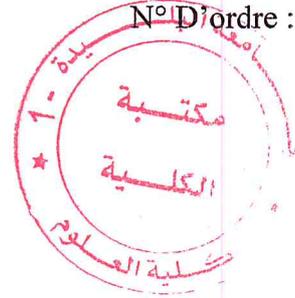


République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab Blida

N° D'ordre :



Faculté des sciences

Département d'informatique

Mémoire Présenté par :

Sahraoui Loukmane Laidi Nacereddine Oussama

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique
Spécialité : Informatique
Option : Ingénierie de logiciel

Sujet : Réalisation et Conception d'un système de recommandation sensible au contexte des applications mobile

Thème : Application mobile dans le domaine du tourisme

Soutenu le :

Mme.Rezzoug Nachida
M.Samir Benaskeur
M.Zebouchi Mohamed
M.Chikhi Fatah
Mme. Cherfa

Promotrice
Encadreur
Encadreur
Président
Examineur

MA-004-255-1

Promotion
2014 / 2015

Sommaire :

Remerciement..... 2

Résumé..... 3

Liste des Figures..... 5

Liste des Tableaux..... 7

Introduction Générale..... 8

1- Introduction :.....

2- Problématique :

3- Objectifs :

I. Chapitre 1 : Etat de l'art : Définitions :

1- Introduction :..... 10

2- Système sensible au contexte:..... 10

2.1- Définition :..... 10

2.2- Caractéristique d'un système sensible au contexte :..... 11

2.3- Contexte :..... 11

2.4- Acquisition du contexte :..... 12

2.5- Utilisation du contexte :..... 12

2.6- Exemple d'utilisation du contexte :..... 12

3- Les types de systèmes sensibles au contexte :..... 13

3.1- Définition d'un Système informatique sensible au contexte:... 13

3.2- Les Systèmes Ubiquitaires :..... 13

3.3- Définition d'un système ubiquitaire :..... 14

4- Architecture proposée dédiée aux systèmes ubiquitaires :..... 14

4.1- Capture de contexte :

4.2- Interprétation de contexte :..... 15

4.3- Gestion de contexte :..... 16



4.4-	Adaptation au contexte :.....	17
5-	Services Peer to Peer & SMA:.....	17
5.1-	Systèmes Pair à Pair (P2P) :.....	18
5.2-	Exemple des systèmes P2P :.....	18
5.3-	Systèmes Multi-Agents SMA :.....	18
5.4-	Exemple des systèmes Multi-Agent :.....	20
6-	Conclusion :.....	20

II. Chapitre 2 : Etat de l'art :

1-	Introduction :.....	21
2-	Les Systèmes de recommandations :	21
2.1-	Définitions :.....	21
2.2-	Les Etapes de recommandation :.....	22
2.3-	Type des systèmes de recommandations :.....	22
3-	Systèmes de recommandation Sensible au contexte :.....	26
3.1-	La naissance des SRSC :.....	26
3.2-	Définition:.....	27
3.3-	Les caractéristiques d'un SRSC :.....	27
3.4-	L'utilité d'un SRSC :.....	28
4-	L'intégration de contexte dans les systèmes de recommandation :.....	28
4.1-	Types d'intégration de contexte :.....	28
4.1.1-	Pré-filtrage contextuelle :	29
4.1.2-	Post-filtrage contextuelle :.....	29
4.1.3-	Modélisation contextuelle :	30
5-	Exemples des SRSC:.....	30
5.1-	Pandora Radio :.....	30
5.2-	NetFlix :.....	31
5.3-	Sailendra :.....	31
6-	Conclusion :.....	32

III. Chapitre 3 : Travaux Existants :

1- Introduction :	33
2- Domaine d'utilisation et utilités des SRSC :	33
3- Travaux existants :	33
3.1- Les Travaux basés sur les contextes pertinents :	34
3.2- Travaux basé sur la corrélation entre le contexte et les préférences :	36
3.3- Travaux basé sur la corrélation du contexte / préférences et le choix du type de la recommandation:	41
3.3.1- SRSC utilisant le filtrage collaboratif :	41
3.3.2- SRSC utilisant des recommandations basées sur le contenu :	43
3.4- Nos Exemples :	44
3.5- Autre Travaux du même type [5] :	44
4- Discussion générale:	45

IV. Chapitre 4 : Conception et Réalisation :

1. Introduction :	47
2. Domaine d'application :	47
3. Architecture Dédiée à Notre Système :	48
3.1- L'architecture proposée :	48
3.2- Description des termes :	49
3.3- Identification des agents :	49
3.4- Identification des interactions :	49
4. L'ontologie du contexte :	50
4.1- Composant de l'architecture :	51
4.1-1. Acquisition du contexte [36] :	51
4.1-2. Le Matching :	52
4.1-3. Approche de Classification Bayésienne:	53
4.1-4. SRSC et Processus de corrélation :	53

4.1-5. La Base de données:.....	55
4.1-6. Diagramme de classe :.....	55
4.2- Input / Output des composants de l'architecture :.....	56
5. Conclusion :.....	59

V. Chapitre 5 : Implémentation :

1. Introduction :.....	60
2. Environnement de travail :.....	60
3. Environnement Logiciel :.....	60
3.1. Environnement de développement (Plateforme et Logiciels):.....	60
3.1.1. Pourquoi nous avons choisie Android ?.....	60
3.1.2. Les Différentes versions d'Android et la version choisie :.....	61
3.1.3. Les outils et les données auxiliaires :.....	62
3.2. Technologies utilisées (Base de données) :.....	64
3.2.1. Les Bases de données pour Mobile :.....	64
3.2.2. Définition des BaaS (Backend as a Service) [40] :.....	65
3.2.3. Base de données choisie : (Parse.com) :.....	65
3.2.4. Mais c'est quoi Parse.com ?.....	65
3.2.5. La notion de classe :.....	65
4. Test d'intégration :.....	66
4.1.1. Présentation de l'application (Bino) :.....	66
4.1.2. Représentation des interfaces de l'application :.....	67
5. Conclusion :.....	75

Conclusion Générale.....	76
--------------------------	----

Bibliographie :.....	77
----------------------	----

Remerciements :

Avant tous je remercie le bon dieu qui veille sur nous et qui nous protège avec sa grâce et générosité accordant le souffle de la vie et la bonne santé à notre être pour accomplir notre destin dans la vie, j'adresse aussi mes remerciements à ma mère et à mes petits frères qui ont fait en sorte que je garde le bon rythme, et à mes valeureux proches, et mon binôme Laidi Oussama qui m'a traité comme un frère durant nos cinq années d'études.

Les travaux présentés dans ce mémoire ont été effectués à Condor ICT, et tutorés par Mr. Samir Benaskeur et Mr. Anouar Mohamed que je tiens à remercier du fond du cœur pour leur indulgence et leur bienveillance tout au long notre stage chez eux, où ils ont toujours répondu présents à nos demande d'aide en nous dirigeant avec gratitude et assurance rendant notre travail un travail digne et professionnel.

Je remercie surtout ma promotrice Mme. Rezzoug qui nous a dirigés avec dignité et à la perfection, avec ses bonnes informations et son unique soutien avec grand cœur.

Sahraoui Loukmane.

Je tiens à remercier avant tout le bon Dieu puissant de nous avoir donné la volonté, le courage et la santé de réaliser ce travail. J'adresse mes plus sincères remerciements à mes chers mes parents, ma sœur et mes deux petits frères, qui m'ont accompagné, aidé, soutenu et encouragé tout au long de la réalisation de ce mémoire.

Je voudrais exprimer ma reconnaissance envers le corps enseignant du département d'informatique de l'université SAAD DAHLEB -- BLIDA, particulièrement notre promotrice Mme. Rezzoug pour ses conseils, son assistance, ses orientations et son suivi parfait.

Un grand remerciement à mon frère, mon binôme et mon meilleur ami pour ces efforts et son esprit de travail.

Je tiens à remercier Mr. Samir Benaskeur qui nous a fourni les outils nécessaires pour réussir ce projet, Mr. Anouar Mohamed notre encadreur qui nous a soutenus au cours de ce stage. Je veux aussi adresser tous mes remerciements aux personnes avec lesquelles j'ai pu échanger et qui m'ont aidé pour la rédaction de ce mémoire.

Laidi Nacereddine Oussama.

Résumé :

De nos jours, l'interaction avec les applications informatiques profitent de plus en plus de technologies mobiles. L'adoption de ces technologies assure plus de flexibilité et permet de créer de nouvelles formes d'utilisation. Ces technologies mobiles attribuent à un environnement de travail des caractéristiques physiques et des propriétés sociales. Les propriétés sociales définissent les habitudes, les règles et les normes sociales d'un environnement mobile. Les paramètres et les propriétés de cet environnement définissent les informations de contexte. Une application sensible au contexte est une application qui répond aux exigences imposées par ces informations de contexte. Une telle application doit être capable de capturer et de gérer les informations de contexte en vue d'apporter des services adaptés, On abordera aussi les différents types de recommandation, et certains travaux existants pour appliquer des critiques et les éviter dans notre développement du système de recommandation.

Cette thèse sera l'implémentation de cet environnement en une application mobile dans le domaine touristique qui fera interaction avec l'utilisateur, en capturant ses point d'intérêts (POI), son activité et ses contextes avec une architecture basée sur un système multi-agents (SMA), qui calculera le modèle du contexte d'après un matching pour arriver à une recommandation bien précise et un résultat fiable à afficher.

Mots Clés :

Sensibilité au contexte, Système de recommandation, Environnement mobile, Points d'intérêts, Architecture, Système multi-agents, Matching.

Abstract :

Nowadays, the interaction with software applications is mainly oriented towards mobile technologies. The adoption of these technologies allows providing more flexibility and creating new forms of interactions. These mobile technologies attribute to a working environment a set of physical parameters and social properties. The social properties define habits, rules and social norms of an environment. These parameters and properties constitute the context information. A context-aware application is an application that meets the requirements of the context information. Such an application must be capable of capturing and managing context information in order to provide adapted services. We will also discuss the different types of recommendations, and some existing work projects to apply some critics and avoid them in our development of the recommendation system .

This thesis is the implementation of the environment within a mobile application in the tourism domain; it will make interaction with the user , by capturing his points of interest (POI), his activity and his contexts with an architecture based on a multi-agent system (SMA) ,that calculates the model of context after the matching to get to a specific recommendation and reliable results to be well displayed.

Keywords :

Context Awareness, System of Recommendation, Mobile Environment, Points of Interest, Architecture, Multi-agent System, Matching.

Liste des Figures :

Figure 1- Interface Macintosh.....	
Figure 2- Architecture Proposee Dediee Aux Systemes Ubiquitaires.....	
Figure 3- Recommandation Personnalisee.....	
Figure4- Filtrage Collaboratif.....	
Figure 5- Recommandation Basee Sur Le Contenu.....	
Figure 6- Recommandation Hybride.....	
Figure 7- Application Tenant Compte Du Contexte.....	
Figure 8- L'interface Deapz.....	
Figure 9- Interface Utilisateur Pour La Specification Du Contexte.....	
Figure 10- La Liste Recommandee.....	
Figure 11- Detail D'un Choix De La Recommandation.....	
Figure 12- Interface De Validation Des Preferences.....	
Figure 13- Interface De La Recommandation Finale.....	
Figure 14- Interface Flipboard.....	
Figure 15- Recommandation Personnalise Du Play Store En Utilisant Le Contexte De Lieu.....	
Figure 16- Recommandation En Utilisant Le Calcul De La Similarite Et Le Contexte De Lieu.....	
Figure 17- Architecture Generale Du Systeme.....	
Figure 18- Ontologie De Contexte.....	
Figure 19- Arbre de décision.....	
Figure 20-Diagramme de classe.....	
Figure 21- Différente Version D'android.....	
Figure 22- Environnement Android Studio.....	
Figure 23- Version Java JDK utilisé :.....	

- Figure 24- Version de l'API Android (SDK).
- Figure 25- Interface GenyMotion (Machine Virtuel Lancée).
- Figure 26- Interface Parse
- Figure 27- Aperçue de notre application sous notre Téléphone.
- Figure 28- Interface Splash Screen.
- Figure 29- Interface Login (Connexion aux Comptes).
- Figure 30- Interface Sign-Up 1^{ère} étape.
- Figure 31- Interface Sign-Up 2^{ème} étape.
- Figure 32- Menu Déroulant de l'application (DrawerMenu).
- Figure 33- Interface « What's New ?! » coté articles.
- Figure 34- Interface « What's New ?! » coté Vidéos.
- Figure 35- Interface service « Near By », sous service « Events » :
- Figure 36- Interface service « Near By », sous service « Map » :
- Figure 37- Interface Service « Let's Go ! », soumission de données.
- Figure 38- Interface Service « Let's Go ! », résultats affichés.
- Figure 39- Interface Settings (Paramètre des POI du profil).
- Figure 40- Interface Settings (Paramètre du compte).
- Figure 41- Interface Settings (Paramètre de langage).
- Figure 42- Interface About Us.

Liste des Tableaux :

Tableau 1- Comparaison Entre Les Types De Recommandation.....	
Tableau 2- Avantages Et Inconvénients De Pandora Radio.....	
Tableau 3- Avantages Et Inconvénients De Netflix.....	
Tableau 4- Avantages Et Inconvénients De Sailendra.....	
Tableau 5- Analyse Et Critique D'ubiquito.....	
Tableau 6- Analyse Et Critique De Moma.....	
Tableau 7- Analyse Et Critique De Deapz.....	
Tableau 8- Analyse Et Critique De Rerex.....	
Tableau 9- Analyse Et Critique De Blinkfeed.....	
Tableau 10- Analyse Et Critique De Flipboard.....	
Tableau 11- Comparaison Entre Les Types Et Les Travaux Présentés....	
Tableau 12 – Analyse de l'architecture.....	
Tableau 13– POI de l'utilisateur.....	
Tableau 14 – Déroulement de l'exemple précédent.....	

Introduction générale :

1- Introduction :

De nos jours le business a changé est devenu beaucoup plus développé en technologie que ce qu'il était auparavant, et d'une certaine manière il a surpassé la capacité de l'utilisateur à tout se procurer mais ce dernier n'a jamais fait pas en arrière en achetant toute les nouveautés du marché, des plus fines et intelligente télévision au plus puissant des mobiles.

Puis vient le rôle des media à englober les informations que veut l'utilisateur, en lui offrant plusieurs données éparpillées parfois sur le net, parfois sur des chaînes Tv, donc il s'enlace et il passe à chercher certaines news susceptible à ses intérêts et besoins culturelle en perdant beaucoup de temps car chaque intérêt diffère de l'autre (Sport, Politique, Religion, Musique...), et se plonge dans un flux constant de mises à jour, et d'informations, aussi l'être humain cherche aussi son comble d'informations dans ses sorties qui deviennent de plus en plus flou et se faite seulement par les agences de tourisme qui coûtent chères et parfois pour des sorties simple il trouve pas exactement l'endroit d'après ses points d'intérêts .

Bien que les Smartphones soient l'un des principaux moyens de rester en contact avec les personnes et les informations qui nous intéressent, les conceptions classiques n'ont pas réussi à suivre le rythme sur lequel les gens les utilisent réellement (Site web d'actualités et de tourisme).

2- Problématique :

La surcharge importante en termes d'information et de contenus disponible et accessible des systèmes d'information actuels traitent une grande quantité de données et délivrent par conséquent un nombre important de réponses aux requêtes de l'utilisateur, se retrouvent ainsi incapable de distinguer l'information qui lui est pertinente de celle qui ne l'est pas, donc au lieu d'aboutir à une bonne information, il se retrouve avec plein d'actualités ou avec pleines de destinations et pas d'assez d'information surtout sur le coût par rapport à sa durée et à ses préférences.

3- Objectifs :

Notre objectifs et de définir un système de recommandation comme une forme spécifique de filtrage de l'information visant à présenter les éléments d'information (films, musique, livres, news, images, pages Web, etc) et les lieux en lien avec le tourisme et les sorties de notre utilisateur qui sont susceptibles de l'intéresser.

Autrement dit, un système de recommandation cherchant à prédire la valorisation ou préférence qu'un utilisateur attribuerait à un objet (livre, musique, film...) ou à un élément social (personne, groupe, communauté) qu'il n'avait pas encore considéré et aussi un guide touristique bien définie qui l'aidera à bien connaître l'endroit à destination et les lieux fameux sur une Map bien clair.

Et Pour être pertinent, un système de recommandation doit pouvoir faire des prédictions sur les intérêts des utilisateurs. Il faut donc pouvoir collecter un certain nombre de données sur ceux-ci afin d'être capable de construire un profil pour chaque utilisateur selon les contextes qu'on a choisies.

Notation : c'est juste l'objectif principal, l'idée va être élargie avec plus de fonctionnalités pour enrichir le projet en fonction des demandes des encadreurs.

I. Chapitre 1 : Etat de l'art : Définitions :

1. Introduction :

De nos jours, l'informatique n'est plus liée à un poste de calcul fixe : nos téléphones sont devenus de petits ordinateurs, et même d'autres objets autour de nous sont informatisés. De plus, les systèmes informatiques gagnent en complexité : un très grand nombre d'entités participent à l'interaction, plusieurs utilisateurs se trouvent dans des endroits ouverts aux technologies mobiles, nous avons besoin d'accéder à des services à distance, etc. Nous passons ici en revue quelques-uns des domaines qui sont reliés à nos recherches : l'informatique ubiquitaire, les systèmes sensibles aux contextes et les Services P2P.

Pour éviter tout risque de confusion entre les termes utilisés dans le domaine de l'informatique permansive, nous présentons les définitions suivantes qui le caractérisent [Laforest et Mouël, 2005/2006]; [Saha et Mukherjee, 2003]; [Weiser, 1993] :

- **Ubiquitaire** : Accessible de n'importe où ;
- **Mobile** : Qui intègre les terminaux mobiles ;
- **Sensible au contexte** : Qui prend en compte le contexte d'exécution ;

2. Système sensible au contexte:

L'utilisation du contexte est nouvelle en informatique, il y a ce qu'on appelle des systèmes sensibles au contexte, mais en fait qu'est-ce qu'un système sensible au contexte, quelle est la définition d'un contexte ?

2.1. Définition :

- Les systèmes sensibles au contexte sont des systèmes dont l'interaction dépend de l'environnement physique de l'utilisateur. [2]
- Un système est sensible au contexte s'il utilise du contexte pour fournir de l'information pertinente et/ou des services à l'utilisateur, où la pertinence dépend sur la tâche de l'utilisateur. [3]
- Un Système sensible au contexte est un système dont le comportement ou la structure peut varier en fonction de l'état de l'espace des informations de contexte.
- On dit d'un système qu'il est sensible au contexte s'il peut tirer, interpréter et utiliser des informations issues du contexte et adapter sa réponse en fonction du contexte d'utilisation. [1]

.. Caractéristique d'un système sensible au contexte [4] :

Les informations et les services peuvent être présentés à l'utilisateur dans le contexte courant. Ceci inclut la sélection d'informations de proximités.

- Exécution automatique d'un service dans un certain contexte. Ceci inclut les actions déclenchées par des capteurs de contexte.
- Etiquetages du contexte à l'information pour une restitution ultérieure.

2.3. Contexte :

Il est beaucoup plus dur de donner une définition précise à la notion de contexte, Les premières définitions de la notion de contexte en remontent aux travaux d'Ingenwersen et de Saracevic qui ont placé le contexte en avant de l'interaction utilisateur-SRI.

Gaetan Rey et Joelle Coutaz proposent quatre axes de définitions du contexte sur lesquels s'accordent une majorité des chercheurs [5]:

Définition 1 :

- Il n'y a pas de contexte sans contexte : La notion de contexte doit se définir en fonction d'une finalité. Par exemple, on cherche à adapter dynamiquement les capacités interactives d'un système.
- Le contexte est un espace d'information qui sert l'interprétation : la capture du contexte n'est pas une fin en soi mais les données capturées doivent servir un objectif.
- Le contexte est un espace d'informations partagé par plusieurs acteurs : ici, il s'agit de l'utilisateur et du système.
- Le contexte est un espace d'informations infini et évolutif : le contexte n'est pas figé une fois pour toutes mais se construit au cours du temps.

K. Dey et Gregory D propose une autre définition du contexte qui paraît plus précise et formelle [6].

Définition 2 :

Le contexte est l'ensemble de toutes les informations qui peuvent être utilisées pour caractériser la situation d'une entité. Une entité pouvant être un acteur, un lieu, un objet de l'environnement considéré comme utile à l'interaction entre la personne et le système.

Remarque :

Cette définition permet à un développeur de savoir instantanément ce qui est du contexte et ce qui ne l'est pas. Si une information peut être utilisée pour caractériser ou influencer sur les

interactions entre un utilisateur et le système, alors, cette information fait partie du contexte. Elle est hors contexte dans le cas inverse.

Quelques exemples de sensibilité au contexte :

- Démarrage d'une synchronisation lorsque le réseau est rétabli (réaction)
- Avertissement lorsqu'une personne est à proximité (réaction)
- Affichage une position géographique (connexion contexte – attribut)

Mais comment un tel système acquiert du contexte et l'utilise ?

2.4. Acquisition du contexte [7] :

On peut distinguer principalement deux techniques de collecte des données du contexte, l'acquisition explicite et l'acquisition implicite :

- **L'acquisition explicite :** repose principalement sur les techniques de feedback explicites, Ces techniques permettent une construction contrôlée du profil utilisateur. Cependant elles présentent des limites à cause de l'effort supplémentaire imposé à l'utilisateur à spécifier explicitement ses besoins. Ces limitations ont orienté les travaux vers des techniques d'acquisition implicite des données du contexte utilisateur.
- **L'acquisition implicite :** consiste à collecter à l'aide d'algorithmes d'acquisition implicite les données de l'utilisateur en observant ses interactions avec le système durant les activités de recherche, L'avantage de cette approche est qu'elle ne nécessite aucune implication directe de l'utilisateur. Le répertoire d'informations collectées constitue un riche répertoire de données qui peut éventuellement être exploité selon diverses techniques.

2.5. Utilisation du contexte :

L'utilisation du contexte se fait souvent en mélangeant les données en provenance de plusieurs capteurs. La déduction indirecte d'éléments du contexte est difficile pour un système automatique.

2.6. Exemple d'utilisation du contexte [8] :

C'est un widget particulier d'interface introduit par le Macintosh qui peut être appelé par la souris n'importe où au sein de l'écran ou d'une fenêtre et que son contenu dépend de cet endroit.

Ici le contexte récupéré est celui de l'application, telle qu'il a été prévu par le concepteur de l'application, le contexte utilisateur n'influence pas sur le menu contextuel. L'utilisation de

menus contextuels permet à l'utilisateur de disposer d'un panorama des possibilités offertes par l'application dans un contexte donné.

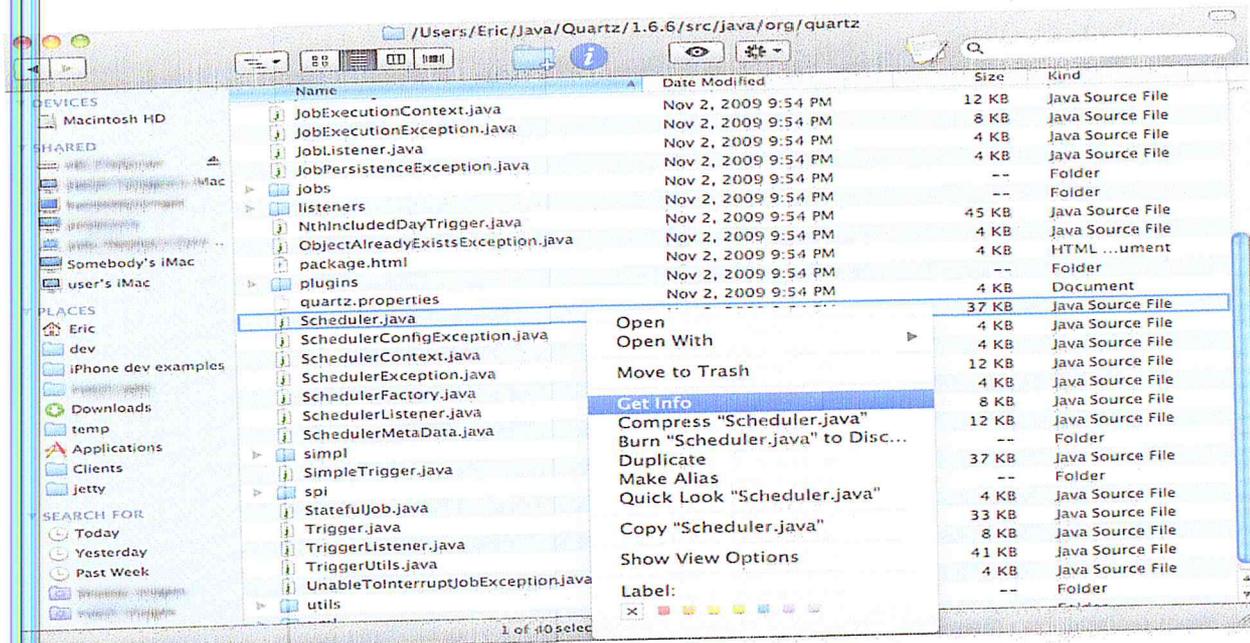


Figure 1 -- Interface Macintosh --

3. Les types de systèmes sensibles au contexte :

Il existe plusieurs types de systèmes sensibles au contexte, parmi ces systèmes nous s'intéressons aux systèmes informatiques sensibles au contexte, plus précisément les systèmes ubiquitaires.

3.1. Définition d'un Système informatique sensible au contexte:

Un système informatique sensible au contexte est un ensemble des moyens informatiques ayant pour finalité délaborer, traiter, stocker, acheminer, présenter ou détruire des données dont son comportement dépend de l'environnement physique de l'utilisateur (le contexte).

3.2. Les Systèmes Ubiquitaires :

Dans l'informatique, les capacités informatiques s'intègrent dans la vie de tous les jours, dans les différents objets qui nous entourent, tels que les imprimantes, nos vêtements, les appareils électroménagers... Les systèmes ubiquitaires suppose donc de déterminer l'état courant du contexte grâce aux caractéristiques permansives du dispositif pour fournir des informations/interactions adaptées à la situation de l'utilisateur.

3.3. Définition d'un système ubiquitaire [10]:

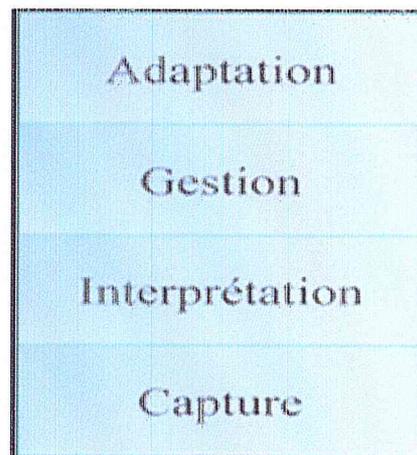
Un système ubiquitaire est un environnement dans lequel les ordinateurs et réseaux sont « enfouis », « intégrés » et « omniprésents » dans le monde réel. L'utilisateur a accès à un ensemble de services à travers des interfaces distribuées se voulant intelligentes, dont il est entouré. Ces interfaces s'appuient sur des technologies intégrées dans les objets familiers.

Le W3C définit l'informatique ubiquitaire, comme un paradigme émergeant de l'informatique personnelle, elle est caractérisée par :

- L'utilisation de dispositifs légers, manipulables et sans fil.
- La nature permansive et sans fil des dispositifs qui demande des architectures réseaux supportant une configuration automatique.
- La haute distribution, hétérogénéité, mobilité et autonomie de l'environnement.

4. Architecture proposée dédiée aux systèmes ubiquitaires :

Plusieurs travaux ont proposé des architectures sensibles au contexte. Un accord général, dans ces travaux, sur la séparation entre la capture des informations de contexte et l'utilisation de ces informations afin d'assurer l'extensibilité et la réutilisation du système. La figure suivante (Figure 2) montre l'architecture générale d'un système sensible au contexte. Elle est composée des couches suivantes : Capture de contexte, Interprétation de contexte, Gestion de contexte et Adaptation au contexte.



-Figure 2- Architecture proposée dédiée aux systèmes ubiquitaires.

4.1. Capture de contexte :

Les systèmes sensibles au contexte sont désignés pour être à l'écoute aux changements de l'environnement de système. La couche de capture est composée d'un ensemble de capteurs. Ces capteurs peuvent être classifiés selon qu'ils soient physiques ou logiques.

- **Capteurs physiques** : ces capteurs capturent des grandeurs physiques tels que la température, la position géographique, le son, la lumière, etc. Parmi ces capteurs on trouve : le GPS pour déterminer les coordonnées d'un utilisateur, des caméras, des microphones, etc.

- **Capteurs logiques** : ces capteurs peuvent rassembler des données à partir des applications et des services logiciels. Par exemple, on consulte un calendrier électronique pour déterminer la position ou l'activité courante d'un utilisateur.

Dans le travail de Indulska, on donne une catégorisation des capteurs selon : physiques, virtuels et logiques. Les capteurs physiques sont pour les grandeurs physiques. Les capteurs virtuels sont les capteurs qui se basent sur les applications et les services et les capteurs logiques sont les combinaisons entre les capteurs physiques et virtuels.

Nous considérons que la transformation est une sorte d'interprétation d'informations de contexte. Afin d'assurer la réutilisation de l'information de contexte nous séparons la couche interprétation de la couche capture de contexte. Dans le paragraphe suivant, nous expliquons la couche destinée pour l'interprétation de contexte.

4.2. Interprétation de contexte :

Cette couche a pour but d'interpréter les données contextuelles fournies par les capteurs. Elle sert à l'analyse et à la transformation des données brutes, fournies par la couche de capture, d'autres formats plus expressifs à l'application. Les transformations effectuées sur les données brutes concernent l'extraction, la quantification et l'agrégation.

Par exemple, les coordonnées GPS d'une personne peuvent être moins significatives qu'une adresse sous forme de numéro de rue et de ville. Cette couche peut aussi avoir le rôle de résolutions de conflits causés par l'utilisation de plusieurs sources de contexte. Ces sources peuvent donner des résultats contradictoires ou peuvent aboutir à des situations imprécises. Cette couche doit donc avoir des moyens pour résoudre ces conflits.

4.3. Gestion de contexte :

A ce niveau, le contexte capturé et interprété doit être bien géré pour faciliter l'utilisation. La gestion de contexte contient le stockage et la représentation formelle des informations de contexte. Le stockage peut être centralisé ou distribué.

- Le stockage centralisé est le plus répandu et le plus utilisé puisqu'il facilite des mises à jour et des modifications des valeurs de contexte. Pourtant, cette solution se limite avec les contraintes de la capacité des équipements mobiles utilisés dans la sensibilité au contexte. Ces équipements sont caractérisés par leurs espaces de stockage restreints.
- Le stockage distribué remédie bien à la contrainte de l'espace de stockage des équipements utilisés. Cependant, il inflige des fonctions additionnelles relatives à la synchronisation et à l'actualisation des valeurs de contexte.

La représentation de contexte est la structuration de ce contexte selon un modèle. Le choix du modèle dépend fortement de mécanismes choisis pour adapter les actions du système au contexte. Plusieurs méthodes de représentation ont été proposées dans la littérature, en invoquant les diverses technologies méthodes sont présentées selon :

- Représentations à base de pair (attribut, valeur) : c'est la structure des données la plus simple pour la modélisation des informations contextuelles. L'attribut représente un élément de contexte tel que la localisation de l'utilisateur. La valeur est la valeur actuelle de cette information.
- Représentations basées sur XML : ces modèles utilisent une structure de données hiérarchique. Plusieurs modèles de description des informations de contexte se dérivent à partir de ce langage.
- Le langage CC/PP est une recommandation de W3C (World Wide Web Consortium) Il est basé sur RDF (Resource Description Framework) qui est utilisé afin de créer des profils décrivant les capacités des terminaux et les préférences d'un agent utilisateur. CC/PP est utilisé pour personnaliser le contenu sur la base des capacités d'un terminal et les préférences de l'utilisateur.

Ces modèles fournissent une description des éléments de contexte (les ressources) en incluant des contraintes élémentaires et des relations entre ces éléments de contexte Pour construire un profil. Ces modèles sont plus expressifs relativement aux modèles basés sur Attributs/valeurs.

Représentations graphiques : Ces modèles consistent à modéliser les informations contextuelles selon un graphe conceptuel. Quan et al. ont utilisé une représentation graphique basée sur UML pour la modélisation des informations contextuelles pour la gestion des services web.

4.4. Adaptation au contexte :

L'adaptation au contexte est l'ensemble des mécanismes de réactions prévues suites aux changements de contexte. L'adaptation se base sur un ensemble de règles d'adaptation. Ces règles sont implémentées selon des langages de programmations traditionnels ou bien en utilisant la logique de prédicats. Pour remédier au caractère incertain de contexte, certains travaux adoptent la logique floue ou la logique probabiliste. Plusieurs formes d'adaptation peuvent être présentées :

- **Adaptation architecturale** : désigne les changements faits, en temps réel, dans la structure des composants du système et/ou dans les interactions entre eux en utilisant un modèle architectural du système.
- **Adaptation compositionnelle** : concerne les modifications de la structure et du comportement d'une entité logiciel, en temps réel, en réponse aux changements arrivés à son environnement d'exécution.
- **Adaptation structurelle** : consiste en la mise à jour de sa structure en préservant son comportement et ses services. Une adaptation structurelle signifie le changement dynamique de type des composants de l'application tel qu'une signature d'une méthode.
- **Adaptation comportementale** : désigne les changements dynamiques dans la phase de l'exécution d'un composant logiciel d'une manière non intrusive (par exemple : en changeant sa configuration ou bien en interceptant ses requêtes et réponses).
- **Adaptation de contenu** : concerne la transformation et la manipulation des contenus en se basant sur les caractéristiques de l'application et des terminaux utilisés.

5. Services Peer to Peer & SMA :

Dans cette partie, nous présentons les concepts liés à l'informatique ubiquitaire, un paradigme récent de l'informatique personnelle dont l'objectif est de permettre aux utilisateurs de consulter des données n'importe quand, n'importe où.

L'informatique ubiquitaire a besoin d'architectures et de technologies dotées de fortes capacités de communication. Shizuoka ne considère que les systèmes Pair à Pair (P2P) soient

une bonne solution, permettant que les pairs soient ubiquitaires malgré les changements de localisation. La section suivante leur est consacrée.

5.1. Systèmes Pair à Pair (P2P) :

Stuckenschmidt et Xu définissent l'architecture Pair à Pair (« Peer to Peer », P2P) comme une forme d'informatique distribuée qui implique un grand nombre de nœuds (les pairs) informatiques autonomes et coopératifs qui communiquent et partagent des ressources et des services disponibles, en l'absence de tout contrôle centralisé. De tels systèmes sont hautement dynamiques : les nœuds rejoignent ou quittent le système et changent leurs contenus et de rôles constamment.

D'après Zhuge, un pair peut jouer le rôle de serveur lorsqu'il fournit des données, de l'information et des services, de médiateur lors de l'acheminement des besoins de requêtes, et de client lorsqu'il accède à l'information détenue par d'autres pairs.

5.2. Exemple des systèmes P2P :

Plusieurs travaux concernant les systèmes P2P ont été développés pour différentes activités telles que le traitement de requêtes, la gestion de la connaissance et l'échange d'information à l'intérieur d'un groupe de travail. Nous présentons des exemples de systèmes P2P pour chacune de ces activités : Vdovjak ont décrit des architectures et des systèmes P2P qui soumettent des requêtes à des sources de données RDF. Ce travail montre comment dans des systèmes P2P tels que Gnutella et Freenet les ressources sont atteintes en utilisant une stratégie de routage générique

5.3. Systèmes Multi-Agents SMA :

Carabelea définissent un SMA comme « une fédération d'agents logiciels qui interagissent dans un environnement partagé coopérant et coordonnant leurs actions compte tenu de leurs buts et de leurs plans ». D'après Korhonen, les agents utilisent une plateforme qui fournit une infrastructure partagée et une interface pour l'envoi et la réception de messages. Ces messages peuvent être utilisés pour accéder à un SIW et pour échanger de l'information.

Durant la conception d'un SMA, il est nécessaire de définir les différents rôles que les agents peuvent jouer (par exemple, client, serveur, modérateur, coordinateur, etc.). Il est aussi nécessaire de préciser la manière dont un agent peut communiquer avec d'autres agents pour accomplir des tâches propres et/ou collectives. Panti décrit un Système Multi Agent Pair basé sur le concept « pair ». Un pair représente une personne ou une organisation et possède

un agent qui gère l'information qui lui est associée. Il possède aussi la capacité de gérer et de contrôler un « workflow » simple.

Concept et définition d'un agent :

Le concept d'agent est le résultat de plusieurs années de recherche en IA et en robotique. L'idée d'une entité conceptuelle capable de réaliser des tâches au profit d'un utilisateur était déjà bien établie au milieu des années 70, de cette origine, découlent des bases théoriques telles que les concepts de raisonnement, de représentation de connaissances et d'apprentissage.

L'une des définitions d'agent les plus communes a été fournies par Ferber [11]:

« Un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui, dans un univers multi agent, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents. »

Les Avantages des SMA [12]:

Parmi les plus grands avantages des SMA, nous citons les suivants :

- Augmente la productivité individuelle ou de groupe.
- Réduit la quantité d'information présentée
- Réduit la charge du travail.
- Réduire le temps de formation.

Propriétés des Systèmes multi agents :

Les systèmes multi agents sont caractérisés par :

- Chaque agent a des informations ou des capacités de résolution de problèmes limitées, ainsi chaque agent a un point de vue partiel
- Les données sont décentralisées
- Le calcul et les traitements sont asynchrones
- La flexibilité et le traitement des systèmes à grandes échelles : on peut toujours augmenter le nombre d'agents pour traiter des systèmes de plus en plus gros ; sans pour autant perturber le travail des agents existants
- L'efficacité des traitements : les agents travaillent en parallèle et communiquent de façon asynchrone

5.4. Exemple des systèmes Multi-Agent :

KODAMA est un système multi agent conçu pour livrer de l'information publicitaire aux clients des centres commerciaux à Nagoya (Japon).

Chaque client est muni d'un téléphone portable avec un transmetteur qui envoie des signaux aux récepteurs situés dans le centre commercial. Le récepteur le plus proche du téléphone portable du client active les agents du système de publicité (afin de chercher l'information sur les magasins qui sont proches du client) et envoie des messages au téléphone du client par le biais d'emails. Son architecture multiniveaux modélise des systèmes distribués basés sur des agents.

6. Conclusion :

On a conclu dans ce chapitre l'état de l'art essentiel à chaque projet de fin d'étude, où on a mis en œuvre une description explicite sur les différents services présents dans notre thèse liée au développement mobiles, en passant par les étapes qu'il faut prendre pour atteindre la satisfaction du développement.

Les technologies ubiquitaires et celles liées au SMA et P2P sont conformes à nos besoins de localisation et d'actualisation de contexte, et donnent à l'utilisateur un environnement mobile dynamique, lui offrant le résultat propre à ces préférences.

II. Chapitre 2 : Etat de l'art :

1. Introduction :

Dans le présent chapitre nous allons définir les systèmes de recommandation ainsi que leurs différents types, le but de ces systèmes et les avantages / inconvénient de chaque type. Ce chapitre permet, en outre, d'identifier les limites des solutions/ travaux existantes par rapport à nos objectifs, mais aussi la naissance d'une nouvelle solution, les SRSC (systèmes de recommandation sensible au contexte), et nous terminons notre chapitre par une brève conclusion qui résume l'idée générale du chapitre.

2. Les Systèmes de recommandations :

Dans la partie suivante, nous allons définir les systèmes de recommandations et à quoi consiste exactement.

2.1. Définitions :

- Un système de recommandation est une forme spécifique de filtrage de l'information visant à présenter les éléments d'information (films, musique, livres, news, images, pages Web...) qui sont susceptibles d'intéresser l'utilisateur. [13]
- Les systèmes de recommandation ou des systèmes de recommandation sont une sous-classe de système d'information de filtrage qui cherche à prédire la «note» ou «préférence» cet utilisateur donnerait à un élément. [14]
- Dit autrement, un système de recommandation cherche à prédire la valorisation ou préférence qu'un utilisateur attribuerait à un objet (livre, musique, film...) ou à un élément social (personne, groupe, communauté) qu'il n'avait pas encore considéré.
- Un système de recommandation (RS) aide les utilisateurs qui n'ont pas suffisamment d'expérience ou la compétence nécessaire pour évaluer le nombre, potentiellement important, d'alternatives offertes par un site (web). [15]
- Dans leur forme la plus simple les RS :
 - recommandent à leurs utilisateurs des listes personnalisées et classées d'articles
 - fournissent aux consommateurs des renseignements pour les aider à décider quels articles acheter [15]

Mais quelles sont les étapes de recommandations ??

2.2. Les Etapes de recommandation :

Un système de recommandation requiert généralement 3 étapes:

- La première consiste à recueillir de l'information sur l'utilisateur.
- La deuxième consiste à bâtir une matrice ou un modèle utilisateur contenant l'information recueillie.
- La troisième consiste à extraire à partir de cette matrice une liste de recommandations. [16]

Il existe plusieurs types de SR, nous citons les principaux dans le prochain titre !

2.3. Type des systèmes de recommandations :

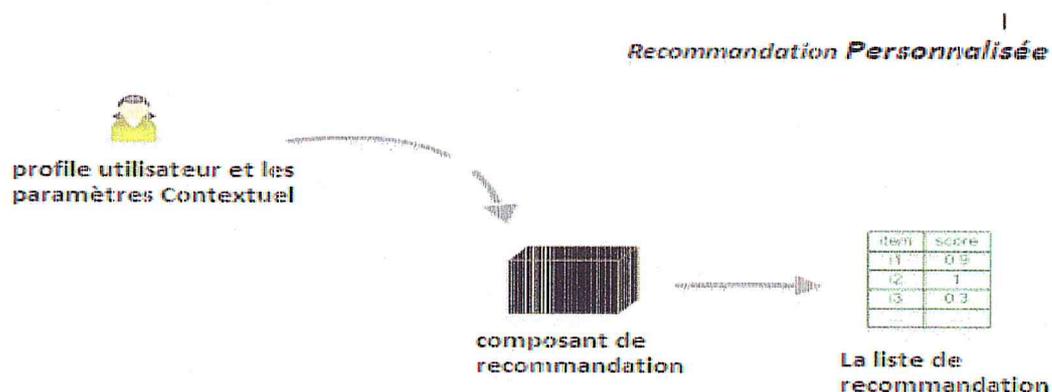
On a l'habitude de présenter 4 approches possibles pour un système de recommandation:

- a) Personnalisée [Personalized recommendations].
- b) Le filtrage collaboratif – Collaborative filtering.
- c) Les recommandations basées sur le contenu – Content-based.
- d) Les approches hybrides [17].

a) Recommandation Personnalisée :

Il s'agit de recommander des objets sur la base du **comportement passé de l'utilisateur**.

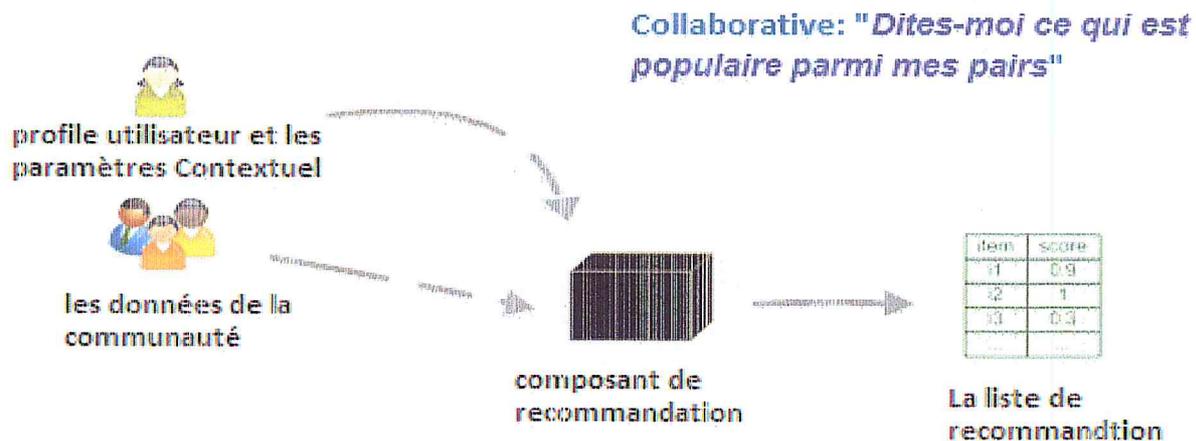
Par exemple, Des produits achetés ou sélectionnés sur un site de e-commerce, ainsi qu'un certain nombre d'actions ou décisions effectuées par l'utilisateur qui permettent de prédire de nouveaux produits susceptibles de l'intéresser [18]



-Figure 3- Recommandation Personnalisée.

b) Filtrage collaboratif :

Les systèmes basés sur le filtrage collaboratif produisent des recommandations en calculant la similarité entre les préférences d'un utilisateur et celles d'autres utilisateurs. De tels systèmes ne tentent pas d'analyser ou de comprendre le contenu des items à recommander. Ils sont capables de suggérer de nouveaux items à des utilisateurs qui ont des préférences similaires à d'autres.



-Figure 4- Filtrage collaboratif.

La méthode consiste à faire des prévisions automatiques (filtrage) sur les intérêts d'un utilisateur en collectant des avis de nombreux utilisateurs (collaborateurs).

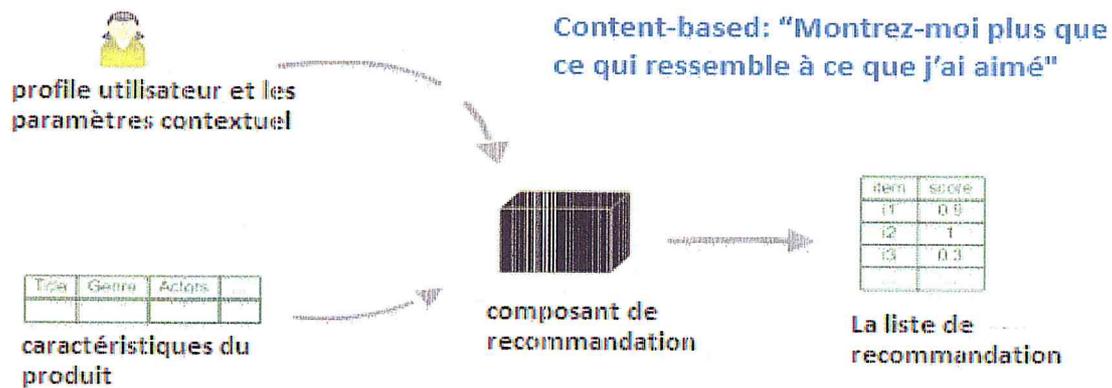
Idée :

- Essayer de prédire l'opinion de l'utilisateur qu'il aura sur les différents items et être en mesure de recommander le «meilleur» item à chaque utilisateur en fonction de : des goûts/avis précédents de l'utilisateur et des avis d'autres utilisateurs qui lui sont semblables [19]

c) Recommandation basée sur le contenu :

- Pour les recommandations basées sur le contenu, le système essaie de recommander des items qui correspondent au profil de l'utilisateur.
- Le profil est basé sur les items que l'utilisateur a aimés dans le passé ou sur les intérêts qu'il a explicitement définis.
- Un RS basé sur le contenu fait coïncider le profil de l'item avec le profil de l'utilisateur pour décider de sa pertinence pour l'utilisateur.

- Cette méthode permet de développer des modèles afin de trouver des patterns ou motifs semblables entre différentes données. Ils évaluent à quel point un contenu pas encore vu par l'utilisateur est similaire aux contenus que celui-ci a évalués positivement dans le passé. Pour ce faire, on utilise la notion de similarité qui peut être mesurée de plusieurs manières. [20]



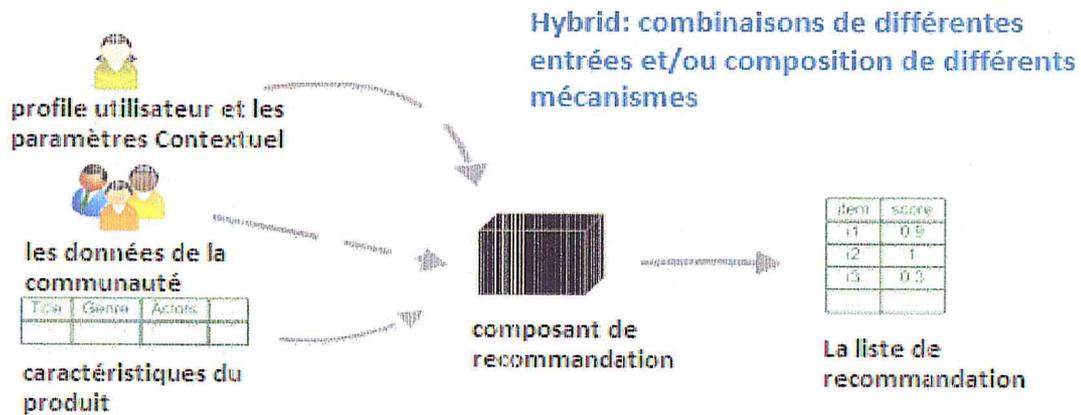
-Figure 5- Recommandation basée sur le contenu.

Idée :

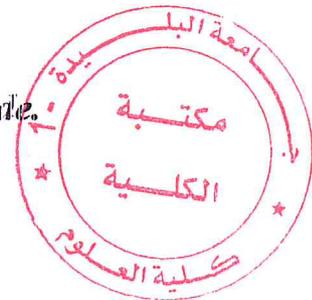
- Il s'agit de recommander des objets (ou contenus) en se basant sur les qualités et propriétés intrinsèques de l'objet lui-même et en les corrélant avec les préférences et intérêts de l'utilisateur. Ce type de système va donc extraire un certain nombre de caractéristiques et attributs propres à un contenu, afin de pouvoir recommander à l'utilisateur des contenus additionnels possédant des propriétés similaires. Cette méthode crée un profil pour chaque objet ou contenu, c'est-à-dire un ensemble d'attributs/propriétés qui caractérisent l'objet. [20]

d) Recommandation hybride :

Combinaison du filtrage collaboratif et basé sur le contenu. Une **combinaison des trois approches ci-dessus**. Les méthodes hybrides sont de plus en plus utilisées, car elles permettent de résoudre des problèmes comme le *cold start* et la rareté qu'on retrouve dans une approche de recommandation uniquement sociale. [21]



-Figure 6- Recommendation hybride.



e) Discussion : [22]

Le tableau suivant (Tableau 01) présente la différence entre les types cité précédemment, notamment les avantages et les inconvénients.

Avantage :

- A. Utilise des recommandations d'autres utilisateurs (score) pour évaluer l'utilité des items.
- B. L'essentiel est de trouver des utilisateurs ou groupes d'utilisateurs dont les intérêts correspondent à l'utilisateur courant.
- C. Plus il y a d'utilisateurs, plus il y a de scores : meilleurs sont les résultats.
- D. Le profil des utilisateurs est la clé.
- E. Le matching entre les préférences de l'utilisateur et les caractéristiques des items fonctionne aussi pour les données textuelles.
- F. Pas besoin de données sur les autres utilisateurs.
- G. Pas de problème de démarrage à froid ou de faible de.
- H. Sensible à la variance à court terme.
- I. Peut identifier des niches précisément.
- J. Recommander des items similaires à ceux que les utilisateurs ont aimés dans le passé.

Inconvénient :

- A. Trouver des utilisateurs ou groupes d'utilisateurs similaires est difficile.
- B. Démarrage à froid :
 - a. Nouvel utilisateur = Pas de préférences
 - b. Nouvel item = Pas de score
- C. Des items représentés par le même ensemble de mots-clés ne peuvent pas être distingués.
- D. Les utilisateurs avec des milliers d'achats/items sont un problème.
- E. Nouvel utilisateur : pas d'historique.
- F. Impossible d'exploiter les jugements des autres utilisateurs.

Techniques	Avantages	Inconvénients
<i>Recommandation basée sur le contenu</i>	D, E, F, G, H, J	N, O, P, Q, R
<i>Filtrage collaboratif</i>	A, B, C, I	L, M
<i>hybride</i>	A, B, C, D, E, F, G, H, I, J	R

-Tableau 1- Comparaison entre les types de recommandation.

3. Systèmes de recommandation Sensible au contexte :**3.1. La naissance des SRSC :**

Une nouvelle famille de systèmes de recommandation appelés **Systèmes de Recommandation Sensibles au Contexte (SRSC)** se développe de plus en plus dans la littérature comme par exemple (Bader et al. 2011) (Baltrunas et al. 2011), (Woerndl et al. 2011) qui se basent sur le contexte de l'utilisateur en modélisant des données comme sa localisation, l'heure et les personnes aux alentours. La principale limite de ces approches est qu'elles ne tiennent pas compte de la dynamité des contenus.

Une façon d'appréhender le problème des systèmes de recommandation, à savoir présenter à l'utilisateur des contenus pertinents, est d'utiliser l'apprentissage automatique pour faire des prédictions sur des données non précédemment vues en apprenant de l'expérience passée. Ce problème s'apparente au problème du bandit puisqu'il peut se traduire comme un compromis entre exploration et exploitation : dans la phase d'exploration, le but est d'obtenir des récompenses pour améliorer la qualité des prédictions ; dans la phase d'exploitation, le but est de présenter des items avec la plus haute évaluation avec les informations obtenues jusqu'ici.

3.2. Définition:

- On dit d'un système qu'il est sensible au contexte s'il peut tirer, interpréter et utiliser des informations issues du contexte et adapter sa réponse en fonction du contexte d'utilisation [23].
- Les systèmes de recommandations sensibles au contexte (SRSC) combinent les caractéristiques des systèmes sensibles au contexte et des systèmes de recommandation afin de fournir des informations personnalisées aux utilisateurs dans des environnements ubiquitaires. Dans cette perspective où tout ce qui concerne l'utilisateur est dynamique, les contenus qu'il manipule et son environnement, deux questions principales doivent être adressées :
 - Comment prendre en compte la dynamique des contenus de l'utilisateur ?
 - Comment éviter d'être intrusif en particulier dans des situations critiques ?

La plupart des systèmes de recommandation sensibles au contexte se concentrent d'avantage sur la recommandation de documents pertinents à un utilisateur tout en intégrant des données contextuelles sans pour autant considérer le problème de l'évolution des contenus.

3.3. Les caractéristiques d'un SRSC :

Un bon SRSC doit être :

- Facile à utiliser.
- Répondre aux besoins des utilisateurs.
- Produit des résultats précis et explicites.
- Fait des recommandations suivant les différents contextes qu'il exploite.
- Prendre en compte les aspects temporels, géographiques, ...

3.4. L'utilité d'un SRSC :

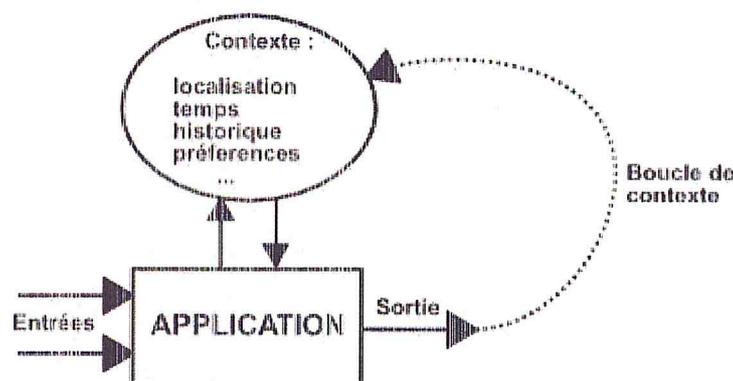
- Améliore les techniques de filtrage et de recommandation.
- Il aide à Combiner les différentes techniques existantes.
- Un outil d'aide à la décision.
- Gain de temps et de cout.

Exemple :

Après que l'application recommande un hôtel, et que le l'utilisateur fait une réservation dans cet hôtel, il est invité après à donner une note à l'hôtel et à cette recommandation indiquant sa satisfaction, sans oublié que cette recommandation doit respecter le contexte de temps, de lieu et de préférences de l'utilisateur.

4. L'intégration de contexte dans les systèmes de recommandation : [24]

L'intégration du contexte au sein des SR, plus précisément dans les techniques précédentes et un grand défi, Les applications qui utilisent la notion de contexte sont de plus en plus nombreuses. Devant la multiplication et les avancées logicielles du domaine, des équipes de recherche se sont intéressées à la définition d'une architecture commune pour les applications contextuelles, celles-ci devant permettre l'interopérabilité et l'intégration de contextes de plus en plus riches sans pour autant avoir à reprendre toute l'architecture du logiciel de zéro.



-Figure 7- Application tenant compte du contexte.

4.1. Types d'intégration de contexte [25] :

En particulier, du point de vue algorithmique, les SRSC travailleraient avec les données de la forme $U \times I \times R \times C$, où C est un contexte supplémentaires, et essaye de produire une liste

de recommandations contextuelles i_1, i_2, i_3, \dots pour chaque utilisateur U , l'intégration est faite grâce à l'une des technique suivante, en fonction de la façon dont l'information contextuelle est utilisé, pré filtrage contextuelle, post filtrage contextuelle, ou modélisation contextuelle.

4.1.1- Pré-filtrage contextuelle :

L'intégration du contexte est faite avant la production du résultat et de la liste de recommandation, le système utilise l'information contextuelle pour sélectionner les données les plus pertinents afin de les générer pour faire la meilleure recommandation. Un avantage majeur de cette approche est qu'elle permet un déploiement de l'un des de nombreuses techniques traditionnelles de recommandation précédemment proposées dans la littérature (Adomaviciuset Tuzhilin 2005).

En particulier, l'utilisation de cette approche, le contexte C sert essentiellement à une requête (ou un filtre) pour sélectionner des données pertinents.

Un exemple d'un filtre de données contextuelles pour un SR de cinéma: si une personne veut voir un film le samedi, seules les données de note samedi sont utilisées pour recommander des films. Notez que cet exemple représente un préfiltre exacte parce que le des données ont été filtré en utilisant exactement le contexte spécifié.

4.1.2 - Post-filtrage contextuelle :

L'intégration du contexte est faite à la fin de la procédure de recommandation, donc le système ignore l'information contextuelle de début de la recommandation, mais il l'utilise à la fin pour ajuster la liste de recommandation obtenue pour chaque utilisateur.

Les ajustements de la liste de recommandation peut être faite par: (1) de filtrage des recommandations qui ne sont pas pertinents dans un contexte donné, ou (2) par réglage du classement des recommandations de la liste.

Par exemple, dans une application de recommandation de film, si une personne veut voir un film dans le week-end, et que cette personne préfère des films de comédies, le système peut tous filtrer non-comédies de la liste recommandée.

Comme avec de nombreuses techniques de recommandation, les approches post-filtrage contextuels peuvent être classées dans des techniques basées sur heuristique et modèle.

4.1.3 - Modélisation contextuelle :

La modélisation contextuelle utilise les informations contextuelles directement dans la fonction de recommandation, autrement dit le système utilise le contexte durant toute la procédure de recommandation.

Il y a eu aussi quelques nouvelles techniques développées spécifiquement pour la sensibilité au contexte des systèmes basés sur la modélisation contextuelle.

Par exemple, Oku et al. (2006) propose d'intégrer les dimensions contextuelles supplémentaires directement dans l'espace et l'utilisation des techniques d'apprentissage automatique pour fournir des recommandations pour un système d'un restaurant.

5. Exemples des SRSC:

5.1. Pandora Radio : [26]

Pandora utilise principalement une approche de recommandation objet (basées sur le contenu). Le système se base sur les propriétés de la chanson et/ou de l'artiste, c'est à dire il décompose dans un premier temps les morceaux de musique afin de mettre en évidence ses propriétés intrinsèques. Chaque morceau de musique disposerait de plus de 400 attributs, permettant de déterminer une sorte de code génétique de là. En comparant les propriétés similaires entre différents morceaux, le système arrive alors à nous proposer de nouvelles chansons à écouter, c'est-à-dire une sorte de station radio qui diffuse de la musique avec des propriétés susceptibles de correspondre à nos goûts. L'utilisateur peut ensuite indiquer s'il aime ou pas le morceau en écoute, et permet ainsi à Pandora d'affiner son filtre de recommandation en privilégiant (ou pas) certains attributs selon l'information venue en retour de l'utilisateur.

Avantages	Inconvénients
<ul style="list-style-type: none">• Souplesse d'utilisation• Sélectivité selon le profil des lecteurs	<ul style="list-style-type: none">• Encombrement publicitaire

-Tableau 2- Avantages et inconvénients de Pandora Radio.

5.2. NetFlix : [27]

Le principe (et succès) de Netflix s'appuie sur la recommandation sociale (filtrage collaboratif) de films. 75% des inscrits utilisent la plateforme principalement pour son moteur de recommandation. Grâce à un algorithme secret et aux millions de données récoltées, Netflix pousse des propositions personnalisées à chaque nouvelle connexion. Il analyse autant les films recherchés et visionnés par les utilisateurs que les notations données à chaque film par ces derniers.

Netflix fait ce qu'on appelle « **une recommandation hybride** ». D'une part, la plateforme collecte les données au niveau de l'objet lui-même, le film. Il fait le lien entre les caractéristiques du film (son genre, les acteurs qui y jouent) et l'historique de l'utilisateur. D'autre part, elle compare également des données sur des utilisateurs ayant des goûts similaires, susceptibles d'aimer le même genre de film, ce dernier point devenant pertinent avec un nombre très élevé d'abonnés. Bref, un service complet qui répond le plus précisément possible aux goûts et intérêts des internautes.

Avantages	Inconvénients
<ul style="list-style-type: none">• Facile d'utilisation et extrêmement mobile	<ul style="list-style-type: none">• Un service payant

-Tableau 3- Avantages et inconvénients de NetFlix.

5.3. Sailendra [28] :

Moteur de filtrage collaboratif pour les produits, documents ou articles (indépendant de la langue et du format).

Avantages	Inconvénients
<ul style="list-style-type: none">• Proposez des recommandations pertinentes, personnalisées et de qualité.	<ul style="list-style-type: none">• Service payant

-Tableau 4- Avantages et inconvénients de Sailendra.

6. Conclusion :

Nous avons présenté dans ce chapitre la notion d'un système de recommandation et ces différents types, avec les points forts et faibles de chacun d'eux, mais comme cité ce dernier n'est pas suffisant pour répondre au besoin de l'utilisateur et surtout gérer la dynamique des données, ce qui a invoqué la naissance des SRSC qui prennent en considération le contexte à fin de satisfaire l'utilisateur en lui faisant part du système avec le déploiement de ses intérêts et de son type profil sur la plupart des fonctionnalités de l'application.

On va aborder dans le chapitre suivant beaucoup de travaux existants, qui utilisent les SRSC pour résumer le comportement de l'utilisateur en un système intelligent, pour la bonne interaction avec les préférences de l'utilisateur.

III. Chapitre 3 : Travaux Existants :

1. Introduction :

Les systèmes de recommandation sont des outils et des techniques logiciels de suggestions pour des articles qui devraient être utiles pour un utilisateur donné, mais bien sûr sans tenir compte de l'information contextuelle de l'utilisateur.

Les systèmes de recommandation traditionnels sont limités et ne satisfont pas vraiment l'utilisateur, contrairement aux SRSC qui tiennent compte de l'utilisation de l'information du contexte pour optimiser la liste de recommandation. Pour cette raison, les SRSC gagnent de plus en plus d'attention dans plusieurs domaines d'application.

2. Domaine d'utilisation et utilités des SRSC :

La recommandation de l'information a été un sujet très populaire pour les flux des systèmes de recommandations, et surtout dans la recommandation des nouvelles pages web. Dans ce domaine, les SR ont été développés avec les technologies basées sur le contenu ou l'hybridation de contenu avec des approches de collaboration.

Les SRSC sont utilisés dans plusieurs domaines tels que le cadre commercial, publicitaire et même éducatif. Le domaine d'application qui a reçu la plus grande attention est le tourisme. Les fonctions prises en charge par des guides touristiques sont liées à trouver des services pertinents, ou de soutenir l'exploration d'une zone. La position de l'utilisateur est souvent utilisée pour personnaliser les résultats, comme les attractions de la ville (les musées, les hôpitaux, etc...) sont les articles les plus recommandés de ces systèmes de recommandations, et ils sont visualisés soit dans une interface basée sur la liste traditionnelle, ou comme un ensemble de recommandations sur une carte.

3. Travaux existants :

Nous allons maintenant illustrer quelques exemples de systèmes d'information et SRSC développés spécialement pour le scénario mobile, en analysant les différents aspects touchés du contexte, les différentes méthodes utilisées, leurs caractéristiques et les insuffisances de chacun.

En résumé, il existe trois différents types de travaux et on va les aborder selon leurs simplicités, du plus simple vers le plus compliqué :

- Travaux basés sur les contextes pertinents.
- Travaux basés sur la corrélation du contexte et les préférences de l'utilisateur.

- Travaux basés sur la corrélation du contexte / préférences et le choix du type de la recommandation.

3.1. Les Travaux basés sur les contextes pertinents :

Dans cette partie nous allons parler des SRSC qui utilisent les contextes les plus pertinents, mais c'est quoi un contexte pertinent ?

Un contexte pertinent est un ensemble des conditions naturelles, sociales, culturelles qui jouent un rôle principal dans la recommandation, comme le contexte du temps et du lieu (localisation).

L'acquisition du contexte se fait d'une manière implicite, le système collecte les informations nécessaires en observant les interactions de l'utilisateur avec ce dernier.

a) UbiquiTO [29]:

C'est un guide touristique qui utilise le contexte d'une façon intelligente. Il intègre les différentes formes d'adaptation de contexte lié: le type de périphérique multimédia, les caractéristiques et les préférences des utilisateurs, au contexte physique de l'interaction (emplacement et temps). UbiquiTO utilise une approche de modélisation pour adapter le contenu de la recommandation prévue, telles que la quantité / le type d'informations/ Caractéristiques associées à chaque recommandation.

Algorithmes d'intégration	Attribut de contexte	Insuffisance
<ul style="list-style-type: none"> • Modélisation Contextuelle 	<ul style="list-style-type: none"> • Contexte de Lieu • Type de l'appareil • Temps (l'heure du jour) 	<ul style="list-style-type: none"> • Peu d'attribut de contexte

-Tableau 5- Analyse et critique d'UbiquiTO.

b) MoMa [30]:

MoMa est un SRSC des annonces (vente/achat), (Bulander et al. 2005) propose des recommandations proactives en utilisant un post-filtrage approche pour faire correspondre les spécifications de la commande avec des offres.

Lors de la création d'une commande, l'application client remplit automatiquement dans le contexte du profil appropriée des paramètres physiques, exemple « localisation ».

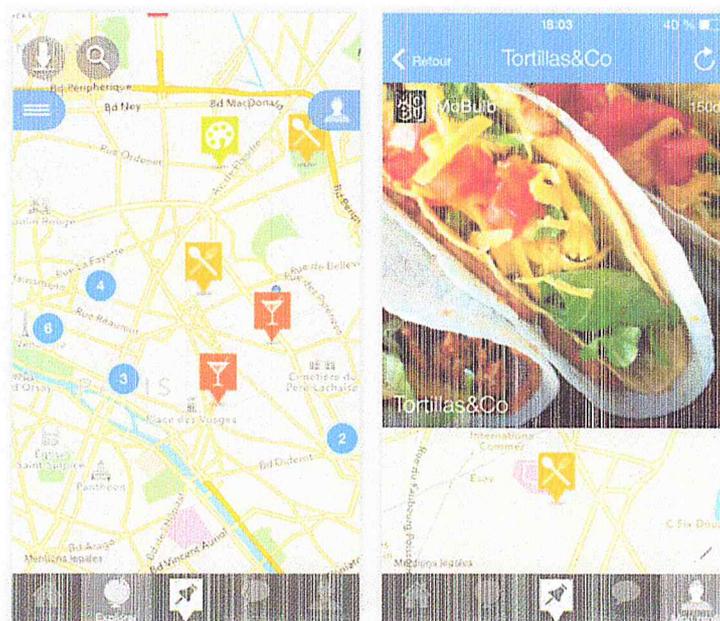
De l'autre côté, les fournisseurs des annonceurs mettent leurs offres dans le système MoMa. Ces offres sont également formulées selon le catalogue. Lorsque le système détecte une paire de contexte, l'utilisateur est notifié, de la manière préférée (par exemple, SMS, e-mail). Et donc, l'utilisateur peut décider de contacter l'annonceur pour accepter l'offre.

Algorithmes d'intégration	Attribut de contexte	Insuffisance
<ul style="list-style-type: none"> • Post-filtrage contextuelle 	<ul style="list-style-type: none"> • Contexte de Lieu • Temps 	<ul style="list-style-type: none"> • le types de l'appareil ne sont pas présent en considération dans la recommandation.

-Tableau 6- Analyse et critique de MoMa.

c) Deapz [31]:

Deapz est une sorte de réseau social, ou plutôt de « carnet d'adresses social », qui se présente sous la forme d'une application iPhone très visuelle, accordant beaucoup d'importance aux photos. Le concept est de permettre aux gens de noter sur une carte les lieux qu'ils aiment, que ce soit des restaurants, des points de vue, des musées, des ccins de nature, et autres lieux intéressants. Le but est de pouvoir les garder en mémoire, mais surtout de pouvoir les partager avec leurs amis. Le partage se fait directement dans l'application, mais aussi grâce à un système qui permet de partager sa carte et ses lieux avec les gens qui n'ont pas l'application, directement sur Facebook, Twitter, ou grâce à un simple lien.



-Figure 8- l'interface Deapz.

Algorithmes d'intégration	Attribut de contexte	Insuffisance
<ul style="list-style-type: none"> • Pré-filtrage contextuelle 	<ul style="list-style-type: none"> • Contexte de Lieu • Type de l'appareil • Temps 	<ul style="list-style-type: none"> • Peu d'attribut de contexte • N'est pas disponible sur toutes les plateformes mobiles (Ex : Android)

-Tableau 7- Analyse et critique de Deapz.

Conclusion :

On vient de voir certains travaux basé sur les contextes pertinents, où l'utilisateur lui-même cause le changement dans la recommandation, tout dépend de ses choix et comportements avec les contextes principaux qui sont le temps et le lieu.

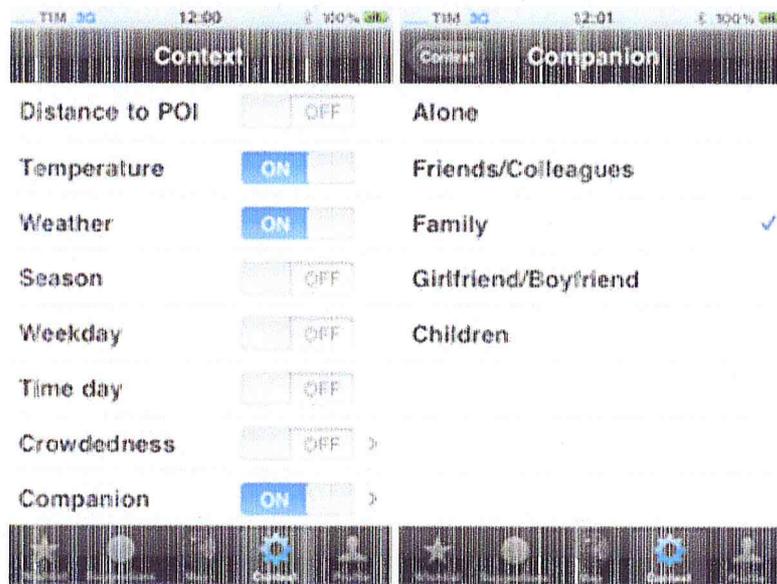
3.2. Travaux basé sur la corrélation entre le contexte et les préférences :

Maintenant dans ce qui suit nous allons voir des travaux du même contexte mais aussi avec une certaine corrélation entre le contexte et les préférences pour optimiser leurs résultats.

a) ReRex [32]:

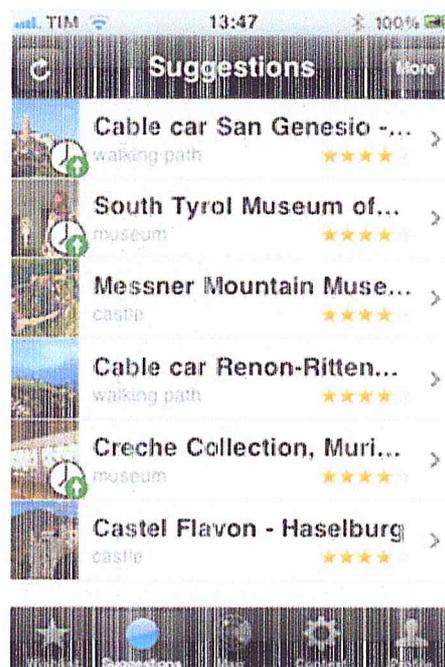
ReRex offre des recommandations des lieux, mais identifie également les états contextuels avec le plus grand impact. Cette condition est utilisée comme un argument pour expliquer ce qui rend le produit plus adaptée à la situation actuelle de l'utilisateur.

ReRex est une application de planification de voyage visant à recommander des points d'intérêt (POI) aux utilisateurs mobiles. Elle offre deux fonctionnalités. Tout d'abord, dépend du contexte. Deuxièmement, aide à la préparation d'un itinéraire complet et la modification en fonction des circonstances et éventualités qui peuvent arrivés pendant l'itinéraire.

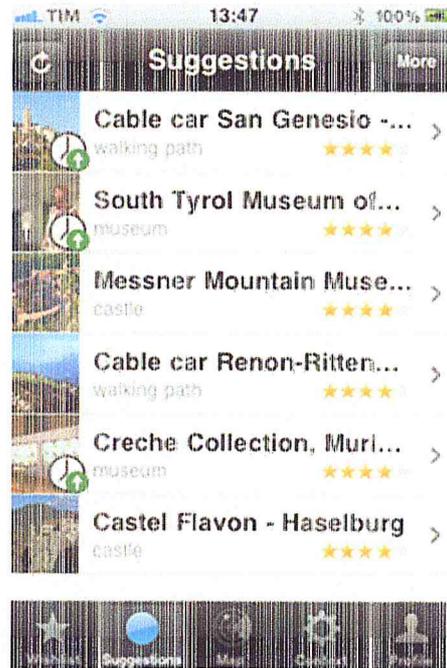


-Figure 9-- Interface utilisateur pour la spécification du contexte.

En particulier, l'utilisateur peut demander des conseils dans un contexte particulier. L'application présente les recommandations générées par le modèle prédictif et justifie les recommandations avec l'explication de la raison principale pour laquelle un élément est recommandé. Ainsi, le système peut justifier la recommandation d'un POI particulier (Ex : Musée) parce que l'utilisateur se déplace avec les enfants. Par ailleurs, l'application mobile peut notifier de façon asynchrone l'utilisateur que l'état du contexte (par exemple, temps) est susceptible de changer et réviser donc les recommandations pour l'utilisateur.



-Figure 10-- La liste recommandée.



-Figure 11– Détail d'un choix de la recommandation.

La recommandation se fait après que l'utilisateur choisit le contexte voulu pour que l'application calcul la liste de recommandation, le système cherche des comportements déjà passé où bien fait un nouveau calcul, en prenant le contexte sélectionné la corrélation avec les préférences de l'utilisateur est faite, pour donner la recommandation des lieux, voyages etc...

Algorithmes d'intégration	Attribut de Contexte	Insuffisance
<ul style="list-style-type: none"> • Pré-filtrage contextuelle 	<ul style="list-style-type: none"> • Contexte de Lieu • Temps (l'heure du jour) • La saison • Compagnon • La température • La météo 	<ul style="list-style-type: none"> • La recommandation se résume que dans la recommandation des lieux (POI). • N'est pas disponible sur toutes les plateformes mobiles (Ex : Android)

-Tableau 8– Analyse et critique de ReRex.

b) BlinkFeed :

BlinkFeed est une application mobile et un SRSC permet la recommandation des articles tout en tenant compte les préférences de l'utilisateur et les informations contextuelles et de posséder sur son écran d'accueil un flux d'informations. Les sources sont diverses, des

réseaux sociaux aux réseaux d'actualités en passant par le divertissement ou encore les photos. Ainsi, toutes ces informations sont présentes à un seul endroit sur le téléphone et directement au sein de l'écran d'accueil, sans avoir à lancer une application.

Avec plus de 1400 sources et medias différents dont 130 dans l'hexagone, le constructeur assure plus de 10.000 articles par jour disponibles en 16 langues.

L'application propose au début à l'utilisateur de choisir ses préférences comme le montre la figure 12, et puis elle commence à faire des recommandations.

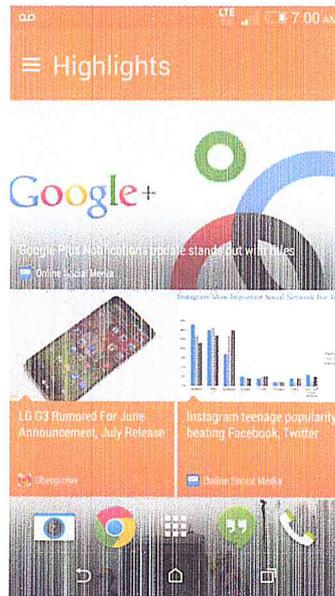


-Figure 12- Interface de validation des préférences.

La recommandation de cette application ne base pas seulement sur le profil utilisateur, mais aussi sur sa localisation pour lui proposer des articles du plus intéressant vers le moins, elle peut même détecter la langue automatiquement en utilisant sa localisation ou bien par la changé manuellement dans les paramètres.

Algorithmes d'intégration	Attribut de contexte	Insuffisance
<ul style="list-style-type: none"> • Pré-filtrage contextuelle 	<ul style="list-style-type: none"> • Contexte de Lieu. • Type de l'appareil • Préférence de l'utilisateur. 	<ul style="list-style-type: none"> • Ne permet pas d'ajouter un site personnel en tant que source d'actualité

-Tableau 9- Analyse et critique de BlinkFeed.



..Figure 13– Interface de la recommandation finale.

e) Flipboard :

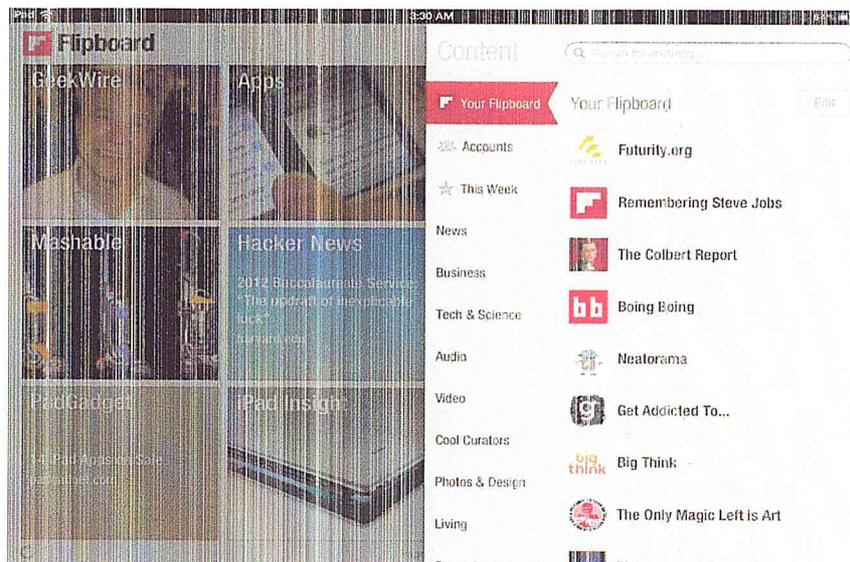
Cette application est un "agrégateur" d'informations qui fonctionne à la manière de Google Actualités. Elle agrège des flux RSS et restitue leur contenu aux utilisateurs. La différence : l'esthétisme. Flipboard propose à l'utilisateur de "composer lui-même son propre magazine, qui ressemble à la version papier mais avec du contenu provenant d'Internet". Pratique pour ceux qui ne veulent plus jongler d'application en application pour connaître les derniers scoops et qui regrettent l'ergonomie et le design du magazine papier.

Cibler ces centres d'intérêts :

L'utilisateur peut lui-même personnaliser son magazine en choisissant de cibler ses centres d'intérêt. Flipboard se charge ensuite de présélectionner du contenu pour répondre à ses attentes. Technologie, politique, musique, mode, économie ou encore cinéma : chacun a la possibilité de créer sa propre interface en fonction de ses goûts et des formats qui l'intéressent (texte, photo ou vidéo). En un "flip", on peut passer d'une information à une autre.

Algorithmes d'intégration	Contexte Utiliser	Insuffisance
<ul style="list-style-type: none"> • Pré-filtrage contextuelle 	<ul style="list-style-type: none"> • Type de l'appareil • Préférence de l'utilisateur. • Temps 	<ul style="list-style-type: none"> • Ne fournis pas des articles spécial pour une localisation donnée (un pays par exemple)

-Tableau 10- Analyse et critique de Flipboard



-Figure 14– Interface Flipboard.

Conclusion :

Les applications qui font la corrélation entre le contexte et les préférences, ont plus de pouvoir à restreindre la recommandation pour l'utilisateur, du plus pertinent vers le moins voulue en analysant bien ses préférences enregistrer lors de la création de son profil.

3.3. Travaux basé sur la corrélation du contexte / préférences et le choix du type de la recommandation:

Ce type est le plus évolué de tous les SRSC, car ce dernier est basé sur la corrélation du contexte et des préférences de l'utilisateur comme les travaux qu'on a précédemment présenté, mais aussi le bon choix du type de la recommandation avant de donner des résultats.

Car faire une telle recommandation sans choisir le bon type, peut souvent donner des résultats inutiles ou bien pas précis.

Pour cela dans la partie suivante, nous allons présenter des travaux qui englobent tous ces critères, et se concentrer surtout sur les travaux utilisant les types de recommandation suivant :

- Le filtrage collaboratif
- Les recommandations basées sur le contenu

3.3.1- SRSC utilisant le filtrage collaboratif :

Les SRSC utilisant le filtrage collaboratif font des recommandations en effectuant une corrélation entre des utilisateurs ayant des préférences et intérêts similaires. On utilise des méthodes qui collectent et analysent des données sur le comportement, les activités, les

préférences des utilisateurs et des algorithmes tentent de prédire ce que l'utilisateur aimera en cherchant des utilisateurs qui ont les mêmes comportements que l'utilisateur à qui l'on souhaite faire des recommandations.

L'idée sous-jacente est de dire que si une personne a la même opinion (ou les mêmes goûts) qu'une personne B sur un objet x, alors la personne a plus de chance d'avoir la même opinion que B sur un autre objet y, plutôt que d'avoir la même opinion que quelqu'un choisi au hasard pour l'objet y.

Le contexte utilisé dans ce cas-là est le lieu, généralement le lieu fait des recommandations presque similaires au gens d'une même région (mécanisme basé sur le voisinage proche).

Exemple : Play Store

Le meilleur exemple qu'on peut donner sur ça est Le Play Store de Google, un magasin en ligne d'application mobile sous-système Android.

Cette application utilise comme contexte le temps, le lieu et le type de l'appareil, par exemple les recommandations faites pour quelqu'un qui est en Algérie n'est pas la même pour quelqu'un d'autre en France, s'il s'agit de la fin de l'année l'application propose des fonds d'écran spécial pour cette période. Les recommandations pour un téléphone mobile et une tablette sont pas les mêmes car la taille de l'écran, la performance voir même la compatibilité se diffère.



-Figure 15- Recommandation personnalisée du Play Store en utilisant le contexte de lieu.

Après avoir téléchargé une application, le système propose à l'utilisateur d'autres applications à télécharger et ça est calculé sur deux façons (voir figure 16) :

- Soit d'autres utilisateurs ont un profil similaire que lui, ont téléchargé la même application (filtrage collaboratif).
- Soit d'autres utilisateurs ayant un profil similaire se trouvent dans la même région que lui (mécanisme basé sur le voisinage proche).



-Figure 16- Recommandation en utilisant le calcul de la similarité et le contexte de lieu.

3.3.2- SRSC utilisant des recommandations basées sur le contenu :

Ce type de système va extraire un certain nombre de caractéristiques et attributs propres à un contenu, afin de pouvoir recommander à l'utilisateur des contenus additionnels possédant des propriétés similaires. Cette méthode crée un profil pour chaque objet ou contenu, c'est-à-dire un ensemble d'attributs/propriétés qui caractérisent l'objet.

Il s'agit donc de recommander des objets en se basant sur les qualités et propriétés intrinsèques de l'objet lui-même et en les corrélant avec les préférences et intérêts de l'utilisateur tout en tenant compte du contexte, et parmi les travaux les plus référencés de ce type nous avons :

- **ReRex** : comme on a déjà parlé de cette application, ReRex est un système de planification de voyage visant à recommander des points d'intérêt aux utilisateurs mobiles, ReRex fait des recommandations en se basant sur les caractéristiques des

voyages et des préférences de l'utilisateur, en faisant le matching de ces caractéristiques et en ajoutant les contextes choisis par l'utilisateur ou bien les détecter automatiquement, pour présenter la liste finale de recommandation.

3.4. Nos Exemples :

- Pour décider la suggestion d'un article ou non, le système peut se baser sur les mots-clés principaux de l'article et les comparer avec les mots-clés apparaissant dans d'autres articles que l'utilisateur a évalués positivement dans le passé.
- Dans le même exemple, recommander un article de sport au weekend à un utilisateur qui habite en Algérie, sachant que cet utilisateur a déjà fait ce comportement, jusqu'ici on a utilisé une recommandation basée sur le contenu et on a touché le contexte de temps et de lieu tout en respectant les préférences de l'utilisateur.

3.5. Autre Travaux du même type [33] :

- NAMA (Kwon et al. 2005) est un système de rappel qui fournit des services personnalisés sensibles au contexte. Par exemple, au shopping, s'il y a certains produits que les utilisateurs ont marqués dans leur to-do-list (à faire) près de chez eux, le système leur recommande la liste enregistrée.
- Doulkeridis et al. (2006) exige aux utilisateurs d'entrer les mots-clés pour l'information qu'ils veulent pour fournir les services et satisfaire son contexte et les préférences simultanément comme un moteur de recherche web. Dans ce cas-là, si l'utilisateur a besoin de quelques photos, le système lui fournit des services basés sur le contexte tels que le statut actuel des dispositifs.
- GUIDE fournit l'information personnalisée avec les voyageurs basée sur les profils des utilisateurs ou la préférence de l'utilisateur, tels que l'intérêt de l'utilisateur et la langue, enregistrés dans l'application (Cheverst et al. 2000).
- Dans la recherche d'information et la classification des documents, Middleton et al. (2004) ont développé un système de recommandation basé k-NN qui recommande les documents de recherche sur la base de la similarité des préférences des utilisateurs et utilise l'ontologie pour analyser les profils des utilisateurs.
- Shardanand et Maes (1995) ont suggéré "Ringo" pour recommander la musique. Comme ça, la recherche sur les services de personnalisation a été réalisée par de nombreux chercheurs dans les différentes zones à l'exception pour le contexte.

Conclusion :

Dans le titre précédent on a entamé quelques travaux qui définissent les SRSC avec leur type de recommandation, avec différents exemples gérant les profils avec du filtrage collaboratif ou bien avec des recommandations basées sur le contenu, pour proposer à l'utilisateur que des données intéressantes selon son contexte.

4. Discussion générale:

Après avoir présenté les trois types de travaux, nous remarquons que ces systèmes n'utilisent pas les mêmes algorithmes d'intégration de contexte, c'est à cause des concepts qui se diffèrent d'une application à une autre. Nous remarquons aussi que dans le Mobile, le domaine qui a reçu le plus d'attention est le domaine touristique, car on peut réaliser plusieurs services en un seul système.

Pour le premier type, on a vu pourquoi l'utilisation d'un contexte pertinent (lieu, temps) est nécessaire, mais pas suffisante, ce qui a invoqué le 2ème type qui prend en considération ce que l'utilisateur préfère et fait la corrélation entre le contexte et ses préférences.

Le 3ème type est né pour optimiser son antécédent, ajouter le choix d'un bon type de recommandation, « basée sur le contenu » ou bien « filtrage collaboratif », et chaque système présenté utilise soit l'un des deux, soit les deux en même temps.

Concernant les insuffisances de chacune de ces applications, il s'agit de plusieurs cas, le plus souvent un contexte ignoré, du non disponibilité de l'application dans une grande plateforme mobile (Exp: Android), où bien dans un autre cas, la recommandation se résume à une seule chose (Ex : lieu), comme l'application qu'on va présenter ci-dessus (ReRex). Le tableau suivant (Tableau 11) montre la différence entre les trois types, avec les applications présentés pour chaque type.

Description des symboles / abréviations utilisés :

X → signifie les deux contextes pertinents (temps, lieu).

TA → signifie « Type de l'appareil », « / » → Valeur NULL.

M → Météo, **C** → Compagnon.

T → Température, **UP** → Préférences de l'utilisateur.

MC → Modélisation contextuelle.

POF → Post – Filtrage contextuel.

PRF → Pré- Filtrage contextuel.

Types		Axes de contexte		Contexte pertinent	Autre attribut Contexte	Acquisition du contexte	Algorithme d'intégration
SRSC basé sur les Contextes pertinents 1.	UbiquiTO	X	TA	Implicite	MC		
	MoMa	X	/	Explicite	POF		
	Deapz	X	TA	Explicite	PRF		
SRSC basé sur la Corrélation du contexte / préférence Utilisateur 2.	ReRex	X	M / C / T	Implicite	PRF		
	BlinkFeed	Lieu	TA / UP	Implicite	PRF		
	Flipboard	Temps	TA / UP	Implicite	PRF		
2 + Choix de la recommandation	Filtrage collaboratif	PlayStore	X	TA	Explicite	MC	
	Basé sur le contenu	ReRex	X	M / C / T	Implicite	PRF	

-Tableau 11- Comparaison entre les types et les travaux présentés.

IV. Chapitre 4 : Conception et Réalisation :

1. Introduction :

Dans ce chapitre on va entamer la conception de notre projet, où on va décrire le domaine d'application utilisé, en citant les différentes techniques et approches utilisées dans la proposition des SRSC, et ainsi on arrive à l'architecture du système qui permettra de prendre en charge les utilisateurs dans un contexte donné. Cette architecture est basée sur un système multi agent où on va décrire chaque composant convenablement en citant les inputs outputs de chacun. Nous allons aussi parler de l'algorithme bayes naïve et de l'otologie de contexte en expliquant la contribution de cet algorithme dans notre système et comment notre modèle de contexte est basé sur cette ontologie.

2. Domaine d'application :

Comme on a déjà mentionné dans le chapitre précédent, le domaine qui a reçu plus d'attention est le domaine touristique, Pour cela notre application est destinée à ce dernier. Pour répondre à la diversité des préférences des utilisateurs, nous proposons de mettre en place un système de recommandation des lieux qui est sensible au contexte des applications mobiles et se base sur les préférences de l'utilisateur.

Pour cela, il faut rappeler que ces dernières années, des dispositifs mobiles disposant de fonctionnalités avancées (Smartphones) sont apparues. Ils permettent à leurs utilisateurs d'accéder à des services à tout moment, quel que soit l'endroit où ils se trouvent. Ces services peuvent aussi s'adapter à la position de l'utilisateur grâce à la fonctionnalité de géolocalisation.

Le but principal est de construire un système sur dispositif mobile capable d'améliorer l'expérience de l'utilisateur lors d'un séjour / voyages en lui recommandant des lieux à visiter qui correspondent à ses goûts et en l'aidant à construire un parcours de visite adapté au contexte et à ses préférences.

Notre proposition consiste à concilier deux approches : une approche basée sur la représentation sémantique du modèle du contexte, et une approche collaborative qui utilise l'avis d'utilisateurs similaires. Nous cherchons également à prendre en compte le contexte (localisation, temps, environnement physique. . .) pour la génération de parcours de visite dynamiques et personnalisés.

3. Architecture Dédiée à Notre Système :

Comme on a mentionné précédemment, nous avons choisi de nous orienter vers la conception d'un SRSC destiné au domaine touristique. La corrélation des préférences et du contexte est réalisée dynamiquement en fonction des recommandations issues de ce système. Cette architecture est basée sur l'architecture multi - agents (SMA), choisi et définit comme suit:

Parmi plusieurs architecture dédié au système ubiquitaire, Nous avons choisis de travaillé sur le concept de l'architecture SMA, et pour rappeler sur cette dernière :

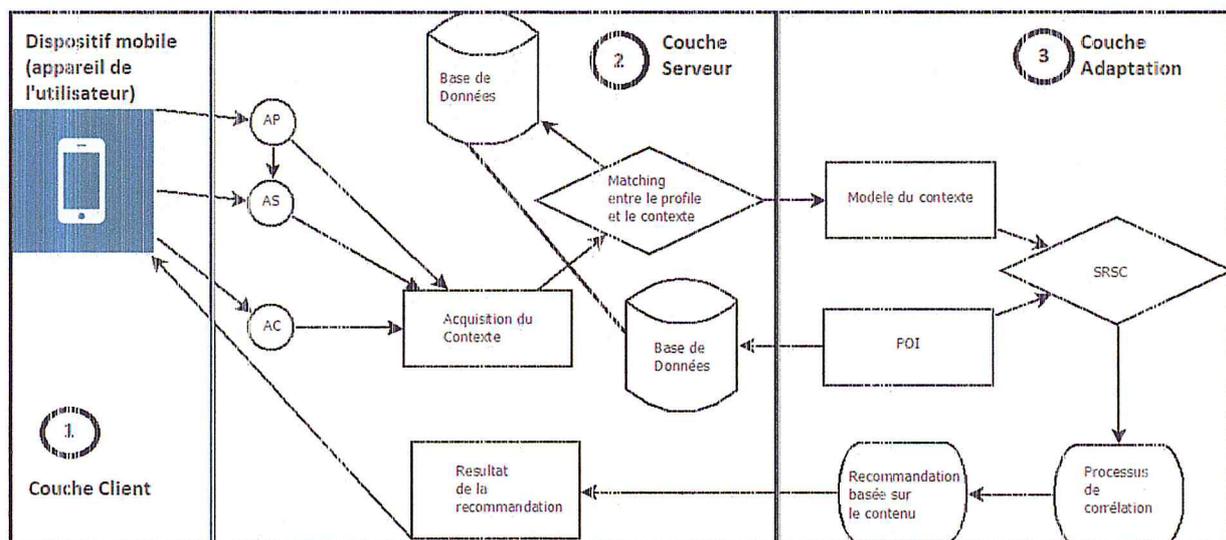
« [34] Un système multi agents est un système distribué composé d'un ensemble d'agents. Contrairement aux systèmes d'IA, qui simulent dans une certaine mesure les capacités du raisonnement humain, les SMA sont conçus et implantés idéalement comme un ensemble d'agents interagissant, le plus souvent, selon des modes de coopération, de concurrence ou de coexistence. »

- Pourquoi on a choisi l'architecture SMA ?

Pour notre système cette dernière est plus adéquate, car les agents se basent sur un système d'aide à la décision et vont interagir entre eux, contrairement à P2P par exemple qui dispose d'un système ou une entité a besoin de faire des actions en tant que serveurs, et en tant que client comme envoyer et recevoir des paquets de traitement.

3.1. L'architecture proposée :

La figure suivante (figure 17) représente l'architecture proposée à notre système:



-Figure 17- Architecture générale du système.

L'architecture qu'on vient de proposer se base sur l'ontologie de contexte dans la figure 18, qui sera présenté au-dessous.

Notre architecture est composée de trois couches :

- **Couche client** : Contient l'appareil mobile de l'utilisateur
- **Couche serveur** : Contient la base de données sur Parse, le modèle de contexte qui est composé d'un couple (profil, contexte) est créé dans cette couche
- **Couche adaptation** : c'est là où SRSC reçoit le modèle de contexte plus les POI pour lancé le processus de corrélation afin de calculé la liste de recommandation.

3.2.Description des termes :

POI : point d'intérêts

SRSC : système de recommandation sensible au contexte.

AP : Agent principal

AS : Agent secondaire

AC : Agent capteur (chercheur)

3.3.Identification des agents :

AP : Agent principal, responsable de :

- La création des comptes et de l'authentification.
- Créé un agent secondaire pour compléter le profile utilisateur.

AS : Agent secondaire, responsable de :

- Fournit les informations essentielles lors de l'acquisition de contexte.

AC : Agent capteur (chercheur), responsable de :

- La capture du contexte pertinent (temps, lieu) depuis le dispositif de l'utilisateur

3.4.Identification des interactions :

AP → AS : l'agent principal est responsable de la création des compte et l'authentification, et l'agent secondaire complète le profile utilisateur, donc l'agent secondaire ne peut pas commencer sa tache si l'agent principale ne termine pas la sienne, et il a besoin aussi des informations de l'utilisateur pour compléter cette procédure.

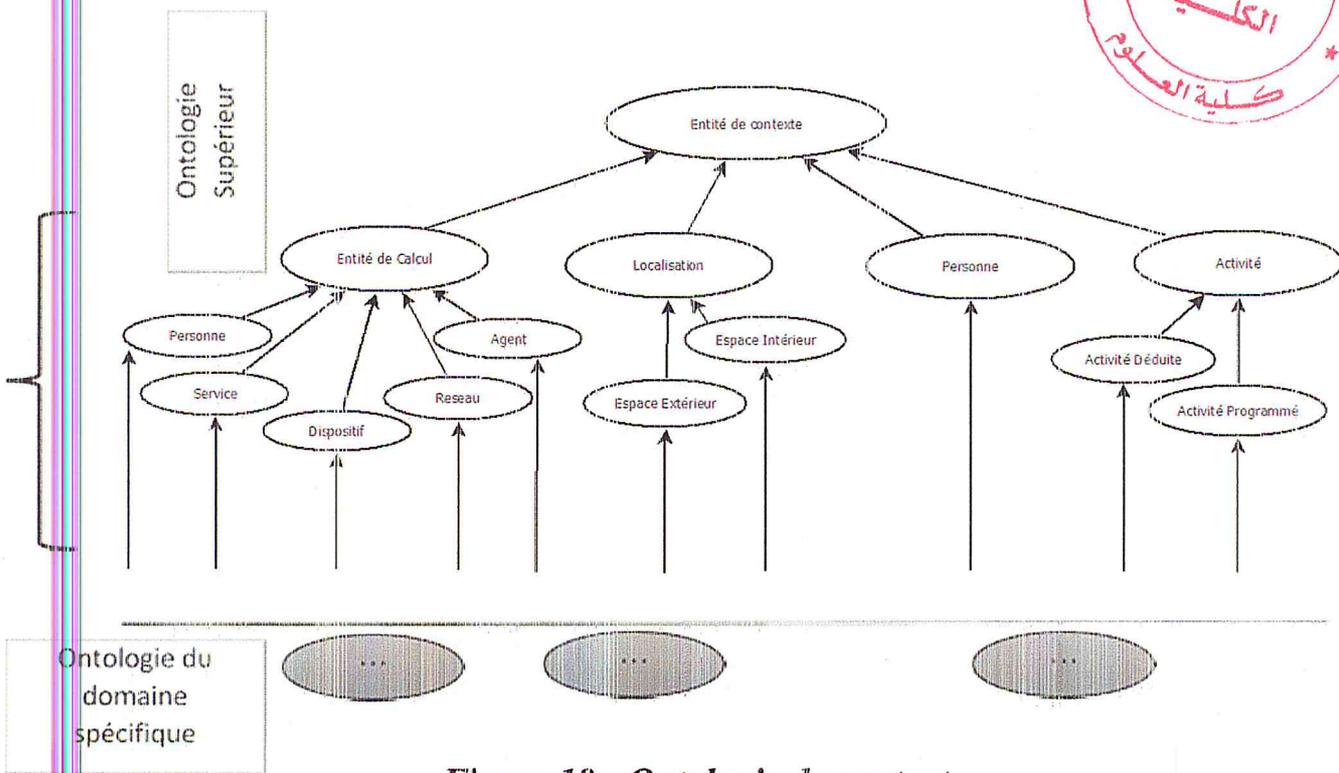
Dispositif mobile → AC : l'agent capteur est chargé de capter le contexte actuel de l'utilisateur, cela implique recevoir les données concernant la localisation et le temps depuis l'appareil mobile de l'utilisateur.

4. L'ontologie du contexte : [35] :

Le concept de base de notre modèle est basé sur une ontologie qui fournit un vocabulaire pour représenter des connaissances sur un domaine et pour décrire des situations spécifique dans un domaine. L'ontologie du contexte définit un vocabulaire commun pour partager des informations de contexte dans un omniprésent, et inclue des définitions, des notions de base dans le domaine et des relations interprétable par la machine. Le principal avantage de notre modèle de contexte est de partager une compréhension commune de la structure du cadre d'informations entre les utilisateurs, les matériels et les services pour permettre l'interopérabilité sémantique. Il permet également la réutilisation du domaine de connaissances, à savoir, la construction d'une grande ontologie en intégrant plusieurs ontologies décrivant des parties du grand domaine.

Surtout, il permet une analyse formelle du domaine de la connaissance, par exemple, le contexte devient un raisonnement possible en définissant explicitement son ontologie.

Le modèle suivant dans la figure 18, représente l'ontologie du contexte.



-Figure 18- Ontologie de contexte.



L'ontologie de haut niveau (Supérieur) définit les concepts de base (personne, Localisation, entité de calcul, activité) comme indiqué dans Figure précédente. L'entité de contexte fournit un point d'entrée référencé pour déclarer l'ontologie supérieure. Une instance de cette entité existe pour chaque utilisateur distinct, un agent ou un service.

Chaque instance de l'entité de contexte présente un ensemble de classes descendantes de personne, Localisation, entité de calcul et de l'activité. Les détails de ces concepts de base sont définis dans l'ontologie du domaine spécifique qui peut varier d'un domaine à un autre. Nous avons défini toutes les classes sont dérivées dans un environnement de maison intelligente et un ensemble de propriétés et des relations qui sont associés à ces classes.

4.1.Composant de l'architecture :

Dans cette section nous allons présenter les différents composants de notre architecture en donnant des exemples de chaque composant.

4.1.1. Acquisition du contexte [36] :

L'acquisition du contexte qu'on vient de proposer se base sur les travaux de (Zimmermann et al. 2007).En l'appliquant à nos besoins spécifiques dans le cadre de la création d'un SRSC pour le domaine touristique.

On la partage à deux catégories : contexte pertinent, et contexte secondaire.

- **Contexte principal** : Ce type de contexte est acquis implicitement, directement du dispositif mobile, il est composé de deux attributs principaux :
 - **Localisation** : la localisation actuelle de l'utilisateur est une information et des lieux qu'il l'entour. Ces informations sont importantes et interviennent lors de la sélection des objets à recommander suivant le profil de l'utilisateur.
 - **Temporalité** : information concernant le temps qui est aussi très importante, elle est nécessaire pour optimiser le choix de la recommandation et pour avoir une idée sur ce que l'utilisateur préfère, et à n'importe quel moment.

L'agent capteur (AC) est responsable de l'acquisition du contexte comme le montre notre architecture, ces informations contextuelles seront capturées et reçues depuis le dispositif (l'appareil de l'utilisateur) d'une façon explicite, localisé en utilisant l'option de la

géolocalisation pour avoir la position, et le temps (l'heure) actuelle qui s'affiche sur l'écran de l'utilisateur.

- **Contexte Secondaire :** Ces contexte est acquis d'une manière explicite du fait que l'utilisateur lui-même qui doit entrer les valeurs de ces attributs, on parle ici d'autres contextes qui peuvent être présent en considération aussi, ces contextes-là jouent un rôle assez important dans la recommandation, mais aussi de poids pour optimiser la liste recommandé. Ces contexte sont activés/désactivés par l'utilisateur lui-même.

L'agent secondaire (AS) est responsable de ce contexte, après l'authentification par l'agent principal, ce dernier apporte quelque contexte secondaire choisit par l'utilisateur lui-même, parmi ces contexte nous citons les suivants :

- **Activité :** selon l'activité de l'utilisateur, voir même les activités qu'il occupe au cours de son voyage, le système doit prendre en considération cette information contextuelle.
- **Budget :** le budget de l'utilisateur peut se varier, donc c'est une information dynamique que l'utilisateur peut la changé de temps en temps pour recalculer la liste de recommandation selon ses besoins (hôtels, musée ...)

Exemple :

- Pour le contexte principal, Lieu= Alger et Temps=10h du matin
- Pour le contexte secondaire, Activité=Photographe et budget = 54 000 DA

4.1.2. Le Matching :

Le matching concerne le profil de l'utilisateur d'un côté, et une ressource de donnée de l'autre coté (le contexte dans notre cas) qui est une approche basée sur le contenu.

Lorsque le matching considère plusieurs profils d'utilisateurs, il est appelé une approche collaborative.

Pour faire le matching du profil et le contexte acquis, ces informations sont définies dans la même ontologie que dans la figure 18 au-dessous. Dans l'ontologie présentée, le profil se compose d'un ensemble de concepts qui représentent son contenu. Les concepts représentant un élément sont les plus spécifiques dans une certaine branche de la hiérarchie.

Un profil basé sur le contenu de l'utilisateur se compose d'une liste pondérée des concepts d'ontologie représentant ses intérêts, par exemple, si le profil de l'utilisateur peut comprendre les concepts de sports, le matching se fait en cherchant des lieux contenant « sport » dans

leurs concepts et calcule la similarité entre les deux, puis sortir d'un modèle de concept contenant un couple (profil, contexte).

Pour cela, nous avons utilisé la une méthode de classification, elle est basé sur la méthode Bayes Naïve et elle est présenté et utiliser comme suit:

4.1.3. Approche de Classification Bayésienne:

Nous avons adopté la classification Bayésienne pour classifier nos données, la classification Bayésienne qui est un type de classification probabiliste simple basée sur le théorème de Bayes avec une forte indépendance (dite naïve) des hypothèses. Elle met en œuvre un classifieur bayésien naïf, ou classifieur naïf de Bayes, appartenant à la famille des classifieurs linéaires.

Par exemple : Un fruit peut être considéré comme une pomme s'il est rouge, arrondi, et fait une dizaine de centimètres. Même si ces caractéristiques sont liées dans la réalité, un classifieur bayésien naïf déterminera que le fruit est une pomme en considérant indépendamment ces caractéristiques de couleur, de forme et de taille.

Dans notre cas, nous proposons de regrouper le contexte et les préférences de l'utilisateur sous formes de couple contenant 2 données, contexte pertinent et profil utilisateur:

$C \times PU = (C, PU)$ ou C représente le contexte et PU les préférences de l'utilisateur.

Nous Avons aussi créé des profils standards pour faire le matching avec le profil utilisateur du couple, l'utilisateur peut choisir un des profils standards proposé, sinon il peut créer son propre profil personnalisé.

En prenant l'exemple du profil d'un artiste et en ajoutant le contexte acquis par exemple lieu = Alger, temps = 10h, la classification et le matching nous donnera un résultat comme: "Un artiste est à Alger à 10h"

On peut même proposé de diviser le contexte de temps en plusieurs classes, par exemple le diviser en deux classes, semaine ou week-end, même chose pour le lieu on peut le deviser en plusieurs classes comme "à l'extérieur", "dans l'hôtel", "à la plage" etc.

4.1.4. SRSC et Processus de corrélation :

Pour lancer le processus de corrélation, le SRSC doit recevoir des données essentiels, il s'agit du modèle de contexte déjà présenté et les POI, mais en fait c'est quoi les POI ??

- **POI** : les points d'intérêt de l'utilisateur, ce qu'il aime voir et ce qu'il a aimé sera aussi pris en considération dans l'architecture qu'on va proposer. , exemple des POI

une personne intéressée par l'art, aura après le processus de corrélation dans la liste recommandé des lieux qui font partie de ce concept comme des musées des théâtres etc. ... et même chose pour les autres POI, donc chaque lieu est classé sous un concept donné pour faciliter la tâche de le recommander.

La matrice de corrélation :

Vu qu'on utilise un pré filtrage contextuel dans notre application, pour faire la corrélation entre le modelé du contexte et les POI de l'utilisateur, on est obligé à adopter une matrice de corrélation.

Cette matrice travaille avec des données de la forme : $UP \times I \times C \times POI$ où C représente le contexte qui produit une liste de recommandation 1,i 2,i 3,i etc. ... UP le profil utilisateur et POI les points d'intérêt de l'utilisateur.

Exemple : un petit exemple en utilisant l'arbre de décision.

Un architecte est à Oran, il préfère Arts, Photographie et Sport, c'est un pour voir comment la décision est prise en cas de l'utilisation d'un arbre de décision après l'exécution avec une matrice de corrélation.

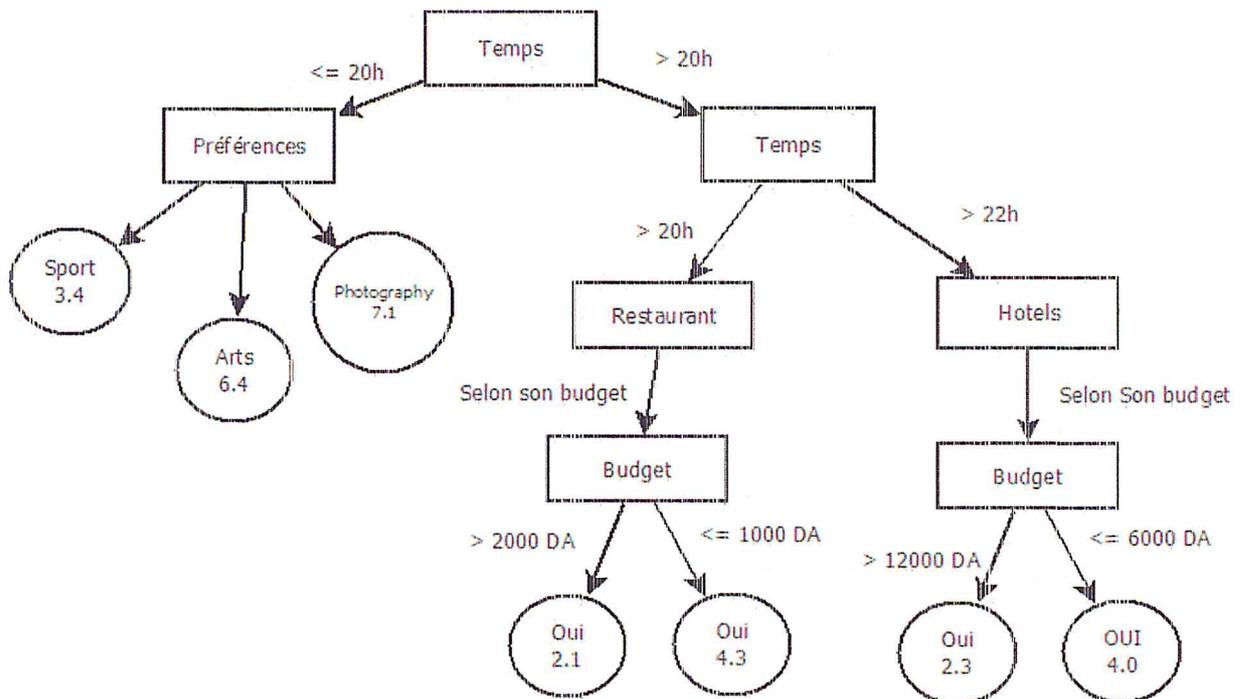


Figure 19 – Arbre de décision

Le processus de corrélation fait la fusion de données reçus pour calculer la liste de recommandation et cette dernière sera recommandée selon la méthode suivante :

Les techniques de recommandations :

Selon Xi Ning ZHU (systèmes de recommandations et sécurités, 2006), il existe huit techniques de recommandations, mais dans notre cas on va se baser sur la technique suivante :

- **Les recommandations basées sur le contenu :**

Les recommandations basées sur le contenu recueillent d'abord, au moyen du système, les données de l'utilisateur, par la suite, elles analysent et créent son profil de préférence pour enfin suggérer les articles qui lui conviennent.

Par exemple, un article déjà aimé par l'utilisateur, cet article et d'autre de son genre seront proposé à l'utilisateur afin d'améliorer la corrélation et la liste recommandé.

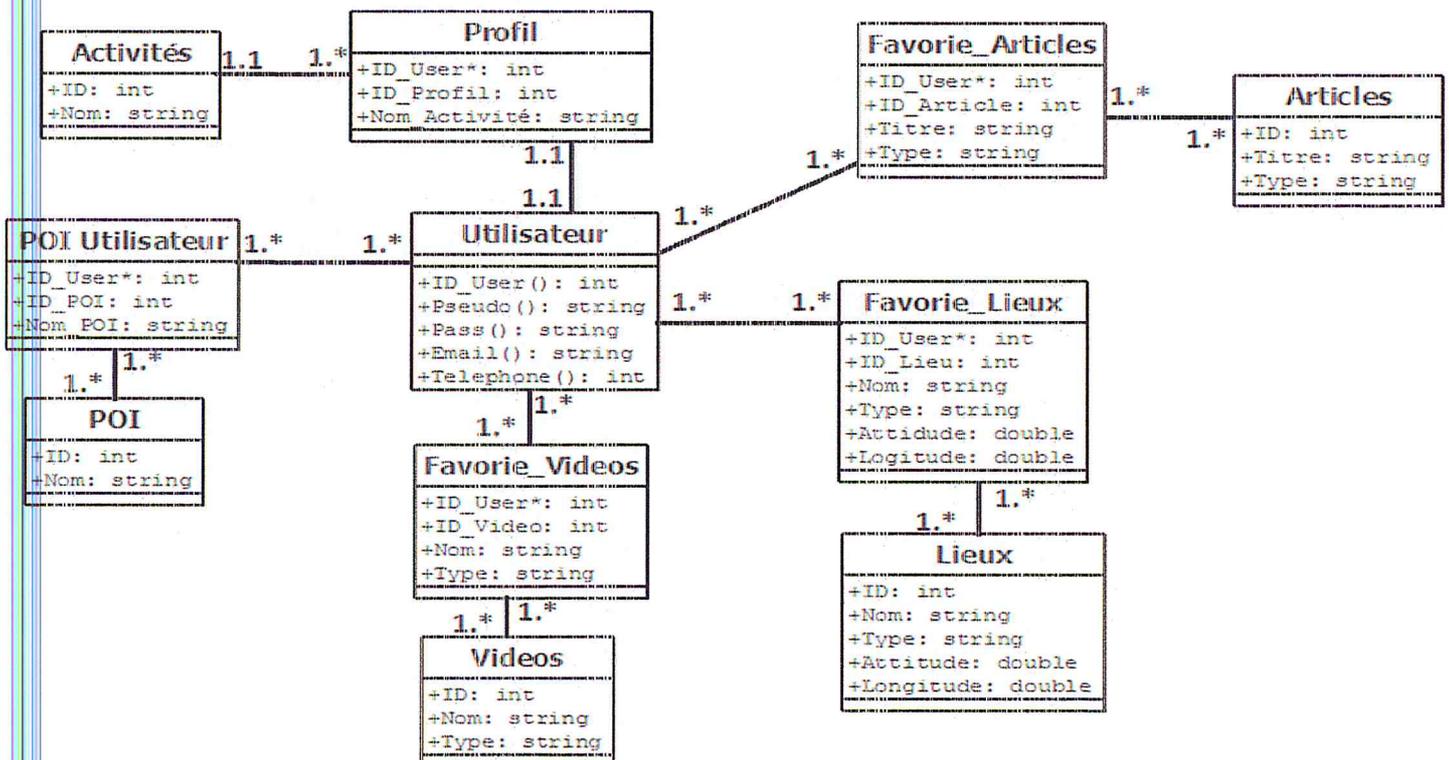
4.1.5. La Base de données:

Comme notre base de données se trouve au niveau de Parse.com qui sera présenté dans le prochain chapitre d'implémentation, Parse fournit la notion de classe, ce qui nous a permis de créer une base de données qui se compose de plusieurs classes.

Pour la représentation des classes, on a utilisé la méthode UML pour la modélisation en donnant un diagramme de classes contenant les différentes classes ainsi que leurs attributs.

4.1.6. Diagramme de classe :

Le Schéma suivant représente le diagramme de classe de notre système.



-Figure 20- Diagramme de classe.

4.2. Input / Output des composants de l'architecture :

Après avoir bien présenté les différents composants de notre architecture, ainsi que le rôle de chacun et ces interactions avec chaque composant adjacent, nous allons maintenant dans cette section présenter les Input (données entrants) et les Output (données sortants) de toutes les parties en donnant un exemple d'un simple scénario.

Le Tableau suivant représente les Input / Output des composants de notre architecture :

Composant	Inputs	Outputs
Dispositif Mobile	Liste de recommandation basée sur le contenu	<ul style="list-style-type: none"> • Attribut de contexte pertinent • Identifiant de l'utilisateur • Profil utilisateur
Acquisition de contexte	Attribut de contexte pertinent Identifiant de l'utilisateur Profil utilisateur	Couple de la forme (Contexte, Profil utilisateur)
Matching entre le profil et le contexte	Identifiants, Couple (contexte, profil utilisateur)	Modèle du contexte
SRSC // Processus de corrélation	Modèle (ID, Contexte, Profil utilisateur) POI	Des recommandations
Recommandation Basée sur le contenu	Des recommandations	Liste de recommandation basée sur le contenu

Tableau 12 – Analyse de l'architecture

Scénario :

Pour le déroulement de cette procédure, nous allons faire un petit scénario pour bien comprendre l'architecture, les données et le résultat ne sont pas vraiment exacts, mais il y aura un autre déroulement illustré dans le prochain chapitre d'implémentation.

Identifiant :

- ID: 001
- User: Amine
- Password: azerty123

Profile:

- Activité: Acteur
- Budget: 60000 DA

Attribut de contexte :

- Lieu : Alger
- Temps : 10h

POI :

Articles	Lieux
<ul style="list-style-type: none">• Politique• Sport• Santé• Style de vie	<ul style="list-style-type: none">• Arts• Hôtels• Restaurant• Plages

Tableau 13– POI de l'utilisateur

Donc avec ces données-là, l'agent principal utilise les ID pour l'authentification, dès qu'il termine sa tâche les agents secondaire et capteur apportent le profil utilisateur et capteur les attributs de contexte actuel pour former un couple de la forme : (« Amine, azerty123, Acteur, 6000DA », « Alger, 10h »)

Le matching nous trouvera un profil standard pareil en utilisant la méthode bayésienne, du genre « Artiste, Alger, 10h », et nous mèneras vers la création d'un modèle de contexte, En ajoutant les POI avec le modèle du contexte obtenue, le système de recommandation acquies ces informations et lance le processus de corrélation, on aura donc des résultats concernant les informations initiales, par exemple dans la section article on aura des articles de plusieurs

domaine, même chose pour les lieux on prend en considération tout ce qui entoure l'utilisateur à sa position, en prenant ces POI et en utilisant la méthode basé sur le contenu, les recommandation sont personnalisées suivant ce que l'utilisateur aime voire, encore une fois pour la section article des article selon les thèmes choisit par l'utilisateur, même chose pour la recommandation de la localisation on aura des Hôtels et des restaurants etc... tout en tenant compte de son profil et surtout son budget, et bien sur les POI aimé.

L'exemple ci-dessus est déroulé dans le tableau suivant en spécifiant les inputs/outputs et le résultat final :

Composant	Inputs	Outputs
Dispositif Mobile	Résultat = (Différents Lieux aimé par l'utilisateur comme des restaurants des hôtels etc., même chose pour les articles selon les thèmes choisit par l'utilisateur	Identifiant, Profil (Acteur, 60000da) Contexte (Alger, 10h)
Acquisition de contexte	Identifiant, Profil (Acteur, 60000da) Contexte (Alger, 10h)	Couple (Profil, Contexte)
Matching entre le profil et le contexte	Couple (Profil, Contexte)	Modèle de contexte (« Acteur, 60000da », « Alger, 10h »)
SRSC // Processus de corrélation	Modèle de contexte (« Acteur, 60000da », « Alger, 10h ») + POI = (« Politique, Sport, Santé, Style de vie » « Arts, Hôtels, Restaurants, Plages »)	Résultat = (Différents Lieux aimé par l'utilisateur comme des restaurants des hôtels etc., même chose pour les articles selon les thèmes choisit par l'utilisateur)

Tableau 14 – Déroulement de l'exemple précédent

5. Conclusion :

On a entamé dans ce chapitre la conception de notre projet, les outils qui Co-fonctionnent sous l'interface utilisateur et ce que ce dernier aura comme interactions d'après ses points d'intérêts et notre SRSC (Système de Recommandation Sensible au Contexte), où on a présenté l'architecture principale qui à mener à faire notre application. On a aussi bien définie l'ontologie avec tous les contextes et les classes et les Sous-Classes présentent en clôturant avec une définition très claire de notre service de base de données choisie (Parse.com).

Tout ce chapitre nous mènera vers le bien de l'implémentation dans le chapitre suivant, où on va introduire nous outils de programmation, et l'interface finale de notre application, visant à bien éclairer l'interface et l'expérience utilisateur.

Chapitre 5 : Implémentation :

1. Introduction :

Nous arrivons enfin au dernier chapitre de notre projet, qui consistera à mettre en œuvre la partie plus importante qui est le cœur du développement de notre application mobile, où on va présenter étape par étape les outils et les environnement de travail et de développement, tout en détaillant les technologies utilisées menant à la bonne implémentation de notre application sur la plateforme qui nous convient.

2. Environnement de travail :

On a introduit la manière basique avec l'environnement Windows Seven et des PC de type HP Pavilion avec des processeurs Intel i7 et des mémoires vives de 4 à 6Go de RAM, une configuration assez bonne pour le bon déroulement des logiciels et technologies utilisées.

3. Environnement Logiciel :

Dans cette partie on va parler des outils qui ont aidé à la création et à l'implémentation de notre application de A à Z.

3.1. Environnement de développement (Plateforme et Logiciels) [37]:

La plateforme sur laquelle se tient notre application s'appelle Android, un système d'exploitation mobile pour Smartphones, tablettes tactiles, smartwatches (version Wear) et terminaux mobiles. C'est un système open source utilisant le noyau Linux. Il a été lancé par une startup du même nom rachetée par Google en 2005. D'autres types d'appareils possédant ce système d'exploitation existent, par exemple des téléviseurs, des radioréveils, des montres connectées, des autoradios et même des voitures.

3.1.1. Pourquoi nous avons choisie Android [38]?

La question qui fait couler beaucoup d'encre... Pourquoi choisir Android (Google) plutôt qu'iOS (Apple) ou bien Windows Phone (Windows). Alors on va résumer notre choix en quelques points qui sont :

- **Le Système :** Contrairement à iOS d'Apple et Windows Phone Android est Open Source.
- **La gestion des données :** Avec un téléphone Android la simplicité est de mise. On copie les fichiers (musique, photos, vidéos) où l'on veut sur la carte SD (ou la mémoire interne) et tout est détecté par le système .

- **Les Services Google :** Au premier démarrage du mobile on vous propose de créer un compte. L'identifiant du compte sera une adresse @gmail.com permettant entre autre l'accès à tous les services Google gratuitement, l'Android Market (Play Store), Maps, g+, Gmail, Contacts et répertoire, etc.
- **Les Widgets :** Un petit programme qui va se placer sur le bureau pour accéder directement sans lancer l'application à un contenu dont on se sert très souvent.
- **Le Market :** On trouve sur le Play store de Google plus d'application gratuite et pour les comptes développeur c'est 25\$ à vie contrairement à iOS 99\$ par an (faut acheter un MacBook pour développer sous iOS car leur logiciel Swift et seulement sur MacOS) et sur Windows deux types de comptes à 19\$(individuel) et à 99\$(entreprise).

3.1.2. Les Différentes versions d'Android et la version choisie :

L'historique des versions d'Android a débuté avec la sortie de la version 1.0 en septembre 2008, la Figure suivante montre toutes les versions après la 1.0 (Beta) en résumé :

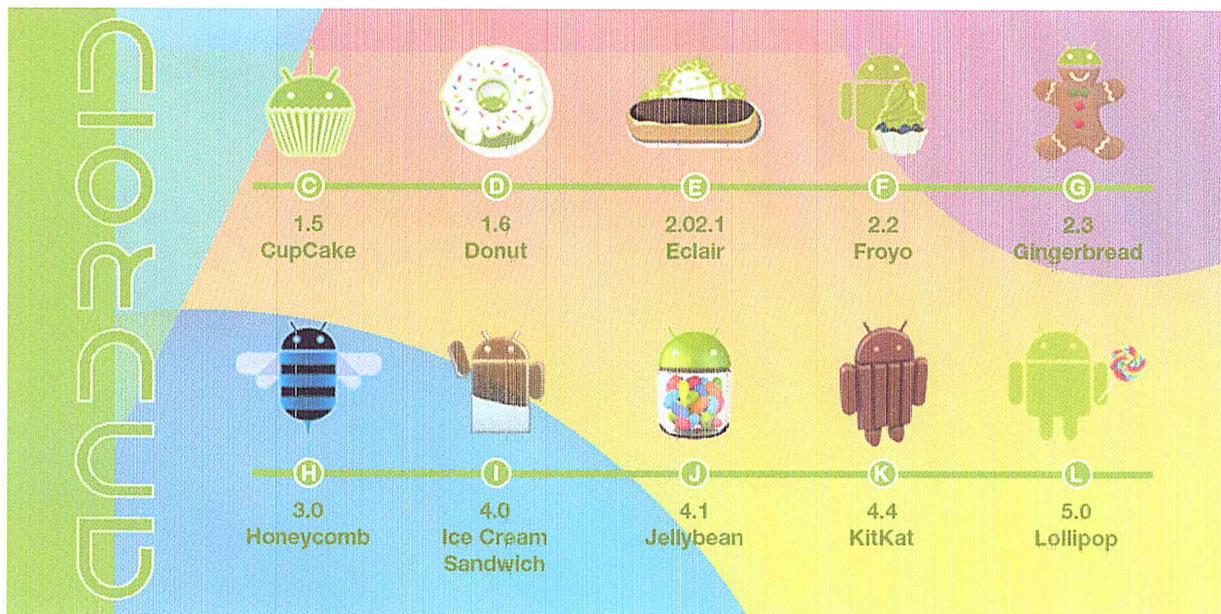


Figure 21 – Différentes Version Android Depuis sa Création.

Sans oublier que récemment y a eu la Version 6.0 (Android M) et qui est toujours sous version d'essai. Bien sûr notre application est compatible avec les téléphones contenant Android 4.0 (Ice Cream Sandwich) ou plus, touchant plus de 93,57% des Smartphones Android disponibles chez les utilisateurs.

3.1.3. Les outils et les données auxiliaires :

le développement du projet est aussi Open Source que jamais, avec des outils offrant la meilleure performance et la bonne utilisation, et qui été facile à entreprendre, en utilisant les logiciels suivant :

- **L'environnement Android Studio :**

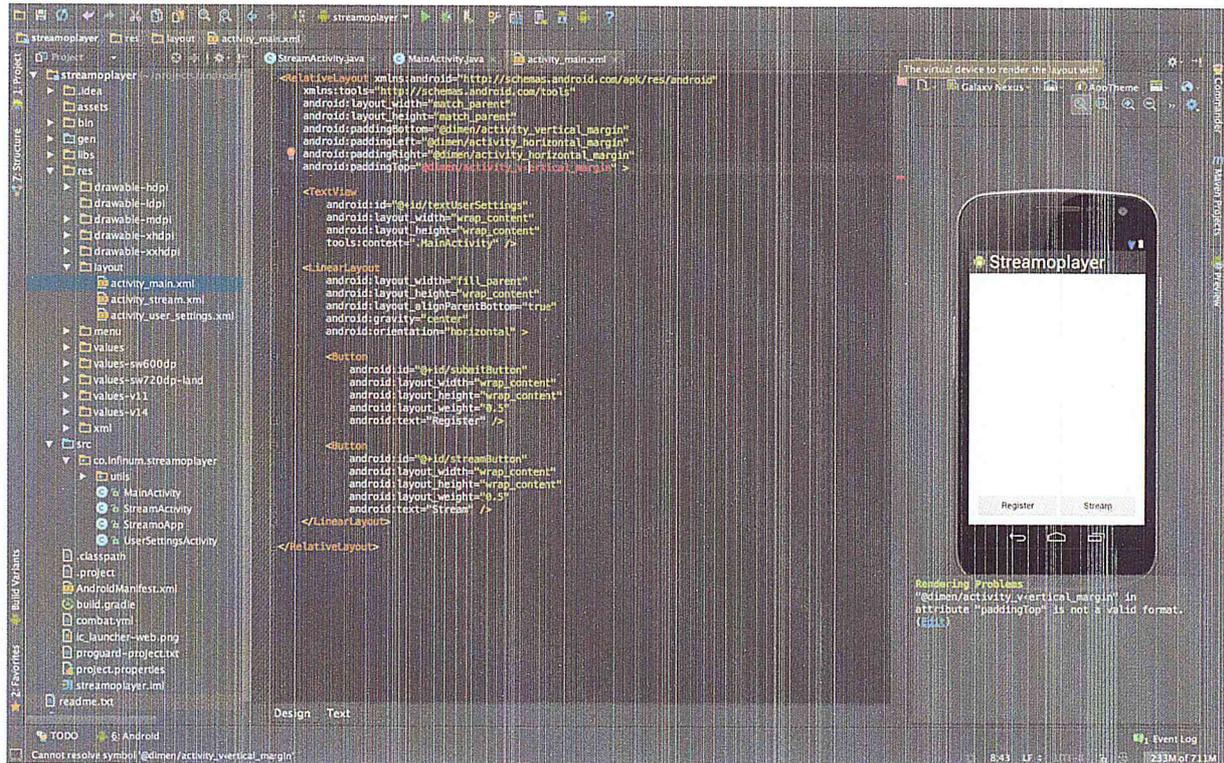


Figure 22- Environnement Android Studio.

Android Studio est un environnement de développement pour développer des applications Android. Il est basé sur IntelliJ IDEA. Avant Android Studio, de 2009 à 2014, Google proposait comme environnement de développement officiel une distribution spécifique de l'environnement Eclipse, contenant notamment le SDK d'Android. Le 8 décembre 2014, Android Studio passe de version bêta à version stable 1.0. L'environnement devient alors conseillé par Google, et Eclipse est délaissé.

On a choisie Android Studio car il permet principalement d'éditer les fichiers Java et les fichiers de configuration d'une application Android en XML, et il est proposé et agréé par Google.

Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page des écrans sur des écrans de résolutions variées simultanément.

- **JDK :**

Le Java Development Kit (JDK) désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé. Sans JDK la programmation java est impossible, et rien de ce qui est développé en JAVA ne marchera, la Figure 23 ci-dessous montre la version que l'on a utilisé.



Figure 23- Version Java JDK utilisé :

- **SDK :**

SDK est une abréviation qui peut faire référence à Software development kit, et c'est l'ensemble d'outils que l'utilise notre Android Studio permettant l'installation et la création des application Android, mais le SDK est aussi utilisé sur toutes les plateforme utilisant java, la Figure 23 montre bien notre version SDK de l'API 14 vers l'API 21 :

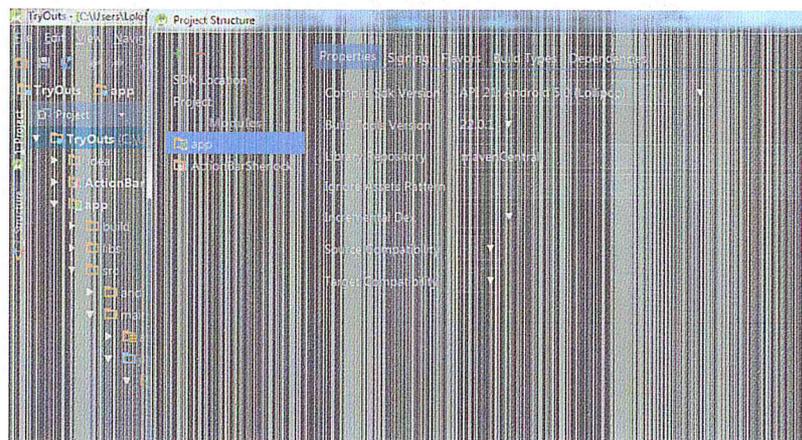


Figure 24- Version de l'API Android (SDK).

- **L'émulateur GenyMotion :**

Genymotion est un émulateur Android rapide et facile à utiliser pour exécuter et tester les applications Android. Facile à installer et possède une machine virtuelle qui affichera l'interface Android d'après notre configuration donnée. L'émulateur est relié à Android Studio et s'exécute à chaque fois qu'on compile le projet :

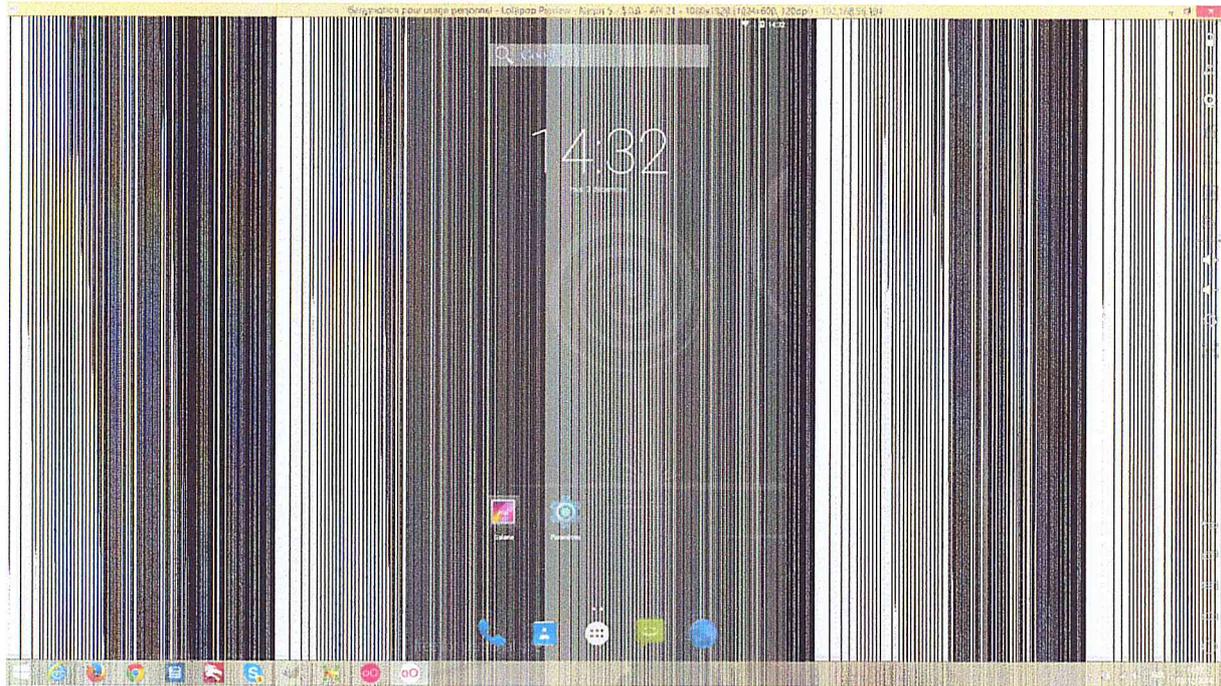


Figure 2-1: Interface GenyMotion (Machine Virtuelle Lancée).

3.2. Technologie utilisées (Base de données) :

3.2.1. Les Bases de données pour mobile [39] :

La sauvegarde d'informations dans une base de données est immédiate dans une application mobile, et généralement en utilise trois genre de base de données :

SQLite : La base SQLite est interne à l'application donc spécifique à chaque appareil. Elle peut à la fois contenir des données provenant de sources externes et des données créées par l'utilisateur lui-même. Sa rapidité en fait un choix idéal lorsqu'il y a beaucoup d'informations à traiter.

Base en Cache : La base en cache de données intervient lorsque celles-ci sont récupérées via des sources externes, mais sur lesquelles l'utilisateur n'intervient pas, comme un flux de données ou des images par exemple. Elle permet à la fois, un chargement plus rapide de l'information, mais aussi un accès en ligne. Le cache est géré par l'appareil lui-même pour ne pas perturber l'utilisateur avec des données trop important (Mo).

Préférences utilisateur (Notre choix) : La gestion de tout ce qui est propre à un utilisateur sur un appareil spécifique, est réalisée par le biais de fichiers et la sauvegarde qui consiste à sauvegarder toutes les données dans une base de données dans le Cloud (en ligne) sous le pseudo de l'utilisateur.

3.2.2. Définition des BaaS (Backend as a Service) [40] :

Backend mobile en tant que service (M-BaaS), aussi connu comme "backend en tant que service" (BAAS), est un modèle pour fournir aux développeurs web et d'applications mobiles un moyen de lier leurs applications au backend stockage cloud et API exposés par les applications back-end tout en offrant des fonctionnalités telles que la gestion des utilisateurs, les notifications push, et l'intégration avec les services de réseautage social.

Ces services sont fournis via l'utilisation de la coutume des kits de développement logiciel (SDK) et des interfaces de programmation d'application (API). BaaS est un développement relativement récent dans le cloud computing, avec la plupart BaaS startups datant de 2011 ou plus tard. Bien que d'une industrie assez naissante, les tendances indiquent que ces services gagnent du terrain main Stream avec les consommateurs de l'entreprise.

3.2.3. Base de donnée choisie : (Parse.com) :

Pour une base de données se trouve au niveau du Cloud, on a choisi d'utiliser Parse comme BaaS (Backend as a Service), car c'est le seul qui permet de commencer contrairement aux autres qui exige de payer avant de commencer.

3.2.4- Mais c'est quoi Parse.com ?

Parse.com fournit un service de Backend pour les développeurs. Si l'application mobile ou de bureau nécessite un Backend sur Internet, Parse est l'une des meilleures options à choisir.

Parse est l'une des meilleures solutions car il offre trois plans différents pour le moment. Le premier est gratuit et permet à votre application pour faire 1 million de demandes par mois avec une limite d'échelonnement de 10 requêtes par seconde. Ils fournissent également 1 million de pushes par mois.

Si votre application est populaire et vous avez besoin plus que de sauter jusqu'à 15 millions de demandes par mois vous coûtera \$ 199 qui, pour une application de cette taille, il est relativement pas cher.

3.2.5- La notion de classe :

Parse vous permet d'accéder à vos données à travers les objets. Lorsque vous créez une nouvelle application dans Parse, vous pouvez créer une nouvelle classe, puis importer des données pour cette classe. Les importations peuvent être à partir de fichiers CSV, XLS, TXT et JSON, ou vous pouvez créer manuellement la table dans chaque classe en définissant les colonnes et les types de données stockées.

L'ajout de données et la création des colonnes ne prennent que quelques lignes de code. Dans le menu de démarrage rapide, vous pouvez voir des moyens spécifiques pour tester votre application sélectionnée.

La figure suivante présente l'interface Parse, ainsi que les différentes classes qu'on utilise :

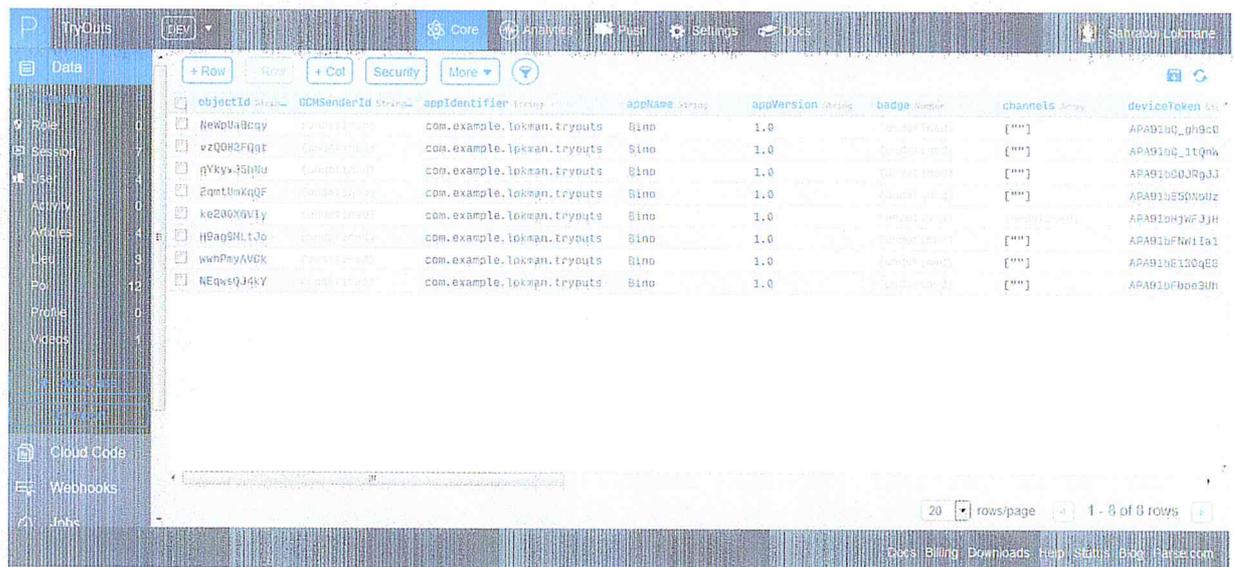


Figure 26- Interface Parse.

4. Test d'intégration :

4.1.1. Présentation de l'application (Binno) :

En une phrase, Binno est une application touristique qui vise à aider les utilisateurs dans la recherche d'endroits qui peuvent combler leurs sorties et leur intérêts.

Elle contient divers services qui ont aidé à la naissance de ce projet, en commençant par la partie des informations (What's New ?) qui est un service de feed d'articles et de vidéos, qui affiche ces derniers en fonction des POI de l'utilisateur et d'après son profil. Il y a aussi le service des endroits proches de l'utilisateur (Map Me !) qui montre à l'individu sa position et les endroits et les événements proches, puis vient le service le plus important de notre projet qui est le guide touristique (Let's Go !) qui d'après le Budget et la durée du séjour déterminera les endroits convenables pour la sortie de l'utilisateur, et tout ça d'une manière générale cherchant à mettre les contextes de l'utilisateur en évidence.

La Figure 27 ci-dessous est juste un petit aperçu de l'application :

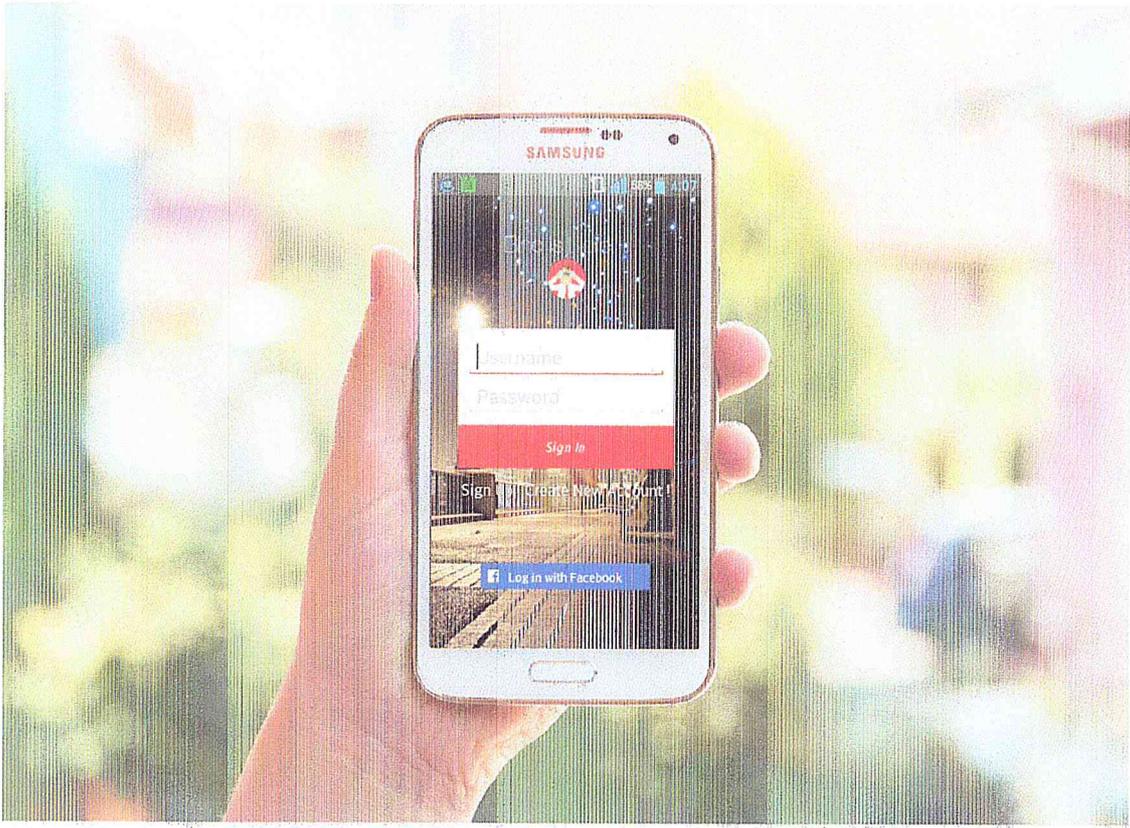


Figure 27- Aperçue de notre application sous notre Téléphone.

4.1.2. Représentation des interfaces de l'application :

» Interface Splash Screen :

La plupart des applications mobiles ont des Splash Screen, qui est un écran introductif en cinq secondes au maximum affichant le nom de l'application avec son Logo et la société qui l'a développé, comme le montre la Figure 28 ci-dessous, l'application « Bino » avec son Logo, et le nom de société qu'on a choisie « Pixel » :

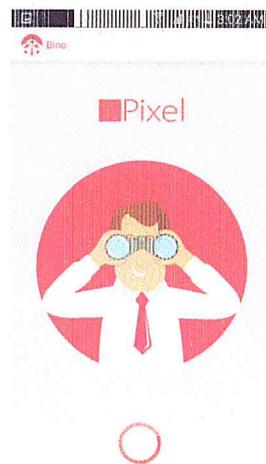


Figure 28- Interface Splash Screen.

• Interface Login :

La Figure 27 montre l'interface Login, où l'utilisateur « loko » se connecte à son propre compte pour naviguer et utiliser nos services, ou bien se crée un nouveau compte si c'est la première fois qu'il utilise « Eino ». Faire remarquer que chaque utilisateur à son propre compte augmente la sécurité et ajoute la diversité dans notre application pour nous aider à bien gérer nos utilisateurs et à les garder bien à jour en connaissant les POI de chacun d'eux à part, et pourquoi pas plus tard faire une version qui va lier les utilisateurs avec un partage critique des meilleurs endroits.

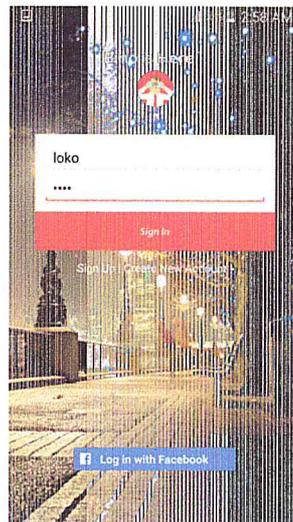


Figure 27- Interface Login (Connexion avec Comptes).

• Interface Sign-up (Inscription) :

À la première contact avec l'application, l'utilisateur doit se créer un compte propre à lui et à ses désirs, nous faisant part de sa profession avec un compte standard pour nous laisser gérer, ou bien de personnaliser son compte avec ses POI préférées pour qu'on lui recommande des résultats d'après ces choix et cela ne lui coûtant rien à part un mot de passe et une adresse email, qu'il doit utiliser pour se connecter plus tard, les Figures 28 et 29 montrent les étapes d'inscription de l'utilisateur « Loko » et le choix de ses points d'intérêts et son activité :



Figure 30- Interface Sign-Up 1^{ère} étape.

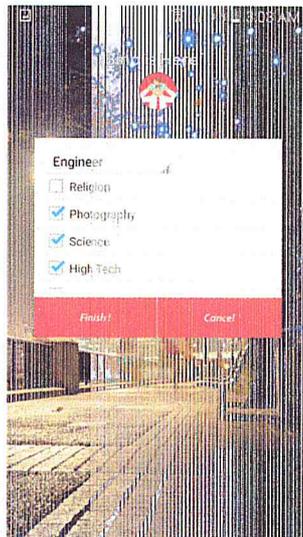


Figure 31- Interface Sign-Up 2^{ème} étape.

• Interface d'accueil et Menu d'accueil

Dès que l'utilisateur « loko » se connecte à son compte, il est accueilli par le service « What's New ?! » qu'on va expliquer juste après, et en haut à gauche comme dans la plupart des application Android, on a utilisé ce qu'on appelle un Drawer menu, un menu déplaçant à gauche de l'écran permettant à chacun de naviguer entre les différents services qu'on propose, d'une manière clair sans avoir à demander le guide d'utilisation, aussi on a enrichi l'application avec un beau design de Tab, des onglets glissent l'utilisateur entre les sous services de chaque service.



Figure 3.11- Menu Déroulant de l'application (DéroulerMenu).

• **Interface** du service d'informations « What's New ?! » :

Le service « What's New ?! » recommande à l'utilisateur des articles et des vidéos à partir de critères généraux, mais principalement, il lui offre ces informations que d'après ses choix soit par sa profession ou par ses préférences, les articles en tant que son service et les vidéos aussi de l'autre côté de l'onglet pour garder notre utilisateur à jour de tout ce qui se déroule dans ses domaines d'intérêt, et sans oublier qu'il a mis actif le service Push Notifications qui alertera l'utilisateur de chaque nouveauté dans l'application que ce soit des articles, vidéos, où parfois mise à jours, comme le confirme les figures 31 et 32 (le jeu d'essai sera prêt d'ici la soutenance):



Figure 3.12- Interface « What's New ?! » avec articles.

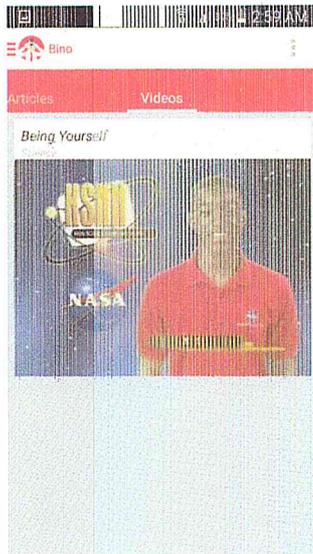


Figure 34- Interface « What's New ?! » sous Vidéos.

• **Interface service «Near By !»**

Le Service «Near By !» est très utile à notre application notamment touristique, car il contient deux sous service qui sont obligatoire dans le tourisme, le sous service « Events » qui présente une liste d'évènements dans les prochains dates de tout genre (Concert, soirée, événement, séminaire, etc.) et le sous service « Map Me » qui montre à l'utilisateur tous nos endroits proposés et lui donne la possibilité de se géolocaliser pour se retrouver entre plein d'endroits qui peuvent l'attirer vers une sortie plaisante, en dessous on retrouve les deux sous service de l'interface « Near By » :



Figure 35- Interface service « Near By ! », sous service « Events »



Figure 36- Interface service « Let's Go ! », sous service « Map Me »

• Interface service Guide Touristique “Let’s Go !”

Nous arrivons au service le plus important de tout notre projet qu'on a nommé « Let's Go ! ». L'utilisateur « client » lance le service qui est en lui-même un guide touristique qui recommande des endroits en relation avec le contexte Budget donné par l'utilisateur, et la longueur de son séjour à l'endroit où se trouve. Après soumission des informations demandées le service lance une recommandation de lieux qui peuvent satisfaire le taux d'argent de l'intéressé comme le montre la Figure 35-36 avec un budget de « 1200 000 » et une durée de séjour de 15 jours :

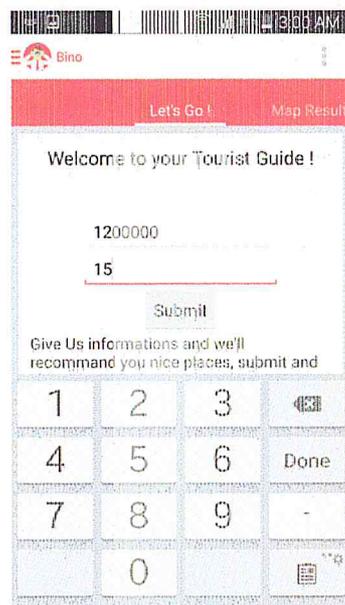


Figure 37- Interface Service « Let's Go ! », sous sous-section de données.



Figure 36 - Interface Service « Let's Go ! », éléments affichés.

• **Interface Paramètres (Paramètres de l'application) et About Us:**

Les dernières interfaces qu'on présente maintenant se réfèrent à l'interface Paramètres de l'application où l'utilisateur « loko » va gérer son profil en cas où il veut changer ses POU, son compte en changeant son adresse mail ou son mot de passe, et aussi les paramètres de langages, et à l'interface About Us un petit résumé sur les développeurs de l'application, dans les Figures de 37 à 40.

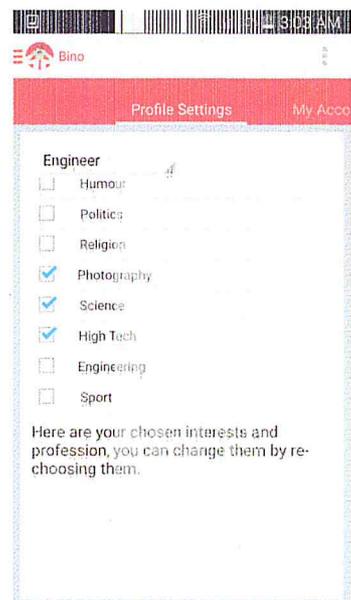


Figure 37 - Interface Settings (Paramètres de l'application et profil).

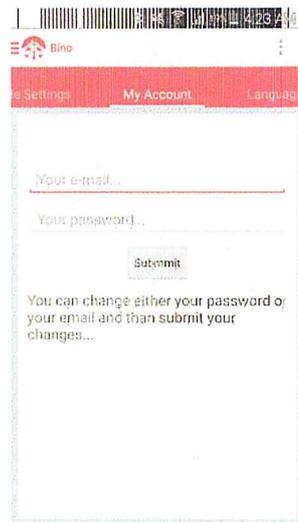


Figure 40- Interface Settings (Paramètre du compte)



Figure 41- Interface Settings (Paramètre de langage)



Figure 42- Interface About Us

Conclusion :

En conclusion ce chapitre qui se nomme le dernier de tous les chapitres, où on a clôturé avec les technologies qui ont permis à l'implémentation de notre projet du premier outil qui nous a permis la possibilité de faire notre première ligne de code en java, vers la création de toutes ces interfaces grâce à Android Studio en passant par notre Base sur le Cloud « PostgreSQL » qui nous a procuré la collecte des données de chaque utilisateur individuellement pour le bon déroulement de nos recommandations avec une expérience et une interface utilisateur digne de ce nom.



Conclusion Générale :

Nous Arrivons à la fin de notre thèse où on peut conclure que dedans, nous avons commencé par une étude sur les travaux existants. Cette étude nous a ramené à déduire que ces travaux se basent sur l'ingénierie de réalisation d'un système sensible au contexte. Ils se focalisent sur l'énumération des informations de contexte et sur comment les utiliser pour apporter des adaptations aux actions du système selon ces informations.

Ces travaux proposent plusieurs modèles et méthodes de recommandation sensible au contexte. Cependant ils considèrent que la sensibilité au contexte est une réaction liée aux changements de paramètres de l'environnement. Donc suite à nos objectifs, nous avons montré selon notre architecture basée SMA (System multi-agents) qu'il y a aussi d'autres changements sur lesquelles on peut se baser et qui sont les points d'intérêts (POI) de l'utilisateur et le contexte de son budget et la durée de séjour touristique.

De L'architecture on est passé au déploiement de l'application mobile, et pendant son développement on a touché à toutes les étapes nécessaire à son implémentation, avec une conception précise de la base de donnée selon le diagramme de classe qu'on a mis en place pour mieux là démontrer à savoir qu'elle est sur parse.com dans le Cloud et que presque toutes les applications android modernes utilise des BaaS (Backend as a Service) pour le bien et la facilité de son code presque prêt à utiliser dans l'environnement de développement (Android Studio).

A la fin on a bien clôturé avec l'implémentation en démontreront les outils qu'on a utilisé pour arriver à nos fin de programmation dans un environnement paisible sans difficulté la plupart du temps, puis on a démontré toute notre conception avec les interfaces finales de notre application, qui a bien prouvé notre architecture et notre base de donnée avec le jeu d'essai qu'on a mis en place dans les Figures précédente du chapitre 5.

Notre application prénommé Bino a été l'objet d'une étude bien finie sur la recommandation sensible au contexte dans le domaine touristique, tous les objectifs ont été atteints et la problématique a été bien résolue.

Référence Bibliographique :

[1] : L'informatique sensible au contexte, P 6.

[2] : L'informatique

mobile, <http://deptinfo.unice.fr/~renevier/corrections/theseHTML/informatiquemobile.html>, (01-04-2015).

[3] : Angela Cristina Carrillo Ramos, Agents ubiquitaires pour un accès adapté aux systèmes d'information, Université Joseph-Fourier - Grenoble I, 2007. France, P 61.

[4] : L'informatique sensible au contexte, P 7.

[5] : L'informatique sensible au contexte, P 4.

[6] : L'informatique sensible au contexte, P 5.

[7] : Mme RESSAD-BOUIDGHAGHEN Ourdia, Accès contextuel à l'information dans un environnement mobile : approche basée sur l'utilisation d'un profil situationnel de l'utilisateur et d'un profil de localisation des requêtes, Doctorat de l'université de Toulouse – France, P 45.

[8] : L'informatique sensible au contexte, P 8.

[9] : Apport des modèles pour l'adaptation d'applications en environnement ubiquitaire, P 20.

[10]: Wikipedia, <http://fr.wikipedia.org/wiki/Ubiquit%C3%A9> , (01-04-2015).

[11]: Francois Bourdon, « systèmes multi agents cours exposé dans le cadres du DEA IAA » ,2001

[12] : Alper Caglayan , « les agent application bureautiques Internet et intranet », 1998

S.Ladidi « de l'intelligence Artificielle Distribuée au systèmes multi agents » , 1993

[13] : Wikipédia, http://fr.wikipedia.org/wiki/Syst%C3%A8me_de_recommandation, (25/02/2015).

[14]: Podcast science, <http://www.podcastscience.fm/dossiers/2012/04/25/les-algorithmes-de-recommandation/>, (25/02/2015).

[15]: Negre Elsa, Les systèmes de recommandations, université de Paris, P 3.

[16]: Podcast science, <http://www.podcastscience.fm/dossiers/2012/04/25/les-algorithmes-de-recommandation/>.

[17]: Podcast science, <http://www.podcastscience.fm/dossiers/2012/04/25/les-algorithmes-de-recommandation/>, Negre Elsa, Les systèmes de recommandations, université de Paris, P 6.

[18]: Podcast science, <http://www.podcastscience.fm/dossiers/2012/04/25/les-algorithmes-de-recommandation/>. Negre Elsa, Les systèmes de recommandations, université de Paris, P 7.

[19] : Negre Elsa, Les systèmes de recommandations, université de Paris, P 12..13..14.

[20]: Podcast science, <http://www.podcastscience.fm/dossiers/2012/04/25/les-algorithmes-de-recommandation/>.

[21]: Podcast science, <http://www.podcastscience.fm/dossiers/2012/04/25/les-algorithmes-de-recommandation/>.Negre Elsa, Les systèmes de recommandations, université de Paris, P 27...28.

[22] : Negre Elsa, Les systèmes de recommandations, université de Paris, P 17..18..24..25.Robin Bruke, Integrating Knowledge-based and Collaborative-filtering Recommender Systems, 1999, university of California, P 70.

[23]: L'informatique sensible au contexte, P 7.Robin Bruke, Integrating Knowledge-based and Collaborative-filtering Recommender Systems, 1999, university of California, P 71.

[24]: Gediminas Adomavicius, Bamshad Mobasher, Francesco Ricci, and Alex Tuzhilin, Context-Aware Recommender Systems, P 71...72.

[25]: L'informatique sensible au contexte, P 6.

[26]: Podcast science, <http://www.podcastscience.fm/dossiers/2012/04/25/les-algorithmes-de-recommandation/>, (01/03-2015).

[27]: AntVoice, <http://www.antvoice.com/fr/recommandation-predictive/pourquoi-netflix-nas-pas-encore-atterri-en-france/#.VPb6zfmG9Np>, (01/03/2015).

[28]: Wikipedia, http://fr.wikipedia.org/wiki/Syst%C3%A8me_de_recommandation, (11-03-2015).

[29]: Francesco Ricci, Mobile Recommender Systems, University of Bolzano, Italy, 2010, P 12.

- [30]: Francesco Ricci, Mobile Recommender Systems, University of Bolzano, Italy, 2010, P 18.
- [31]: <http://www.blogdumoderateur.com/interview-deapz/> , (14-02-2015).
- [32]: Linas Baltrunas, Bernd Ludwig, Stefan Peer & Francesco Ricci, Context relevance assessment and exploitation in mobile recommender systems, 2011, P 05/13.
- [33]: Jongyi Hong a, Eui-Ho Suh a, Junyoung Kim b, SuYeon Kim c, Context-aware system for proactive personalized service based on context history, 2009, P 02.
- [34] : Mr Mili Seif Eddine, « CONCEPTION D'UNE ARCHITECTURE BASEE AGENTS POUR LA CREATION D'UN MARCHE VIRTUEL », mémoire pour l'obtention d'un diplôme de magister en informatique, P 31, 2008.
- [35]: Tao Gu, Xiao Hang Wang, Hung Keng Pung , Da Qing Zhang, "An Ontology-based Context Model in Intelligent Environments" , Department d'informatique Singapore.
- [36]: Zimmermann A., Lorenz A., Oppermann R., « An operational definition of context », Modeling and using context, Springer, p. 558-571, 2007.
- [37]: <http://fr.wikipedia.org/wiki/Android>, (10-06-2015).
- [38]: <http://www.hypnoz.net/android/pourquoi-choisir-android/>, (10-06-2015).
- [39]: <http://toutdanslapoche.fr/savoir-faire/base-de-donnees/>, (10-06-2015).
- [40]: http://en.wikipedia.org/wiki/Mobile_Backend_as_a_service, (10-06-2015).