

Recensement des tâches : Projet BOOKING <small>by Simpion</small>						
N°	Tâches	Durée jours	Tâches requises	Objectifs	Critères d'achèvement	Livrables
Phase 0.5 : Mise en place du projet et de sa gestion	1 Création du dépôt github	0,25	X	Création d'un dépôt sur github pour regrouper le travail du projet	Github opérationnel et prêt a accueillir les données du projet	Dépôt Github en ligne
	2 Installation des outils collaboratifs	0,5	X	Installation sur nos postes de travail de GitKraken pour synchronisation du code avec l'équipe	Gitkraken est installé sur tous les postes de l'équipe	Application installer sur nos postes
	3 Création du kanban	0,5	X	Création du board Kanban pour organisé le travail de l'équipe de developpement	Tableau fonctionnel et prêt à être utiliser pour la gestion du flux de travail	Board initialiser dans notre application GitKraken
Phase 1 : Création de la base de données	4 Etablissement du dictionnaire de données	0,5	X	Etablir un dictionnaire de données pour recenser les données utile	Les données sont recensé dans un dictionnaire de données	Fichier excel contenant le dictionnaire de données
	5 Modèle conceptuel des données	0,5	4	Définition des entité ainsi que leurs relation avec leur cardinalités	Toutes entités sont crée ainsi que leurs relation avec les cardinalité et leur nom de colonne	Fichier Drawio et .png affichant le modèle
	6 Modèle logique des données	0,5	5	Modélisation de la structure de la base de donnée	Les dépendances sont fonctionnelles, définition des identifiant, formes normalisé	Fichier Drawio et .png affichant le modèle
	7 Modèle relationel des données	0,25	6	Formaliser la struture de stockage des données, formaliser la manipulation des données	Première, deuxième et troisième forme normale avec établissement des clé primaire et clé étrangère	Fichier .PDF contenant le modèle relationel
	8 Modèle physique	1	7	Retranscrire le modèle logique en langage SQL	Possibilité d'implémenter des données	Code SQL de la base de données
	9 Création des requête CRUD	5	8	Créer, lire, mettre a jour et supprimé les donnée de la base de donnée	Test unitaire fonctionnel	Procédure stockée format .SQL
Phase 2 : Création du programme python (POO)	10 Création de la classe reservation (python)	1,5	X	Une classe reservation qui contient les données relative à une demande de réservation : le nom de la salle, la date de réservation, et l'heure souhaité. Cette class sert au transport de l'information entre les différentes partie du programme.	Classe reservation prête à accueillir les données des mails	Fichier python (programme) contenant la classe
	11 Création de la classe interface_mail (python)	1,5	11	Une classe interface_mail qui est capable de lire les mails (des fichiers texte) et de stocker les informations dans un objet de type reservation.	Classe interface_mail capable de lire les mails et de stocker les informations dans un objet réservation	Fichier python (programme) contenant la classe
	12 Création classe DAO (python)	1,5	12	Une classe dao (Data Access Object) qui gère la connexion à une base de données et l'envoi des requêtes.	Classe capable de gérer la connexion entre une base de données et l'envoi des requêtes	Fichier python (programme) contenant la classe
	13 Création de la classe moteur (python)	1,5	13	Une classe moteur qui fait le lien entre toutes les autres : elle récupère les données reçus par interface_mail, demande au dao les informations lui permettant lui permettant de vérifier si les demandes sont compatibles avec les réservations précédentes et si c'est le cas met à jour la base de données.	Classe capable de vérifier et de traiter les demande de réservations lui permettant de vérifier si les demandes sont compatibles avec les précédentes réservations et si c'est le cas met à jour la base de données.	Fichier python (programme) contenant la classe
Phase 3 : Création de l'application FLASK	14 Auto-formation au framework Flask HTML & CSS	1	X	S'auto former sur les outils nécessaire à la réalisation du projet	Trouver des pistes d'ouvertures cohérentes (vidéo, tutos, stackoverflow) pour commencer l'application FLASK	Connaissance acquise et fichier .py de suivi d'un tutoriel
	15 Recherche de templates	0,25	X	Recherche de template pour la création de la partie HTLM de l'application FLASK	Trouver un template cohérent avec un logiciel de réservation sur internet	Templates avec fichier HTML et CSS libre de droit
	16 Définir les routes de l'application flask	0,5	X	Définir les différentes entrées possible sur l'application	Les différentes routes sont nommées et prêtes à être créer	Schema de l'application .xls
	17 Définir les champs formulaire login	0,5	16	Définir le formulaire de connection à l'application	Les différents champs (email/mot de passe) du formulaire login sont prêts a être inséré sur la page HTML adéquate	Fichier HTML avec formulaire de connexion
	18 Définir les champs formulaire register	0,5	16	Définir le formulaire pour créer un compte sur l'application	Les différents champs (email/prénom/nom/poste/organisation/mot de passe) du formulaire register sont prêts à être inséré sur la page HTML adéquate	Fichier HTML avec formulaire d'enregistrement
	19 Création de la page accueil " /home "	2	16	Création de la page d'accueil	La page d'accueil est fonctionnelle	Fichier HTML&CSS fonctionnant avec la route Flask
	20 Création de la page login " /login "	2	17	Création de la page de connexion	La page de connexion est fonctionnelle et permet de se connecter sur un compte crée	Fichier HTML&CSS fonctionnant avec la route Flask
	21 Création de la page register " /register "	2	18	Création de la page d'enregistrement	La page d'enregistrement est fonctionnelle et permet de créer un compte sur l'application	Fichier HTML&CSS fonctionnant avec la route Flask
	22 Création de la page accueil login " /login/home"	1	19	Création de la page d'accueil de l'utilisateur	La page d'accueil de l'utilisateur est fonctionnelle après s'être connecter avec son compte	Fichier HTML&CSS fonctionnant avec la route Flask
	23 Création de la page profile page " /login/profile"	1	20	Création de la page de profil de l'utilisateur	La page de profil de l'utilisateur est fonctionnelle et lui permet de consulter ses informations personnelles	Fichier HTML&CSS fonctionnant avec la route Flask
	24 Création de la page réservation " /login/reservation "	2	20	Création de la page de demande de réservation de l'utilisateur	La page de demande de réservation est fonctionnelle, elle permet de prendre une réservation sous réserve d'être connecté avec un compte sur l'application	Fichier HTML&CSS fonctionnant avec la route Flask
	25 Ajout des redirections des routes entres elles	1	24	Faire communiquer les pages entre par les hyperliens ainsi que les restrictions des routes	Le site est fonctionnel et toute les pages interagissent entre elles et prennent en compte les différentes restrictions imposées	Fichier HTML&CSS fonctionnant avec les routes Flask qui interagisse entres elles
	26 Création de la page de consultation de réservation	1	25	Création de la page pour que l'utilisateur consulte ses réservation	Un compte enregistrer doit pouvoir consulter ses demande de réservation sur la page de consultation de ses réservation	Fichier HTML&CSS fonctionnant avec la route Flask
	Définir les données pour les graphiques utilisé pour la datavisualtion de la page utilisateur	1	24	Définir les données à mettre en avant dans un tableau de bord ou graphique	Toutes les données jugées pertinentes sont prêtes à être utilisé pour la datavisualisation	Fichier .ipynb avec dashboard
	Création de la page utilisateur contenant la dataviz " /login/dashboard"	2	27	Création de la page de tableau de bord de booking pour l'utisateur	Une page dataviz est disponible pour l'utilisateur avec différents graphiques et statistiques utiles concernant ses réservations	Fichier HTML&CSS fonctionnant avec la route Flask