

COMPTE RENDU TP1 et TP2

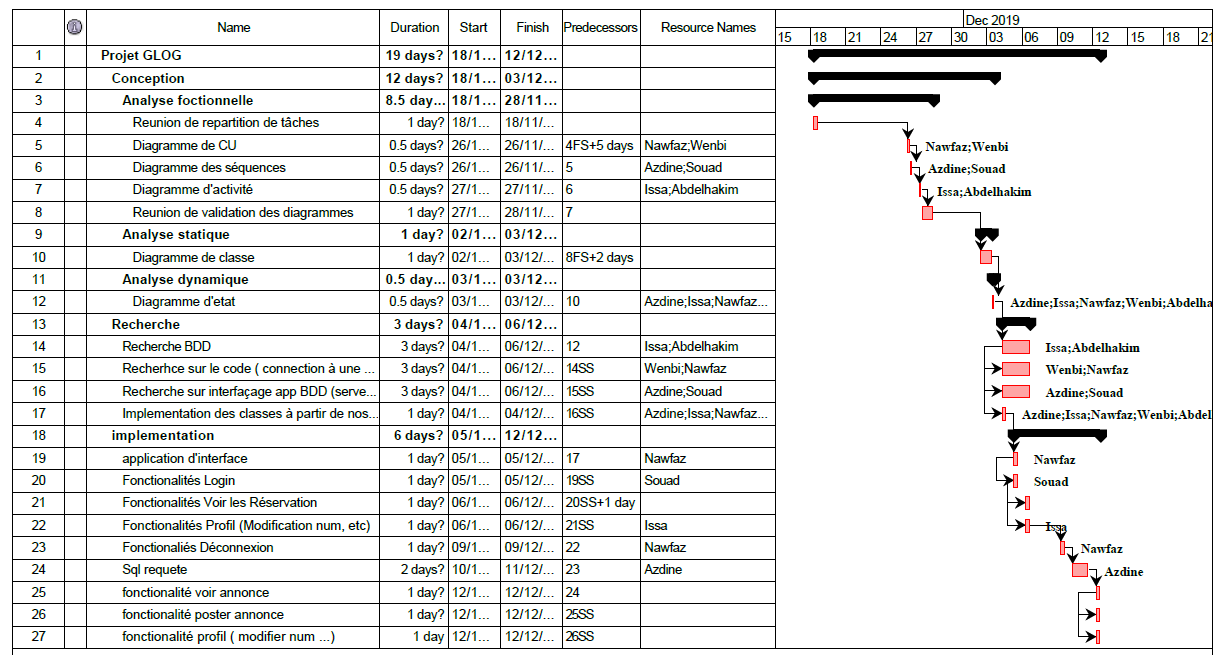


|  |  |
| --- | --- |
| Yahia IHDENE | Souad AIT HAMOUDA |
| Wenbi LIN | Hakim OUBRAHAM |
| Issa DHIFI | Nawfaz JAUFURALLY |

**CONTEXTE** :

Comme il a été dit dans le cahier des charges. Le but de notre projet est de réaliser une application capable d’organiser facilement des évènements de sport, rassembler des amis pour prévoir des matchs de tout type de sport.

**Séance 1**

Lors de la première séance de TP, il a fallu réfléchir sur la manière de procéder et de comment on allait découper les taches : 

**Figure 1 : Diagramme de GANTT**

Toujours durant cette première séance, mon équipe et moi-même avons identifié les acteurs ainsi que les différents cas d’utilisations pour différents scénarios afin d’établir un diagramme d’utilisation.

**Séance 2**

Lors de la séance de TP numéro 2, nous avons commencé à travailler en binôme.

Issa :

2ème semaine [01/12 – 06/12]

- Installation Android studio

- Initiation d’Android Studio avec Nawfaz

- Diagramme d’activité avec Hakim

- Installation de l’IDE IntelliJ IDEA

- Installation de Mysql et gestion de bases

Hakim :

1ère semaine du projet [24/11 - 30/11] :

- Initiation d’Android Studio

2eme semaine [01/12 – 06/12] :

- Diagramme d’activité ‘avec issa

07/12 – 17/12 :

* Configuration de l'ide intellij avec maven, installation et configuration du serveur tomcat,
* Création des class User, DBAccessor côté serveur Web de l'application pour la **« connexion » « inscription » « modification profil »** et « **déconnexion »**

**→**  (coté client avec Souad)

* Tests des différentes fonctionnalités implémentées entre la partie client et serveur (avec souad)
* création d’un fichier qui permet la communication entre les différentes activités
* Rédaction de la parti serveur web dans le rapport
* integration de toutes les fonctionnalité du code ( avec souad)

Yahia :

1ère semaine du projet [24/11 - 30/11] :

- Création et ajout des membres de l’équipe dans github

- Configuration du serveur web apache

- initiation sur Android studio

- Diagramme de Séquence

2eme semaine [01/12 – 06/12] :

- Mise en place du serveur web en interne

- Installation et configuration de la base de données

- Externalisation du serveur web et migration de la base de données MySQL

- Création de la page d’entrée de l’application Android MySport

- mise en place d’une 1ère version d’architecture de base de données

07/12 - 17/12

* Configuration de l'ide intellij avec maven, installation et configuration du serveur tomcat, installation du reverse proxy avec in dns pour la connexion ecterne ngrok
* Répartition. Ded tâches avec les membres de l'équipe
* Création des class réservations, annonces et terrains smdznd le côté serveur Web de l'application
* Costumisation de la base de donnée en changeant quelques table mise par Wenbi.
* Création du code pour l'ajout d'une réservation, la modifier, la supprimer ou voir toutes les réservations faites par l'utilisateur
* Création d'une première version de code pour l'ajout d'une annonce, suppression, et lister les annonce en s'appuyant sur le désérialiser de Wenbi
* Contribution au debygage de l'erreur d'envoi d'informations du côté serveur vers les clients
* Redaction de la parti base de donnée dans le rapport
* Merge du code reservation et annoncé avec Wenbi
* Tester le bon fonctionnement des méthode du serveur tomcat avec le code du client Android avec Nawfaz.

Souad :

1ère semaine du projet [24/11 - 30/11] :

- Création de la maquette sur powerPoint

- initiation sur Android studio et mySQL

- Diagramme de Séquence

2eme semaine [01/12 – 06/12] :

- Installation de l’IDE IntelliJ IDEA

- Installation de Mysql et gestion de bases

- Création de l’interface de l’application ( pages : Mes reservation – Mes annonces – Ajouter une annonce – reserver une annonce – Modifier profil – Recherche et Deconnexion )

07/12 - 17/12

* Création des class User, DBAccessor ModifProfil côté client de l'application pour la **« connexion » « inscription » « modification profil »** et « **déconnexion »** (avec Hakim)
* Tests des différentes fonctionnalités implémentées entre la partie client et serveur avec Hakim

Rédaction du rapport sur les parties :

1. Acteurs, cas d’utilisation et scenario
2. Diagramme Cas d’utilisation
3. Diagramme de séquence

Wenbi :

-Créer le projet public sur github, avec Hakim, nous avons clone le projet de github mais nous n’arrivons pas push sur le serveur distant (en passant par ssh et http).

-Installer MySQL sur pc et apache PHP

- test : on partagea la connexion en wifi, on arrive à accéder sur la page web à partir de téléphone

- Diagramme de classe, Diagramme de cas d’utilisation

- installer Docker et la volume LAMP : pour tester les requêtes de sql.

**Lundi 9 déc** : mettre à jour les tables de sql en ajoutant les contraintes avec Yahia

**Mardi 10 decembre :** hash un string, récrire les requêtes de SQL, comprendre les fonctionnements de refrofit 2

**Jeudi 12 decembre et Vendredi :** Déployer le code sur tomcat sous window avec l’intellig, mais j’ai une erreur 500, Le serveur est actif mais la compilation de code java ne fonctionne pas, il lui manque des librairies.

Passer sous Ubuntu, installer Tomcat et configurer :

Problème : Tomcat ne démarrer pas :

Cause : Tomcat n’a pas le droit d’écriture et exécuté sur certains dossiers, et intellig ne trouve pas de tomcat

Fonction : ajout une annonce dans la base de données en lui fournissant une json d’Annonce que l’utilisateur a envoyée (simulation de client avec le browser) => OK données bien insérer.

Sérialisation et Désérialisation d’annonce

Problèmes rencontrés : Rétrofit propose une librairie Gson permet de converti les objets en string de json (sérialisation) et de json en Objet (désérialisation).

Gson n’arrive pas reconstruit Annonce, en effet gson n’arrive pas gérer des objets polymorphismes

* Solution : créer une désérialiser customiser pour l’annonce.

Recherche pour réaliser un arrayListe d’Annonce avec le gson customisé.

Tester convertir en json avec Gson et utiliser le désérialiser pour reconstruire l’objet Annonce avec les bons types d’item : terrain.

Fonction : recevoir une liste des annonces d’un utilisateur, en entrée l’id d’utilisateur => ok avec le browser j’ai récupéré une liste.

Fonction : recevoir toutes les annonces. =>ok avec le browser

**Weekend :**

Merge les Annonces et mes Réservations avec Yahia

Debug Avec Yahia problème de cote Client, l’application fait une requête au page de Mes Annonces / Annonces : le serveur a pu retourner une réponse en des liste des objets Annonces sous format d’un json, mais l’application n’a pas pu récupérer ce fichier.

En effet : Nous avons construit utiliser un Retrofit avec un convertisseur de Gson, qui convertie un json en string à l’objet choisie. Pour convertir en une liste Annonce, nous devrons utiliser la désérialiser customisé,

Lundi : Régler le problème des annonces : en enlevant le Gson simple et récupérer un objet de type ReponseBody qui contient le string et on désérialise avec Gson customise.

**Nawfaz :**

Première séance :

1. Diagramme de classe.
2. Création des interfaces de l'application(Page connexion,inscription,Accueil).

Deuxième séance :

Implémentation des fonctionnalités : afficher une liste d'annonce en utilisant le design pattern Adapter qui étend la classe BaseAdapter (et ce dernier implémente l'interface ListAdapter) dans android studio. Utilisation du design pattern Factory method pour créer les types d’Item (Terrain etc..) afin que cette partie de l'application soit fermé à la modification mais ouvert à l'extension. Création des classes POJO (Annonce, Terrain, Réservation, User, Item, FactoryItem etc.)

3ème – 6ème séance : Développement de l’application **côté client** sur les parties suivantes :

1. Page de l’application : Accueil 🡪 Affiche la liste d’annonce
2. Page de l’application : Ajouter une annonce 🡪 poster une annonce vers la base de données. (Avec Yahia côté serveur)
3. Page de l’application : Mes Annonces 🡪 Supprimer une annonce posté dans la base de données. (Côté serveur : Yahia)
4. Page de l’application : Rechercher des annonces et les afficher.
5. Page de l’application : Mes réservations 🡪afficher mes Réservations dans la listView
6. Page de l’application : Réserver une annonce 🡪 Envoie un objet Reservation vers la base en format Json. (Avec Yahia côté serveur).
7. Page de l’application : Supprimer une réservation 🡪 Envoie d’un objet Réservation sous format Json vers le serveur pour supprimer une réservation.(Avec Yahia côté Serveur)

Rédaction du rapport sur les parties suivantes :

1. Application MySport (Introduction)
2. Cahier de Charge
3. Les Design pattern