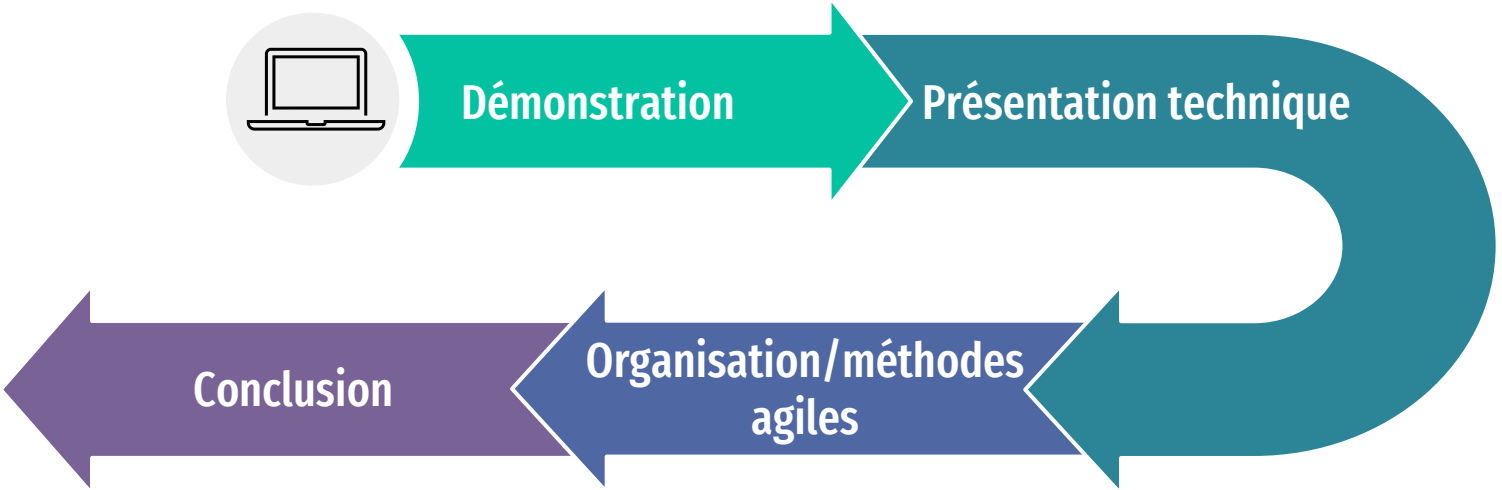
Projet long TOB

Réalisation d'un gestionnaire de ticket de caisse

Plan


Démonstration
Présentation technique
Organisation/méthodes agiles
Conclusion

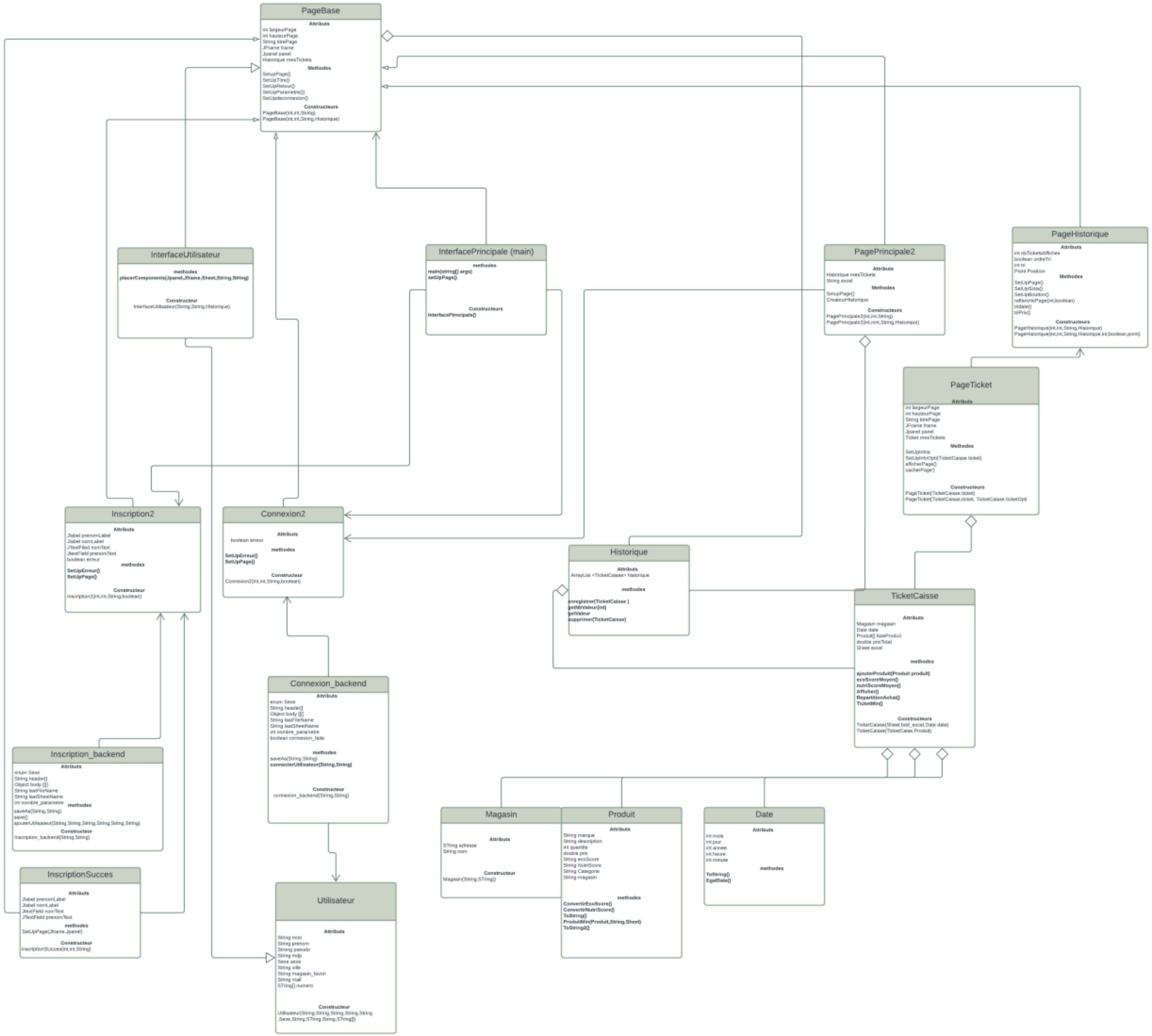
INTRO

* En France, 12,5 milliards de tickets de caisse sont imprimés chaque année, soit l'équivalent de 150000 tonnes de papier.

* Notre application TRICKET a pour objectif de stocker intuitivement les tickets de caisses des consommateurs et d'en extraire les données.

1) Démonstration

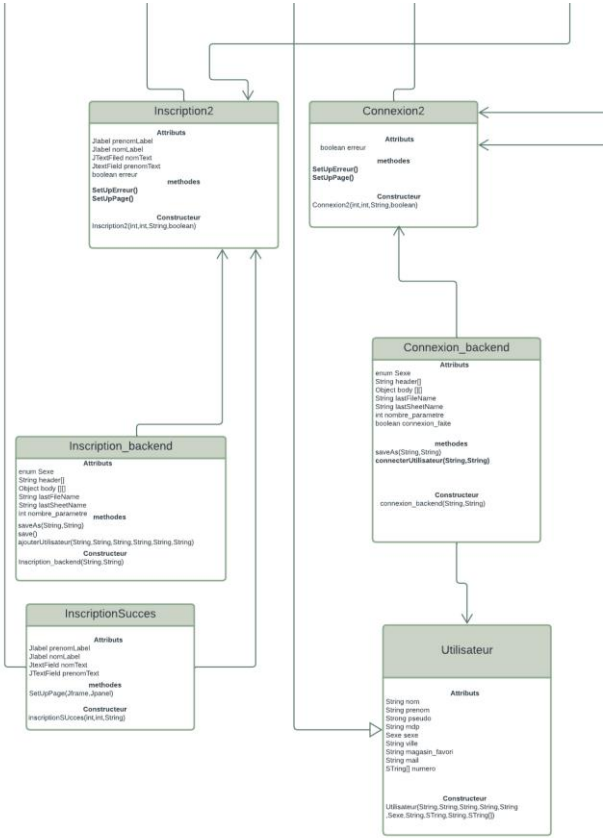
2) Présentation technique



PageBase
Attributs
int largeurPage
int hauteurPage
String titrePage
JFrame frame
Jpanel panel
Historique mesTickets
Methodes
SetupPage()
SetUpTitre()
SetUpRetour()
SetUpParametre()
SetUpdeconnexion()
Constructeurs
PageBase(int,int,String)
PageBase(int,int,String,Historique)
InterfaceUtilisateur
methodes
placerComponents(Jpanel,Jframe,Sheet,String,String)
Constructeur
InterfaceUtilisateur(String,String,Historique)
InterfacePrincipale (main)
methodes
main(string[] args)
setUpPage()
Constructeurs
InterfacePrincipale()
PagePrincipale2
Attributs
Historique mesTickets
String excel
Methodes
SetupPage()
CreateurHistorique
Constructeurs
PagePrincipale2(int,int,String)
PagePrincipale2(int,nint,String,Historique)
PageHistorique
Attributs
int nbTicketsAffiches
boolean ordreTri
int tri
Point Position
Methodes
SetUpPage()
SetUpSlots()
SetUpBouton()
raffarichirPage(int,boolean)
tridate()
triPrix()
Constructeurs
PageHistorique(int,int,String,Historique)
PageHistorique(int,int,String,Historique,int,boolean,point)
PageTicket
Attributs
int largeurPage
int hauteurPage
String titrePage
JFrame frame
Jpanel panel
Ticket mesTickets
Methodes
SetUpInfos
SetUpInfoOpti(TicketCaisse ticket)
afficherPage()
cacherPage')
Constructeurs
PageTicket(TicketCaisse ticket)
PageTicket(TicketCaisse,ticket, TicketCaisse ticketOpti
Inscription2
Attributs
Jlabel prenomLabel
Jlabel nomLabel
JTextFiled nomText
JtextField prenomText
boolean erreur
methodes
SetUpErreur()
SetUpPage()
Constructeur
Inscription2(int,int,String,boolean)
Connexion2
Attributs
boolean erreur
methodes
SetUpErreur()
SetUpPage()
Constructeur
Connexion2(int,int,String,boolean)
Historique
Attributs
ArrayList <TicketCaisse> historique
methodes
enregistrer(TicketCaisse)
getNbValeur(int)
getValeur
supprimer(TicketCaisse)
TicketCaisse
Attributs
Magasin magasin
Date date
Produit[] listeProduit
double prixTotal
Sheet excel
methodes
ajouterProduit(Produit produit)
ecoScoreMoyen()
nutriScoreMoyen()
Afficher()
RepartitionAchat()
TicketMin()
Constructeurs
TicketCaisse(Sheet bdd_excel,Date date)
TicketCaisse(TicketCaise,Produit)
Connexion_backend
Attributs
enum Sexe
String header[]
Object body [][]
String lastFileName
String lastSheetName
int nombre_parametre
boolean connexion_faite
methodes
saveAs(String,String)
connecterUtilisateur(String,String)
Constructeur
connexion_backend(String,String)
Inscription_backend
Attributs
enum Sexe
String header[]
Object body [][]
String lastFileName
String lastSheetName
int nombre_parametre
methodes
saveAs(String,String)
save()
ajouterUtilisateur(String,String,String,String,String,String)
Constructeur
Inscription_backend(String,String)
InscriptionSucces
Attributs
Jlabel prenomLabel
Jlabel nomLabel
JtextField nomText
JTextField prenomText
methodes
SetUpPage(Jframe,Jpanel)
Constructeur
inscriptionSUcces(int,int,String)
Utilisateur
Attributs
String nom
String prenom
Strong pseudo
String mdp
Sexe sexe
String ville
String magasin_favori
String mail
STring[] numero
Constructeur
Utilisateur(String,String,String,String,String
,Sexe,String,STring,String,STring[])
Magasin
Attributs
STring adresse
String nom
Constructeur
Magasin(String,STring[])
Produit
Attributs
String marque
String description
int quantite
double prix
String ecoScore
String NutriScore
String Categorie
String magasin
methodes
ConvertirEcoScore()
ConvertirNutriScore()
ToString()
ProduitMin(Produit,String,Sheet)
ToString2()
Date
Attributs
int mois
int jour
int annee
int heure
int minute
methodes
ToString()
EgalDate()

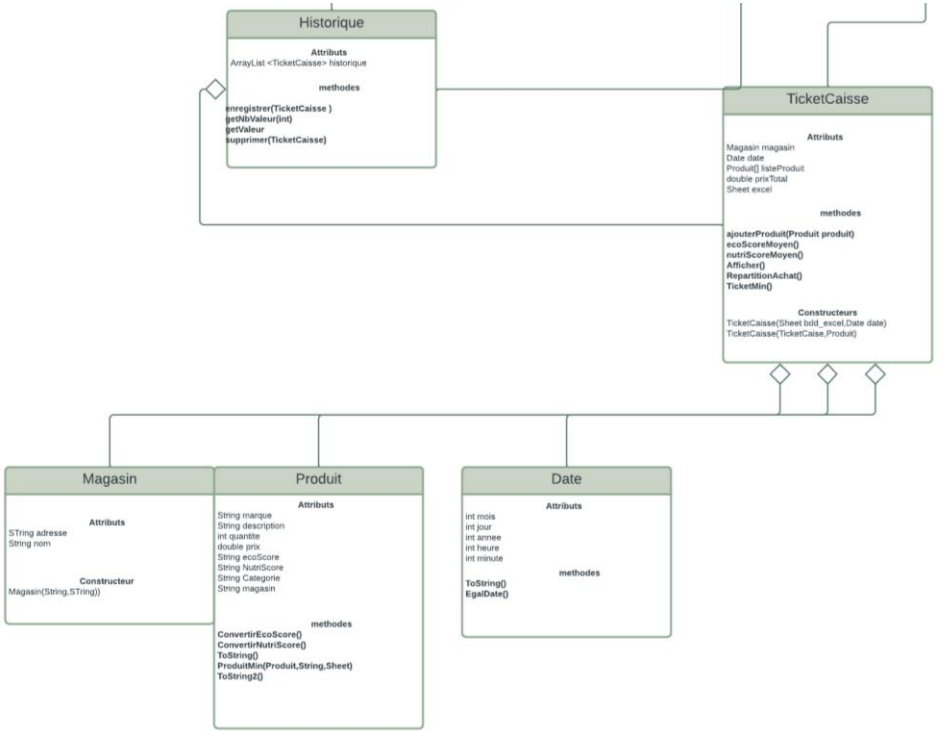
Partie connexion/inscription

- Classe utilisateur : création d'une classe utilisateur afin de formaliser un utilisateur
- Connexion_backend : rechercher l'utilisateur déjà inscrit (base de donnée: Apache)
- Inscription_backend : analogue à « Connexion_backend »
- Connexion2 : front-end de la page de connexion: JtextField
- Inscription2 : front-end de la page d'inscription: JtextField
- Inscription_succès : message de succès d'inscription



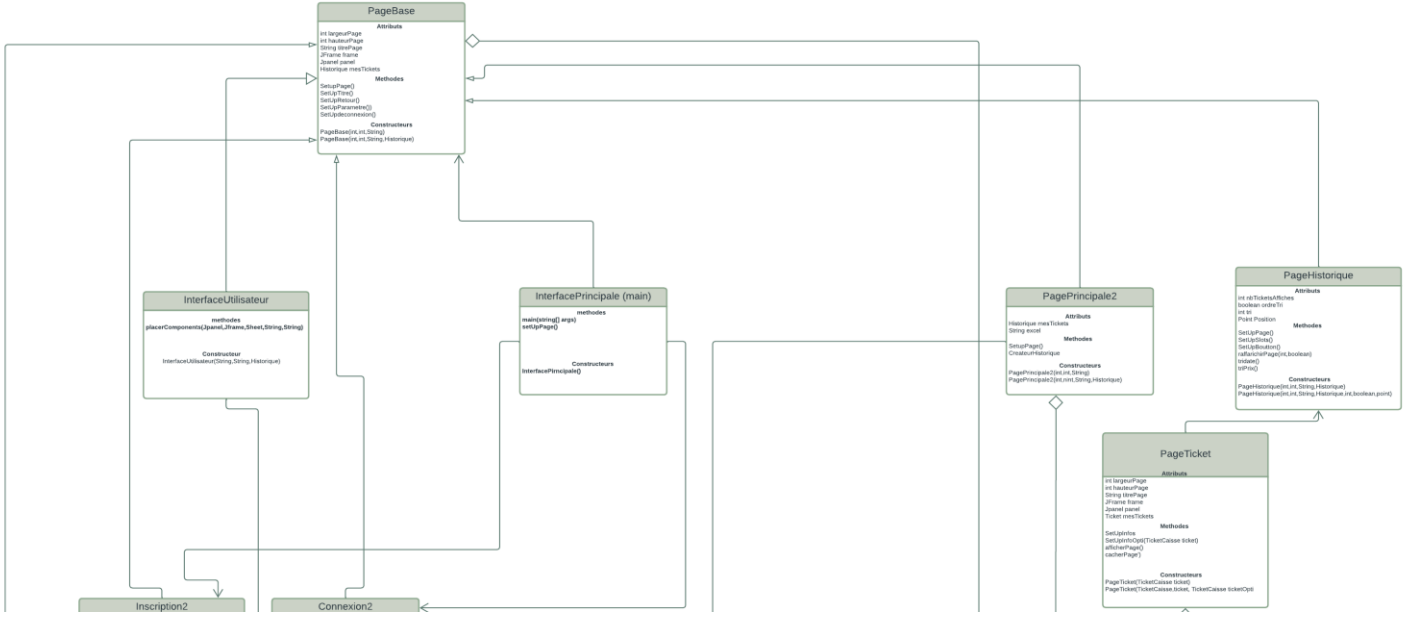
Partie ticket/historique

- Tickets de caisse générés aléatoirement à partir d'une base de données
- Tickets de caisse composés d'une date, magasin, liste de produits
- Historique : liste de tickets de caisse
- Bibliothèque Apache pour la gestion des BDD



Partie création des pages

- PageBase : créer une nouvelle fenêtre taille générique; met en place les éléments communs aux pages utilisateurs. Les autres pages sont des héritages de PageBase en ajoutant leurs particularités.
- InterfacePrincipale : contient le main et ajout des boutons de « connexion » et « inscription »
- Interface utilisateur: reliée à la base de donnée afin de récupérer les infos de la personne se connectant.
- PagePrincipale2 : extend de PageBase : mise en place des boutons de la page d'accueil
- PageHistorique : mise en place de l'historique

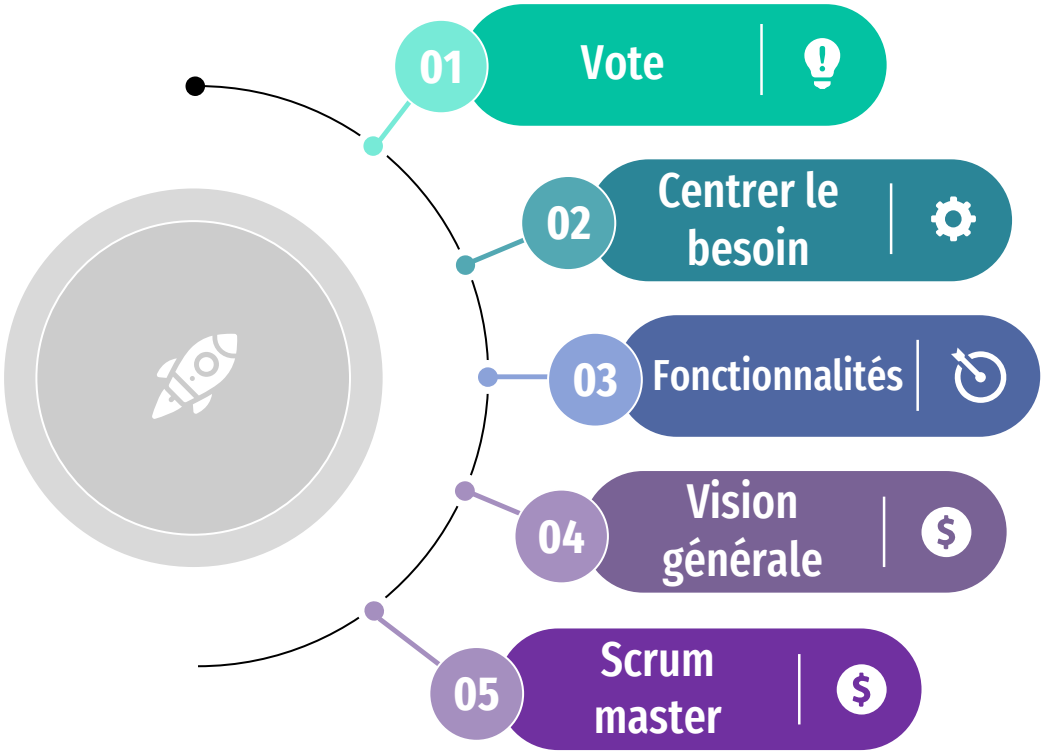


Principaux choix techniques

- → Utilisation des bases de données pour simuler un client qui passe en caisse: bibliothèque Apache
- → Base de données Excel
- → Interface graphique sur Swing
- → Travail avec une base de données pour assurer la connexion et l’inscription

3) Organisation et méthodes agiles

Début du projet : définition du besoin



Organisation d'un vote (gform) pour classer les idées de chacun anonymement

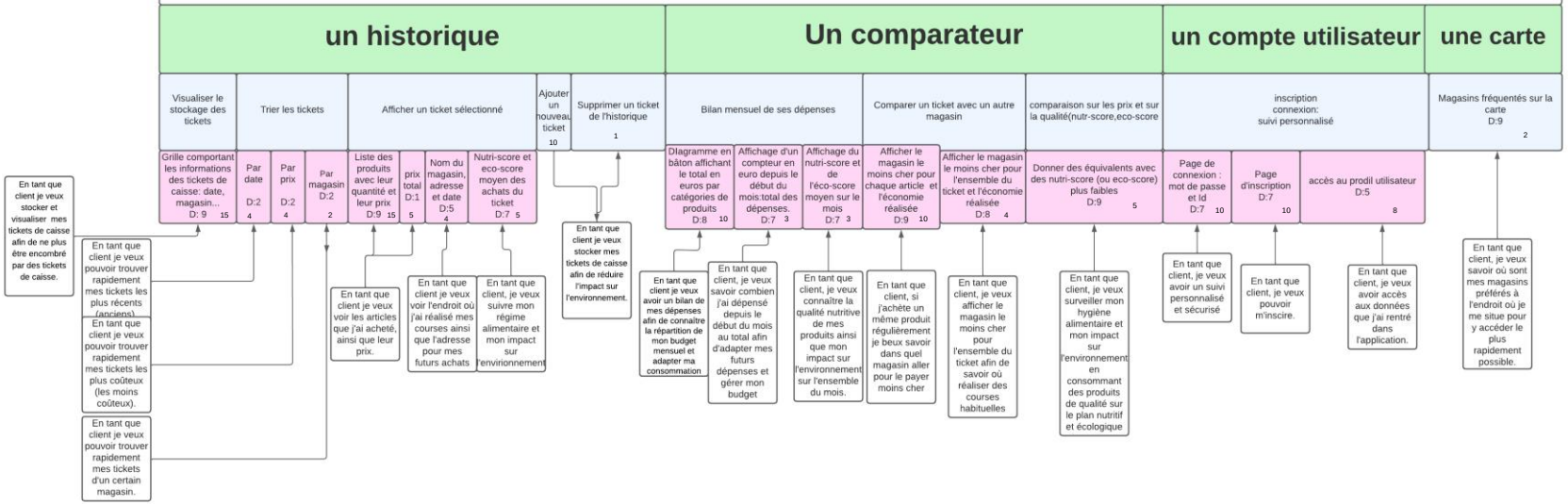
Précision sur le projet et les besoins de l'utilisateur : pitch elevator

Choix des fonctionnalités : environ une par personne

Vision générale de l'application : backlog produit et design complet de l'application sur Canva : définition d'une charte graphique

Désignation du scrum master : organiser, suivre, veiller

Gestionnaire de ticket de caisse



Organisation générale

Equipe front-end

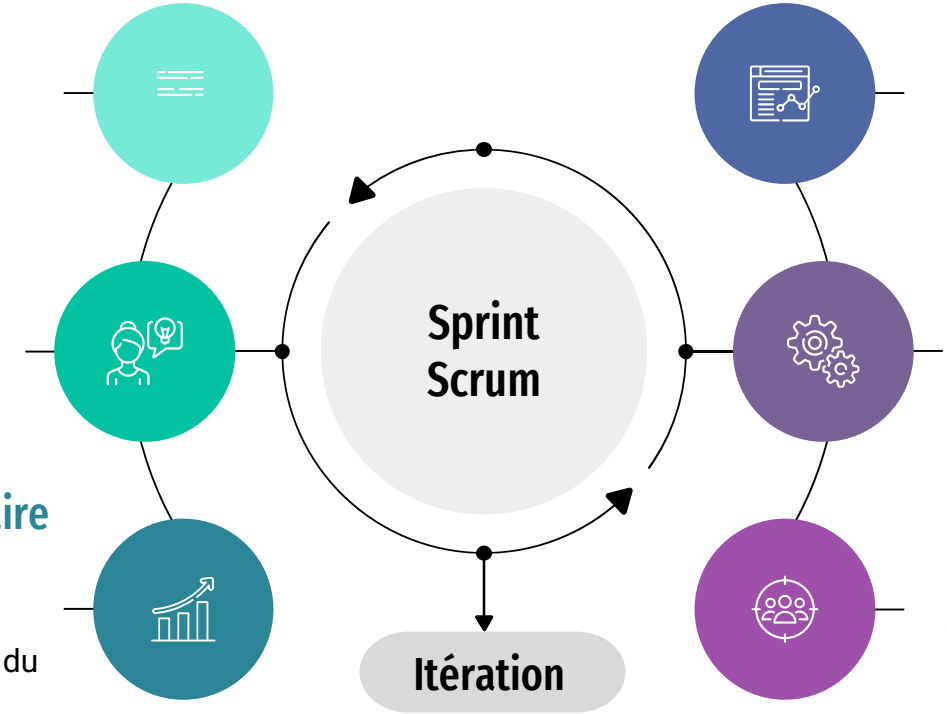
Interface graphique de l'application

Equipe back-end

Codage des méthodes/classes de l'application

Réunion hebdomadaire

Mise en commun des codes, montrer les avancées, définir la suite du projet



Rapports

Rédaction de rapports intermédiaires+ rapports personnels

Manuel utilisateur

Rédaction d'un manuel utilisateur à chaque itération

Vote sur les décisions

Valeurs métiers: scrum time

Conclusion : ce que nous avons appris grâce à ce projet

Compétences technique

Gestion des bases de données, maîtrise de swing, plus généralement la maîtrise de java

Gestion du travail d'équipe

Organiser, communiquer

Gestion du temps

Prioriser: methodes agiles

Besoin client

Définir un besoin précis