

Master of Science HES-SO in Engineering

Orientation : Technologies de l'information et de la communication (TIC)

AGENDA DE L'ESPACE ROMAND DE LA FORMATION (APP CIIP).

Développement d'une application Android de gestion des
manifestations pour l'Espace romand de la formation.

Fait par

Gaël Tegua Tegua

Sous la direction de

Prof. Fatemi Nastaran

Lausanne, HES-SO//Master, 2017

Table des matières

1. Introduction	5
2. Analyse et conception	7
2.1 Cadre, cible et contenus	7
2.2 Scenarios d'utilisations	8
2.3 Planification	12
2.4 Ergonomie et navigation	14
2.5 Technologies et outils	19
2.6 Architecture	22
2.7 Methodologie	23
3. Implémentation	24
3.1 Serveur REST	25
3.1.1 Base de données	25
3.1.2 Diagramme de classes	27
3.1.3 Interfaces	33
3.2 Client ANDROID	39
3.2.1 Diagramme de classes	39
3.2.2 Interfaces	43
4. Tests, maintenance et documentation	51
5. Prolongement et ameliorations	52
6. Conclusion	54
7. Références	56

1. INTRODUCTION

La Conférence intercantonale de l'instruction publique de la Suisse romande et du Tessin (CIIP) est une institution de droit public ayant pour but la coordination des cantons membres en matière de formation et de culture. Son secrétariat général (SG-CIIP) met à la disposition de publics spécifiques (délégués, responsables d'établissements scolaires, enseignants, ...), ainsi que du grand public, diverses informations de type agenda. Sur le site internet de l'institution ces informations sont des plus consultées, pour autant que les utilisateurs sachent où chercher. L'information de ce type qui est la plus demandée est le calendrier des vacances scolaires des divers cantons pour l'année en cours et les deux années suivantes.

Le SG-CIIP organise par ailleurs chaque année diverses manifestations auxquelles il est possible de s'inscrire. Il s'agit principalement de la semaine de la langue française et de la francophonie (mi-mars), de la semaine des médias à l'école (automne), et des journées de réflexion/formation (technologies et éducation aux médias, pédagogie spécialisée, développement durable).

Dans le cadre de la commission consultative des partenaires (COPAR) se déroule un échange intensif d'informations avec les associations faïtières romandes ou latines reconnues comme partenaires par les statuts de la CIIP : parents, enseignants, chefs d'établissement et chercheurs. Chacune d'entre elles s'intéresse aux manifestations mises sur pied dans l'Espace romand de la formation et organise également elle-même des colloques, des semaines thématiques, des congrès, dont l'accès peut être public.

Pour simplifier l'accès aux informations mentionnées ci-dessus le présent projet de développement initié dans le cadre de mon projet d'approfondissement, a consisté à consolider le programme de manifestations et les vacances scolaires cantonales au sein d'une application.

Celle-ci donne aux membres de l'Espace romand de la formation, la CIIP, ses partenaires associatifs reconnus, ainsi qu'au grand public les possibilités de rechercher, de consulter et de s'inscrire aux diverses manifestations. L'accès aux informations est facilité grâce à la centralisation de celles-ci. Des notifications automatiques basée sur la localisation proposent aux utilisateurs des évènements intéressants dans les environs.

Cette application récupère les données depuis un serveur possédant une base de données. Les vacances et les manifestations sont créés via l'interface web du serveur.

Le nom d'application, APP CIIP a été choisi de manière à être simple à retenir. Il fait référence à l'AP CIIP, Assemblée Plénière de la CIIP réunissant l'ensemble des conseillers d'états romands en charge de l'éducation.

Le choix de la technologie Android a été fait à cause de l'ouverture et de la polyvalence de sa plateforme de développement.

Pour effectuer la gestion du projet, j'ai fait un découpage en 3 grandes phases. Une première phase d'analyse et de conception, où il a fallu définir des fonctionnalités et designer une architecture. Ensuite, une phase d'implémentation, où j'ai réalisé un prototype. Enfin une phase de tests, de maintenance et de documentation.

Ces différentes phases et les divers choix effectués sont étayés dans la suite de ce rapport.

2. ANALYSE ET CONCEPTION

La première phase du projet a consisté à décrire formellement le système. Les entretiens avec mon encadreur académique madame Nastaran Fatemi et monsieur Olivier Maradan, secrétaire général de la CIIP, ont permis de spécifier ce que doit faire l'application et de discuter comment réaliser le tout.

J'ai débuté par un brainstorming me permettant de répertorier les besoins, les moyens, les mesures et les objectifs. Ensuite, je me suis documenté sur les sujets et domaines liés au projet (développement Android, API Restful, ...).

En partant de ces éléments d'analyse, j'ai défini un planning projet, choisi les technologies à utiliser, et effectué divers designs et architectures.

2.1 CADRE, CIBLE ET CONTENUS

La première étape de la phase d'analyse et de conception a été de regrouper des informations sur le cadre, la cible et les contenus de l'application. Ces informations ont influencé les différents choix graphiques et technologiques.

Le besoin que l'application essaye de résoudre est l'accès facilité aux informations sur les manifestations et les vacances scolaires. La région visée par l'application est principalement la partie romande de la Suisse. L'audience visée est a priori l'ensemble des collaborateurs travaillant dans le domaine de l'éducation et les parents d'élèves essentiellement.

Ces éléments m'ont poussé à définir le français comme langue de l'application et à utiliser des interfaces simples et intuitives.

Les contenus de l'application sont uniquement constitués de textes et d'images illustratives. Pour la production de ceux-ci j'ai choisi l'utilisation d'une interface web et d'une base de données relationnelles pour le stockage.

L'application cliente reçoit les contenus via des requêtes sur un API et les mets en forme.

2.2 SCENARIOS D'UTILISATIONS

Après avoir analysé le contexte il a fallu définir un scope pour le projet d'approfondissement, c'est-à-dire un périmètre de fonctionnalités pouvant être réalisées. Ce périmètre de fonctionnalités constitue une sorte de cahier de charge fonctionnel que j'ai choisi de regrouper en fonctionnalités Serveur et fonctionnalités Client.

FONCTIONNALITÉS SERVEUR

- Créer une vacance scolaire

Via le formulaire du serveur REST, on peut introduire une vacance en spécifiant son titre, une description, les dates de début, de fin, et le canton concerné. Ces informations sont stockées dans la base de données relationnelles.

Les images des drapeaux des cantons sont associées automatiquement à partir de leurs noms.

- Consulter la liste de vacances scolaires

A partir d'une URL RESTful, l'application web retourne la liste des vacances au format JSON. Le résultat retourné pourra ainsi être « consommé » par une autre application. Par défaut les vacances retournées sont triées par date croissante.

- **Rechercher des vacances scolaires par mots clés**

En recevant un mot clé en paramètre d'une URL donnée, l'application retourne une liste de vacances au format JSON contenant ce mot clé dans l'un de ses attributs.

- **Créer une manifestation**

De même que pour les vacances scolaires, au travers d'un formulaire web on peut créer une manifestation en indiquant son titre, sa description, les horaires de début et de fins, une image illustrative, l'organisateur et le site web correspondant. L'ensemble des informations introduites est stocké dans la base de données.

- **Consulter la liste de manifestations**

A partir d'une URL RESTful, l'application web retourne la liste des manifestations au format JSON. Le résultat retourné pourra ainsi être « consommé » par une autre application. Par default les manifestations sont aussi triées par date croissante.

- **Rechercher des manifestations par mots clés**

En recevant un mot clé en paramètre d'une URL donnée, l'application retourne une liste de manifestations au format JSON contenant ce mot clé dans l'un de ses attributs.

- **Rechercher des manifestations par localisation**

En recevant une localisation en paramètre d'une URL donnée, l'application retourne une liste de manifestations au format JSON en précisant pour chacune d'elles la distance jusqu'à son lieu de déroulement.

- **S'inscrire à une manifestation**

En recevant en paramètre les informations d'un utilisateur (nom, prénom, email, téléphone) et l'identifiant d'une manifestation l'application introduit dans sa base de données une inscription.

FONCTIONNALITÉS CLIENT

- **Afficher la liste de manifestation**

Via l'URL du serveur l'application cliente transmet sa position géographique et reçoit en retour une liste de manifestations au format JSON. L'application met en forme les informations reçues et affiche dans une liste déroulante, le titre, la description, la date de début, l'adresse et l'image illustrative miniaturisée de chaque manifestation.

Si le titre et/ou la description sont trop longs, ils ne sont pas entièrement affichés.

- **Afficher les details d'une manifestation**

En sélectionnant une des manifestations de la liste, un affichage détaillé est présenté. En plus des attributs de la liste on peut voir la date de fin, la météo courante et l'image illustrative en plus grande taille.

- **Rechercher une manifestation par mots clés**

Dans la barre d'action de l'application on peut introduire un mot clés et recevoir une liste de manifestations contenant ce mot dans ses attributs.

- **S'inscrire à une manifestation**

L'application ne disposant pas de profil, l'utilisateur s'inscrit à une manifestation via un formulaire dans lequel il saisit son nom, prénom, email, numéro de téléphone, et un message. Ces informations ainsi que l'identifiant de la manifestation concernée sont sauvegardés dans la base de données du serveur.

- **Notifier une manifestation**

A partir de la position géographique courante de l'utilisateur, l'application cliente reçoit toute les une heure une notification d'une manifestation à proximité.

En cliquant sur la notification l'utilisateur affiche les détails de la manifestation.

- **Afficher la liste de vacances scolaires**

Via l'URL du serveur l'application cliente reçoit une liste de vacances au format JSON. L'application met en forme les informations reçues et affiche dans une liste déroulante, le titre, la description, la date de début et l'image illustrative miniaturisée de chaque vacance.

- **Afficher les détails d'une vacance scolaire**

En sélectionnant une des vacances de la liste, un affichage détaillé est présenté.

- **Rechercher une vacance scolaire**

Dans la barre d'action de l'application on peut introduire un mot clés et recevoir une liste de vacances contenant ce mot dans ses attributs.

2.3 PLANIFICATION

Du 24 février au 9 juin 2017, 180 heures de travail ont été prévues pour ce projet d'approfondissement.

Pour l'encadrement de celui-ci des entretiens hebdomadaires ont été fixés tous les lundis avec mon encadreur madame Fatemi.

Pour chacune des taches j'ai été l'unique responsable et pour certaines d'entre elles j'ai pu bénéficier de la participation de monsieur Maradan ou de madame Fatemi.

Pour la gestion de mon projet et sa réalisation dans le temps imparti l'échéancier suivant a été fixé.

SEMAINE	DATE	TACHES	PARTICIPANT(S)	LIVRABLE	PHASE
01	24.02	<ul style="list-style-type: none"> - Brainstorming. - Analyse du cahier des charges et du contexte. - Clarification du projet. 	G. Tegua O. Maradan	Cahier de charge. Liste des taches.	Analyse et conception
02	03.03	<ul style="list-style-type: none"> - Spécification des fonctionnalités. - Design de architecture. - Planification 	G. Tegua N. Fatemi	Liste des uses-cases. Architecture. Planning.	Analyse et conception
03	10.03	<ul style="list-style-type: none"> - Conception de la navigation. - Design des vues des activités. - Choix des technologies 	G. Tegua	Arborescence. Mockups.	Analyse et conception

04	17.03	<ul style="list-style-type: none"> - Mise en place de l'environnement de développement. - Modélisation et implémentation de la base de données. 	G. Teguia	Modèle Logique de Données.	Implémentation
05	24.03	<ul style="list-style-type: none"> - Création d'une manifestation via formulaire web. - Affichage de la liste de manifestation via l'API REST. 	G. Teguia	Serveur REST.	Implémentation
06	31.03	<ul style="list-style-type: none"> - Affichage du SplashScreen. - Affichage de la liste de manifestation sur le Client Android. 	G. Teguia	Client Android.	Implémentation
07	07.04	<ul style="list-style-type: none"> - Affichage d'une manifestation détaillée - Recherche par mots clés de manifestations. 	G. Teguia	Client Android.	Implémentation
08	14.04	<ul style="list-style-type: none"> - Recherche par localisation de manifestations. - Inscription à une manifestation. 	G. Teguia	Client Android. Serveur REST.	Implémentation
09	28.04	<ul style="list-style-type: none"> - Creation d'une vacance via formulaire web. - Affichage de la liste de vacances via l'API REST. 	G. Teguia	Client Android.	Implémentation
10	05.05	<ul style="list-style-type: none"> - Affichage de la liste de vacances sur le Client Android. 	G. Teguia	Serveur REST. Client Android.	Implémentation

11	12.05	<ul style="list-style-type: none"> - Affichage d'une vacance détaillée. - Recherche par mots clés de vacances. 	G. Teguia	Client Android.	Implémentation
12	19.05	<ul style="list-style-type: none"> - Notification de manifestation sur le Client Android. 	G. Teguia	Client Android.	Implémentation
13	26.05	<ul style="list-style-type: none"> - Tests, debugs et évaluation de l'application. 	G. Teguia	Serveur REST. Client Android. Rapport	Tests, maintenance et documentation
14	02.06	<ul style="list-style-type: none"> - Tests, debugs et évaluation de l'application. - Rédaction du rapport final. 	G. Teguia	Serveur REST. Client Android. Rapport.	Tests, maintenance et documentation
15	09.06	<ul style="list-style-type: none"> - Rédaction et relecture du rapport final. 	G. Teguia	Rapport.	Tests, maintenance et documentation

2.4 ERGONOMIE ET NAVIGATION

Dans le but de faciliter l'accès aux informations et fonctionnalités, j'ai essayé de designer un plan des plus simples pour l'application mobile. L'arborescence définie a été la suivante:

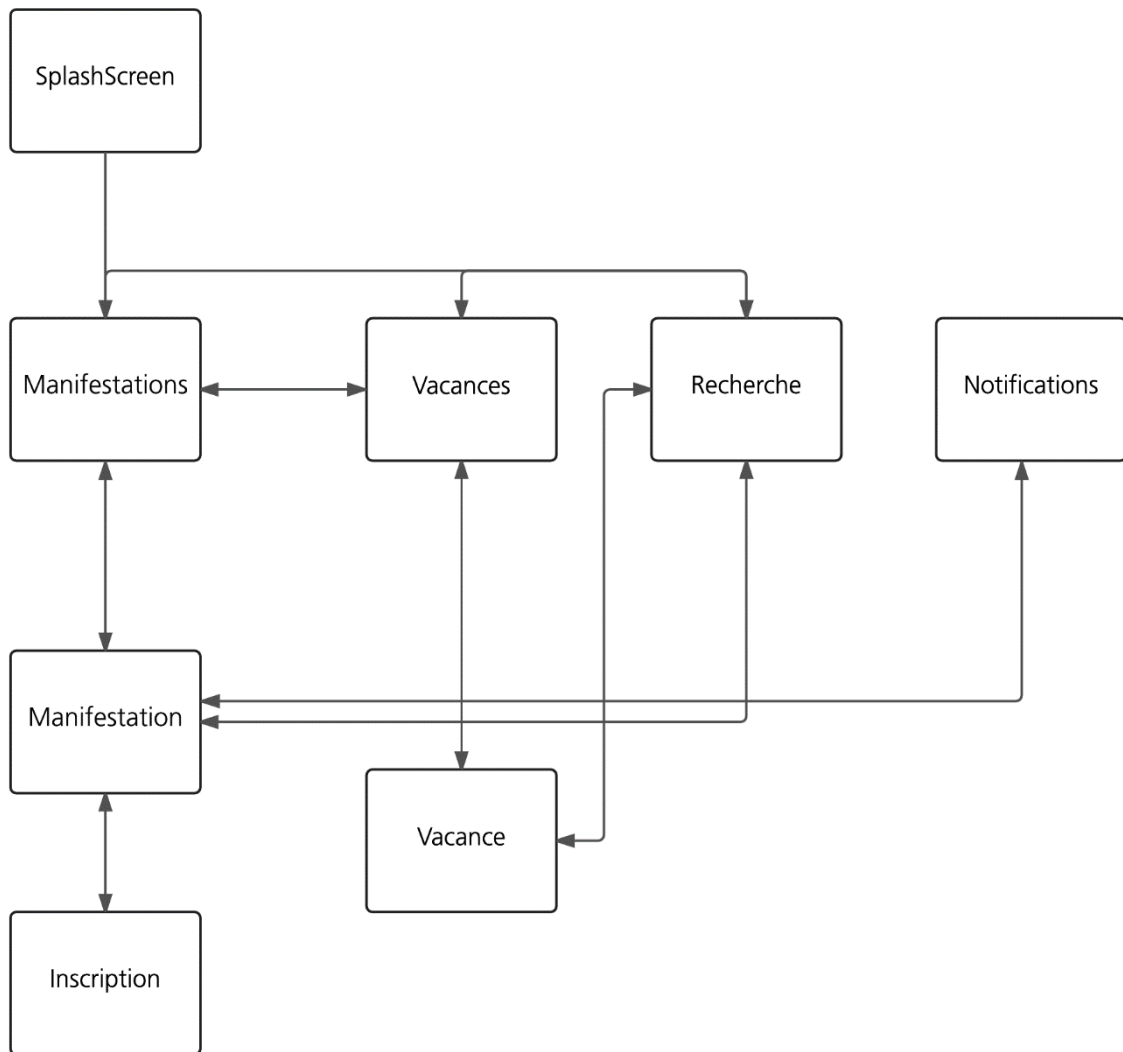


Figure 1 – Arborescence application mobile

Pour l'optimisation de la navigation, un unique « objectif » est visée pour chaque vue. En dehors du SplashScreen de démarrage, chaque vue est subdivisée en deux partie : Une zone d'entête et une zone de contenu. Depuis la vue principale on peut accéder à la liste des manifestations comme à la liste des vacances.

La recherche est à chaque fois accessible depuis la zone d'entête.

Lors du scroll, l'entête reste fixe. Le logo de la CIIP est utilisé pour le SplashScreen et pour toute les images par défaut. Pour avoir une représentation de la répartition des contenus j'ai réalisé les maquettes/mockups suivants.



Figure 2 – SplashScreen

C'est la première fenêtre affichée au lancement de l'application. Elle présente le logo de l'application pendant le chargement des informations.



Figure 3 - Vue Manifestations

Il s'agit de la vue principale de l'application. Au démarrage elle affiche la liste des manifestations reçues via l'API et donne la possibilité de basculer vers la liste des vacances. En haut, dans la barre d'action on peut réaliser une recherche par mots clés de manifestations. En cliquant sur l'un des éléments de la liste on passe à la vue Manifestation.



Figure 4 - Vue Manifestation

Elle affiche les détails d'une manifestation sélectionnée notamment la météo courante et donne la possibilité de s'y inscrire via la vue Inscription.



Figure 5 - Vue Inscription

Elle permet à l'utilisateur de s'inscrire à une manifestation en transmettant ses informations de contact.

L'affichage de la liste de vacances et d'une vacance détaillée reprennent les mêmes principes que ceux des manifestations.

2.5 TECHNOLOGIES ET OUTILS

Pour réaliser mon serveur REST et mon client Android, les différentes technologies et outils suivants ont été utilisés.

- **Android Studio 2.3.2**

J'ai utilisé cet environnement totalement dédié aux développements Android pour la réalisation de mon application cliente. Son interface est simple et intuitive. Et les composants nécessaires tels que le SDK Android y sont directement préinstallés.

- **Eclipse Java EE IDE for Web Developers. Neon.1 Release (4.6.1)**

J'ai utilisé cet environnement de développement pour implémenter mon serveur REST. Il est open source, simple d'utilisation et comprend de façon native les outils de la plateforme Java Entreprise. Il facilite l'importation de librairies et de plugins, permettant ainsi d'étendre ses fonctionnalités.

- **MySQL version 6.3.7**

J'ai utilisé cet environnement pour la réalisation et la gestion de ma base de données. C'est un logiciel gratuit, rapide, multi-utilisateurs, portable et facile à administrer.

- **Tomcat v7.0**

J'ai utilisé Tomcat comme serveur http pour mon application Web Java EE. C'est un serveur Web open source développé par Apache SoftwareFoundation. Il comprend un conteneur de servlets et implémente les spécifications Java Servlet et JavaServer Pages (JSP).

L'intégration à Eclipse est simplifiée et il est très pratique comme serveur de développement pour la construction d'applications ou autres technologies Web dynamiques basées sur Java.

En production, l'application pourrait être déployée sur des serveurs plus robustes tels que Glassfish, WebLogic, WebSphere, ou encore JBoss,

- **JAX-RS (Java API for RESTful Web Services)**

J'ai utilisé cet API Java pour implémenter mon service Web Restful. Il est léger, simple, et permet le transfert de représentation de ressources (exposition des « session bean » en tant que service). C'est une norme faisant partie de Java EE, qui peut être déployée sur n'importe quel serveur d'applications Java et qui facilite grandement le développement lorsqu'elle est utilisée avec d'autres technologies Java EE.

- **lucidchart.com**

J'ai utilisé cette plateforme web pour la réalisation de diagrammes (arborescence, architecture) et de mockups.

- **ObjectAid UML**

J'ai utilisé ce plugin pour la réalisation des diagrammes de classes UML.

- **Bootstrap version 3**

J'ai utilisé cette collection d'outils pour réaliser les interfaces (formulaires web) de mon serveur REST.

- **GitHub**

J'ai utilisé ce service pour la gestion de mon code. Il est rapide et propose un excellent algorithme de fusion. Le code est disponible sous github.com/gaelteguia/appciip

2.6 ARCHITECTURE

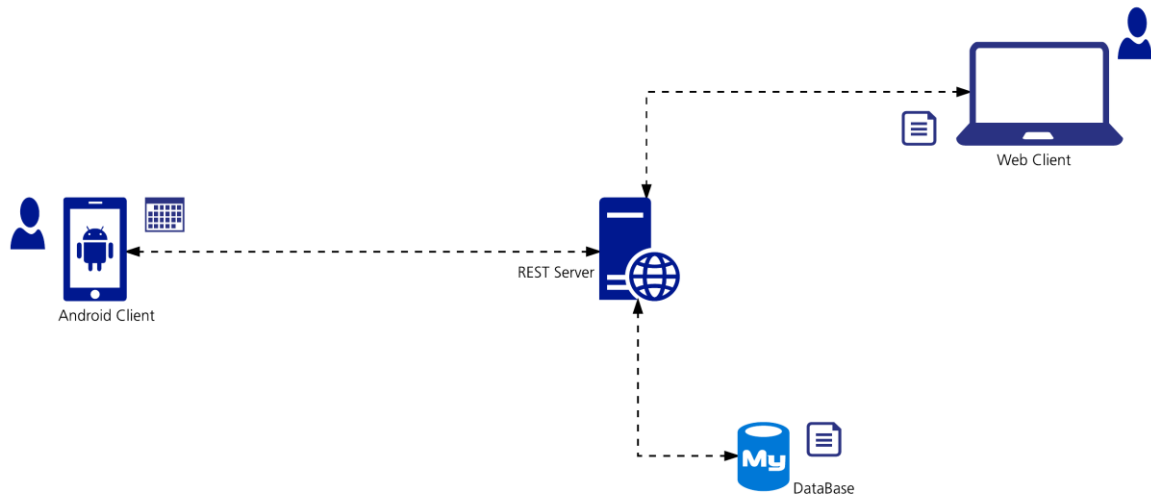


Figure 6 – Architecture

L'architecture choisie est de type Client-Serveur, utilisant le protocole HTTP pour les échanges et le format JSON pour les ressources.

Le client web envoie des requêtes http pour introduire des contenus (manifestations, vacances) ou pour les recevoir via l'API REST.

Le client Android interroge le serveur web qui lui répond avec des informations, qu'il met en forme et affiche.

Le serveur « écoute » en permanence les requêtes et renvoie des réponses http correspondantes aux clients. Il est de type REST (Représentation State Transfer) pour faciliter la représentation des ressources et leurs identifiants.

La base de données relationnelle, permet le stockage persistant des éléments (manifestations, vacances, utilisateurs, ...).

2.7 METHODOLOGIE

Le temps et les ressources à disposition étant relativement limités, il a fallu définir une méthodologie succincte. J'ai utilisé des principes connus du modèle incrémental itératif.

J'ai choisi de consacrer plus de temps pour l'implémentation et les tests, que pour la spécification et la conception.

Chaque étape de la planification représente une itération qui contenait elle-même sa phase d'analyse, de conception, d'implémentation et de test. Ce qui m'a permis d'intégrer petit à petit des fonctionnalités concrètes et factuelles, et de gérer les changements en cours de route.

Durant chaque itération j'ai essayé de développer le scénario complet entre le serveur et le client. Par exemple, l'affichage de la liste des manifestations sur le client Android a intégré la connexion du serveur avec la base de données, l'interrogation SQL, la servlet d'écoute de la requête, le formatage JSON, et l'Activity permettant d'affichage.

J'ai mis en œuvre une architecture centrale dès les premières itérations et par la suite les développements ont été pilotés par les cas d'utilisation.

Pendant la rédaction du code, aucune importance n'a été portée aux commentaires.

3. IMPLÉMENTATION

Dans cette phase il a fallu mettre en œuvre les scénarios définis en phase d'analyse.

J'ai commencé par mettre en place les différents environnements de développement. Ensuite j'ai réalisé les modules et fonctions nécessaires. Tout à la fin, j'ai saisi des informations et réalisé de tests complets sur l'ensemble.

Le design pattern MVC (Modèle-Vue-Contrôleur) a été utilisé pour la partie Serveur comme pour la partie Client. Cette solution de conception préconise la séparation des données, de la vue et du contrôleur, pour avoir une souplesse et un gain de temps lors de la maintenance/évolution de la plateforme.

Le modèle encapsule le cœur fonctionnel de l'application, le domaine logique.

Le contrôleur joue le rôle d'un aiguilleur qui reçoit les données et les transmet au modèle ou à la vue.

Les données sont envoyées, par le modèle, à la vue qui les présente à l'utilisateur.

Les modifications effectuées sur n'importe quelle couche de l'application n'affectent pas les autres couches. Les paragraphes suivants décrivent brièvement les composants de chaque partie.

3.1 SERVEUR REST

3.1.1 BASE DE DONNÉES

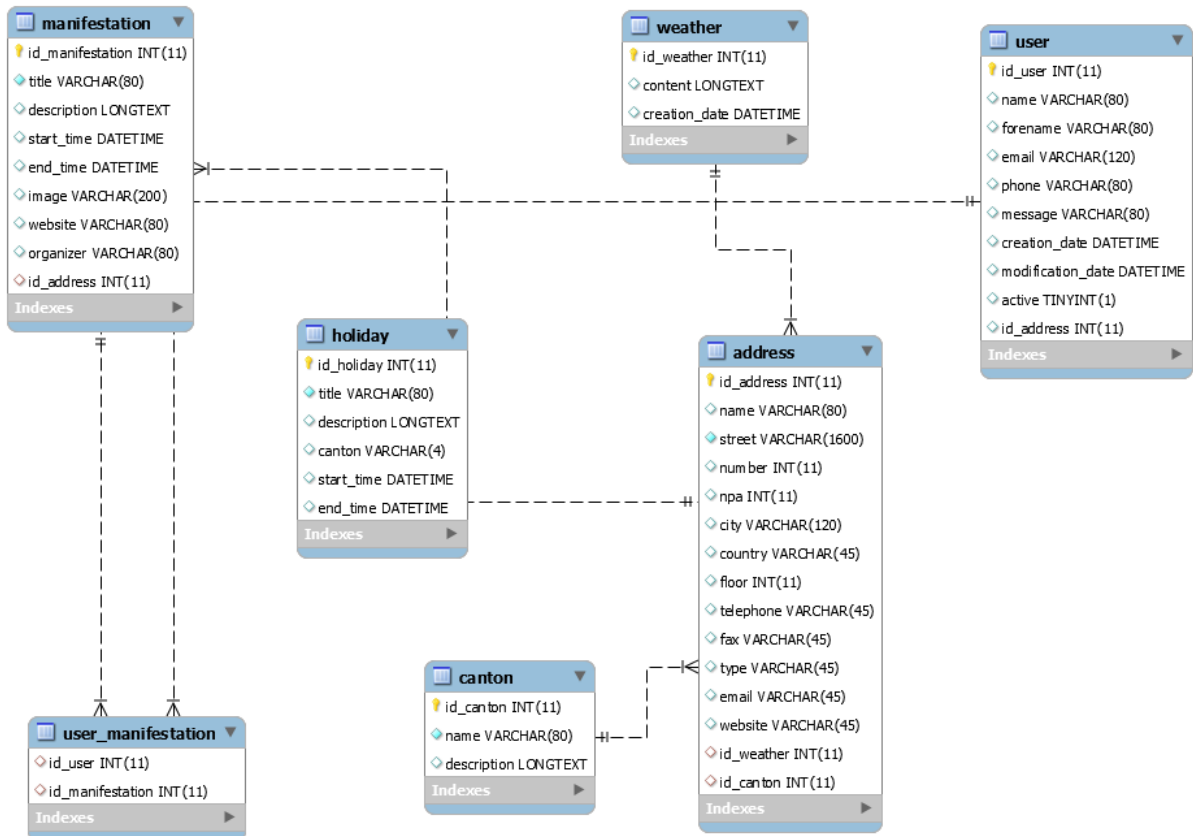


Figure 7 – Modèle logique de la base de données

Ci-dessus est présenté le modèle logique de la base de données utilisée.

Les tables manifestation, holiday, weather, user, address, canton permettent de représenter respectivement les manifestations, les vacances, la météo, les utilisateurs, les adresses et les cantons.

La table user_manifestation permet d'associer les utilisateurs et les manifestations (inscriptions).

La contrainte la plus importante a été de modéliser une base de données évolutive. L'introduction de nouvelles tables et/ou de nouvelles associations devant rester possible.

Il est à noter que pour la détermination de la manifestation la plus proche l'utilisation de requête spatiale postGIS aurait été plus efficace. En effet avec postGIS, on a une gestion simplifiée et efficace des données géographiques dans un SGBD. Les fonctionnalités spatiales de MySQL sont extrêmement limitées par rapport à postGIS. Et vu le temps imparti pour la réaliser ce Proof Of Concept, j'ai choisi de réaliser un simple calcul de distance avec la localisation courante de l'utilisateur et les adresses géocodées.

3.1.2 DIAGRAMME DE CLASSES

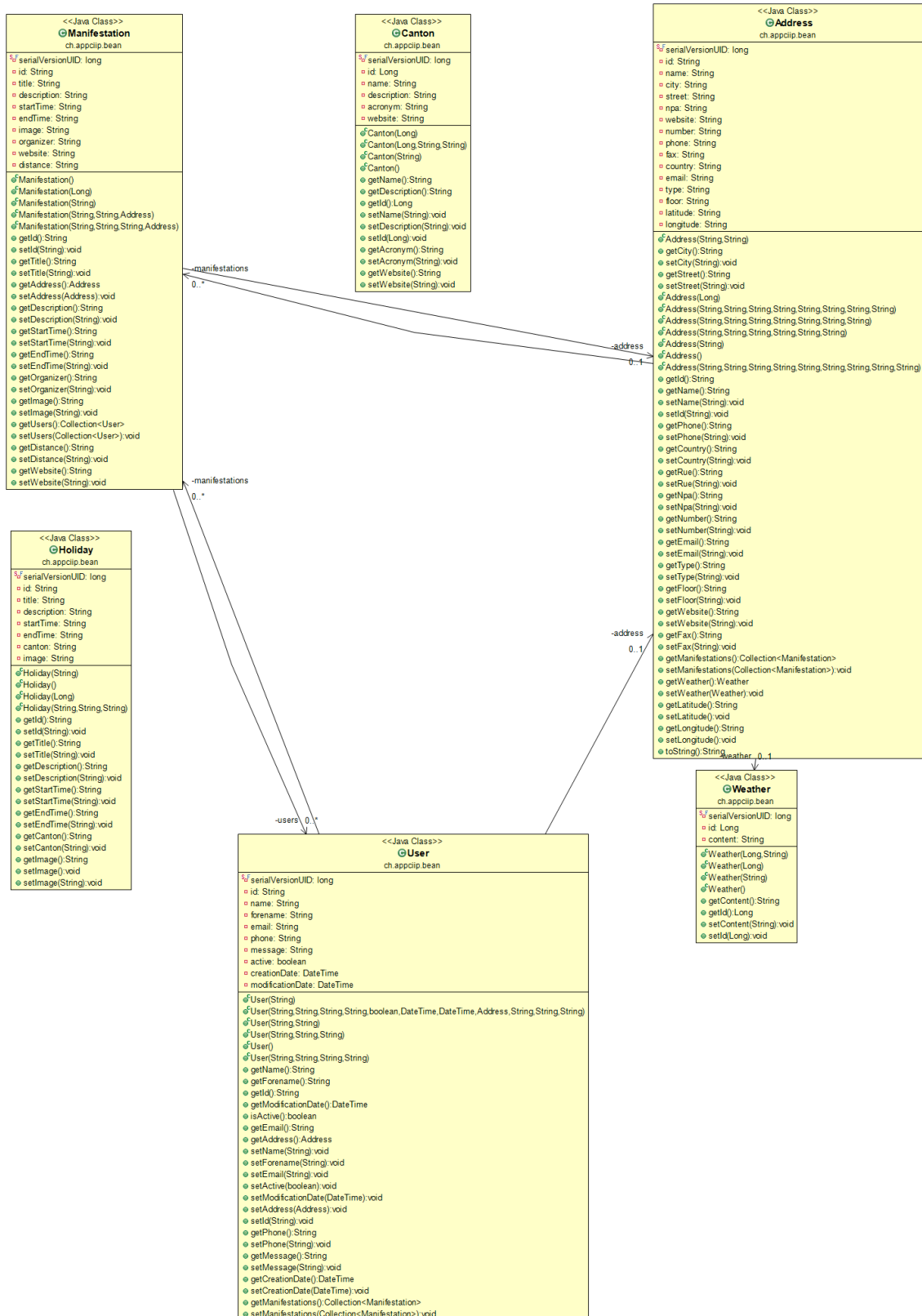


Figure 8 – Bean

Ci-dessus la représentation UML des java bean utilisés pour la partie Serveur. Les beans permettent d'implémenter le paradigme "write-once, run-anywhere" de Java. Ils permettent d'assurer une meilleure lisibilité et une meilleure maintenabilité du code.

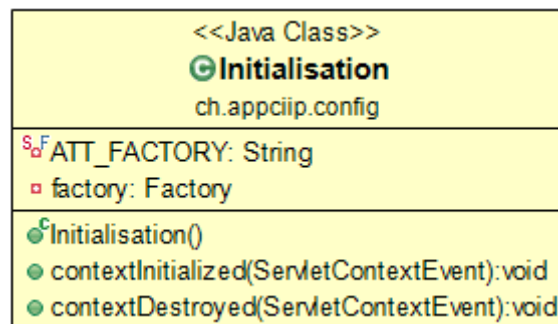


Figure 9 – Config

Il s'agit ici du Listener permettant de gérer le démarrage de la terminaison de l'application web. Il permet notamment la création d'une instance de la Factory qu'il met à disposition de l'application entière via l'objet ServletContext.

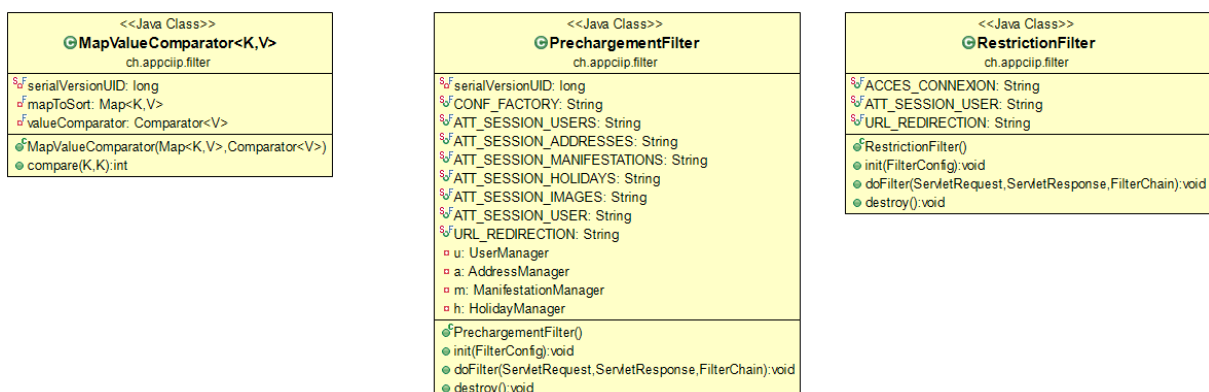


Figure 10 – Filter

Ci-dessus les classes « filtres » permettant la gestion des objets échangés lors des requêtes http.

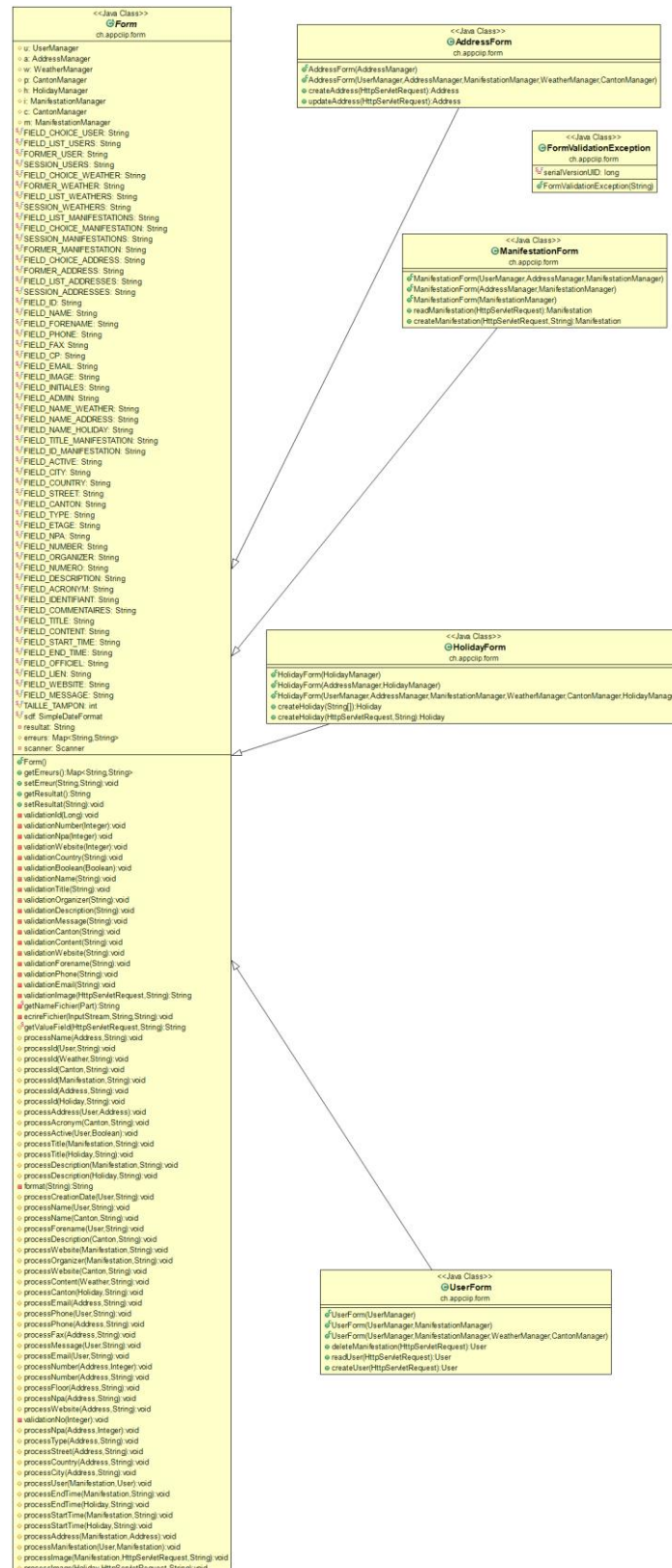


Figure 11 - Form

Ci-dessus les classes permettant de traiter les paramètres reçus dans les requêtes HTTP et de préparer les résultats à retourner. Par exemple, vérifier que le champ titre d'une manifestation est bien présent et possède plus de deux caractères, en cas d'erreur retourner un message via la servlet.

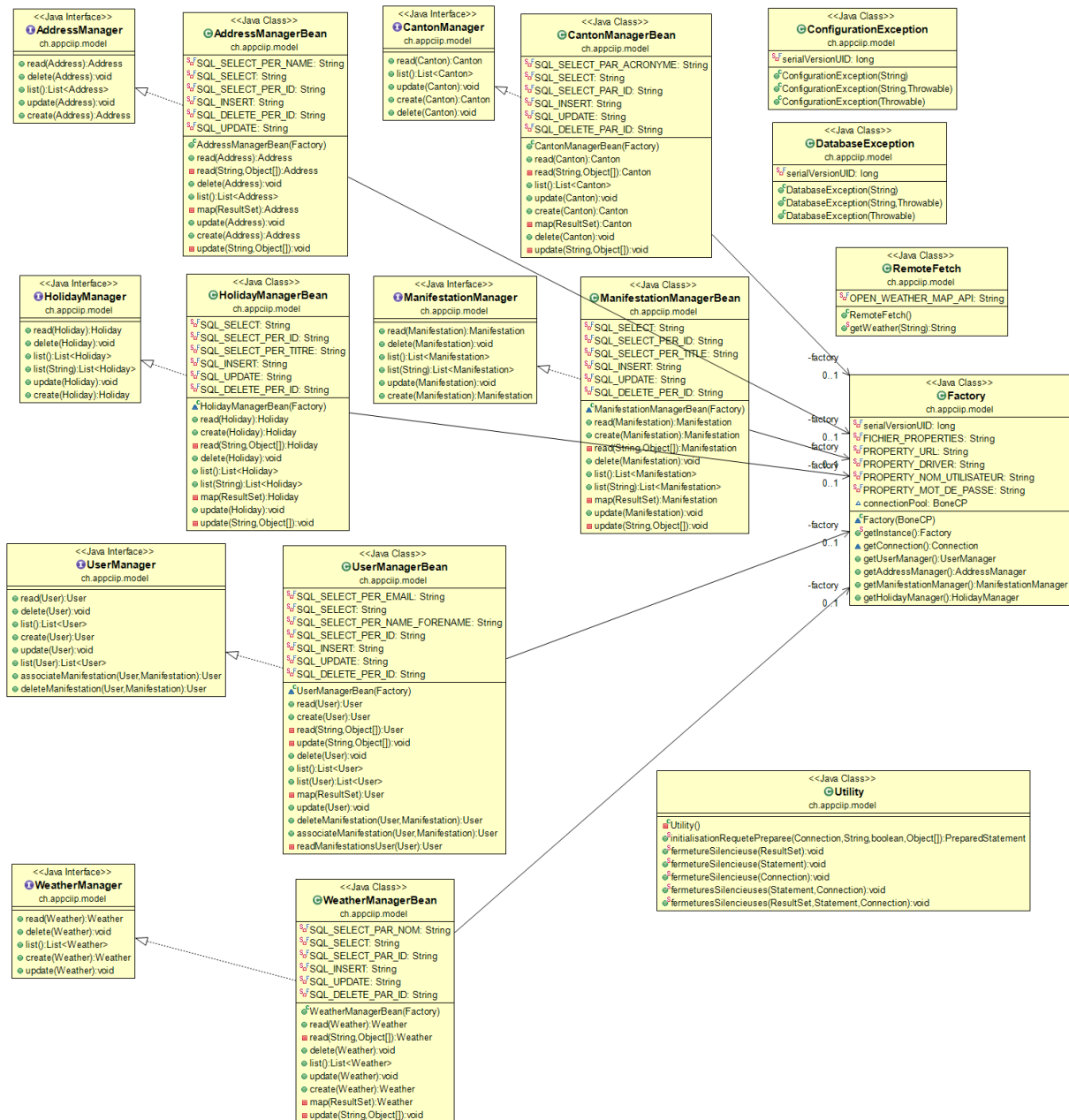


Figure 12 - Model

Ci-dessus l'ensemble des classes permettant la communication avec la base de données (gestions des pools de connexions, insertions, modifications, suppressions, lecture des résultats, fermeture des connexions, ...).

Les requêtes préparées y sont utilisées pour sécuriser les champs contre les injections SQL.

La classe RemoteFetch permet de récupérer la météo et la position (latitude, longitude) en passant en paramètre une adresse au format postal.

<pre> <<Java Class>> @CreateHoliday ch.appclip.servlet CONF_FACTORY: String ATT_HOLIDAY: String ATT_USER: String ATT_FORM: String ATT_SESSION_USER: String APPLICATION_USERS: String APPLICATION_INSTITUTIONS: String SESSION_INSTITUTIONS: String SESSION_USERS: String SESSION_HOLIDAYS: String SESSION_ADDRESSES: String serialVersionUID: long CHEMIN: String MSG: String SUCCES: String succes: boolean VUE_SUCCES: String VUE_FORM: String a: AddressManager m: HolidayManager CreateHoliday() init():void doGet(HttpServletRequest,HttpServletResponse):void doPost(HttpServletRequest,HttpServletResponse):void </pre>	<pre> <<Java Class>> @CreateManifestation ch.appclip.servlet CONF_FACTORY: String ATT_MANIFESTATION: String ATT_USER: String ATT_FORM: String ATT_SESSION_USER: String APPLICATION_USERS: String APPLICATION_INSTITUTIONS: String SESSION_INSTITUTIONS: String SESSION_USERS: String SESSION_MANIFESTATIONS: String SESSION_ADDRESSES: String serialVersionUID: long CHEMIN: String MSG: String SUCCES: String succes: boolean VUE_SUCCES: String VUE_FORM: String a: AddressManager m: ManifestationManager CreateManifestation() init():void doGet(HttpServletRequest,HttpServletResponse):void doPost(HttpServletRequest,HttpServletResponse):void </pre>	<pre> <<Java Class>> @Image ch.appclip.servlet serialVersionUID: long TAILLE_TAMPON: int Image() doGet(HttpServletRequest,HttpServletResponse):void </pre>
		<pre> <<Java Class>> @Register ch.appclip.servlet CONF_FACTORY: String ATT_USER: String ATT_FORM: String ATT_SESSION_USER: String APPLICATION_USERS: String APPLICATION_INSTITUTIONS: String SESSION_INSTITUTIONS: String SESSION_USERS: String SESSION_ADDRESSES: String serialVersionUID: long CHEMIN: String MSG: String SUCCES: String succes: boolean VUE_SUCCES: String VUE_FORM: String u: UserManager m: ManifestationManager Register() init():void doGet(HttpServletRequest,HttpServletResponse):void doPost(HttpServletRequest,HttpServletResponse):void </pre>

Figure 13 – Servlet

Ci-dessus les servlets utilisés pour l'aiguillage des requêtes. Il s'agit des portes d'entrée de l'application, recevant les demandes des clients. Elles transmettent les demandes au modèle pour traitement (par exemple une manifestation à créer), reçoivent des résultats du modèle (succès de la création, messages d'erreurs, ...) et les insèrent dans la vue à envoyer au client.

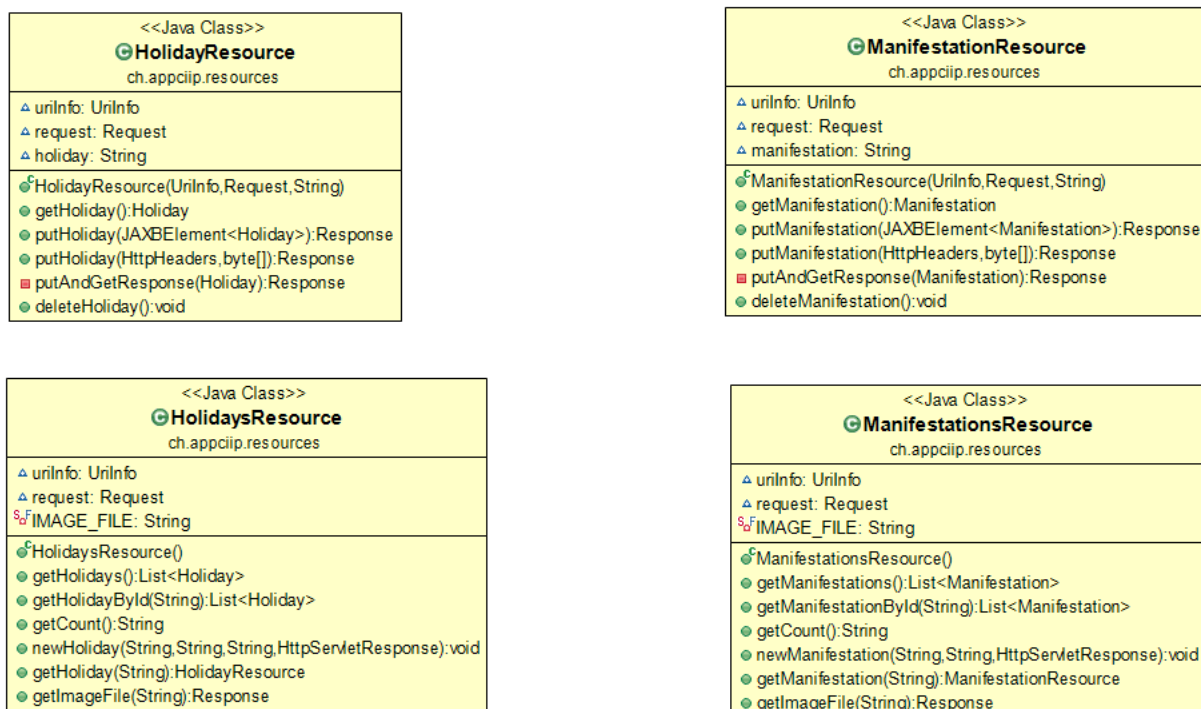


Figure 14 – Resources

Ci-dessus les classes permettant la gestion des ressources grâce à l'API REST. Elles reçoivent, interprètent les requêtes et permettent de présenter les résultats au format JSON par exemple.

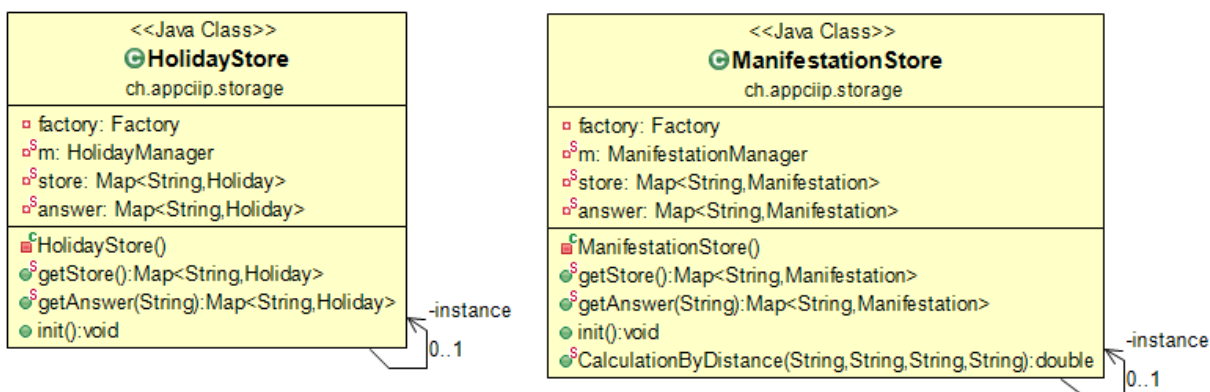


Figure 15 – Storage

Ci-dessus les classes permettant de gérer la persistance des données.

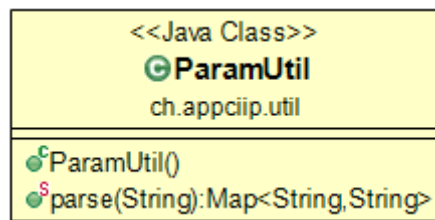


Figure 16 – Util

Ci-dessus une classe utilitaire permettant de traiter les URL. Par exemple récupérer les paramètres passés via la méthode http GET.

3.1.3 INTERFACES

J'ai utilisé des formulaires pour l'introduction des données. Pour ces formulaires j'ai dû respecter quelques principes du monde web. Par exemple, les champs obligatoires sont annoncés avec une étoile rouge. Un focus est positionné sur le champ rempli. Des messages sont envoyés en cas d'échecs précisant les erreurs.

APPCIIP

Soumettre une manifestation
Introduire une vacance

6 juin 2017

APPCIIP Server

APPCIIP CRUD Restful Webservice.

Création d'une manifestation

Formulaire de soumission d'une nouvelle manifestation.

Titre *

Semaine de la langue française et de la francophonie

Description *

La Semaine de la langue française et de la francophonie (SLEF) s'articule chaque année autour du 20 mars, Journée internationale de la Francophonie, en proposant de nombreuses manifestations placées sous le signe du français dans plusieurs pays et régions francophones.

Horaire de debut 16 March 2017 - 08:00

Horaire de fin 26 March 2017 - 17:00

Image

Parcourir... slff-title-2017-top-1000x440px.png
Selectionnez une image illustrative

Titre *

Semaine des Médias

Description *

La Semaine des médias à l'école en Suisse romande est une proposition pédagogique destinée à s'insérer dans le programme habituel des cours. L'ambition est de consacrer une modeste fraction de l'horaire aux médias, aux images, au numérique. C'est surtout l'occasion de mettre en œuvre les objectifs MITIC du Plan d'études romand. Du 27 au 31 mars 2017, la 14e édition se déclinait sur le thème : "Toujours connectés ?".

Horaire de debut 27 March 2017 - 08:00

Horaire de fin 31 March 2017 - 17:00

Image

Parcourir... showPhoto.png
Selectionnez une image illustrative

Rue

Faubourg de l'Hôpital

March 2017						
Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8
Today						

Titre *

Semaine des Médias

Description *

La Semaine des médias à l'école en Suisse romande est une proposition pédagogique destinée à s'insérer dans le programme habituel des cours. L'ambition est de consacrer une modeste fraction de l'horaire aux médias, aux images, au numérique. C'est surtout l'occasion de mettre en œuvre les objectifs MITIC du Plan d'études romand. Du 27 au 31 mars 2017, la 14e édition se déclinait sur le thème : "Toujours connectés ?".

Horaire de debut 27 March 2017 - 08:00

Horaire de fin 31 March 2017 - 17:00

Image

Parcourir... showPhoto.png
Selectionnez une image illustrative

Rue

Faubourg de l'Hôpital

27 March 2017				
0:00	1:00	2:00	3:00	
4:00	5:00	6:00	7:00	
8:00	9:00	10:00	11:00	
12:00	13:00	14:00	15:00	
16:00	17:00	18:00	19:00	
20:00	21:00	22:00	23:00	
Today				

Rue
Faubourg de l'Hôpital

No. Rue
43

Npa
2002

Ville
Neuchâtel

Pays
Suisse

Site web
www.siff.ch

Organisateur
CIIP

Valider Remettre à zéro

© CIIP 2017.

Figure 17 - Création d'une manifestation

Ci-dessus l'interface web de création d'une manifestation utilisant un date/hour picker pour l'introduction des horaires.

Création d'une manifestation

Nouvelle manifestation.

Succès de la création de la manifestation.

Image : showPhoto.png Voir

SEMAINE DES MÉDIAS

La Semaine des médias à l'école en Suisse romande est une proposition pédagogique destinée à s'insérer dans le programme habituel des cours. L'ambition est de consacrer une modeste fraction de l'horaire aux médias, aux images, au numérique. C'est surtout l'occasion de mettre en œuvre les objectifs MITIC du Plan d'études romand. Du 27 au 31 mars 2017, la 14e édition se déroulait sur le thème : "Toujours connectés ?".

lundi 27 mars 2017 08:00

vendredi 31 mars 2017 05:00

Faubourg de l'Hôpital 68, 2002 Neuchâtel

www.semainedesmedias.tv

CIIP

Figure 18 – Récapitulatif manifestation

Ci-dessus un exemple de résultat html récapitulatif lors de la création d'une manifestation.

```

42:
  address:
    city: "Neuchâtel"
    country: "Suisse"
    email:
    fax:
    floor: "0"
    id: "13"
    latitude: "46.99"
    longitude: "6.93"
    manifestations:
      name: "FAUBOURG DE L'HÔPITAL 43"
      npa: "2002"
      number: "43"
      phone:
      rue: "Faubourg de l'Hôpital "
      street: "Faubourg de l'Hôpital "
      type:
  weather:
    content:
      [{"dt":1496820000,"coord":{"lon":6.93,"lat":46.99},"visibility":10000,"weather":[{"icon":"04d","description":"broken clouds","main":"Clouds","id":803},"name":"Neuchatel","cod":200,"main":{"temp":11.31,"temp_min":6,"humidity":58,"pressure":1020,"temp_max":13},"clouds":{"all":75},"id":2659496,"sys":{"country":"CH","sunrise":1496806687,"sunset":1496863483,"id":6815,"type":1,"message":0.0027},"base":{"stations"},"wind":{"deg":220,"speed":7.2}}]
    id:
    website:
  description: "La Semaine de la langue française et de la francophonie (SLFF) s'articule chaque année autour du 20 mars, Journée internationale de la Francophonie, en proposant de nombreuses manifestations placées sous le si
    du français dans plusieurs pays et régions francophones."
  distance:
  endTime: "dimanche 26 mars 2017 05:00"
  id: "43"
  image: "siff-title-2017-top-1000x440px.png"
  organizer: "CIIIP"
  startTime: "jeudi 16 mars 2017 08:00"
  title: "SEMAINE DE LA LANGUE FRANCAISE ET DE LA FRANCOPHONIE"
  users:
  website: "www.siff.ch"

```

Figure 19 – JSON Manifestation

Ci-dessus un exemple de résultat JSON lors de la création d'une manifestation.

APPCIIP
Soumettre une manifestation
Introduire une vacance
7 juin 2017

APPCIIP Server

APPCIIP CRUD Restful Webservice.

Création d'une vacance

Formulaire d'introduction d'une nouvelle vacance.

Titre *

Description *

Horaires de début 03 April 2017 - 08:00 ✕

Horaires de fin 17 April 2017 - 23:55 ✕

Canton
NE

Valider Remettre à zéro

© CIIP 2017.

Figure 20 - Création d'une vacances

Ci-dessus l'interface web de création d'une vacances.

Création d'une vacances scolaires
Nouvelle vacances scolaires.

Succès de la création de la vacances.

Image : JU.png Voir

PRINTEMPS 2017 JU

Vacances de printemps.

lundi 03 avr. 2017
lundi 17 avr. 2017
JU

© CIIP 2017.

Figure 21 - Récapitulatif vacances

Ci-dessus un exemple de résultat html récapitulatif lors de la création d'une vacances.

```
▼ 5:
  canton: "VD"
  description: ""
  endTime: ""
  id: "13"
  image: "VD.png"
  startTime: "lundi 21 août 2017"
  title: "RENTRÉE SCOLAIRE VD"
▼ 6:
  canton: "GE"
  description: "Vacances de printemps."
  endTime: "lundi 17 avr. 2017"
  id: "25"
  image: "GE.png"
  startTime: "lundi 03 avr. 2017"
  title: "PRINTEMPS 2017 GE"
▼ 7:
  canton: "VD"
  description: ""
  endTime: ""
  id: "14"
  image: "VD.png"
  startTime: "lundi 18 sept. 2017"
  title: "LUNDI DU JEÛNE FÉDÉRAL VD"
▼ 8:
  canton: "JU"
  description: "Vacances de printemps."
  endTime: "lundi 17 avr. 2017"
  id: "26"
  image: "JU.png"
  startTime: "lundi 03 avr. 2017"
  title: "PRINTEMPS 2017 JU"
```

Figure 22 – JSON Vacances

Ci-dessus un exemple de résultat JSON lors de la création d'une vacances.

3.2 CLIENT ANDROID

3.2.1 DIAGRAMME DE CLASSES

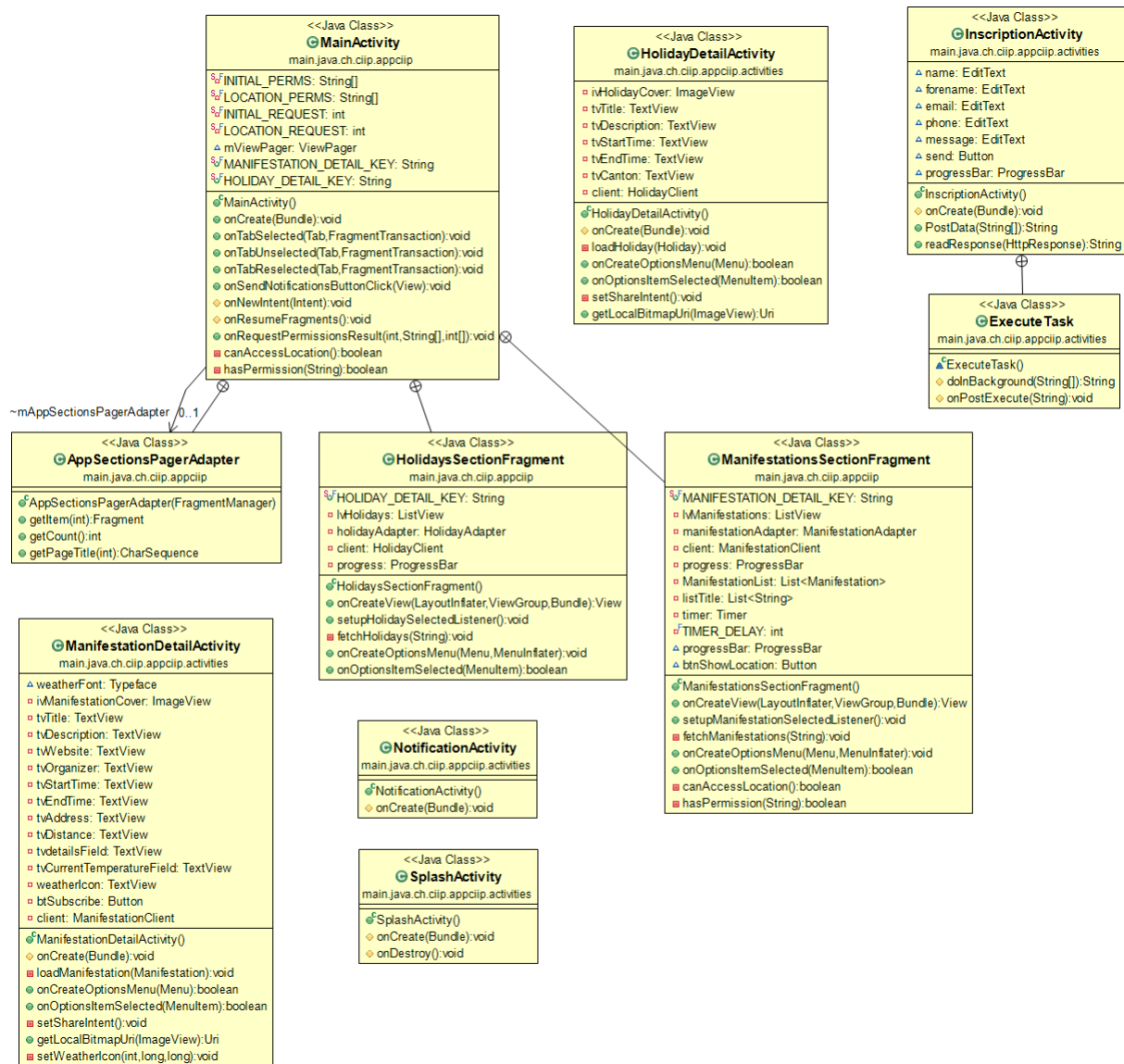


Figure 23 - Activities

Ci-dessus les différentes activités utilisées pour mon application Android.

L'activité principale utilise des Fragments facilitant l'affichage sur le même écran d'activités différentes (liste de manifestations et liste de vacances).

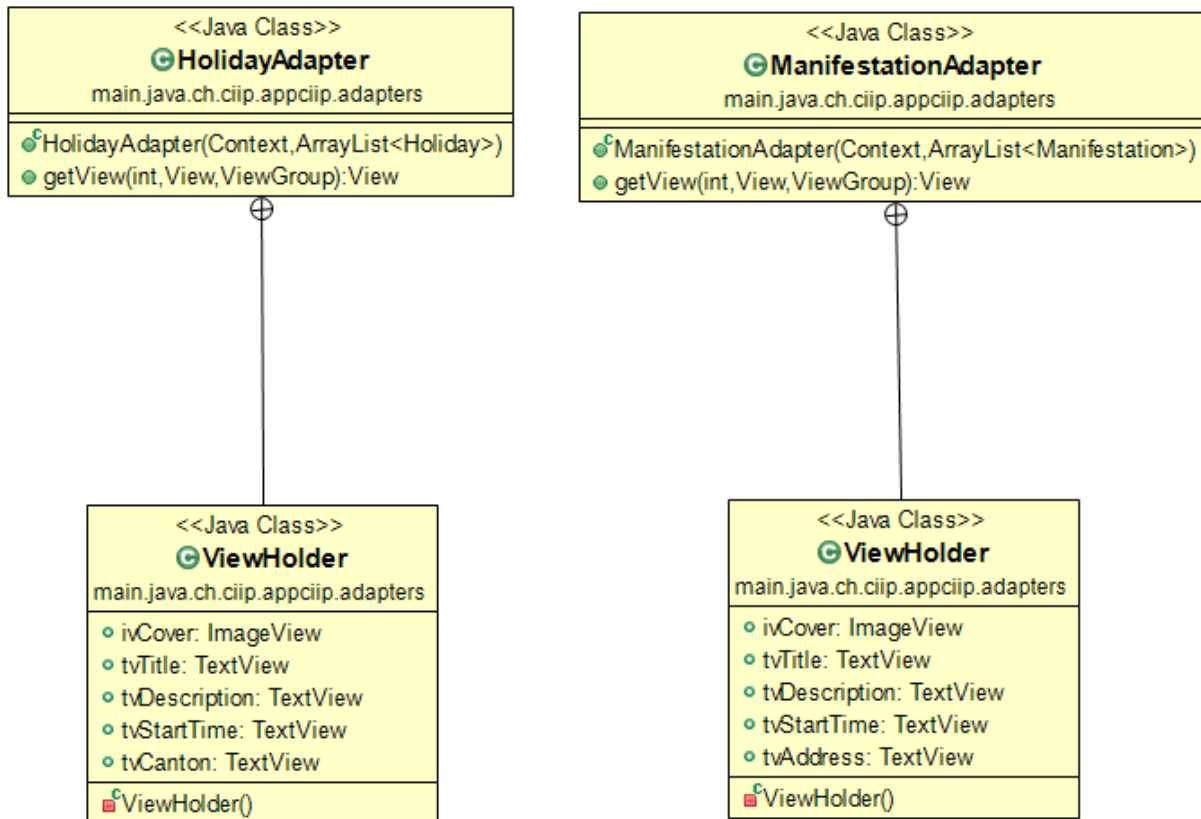


Figure 24 - Adapters

Ci-dessus la liste d'adapters utilisés dans mon application. L'adapter agit comme un pont entre un AdapterView et les données sous-jacentes pour cette vue. Il m'a permis d'accéder aux éléments des listes et de créer à la volée une vue pour chacun d'entre eux.

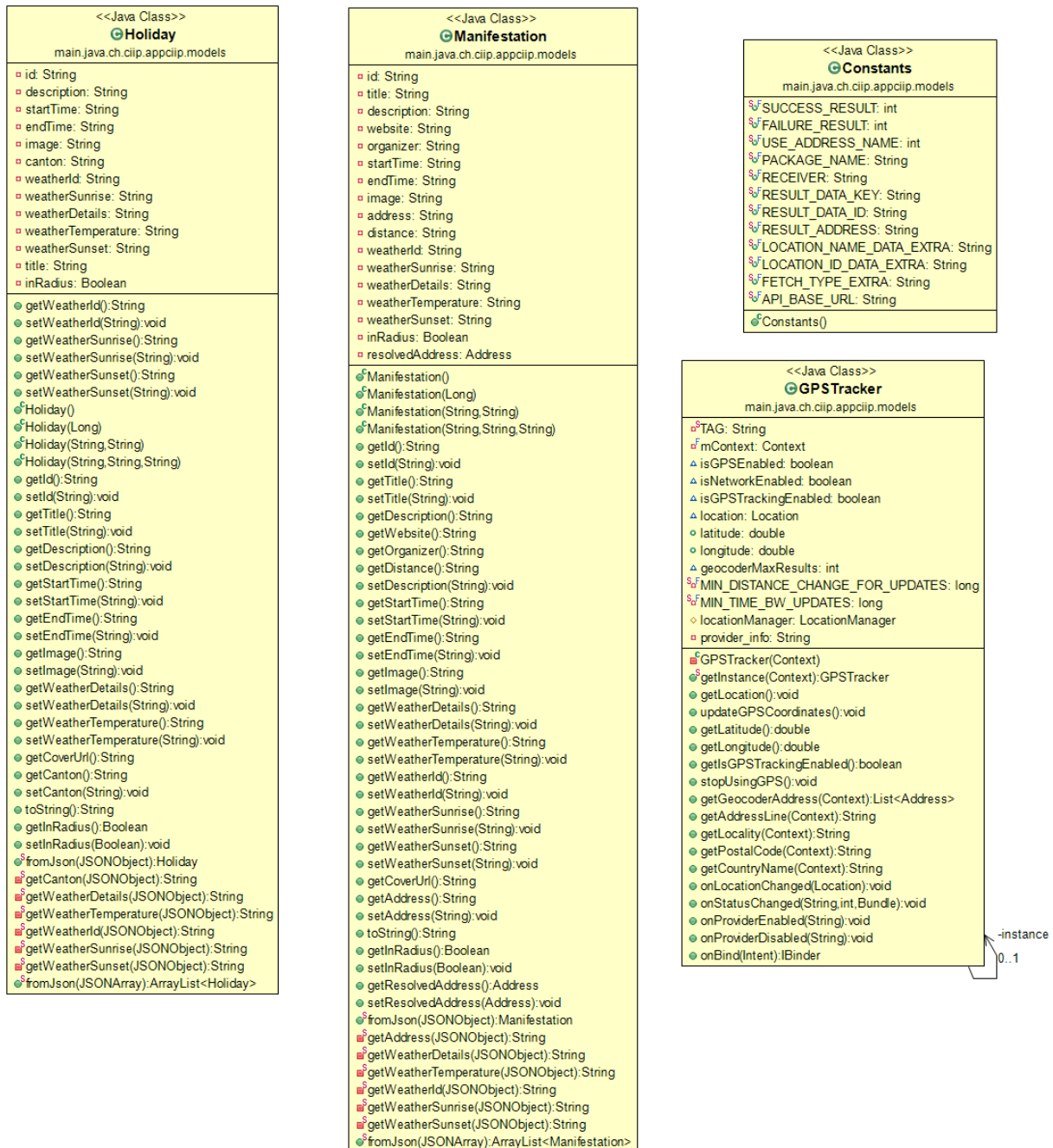


Figure 25 – Models

Ci-dessus les classes utilisées pour la gestion et le traitement des données. Elles représentent la couche métier » de l'application. Les classes Holiday et Manifestation permettent de parser les manifestations et les vacances reçues au format JSON. La classe GPSTracker permet de gérer la position courante du mobile et la classe Constants permet de gérer les attributs partagés (par exemple l'adresse IP du serveur).

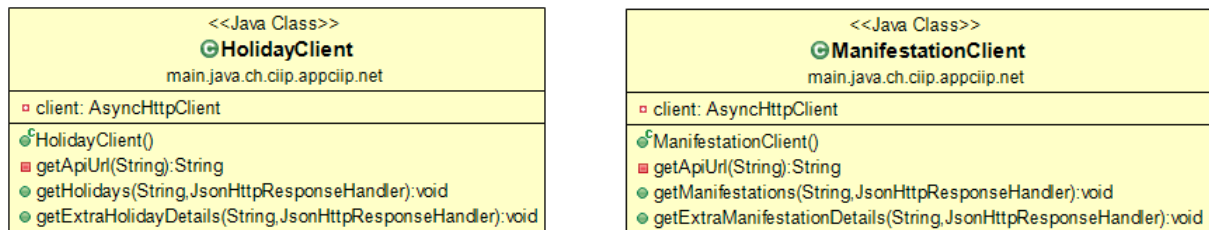


Figure 26 - Net

Ci-dessus les classes permettant de se connecter à l'API du serveur REST.

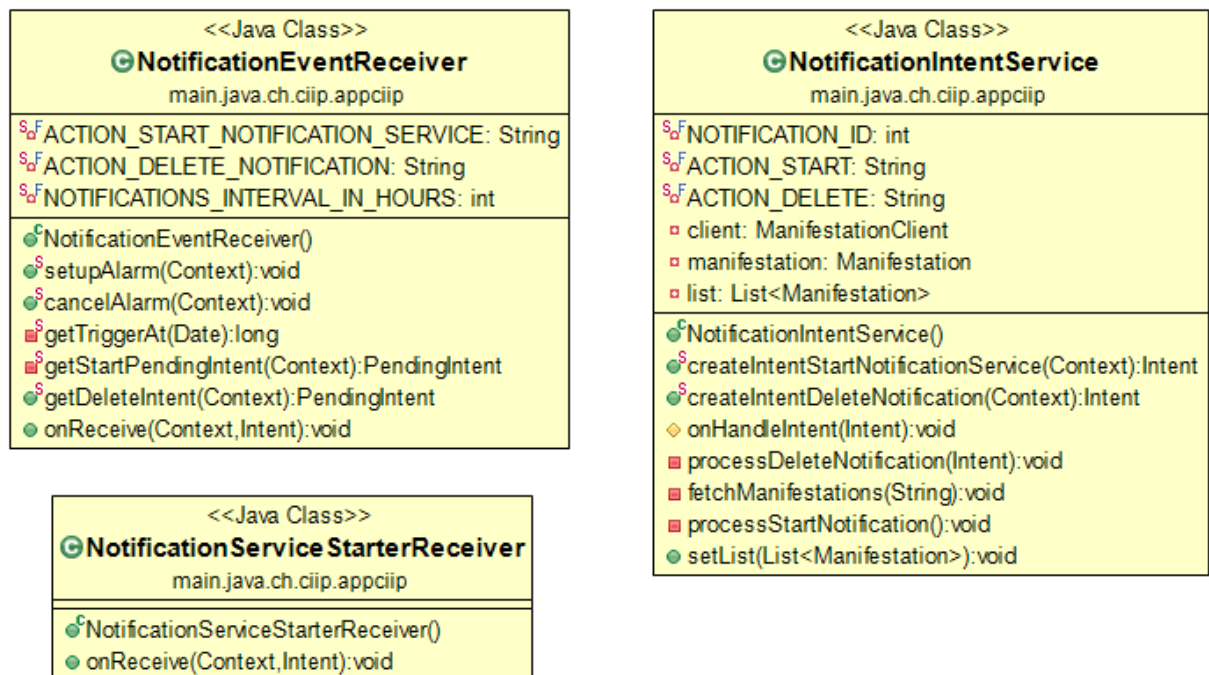


Figure 27 - Notifications

Ci-dessus l'ensemble des classes utilisées pour la gestion des notifications.

3.2.2 INTERFACES



CONFÉRENCE INTERCANTONALE
DE L'INSTRUCTION PUBLIQUE DE
LA SUISSE ROMANDE ET DU TESSIN

Figure 28 - SplashScreen

Ci-dessus la fenêtre de démarrage de l'application.



Figure 29 - Vue Principale

Ci-dessus la fenêtre principale de l'application.



Figure 30 - Manifestation

Ci-dessus la fenêtre d'affichage d'une manifestation.

The screenshot shows an Android application interface for registration. At the top, a black status bar displays various icons and the time 12:17. Below it, a dark header bar contains the text 'SEMAINE DES MÉDIAS'. The main content area has a light gray background. The title 'Inscription' is centered in a large, dark font. Below the title, there are five input fields: 'Teguia' (with a blue underline and a cursor), 'Prenom', 'Email', 'Telephone', and 'Votre Message'. At the bottom, there is a gray button labeled 'ENVOYER'.

Figure 31 - Inscription

Ci-dessus la fenêtre d'inscription à une manifestation.

Un message de confirmation de l'inscription est retourné sous forme de Toast Android, après quoi l'application retourne sur la vue de la Manifestation.



Figure 32 - Recherche

Ci-dessus une illustration de la recherche par mots clés.

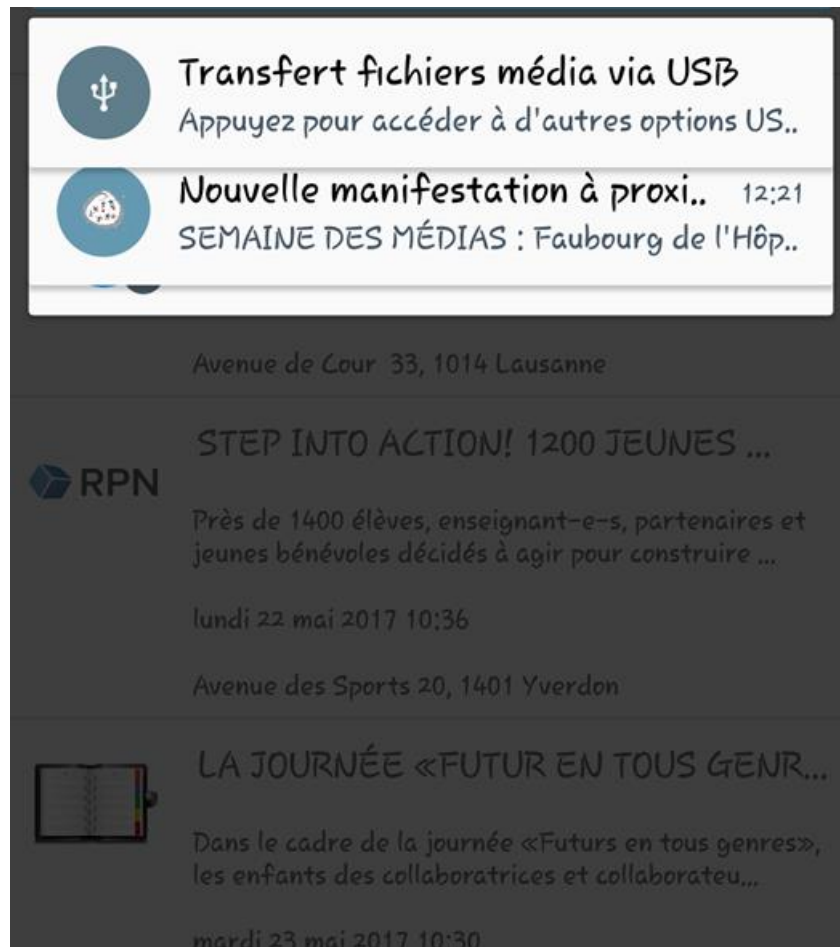


Figure 33 - Notification

Ci-dessus une illustration de notification. En déroulant les notifications on peut voir la proposition d'une manifestation à proximité. En cliquant sur la notification on affiche les détails de celle-ci.



Figure 34 - Vacances

Ci-dessus la fenêtre affichant la liste de Vacances.



Figure 35 - Vacance

Ci-dessus la fenêtre affichant les détails d'une vacance.

4. TESTS, MAINTENANCE ET DOCUMENTATION

Dans cette partie il a fallu vérifier que l'ensemble des parties de l'application cliente et du serveur fonctionnent correctement. Pour les anomalies découvertes il a fallu trouver des ajustements.

Cette phase a été particulièrement importante car ne disposant pas toujours d'assez de temps pour pouvoir effectuer des tests de non-régression au terme de chaque itération. Pour effectuer chaque test il fallait fixer l'adresse IP du serveur dans le code, démarrer le serveur Tomcat et enfin démarrer l'application mobile.

L'évaluation a essentiellement porté sur les exigences fonctionnelles. J'ai testé l'introduction de données pour les manifestations et les vacances, via le formulaire web. L'affichage des données créées, l'exactitude des informations (titre, dates, distances...), la réception de notifications,

En parallèle de ces différents tests, j'ai rédigé le présent rapport.

5. PROLONGEMENT ET AMELIORATIONS

Ce projet d'approfondissement ouvre des perspectives d'améliorations intéressantes pouvant déboucher sur une application finalisée et déployée.

Un certain nombre de primitives n'ont pas pu être traitée vu le temps imparti et un prolongement pourrait être envisagé sur les fonctionnalités et l'utilisation de l'application. Les uses-cases développés pourraient être améliorés et de nouveaux scénarios implémentés en extension de ce projet d'approfondissement.

Depuis l'interface web du serveur les administrateurs devraient pouvoir en plus de créer, modifier ou supprimer les manifestations et les vacances. On devrait pouvoir choisir pour une manifestation donnée, un artiste proposé par la CIIP.

La traduction des contenus dans d'autres langues (italien, allemand, anglais, romanche 😊) pourrait être réalisée.

On pourrait donner la possibilité à l'utilisateur de trier les manifestations par distance, de partager une manifestation, de configurer les notifications (intervalles de réception, ...), d'administrer les paramètres de l'application. On pourrait améliorer la gestion des erreurs, par exemple afficher un message en cas de non disponibilité du serveur.

La gestion des informations introduites dans la base de données devrait être réservée exclusivement aux collaborateurs de la CIIP. Ceux-ci seraient dès lors responsables de l'intégrité sémantique des données.

Il faudrait donc introduire la notion de profil utilisateur et développer leur gestion (inscription, connexion/déconnexion, modification de profil, suppression de profil).

Au niveau de l'application Android, ces profils devrait donner la possibilité de mettre en favoris une manifestation et de donner des appréciations. Au moment de l'inscription d'un utilisateur, les conditions de traitement de données devraient lui être rappeler.

La collecte de données via l'application pourrait ouvrir un champ d'étude intéressant quant aux comportements utilisateurs. Un historique des « recherches » pourrait être produit à partir des requêtes récoltées afin de réaliser des mécanismes de recommandations intelligentes de manifestations (Machine Learning). Les paramètres de recherche antérieures, de géolocalisations, d'intérêts, de domaines, de dates seraient agrégés pour fournir des propositions adéquates à chaque utilisateur.

Il faudrait aussi choisir et paramétrer un outil de tracking pour pouvoir récupérer les statistiques de consultation et les comportements des utilisateurs (nombre de visiteurs, nombre de visites, temps passé sur l'application, localisation des visiteurs, types d'appareils utilisés, volumes de recherche, ...).

Cette analyse de l'audience permettrait d'effectuer un suivi, et d'améliorer l'efficacité et la facilité d'utilisation.

Il faudrait définir une stratégie de communication et de présence en ligne pour l'application. Une analyse des couts (développement de la version iPhone, budget promotion et référencement, ...), et un plan de maintenance/mise à jour. Il faudrait tester et évaluer les qualités systémiques et exigences non-fonctionnelles du système. A savoir les performances, la sécurité, la facilité d'administration, la facilité d'utilisation, les temps de réponse, la capacité, la scalabilité, la disponibilité, l'évolutivité et la fiabilité.

Suite à ces améliorations la mise en production pourrait être envisager. Il faudrait des lors paramétrer et déployer le serveur REST et publier l'application sur Google Play.

6. CONCLUSION

Il a été question pour moi durant ce projet d'approfondissement de développer le prototype POC (Proof Of Concept) d'une application de gestion des manifestations et des vacances scolaires cantonales.

Pour ce faire j'ai choisi de réaliser une architecture de type Client-Serveur et j'ai découpé ma planification en trois grandes phases.

Les séances d'échanges et d'analyses avec madame Fatemi et monsieur Maradan m'ont permis de clarifier les objectifs, les étapes, et les travaux à effectuer.

J'ai fait des choix technologiques pour construire un système évolutif, interopérable et extensible. Et des choix graphiques pour maintenir une cohérence visuelle.

Au final, à partir de l'interface du serveur web un utilisateur peut introduire des manifestations ou des vacances qui seront ensuite disponibles via un API REST. En « interrogeant » le serveur, l'application Android peut afficher des vacances scolaires ainsi qu'une liste de manifestations dans le domaine de l'éducation. L'utilisateur peut rechercher une manifestation, consulter ses détails (date de début, de fin, météo, site web contenant le programmes étayé, ...) et s'y inscrire. Il reçoit de manière courante une notification lui proposant une manifestation proche de sa position géographique.

Le prototype délivré permet de se rendre compte de la faisabilité mais aussi des améliorations possibles sur l'application. En extension, l'application pourrait informer de manière proactive les utilisateurs sur les différentes activités en tenant compte de leur planification, de leur géolocalisation, et de leur intérêt.

Ce projet intéressant m'a permis de mettre en œuvre les notions de programmation orientée objet, de développement d'applications mobiles et de méthodologie gestion de projet vues dans mon cursus Master HES. J'ai ainsi pu approfondir mes connaissances dans le domaine des technologies de l'information et de la communication.

Lausanne, 09 Juin, 2017.

Gaël Tegua Tegua

7. RÉFÉRENCES

- <https://developer.android.com/index.html>
- <https://openclassrooms.com/courses/creez-votre-application-web-avec-java-ee/introduction-au-java-ee>
- <https://github.com/martykan/forecastie>
- <https://github.com/codepath/android-booksearch-demo>
- <http://alvinalexander.com/android/android-asynctask-http-client-rest-example-tutorial>

Cours MSE:

- T-MobOp - Systèmes d'exploitation mobiles et applications
- MAGHY - Méthodologies Agiles et Hybrides