KHRAIMECHE Salim - DUCHESNAY Simon - DEBAS Matthieu
GOMIS Michel - RAKOTONDRATSIMA Mahefa
Institut Universitaire de Technologie
LP SICI - Année 2014/2015

Projet de Tutorat MP2

Mise en place d'une plate-forme web d'optimisation de tournées de livraison

Supervisé par : M. DUVALLET Claude

Remerciements

Nous aimerions tout d'abord remercier, M. DUVALLET et M. DIARRASSOUBA, pour le sujet proposé. Ce projet est complet, de part **sa mise en situation réelle**, ainsi que **son nombre de tests et de tâches à exécuter** pour la réalisation de ce projet. Cet ensemble de tâches nous a permis de pouvoir **spécifier, les différents langages acquis au sein de notre formation**.

Nous souhaiterions également remercier M. DECLOMESNIL et M. LEGRAS, pour les nombreux cours, sur la rédaction de documents officiels. Les connaissances ainsi obtenues, comme la mise en place du cahier des charges, des fiches de test, des spécifications fonctionnelles et techniques, **nous ont permis de rendre notre projet bien plus concret.**

Enfin, **nous adressons des remerciements à M. GRIEU**, pour la gestion du travail. Les travaux et cours sur la planification des tâches, sur l'anticipation d'avancement du projet nous ont permis d'avancer dans le projet dans les meilleures conditions. **Son expérience et sa formation sur GANTT et sur la méthode PERT nous ont permis de réaliser des documents valides à présenter pour appuyer notre gestion de projet.**

Sommaire

Introduction

Dans le cadre de la Licence Professionnelle S.I.C.I., nous avions choisir de prendre le sujet :

Mise en place d'une plate-forme web d'optimisation tournées de livraison

Les travaux à rendre pour ce projet se présentent **sous la forme d'un rapport écrit**, présentant **un compte-rendu de notre projet de A à Z.** Ce rendu devra comporter le projet complet (arborescence et fichier), ainsi qu'**une présentation orale** avec des diapositives.

Le projet sera à rendre au commanditaire, quelques jours avant sa présentation orale, afin de présenter le rendu définitif à ce dernier.

Ce travail est à réaliser en groupe, composé de cinq personnes de la même promotion, afin de réaliser tous les travaux demandés précédemment.

Ce projet conséquent sera montré au jury enseignant, **lors d'un oral de présentation de vingt-cinq minutes maximum**. Pendant cette durée, les étudiants devront exposer leur réalisation finale, par une possible démonstration, mais également argumenter sur les différentes phases de leur projet.

Pour ce rapport écrit, nous allons diviser ce compte rendu en plusieurs parties distinctes, afin de présenter le projet sous toutes ses différentes phases abordées.

Premièrement, une partie « Définition du projet » vient présenter le projet, l'équipe en charge de ce dernier,le planning et les outils utilisés pour sa réalisation.

Deuxièmement, la partie « Réalisation du projet », viendra présenter la conception de notre site dans son ensemble : cible de notre site, choix du contenu, arborescence et le contenu lui-même.

La troisième partie « Analyse des Résultats » nous permettra d'évoquer notre recul sur notre projet et nos réalisations. **Enfin, une conclusion** viendra clôturer notre rapport.

Définition du projet

Cahier des charges

Commanditaire : M. DIARRASSOUBA Ibrahima

Objectifs/Souhaits

Notre client souhaite obtenir une solution de gestion de trajets et de tournées pour une application. Cette dernière a pour objectif de servir de simulation au sein de son propre cours.

Cette simulation comporte deux actions :

- Dans un premier temps, ce service doit permettre à l'usager d'y placer une liste de clients, de commandes par le biais de cette interface. Cette dernière ira questionner un serveur, qui se chargera d'optimiser le trajet et d'y établir un ordre.

- Dans un second temps, cette application doit pouvoir interpréter la réponse du serveur d'optimisation, afficher à l'utilisateur un récapitulatif complet du trajet et ainsi qu'un aperçu d'un trajet au sein d'une carte. Le client pourra alors ainsi commencer sa tournée.

Entrée/Affichage des informations :
L'envoi de l'usager devra contenir les informations suivantes :

- Informations relatives à chaque client : Id et Adresse du client,
- Adresse de dépôt / de matériaux,
- Indications relatives au colis : Poids/Charge/Surface,
- Contraintes relatives au transport : Temps/Pause/Véhicule,

L'Affichage de la réponse devra comporter les informations suivantes :

- Numéro de tournée,
- Poids/Surface des commandes,
- Coûts de notre tournée (fixes et variables),
- Affichage complet de chaque étape (client),

Spécifications fonctionnelles

SF_001 : Support du produit

L'application devra être disponible sur un support PC.

SF_002 : Navigateur Internet

Ce webservice doit être compatible avec les navigateurs internet Google Chrome et Mozilla Firefox (compatibilité avec les 5 dernières versions).

SF_003 : Utilisation d'un service de mapping

Ce webservice devra contenir un système de mapping.

SF_004 : Formulaire

Mise en place d'un formulaire, pour insérer les différentes informations nécessaires à la création d'une tournée.

SF_005 : Modularité du formulaire

Possibilité d'ajouter des champs au formulaire, lorsque l'on a un grand nombre de clients, ou plus d'informations au formulaire.

SF_006 : Envoi du formulaire

Permettre au webservice de récupérer les informations du formulaire et de créer une requête à envoyer au serveur d'optimisation

SF_007 : Réponse du serveur

L'application doit être en mesure d'écouter et de réceptionner la réponse du serveur.

SF_008 : ***Mise en place d'un distancier***

Le distancier est placé ici, pour obtenir les distances entre deux points.

SF_009 : ***Traitement de la réponse du serveur***

Le webservice doit être en mesure de traiter la réponse afin de créer l'affichage final pour notre usager.

SF_010 : ***Affichage de la réponse***

Grâce au traitement de la réponse, l'application doit permettre un affichage de notre tournée : informations complètes textuelles, ainsi qu'un affichage du trajet par le biais d'une map.

SF_011 : ***Choix des tournées***

Mettre en place une sélection, afin de réunir les tournées et de les choisir chacune leur tour.

Spécifications techniques

ST_001 : Création d'une interface internet
Interface internet finalisé en pages HTML/CSS
Page généré en HTML/CSS contenant du PHP et du JavaScript.

ST_002 : Système de Mapping
Utilisation de l'API Google Maps, afin de générer la map et d'y implémenter les différents trajets. Module en JavaScript.

ST_003 : Création de Client
Possibilité de créer des clients pour garder les informations en mémoire.
Client en JavaSript ou en PHP.

ST_004 : Création de Tournée
Possibilité de créer des tournées pour garder les informations en mémoire.
Tournée en JavaSript ou en PHP.

ST_005 : Création d'une requête vers le serveur
Socket PHP qui générera une chaîne XML à envoyer au serveur.

ST_006 : Réponse du Serveur
Écoute, réalisée en PHP, qui viendra récupérer la chaîne XML réponse du serveur.

ST_007 : Traitement de la réponse XML
Utilisation d'un module JavaScript pour parcourir le fichier et générer les tournées et les clients.

ST_008 : ***Affichage des résultats***

Traitement JavaScript pour modifier les différents éléments de la page HTML.

ST_009 : ***Affichage de la Map***

Utilisation de Tournées et des Clients pour parvenir à placer les différents trajets sur la map. Module PHP ou JavaScript.

ST_010 : ***Sélection des Tournées***

Utilisation d'un élément HTML (select), afin de générer une liste de toutes les tournées. Utilisation du HTML couplé au JavaScript, afin de générer la liste des informations et la création du trajet en simultané.

ST_011 : ***Formulaire***

Création d'un formulaire en HTML avec une redirection vers une page de traitement PHP.

ST_012 : ***Modularité du Formulaire***

Utilisation de fonctions JavaScript afin de régénérer des éléments HTML du contenu du formulaire

Fiches de tests

« Tests Graphiques »

FT_001 : La Cartographie
Affichage de la map (API Google Maps) au sein de notre site internet

FT_002 : Affichage de notre page Internet
Affichage de notre Select, de nos informations et de notre map, en simultanée. Mise en page CSS.

FT_003 : Navigation au sein de la map
Possibilité d'interagir avec la map : zoom, déplacement, clic,...

FT_004 : Formulaire de page
Affichage du formulaire de création des tournées.

« Tests Fonctionnels »

FT_005 : Modularité du formulaire
Le module JavaScript doit être en mesure de pouvoir générer des éléments supplémentaires dans la formulaire, ainsi que de rendre possible leur suppression.

FT_006 : Génération de la chaîne d'envoi XML
Récupération des informations du formulaires et génération du String XML.

FT_007 : Envoi de la requête XML
Bonne réception de la chaine, en intégralité, par la serveur d'optimisation.

FT_008 : Réception du Serveur
Voir si le serveur a bien reçu la requête XML de notre service.

FT_009 : ***Écoute de la réponse serveur.***

La Page PHP doit être capable de devoir récupérer la chaîne XML en réponse du serveur d'optimisation.

FT_010 : ***Génération des tournées par rapport à la chaîne XML réponse.***

Mise en mémoire des informations de chaque tournée, en JavaScript.

FT_011 : ***Génération des clients, par rapport à la chaîne XML réponse.***

Mise en mémoire, par le biais du JavaScript, des information de chaque client de la réponse en XML.

FT_012 : ***Remplissage du DOM HTML***

Le JavaScript doit être en mesure de remplir les éléments HTML, afin d'afficher les informations des tournées et des clients

FT_013 : ***Initialisation de l'API Google Maps***

Mise en place de l'API, avec les paramètres généraux.

FT_014 : ***Placement des markers sur la carte***

Grâce aux Prototypes JavaScript, l'application doit pouvoir placer les points sur la map en fonction des coordonnées de chaque étape d'une tournée.

FT_015 : ***Interaction des tournées***

Utilisation de la balise ‹select›, afin de pouvoir switcher et générer dynamiquement le trajet sur la map, ainsi que l'affichage des étapes de cette dernière.

Matrice de traçabilité

Cahier des charges	Spécifications Fonctionnelles	Spécifications Techniques	Fiches de Tests
Interface WEB - Ergonomie	1 2	1	2
API Map Google	3	2	1 3 13
Formulaire de création de tournées	4 5	11 12	4 5
Socket PHP / Envoi + Serveur	6 8	5	6 7 8
Réponse Serveur (Écoute)	7	6	9
Génération des tournées/client (JS)	9	3 4 7	10 11
Affichage de la réponse serveur	10	8 9	12 14
Sélection des tournées (select)	11	10	15

Attentes/Objectifs du projet

Notre projet comporte plusieurs tâches distinctes, qui permettront de traiter tous les aspects d'un projet, tels que la conduite, la synthèse ou encore son organisation temporelle.

Les tâches sont les suivantes :

- **Réalisation du rapport écrit :** le rapport devra comporter les différentes informations sur l'ensemble du projet, la méthodologie, la description des étapes.

- **Conception/Production de notre projet.**

- **Soutenance Orale d'Examen:** devant un jury composé d'enseignants de la section informatique ou de candides, le groupe devra présenter le projet dans son intégralité.

Les objectifs de ce projet comportent trois points précis :

- **Un projet ancré dans la réalité :** en effet, les projets proposés lors du MP2 sont bien concrets, avec un besoin et un client réels. Par conséquent, l'équipe de projet devra s'entretenir tout au long du projet avec le commanditaire. La progression du projet doit pouvoir être réguliere, afin de pouvoir présenter les nouveautés au client à chaque nouvelle rencontre.

- **Le travail en équipe :** le projet étant plus conséquent que ceux proposés au MP1, l'organisation et la répartition des tâches sont des éléments cruciaux de la réussite du projet.

- **Présenter et vendre son projet à l'oral :** la communication est importante au sein du projet MP2. Que ce soit pour le commanditaire ou pour le jury, l'équipe doit être capable d'expliquer le fonctionnement du projet, mais également argumenter ses choix de conception et de développement. Les différentes rencontres et l'examen oral auront pour but d'obtenir une aisance à l'orale et de se faire mieux comprendre auprès d'un client, néophyte ou non.

Outils utilisés pour le projet

Développement du site WEB :

Sublime Text
- Éditeur de texte avec colorisation syntaxique.
- Arborescence des fichiers.

API Google Maps
- API simple à mettre en place et à manipuler
- Service gratuit et complet.

Planification du projet :

Apache Subversion (SVN)
- Gestion du projet, diagramme, chemin critique.
- Mise à disposition des versions du projet pour l'équipe et le client.

GANTT Project
- Gestion du projet, diagramme, chemin critique.
- Diagramme de PERT.

Rédaction/Présentation des documents :

Adobe InDesign CC
- Rédaction, Mise en page, Exportation en PDF.

Microsoft Office PowerPoint 2013
- Mise en place des diapositives
- Rédaction du contenu.

Navigateurs Internet :

Google Chrome
- Navigateur efficace, utilisé par la plupart des internautes.

Mozilla Firefox
- Navigateur utilisé par notre commanditiare.

Équipe de projet

L'équipe est constituée de 5 étudiants de Licence professionnelle SICI :

- **Salim KHRAIMECHE** (Chef de projet)
- **Matthieu DEBAS**
- **Simon DUCHESNAY**
- **Michel GOMIS**
- **Mahefa RAKOTONDRATSIMA**

Au vu des tâches et de la compléxité de chacune, il est logique que la plupart de celles-ci ont été réalisées en groupe. Voici la liste des tâches et la répartition des éléments au sein de ces dernières :

cf. Annexe 9, p.XV

- *Interface WEB (HTML/CSS) :*
  - › Simon
- *Recherches sur les différentes API :*
  - › Michel/Salim
- *Implémentation de l'API Google Maps au sein de la page :*
  - › Salim
- *Mise en place du formulaire (HTML/PHP/CSS) :*
  - › Mahefa / Matthieu / Michel
- *Mise en place des sockets WEB :*
  - › Salim
- *Traitement des envois/réponses XML :*
  - › Mahefa / Simon
- *Création des prototypes JS :*
  - › Simon
- *Réalisation du PowerPoint :*
  - › Salim / Matthieu

› *Réalisation du rapport écrit :*

- › Simon / Michel

Réalisation du projet

Notre projet s'est tourné autour des rencontres avec Mr DIARRASSOUBA, qui est le commanditaire de ce projet. Chaque rencontre a été déterminante, pour le suivi de notre projet, et les principaux objectifs à accomplir pour la suivante.

Par conséquent, notre projet a subi une étape de transformation de projet à chaque rencontre. Cela nous **permettra d'expliquer davantage les choix de conception, les remises en question du MP2, ainsi que les réorientations du projet.**

Réalisation du projet

Rencontre 1 : Mise en place du projet

Durant cette première rencontre, nous avons placé les **objectifs primaires** de notre projet, à savoir permettre à l'internaute de rentrer des informations, de les expédier, sur le serveur qui apportera une solution qui sera interprété par la page web de l'usager.

Après la rencontre, nous nous sommes fixés plusieurs objectifs :

1 - Mise en place d'une map routière : Notre service web doit pouvoir être en mesure de configurer un trajet, de placer les points et de les lier, afin de présenter significativement le trajet à l'usager. Pour cette étape, le résultat peut être obtenu en plaçant des données brutes dans le programme.

cf. Annexe 1, p.I-III

2 - Trouver un moyen de pouvoir dialoguer avec le serveur d'optimisation (délivré au préalable par notre commanditaire). Le moyen évoqué lors de cette intervention, était la possibilité d'utiliser des sockets PHP. L'objectif pour cette étape est donc de retrouver la chaîne envoyé sur l'affichage du serveur en ligne.

cf. Annexe 2, p.IV

3 - Possibilité de placer en mémoire les informations nécessaires à la réalisation de nos tournées. L'éventualité d'un objet pour les clients, de tournées et de dépôts est évoquée.

cf. Annexe 3, p.V-VII

4 - Mettre en place le formulaire, pour pouvoir entrer chaque information du trajet et pouvoir par la suite, les réutiliser au sein d'autres pages. Mise en place d'un formulaire en HTML/PHP.

cf. Annexe 4, p.VIII-IX

À la suite de cette première étape, nous en sommes arrivés à plusieurs problèmes de conception et de développement. **La limite de notre API de Mapping :** L'API Google Maps nous propose de créer des trajets, et de placer des points sur la map. Cependant, **le service gratuit de base de Google limite l'action à plusieurs paramètres :**

› ***Limite d'itinéraires/de tournées :***
Nous ne pouvons créer que 10 tournées par page au maximum.

› ***Limite de points par tournée :***
Chaque tournée ne peut prendre en compte que 10 points.

› ***Limite du nombre de requêtes pour le géocoder :***
Le service ne propose qu'un nombre de 2000 requêtes par jour.

Réalisation du projet

Rencontre 2 : Mise en place du formulaire et de la réponse

Durant la deuxième rencontre, nous avons présenté les résultats apportés au commanditaire. **Ce dernier perçoit bien la mise en place des sockets PHP, de la réception du serveur et de l'envoi d'une réponse par ce dernier**. Notre page de réception de réponse **récupère bien une chaîne XML, afin de la parcourir et d'y apporter une réponse visuelle.**

Pour ce qui est de la mise en place du mapping, M. DIARRASSOUBA **nous laisse un nouveau délai afin de pouvoir tester des nouveaux systèmes de cartographie**, ainsi que de mise en place de trajet. La tendance nous pousse à essayer de regarder au niveau du deuxième service à notre disposition, Open Street Map.

Dès lors, nous avons maintenant de nouveaux objectifs :

1 – Proposer une solution à envoyer depuis le serveur :
Le serveur d'optimisation doit être en mesure d'envoyer une réponse « cohérente », et doit être recevable pour la page réponse du client.

cf. Annexe 5, p.X

2 – Mise en place de la réponse / Affichage de la map :
Cette réponse doit être correctement interprété, et le client doit pouvoir obtenir à l'écran l'information complète de ses tournées, ainsi que le trajet correspondant.

cf. Annexe 1, p.I-III

3 – Possibilité de présenter un choix de tournée :
Si notre solution se découpe en plusieurs tournées (selon les contraintes), l'usager doit être en mesure de visionner chaque trajet, avec un rafraîchissement automatique de la page.

cf. Annexe 6, p.XI

À la suite de nos avancés, nous sommes confrontés aux problèmes suivants :

- **Open Street Map est trop complexe à gérer,** tant sur la mise en place de la carte, des points, mais également la mise en place des trajets. Par conséquent, nous sommes repassés sur l'API Google Maps.

- **Nous n'avons pas encore une mise en place complète du formulaire complet.** Par conséquent, nous devrons nous attaquer davantage sur le formulaire et la génération de la requête à envoyer. Le traitement de la réponse et l'envoi de socket PHP étant réalisés, il faut donc se centrer davantage sur le formulaire.

- Trouver un moyen d'avoir les coordonnées géographiques des villes, afin de pouvoir générer plus facilement les trajets, et éviter le problèmes des limites de requêtes.

Réalisation du projet

Rencontre 3 : Distancier/Remplissage et lien du formulaire

Durant la dernière rencontre, nous avons réalisé une démonstration des différentes étapes : Envoi d'une requête « type » au serveur d'optimisation, vérifier la réception sur ce dernier, son envoi de solution et le traitement final de la solution VRP XML.

À la suite de la présentation, nous avons recueilli les critiques et volontés d'amélioration de M. DIARRASSOUBA :

1 - La présentation du trajet ne correspond pas totalement aux attentes du commanditaire. En effet, la liaison rectiligne, entre les points d'une tournée, n'est pas très significative. Par conséquent, il nous faut trouver un moyen de détourner les limites de points, pour optimiser les contours du tracé de tournée.

cf. Annexe 1, p.I-III

2 - Le client souhaiterait une solution, afin d'avoir les distances sur la carte. Nous évoquons la mise en place d'un distancier ou tout autre moyen d'obtenir des informations, vis-à-vis de la map.

cf. Annexe 1, p.I-III

3 - Il souhaiterait voir l'action complète du formulaire, ainsi que les différentes contraintes, liés au transport et à la livraison (heures de livraison, temps de pause, temps maximal de trajet, etc).

cf. Annexe 7, p.XII-XIII

Analyse des résultats

Points forts de notre application

Notre application web possède toutes les étapes de simulation de la création de tournées. Le formulaire permet de placer toutes les informations nécessaires, que ce soit les adresses de passage ou les contraintes liées au trajet.
La géolocalisation des villes et la gestion des contraintes sont respectées selon les critères de bases et les techniques efficaces.

L'envoi de notre chaine d'informations, au serveur, est bien envoyé. Les sockets PHP ont été correctement mises en place, pour l'envoi au serveur d'optimisation. **Le serveur reçoit bien notre requête**. Le traitement, réalisé sous Java, nous permet de réaliser la meilleure solution, avant de générer notre chaine XML.

Pour ce qui est de la réponse, notre page d'affichage reçoit bien la chaine XML du serveur. La réponse est bien parcourue, mise en mémoire par le biais de prototypes JavaScript et ensuite mise en place sur la carte. **La mise en place d'une sélection, couplée au JavaScript permet de générer et afficher dynamiquement les différentes solutions de transport.**

Points faibles de notre application

Néanmoins, notre application possède des points faibles. D'un côté, il y a le manque de temps, pour l'amélioration technique et esthétique de notre application. De l'autre, il s'agit de problèmes liés à certains services utilisés et à certaines parties de développement.

Premièrement, nous avons l'interface du site et son ergonomie. Le nombre de tâches et d'éléments d'architecture à réaliser ont été tellement nombreux et complexes pour nous que l'interface n'a pas été pensée davantage. Étant donné que le principal objectif était une simulation efficace, notre choix s'est porté naturellement sur un fonctionnement qui soit le plus efficace possible. Bien entendu, cela a peiné sur le plan graphisme et sur le responsive design de notre site.

Deuxièmement, l'API Google Maps nous a posé bon nombre de soucis, dont certains encore non-résolvables : par exemple, le nombre limite d'étapes a été palié par un découpage et lien de chaque trajet, mais cela n'enlève toujours pas la limite de requêtes journalière. Il faudra donc tâcher de revoir ce point, ou d'observer attentivement les services plus optimisés, mais payant, de ces API.

Enfin, notre dernier problème vient d'un soucis de création de trajet. La plupart des requêtes de trajet se déroulent normalement, mais il peut arriver que certaines villes nous posent problème. En effet, ces dernières sont bien trouvées, et marquées, mais pas sur un élément traçable et routier. L'élément pointé se permet pas de bien réaliser le trajet. Les chiffres, ainsi que le trait, sont donc faussés, ou non éxécutés par l'application.

Conclusion

Pour conclure, nous pouvons dire que ce projet était l'un des projets les plus complets réalisés lors de notre formation.

Premièrement, par son ancrage dans une situation réelle : la possibilité d'interagir avec un client réel est un moyen de se confronter à des objectifs concrets. Les délais courts nous ont permis d'acquérir des habitudes de communication efficaces. De ce point de vue, le projet n'a fait qu'améliorer et aiguiser notre manière d'organiser notre travail et de gérer un planning.

Deuxièmement, de part la souplesse des technologies à utiliser, la multiplicité des actions à réaliser, et sa complexité, ce projet nous a repoussé dans des exercices de développement encore jamais réalisés, pour la plupart d'entre-nous. Il nous a permis d'entrevoir des technologies, des services, dans lesquels, nous avons maintenant une certaine aisance, que ce soit dans la conception ou la pratique.**Cette veille régulière, durant le projet, nous a aussi poussé à mettre à jour notre panel d'outils et de solutions web** pour le projet. Tout ceci renforce nos débuts dans l'expertise professionnelle.
L'architecture et les tâches demandées, nous ont également permis de toucher également à beaucoup de langages de programmation, vus au sein de notre formation, ce qui nous a permis de pratiquer davantage et de ne mettre aucune compétence de côté.

Enfin, la manière dont s'est rythmé le projet, nous a préparé davantage pour les futurs entretiens et présentations professionnels. Le fait d'avoir séquencer notre projet , en plusieurs rencontres avec l'intervenant, nous a permis de nous exprimer à l'oral, d'argumenter au fur et à mesure nos choix d'orientation du projet. Ces échanges nous ont été bénéfiques pour la préparation à l'oral, mais **elles nous ont appris également à nous exprimer face à un public avec les termes appropriés et à adapter notre vocabulaire selon la cible de notre présentation.**

Ce sont tous ces points qui nous confortent dans l'idée que ce projet était d'une grande richesse, tant dans la conception, le devéloppement ou la manière dont nous l'avons dirigé. Il n'y a pas de doute sur le fait que ce projet restera une expérience plus qu'enrichissante et bénéfique dans notre formation. **Ce projet et son rendu final sont l'union de toutes les compétences obtenues durant notre formation**, afin d'obtenir un résultat concluant sur une vrai besoin scolaire et professionnel.

Annexes du rapport

Annexe 1 - (R1-1/R2-2/R3-1/R3-2)

```
function afficheMap(t)
{
  // Initialisation de la Map + Paramètres (Zoom,...)
  var mapOptions =
        { zoom: 6, center: new google.maps.LatLng(47.0810120000,2.3987820000)};
  var map =
        new google.maps.Map(document.getElementById('map_canvas'),mapOptions);
  // On place la ville
  var ville= [];
  var marker = [];
  var i;
  for(i=0 ;i<t.nbClients()-1;i++)
  {
    ville[i]={
      location:
      new google.maps.LatLng(t.getClient(i+1).positionY,t.getClient(i+1).positionX),
      stopover:true
    }

      marker[i] = new google.maps.Marker({
      position:
      new google.maps.LatLng(t.getClient(i+1).positionY,t.getClient(i+1).positionX),
      map: map,
      title:»étape «+(i+1),
      icon:'./images/marker_base.png'
    });

    google.maps.event.addListener( marker[i], «click», (function(marker,i)
    {
      return function()
      {
        new google.maps.InfoWindow(
        {content: «<h1>etape «+(i+1)+»</h1>»+t.getClient(i+1).afficherClient()}
        ).open(map, marker);
      }
    })(marker[i],i));
  }

 marker[i] = new google.maps.Marker({
      position:new google.maps.LatLng(t.depotY,t.depotX),
      map: map,
      title:»depot»,
      icon:'./images/marker_depot.png'
    });
```

```
    google.maps.event.addListener( marker[i], «click»,(function(marker,i)
    {
      return function()
      {
                new google.maps.InfoWindow(
                {content: «Depot <br\> ville du dêpots : «+t.nom_depot}
                ).open(map, marker);
      }
  })(marker[i],i));

  var listeVille = [];
  listeVille[0]=[];
  var cptwaypoints=0;
  var cptListeVille=-1;

  for(var cpt=0; cpt < ville.length;cpt++)
  {
        if(cptwaypoints == 0)
        {
              cptListeVille++;
              listeVille[cptListeVille]=[];
        }

        listeVille[cptListeVille][cptwaypoints]=ville[cpt];

        if(cptwaypoints == 7)
              cptwaypoints =0;
        else
              cptwaypoints++;
  }
  var service;
  var display;
  var origin;
  var destination;
  var waypoints;

for(var cpt = 0; cpt< listeVille.length;cpt++)
{
        service = new google.maps.DirectionsService();
        display = new google.maps.DirectionsRenderer({'map': map});
        if(cpt == 0)
        {
              origin = new google.maps.LatLng(t.depotY,t.depotX);
        }
        else
        {
              origin = listeVille[cpt-1][listeVille[cpt-1].length-1].location;
        }
```

```
        if(cpt == listeVille.length-1)
        {
            destination = new google.maps.LatLng(t.depotY,t.depotX);
            waypoints=listeVille[cpt];
        }
        else
        {
            waypoints=[];

            for(var i =0; i < listeVille[cpt].length-1;i++)
            {
                waypoints[i]=listeVille[cpt][i];
            }

            destination = listeVille[cpt][listeVille[cpt].length-1].location;
        }

        itineraire(service,display,origin,listeVille[cpt],destination);
    }
    // On place le dépot en premier point de rendez-vous

}
```

Annexe 2 - (R1-2)

```
function clientSocket($host, $port) {
    if(!($this->sock = socket_create(AF_INET, SOCK_STREAM, 0)))
    {
        $errorcode = socket_last_error();
        $errormsg = socket_strerror($errorcode);
        die(«Couldn’t create socket: [$errorcode] $errormsg \n»);
    }

    if(!socket_connect($this->sock , $host , $port))
    {
        $errorcode = socket_last_error();
        $errormsg = socket_strerror($errorcode);
        die(«Could not connect: [$errorcode] $errormsg \n»);
    }
}

 function envoiMessage($xml) {

    $message = «VRP#».$xml;
    //Send the message to the server
    if( ! socket_send ( $this->sock , $message , strlen($message) , 0))
    {
        $errorcode = socket_last_error();
        $errormsg = socket_strerror($errorcode);

        die(«Could not send data: [$errorcode] $errormsg \n»);
    }
    //echo «Message sent successfully \n»;
}

 function receptionMessage()
 {
    $xml = «»;

    socket_recv($this->sock, $xml, 2048, MSG_WAITALL);

    $xml = substr($xml,0,strpos($xml,»</VRPXMLResponse>»)+strlen(«</VRPXMLRes-
ponse>»));

    socket_close($this->sock);
    return $xml;
 }
```

Annexe 3 (R1-3)

```
/* ---------------------------------------- */
/*                Client                    */
/* ---------------------------------------- */

// Prototype pour le stockage des données d'un Client
function Client(id,ordre,positionX,positionY,ville_client,demande,horaires_min,ho-
raires_max)
{
    // Définition des variables du prototype Client
    this.id = id;
    this.ordre = ordre;
    this.positionX = positionX;
    this.positionY = positionY;
    this.ville_client=ville_client;
    this.demande=demande;
    this.horaires_min=horaires_min;
    this.horaires_max=horaires_max;

    // Fonction message qui retourne une chaine texte
    this.afficherClient = function()
    {
        var texte = «»;
        // Affichage de notre récapitulatif Client
        texte += «id = « + this.id + «<br/>» +
            «ordre = « + this.ordre + «<br/>» +
            /*»position X = « + this.positionX + «<br/>» +
            «position Y = « + this.positionY + «<br/>» +*/
            «ville du client = « + this.ville_client + «<br/>» +
            «demande = « + this.demande + «<br/>» +
            «horaires_min = « + this.horaires_min + «<br/>» +
            «horaires_max = « + this.horaires_max + «<br/><br/>»;

        return (texte);
    };
}
```

```
/* ---------------------------------------- */
/*                 Dépôt                    */
/* ---------------------------------------- */

// Prototype pour le stockage des données d'un Dépôt
function Depot(id,positionX,positionY,ville_depot)
{
    // Définition des variables du prototype Dépôt
    this.id = id;
    this.positionX = positionX;
    this.positionY = positionY;
    this.ville_depot=ville_depot;

    // Fonction message qui retourne une chaine texte
    this.afficherDepot = function()
    {
        var texte = «»;
        // Affichage de notre récapitulatif Client
        texte += «   id = « + this.id + «<br/>» +
            /*»   position X = « + this.positionX + «<br/>» +
            «   position Y = « + this.positionY + «<br/>» +*/
            «   ville du dépôt = « + this.ville_depot + «<br/><br/>»;

        return (texte);
    };
}

/* ---------------------------------------- */
/*                 Tournée                  */
/* ---------------------------------------- */

// Prototype pour le stockage des données d'une Tournée
function Tournee(id,cout,duree,charge,depotX,depotY,nom_depot)
{
    // Définition des variables du prototype Tournée
    this.id=id;
    this.cout = cout;
    this.duree = duree;
    this.charge = charge;

    // Liste des clients de notre tournée
    this.listeClients = [];

    // On indique le dépot
    this.depotX = depotX;
    this.depotY = depotY;
    this.nom_depot = nom_depot;
```

```
    this.nbClients=function()
    {
        return this.listeClients.length;
    };

    // Fonction qui montre le trajet
    this.afficherTournee = function()
    {
        var chaine = «Récaptulatif de la Tournée :<br/>» +
            «   Cout : « + this.cout + «<br/>» +
            «   Durée : « + this.duree + «<br/>» +
            «   Charge : « + this.charge + «<br/>» +
            «   Ville Dépot : « + this.nom_depot + «<br/>» /*+
            «   Pos X dépot : « + this.depotX + «<br/>» +
            «   Pos Y dépot : « + this.depotY + «<br/><br/>»;*/

        return (chaine);

    };

    // Fonction message qui retourne une chaine texte
    this.afficherClientsTournee = function()
    {
        var chaine = «Récaptulatif des Clients :<br/><br/>»;
        for(var i=1;i<this.listeClients.length;i++)
        {
            chaine += «<h4>Etape n° « + i + «</h4>»;
            chaine += this.listeClients[i].afficherClient();
        }
        return (chaine);
    };

    //Fonction pour récupérer un client
    this.getClient = function(ordre)
    {
        for(var i=1;i<this.listeClients.length;i++)
        {
            if(this.listeClients[i].ordre == ordre )
            {
                return this.listeClients[i];
            }
        }
    };

    //Fonction pour récupérer un client
    this.ajouterClient = function(client)
    {
        this.listeClients[client.ordre] = client;
    };
}
```

Annexe 4 - (R1-4)

```
<form id=»myForm» class=»formulaire» method=»post» action=»map.php»>
    <h2>Véhicule : </h2>
    <div class=»Vehicle» id=»Vehicle1»>
      <label>Véhicule 1 : </label>
      <label for=»VehicleCapacity1»>Capacité : </label><input id=»capacity1»
name=»VehicleCapacity1» type=»number» value=»» min=»0» required />
      <label for=»VehicleSpeed1»>Vitesse : </label><input id=»speed1» name=»VehicleS-
peed1» type=»number» value=»» min=»0» required/>
      <label for=»VehicleFixCost1»>Coût Fixe : </label><input id=»fixcost1»
name=»VehicleFixCost1» type=»number» value=»» min=»0» required/>
      <label for=»VehicleVarCost1»>Coût Variable : </label><input id=»varcost1»
name=»VehicleVarCost1» type=»number» value=»» min=»0» required/>
    </div>
    <a id=»addVehicle» href=»#»>Ajouter</a> <a href=»#» id=»delVehicle»>Supprimer</a>
    <h2>Dépôt:</h2>
    <div class=»Depot» id=»Depot1»>
      <label>Dépôt 1 : </label>
      <label for=»Adress1»>Ville : </label><input id=»DepotAdress1» name=»Depo-
tAdress1» type=»text» value=»» />
    </div>
    <a id=»addDepot» href=»#»>Ajouter</a> <a href=»#» id=»delDepot»>Supprimer</a>

    <h2>Client:</h2>
    <div class=»Client» id=»Client1»>
      <label>Client 1 : </label>
      <label for=»Adress1»>Ville : </label><input id=»ClientAdress1» name=»Clien-
tAdress1» type=»text» value=»» />

      <label for=»Demand1»>Demande : </label><input id=»ClientDemand1»
name=»ClientDemand1» type=»text» value=»» required />
      <label for=»TimeWindowMin1»>Heure de livraison minimum : </label><input
id=»ClientTimeWindowMin1» name=»ClientTimeWindowMin1» type=»time» value=»»
required />
      <label class=»paddingTWMax» for=»TimeWindowMax1»>Heure de livraison maxi-
mum : </label><input id=»ClientTimeWindowMax1» name=»ClientTimeWindowMax1»
type=»time» value=»» required />
    </div>
```

```
        <a id=»addClient» href=»#»>Ajouter</a> <a href=»#» id=»delClient»>Supprimer</a>
        <h2>Contrainte:</h2>
        <div class=»Contrainte»>
            <label for=»MaxTripDuration1»>Temps maximum de la tournée : </label><input id=»maxTripDuration» name=»MaxTripDuration» type=»number» value=»» min=»0» required />
            <label for=»MaxTripWithoutRestDuration1»>Temps maximum de la tournée sans repos : </label><input id=»maxTripWithoutRestDuration» name=»MaxTripWithoutRestDuration» type=»number» value=»» min=»0» required />
        </div>

        <input type=»submit» class=»boutonEnvoyer» value=»Envoyer» />
        <input type=»reset» class=»boutonEnvoyer» value=»Réinitialiser» />
    </form>
```

Annexe 5 - (R2-1)

```
<VRPXMLResponse>
    <Solution>
        <General>
            <Status>ok</Status>
            <TotalCost>5000</TotalCost>
        </General>
        <Trips>
            <Trip>
                <Cost>1000</Cost>
                <Duration>12</Duration>
                <TotalCharge>10</TotalCharge>
                <Depot>
                    <Id>1</Id>
                </Depot>
                <Client>
                    <Id>1</Id>
                    <Order>1</Order>
                </Client>
            </Trip>
            <Trip>
                <Cost>1000</Cost>
                <Duration>12</Duration>
                <TotalCharge>10</TotalCharge>
                <Depot>
                    <Id>1</Id>
                </Depot>
            </Trip>
        </Trips>
    </Solution>
</VRPXMLResponse>
```

Annexe 6 - (R2-3)

```
$($xml).each(function()
{
    $(this).find("VRPXMLResponse>Solution>Trips").each(function()
    {
      $(this).find("Trip").each(function()
      {
        var i=0;
                var idDepot = $(this).find("Depot").find("Id").text();
                console.log(idDepot);
        tournees[cptTrip] = new Tournee(cptTrip,
          $(this).find("Cost").text(),
          $(this).find("Duration").text(),
          $(this).find("TotalCharge").text(),
          tabDepot[idDepot-1].positionX,
          tabDepot[idDepot-1].positionY,tabDepot[idDepot-1].ville_depot);
        $(this).find("Client").each(function()
        {
          var id = parseInt($(this).find("Id").text())-1;

                    tabClient[id].ordre=$(this).find("Order").text();
          tournees[cptTrip].ajouterClient(tabClient[id]);
          i++;
        });
        cptTrip++;
      });

    });
  });

  recap.innerHTML=tournees[0].afficherTournee()+"<br/>"+tournees[0].affi-
cherClientsTournee();
  for(var i=0;i<tournees.length;i++)
  {
    select.innerHTML+="<option value="+i+"> Tournée n° "+(i+1)+"</option>";
  }

  $("#select").change(function()
  {
    var t =tournees[$(this).val()];
    recap.innerHTML=t.afficherTournee()+"<br/>"+t.afficherClientsTournee();
    afficheMap(tournees[$(this).val()]);
  });
```

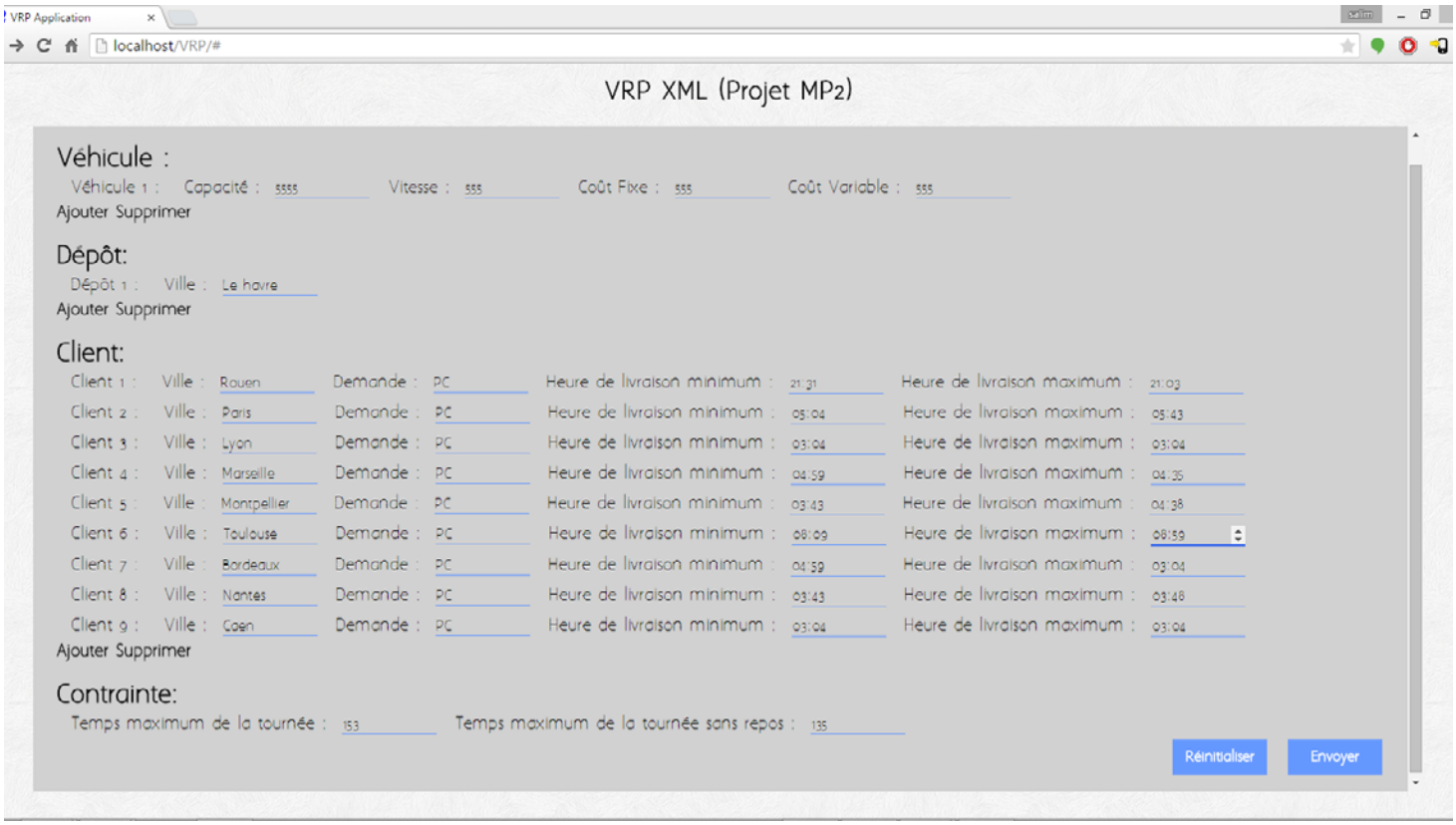
Annexe 7 - (R3-3)

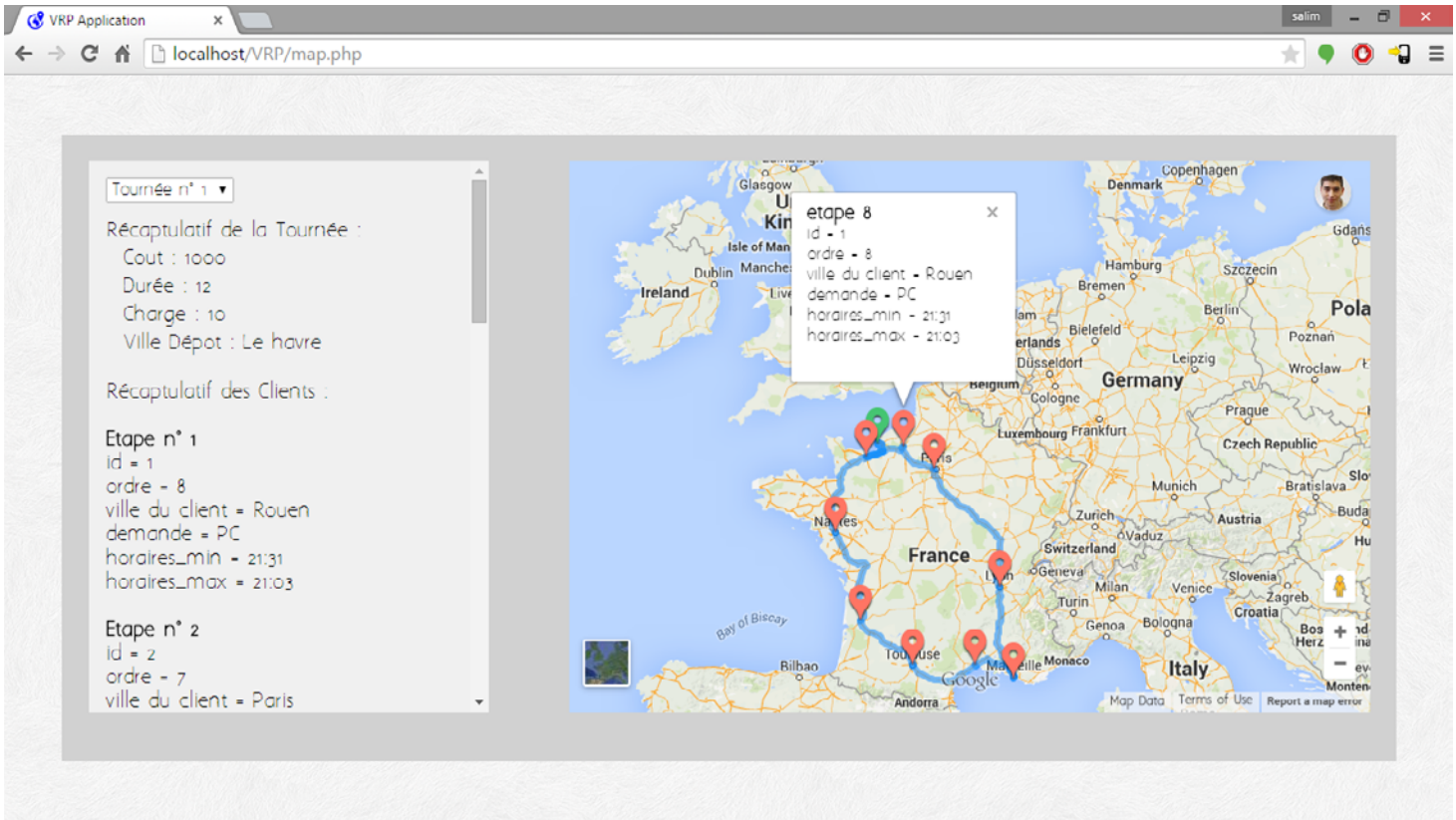
```
function genereXML($infos,$bdd)
{

  $xml = «<VRPXMLQuery>»;
  $xml .= «<ProblemData>»;
  $xml .= «<Vehicles>»;
  $vehiclesloop = false;
  $i = 1;
  while (!$vehiclesloop)
  {
   $xml .= «<Vehicle>»;
   $xml .= «<Capacity>».$infos['VehicleCapacity'.$i].»</Capacity>»;
   $xml .= «<Speed>».$infos['VehicleSpeed'.$i].»</Speed>»;
   $xml .= «<FixCost>».$infos['VehicleFixCost'.$i].»</FixCost>»;
   $xml .= «<VarCost>».$infos['VehicleVarCost'.$i].»</VarCost>»;
   $xml .= «</Vehicle>»;
   $i++;
   if (!(isset($infos['VehicleCapacity'.$i])) || $infos['VehicleCapacity'.$i] == «»)
   {
    $vehiclesloop = true;
   }
  }
  $xml .= «</Vehicles>»;
  $xml .= «<Depots>»;
  $depotsloop = false;
  $i = 1;
  while (!$depotsloop) {
   $xml .= «<Depot>»;
   $xml .= «<ID>».$i.»</ID>»;
   $xml .= «<Address>».$infos['DepotAdress'.$i].»</Address>»;
   $coor=$bdd->getLatLngByAdress($infos['DepotAdress'.$i]);
   $xml .= «<X>».$coor['longitude'].»</X>»;
   $xml .= «<Y>».$coor['latitude'].»</Y>»;
   $xml .= «</Depot>»;
   $i++;
```

```
    if (!(isset($infos['ClientAdress'.$i])) || $infos['ClientAdress'.$i] == «»)
    {
        $clientsloop = true;
     }
   }
   $xml .= «</Clients>»;
   $xml .= «<Constraints>»;
   $xml .= «<MaxTripDuration>».$infos['MaxTripDuration'].»</MaxTripDuration>»;
   $xml .= «<MaxTripWithoutRestDuration>».$infos['MaxTripWithoutRestDuration'].»</
MaxTripWithoutRestDuration>»;
   $xml .= «</Constraints>»;
   $xml .= «</ProblemData>»;
   $xml .= «</VRPXMLQuery>»;
   return $xml;
 }
```

Annexe 8 - Interface de la web-application (2 pages)





Annexe 9 - Diagramme de GANTT - Répartition des tâches