

Jules HIRTZ
Yann MIJATOVIC
Théo PINCHON
Alexandre PERROT

Mimir
Tuteur : Yann BONIFACE
Année 2023 - 2024

MIMIR

Étude préalable

Sommaire

Présentation du projet tutoré.....	3
Liste des fonctionnalités du système.....	4
1. Gestion de compte utilisateur.....	4
2. Tableau d'analyse de sa progression.....	4
3. Algorithme de Leitner.....	4
4. Système de partage entre les utilisateurs.....	5
5. Convertir des PDF vers des cartes.....	5
6. Chronomètre Pomodoro.....	6
7. Système de création de cartes.....	6
8. Modération du système.....	7
Cas d'utilisation - Scénarios.....	9
• Cas d'utilisation.....	9
• Scénarios avec leur diagramme d'activité respectif.....	11
Maquette de l'application.....	15
Recensement et évaluation des risques.....	16
1. Utilisation d'un GPT.....	16
2. Extraction de PDF vers texte.....	16
3. Le "vérificateur de réponse".....	17
4. Modération.....	17
5. Évolutivité de notre application.....	18
6. Pas assez de données pour rassembler des statistiques intéressantes.....	18
7. Pas assez d'utilisateurs pour créer des decks et ainsi pas d'entraide dans la communauté.....	18
Planning prévisionnel.....	19
• Itération 1 (38h du 18/12 au 12/01).....	19
• Itération 2 (28h du 15/01 au 19/01).....	19
• Itération 3 (28h du 22/01 au 27/01).....	19
• Itération 4 (44h du 29/01 au 16/02).....	19
• Itération 5 (24h du 19/02 au 08/03).....	19
• Itération 6 (8h le 18/03).....	20
Annexes.....	21
Diagramme d'activité Scénario 1.....	21
Diagramme d'activité Scénario 2.....	22
Diagramme d'activité Scénario 3.....	23
Diagramme d'activité Scénario 4.....	24
Diagramme d'activité Scénario 5.....	25
Diagramme d'activité Scénario 6.....	26
Diagramme d'activité Scénario 7.....	27
Diagramme d'activité Scénario 8.....	28
Diagramme d'état Deck.....	29
Diagramme d'état flashcard.....	30

Présentation du projet tutoré

Le projet que nous avons imaginé est de proposer une sorte de plateforme éducative. L'application se nomme Mimir en référence à la divinité nordique responsable de la sagesse et de la connaissance.

Dans un premier temps l'objectif principal est de créer un système de flashcard (questions / réponses), dans des decks. (*un deck signifie un paquet de cartes*)

Prenons l'exemple d'un étudiant ayant pour but d'aller dans des écoles d'ingénieur à la suite de son BUT à l'IUT Nancy-Charlemagne dans la branche ingénierie logicielle. Afin de réviser et d'apprendre son cours de logique le mieux possible, il aura simplement à se connecter à l'outil et ses révisions seraient organisées automatiquement par le système. Plus besoin de perdre des heures à faire des fiches de révisions, plus de stress pour savoir par quels cours commencer, comment se tester, ou si nos connaissances sont réellement apprises et mémorisées ou si c'est dans notre mémoire instantanée.

Notre application gèrera tous les aspects de l'apprentissage profond et de la compréhension des concepts à travers une sorte de quizz (à l'image d'ANKI, en mieux).

Par ailleurs, si le système venait à être utilisé par plusieurs personnes, on pourrait imaginer un partage de decks (groupe de flashcards) entre étudiants, voire entre professeurs et étudiants. C'est-à-dire qu'un professeur pourrait préparer facilement les cartes liées à la théorie du cours et partager le deck à ses groupes de classe, s'assurant ainsi de la qualité des informations directement retenus par les étudiants. En plus de vouloir créer un outil que nous aurions aimé avoir, nous cherchons à faire d'une pierre deux coups en utilisant le plus possible les technologies les plus efficaces et en vogue dans le domaine de l'ingénierie logicielle web. De plus nous avons déjà pensé à une architecture facilement scalable puisque nous avons l'ambition de construire une solution pratique et efficace pour accompagner les étudiants et les aider au long de leur scolarité et même après.

Liste des fonctionnalités du système

1. Gestion de compte utilisateur

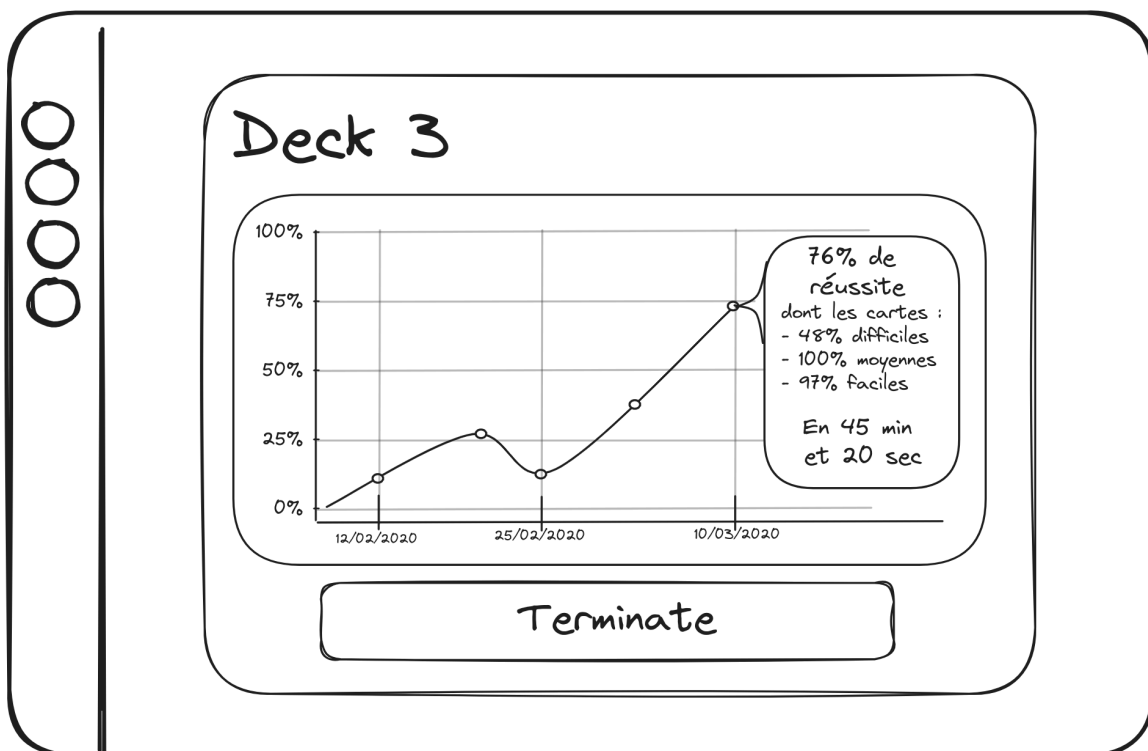
On souhaite pouvoir proposer le site Web qu'aux utilisateurs inscrits afin qu'ils puissent avoir un espace privé pour leurs cartes, de partager leurs créations à d'autres utilisateurs, de participer à des decks éducatifs proposés par leur professeur, ...

2. Tableau d'analyse de sa progression

L'utilisateur doit pouvoir suivre sa progression grâce à un tableau d'analyse qui relèvera le temps passé sur chaque question, les erreurs qu'il a faites, de ce qu'il a le mieux retenu, ...

Il pourra également voir sa progression par rapport aux jours précédents.

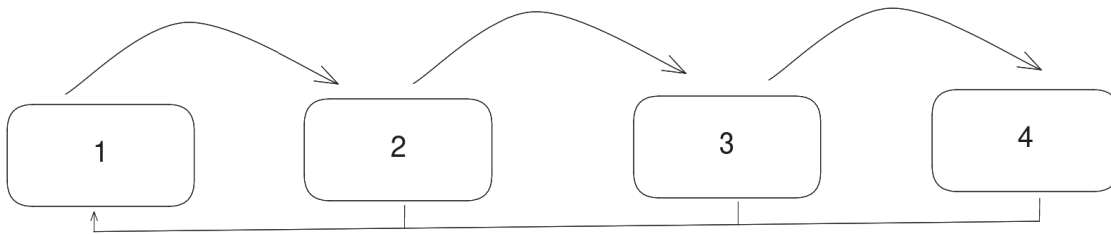
Ces données permettront à l'utilisateur de voir s'il apprend efficacement, ce qu'il doit réviser à nouveau.



3. Algorithme de Leitner

Pour qu'un utilisateur puisse valider un deck sur un cours. Nous allons utiliser le système de Leitner qui consiste à mettre en place des paliers.

A chaque validation de réponse, si la carte est jugée correcte, elle passera au palier supérieur. Dans le cas contraire, elle passe dans un palier inférieur.



Le but ici est de faire apparaître les paliers de bas niveaux plus fréquemment pour faire apprendre les cartes que l'utilisateur maîtrise le moins. Et les palier supérieurs apparaîtront moins souvent car l'utilisateur maîtrise déjà suffisamment ces cartes. Une fois que toutes les cartes atteignent un dernier palier, s'il les valide toutes, le cours est en théorie bien appris. On veut mettre en place un système de révision qui demandera à l'utilisateur de refaire ce deck pour être sûr qu'il n'a rien oublié même des jours après.

4. Système de partage entre les utilisateurs

Ce projet repose aussi principalement sur le partage, et ce pour cela qu'il est important de pouvoir mettre à disposition des moyens de partages :

- Via un URL (il y aura également un système d'édition pour que les élèves puissent modifier le deck)
- Mise en place d'une bibliothèque publique (comme un Play Store) où un utilisateur partage son deck à tous les utilisateurs du site Web.

5. Convertir des PDF vers des cartes

Nous voulons simplifier le processus de création des cartes afin que l'étudiant puisse se concentrer sur la partie apprentissage que de la partie création des cartes en copiant un par un les problèmes dans son cours. Nous allons utiliser une intelligence artificielle afin de faire ce processus, elle aura pour objectif d'extraire les informations correspondant à notre besoin (extraire les problèmes et ses solutions) et les convertir en cartes.

Exercice 1 Soit $\varphi = a \wedge \neg(b \rightarrow (c \vee \neg a))$. Est-ce une tautologie ? Est-elle satisfiable ? Si oui, donnez une interprétation de φ . Si non, montrer qu'elle n'est pas satisfiable en utilisant une table de vérité.



Recto

Soit $\varphi = a \wedge \neg(b \rightarrow (c \vee \neg a))$. Est-ce une tautologie ? Est-elle satisfiable ? Si oui, donnez une interprétation de φ . Si non, montrer qu'elle n'est pas satisfiable en utilisant une table de vérité.

Verso

a	b	c	φ
F	F	F	F
F	F	V	F
F	V	F	V
F	V	V	F
V	F	F	V
V	F	V	V

6. Chronomètre Pomodoro

Une fonctionnalité qui permet de mettre en place une méthode d'apprentissage où l'on effectue 25 minutes de travail et 5 minutes de pause, et cela recommence jusqu'à que l'utilisateur choisisse d'arrêter sa session.

7. Système de création de cartes

Flashcards dans des decks qui peuvent se trouver dans des groupes de decks

Les decks peuvent être créés par n'importe quel utilisateur du site.

Pareil pour les groupes de decks, sauf que les decks contenus à l'intérieur ne seront pas forcément des créations de l'administrateur du groupe. Le deck sera donc supprimé du groupe si l'utilisateur décide de rendre son deck privé ou de le supprimer.

[\[Voir Annexe - Diagramme d'état Deck\]](#)

[\[Voir Annexe - Diagramme d'état Flashcard\]](#)

Les utilisateurs peuvent aussi télécharger un deck afin d'en avoir une copie dont on est sûrs qu'elle ne sera pas supprimée. Les decks téléchargés sont re-postables **UNIQUEMENT** si c'est dans le cadre d'un groupe (pour éviter de télécharger puis ré-uploader le travail de quelqu'un d'autre).

Deux types de decks et groupe de decks :

- **classique** créable par tout le monde et “up-votable” pour rendre certains decks ou groupes de decks pertinents ou non selon les avis utilisateurs.
- **éducatif** avec option proposition de flashcards (pas de decks) par les lecteurs. Les propositions sont validées ou non par le créateur et les propositions sont ajoutées directement au deck en question. Le créateur a accès à une gestion multicouches de l’anonymat des utilisateurs.

Au niveau de l’anonymat des decks/groupes :

- **classique** : une moyenne de réussite pour chaque flashcards et/ou decks sera disponible à la vue de tous. Les résultats sont anonymes à chaque fois.
- **éducatif** : Par défaut, le créateur du deck/groupe a accès aux résultats individuels de chaque personne **ANONYMEMENT** (si tout le monde a consulté un deck par exemple et les statistiques). Les utilisateurs ont le choix ou non de retirer leur anonymat aux yeux du créateur, ce qui peut aider à la pédagogie sans pour autant compromettre l’intégrité de l’utilisateur, qui a le choix de l’anonymat ou non.

Au niveau de l’accessibilité des decks/groupes :

- **classique** : Il est accessible à tout le monde si le créateur a décidé de rendre public son travail ou non.
- **éducatif** : Le deck ne peut pas être publié aux yeux de tous, mais seulement partagé à une poignée d’utilisateurs.

Pour l’ajout des decks dans les decks suivis :

- **classique** : il faut cliquer sur l’option “Suivre le deck” pour que le deck soit suivi
- **éducatif** : le deck est automatiquement ajouté dans les decks suivis
- Les decks suivis sont des copies quasi-similaires au deck original mais diffèrent sur certains points pour pouvoir avoir un suivi individuel de chaque utilisateur suivant le deck ou le groupe (nécessité de duplication afin de rendre possible la “killer feature” de l’application web).
- Si le deck original est modifié, une situation de merge est créée (comme sur GitHub) et l’utilisateur peut choisir d’accepter les ajouts et modifications ou non. Il peut simplement garder le deck qu’il a décidé de suivre au départ.

8. Modération du système

Au niveau de l’administration des decks, il faudrait pouvoir gérer plusieurs cas.

- Il faudrait gérer le fait des erreurs dans les flashcards contenues dans le deck par un système de signalement de flashcard accompagné d'un message envoyé par le lecteur qui serait directement transmis à l'auteur du deck.
- Le cas où le deck est "problématique" dans plusieurs sens du terme. Tout d'abord, si le deck contient des propos injurieux, racistes etc... qui n'ont pas lieu d'être sur le site, un signalement de deck sera disponible et sera directement remonté aux administrateurs et modérateurs, qui pourront supprimer le deck voire l'utilisateur lui-même. Un autre type de signalement de decks sera le cas où le contenu ne correspond pas à ce qui était attendu par le lecteur. Par exemple, si des questions de français se retrouvent sur un deck prévu pour des mathématiques, le signalement sera envoyé au créateur du deck et si celui-ci ne réagit pas au bout d'un certain temps et d'un certain nombre de signalements (pour éviter les signalement abusifs), l'information sera remontée aux administrateurs et modérateurs qui pourront supprimer le deck ou changer le libellé si le deck reste pertinent.

Les rôles disponibles :

- L'utilisateur lambda : Il peut seulement créer des cartes et des groupes de cartes et administrer ces mêmes decks et groupes dont il est créateur ou désigné administrateur par un autre utilisateur lambda créateur.

Pour résumer les rôles au sein même des decks et groupes :

- Le lecteur : il est seulement autorisé à lire et consulter les decks et groupes
- Le "demandeur"/collaborateur : seulement disponible dans les decks éducatifs, c'est un lecteur lambda sauf qu'il peut soumettre des propositions de flashcards au créateur du deck, qui peut accepter ou non d'inclure les propositions dans son deck
- Le créateur/administrateur : C'est le créateur du deck ou du groupe. Le rôle peut aussi se partager selon la volonté du créateur. Il permet de gérer les options du deck et d'ajouter et retirer des cartes / decks.
- Le modérateur : Il a les mêmes fonctionnalités que l'utilisateur sauf qu'il peut en plus envoyer des messages privés aux utilisateurs sans les avoir en ami et supprimer des decks et des groupes.
- L'administrateur : Comme un modérateur mais encore plus privilégié. Administre les modérateurs. Il peut aussi envoyer des messages à tous les utilisateurs ou à un groupe fermé d'utilisateurs (pour des annonces par exemple).

Cas d'utilisation - Scénarios

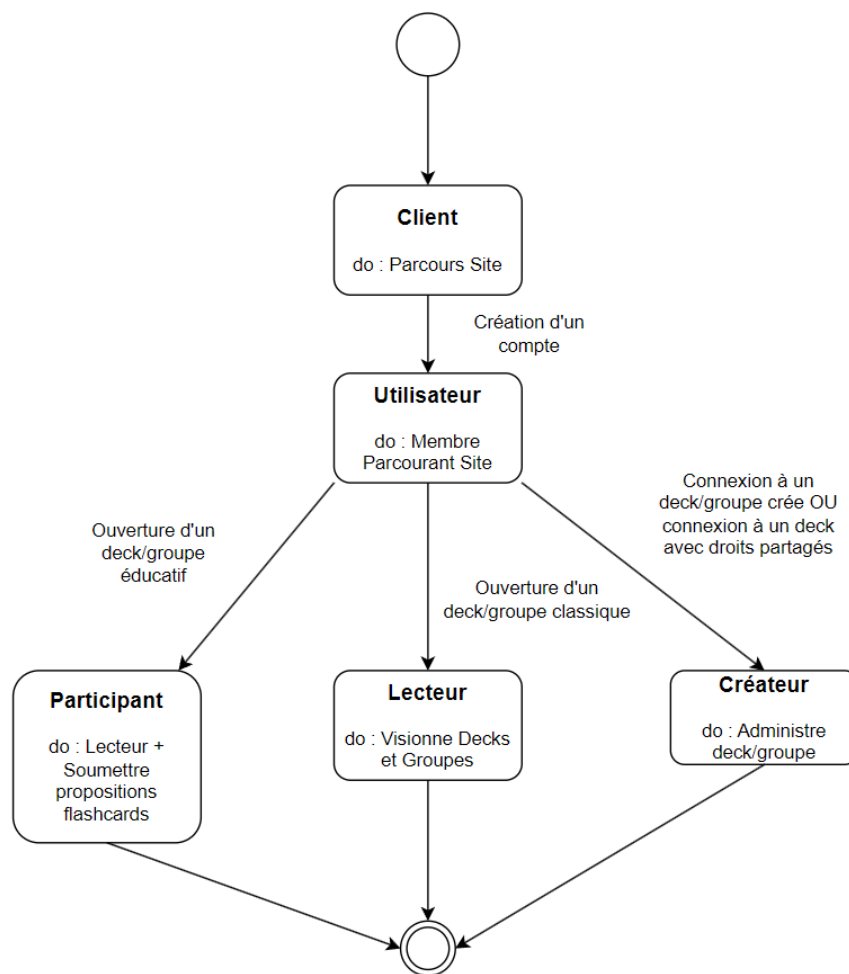
- **Cas d'utilisation**

Le premier rôle est celui d'élève ou dit lecteur. L'utilisateur possède les fonctionnalités classiques de pouvoir répondre au carte ainsi que vérifier ses réponses.

Le second rôle est une spécialisation du premier, appelé demandeur. Lorsque le lecteur soumet une proposition de carte pour un deck, il change de rôle. Ce rôle ne peut être accessible uniquement pour les decks dit "ouvert aux propositions".

Le troisième rôle correspond au professeur ou créateur. Ce dernier possède les droits d'administration sur le deck qu'il crée. C'est-à-dire pouvoir y ajouter ou enlever des cartes, les modifier.





- **Scénarios avec leur diagramme d'activité respectif**

- **Un professeur veut créer et partager un deck à ses étudiants**

1. Le professeur crée le deck en cliquant sur l'option "éducatif"
2. Le professeur ajoute des flashcards jusqu'à avoir un support de travail convenable pour les étudiants
3. Le professeur clique sur l'option "Ajouter des membres" sur le deck
4. Le professeur ajoute ses élèves qu'il possède comme contacts sur Mimir
5. Le professeur recherche les élèves qu'il n'a pas dans sa liste de contacts avec la fonction "Rechercher"
6. Les élèves reçoivent une notification sur le site leur indiquant qu'ils sont invités à rejoindre un groupe
7. Les élèves acceptent d'être invités sur le deck
8. Un choix est proposé par le système aux élèves : voulez-vous participer à ce deck anonymement ?
9. Les élèves voient maintenant le deck dans l'onglet "Decks en cours"
10. Le professeur peut voir les participants ayant acceptés de participer au deck

[\[Voir Annexe - Diagramme d'activité Scénario 1\]](#)**- La classe complète le deck éducatif partagé par un professeur**

1. Les élèves répondent à toutes les questions
2. Le professeur voit sur la vue du deck que tous les participants ont complété le deck
3. Le professeur clique sur l'option "Statistiques du deck" pour y accéder
4. Le professeur voit sur la page principale des statistiques la moyenne des bonnes réponses sur toutes les questions et la moyenne finale générale.
5. Le professeur a accès aux statistiques individuelles de chaque utilisateur
6. Le professeur voit la note finale et la réponse à chaque question pour chaque utilisateur

[\[Voir Annexe - Diagramme d'activité Scénario 2\]](#)**- Un utilisateur veut compléter et suivre un deck "Bases de Scala"**

1. L'utilisateur clique sur l'onglet "Decks"
2. L'utilisateur décide d'utiliser la fonction "Rechercher" de cet onglet pour trouver un deck sur les bases de scala en écrivant "Bases scala"
3. Le site lui affiche les decks les plus pertinents correspondants à sa recherche juste en dessous de la barre de recherche
4. L'utilisateur clique sur le premier résultat qui est le deck le plus populaire
5. L'utilisateur ajoute le deck à ses decks suivis en cliquant sur l'option "Suivre ce deck"
6. L'utilisateur décide de compléter le deck
7. L'utilisateur a accès à ses statistiques à la fin du deck qui lui indique sa note finale et la réponse à chaque question

[\[Voir Annexe - Diagramme d'activité Scénario 3\]](#)**- Un utilisateur veut compléter son deck du jour "Additions"**

1. L'utilisateur clique sur l'onglet "Decks suivis"
2. L'utilisateur clique ensuite sur le deck "Additions" qu'il a ajouté précédemment.
3. Il complète toutes les questions avec 5 fautes
4. L'utilisateur répond une nouvelle fois aux questions où il a échoué précédemment
5. L'utilisateur a accès aux statistiques de suivi de ce deck à la fin de sa complétion quotidiennes, lui indiquant les progrès par rapport au jour précédent et les progrès généraux.
6. Le deck est considéré comme complété pour la journée

[\[Voir Annexe - Diagramme d'activité Scénario 4\]](#)**- Un utilisateur veut créer un deck classique sur le “English preterite”**

1. L'utilisateur clique sur l'option dans la page d'accueil “Créer un nouveau deck”
2. Il définit ensuite un nom pour le deck, ici “English preterite”
3. Il coche ensuite l'option “Deck classique”
4. Il coche ensuite l'option “Public” pour partager son deck au monde entier
5. Il crée ensuite les flashcards contenues dans ce deck une à une
6. Il appuie ensuite sur “Créer”

[\[Voir Annexe - Diagramme d'activité Scénario 5\]](#)**- Un élève veut proposer une flashcard dans un deck éducatif**

1. L'élève clique sur le deck éducatif qui lui a été partagé dans l'onglet “Decks partagés”
2. Sur la page du deck, il clique sur l'option “Proposer une nouvelle carte”
3. Un nouvel onglet où il peut créer une carte s'ouvre
4. L'élève crée donc une nouvelle carte avec une question et une réponse
5. Il clique sur “Proposer cette carte au propriétaire”
6. Il revient sur l'onglet du deck avec une notification “Proposition envoyée !”
7. Le professeur (créateur et gestionnaire du deck) reçoit une notification “Un utilisateur du deck NOM_DECK + ID_DECK a proposé une nouvelle carte”
8. Il clique sur la notification et arrive sur une page où il voit la carte créée par l'élève
9. Il choisit de cliquer sur l'option “Accepter cette nouvelle carte dans le deck”
10. Le deck contient maintenant une carte de plus : celle de l'élève

[\[Voir Annexe - Diagramme d'activité Scénario 6\]](#)**- Un utilisateur veut créer un groupe de deck sur la logique**

1. L'utilisateur clique sur l'option “Créer un groupe de deck”
2. L'utilisateur donne le nom “Logique” pour son groupe et décide de le rendre public.
3. Il valide la création de son groupe en cliquant sur “Créer”
4. Il clique ensuite sur “Ajouter un deck personnel au groupe”
5. Il a le choix d'ajouter un deck parmi tous les decks que l'utilisateur a déjà créés
6. Il choisit son deck “Logique propositionnelle”
7. Il répète l'opération pour ajouter son deck “Logique du premier ordre”
8. Il clique ensuite sur l'option “Rechercher un deck à ajouter”

9. Dans la barre de recherche ouverte en cliquant sur cette option, l'utilisateur recherche les mots-clés "Logique deuxième ordre"
10. Il clique sur le premier résultat (le plus pertinent selon l'algorithme de référencement) pour l'ajouter à son groupe

[\[Voir Annexe - Diagramme d'activité Scénario 7\]](#)

- Un client veut créer un compte sur Mimir

1. Le client recherche "Mimir" dans la barre de recherche de son navigateur
2. Le client clique sur le résultat correspondant à notre application web.
3. Sur le site, il clique sur l'option "S'inscrire"
4. Il entre ensuite les informations nécessaires à la création de son compte
5. Il reçoit ensuite un mail de validation
6. Il se rend sur sa boîte mail et valide son compte en cliquant sur un lien
7. Il clique maintenant sur "Se connecter"
8. Il entre son identifiant ou courriel ainsi que son mot de passe
9. Le client est maintenant devenu un utilisateur étant connecté sur Mimir

[\[Voir Annexe - Diagramme d'activité Scénario 8\]](#)

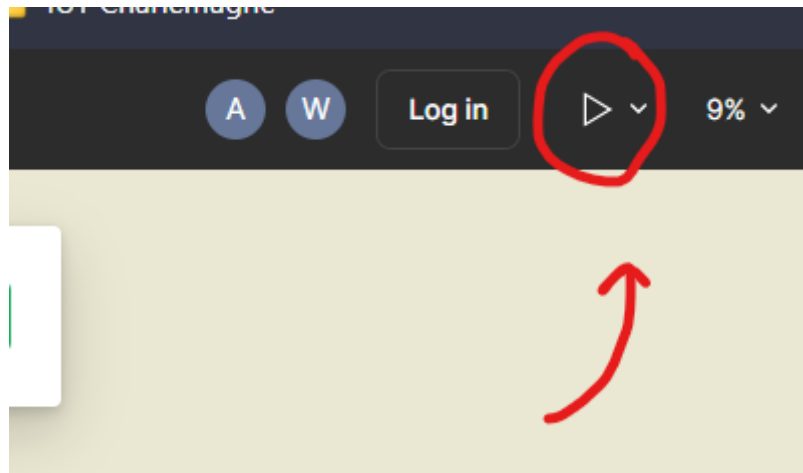
Maquette de l'application

Nous avons réalisé la maquette de l'application via Figma

Voici le lien pour visualiser la maquette :

<https://www.figma.com/file/Wifj3LLdOS5zvpNooBIMam/mimir?type=design&node-id=0-1&mode=design>

Vous pouvez interagir avec la maquette comme si vous étiez un utilisateur du site Web en cliquant sur la flèche en haut à droite.



Recensement et évaluation des risques

On souhaite mettre en place une application Web d'apprentissage en ligne via des cartes.

Je vais recenser ci-dessous les risques que peuvent avoir notre projet (d'un point de vue technique et commercial) et les évaluer du plus risqué au moins risqué :

1. Utilisation d'un GPT

Ci-dessous, on discutera de l'extraction du PDF vers du texte. Après avoir extrait le PDF vers du texte, il faut maintenant traiter le texte pour trouver les phrases **intéressantes** (*définir intéressante dans ce contexte, on conserve que les phrases interrogative qui demandent à l'utilisateur de réfléchir, ou bien on garde les phrases qui ont le plus d'intérêt à être apprises par les utilisateurs*) et d'en générer sa réponse. Car une carte possède une face recto (pour la question) et verso (pour la réponse), mais dans la majorité des cas, les PDF n'incluent pas de réponses avec les questions et donc pour palier à ce problème, on souhaite utiliser une I.A générative pour faire cela.

Le gros problème ici, c'est juste le fait d'utiliser une IA générative pour répondre aux questions, comme vous le savez, l'IA n'a pas conscience de ce qu'elle raconte, parfois elle génère la bonne réponse à notre question, parfois elle est complètement à côté de la plaque, et on veut éviter cette situation.

L'autre gros problème, c'est comment utiliser une I.A générative ? Est-ce qu'on passe par l'API de OpenAI (payante et peut être très coûteuse, impossible pour une plateforme gratuite). On envisage d'utiliser un GPT local et gratuit, mais dont le modèle est moins entraîné que celui de OpenAI et par conséquent, sera encore moins précis.

En théorie, on souhaiterait générer les réponses avec les questions de chaque carte, idéalement avec une vérification par l'utilisateur après pour s'assurer de la qualité de la réponse.

2. Extraction de PDF vers texte

Dans notre application, on souhaite faciliter la création des cartes à l'utilisateur en proposant une fonctionnalité de **conversion d'un PDF vers un deck** directement

On peut imaginer un cours de Géographie, au lieu d'écrire les pays à la main un par un, on peut utiliser le support du cours (un PDF), et l'application extrait les informations du PDF pour en créer un deck.

Cette fonctionnalité s'avère très intéressante car elle permet d'accélérer le **processus** de création et ainsi, au lieu de se concentrer sur la mise en place des cartes, on passe plus de temps à s'entraîner dessus.

Mais, pour ce faire, il faut implémenter une fonctionnalité **d'extraction d'informations à partir d'un PDF**. Après une étude de marché, il y a plusieurs librairies open-source qui nous permettent d'extraire un PDF vers du texte directement, ce qui nous permet de travailler directement sur du texte.

Le problème c'est que les librairies ne peuvent pas **extraire à 100%** un PDF, en effet, ils indiquent tous un pourcentage de fiabilité d'extraction (qui tourne aux alentours de 95% pour les meilleurs d'entre elles), cela signifie que dans le processus, on perdra des informations

Ce n'est pas tout, on ne pourra pas extraire les informations **sous forme d'image** (cela implique de la reconnaissance d'image et de les interpréter vers du texte en langage naturel) ce qui sort de nos compétences.

Il y a aussi une forte probabilité que l'extracteur **dispose incorrectement** les informations, à cause de la nature des PDF, et ça peut aussi mal reconnaître les caractères.

3. Le "vérificateur de réponse"

Un point qui nous semble important dans ce projet, c'est de pouvoir évaluer les réponses correctement. En effet, lorsqu'un utilisateur répond à une carte, on doit pouvoir évaluer sa réponse (et non en utilisant un simple "equals()") car, en posant une question telle que 2+2, on s'attend à que l'utilisateur réponde 4 en chiffre, mais il existe des cas (bien qu'improbable), que l'utilisateur, réponde quatre en toute lettre (il y a d'autres exemples plus pertinent que celui-ci)

Donc le risque ici est que la réponse soit jugée incorrecte, alors qu'en réalité, elle l'est.

(on peut effectivement implémenter un système de règle, où on dit à l'utilisateur qu'on s'attend à une réponse en chiffre par exemple)

4. Modération

On souhaite pouvoir détecter le contenu inapproprié dans des decks, même si on "embauche" des modérateurs afin de vérifier les decks pour les valider et que le reste de la communauté puisse l'utiliser.

Le problème qui suit, c'est la rapidité de la modération, les modérateurs restent humains et ne peuvent pas tout vérifier d'un coup très rapidement (surtout si l'application prend de l'ampleur)

On pourrait utiliser un système qui détectera automatiquement un contenu inapproprié (mais reste à voir comment implémenter un tel système)

5. Évolutivité de notre application

Comme dit auparavant, on a conceptualisé un schéma afin de pouvoir stocker les informations pertinente et efficacement de notre application, mais vu la nature de notre application aussi, il y a une crainte de ne pas réussir à stocker tout ce qu'on veut exactement, car il y a plusieurs possibilités de créer une carte (question par image, question en répondant avec du code, etc...)

C'est une question de savoir si on peut réussir à adapter l'application aux besoins futurs sans qu'on ait besoin de tout refaire de zéro.

6. Pas assez de données pour rassembler des statistiques intéressantes

C'est une crainte qui concerne les statistiques pour suivre la progression de l'utilisateur, bien qu'on ait réfléchi à comment implémenter ça dans notre application et sur notre base de donnée, on souhaite stocker chaque action de l'utilisateur lorsqu'il répond aux cartes (bonne réponse ? temps passé dessus, ...) afin d'avoir un tableau d'analyse détaillé et de pouvoir suggérer des améliorations à l'utilisateur.

On souhaite aussi croiser les données afin qu'elles soient pertinentes aussi.

7. Pas assez d'utilisateurs pour créer des decks et ainsi pas d'entraide dans la communauté

Ici, c'est un point qui concerne le partage de decks entre les utilisateurs, afin que notre application soit intéressante, il faut qu'il y ait des decks prédéfini (créé par nous-même pour une bonne base) et que les utilisateurs viennent dans l'application et créent à leur tour leur deck, et le partage au marketplace, et ainsi que d'autres utilisateurs soient intéressés et vont à leur tour créer un deck etc...

Planning prévisionnel

- **Preliminaires**

Apprentissage :

- TypeScript
- NodeJS
- Tailwind

- **Itération 1 (38h du 18/12 au 12/01)**

Création d'une application morte. (Jules / Théo)

Mise en place de la base de données mongo. (Jules / Théo)

Preuve de concept de l'IA : 2 membres (Yann / Alexandre)

- **Itération 2 (28h du 15/01 au 19/01)**

Classes métier + Création de l'API et implémentation de la gestion des cartes (Théo / Jules)

Mise en place du système d'authentification (Yann)

Création d'une charte graphique et redesign des maquettes (Théo / Jules)

Gestion des contacts entre utilisateurs (Yann)

Système de partage des decks URL (Alexandre)

- **Itération 3 (28h du 22/01 au 27/01)**

Finaliser le système de cartes (upvote, modération, ...) (Yann)

Extracteur de PDF vers cartes (Si PoC validée) (Alexandre)

Intégration de l'éditeur de texte complexe pour les cartes.(Jules)

Implémentation des méthodes liées à l'analyse statistique dans l'API. (Théo)

- **Itération 4 (44h du 29/01 au 16/02)**

Tableau de bord d'analyse statistique. (Jules / Théo)

Finaliser les fonctionnalités importantes (Toute l'équipe)

Tester l'application de fond en comble (unit tests, end test) et régler les bugs (Alexandre / Yann)

Ajouter les fonctionnalités optionnelles si possible (Toute l'équipe)

- Question par image
- live-code

Préparer soutenance fin de semestre (Toute l'équipe)

- **Itération 5 (24h du 19/02 au 08/03)**

Finaliser fonctionnalités optionnelles (Yann / Jules)

Continuer à tester l'application et régler les bugs (Alexandre)

Peaufiner l'application (Théo)

- **Itération 6 (8h le 18/03)**

Préparer la soutenance finale (Toute l'équipe)

Annexes

Diagramme d'activité Scénario 1

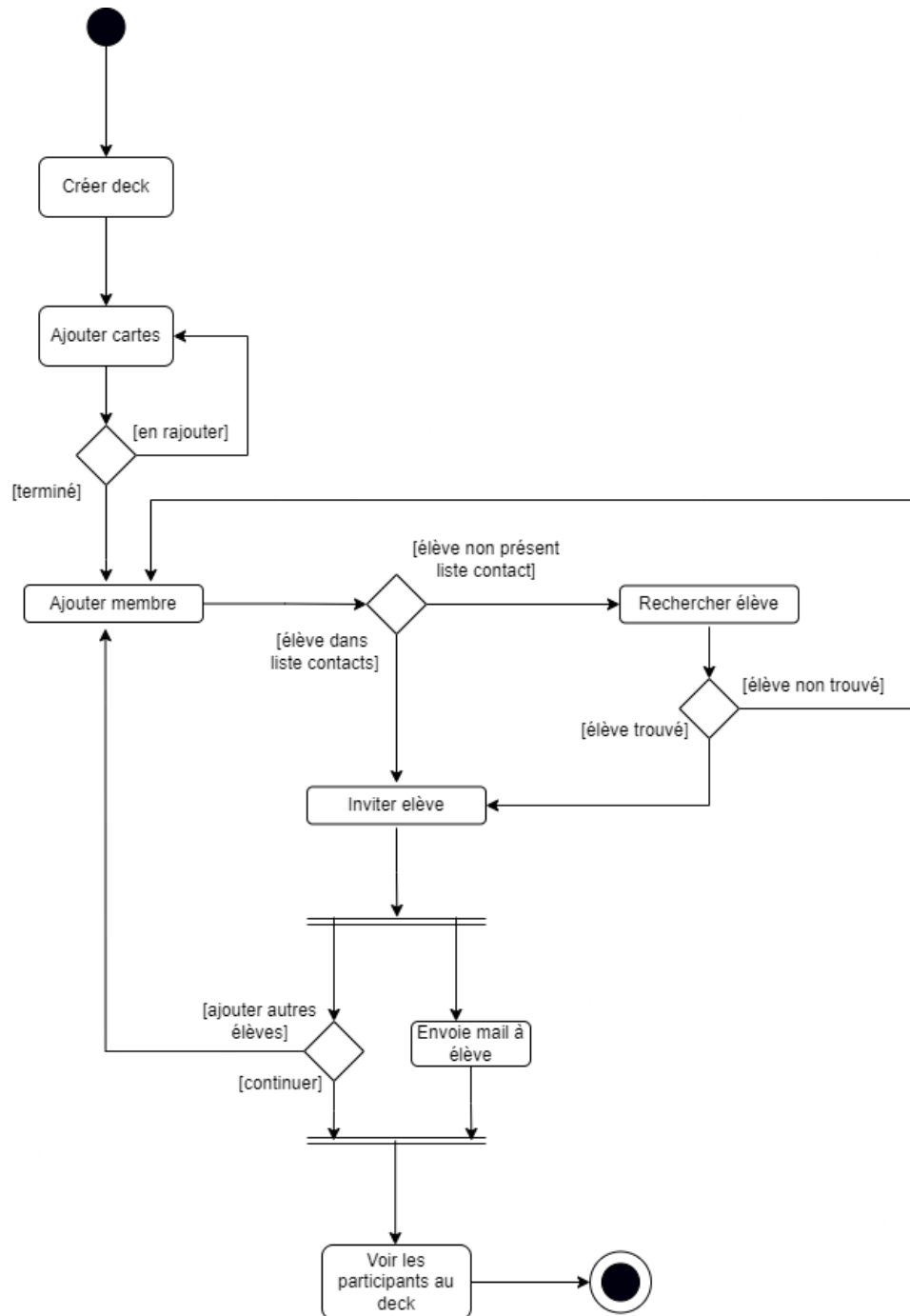


Diagramme d'activité Scénario 2

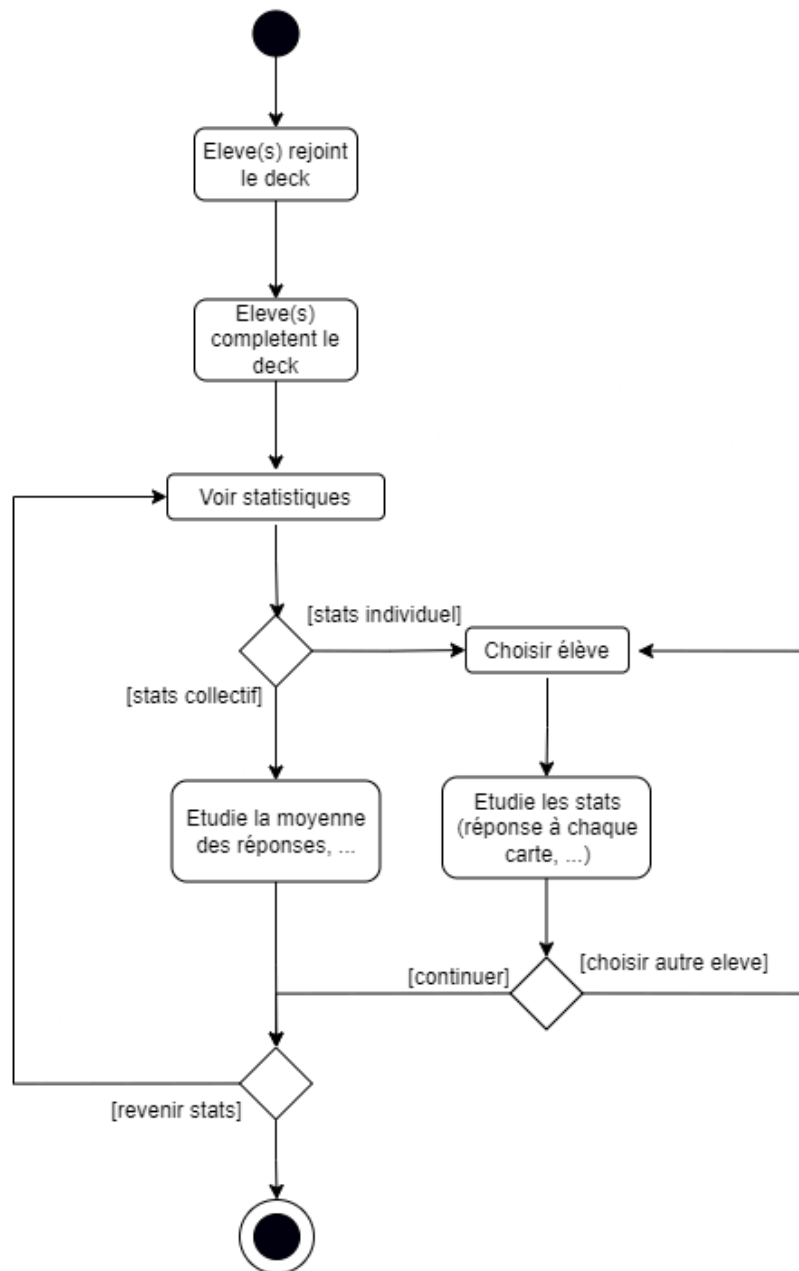


Diagramme d'activité Scénario 3

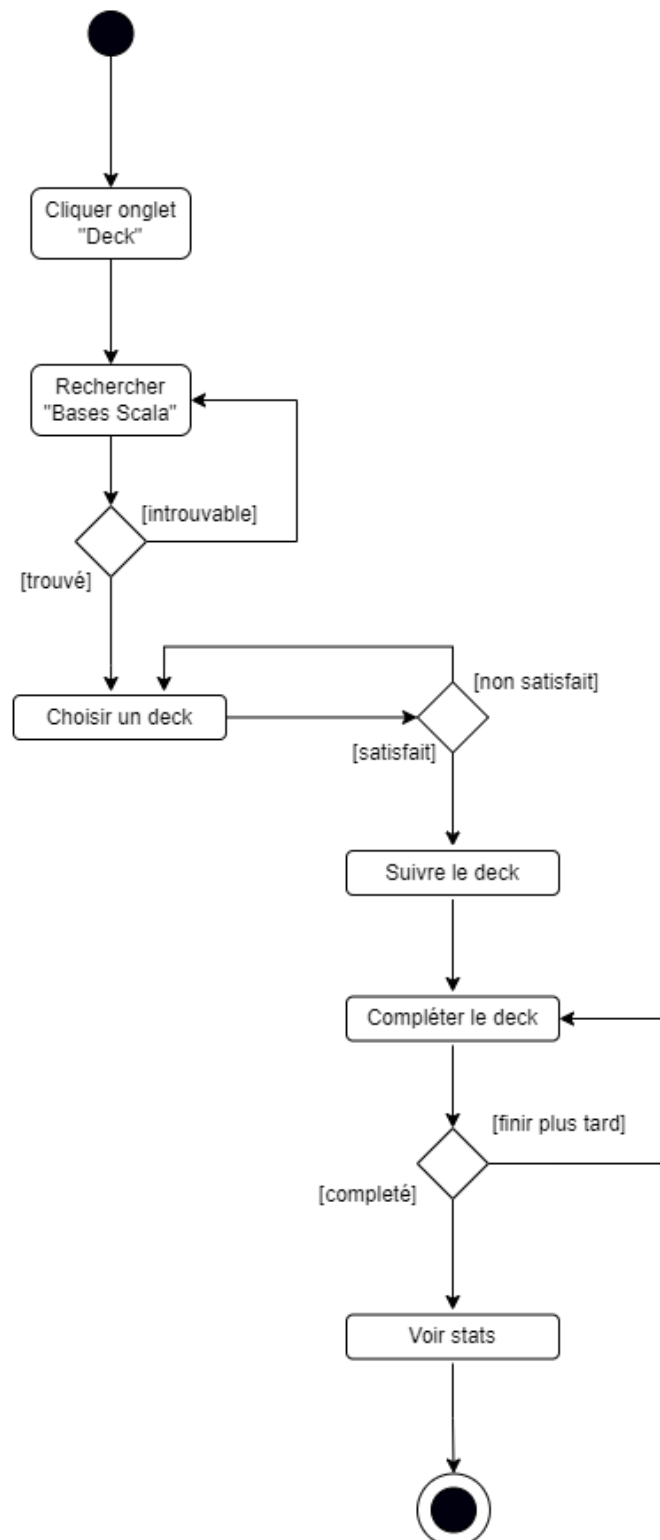


Diagramme d'activité Scénario 4

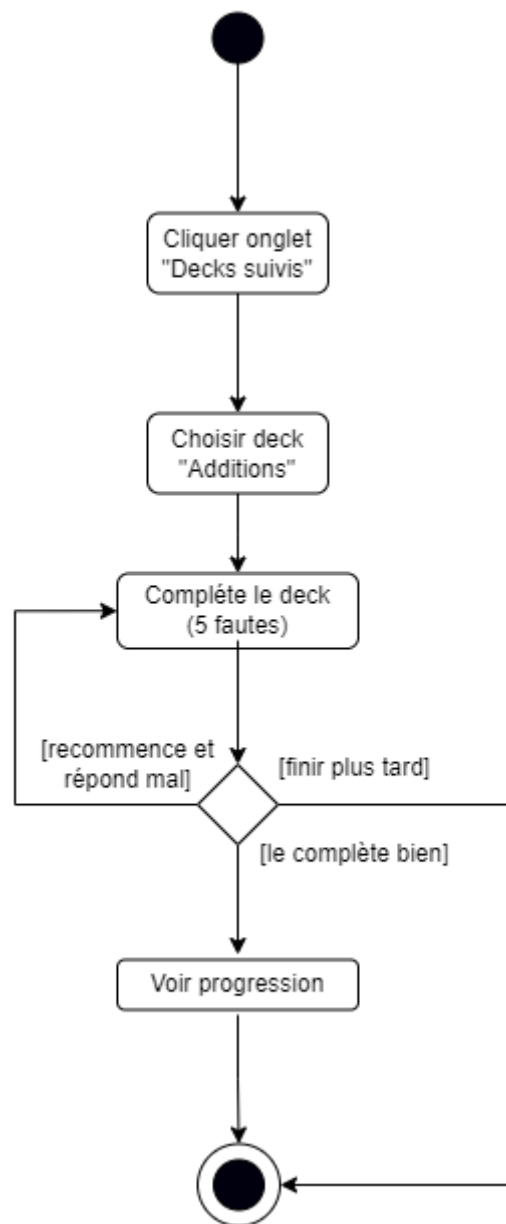


Diagramme d'activité Scénario 5

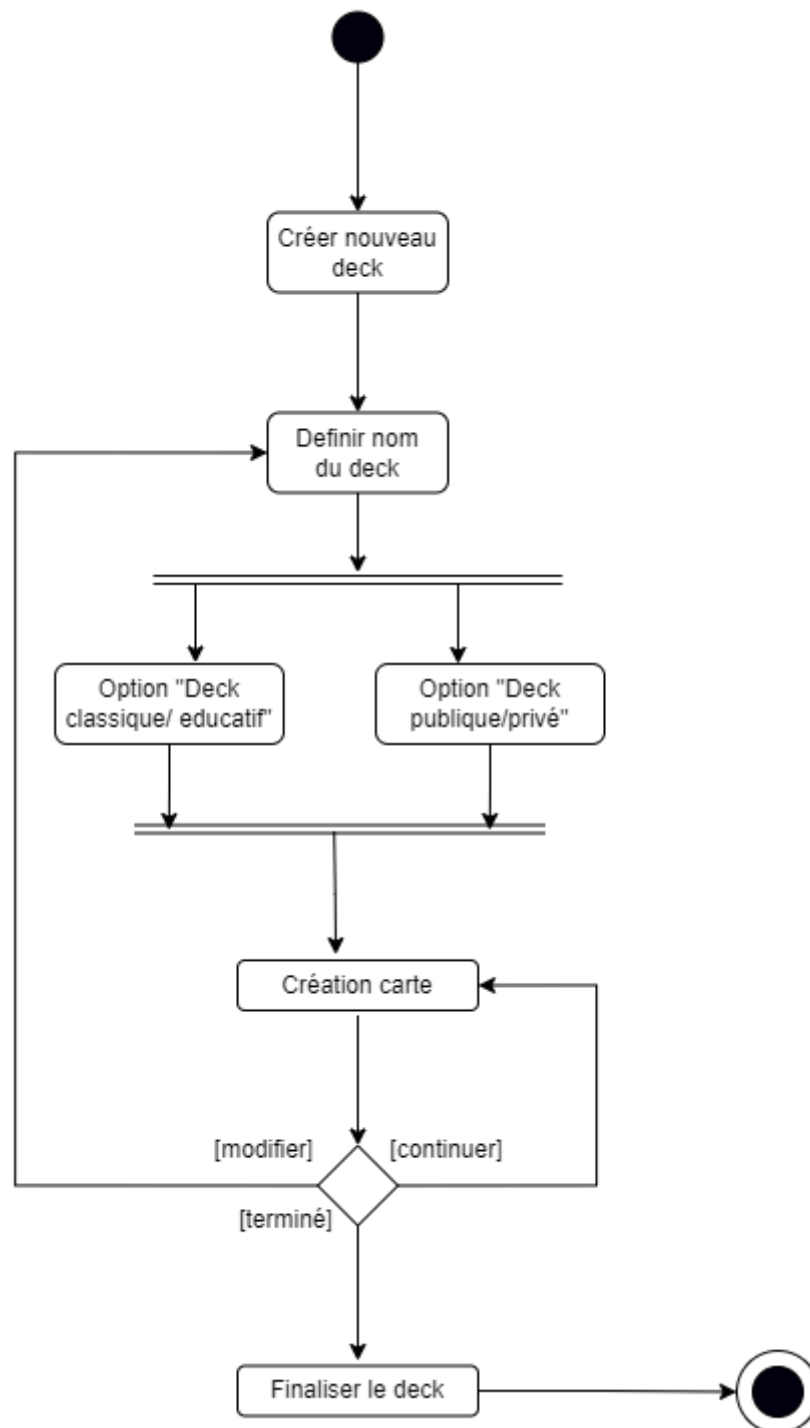


Diagramme d'activité Scénario 6

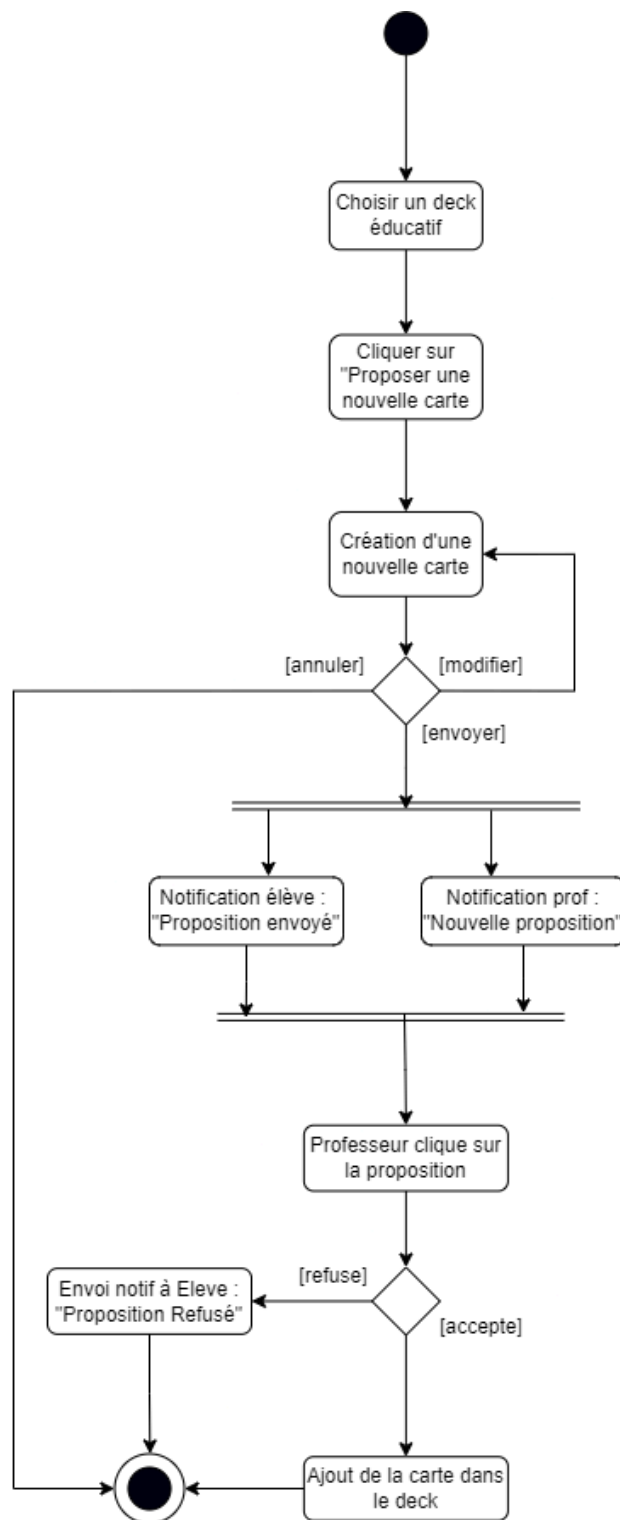


Diagramme d'activité Scénario 7

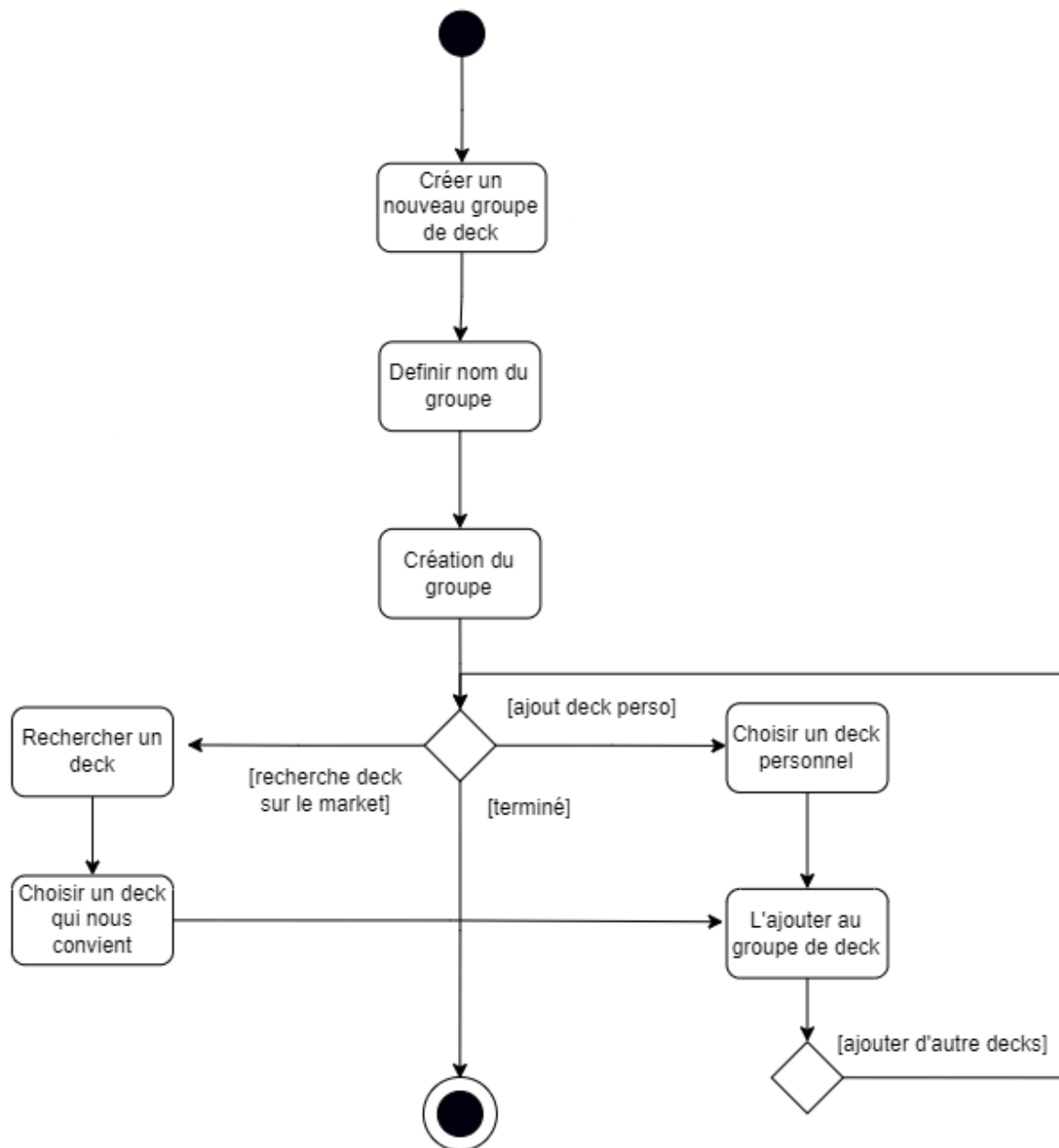


Diagramme d'activité Scénario 8

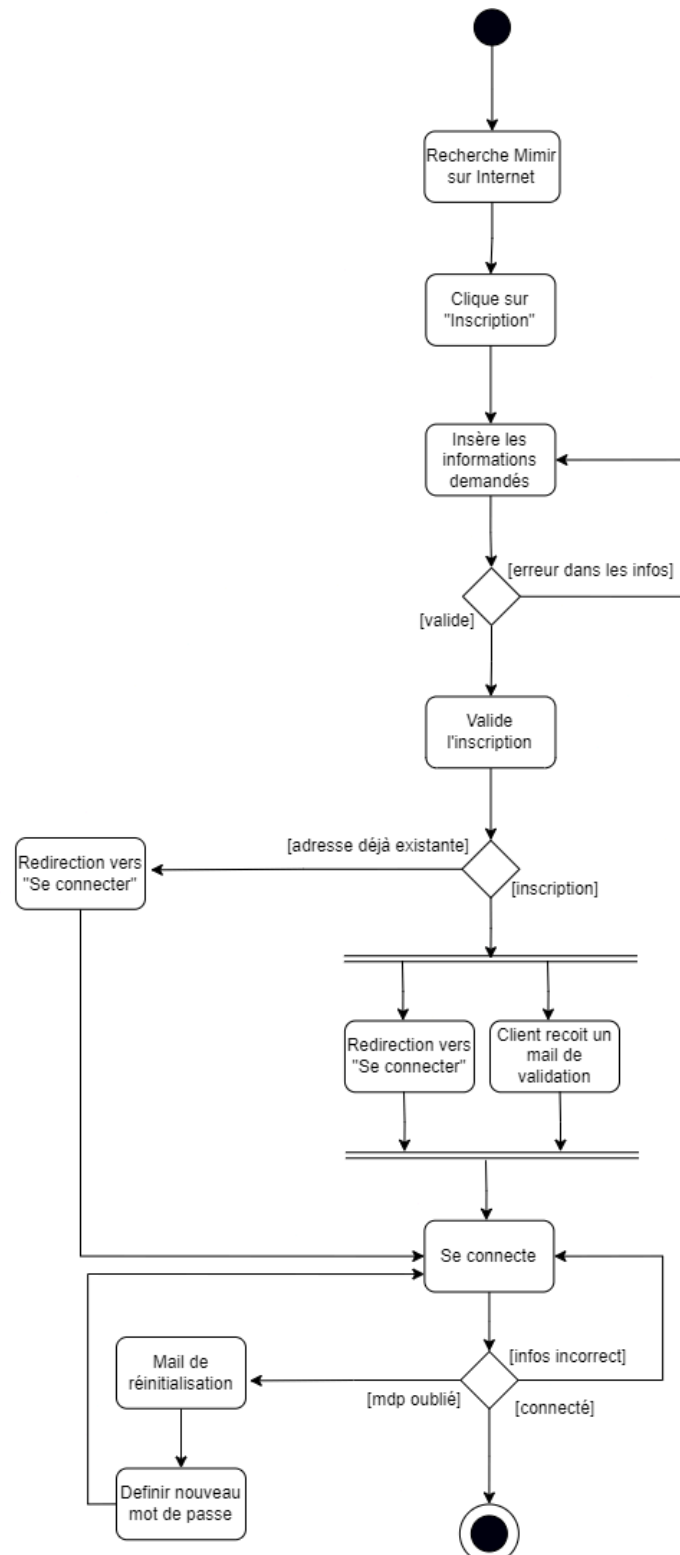


Diagramme d'état Deck

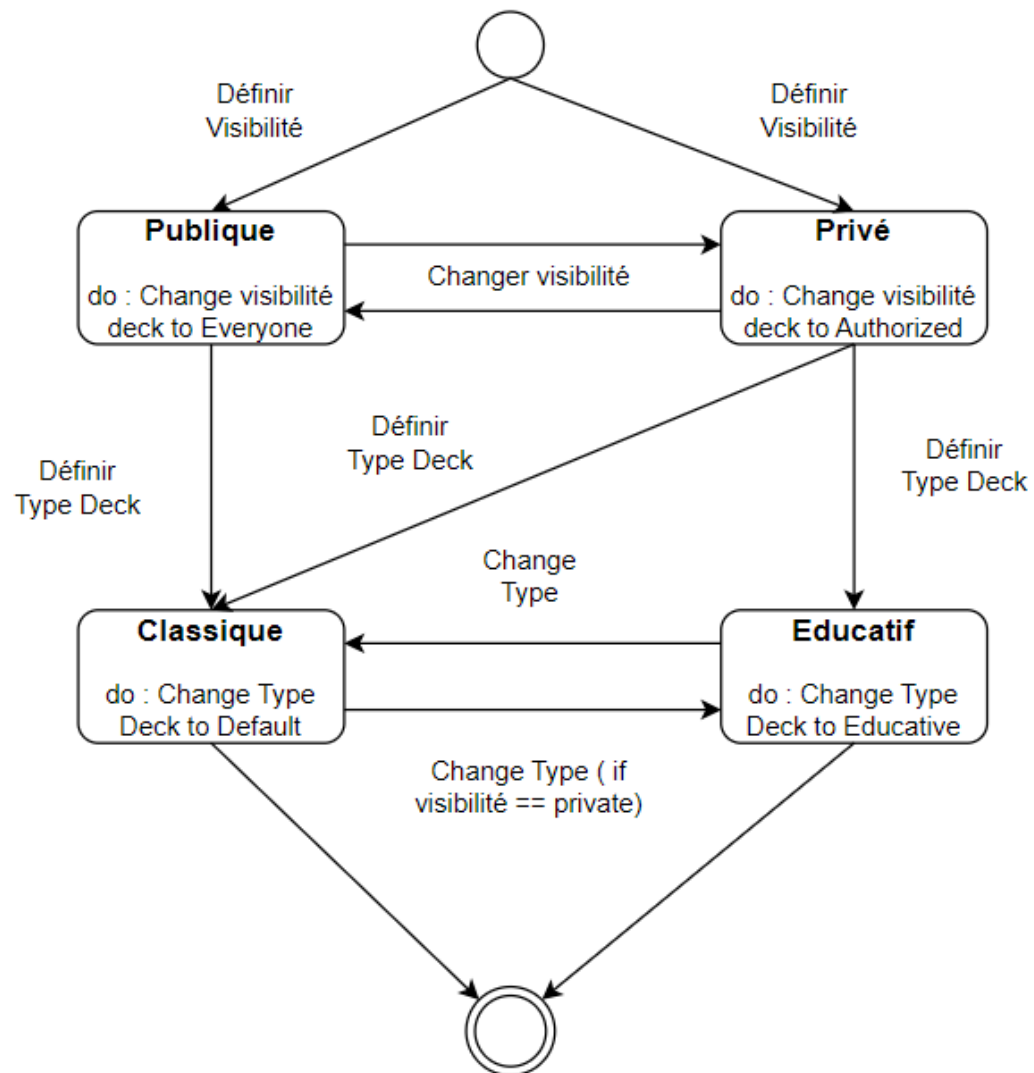


Diagramme d'état flashcard

