

# PROJET ANDROID

## JAVA

## AKEI



# 1. SOMMAIRE

[1. Sommaire](#)

[2. Contexte](#)

[2.1. Service de location](#)

[2.2. Application mobile](#)

[3. Cahier des charges fonctionnel](#)

[4. Cahier des charges technique](#)

[4.1. Conseils de réalisation](#)

[4.2. Conseils techniques pour la mise au point de votre application](#)

[5. Productions attendues](#)

[6. Feuille de route du développement](#)

## 2. CONTEXTE

La société **Akei** est spécialisée dans la **vente de meubles et de produits de la maison** à prix attractif. Cette organisation possède de nombreux magasins en France et dans le monde entier.

Les produits sont visibles dans la **partie exposition** du magasin et sont organisés par « pièce » de la maison

- chambre,
- cuisine,
- séjour,
- etc.

Dans chaque magasin, les employés sont spécialisés par pièce.

Par exemple **Jean Baptiste** est toujours affecté à **l'espace chambre** (on parle aussi de rayon). Il doit connaître, ou du moins pouvoir renseigner au mieux les clients sur l'ensemble des produits de son espace :

- nom,
- description technique,
- prix,
- dimensions,
- poids.

Lorsque les équipiers se déplacent dans le magasin, il arrive parfois que des clients interrogent un employé sur un domaine qu'il ne connaît pas. Il faudrait que l'employé puisse orienter le client vers un de ses collègues en charge du produit dont il est question.

### 2.1. Service de location

Akei vend des produits parfois très volumineux. Les colis ne sont pas toujours transportables dans les véhicules des clients.

**Un service de location** est disponible dans chaque magasin où le client peut louer des camions et des véhicules utilitaires.

Chaque magasin a un parc de véhicules. Les clients peuvent choisir parmi ces véhicules celui qui leur semble le plus adapté.

Par exemple le Fiat Ducato de 20m3 a les dimensions intérieures suivantes :

- longueur : 2670mm
- largeur : 1870mm
- hauteur : 1662mm

On enregistre aussi

- la plaque d'immatriculation,
- le type de carburant, et
- le nombre de km actuel.

Akei a confié le développement de la solution applicative à **l'ESN Stesio**.

### 2.2. Application mobile

L'application mobile destinée aux employés d'Akei doit permettre de répondre facilement aux questions des clients sur les trois thèmes suivants:

- les produits et leurs caractéristiques
- les camions disponibles à la locations
- les employés et leurs spécialités

### 3. CAHIER DES CHARGES FONCTIONNEL

Les objectifs fixés dans ce contexte sont, dans un premier temps, la **visualisation d'informations sur un téléphone mobile** selon trois axes :

- **obtenir la liste des rayons** et pour chacun d'entre eux **les produits et leurs caractéristiques**.
- obtenir la **liste complète des magasins** Akei proposant des locations, puis les véhicules proposés ainsi que leurs caractéristiques
- obtenir l'**ensemble des rayons du magasin**, et pour chacun d'entre eux, **les employés spécialisés** ainsi que leurs caractéristiques.

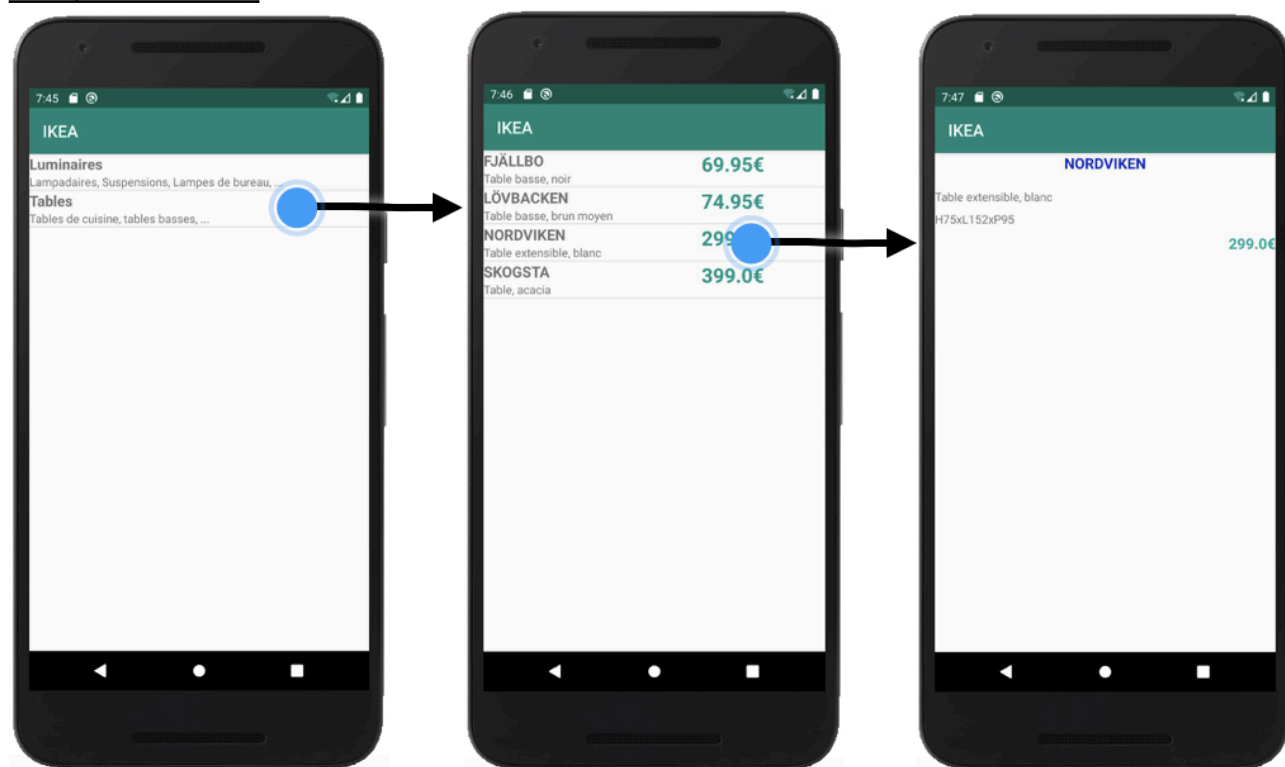
Pour ces trois axes, l'affichage devra être réalisé sur un même principe :

Affichage d'une liste principale, et, sur sélection d'un élément de la liste, affichage de la « sous liste ».

Après avoir sélectionné un élément de la sous-liste l'application affiche en détail l'élément sélectionné.

Chaque affichage fera l'objet d'un développement spécifique réalisé par un membre du groupe.

Exemple d'interface :



Un **champ de recherche** doit être disponible dans la 2ème interface pour permettre à l'utilisateur de filtrer les résultats selon le mot saisi. Le mot en question recherchera dans le nom du produit et dans ses caractéristiques.

Par exemple dans la capture d'écran ci-dessus, lorsque l'utilisateur saisit "table basse", les produits suivants sont affichés : FJÄLLBO et LÖVBACKEN.

Pour les autres axes, la fonctionnalité de recherche se basera sur les critères suivants :

- Véhicules : nom, marque, capacité
- Employés : nom, prenom, nom du magasin.

## 4. CAHIER DES CHARGES TECHNIQUE

La conception préalable des classes métiers du modèle sera faite en équipe, puis les axes spécifiques feront l'objet d'une mission distincte.

La réalisation du projet nécessite de suivre les étapes suivantes :

1. Conception du schéma global de données sous forme d'un **diagramme de classe UML**
2. Transformation du schéma de données en **schéma relationnel** puis en requêtes de création de tables
3. Ecriture des classes métiers
4. Conception des maquettes des interfaces utilisateurs permettant :
  1. De consulter la liste principale
  2. De consulter la sous liste correspondant à un item de la liste principale
  3. D'afficher les informations détaillées du sous-item sélectionné
5. Coder les classes DAO en Java permettant de gérer les données dans la base de données SQLite
6. Présenter la liste majeure (Java)
7. Présenter l'item majeur et la liste mineure (Java)

### 4.1. Conseils de réalisation

Pour réaliser votre application, vous pourrez dissocier les travaux de :

- Gestion des données
- Visualisation de la liste majeure
- Visualisation de la liste mineure

### 4.2. Conseils techniques pour la mise au point de votre application

Deux approches sont possibles pour générer les listes sur les deux premiers écrans :

#### Approche par ListView :

- Le type **ListView** peut être utilisé pour lister les éléments, ce type d'objet est proche du Spinner dans son implémentation. Cependant quelques adaptations sont nécessaires.
- La **consultation d'une liste** s'appuiera d'abord sur un simple Adapter, puis devra basculer vers un Adapter plus élaboré pour pouvoir visualiser de façon plus confortable les propriétés d'un objet métier, et arriver à la maquette dessinée.
- La **consultation de la liste mineure** sera mise au point en affichant d'abord l'item de liste majeure qui a été sélectionné. Puis la visualisation de la liste mineure pourra alors être mise au point.
- N'oubliez jamais qu'une solution est un ensemble de briques, mais que chaque brique peut être mise au point avant l'assemblage final.

#### Approche par LinearLayout :

Il est possible de ne pas faire appel aux **ListView** et générer l'affichage à l'aide de layouts. Dans ce cas, aucun Adapter n'est nécessaire. Cependant cette solution demande potentiellement davantage de code.

## 5. PRODUCTIONS ATTENDUES

- le schéma de données sous forme d'un **diagramme de classes**
- les **maquettes** dessinées des interfaces à réaliser
- les **captures d'écran** des interfaces réalisées, accompagnées d'un bilan qui tracera les différentes phases de réalisation de chaque écran ainsi que les difficultés rencontrées au cours de leur réalisation.
- le **code source JAVA** de l'application, pertinemment commenté.

## 6. FEUILLE DE ROUTE DU DÉVELOPPEMENT

Voici la feuille de route que vous devrez suivre :

Séance	Résultats attendus	Groupe/ Individuel
n°1 13/12	Répartition du travail Schéma de données sous forme d'un diagramme de classe Copies d'écran de Trello (à montrer chaque semaine...)	G G G
n°2 21/02	Schéma de données sous forme d'un diagramme de classe Développements du code Java Copies d'écran de Trello (à montrer chaque semaine...)	G I G
n°3 14/03	Développements du code Java Documentation du code Java	I I
n°5 21/03	Tests des méthodes Jeu d'essai Copies d'écran de Trello (à montrer chaque semaine...)	I I I
n°5 28/03		
n°6 04/04	Fusion des fonctionnalités dans une application unique Copies d'écran de Trello (à montrer chaque semaine...)	G G
n°7 11/04	Fusion des fonctionnalités dans une application unique Copies d'écran de Trello (à montrer chaque semaine...)	G G
n°8 18/04	Remise des documents	G

Une équipe de 2 ou 3 personnes est affectée à la réalisation de cette solution.

L'échéance de remise de l'application est prévue après **8 séances d'AP**