

**MobOp** : Capitole - Le catalogue de films sous Android  
Rapport du projet de semestre

Dan LUONG BA & Frédéric MONTET

**Professeurs** : Pascal BRUEGGER & Ninoslav MARINA

December 31, 2015



# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Conception de l'application</b>	<b>3</b>
2.1	But . . . . .	3
2.2	Concept . . . . .	3
2.3	Mock-ups . . . . .	4
<b>3</b>	<b>Développement</b>	<b>7</b>
3.1	Structure de l'application . . . . .	7
3.2	Base de donnée . . . . .	8
3.3	Liaisons avec les bases de données cinématographiques . . . . .	9
<b>4</b>	<b>Mode d'emploi</b>	<b>11</b>
4.1	Interface principale . . . . .	11
4.2	Détail des films . . . . .	11
4.3	Recherche de films . . . . .	12
4.4	Evaluation d'un film . . . . .	12
<b>5</b>	<b>Conclusion</b>	<b>13</b>

# Chapter 1

## Introduction

Dans la vie de tous les jours, il arrive parfois de ne pas savoir quel film regarder, voir même d'oublier quels sont les films que nous désirons voir. Capitole est une application apportant une solution à ce problème en proposant une gestion des films simple et efficace selon 3 listes: les films suggérés, les films à voir et les films vu.

Ce rapport explique les différents point clé de la conception et de la réalisation de cette application. Les problèmes rencontrés pendant ces étapes et un guide d'utilisation.

## Chapter 2

# Conception de l'application

### 2.1 But

L'application Capitole est basée sur la proposition de projet de semestre suivante:

“FilmList – Create a mobile application that builds a personal movie database – allow different categories and ratings. Connect it to the existing movies databases”

Le but est donc de développer une application pour la gestion d'un catalogue de film. Cette dernière doit permettre de classer les films d'un utilisateurs par catégorie et d'attribuer des notes pour les films. De plus, elle doit être connectée à une base de donnée de films en ligne pour récolter ses informations.

### 2.2 Concept

#### Le nom

Tout d'abord, nous avons changé le nom de projet de semestre donné (FilmList) par Capitole. Ceci pour donner une réelle identité à notre application et rappeler une des plus ancienne et plus grande salle de cinéma de Suisse toujours en activité, qui est située à Lausanne, le Capitole.<sup>1</sup>

#### Structure de l'interface

Les différentes situations qui confronte une personne au monde du cinéma peuvent se résumer de la manière suivante:

- La personne ne sait pas quoi regarder comme film.
- La personne ne se souvient plus quels sont les films qu'elle a envie de voir
- La personne ne se souvient plus les films qu'elle a vu.

---

<sup>1</sup>Le lien de vers le site du Capitole: <http://www.lecapitole.ch>

Du fait de ces 3 situations, pour l'application Capitole, nous avons fait le choix de créer 3 pages qui apportent une solutions aux problème cités ci-dessus. Soit, les pages suivantes:

**Suggestion** Une liste de proposition de films à l'affiche

**To see** Une liste de films sélectionné par l'utilisateur

**Seen** Une liste des films vu par l'utilisateur

Ces 3 pages seront mises dans l'ordre ci-dessus dans l'application car c'est cet ordre qui est, chronologiquement et logiquement, celle d'un film dans l'application Capitole. De la proposition d'un film à son archivage en passant par sa phase "à voir".

## Problématique de l'application

Une des problématiques de Capitole est la suivante:

"Comment obtenir les données des films et en proposer à l'utilisateur?"

Pour résoudre ce problèmes, plusieurs solutions sont envisageables:

1. La saisie des données des films est faite par l'utilisateur
2. Les films proposés sont uniquement ceux à l'affiche donné par un webservice
3. Les films proposés le sont grâce à un algorithme de suggestion en utilisant la base de donnée d'un webservice
4. L'utilisateur recherche lui même les films qu'il a envie de voir sur la base de donnée d'un webservice

La meilleur solution serait de faire des proposition de films à l'utilisateur qui sont pertinentes pour lui, soit d'envisager la troisième solution de la liste ci-dessus. Malheureusement, cette dernière possède un niveau de complexité que nous avons choisi d'éviter et nous nous rabattons donc sur l'association avantageuse d'une proposition de films à l'affiche et de films issus d'une recherche utilisateur.

## 2.3 Mock-ups

Les mock-ups de l'application sont les suivants:



(a) Page principale de Capitole

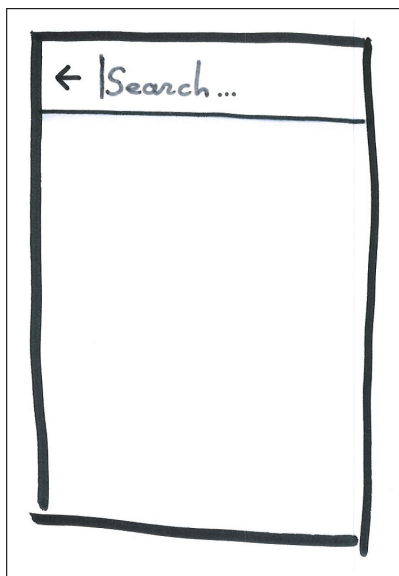


(b) Page du détail d'un film

Figure 2.1 – Mock-ups de Capitole - Principaux

Dans la figure 2.1 ci-dessus, nous avons prévu d'utiliser la `tabView` d'Android avec 3 fragments. Il sera possible de swiper pour passer d'un onglet à l'autre de l'application et ainsi voir les différentes listes de films.

Pour notre cas d'utilisation nous sommes chanceux car ce composant UI correspond exactement au besoin de notre interface et suit les guideline donnée par Google <sup>2</sup> concernant l'utilisation des onglets dans une application.



(a) Recherche de films



(b) Résultat d'une recherche

Figure 2.2 – Mock-ups de Capitole - Recherche

<sup>2</sup>Utilisation des tabs dans Android: <http://www.google.ch/design/spec/components/tabs.html#tabs-usage>

Ces derniers mock-ups mettent en avant l'utilisation du module de recherche de l'ActionBar d'Android. A l'aide de cet outil UI, il est possible d'intégrer un champ de recherche dans une interface utilisateur mobile de manière sobre et discrète.

## Chapter 3

# Développement

### 3.1 Structure de l'application

#### Séparation des responsabilités

Abordant Android pour la première fois, il a été difficile de créer un programme qui adopte un patron structurel <sup>1</sup> tout en prenant en compte les contraintes dues à la manière de programmer une application Android. Dans l'incertitude, nous avons choisi le modèle MVC pour son côté générique.

L'application Capitole est structurée ainsi:

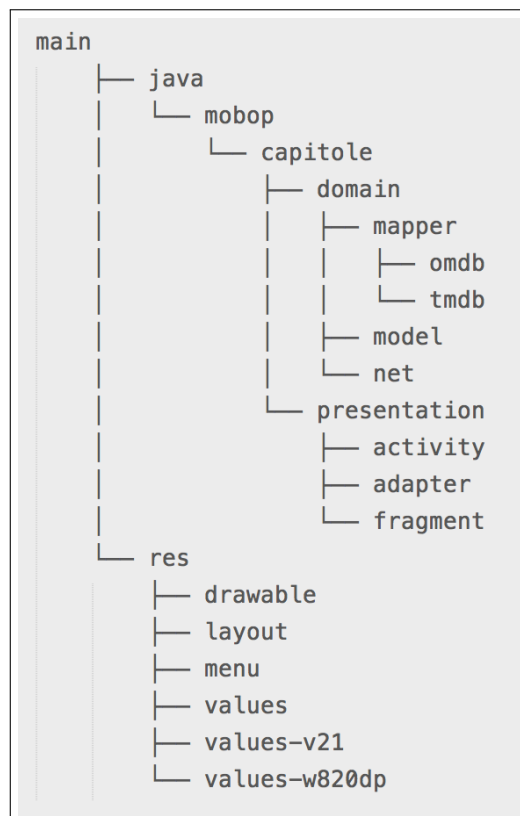


Figure 3.1 – Structure de l'application

---

<sup>1</sup>MVC, MVP, MVVC, etc.



Les principaux composant logiques de l'appli se trouvent dans le dossier Capitole. Dans ce dernier se trouve le dossier domain qui regroupe les composants suivant:

**mapper**

Les classes transformant les objets JSON reçu des webservice en objet Movie de Capitole

**model**

Les classes représentant la base de donnée

**net**

Les classes permettant de faire les appels aux webservice

Dans le dossier presentation se trouvent les composants classique de toute application Android, à savoir les classes Activity, Fragment et les Adapter. Ces derniers permettent, sans surprise, de gérer les vues de l'application qui se trouvent dans les XML situé dans le dossier res de l'application.

Concernant les contrôleurs de l'application, il n'y en a qu'un. Il est situé dans le dossier domain de l'application et

**Problèmes rencontrés****Couplage imposé**

En abordant le développement Android, il est facile de commencer à mettre du code métier dans les fragments et activités alors que ces classes ont comme but la gestion du cycle de vie d'une vue. Une telle pratique aurait comme résultat un code "spaghetti" difficilement maintenable et fortement couplé.

Afin de bien séparer les responsabilités des différentes classes, nous avons consulté certains blogs, notamment: <http://fernandocejas.com/2014/09/03/architecting-android-the-clean-way/> qui traite de ce sujet. La structure finale de l'application dont parle ce blog est pleine de bonnes pratiques cependant, la mise en pratique de ces dernières pour Capitole n'aurait pas été bénéfique en terme de productivité. Nous avons donc dû faire le choix de se rabattre à une architecture plus simple, un peu plus couplée et reposant sur la base de donnée Realm que nous verrons dans la section suivante.

## 3.2 Base de donnée

**Diagramme de classe**

Au début de la phase de conception de Capitole, nous avons créé le diagramme de classe. Le but de ce dernier, en plus de la conception de la base de donnée, était d'améliorer la compréhension de ce que nous allions faire dans notre application, sa structure et ses besoins.

**Schémas relationnel**

Une fois le diagramme de classe fait, nous l'avons décliné en schéma relationnel pour pouvoir l'implémenter avec SQLite et les différents outils Android pour la gestion de base de donnée.

## Problèmes rencontrés

### Conception trop hative du diagramme de classe

La conception du diagramme de classe a été faite avant de prendre en compte les contraintes inhérentes aux données provenant d'une API en ligne. De ce fait, nous avons commencé à développer selon un modèle de donnée et avons adapté ce modèle selon l'API. En soi, cette approche n'est pas mauvaise mais elle nécessite un temps plus important. Une manière plus productive aurait été de prendre en compte le modèle de donnée de l'API nous fournissant les données des films plus tôt pour pouvoir concevoir le modèle de donnée de Capitole en évitant de devoir faire des mapper. Cette dernière solution nous aurait permis de faire un mapping direct d'un objet JSON à un objet Java.

### Changement de base de donnée

Un des autres problème rencontré a été celui du choix de base de donnée. Au début, nous nous sommes dirigés vers l'outil basique de base de donnée relationnel Android: SQLite. Il s'est avéré que cet outil, bien que couvrant nos besoins, nécessite plus de code pour pouvoir être implémenté. Nous nous sommes donc orienté vers le base de donnée mobile non structurée Realm. Cette dernière a nécessité un temps d'apprentissage relativement court pour être prise en main et, contrairement à plusieurs autres SGBD pour mobile, est maintenu par une entreprise, ce qui implique une bonne maintenance.

## 3.3 Liaisons avec les bases de données cinématographiques

### Offre des bases de donnée cinématographique

Plusieurs base de donnée cinématographique en ligne existent, notamment:

- The Open Movie Database
- The Movie Database
- Internet Movie Database

La plus populaire étant sans aucun doute Internet Movie Database (IMDb) malheureusement, cette dernière ne fournit pas d'API public. La solution la plus ouverte est the Open Movie Database (OMDb) car cette API est complètement ouverte. C'est vers ce choix que c'est orienté notre développement dans un premier temps.

## Problèmes rencontrés

### Choix de l'API de films

Comme cité précédemment, c'est vers The Open Movie Database (OMDb) que s'est orienté notre choix de base de donnée cinématographique en ligne en premier lieu. Cependant ce choix n'était pas judicieux car ce webservice ne fournit que des fonctionnalités très limitées. Après avoir mieux consulté The Movie Database (TMDb), cette dernière propose une offre bien meilleure en terme de fonctionnalité et nécessite une authentification avec un token unique. Nous avons donc finalement opté pour TMDb.

### Chargement des données en ligne

Il a été plus compliqué que prévu de charger les données en ligne sur le terminal mobile. Pour commencer, le chargement de celles-ci a été effectué sur le main thread, ce qui a évidemment créé une erreur au lancement de l'application. Le chargement des données en ligne doit se faire sur un autre thread que celui qui gère l'interface utilisateur. Ainsi, la fluidité de l'interface est assurée. Pour faciliter le téléchargement des données de films, nous avons utilisé la librairie de Google: Volley<sup>2</sup>. A l'aide de cette librairie, le chargement des données a été aisé, qu'il s'agisse des données JSON ou des images.

---

<sup>2</sup><http://developer.android.com/training/volley/index.html>

## Chapter 4

# Mode d'emploi

### 4.1 Interface principale

L'interface de Capitole est composée d'une manière analogue à l'application de messagerie Whats App. L'emploi de son interface devrait donc être aisé pour tous. L'interface principale est composée de 3 onglets qui correspondent aux différents état qu'un film peut avoir.

#### Onglet Suggestion

C'est l'onglet qui s'ouvre à l'ouverture de l'application. Une fois celui-ci ouvert, les affiches des films se chargent dans l'interface et l'utilisateur peut scroller et cliquer sur les films pour voir leur détail. Si nécessaire, l'utilisateur peut actualiser la liste de films proposé en utilisant le geste "Swipe-to-Refresh"<sup>1</sup>.

#### Onglet To See

Dans cet onglet, l'utilisateur peu voir les films qu'il a choisi est ajouté à sa liste de films à voir. En faisant un clique long sur un des éléments de la liste, les différentes options disponibles s'affichent à l'écran dans un ContextMenu.

#### Onglet Seen

L'interface de cet onglet est quasi similaire à l'onglet To See. Une liste des films vu y est présente, en cliquant sur les éléments de la liste, on observe le détail du film et en cliquant longtemps, on voit les différentes action qu'il est possible d'effectuer avec un film.

### 4.2 Détail des films

Quelque soit l'état du film (suggéré, à voir ou vu), il est possible d'afficher un détail de ce dernier. Ce détail diffère légèrement selon l'état dans lequel le film se trouve.

#### Détail d'une suggestion

Dans cette interface, il est possible de consulter le détail d'un film suggéré et de l'ajouter à une des 2 liste de films de Capitole. À savoir, "To See" ou "Seen". On ajoute un film à une de ces deux liste

---

<sup>1</sup><http://www.google.ch/design/spec/patterns/swipe-to-refresh.html>

en cliquant sur le FAB button.<sup>2</sup> Un film ne peut être ajouté qu'une fois dans quelque soit la liste dans laquelle il a été ajouté.

### Détail d'un film à voir

Dans ce détail de film à voir, les informations affichées proviennent de la base de données locale. Deux actions sont disponibles: évaluer le film ou supprimer le film. L'évaluation d'un film se fait en cliquant sur le fauteuil. Lors d'une évaluation, il est implicite que le film a été visionné par l'utilisateur, il va donc passer dans la liste de films vus. Lors de la suppression d'un film suite à un clic sur la corbeille, le film va simplement être supprimé de la liste de films à voir.

### Détail d'un film vu

Le détail d'un film vu est analogue à celui d'un film à voir à la différence qu'il contient le commentaire et l'évaluation de l'utilisateur. Les actions disponibles dans l'Action Bar sont: Le retour du film dans la liste de films à voir ou la suppression du film.

## 4.3 Recherche de films

La recherche des films dans la base de données de l'API en ligne est possible depuis l'interface principale en cliquant sur la loupe disponible dans l'Action Bar. Une fois la loupe cliquée, un clavier s'affiche, l'utilisateur peut saisir le titre du film qu'il désire rechercher et une liste de résultats s'affiche. En cliquant sur un des résultats, l'utilisateur arrive sur la même interface de détail que celle utilisée lors des suggestions de film. L'utilisateur peut mettre ce film dans ses listes de films à voir ou vus.

## 4.4 Évaluation d'un film

Un film peut être évalué dans le cas où il est envoyé dans la liste de films vus. Ce passage peut se faire depuis deux emplacements de l'application. Soit depuis le détail d'une suggestion de film, soit lorsqu'un film est dans la liste "To See" et sera mis dans la liste "Seen" après avoir cliqué sur le fauteuil. À la suite d'une de ces actions permettant l'évaluation, un `AlertDialog` s'affiche et invite l'utilisateur à évaluer le film à l'aide d'une `RatingBar` et à y ajouter un commentaire.

---

<sup>2</sup>Le Floating Action Button représente l'action primaire qu'il est possible d'effectuer sur une interface <http://www.google.ch/design/spec/components/buttons-floating-action-button.html>

## Chapter 5

# Conclusion

Lausanne, December 31, 2015

Dan LUONG BA

Frédéric MONTET