

# INFO-F-307 : Génie logiciel et gestion de projets

## Projet Flashcards

Professeur : FRÉDÉRIC PLUQUET

Assistants : HUGO CALLEBAUT, ANTHONY CNUDE, MARTIN COLOT,  
ABEL LAVAL, et YANNICK MOLINGHEN

Université Libre de Bruxelles  
2022-2023

## 1 Introduction

Pour ce projet, vous allez développer une application pour la gestion de projets exécutable sous *Linux*, *Windows* et *MacOS*. Pour ce faire, vous emploierez le langage de programmation *JAVA*.

Le présent document est composé de deux parties. La première est technique et elle a pour but de vous expliquer la structure du dépôt utilisé pour le projet et de vous aider à configurer l'accès à ce dépôt. La seconde partie reprend les histoires que nous vous demandons d'implémenter lors des itérations.

## 2 Détails techniques

Nous utilisons la variable  $xx$  pour votre numéro de groupe (i.e. 01) et  $i$  pour l'itération en cours.

### 2.1 Logiciels

**Jetbrains IntelliJ** disponible sur <https://www.jetbrains.com/idea/>

**N.B.** Chaque projet qui, lors d'une livraison, ne pourra pas être correctement importé et démarré à partir de la version IntelliJ 2022.3 recevra une note nulle pour l'itération concernée.

**Java >= 18** téléchargeable sur <http://www.oracle.com>

**N.B.** Toute version à partir de Java 18 peut être utilisée pour le projet. Veillez à indiquer dans le fichier `README.md` la version à utiliser pour votre projet, ainsi que les instructions pour importer correctement votre projet dans IntelliJ.

## 2.2 Structure du dépôt

Chaque groupe aura accès à son propre dépôt Git, disponible à l'adresse :

<https://gitlab.ulb.be/ulb-infof307/2023-groupe-x.git>

Le structure principale du dépôt ne peut en aucun cas être modifiée. Voici la structure initiale et la signification des dossiers :

**src** dossier source qui va contenir le code source de votre application sous le package `ulb.infof307.gxx`. La classe principale qui lancera votre application est la classe `Main` de ce même package. Vous pouvez bien sûr ajouter des sous-packages, des classes et modifier l'implémentation de cette classe `Main`.

**test** est également un dossier source contenant tous vos tests unitaires. Les classes de tests doivent être nommées selon la convention `TestClass` et implémenter les tests unitaires et/ou fonctionnels de la classe `Class`.

**dist** contiendra vos jars exécutables sous le format `gxx-iteration-i.jar`. Pour exporter un fichier jar dans IntelliJ, il faut d'abord définir un artifact et ensuite le compiler.

Pour définir l'artifact du jar :

1. File → Project Structure...
2. Artifact → Cliquez sur "+" → JAR → From module with dependencies...
3. Indiquez la classe main (e.g., `ulb.infof307.gxx.Main`) puis cliquez sur OK
4. Vous pouvez alors modifier le nom du jar (champs Name), le dossier de sortie (champs Output directory),...
5. Cliquez sur OK

Pour compiler le jar :

1. Build → Build Artifacts...
2. Sélectionnez l'artifact que vous venez de créer et cliquez sur Build.

**doc** contiendra la Javadoc générée depuis l'outil Java (depuis le menu d'IntelliJ, Tools, Générer la Javadoc, ne sélectionner que le sous-dossier "src").

**lib** contiendra les libraires que vous utiliserez après avoir reçu un accord préalable des assistants.

**team** contiendra la version courante des documents que vous utiliserez pour la gestion de votre équipe pendant le projet, c'est-à-dire uniquement les fichiers `*.md`, `*.tex`, `*.xls`, `*.ods` et `*.pdf` contenant la répartition des tâches, les burndown charts et tout autre diagramme que vous auriez employé dans la gestion de votre projet. Dans ce dossier, vous trouverez un fichier template pour l'encodage de la répartition des charges et la réalisation des burndown charts. Vous devrez mettre à jour **régulièrement** ce fichier sur base des activités réalisées dans le cadre des différentes itérations. Si vous voulez utiliser des images, vous devrez créer un sous-dossier "images".

### 3 Situation et objectifs

La session d'examens de janvier vient de se terminer et le taux de réussite en première année de sciences informatiques est consternant. Seulement 10.8% de réussite, c'est le taux le plus bas depuis l'invention de l'ordinateur.

Les plus grands experts du milieu ont été contactés et le verdict est clair. Tous s'accordent pour dire que le problème vient des étudiants qui n'ont pas appris à étudier correctement entre autres à cause du Covid. Étant vous-mêmes des étudiants et vous sentant concernés par ce problème vous vous êtes proposés pour créer (bénévolement) une application qui les aidera à réussir leurs études.

Pour ce faire, vous allez développer une application d'aide à l'étude basée sur un système de flashcards. Les flashcards sont un outil de révision simple et efficace ; au recto de la carte est inscrit une question et au verso la réponse à la question. L'étudiant peut alors parcourir les questions et tester ses connaissances grâce à ces cartes.

Le principe de base étant très simple, il vous sera aussi demandé d'ajouter de nouvelles fonctionnalités pour vous différencier de la concurrence telles que la correction automatique des questions, une probabilité de sélection des cartes plus élevée pour les cartes les moins bien maîtrisées, ... Les étudiants devront évidemment aussi être capables de créer de nouveaux jeux de cartes et de se les partager entre eux.

De plus, pour enfoncer le clou et laisser toute la concurrence derrière vous, vous pourrez aussi ajouter un mode détente à votre application. En effet, avec la multiplication des écrans, la durée de concentration a chuté et pour éviter que celle-ci ne dérive trop, il est souhaitable que les étudiants restent sur l'application dédiée aux études. Dans ce mode détente les règles seraient différentes, et un système de points serait mis en place. Deux joueurs pourraient par exemple s'affronter en duel et un système de récompenses pourrait être mis en place.

Finalement le but ultime de votre application serait d'être déployée en version mobile (Android) pour toucher un maximum d'étudiants dans le besoin.

## 4 Histoires

### Histoire 1 : GESTION DES COMPTES UTILISATEURS

Description : Pour accéder au programme l'utilisateur doit entrer son pseudo et son mot de passe. S'il n'a pas encore de compte, il peut en créer un (local à l'ordinateur) en choisissant un pseudo et un mot de passe. Le logiciel vérifie que le pseudo n'est pas déjà utilisé. Chaque utilisateur aura alors accès à ses propres jeux de cartes et statistiques. Il peut également modifier son mot de passe grâce à un menu d'édition de compte.

Priorité Client : 1

Risque Développeurs : 1 2 3

Introduit dans l'iteration : 1

État : fini

6

### Histoire 2 : GESTION DES PAQUETS DE CARTES

Description : L'utilisateur peut créer un jeu de cartes, lui donner un nom et une ou plusieurs catégories (par exemple, le nom d'un cours pour lequel le paquet aide à la révision ou le type de questions qu'on trouve dans le paquet). Une fois le paquet de cartes enregistré, il apparaît dans un menu permettant de parcourir les jeux de cartes.

Il est possible de filtrer les jeux de cartes pour n'afficher que ceux qui appartiennent à une certaine catégorie. L'utilisateur peut également supprimer un jeu de cartes ou le sélectionner pour l'éditer.

Priorité Client : 1

Risque Développeurs : 1 2 3

Introduit dans l'iteration : 1 et 2

État : fini

18

### Histoire 3 : CRÉATION DE CARTES DANS UN PAQUET

Description : Lorsque l'utilisateur édite un paquet de cartes, il peut y ajouter autant de cartes qu'il le souhaite. Chaque carte est composée de deux faces : un recto (contenant une question) et un verso (avec la réponse). Il peut également supprimer une carte ou en modifier une existante.

Priorité Client : 1

Risque Développeurs : 1 2 3

Introduit dans l'iteration : 1

État : fini

6

## Histoire 4 : ÉTUDIER UN JEU DE CARTES

Description : L'utilisateur peut étudier un jeu de cartes en le sélectionnant dans le menu. Le programme lui présente chaque carte du paquet dans un ordre aléatoire, côté question. Quand l'utilisateur le demande, la réponse est affichée.

Il indique alors au programme s'il connaissait ou non la réponse grâce à des boutons. Plusieurs boutons sont disponibles pour indiquer son degré de connaissance : *Très bonne, bonne, moyenne, mauvaise ou très mauvaise connaissance.*

Priorité Client : 1

Risque Développeurs : 1 ② 3

Introduit dans l'iteration : 2

État : *fin:*

5

## Histoire 5 : MÉTHODE DE TIRAGE

Description : Quand l'utilisateur étudie un paquet de cartes, l'application lui présente les cartes de sorte à ce que celles qui sont le moins bien connues apparaissent plus souvent.

Priorité Client : 2

Risque Développeurs : 1 2 ③

Introduit dans l'iteration : 1

État : *Fini*

6

## Histoire 6 : MODES DE JEU

Description : Deux modes de jeu sont proposés.

Le premier est le mode «révision» où les questions les moins maîtrisées reviennent plus souvent de sorte à apprendre la matière.

Le second est le mode «libre» où l'utilisateur peut s'amuser librement avec les cartes et essayer d'obtenir le meilleur score sur un jeu de carte choisi.

Priorité Client : 2

Risque Développeurs : 1 2 ③

Introduit dans l'iteration :

État :

4

## Histoire 7 : QUESTIONS DE DIFFÉRENTS TYPES

Description : Lorsqu'un utilisateur crée une carte, il doit pouvoir choisir le type de carte (textuelle, audio, vidéo, ...). On doit pouvoir mélanger les types de cartes dans un même paquet.

Priorité Client : 3

Risque Développeurs : ① 2 3

Introduit dans l'iteration :

État :

8

## Histoire 8 : RÉPONSES DE DIFFÉRENTS TYPES

Description : Les réponses à une question peuvent être de différente nature :

- Texte à trous à compléter
- Choix multiple
- Réponse ouverte
- ...

Priorité Client : 2

Risque Développeurs : 1 ② 3

Introduit dans l'iteration : 2

État : fini

9

## Histoire 9 : STATISTIQUES D'ENTRAÎNEMENT

Description : Le programme offre deux niveaux de statistiques à l'utilisateur. D'abord, des statistiques globales comprennent le temps d'entraînement total et par jour, le nombre de jeux de cartes utilisés et une liste des jeux de cartes les plus utilisés.

Il donne également accès à des statistiques par paquet de cartes. Celles-ci reprennent le nombre de sessions d'études réalisées sur ce jeu et un indice de maîtrise du paquet. Il permet également de voir l'évolution de ces statistiques au fil des jours sous la forme d'un diagramme d'évolution. Pour chaque paquet, il est possible de voir les cartes triées en fonction de leur connaissance par l'utilisateur.

Priorité Client : 3

Risque Développeurs : ① 2 3

Introduit dans l'iteration :

État :

10

## Histoire 10 : CONNEXION À UN SERVEUR

Description : L'utilisateur peut se connecter à un serveur à l'aide d'un login et d'un mot de passe. Ce serveur retient les jeux de cartes de l'utilisateur.

Priorité Client : 2

Risque Développeurs : ① 2 3

Introduit dans l'iteration :

État :

10

## Histoire 11 : STORE DE JEUX DE CARTES

Description : Lorsque l'utilisateur est connecté au serveur, il peut se rendre sur un *store* où il peut publier des jeux de cartes et télécharger ceux publiés par d'autres utilisateurs.

Priorité Client : 3

Risque Développeurs : ① 2 3

Introduit dans l'iteration :

État :

10

## Histoire 12 : SCORE ET CLASSEMENT

Description : Pour chaque jeu, l'utilisateur peut enregistrer son score basé sur sa connaissance des cartes et sur son temps de réponse. S'il le souhaite, son score est ensuite mis en ligne.

L'utilisateur peut ensuite consulter un classement en ligne qui recense les scores des joueurs pour chaque jeu de cartes.

Priorité Client : 3

Risque Développeurs : 1 ② 3

Introduit dans l'iteration :

État :

6

## Histoire 13 : VERSION MOBILE

Description : L'utilisateur peut se connecter sur son compte à l'aide de son téléphone et effectuer ses révisions dessus.

Priorité Client : 3

Risque Développeurs : ① 2 3

Introduit dans l'iteration :

État :

12

## Histoire 14 : LECTURE AUDIO

Description : Comme certains utilisateurs ont une meilleure mémoire auditive que visuelle, l'utilisateur doit pouvoir se faire dicter le contenu textuel d'une carte à l'aide d'une synthèse vocale.

Priorité Client : 3

Risque Développeurs : 1 2 ③

Introduit dans l'iteration :

État :

4

## Histoire 15 : IMPORT/EXPORT DE JEUX DE CARTES

Description : L'utilisateur peut exporter un jeu de cartes sous forme de fichier. Ce fichier doit être ré-importable dans l'application.

Priorité Client : 2

Risque Développeurs : 1 ② 3

Introduit dans l'iteration :

État :

6

## Histoire 16 : IMPRESSION DE JEUX DE CARTES

Description : Comme certains utilisateurs préfèrent les supports papiers, l'utilisateur doit pouvoir exporter ses jeux de cartes sous forme d'image ou de PDF imprimable.

Priorité Client : 2

Risque Développeurs : 1 ② 3

Introduit dans l'iteration :

État :

4

Histoire 17 : ~~TeX~~ ET HTML DANS LES CARTES

Description : L'utilisateur peut écrire des cartes contenant du code ~~TeX~~ et/ou HTML qui est correctement interprété lors de la visualisation.

Priorité Client : 1

Risque Développeurs : 1 ② 3

Introduit dans l'iteration :

État :

8



## Histoire 18 : Succès

Description : Lorsqu'un utilisateur publie un jeu de cartes sur le store, il peut spécifier des *succès* qui y sont propres. Un succès se caractérise par un titre, une description et une progression. Des exemples de succès propres aux jeux de cartes sont

- **Top chrono** : Répondre aux questions en un temps minimal
- **Perfectionniste** : Répondre correctement à toutes les questions d'un jeu
- ...

De plus, le store prévoit des *succès* génériques qui sont transversaux à tous les jeux de cartes comme

- **Un nouvel espoir** : Mettre son premier score en ligne
- **Question pour un Champion** : Répondre correctement à 100 cartes différentes
- **Petit producteur** : Publier 5 jeux de cartes

Priorité Client : 3

Risque Développeurs : ① 2 3

Introduit dans l'iteration :

État :

18

## Histoire 19 : NOTIFICATIONS DE RÉVISION

Description : L'utilisateur peut spécifier un objectif de révisions, par exemple en nombre de révisions par semaine. L'application envoie des notifications en conséquence pour inciter l'utilisateur à réviser.

Priorité Client : 3

Risque Développeurs : 1 ② 3

Introduit dans l'iteration :

État :

6

## Histoire 20 : CARTES À CHOIX DE DIFFICULTÉ

Description : Après avoir reçu la question, l'utilisateur a le choix de :

- Donner une réponse librement
- Demander quatre options pour un choix multiple
- Demander deux options pour un choix multiple

Le score de l'utilisateur augmentera plus ou moins en fonction de la difficulté choisie en cas de réponse correcte.

Priorité Client : 2

Risque Développeurs : 1 2 3

Introduit dans l'iteration :

État :

9

## Histoire 21 : CARTES MODULABLES

Description : Une même question peut avoir des paramètres aléatoires. Par exemple une question demandant de calculer la somme de deux nombres pourrait avoir deux nombres aléatoires comme opérandes. La réponse correcte sera dépendante de ces paramètres.

Il doit donc être possible de créer des formules lors de la création de cartes qui seront évaluées lors de l'affichage de la carte.

Priorité Client : 3

Risque Développeurs : 1 2 3

Introduit dans l'iteration :

État :

4

## Histoire 22 : DUEL

Description : Plusieurs utilisateurs peuvent s'affronter dans un mode duel dans lequel la même question est présentée aux joueurs. Chacun gagne des points en fonction de sa performance.

Priorité Client : 3

Risque Développeurs : 1 2 3

Introduit dans l'iteration :

État :

16