

---

Conception Objet- Projet Book7  
Rapport

---

I/ Objectif du projet

II/ Gestion de projet

II.1 Outils et organisation

II.2 Des features au user story

II.3 graphe de dépendance

III/ Présentation technique

III.1 Modèle

III.2 Interface graphique

III.3 Choix de conception

IV/ Conclusion

## I/ Objectif du projet

Objectif du projet : L'application est faite pour les étudiants de l'ENSEEIH

Qui souhaitent numériser leur cahiers et prendre des notes.

Notre produit est une application de gestion de cahier numériques qui simule un cahier manuscrit avec la souplesse du numérique à la différence des PDF ou du simple cahier

le cahier numérique permet d'inclure des PDF sur le cahier numériques et fournir des outils de mise en forme

Objectif du sprint 3:

A la fin du sprint, l'utilisateur pourra styliser le contenu de son cahier et y rajouter des pdf. Il aura la possibilité de le partager et de récupérer le cahier de ses collègues.

Nous n'aurons pas le temps de traiter toutes les user story restantes pour ce dernier sprint. Nous privilégions donc celles avec le meilleur rapport points d'efforts/valeur métier.

Users Story associées:

- Sélection pdf
- Parsage pdf
- import des diapos
- Alignement du texte
- Style texte
- insérer code
- Fond page
- couleur
- export lecture (au format pdf)
- export module lecture/écriture

(+ changement du mécanisme de sauvegarde auto)

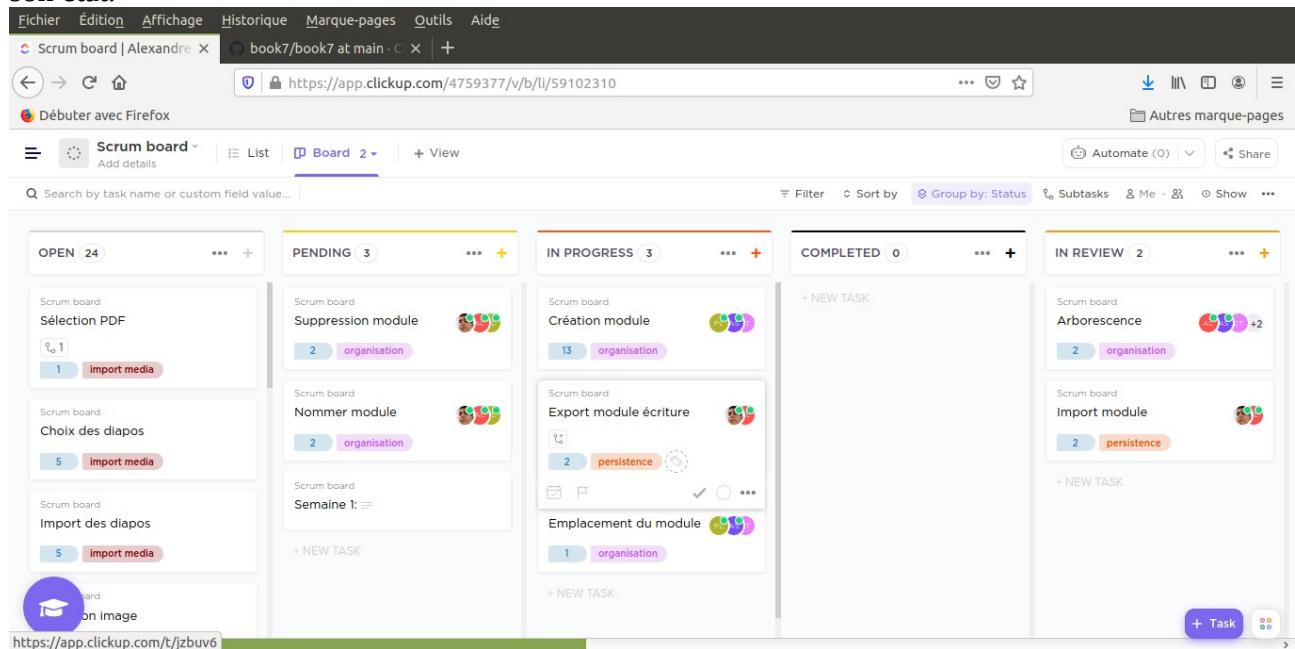
## II. Gestion de projet

On a Clément Torti qui est le product owner et Nassim Bennouar en tant que Scrum master. Tout le monde prend le rôle de développeur.

### II.1/ Outils et organisation

Clickup: Application de gestion de projet en mode agile.

Contient les Features-Epics-Users story avec les points d'efforts associés et les membres du groupe assignés. Chaque user story est placée dans le bon "board" (pending, in progress ...) en fonction de son état.



### Pair programming:

Nous avons une équipe de 3 élèves provenant de réseaux et 2 d'informatique. Pour palier cette hétérogénéité, nous avons travaillé par groupe 2-3 sur les premières users story.

### GitHub:

Le repository du projet est sur github: <https://github.com/Clement-Torti/book7>  
Envoyez un message à Clément pour obtenir les droits de cloner le projet

### Autre Outils:

On a également utilisé starUML pour l'UML, Balsamic mockup pour le design des vues des différentes fenêtres. Draw.io pour le dessin des dépendances des user-stories et Discord pour la communication en distanciel et le partage d'information.

## II.2 Des features aux user-stories

Définition des features :

Features : Import Media

**En tant que** étudiant de l'enseiht,

**Je veux** pouvoir importer des media (image/pdf) dans mon cahier numerique

**Afin de** pouvoir le completer

Features : Organisation

**En tant que** étudiant de l'enseiht,

**Je veux** organiser mes cahiers selon les modules de l'enseiht et un cahier selon les sections cours TD TP

**Afin de** retrouver facilement des informations

Features : Mise en forme

**En tant que** étudiant de l'enseiht,

**Je veux** Mettre en forme mes notes

**Afin de** personnaliser mon cahier et mettre en valeur le contenu

Features : Persistence

**En tant que** étudiant de l'enseiht,

**Je veux** Pouvoir enregistrer mon cahier et l'exporter/imprimer

**Afin de** conserver et partager le contenu du cahier

*Des features aux epics :*

Features	Epics
Import Media	<ul style="list-style-type: none"><li>• Importer PDF Ajout des diapo d'un pdf dans le cahier</li><li>• Importer JPG/PNG Ajout d'une image dans le cahier</li></ul>
Organisation	<ul style="list-style-type: none"><li>• Section des cahiers Les sections Cours/TD/TP d'un cahier sont séparés et indépendantes.</li><li>• Organisation des modules Cahiers placés dans des dossiers « semestres » aux noms pré-établis.</li><li>• Navigation L'utilisateur doit rechercher et accéder rapidement à l'information recherchée (metadonnee sur les elements)</li></ul>

Mise en Forme	<ul style="list-style-type: none"> <li>• Format du texte (tappé au clavier) Possibilité de choisir le format d’affichage du texte (gras/paragraphe/couleurs/code/puces)</li> <li>• Formes Ajout de forme ancrées sur des éléments du cahier (crayon/fleche)</li> </ul>
Persistence	<ul style="list-style-type: none"> <li>• Sauvegarde du cahier Permet de conserver le contenu des cahier sur le disque dur</li> <li>• Partage Transférer un cahier d’un ordinateur à l’autre</li> </ul>

### *Les user-stories :*

En parenthèse est indiqué le numéro du sprint où l’user-story a été réalisé. La remarque “pas fait” indique que la user story n’a pas été réalisée et “en cours” quelle n’a pas pu être terminée.

### Organisation :

#### Import Media :

Choix des diapo (3)

sélection pdf (3)

importer un pdf (3)

parcours du pdf (3)

sélection image (2)

importer image (2)

rognage image (pas fait)

création module (1)

nommer le module (1)

supprimer un cahier (1)

Arborescence (1)

localisation du module dans l'arborescence (1)

sections du cahier (1)

changer de page (2)

Ctrl+F (pas fait)

### Mise en Forme :

#### Persistence :

export en lecture (pdf) (pas fait)

export en écriture (2)

sauvegarde automatique (2)

ctrl+Z (pas fait)

import d'un cahier (en cours ,3)

ajout de date automatique (pas fait)

style texte (3)

Ajout de texte (2)

Alignement du texte (pas fait)

Insérer Code (3)

Ajout forme (pas fait)

redimensionnement forme (pas fait)

suppression formes (pas fait)

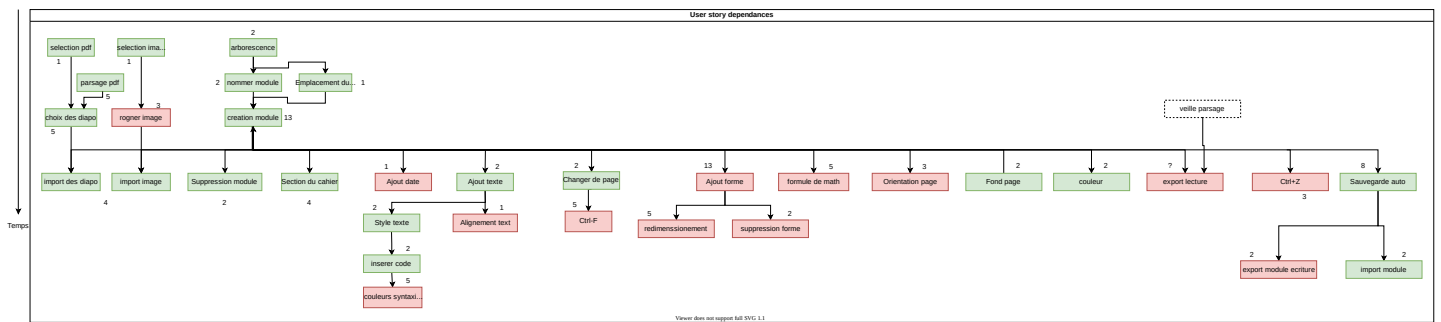
formules de math (pas fait)

orientation de la page (pas fait)

fond de la page (3)

couleur (texte et des formes) (3)

## IV/ Dépendance des user-stories

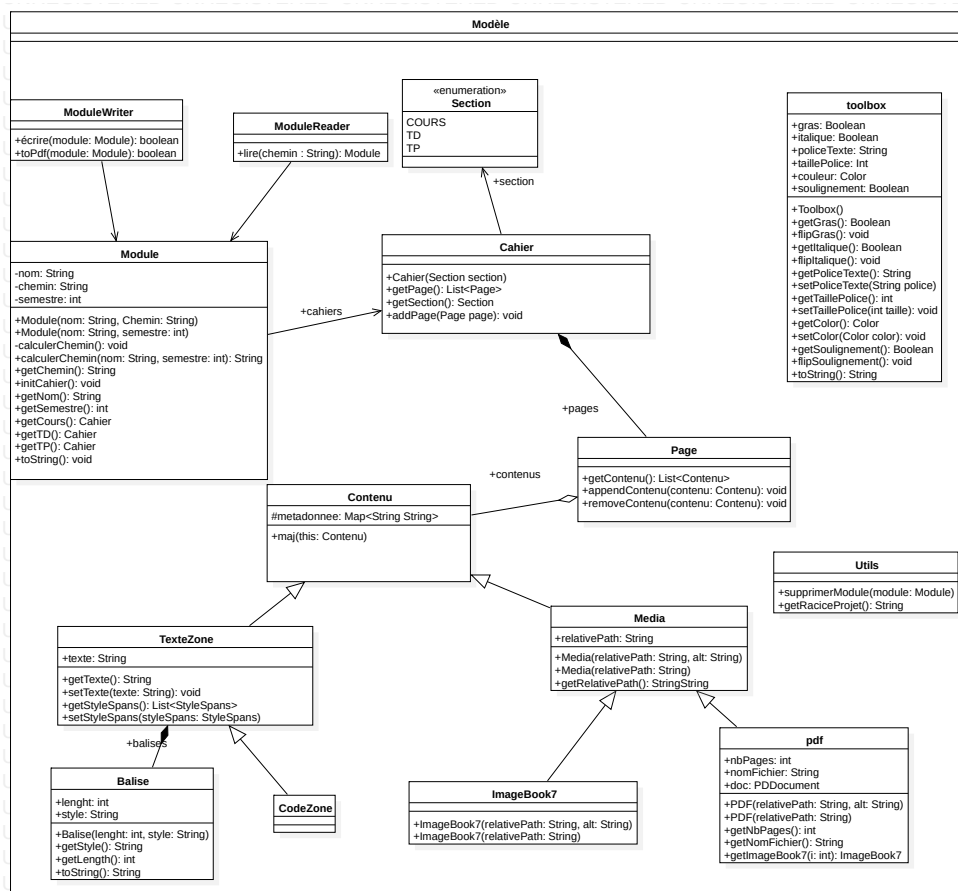


vert: terminée  
Rouge: Annulé

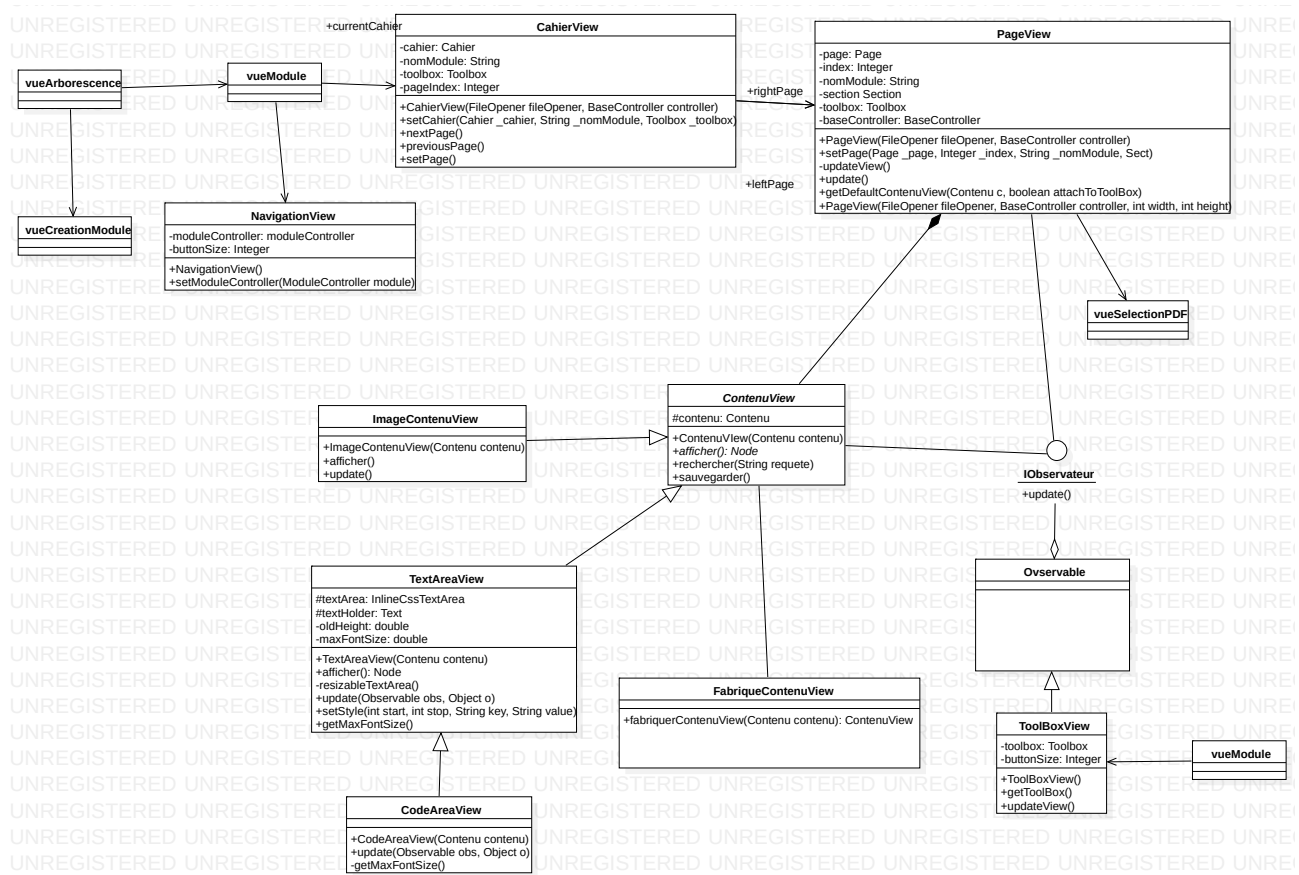
## III/ Présentation technique

### III.1 Modèle/

La version pdf est disponible sur le git [ressources/UMI](#)



### III.2/ Interface graphique



### III.2/ Choix de conception

Traduction du langage naturel en UML.

Choix des patrons de conception Fabrique/Observateur adaptés aux problématiques du projet.

#### *IV/ Travail effectué*

##### *Sprint 3 :*

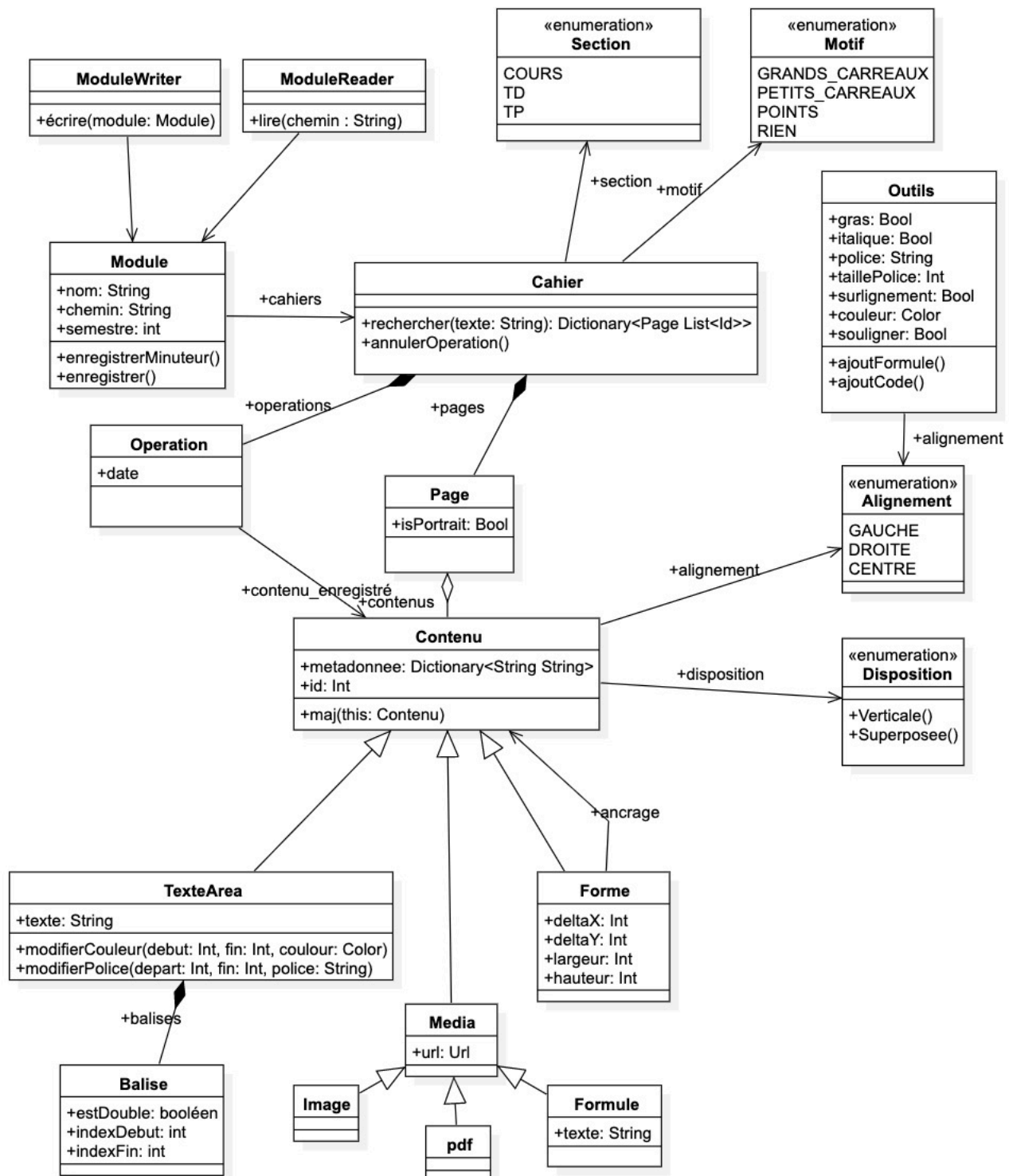
Anna	J'ai travaillé avec Nassim sur le passage du PDF l'extraction de texte et l'ajout d'un Diapo dans le cahier.
Clément	J'ai jonglé entre les différents groupes de pair-programming pour supporter l'ajout des fonctionnalités. J'ai touché à la ToolBox avec Alexandre et Philippe et à l'ajout des PDF avec Nassim et Anna. J'ai également débuggé d'anciennes fonctionnalités et rajouté l'import de .b7 externe et l'ajout de code.
Philippe	Fonctionnalités de la toolbox - Veille sur la classe RichTextFX
Alexandre	Cette semaine j'ai continué à travailler sur des fonctionnalités de la toolbox avec et sans Phillippe sur le plan technique et graphique.
Nassim	Cette semaine je me suis chargé de l'ajout de PDF et en ai profité pour améliorer la gestion des entrées sorties avec des chemins relatifs afin de rendre le programme portable. J'ai également commencé à implémenter la fonctionnalité de recherche que je n'ai malheureusement jamais fini. J'ai cependant rendu le sol fertile à l'implémentation du Ctrl+F.

#### *V/ Conclusion*

*Ce Projet nous a permis de travailler en équipe et de découvrir les différentes notions d'agilité et gestion de projet mais aussi de monter en compétences en programmation objet (Java) et les interfaces graphique (Javafx).*

*Tout au long du projet on a travaillé dans esprit collaboratif où tout le monde à participer et contribuer. Ceci nous a permis de monter rapidement en compétences et d'aboutir à un résultat satisfaisant.*





## Conception Objet: Setup

### Table des matières:

- Setup IDE
- Setup Repository
- Lancer le projet pour la première fois
- Modifier le projet
- Sauvegarder le projet

### Setup IDE: IntelliJ

1. Allez à: <https://www.jetbrains.com/fr-fr/idea/download/#section=linux>
2. Téléchargez: Community Edition
3. Extraire le fichier dans le dossier de votre choix
4. Allez chercher dans ce dossier le fichier `bin/idea.sh` en faire un raccourci (le drag and drop dans la bar des tâches en haut de ubuntu)
5. Double-cliquez sur ce raccourci pour lancer IntelliJ
6. Faire les étapes suivantes

### Setup Repository: Github

1. Créer un compte si vous n'en avez pas
2. Demandez à Clement les droits sur le projet en envoyant l'adresse mail du compte github
3. Ouvrir un terminal
4. Allez dans le dossier qui va contenir le dossier gith
5. lancer: **apt install git**
6. lancer: **git clone <https://github.com/Clement-Torti/book7.git>**  
en fournissant les user info

### Lancer le projet pour la première fois:

1. Cliquez sur votre raccourci d'IntelliJ
2. Ouvrir projet
3. Cherchez le projet dans le dossier git: `/chemin/pour/le/git/book7/book7`
4. Il faut télécharger la librairie javafx. Pour se faire, depuis IntelliJ: File>New>Project
5. Choisir javafx>next. Donnez un nom bidon et Cliquez OK pour télécharger la librairie.
6. Supprimer le projet bidon dans le dossier `n7Student/IdeaProjects`
7. Build le projet en cliquant sur le marteau vert en haut de l'IDE
8. Créer une configuration pour lancer le projet: >"Add configuration" dans la barre du haut
9. "Add new run configuration"
10. L'appellez: "book7"
11. Cliquez sur "Main class" et choisir la classe "Main"
12. Cliquez sur OK
13. Cliquez sur le bouton vert "play" et une fenêtre doit apparaître

### Modifier le projet

1. Ouvrir un terminal et allez dans le dossier github du projet
2. !! lancer: **"git pull"** avant d'effectuer tout changement dans le code projet !!
3. Lancer le projet sur IntelliJ

#### 4. Effectuer les modification de code

Sauvegarder le projet:

**!! Il faut sauvegarder au moins une fois après chaque session de codage !!**

0. Ouvrir un terminal et allez dans le dossier github

1. lancer: **git add .**

2. lancer: **git commit -m "message indiquant les modif effectué"**

3. lancer: **git push**

4. Les changements sont sauvegardés, sauf s'il y a un conflit à résoudre. En cas de conflit, demander à Clément ou Nassim

Lancer le fix par défaut

Add configuration

GitHub Branching workflow:

En debut de sprint:

- Une branche master (contenant le code stable de production maj toutes les fin de sprint)
- creation d'une branche dev à partir du master
- une branche par equipe de dev à partir du master

En fin de sprint:

Pour chaque branche d'équipe:

- Push tout le code sur sa branche
- Merge dev->branche (via l'option git sur intellij)
- Resolution de conflit (via l'outils intellij)
- Merge branche->main (via github -> create pull request)

A la fin:

merge dev->master