

Rapport de projet

Semestre 5



Alex **CAMPOS**

Fabien **LAURENT**

Bachelor 3 CS2i Bourgogne

Remerciements :

Nous tenons tout d'abord à remercier notre enseignant, Monsieur Berthet, de nous avoir aiguillé, conseillé et apporté les connaissances nécessaires afin de mener à bien la première partie de notre projet.

Sommaire :

Remerciements :	2
Sommaire :	3
Présentation du sujet :	4
Organisation du travail :	5
Les étapes d'avancement :	6
Cahier des charges :	12
Contexte :	12
Objectif :	12
Périmètre :	12
Calendrier :	12
Fonctionnalités souhaitées :	12
Choix des technologies :	13
Maquettes :	17
Conclusion :	21
Sources utilisées :	22

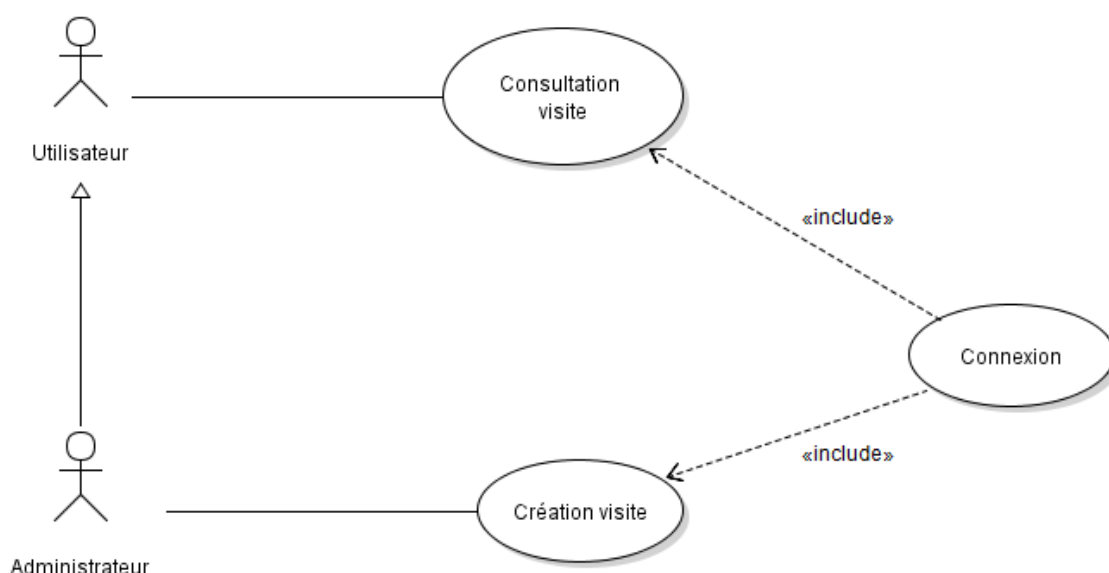
Présentation du sujet :

Cette année, notre projet ne dure pas un semestre, mais est étendu sur toute l'année scolaire. Nous avons donc choisi avec mon camarade, de travailler sur un plug-in de visite d'appartement/maison en 3D.

Lorsqu'un utilisateur quelconque arrivera sur un site tel que Leboncoin ou bien un site de vente/location de maison ou appartement, il pourra se munir du plug-in et pourra lancer une visite de l'endroit directement en 3D.

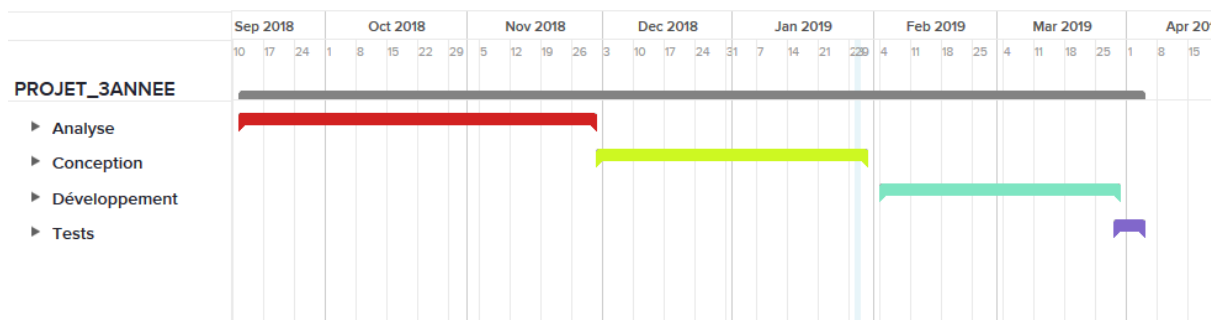
L'utilisateur pourra se déplacer dans chaque pièce, modifier l'emplacement du soleil, mais aussi grâce à un onglet « Plus », rajouter des meubles, au préalable déjà rentré dans le plug-in ou bien, créer son propre meuble avec ses propres dimensions.

Mais il pourra aussi grâce à cette onglet, visualiser plusieurs informations sur la bâtisse et aussi visiter en street view.

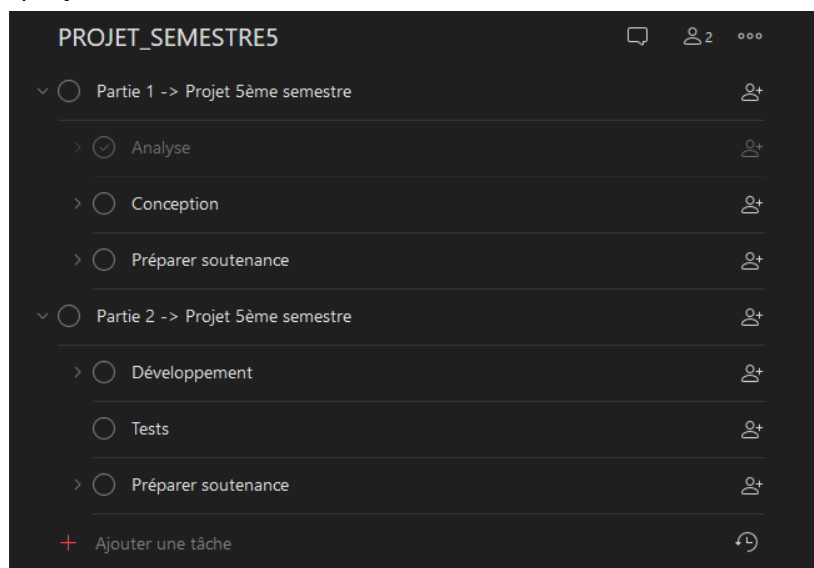


Organisation du travail :

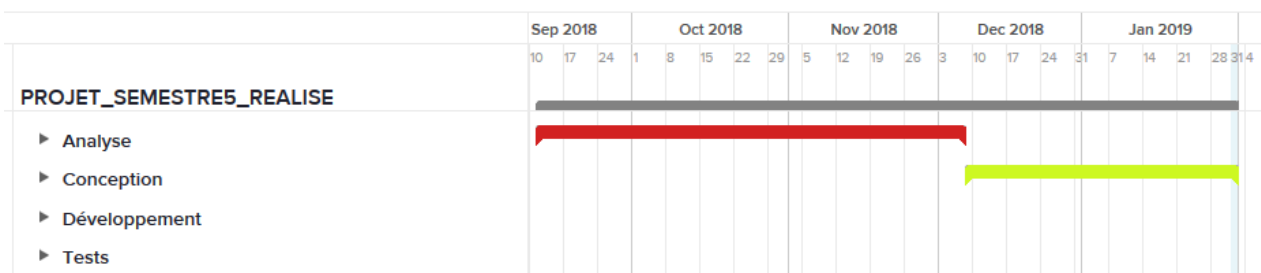
Nous avons tout d'abord organiser et planifier le travail à fournir pour ce projet en établissant un diagramme de **Gantt prévisionnel**.



Grâce à l'outil en ligne **ToDoist**, nous avons pu mettre chaque tâche à effectuer pour les deux parties du projet.

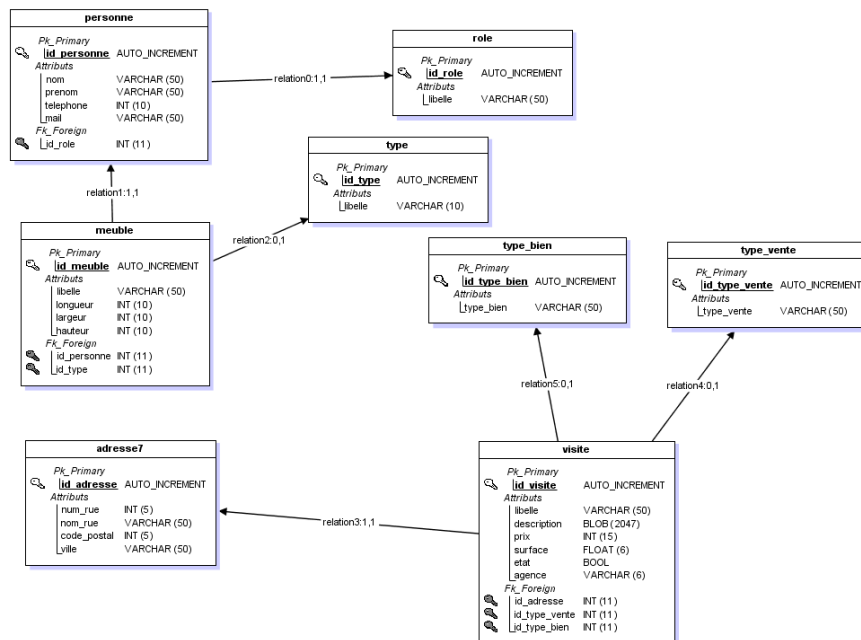


Après avoir organisé et planifié les tâches du projet, nous avons réparti celles-ci au sein du groupe. Nous avons donc réalisé nos diagrammes d'activité et de cas d'utilisations ensemble avec le logiciel **VioletUML**. Pour finir sur la partie **Analyse**, nous avons établi un diagramme de Gantt réalisé.



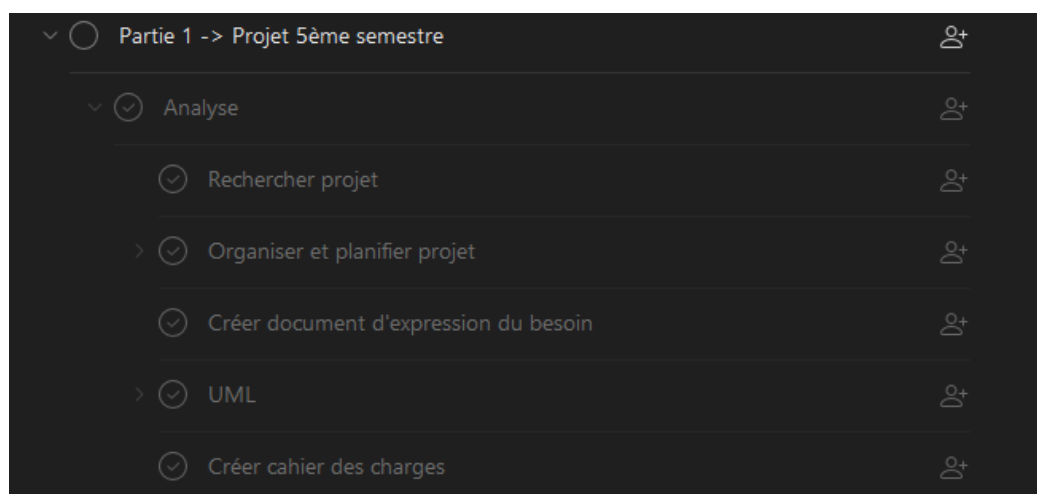
Ensuite, nous sommes passés à la partie **Conception** avec chacun, des tâches différentes à effectuer que nous avons mis en commun, afin de savoir où nous en étions dans celles-ci.

Par exemple, voici le Modèle Physique des Données qui a servi à implanter une base de données pour notre projet :



Les étapes d'avancement :

Pour mener à bien ce projet, nous ne voulions pas partir tête baissée, c'est pour cela que nous avons organisé et listé chaque tâche à réaliser pour le projet. En premier lieu, nous avons la partie « **Analyse** » qui regroupe toutes les tâches à réaliser avant de passer à la conception du projet :



Nous n'avons pas rencontré trop de problèmes avec cette partie car nous avons beaucoup travaillé ensemble afin d'être le plus performant possible.

Voici les diagrammes que nous avons réalisés pour le projet :

Diagramme cas utilisation plug-in :

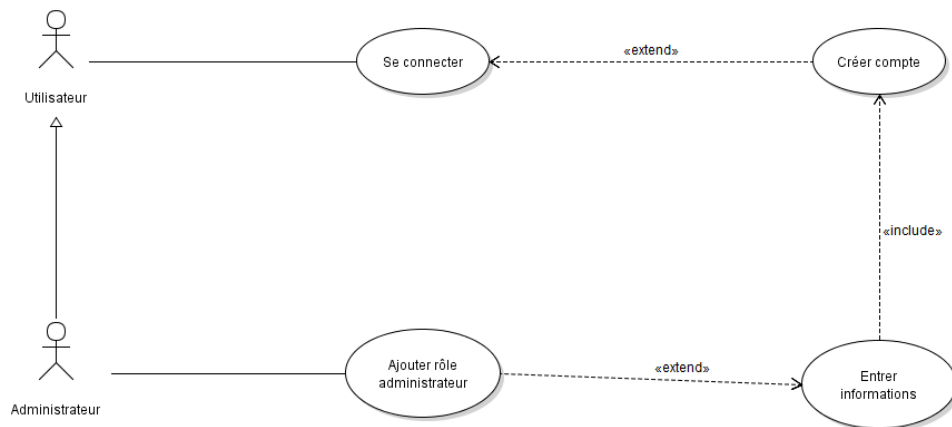


Diagramme cas utilisation création visite :

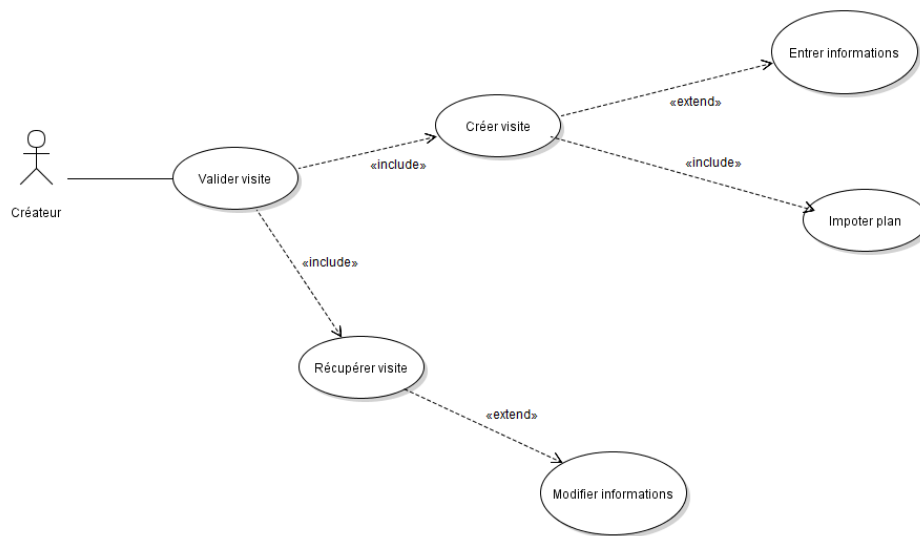


Diagramme cas utilisation visite :

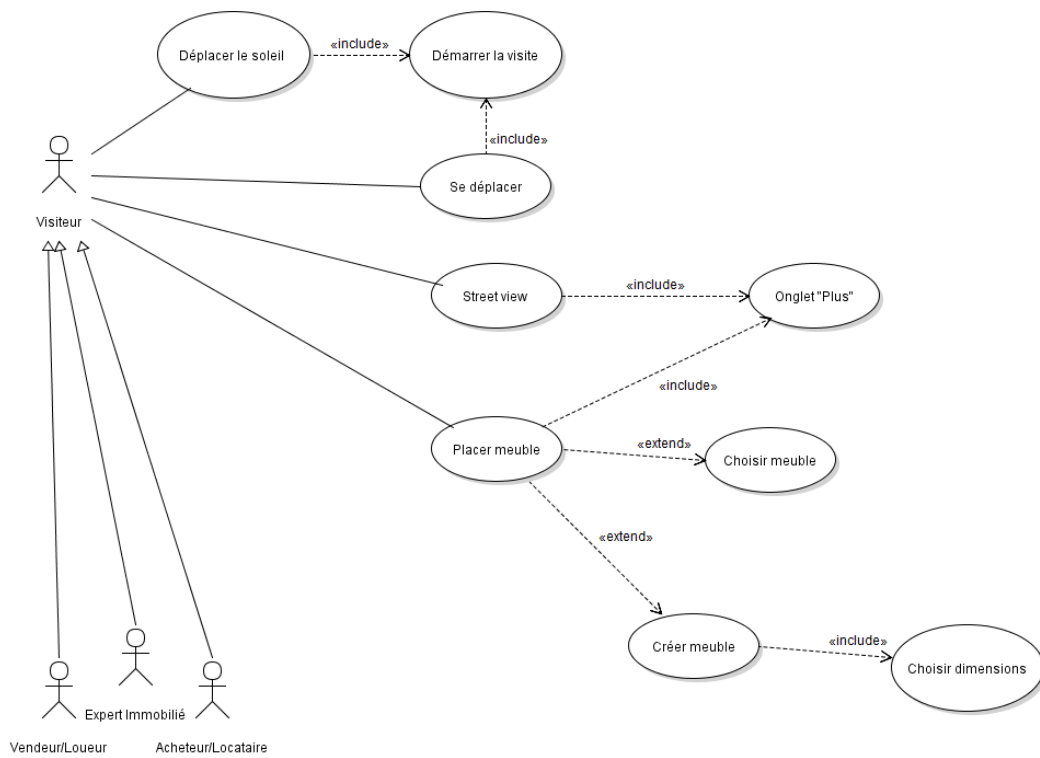


Diagramme d'activité création visite :

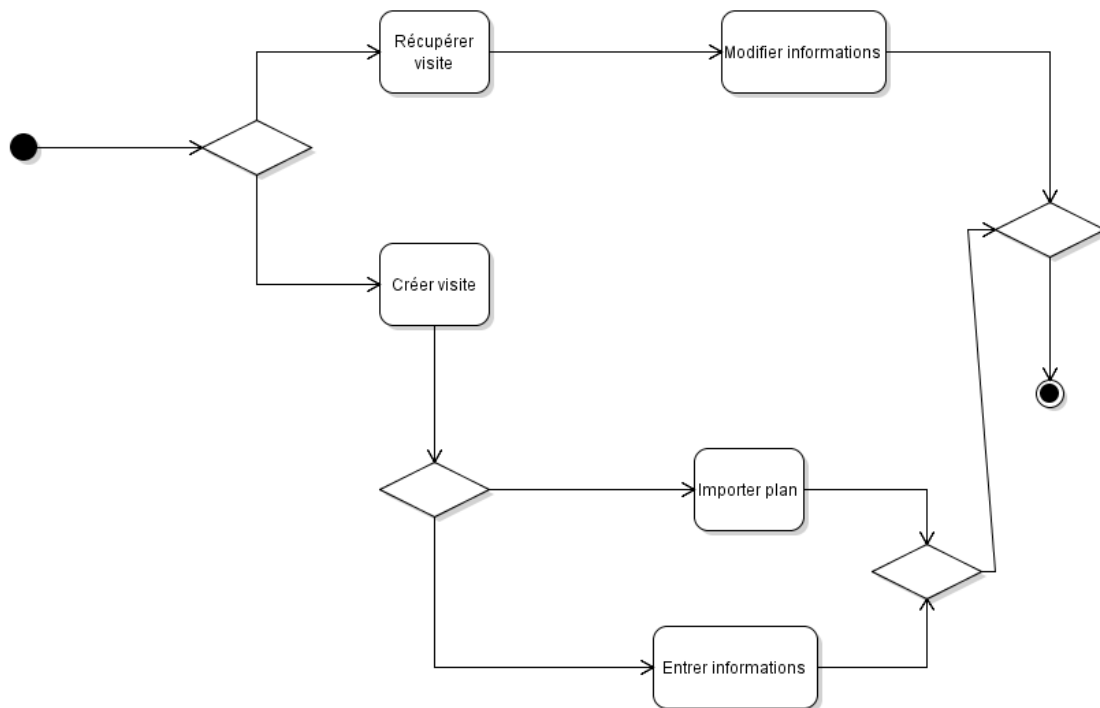
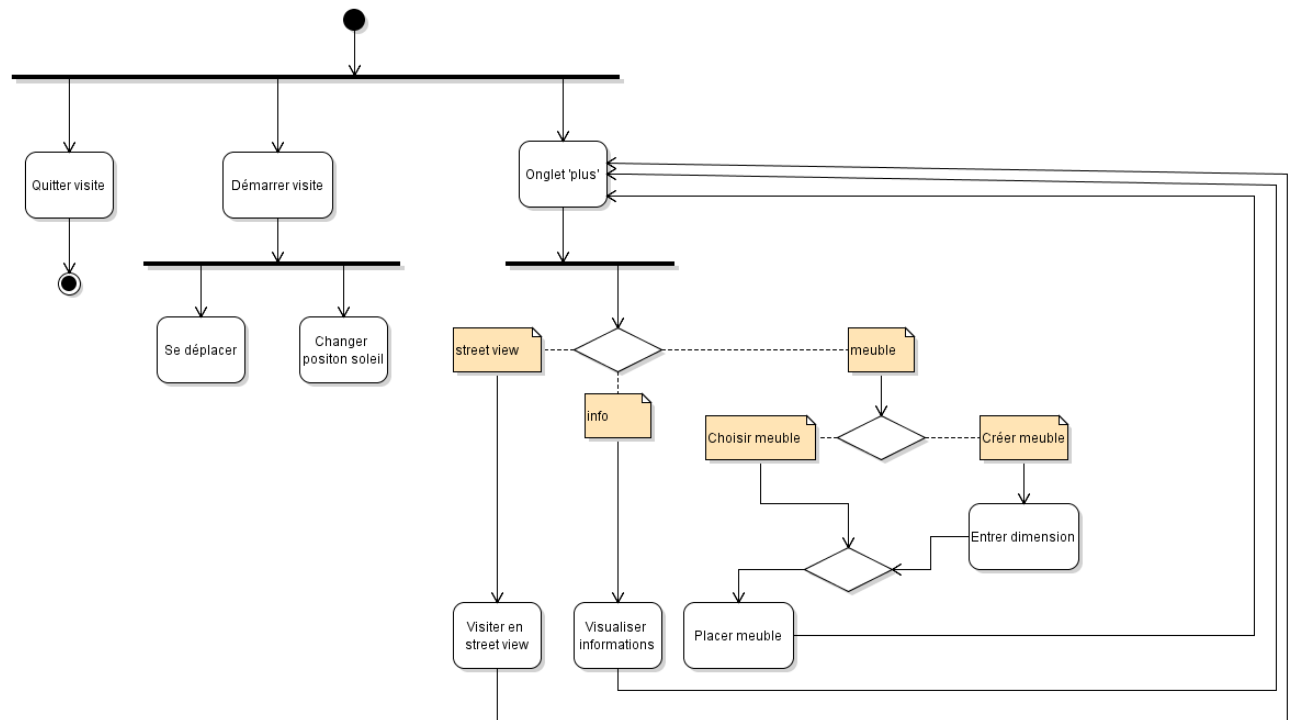
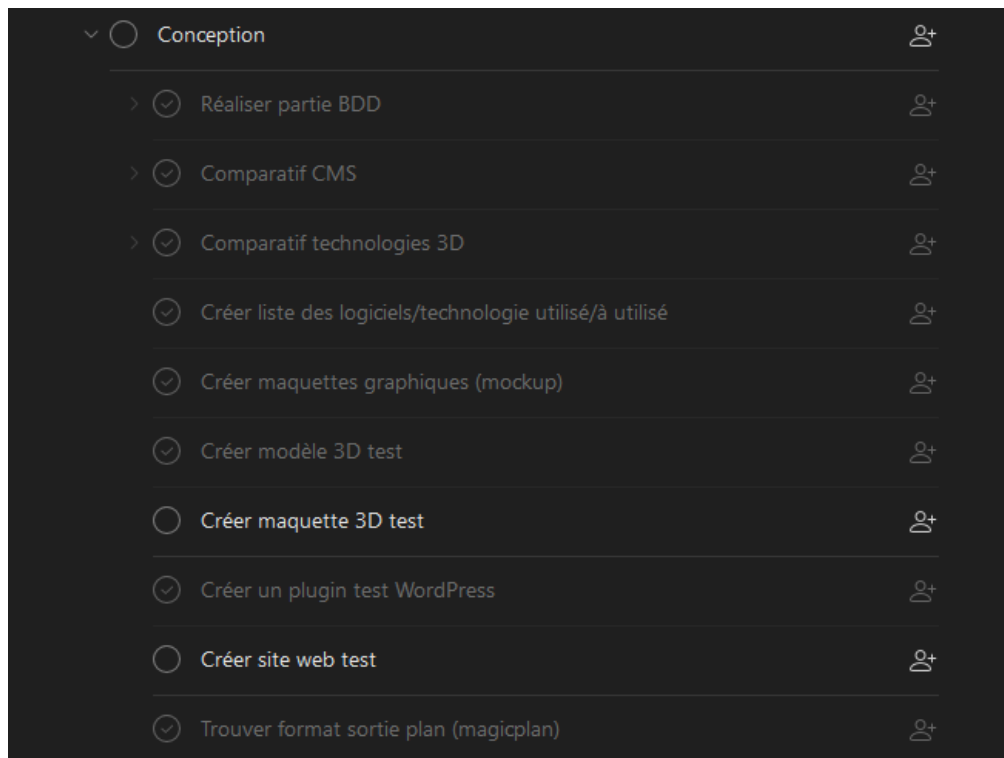


Diagramme d'activité visite :



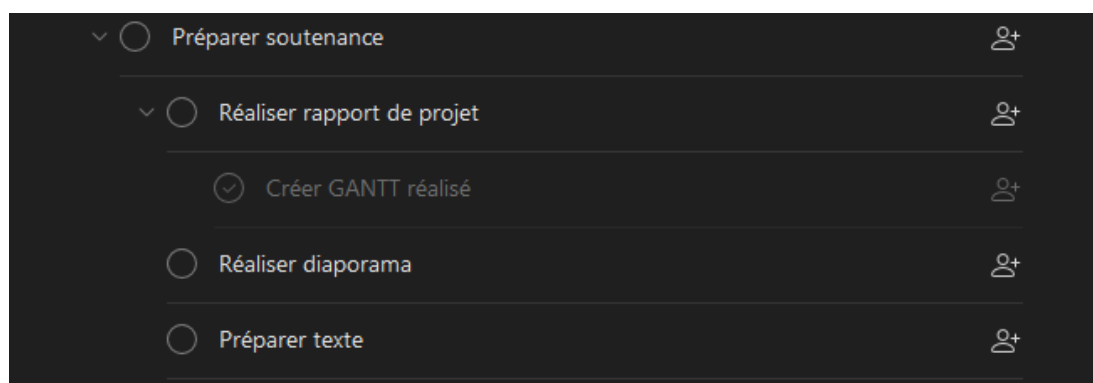
En deuxième partie, nous avons la partie « **Conception** ». Dans cette partie, nous allons retrouver les tâches à effectuer avant le développement.



Dans cette partie, nous nous sommes partagé les tâches afin de ne pas perdre de temps sur le projet :

Alex	Fabien
Comparatif CMS	Réaliser partie BDD
Comparatif technologie 3D	Créer maquettes graphiques (mockup)
Créer liste des logiciels/technologies utilisées/à utiliser	Créer plug-in test WordPress
Créer modèle 3D test	Trouver format sortie plan (magicplan)

Pour finir, la dernière partie est la « **Préparation de soutenance** ».



- ☐ Préparer soutenance
- ☐ Réaliser rapport de projet
 - ☒ Créer GANTT réalisé
 - ☐ Réaliser diaporama
- ☐ Préparer texte

Voici la répartition des tâches dans le groupe :

Alex	Fabien
Réaliser diaporama	Réaliser rapport de projet
Préparer texte	Créer Gantt réalisé

Cahier des charges :

Contexte :

Le sujet pour ce projet est libre, nous pouvons utiliser n'importe quel langage ou technologie. Les logiciels ou applications pour le partage et le versionning du projet sont souhaités.

Objectif :

Notre objectif pour ce projet est de créer un plug-in de visite 3D.

Périmètre :

Nombre de personnes : 2

Ressources : 2 ordinateurs

Calendrier :

Toute l'année de Bachelor 3

Fonctionnalités souhaitées :

- Page de connexion
- Page d'inscription
- Déplacer dans environnement 3D
- Changer position du soleil
- Onglet « Plus »
 - Placer meubles (avec dimensions/mesures)
 - Déplacer meubles
 - Importer modèle 3D
 - Visualiser informations pièces
 - Visiter en street view

Choix des technologies :

Pour ce projet, nous voulions créer une application qui serait disponible sur un maximum de plateformes et de médias. C'est pourquoi nous avons décidé de créer un plug-in au travers d'un CMS qui serait accessible au plus grand nombre.

Pour le choix du CMS :

Wordpress	Plus grande communauté au monde Milliers de plug-ins Applications smartphone Gratuit et mis à jour régulièrement
Drupal	Plus accès côté "blog" donc utile pour les sites galerie, photos, etc. Grande communauté Mis à jour régulièrement
Joomla	Grande communauté Milliers de plug-ins Interface simple d'utilisation
ExpressionEngine	Licence annuelle de 300 dollars Support disponible Centaines de plug-ins

Notre choix s'est porté sur Wordpress. Nous sommes novices dans l'utilisation de CMS et Wordpress comporte la plus grande communauté (pour le CMS) au monde. De plus, il contient déjà des milliers de plug-ins, des applications pour smartphone (responsive design). Il est gratuit et mis à jour régulièrement.

Après avoir choisi le CMS, nous devons choisir des technologies qui nous permettraient d'utiliser un environnement 3D pour nos visites :

Moteur rendu 3D	Caractéristiques	Avantages	Limites
WebGL	<ul style="list-style-type: none"> - Compatible avec tous les navigateurs et présent de manière native - Utilise JavaScript - Libre et gratuit 	<ul style="list-style-type: none"> - Utilisé et compatible avec de nombreux framework et environnements - Natif sur tous les navigateurs donc possibilité de lecture des contenus WebGL avec n'importe quel appareil - Documentation riche 	<ul style="list-style-type: none"> - Gourmand en ressources - Faibles performances sur appareil mobile
Flash	<ul style="list-style-type: none"> - Pré-installé sur de nombreux navigateurs mais est souvent désactivé par défaut - Absent nativement des navigateurs mobiles - Utilise ActionScript - Propriétaire et gratuit 	<ul style="list-style-type: none"> - Langage de très bas niveau (proche de l'assembleur) permettant une manipulation précise de la machine et donc une meilleure gestion des performances - Documentation riche 	<ul style="list-style-type: none"> - Doit être activé et installé pour lire les contenus flash - Absent sur les appareils mobiles - Non normalisé par le W3C
DirectX	<ul style="list-style-type: none"> - Pas d'information sur les navigateurs compatibles - Possibilité de l'utiliser sur le web - Indisponible sur mobile - Compatible avec les systèmes Windows et autres via Wine - Utilise tout langage supporté par le framework .Net - Propriétaire et gratuit 	<ul style="list-style-type: none"> - Documentation riche 	<ul style="list-style-type: none"> - Doit être installé pour lire les contenus DirectX - Incompatibilité avec les autres systèmes sans installation de Wine - Absent sur les appareils mobiles - Non normalisé par le W3C

Nous avons choisi WebGL comme moteur de rendu 3D car celui-ci répond à tous nos critères de compatibilité, en plus d'être très bien documenté.

Technologie	Caractéristiques	Avantages	Limites
Unity 3D	<ul style="list-style-type: none"> - Compatible avec de nombreux systèmes d'exploitation - Compatible avec de nombreuses bibliothèques 3D - Supporte de nombreux formats 3D - Utilise JavaScript ou C# - Propriétaire et payant (version professionnelle) - Supporte les formats: .fbx/.dae/.3ds/.obj/.dxf/.skp 	<ul style="list-style-type: none"> - Simplicité de mise en place d'un projet "temps réel" - Simplicité d'utilisation et de création de plus gros projet WebGL - Documentation très riche 	<ul style="list-style-type: none"> - Fichiers générés en WebGL lourds - Temps de chargement plus long
Vulkan	<ul style="list-style-type: none"> - Compatible avec de nombreux systèmes (sauf MacOS et iOS) - Utilise WebGL - Semi-libre et gratuit - Supporte les formats: .obj/.fbx 	<ul style="list-style-type: none"> - Fichiers légers - Temps de chargement plus court - Peut être utilisé avec Unity 3D - Documentation riche 	<ul style="list-style-type: none"> - Nécessite MoltenVK ou SDKLunarG pour fonctionner sous MacOS ou iOS
Three.js	<ul style="list-style-type: none"> - Compatible avec tous les navigateurs - Utilise WebGL - Libre et gratuit - Supporte les formats: .gltf/.dae/.fbx/.obj/.3ds/.dxf(non natif) 	<ul style="list-style-type: none"> - Fichiers légers - Temps de chargement plus court - Documentation riche 	<ul style="list-style-type: none"> - Difficulté de la mise en place de grands projets
Babylon js	<ul style="list-style-type: none"> - Compatible avec tous les navigateurs - Utilise WebGL - Libre et gratuit - Supporte les formats: .gltf/.babylon/.obj 	<ul style="list-style-type: none"> - Fichiers légers - Temps de chargement plus court - Documentation riche 	<ul style="list-style-type: none"> - Difficulté de la mise en place de grands projets

Nous avons décidé de prendre BabylonJS car il génère des fichiers légers, possède une grande communauté, une grande documentation et possède un avantage par rapport à Three.JS (les deux bibliothèques sont très similaires) : il possède un environnement qui permet de tester et partager nos projets.

Au niveau du choix de programmation et technologies, nous utilisons le CMS **WordPress** qui utilise les langages PHP et HTML, la bibliothèque JavaScript, **BabylonJS** (rendu 3D) avec comme moteur de rendu, **WebGL** et enfin pour créer des modèles 3D, le logiciel **Blender**.



Afin de mener à bien notre projet nous utilisons le logiciel **Discord** pour communiquer, **GitHub** pour partager nos fichiers et pour montrer l'évolution dans le projet, **ToDoist** qui nous sert à planifier nos tâches, **VioletUML** pour concevoir nos diagrammes, pour les documents comme le rapport ou le PowerPoint nous disposons d'une suite **LibreOffice** et **Office**, pour la conception de maquette **Balsamiq Mockups**, pour le développement **NotePad++** et **Visual Code**, pour la création de plans **Magicplan** et enfin pour visualiser ceux-ci, **ABViewer**.



Maquettes :

Ici, nous allons vous montrer les maquettes que nous avons réalisés pour le projet.

Ci-dessous, vous verrez des maquettes Mockup réalisé pour imaginer à quoi ressemblera notre plug-in :

Page d'accueil/connexion et création compte :

Diagramme illustrant les maquettes de la page d'accueil/connexion et création compte. La page d'accueil (gauche) est intitulée 'Bienvenue' et contient un onglet 'Connexion' avec des champs pour 'Email*' et 'Mot de passe*', un lien 'Contact', un lien 'Créer un compte' (surligné), et un bouton 'Valider'. La page de création de compte (droite) est intitulée 'Créer un compte' et contient des champs pour 'Nom*', 'Prénom*', 'Téléphone*', 'Email*', et 'Mot de passe*', un bouton 'Retour', un lien 'Contact', et un bouton 'Valider'. Une ligne de connexion relie le lien 'Créer un compte' de la page d'accueil au bouton 'Retour' de la page de création de compte.

Page créateur/créer visite :

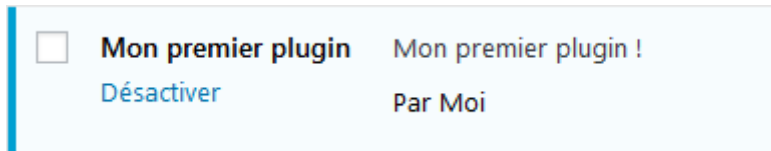
Diagramme illustrant les maquettes de la page créateur/créer visite. La page créateur (gauche) est intitulée 'Créateur' et contient un onglet 'Nom Créateur', un bouton 'Créer visite' (surligné), un bouton 'Récupérer visite', et un bouton 'Retour'. La page de création de visite (droite) est intitulée 'Créer visite' et contient des champs pour 'Libelle:', 'Prix:', 'Surface:', 'Description:', 'Etat:' (avec une liste déroulante : 'A vendre', 'Vendu', 'A louer', 'Louer'), 'Type' (avec des boutons radio : 'Appartement', 'Maison'), 'Adresse:', 'Agence:', 'Photo:', et un bouton 'Insérer'. Une ligne de connexion relie le bouton 'Créer visite' de la page créateur au bouton 'Insérer' de la page de création de visite.

Page utilisateur/création meuble :

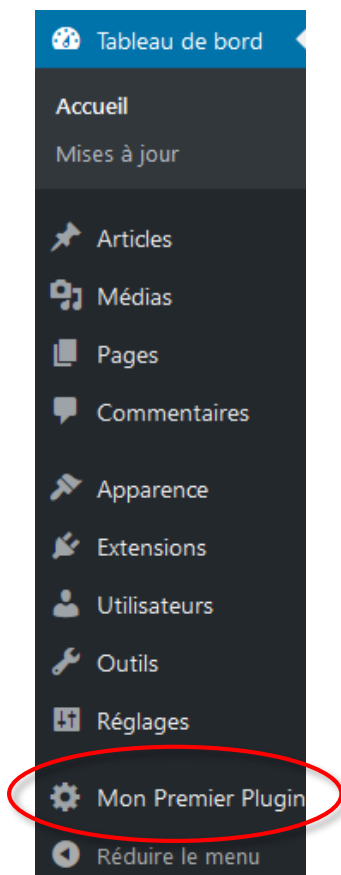
Diagramme illustrant les maquettes de la page utilisateur/création meuble. La page utilisateur (gauche) est intitulée 'Utilisateur' et contient un bouton 'Retour', un bouton '+ Plus' (surligné), et une image d'un meuble. Le bouton '+ Plus' est connecté à un menu déroulant (au centre) qui contient les options : '+ Plus', 'Informations', 'Street View', 'Meuble' (surligné), 'Choisir meuble', et 'Créer meuble'. Le menu déroulant est connecté à la page de création de meuble (droite), intitulée 'Formulaire création meuble', qui contient des champs pour 'Longueur', 'Hauteur', 'Largeur', 'Type meuble', et des boutons 'Valider' et 'annuler'.

Ensuite, voici le plug-in test créer sur **WordPress**.

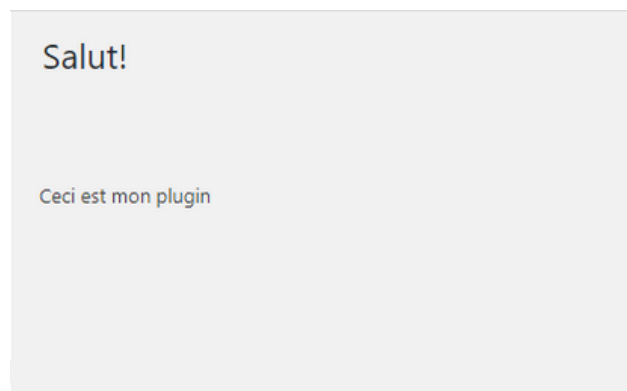
On peut voir ci-dessous ce que l'on retrouve, quand nous allons à la page extension sur le Tableau de bord d'administration de WordPress. Il y a le nom du plug-in, sa description et le nom de l'auteur du plug-in.



Lorsque le plug-in est activé, on retrouve ceci dans le menu du Tableau de bord de WordPress.



Enfin, lorsque l'on appuie sur « Mon Premier Plugin » voici le résultat :



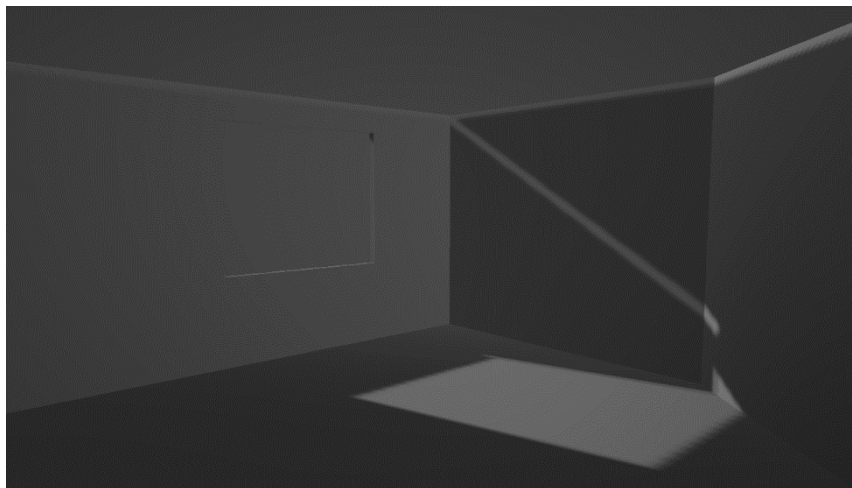
Pour se familiariser avec la 3D, nous avons créé un modèle 3D et un environnement de test en 3D. Cet environnement a été conçu grâce à la bibliothèque JavaScript, **BabylonJS**. Ce sont deux terrains plats, un en haut et un en bas et des colonnes disposées entre ces deux terrains plats.



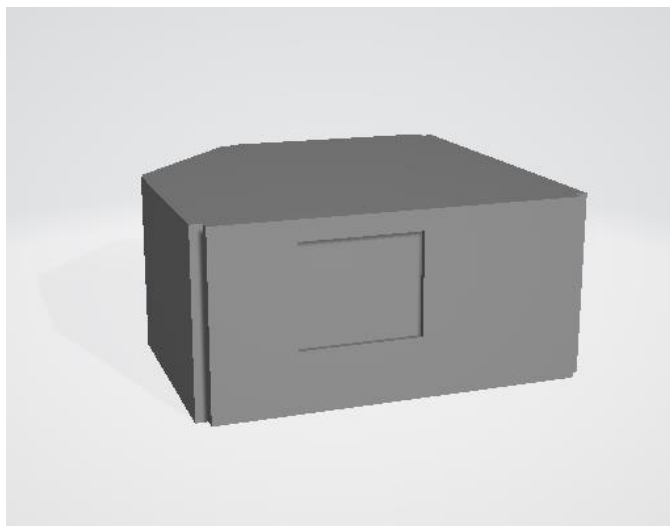
C'est dans cet environnement que nous placerons les modèles 3D des visites. **BabylonJS** nous offre la possibilité d'importer nos propres modèles. Dans un premier temps, nous avons imaginé une méthode qui aurait permis de récupérer des plans en 2D au travers du logiciel **Magicplan**, puis de l'exporter sous un format 3 dimensions afin de le récupérer dans notre environnement 3D. Cela aurait permis de faciliter la création des visites, mais le format de sortie de **Magicplan** est propriétaire et incompatible avec **BabylonJS**.

Le modèle 3D ci-dessous a été conçu grâce au logiciel de modélisation 3D **Blender**.

Intérieur :



Extérieur :



Conclusion :

L'organisation et la planification de ce projet ont été efficace et ont facilitées l'analyse et la conception malgré des difficultés rencontrées, notamment causé par les départs des anciens membres de nos projets respectifs.

Au travers de différents types de maquettes (modèle 3D, plugin test WordPress, etc.) et d'une analyse poussée sur le sujet, nous avons pu réunir tous les éléments nécessaires à la bonne réalisation de ce projet.

Cependant, nous avons imaginé une méthode, permettant de récupérer des plans et ainsi de faciliter la création des visites. Cette méthode était malheureusement incompatible avec notre projet et nous n'avons pas encore trouvé de solutions pour l'y incorporer.

Sources utilisées :

w3c.fr, developer.mozilla.org, wikipedia.org, khronos.org, adobe.com, journaldunet.com, developpez.net, namide.com, scriptol.fr, babylonjs.com, threejs.com, tomshardware.fr, unity3d.com, gijifo.com, angezanetti.com, sat.qc.ca, malabardesign.fr, blog.kaliop.com, aslacreations.com, cio.com, openclassrooms.com, ichemlabs.com, codeflow.org, html5gamedev.com, harvardarvr.club, msdn.microsoft.com, digitaltrends.com, zdnet.fr, developer.apple.com, macg.co, macbidouille.com, reddit.com, amd.com, stackoverflow.com, caniuse.com, afecreation.fr