



## TelecomNancy FlashCards

“ TelecomNancy FlashCards est une application qui permet de favoriser l'apprentissage en utilisant des flashcards (ou cartes questions-réponses numériques). Cette application propose deux modes d'utilisation : le mode "création" et le mode "apprentissage". Le mode création permet de créer des piles de cartes dont une face contient une question et l'autre face la réponse à cette question. Le mode apprentissage permet de réviser le contenu des cartes. ”

### Sujet

L'objectif de ce projet est de réaliser une application de flashcards qui sont largement utilisées comme un exercice d'apprentissage d'un sujet en reposant sur l'apprentissage actif active recall testing et la mémorisation par répétition espacée (spaced repetition). Il existe bon nombre d'applications de ce genre <sup>1</sup>. Si vous n'êtes pas familier avec ce genre d'applications nous vous invitons vivement à étudier le fonctionnement des applications existantes et notamment la page de présentation de l'application (Anki) <sup>2</sup> <https://docs.ankiweb.net/background.html#active-recall-testing> : Installez et essayez cette application.

Le mode "création" de l'application devra permettre :

- de créer une carte, d'éditer son contenu (face question, face réponse), de supprimer une carte ;
- de créer une pile (en précisant a minima un nom et une description), de modifier une pile, de supprimer une pile ;
- d'exporter et d'importer une pile de cartes (dans votre propre format de stockage).

Le mode "apprentissage" de l'application devra permettre :

- de travailler l'apprentissage d'une pile de cartes sélectionnée au préalable. L'application devra présenter la question, laisser un temps de réflexion, montrer la réponse, permettre à l'utilisateur de s'auto-évaluer sur la carte présentée ;
- de configurer l'apprentissage en définissant différents paramètres influençant la répétition des cartes (l'algorithme de répétition, les fréquences de répétition, l'ordre de présentation) ;
- de consulter les statistiques d'apprentissage d'une pile ou de l'ensemble des piles. Ces statistiques pourront être présentées sous forme de graphiques.

Il est possible d'envisager un certain nombre d'extensions aux fonctionnalités basiques présentées ci-dessous :

- partager en ligne des piles de cartes (dans un répertoire GoogleDrive, un dépôt DropBox, un dépôt Git, etc.) ;
- intégrer des données multimédia à vos cartes (images, sons, vidéos, etc.) ;
- ajouter différents formats de rendu du contenu de vos cartes (rendu de formules mathématiques, de diagramme plantuml, etc.) ;
- importer les piles de cartes provenant d'une autre application (par exemple le format `.apkg` d'Anki) ;
- proposer une meilleure organisation des cartes et des piles de cartes (en partageant des cartes entre différentes piles, en ajoutant des étiquettes (tags) sur les cartes et les piles, en organisant les piles par leçons) ;
- modifier le style de présentation de cartes ;
- proposer un mode où l'apprenant doit saisir la réponse.
- ...

---

1. [https://en.wikipedia.org/wiki/List\\_of\\_flashcard\\_software](https://en.wikipedia.org/wiki/List_of_flashcard_software)

2. <https://apps.ankiweb.net/>

# Organisation

## Présence et assiduité

La semaine a été bloquée pour vous permettre de vous concentrer sur cette épreuve. Des salles de TD et TP sont réservées spécifiquement pour l'épreuve (salles S0.8, S0.9, S1.12, E1.15, E1.16, E1.17, E1.18, E1.21, E2.41). Si vous vous installez dans une salle non réservée, nous vous prions de bien vouloir libérer celle-ci lorsque des enseignements y sont prévus dans la semaine. Il vous est demandé d'indiquer à votre encadrant(e) dans quelle salle votre groupe se trouve.

## Encadrement

Chaque groupe se verra assigner un enseignant référent. Chaque groupe rencontrera son enseignant à intervalles réguliers. Au cours des journées, plusieurs enseignants seront disponibles pour répondre à vos questions. Ceux-ci seront joignables à travers le forum ou dans une salle dont le numéro vous sera communiqué ultérieurement.

## Travail et collaboration

Suite à l'enregistrement de votre groupe, un dépôt privé Git vous sera créé où vous déposerez le code source et les différentes informations nécessaires à la compilation et à l'exécution de votre application (Java).

Il est nécessaire de bien organiser le contenu de ce dépôt et de "commiter" régulièrement afin que l'ensemble des membres du groupe ait accès aux données les plus récentes. Il sera tenu compte des contributions de chacun des membres du groupe. Si un des membres n'a pas su "commiter" ses contributions, nous considérerons sa participation comme nulle.

Vous devrez :

- documenter la planification et l'avancement du projet ;
- vous assurer des livraisons (releases) quotidiennes et de la bonne utilisation de Git ;
- mettre en œuvre des tests et des procédures assurant la qualité du logiciel ;
- documenter votre architecture et votre conception pour les rendre compréhensibles ;
- documenter l'installation et l'exécution de votre application.

Ces différents points seront pris en compte pour un tiers de la note.



Quand vous utilisez Git faites attention à ne pas publier vos clés d'accès au serveur GitLab <sup>a</sup>. Pour cela, il convient de placer ces clés dans un fichier de configuration et de filtrer ce fichier en utilisant le fichier `.gitignore`. Vous devez vous arranger pour que chacun utilise ses propres clés d'accès aux API. Si possible, prévoyez des clés d'accès pour vos démonstrations, sinon documentez la procédure pour obtenir et fournir ces clés.

---

a. Ars Technica. *PSA : Don't upload your important passwords to GitHub*. Janvier 2013

## Évaluations et rendus

Le développement se fait de façon itérative et incrémentale. Vous devez produire un logiciel opérationnel chaque jour. Nous vous recommandons d'utiliser un outil tel que Gradle pour automatiser la compilation, l'exécution des tests et la construction de votre application. Vous y ajoutez des fonctionnalités en fonction d'une feuille de route (roadmap) que vous aurez définie à l'avance et que vous ferez évoluer en fonction de votre avancée. Il n'est pas question d'avoir plusieurs morceaux de programmes fonctionnant indépendamment les uns des autres. Le programme doit être intégré chaque jour et vous devez à tout moment pouvoir faire une démonstration de la dernière livraison. Le document **README** du dépôt doit indiquer comment exécuter ce programme.

## Rendus intermédiaires

Chaque groupe devra effectuer un rendu journalier sous la forme de commits Git (cela ne signifie pas qu'il faut se limiter à un commit par jour dans le dépôt, au contraire). Ce rendu devra être étiqueté (au moyen de la commande `git tag` dans

la branche master) selon la convention de nommage suivante : **RELEASE\_DAY\_X** où X indique le jour de la semaine. Il s'agira d'une livraison, donc d'un système déployable et opérationnel. Par défaut, la commande **git push** ne partage pas les tags, il faut explicitement partager ces tags (en utilisant l'option **--tags**).



La contribution de chacun des participants d'un groupe pourra être mesurée à son activité sur GitHub, i.e. le nombre de commit qu'il aura pu faire et leur taille. Il ne sera pas admis que certains d'entre vous n'aient pas commité de code à leur nom. Il est donc important de bien configurer votre client Git afin que vos commits vous soient attribués <sup>a</sup>.

a. Vérifiez les variables **user.name** et **user.email** au moyen de la commande **git-config --list**.

## Rendu final

Le rendu final sera étiqueté (toujours la branche master) selon la convention de nommage suivante : **RELEASE\_FINAL**.

Il devra comporter a minima :

- les documents d'analyse et de conception que vous aurez réalisés ;
- le code source de l'application client lourd (Java), les instructions textuelles (texte brut ou fichier PDF) indiquant comment compiler, configurer et exécuter cette application (précisant les dépendances externes et comment avoir accès à ces dépendances).

Les derniers commit auront lieu **avant 18h le vendredi 6 janvier 2023**.



Il ne sera pas possible de faire de démonstration le dernier jour. Vous nous rendrez une démo de 10 mn de votre application sous la forme d'un screencast présentant ses fonctionnalités. Vous pouvez utiliser un outil de capture d'écran tel que OBS Studio <https://obsproject.com/> ou tout autre outil. Le plus simple est de déposer votre vidéo sur une plateforme en ligne telle que YouTube ou Dailymotion et de publier ensuite le lien dans votre fichier **README** de votre dépôt Git (ce n'est pas un problème si le commit publiant ce lien est réalisé après l'heure de rendu limite de votre projet).

Cette vidéo devra être produite et mise en ligne avant le **samedi 7 janvier 2023, 12h**.

## Critères d'évaluation

- Les résultats concrets présentés à la fin de la semaine ;
- le respect des méthodes ;
- l'organisation du travail ;
- la mise en œuvre de tests démontrant le fonctionnement du logiciel ;
- la qualité du développement (principe de conception mis en œuvre, architecture adaptée) ;
- l'invention de nouveaux services et caractère innovant du logiciel.

## Communication avec les encadrants

Afin de poser vos questions et de discuter durant la semaine, nous avons créée sur la plateforme de messagerie Microsoft Teams une équipe avec différents canaux (nous créerons un canal pour chaque équipe également). N'hésitez pas à y poser des questions sur le sujet ou sur des points techniques.



Afin de rejoindre l'équipe dédiée à la codingweek, vous devrez utiliser **ce lien d'invitation** à votre première connexion.

Afin de centraliser les différentes réponses, nous avons mis également en place un document partagé. N'hésitez pas à consulter/éditer ce document régulièrement au cours de la semaine.

[https://docs.google.com/document/d/1Xyk\\_p-roVaKcngHYGKlQwj4KFu8EvRumvHJALwrGv2l/edit](https://docs.google.com/document/d/1Xyk_p-roVaKcngHYGKlQwj4KFu8EvRumvHJALwrGv2l/edit)