilisateur comprenne facilement le programme. On pourra également penser à implémenter une interface tactile, si l'appareil supporte cette technologie.

C.2. Le planning

Ajoutez un agenda des dates souhaitées pour la validation des différentes étapes :

- Date de la création des prototypes
- Dates des tests
- Date de la validation des prototypes
- Date de mise en ligne

Le planning respectera l'agenda du cours, à savoir:

- Rendu du cahier de charges le 11/02/2019.
- Rendu de l'analyse le 18/02/2019, incluant les diagrammes de classes, d'utilisation, UML,
- Rendu de la conception le 11/03/2019
- Rendu de l'implémentation le 17/5/2019, avec les classes de test.

Les prototypes seront dans ce cas les classes de test, qui seront implémentés après les classes, pour résoudre les bugs, dans le cas où les tests détectent les bugs.