



EMACovoit'

Analyse & Conception

Projet GSI – IFIE 2016

Solène Pettier
Paul Girault
Arnaud Ballet
Grégoire Lecomte



TABLE DES MATIERES

PARTIE I : Modélisation « métier » des processus	4
Le formalisme de description des processus : Modélisation BPMN	4
La modélisation des processus	5
Authentifier l'utilisateur d'EMACovoit'	5
Accéder aux propositions de covoiturage.....	6
Consulter des propositions de covoiturage	7
Publier une annonce de covoiturage	8
Modifier une annonce de covoiturage.....	9
Supprimer une annonce de covoiturage.....	10
PARTIE II : Les différents interfaces d'EMACovoit'	11
Description de l'architecture : Modèle-Vue-Contrôleur (MVC)	11
Definition des routes de l'application WEB EMACovoit'	12
Navigation dans l'application.....	13
Description des différentes vues de l'application EMACovoit'	14
Vue login.....	14
Vue barre de navigation.....	14
Accueil EMACovoit'	15
Recherche de trajets.....	16
Annonce de covoiturage.....	17
Publier une annonce de covoiturage	18
Description du Design d'EMACovoit'	20
Esprit du design	20
Couleurs.....	20
PARTIE III : Les spécifications d'implémentation.....	21
Modèle conceptuel des données (MCD)	21
Règles de gestion de l'outil	23
PARTIE IV : L'estimation de la taille et de la charge pour l'étape de développement	25
Méthode d'estimation de la taille et de la charge du projet.....	25
Estimation charge totale pour une difficulté maximale.....	27
Estimation charge totale pour une difficulté minimale	28
Estimation charge totale pour une difficulté moyenne	29
Comparaison des solutions.....	30
PARTIE V : Glossaire	31
PARTIE VI : Table des illustrations.....	32

PARTIE VII : Annexes	33
Modèle relationnel de données (MRD)	33
Feuille de calcul de l'estimation de la taille du projet et de la charge associée à chaque fonction ...	34

HISTORIQUE DES REVISIONS DU DOCUMENT – ANALYSE & CONCEPTION			
VERSION	DATE	NOM	DESCRIPTION
1.1	04/12/2015	Grégoire Lecomte	Collecte des premières informations et mise au propre du document
1.2	10/12/2015	Grégoire Lecomte	Prise en compte des indications suite au premier jalon avec M.Otton et M.Gaborit
1.3	28/12/2015	Grégoire Lecomte	Rédaction après relecture par les membres de l'équipe GSIGN
1.4	03/12/2015	Grégoire Lecomte	Rédaction après relecture par les membres de l'équipe GSIGN post trêve de Noël
1.5	07/12/2015	Grégoire Lecomte	Dernière relecture par les membres de l'équipe GSIGN

PARTIE I : MODELISATION « METIER » DES PROCESSUS

LE FORMALISME DE DESCRIPTION DES PROCESSUS : MODELISATION BPMN

QU'EST-CE QUE LE BPMN

Le BPMN est le langage standard pour modéliser graphiquement un processus métier. Il a été créé par le BPMI (Business Process Management Initiative) et est maintenu par l'OMG. Ce langage est supporté par les principaux éditeurs de solution SOA.

Le BPM est destiné aux utilisateurs des processus: analystes, développeurs, managers...

A QUOI SERT LE BPMN ?

Le BPMN fourni un ensemble d'objets graphiques qui permet de modéliser tout processus d'entreprise. Les concepts de cette modélisation sont assez similaires au modèle dynamique que propose EPC (l'événement, la synchronisation,...)

Pour l'application EMACovoit', nous avons choisi de modéliser nos processus métier avec ce formalisme.

LA MODELISATION DES PROCESSUS

AUTHENTIFIER L'UTILISATEUR D'EMACOVOIT'

Une fois sur la page d'accueil, l'utilisateur doit s'authentifier. Le système d'information (SI) affiche le formulaire d'authentification et vérifie les informations de l'utilisateur. Si elles sont correctes l'utilisateur est connecté ; sinon un nouveau formulaire d'authentification est affiché.

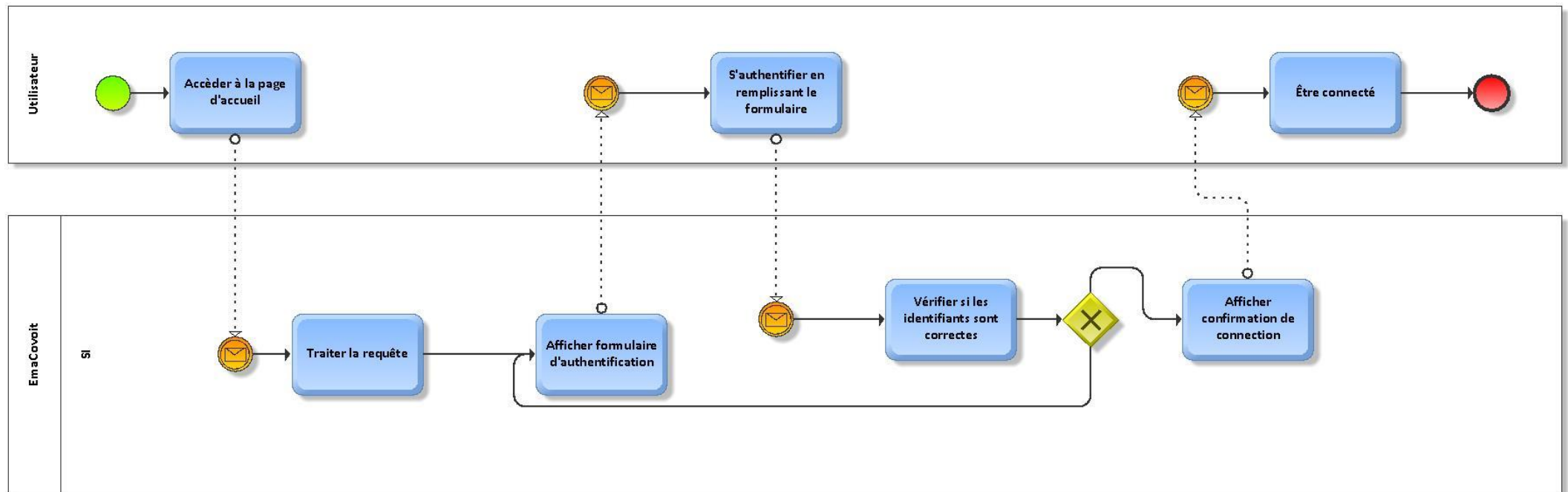


FIGURE 1 : MODELISATION BPMN « AUTHENTIFIER L'UTILISATEUR »

ACCEDER AUX PROPOSITIONS DE COVOITURAGE

L'utilisateur peut effectuer une recherche en réalisant une demande de recherche. Le SI affiche un formulaire de recherche. L'utilisateur remplit ce formulaire. Le SI traite alors la requête et affiche les propositions de covoiturage répondant aux critères définis précédemment par l'utilisateur.

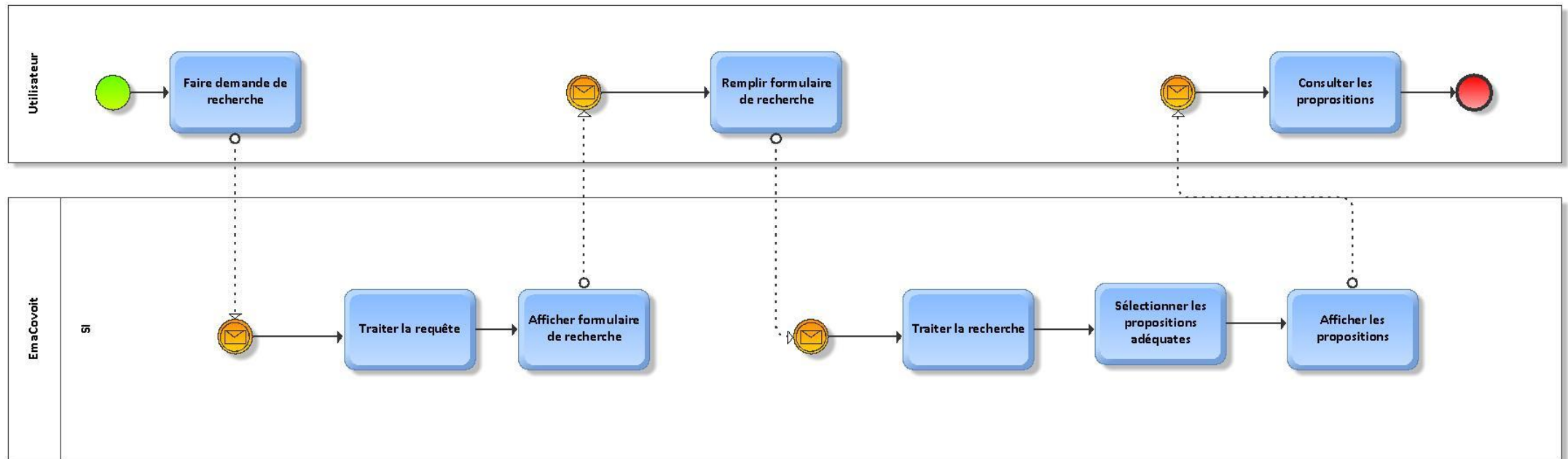


FIGURE 2 : MODELISATION BPMN « ACCEDER AUX PROPOSITIONS DE COVOITURAGE »

CONSULTER DES PROPOSITIONS DE COVOITURAGE

L'utilisateur possède la possibilité de consulter ses propres propositions de covoiturage déjà publiées.

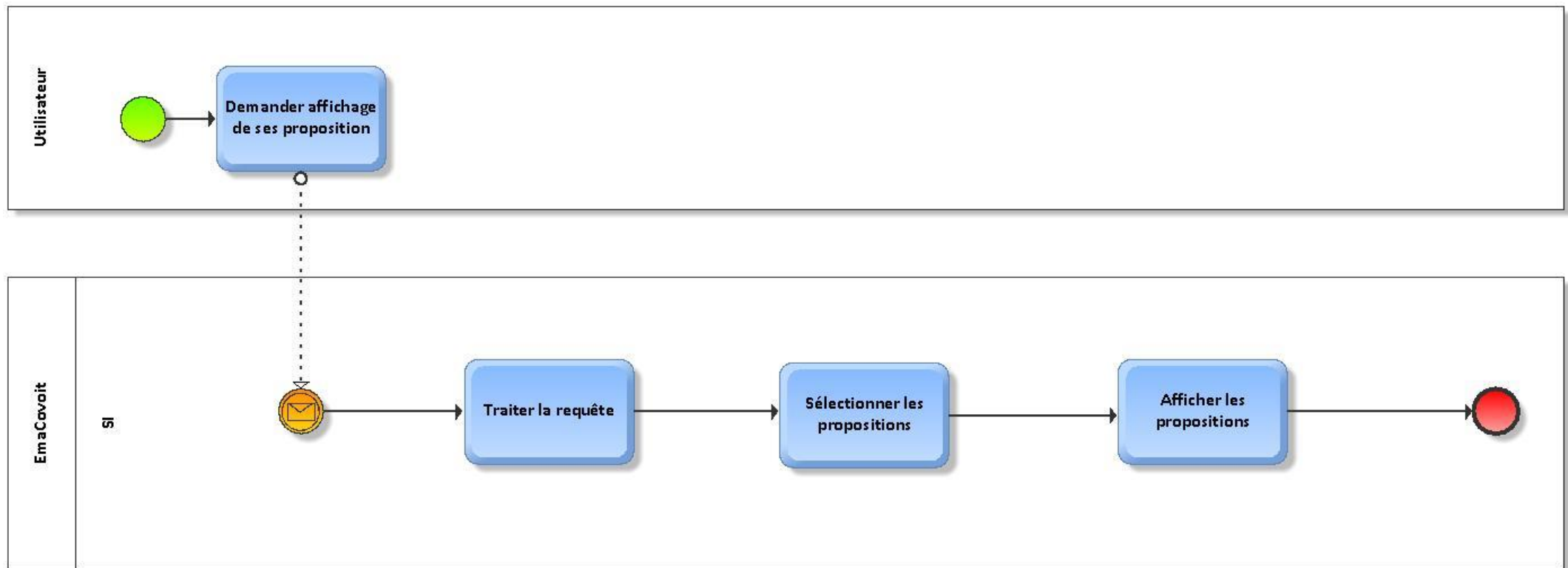


FIGURE 3 : MODELISATION BPMN « CONSULTER DES PROPOSITIONS DE COVOITURAGE »

PUBLIER UNE ANNONCE DE COVOITURAGE

L'utilisateur a la possibilité de publier une proposition de covoiturage. Pour cela il accède au formulaire de publication. Le SI affiche le formulaire, l'utilisateur le remplit et le valide. Si le formulaire est vérifié, le SI stocke la nouvelle publication et la publie.

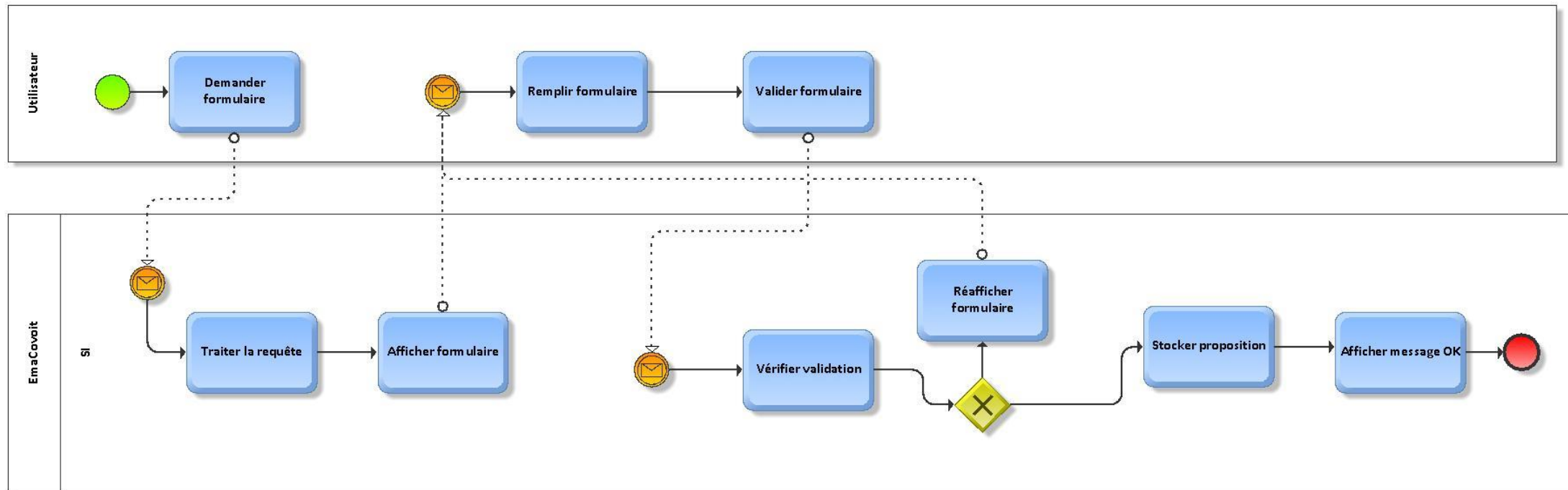


FIGURE 4 : MODELISATION BPMN « PUBLIER UNE ANNONCE DE COVOITURAGE »

MODIFIER UNE ANNONCE DE COVOITURAGE

L'utilisateur a la possibilité de modifier une publication de covoiturage qu'il a déjà publié. Pour cela il choisit la publication à modifier. Le SI affiche le formulaire de modification, l'utilisateur le remplit et le valide. Si le formulaire est vérifié, le SI stocke la nouvelle publication modifiée.

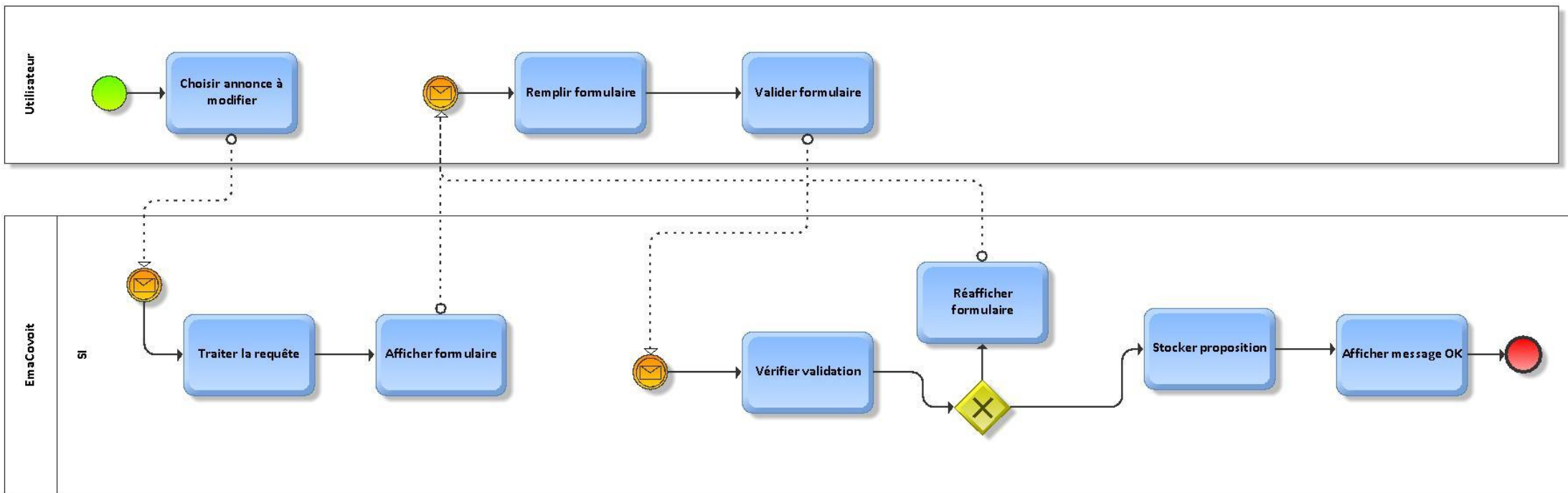


FIGURE 5 : MODELISATION BPMN « MODIFIER UNE ANNONCE DE COVOITURAGE »

SUPPRIMER UNE ANNONCE DE COVOITURAGE

L'utilisateur choisit l'annonce à supprimer, le SI affiche l'annonce à supprimer. L'utilisateur valide l'action puis le SI traite la requête et efface l'annonce de la base de données et affiche les annonces restantes.

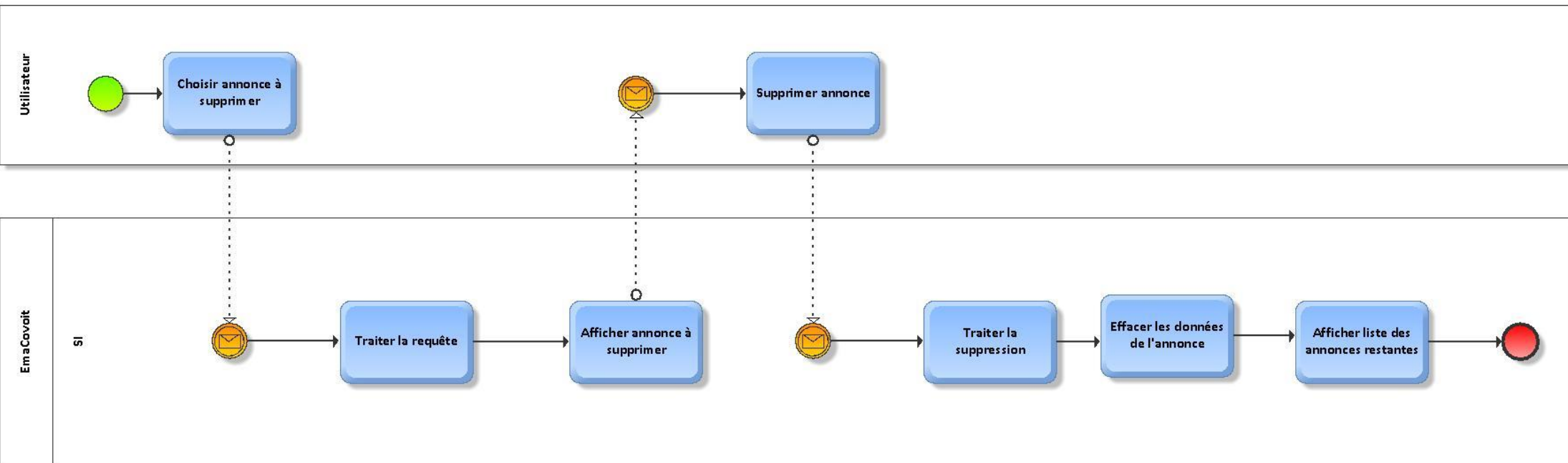
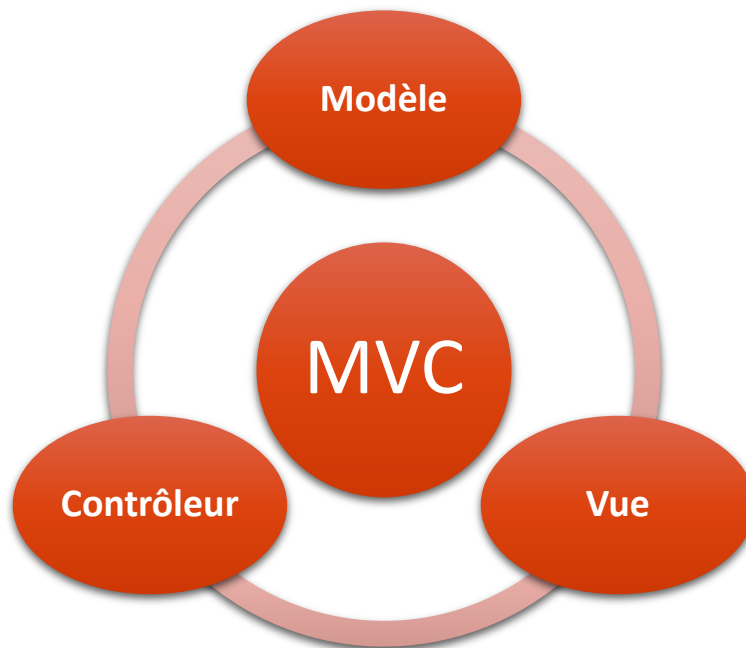


FIGURE 6 : MODELISATION BPMN « SUPPRIMER UNE ANNONCE DE COVOITURAGE »

PARTIE II : LES DIFFERENTS INTERFACES D'EMACOVOIT'

DESCRIPTION DE L'ARCHITECTURE : MODELE-VUE-CONTROLEUR (MVC)

MVC est l'acronyme de Modèle-Vue-Contrôleur :



LE MODELE

Le modèle représente le cœur de l'application : traitements des données, interactions avec la base de données, etc. Il décrit les données manipulées par l'application. Il regroupe la gestion de ces données et est responsable de leur intégrité. La base de données sera l'un de ses composants. Le modèle comporte des méthodes standards pour mettre à jour ces données (insertion, suppression, changement de valeur)

LA VUE

L'entité avec laquelle l'utilisateur interagit se nomme précisément la vue. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toute action de l'utilisateur. Ces différents événements sont envoyés au contrôleur. La vue n'effectue pas de traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

LE CONTROLEUR

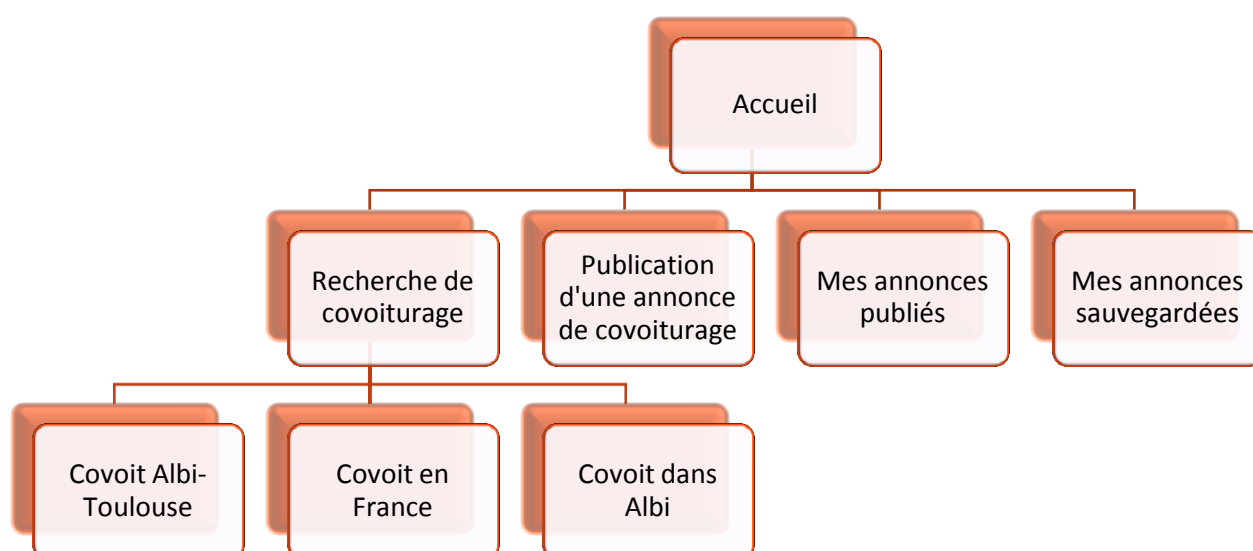
Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de la vue et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle afin que les données affichées se mettent à jour

DEFINITION DES ROUTES DE L'APPLICATION WEB EMACOVOIT'

Ce tableau récapitule les différentes routes de l'application, et présente pour chacune d'entre-elle la ou les fonction(s) réalisées par le contrôleur. Pour accéder à l'application, l'utilisateur devra être authentifié grâce au système d'annuaire LDAP de l'école. Cette restriction d'accès se paramètre dans le fichier de configuration du serveur.

Route	Fonctions effectuées
GET /accueil	Si l'utilisateur est authentifié, on affiche la page d'accueil, sinon on redirige l'utilisateur vers la route /login.
GET /annonce/rechercher/ :type/ :départ/ :arrivée/:date	Affichage du formulaire de la vue rechercher.tt Le paramètre type peut prendre 3 valeurs possibles : Albi/Albi-Toulouse/France. Les paramètres :départ, :arrivée, :date sont facultatifs et dépendront du type de trajet recherché.
GET /annonce/ :id	Affichage de la vue annonce.tt et transmission de l'annonce d'identifiant :id.
GET /annonce/publier/	Affichage de la vue publierAnnonce.tt
POST /annonce/publier/	Création d'une nouvelle annonce de covoiturage : Contrôle des valeurs transmises. Si elles sont conformes par rapport aux règles de gestion, on insère une ligne dans la table covoiturage. Sinon afficher message d'erreur et réafficher la vue publierAnnonce.tt.
GET /annonce/modifier/ :id	Afficher la vue publierAnnonce.tt, avec la transmission l'annonce correspondant à l'identifiant :id.
POST /annonce/modifier /	Modification d'une annonce de covoiturage : Contrôle des valeurs transmises. Si elles sont conformes par rapport aux règles de gestion, on insère une ligne dans la table covoiturage. Sinon afficher message d'erreur et réafficher la vue publierAnnonce.tt.
GET /annonce/supprimer/ :id	Afficher la vue confirmer.tt
POST /annonce/sauvegarder/	Sauvegarde une entrée dans la table « Sauvegardes » liant l'utilisateur à l'annonce et redirige vers l'annonce.
POST /annonce/supprimer/	Suppression d'une annonce de covoiturage. Redirection vers l'accueil avec l'affichage d'un message pour confirmer la bonne suppression de l'annonce.
GET /annonces/publiees/	Affichage de la vue mesAnnonces.tt et transmission de liste des annonces postées par l'utilisateur.
GET /annonces/sauvegardees/	Affichage de la vue mesAnnonces.tt et transmission de la liste des annonces sauvegardées par l'utilisateur.

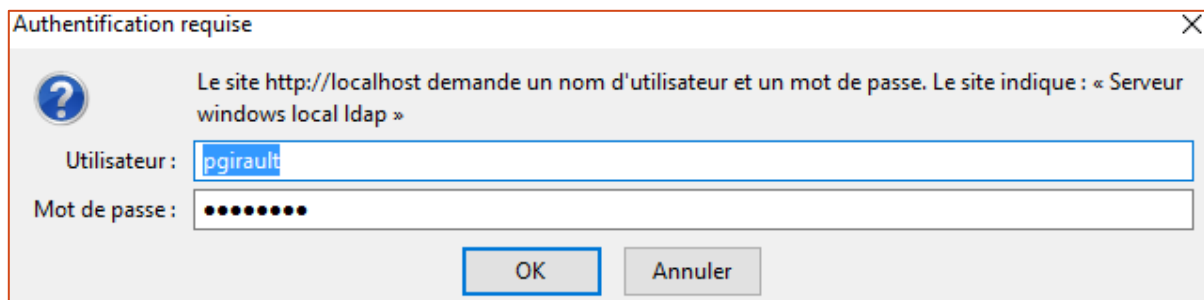
NAVIGATION DANS L'APPLICATION



DESCRIPTION DES DIFFERENTES VUES DE L'APPLICATION EMACOVOIT'

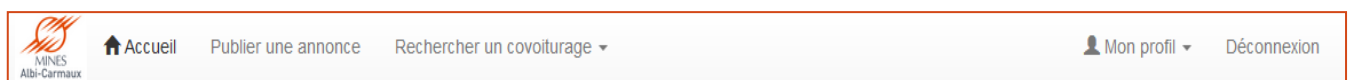
Afin de bien illustrer comment se présentera l'application et son fonctionnement, nous avons réalisé une ébauche des principales vues qu'utilisera notre application.

VUE LOGIN



L'interface de connexion est implémentée automatiquement par le serveur Apache, une fois celui-ci configuré. La connexion se fera donc avec le login et le mot de passe école.

VUE BARRE DE NAVIGATION



Sur chaque page de l'application sera présent une barre de navigation comportant :

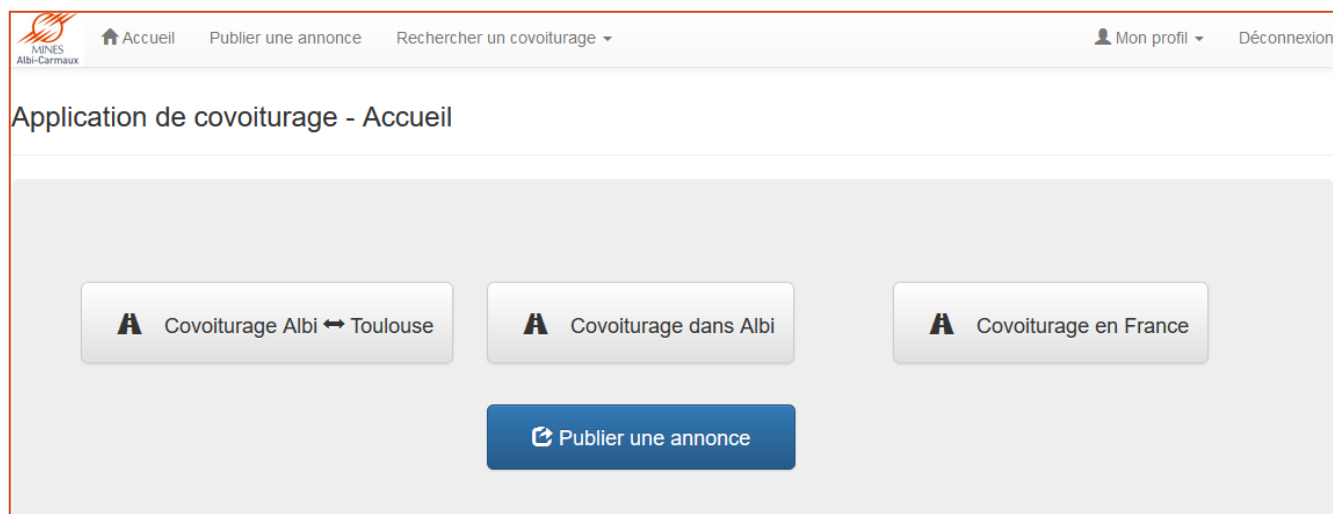
- Un bouton de déconnexion
- Une liste déroulante « Mon profil »
- Un lien vers l'accueil.

La liste déroulante « Mon profil » devra comporter les éléments suivant :

- Mes annonces sauvegardées
- Mes annonces
- Mon profil

ACCUEIL EMACOVOIT'

VUE

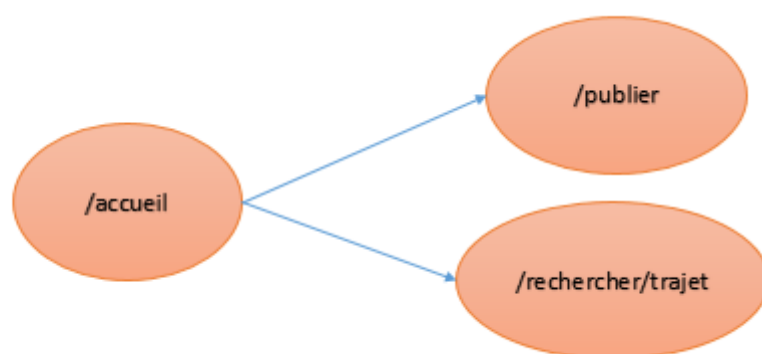


La page d'accueil sera présentera ainsi.

Elle permet d'accéder à la route /annonce/publier via le bouton « Publier une annonce ». Elle permet également d'accéder à la route /annonce/rechercher/ :type via les boutons : « Covoiturage Albi<->Toulouse », « Covoiturage dans Albi », « Covoiturage en France ».

Selon le bouton choisi, le paramètre type prendra une valeur différente.

GRAPHE



RECHERCHE DE TRAJETS

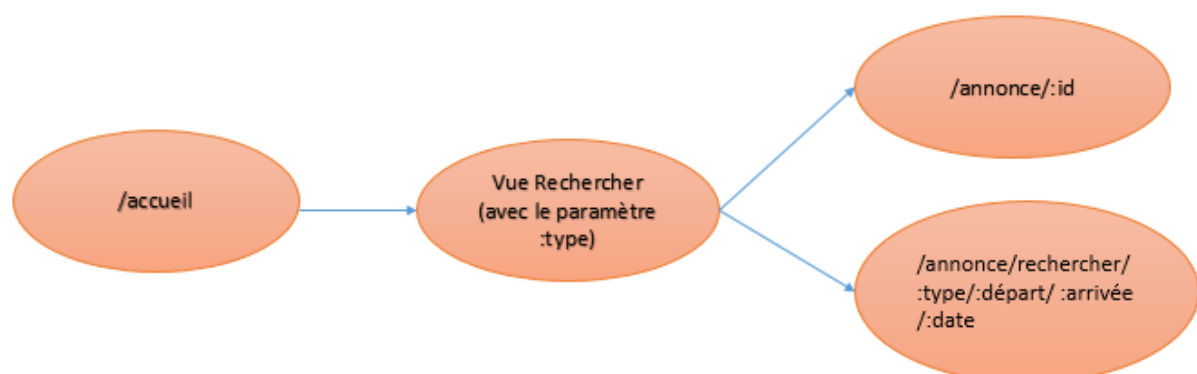
VUE

The screenshot shows the 'Application de covoiturage' interface. At the top, there is a navigation bar with links: Accueil, Publier une annonce, Rechercher un covoiturage, Mon profil, and Déconnexion. Below the navigation bar, the main heading is 'Application de covoiturage'. The central part of the interface is a search form titled 'Rechercher un covoiturage'. It includes a dropdown menu for 'Sélectionnez un type de covoiturage', two input fields for 'De' and 'À' (each with a location icon), a date input field labeled 'Date', and a blue 'Rechercher' button. Below the search form, there is a section titled 'Le JJ/MM/AAAA' showing a specific trip: 'Albi - Toulouse - Bordeaux' for '30 €' and '3 places', with the driver 'Paul Girault-Matz - IFIE 2016'.

La vue de recherche d'un trajet se présentera de cette manière. Pour accéder à cette vue plusieurs paramètres peuvent être précisés. Le type de trajet peut être donné ou non pour arriver sur la page. Via le bouton rechercher l'utilisateur accédera à la même vue, avec cette fois les paramètres lieu de départ, lieu d'arrivée et date qui auront été transmis via le formulaire. La vue comprendra en dessous du formulaire de recherche la liste des covoiturages correspondant aux paramètres fournis.

Chacune de ces annonces pour mener au détail de l'annonce via la route : `/annonce/:id`.

GRAPHE



VUE

La vue d'une annonce de covoiturage se présentera ainsi. La route permettant d'afficher cette vue prend en paramètre l'id de l'annonce. Les boutons « Modifier » et « Supprimer » sont affichés si l'utilisateur courant est l'auteur de la publication ou s'il est administrateur.

Le bouton sauvegarder mène vers la route /annonce/sauvegarder/ qui redirige vers l'annonce en question une fois celle-ci sauvegardée.

```

graph LR
    A("/annonce/rechercher/ :type/:départ/:arrivée/:date") --> C("Vue annonce (:id de l'annonce en paramètre)")
    B("/annonces/publiees/") --> C
    D("/annonces/sauvegardees/") --> C
    C --> E("/annonce/modifier/:id")
    C --> F("/annonce/supprimer/:id")
    C --> G("/annonce/sauvegarder/")
  
```

PUBLIER UNE ANNONCE DE COVOITURAGE

VUE

The screenshot shows a web form titled "Publier une annonce :". It contains several input fields and dropdown menus. The first section includes a dropdown for "Type de covoiturage", followed by text inputs for "Date de départ", "Heure de départ", "Prix", "Départ", "Destination", and a larger text area for "Description" with a placeholder "Entrez une description". A second "Type de covoiturage" dropdown follows. Below these are dropdowns for "Nombre de places" and "Taille des bagages". At the bottom are two buttons: "Annuler" and "Publier".

Label	Input Type	Placeholder/Options
Type de covoiturage	Dropdown	Sélectionnez un type de covoiturage
Date de départ	Text	Date de départ
Heure de départ	Text	Heure de départ
Prix	Text	Prix
Départ	Text	Départ
Destination	Text	Destination
Description	Text Area	Entrez une description
Type de covoiturage	Dropdown	Sélectionnez un type de covoiturage
Nombre de places	Dropdown	Sélectionnez un nombre de places
Taille des bagages	Dropdown	Sélectionnez la taille des bagages

Buttons: Annuler, Publier

La vue de publication d'une annonce se présentera ainsi. Le formulaire évoluera selon le type de covoiturage à publier :

Pour un covoiturage Albi<->Toulouse, le lieu de départ sera Albi et la destination sera Toulouse. Ces lieux pourront être inter changés.

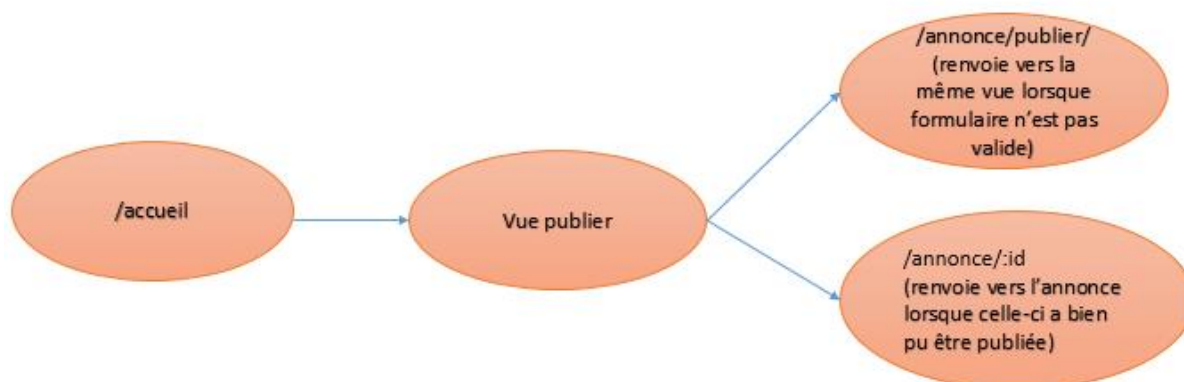
Les champs prix, taille des bagages, description seront facultatifs. Des lieux seront proposés lors de la complétion du champ départ ou destination soit à l'aide du service OpenStreetMap ou bien un script ajax proposant des lieux contenus en base de données.

Cette vue sera réutilisée pour la modification d'une annonce. Elle sera également renvoyée avec des messages d'erreurs lorsque l'utilisateur n'a pas bien rempli le formulaire

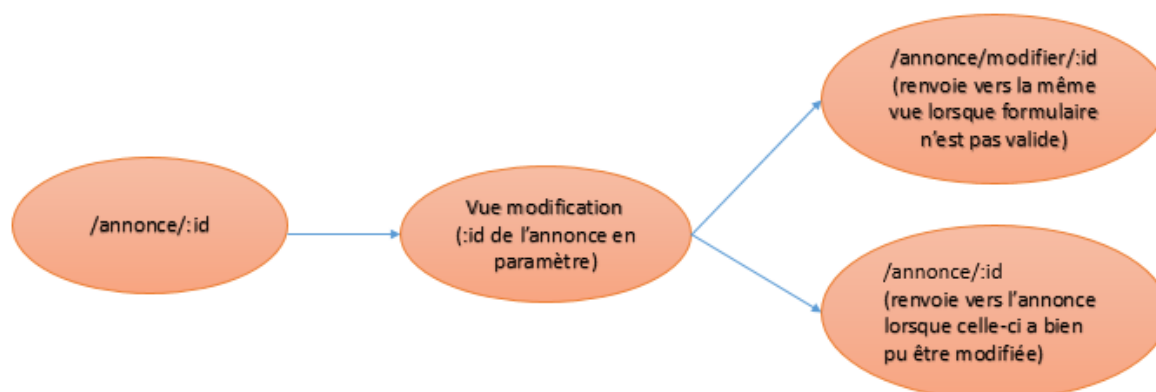
Concernant la suppression d'une annonce, la vue consistera en un simple formulaire de confirmation de suppression.

GRAPHES

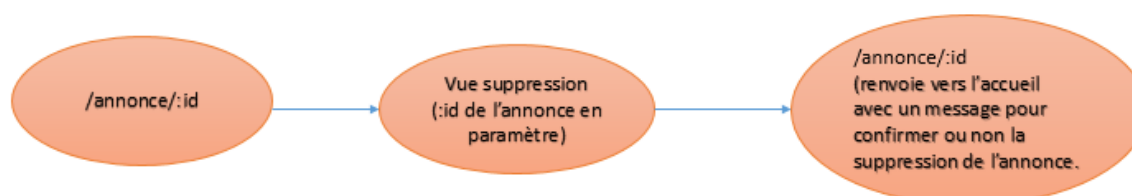
GRAPHE DE PUBLICATION



GRAPHE DE MODIFICATION



GRAPHE DE SUPPRESSION



DESCRIPTION DU DESIGN D'EMACOVOIT'

L'utilisation de fichiers CCS permettra à l'application WEB EMACovoit' d'avoir un design qui lui est propre et en accord avec les valeurs de l'association ID2.

Les feuilles de style en cascade appelées fichiers CSS (Cascading Style Sheets), forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.

ESPRIT DU DESIGN

L'application EMACovoit devra posséder le même esprit épuré que la plateforme CAMPUS de l'école (Bootstrap) afin que les utilisateurs (membres de l'EMAC) évoluent dans une plateforme déjà familière et cohérente avec les autres applications déjà existantes et disponibles sur l'Ecole des Mines d'Albi Carmaux.

COULEURS

Si l'application devait utiliser des couleurs supplémentaires à celles déjà utilisées par le style de CAMPUS. Nous choisirions le verts et ses nuances décrites ci-contre afin garder l'esprit écologique et de développement durable qui caractérise l'association I2D. Pour plus de contraste nous associerons à ce vert sa couleur complémentaire (Orange), elle aussi décrite ci-contre.

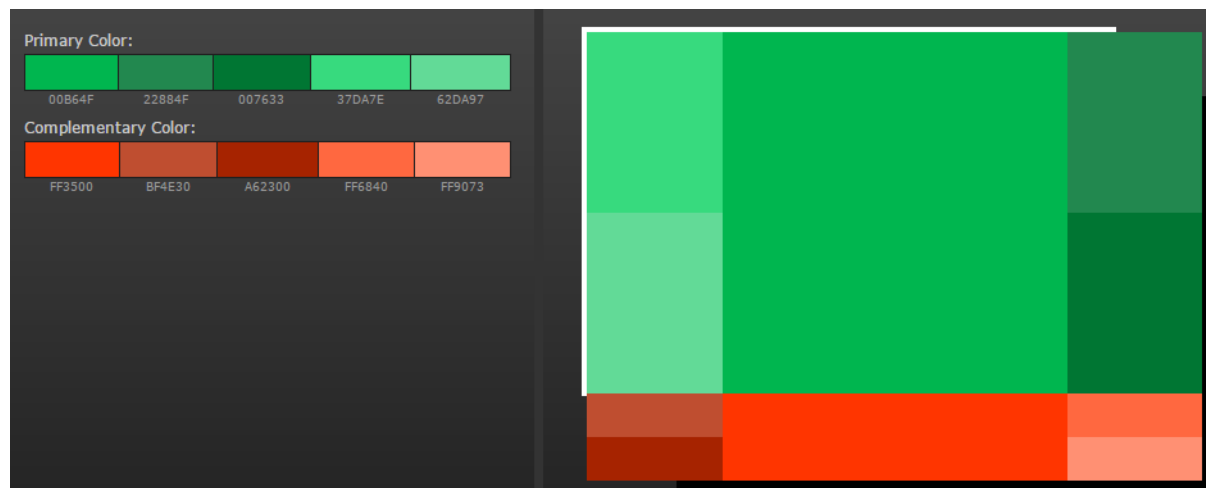


FIGURE 7 : DESCRIPTION DES COULEURS POTENTIELLEMENT UTILISABLE DANS L'APPLICATION EMACOVOIT'

PARTIE III : LES SPECIFICATIONS D'IMPLEMENTATION

MODELE CONCEPTUEL DES DONNEES (MCD)

Le modèle conceptuel des données (MCD) a pour but de représenter de façon structurée les données qui seront utilisées par le système d'information EMACovoit'. Le modèle décrit la sémantique c'est à dire le sens attaché à ces données et à leurs rapports.

L'équipe projet a recensé l'ensemble des données nécessaires au fonctionnement de l'outil. Puis, elle a étudié les relations existantes entre ces données (les dépendances fonctionnelles), pour aboutir au MCD.

Ce modèle inclue les données relatives à l'utilisateur ainsi qu'au covoiturage.

Nous avons choisi d'utiliser le terme « covoiturage » afin de parler indifféremment (pour un utilisateur donné) d'une proposition de covoiturage posté par lui-même ou bien d'une annonce de covoiturage postée par les autres utilisateurs.

La distinction entre les étudiants et les cadres de l'école a été faite à titre indicatif afin de souligner le caractère générique de l'application. Le type de connexion sera le même quel que soit l'utilisateur.

Enfin, nous avons choisi de faire apparaître une classe « Liste_type » dans laquelle seront recensés les 3 types de covoiturages : Long, Court, Albi-Toulouse.

Le reste du MCD est en accord avec le fonctionnement décrit dans le cahier des charges.

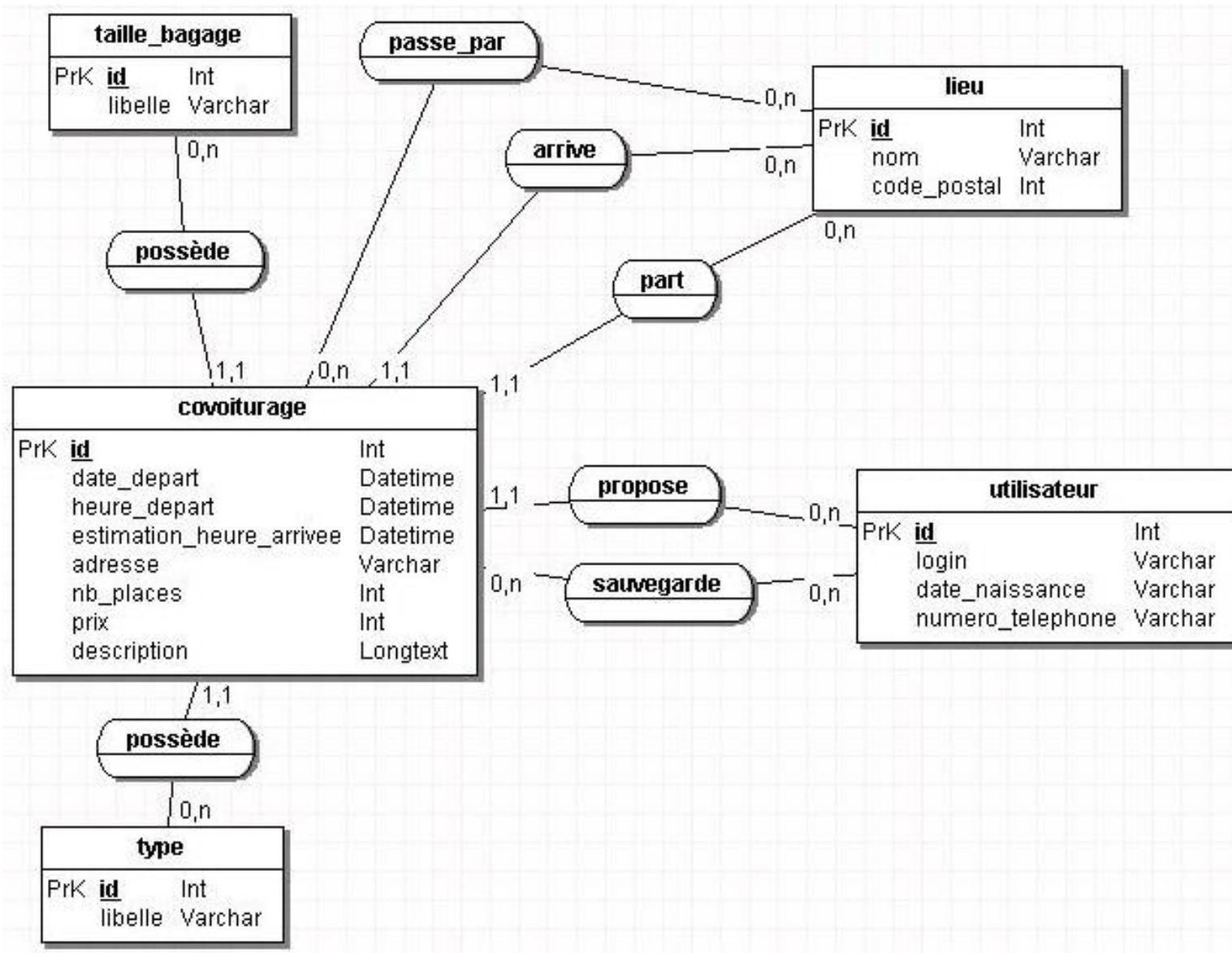


FIGURE 8 : MODELE CONCEPTUEL DES DONNEE DE L'APPLICATION EMACOVOIT'

REGLES DE GESTION DE L'OUTIL

Les règles de gestion de l'application EMACovoit' permettent de caractériser toutes les données échangées dans l'application. Ce tableau rassemble l'ensemble des données métier que le système d'information manipulera avec leurs règles de gestion.

Classe	Libellé de la donnée	Type de donnée	Règle(s) de gestion
Utilisateur	login	Int	Obligatoire Issue du ldap de l'EMAC
	nom	Varchar	Complétion automatique avec la connexion de l'utilisateur Non-Modifiable Récupération via ldap
	prenom	Varchar	Complétion automatique avec la connexion de l'utilisateur Non-Modifiable Récupération via ldap
	date_naissance	Varchar	Facultatif Complétion manuelle
	numero_telephone	Varchar	Facultatif Complétion manuelle
	email	Varchar	Complétion automatique avec la connexion de l'utilisateur, possibilité de changer l'adresse Recupération via ldap
Covoiturage	Id	Int	Obligatoire Auto incrémenté
	Id_type	Int	Obligatoire Sélection préalable
	Date_depart	Datetime	Obligatoire Complétion manuelle Postérieure à la date à laquelle l'annonce est créée ou modifiée
	Date_publication	Datetime	Obligatoire Complétion automatique Non-Modifiable Correspond à la date à laquelle l'annonce est publiée ou modifiée
	Nb_place	Int	Obligatoire Complétion manuelle

	Supérieur ou égal à 1		
	Taille_bagages	Varchar	Obligatoire Complétion manuelle par choix parmi une liste de propositions
	Prix	Int	Facultatif Complétion manuelle
	Description	Longtext	Facultatif Complétion manuelle
Lieu	Id	Int	Obligatoire
	Code_postal	Int	Obligatoire Complétion manuelle Proposition d'auto-complétion
	Nom	Varchar	Obligatoire Complétion manuelle
	Ville	Varchar	Obligatoire Complétion manuelle Proposition d'auto-complétion
	Intitule	Varchar	Obligatoire Complétion manuelle Proposition d'auto-complétion
	Region	Varchar	Facultatif Complétion manuelle Proposition d'auto-complétion Ou choix parmi une liste de propositions
Liste_type	Id	Int	Obligatoire
	Libelle	Varchar	Obligatoire Choix par sélection de l'utilisateur

COMPLEMENT AUX REGLES DE GESTION DE L'OUTIL EMACOVOIT'

Les points suivants correspondent aux différentes règles de gestion supplémentaires, non-décrites dans le tableau ci-dessus :

- La gestion de certaines données est effectuée par l'utilisateur
- La gestion des annonces et de leur obsolescence est automatiquement gérée par le système qui met à jour la base de données chaque semaine via un script de gestion automatique.
- Le gestionnaire de l'application possède aussi la possibilité d'interagir avec les données de chaque utilisateur mais aussi de nettoyer la base de données entière.

PARTIE IV : L'ESTIMATION DE LA TAILLE ET DE LA CHARGE POUR L'ETAPE DE DEVELOPPEMENT

METHODE D'ESTIMATION DE LA TAILLE ET DE LA CHARGE DU PROJET

Les étapes de réalisation du projet sont regroupées de la façon suivante :

- Etablissement du support de travail : planning et suivi du projet
- Création de la base de données en MCD puis SQL
- Création de la fonction Login
- Création de la fonction Publier une annonce
- Création de la fonction Afficher mes annonces
- Création de la fonction Afficher une annonce (détail)
- Création de la fonction Modifier une annonce
- Création de la fonction Supprimer une annonce
- Création de la fonction Rechercher
- Création de la fonction Sauvegarder une annonce
- Création des fichiers CSS
- Intégration déploiements et tests du SI
- Rédaction de documents annexes pour présenter et accompagner le projet

Chaque étape comprend des tâches de taille et de difficulté variable, à réaliser avec des outils différents.

Pour avoir une idée de la charge de travail pour chaque fonction, nous avons travaillé avec des estimations relatives. En effet, il était trop incertain de donner des approximations de la charge pour chaque tâche. Nous avons donc décidé de donner des coefficients de temps pour chaque fonction selon un référentiel. Ces coefficients nous permettent ensuite d'évaluer les charges de façon cohérente pour chaque tâche.

FONCTION AFFICHER MES ANNONCES PUBLIEES

Fonction simple à réaliser qui implique de retrouver en base de données les annonces publiées par l'utilisateur courant. Cette tâche ne présente pas de réelles difficultés, et constitue donc notre tâche de référence. Nous lui affectons un coefficient de 1.

FONCTIONS SUPPRIMER ET AFFICHER UNE ANNONCE

Il s'agit de fonctions plutôt simples mais il convient tout de même de vérifier si l'utilisateur est administrateur ou s'il est l'hôte de l'annonce. Il aura accès à la modification et suppression de l'annonce. Ces 2 fonctions constituent également nos tâches de référence, nous leur affectons à chacune un coefficient de 2.

FONCTION LOGIN

Cette fonction serait compliquée à gérer si nous gérons tous les aspects de la connexion. Or nous utilisons l'annuaire LDAP de l'école. Cette tâche implique de se familiariser avec le fonctionnement d'un annuaire LDAP. Nous devons tout de même sauvegarder en base de données les informations concernant l'utilisateur qui ne sont pas déjà contenus dans l'annuaire ldap. Nous affectons à cette fonction un coefficient de 3.

FONCTION PUBLIER UNE ANNONCE

Il s'agit d'une fonction majeure de l'application. Elle nécessite de vérifier les contraintes de gestion concernant les données du formulaire de publication. Il faut également gérer les champs facultatifs et obligatoires au niveau du formulaire.

Nous envisageons plusieurs possibilités afin de dynamiser ou non certains champs du formulaire :

- **Difficile** (coefficient 10) : Proposer des lieux de départ et de destination lors de la complétion de ces champs par l'utilisateur à l'aide du service OpenStreetMap. L'utilisation de ce service implique de comprendre comment celui-ci fonctionne. L'heure d'arrivée pourra également être calculée par ce service.
- **Moyenne** (coefficient 7.5) : Nous pouvons aussi stocker en base de données chaque nouveau lieu proposé par l'utilisateur lors de la publication d'une annonce de covoiturage. Nous pouvons ainsi proposer les lieux déjà contenus en base de données à l'aide de JavaScript. L'heure d'arrivée sera à rentrer manuellement par l'utilisateur.
- **Facile** (coefficient 5) : Même solution qu'au-dessus sans JavaScript.

FONCTION MODIFIER UNE ANNONCE

Cette fonction reprend une grande partie de la logique de la fonction publier. Une fois celle-ci réalisée, elle sera simple à réaliser. (Coefficient 2).

FONCTION RECHERCHER :

Cette fonction est également difficile à réaliser. Concernant le formulaire de recherche, il faut gérer les différents paramètres, donc certains sont facultatifs. Ensuite, nous devons réaliser l'interrogation de la base de données et la récupération des informations pertinentes. Pour le champ date nous envisageons la solution d'utiliser un plugin jQueryUI proposant un calendrier pour choisir la date.

Nous envisageons plusieurs possibilités afin de dynamiser ou non certains champs du formulaire :

- **Difficile** (Coefficient 10) : Utilisation du service OpenStreetMap pour proposer les lieux de départ et d'arrivée.
- **Moyenne** (Coefficient 7.5) : Utilisation de JavaScript proposant les lieux déjà enregistrés en base de données.
- **Facile** (Coefficient 5) : Formulaire de recherche sans JavaScript.

FONCTIONS SAUVEGARDER UNE ANNONCE

Il s'agit d'une fonction de difficulté moyenne. Il faut gérer l'insertion d'une entrée cohérente dans la table « Sauvegarde ». Afin que les contraintes de clés étrangères soient vérifiées.

(Coefficient 3)

Les coefficients correspondent chacun à un nombre d'heure particulier de travail sur chaque fonction. On admet l'équivalence suivante, qui est constitué une marge d'erreur de 90% :

Coefficient	Charge approximative
1	Entre 1 et 20 heures
2	Entre 20 et 40 heures
3	Entre 40 et 60 heures
5	Entre 60 et 100 heures
7,5	Entre 100 et 150 heures
10	Entre 150 et 200 heures

ESTIMATION CHARGE TOTALE POUR UNE DIFFICULTE MAXIMALE

En premier lieu, on décide de visualiser la charge totale de travail dans le cas de difficulté maximum, c'est-à-dire lorsque les fonctions Publier et Rechercher sont codées sous JavaScript et donc présentent un coefficient de 10.

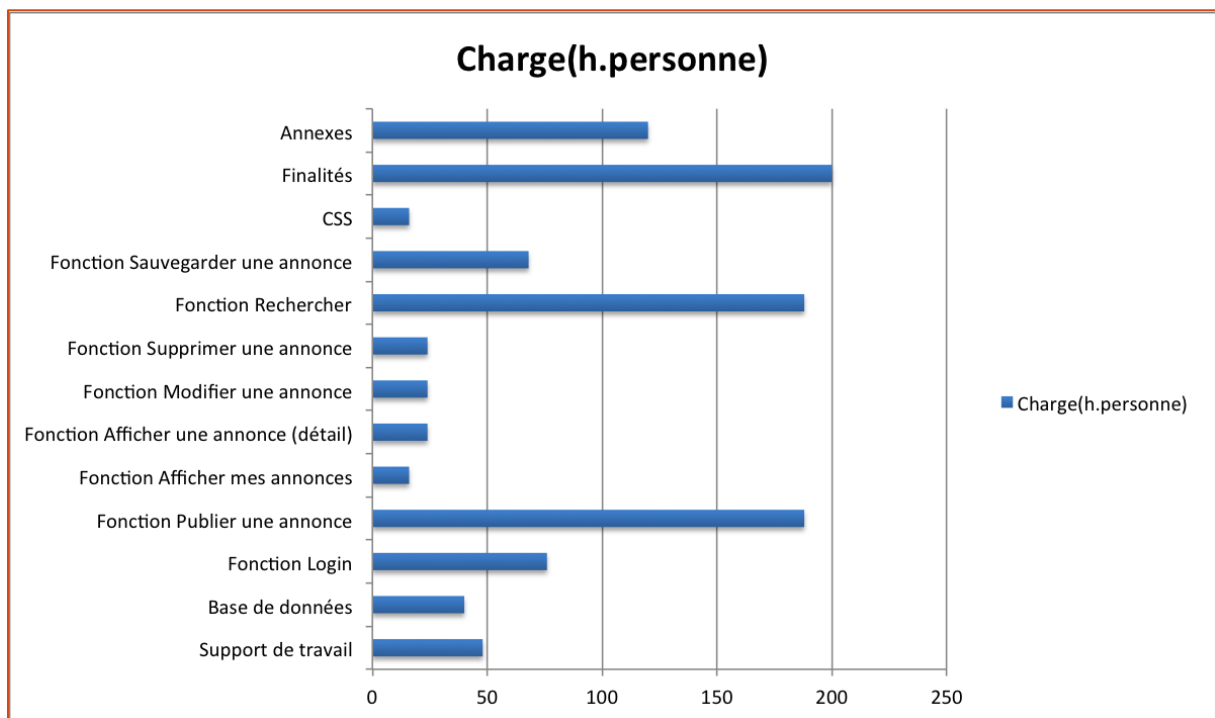


FIGURE 9 : CHARGE EN H.PERSONNE POUR UNE DIFFICULTE MAXIMALE

Selon l'estimation, le nombre total d'heures nécessaires pour arriver à terme de ce projet est de **1032 heures**. Cependant, notre emploi du temps nous accorde seulement 488 heures de travail en autonomie et, en admettant une charge de travail hors emploi du temps de 10 heures par personne par semaines, nous pouvons ajouter 440 heures supplémentaires. Nous avons donc la possibilité de travailler 928 heures sur ce projet qui en demande 1032. Il va donc falloir prioriser certaines phases de développement afin d'être sûr de ne pas s'attarder sur des étapes facultatives longues et difficiles : notre temps doit être optimisé.

ESTIMATION CHARGE TOTALE POUR UNE DIFFICULTE MINIMALE

Afin d'évincer certaines tâches et choisir les plus pertinentes pour la réussite de notre projet et la satisfaction de la demande client, nous avons catégorisé toutes les tâches en fonction de leur nécessité et de leur apport au projet. Certaines des tâches à réaliser mentionnées ci-dessus sont optionnelles. Dans un souci d'efficacité, nous avons choisi d'éliminer les tâches suivantes :

- Les rendez-vous client si ils ne sont pas provoqués par le client lui-même
- La vue HTLM de monProfil
- La personnalisation du bootstrap en fichier CSS
- Le codage des champs d'autocomplétion en JavaScript
- L'utilisation du service OpenStreetMap

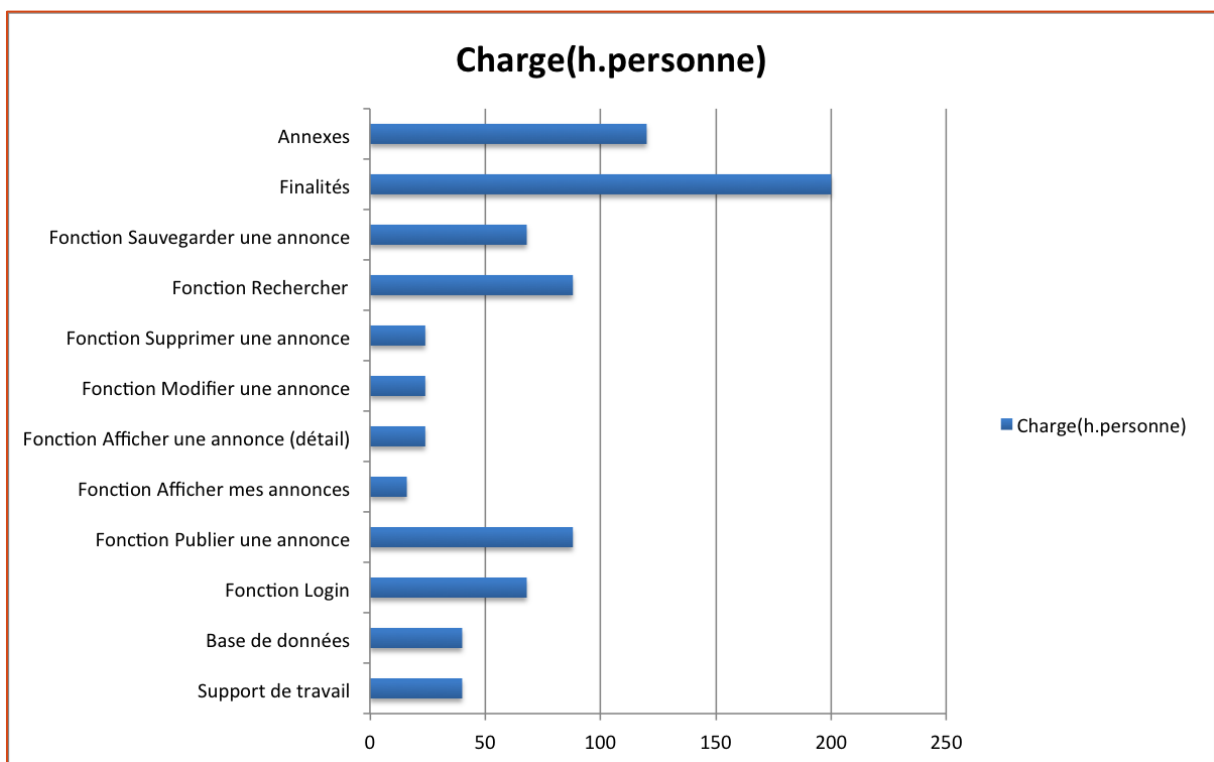


FIGURE 10 : CHARGE EN H.PERSONNE POUR UNE DIFFCULTE MINIMALE

Pour ce nouveau plan d'action, nous avons besoin d'un **total de 800 heures de travail** sur le projet. Comme dit précédemment, nous avons la possibilité de travailler 928 heures sur ce projet. Ainsi, nous pouvons nous permettre d'ajouter quelques options supplémentaires à notre SI.

ESTIMATION CHARGE TOTALE POUR UNE DIFFICULTE MOYENNE

Nous choisissons de rajouter certaines étapes de développement supprimées précédemment :

- La personnalisation du bootstrap en fichier CSS
- Le codage des champs d'autocomplétion en JavaScript

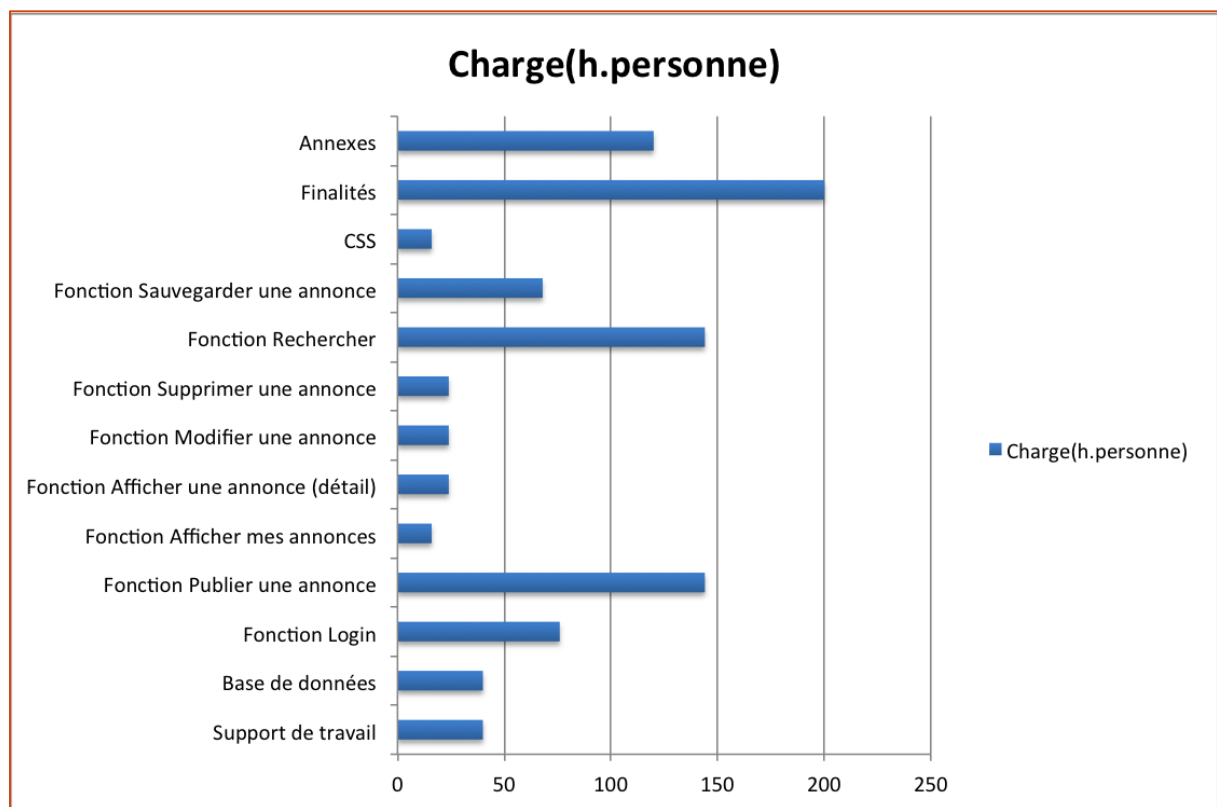


FIGURE 11 : CHARGE EN H.PERSONNE POUR UNE DIFFICULTE MOYENNE

Cette dernière solution nous demande **936 heures de travail** pour une capacité de 928 heures de travail ce qui nous semble raisonnable.

COMPARAISON DES SOLUTIONS

Nous pensons donc nous concentrer sur la dernière solution. Si nous prenons trop de retard, nous nous rabattons sur la solution de difficulté minimale et si nous arrivons à gagner du temps, nous essaierons d'intégrer le service OpenStreetMap de la solution de difficulté maximum.

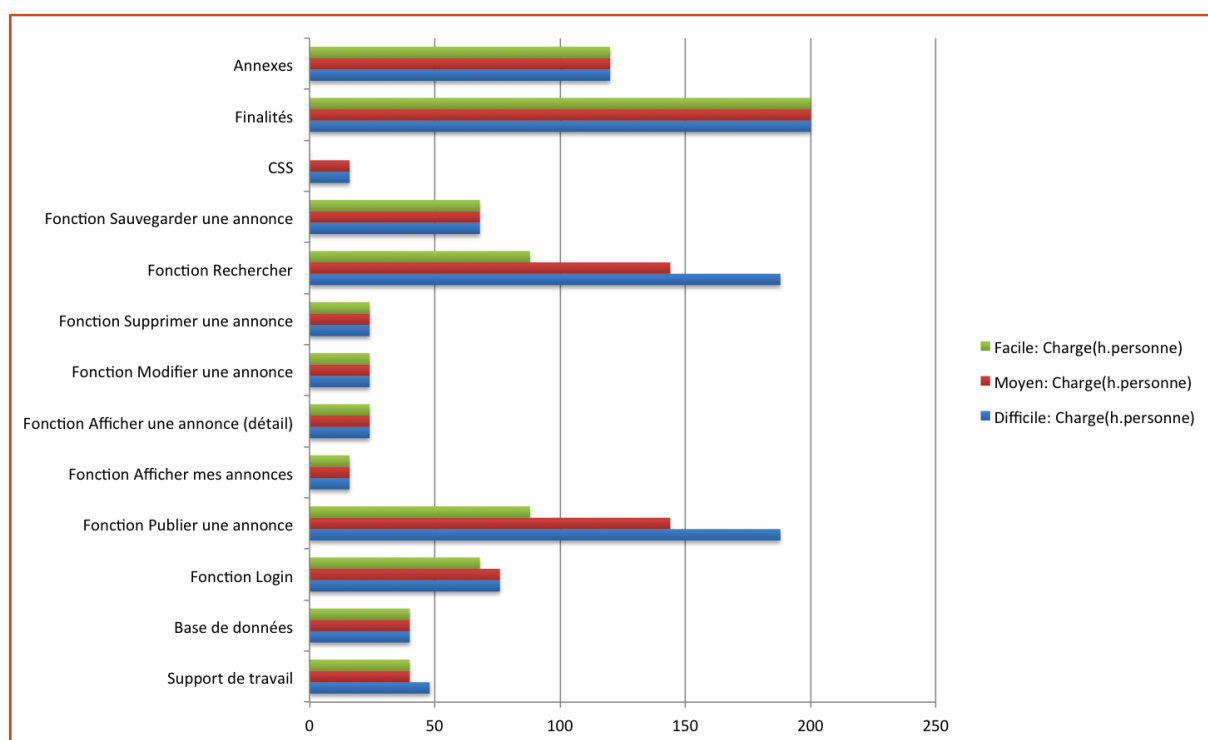


FIGURE 12 : COMPARAISON DE TOUTES LES SOLUTIONS SELON LEUR DIFFICULTE DE MISE EN PLACE

PARTIE V : GLOSSAIRE

Le glossaire suivant a pour but de répertorier le vocabulaire lié au projet et à son développement

ANNONCE	Événement créé par un conducteur décidant de partager son trajet.
CLIENT	L'association I2D.
CONDUCTEUR	Individu qui propose le covoiturage (ici qui possède le véhicule).
COVOITURAGE	Le covoiturage est l'utilisation conjointe et organisée (à la différence de l'auto-stop) d'un véhicule, par un conducteur non professionnel et un ou plusieurs tiers passagers, dans le but d'effectuer un trajet commun.
PASSAGER	Personne qui participe au covoiturage en tant que voyageur ponctuel.
PROPOSITION	Proposition de covoiturage (cf « Annonce »).
UTILISATEUR	Personne qui va utiliser le service de covoiturage, aussi bien passager que conducteur.
TRAJET	Parcours du covoiturage d'un endroit à un autre.
API GOOGLE MAPS	Api de Google permettant de géolocaliser des adresses sur une carte à l'aide de sa latitude et de sa longitude.
OMG	Object Management Group Association américaine à but non lucratif créée en 1989 dont l'objectif est de standardiser et promouvoir le modèle objet sous toutes ses formes.

PARTIE VI : TABLE DES ILLUSTRATIONS

Figure 1 : Modélisation BPMN « Authentifier l'utilisateur »	5
Figure 2 : Modélisation BPMN « ACCEDER AUX PROPOSITIONS DE COVOITURAGE »	6
Figure 3 : Modélisation BPMN « CONSULTER DES PROPOSITIONS DE COVOITURAGE »	7
Figure 4 : Modélisation BPMN « PUBLIER UNE ANNONCE DE COVOITURAGE »	8
Figure 5 : Modélisation BPMN « MODIFIER UNE ANNONCE DE COVOITURAGE »	9
Figure 6 : Modélisation BPMN « SUPPRIMER UNE ANNONCE DE COVOITURAGE »	10
Figure 7 : Description des couleurs potentiellement utilisable dans l'application EMACovoit'	Erreur !
Signet non défini.	
Figure 8 : Modèle Conceptuel des donnée de l'application EMACovoit'	22
Figure 9 : Charge en h.personne pour une difficulté maximale	27
Figure 10 : Charge en h.personne pour une difficulté minimale	28
Figure 11 : Charge en h.personne pour une difficulté moyenne	29
Figure 12 : Comparaison de toutes les solutions selon leur difficulté de mise en place	30
Figure 13 : Modèle Relationnel des DONNÉES de l'application EMACovoit'	33

PARTIE VII : ANNEXES

MODELE RELATIONNEL DE DONNEES (MRD)

Le modèle relationnel de données (MRD) représente la base de données du système d'information comme un ensemble de tables. Elles constituent donc la structure logique du modèle relationnel. Le système est ensuite libre d'utiliser n'importe quelle technique et méthode de stockage dès lors qu'il est possible de relier ces structures à des tables au niveau logique.

On peut définir le modèle relationnel de données de la manière suivante :

- les données sont organisées sous forme de tables à deux dimensions, encore appelées relations, dont les lignes sont appelées n-uplet.
- les données sont manipulées par des opérateurs de l'algèbre relationnelle.
- l'état cohérent de la base est défini par un ensemble de contraintes d'intégrité.
- Le modèle relationnel de données est obtenu à partir du modèle conceptuel des données (MCD) grâce aux règles de transformations MCD vers MRD.

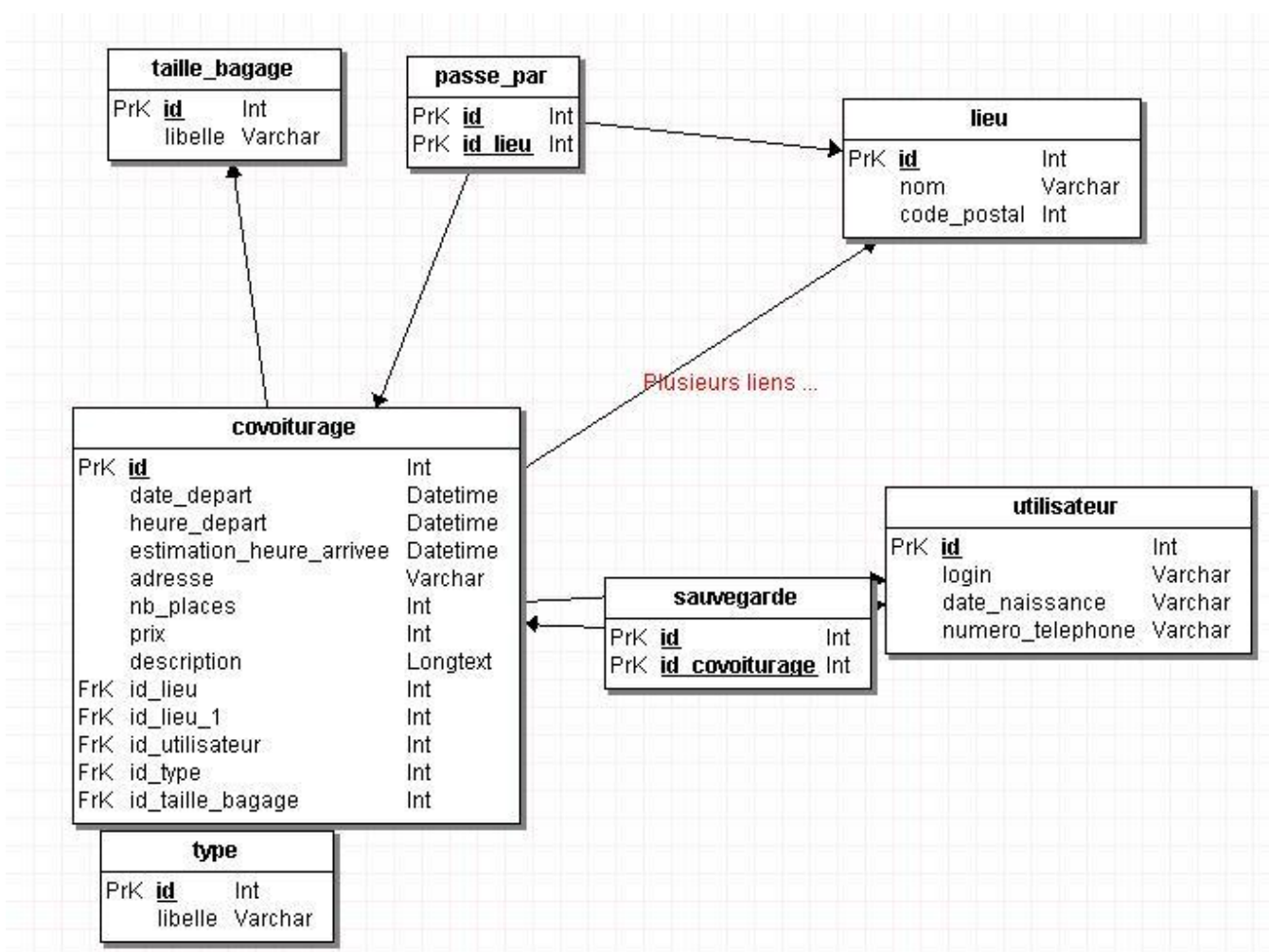


FIGURE 13 : MODELE RELATIONNEL DES DONNEES DE L'APPLICATION EMACOVOIT'

Feuille de calcul de l'estimation de la taille du projet et de la charge associée à chaque fonction

FEUILLE DE CALCUL DE L'ESTIMATION DE LA TAILLE DU PROJET ET DE LA CHARGE ASSOCIEE A CHAQUE FONCTION

ESTIMATION SUR LE CHOIX DE DIFFICULTÉ FACILE DES TÂCHES À EFFECTUER						
Tâches à réaliser	Coefficient	Charge(h.4personnes)	Charge(h.personne)	Outil	Difficulté	Commentaire/Nombre de tâches
Support de travail		10	40			2
Plannifier le developpement		5	20		Moyen	Obligatoire
Suivre l'avancement		5	20		Moyen	Obligatoire
Base de données		10	40			2
MCD - MRD		5	20		Moyen	Obligatoire
Construction de la base de données en SQL		5	20		Facile	Obligatoire
Fonction Login	3	17	68			2
Vue accueil		2	8	HTML	Facile	Obligatoire
Interfaçage de l'annuaire LDAP		15	60		Difficile	Obligatoire
Fonction Publier une annonce	5	22	88			2
Vue formulaire de publication		2	8	HTML	Facile	Obligatoire
Publier une annonce		20	80	Perl/Dancer	Moyen	Obligatoire
Fonction Afficher mes annonces	1	4	16			2
Vue liste des annonces		2	8	HTML	Facile	Obligatoire
Afficher mes annonces		2	8	Perl/Dancer	Moyen	Obligatoire
Fonction Afficher une annonce (détail)	2	6	24			2
Vue annonce		2	8	HTML	Facile	Obligatoire
Afficher le détail d'une annonce		4	16	Perl/Dancer	Moyen	Obligatoire
Fonction Modifier une annonce	2	6	24			2
Vue modification annonce		2	8	HTML	Facile	Obligatoire
Modifier une annonce		4	16	Perl/Dancer	Moyen	Obligatoire
Fonction Supprimer une annonce	2	6	24			2
Vue suppression annonce		2	8	HTML	Facile	Obligatoire
Supprimer une annonce		4	16	Perl/Dancer	Moyen	Obligatoire
Fonction Rechercher	5	22	88			2
Vue rechercher		2	8	HTML	Facile	Obligatoire
Rechercher parmi les annonces		20	80	Perl/Dancer	Moyen	Obligatoire
Fonction Sauvegarder une annonce	3	17	68			2
Vue sauvegarder		2	8	HTML	Facile	Obligatoire
Sauvegarder d'une annonce		15	60	Perl/Dancer	Moyen	Obligatoire
Finalités		50	200			3
Intégration		20	80		Difficile	Obligatoire
Déploiement		10	40		Difficile	Obligatoire
Tests		20	80		Facile	Obligatoire
Annexes		30	120			3
Rédaction d'un rapport		15	60		Moyen	Obligatoire
Rédaction d'une documentation/aide		5	20		Moyen	Obligatoire
Préparation de la présentation finale		10	40		Moyen	Obligatoire

Feuille de calcul de l'estimation de la taille du projet et de la charge associée à chaque fonction

Total heures de travail à fournir	200	800	3 tâches difficiles	26
Heures allouées au projet dans emploi du temps	122	488		
Heures de travail personnel (10h)	110	440		
Total des heures de travail disponibles	232	928		
Charge à rattraper	-32	-128		

Le tableau ci-dessus donne l'ensemble des estimations des tâches à effectuer dans le cas où la difficulté serait minimale

ESTIMATION SUR LE CHOIX DE DIFFICULTÉ MOYEN DES TÂCHES À EFFECTUER						
Tâches à réaliser	Coefficient	Charge(h.4personnes)	Charge(h.personne)	Outil	Difficulté	Commentaire/Nombre de tâches
Support de travail		10	40			2
Plannifier le développement		5	20		Moyen	Obligatoire
Suivre l'avancement		5	20		Moyen	Obligatoire
Base de données		10	40			2
MCD - MRD		5	20		Moyen	Obligatoire
Construction de la base de données en SQL		5	20		Facile	Obligatoire
Fonction Login	3	19	76			3
Vue accueil		2	8	HTML	Facile	Obligatoire
Vue monProfil		2	8	HTML	Facile	Facultatif
Interfaçage de l'annuaire LDAP		15	60		Difficile	Obligatoire
Fonction Publier une annonce	7,5	36	144			6
Vue formulaire de publication		2	8	HTML	Facile	Obligatoire
Publier une annonce		20	80	Perl/Dancer	Moyen	Obligatoire
Autocomplétion des lieux		4	16	JavaScript	Difficile	Facultatif
Ajouter un lieu		4	16	JavaScript	Difficile	Facultatif
Vérification de l'existence des lieux		4	16	JavaScript	Difficile	Facultatif
Champ d'autocomplétion GoogleMap		2	8	JavaScript	Moyen	Facultatif
Fonction Afficher mes annonces	1	4	16			2
Vue liste des annonces		2	8	HTML	Facile	Obligatoire
Afficher mes annonces		2	8	Perl/Dancer	Moyen	Obligatoire
Fonction Afficher une annonce (détail)	2	6	24			2
Vue annonce		2	8	HTML	Facile	Obligatoire
Afficher le détail d'une annonce		4	16	Perl/Dancer	Moyen	Obligatoire
Fonction Modifier une annonce	2	6	24			2
Vue modification annonce		2	8	HTML	Facile	Obligatoire
Modifier une annonce		4	16	Perl/Dancer	Moyen	Obligatoire
Fonction Supprimer une annonce	2	6	24			2
Vue suppression annonce		2	8	HTML	Facile	Obligatoire
Supprimer une annonce		4	16	Perl/Dancer	Moyen	Obligatoire
Fonction Rechercher	7,5	36	144			6
Vue rechercher		2	8	HTML	Facile	Obligatoire
Rechercher parmi les annonces		20	80	Perl/Dancer	Moyen	Obligatoire
Autocomplétion des lieux		4	16	JavaScript	Difficile	Facultatif
Ajouter un lieu		4	16	JavaScript	Difficile	Facultatif
Vérification de l'existence des lieux		4	16	JavaScript	Difficile	Facultatif
Champ d'autocomplétion GoogleMap		2	8	JavaScript	Moyen	Facultatif

Feuille de calcul de l'estimation de la taille du projet et de la charge associée à chaque fonction

Fonction Sauvegarder une annonce	3	17	68			2
Vue sauvegarder		2	8	HTML	Facile	Obligatoire
Sauvegarder d'une annonce		15	60	Perl/Dancer	Moyen	Obligatoire
CSS	4	16				1
Personnaliser bootstrap		4	16	CSS	Moyen	Facultatif
Finalités	50	200				3
Intégration		20	80		Difficile	Obligatoire
Déploiement		10	40		Difficile	Obligatoire
Tests		20	80		Facile	Obligatoire
Annexes	30	120				3
Rédaction d'un rapport		15	60		Moyen	Obligatoire
Rédaction d'une documentation/aide		5	20		Moyen	Obligatoire
Préparation de la présentation finale		10	40		Moyen	Obligatoire
Total heures de travail à fournir	234	936		9 tâches difficiles		35
Heures allouées au projet dans emploi du temps		122	488			
Heures de travail personnel (10h)		110	440			
Total des heures de travail disponibles		232	928			
Charge à rattraper	2	8				

Le tableau ci-dessus donne l'ensemble des estimations des tâches à effectuer dans le cas où la difficulté serait moyenne

ESTIMATION SUR LE CHOIX DE DIFFICULTÉ MAXIMUM DES TÂCHES À EFFECTUER						
Tâches à réaliser	Coefficient	Charge(h.4personnes)	Charge(h.personne)	Outil	Difficulté	Commentaire/Nombre de tâches
Support de travail		12	48			3
Plannifier le développement		5	20		Moyen	Obligatoire
Suivre l'avancement		5	20		Moyen	Obligatoire
Rendez-vous avec les clients		2	8		Moyen	Facultatif
Base de données		10	40			2
MCD - MRD		5	20		Moyen	Obligatoire
Construction de la base de données en SQL		5	20		Facile	Obligatoire
Fonction Login	3	19	76			3
Vue accueil		2	8	HTML	Facile	Obligatoire
Vue monProfil		2	8	HTML	Facile	Facultatif
Interfaçage de l'annuaire LDAP		15	60		Difficile	Obligatoire
Fonction Publier une annonce	10	47	188			3
Vue formulaire de publication		2	8	HTML	Facile	Obligatoire
Publier une annonce		20	80	Perl/Dancer	Moyen	Obligatoire
Utilisation sur service OpenStreetMap		25	100	JavaScript	Difficile	Facultatif
Fonction Afficher mes annonces	1	4	16			2
Vue liste des annonces		2	8	HTML	Facile	Obligatoire
Afficher mes annonces		2	8	Perl/Dancer	Moyen	Obligatoire
Fonction Afficher une annonce (détail)	2	6	24			2
Vue annonce		2	8	HTML	Facile	Obligatoire
Afficher le détail d'une annonce		4	16	Perl/Dancer	Moyen	Obligatoire
Fonction Modifier une annonce	2	6	24			2
Vue modification annonce		2	8	HTML	Facile	Obligatoire
Modifier une annonce		4	16	Perl/Dancer	Moyen	Obligatoire

Feuille de calcul de l'estimation de la taille du projet et de la charge associée à chaque fonction

Fonction Supprimer une annonce	2	6	24			2
Vue suppression annonce		2	8	HTML	Facile	Obligatoire
Supprimer une annonce		4	16	Perl/Dancer	Moyen	Obligatoire
Fonction Rechercher	10	47	188			3
Vue rechercher		2	8	HTML	Facile	Obligatoire
Rechercher parmi les annonces		20	80	Perl/Dancer	Moyen	Obligatoire
Utilisation sur service OpenStreetMap		25	100	JavaScript	Difficile	Facultatif
Fonction Sauvegarder une annonce	3	17	68			2
Vue sauvegarder		2	8	HTML	Facile	Obligatoire
Sauvegarder d'une annonce		15	60	Perl/Dancer	Moyen	Obligatoire
CSS		4	16			1
Personnaliser bootstrap		4	16	CSS	Moyen	Facultatif
Finalités		50	200			3
Intégration		20	80		Difficile	Obligatoire
Déploiement		10	40		Difficile	Obligatoire
Tests		20	80		Facile	Obligatoire
Annexes		30	120			3
Rédaction d'un rapport		15	60		Moyen	Obligatoire
Rédaction d'une documentation/aide		5	20		Moyen	Obligatoire
Préparation de la présentation finale		10	40		Moyen	Obligatoire
Total heures de travail à fournir		258	1032	5 tâches difficiles		30
Heures allouées au projet dans emploi du temps		122	488			
Heures de travail personnel (10h)		110	440			
Total des heures de travail disponibles		232	928			
Charge à rattraper		26	104			

Le tableau ci-dessus donne l'ensemble des estimations des tâches à effectuer dans le cas où la difficulté serait élevée

Tâches à réaliser	Difficile: Charge(h.personne)	Moyen: Charge(h.personne)	Facile: Charge(h.personne)
Support de travail	48	40	40
Base de données	40	40	40
Fonction Login	76	76	68
Fonction Publier une annonce	188	144	88
Fonction Afficher mes annonces	16	16	16
Fonction Afficher une annonce (détail)	24	24	24
Fonction Modifier une annonce	24	24	24
Fonction Supprimer une annonce	24	24	24
Fonction Rechercher	188	144	88
Fonction Sauvegarder une annonce	68	68	68
CSS	16	16	0
Finalités	200	200	200
Annexes	120	120	120

Le tableau ci-contre récapitule la charge en h.personne pour chaque tâche à réaliser et dans chaque configuration :

- Difficile
- Facile
- Moyen