

# **Documentation**

Travailler en mode projet

Analyser les objectifs et les modalités d'organisation d'un projet .....	3
Objectifs et modalités d'organisation .....	3
Analyse des besoins et de l'existant.....	3
Planifier les activités.....	5
Méthode agile .....	5
Méthode scrum .....	6
Évaluer les indicateurs de suivi d'un projet et analyser les écarts.....	9
Conformité du projet.....	9
Mise en conformité et sécurité .....	10
Rapport de mise en conformité .....	11
Documentation du projet.....	12
Communication écrite et/ou oral adaptée à l'interlocuteur.....	16

# Analyser les objectifs et les modalités d'organisation d'un projet

En vue du projet « OpenVia », une bibliothèque en ligne, un cahier des charges a été effectué, celui-ci répondant à plusieurs des critères qui vont venir.

## Objectifs et modalités d'organisation

### LE CONTEXTE/LA DEFINITION DU PROJET

La bibliothèque [OpenVia](#) est un modeste établissement d'une municipalité de village comptant 450 inscrits.

Le site actuel a été livré début janvier 2021 par un des stagiaires en BTS SIO de la bibliothèque. Il est hébergé sur un serveur gratuit et souffre de quelques défauts. Le stagiaire étant parti, la bibliothèque n'a pas les compétences en interne pour faire évoluer le site. Elle a donc fait appel à l'entreprise de prestation [Codicillus](#) pour mener à bien les évolutions souhaitées.

### LES OBJECTIFS DU PROJET

La bibliothèque souhaite un site plus accessible, moderne et fonctionnel. Afin de gagner du temps sur leur planning déjà très chargé et de mettre à jour la liste des livres disponibles plus rapidement, les bibliothécaires pourront gérer les données eux-mêmes directement sur le site afin de ne pas avoir à passer par un développeur. Pour éviter les retards de rendu de livre, les inscrits à la bibliothèque pourront se connecter à leur compte en ligne pour vérifier la date de retour et faire une prolongation si nécessaire.

La bibliothèque [OpenVia](#) souhaite mettre en ligne une nouvelle version du site pour la date anniversaire de l'établissement le mercredi 26 février 2021. Une campagne de publicité est déjà prévue pour cette date et a pour objectif de faire connaître la bibliothèque auprès des habitants du quartier. Afin de laisser à l'équipe L'environnement de recette devra être prêt pour le jeudi 20 février 2021 afin que nos testeurs aient le temps de remonter les potentielles anomalies.

## Analyse des besoins et de l'existant

### CADRER LA DEMANDE

Les grands axes d'amélioration du site :

- ☐ Un design responsive
- ☐ Augmenter le nombre de visiteurs simultanés autorisé
- ☐ Une interface d'administration pour que les bibliothécaires puissent gérer eux-mêmes les livres et en ajouter. De même pour l'ajout/la suppression/modification d'un nouvel inscrit dans la base.
- ☐ Création d'un compte en ligne pour chaque membre à l'aide de leur identifiant présent sur leur carte de bibliothèque.
- ☐ Mettre à disposition des inscrits la liste de leurs réservations en cours et un système de prolongation de 15 jours renouvelable 1 fois.

Les membres ne peuvent pas créer un numéro de carte en ligne, seuls les administrateurs peuvent faire cette manipulation.

## LES CONTRAINTES

### Les contraintes de délais:

- Le site internet doit être opérationnel pour le 26 février 2021.

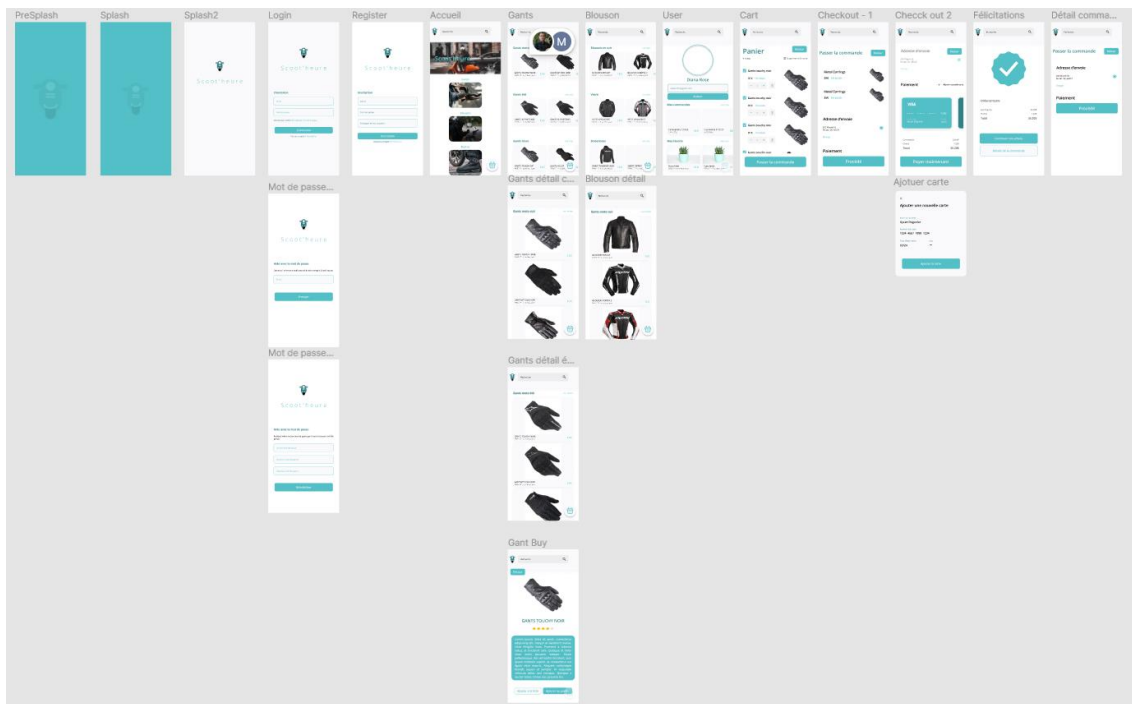
### Les contraintes techniques:

- Le site internet doit être accessible aux personnes malvoyantes ou aveugles.
- Le site internet doit pouvoir contenir une très grande quantité de livres et de comptes.

### Les contraintes logistiques:

- Le site internet doit pouvoir différencier un bibliothécaire voulant un compte admin, d'un intervenant ayant accès qu'au compte commun.

Pour le projet M2L, un Figma a été créé pour être sûr de répondre au besoin et d'avoir une idée claire du projet pour mieux se répartir les tâches :



# Planifier les activités

## Méthode agile

La méthodologie appliquée pour le projet OpenVia, fait par 3 personnes, était la méthode agile.  
Mais qu'est-ce donc ?

La méthode agile est une méthode de travail et un état d'esprit plaçant l'évolution du projet sur la satisfaction, **le besoin** du client.

Le projet est fait petits bouts par petits bouts afin d'être sûr que cela convienne au client et qu'il n'y ai pas de problème pour ensuite tout rassembler ce qui va nous donner le projet final, répondant au mieux à toute les exigences du client.

L'analyse du besoin étant fait, voici maintenant la liste des fonctionnalités importantes pour le projet.

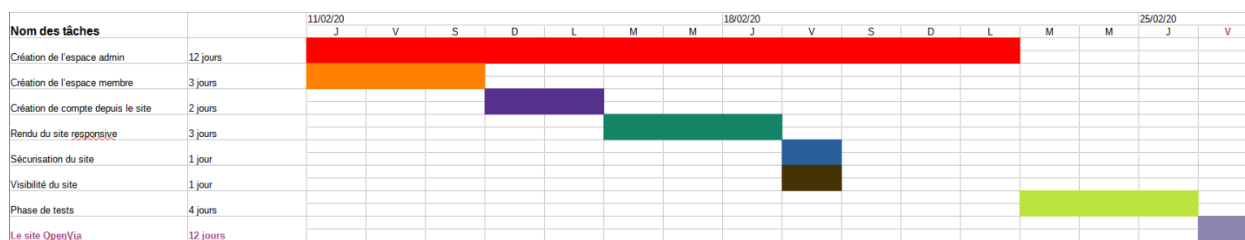
### LISTER ET DECRIRE LES FONCTIONNALITES

Liste les fonctionnalités par réalisation/produit.

Produit : ...

N° de la fonctionnalité	Version	Priorité	Description de la fonctionnalité
A1	Création d'un espace administrateur.	1	- Les bibliothécaires auront un compte leur permettant d'accéder à une interface*1 administrateur. Celui-ci confère tout les droits concernant le site.
A2	Création d'un espace membre.	1	- Les internautes*2 auront leur espace personnel leur permettant de connaître la date de rendu de livre ainsi que de prolonger cette durée de 15 jours une seule fois par emprunts.
A3	Création de comptes depuis le site.	2	- Les internautes auront la possibilité de créer leur propre compte comprenant le nom, le prénom, le mail, le mot de passe.
A4	Sécurisation du site.	3	- Les membres ne doivent pas avoir accès à l'espace admin ou bien à la base de donnée du site. Par précautions, le mot de passe de tout les comptes sera chiffrer*3.
A5	Rendu du site responsive*4.	3	- Le site devra être responsive.
A6	Visibilité du site.	4	- Le site devra être visible par les personnes aveugles ou malvoyantes.

La planification du projet.



Le coût du projet.

<b>PRIORITE</b>	<b>N°FONCTIONNALITE</b>	<b>RESSOURCES</b>	<b>NOMBRE DE JOURS</b>	<b>PRIX</b>
<b>1</b>	A1	2 développeurs	12 jours	6000 euros
<b>1</b>	A2	1 développeur 1 chef de projet	3 jours	1650 euros
<b>2</b>	A3	1 développeur	2 jours	500 euros
<b>3</b>	A4	1 développeur	1 jour	250 euros
<b>3</b>	A5	2 développeurs	3 jours	1500 euros
<b>4</b>	A6	1 développeur	1 jour	250 euros
<b>TOTAL SITE INTERNET (toutes options comprises)</b>				10150 euros
<b>TOTAL HEBERGEMENT*5</b>				15600 euros
<b>TOTAL À PAYER</b>				25750 euros

## Méthode scrum

La méthode utilisée pour un projet appeler « Bataille Navale », un jeu développé en Java en équipe de 6 personnes.

La méthode scrum est une extension de la méthode agile, c'est pourquoi on entend souvent parler de « scrum agile ».

Cette méthode est donc également orientée sur la satisfaction, le besoin du client.

Avec cette méthode aussi il faut faire les fonctionnalités petit à petit pour ensuite les regrouper pour le projet final.

Chaque personnes du projet a un quota de tâches à faire et une durée pour terminer celle-ci (une semaine ou deux par exemple), il faut donc commencer par les principales pour ne pas retarder les autres.

Dans notre cas, une liste des fonctionnalités nous a été donnée avec une couleur nous indiquant l'importance pour chacune d'elles.

Chaque fonctionnalité, après avoir été codée, est suivi d'un test fait par deux autres personnes du groupe et chacune accepté par le scrum master et donc le client.

Le scrum master étant le chef de projet et entretenant la liaison avec celui-ci.

Liste des fonctionnalités :

1

2

*getValue* renvoie la valeur de la matrice à la coordonnée en paramètre.  
**public int** getValue(**int** column, **int** line)

*initMat* initialise les cases de la matrice à la valeur 0.  
**private void** initMat()

*randomInit* initialise la grille avec des bateaux aux positions aléatoires. Fait appel à la méthode *isValidShip* avant de placer le bateau avec l'appel à la méthode *placeShip*.  
**public void** randomInit()

*addShot* met à jour la grille d'attaque avec la valeur 1 si c'est touché (success est vrai) ou avec la valeur 4 si c'est raté (success est faux) à la coordonnée en paramètre.  
**public void** addShot(**int** column, **int** line, **boolean** success)

*isValidShip* renvoie true si le bateau peut être inséré à cette coordonnée.  
**private boolean** isValidShip(**int** c, **int** l, **int** size, **int** dir)

*addNewShip* vérifie si le bateau peut être inséré avec *isValidShip* puis fait appel à *placeShip* pour l'insérer dans la matrice. « s » est la taille du bateau à insérer.  
**public boolean** addNewShip(**int** column, **int** line, **int** s, **int** d)

*placeShip* modifie la matrice d'entiers pour mettre la valeur 1 aux cases contenant le nouveau bateau  
**private void** placeShip (**int** c, **int** l, **int** size, **int** dir)

*toString* est une méthode très utilisée pour afficher un objet. La grille affiche sa matrice en colonnes et lignes comme ci-dessous.

	A	B	C	D	E	F	G	H	I	J
0	1	0	0	0	0	0	0	0	1	0
1	1	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	1	1	1	1	0
4	1	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	1	1
9	0	0	1	1	0	0	0	0	0	0

**public** String toString()

La classe Player

Les attributs

La classe Player contient le nom du joueur et deux objets Grid (l'un pour enregistrer l'emplacement de ses bateaux, l'autre pour enregistrer ses attaques).

Les méthodes

*addNewShip* fait appel à la méthode *addNewShip* de la grille d'enregistrement des bateaux du joueur et retourne ce qu'elle renvoie.

**public boolean** addNewShip(**int** column, **int** line, **int** s, **int** d)

*recordShot* fait appel à la méthode *getValue* de la grille d'enregistrement des attaques. Si l'attaque n'a pas déjà été faite à cette coordonnée-ci, vérifier que le joueur p a un bateau à cet endroit avec la méthode *hasShip*. Dans les deux cas, faire appel à la méthode *addShot* de la matrice d'attaque. Retourne true si l'attaque est un succès.

**public boolean** recordShot(**int** column, **int** line, Player p)

*hasShip* renvoie true si le joueur a un bateau à la coordonnée indiquée, false sinon.

**private boolean** hasShip(**int** c, **int** l)

*initGridRandom* initialise la grille du joueur avec des bateaux à des positions aléatoires (pour le mode 1 joueur) en appelant *randomInit* de la grille du joueur.

**public void** initGridRandom()

*displayGrid* fait appel à *toString* de la grille du joueur.

**public void** displayGrid()

*displayShotGrid* fait appel à *toString* de la grille d'attaque.

**public void** displayShotGrid()

*hasWin* renvoie true si le joueur a coulé tous les bateaux de l'autre.

**public boolean** hasWin()

3

4

*getName* renvoie le nom du joueur.  
**public** String getName()

La classe Game

Cette classe vous est fournie.

Les attributs

Contient les deux joueurs, le joueur courant et le mode de jeu.

Les méthodes

*fakelInitPlayerGrid* est une méthode permettant au développeur de ne pas avoir à saisir de nombreuses coordonnées pour pouvoir tester le programme. A utiliser en phase de développement uniquement.

**private static void** fakelInitPlayerGrid(Player p)

*initPlayerGrid* est la méthode de mise en place d'une grille pour un joueur humain.

**private static void** initPlayerGrid(Player p)

*launch2players* initialise le jeu au mode 2

**public static void** launch2players()

*launch1player* initialise le jeu au mode 1

**public static void** launch1player()

*initGame* est appelé par *launch2players* et *launch1player* pour initialiser le jeu. Cette méthode fait appel à *initPlayerGrid* ou *fakelInitPlayerGrid* pour chaque joueur. Dans le cas du mode 1 joueur, fait appel à *initGridRandom*.

**private static void** initGame()

*shot* fait jouer un joueur en lui demandant une coordonnée. Retourne true si touché, false si raté.

**public static boolean** shot()

*changeCurrentPlayer* est appelé par la méthode *nextPlayer* pour changer le joueur en fonction du mode de jeu.

**public static void** changeCurrentPlayer()

*isOver* renvoie true si un des deux joueurs a coulé tous les bateaux de l'autre. Fait appel à la méthode *hasWin* de chaque joueur.

**public static boolean** isOver()

La classe Battleship

Cette classe vous est fournie.

La classe contenant la méthode main. Fait appel à Game.

## AMELIORER LE JEU

### Coulé

Dans le jeu présenté à la partie précédente, la notion de bateau coulé n'est pas mise en place. A chaque attaque réussie, vérifiez que le bateau n'a pas été coulé. Il faut ajouter des méthodes et modifier *recordShot*.

### Intelligence artificielle

Robot niveau 0 : attaque aléatoirement le joueur adverse.

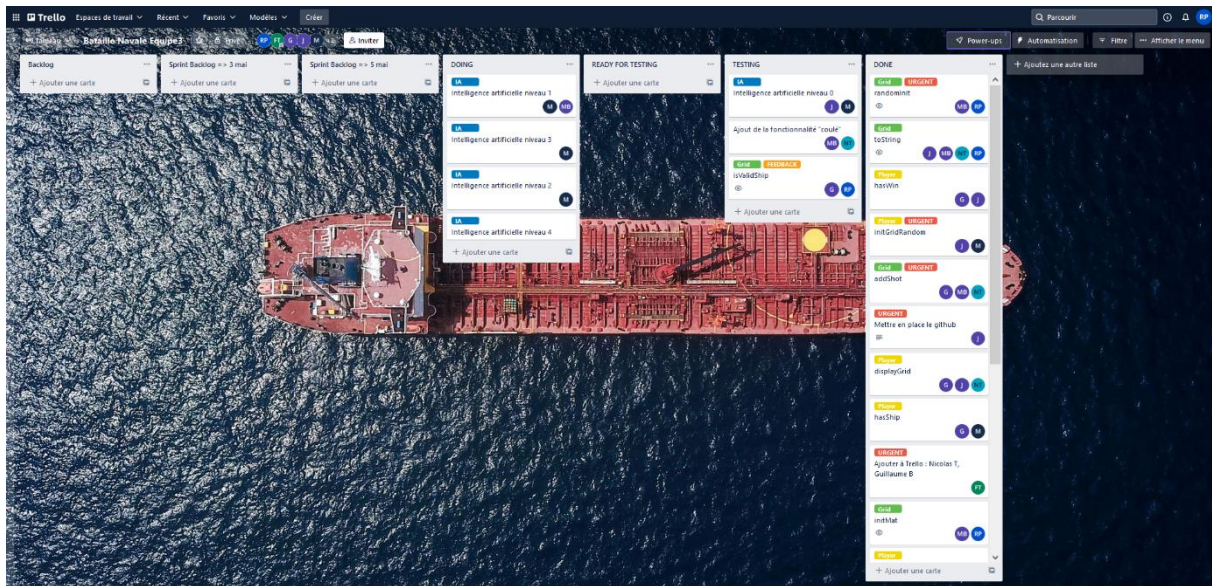
Robot niveau 1 : attaque aléatoirement le joueur adverse sur une coordonnée sur laquelle le joueur n'a pas encore attaqué.

Robot niveau 2 : attaque aléatoirement une case proche d'un bateau déjà touché (non coulé) et qui n'attaque pas les cases à proximité d'un bateau coulé.

Robot niveau 3 : améliore le niveau 2 par la détection de la direction d'un bateau touché. La case choisie est une des extrémités du bateau touché.

Robot niveau 4 : à vous de trouver des améliorations pour rendre le robot plus intelligent !

Pour organiser ce projet, nous avons donc utiliser Trello, une application de gestion de projet permettant entre autres de mettre des tags sur les tâches à faire (des mots clé pour mieux nous repérer) ainsi que d'assigner une tâche à une ou plusieurs personnes et y ajouter des commentaires.



Comme vous pouvez le voir, l'une des tâches ayant été faite et étant très importante dans un projet comme celui-ci... le github.

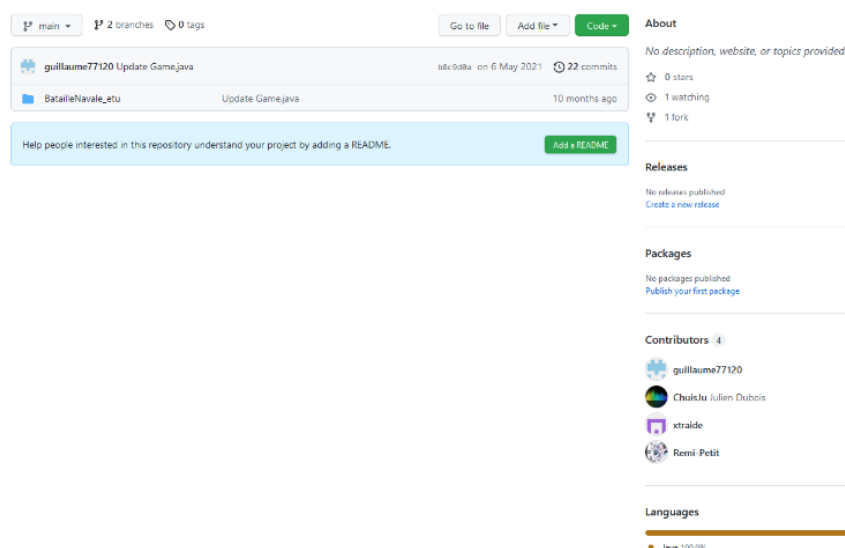
Github est une plateforme où les développeurs peuvent partager du code, travailler en mode projet. En effet, les projets sur github s'appellent les repositories et nous pouvons y ajouter nos camarades, le point fort étant la récupération du code qui est facilité mais surtout le regroupement du code de chaque personne y participant et ce manière simple et rapide, il suffit de 3 lignes de commandes (en partant du principe que notre projet est déjà lié au github).

**git add -A** (= pour inclure tout les fichiers de notre projet après s'être placé dans le bon dossier à l'aide de la commande « cd »..)

**git commit -m "message"** (= un commit étant une modification apportée au github et donc au projet, il est obligatoire de préciser pourquoi nous faisons tel ou tel action pour plus de clarté au sein du groupe.)

**git push origin master** (-> par exemple, pour apporter les modifications directement sur la branche principale du projet, master.)

Lien github : <https://github.com/ChuisJu/BatailleNavale>





## Évaluer les indicateurs de suivi d'un projet et analyser les écarts

### Conformité du projet

Certains projets utilisant par exemple des données à caractères personnelles (nom, prénom, âge, mail, numéro de téléphone, adresse, adresse ip etc...) sont soumis à des conditions à respecter par la CNIL (Commission nationale de l'informatique et des libertés) appelé RGPD (règlement général sur la protection des données).

Dans notre cas, le projet OpenVia utilise justement des informations personnelles et doit donc respecter ces lois.

Voici les liens utilisés pour répondre connaître la loi à appliquer :

<https://biblionumericus.fr/2018/05/23/quelques-pistes-pour-se-conformer-au-rgpd/>

[https://biblio.gironde.fr/images/stories/auteur\\_bdp/lisa/bao\\_rgpd\\_978f0.pdf](https://biblio.gironde.fr/images/stories/auteur_bdp/lisa/bao_rgpd_978f0.pdf)

[https://mediatheque.ille-et-vilaine.fr/images/template\\_rgaa/srcclient/MDIV/Boite\\_a\\_outils/Art3\\_RGPD.pdf](https://mediatheque.ille-et-vilaine.fr/images/template_rgaa/srcclient/MDIV/Boite_a_outils/Art3_RGPD.pdf)

<https://www.cnil.fr/sites/default/files/atoms/files/affiche-adoptez-les-6-bons-reflexes.pdf>

<https://www.cnil.fr/>

N°	Règle du RGPD	En quoi cette règle concerne la bibliothèque	Ce qui doit être mis en place pour y répondre	* : à mettre en place intégralement ** : en partie réalisé *** : déjà opérationnel
1	<i>Minimiser la collecte des données.</i>	<i>Le Règlement impose le principe de la minimisation des données collectées de tous les systèmes. Notre bibliothèque se sert des données collectées à des fins de gestion.</i>	<i>Établir la liste des données et leur finalité (faire un tableau récapitulatif). Supprimer la collecte des données non essentielles.</i>	*** <i>La bibliothèque ne demande pas d'informations qui ne sont pas essentielles à son fonctionnement.</i>
2	<i>Consentement libre et de clarté</i>	<i>Le Règlement impose que le formulaire d'inscription soit élaboré de telle sorte que l'utilisateur choisisse volontairement et librement. Notre bibliothèque se sert de formulaires d'inscriptions à des fins de gestion.</i>	<i>Établir la liste des données tout en respectant le consentement libre et le consentement de clarté.</i>	*** <i>La bibliothèque ne coche aucune case par défaut, elle laisse le choix aux utilisateurs de mettre leurs informations ils veulent en avoir un meilleur accès (empreint de livre par exemple).</i>
3	<i>RGS</i>	<i>Le Règlement impose que les données des utilisateurs soient correctement sécurisées.</i>	<i>Établir des protections dans tous les onglets où les utilisateurs peuvent insérer du texte ainsi que sur la BDD.</i>	** <i>La bibliothèque possède du htmlspecialchars ce qui sécurise l'accès à la BDD. Pour plus de sécurité, il n'est pas possible d'avoir un accès admin par l'URL mais elle ne crypte pas les données qui y sont, il faudrait améliorer ce point.</i>
4	<i>Doit à l'effacement et droit à la portabilité</i>	<i>Le Règlement impose que les utilisateurs aient accès à leurs données afin de pouvoir les modifier ou les supprimer.</i>	<i>Établir un moyen de récupération de données pour les utilisateurs ainsi que le droit de les supprimer.</i>	*** <i>La bibliothèque permet de récupérer le nom, prénom et le mail. Pour le mot de passe, l'utilisateur va devoir passer par un admin. Il est possible pour les utilisateurs de modifier leurs informations, s'ils veulent supprimer leur compte, il va falloir passer par un admin.</i>
5	<i>Norme simplifiée NS-009</i>	<i>Le Règlement impose que les utilisateurs doivent savoir à quelle fin va servir leurs données, à qui ils seront envoyés, une sécurisation des données ainsi qu'une durée de conservation des données limitée.</i>	<i>Établir un astérisque à côté de chaque collecte de données dans un formulaire afin de prévenir l'utilisateur de son utilité. Informer les utilisateurs de ceux qui vont recevoir ces données sans oublier de les protéger et de ne pas conserver ces informations au-delà d'un an après utilisation.</i>	** <i>La bibliothèque prend conscience des risques de sécurité bien qu'il soit encore possible de l'améliorer. Elle n'informe pas directement les utilisateurs de la façon dont sont traitées les données mais, un compte étant créé à l'aide d'un code donné par un bibliothécaire sur place, celui-ci pourra se charger de répondre aux questions. Les personnes recevant les informations sont inscrites dans le footer de la page. Aucune conservation de données n'est enregistrée dans un délai de plus d'un an en ce qui concerne les réservations. Une amélioration de la sécurité au niveau de la BDD par le biais d'un chiffrement est de mise.</i>

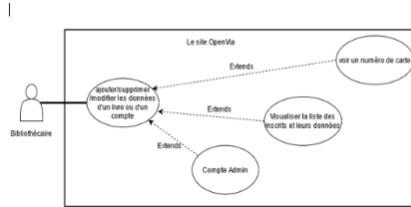
## Rapport de mise en conformité

N°	Rapport
1	<p><i>Liste des données collectées + finalité :</i></p> <ul style="list-style-type: none"> <li>- <i>Le prénom, il sert à savoir à quelle session appartient l'utilisateur, s'il est admin ou membre. Il peut également servir à savoir quelle personne veut envoyer un message dans la partie contact.</i></li> <li>- <i>Le nom, il a la même utilité que le prénom mais comme plusieurs peuvent avoir le même prénom, statistiquement c'est plus sûr sans compter qu'il est plus simple pour l'admin de retrouver la personne dans la base de données pour divers besoins. Il peut également servir à savoir quelle personne veut envoyer un message dans la partie contact.</i></li> <li>- <i>Le mail, il permet de contacter la personne si elle ne rend pas le livre à temps par exemple.</i></li> <li>- <i>Le mot de passe, sert à la sécurité de la cession de chacun.</i></li> <li>- <i>L'Identifiant client, sert également à la sécurité, n'importe qui n'a pas accès à un compte dans la bibliothèque. De plus, il sert à l'optimisation de la bibliothèque en ce qui concerne le prêt des livres par exemple.</i></li> <li>- <i>Le message est collecté afin de permettre à l'utilisateur d'envoyer un message aux personnes qui ont développé le site.</i></li> </ul>
2	<p><i>La bibliothèque est accessible sans avoir de comptes au préalable, donc sans avoir le besoin de fournir d'informations personnels.</i></p> <p><i>Pour réserver un livre, il sera en effet obligé de s'inscrire en tant que membre, au bon vouloir de l'utilisateur.</i></p> <p><i>En ce qui concerne la partie contact, rien n'oblige l'utilisateur à envoyer un message aux développeurs du site, il en a simplement l'occasion.</i></p>
3	<p><i>Le htmlspecialchars est actif à chaque input de la bibliothèque ce qui empêche les accès à la base de données depuis des commandes faites via le site.</i></p>
4	<p><i>Lorsque nous sommes connectés sur notre cession, en cliquant sur «votre espace», nous avons accès à nos données et pouvons les modifier.</i></p> <p><i>En ce qui concerne la suppression du compte, il faut un droit spécifique, celui d'admin. Il faut donc demander directement à un bibliothécaire de le supprimer, cette opération se fait sur l'interface admin, dans «Liste des comptes», il suffit de cliquer sur la corbeille qui se trouve à côté de compte que l'on souhaite supprimer.</i></p>
5	<p><i>Le prénom et le mail des développeurs du site se trouve dans le pied de page du site. Les données liants les membres à leur livre réservé sont supprimées lorsqu'ils rendent le livre. Un chiffrement des données de la base de données est nécessaire pour la sécurité des données personnels de chaque personne inscrite dans la bibliothèque.</i></p>

# Documentation du projet

## Documentation OpenVia :

Les diagrammes de cas d'utilisation :

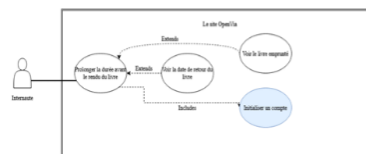


Cas d'utilisation : Prolonger la durée avant le rendu du livre

Acteur : Internaute

Données d'entrée : aucune

Le cas d'utilisation commence lorsque l'internaute clique sur «[Voir mon profil](#)»



Scénario alternatif :

- 1) Déclenchement du cas « [Se connecter](#) »
- 2) Fin du cas « [Se connecter](#) »

Cas d'utilisation : ajouter/modifier/supprimer un livre

- 1) Déclenchement du cas « [Se connecter](#) »
- 2) Fin du cas « [Se connecter](#) »

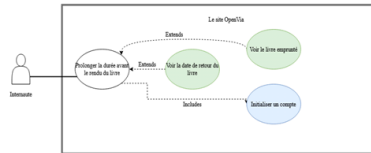
- 3a) Déclenchement du cas « ajouter un livre »
- 4a) Ajout des informations du livre et clique sur le bouton « envoyé »
- 5a) Le système ajoute le livre sur la base de [donnée](#).
- 6a) Fin du cas « ajouter un livre »
- 5b) Le système affiche un message « Vous devez mettre « information manquante » »
- 5b) Fin du cas « ajouter un livre »

- 3a) Déclenchement du cas « modifier un livre »
- 4a) Ajout des nouvelles informations du livre et clique sur le bouton « envoyé »
- 5a) Le système modifie les informations modifier du livre sur la base de [donnée](#).
- 6a) Fin du cas « modifier un livre »

- 3a) Déclenchement du cas « supprimer un livre »
- 4a) ajouter le titre du livre et clique sur le bouton « envoyé »
- 5a) Le système supprime le livre de la base de [donnée](#).
- 6a) Fin du cas « supprimer un livre »

Cas d'utilisation : ajouter un genre

- 3b) Déclenchement du cas « ajouter un genre »
- 4a) Ajout du genre et clique sur le bouton « envoyé »
- 5a) Le système ajoute le genre sur la base de [donnée](#).
- 6a) Fin du cas « ajouter un genre »
- 5b) Le système affiche un message « Vous devez mettre un genre »
- 5b) Fin du cas « ajouter un genre »



3a) Déclenchement du cas « [Voir le livre emprunté](#) »

4a) Fin du cas « [Voir le livre emprunté](#) »

3b) Déclenchement du cas « [Voir le livre emprunté](#) »

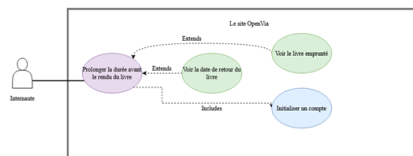
Le système affiche le livre emprunté

3b) Fin du cas « [Voir le livre emprunté](#) »

4b) Déclenchement du cas « [Voir la date de retour du livre](#) »

4c) Le système affiche affichera la date en rouge pour avertir l'internaute du retard.

4c) Le système affiche le nombre de temps qu'il reste et la possibilité de le rallonger



5) Déclenchement du cas « [Prolonger la durée avant le rendu du livre](#) »

6) Le système affiche que la demande a bien été prise en compte, le temps avant rendu du livre est prolongé de 15 jours.

5d) Déclenchement du cas « [Prolonger la durée avant le rendu du livre](#) »

6d) Le système affiche un message expliquant que prolonger la durée n'est possible qu'une seule fois.

7) Fin du cas « [Prolonger la durée avant le rendu du livre](#) »

8) Déclenchement du cas « [Se déconnecter](#) »

9) Fin du cas « [Se déconnecter](#) »

Cas d'utilisation : ajouter un [éditeur](#).

3b) Déclenchement du cas « ajouter un [éditeur](#) »

4a) Ajout du [éditeur](#) et clique sur le bouton « envoyé »

5a) Le système ajoute le [éditeur](#) sur la base de [donnée](#).

6a) Fin du cas « ajouter un [éditeur](#) »

5b) Le système affiche un message « Vous devez mettre un [éditeur](#) »

5b) Fin du cas « ajouter un [éditeur](#) »

Cas d'utilisation : ajouter une langue

3b) Déclenchement du cas « ajouter une langue »

4a) Ajout de la langue et clique sur le bouton « envoyé »

5a) Le système ajoute la langue sur la base de [donnée](#).

6a) Fin du cas « ajouter une langue »

5b) Le système affiche un message « Vous devez mettre une langue »

5b) Fin du cas « ajouter une langue »

Cas d'utilisation : Liste des réservations (Interface Admin)

3b) Déclenchement du cas « liste des réservations »

4a) Le système affiche la liste des livres réservés par tous.

5a) Fin du cas « liste des réservations »

4b) Le système affiche un message « Il n'y a aucun livre de réservé »

5b) Fin du cas « liste des réservations »

Cas d'utilisation : Ajouter un compte

3b) Déclenchement du cas « Ajouter un compte »

4a) Ajout de toute les informations demandée concernant l'utilisateur et clique sur « Envoyer ».

5a) Fin du cas « liste des réservations »

4b) Le système affiche le message « Vous devez mettre « information manquante » »

5b) Fin du cas « liste des réservations »

**Cas d'utilisation :** Modifier un compte

3b) Déclenchement du cas « Modifier un compte »

4a) Changement des informations souhaitées concernant l'utilisateur et clique sur « Envoyer ».

5a) Le système change les informations de l'utilisateur dans la base de donnée.

6a) Fin du cas « liste des réservations »

**Cas d'utilisation :** Supprimer un compte

3b) Déclenchement du cas « Supprimer un compte »

4a) Le bibliothécaire met les informations de l'utilisateur concerné et clique sur « Envoyer ».

5a) Le système supprime les informations de l'utilisateur de la base de donnée.

6a) Fin du cas « liste des réservations »

4b) Le système affiche le message « Cette utilisateur n'est pas enregistré dans la base de donnée. »

5b) Fin du cas « liste des réservations »

**Cas d'utilisation :** Liste des réservations (Interface Internaute)

3b) Déclenchement du cas « liste des réservations »

4a) Le système affiche la liste des livres qu'il a réservé.

5a) Le système affiche la possibilité de renouveler la durée de rendu du livre une fois par livres.

6a) Fin du cas « liste des réservations »

6b) L'internaute sélectionne un livre et clique sur le bouton « envoyer »

7b) Le système ajoute 15 jours supplémentaires concernant le rendu du livre sélectionner.

8b) Le système affiche la nouvelle date de rendu du livre

9b) Fin du cas « liste des réservations »

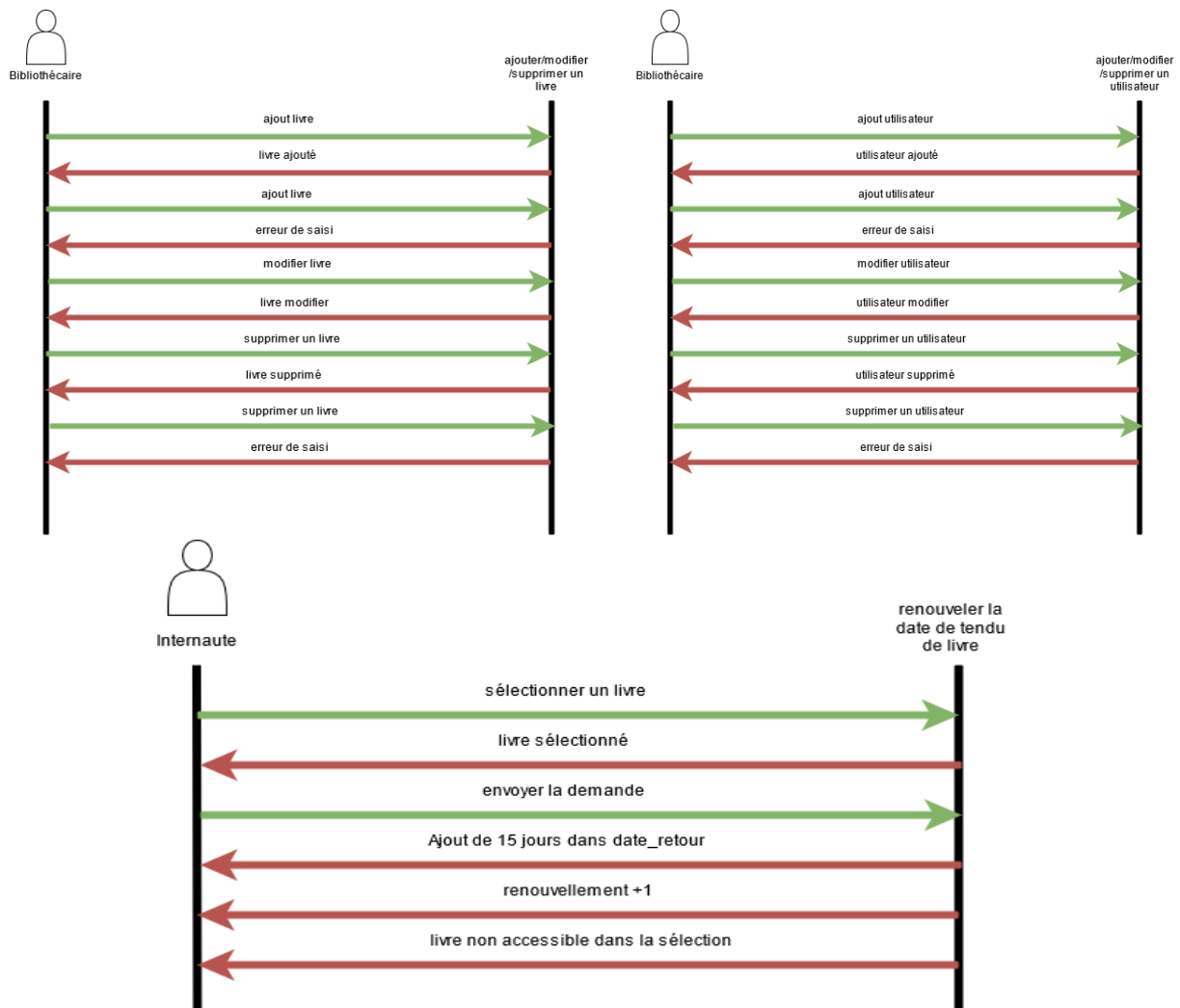
6c) L'internaute ne peut pas sélectionner un livre déjà renouveler.

7c) Fin du cas « liste des réservations »

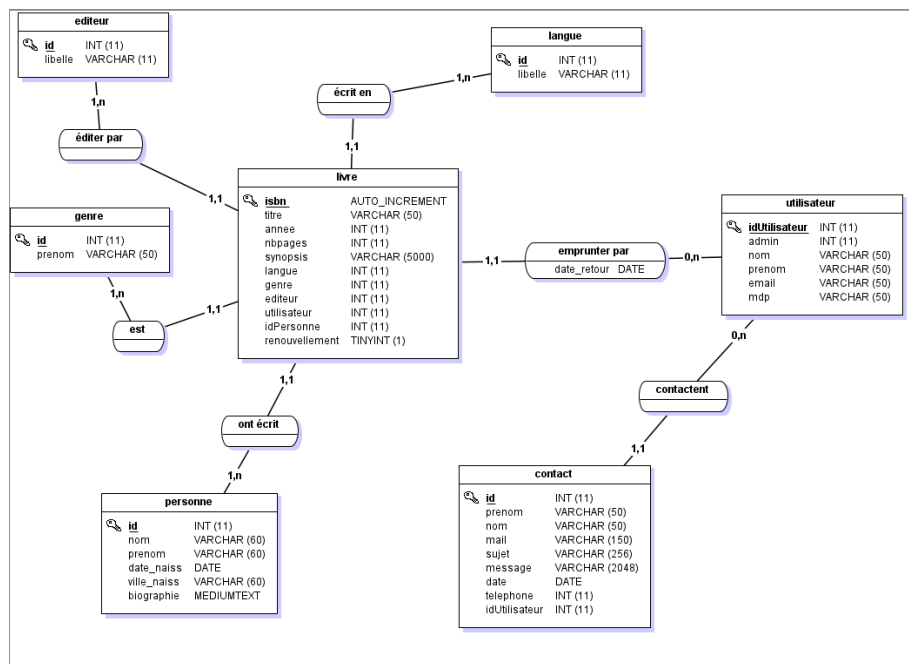
5d) Le système affiche un message « Il n'y a aucun livre de réservé »

5d) Fin du cas « liste des réservations »

## Diagramme de séquence :



## Schéma entité/association :



Sans oublier le manuel d'installation :

Pour installer le site OpenVia sur votre machine veuillez suivre cette procédure

### **Etape 1 : Déplacement du dossier du site vers le serveur**

- Déplacer le dossier nommé vers votre serveur.
- Le chemin vers le serveur est C:\wamp64\www\ et mettez le dossier dans www.

### **Etape 2 : Installation de la base de donnée**

- Ouvrez WampServer
- Cliquez sur le logo en bas à droite
- Cliquez sur une fenêtre internet s'ouvre.
- Inscrivez-y en Utilisateur : root et en Mot de passe :
- Créé une base de donnée, nommez la comme bon vous semble.
- Ensuite cliquez sur Importer en haut à droite.
- Dans la section **Fichier à importer** : cliquez sur le bouton **Choisir un fichier**
- Rendez-vous dans le chemin C:\wamp64\www\bib\SQL et ouvrez le fichier **biblio.sql**
- Dans la section **Options spécifiques au format** : puis cliquez sur

### **Etape 3 : Modification du chemin absolu du site**

- Rendez-vous vers le chemin C:\wamp64\www\bib\bibliotheque\header
- Ouvrez le fichier header.php
- A la ligne 3, `$CONFIG = array("root_path" => "/bibliotheque");` modifier le `/bibliotheque`
- Ecrivez-y le chemin absolu à partir de la racine www/

### **Etape 4 : Modification connexion à la bdd**

- Rendez-vous vers le chemin C:\wamp64\www\bib\bibliotheque\bdd
- Ouvrez le fichier config.php
- A la ligne 3 `dbname=biblio;charset=UTF8,'root',''`
- Remplacez **biblio** par le nom que vous avez mis à votre base de donnée
- Puis **root** par votre utilisateur pour se connecter à phpmyadmin
- Entre les apostrophes juste après le **root** mettez votre mot de passe si vous en avez un.

### **Etape 5 : Ouvrir le site**

- Lancez votre navigateur
- Dans l'URL inscrivez-y localhost/CheminAbsoluDuSite

Votre site est maintenant opérationnel !

Il en va de même pour le projet RespireStat également, où toutes les informations sont sur le portfolio, à cette URL :

<https://www.remipetit.fr/pages/informations.php?nom=RespireStat>

Ou bien encore le projet TodoList, à cette URL ci :

<https://www.remipetit.fr/pages/informations.php?nom=TodoList>

## Communication écrite et/ou oral adaptée à l'interlocuteur

Le cahier des charges comporte un glossaire avec la définition de l'ensemble des mots technique utilisés pour le projet OpenVia.

### GLOSSAIRE

*Définition des termes techniques, complexes, spécifiques que vous avez utilisé dans ce document.*

*Interface\*1 : Une interface est ce qui permet de relier le système de façon simplifier pour ses utilisateurs. Les bibliothécaires pourront donc faire toutes les actions possibles sur le site à travers cette interface.*

*internautes\*2 : Internaute désigne une personne qui utilise Internet, quelqu'un qui va sur le web, que ce soit de manière régulière ou occasionnelle.*

*Chiffrer\*3 : Opération qui consiste à transformer un message à transmettre, dit «message clair», en un autre message, inintelligible pour un tiers, dit «message chiffré», en vue d'assurer le secret de sa transmission.*

*Le mot de passe «motdepasse» sera alors écrits sur la base de donnée comme ceci par exemple :*

b6edd10559b20cb0a3ddaeb15e5267cc

*Responsive\*4 : c'est lorsque le site s'adapte à la taille de l'écran sur le quel il est utilisé (ordinateur, tablette etc..).*

*Hebergement\*5 : Votre site a besoin d'un hébergeur, le rôle de celui ci est de stocké votre site et par la même occasion, mettre votre site sur Internet.*

L'ensemble des conversations avec la cliente, Flavie Tonon, étant passée à travers l'application Teams de Microsoft, permettant ainsi l'envoi de messages, d'images... et la possibilité de faire des appels où nous pouvons par exemple faire un partage de notre écran, ce qui est pratique pour avoir l'avis du client et lui montrer l'avancée du projet.

