



ATARI

Annuaire technique d'attributs pour responsables informatiques

Travail Pratique Individuel

Candidat : Jérôme Cosandey

Experts : Xavier Carrel et Alain Girardet

Chef de projet : Nicolas Borboën

Année de réalisation : 2023

ATARI

It works on my machine

Table des matières

Introduction.....	5
Mise en contexte.....	5
Problématique.....	5
Solution.....	6
Analyse.....	7
Méthodologie de gestion de projet.....	7
Objectifs.....	7
Frontend.....	8
Recherche de personnes.....	8
Affichage des résultats.....	8
Outils.....	9
Backend.....	9
API.....	9
Contrôle d'accès.....	10
Vue d'ensemble.....	10
Base de données.....	10
Planning initial.....	11
Conception.....	12
Schéma d'architecture.....	12
Frontend.....	12
Recherche de personnes.....	12
Affichage des résultats.....	13
Stratégie de test.....	14
Réalisation.....	15
Infrastructure.....	15
Choix des technologies.....	15
Meteor.....	15
Docker.....	15
Résultats des tests.....	15
Organisation des résultats de travail.....	15
Glossaire.....	17
Meteor.....	18
Docker.....	19
React.....	20
React est une bibliothèque JavaScript open-source développée par Facebook, utilisée pour la création d'interfaces utilisateur interactives et réactives. Elle est largement utilisée dans le développement d'applications web et mobiles. Voici quelques points clés à connaître sur React :	
React :	20
Conclusion.....	23
Résultats.....	23
Difficultés techniques.....	23
Bilan personnel.....	23
Glossaire.....	24
Annexes.....	25
Résumé du rapport de TPI.....	25
Situation de départ.....	25
Mise en œuvre.....	25
Résultats.....	25
Liens dépôts.....	25
Améliorations possibles.....	25

ATARI

Introduction

L'École polytechnique fédérale de Lausanne (EPFL) est une université suisse de premier plan située à Lausanne, en Suisse. Elle est considérée comme l'une des meilleures écoles d'ingénieurs d'Europe et du monde.

Elle a été fondée en 1969 et est aujourd'hui l'une des deux Écoles polytechniques fédérales suisses, l'autre étant l'ETH Zurich. L'université se concentre sur la recherche scientifique, la technologie et l'enseignement des sciences naturelles et de l'ingénierie.

Le Travail Pratique Individuel (TPI), est un travail que chaque apprenti informaticien de 4ème année doit effectuer pour obtenir son CFC. Le cadre de cette épreuve est fixé par l'article 20 de l'Ordonnance du SEFRI sur la formation professionnelle initiale, son évaluation définie selon les critères d'évaluation ICT, détaillés dans le document fourni par iCQ-VD.

L'intérêt de ce travail est de permettre à l'apprenti de prouver ses aptitudes à traiter des problématiques complexes tout en adoptant une attitude professionnelle. Ce qui se traduit, entre autres, par des échanges formels réguliers avec les experts et le chef de projet, ainsi que la rédaction d'un rapport détaillé indiquant notamment les différentes étapes par lesquelles le candidat a dû passer afin de mener à bien le projet imposé.

Ce TPI se déroule au sein d'une équipe de développement nommée ISAS-FSD qui s'occupe de nombreux systèmes informatiques essentiels au bon fonctionnement de l'école par la mise en application des pratiques *DevOps*.

Mise en contexte

Problématique

« Pour répondre aux différentes demandes des membres de L'EPFL, le personnel technique doit sans cesse aller à la pêche aux informations. Qui peut accréditer une personne, dans quelle(s) unité(s) est-elle accréditée, son e-mail est-il activé, a-t-elle les droits pour accéder à tel ou tel service, a-t-elle activé l'authentification à double facteur, etc. L'outil actuel, IDP-EXOP, permet de répondre de manière centralisée à ces questions en regroupant les informations disponibles sur différents systèmes de l'EPFL.

Cette application est uniquement disponible aux personnes autorisées et accessible dans le réseau de l'EPFL à l'adresse <https://idp-exop.epfl.ch>. L'application ne répond plus aux normes de développement modernes et devient difficile à maintenir. » (cahier des charges, annexe)

Solution

« L'objectif de ce TPI est de redévelopper la solution en répondant aux normes de développement modernes et offrant une API qui permettra d'anticiper les évolutions futures du système d'information de l'EPFL. » (cahier des charges, annexe)

Analyse

Méthodologie de gestion de projet

La méthodologie employée ici est la méthode en cascade. C'est une approche linéaire qui implique une série d'étapes distinctes :

1. Analyse : compréhension des exigences du client et la définition des objectifs du projet.
2. Conception : élaboration d'une solution pour répondre aux besoins du projet. Cela peut inclure la conception de plans, de schémas, de maquettes, etc.
3. Développement / réalisation : création du produit ou du service selon les spécifications définies lors de la phase de conception.
4. Tests : vérification de l'aspect fonctionnel de l'application défini lors de la phase de conception et correction des erreurs s'il en reste.
5. Mise en production : mise en service de l'application dans l'environnement opérationnel.
6. Maintenance : maintient du bon fonctionnement de l'application tout au long de son cycle de vie.

À noter que les étapes 5 et 6 ne seront pas mises en pratique durant ce TPI car elles se déroulent durant une période post-projet.

Chaque étape doit être terminée avant de passer à la suivante. Cette approche présente l'avantage de permettre une planification précise et une gestion rigoureuse du projet, mais à contrario elle peut également être rigide et difficile à adapter en cas d'imprévus, de changements ou d'évolution du projet.

Les risques liés à aux imprévus de la méthodologie seront pondérés par une exagération volontaire de l'estimation du temps nécessaire à la réalisation de chaque tâche.

La méthode en cascade a été choisie pour ce projet car elle permet de se faire une idée précise de l'état du projet par rapport au planning. Les phases de projet sont identiques à la structure de base choisie pour les grands titres de ce rapport de TPI (analyse, conception, réalisation, tests). Cette approche rend davantage visible sa mise en œuvre et encourage son utilisation correcte tout au long du projet.

Un Kanban sera utilisé en complément de la méthodologie principale afin de pouvoir suivre l'évolution du projet sur github.

Objectifs

Dans cette partie, se trouve la liste des objectifs du mandat du cahier des charges analysés et réinterprétés.

Frontend

Recherche de personnes

Dans le cahier des charges, la fonction se nomme «recherche de personnes».

L'utilisateur doit pouvoir rechercher des personnes dans un champ. Celui-ci doit présenter la possibilité de faire des recherches en fonction de plusieurs attributs de la personne :

son numéro de téléphone, sciper, gaspar, non, prénom et adresse mail.

Le champ est capable d'adapter automatiquement sa recherche en fonction du type d'information saisie :

- Si le texte de la saisie ne contient que des chiffres, la recherche devra être faite en fonction du sciper de la personne ou de son numéro.
- Si la saisie contient un point ou une arobase, la recherche devra être faite en fonction du mail de la personne
- Si la saisie contient du texte et/ou des chiffres sans arobase ni point, la recherche devra être faite sur tous les champs.

La fonctionnalité de recherche partielle doit être conçue de manière à limiter la charge du back-end et le nombre de résultats affichés : limite de 10 entrées.

Les difficultés identifiées de cette tâche sont de pouvoir optimiser la fonction de recherche et de devoir permuter entre les différents attributs en fonction du type de saisie effectuée.

Durée estimée de la tâche : 16h



The screenshot shows a web form titled "Identification des personnes". It contains three input fields with labels: "Sciper/User/Email:" with the value "123456 / username / prenom.nom@epfl.ch", "Nom/Prenom:" with the value "nom autocomplété", and "Téléphone:" with the value "[+41 21 69] 11122". Below the fields are two buttons: "Chercher" and "Effacer".

Figure 1: écran d'accueil de l'application actuelle (c.f. cahier des charges en annexe)

Affichage des résultats

Le frontend doit être capable d'afficher les mêmes informations concernant les utilisateurs que celles actuellement disponibles sur le site actuel de idp-exop.epfl.ch (accès authentifié requis).

Si les résultats de recherche ne concernent plus qu'une seule personne (que ce soit parce qu'il n'y a plus qu'une seule correspondance dans la recherche partielle, ou parce que l'utilisateur a choisi explicitement un résultat de recherche partielle), une vue d'ensemble de la personne s'affiche avec ses informations.

Kermit La Grenouille

Tools:

Accreds (copy sciper 1st)
Check AD (copy username 1st)

Check LDAP
ServiceNow


[login people](#)

Prénom, Nom: Kermit, La Grenouille

Sexe: Homme

Sciper créé par: SID

Sciper, Username: 133134, lagrenou

Téléphone, Local: [MA A0 393](#)

Unité(s):

1. SI Services Académiques Gestion
[EPFL](#) → [VP0-SI](#) → [ISAS](#) → [ISAS-GE](#)
Fonction: Consultant
Statut: Hors EPFL

2. Middleware Services
[EPFL](#) → [VP0-SI](#) → [ITOP](#) → [ITOP-MWS](#)
Fonction: Consultant
Statut: Hors EPFL

Active Directory:

Domaine\login:	INTRANET\133134
Status du compte:	Compte désactivé
Expiration du compte:	Jamais
Dernière connexion :	21 Oct 2020 08:32
Dernier mot de passe erroné:	25 Oct 2020 21:25
Nombre d'essais erronés de password:	1

Figure 2: écran de détails d'une personne : Kermit La Grenouille (c.f. cahier des charges en annexe)

Durée estimée de la tâche : 8h

Outils

Des liens vers les différents outils utilisés par le service desk doivent être présents sur la page d'affichage des résultats :

- Un lien vers <https://accred.epfl.ch/>
- Un lien vers <https://windows.epfl.ch/checkad/default.aspx>
- Un lien vers une page de l'application qui affiche tous les champs LDAP de la personne
- Un lien vers la page ServiceNow de la personne
- Un lien vers la page de statut du mail : <https://mailwww.epfl.ch/emailStatus.cgi?query>
- Un lien de la page people de la personne : <https://search.epfl.ch/>

Backend

API

Le projet dispose d'une API qui centralise des informations provenant de différentes sources et fournit des données uniformisées au frontend. Cette API permet également de créer, modifier ou supprimer différentes sources de données et facilite les évolutions du projet (comme l'ajout d'un deuxième client à l'API). Des mesures doivent être prises pour assurer que l'API soit séparée du frontend, qu'elle ne soit pas librement accessible et que l'accès à l'API soit authentifié.

Dans le cadre du TPI, l'API devra être connectée à la base de données, tandis que la connexion au LDAP pourra être effectuée directement depuis le serveur, sans nécessiter d'authentification. Il ne sera pas nécessaire de se connecter à l'Active Directory ou à d'autres services (tels que websrv.epfl.ch, une API qui fournit des informations sur les rôles d'une personne).

Contrôle d'accès

Le contrôle d'accès à l'application (frontend) doit être géré en fonction des rôles, des droits ou des groupes auxquels appartiennent les utilisateurs. Ces informations sont fournies par l'IdP3 de l'EPFL, Tequila. Il est possible que les informations présentées aux utilisateurs varient à l'avenir en fonction de leurs droits, par exemple un utilisateur de la faculté A ne pourra pas voir les informations des utilisateurs de la faculté B. Bien qu'il ne soit pas nécessaire d'implémenter cette fonctionnalité pour le moment, elle doit être prise en considération lors de la conception de la solution.

L'accès à l'API est pour le moment limité strictement à l'application (frontend) et doit être authentifié. L'accès authentifié à l'API par d'autres application est envisagé à l'avenir. La gestion des tokens ou des "rate limit" n'est pas obligatoire mais doit être prise en compte pour permettre des améliorations futures.

Vue d'ensemble

En résumé, ...

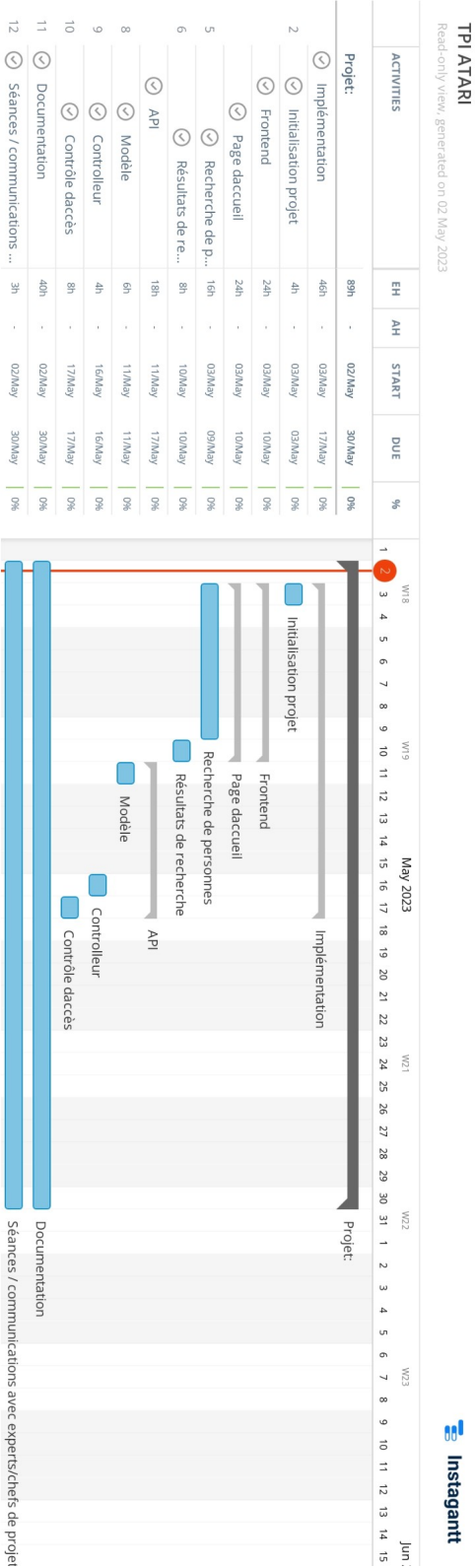
Base de données

La base de données est constituée par défaut comme suit :

1. champ 1
2. champ 2

[image schéma db]

Planning initial



Conception

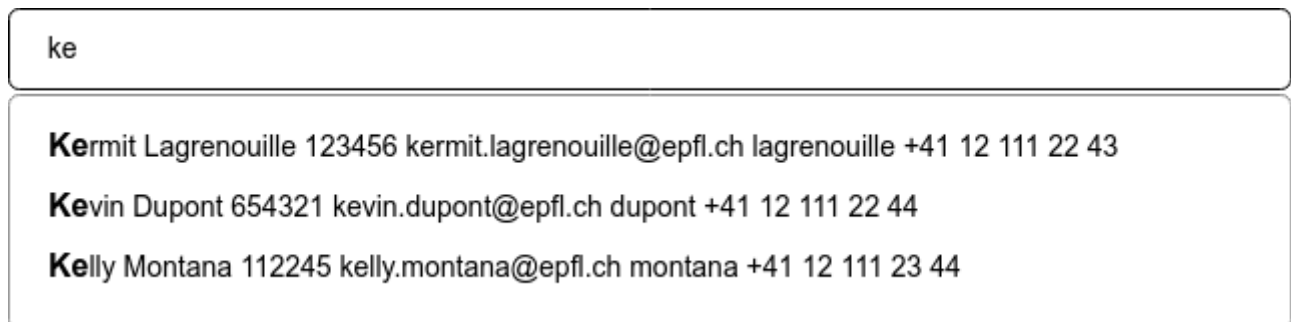
Conception : élaboration d'une solution pour répondre aux besoins du projet. Cela peut inclure la conception de plans, de schémas, de maquettes, etc.

Schéma d'architecture

Frontend

Recherche de personnes

Cette fonctionnalité devra être capable d'effectuer une recherche parmi les membres de l'EPFL.

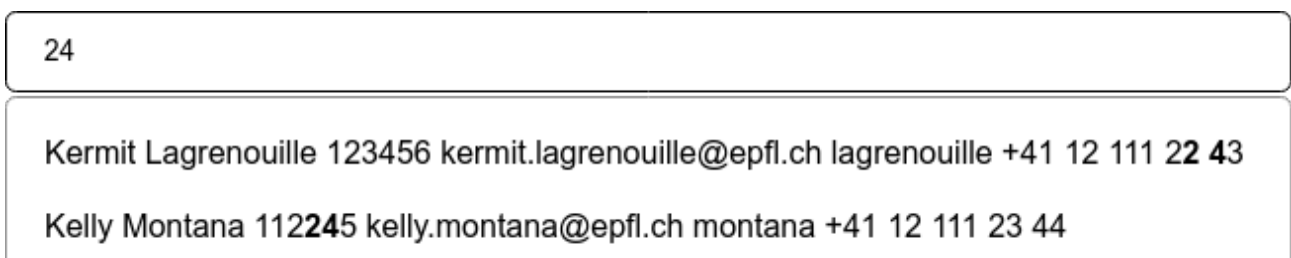


ke
K ermit Lagrenouille 123456 kermit.lagrenouille@epfl.ch lagrenouille +41 12 111 22 43
K evin Dupont 654321 kevin.dupont@epfl.ch dupont +41 12 111 22 44
K elly Montana 112245 kelly.montana@epfl.ch montana +41 12 111 23 44

Figure 3: Recherche en fonction du prénom de l'utilisateur

Lorsque l'utilisateur commence à écrire dans le champ de recherche, des suggestions de personnes en fonction du texte déjà écrit devront être présentées à l'utilisateur.

La difficulté de cette fonctionnalité réside dans le choix du champ dans lequel la recherche doit être effectuée et d'adapter la recherche en fonction du type de saisie, c'est à dire



24
K ermit Lagrenouille 123456 kermit.lagrenouille@epfl.ch lagrenouille +41 12 111 22 43
K elly Montana 112245 kelly.montana@epfl.ch montana +41 12 111 23 44

Figure 4: La recherche peut être effectuée sur d'autres champs, comme le numéro sciper ou le numéro de téléphone

Note : les caractères en gras sont présents à titre d'illustration et d'aide à la compréhension. Ils ne seront pas présents dans l'application en l'état rendue à la fin de ce TPI.

Affichage des résultats

Stratégie de test

Par soucis de clareté et de simplicité, la procédure de test est effectuée de façon manuelle.

Voici les tests effectués :

Description	Marche à suivre	Résultat attendu

Réalisation

Infrastructure

L'application

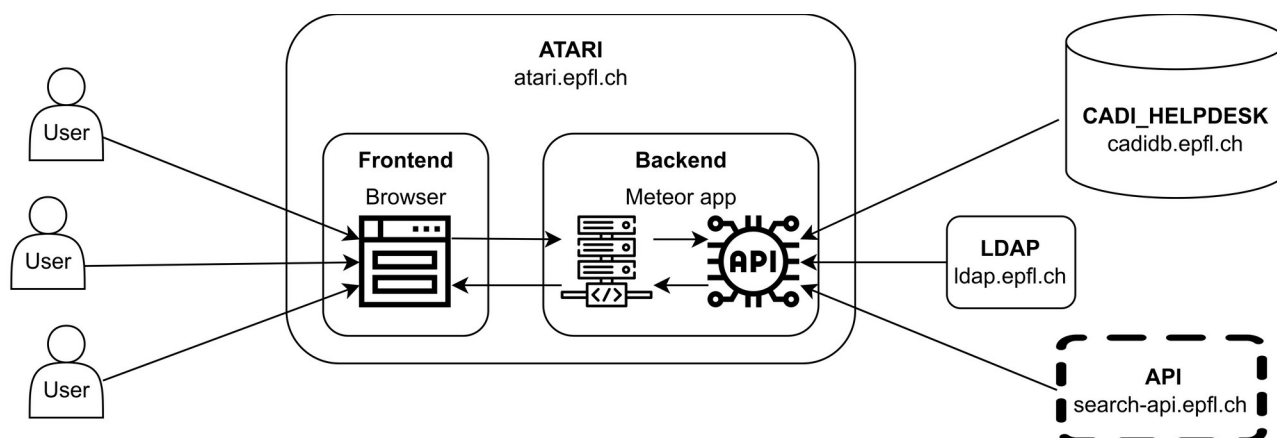


Figure 5: Architecture du site

Choix des technologies

Afin de permettre une bonne cohérence de l'ensemble de la solution logicielle à implémenter, il est important de pouvoir distinguer quelles sont les meilleures technologies qui peuvent être utilisées. Les choix de ces dernières ont été faits en fonction de plusieurs critères, notamment le niveau de maturité, la fiabilité ainsi que la simplicité d'utilisation.

Meteor

Meteor a été choisi, car il s'agit d'un framework puissant qui permet de créer des applications web réactives de manière efficace. Sa combinaison de réactivité, d'isomorphisme et de gestion de packages simplifie la mise en place d'un projet et permet de se concentrer pleinement sur le développement des fonctionnalités.

Docker

Pour l'environnement de développement, l'application doit être entièrement virtualisée en utilisant la technologie de conteneurisation Docker. Il s'agit d'un des points spécifiques techniques décrits dans le cahier des charges.

React

Pour le développement frontend, la librairie React a été choisie car elle présente l'avantage d'offrir une liberté totale dans le choix de la structure du frontend de l'application. Elle permet également de découper l'application en petits modules séparés et réutilisables, appelés «composants», qui

encouragent l'utilisation du principe DRY, (don't repeat yourself). Sa simplicité d'utilisation en fait pour beaucoup de développeurs, la librairie idéale pour le frontend.

Recherche de personnes

La recherche de personnes est une fonctionnalité difficile à implémenter car s'il est en principe possible rapide d'effectuer une recherche sur parmi des milliers d'entrées sur une base. elle doit pouvoir en temps réel effectuer une recherche sur un ou plusieurs champs parmi des dizaines de milliers d'entrées.

L'approche préconisée dans le cahier des charges consiste à effectuer des requêtes dans la base de données à intervalle régulier afin de mettre à jour les résultats suggérés.

Une autre approche consisterait à charger localement uniquement les champs sur lesquels on souhaiterait effectuer une recherche, puis, lorsque l'utilisateur est sélectionné, charger les éléments restants.

Une comparaison a été faite entre la méthode classique et une liste virtualisée, et le constat est sans appel : (démontrer et montrer les autres possibilités, comme par exemple faire des requêtes directement à l'API)

Il est également possible d'utiliser des fonctions javascript à la place de regex (comme par exemple la méthode «includes»), mais celles-ci auraient eu un impact négatif sur les performances de l'application, la lisibilité du code, et rendu l'implémentation beaucoup plus compliquée.

Résultats des tests

Voici les résultats des tests menés sur le projet :

Description	Marche à suivre	Résultat attendu	Résultat

Organisation des résultats de travail

To do

Tous les jours à 18h, sauvegarde en utilisant la stratégie de backup 3-2-1
<https://www.backblaze.com/blog/the-3-2-1-backup-strategy/>

Donc une copie dans une archive dossier compressé en l'état chaque soir dans un drive, un autre sur github (commit), et un autre sur un disque dur externe. L'historique est retracable.

ATARI

Glossaire

ATARI

ATARI

Conclusion

Résultats

Difficultés techniques

Bilan personnel

Glossaire

Annexes

Ajouter cahier des charges

Résumé du rapport de TPI

Situation de départ

Mise en œuvre

Résultats

Liens dépôts

Lien du dépôt ATARI :

Améliorations possibles

Parmi les perspectives d'améliorations futures, nous avons...

ATARI