

# Guide de projet L023/AI12 (A23) : Logiciel de partage de musique

## 1. Fonctions du logiciel

### *Créer un profil*

---

- **Déroulement** : L'utilisateur choisit un pseudo et un mot de passe, créant un profil local sur sa machine. Des données personnelles optionnelles (nom, prénom, âge, icône, etc.) sont également demandées.
- **Remarque** : Attention, rien n'empêche a priori deux utilisateurs de prendre le même pseudo (par exemple avec une application client/serveur en créant un compte de même pseudo sur deux serveurs différents, puis l'un des deux comptes se connectant par la suite sur le serveur de l'autre).
- **Fonctionnement avancé** : L'utilisateur renseignera aussi un avatar (une image) dans son profil.

### *Connexion et découverte d'utilisateurs*

---

- **Déroulement** : L'utilisateur se connecte à l'application par son pseudo et mot de passe. L'application affiche la liste des utilisateurs distants actuellement connectés.
- **Piste de travail** : L'application peut être de type client/serveur ou de type client décentralisé. Attention à bien étudier les avantages et défauts de chacun avant de choisir (l'argumentaire du choix fera partie du dossier de conception). Pour un type décentralisé, la découverte d'autres applications se fait à travers des noeuds initiaux dont les adresses et le numéro de port sont spécifiés par l'utilisateur. Pour un type client/serveur, l'adresse et le port du serveur est à spécifier ; plusieurs serveurs différents peuvent fonctionner en parallèle sans se connaître, et un utilisateur peut changer de serveur sans perdre ses données personnelles et sa musique.

### *Partager des morceaux de musique*

---

- **Déroulement** : L'utilisateur peut choisir des morceaux de musique (format mp3) et les enregistrer dans l'application pour les partager aux autres utilisateurs. Un morceau a un titre, et peut avoir un album et un artiste.
- **Fonctionnement avancé** : Plusieurs tags peuvent être associés à un morceau de musique, et l'utilisateur peut choisir s'il est partagé ou non aux autres utilisateurs. Les autres utilisateurs peuvent noter le morceau, et mettre des commentaires.

### *Supprimer un morceau*

---

- **Déroulement** : L'utilisateur peut voir la liste de ses morceaux pour les écouter et supprimer un morceau de sa liste.

### *Voir la liste des morceaux de musique proposés par un utilisateur et les écouter*

---

- **Déroulement** : L'utilisateur peut voir la liste des morceaux partagés par un autre utilisateur connecté. Il peut voir leurs informations et les écouter. L'utilisateur peut sauvegarder un morceau qui lui plaît sur son ordinateur.
- **Fonctionnement avancé** : L'utilisateur peut noter, commenter et voir les notes et les commentaires des morceaux.

### *Rechercher de la musique*

---

- **Déroulement** : L'utilisateur peut faire des recherches parmi tous les morceaux actuellement partagés, par album, artiste ou titre. Il peut voir les informations, des morceaux résultats de la recherche, et les écouter. L'utilisateur peut sauvegarder un morceau qui lui plaît sur son ordinateur.
- **Fonctionnement avancé** : L'utilisateur peut faire des recherches par tag et par note. Il peut voir les notes et commentaires des morceaux et les noter et commenter lui-même.

### *Exporter et importer un profil*

---

- **Déroulement** : L'utilisateur peut exporter son profil local pour l'importer et écouter ses morceaux sur une autre machine.
- **Fonctionnement avancé** : les notes et commentaires des morceaux font également partie du profil.

### *Fonctionnement avancé : Gérer ses listes de contact*

---

- **Déroulement** : L'utilisateur peut ajouter des contacts dans une liste, et les gérer en différentes catégories (amis, famille, pop, rock, etc.). Il peut donner des droits à chaque catégorie pour chacun de ses morceaux : droit de voir les informations du morceau, droit de l'écouter, et droit de noter et commenter.

## **2. Moyens**

### *Environnement informatique*

---

- Gestion de projet :
  - Ganttproject
  - Microsoft Office Project
  - gestion d'issues Github
  - kanban picasoft (type trello) : <https://kanban.picasoft.net/>
- UML
  - PlantUML (recommandé) : [plantuml.com](http://plantuml.com), [planttext.com](http://planttext.com)
  - Dia
- Environnement de programmation
  - Netbeans
  - Eclipse
  - sous Java 1.8+
- Gestion de version :
  - Gitlab utc
  - SVN (à la demande)
  - plugin SVN, Git et CVS sous Netbeans
  - outil Tortoise SVN
  - git en ligne de commande
- Autres logiciels de l'UTC:
  - Microsoft Office,
  - owncloud : répertoire partagé pour l'échange de documents,
  - etc.

### *Autres logiciels (non fournis par l'UTC)*

---

- Outils UML libres utilisables sans installation : dotproject, White Star UML, Websequence diagrams (outil en ligne), Modelio, ArgoUML, Topcased, etc.

- SVN repository gratuit : Google Code, Unfuddle, OpenSVN
- Git : github
- Logiciel de gestion de projet libre : dotproject, etc.
- Outils de qualimétrie : **Sonar** (sonarqube, sonarlint), XRadars, etc.

### *Echange d'information*

---

- Discussions pendant les séances (formelles / informelles)
- SVN, CVS, Git (code source et documentation)
- Mattermost : [https://team.picasoft.net/signup\\_user\\_complete/?id=xaf4g5xpwfdgikmd5usmfmdxfw&md=link&sbr=su](https://team.picasoft.net/signup_user_complete/?id=xaf4g5xpwfdgikmd5usmfmdxfw&md=link&sbr=su)  
Moodle : forum étudiant  
Autres forums ?
- Mail
- autres possibilités : slac, discord, google groups, facebook, etc.

### *Pour JAVA : tutoriels et docs*

---

#### **Qwant/DuckDuckGo est votre ami !**

- La JAVAdoc : INDISPENSABLE
  - en anglais (Java 1.8) : <https://docs.oracle.com/javase/8/docs/api/>
  - indice : la classe MediaPlayer de JavaFX peut être utile au projet.
- JAVA
  - en anglais : *tutoriel oracle* <http://download.oracle.com/javase/tutorial/>
  - en français : <http://jmdoudoux.developpez.com/cours/developpons/java/>
- Points particuliers :
  - Serialization (pour transporter des objets JAVA à travers le réseau)
    - en anglais : <http://www.oracle.com/technetwork/articles/java/javaserial-1536170.html>
    - en français : <http://ydisanto.developpez.com/tutoriels/j2se/serialisation/partie1/>
  - IHM en JavaFX
    - en anglais : [http://docs.oracle.com/javafx/2/get\\_started/jfxpub-get\\_started.htm](http://docs.oracle.com/javafx/2/get_started/jfxpub-get_started.htm)
    - en français : <https://code.makery.ch/fr/library/javafx-tutorial/>

## **3. Déroulement du projet**

### *Demande*

---

L'application doit être développée en programmation orientée objet (langage au choix). Vous pouvez utiliser des bibliothèques (attention aux droits et licences).

### *Découpage structurel imposé*

---

Quatre sous-projets :

1. Communication : Ce module est responsable de l'acheminement des données dans le logiciel, et de l'envoi et de la réception des messages réseaux. Si le choix d'une architecture serveur est faite, il devra aussi implémenter l'IHM serveur (modification du port de connexion, par exemple).
2. Gestion et traitement des données : Ce module comprend les structures de données liées au profil utilisateur, et aux moreaux. Il est responsable de leur chargement et de leur sauvegarde.
3. IHM-Main : Ce module gère l'interface graphique principale de l'application : connexion, liste des utilisateurs et des morceaux, recherche de morceaux...

4. IHM-Music : Ce module gère l'interface graphique pour un morceau particulier : ajout de morceau, écoute de morceaux, visualisation et modification des paramètres d'un morceau, notes et commentaires d'un morceau...

Vous pouvez redéfinir les modules si vous le souhaitez. La décomposition doit être validée par l'encadrement. Elle est à donner dans le dossier de conception et doit être prise en compte dans les diagrammes de séquence.

## Organisation

---

Six responsables pour chaque équipe :

1. Directeur de projet : responsable communication avec le client et les autres équipes de sous-projet
2. Manageur de projet : responsable gestion de projet (estimation des charges, planification, mise à jour des indicateurs)
3. Concepteur : responsable de la conception
4. Développeur : responsable du développement
5. Chef qualité : responsable de la qualité (documentation, tests, qualité du code, fautes d'orthographe, etc.)
6. Etude : responsable des solutions techniques (y compris leur prototypage)

Possibilité de changer de rôle en cours de semestre par demande justifiée.

### **Tout le monde participe à toutes les tâches !**

(mais c'est la responsabilité du responsable si ça ne fonctionne pas)

## Délivrables

---

- Plan de management de projet
  - a. Découpage du projet
  - b. Estimation des charges
  - c. Planification
  - d. Déroulement du projet (problème, reprise des étapes d'analyse, nouvelle planification, etc.)
  - e. Analyse de risque
  - f. Actions qualité
  - g. Réflexion sur l'ingénierie soutenable du logiciel
  - h. Pour chaque membre du module : fiche de travail individuelle indiquant précisément les tâches effectuées, le temps passés sur chacune, le temps initial prévu, et quelques commentaires (facilité de réalisation, problèmes rencontrés, etc.).
  - i. Auto-notation de chaque sous-projet (12 points par membre à répartir comme entre les membres du sous-projet, sauf soi-même).
  - j. Bilan
- Dossier de conception
  - a. Choix de conception : question, solutions possibles, argumentation et décision
  - b. Diagramme des cas d'utilisation
  - c. Diagrammes de séquences (pas forcément tous)
  - d. Interfaces
  - e. Diagramme des classes
- Dossier de réalisation
  - a. Choix technologique : question, solutions possibles, argumentation et décision
  - b. Points durs (difficultés particulières rencontrées)
  - c. Points particuliers (exemples d'algorithme, approfondissements possibles, etc.)
- Code source et exécutable (y compris les bibliothèques nécessaires, s'il y en a, la documentation et les manuels utilisateurs)

## Echéances

---

- Pour le vendredi 6/13 octobre : Dossier de conception, et première version du plan de management de projet

- Pour la soutenance : Plan de management de projet (version finale), dossier de réalisation, code source et exécutable

### *Plan possible pour la soutenance (première semaine de janvier)*

---

- Organisation générale (10mn) :  
organisation et déroulement globaux du projet, cas d'utilisation et classes générales de l'application
- Présentation de chaque sous-groupe (30mn) :
  - présentation de chaque responsable (5mn)
  - question (10mn)  
**Attention** : une question peut être posée à n'importe quelle personne sur n'importe quel sujet !
- Démonstration de l'application (20mn)

**Longue durée** : environ 3 heures = le résultat de 1000 heures de travail (voire plus)