

TEAM JT : DWARF

Groupe 7



MARTIN Joris
JHISTARRY Thomas
INZOUDDINE Ayache

REMY Emeryck
SIGNORET Tanguy
PAPPALARDO Mattéo

Storyboard	2
Cadrage du projet	4
Objet du projet	4
L'objet du projet	4
Nom et logo du projet	4
Parties prenantes	4
Maîtrise d'ouvrage	4
Maîtrise d'oeuvre	4
Objectifs du projet	5
Objectifs principaux	5
Objectifs secondaires	5
Analyse de l'existant	6
Les contraintes	10
Les risques	11
Expression du besoin	13
Ciblage	13
Les besoins	13
Besoins fonctionnels	14
Besoins non-fonctionnels	14
Besoins techniques	14
Les critères de qualité	14
Qualité logicielle	14
Qualité ergonomique	15
Critère de test	16
Solutions techniques envisagées	17
Langages et utilisation	17
Architecture technique	17
Architecture de notre application	17
Architecture du serveur	18
Organisation du travail et découpage du projet	18
Mode de fonctionnement	18
Outils	20
Secondes version des maquettes non-fonctionnelles	21
Navigateur / tablette	21
Smartphone	24
Annexes	27

1. Storyboard

Commençons par vous présenter de manière intuitive le principe même de notre application à travers ce storyboard.



Figure 0 : Storyboard page 1

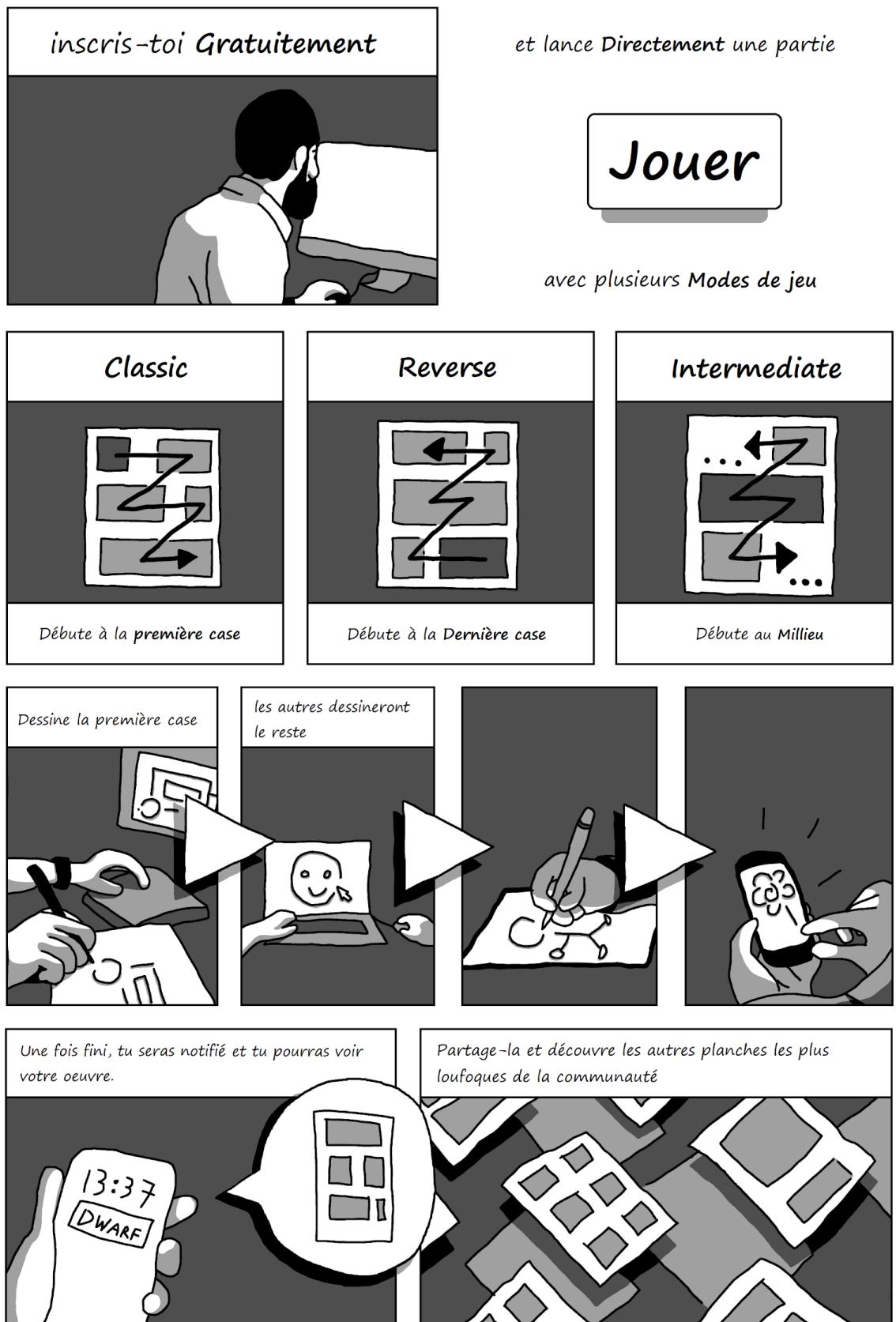


Figure 0 bis : Storyboard page 2

2. Cadrage du projet

a. Objet du projet

I. L'objet du projet

L'objectif du projet est de réaliser une application web pour concevoir une planche de bande dessinée en collaboration. Chaque utilisateur devra réaliser une case d'une planche, en ayant connaissance uniquement de la précédente ou suivante suivant le mode jeu sélectionné. Le projet s'inspire donc du concept du jeu intitulé "cadavre exquis".

Le produit sera gratuit et accessible à tous via un navigateur web ou une application mobile avec une connexion internet.

II. Nom et logo du projet

Le projet se nomme DWARF. C'est un acronyme pour "**D**raw **W**ith **A**mazing **R**andom **F**riends". Le logo (première esquisse ci dessous) représente donc un nain portant un crayon à la place de l'iconique pioche, agrémenté du slogan "Tous à la mine !!!" (jeu de mot avec mine de crayon) (*cf. Figure 1*).



Figure 1 : Logo de DWARF

b. Parties prenantes

I. Maîtrise d'ouvrage

Par le sujet de ce module, les professeurs représentent la maîtrise d'ouvrage, c'est à eux que nous faisons les revues d'avancement et à qui nous livrerons le produit au final.

II. Maîtrise d'oeuvre

La maîtrise d'oeuvre de projet DWARF est notre équipe développeurs et concepteurs, composée de :

- **Ayache INZOUDDINE** : rédaction dossier, base de données
- **Thomas JHISTARRY** : base de données, développement back-end
- **Joris MARTIN** : rédaction dossier, conception, développement full-stack
- **Mattéo PAPPALARDO** : rédaction dossier, développement full-stack
- **Emeryck REMY** : design, ergonomie, conception

- **Tanguy SIGNORET** : chef de projet, rédaction dossier, conception, développement front-end

c. Objectifs du projet

I. Objectifs principaux

Nos objectifs premiers sont de réaliser une application fonctionnelle sur tous types de support : les navigateurs web, les tablettes et les smartphones. Cela implique donc :

- La réalisation d'une case de la planche à l'aide d'un environnement de dessin simple, efficace et ergonomique.
- La conception d'une planche de BD via une interface de création de partie.
- La visualisation et la notation des planches réalisées.
- La recherche d'une planche spécifique ou non afin de la continuer à son tour.
- L'inscription et la connexion à l'application.

II. Objectifs secondaires

Dans un second temps, si tous les objectifs principaux sont accomplis, nous envisageons d'ajouter certaines options à notre application, notamment :

- L'ajout, à l'interface de dessin, d'un outil permettant l'incorporation d'éléments de dessins pré-existants (Exemples : Personnage, gommettes, ...).
- La possibilité d'avoir plusieurs modes de visualisation des planches comme :
 - Un visionnage classique (statique).
 - Un visionnage des planches sous forme d'un livre où l'on tourne les pages.
 - Un visionnage de l'histoire imaginée par les utilisateurs sous forme de flipbook.

d. Analyse de l'existant

Après des recherches approfondies sur l'environnement de notre projet (la bande dessinée), nous n'avons trouvé aucune application/site web correspondant exactement à notre idée.

Il existe cependant plusieurs applications de dessin synchrone, ainsi que plusieurs applications assistant à la création de bandes dessinées. Le tableau 1 ci-dessous présente trois de ces applications :

Nom	Description	Points intéressants	Critiques
	<p>Application mobile, petit jeu de cadavre exquis en multijoueur synchrone : une personne dessine la tête, une autre le corps d'un animal / personnage imposé, le tout en 30s chacun.</p> <p>Le résultat s'affiche et les joueurs peuvent réagir à leur création.</p> <p>Une galerie est présente sur la page d'accueil, montrant les dessins récents.</p> <p>(cf. Figure 2)</p>	<p>Rapide, ergonomique, simple.</p> <p>Réactions au dessins produit à la fin (emojis).</p>	<p>Le jeu est inutilement synchrone.</p> <p>Le concept est vite "essoufflé", le personnage à dessiner n'est découpé qu'en 2 parties (tête et corps).</p> <p>Les personnages / animaux à dessiner sont tirés d'une liste fournie par l'application, c'est-à-dire que le jeu peut vite tourner en rond et redonner les mêmes sujets.</p>
	<p>Site web, jeu de dessin en multijoueur synchrone. Une personne dessine une chose qu'il a choisi parmi une liste de propositions et les autres joueurs doivent deviner ce qu'il dessine en tapant le mot correspondant.</p> <p>(cf. Figure 3)</p>	<p>Amusant, rapide, ergonomique.</p> <p>Plusieurs langues sont disponibles pour les mots à (faire) deviner.</p> <p>Petits avatars amusants.</p>	<p>Les œuvres réalisées sont éphémères.</p>

	<p>Site web permettant de créer des bandes dessinées à partir d'une banque d'images comportant :</p> <ul style="list-style-type: none"> • des personnages personnalisables • des arrière-plans de cases • des bulles de texte • des formes, des pictogrammes • des images importées par l'utilisateur <p><i>(cf. Figure 4)</i></p>	<p>Interface très ergonomique.</p> <p>Il est très facile d'arriver à un résultat convaincant.</p> <p>Aucune compétence en dessin n'est nécessaire.</p> <p>Disposition des cases modifiable à la volée.</p>	<p>Impossible d'ajouter un dessin fait main sur le site, il faut importer une image depuis son ordinateur.</p> <p>Pas d'espace de partage des storyboards réalisés.</p>
---	---	--	---

Tableau 1

Il existe aussi plusieurs applications permettant de tout simplement dessiner. En voici un petit exemple avec une courte présentation de Gimp (le logiciel avec lequel nous avons notamment réalisé notre storyboard) :

Nom	Description	Points intéressants	Critiques
	<p>Logiciel Open Source permettant la réalisation de composants graphiques complexes grâce à de nombreux outils de dessin.</p> <p><i>(cf. Figure 5)</i></p>	<p>Le logiciel est très complet.</p> <p>Permet la réalisation de dessins de grande qualité.</p>	<p>Difficile à prendre en main pour un débutant, certaines manipulations pourtant simples ne sont pas intuitives.</p>

Tableau 1 bis

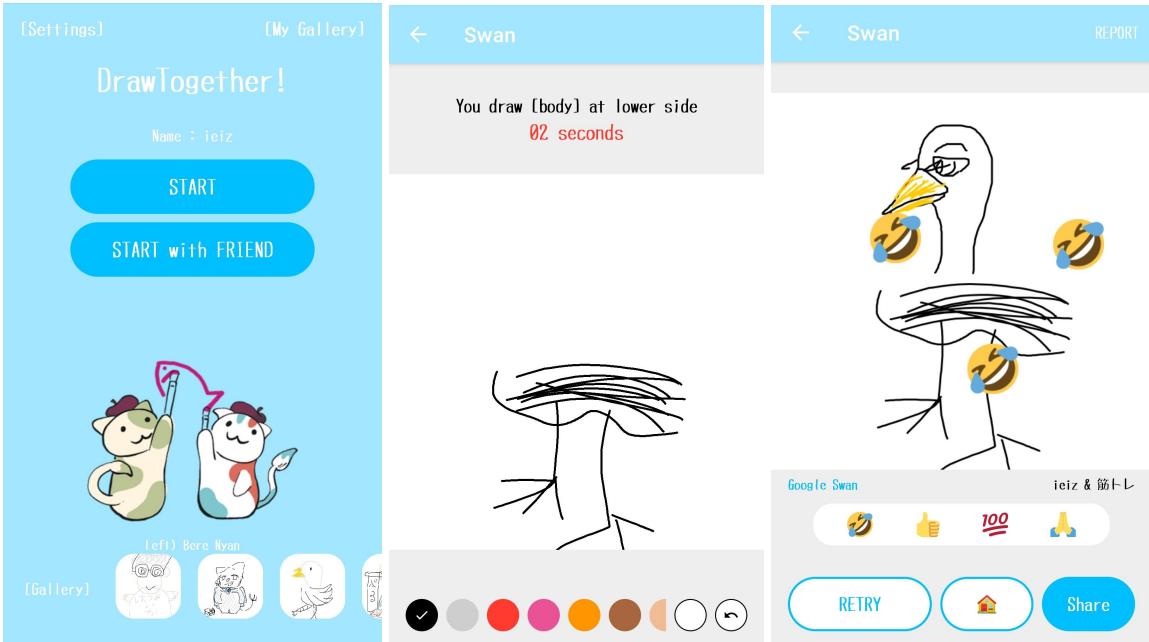


Figure 2 : Trois captures d'écrans de l'application “DrawTogether!”. La première présente le menu principal de l'application. La deuxième présente l'interface de jeu et de dessin, où le but ici était de dessiner le corps d'un cygne. La troisième présente l'interface de fin, où le dessin final est présenté aux deux joueurs.



Figure 3 : Une capture d'écran présentant l'interface de jeu de Skribbl.io. L'avatar ayant un crayon à côté de son nom est celui qui est actuellement en train de dessiner. Les autres joueurs doivent taper le mot correspondant au dessin dans la zone de texte en bas à droite.

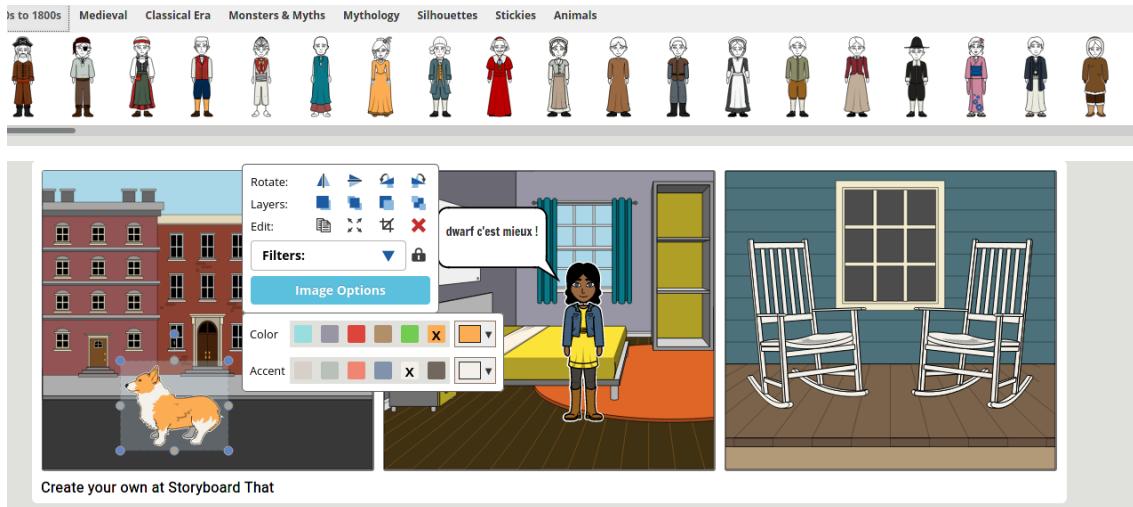


Figure 4 : Une capture d'écran de l'interface de création de bandes dessinées de "StoryboardThat". Les différentes images, bulles, environnements, etc. peuvent être choisies en haut de l'application, et la bande dessinée elle-même ainsi que les options de personnalisations sont en bas.

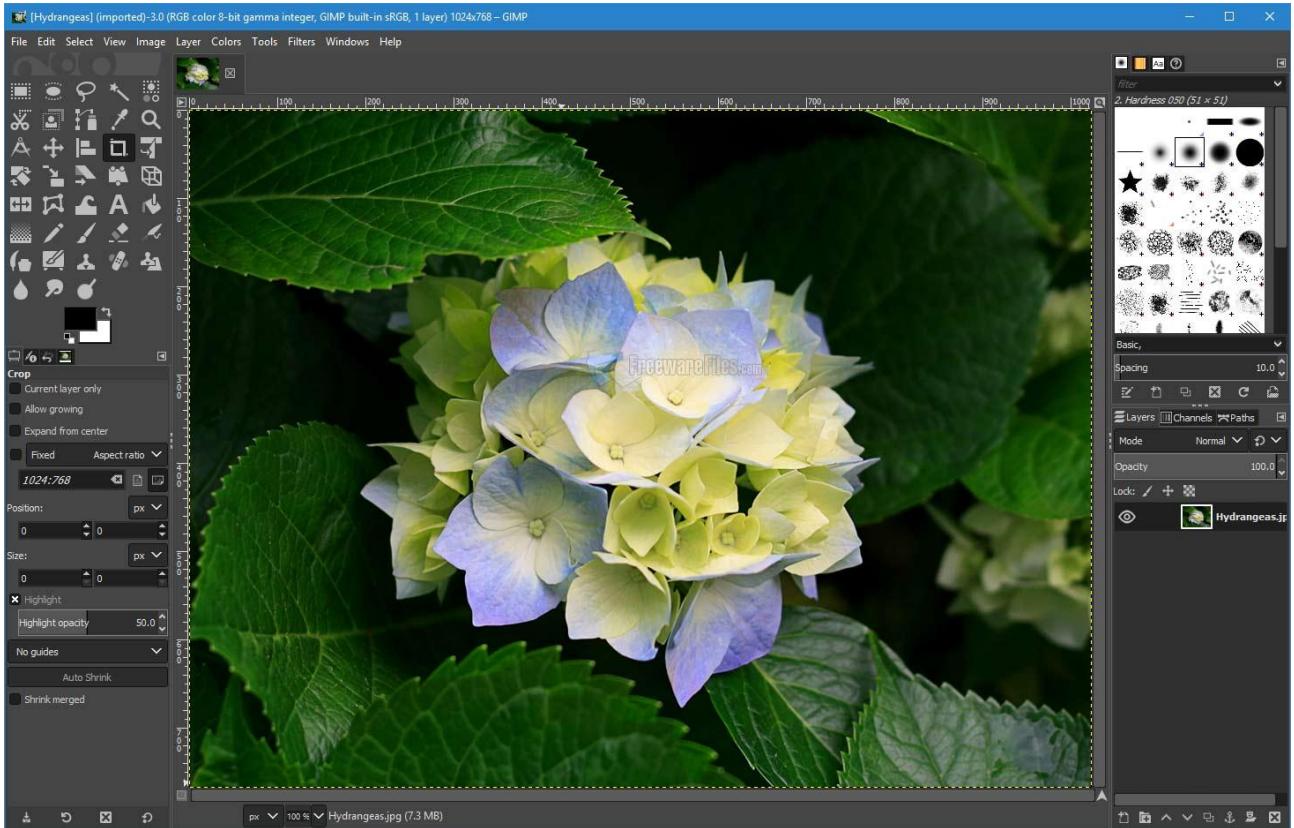


Figure 5 : Voici l'interface du logiciel GIMP. On retrouve un espace de travail central. Et sur les côtés, divers outils de dessins ou de correction d'image.

e. Les contraintes

Durant l'élaboration de cette première partie du projet, il nous est apparu plusieurs contraintes.

Tout d'abord, les règles concernant **l'utilisation des données des utilisateurs**, leur stockage et leur transfert doivent s'effectuer de manière sécurisée. De plus, il nous a semblé important d'être vigilant concernant la publication des œuvres collaboratives réalisées dans le cadre de notre application. C'est pourquoi nous avons fait le choix de placer chaque image réalisée sous une licence Creative Commons qui autorise des personnes tiers à utiliser ces derniers. Il existe différents modèles de cette licence. Nous avons retenu le modèle **CC-BY-NC** qui autorise le partage et la modification mais pas l'utilisation à des fins commerciales car c'est celui qui est le plus adapté à la situation décrite.

Au vu de la définition du projet, il paraît essentiel, de bien se rappeler, que l'application doit être **disponible, accessible et facilement utilisable** sur tous types de supports.

Dans ce même cadre, il nous faudra garder à l'esprit que l'application doit être **supportée par un maximum de plateformes**, de navigateurs, y compris pour des utilisateurs ne possédant pas de supports avec une grande puissance de calcul.

De plus, il est important de garder à l'esprit que les utilisateurs **ne doivent pas être bloqués** si jamais ils se retrouvent **sans connexion internet**.

Comme exprimé précédemment, notre application nécessitera un système de **modération à posteriori** en lien avec la possibilité d'apparition de contenu inapproprié au sein de l'application. Cela implique aussi la mise en place d'un système de blacklist des pseudonymes.

Nous devrons aussi s'assurer que les données de nos utilisateurs soient **stockées de manière sécurisée** et en respectant les principes exposés dans le Règlement Général sur la Protection des Données (RGPD).

Nous sommes aussi confrontés à une **contrainte budgétaire** qui nous oblige à n'avoir qu'un seul serveur supportant l'ensemble de l'application ainsi que ses différents composants.

Enfin, il faudra anticiper la possibilité de travail à distance afin de conserver notre équipe dans son intégralité face au risque de la **COVID-19**.

f. Les risques

Il existe plusieurs risques en lien avec notre projet :

Compatibilité : Notre application sera en partie disponible sur navigateur internet. Certains de ces derniers peuvent se retrouver incompatibles avec les technologies utilisées dans le fonctionnement de l'application et donc rendre cette dernière incompatible.

Contenu : Le risque d'avoir des publications inappropriées sur l'application. Cette dernière étant ouverte à tout public, y compris les personnes mineures, il est possible que ces personnes se retrouvent en face de contenu choquant ou blessant. Un système de modération doit donc être implémenté, qui permettra de modérer les pseudonymes des utilisateurs ainsi que les dessins qu'ils réaliseront.

Matériel : Un risque matériel lié à la machine qui héberge l'application, si cette dernière s'avère hors d'état de marche, notre application ne serait plus disponible aux utilisateurs. Il faudrait donc envisager d'avoir une machine de secours, qui prendra le relais en cas de panne de la machine principale.

Équipe : Étant donné le contexte actuel, notre équipe est confrontée au risque de la perte d'un membre si ce dernier s'avère en contact de la COVID-19. Il faudra donc anticiper la possibilité de travail à distance afin de conserver notre équipe dans son intégralité face au risque COVID-19 grâce à des logiciels tels que Zoom ou Discord.

Public : Notre application cible un public très général et donc des personnes de toute tranche d'âge. Le design de l'application peut donc rencontrer un risque de manque d'ergonomie.

Données : Les données sensibles des utilisateurs peuvent être interceptées ou extraites, ce qui représente un risque important concernant la protection des données. Ces données devront donc impérativement suivre le Règlement Général sur la Protection des Données. (RGPD)

Connexion : Les utilisateurs auront recours à une connexion internet afin d'afficher les réalisations ou même afin de soumettre sa réalisation. Nous devons donc prendre en compte le risque de perte de connexion internet auprès des utilisateurs grâce à un système de mise en cache.

Retard : Nos délais de rendu sont restreints et parfois courts. Il existe donc un risque de retard lors de la soumission des rendus.

Fonctionnalité : Par corrélation avec le dernier risque énoncé, notre application peut s'avérer non-fonctionnelle ce qui représente un risque majeur car notre objectif premier est de rendre cette dernière utilisable et fonctionnelle.

Stratégie de mitigation utilisé pour ces risques :

Compatibilité : acceptation ⇒ Notre application sera adaptée afin qu'elle soit disponible sur un maximum de support, mais si toutefois un navigateur se montre incompatible ne n'auront pas d'autre choix que de l'accepter.

Contenu : protection ⇒ Modération périodique des différentes publications postées.

Matériel : acceptation ⇒ Comme décrit dans les contraintes, nous ne pouvons pas avoir un serveur de secours, nous acceptons donc ce risque.

Équipe : réduction ⇒ Le contexte actuel (confinement) réduit le risque lié à la COVID-19 mais il n'est pas inexistant.

Public : réduction ⇒ Nous allons faire tester notre application à un large public afin de réduire au maximum ce risque.

Données : évitement et réduction ⇒ Nous allons sécuriser au maximum le stockage et le transit des informations qui ont lieu sur notre application.

Connexion : acceptation ⇒ Ce risque dépend uniquement de l'utilisateur, nous ne pouvons pas agir sur ce dernier et donc nous devons l'accepter.

Retard : évitement ⇒ La mise en place d'un Trello et d'un diagramme de GANTT nous permettent d'éviter ce risque (*cf. Annexe 3, Figure 7*).

Fonctionnalité : réduction ⇒ L'ensemble des méthodes et outils de travail instauré nous permettent de réduire au maximum ce risque.

		Impact		
		Faible	Modéré	Fort
Probabilité	Faible	Compatibilité	Public	Contenu - Fonctionnalité
	Modéré	Équipe	Contenu - Retard	Données
	Fort	Connexion		

Tableau 2 : Matrice de criticité des risques

3. Expression du besoin

a. Ciblage

Notre application se positionne comme un moyen de divertissement pour tout type d'âge, ne demandant pas de compétences d'artiste pour participer, utilisable partout et à n'importe quel moment.

Notre cible est donc très large : nous souhaitons toucher les plus jeunes comme les plus âgés, de tout horizon, afin de réunir les générations autour d'un jeu familial.

b. Les besoins

Nous identifions trois besoins principaux: **Dessiner**, **Collaborer** et **Partager**.

Ces trois besoins principaux s'expriment à leur tour des besoins secondaires. Par exemple, pour dessiner, il faut pouvoir choisir ses outils, visualiser sa case ou encore visualiser la case adjacente. Ces besoins secondaires peuvent à leur tour exprimer d'autres besoins.

L'ensemble des besoins est donc modélisé à travers un diagramme (*cf. Figure 6*).

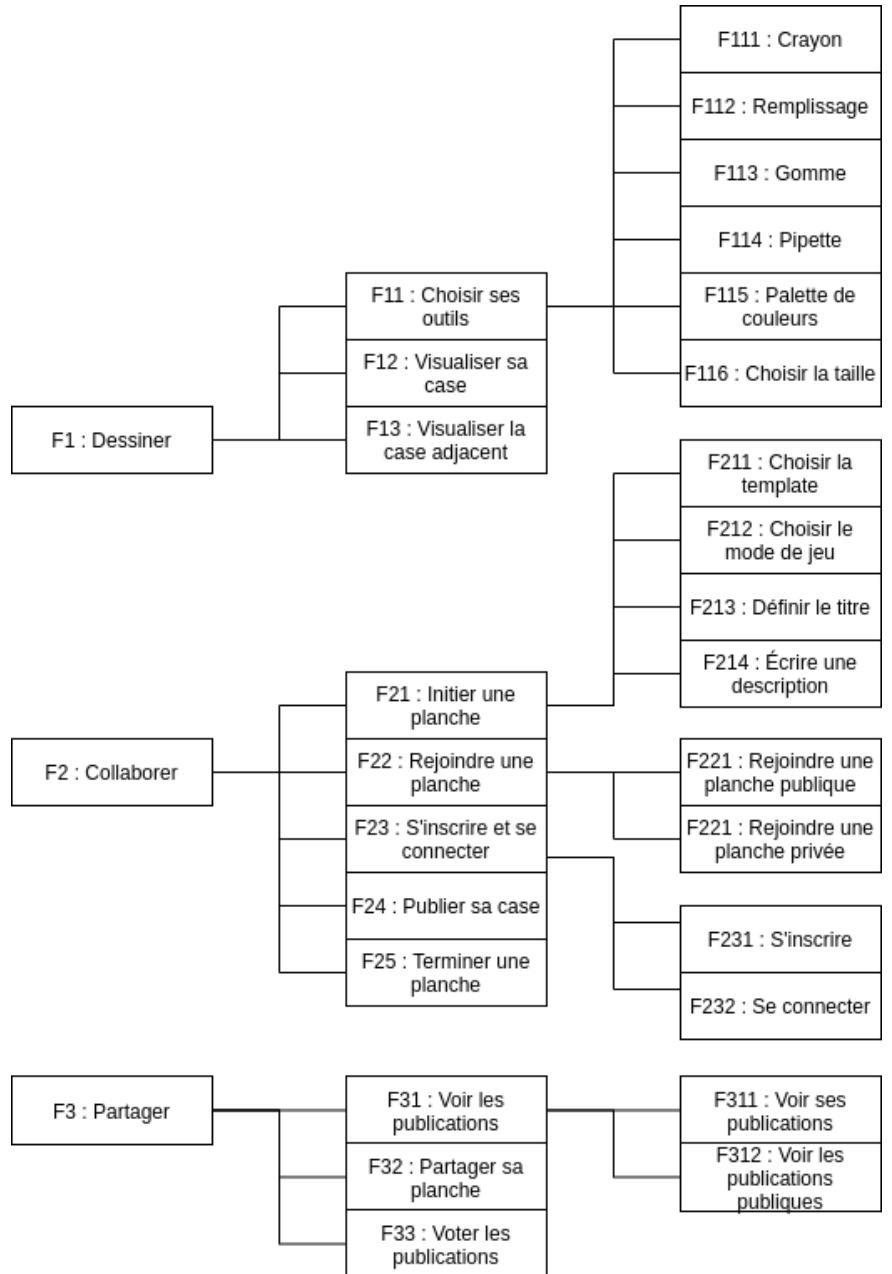


Figure 6 : Diagramme FAST des besoins.

I. Besoins fonctionnels

Afin que notre application soit fonctionnelle, il faut que les utilisateurs puissent :

- Avoir accès à une interface de dessin permettant la réalisation du premier dessin d'une planche ou d'un dessin intermédiaire en pouvant visualiser la création précédente.
- Collaborer à plusieurs sur une même planche de bande dessinée.
- Avoir accès à une interface permettant de naviguer entre les planches finalisées et les planches en cours de réalisation.
- Avoir accès à une interface de connexion et d'inscription permettant la connexion à son compte ou l'enregistrement d'un nouveau compte.

II. Besoins non-fonctionnels

Pour que notre application réponde aux attentes de l'utilisateur, il faut que :

- Elle puisse être utilisée sur tout type de support :
 - PC : plusieurs navigateurs, plusieurs tailles d'écran
 - Tablettes : plusieurs systèmes d'exploitations, plusieurs tailles d'écran
 - Smartphones : idem
- Les interfaces soient ergonomiques et intuitives,
- Les données (planches, cases et utilisateurs) soient conservé de manière sécurisé,
- La navigation soit rapide et fluide.
- L'affichage soit correct pour les différents composants.

III. Besoins techniques

Au vu des besoins fonctionnels et non fonctionnels exprimés précédemment, il apparaît certains besoins techniques :

- Un serveur qui pourra supporter l'application, son fonctionnement en arrière-plan et une base de données.
- Un appareil permettant l'utilisation de l'application comme un ordinateur, smartphone ou tablette.
- Une connexion internet.

c. Les critères de qualité

I. Qualité logicielle

De manière à fournir une application sûre, nous avons décidé d'utiliser les critères de qualité logiciel de la norme ISO 9126 (critères SQUARE). Ils nous apparaît important de détailler certains d'entre eux :

- **La fiabilité** : Comme notre application sera disponible sur tout type de support, elle visera donc un public très large et elle se doit donc d'être fiable. Cela signifie qu'elle doit avoir une certaine tolérance aux fautes du côté utilisateur. Cela inclut aussi la capacité de récupération. Que ce soit des données utilisateur ou de jeux.
- **La facilité d'usage** : Comme énoncé dans le critère précédent, notre application visera un public large et de toute tranche d'âge, il donc impératif que cette dernière soit utilisable de la manière la plus simple possible. Dans la même démarche elle se doit d'être intuitive et facilement compréhensible, c'est-à-dire que les utilisateurs doivent facilement naviguer au sein de l'application et qu'ils doivent facilement s'y retrouver si ils y retournent ultérieurement.
- **La portabilité** : Ce critère est sans doute l'un des plus importants. En effet, notre application à pour but de se retrouver disponible sur tout type de support. Ce critère se décompose en quatre attributs :
 - La facilité d'installation : Ce qui nous concerne uniquement sur la partie d'installation sur tablette et mobile mais cela est respecté car il s'agira ni plus ni moins que d'une simple application à télécharger. La partie sur ordinateur n'est pas concernée car il s'agit d'un simple site internet.
 - La facilité de migration et l'adaptabilité : Ce qui est aussi respecté car comme expliqué plus tard notre application web s'adapte facilement sur des supports plus petits grâce à une technologie permettant de transférer cette dernière en une application mobile tout en gardant son fonctionnement qui reste le même.
 - La conformité : Enfin, l'ensemble de notre application respectera l'ensemble des normes existantes la concernant tel que les normes sur les bases des données ou encore sur le modèle de développement.

Afin de s'assurer du bon fonctionnement de notre application, notre équipe effectuera une grande série de tests. Mais comme cela n'est pas suffisant, nous soumettrons notre application à des utilisateurs "testeurs" qui nous feront un retour du fonctionnement de l'application.

II. Qualité ergonomique

Dans la même démarche de qualité, l'ergonomie de l'application sera soumise à des utilisateurs testeurs sur les différents supports. Ces derniers nous feront part de la même manière de leur expérience sur le point ergonomique.

Pour permettre une analyse plus performante de l'interface, nous utiliserons les critères de qualité de *Bastien et Scapin*. Ils nous paraît important de spécifier certains d'entre eux :

- Le guidage : l'utilisateur devra avoir un parcours optimal au sein de notre application car n'importe quel usager doit pouvoir se repérer dans l'application. Afin de rendre l'expérience utilisateur optimal, il devra avoir :
 - un feedback rapide
 - des incitations lui permettant de se déplacer aisément selon les différentes interfaces ...

- La gestion des erreurs : si l'utilisateur réalise un action auquel il n'est pas autorisé, les messages d'erreur renvoyées devront lui permettre de comprendre sans effort et rapidement son erreur.
- La charge de travail : le consommateur de l'application aura une charge de travail réduite ; les étapes et la lecture entre les différentes vues seront simples et limitées.
- La compatibilité : notre application a pour vocation d'être utilisable par n'importe quelle plateforme, c'est pourquoi nous serons vigilants à ce que toutes nos implémentations soient adaptables sur tout type de support (n'importe quel navigateur, tablette, smartphone).
- Homogénéité : afin que l'utilisateur puisse s'adapter facilement selon son support d'utilisation, nous mettrons en place des interfaces quasiment identiques et cela ne dépendra pas du support d'utilisation.

III. Critère de test

Dans un soucis de réussite, nous avons listé différents éléments qui doivent répondre à des critères précis de test :

Element	Description	Critère de test
Vue de dessin	Dessin réalisable	Le dessin est bien affiché sur l'interface dédiée
	Aide au dessin	Il est possible d'avoir une aide au dessin, incluant la visualisation de la case précédente
	Dessin sauvegardé	Le dessin peut être affiché ultérieurement, il a bien été enregistré dans la base de données
	Interface intuitive	La majorité des utilisateurs test (en dehors de l'équipe de développement) a réussi à dessiner sans aide
Vue de connexion	Création d'un compte	La vue permet de se créer un compte
		La création d'un compte est bien sauvegardé
	Connexion	A partir des comptes déjà créés, il est possible de se connecter
Vue des planches	Visualiser des planches	Il est possible de visualiser les dernières planches

Tableau 3 : Critères de test

Pour valider l'ensemble de ces critères, nous soumettrons notre application à un plusieurs personnes qui seront donc de tout âge de manière à ce que ces tests soient le plus

représentatif de notre public cible. Par la suite nous demanderons à ces mêmes personnes de remplir un formulaire de retour les questionnant sur les divers points cités ci-dessus. (cf. *Tableau 3*)

4. Solutions techniques envisagées

a. Langages et utilisation

Le noyau fonctionnel de notre application peut être implémenté avec de nombreux langages comme PHP, Java, JavaScript, C++ ou encore d'autres langages de programmation.

Ce dernier sera réalisé à l'aide du langage **PHP** notamment car il est adapté au web et maîtrisé par l'ensemble du groupe.

Nous disposerons aussi d'une base de données qui pourrait être PostgreSQL, MySQL ou encore Sqlite.

PostgreSQL a été retenue car la manipulation est maîtrisée par tous notamment grâce aux cours suivis au cours des deux premiers semestres réalisés à l'IUT.

Les données stockées dans cette base communiqueront depuis le back-end vers la front-end au format **JSON** (JavaScript Object Notation), qui sont facilement interprétables par le PHP comme par le JavaScript.

Enfin, notre application doit être adaptée pour tous types de supports. Nous avons différentes possibilités afin d'arriver à cette fin. Réaliser le front-end en HTML et CSS tout cela de manière responsive, et de manière encore plus simple grâce à un framework CSS comme Bootstrap. Ou alors utiliser des framework JavaScript comme Angular ou React qui permettent de réaliser facilement des applications web en page unique ce qui évite le chargement ou recharge de page web.

Nous avons fait le choix en premier lieu de réaliser le front-end (visuel) qui sera disponible sur ordinateur à l'aide du framework **React.js**. Ce choix est pertinent car il nous permettra facilement d'implémenter le front-end sur tablette et mobile à l'aide de **React Native** qui est un autre framework permettant d'utiliser React.js avec les fonctionnalités native sur smartphones.

b. Architecture technique

I. Architecture de notre application

Notre application sera conçue sous le modèle de développement appelé "Modèle Vue Contrôleur" plus connu sous l'acronyme MVC.

Le modèle correspond à l'architecture logique de notre application. Ce dernier communiquera de manière directe avec le contrôleur qui sera chargé d'effectuer toutes les opérations de traitement de l'application. Enfin le contrôleur enverra les informations à

communiquer à l'utilisateur à la vue qui sera la partie graphique et visible de notre application. (*Annexes 4, 5, 6, 7, 8*)

II. Architecture du serveur

Le serveur hébergeant notre application est un VPS (Virtual Private Server) loué chez l'hébergeur français OVH. Le système d'exploitation est Debian 10 (Buster), choisi pour sa stabilité et ses performances. Nous avons adopté Apache comme serveur web car c'est une solution populaire très documentée.

5. Organisation du travail et découpage du projet

a. Mode de fonctionnement

Notre équipe fonctionnera en utilisant les principes de gestion semi-agile, c'est-à-dire une simplification d'une simplification des itérations (réduction du nombre de revues avec la maîtrise d'ouvrage).

Notre équipe se réunira tous les vendredi afin d'effectuer une mise au point de l'avancement des tâches définis la semaine précédente. Par la suite nous fixerons les tâches à réaliser pour le vendredi suivant.

Depuis la première itération, nous réalisons la conception générale de notre application. Suite à cela, nous commencerons la réalisation d'une première maquette fonctionnelle. Une fois la maquette réalisée, nous implémenterons fonctionnalité, par fonctionnalité le reste de l'application.

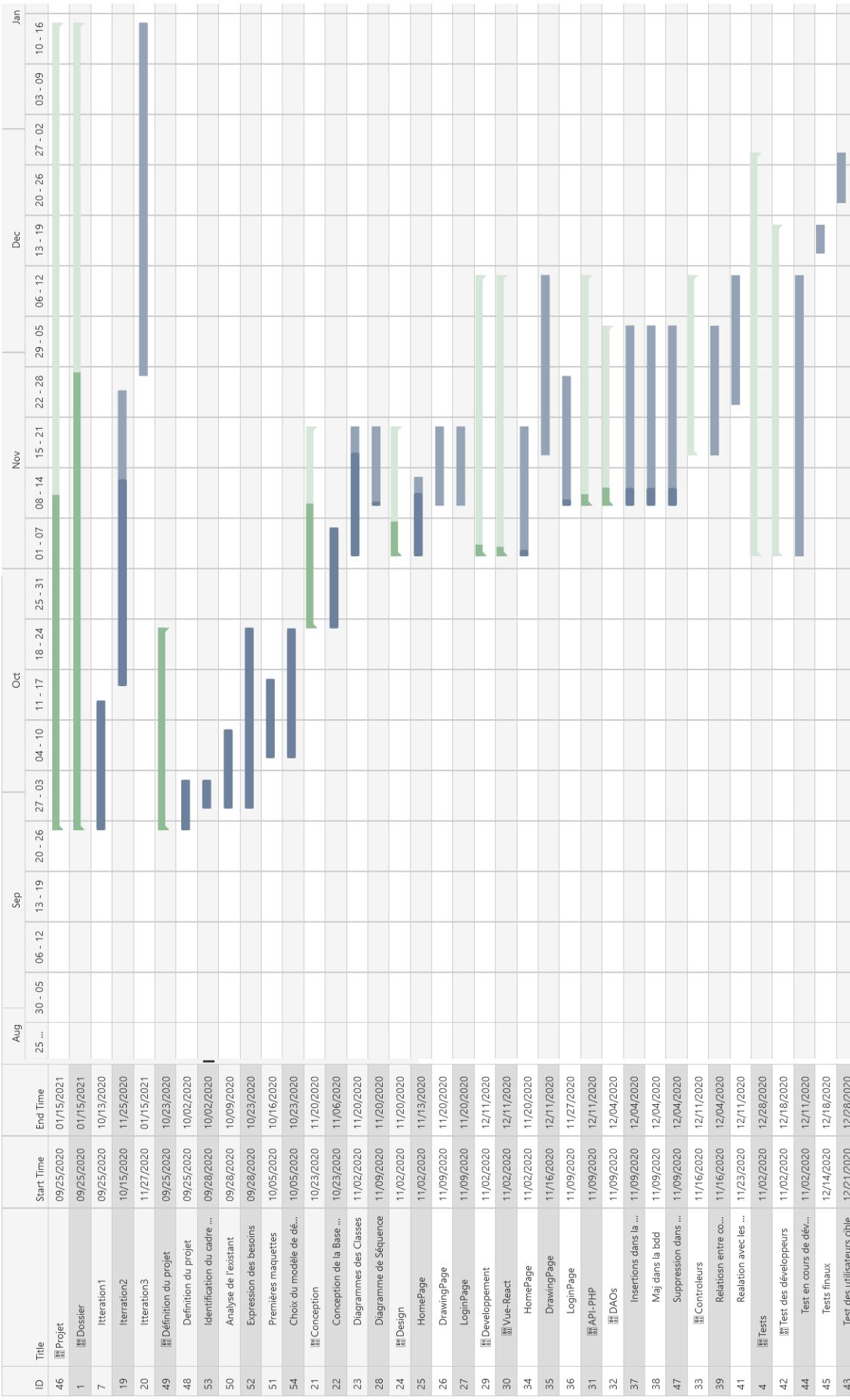


Figure 7 : Road-map prévisionnelle pour la première et deuxième itération

b. Outils

Afin de mener à bien notre projet, nous définissons plusieurs outils dont nous allons avoir besoin :

Les diagrammes de conception de notre application ont été conçus avec **Visual Paradigm**. La première version des maquettes non fonctionnelles a été réalisée avec le site **Draw.io**, et la seconde version avec **Figma**, un outil collaboratif de création d'interfaces.

Pour pouvoir collaborer tous ensemble sur ce projet, nous utiliserons des outils tels que **git** et le serveur **gicad-gitlab** (figure 1 en annexe) mis à disposition par l'université, ainsi que **Google Docs** (figure 2 en annexe) qui nous permet de modifier ce dossier tous en même temps.

Afin de correctement gérer notre avancement au cours du projet, nous utiliserons un outil de méthodologie de gestion de projet kanban appelé **Trello** (figure 3 en annexe).

Nous utilisons également des conversations textuelles et vocales **Discord** pour communiquer efficacement en dehors des heures de cours prévues pour notre projet (et également lors des heures prévues à cause de la Covid-19).

Dans le but de débugger notre back-end sans modifier la front-end, nous utilisons le site **apitester.com** qui permet d'envoyer des requêtes HTTP GET ou POST à notre back-end et de visualiser les données JSON renvoyées. L'outil simule donc les requêtes de notre front-end à notre back-end.

Pour faciliter les déploiements des modifications faites par chacun d'entre nous sur le git, nous avons mis en place **un script bash** sur le serveur permettant de *git pull*, puis compiler le client React et le déployer sur le serveur apache, et enfin déployer les scripts PHP sur ce même serveur.

6. Secondes version des maquettes non-fonctionnelles

a. Navigateur / tablette

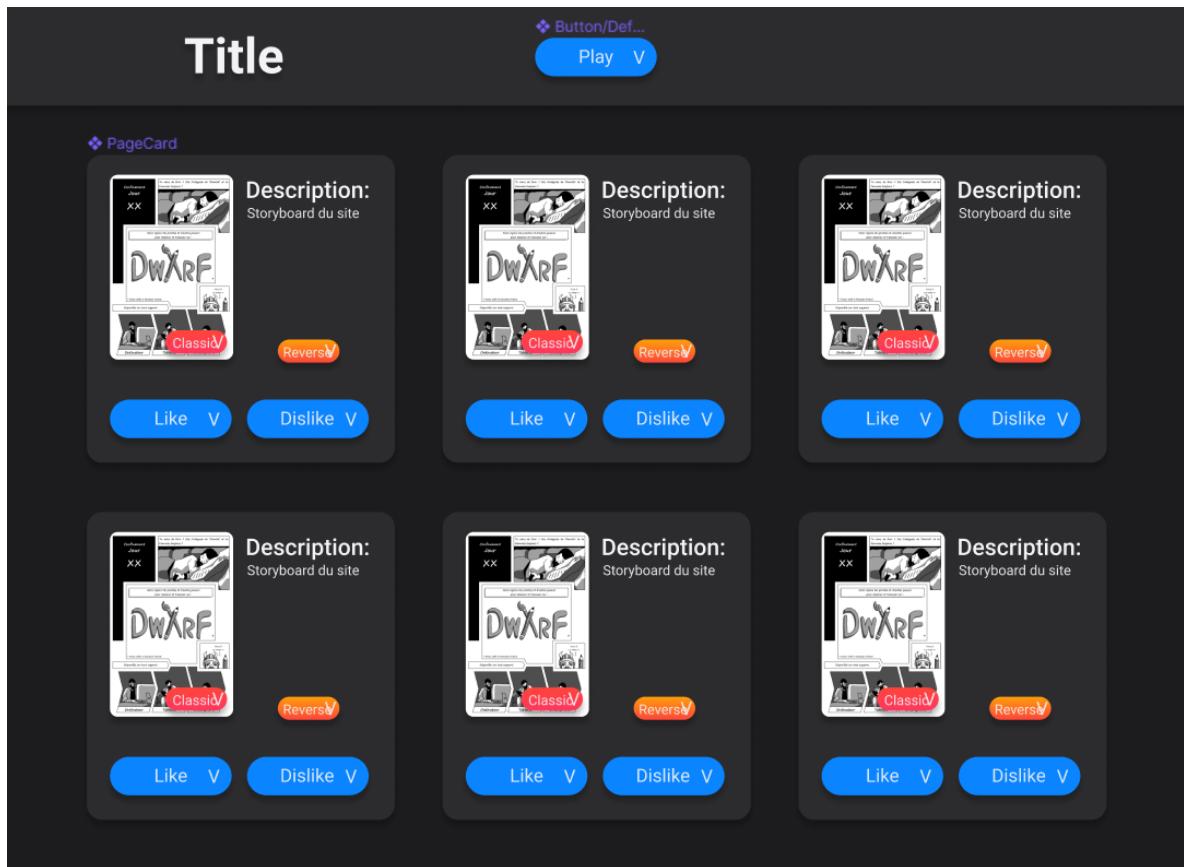


Figure 8 : Page principale

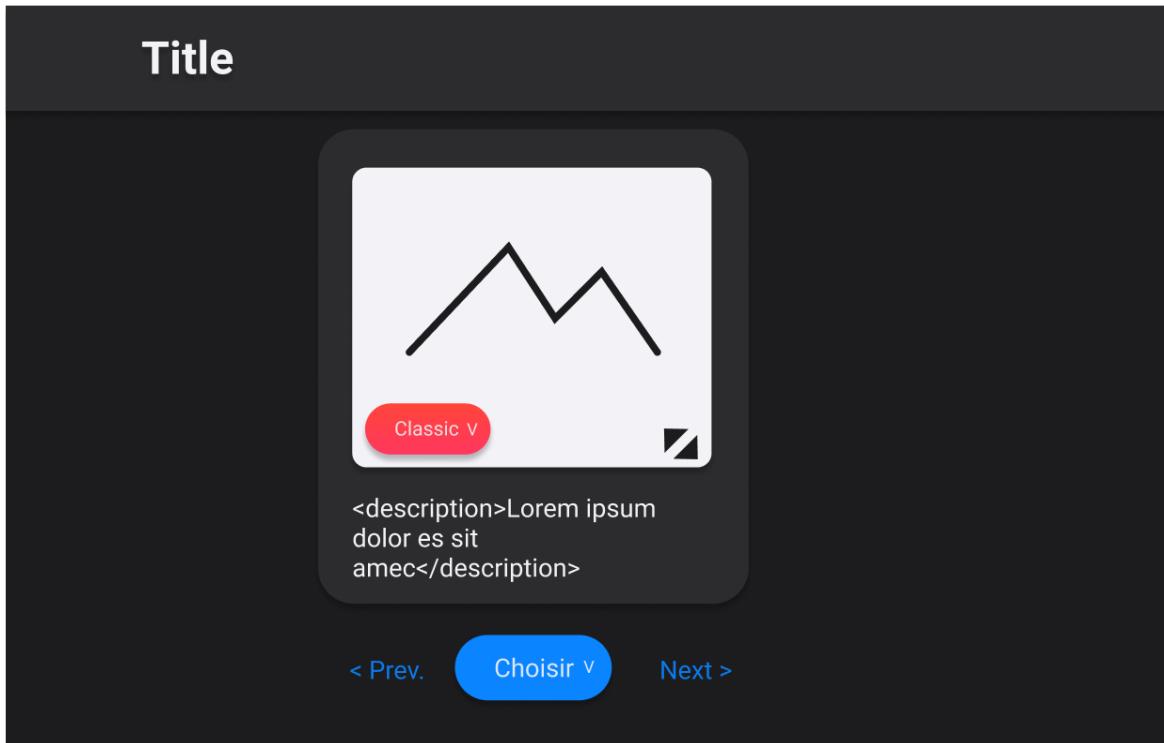


Figure 9 : Navigation entre les planches en cours

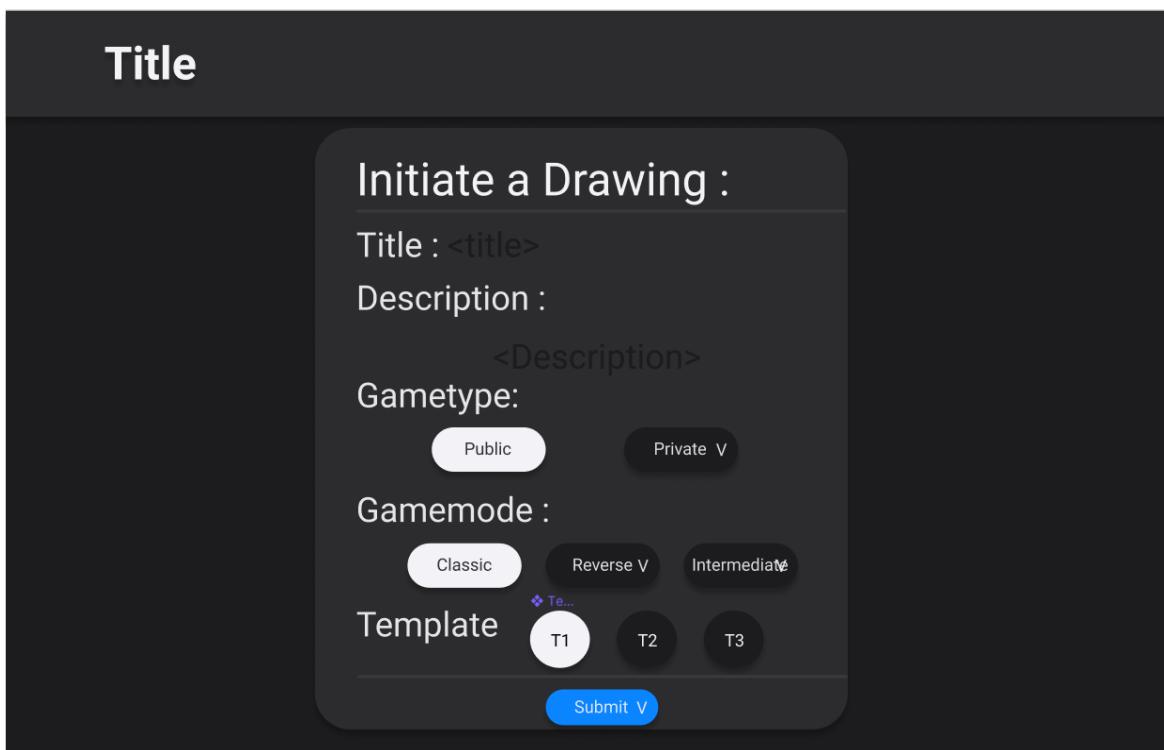


Figure 10 : Formulaire de création de planche

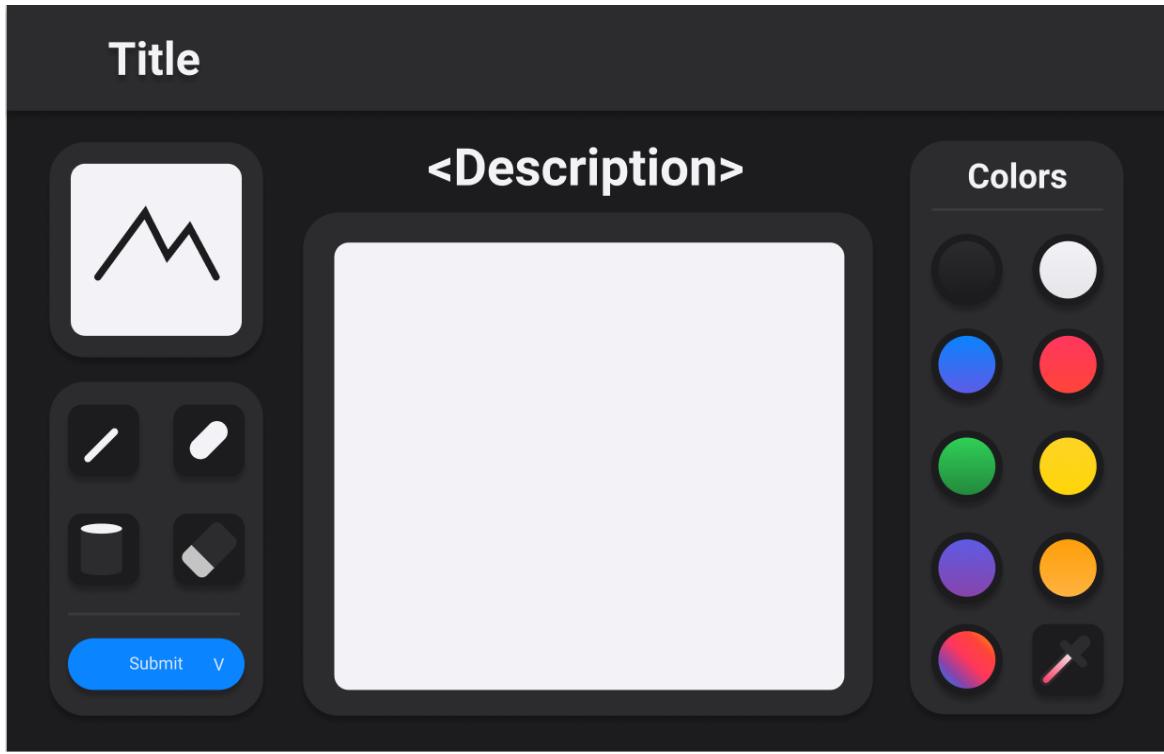


Figure 11 : Interface de dessin

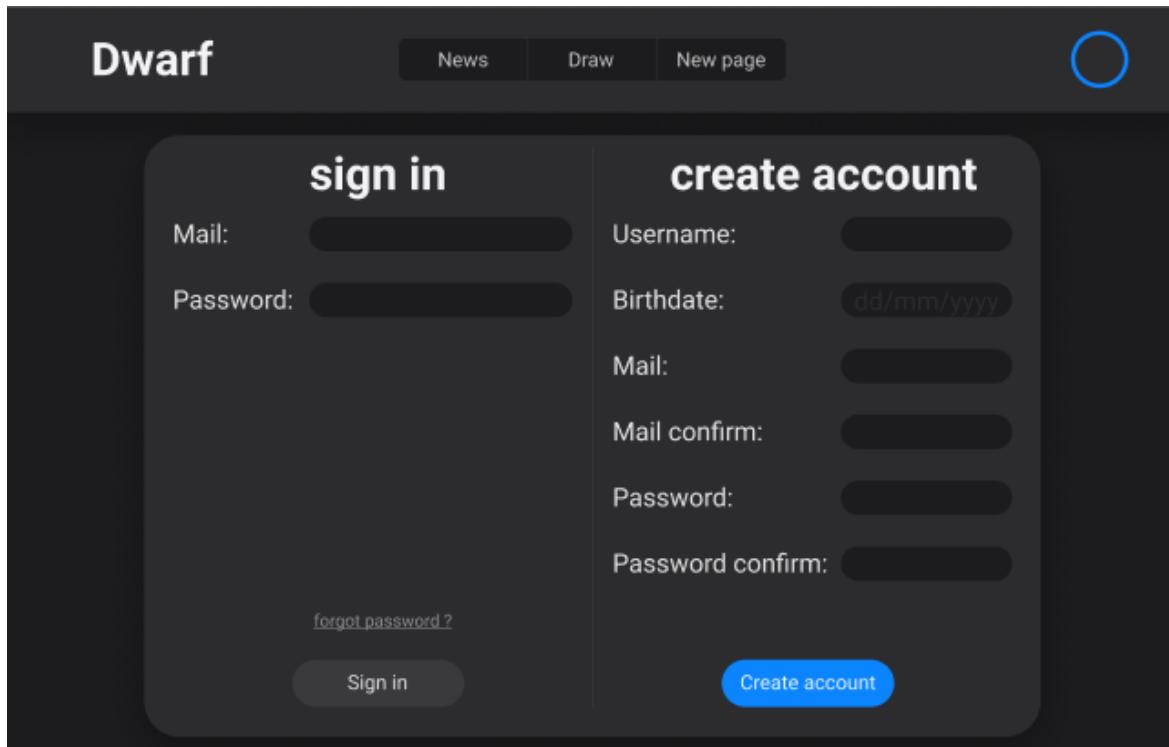


Figure 12 : Interface de connexion / Inscription

b. Smartphone

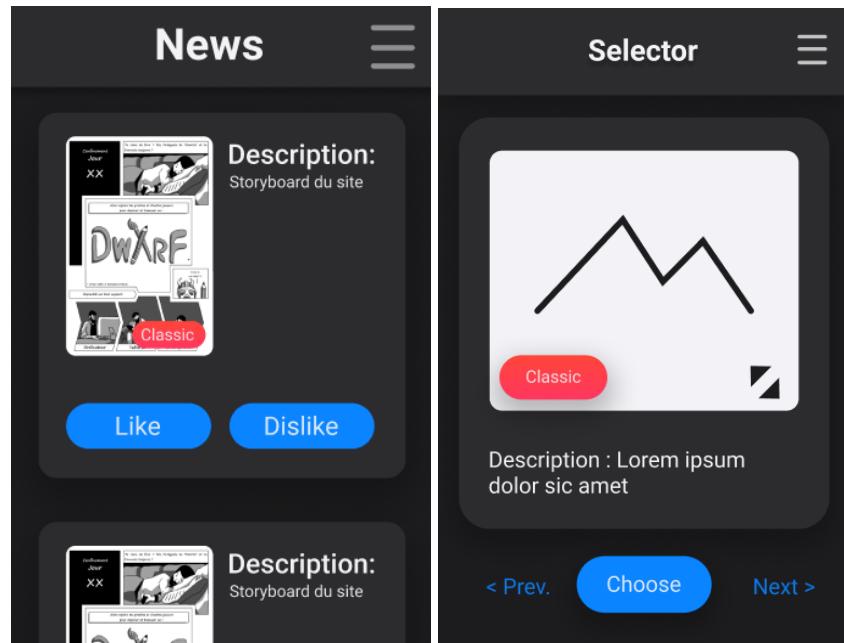


Figure 13 (gauche) : Page principale

Figure 14 (droite) : Navigation entre les planches en cours

Initiate a Drawing :

Title : <title>

Description :
<Description>

Gametype:
Public Private

Gamemode :
Classic Reverse Intermediate

Template T1 T2 T3

Submit

Figure 15 : Formulaire de création de planche

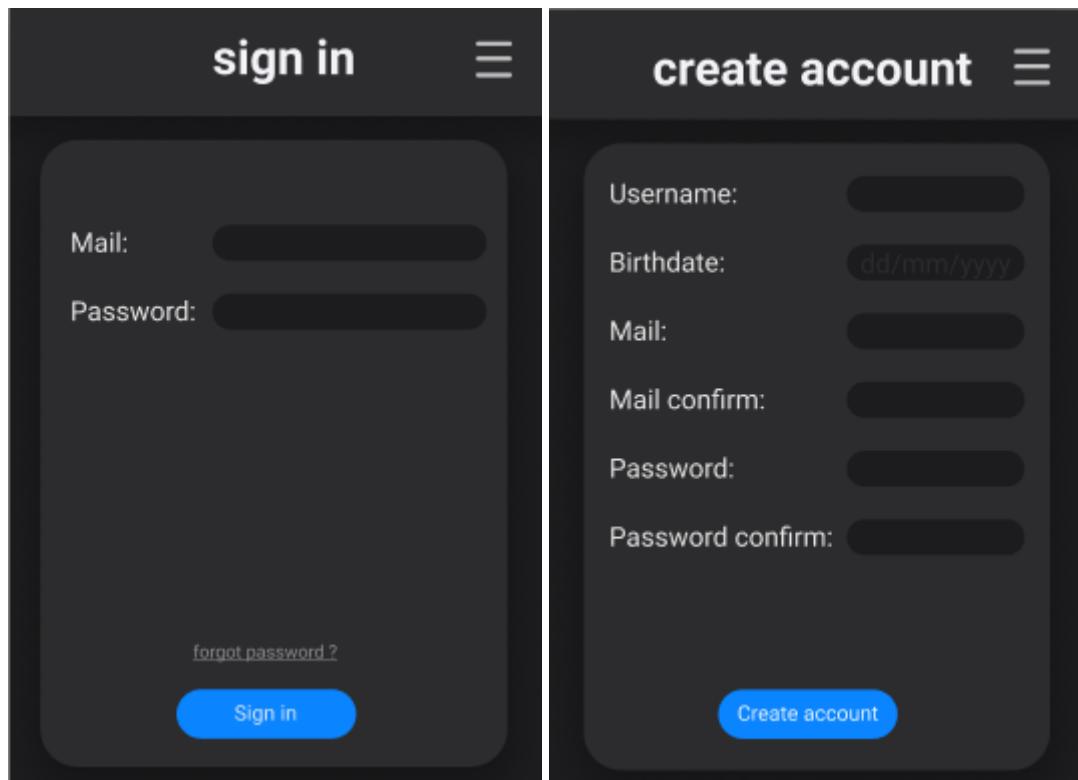
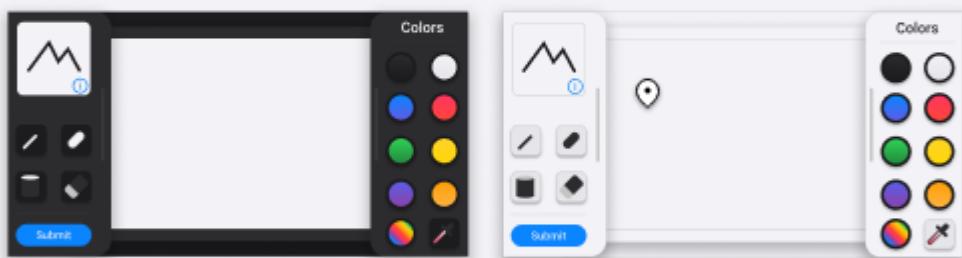


Figure 16 : Interface de connexion / Inscription

Zone de dessin de base :



Menu adjacent déplié :



Menu du bouton (info) de l'image :

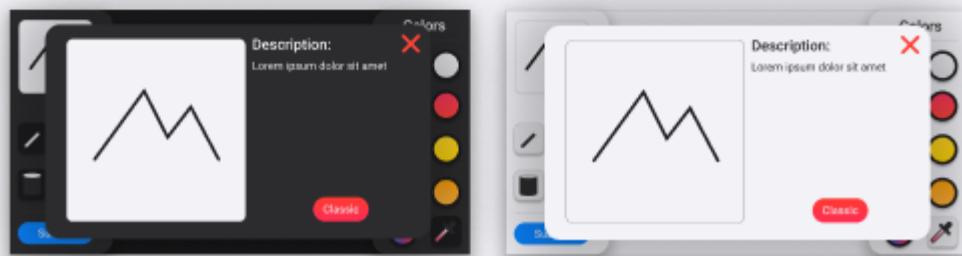


Figure 17 : Interface de dessin (white mode)

7. Annexes

The screenshot shows a GitLab project page for 'rendus'. The sidebar on the left lists various project management sections: Project overview, Details, Activity, Releases, Repository, Issues (0), Merge Requests (0), CI / CD, Operations, Packages & Registries, Analytics, Wiki, and Snippets. The main content area displays the project details: 'rendus' (Project ID: 7710), 157 Commits, 1 Branch, 1 Tag, 6 MB Files, 6 MB Storage. It also shows that the project is forked from an inaccessible project and has integrations with 'chamilo' and 'discord'. Below this is a commit history showing a merge branch 'master' by Tanguy Signoret. A table lists recent commits, including '.gitlab/issue_templates', 'api', 'bdd', and 'docs'. At the bottom, there are links for README, Add LICENSE, Add CHANGELOG, Add CONTRIBUTING, Add Kubernetes cluster, and Set up CI/CD.

Annexe 1 : Gricad qui est une plateforme de l'UGA qui utilise l'outil git qui permet de stocker et de garder une version à jour de notre projet auprès de tous les membres de notre équipe.

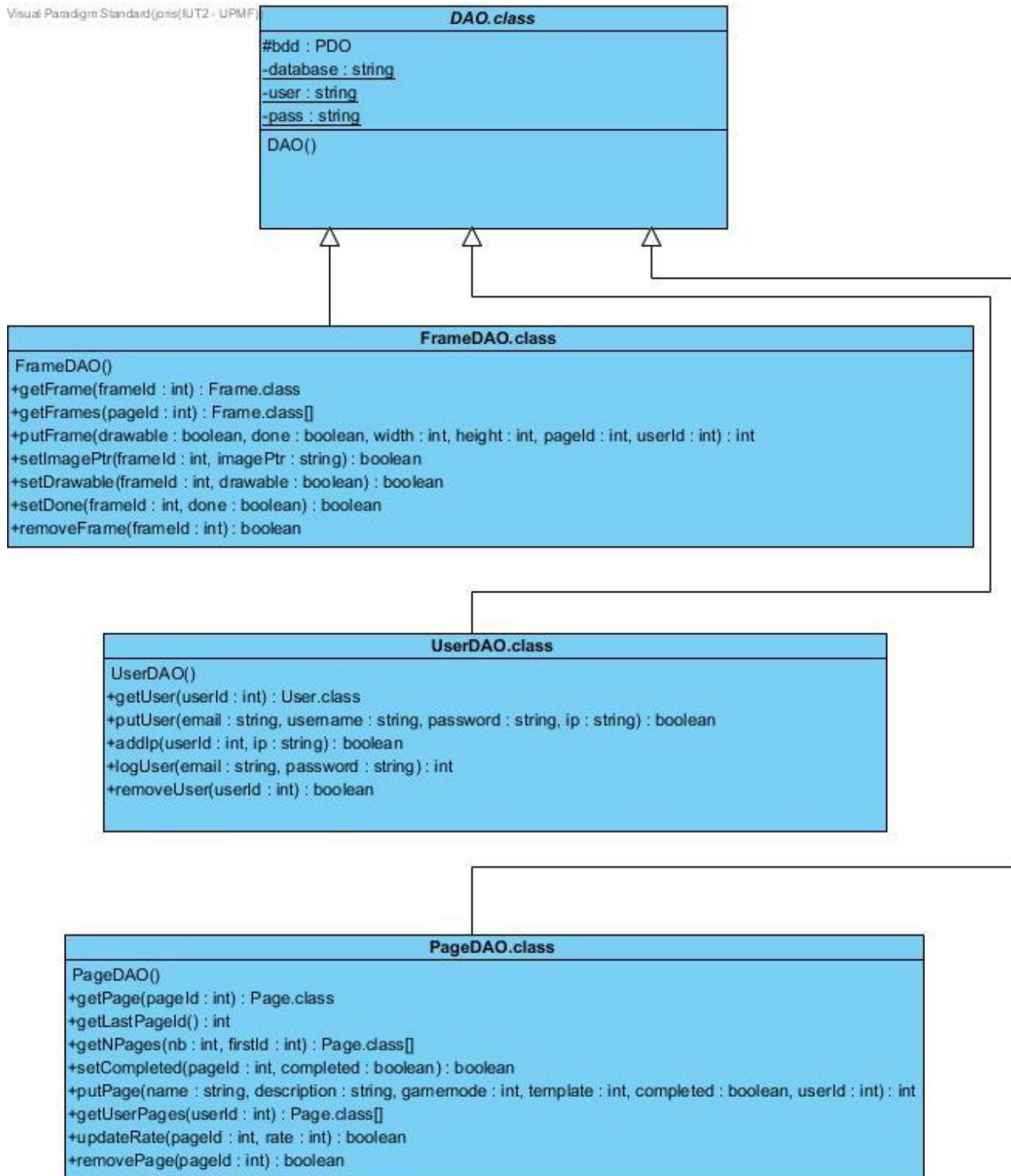
The screenshot shows a list of files shared with the user. The files are: 'BRAINSTORMING' (modified 23 oct. 2020 by Tanguy Signoret), 'CR_Retour_Iteration1' (modified 16 oct. 2020 by Tanguy Signoret), 'Presentation1' (modified 23 oct. 2020 by Tanguy Signoret), 'Presentation2' (modified 23 oct. 2020 by Tanguy Signoret), and 'TEAM DWARF' (modified 13:59 by Tanguy Signoret). The table includes columns for Nom, Propriétaire, Dernière modification, and Taille du fichier.

Nom	Propriétaire	Dernière modification	Taille du fichier
BRAINSTORMING	Tanguy Signoret	23 oct. 2020	—
CR_Retour_Iteration1	Tanguy Signoret	16 oct. 2020	—
Presentation1	Tanguy Signoret	16 oct. 2020	—
Presentation2	Tanguy Signoret	23 oct. 2020	—
TEAM DWARF	Tanguy Signoret	13:59	—

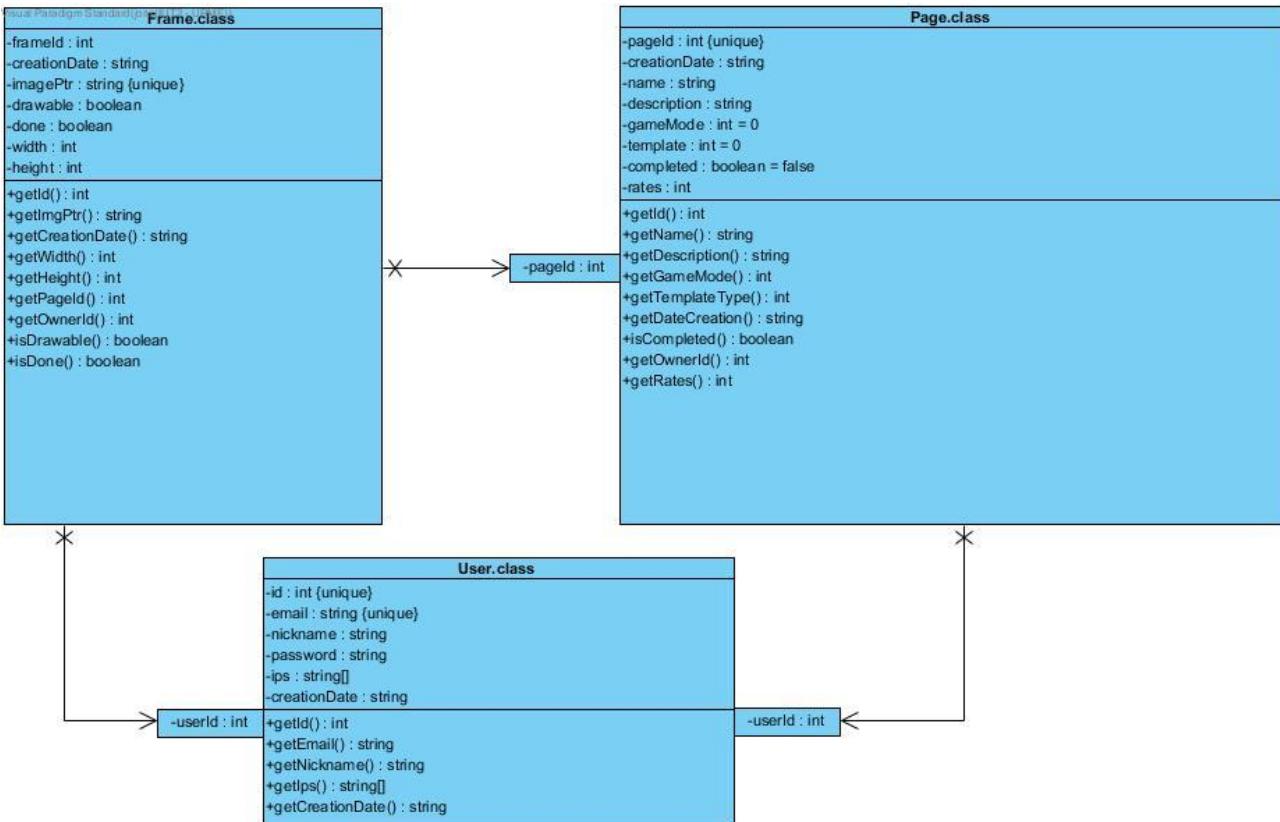
Annexe 2 : Google Drive qui permet l'édition de fichier, notamment de ce dossier, de manière simultané entre plusieurs utilisateurs. Permet aussi de stocker des fichiers.

The screenshot shows a Trello board with five columns: 'Options', 'To Do', 'Doing', 'Done', and 'Testing'. The 'Options' column contains cards for 'Adaptation pour Tablette et mobile -> Fonctionnalités native' and 'Espace personnel (Planche créé, auquel on a participé)'. The 'Systeme de modération' card has a '+ Add another card' button. The 'To Do' column has a card for 'Inscription'. The 'Doing' column contains cards for 'Vue planche en cours, à compléter', 'Système de notation', 'implémentation des tempalte', 'implémentation des modes de jeu', and '+ Add another card'. The 'Done' column has a card for 'Autentification' and '+ Add another card'. The 'Testing' column contains cards for 'BDD - Construction', 'gantt - itération 1 et itteration 3', and '+ Add another card'.

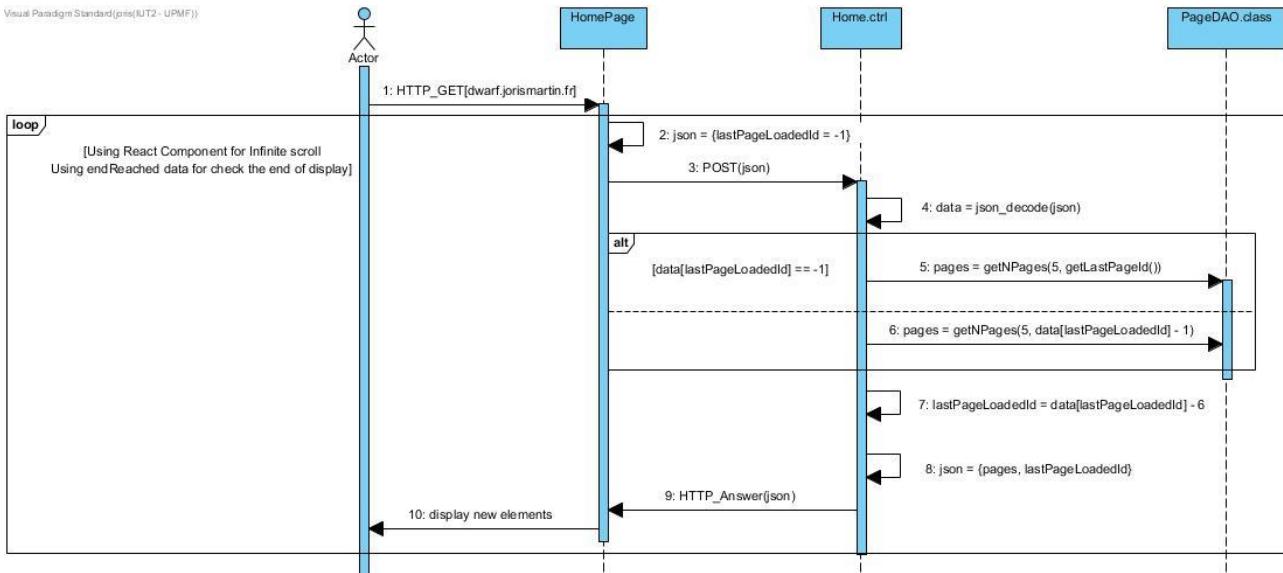
Annexe 3 : Trello qui est un outil permettant de constater l'avancement des tâches



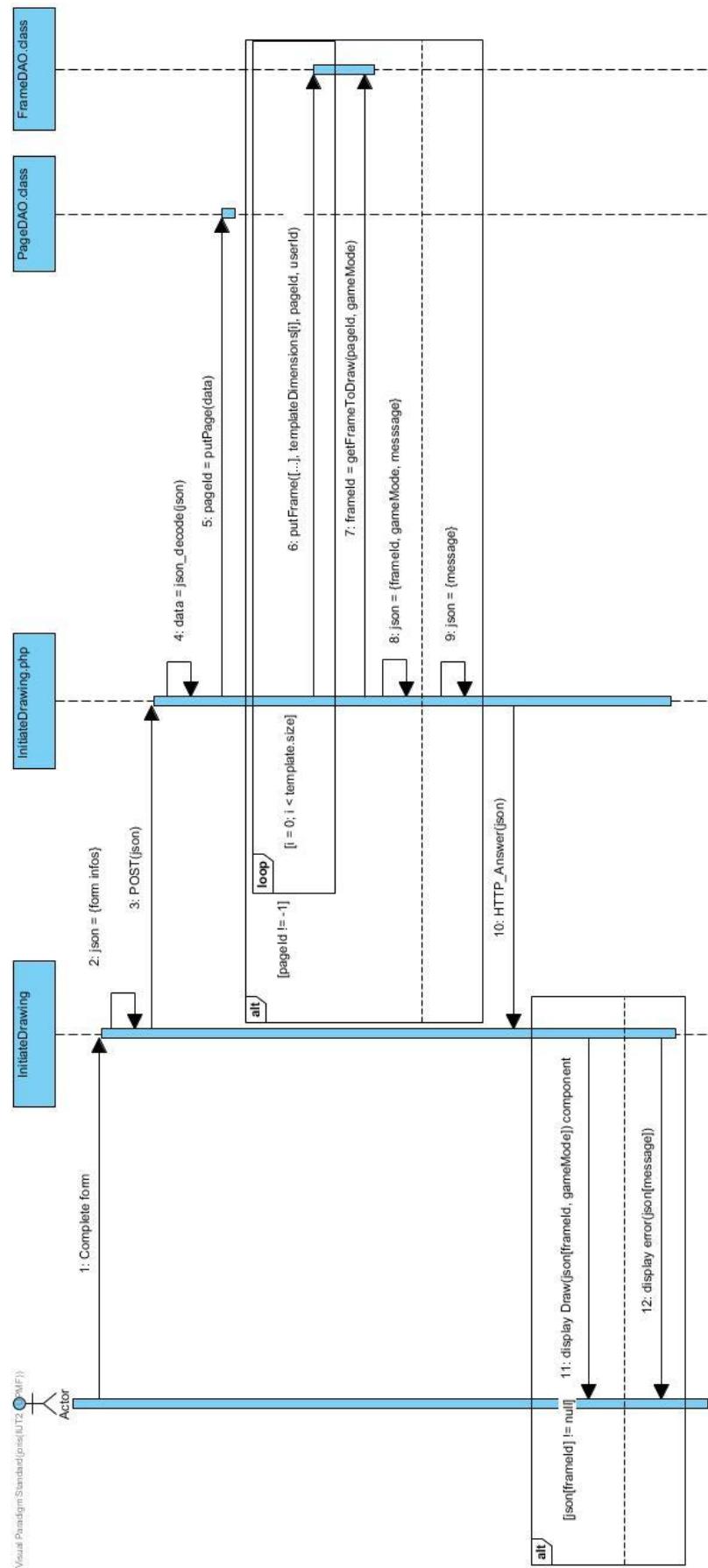
Annexe 4 : Conception ⇒ Diagramme de classes du DAO (Gestion base de données)



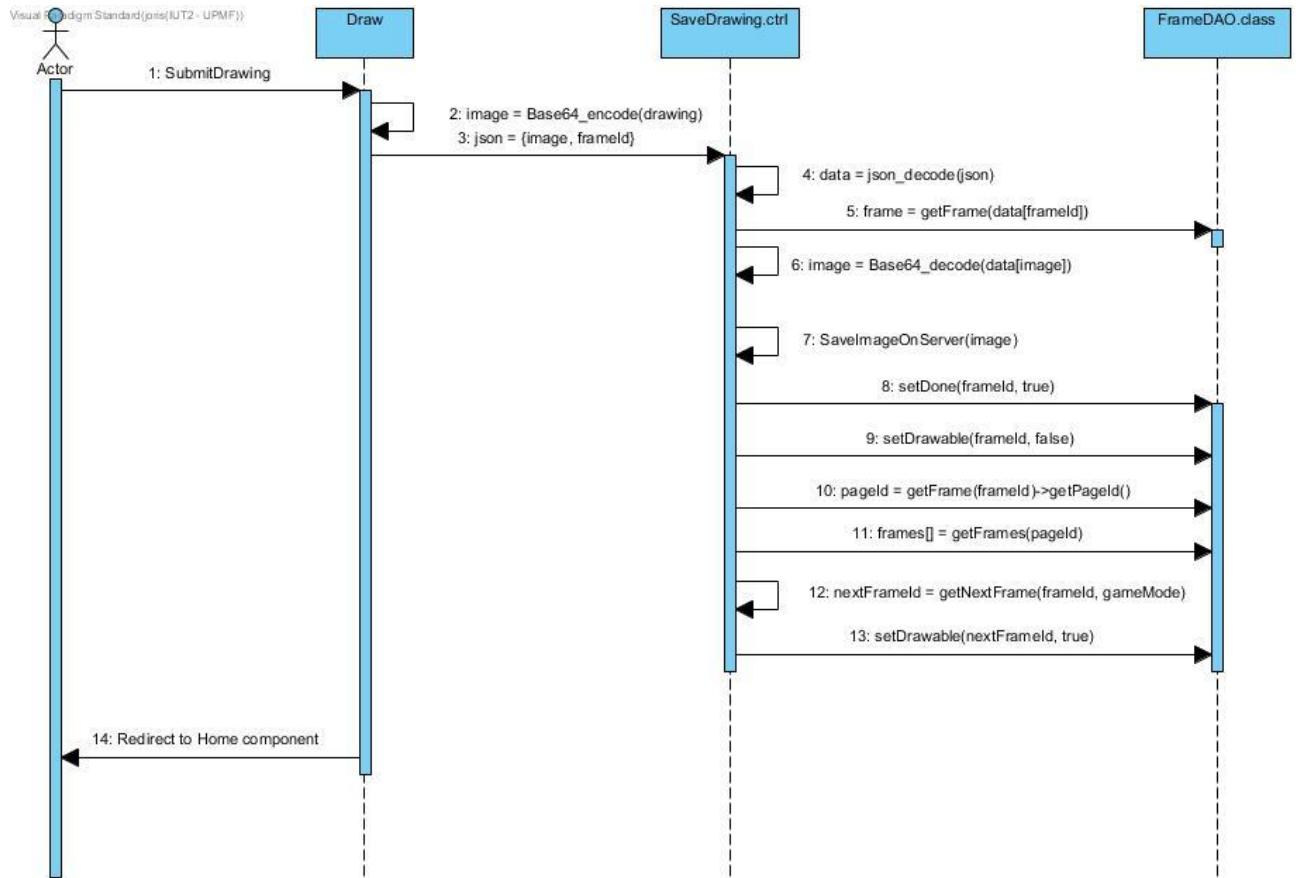
Annexe 5 : Conception ⇒ Diagramme de classes du modèle



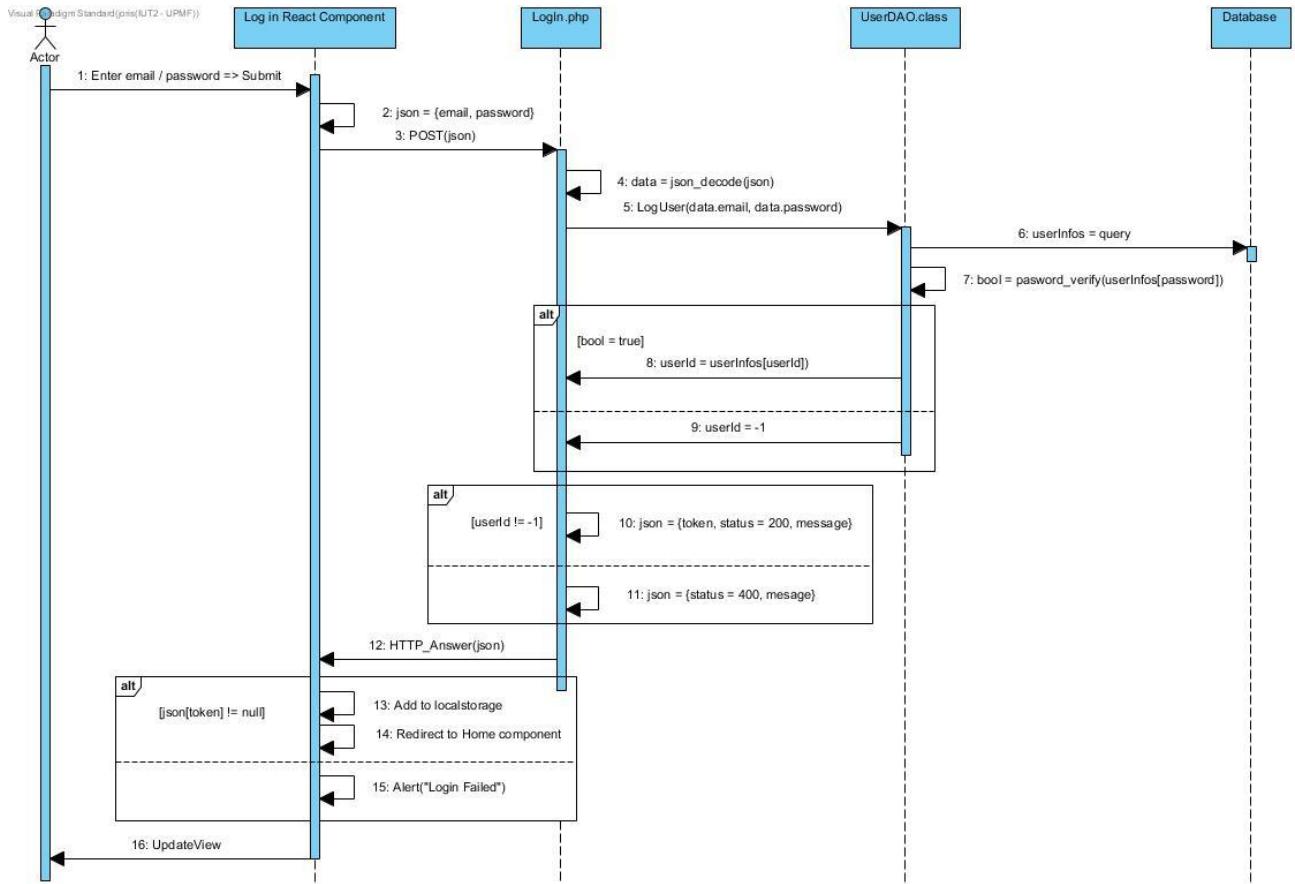
Annexe 6 : Conception ⇒ Diagramme de Séquence de l'affichage des BD sur la page d'accueil. (Diagramme simplifié due à l'utilisation d'un module React “Infinite Scroll” qui rend la modélisation difficile)



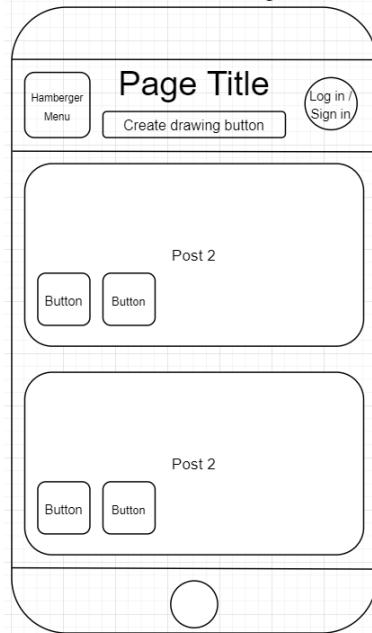
Annexe 7 : Conception ⇒ Diagramme de Séquence de la soumission du formulaire de création d'une planche



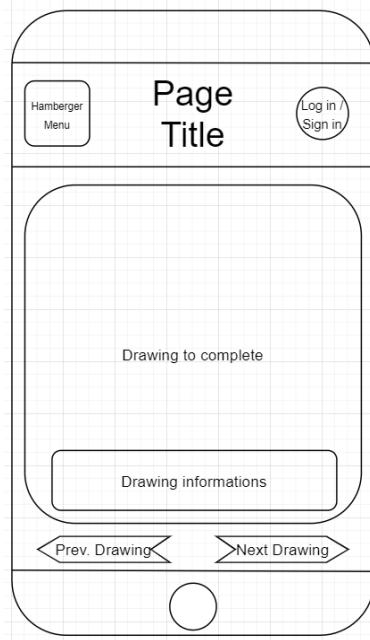
Annexe 8 : Conception ⇒ Diagramme de Séquence de la soumission d'un dessin.
(Diagramme temporaire)



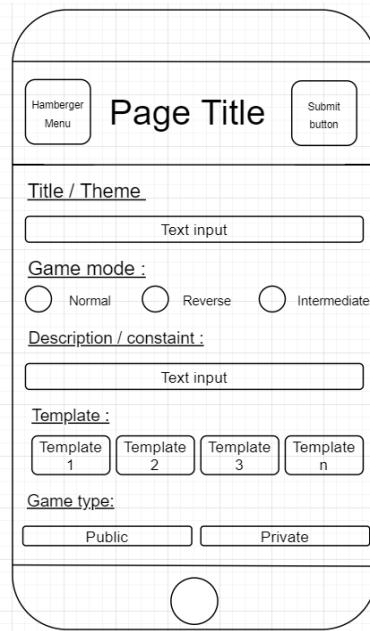
Annexe 9 : Conception ⇒ Diagramme de Séquence de la connexion



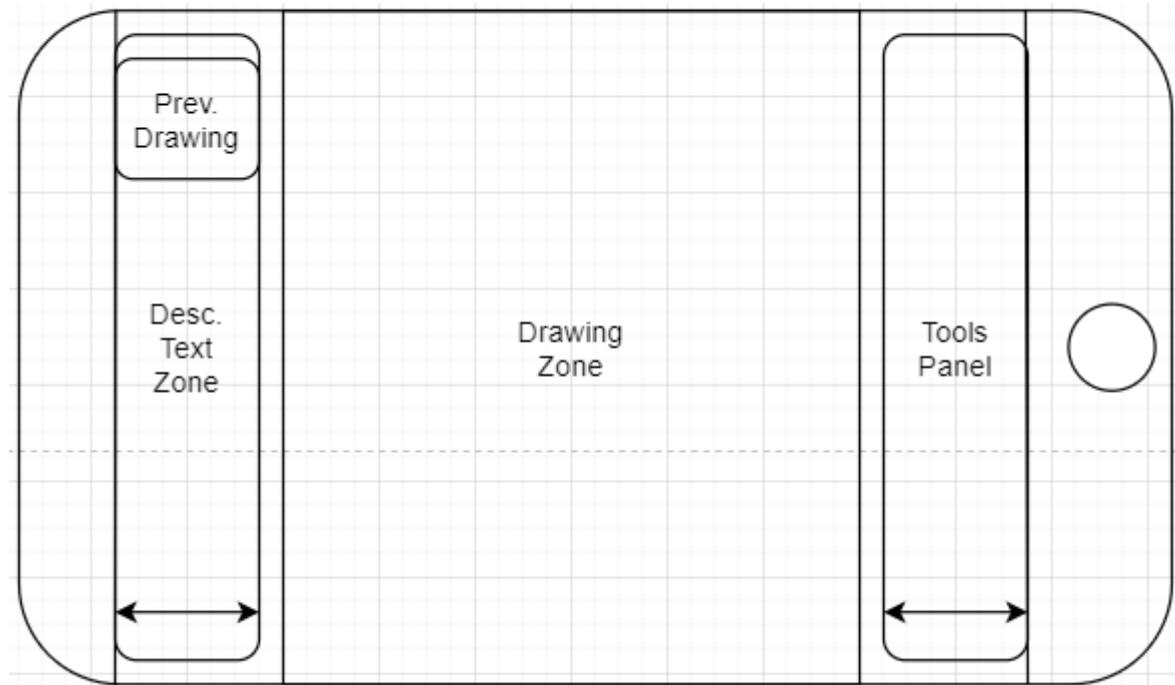
Annexe 10 : Première version maquette smartphone - Page principale



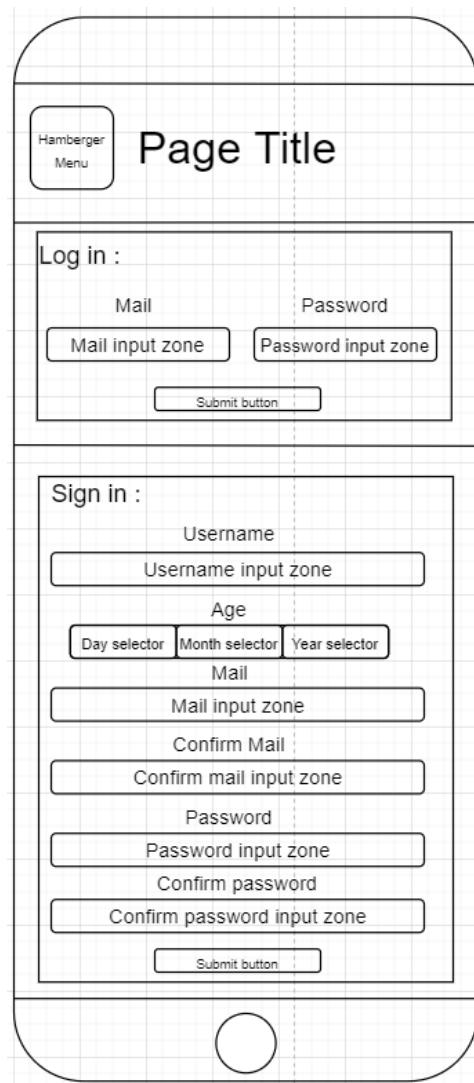
Annexe 11 : Première version maquette smartphone - Navigation entre les planches en cours



Annexe 12 : Première version maquette smartphone - Formulaire de création de planche



Annexe 13 : Première version maquette smartphone - Interface de dessin



Annexe 14 : Première version maquette smartphone - Interface de connexion /
Inscription