

RAPPORT DE PROJECT PLURIDISCIPLINAIRE

Visualisations des Structures de Fichiers et Structures de Données.

MEMBRES DE GROUPE :

1. Houssam BENAYACHE
2. Bilal CHATBI
3. Abdessalam DAI
4. Ouassim HAMDANI
5. Zineddine LOUZANI

Encadreur : Dr. Meroua DAOUDI

Année universitaire : 2021-2022

Table de Contenue

Chapitre 1: Introduction	4
1.1 Présentation du contexte	4
1.2 Définition du problème	4
1.3 La solution proposée	5
Chapitre 2: Présentation Générale du projet	6
2.1 Introduction	6
2.2 Cadre du projet	6
2.2.1 Objectif du projet	6
2.3 Etude préalable	6
2.3.1 Etude de l'existant	7
2.3.2 Critique de l'existant	7
2.4 Les Outils	7
2.4.1 Les Outils De Conception	7
2.4.2 Les Outils de Developpement	8
2.4.3 Les Langages de Programmation et Frameworks	10
2.5 Conclusion	13
Chapitre 3: Analyse des besoins	14
3.1 Introduction	14
3.2 Description du client	14
3.3 Besoins fonctionnles	15
3.4 Besoins Non Fonctionneles	15
3.5 Diagrammes	16
3.5.1 Diagramme de cas d'utilisation	16
3.5.2 Diagramme de Classes	17

3.5.3	Diagramme d'activités	25
3.5.4	Diagramme de Gantt	26
3.6	Conclusion	27
Chapitre 4:	Réalisation	28
4.1	Introduction	28
4.2	Travail réalisée	28
4.3	Logo	40
4.3.1	Signification du logo	40
4.3.2	Processus de création de logo	41
4.3.3	Le choix des couleurs	41
4.4	Fonctionnalités des buttons	42
4.4.1	Page d'accueil :	42
4.4.2	Page de travail dashboard :	42
4.4.3	Page d'édition d'un fichier :	42
4.5	Conclusion	43
Chapitre 5:	Conclusion Général de Projet	44

Remerciement

Nous tenons à remercier notre superviseur d'avoir cru en notre idée et de nous avoir apporté tout le soutien nécessaire pour donner vie à cette excellente idée.

Chapitre 1

Introduction

1.1 Présentation du contexte

Très souvent, les bases en Structures de Données et Structures de Fichiers sont indispensables dans les différents domaines en informatique (structures de la machine, algorithmique et structures de données, systèmes d'informations, bases de données, système d'exploitation, etc.). Donc il est nécessaire de maîtriser et de percevoir l'ensemble de ces éléments et concepts qui seront des pré-requis dans la vie professionnelle.

Le module SFSD est enseigné au deuxième année classe préparatoire aux Écoles Supérieures d'Informatique avec un coefficient élevé et aussi un module fixe dans le concours d'accès au cycle supérieur, ce qui montre l'importance de ce module fondamentale.

1.2 Définition du problème

On a remarqué durant dans la période d'étude du module que la plupart des étudiants ont trouvé des difficultés majeurs pour comprendre et stabiliser quelques concepts (les buffers, les blocs, etc.) parce que les étudiants n'arrive pas à visualiser ce qui se passe à l'intérieur de l'ordinateur, et c'est pour cela les étudiants ont commencé à sous-estimer l'importance du module SFSD, malgré tous les efforts fait par les enseignants. Après une étude approfondie des préoccupations des étudiants avec les chargés de ce module, on est arrivé à définir le problème exact : les concepts de bases sont abstraits.

1.3 La solution proposée

Pour une bonne préparation au concours d'accès au cycle supérieur, et pour bien aider les prochaines promotions, nous proposons la création d'une application Web pour visualiser ces concepts-là, les différents algorithmes, etc.

Chapitre 2

Présentation Générale du projet

2.1 Introduction

Dans ce chapitre, nous allons commencer en un premier lieu par explication du cadre du projet. Puis, faire notre étude préalable qui contient l'étude, la critique de l'existant et nos solutions proposées.

2.2 Cadre du projet

Ce projet est un travail de cinq étudiants dans le cadre du projet pluri-disciplinaire ayant comme sujet « Conception et développement d'un site web pour la visualisation du module Structures Fichiers et Structures de Données (SFSD) »

2.2.1 Objectif du projet

Dans ce projet, on s'est fait demander de faire la conception puis le développement d'une application Web qui facilite aux étudiants de l'Ecole Supérieure en Sciences et Technologies de l'Informatique et Numérique (ESTIN) de comprendre comment se déroulent les différents algorithmes du module SFSD. En effet, elle offre aux enseignants et étudiants la possibilité de pratiquer le module par création des fichiers au sein de leurs pages d'essai dans leurs comptes privés et être capable de partager leurs travaux et leurs documents avec les autres étudiants.

2.3 Etude préalable

Dans cette partie du rapport, on va entamer l'étude de l'existant ,puis on va le critiquer pour déduire les solutions qui seront proposées au sein

de notre projet.

2.3.1 Etude de l'existant

Visual Learning⁷ : C'est une plateforme en ligne créée pour aider les étudiants à comprendre le module SFSD et offre à ses visiteurs de consulter les cours ainsi que la visualisation des différents algorithmes du module.

L'enseignement de ce genre des modules n'est pas facile vu les difficultés de compréhension qui se trouvent au déroulement des algorithmes.

L'outil dont on a déjà parlé propose des solutions pour faciliter l'opération de compréhension. Il essaie d'offrir une solution innovante et efficace. Dans les parties suivantes, on va critiquer ce genre de solutions et on va en proposer d'autres

2.3.2 Critique de l'existant

Certes les outils cités précédemment sont utiles, connus et serviables, mais ils souffrent de quelques limites. Nous allons maintenant les décrire :

- Manque d'information sur le coût d'opération de chaque algorithme.
- Le nom de site ne représente pas son contenu.
- le site se charge lentement.
- Le site ne donne pas importance à ce qui se passe au niveau de la mémoire ce qui est l'un des buts de module.
- Le site est mal conçu.
- L'impossibilité de sauvegarder le contenu des fichiers en base de données.

2.4 Les Outils

2.4.1 Les Outils De Conception

Le conception des applications Web se concentre sur l'apparence, la mise en page, les couleurs, la police et les images utilisées , avec une façon qui permettra à l'utilisateur d'avoir la meilleur expérience, L'outil que nous avons décidé à utiliser sont :

Figma :

L'outil a été conçu dans l'esprit d'un design system, c'est-à-dire un système permettant de réunir tous les éléments visuels, graphiques...

7. <https://visuellearning.netlify.app/>



d'un projet de développement. Figma se présente sous la forme d'une solution en ligne de design d'interfaces collaboratif et dynamique, qui peut être alimentée et partagée au sein d'une équipe de designers.

draw.io :



est un logiciel qui permet de créer des diagrammes et des organigrammes (UML).

2.4.2 Les Outils de Développement

Pour un meilleur workflow de développement de notre application on utilise :

Pycharm :



PyCharm est un environnement de développement intégré utilisé pour programmer en Python. Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django.

Webstorm :

est un environnement de développement intégré pour JavaScript et les technologies connexes. Comme les autres IDE de JetBrains, il rend



l’expérience de développement plus conviviale, en automatisant les tâches répétitives et aidant à gérer les tâches complexes avec facilité.

Git :



c’ est de loin le système de contrôle de version le plus largement utilisé aujourd’hui C’est la solution idéale pour la plupart des équipes de développement actuelles.

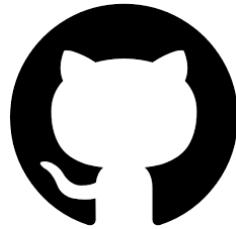
Gmail :



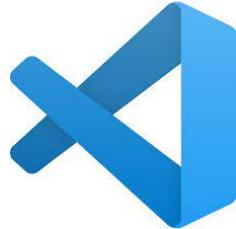
Gmail est une boîte de d’envoi et de réception d’e-mails développée par Google en 2004. Ce service de messagerie est gratuit et à ce jour l’un des plus simple, complet et sécurisé.

GitHub :

GitHub est un site web et un service de cloud qui aide les développeurs à stocker et à gérer leur code, ainsi qu’à suivre et contrôler les modifications qui lui sont apportées.



VSCode :



est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'autocomplétion, la coloration syntaxique, le débogage, et les commandes git.

2.4.3 Les Langages de Programmation et Frameworks

Les Langages du web

HTML :



c'est le langage de balisage conçu pour représenter les pages web. Ce langage permet : d'écrire de l'hypertexte, d'où son nom, de structurer sémantiquement la page, de mettre en forme le contenu.

CSS :

donne la possibilité de styler les éléments html en ajoutant des bordures de couleurs de polices, et des animations interactives...



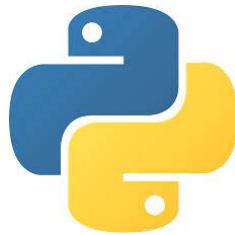
Les Langages de programmations

JavaScript :



JavaScript est un langage de programmation de scripts principalement employé principalement dans les pages web interactives et à ce titre est une partie essentielle des applications web.

Python :



est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.

Les Frameworks

Tailwind CSS :

est un CSS framework permettant aux développeurs de personnaliser totalement et simplement le design de leur application ou de leur site web. Avec ce framework, il est possible de créer un design d'interface au sein même du fichier HTML.



D3 :



est une bibliothèque graphique JavaScript qui permet l'affichage de données numériques sous une forme graphique et dynamique. Il s'agit d'un outil important pour la conformation aux normes W3C qui utilise les technologies courantes SVG, JavaScript et CSS pour la visualisation de données.

Django :



Django est un framework Python de haut niveau, permettant un développement rapide de sites internet, sécurisés, et maintenables. Django prend en charge la plupart des tracas du développement web.

JQuery :

jQuery est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web.



SQLlite3 :



SQLite est une bibliothèque écrite en langage C qui propose un moteur de base de données relationnelle accessible par le langage SQL.

NodeJs :



est un environnement d'exécution single-thread, open-source et multi-plateforme permettant de créer des applications rapides et évolutives côté serveur et en réseau. Il fonctionne avec le moteur d'exécution JavaScript V8 et utilise une architecture d'E / S non bloquante et pilotée par les événements, ce qui le rend efficace et adapté aux applications en temps réel.

2.5 Conclusion

Dans ce chapitre, on a commencé par une présentation du cadre du projet dans laquelle on a décrit son objectif. Puis, on est passé par une étude préalable qui contient l'étude et la critique de l'existant accompagné de quelques solutions proposées. En ce qui concerne le chapitre suivant, on s'intéressera à l'analyse des besoins fonctionnels et non fonctionnels et à la spécification de ces besoins.

Chapitre 3

Analyse des besoins

3.1 Introduction

Ce chapitre sera divisé en deux parties : la première est une analyse des besoins dans laquelle on va citer en un premier lieu les différents acteurs intervenants dans l'application Web. En second lieu, on va décrire d'une façon détaillée nos besoins fonctionnels et non fonctionnels. Dans la deuxième partie on s'intéresserait à la spécification et de ces besoins à travers les diagrammes de cas d'utilisation de l'UML.

3.2 Description du client

Madame Daoudi Meroua, enseignante de module structures de fichiers et structure des données, elle est responsable de ce module au sein de l'établissement ESTIN (École Supérieure en Sciences et Technologies de l'Informatique et du Numérique Amizour-Bejaia). Elle est aussi le responsable de présentation des cours de ce module et le chargé des travaux dirigés pour certaines groupes. Elle suit aussi la formation des jeunes stagiaires en relation avec ce module. Madame Daoudi cherche un outil simple et efficace pour l'explication des concepts du module, et elle cherche une manière de visualiser les différentes opérations qui se passent au niveau du mémoire central et la mémoire secondaire, telles que la lecture et l'écriture, la suppression, la recherche d'une donnée, dans les différents types de fichiers TOF, TÔF, LOF, LÔF, etc.... (Veuillez consulter le chapitre d'analyse des besoins). Madame Daoudi cherche aussi à simplifier la tâche d'enseignement aux autres enseignants de module SFSD.

3.3 Besoins fonctionnels

Notre application doit permettre aux différents acteurs de s'inscrire, s'authentifier afin d'accéder à leurs données. Ils peuvent ainsi consulter leurs profils et modifier leurs informations. Pour le reste des besoins fonctionnels, ils varient d'un acteur à l'autre comme suite :

○ Client :

- Il est caractérisé par son nom, prénom, email, mot de passe, et son groupe.
- Les informations d'un client doivent être accessibles pour qu'il puisse les consulter et les modifier.
- Il peut consulter n'importe quelle page et suivre ses anciens travaux.
- Il peut construire des fichiers, les sauvegarder, les partager ou les supprimer.

○ Représentant de l'application web :

- C'est un utilisateur inscrit comme étant un client, affecté par le super administrateur à l'application Web (dans notre cas, il est le même avec le super administrateur).
- Il peut gérer les informations de l'application Web.

○ Super administrateur :

- C'est le seul qui peut gérer les clients (étudiants et enseignants)..
- L'administrateur de Visual SFSD est en même temps son représentant.

3.4 Besoins Non Fonctionnelles

Les besoins non fonctionnels forment l'ensemble des contraintes techniques auxquelles notre application doit répondre :

➢ Rapidité :

le temps de traitement exigé par l'application doit être court pour garantir un accès admissible.

➢ Sécurité :

tout type d'accès à la donnée est authentifié.

➢ Performance :

les langages de programmation et technologies utilisés sont perfor-

mants.

➤ **Maintenabilité et convivialité :**

application Web est dédié vers deux catégories seulement, pour les étudiant et les enseignants

➤ **Portabilité :**

aapplication Web est accessible par tous machines connectées à l'internet.

3.5 Diagrammes

3.5.1 Diagramme de cas d'utilisation

Simple utilisateur :

L'utilisateur authentifie et accède au site grâce à les coordonnées de son compte si il est inscrit déjà sinon il sera transférer vers la procédure d'enregistrement.

L'utilisateur a le choix de :

1. Créer un nouveau fichier :

Indexé : dense ou non dense.

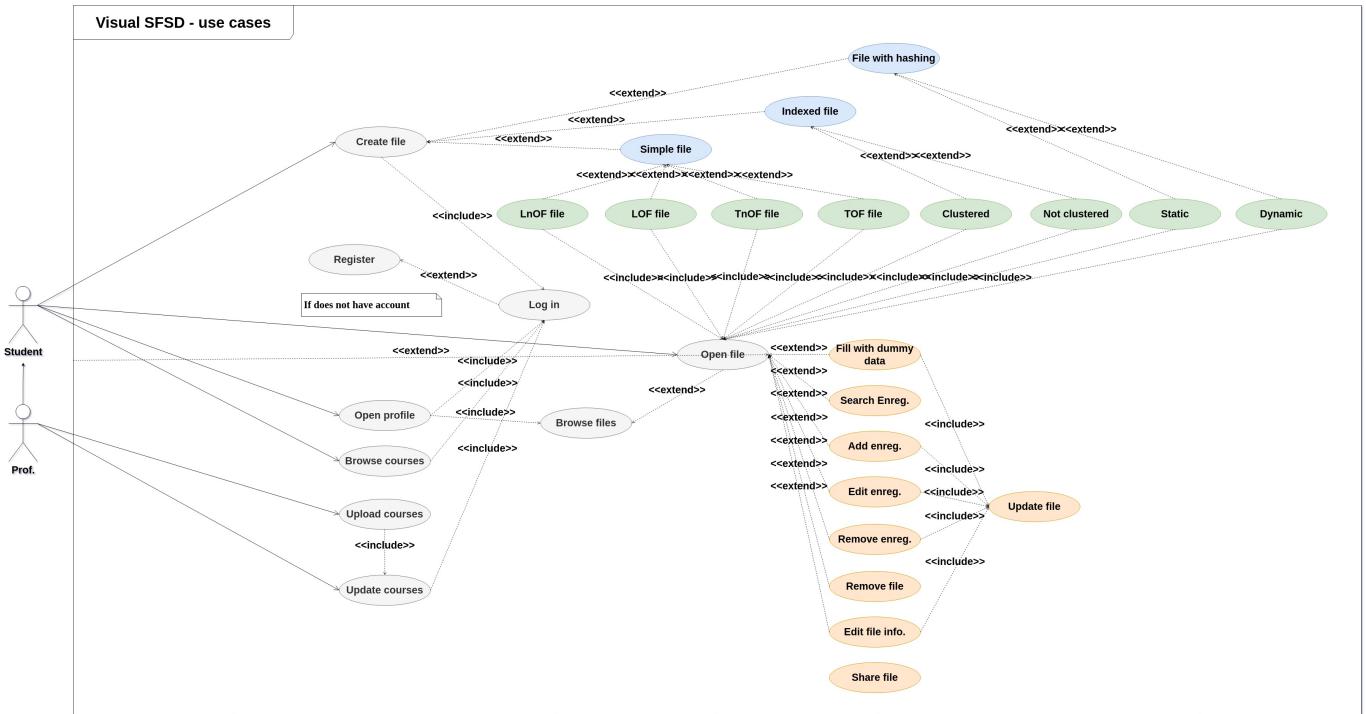
Séquentiel : puis choisir son type de fichier ; TOF, LOF ...

Hashage : statique.

2. Ouvrir un fichier :

(a) Modification de fichier en utilisant les fonctions souhaitées.

3. Ouvrir le profil et la possibilité de le modifier, et consulter les profils des autres (enseignants et étudiants).



3.5.2 Diagramme de Classes

Classe Block

○ Attributs :

- **enregs** : tableau d'enregistrements (de type Enreg).
- **nb** : le nombre d'enregistrements dans le bloc.
- **blockAddress** : l'adresse physique (réelle) du bloc dans la mémoire secondaire

Classe Enreg

○ Attributs :

- **key** : la clé unique de l'enregistrement, elle est de type entier
- **field1** : le champ 1 de type chaîne de caractères.
- **field2** : le champ 2 de type chaîne de caractères.
- **removed** : un booléen pour indiquer si l'enregistrement est supprimé logiquement.

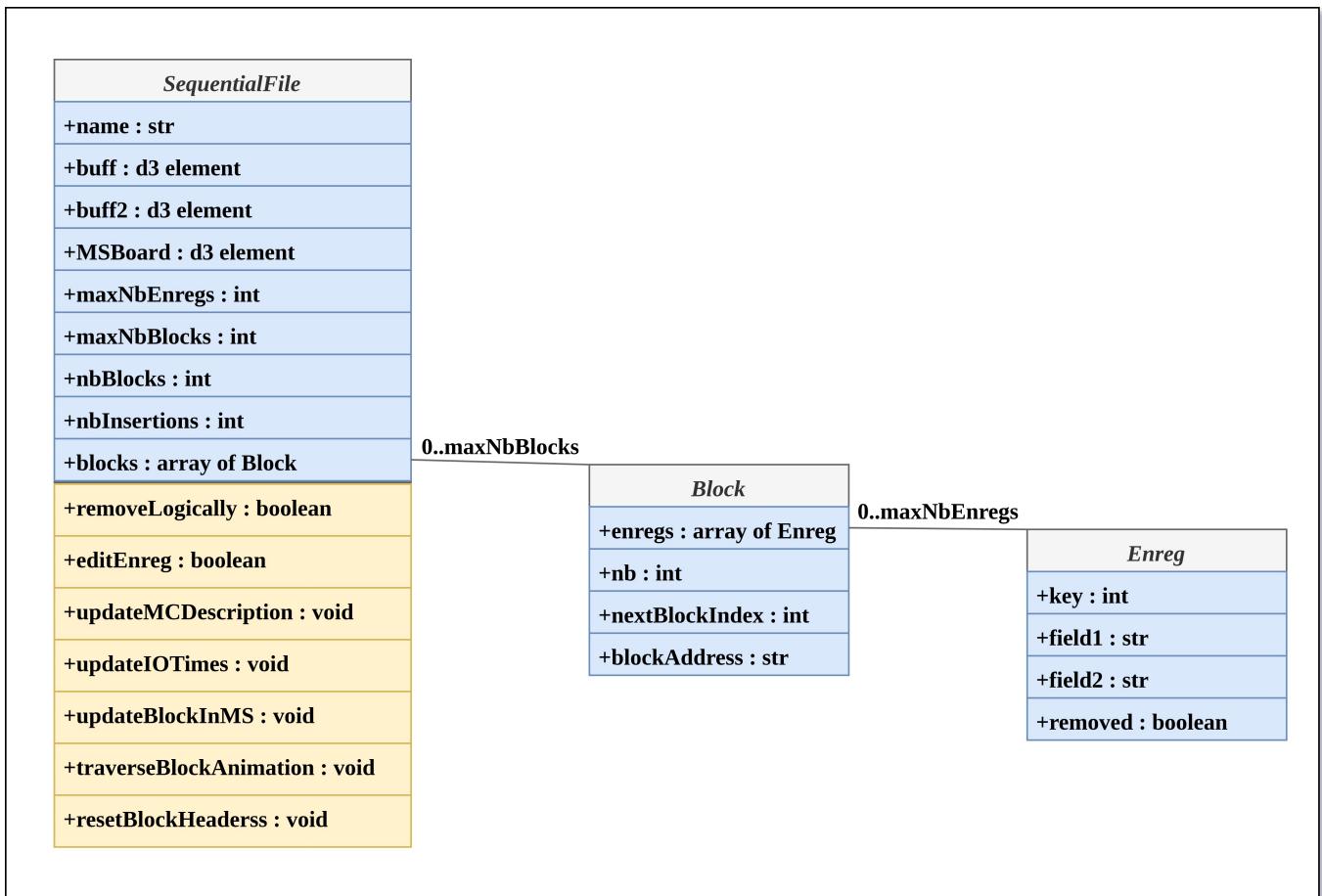
Classe sequentialFile

○ Attributs :

- **name** : le nom du fichier.
- **Blocks** : un tableau de blocs (de type Block).
- **maxNbEnregs** : le nombre maximum d'enregistrements dans un bloc.
- **maxNbBlocks** : le nombre maximum de blocs dans le fichier.
- **nbInsertions** : le nombre d'insertion dans le fichier.
- **blocks** : un tableau de blocs (de type Block).
- **buff** : un élément HTML sélectionné par d3.
- **buff2** : un élément HTML sélectionné par d3.
- **MSBoarad** : un élément HTML sélectionné par d3.

○ Méthodes :

- **rempveLogically** : supprimer un enregistrement logiquement.
- **editEnreg** : modifier un enregistrement dans le fichier.
- **updateMCDescription** : mettre à jour la description de la mémoire centrale.
- **updateIOTimes** : mettre à jour le nombre de d'operation de la lecture et d'écriture.
- **updateBlockInMs** : mettre à jour le block dans la mémoire secondaire.
- **taraverseBlockAnimation** : Ajouter l'animation au block.
- **resetBlockHeaders** : réinitialiser l'affichage de block.



Classe TableFile

○ Méthodes :

- **reset** : réinitialiser les données dans le fichier.
- **getJsonFormat** : transforme vers la forme de JavaScript Object Notation.
- **createBordsBoard** : permettre de créer les éléments HTML pour visualiser le contenu du fichier.
- **isInsertionAllowed** : autorisation de l'insertion dans le fichier.
- **updateBufferEelement** : mettre à jour l'élément buffer.
- **setBlockAdress** : définir l'adresse du bloc.
- **createNewBlockInBuff** : créer un nouveau bloc dans le buffer.

Classe TOF

Cette classe hérite la classe TableFile

○ Méthodes :

- **search** : faire le recherche d'un enregistrement dans le fichier selon

la clé.

- **insert** : insérer un nouveau enregistrement dans le fichier.
- **removeLogically** : supprimer un enregistrement logiquement.
- **removePhysically** : supprimer un enregistrement physiquement.
- **editEnreg** : modifier un enregistrement dans le fichier

Classe TnOF

Cette classe hérite la classe TableFile

○ Méthodes :

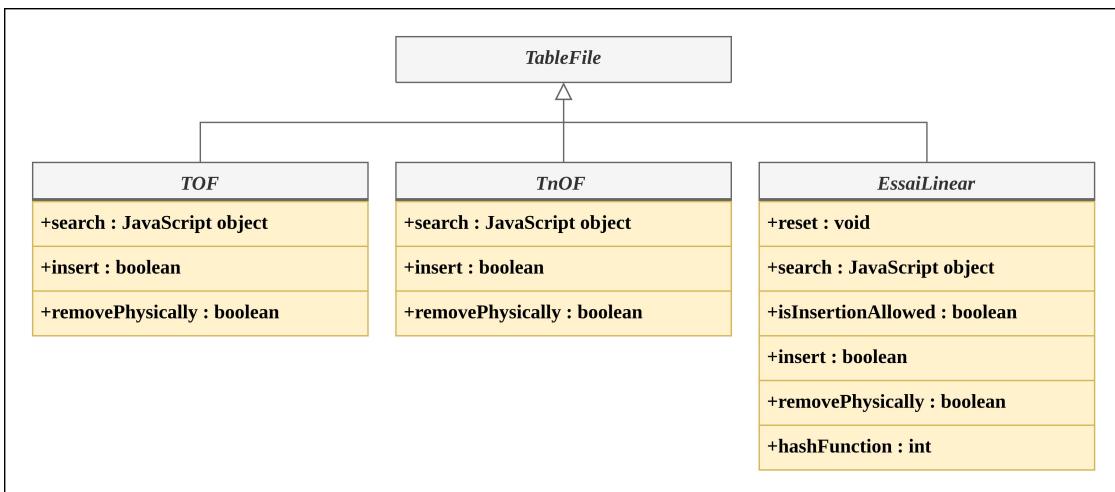
- **search** : faire le recherche d'un enregistrement dans le fichier selon la clé.
- **insert** : insérer un nouveau enregistrement dans le fichier.
- **removePhysically** : supprimer un enregistrement physiquement.

Classe EssaiLinear

Cette classe hérite la classe TableFile

○ Méthodes :

- **search** : faire le recherche d'un enregistrement dans le fichier selon la clé.
- **insert** : insérer un nouveau enregistrement dans le fichier.
- **removePhysically** : supprimer un enregistrement physiquement.
- **isInsrtionAllowed** : autorisation de l'insertion dans le fichier.
- **hashFunction** : la fonction de hashage.



Classe IndexedFile

cette classe herite la classe tableFile

○ Attributs :

- **maxIndex** : l'indice max.
- **indexTableHtml** : un élément HTML sélectionné par d3.
- **indexTable** : un tableau de IndexCouple
- **indexTableConatiner** : un élément HTML sélectionné par d3.

○ Méthodes :

- **reset** : réinitialiser les données d'un fichier
- **createIndexTableDOM** : générer le html représentant la table d'index.
- **scrollToCell** : défiler jusqu'à la cellule
- **getJsonFromat** : transforme vers la forme de JavaScript Object Notation.

Classe ListFile

cette classe herite la classe SequentialFile

○ Attributs :

- **headIndex** : l'indice de tête. .
- **tailIndex** : l'indice de queue .

○ Méthodes :

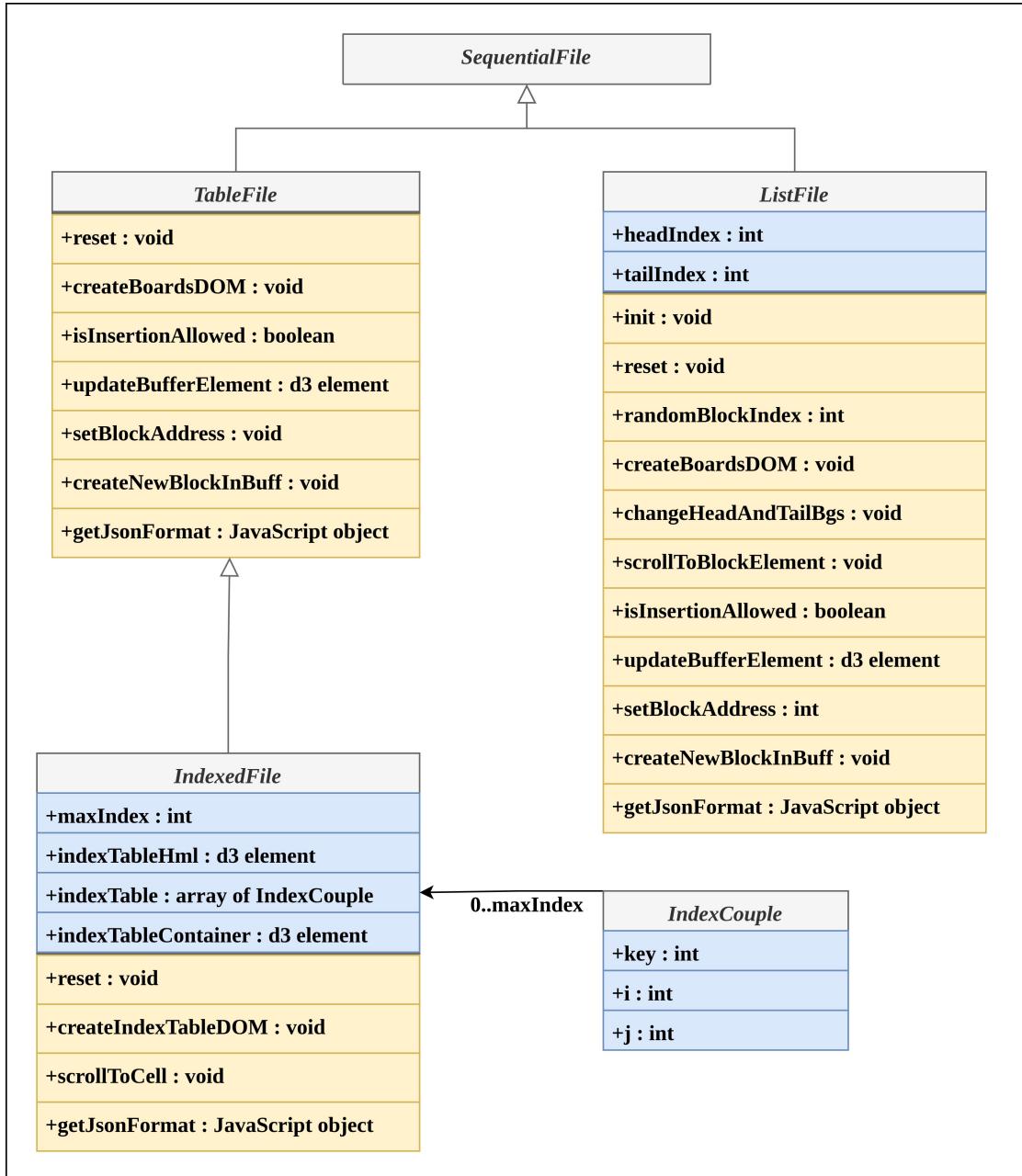
- **init** : initialiser les données d'un fichier
- **reset** : réinitialiser les données d'un fichier
- **randomBlockIndex** : renvoie un index aléatoire pour le block suivant.
- **createBoardsDOM** : permettre de créer les éléments HTML pour visualiser le contenu du fichier.
- **changeHeadAndTailBgs** : changer le fond des deux HTML elements.
- **scrollToBlockElement** : faites défiler jusqu'à l'élément de block.
- **isInsertionAllowed** : autorisation de l'insertion dans le fichier.
- **updateBufferEelement** : mettre à jour l'élément buffer
- **createNewBlockInBuff** : créer un nouveau bloc dans le buffer.

- **getJsonFormat** : transforme vers la forme de JavaScript Object Notation.

class IndexCouple

○ Attributs :

- **key** : la clé du couple d'indices.
- **i** : représentant l'indice de block.
- **j** : représentant l'indice de l'enregistrement dans le block.



Classe LnOF

Cette classe hérite la classe ListFile

○ Méthodes :

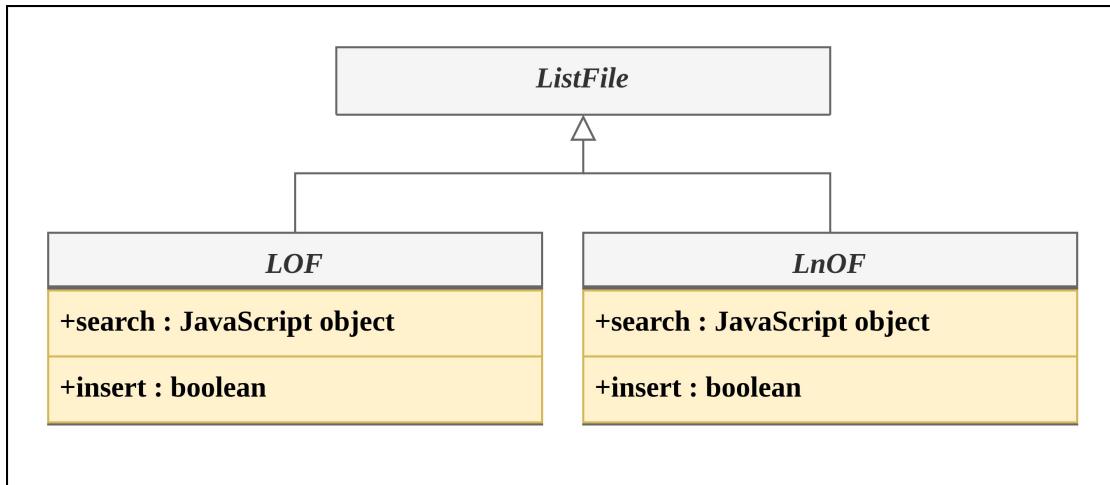
- **search** : faire la recherche d'un enregistrement dans le fichier selon la clé.
- **insert** : insérer un nouveau enregistrement dans le fichier.

Classe LOF

Cette classe hérite la classe ListeFile

○ Méthodes :

- **search** : faire la recherche d'un enregistrement dans le fichier selon la clé.
- **insert** : insérer un nouveau enregistrement dans le fichier.



Classe Clustered

Cette classe hérite la classe IndexedFile

○ Méthodes :

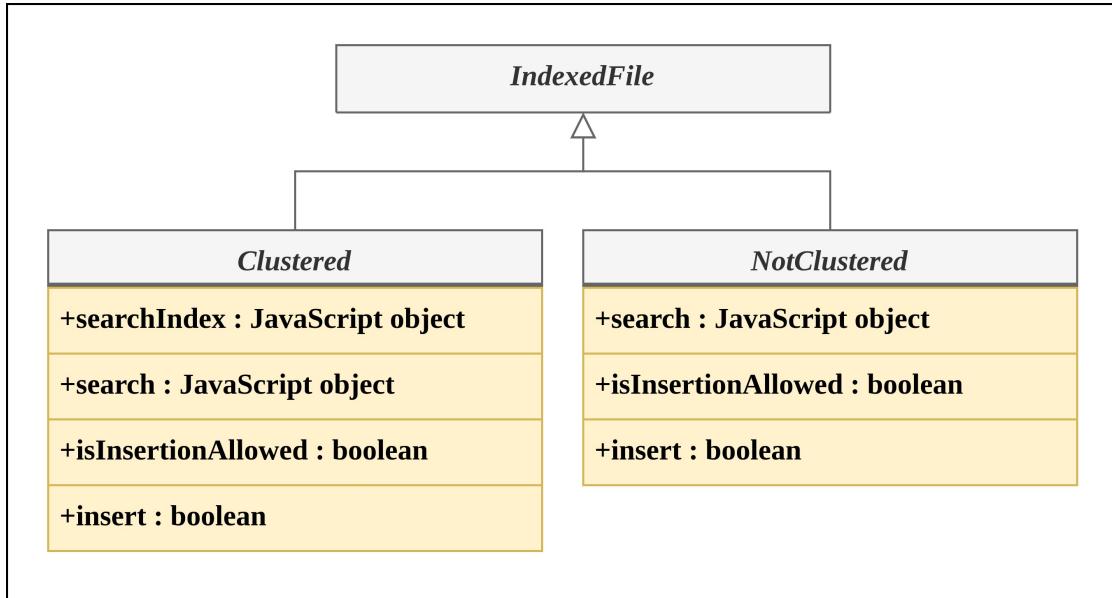
- **searchIndex** : faire la recherche d'un enregistrement dans la table d'index.
- **seach** : faire la recherche d'un enregistrement dans le fichier.
- **isInsertionAllowed** : l'autorisation de l'insertion dans le fichier.
- **insert** : insérer un nouveau enregistrement dans le fichier.

Classe NotClustered

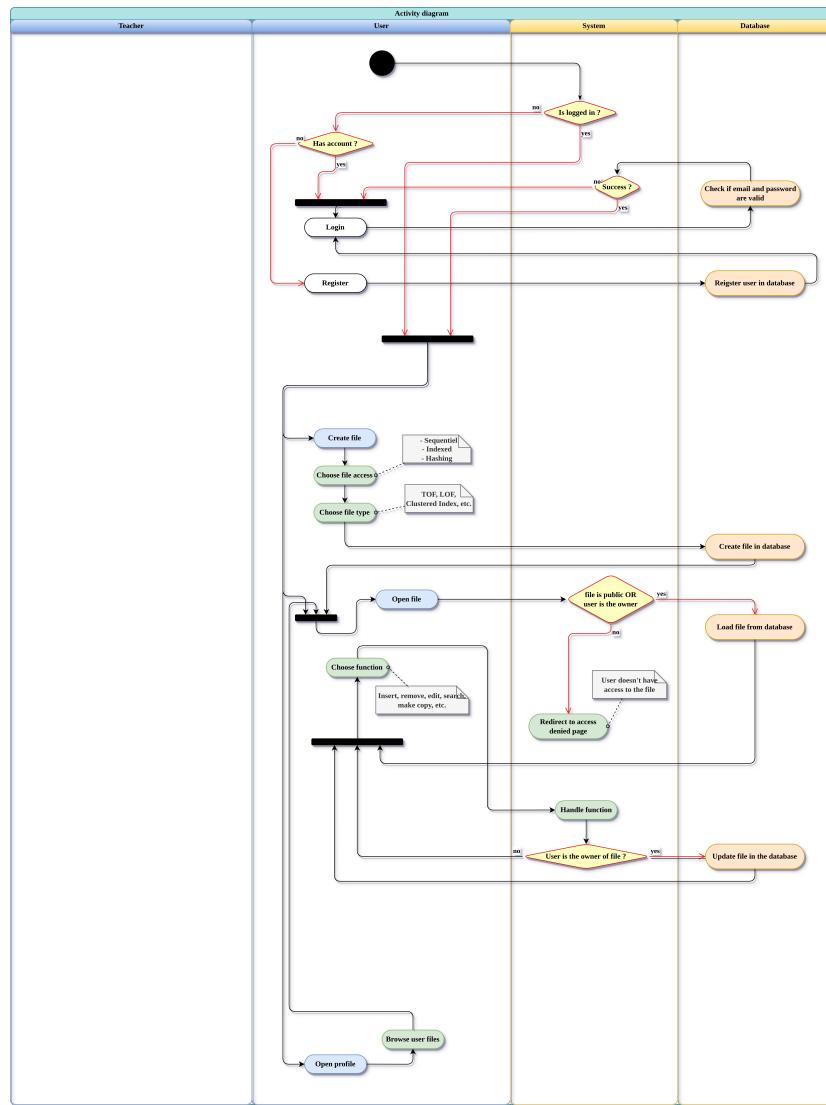
Cette classe hérite la classe IndexedFile

○ Méthodes :

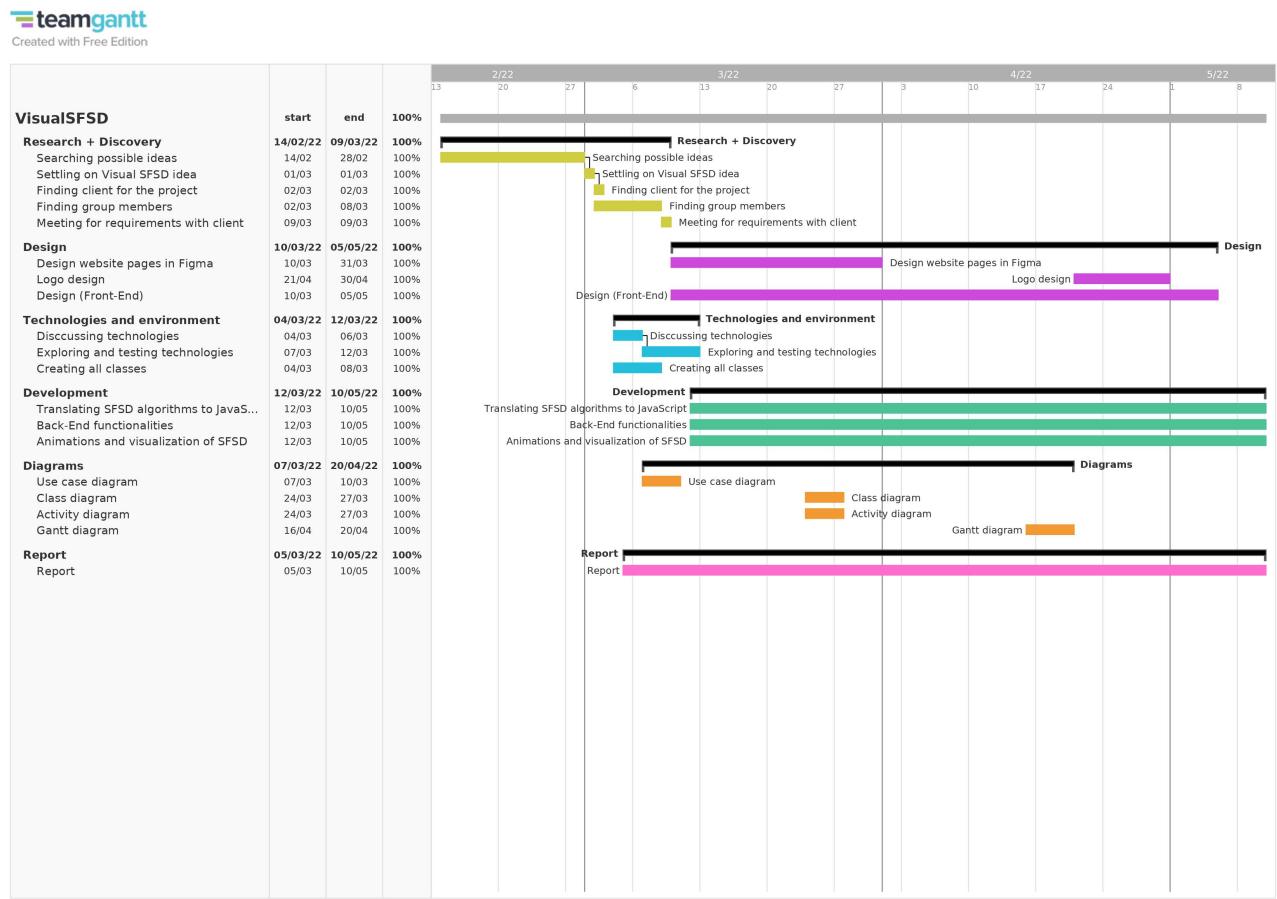
- **search** : faire la recherche d'un enregistrement dans le fichier selon la clé.
- **isInsertionAllowed** : l'autorisation de l'insertion dans le fichier.
- **insert** : insérer un nouveau enregistrement dans le fichier.



3.5.3 Diagramme d'activités



3.5.4 Diagramme de Gantt



3.6 Conclusion

Dans ce chapitre nous avons commencé par une analyse des besoins, on a cité les besoins fonctionnels et non fonctionnels. Puis nous avons décrit la spécification de ces besoins à travers le diagramme de cas d'utilisation. nous avons également introduit toutes les classes que nous avons utilisées pour implémenter l'application.

Chapitre 4

Réalisation

4.1 Introduction

Dans ce chapitre nous allons expliquer comment le site fonctionne en générale, ses différents boutons qui simplifient le processus, la signification du logo et ses couleurs.

4.2 Travail réalisée

Nous allons ici décrire notre travail réalisé et nous allons expliquer le scenario d'exécution à l'aide des captures d'écrans. La figure ci-dessous montre l'interface du site Visual SFSD.

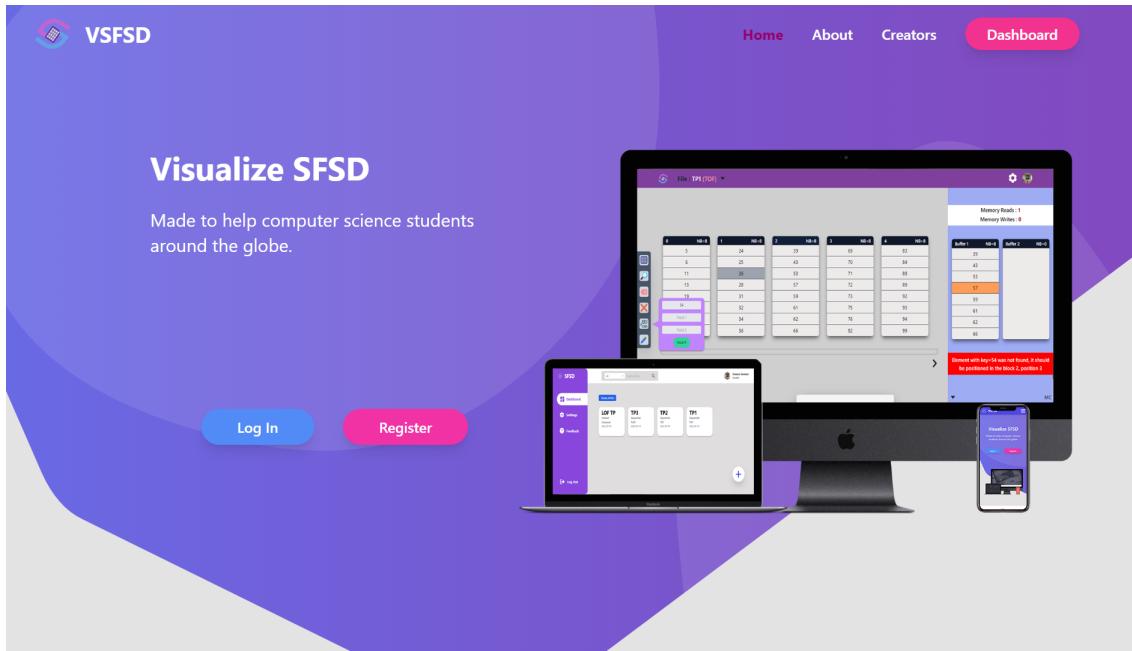


FIGURE 4.1 – page d'accueil.

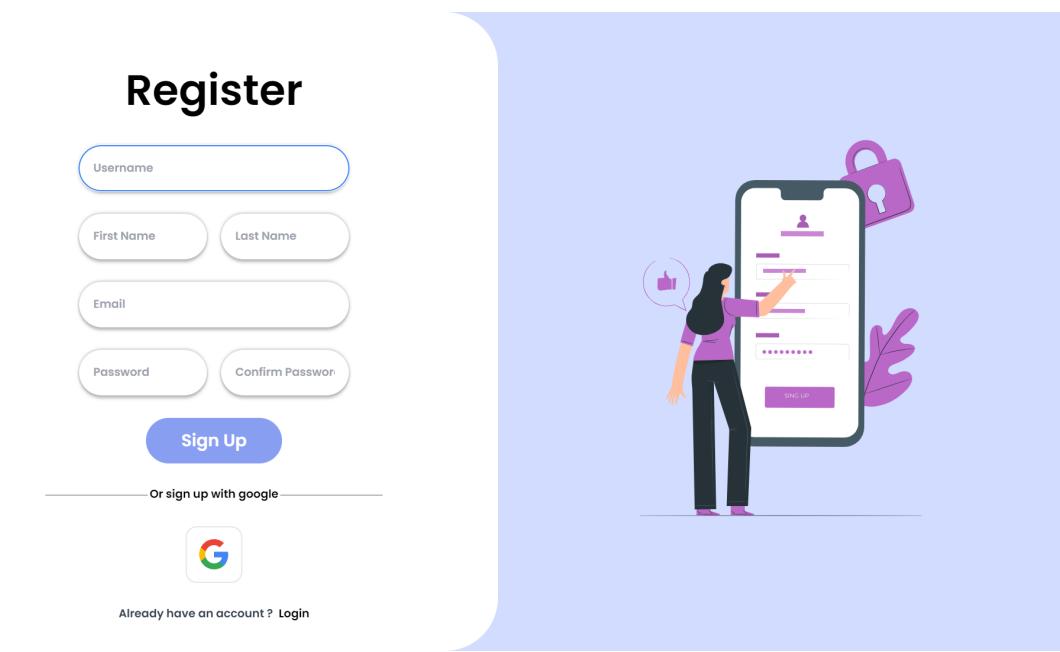


FIGURE 4.2 – page d’inscription.

Sachant que tout type accès doit être authentifié.

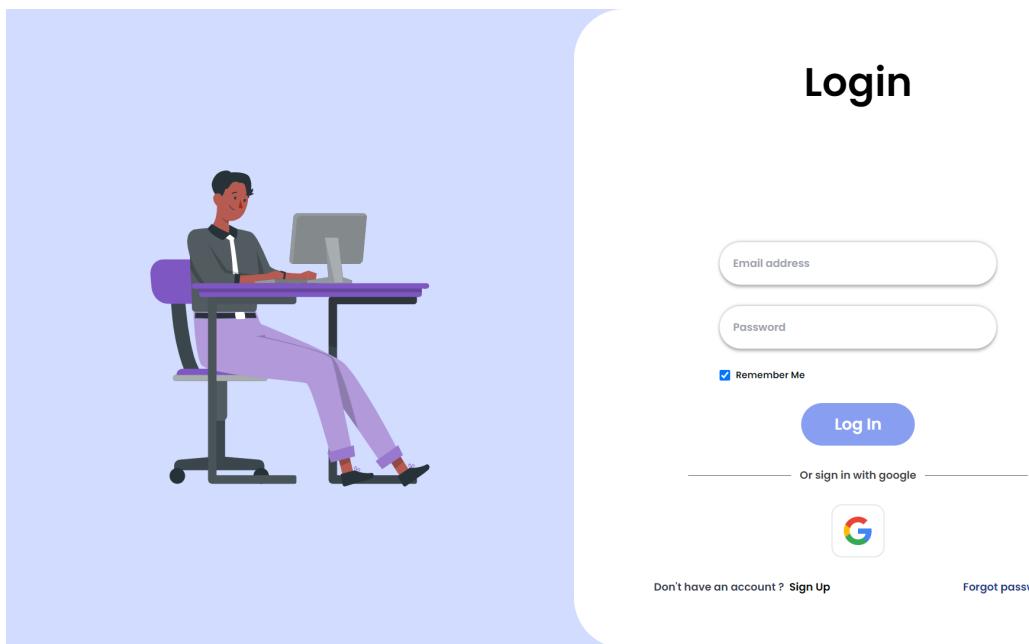


FIGURE 4.3 – page d'authentification.

Après avoir été authentifiée, un étudiant peut consulter ses fichiers et les modifier. La figure 4.3 montre la page du Dashboard que l'étudiant ou l'enseignant peut voir.

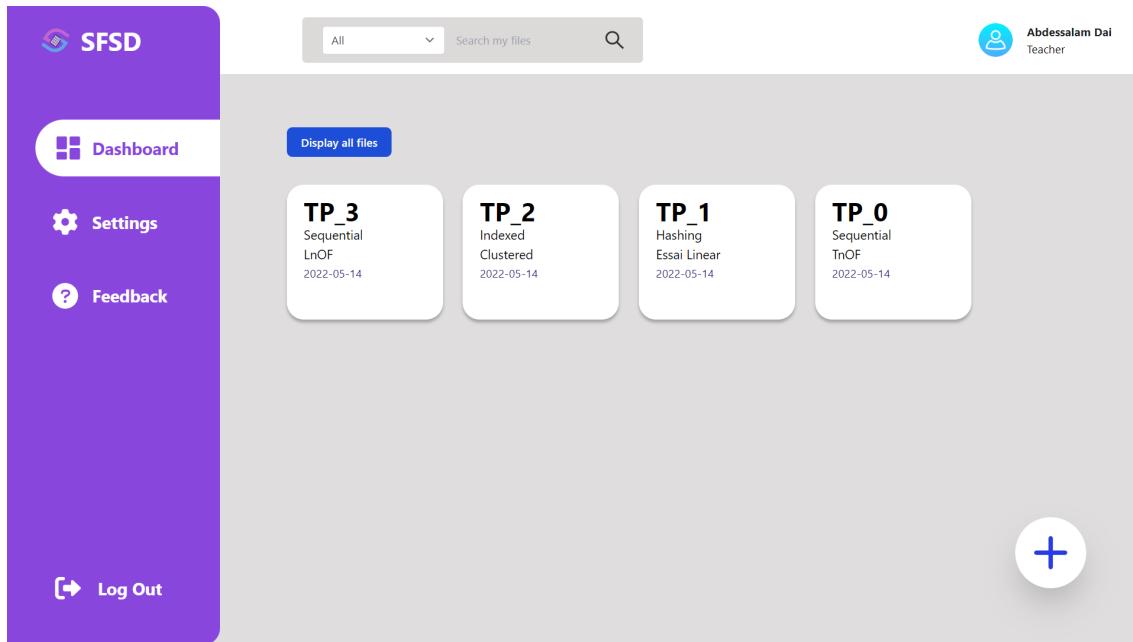


FIGURE 4.4 – tableau de bord.

Sur une clique, l'utilisateur va choisir le nom du fichier à créer.

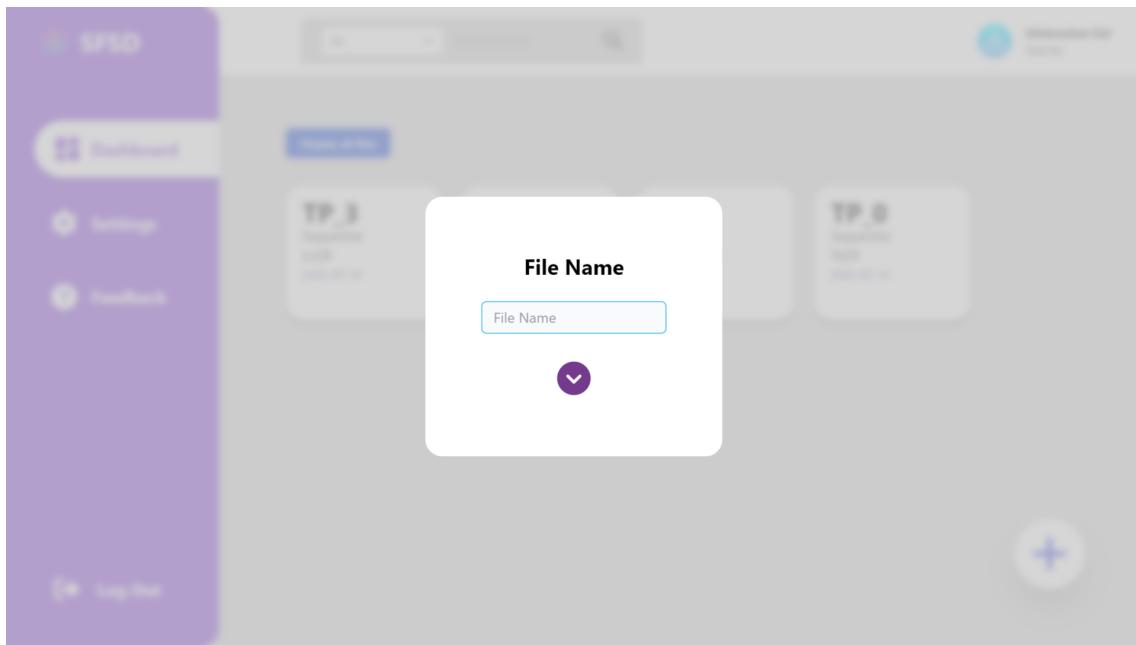


FIGURE 4.5 – Nom de fichier.

l'utilisateur choisira une option pour la structure de fichier

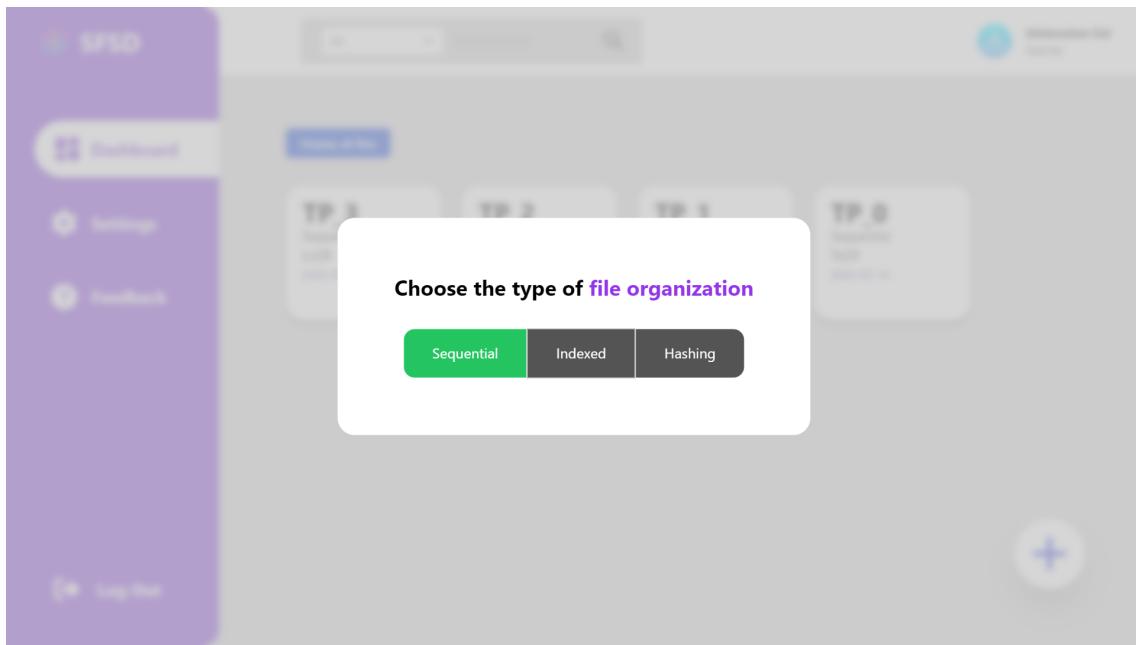


FIGURE 4.6 – structure de fichier.

l'utilisateur choisira une option pour le type de fichier

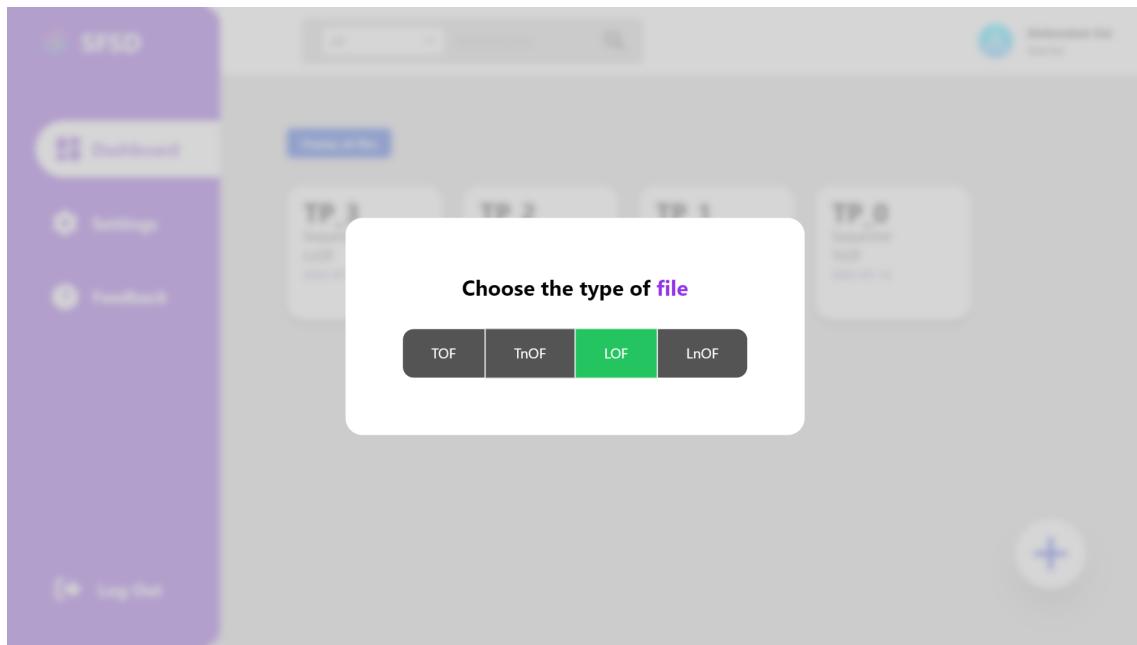


FIGURE 4.7 – type de fichier.

l'utilisateur choisira la configuration de fichier.

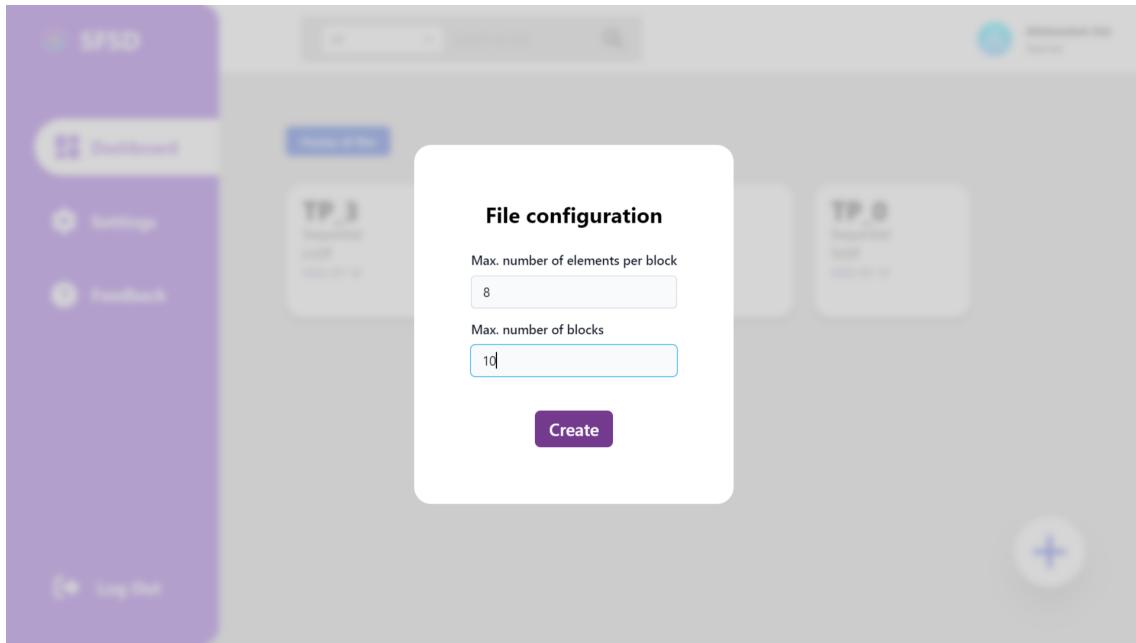


FIGURE 4.8 – configuration de fichier.

Après des simples clicks, le site va afficher l'interface où l'utilisateur peut choisir plusieurs fonctions dans la création du fichier avec la vitesse de l'animation.

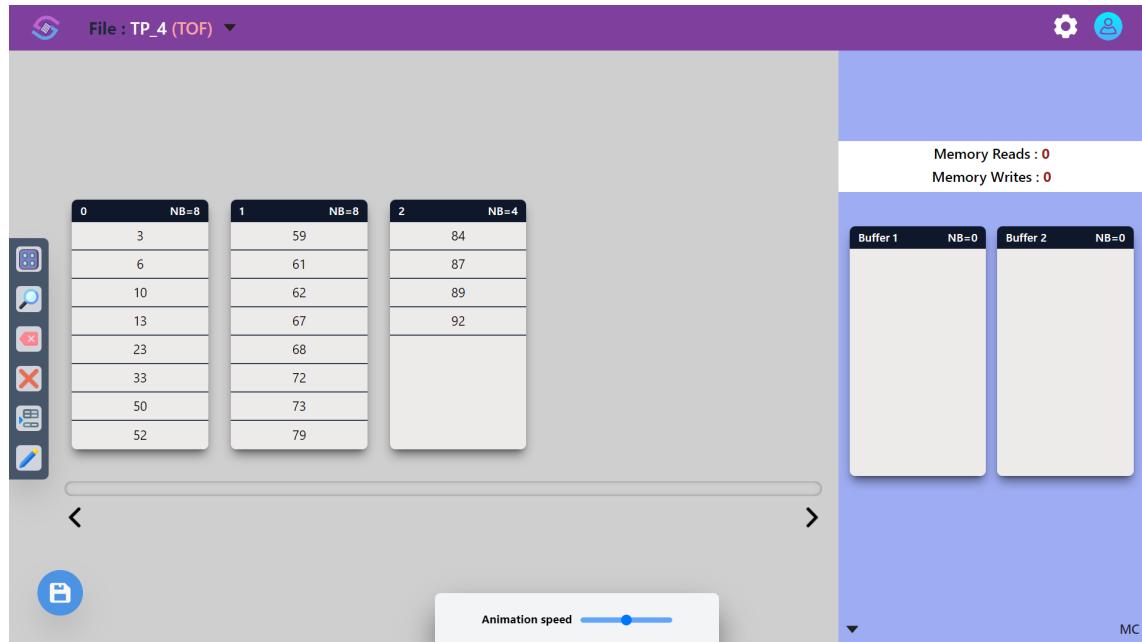


FIGURE 4.9 – l'espace de travail.

L'utilisateur peut changer ses données personnelles.

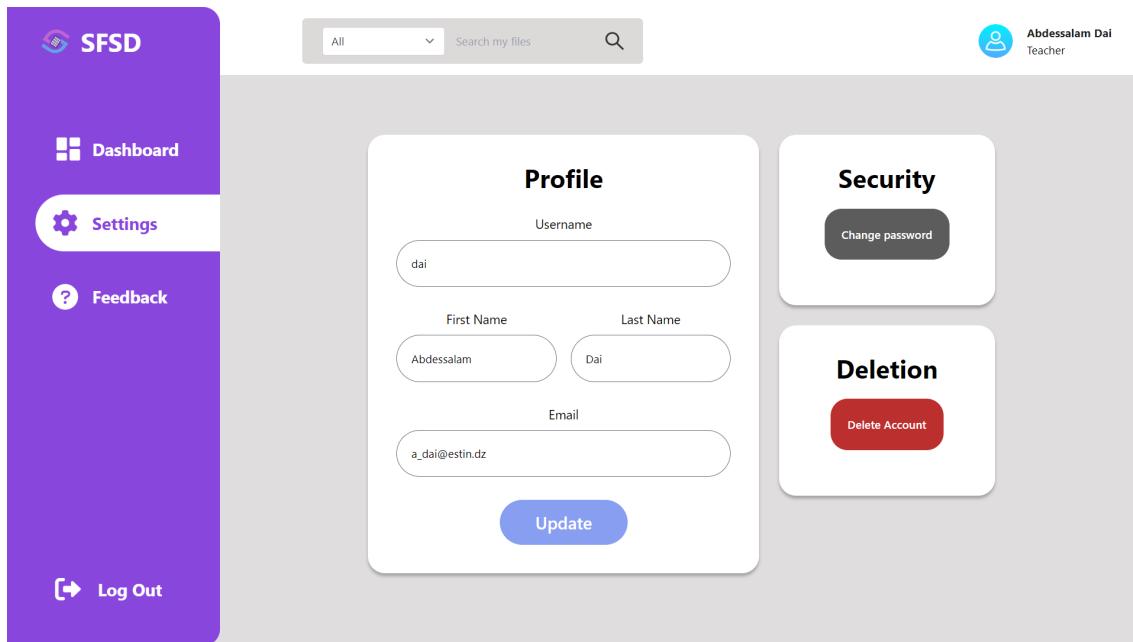


FIGURE 4.10 – settings de compte.

L'enseignant peut consulter la liste des étudiants.

The screenshot shows a user interface for managing student data. On the left, a purple sidebar menu lists navigation options: Dashboard, Admin, Students (which is selected and highlighted in white), Settings, Feedback, and Log Out. At the top right, a teacher profile is displayed with the name 'Abdessalam Dai' and the title 'Teacher'. The main content area shows a grid of four student profiles. Each profile card contains a circular icon representing the student, the student's name, and their email address. The students listed are Benayache Houssam (email: h_benayache@estin.dz), zineddine LOUZANI (email: z_louzani@estin.dz), Ouassim Hamdani (email: o_hamdani@estin.dz), and bilal CHATBI (email: b_chatbi@estin.dz). A search bar at the top of the main area allows users to search for files.

FIGURE 4.11 – Liste des étudiants.

L'utilisateur peut nous contacter et même nous envoyer des commentaires concernant l'application.

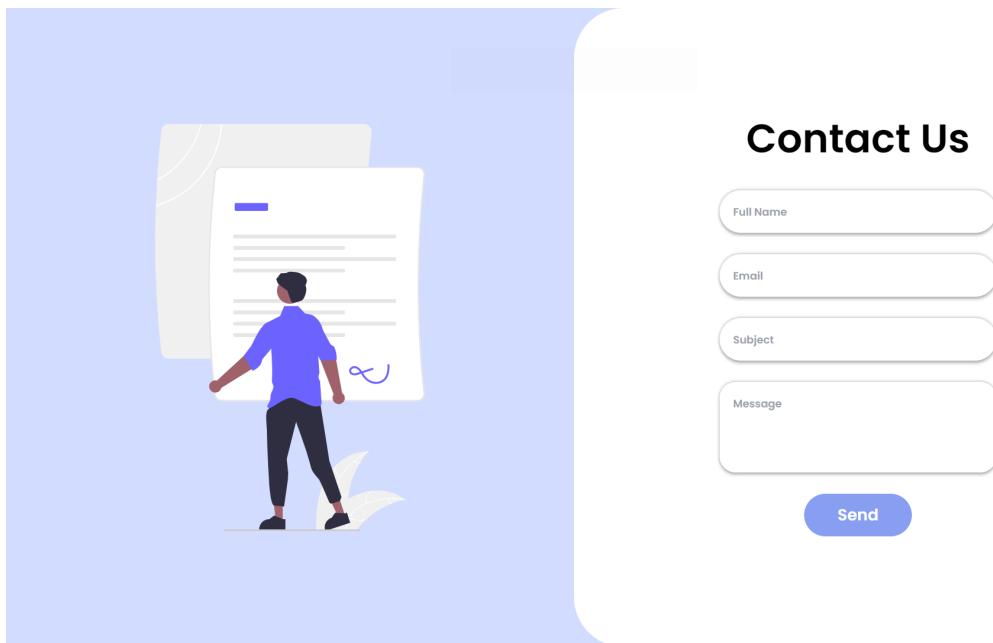


FIGURE 4.12 – Contact

4.3 Logo

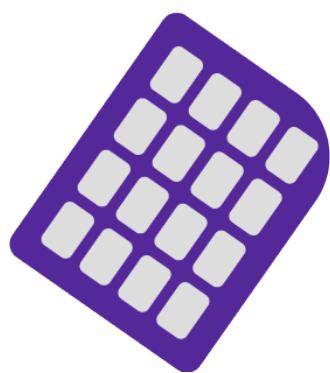
4.3.1 Signification du logo

Détaillons notre logo afin de comprendre la signification de sa forme. Notre logo est composé essentiellement de 2 parties et sert à représenter 3 choses.

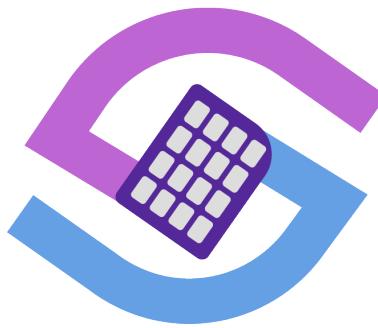


Tout d'abord, nous avons l'œil représenté par une double courbe avec une extrémité droite au centre, ce qui est également spécial de cette forme est quand vous lui donnez un espace central, il va commencer à ressembler à un S qui dans notre cas représente SFSD, et l'œil peut également être vu avec une seule partie, Le but de l'œil est de représenter la visualisation puisque c'est la partie clé de lui et de voir des graphiques.

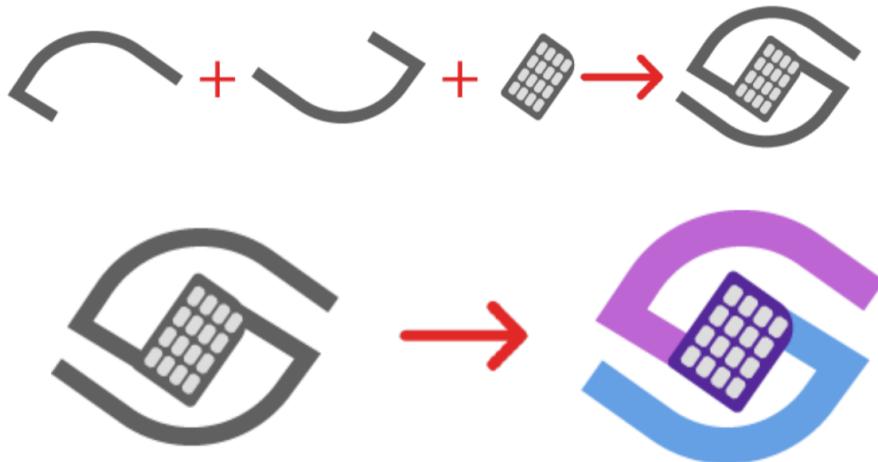
Deuxième Nous avons notre fichier à la place de la pupille de l'œil, le fichier est représenté par un rectangle avec un angle arrondi qui est la conception standard pour les fichiers, et nous avons ajouté quelques-uns de nos ingrédients spéciaux qui sont les blocs de mémoire représentés comme petit rectangle aligné à l'intérieur du fichier, cette partie de la conception sert à représenter les structures de fichiers.



Ensemble, nous obtenons notre logo utilisé pour représenter la visualisation des structures de fichiers qui constitue notre projet.



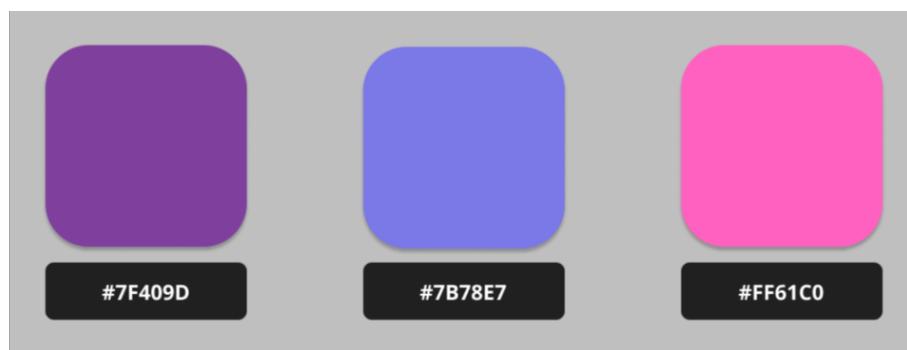
4.3.2 Processus de création de logo



4.3.3 Le choix des couleurs

Ces couleurs mélangées ensemble fournissent une excellente combinaison et schématique pour la partie conception de l'interface utilisateur. La décision de couleur était dirigée vers le rose et le pourpre en raison des préférences personnelles de notre client.

Le bleu qui les accompagne est parfaitement ajusté. Ces couleurs combinées ont créé notre système de coloration pour tout le projet.



Ces couleurs n'ont pas été utilisées par toutes leurs valeurs exactes sur

toutes nos pièces, certaines ont été légèrement modifiées mais pas tellement différentes.

4.4 Fonctionnalités des buttons

4.4.1 Page d'accueil :

- **Home** : retourner à la page d'accueil.
- **about** : aller à une courte présentation de vsfsd web app.
- **creators** : aller à une courte présentation de l'équipe de Projet.
- **dashboard** : lister les fichiers créés ou clonés par l'utilisateur.
- **log in** : aller vers la page d'authentification.
- **register** : aller vers la page d'inscription.

4.4.2 Page de travail dashboard :

- **settings** : aller vers la page de paramètres où on peut modifier le profile , changer le mot de passe et supprimer le compte.
- **feedback** : aller vers la page de contact pour laisser vos feedback.
- **log out** : déconnecter et aller vers la page d'authentification.
- **display all files** : visualiser la liste des fichiers existants.
- **new file** : créer un nouveau fichier en fournissant son nom, son type et sa structure.
- **recherche** :rechercher un fichier au fonction de deux paramètres le type et le nom.
- **profile** : aller vers la page de profile et les fichiers sauvegardés.

4.4.3 Page d'édition d'un fichier :

- **remplir par des données aléatoire** : : remplire la structure par des données aléatoires après avoir rentrer le nombre d'elements, la clé max et la clé min.
- **chercher un élément** : lancer la procédure de recherche selon le type de fichier et la structure avec animation .
- **supprimer logiquement** : lancer la procédure de suppression logique selon le type de fichier et la structure avec animation.
- **supprimer physiquement** : lancer la procédure de suppression physique selon le type de fichier et la structure avec animation.

- **ajouter un élément** : lancer la procédure d'ajout d'un élément selon le type de fichier et la structure avec animation.
- **modifier un élément** : modifier un élément en précisant son clé et les nouvelles données .
- **sauvegarder** : enregistrer les modification sur le fichier.
- **paramètres de fichier** : modifier certaines fonctionnalités comme activation de l'animation , la suppression de fichier.

4.5 Conclusion

Dans ce chapitre nous avons commencé par décrire notre travail réalisé ainsi que une explication du logo, puis une description des fonctionnalités des boutons.

Chapitre 5

Conclusion Général de Projet

Tout au long de la préparation de notre projet VSFSD, nous avons essayé de mettre en pratique les connaissances acquises durant nos études scolaires et nos compétences personnelles et cela dans le but de réaliser une application web de visualisation de différentes type de structures de données et structures des fichiers.

Au cours de ce projet, nous avons étudié et implémenté les différents algorithmes de création et suppression et modification des données et des fichiers comme TOF,TNOF,LOF, LNOF, HAHAGE,INDEXÉ... Afin de simplifier le maximum possible la procédure d'explication et d'enseignement, pour bien expliquer et comprendre les concepts de module SFSD.

Les critères pris en considération dans notre projet sont :

- La simplification.
- La performance.
- L'efficacité.

D'après l'étude comparative que nous avons réalisée, nous remarquons qu'il y a une grande différence entre la théorie et la pratique, l'algorithme sur papier n'est plus le même sur l'ordinateur. Et bien sûr la simplicité des concepts avec animation, ce qui prouve l'utilité de notre projet grâce à l'idée de visualisation des données.

Nous souhaitons pour les prochains projets d'élargir notre travail, et d'ajouter les différentes fonctionnalités sur la manipulation des données et fichiers, et pour quoi pas la disponibilité des applications similaires pour d'autres modules scolaires, et la maintenance et le mettre à jour de cette application web.

Enfin, les temps impartis à la réalisation du projet furent bref et il a fallu faire preuve de flexibilité et de persévérance, parfois pour respecter les délais, parfois pour respecter les contraintes technologiques imposées

par le projet. Nous avons retrouvé lors de ces trois mois, les compétences, les contraintes mais aussi l'excitation d'un projet innovant.