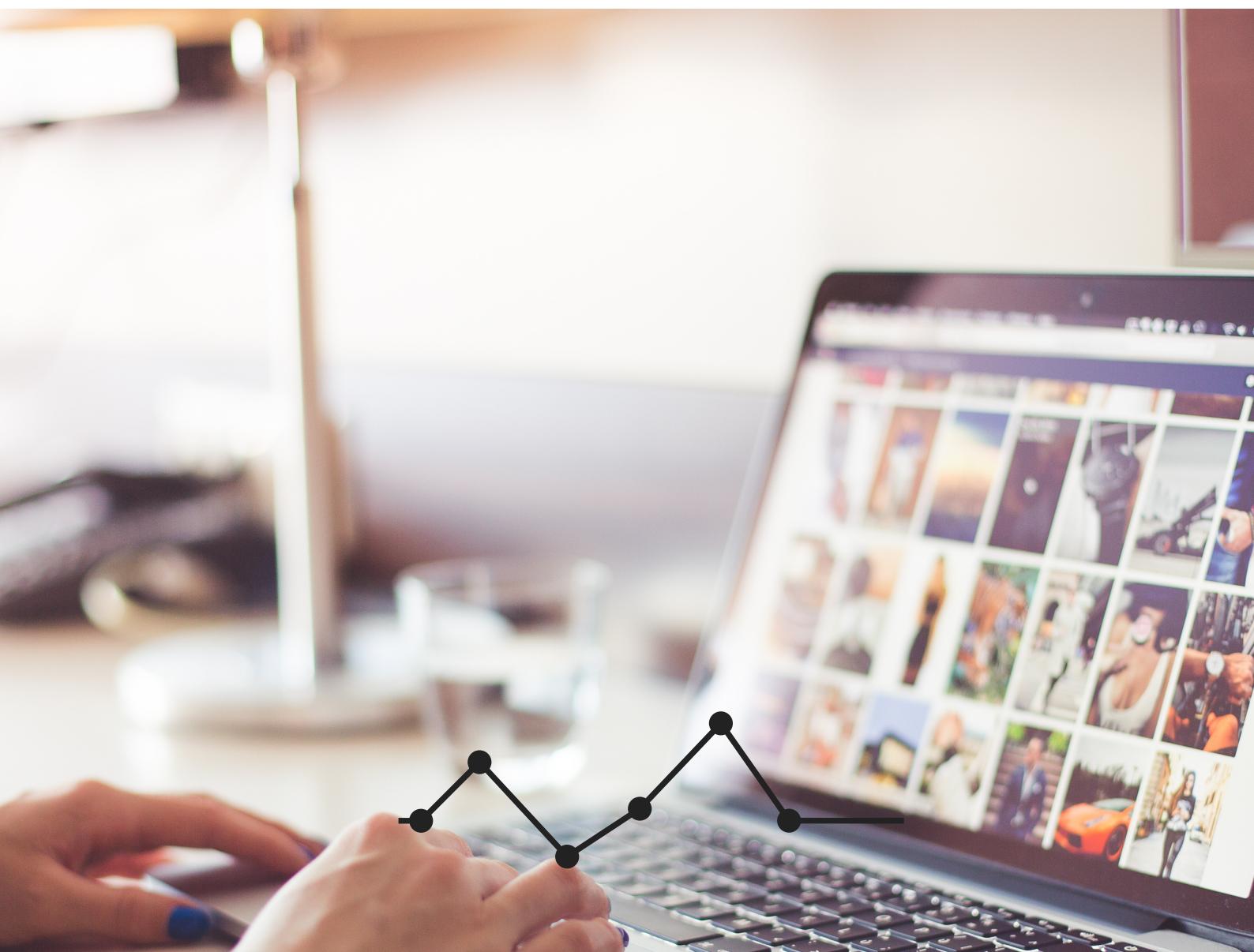


Projet tutoré

Le testament numérique

RAPPORT TECHNIQUE



MONNERVILLE JEAN-EDOUARD

TOULON SARA

LEROY JULIE

Présentation générale

L'application permet aux utilisateurs de gérer leurs données présentes sur les réseaux sociaux en programmant leur devenir après leur décès.

Cibles:

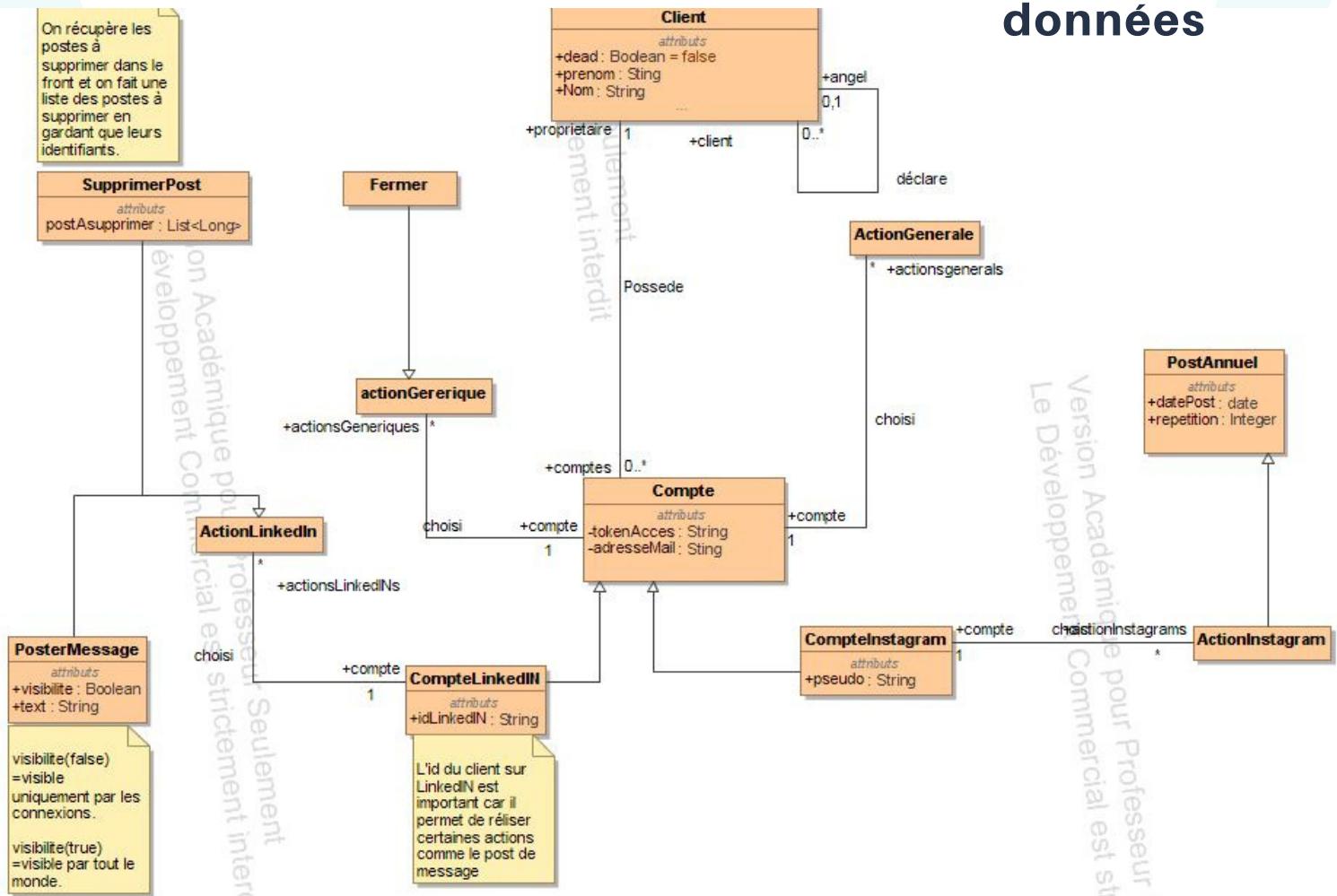
Tout public, quel que soit son âge, possédant un compte LinkedIn. L'utilisateur se connecte grâce à ses identifiants liés à un de ses comptes.

Actions:

Plusieurs actions sont possibles à partir d'un compte. L'utilisateur peut choisir de supprimer certaines de ses publications ou la totalité. Il peut programmer une publication qui se publiera automatiquement à l'annonce de son décès sur le réseau social de son choix.

Afin de savoir quand les actions doivent être déclenchées, l'utilisateur choisit une personne de son entourage, possédant elle aussi un compte LinkedIn. Cette personne sera l'Ange, et sera chargée de déclarer le décès. Un même Ange peut être choisi par plusieurs personnes.

Modèle de données



Modèle de données

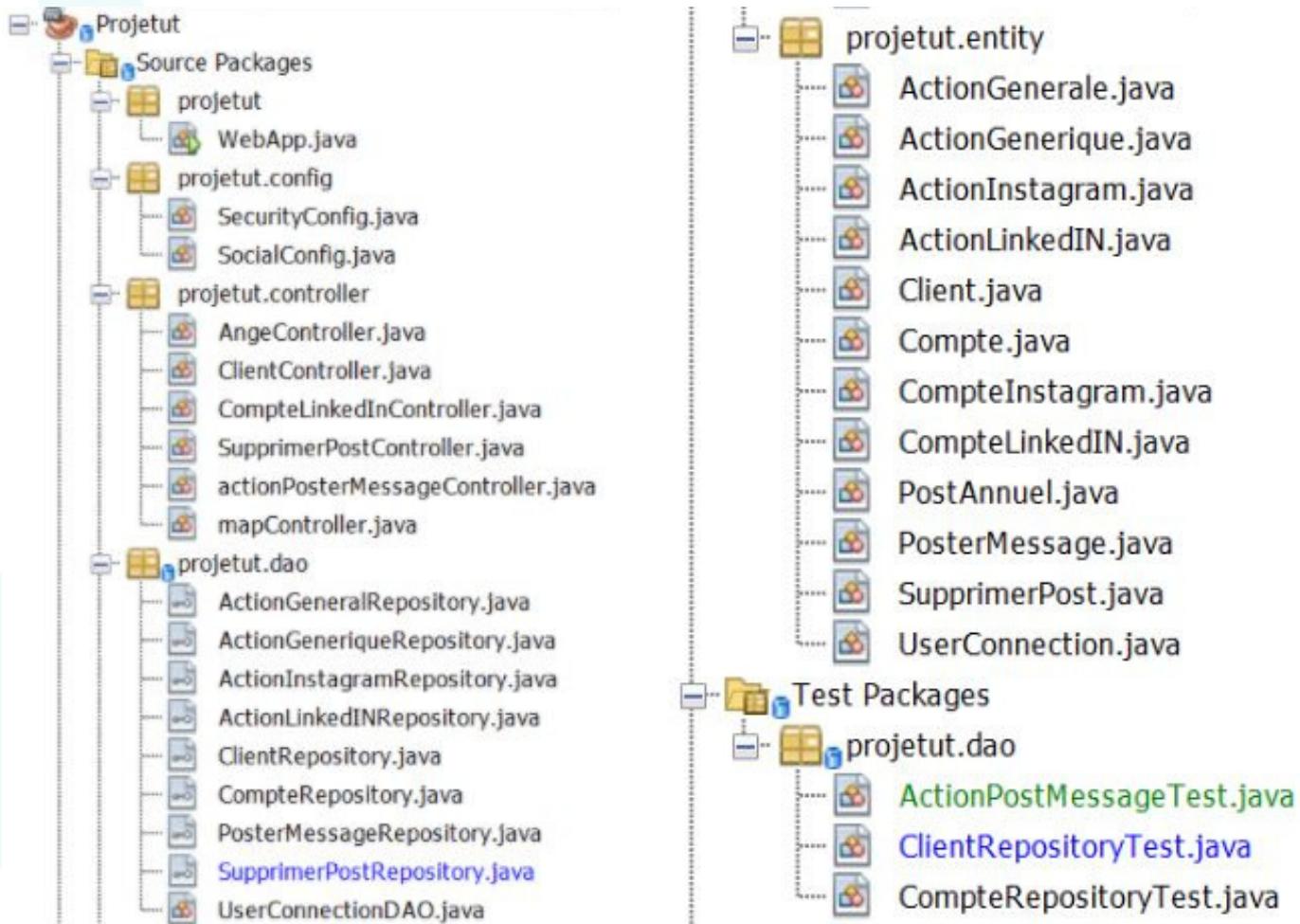


Comme nous pouvons le voir sur le modèle de données, nous avons une classe Client qui possède des propriétés basiques (prénom et nom) et une propriété dead. Dans la base de données, dead permet de savoir si le client est décédé ou pas. La déclaration de la mort s'effectue via un ange qui est lui-même un client et qui est déclaré par le client.

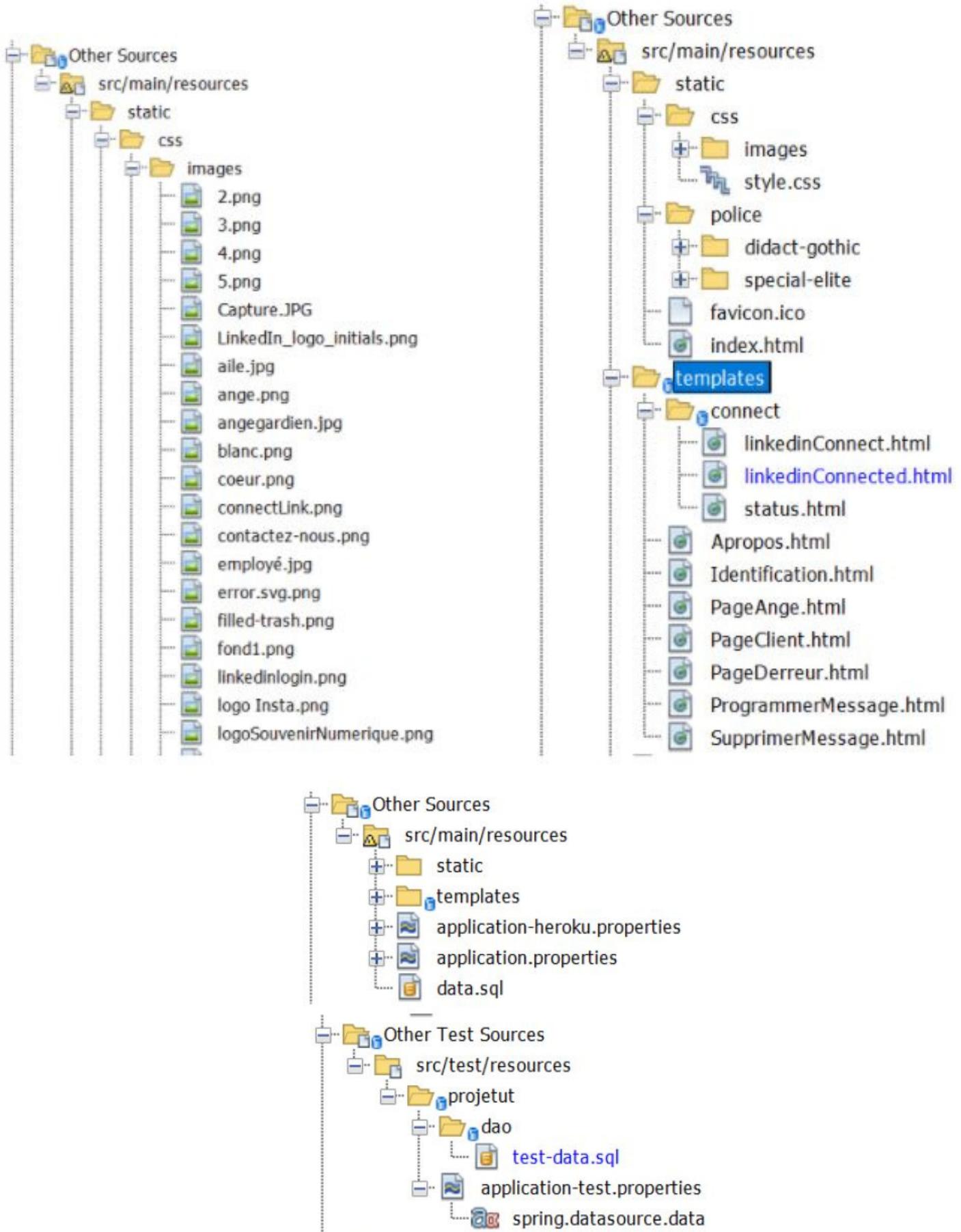
Les différentes actions exposées plus haut sont mises en place grâce aux classes ActionGénérale, ActionGénérique, ActionInstagram, ActionLinkedIn, PostAnnuel, PosterMessage... que l'on peut voir sur le modèle. Le client pourra donc choisir parmi ces différentes actions à effectuer sur son compte lors de son décès. Une autre table se nommant UserConnection est nécessaire pour l'utilisation de Spring social, son utilité n'a pas encore été montrée dans le code.

Structure de l'application

Arborescence



Modèle de données



Documentation technique

JAVA	
Spring MVC (model view controller)	Spring Web Flow Reference Guide
JPA	Spring Data JPA - Reference Documentation
Thymeleaf	Documentation - Thymeleaf
Driver JDBC H2	H2 Database - JDBC Connection - TutorialsPoint
Driver JDBC Postgresql	PostgreSQL JDBC Documentation
Spring Social	1. Spring Social Overview 2. Service Provider 'Connect' Framework (spring.io) 3. Adding Support for a New Service Provider (spring.io) 4. Connecting to Service Providers (spring.io) 5. Signing in with Service Provider Accounts (spring.io)
Spring social LinkedIn	Spring Social LinkedIn Reference Manual
Common codec	Codec – Home (apache.org)
Spring Security	Spring Security

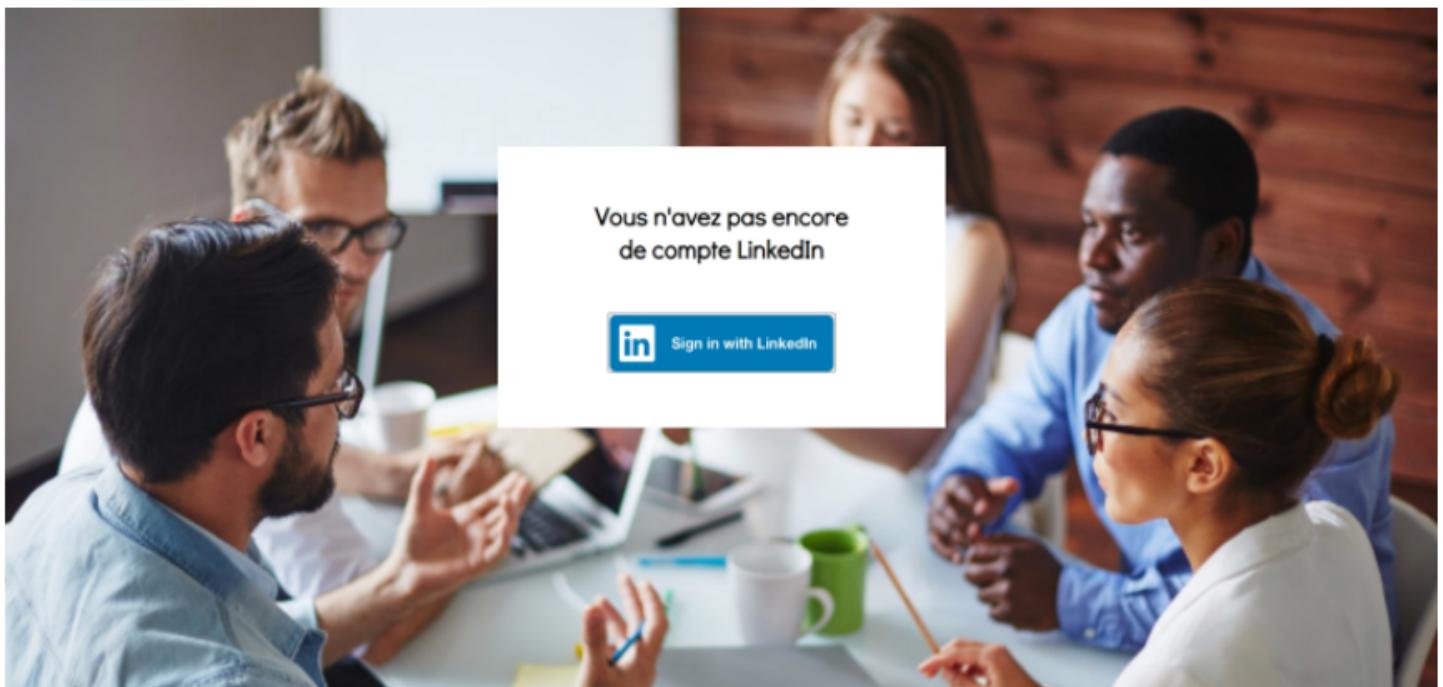
HTML= Langage utilisé pour développer le corps des différentes pages du projet	
CSS = Langage utilisé pour faire le design des pages	
API	
LinkedIn*	
Récupérer les données du profil de l'utilisateur (nom, prénom, adresse mail, url de la photo de profil)	<ul style="list-style-type: none"> ❖ Extrémité à mettre à la fin de l'url de l'api pour accéder aux différentes fonctions ❖ Documentation de l'api "Sign in with LinkedIn"
Gestion des publications (pour publier du contenu ou le supprimer)	<ul style="list-style-type: none"> ❖ Extrémité à mettre à la fin de l'url de l'api pour accéder aux différentes fonctions ❖ Documentation de l'api "Share On Linkedin"

Description générale



Au démarrage, l'application donne sur une page d'accueil (figure 4). Celle-ci permet au visiteur de se connecter via un compte Instagram ou LinkedIn. On utilise uniquement l'API de LinkedIn pour le moment donc le lien "se connecter avec Instagram" renvoie une erreur.

Puis, l'application redirige le client vers une page lui permettant de se connecter à linkedin s'il ne l'a jamais fait (figure 5) ou d'accéder directement à une page assurant sa connexion et le redirigeant vers la prochaine interface l'application (figure 6).



Description générale



figure 6: Connectez-vous à LinkedIn

Ensuite vient l'identification du client. Il se présente soit pour faire son testament ou soit déclarer un décès (figure 7). L'application est partagée en 2 interfaces : l'une pour un utilisateur souhaitant réaliser son testament numérique (figure 9), l'autre pour le fameux Ange, chargé de déclarer le décès (figure 8). Pour ce dernier c'est plutôt simple, une fois connecté, il peut en un clic faire la déclaration. Celle-ci est gérée par un contrôleur.



figure 7: page d'identification

Description générale

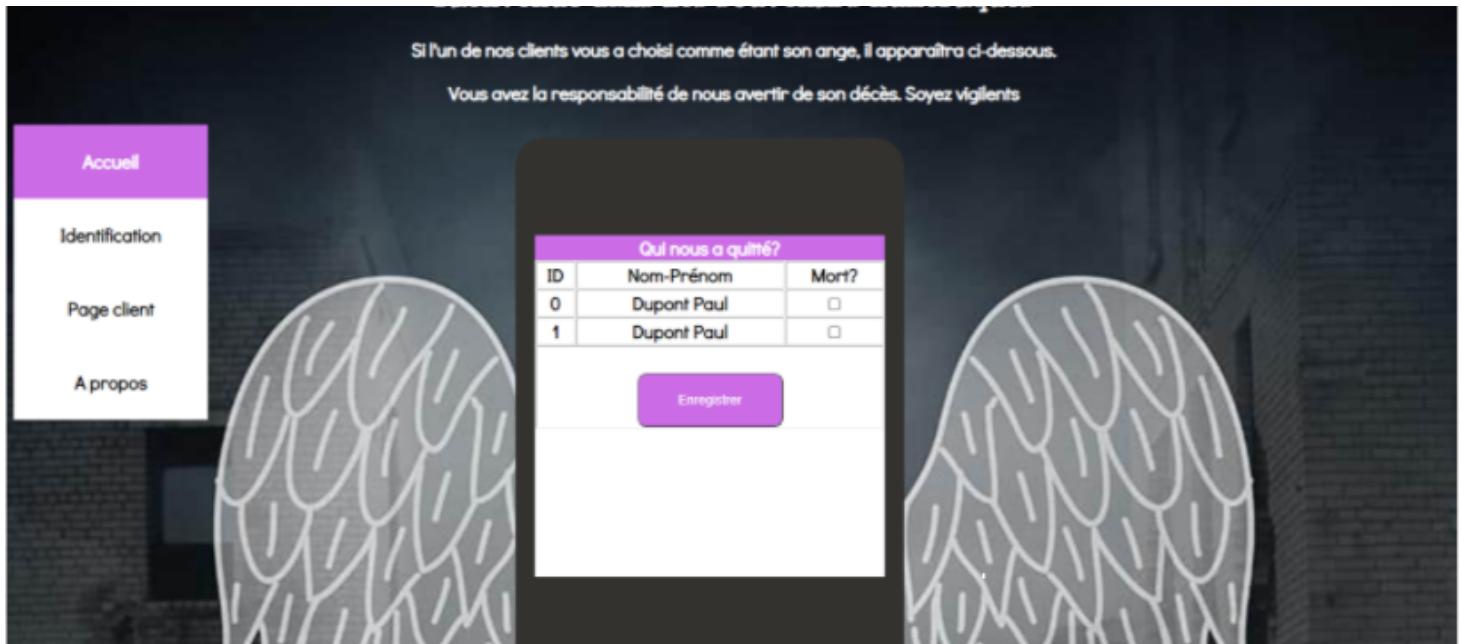


figure 8: page de l'ange



figure 9: page du client

L'utilisateur a accès aux différentes actions dont les modalités sont enregistrées dans la base de données. Celles-ci seront ainsi exécutées après la confirmation de son décès (exemple: partager une publication après le décès). Pour ce faire, en cliquant sur programmer (figure 9) par exemple, il accède à un formulaire (figure 10) lui permettant d'éditer sa publication.

Description générale



figure 10: programmer un message

« TODO », problèmes non résolus, fonctionnalités restant à développer, indications des prolongements possibles pour le projet.



- Ajout de l'ange

Dans notre application, l'ange est sélectionné par le client parmi les autres utilisateurs. Cependant pour faire partie de cette liste d'utilisateurs, il faut s'être déjà connecté à l'application. Or, un ange n'a pas d'intérêt à se connecter avant de déclarer un décès.

Nous avons donc pensé à un moyen pour le client d'entrer les informations de l'ange qu'il choisit (compte linkedIn par exemple) afin de ne pas rencontrer ce problème.

Autre possibilité: l'ange n'est pas forcément une personne possédant un compte sur notre application. Une fois inscrit, le client enregistre l'adresse mail de l'ange qui l'a choisi (cela peut être n'importe qui, pas forcément une personne déjà cliente). La personne recevra donc un mail avec un code secret lui permettant, le jour du décès du client, d'accéder à la plateforme afin d'annoncer sa mort. Les différentes actions que le client aura préalablement mise en place (effacer un compte, envoyer un message...) s'effectueront.

- Instagram

L'API d'Instagram étant gérée par Facebook, elle n'était pas disponible car elle a été récemment désactivée. Nous n'avons pas pu mettre au point la connexion mais cela pourrait être fait une fois l'API disponible.

- LinkedIn

Problème avec le Spring social LinkedIn : Il utilise par défaut un scope dont nous n'avons pas l'accès. Il s'agira alors de soit obtenir les droits sur ce scope, soit modifier le scope utilisé par le Spring social LinkedIn.

- Date des publications programmées

Nous avons choisi d'activer les actions programmées dès que le décès est déclaré. Notre application ne peut donc pas choisir de date prédéfinie par le client et pouvant être répétée selon un intervalle de temps. Il faudrait que l'utilisateur choisisse une date qui nécessite une année. Or, ne connaissant pas sa date de décès, il est compliqué de choisir. Nous avons envisagé d'utiliser un "select" et ainsi y entrer toutes les dates, sans année définie mais cela n'a rien donné.

- Corriger les problèmes d'affichage de la liste des clients dans la page client et dans la page ange.
- Supprimer les actions récentes : permet d'annuler les actions posées
- Avec plus de temps, nous aurions pu développer une application web dans son entiereté. C'est-à-dire, bien développer notre base de données pour pouvoir stocker les données de nos utilisateurs notamment.
- Créer une interface de l'application à part pour accéder gérer et contrôler la base de données
- Configurer et développer l'application pour qu'elle puisse utiliser les autres API telles que Instagram, Twitter et Facebook. Il faudrait aussi mettre à jour la base données étant basée sur ce que les API permettent de faire.
- Augmenter le niveau de cyber sécurité de l'application (par exemple, permettre l'accès à la page identification si et seulement si le client s'est connecté déjà avec au moins un réseau social).
- Ajouter la fonctionnalité permettant l'exécution des actions suite à la déclaration de la mort d'un client.
- Mettre en place un mécanisme de vérification de la mort du client en plus du signalement par l'ange.
- Mécanisme permettant de renouveler les token d'accès aux réseaux sociaux.
- Implémenter les fonctionnalités permettant de supprimer ou poster du contenu sur LinkedIn

*NB= Tout le développement et la recherche sont basés sur l'API LinkedIn jusqu'à maintenant.

Prolongement possible pour le projet :

Dans l'éventualité où toute la to-do-list est checkée, il sera intéressant de voir comment avoir une vue globale sur l'ensemble des actions comme un scénario d'animation illustrant tout ce qui va être fait.