1er rendez-vous

Découverte des outils existants :
Un outil qui traite des demandes de missions, saisie d'informations et l'action passe par plusieurs changements d'étapes.
On peut suivre les missions de recrutement et de stagiaires.
Le template est unifié pour toutes les procédures, on va utiliser codeigniter4 (framework PHP)

1er outil :

Connexion classique
Question de droits d'accès (en fonction du travail au sein du LIG)
Tableau de bord comportant des missions
Les missions ont un état, les différents états comportent un code couleur qui permet de trier en fonction de l'état
On peut faire une demande de mission et ajouter des documents.

2ème outil :
Suivi de recrutement, principalement la même interface

Ajouts éventuels pour le nouvel outil :
Ajouter du tri
Modéliser les procédures
→ L'interface doit rester similaire
Mettre les tableaux sous forme de colonnes et pouvoir trier selon les différents composants
Revoir la fonctionnalité recherche (elle ne marche pas très bien)
Ajout d'une fonctionnalité copier

2ème rendez-vous

Découverte des cahiers des charges pour l'OSS et l'OSR, discussions autour de la bonne compréhension des termes.
1 outil pour regrouper les 2 ? → Ne semble pas complet, on va privilégier de garder les 2 outils.
Discussions autour des données, ce qui va être gardé ou pas, → on en reparlera lors de l'importation des données vers le nouvel outil.
Pour la prochaine fois : détailler les processus (Diagramme de use-case)
Force de proposition au niveau de l'aspect visuel du site, invitation sur GitLab pour nous partager et nous donner accès à des fichiers.

3ème rendez-vous

On a apporté les diagrammes de séquence

Modifications OSR :
C'est soit le responsable de contrat soit le responsable d'équipe qui est notifié en fonction de s'il y a un contrat support ou non
L'assistante d'équipe = la secrétaire de l'équipe (ce sont les mêmes personnes)
Contrat créé à déplacer plutôt vers la cellule RH
C'est la cellule RH qui enregistre les données dans l'OSR puis c'est l'OSR qui envoie les informations dans la base de données/aux stagiaires
Saisir le coût à placer après demande finalisée

Modifications OSS :
A propos du bureau :
S'il y a une demande de bureau commun, on va regarder dans une autre liste s'il y a une disponibilité et elle lui est affectée automatiquement. S'il n'y a pas de bureau commun disponible, la demande passe en attente et c'est à ce moment là qu'on va solliciter le chargé des locaux qui va devoir lui même trouver un bureau à affecter (à priori on ne s'en préoccupe plus)
Le mail est envoyé au RH uniquement quand la demande est initialisée.
La cellule RH n'envoie pas les documents au stagiaire, elle lui envoie de quoi les remplir.
La convention de stage arrive de l'extérieur (UGA) et est envoyée au futur stagiaire.
La RH finalise → envoi d'un mail au stagiaire qui doit fournir les informations
Préciser "du personnel" pour la BDD.
Envoi d'un mail au recruteur uniquement s'il n'est pas déjà chef d'équipe (sinon il va recevoir le mail 2x)
Elle nous a donné la liste des informations sur la base de données, on ne va pas forcément faire des tables pour tout mais plutôt garder en mémoire parce que les informations fluctuent beaucoup.
Pour les nationalités, les groupes, les unités à priori on utilisera des tables.
Proposition : pour que ça soit intuitif, mettre qui doit faire qui et à quel moment au niveau de la visibilité de la demande pour le recruteur pour qu'il puisse savoir qui contacter en cas de besoin ?

4ème rendez-vous

On a complété les diagrammes de séquence, on a fait les use case

Lors de la modification de la demande, il n'y a pas de validation, elle reste à l'état prise en charge. (On ne gère pas ça dans l'OSR)
Prévoir un mail d'avertissement en cas de modification ou poser la question à la personne qui modifie s'il veut envoyer un mail de notification au stagiaire et / ou au recruteur ou non. Si il veut renvoyer un mail, renvoyer le nouveau pdf contenant les détails du stage aux deux.
Pas d'état validé
Remplacer enregistrement des données par envoi des données vers SILOSE
Si le recruteur n'est pas chef d'équipe on envoi un mail de NOTIFICATION pas de CONFIRMATION
Si c'est un avenant on ne redemande pas les pièces

OSR :
Soit c'est le chef de contrat qui accepte/refuse s'il y en a un, soit c'est le chef d'équipe
Envoi du contrat ne part pas de l'OSR mais du Recruté
La cellule RH n'entre pas de données, l'OSR envoie tout seul une fois l'action retour tutelle effectuée

Détails sur la BDD : Quand on insère, on contrôle les doublons sur nom-prenom-date_de_naissance à 1 lettre près et on demande la confirmation à la personne s'il y a une présence de doublons éventuelle.
ENVOI pas ENREGISTREMENT des données
Pas d'état "finalisé" → "retour tutelle" plutôt

sur les use case :
OSR : Pas d'action de déposer le dossier de la tutelle → supprimer
Valider un dossier → clôturer/finaliser/mettre en retour tutelle
On ne voit pas la demande passer aux différents états (pas nécessaire ?)
RH → Mettre le dossier en attente tutelle

OSS : Demander une prolongation de contrat doit inclure valider la demande
Valider un dossier → clôturer/finaliser

Regarder pour faire à peu près la même interface que pour le projet SILORE (disponible sur son GitLab)

5ème rendez-vous

Finalisation des séquence et des use case
Représentation de l'OSR sous bonita

OSR → Outil de Suivi de Recrutement
OSS → Outil de Suivi de Stage

Cellule RH : prendre en charge inclut demande de pièces → envoie un mail
Ajouter un moyen d'attendre que les pièces sont toutes déposées avant de passer en attente tutelle
Contrat crée → la cellule RH passe la demande à l'état de retour tutelle, ce qui envoie une notification
Bien penser à mettre toutes les notifications

On va bien tout réécrire sous une nouvelle technologie, on va pas reprendre de code à priori
La saisie des pièces se fait au fur et à mesure : il faut une liste de pièces avec ce qui est reçu et ce qui reste à recevoir (on discutera de comment on va faire étant donné que la liste change beaucoup et souvent) →
Au moment de la mise en attente tutelle, éventuellement choisir les documents à ajouter dans une liste prédéfinie dans une table ?
Ajouter la duplication aussi dans l'OSS
Les RH doivent voir toutes les demandes à tous les états (l'admin aussi du coup)
Responsable de contrat voit les demandes qu'il a faites et celles qu'il a validé
La remise de l'application se fera sur GitLab, utilisation des méthodes agiles et du modèle MVC
L'assistante de contrat voit toutes les demandes qui ont lieu dans son équipe

Mise à niveau sur CodeIgniter

API → Requête CURL Json, soit on les utilise soit on fait des tables :
Pour les personnes : mémoire
La nationalité : tables
Les groupes/équipes : mémoire
Contrat : tables
On garde les anciennes infos (+ leurs états etc) liées aux demandes

6ème rendez-vous

Réunion visio 17/01

- Présentation Bonitasoft OSS

>attribution de bureau commun : notification de la cellule RH et du demandeur.
>demande finalisée : finaliser la demande est une action (en plus de l'état) qui implique la notification du recruteur ou chef d'équipe avant la transmission de pièces et leur enregistrement dans la BDD.
>On prévient le chef d'équipe uniquement s'il nétait pas dans la boucle. Inutile de renvoyer une notification au demandeur
>Pas de transmission des données de paiement à SILOSE

- Présentation du diagramme de Gantt

> Rallonger la période d'implémentation
> Modélisation de la BDD
> 2 prototypes Interface CSS - interaction
> Tous les contrôles
> Coder les maj en BDD
> Les échanges de mail

MCD
Liste des entités et des attributs
Et les relations entre les entités

7ème rendez-vous

Réunion visio 24/01

- Rendu d'une première modélisation de la BDD et échanges sur les choses à modifier

>informations de recrutement : elles sont récoltées
>lier id_groupe de responsabilité à personnel
>La table des responsabilités est à part car une personne a une ou plusieurs responsabilités
=>Créer une table de lien entre équipe, personne et responsabilités

- Charger le prototype

> Clone HTTP de l'application en CodeIgniter4

- Notre convention est finalement prise en charge !

8ème rendez-vous

Réunion visio 02/02

- Modification de la V2 du MCD

> Indiquer les multiplicités dans le modèle E/R
_ un recruteur est lié à un ou plusieurs groupes de rattachement
> un membre recruté ne fera pas partie du personnel permanent
> créer une table recrutement et non pas “demande + futur recruté”
> on ne gère pas la liste des diplômes, c’est un libellé, saisie libre
_ on peut néanmoins avoir une table de niveau de diplôme
> id_demande est un serial (s’incrémente seul)

=>On va communiquer avec les tables extérieures à la nôtre par le biais de l’API
les informations seront gérées en mémoire (mais on peut quand même les modéliser

- La gestion des pièces fournies

> Plusieurs documents par demande
> Création d’une table “documents génériques” ?
_ avec nom et description

=> La tutelle est dans la demande mais elle est déterminée par rapport à l’organisme gestionnaire qui est dans le contrat

9ème rendez-vous

Réunion visio 07/02

- Les entités sont bien modélisées

=> On a notre modèle entité-relation

- On récupèrera avec des routines quotidiennes les tables contrat, groupe, unité, niveau de diplômes, pays, responsabilités

- Pour les personnels permanents, on recherche à partir du login pour récupérer ses infos : on les garde en mémoire le temps de la connexion : personnel, rattachement, rôle, responsabilités.

> On a des rôles très précis, on ne va pas les maîtriser mais les associer à nos rôles
> Création d'une routine qui va associer les responsabilités entre elles.

- Pour la suite : modèle logique/physique des données

> liste des entités avec les attributs en format texte (clé primaire et secondaire), surtout générer le schéma, script SQL PostGres.

- Création d'une nouvelle section dans le Gitlab pour les rendus.