

# DAR 2017 - Projet

Romain Demangeon

APR, LIP6, UPMC

18/09/2017

## Groupes

- ▶ groupes de 3 étudiants.
  - ▶ les groupes doivent être constitués aujourd'hui 18/09/2017:
    - ▶ envoi de la composition du groupe par mail  
romain.demangeon@lip6.fr.
    - ▶ Google Sheets (comme l'année dernière: <https://goo.gl/w9hSfk>)

## Originalité

L'ensemble du code de l'application doit être écrit par les étudiants du groupe. (bibliothèques publiques autorisées).

- ▶ Faire la différence entre contenu et outils (CSS, Bootstrap).

## Rapport (/4 points)

Rendu en **même temps** que le code, contient :

- ▶ **manuel** utilisateur,
  - ▶ description de l'**interface**,
  - ▶ *use cases*.
- ▶ description des **technologies** utilisées, de l'**architecture** de l'application et des **choix** effectués,
  - ▶ **schéma** de l'application,
  - ▶ choix **techniques** (pas seulement technologiques),
  - ▶ technologies: **courte** descriptions, surtout justification,
  - ▶ **extraits** de code digne d'intérêt.
- ▶ liste des **fonctionnalités**, points forts et points faibles de l'application,
- ▶ **compléments**:
  - ▶ descriptions détaillée des **extensions possibles**,
  - ▶ **étude de marché**, **monétisation**,
  - ▶ **coût** du projet.

- ▶ 4 pts: Rapport **complet** (10-12 pages), **illustré**, **propre** (Latex ?).
- ▶ **Attention**: le rapport **gain/effort** du rapport est élevé.

## Soutenances (/4 points)

- ▶ 24-27 novembre devant **plusieurs** enseignants.
- ▶ soutenance de **20** minutes (10 + 5 + 5):
  - 10 présentation de l'application, sa **mise en valeur**, l'explication des **choix** effectués,
    - ▶ éviter la **description** des technologies,
    - ▶ parler de **votre valeur ajoutée**.
  - 5 **démonstration** de l'application dans un navigateur,
    - ▶ l'application doit être **disponible en ligne** (publiquement) le jour de la soutenance et jusqu'à **trois semaines** plus tard.
  - 5 séance de **questions**.
    - ▶ **précisions**, **sécurité**,
    - ▶ **répartition** charge de travail.
    - ▶ Standard: **une** note pour le groupe.
    - ▶ Cas **exceptionnels**, avec l'accord public des membres du groupe.

- ▶ Deadline: **Dimanche 19 novembre 23h59 (Paris)**.
- ▶ Les soumissions **en retard** seront **pénalisées** ou ignorées.
  - ▶ **0 - 24h** (-2 points), **1 - 7 jours** (-5 points), **+7 jours** (0).
- ▶ Chaque soumission doit consister de **liens** vers:
  - ▶ le **rapport** du projet,
  - ▶ la **présentation** de soutenance (**à rendre après la soutenance**),
  - ▶ le **code source** du projet,
  - ▶ l'**application en ligne** réalisée.
- ▶ Soumissions **2014**: <https://goo.gl/v1ezKa>
- ▶ Soumissions **2015**: <https://goo.gl/KgPfSu>
- ▶ Soumissions **2016**: <https://goo.gl/w9hSfk>

## Barème:

- 4 Rapport
- 4 Soutenance
- 4 Code Source
  - ▶ clarté, commentaires, originalité.
- 8 Application
  - ▶ fonctionnalités, ergonomie, clarté, idées, utilité.

## Sujet

- ▶ Sujet **libre** (originalité prise en compte).
- ▶ Une composante à **respecter**:
  - ▶ **Social**: utilisateurs avec profils, organisés en réseau, pouvant se communiquer des informations.
- ▶ **Exemples** de **projets**:
  - ▶ Site d'évaluation de produits culturels (2013).
  - ▶ Application de rencontre dans les transports en commun (2014).
  - ▶ Application de création de matériels sportifs (2015)
  - ▶ Réseau de recherche de paroles de chansons (2016).

## Consignes Générales

L'application doit:

- ▶ suivre une architecture **client / serveur** web,
  - ▶ **la charge du serveur doit être non triviale**,
- ▶ utiliser Java **Servlet** et éventuellement JSP (serveur),
  - ▶ Pas de **framework** web.
- ▶ interroger **régulièrement** une **API REST**,
  - ▶ requêtes **justifiées, fréquentes** et **dynamiques**.
- ▶ être **hébergée** en ligne gratuitement (perso, ou GAE, ou AWSEC2),
- ▶ utiliser **HTML5, CSS3** et **JavaScript** (client),
- ▶ comporter (au moins) **un appel asynchrone** (AJAX ou JSONP),
- ▶ permettre les **accès simultanés**.

## Consignes API

- ▶ Exemples d'APIs:
  - ▶ Open data Ile-de-France: nombreuses API régionales,
  - ▶ Open data RATP: transports en commun,
  - ▶ Google Maps APIs: localisation, plans, ...
  - ▶ Autres APIs: cinéma, musées, ...
- ▶ Consignes:
  - ▶ L'application doit être plus qu'une interface de l'API.
  - ▶ Bien respecter les règles de l'API.
  - ▶ Utilisation réactive/proactive/statique.

## Valorisation

- ▶ Portée du projet.
- ▶ Travail additionnel: (étude de marché, cadre légal, ...).
- ▶ Sécurité de l'application.