

Mission MV

présentation générale :

Les visiteurs médicaux de GSB ont besoin d'un suivi pour leurs visites.

Contexte :

Les médecins sont vus dans des cabinets médicaux. Un cabinet médical correspond à l'endroit où est vu le médecin, c'est une adresse postale. On considère qu'un médecin n'intervient que dans un seul cabinet. Plusieurs médecins peuvent travailler dans un même cabinet.

Des médecins sont affectés à chaque visiteur. Un médecin n'est vu que par un seul visiteur.

Une personne du service (site de Paris) sera chargée de mettre à disposition les informations concernant les cabinets et les médecins.

Chaque visite effectuée doit être mémorisée :

- la date de la visite
- est-ce qu'elle était sur rendez-vous
- l'heure d'arrivée au cabinet
- l'heure de début d'entretien avec le médecin
- l'heure de départ du cabinet

Un médecin peut être visité plusieurs fois à des dates différentes.

Ces informations doivent pouvoir être saisies au fur et à mesure de la journée par le visiteur à partir de son smartphone ou sa tablette.

Le téléchargement de l'application sera disponible à partir d'un site web en ligne accessible grâce à une double identification.

~~Lors de la l'installation sur l'appareil mobile, le visiteur s'authentifie.~~

~~L'identifiant du visiteur qui installe l'application doit être mémorisé dans un fichier sur le téléphone.~~

~~Les échange entre le mobile et la base de données ne se fera que via l'utilisation d'API sécurisées (utilisation de token).~~

~~L'application mobile de suivi de l'activité du visiteur doit permettre de saisir les visites du visiteur.~~

L'application doit proposer au visiteur sa liste des médecins (et des cabinets) à visiter pour le

Le visiteur peut choisir le cabinet dans la liste des cabinets qui le concerne mais le cabinet proposé par défaut dans la liste est celui le plus proche par rapport à position gps.

L'application mobile doit permettre :

- ~~- de voir une page d'accueil avec un logo, un texte et 2 boutons (voir les visites / ajouter une visite)~~
- ~~- de visualiser les visites~~
- ~~- d'ajouter une visite~~

La création de la base de données, des classes et des API doivent être faites par équipe.
La création de l'application mobile doit être faite individuellement.

Document de suivi de projet à fournir **pour chaque sprint** :

	État - date	Nom étudiant 1		Nom étudiant 2				Nom étudiant 3			
SPRINT		Tps prev	Tps réel								
Us + tâche											

État : à faire / en cours (date/heure début) / fait (date/heure de fin) / validé

contraintes :

chaque étudiant doit faire au moins un sprint d'API

pour les autres sprints , chaque sprint doit être traité par les 3 étudiants

n°		prédécesseurs	deadLine
01	Sprint TourLourdV0	-	15 mars minuit
02	Sprint TourLourdV1	01	15 mars minuit
03	Sprint TourLourdV2	02 / 05 / 06 / 07	
04	Sprint TourLourdV3	02 / 05 / 06 / 07	
05	Sprint APITourLourd_VISITEUR	12	15 mars minuit
06	Sprint APITourLourd_MEDECIN	12	15 mars minuit
07	Sprint APITourLourd_VISITE	12	15 mars minuit
08	Sprint MyTourV1	-	15 mars minuit
09	Sprint MyTourV2	05 / 06 / 07	
10	Sprint MyTourV3	11 (08)	
11	Sprint APIMyTour	12	
12	Sprint BDD	-	15 mars minuit

Pour la deadLine :

- suivi de projet
- contenu du sprint
- gitHub à jour

Sprint V0 (aucun accès à la base de données)

tâche : concevoir classe abstraite Personne

tâche : concevoir classe Visiteur (mettre en place la double navigabilité entre Visiteur et Visite)

tâche : concevoir classe Médecin

tâche : concevoir classe Cabinet

tâche : concevoir classe Visite

tâche : documentation technique (diagramme de classe + extraction de documentation de code)

tâche : livraison

Sprint BDD

tâche : concevoir la BDD

tâche : validation BDD

tâche : mise en ligne du script de la structure

tâche : jeu d'essai

tâche validation JEU ESSAI

tâche : mise en ligne du script jeux de données

tâche : documentation technique

tâche : livraison

Sprint DAOVISITEUR

tâche : concevoir get tous les visiteurs (GET) + script de test.
tâche : get un visiteur en fonction id + script de test.
tâche : ajouter un visiteur (POST) + script de test.
tâche : modifier un visiteur (PATCH) + script de test.
tâche : supprimer un visiteur (DELETE) + script de test.
Tâche : authentification (mail + mdp) + script de test
tâche : documentation technique
Tâche : livraison/validation visiteur

Sprint APIVISITEUR

tâche : concevoir API visiteur (GET) + script de test.
tâche : modifier API visiteur (GET , un visiteur en fonction id) + script de test.
tâche : modifier API visiteur (POST) + script de test.
tâche : modifier API visiteur (PATCH) + script de test.
tâche : modifier API visiteur (DELETE) + script de test.
tâche : modifier API visiteur AUTHENTIFICATION + script de test.
tâche : documentation technique
Tâche : livraison/validation API visiteur

Sprint DAOVISITE

tâche : concevoir get toutes les visites (GET) + script de test.
tâche : get une visite en fonction id + script de test.
tâche : ajouter une visite (POST) + script de test.
tâche : modifier une visite (PATCH) + script de test.
tâche : supprimer une visite (DELETE) + script de test.
tâche : documentation technique
Tâche : livraison/validation visite

Sprint APIVISITE

tâche : concevoir API visite (GET, toutes les visites) + script de test
tâche : modifier API visite (GET, les visites en fonction id visiteur) + script de test
tâche : concevoir API visite (POST, infos visites) + script de test
tâche : modifier API visite (PUT) + script de test.
tâche : concevoir API visite (DELETE) + script de test.
tâche : documentation technique
Tâche : livraison/validation API visite

idem pour CABINET et MEDECIN

Sprint MOBILE v1

tâche : se former à la conception d'interfaces graphiques

User story : En tant que visiteur, je veux pouvoir visualiser mes visites dans mon appli

tâche : Conception interface graphique visualiser mes visites (avec jeu d'essai de visites sans accès bdd)

User story : En tant que visiteur, je veux pouvoir créer une nouvelle visite dans mon appli

tâche : Conception interface graphique créer une visite (avec bouton de validation mais sans ajout dans BDD)

User story : En tant que visiteur, je veux pouvoir m'authentifier sur mon appli mobile

tâche : Conception interface graphique d'authentification

tâche : coder appel page accueil ou pas (sans accès bdd)

tâche : mettre en place authentification qu'à l'installation

User story : En tant que visiteur, je veux accéder à la page d'accueil de l'application

tâche : Conception interface graphique page d'accueil

tâche : Coder appel pages voir et créer à partir de la page d'accueil

tâche : livraison / validation client

Sprint MOBILE API(intégration API => nouvelle branche sur gitHub)

tâche : se former à l'appel d'une API GET (tous) dev mobile

tâche : se former à l'appel d'une API GET(un) dev mobile

tâche : se former à l'appel d'une API POST dev mobile

User story : En tant que visiteur, je veux pouvoir visualiser mes visites dans mon appli

tâche : Coder l'appel de l'api qui renvoie toutes les visites

tâche : Coder l'appel de l'api qui renvoie les visites d'un visiteur (en dur)

User story : En tant que visiteur, je veux pouvoir me créer une nouvelle visite

tâche : Coder l'appel de l'api qui crée une visite dans la base de données pour un visiteur (en dur)

tâche : livraison / validation client

Sprint MOBILE AUTHENTIF

tâche : se former à l'appel d'une API authentification avec gestion du token dev mobile

User story : En tant que visiteur, je veux pouvoir me connecter sur mon appli mobile

tâche : Coder l'appel de l'api qui teste le mail et le mdp du visiteur

User story : En tant que visiteur, je veux pouvoir visualiser mes visites dans mon appli

tâche : Coder l'appel de l'api qui renvoie les visites du visiteur authentifié

User story : En tant que visiteur, je veux pouvoir me créer une nouvelle visite dans mon

tâche : Coder l'appel de l'api qui crée une visite dans la base de données pour le visiteur authentifié

tâche : livraison / validation client