COMPTE RENDU D’ACTIVITE

Application mobile - MyApplifrais

Sciences – U Lyon

Epreuve E4

**Compte Rendu d’Activité : «MyApplifrais»**

**Application Mobile de Gestion de Frais**

## Composition de l’équipe :

* Eddy CRAHET : Développeur - Designer
* Florian PECHOUX : Développeur
* Matthis BERRARD : Développeur
* Matthieu COLIN : Développeur – Designer

## Commentaire :

Nous avons étudié la problématique de l’entreprise Galaxy Swiss Bourdin au niveau de la gestion des frais.

Réalisation, dans un premier temps, d’une activité de connexion, avec son rendu graphique.

Ensuite, création du controller de l’application mobile, gérant la connexion de l’utilisateur, la récupération et l’affichage de ses frais.

Dans un troisième temps, réalisation de l’activité d’affichage des frais

## Contexte :

Galaxy Swiss Bourdin, ou GSB, est une entreprise de pharmaceutique. Son objectif est d’améliorer son activité concernant la visite médicale.

Son patrimoine informatique est composé :

* Serveurs (aux 6ème Etage et virtualisés)
* Applications métier (progiciel de gestion, …)
* De bases de données dédiées à chaque application (BDD frais de visite, BDD produits du labo, …)
* Ordinateurs de bureau et portables
* Smartphone ou PDA (pour les visiteurs)

L’équipe du SI est constituée : du DSI (qui gère les différents projets et services), d’une équipe d’administrateurs réseaux (pour contrôler l’accès des utilisateurs, …), et d’une équipe de développement (gestion de frais, des comptes rendus, …)

L’entreprise possède une salle serveur d’une centaine d’unités ainsi qu’environ 500 poste répartis sur tout le site. Des ordinateurs portables et phablette sont aussi à dénombrer. Au niveau logiciel, les applications de gestion de frais et de comptes rendus existent mais doivent évoluer.

## Expression des besoins

Analyse et correction de la BDD :

Faire évoluer la BDD afin de prendre en compte les nouvelles fonctionnalités. Le responsable souhaite sécuriser les mots de passe qui sont enregistrés en clair en BDD.

Création du module de gestion de frais sur mobile

* Création des activities
* Configuration des intents
* Design des layout

Sécurisation de l’accès à l’application

* Connexion par identifiant
* Création d’une bdd embarquée

*Matrice RACI*

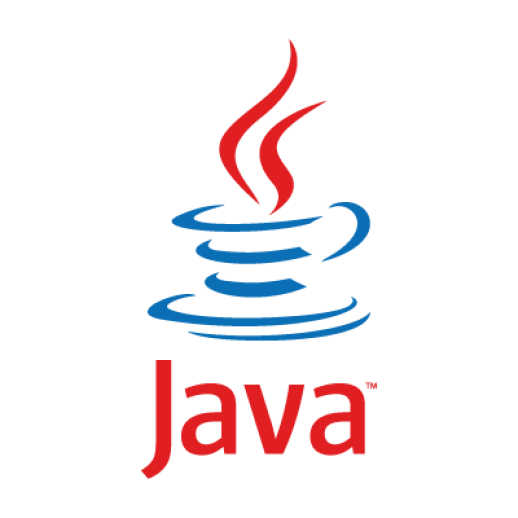
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Activité** | **Eddy** | **Florian** | **Matthis** | **Matthieu** |
| Choix de la mission | C | C | C | A |
| Définition des rôles | I | I | I | A |
| Installation- configuration des outils | R | I | I | A |
| Création des entitées | I | I | I | A |
| Création des activitées | R | R | R | A |
| Création des layouts | R | R | R | A |
| Création du controller | I | I | I | A |
| Création de la base SQLite | I | I | I | A |

# Choix des technologies

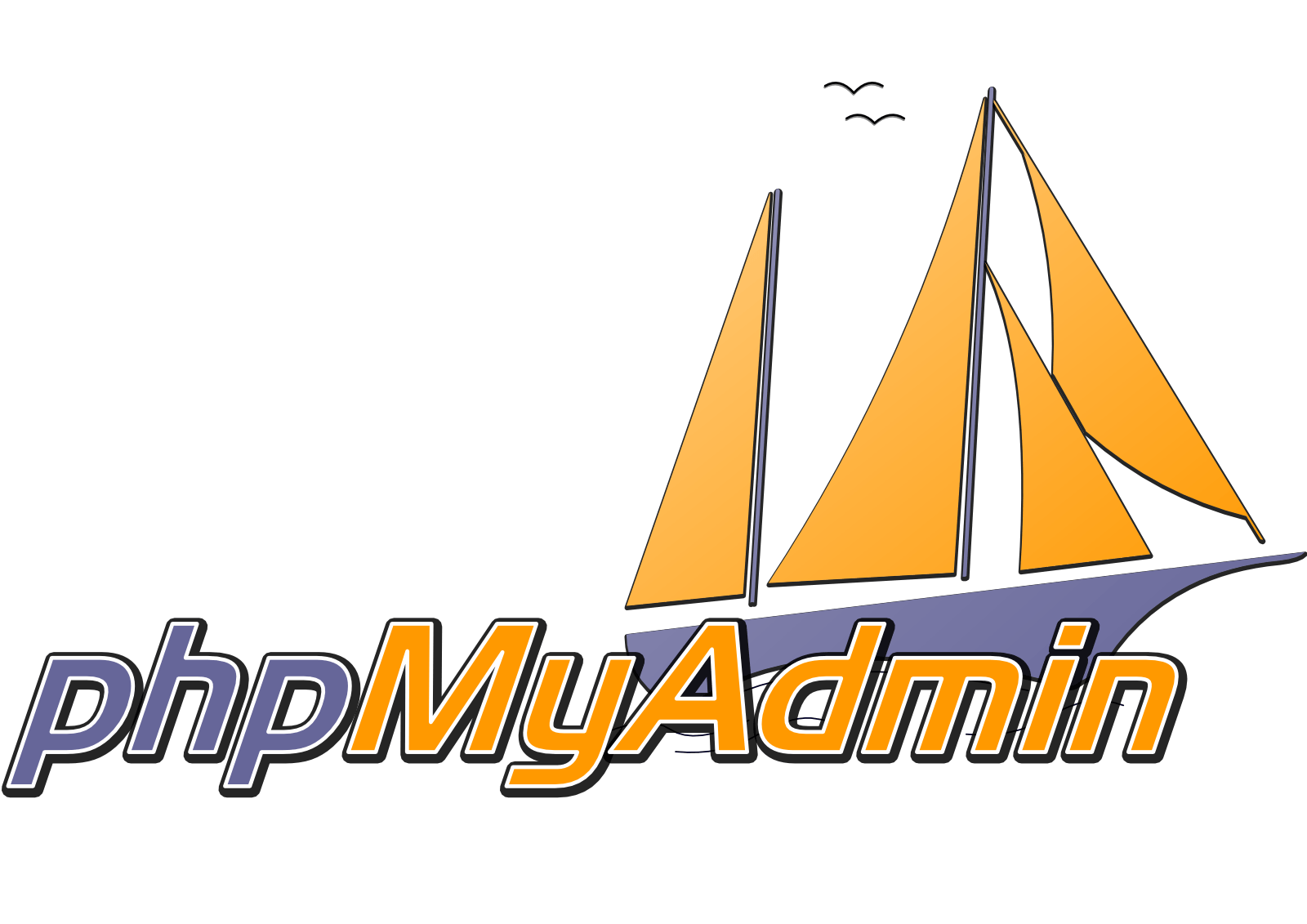
Les technologies ont été choisis en cohérence avec l’environnement de développement (PHP)



IDE : *Android Studio*

Langages : Java, PHP, JSON, HTML



SGBD : *PHP* *MyAdmin*



Serveur : *wamp* *server*

Avant réalisation :

Le projet choisi est le développement d’une application mobile pour la gestion des frais de visites.

Date : 07/10/2017

* Analyse du cahier des charges
* Choix des technologies
* Définitions des tâches

Pendant réalisation :

Problèmes rencontrés et solutions retenues :

* Récupération de la fiche de l’utilisateur
  + Recherche de fiches dans la base selon l’id du visiteur et les mois en cours
* Envoi des données de la fiche (en JSON)
  + Envoi des données des frais dans un tableau
* Problème de récupération des frais
  + En recherche de solution

Après réalisation :

Date de fin : 12/05/2017

Documentation et source sur : <https://github.com/mcolinn/AppliFrais-Mobile>

# Taches réalisées :

# Création du layout

@Override  
**protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);

Récupération du layout dans le dossier de ressource  
 setContentView(R.layout.***activity\_login***);

Création du bouton de connexion  
 Button btnConnexion = (Button) findViewById(R.id.***connect\_button***);

Appel sur évènement de clic  
 btnConnexion.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View button) {  
 **int** id = button.getId();  
 **switch** (id) {  
 **case** R.id.***connect\_button***:

Récupération du texte de l’identifiant sur le layout   
 **final** EditText username\_txt = (EditText) findViewById(R.id.***username\_txt***);

Récupération du texte du mot de passe sur le layout  
 **final** EditText password\_txt = (EditText) findViewById(R.id.***password\_txt***);

Conversion en String pour le traitement  
 String username = username\_txt.getText().toString();  
 String password = password\_txt.getText().toString();

Appel de la fonction de connexion de la classe Asynchrone connect**new** connect().execute(username, password);  
 }  
 }  
 });

# Fonction de connexion

Avant exécution (*onPreExecute()*) affichage d’un message de connexion à la base

En arrière-plan (*doInBackground()*) appel de la fonction de connexion du controller PhoneApi

public function connectAction(Request $request)

{

Récupération du texte des identifiants de l'utilisateur dans la requête

$username = $request->get('username');

$password = $request->get('password');

Recherche de l'utilisateur dans la base

$em = $this->get('doctrine')->getEntityManager();

$user = $this->getDoctrine()->getRepository('FrontBundle:Utilisateur') >findOneByUsername($username);

Récupération et vérification du password

$success = $encoder->isPasswordValid($user->getPassword(), $password, $user->getSalt());

if ($success == true) {

$token = new UsernamePasswordToken($user, $user->getPassword(), 'main', $user->getRoles());

$context = $this->get('security.token\_storage');

$context->setToken($token);

Ecriture de la réponse JSON

$message = $user->getId();

$response = new JsonResponse();

$response->setData(array(

'success' => $success,

'message' => $message,

));

Renvoi de l’id de l’utilisateur connecté

return $response;

}

# Ouverture de l’activité principale

Après succès de la connexion, l’utilisateur est redirigé vers la page d’affichage de ses frais.

Cette redirection est rendue possible grâce à un Intent

Intent listeFiches = **new** Intent(getApplicationContext(), MainActivity.**class**);  
listeFiches.putExtra(**"id"**, **message**);  
startActivity(listeFiches);

# Création du layout

@Override  
**protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);

Récupération du layout dans le dossier de ressource  
 setContentView(R.layout.***activity\_main***);

Création de la barre de menu  
 Toolbar toolbar = (Toolbar) findViewById(R.id.***toolbar***);  
 setSupportActionBar(toolbar);

Récupération de l’identifiant de l’utilisateur

Bundle extras = getIntent().getExtras();  
**if**(extras != **null**)  
 **id** = extras.getString(**"id"**);

Création d’une liste de frais et récupération de la liste du layout

**lesFrais** = **new** ArrayList<>();  
**lv** = (ListView) findViewById(R.id.***ListeFrais***);

Appel de la fonction de chargement des frais  
**new** loadFrais().execute(**id**);

# Fonction de chargement

De même qu’avec la classe connect, la classe loadFrais est asynchrone.

On affiche un message de chargement en poste et on fait appel à la fonction listFicheAction du controller PhoneApi pour charger et afficher les frais correspondant à la fiche de l’utilisateur

public function listFicheAction(Request $request)

{

//récupération de l'id du visiteur dans la requête

$visiteur = $request->get('id');

//récupération de la fiche du mois correspondant à cet id

$em = $this->get('doctrine')->getEntityManager();

$ficheFrais = $this->getDoctrine()->getRepository('FrontBundle:FicheFrais')->findOneBy(

array('utilisateur' => $visiteur, 'mois' => $mois)

);

//récupération des frais de la fiche

if($ficheFrais != null) {

$em = $this->get('doctrine')->getEntityManager();

$lesFraisForfaits = $this->getDoctrine()->getRepository('FrontBundle:FraisForfait')

->findByFicheFrais($ficheFrais->getId());

//récupération des données des frais de la fiche

foreach ($lesFraisForfaits as $unFrais){

$arrayFrais = array(

'type' => $unFrais->getFraisForfaitType()->getLibelle(),

'prix unitaire' => $unFrais->getFraisForfaitType()->getPrixUnitaire(),

'quantite' => $unFrais->getQuantite() ;

) ;}

}

//renvoi des données relatives à la fiche frais : id, mois, tableau des frais

$response = new JsonResponse();

$response->setData(

array( 'success' => $success, 'id fiche' => $ficheFrais->getId(), 'mois fiche' => $ficheFrais->getMois(),

'frais forfait' => $arrayFrais,

)

) ; return $response;

}

# Fonction d’affichage

ArrayAdapter<String> monAdaptater = new ArrayAdapter<String>(MainActivity.this, android.R.layout.simple\_list\_item\_1, lesDescriptions); //lesDescriptions est la collection regroupant les détails de chaque frais

listFrais = (ListView) findViewById(R.id.ListFrais);

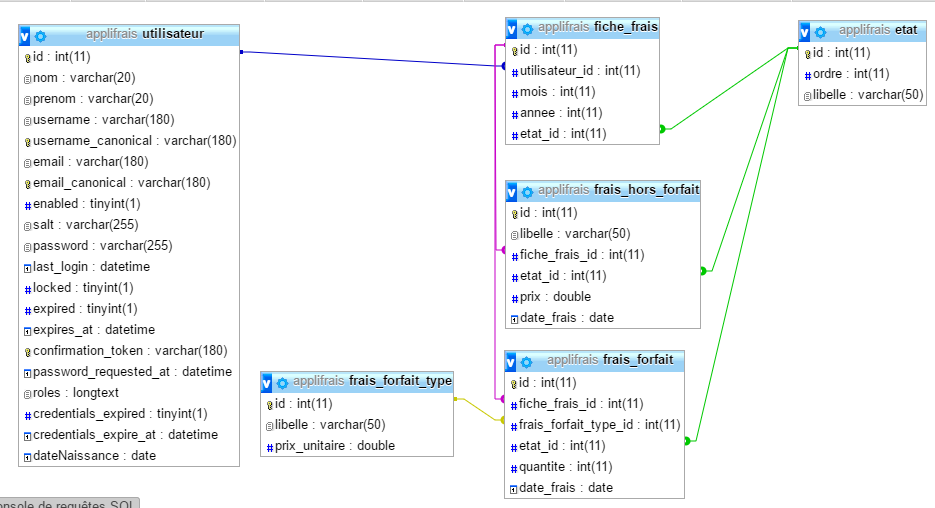
TextView textView = new TextView(MainActivity.this);

textView.setText("Liste des frais");

listFrais.addHeaderView(textView);

listFrais.setAdapter(monAdaptater);

MCD de la base de données :



**Bilan** :

Bien que non totalement fonctionnelle à l’heure actuelle, la réalisation de cette application a permis de découvrir un nouvel environnement de développement (sous android), et de se familiariser avec l’utilisation du JSON pour le transfert de données.

J’ai cependant rencontré quelques problèmes d’organisation car n’ayant pas toujours tous les détails du contexte en tête. J’ai aussi eu des difficultés avec la gestion de l’équipe par un manque de communication et l’inégale évolution de chacun.

Au niveau de l’application, la nécessité de communication avec l’application web a poser des problèmes de transfert de données.

La première évolution que j’apporterai à l’application serai l’ajout du mois et de l’état de la fiche et des frais dans les informations à afficher. Ensuite la modification de l’affichage du message d’authentification qui affichage l’id du visiteur au lieu de son nom .Plus tard, l’ajout d’un écran permettant d’ajout de frais via le mobile. Un autre point intéressant serai aussi l’ajout d’une base de données embarquée SQLite pour permettre l’ajout et la consultation des frais en itinérance, effectuant une synchronisation journalière lors d’une connexion à internet.

Au niveau personnel, il me faudra améliorer ma communication, tout comme mon écoute, et ma présence au sein de l’équipe. En tant que chef de projet je devrais aussi apprendre à déléguer certaines tâches et à motiver l’équipe.