01/12/2023

Alexandre Cubizolle

findByDev

Rapport de Stage

Sommaire

1. Présentation du/des projets
   1. Expression des besoins (cahier des charges)
   2. Organisation du travail (méthode agile ? Git ?)
   3. La stack technique (quelles sont les technologies utilisées ?)
2. Conception du projet (aussi appelé « spécifications fonctionnelles et techniques)
   1. Analyse des besoins (diagrammes UML de cas d’utilisation))

2.2. Architecture du projet

                                                               i.      Conception de BDD

                                                             ii.      Diagrammes de package

                                                           iii.      Diagrammes de classe

* 1. Maquettage (si interface graphique)
  2. Conception des tests unitaires

1. Réalisation (faire un focus sur **une ou deux fonctionnalités** avec des exemples de code pertinents)
   1. Rappel du cas d’utilisation
   2. Contraintes techniques
   3. Développement des fonctionnalités avec description de toutes les couches

                                                               i.      Implémentation de la base de données

                                                             ii.      Composant d’accès aux données (couche DAO ou repository avec configuration de l’ORM)

                                                           iii.      Logique métier (possibilité de mettre du pseudo-code et l’implémentation)

                                                           iv.      Interface graphique

                                                             v.      Tests unitaires

* 1. Résultat

1. Veille sur la sécurité (peut être appliquée au projet)

1. Bilan
   1. Ce qui a été fait (ce qui marche, est-ce déployé ? Que dit le client ?)
   2. Perspective d’évolution (ce qu’il reste à faire)
2. Conclusion sur 1 page (appréciation personnelle du projet, satisfaction
3. **Présentation du Projet**

**Introduction**

The "FindByDev" project addresses a fundamental need in the field of software development. Currently, there isn't a dedicated platform for developers to meet, share their experiences, and build both professional and personal connections. Think of it as a "Tinder" for developers. The main goal of this project is to create a free dating app exclusively for adult developers (aged 18 and above).

The application will have a visual style similar to popular dating apps, with references related to the world of development. Our primary focus is on safeguarding users' personal data and ensuring compliance with data protection regulations.

* 1. **Expression of Needs**

The "FindByDev" app aims to meet several essential needs:

* Allowing users to create profiles.
* Enabling users to "like" other profiles and waiting for a mutual "like" to start a conversation.
* Providing the capability to search for other users who share interests in software development.
* Making communication easy through instant messaging.
* Emphasizing the security of personal data and ensuring compliance with data protection regulations.
* Incorporating features tailored for developers, such as integration with Git, highlighting preferred programming languages, introducing a popularity rating, and adding humor elements.

**1.2 Organisation du Travail**

Dans le cadre de ce projet, j’ai choisi d’utiliser un diagramme de Gantt afin d’obtenir visuellement un planning de gestion du projet. De plus, l'utilisation de Git comme système de gestion de version facilite le partage sur plusieurs postes de travail.

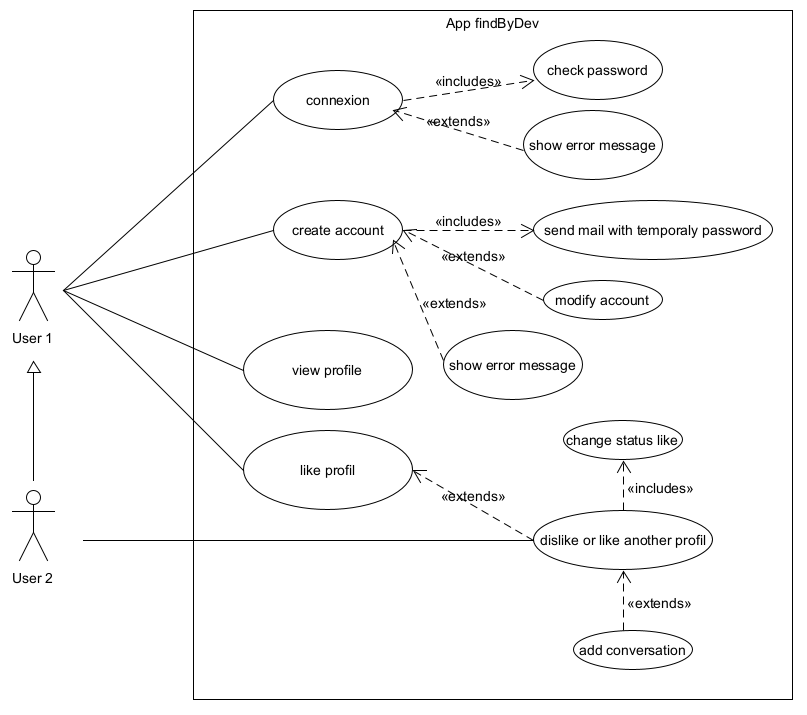
* 1. **La Stack Technique**

La réalisation de l'application "findByDev" implique l'utilisation d'un ensemble de technologies et d'outils. Cela comprendra la mise en place d'une base de données pour stocker les informations des utilisateurs, un composant d'accès aux données (repository) avec la configuration de l'ORM (Object-Relational Mapping), la logique métier, une interface graphique intuitive, et des tests unitaires pour garantir la fiabilité du code.

* Pour réaliser ce projet j’ai utilisé l’IDE VSCode car il est simple et complet.
* Côté conception c’est UMLet pour les USECases et GranttProject pour la planification, LucidChart pour un diagramme de séquence.
* La conception de la BDD a été faîtes avec Looping, qui est très visuel.
* J’utilise PostGrèsSQL pour la BDD et DBeaver pour sa gestion.
* Côté front développé avec React-pwa adapté au mobile, javascript vanilla.
* SpringBoot pour le back avec un structure MVC.
* Postman pour les tests de requêtes.
* <https://websocketking.com> pour les test de websocket.

Dans la suite de ce rapport, je vais aborder en détail les différentes phases du projet, de la conception à la réalisation, en mettant en évidence les principaux éléments et choix techniques. Nous explorerons également les aspects de sécurité qui sont essentiels pour protéger les données sensibles des utilisateurs. Enfin, nous dresserons un bilan des réalisations à ce stade du projet, ainsi que des perspectives d'évolution pour les prochaines étapes.

1. **Conception du projet**
   1. **Analyse des besoins**

****

**Gestion des profils et des interactions**

**Acteurs :**

* User 1 (utilisateur principal)
* User 2 (utilisateur secondaire)

**Description :** Ce cas d'utilisation couvre les différentes fonctionnalités de l'application "findByDev" liées à la gestion des profils, des interactions et des comptes des utilisateurs.

**Scénario principal :**

L'utilisateur User 1 se connecte à l'application. La connexion comprend la vérification du mot de passe et, en cas d'erreur, l'affichage d'un message d'erreur.

Une fois connecté, User 1 peut choisir de créer un compte ou de voir son profil existant.

Si User 1 décide de créer un compte, il doit fournir les informations nécessaires. La création de compte implique également l'envoi d'un e-mail contenant un mot de passe temporaire qu’il devra changer à sa première connexion.

Si User 1 décide de voir son profil existant, il peut afficher les détails de son profil et en modifier les éléments.

User 1 a la possibilité de "liker" un profil d'un autre utilisateur. Cette action peut être étendue pour "disliker" le profil de User 2. La gestion de cette interaction inclut la modification de l'état de "like" et peut entraîner l'ajout d'une conversation.

L'utilisateur User 2, qui hérite des fonctionnalités de User 1, peut "disliker" ou "liker" un profil d'un autre utilisateur qui l’aurai liké. L'interaction est similaire à celle de User 1.

Lors de l'interaction de "like" ou "dislike", l'application peut également gérer un changement de statut de "like" et permettre l'ajout d'une conversation entre les utilisateurs.

Ce cas d'utilisation "Gestion des profils et des interactions" couvre les principales fonctionnalités l’application "findByDev", y compris la création de comptes, la visualisation de profils, les interactions "like" et "dislike", la gestion des conversations et la gestion d'erreurs lors de la connexion ou de la création de compte. Il permet aux utilisateurs de s'engager avec d'autres profils d'utilisateurs et de gérer leurs propres informations de compte.

* 1. **Architecture du projet**

1. **Conception de BDD**

Avant tout chose, j’ai créé un dictionnaire des données pour identifier les principales interactions.

****Il va me servir à organiser et décrire toutes les données pertinentes qui seront utilisées dans l’application. Cela documente le projet et permet une meilleure compréhension pour un œil extérieur.

C’est un échantillon de ce que saura la BDD, avec les futurs noms de colonne qui, au court du développement pourront être adaptés selon les besoins encore non-identifiés. Cela m’a aussi permis d’identifier les futures relations qui seront mise en place.

1. **Looping**

MCD :

Ma BDD contient 7 tables principales

