

Rapport projet I4 : MapMyCalendar

Aloui Slim

Dufaÿ Sébastien

Marleix Mathieu

Noël Romain



ESIEE
PARIS

Présentation du cadre de travail

Ce projet, au départ réalisé dans le cadre du concours ATOS IT Challenge, a été intégré au sein d'un projet d'étude d'ESIEE Paris en quatrième année de majeure informatique.

Ce projet d'application vise la mise en application et l'approfondissement des connaissances en langage java et système embarqué Android.

Résumé

L'accès aux ressources personnelles d'un élève d'ESIEE Paris est pour le moment une opération assez fastidieuse, nécessitant l'accès à un poste informatique ou une utilisation d'outils de société externe. L'application intitulée MapMyCalendar, vise à simplifier la consultation quotidienne d'information par le biais d'une application Android proposant un affichage de l'agenda d'un membre d'ESIEE Paris ainsi que l'affichage des horaires de transports environnant ESIEE Paris.

Table des matières

Présentation du cadre de travail	2
Résumé.....	2
Introduction.....	4
I. Environnement de travail	4
II. Etude de solutions	4
III. Objectifs du projet.....	5
IV. Android.....	8
1. Base du système	8
2. Activité.....	8
3. XML.....	8
V. Intégration d'un agenda dynamique.....	9
1. Récupération des données.....	9
2. Traitement des données	9
3. Affichage des données	10
RSSActivity	10
CalendarActivity	13
4. Ecran de préférence	13
5. Menu	16
6. Diagramme d'interaction des classes.....	16
VI. Intégration d'horaires de transport	17
1. Récupération des données.....	17
2. Affichage des données	17
3. Diagramme d'interaction des classes.....	18
VII. Intégration d'une partie géolocalisation.....	20
Conclusion	21
Annexes	22
Cahier des charges.....	22

Introduction

L'application MapMycalendar se décline en trois parties principales :

- La synchronisation de notre agenda avec l'agenda d'ESIEE Paris,
- La récupération des données RATP,
- La géolocalisation.

Nous allons commencer par décrire la partie théorique du projet. Ensuite, nous aborderons les différentes parties techniques : La synchronisation de l'agenda, sa mise en œuvre graphique ainsi que la gestion des flux RSS nécessaire à cette synchronisation. Nous introduirons ensuite la récupération des données RER et bus proches de l'ESIEE nécessitant un traitement plus simple des flux, ainsi que leur aspect graphique. En suivant, nous expliquerons comment nous avons ajouté la géolocalisation et l'API Google Maps. Nous concluons sur la mise en commun de ces parties pour former notre application.

I. Environnement de travail

Le développement d'applications pour la plateforme Android nécessite l'utilisation de kit de développement spécifique à cette plateforme. Le SDK (Software Development Kit) proposé par Google favorisant l'utilisation de l'IDE (Integrated Development Environment) Eclipse, ayant déjà développé avec celui-ci et estimant avoir un accès à une documentation plus importante puisque majoritairement utilisé par la communauté, nous avons décidé d'utiliser Eclipse en lieu et place d'un autre IDE (notons que le développement au travers d'un autre IDE est possible via Netbeans et le plugin nbandroid).

Pour permettre un développement collaboratif, un dépôt subversion a été effectué sur un hébergement Google Code. Le projet est donc disponible sous une licence Apache 2.

II. Etude de solutions

Comme précédemment énoncé, il est déjà possible pour un élève d'ESIEE Paris de récupérer son agenda grâce à un précédent projet effectué au sein d'ESIEE Paris. Ce projet nommé Hélium et sa

documentation est disponible à l'adresse suivante :
https://intra.esiee.fr/actu.php?id_info=3297&id_rub=18&id_srub=59

Celui-ci renvoi un flux ICS/iCal en fonction de ressources passées en argument. La réutilisation de cet outil aurait pu être profitable, mais n'ayant eu le code source que tardivement, nous n'avons pas réutilisé cet outil. Il est nécessaire de comprendre que le fonctionnement de cet outil est basé sur l'utilisation de l'API de la gestion d'emploi du temps d'ESIEE Paris, à savoir ADE Campus. Lors du développement d'outils basé sur cet API, il est nécessaire de porter attention au nombre de requêtes possiblement émises par l'application, le système actuel supportant une charge de requêtes limitée.

Nous avons donc préféré une alternative utilisant une fonctionnalité déjà implémentée au sein de l'emploi du temps général, à savoir l'exportation des ressources sous forme de flux RSS. Cette solution présente à la fois des avantages et des inconvénients. Contrairement à la première solution, une seule requête étant émise pour récupérer les ressources, le risque de surcharge est très faible. L'utilisation d'une fonctionnalité existante permet à l'application de ne pas avoir à être mise à jour annuellement en fonction des modifications dans la gestion des systèmes d'ESIEE Paris.

Cependant il est impossible, via cette solution, de récupérer les ressources des jours précédant la date de consultation du flux. Il est donc impossible de bénéficier d'une consultation des événements a posteriori. La consultation d'un agenda n'étant pas souvent effectuée a posteriori, mais plutôt dans un but de consultation des événements futur, nous avons conclu que ce désavantage n'était pas trop handicapant dans le développement de notre application.

III. Objectifs du projet

Faciliter l'obtention d'informations clés pour les membres d'ESIEE Paris reste l'objectif majeur de ce projet. Il existe déjà des outils, mais ils sont soit réservés aux membres initiés soit disponibles seulement dans les locaux d'ESIEE Paris. Pour simplifier l'obtention de l'agenda, une idée a été proposée :

Il existe déjà plusieurs outils utilisés via l'intranet de l'ESIEE, à savoir :

- 'Ma scolarité, mes notes, mes absences' <https://extra.esiee.fr/intranet/scolarite/etudiants/>
- 'ADE Campus' <http://planif.esiee.fr:8080/ade/custom/esiee/easyMyPlanning.jsp>

Pour obtenir son agenda personnalisé, l'étudiant doit rentrer ses ressources via une liste déroulante

dans ADE Campus, et à chaque semestre, changer ses ressources (Ci-dessous l’arbre des ressources).

The screenshot displays the MapMyCalendar interface. The browser address bar is highlighted with a red box, showing the URL: `planif.esiee.fr:8080/ade/custom/modules/plannings/direct_planning.jsp?resources=2636,2637,2638,2757,1288,1346,1347,1350,2063,2571,263`. On the left side, a tree view of resources is highlighted with a red box, showing the following structure:

- Classes-Groupes
 - ESIEE Engineering 2011-2012
 - I1
 - I2
 - I3
 - BFM
 - BFR
 - I4
 - Majeures période 1
 - A4
 - F4
 - I1
 - 1a
 - 1b
 - IF4(1)
 - Listes IF4
 - liste 2a
 - liste IF4
 - liste IF4 1
 - liste IF4 2
 - liste IF4 3
 - liste IF4 4
 - SE4
 - ST4
 - mixte majeures
 - liste mixte maj 1
 - liste mixte maj 2
 - liste mixte maj 3
 - Majeures période 2
 - A4
 - F4
 - I1

The main calendar grid shows lessons and projects for the week of January 23-27, 2012. The grid is organized by time slots (08h00 to 19h30) and days. Lessons are represented by colored blocks with labels such as 'IF4-ALG2:COURS', 'PROJET:INFO IF4', 'IF4-ALG2:TDR', 'IF4-BDC:COURS', 'IF4-BDC:TDR', 'IF4-DIST:TP', and 'IF4-DIST:TP'. The calendar also shows various lists and resources associated with each lesson.

Nous avons remarqué que dans l’application ‘Ma scolarité, mes notes, mes absences’ il existait identiquement une liste de ressources et que celle-ci était remplie automatiquement pour chaque élève (Ci-dessous les champs de ressources).

Rapport projet I4 : MapMyCalendar – ESIEE Paris

planif.esiee.fr:8080/ade/custom/modules/plannings/direct_planning.jsp?resources=2636,2637,2638,2757,1288,1346

Code Unité	Titre	Matière	Grp.	Sem.	ECTS	Note	Bonus/Malus
AE3S01	Stage d'extension I3S	STAGE		1			
EL413	Filtres	ELECTRONIQUE	1	1	2.5		
EN401	CV and Cover Letters	ANGLAIS		1	1		
EN411	English Electives Courses Term 1	ANGLAIS		1	1		
EN412	English Electives Courses Term 2	ANGLAIS		2	1		
IF4-ALG1	Algorithmique et applications en imagerie	INFORMATIQUE	2	1	4.5		
IF4-ALG2	Algorithmique et applications	INFORMATIQUE	2	2	4.5		
IF4-ARCH	Architecture	INFORMATIQUE	2	1	4.5		
IF4-BDC	Bases de données et Compilation	INFORMATIQUE	2	2	4.5		
IF4-DIST	Informatique distribuée	INFORMATIQUE	2	2	4.5		
IF4-LLC	Langage, logique et compilation	INFORMATIQUE	2	1	4.5		
IF4-PROJ	Projet interne	STAGE		2	5		
IN411	Réseaux informatiques	INFORMATIQUE	1	1	2.5		
IN412	Systèmes temps réel	INFORMATIQUE	1	1	2.5		
IN413	Design Patterns en Java	INFORMATIQUE	1	1	2.5		
MA412	Traitement statistique de données	MATHEMATIQUES	1	1	2.5		
MGT401	Management et Innovation	SC. HUMAINES	1	1	2.5		
MGT4ELEC	Elective Management d'équipe	SC. HUMAINES		2	1		

8

Il serait peut-être alors possible de lier les deux listes pour obtenir automatiquement les ressources d'un étudiant. Cependant cette idée n'a pas pu être mise en place suite à des problèmes administratifs interne.

IV. Android

1. Base du système

Android est le système mobile commercialisé par l'entreprise Google, basé sur un OS Linux, la programmation d'application est effectuée en Java. Les applications produites sous ce système doivent répondre au pattern [Modèle Vue Contrôleur](#) séparant les fonctionnalités applicatives de l'affichage de l'application.

Pour répondre à ces critères, Google utilise un système d'activités et de XML. Nous allons brièvement passer en revue ces deux types de fichier.

2. Activité

Les activités sont des classes susceptibles de jouer un rôle dans la modification de l'interface de l'application. Elles sont composées de plusieurs méthodes de classe détaillées dans la documentation Android disponible à l'adresse : <http://developer.android.com/reference/android/app/Activity.html>

Une activité implémente nécessairement une méthode onCreate jouant le rôle du constructeur de la classe. Une activité possède plusieurs états : elle peut être mise en pause, détruite ou ses paramètres peuvent être susceptibles d'évoluer. Il est donc nécessaire si besoin est de redéfinir les méthodes de ces classes.

3. XML

Les XML jouent un rôle important au sein d'Android, non seulement de définir l'interface, ils interviennent aussi dans le stockage des textes de l'application, dans le lancement des activités et dans la gestion des permissions de l'application.

V. Intégration d'un agenda dynamique

1. Récupération des données

Notre choix de récupérer l'agenda par flux RSS nous a tout d'abord amené à comprendre le format d'un flux RSS. Une brève recherche sur internet nous permet de ramener les informations suivantes :

« RSS (sigle venant de l'anglais « *Rich Site Summary* ») désigne une famille de formats XML », nous avons donc à traiter un flux de donnée comprenant différentes balises traditionnelles. On a donc pensé à « parser » (récupérer et analyser) les données pour en tirer les informations utiles. En effet, en utilisant une adresse précise et en renseignant les ressources utilisateurs, on récupère un flux sur une durée variable contenant les données de l'utilisateur.

Pour obtenir les données, on utilise les bibliothèques URL et InputSource nous permettant de récupérer les données de la page du flux RSS qu'on vient ensuite « parser » une première fois à l'aide des bibliothèques SAX et XMLReader et de la classe RSSHandler. Une fois ces données obtenues on les classe dans la classe RSSFeed pour pouvoir commencer le traitement.

2. Traitement des données

IN213:PROJ jeudi 3 mai 2012 05:31 23/05/2012 16h00 - 17h00 Ressources 1 GRANDPIERRE T. 5201 AKIL M. BDE:DISCO jeudi 3 mai 2012 05:31 11/05/2012 07h30 - 08h00 Ressources 1 I1 I2 I3 I3FE I3FM I3FR I4 I4FE I4FR I5 I5FE I5FR I5BS PIM (N+I)

Le format obtenu est donc de la forme suivante :

URL, le 03 mai 2012 ;

<http://planif.esiee.fr:8080/ade/rss?projectId=4&resources=2371,79,2514,2513,2386,2061,2062,2376,1962,661,662,1673,2946,1553,1382,1565,2135,2396,2398&nbDays=30&x=26&y=2>

(On ne détaillera pas ici le code entier du flux, si besoin le lecteur peut le consulter par lui-même à l'adresse ci-dessus).

Les ressources telles qu'elles sont affichées ici sont réparties à travers plusieurs balises. Il est ainsi simple d'obtenir et de trier un élément comme le titre d'une unité contenu dans la balise <title>, mais le reste des informations se trouve contenu dans une seule

balise <description>.

Il a donc été nécessaire d'utiliser des outils nous permettant de récupérer des informations précises au sein d'un texte.

On vient donc, au sein de l'activity « ShowDescription », « parser » les différents éléments à l'aide d'expressions régulières. Les librairies utilisées dans ce traitement sont Pattern et Matcher. Ces librairies ont posé certains problèmes lors du développement.

En effet, celles-ci possèdent un bug de traitement introduisant une récursivité dans la mémorisation conduisant à un « buffer overflow ». Il a donc fallu réfléchir à des règles spécifiques ne produisant pas cette situation.

L'affichage de la description correspondant à un item isolé, il nous fallait alors une classe permettant d'afficher la liste des items du flux.

3. Affichage des données

RSSActivity

Notre premier affichage du flux ne devant pas forcément être pratique devait prouver le fonctionnement du système. On a donc d'abord privilégié une intégration simple directement sous forme d'une liste des éléments du flux. On vient donc chercher dans cette activité un objet de type « RSSFeed » pour ensuite afficher ses informations dans une « ListView ». Un « ItemClickListener » est lié à cette « ListView » qui, lors d'un clic de souris sur un des items de la liste, fait basculer l'affichage courant vers la description de cet objet.

Un premier problème a été découvert ici : la date de publication des items correspondant à la date de relevé du flux, les items n'étaient pas triés dans la liste et s'affichaient dans un ordre aléatoire. Il a donc fallu retoucher la classe « RSSFeed » pour ajouter un tri des objets en fonction de la date de l'évènement sélectionné. L'utilisation d'une « TreeMap » au lieu de la « HashMap » précédemment utilisée et la transformation de la date, stockée sous forme de string, vers un objet de type « Date », nous ont permis de trier convenablement la liste des ressources.

Le second problème, lié au précédent, a été la retouche nécessaire de la liste contenant la description des objets puisque le lien correspondant au deux était modifié lors du tri.

Pour obtenir les données, ne pouvant pas récupérer automatiquement les ressources, nous avons implémenté un écran de préférence où l'utilisateur dispose d'un lien vers ADE Campus, d'un champ pour entrer ses ressources et d'autres options.

Une fois le test fonctionnel effectué, nous avons décidé de passer à une interface plus ergonomique et correspondant à l'application : une vue sous forme de calendrier.



Vue calendrier



Détail d'activité



Vue des prochaines activités

CalendarActivity

Notre première intention a été de chercher des implémentations d'agenda préexistant pour pouvoir les adapter à notre besoin. Force est de constater qu'il n'en existait qu'un seul convenant réellement à nos besoins, nous avons donc adapté le travail de Chris Gao pour l'intégrer au sein de notre application. Nous avons donc ajouté en plus de la vue disponible une « ListView » affichant la liste des événements du jour sélectionné et implémenter différentes méthodes permettant de récupérer seulement les ressources d'un jour donné.

Les problèmes rencontrés dans cette partie sont liés à la manipulation dynamique de l'affichage : en cas de problème de connexion, il fallait ajouter un message d'erreur, or la liste masquait ce message. Il a donc fallu retirer la liste et créer le « TextView » affichant l'erreur si besoin et inversement.

L'autre problème fut la lenteur de l'application. Nous n'utilisons pas encore de thread, l'affichage de l'application était donc relativement lent, ce qui, suite à l'implémentation d'Android provoquait l'affichage d'une pop-up indiquant un dysfonctionnement de l'application.

4. Ecran de préférence

Evoqué un peu plus haut, il a été nécessaire d'implémenter un écran de préférence, nous avons utilisé un « SharedPreferences » pour jouer ce rôle couplé avec une activité dédiée nous permettant de personnaliser la vue des préférences. Ceci nous a permis l'ajout d'un bouton et d'une « pop-up » d'information. D'autre part, c'est au sein de celui-ci que nous avons ajouté les réglages de plusieurs fonctionnalités, décrites ci-dessous.

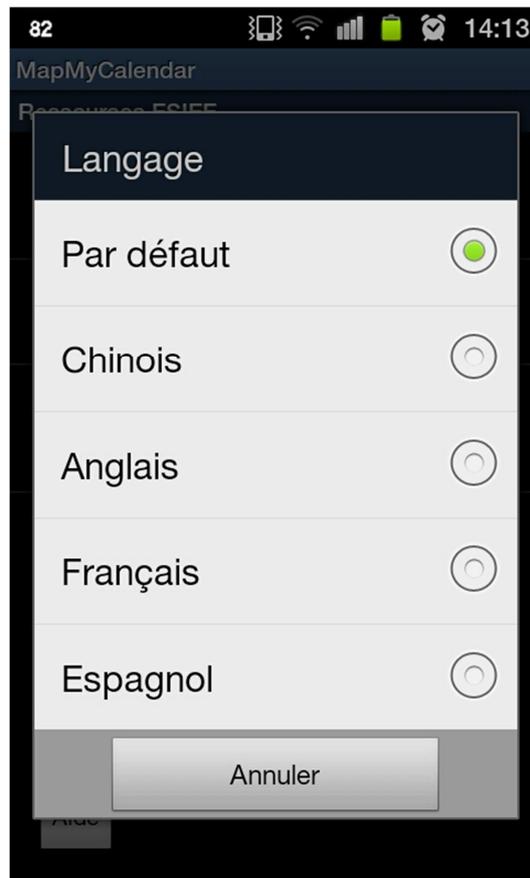
La traduction d'une application est prise en charge automatiquement en fonction des « locales » du téléphone et des traductions proposées par le développeur de l'application. Cependant, nous avons aussi implémenté un changement manuel de la langue de l'application en changeant la localité de l'application. Une des difficultés de cette fonctionnalité fut l'utilisation des « intent » pour permettre un rafraîchissement immédiat de l'interface graphique utilisateur.

Nous avons pensé que l'intégration d'une alarme se déclenchant avant le premier élément de la journée pouvait aussi être une idée très intéressante. En effet, l'horaire des cours changeant chaque matin, il n'est pas rare que l'étudiant ait à reprogrammer son réveil chaque matin. Il était donc judicieux d'implémenter un réveil automatique en fonction du premier événement de la journée.

La dernière fonctionnalité rajoutée fut la mise en place d'une sérialisation des données, autorisant une mise en cache rafraichie selon la volonté de l'utilisateur et améliorant les délais de chargement. Le problème rencontré avec cette fonctionnalité concerne les droits en écriture. En effet, étant sous un système linux, l'application n'a pas accès en écriture à la partition système. Il a donc fallu se placer dans la mémoire externe du téléphone pour pouvoir écrire ces données.



Écran des options



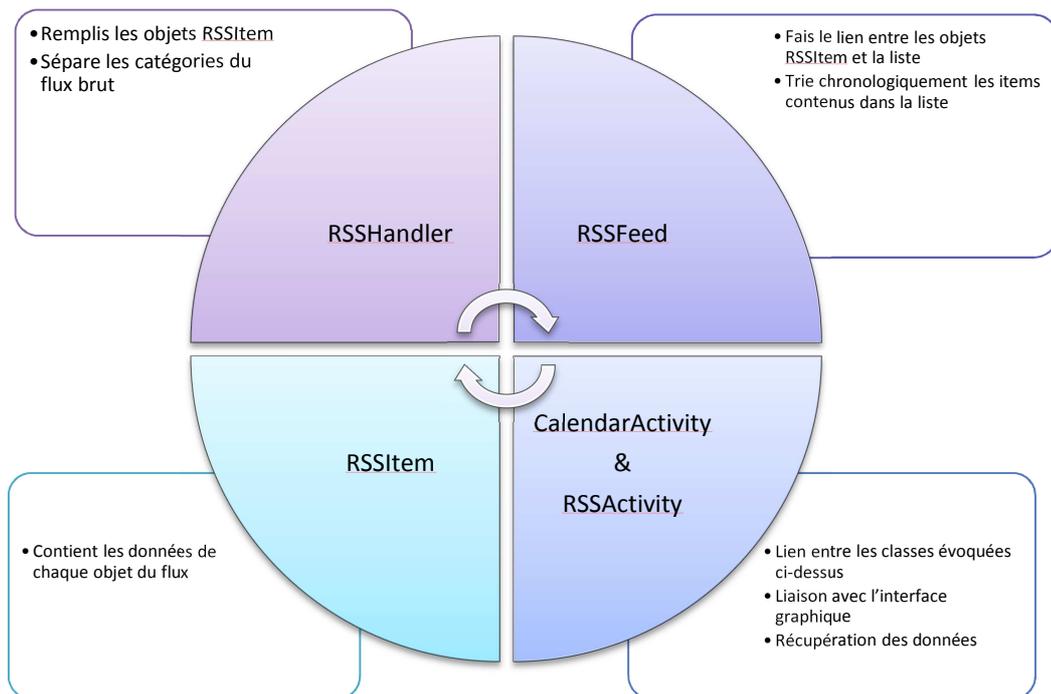
Choix de la langue

5. Menu

Pour permettre à l'utilisateur de naviguer entre les différents écrans de l'application nous avons rajouté un menu s'affichant sur sollicitation de l'utilisateur. Celui-ci est disponible dans toute la partie agenda ainsi que dans les écrans de préférences. L'ajout de ce menu nous a permis de remarquer que toutes les ressources d'icônes fournies au sein d'Android ne sont pas disponibles à l'utilisation : nous avons voulu utiliser des ressources qui, lors de l'ajout dans le code, se sont révélées être privées.

6. Diagramme d'interaction des classes

Le lien entre les différentes classes de la partie agenda peut être représenté tel quel que présenté ci-dessous.



Nous arrivons ainsi à la fin de la partie agenda, nous allons maintenant nous intéresser à la partie horaire de l'application.

VI. Intégration d’horaires de transport

1. Récupération des données

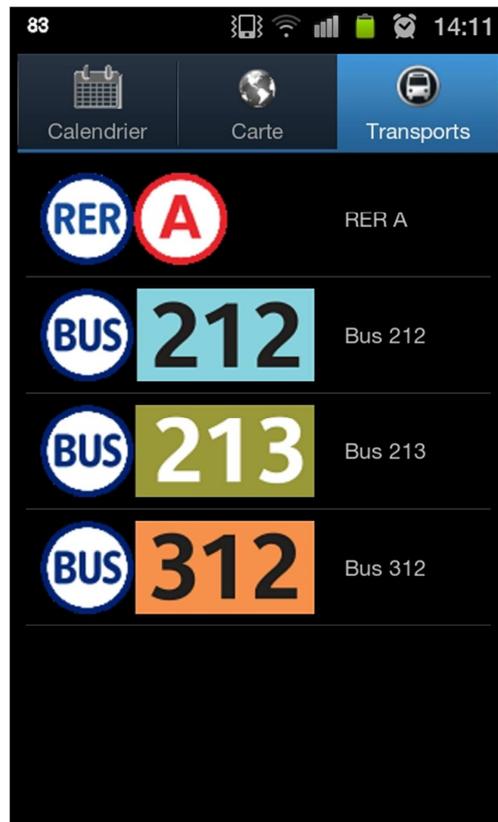
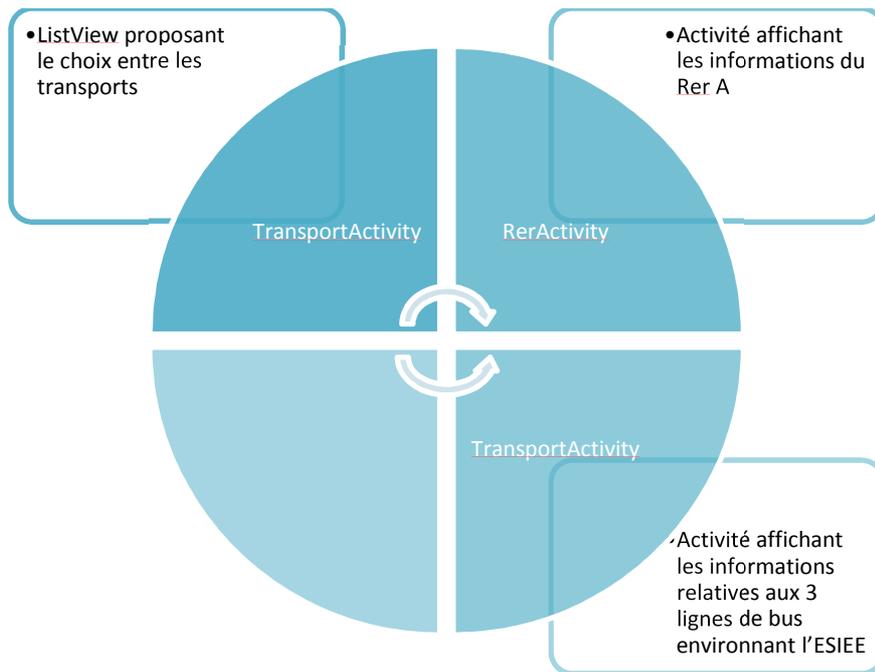
Pour obtenir les données des horaires de transports, n’ayant pas accès à une API, il nous a fallu récupérer les horaires dans le code source des pages même. Pour ce faire, nous avons tout d’abord étudié le code de ces pages pour rechercher des délimiteurs nous permettant d’obtenir seulement les données désirées. Nous avons ensuite découpé chaque page pour obtenir les données nous intéressant.

2. Affichage des données

Pour permettre à l’utilisateur de sélectionner sa ligne de transport, nous avons utilisé une « ListView » contenant le titre du transport ainsi que son icône, permettant une identification plus rapide. Lors d’un clic sur un transport, l’affichage bascule vers une « WebView » qui s’occupe de générer la page correspondant au code tronqué durant l’étape de récupération des données. Une liste déroulante (« Spinner ») a aussi été ajoutée en tête de page pour sélectionner le sens de parcours de la ligne.

La difficulté rencontrée au sein de cette partie est l’affichage des données de la page, car obtenant les données voulues sous forme de page web, deux choix se proposait à nous, éliminer toutes les balises HTML des données et afficher celles-ci comme de simple données où réutiliser le code de la page pour permettre une présentation des données correspondant à celle choisie par la RATP. Nous avons donc opté pour le deuxième choix, affichant les données sous forme de tableau et modifiant les balises pour intégrer le thème proposé par la RATP.

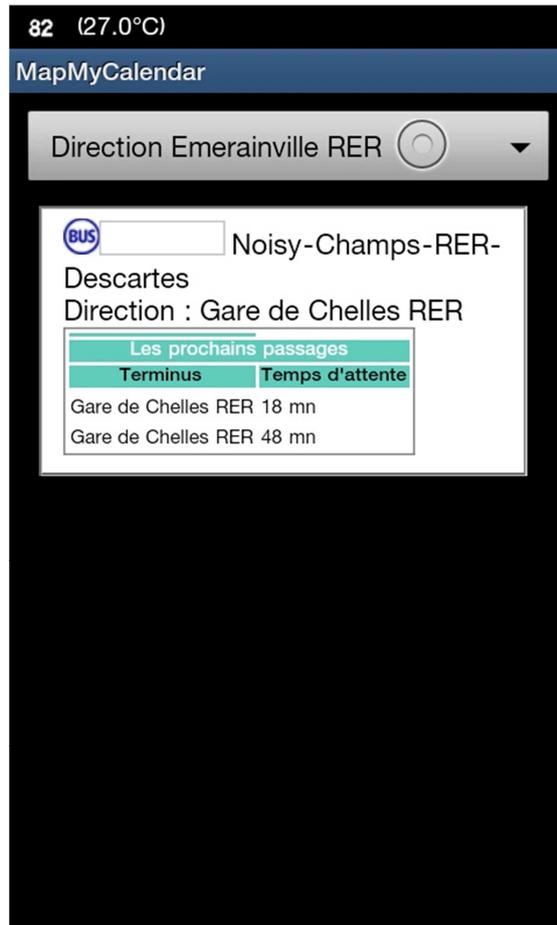
3. Diagramme d'interaction des classes



Choix du moyen de Transport



Vue RER



Vue Bus

VII. Intégration d'une partie géolocalisation

A la base du projet, l'application devait contenir un agenda géo-localisé, permettant une gestion d'emploi interactive en fonction des déplacements de la personne. Par manque de temps et difficulté, cette partie n'a pas pu être intégrée au sein de l'application. Cependant nous avons décidé d'inclure une carte affichant la localisation de l'ESIEE pour pouvoir découvrir l'utilisation de cette fonctionnalité. Il fut difficile de comprendre l'utilisation des différents certificats et la génération de la clé nécessaire au fonctionnement de cette carte : pour chaque développeur, il est nécessaire de disposer de sa propre clé de développement.

Conclusion

Ce projet était relativement ambitieux. Il nous a demandé beaucoup de recherche pour acquérir les diverses connaissances nécessaires à sa réalisation. Ce défi a été difficile à relever mais grâce à un travail d'équipe acharné et à une organisation minutieuse ainsi qu'à une bonne gestion d'équipe, nous avons pu atteindre notre but en y apportant des améliorations.

Ce projet nous a beaucoup appris, que ce soit au niveau technique ou humain. Nous avons réalisé qu'une planification stricte associée à une motivation constante nous permet de réagir plus rapidement face à des contraintes imprévues et de donner le meilleur de nous-mêmes.

MapMyCalendar est terminé, mais il reste un projet très ouvert. Beaucoup d'adaptations et d'améliorations peuvent encore lui être apportées. Sa finalité est de faciliter l'organisation de la vie quotidienne des étudiants autant que des professeurs et des intervenants.



Annexes

Cahier des charges

Afin d'apporter un peu plus d'explications théoriques à ce rapport, le cahier des charges est joint.

Généralités

1. 1. Objectifs du document

Ce document est le cahier des charges de l'application MapMyCalendar. Il vise à fournir une première base de développement pour cerner les différents besoins liés au développement de l'application Android. A travers ce document nous entendons détailler les différentes fonctionnalités de bases proposées par notre application telle qu'elles le seront à la première implémentation.

1. 2. Présentation du projet

Développée dans le cadre, tout d'abord de l'Atos IT challenge, ainsi que dans le cadre du projet de fin d'année de 4^{ème} année d'ingénieur à ESIEE Paris par la suite, l'application MapMyCalendar permet de synchroniser ses rendez-vous (horaires et lieux) sur un calendrier. Mais, à la différence du calendrier d'origine classique proposé sur Smartphone, notre application géo localise en temps réel la position de l'utilisateur sur la carte et calcule le temps de trajet minimum avec pour point de départ la position actuelle de l'utilisateur.

Ainsi l'utilisateur sera averti dès lors que le temps lui restant avant le prochain rendez-vous est approximativement égal au temps de trajet minimum.

Nous souhaitons implémenter un calendrier dynamique donnant la possibilité de s'adapter à l'imprévu. L'utilisateur saura quand passer d'une activité à une autre et sera prévenu des événements susceptibles de gêner le trajet et modifier la durée de parcours.

MapMyCalendar permettrait aussi la consultation de l'emploi du temps relatif aux heures de cours à ESIEE Paris.

1. 3. Fonctionnalités

Liste des premières fonctionnalités de l'application MapMyCalendar.

- Agenda (Vue jour, mois, aujourd'hui, navigation...)
- Itinéraire jusqu'à la prochaine activité (Carte avec itinéraire en fonction de la position actuelle)
- Horaires transports environnant ESIEE Paris

Contraintes : Besoins d'accès aux données nécessaires à l'application (horaire de transport, horaire des cours, connexion au LDAP d'ESIEE Paris...).

1. 4. Parties prenantes

Parties prenantes	Fonctions
ESIEE Paris	Fournis la connexion aux bases de donnée pour utilisateur.
RATP	Fournis les horaires de transport.
Google	Fournis le support logiciel permettant l'exécution de l'application.
Utilisateurs ESIEE	Renseigner identifiants permettant une personnalisation automatique en fonction de l'utilisateur ESIEE.
Utilisateur anonyme	Fonctionnalité de base d'un agenda ajoutant une fonctionnalité de géolocalisation.

1. 5. Utilisateur anonyme

L'utilisateur peut utiliser les fonctionnalités de base de l'application avec l'agenda comme n'importe quelle autre application, il peut ajouter ces propres agenda, en créer au sein de l'application. Il n'a cependant pour le moment pas accès aux différents horaires de transports. Il peut régler dans ses options un délai de préparation pour la première tâche de la journée.

1. 5. Utilisateur ESIEE

L'utilisateur se voit demander au premier lancement de l'application un login et un mot de passe qui seront stockés pour pouvoir accéder à son emploi du temps personnalisé. Il possède une page de configuration pouvant rentrer ses informations personnelles.

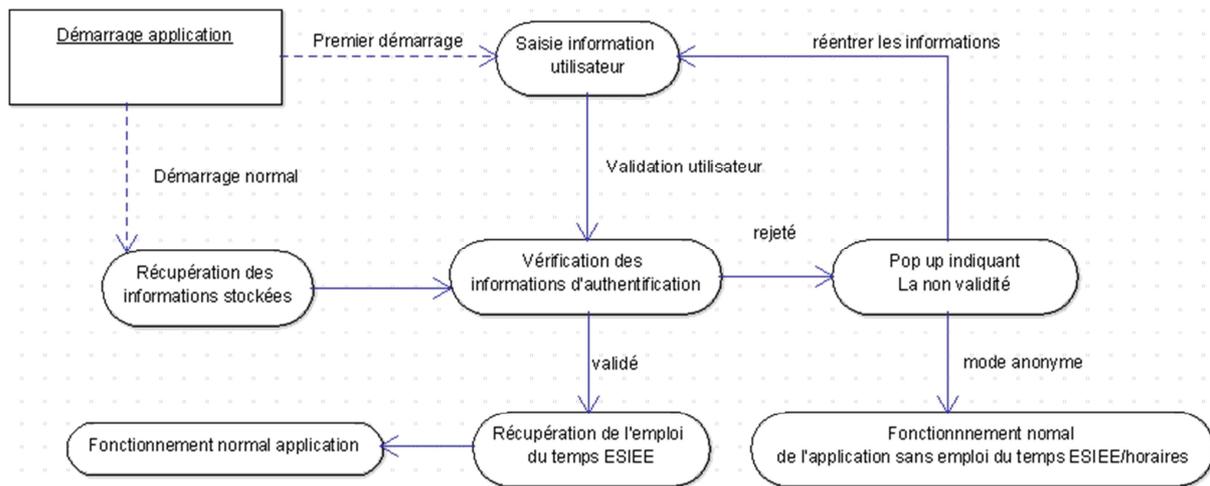
II. Organisation

2. 1. Cas d'utilisation

A. Gestion des utilisateurs

Nom	Connexion
Résumé	Authentification d'un utilisateur
Acteur	Utilisateur authentifié
Pré-condition	Membre ESIEE Paris
Description	Au premier démarrage de l'application, l'utilisateur est invité à remplir une page de configuration comprenant ses informations personnelles et son couple login/mdp relatif à ESIEE Paris.
Post-condition	Le membre accède à son emploi du temps personnalisé en plus de son/ses agenda(s) de bases ainsi qu'aux horaires des transports environnant ESIEE Paris.
Remarques	L'authentification s'effectue via le LDAP de l'école, les données sont stockées sur le téléphone de l'utilisateur, les activités sont récupérées directement via la liste des ressources disponibles dans Ma scolarité/Mes notes.

B. Diagramme d'activité



Ecran prévu :

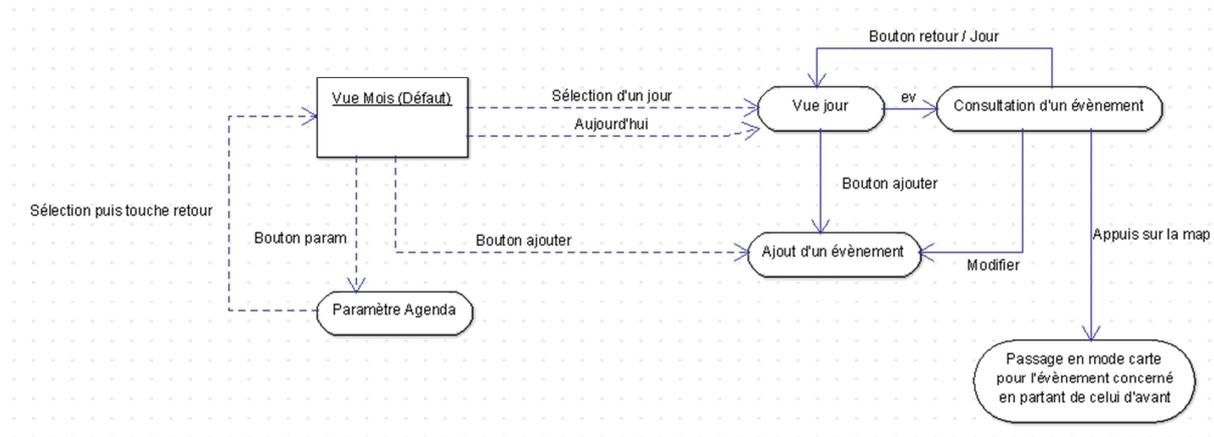
- Démarrage normal : Vue calendrier avec onglet itinéraire et transport
- Premier démarrage : Configuration de l'application

C. Configuration

- Saisie mot de passe et login ESIEE.
- Validation calendrier Gmail
- Calendrier personnel
- Rappel (type de rappel, agenda, délais de rappel, vibreur)
- Vue (premier jours semaine, vue par défaut, météo ? Ville par défaut)
- Temps de préparation

2. 2. Agenda

A. Diagramme d'activité



B. Menu

Nom	Menu
Résumé	Menu de l'agenda
Acteur	Utilisateur
Pré-condition	Etre dans la partie agenda
Description	Accès aux différentes vues de l'agenda, à l'ajout d'activités dans les agendas personnels ainsi qu'au menu des paramètres.
Post-condition	Dans la partie paramètre, dans la partie agenda mois, dans la partie agenda jour.
Remarques	Sélection des différents agenda à afficher.

- Différentes vues possible : Vue jour, vue mois, vue aujourd'hui.
- Accès aux paramètres.
- Ajouter un évènement
- Choix des agendas à afficher.
- Sélection d'un évènement en particulier : affichage des détails évènement.

C. Evènement

Nom	Evènement
Résumé	Création d'un évènement
Acteur	Utilisateur
Pré-condition	Etre dans la partie agenda
Description	L'utilisateur en cliquant sur le bouton + peut ajouter un évènement.
Post-condition	Dans la partie ajout d'un évènement
Remarques	Champs nécessaire avant validation de l'évènement.

- Détail évènement : libellé, horaires et date, lieu, remarques, agenda.
- Ajout d'un évènement : agenda, objet, date, journée ?, lieu (ville, rue), description, rappel, répétition.

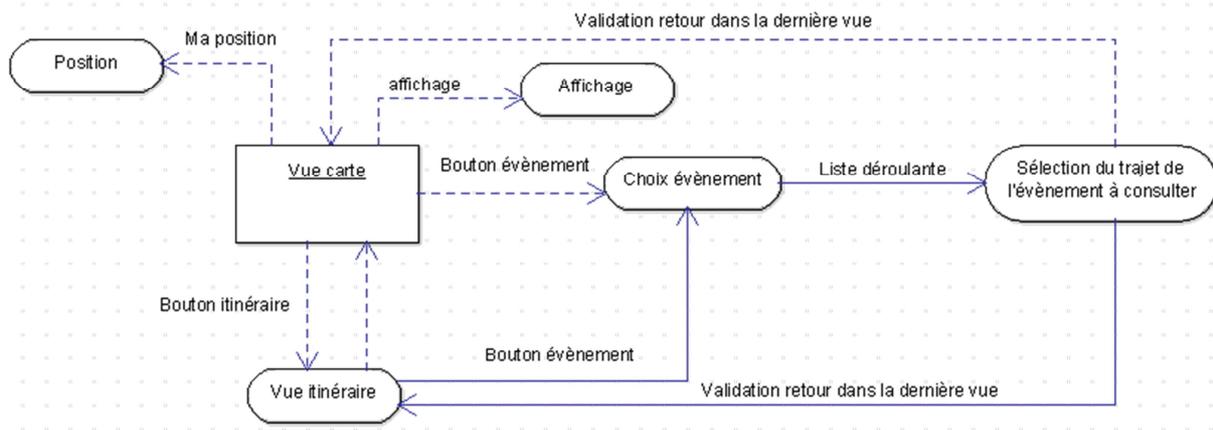
D. Paramètres

Nom	Paramètres
Résumé	Paramètres de l'agenda
Acteur	Utilisateur
Pré-condition	Cliquer sur paramètre à partir de l'agenda.
Description	Accès aux différents paramètres de l'agenda.
Post-condition	Dans la partie paramètre, dans la partie agenda mois, dans la partie agenda jour.
Remarques	Certains champs sont pré-remplis à défaut.

- Login, mot de passe ESIEE
- Oublie du compte ESIEE.
- Démarrage application au boot
- Durée de préparation
- Réveil
- Planification : Agenda à afficher, Type de rappel, ...
- Vues : Vue par défaut, jour de la semaine, météo ? Ville par défaut.

2. 3. Partie Map

A. Diagramme activité



Possibilité de se déplacer sur la map et autres fonctionnalités Google map.

B. Menu

Nom	Menu
Résumé	Menu de la partie carte
Acteur	Utilisateur
Pré-condition	Être dans la partie Map
Description	Accès aux différentes options de la vue carte.
Post-condition	Dans la partie paramètre d’affichage, dans la partie itinéraire
Remarques	La vue carte est celle d’arrivée par défaut.

C. Vue itinéraire

Nom	Vue itinéraire
Résumé	Affichage de l’itinéraire en cours
Acteur	Utilisateur
Pré-condition	Cliquer sur itinéraire à partir du menu carte.
Description	Accès aux détails de l’itinéraire en cours
Post-condition	Dans la partie paramètre, dans la partie vue carte, dans la partie choix de l’itinéraire.

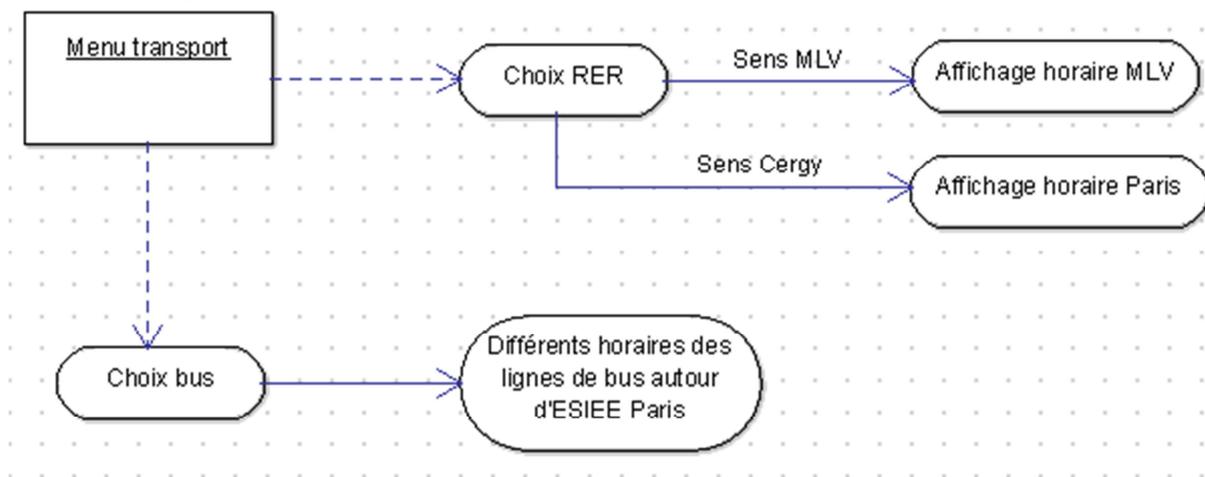
- Affichage avec les transports

D. Choix évènement

Nom	Evènement
Résumé	Choix de l'itinéraire à consulter
Acteur	Utilisateur
Pré-condition	Cliquer sur évènement à partir d'une des vue de la partie Map
Description	Accès aux différents itinéraires.
Post-condition	Dans une des vues de la partie Map, dans les paramètres
Remarques	A voir si cette page ne peux pas être mise dans la partie itinéraire, avec silde gauche et droite.

2. 4. Partie transport

A. Diagramme d'activité



- Choix du mode de transport pour partir de l'ESIEE

B. Choix RER

Nom	Choix RER
Résumé	Choix du RER
Acteur	Utilisateur
Pré-condition	Cliquer sur RER à partir du menu transport
Description	Accès aux prochains horaires dans les deux sens du RER
Post-condition	Dans une des vues horaires du RER
Remarques	A voir si les choix ne peuvent pas tous se faire dans le menu transport.

C. Choix bus

Nom	Choix bus
Résumé	Choix de la ligne de bus
Acteur	Utilisateur
Pré-condition	Cliquer sur bus à partir du menu transport
Description	Accès aux prochains horaires de la ligne de bus choisie dans le sens choisi
Post-condition	Dans une des vues horaires de bus
Remarques	A voir si les choix ne peuvent pas tous se faire dans le menu transport.