



# 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* »  
Anonymous

# Résumé du rapport de TPI

## Situation de départ

Afin de répondre aux diverses demandes des membres de l'EPFL, le personnel technique est constamment confronté à la recherche d'informations. Ils doivent déterminer qui est habilité à accréditer une personne, dans quel(s) département(s) cette accréditation est valide, si son adresse e-mail est activée, si elle possède les autorisations pour accéder à tel ou tel service, si elle a activé l'authentification à double facteur, et bien d'autres informations. L'outil actuel, IDP-EXOP, centralise ces informations provenant de différents systèmes de l'EPFL pour fournir des réponses. Cependant, cette application n'est accessible qu'aux personnes autorisées au sein du réseau de l'EPFL, via l'adresse <https://idp-exop.epfl.ch>. Malheureusement, cette application ne répond plus aux normes de développement modernes et devient difficile à maintenir. L'objectif de ce projet de travail de fin d'études (TPI) est de reconstruire la solution en respectant les normes de développement modernes et en fournissant une API qui permettra d'anticiper les évolutions futures du système d'information de l'EPFL.

## Mise en œuvre

Le fonctionnalité de recherche de personnes avec des résultats suggérés a été ajoutée.

Une page contenant toutes les informations de l'utilisateur sélectionné a été ajoutée.

L'API permettant d'aller chercher des informations dans des sources de données externes a été implémentée dans la partie backend de l'application Meteor.

## Résultats

La majorité des objectifs fixés ont été réalisés. Les résultats obtenus témoignent d'une application utilisable dont l'ensemble des fonctionnalités est satisfaisant. Des améliorations sont citées dans le rapport de ce TPI et garantissent la présence d'un fil rouge pour la continuation du projet. L'ensemble du code effectué durant ce TPI est disponible à l'adresse suivante : <https://github.com/epfl-si/ATARI>.

## Table des matières

Résumé du rapport de TPI.....	3
Situation de départ.....	3
Mise en œuvre.....	3
Résultats.....	3
Introduction.....	6
Mise en contexte.....	6
Problématique.....	6
Analyse.....	7
Méthodologie de gestion de projet.....	7
Objectifs.....	7
Frontend.....	8
Recherche de personnes.....	8
Affichage des résultats.....	8
Backend.....	10
API.....	10
Contrôle d'accès.....	10
Planning initial.....	11
Conception.....	12
Choix des technologies.....	12
Meteor.....	12
Docker.....	12
React.....	12
Schéma d'architecture.....	13
Base de données.....	13
Frontend.....	13
Recherche de personnes.....	13
Affichage des résultats.....	14
Informations générales.....	14
Menus déroulants unités.....	15
Stratégie de test.....	15
Réalisation.....	16
Liens dépôts.....	16
Recherche de personnes.....	16
Approches possibles.....	16
Création composant.....	16
Optimisation des performances.....	17
Test de réactivité.....	17
Types de recherche.....	17
Difficultés rencontrées.....	17
Résultats.....	18
Affichage des résultats.....	18
Général.....	18
Unités.....	19
Outils.....	20
Responsive design.....	20
Backend.....	21
Communications avec DB.....	21
Interfaces.....	21
Contrôle d'accès.....	21
Résultats des tests.....	22

Organisation des résultats de travail.....	22
Améliorations possibles.....	23
Glossaire.....	24
Accréditation.....	24
Active directory.....	24
API.....	24
Base de données.....	24
Docker.....	24
Gaspar.....	25
HTTP.....	25
Identity provider.....	25
LDAP.....	25
Meteor.....	26
React.....	27
Sciper.....	27
Conclusion.....	28
Bilan personnel.....	28
Annexes.....	29
Journal de travail.....	30
Procès verbaux.....	39
Test responsive design.....	43
Cahier des charges.....	45

# 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, l'innovation 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 du 1<sup>er</sup> novembre 2013 <https://www.fedlex.admin.ch/eli/cc/2013/862/fr> sur la formation professionnelle initiale, son évaluation définie selon les critères d'évaluation ICT, détaillés dans le document fourni par iCQ-VD. <https://www.tpivd.ch/files/cfc-ordo2k14/2.%20Criteres%20d%20evaluation%20TPI.PDF>

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 de l'EPFL nommée ISAS-FSD (SI Services académiques) 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)

# 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 : maintien 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.

## Objectifs

Dans cette partie, se trouve la liste des objectifs du mandat du cahier des charges analysés et réinterprétés. Une estimation du temps nécessaire à la réalisation de la tâche sera également précisée.

## 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, *numéro sciper*, *nom d'utilisateur gaspar*, nom, prénom et adresse mail.

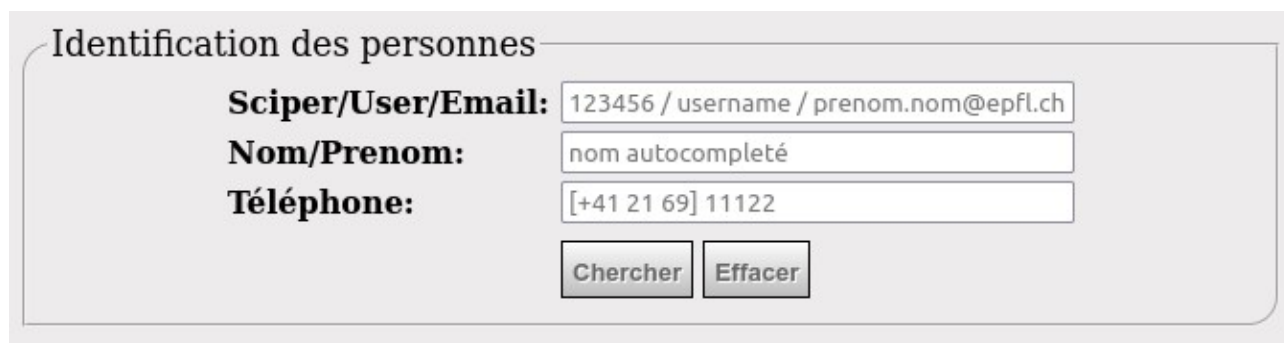
Les résultats de la recherche effectuée via le champ prévu à cet effet s'adaptent automatiquement en fonction du type d'information saisie :

- 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



Identification des personnes

**Sciper/User/Email:** 123456 / username / prenom.nom@epfl.ch

**Nom/Prenom:** nom autocomplété

**Téléphone:** [+41 21 69] 11122

Chercher 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 + VPN 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

**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

  
[login people](#)

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

## 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/>

Durée estimée de la tâche : 8h

## 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).

Durée estimée de la tâche : 10h

### 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 (identity provider)* 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.

Durée estimée de la tâche : 8h

Jérôme Cosandey



# Conception

La phase de conception dans la méthode en cascade consiste à concevoir en détail l'architecture, la structure et les fonctionnalités du système logiciel à développer. Dans cette partie, se trouve la définition de l'architecture globale du système, les différents modules qui la composent ainsi que leurs interactions.

## 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 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

Le projet utilise docker. Il s'agit d'un des points spécifiques techniques décrits dans le cahier des charges. Son utilisation présente plusieurs avantages, notamment d'encapsuler les applications et leurs dépendances, ce qui permet d'exécuter des applications de manière indépendante, sans conflits avec d'autres applications ou le système d'exploitation sous-jacent. Cela permet également de garantir à tous les contributeurs de travailler sur le même environnement de développement.

### 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.

## Schéma d'architecture

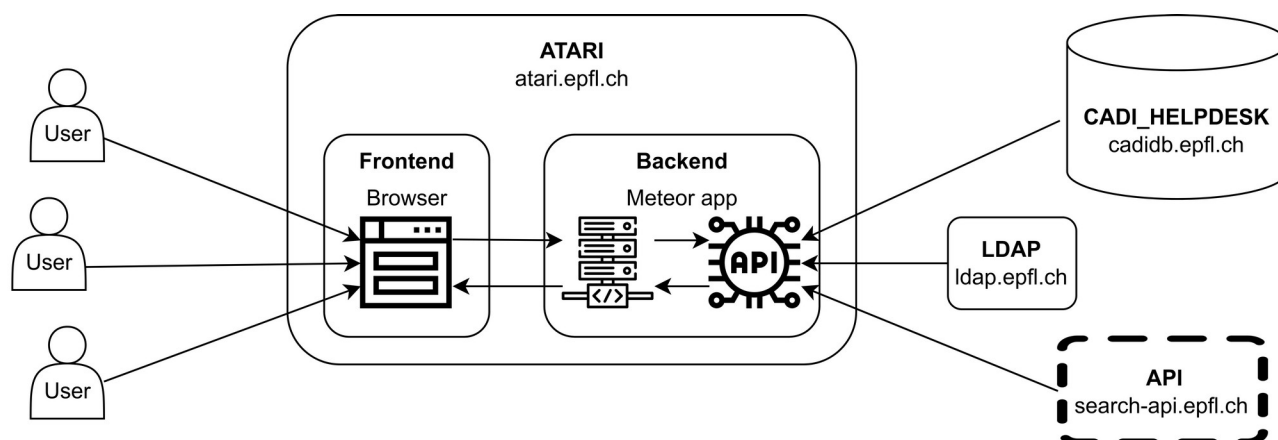


Figure 3: Architecture du site

## Base de données

Pour fournir les données nécessaires au bon fonctionnement de l'application, l'application est connectée à une base de données, CADI\_HELPDESK. Cette base de données contient de nombreuses informations concernant les membres de l'EPFL, utiles pour les activités du service de support de niveau 1 de l'EPFL, le service desk.

## Frontend

### Recherche de personnes

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

**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 4: Recherche en fonction du prénom de l'utilisateur

Lorsque l'utilisateur commence à saisir du texte dans le champ de recherche, des suggestions de personnes basées sur le texte déjà saisi doivent être affichées à l'utilisateur. Cette fonctionnalité est un exemple d'interface homme-machine de recherche incrémentale. Elle est également souvent appelée «search as you type» et définit une interface permettant de rechercher une ou plusieurs occurrences de chaînes de caractères dans un ensemble d'éléments pré-définis. Voici le

lien d'une page wikipedia expliquant en détails le concept :

[https://fr.wikipedia.org/wiki/Recherche\\_incr%C3%A9mentale](https://fr.wikipedia.org/wiki/Recherche_incr%C3%A9mentale)

24

Kermit Lagrenouille 123456 kermit.lagrenouille@epfl.ch lagrenouille +41 12 111 22 43

Kelly Montana 112245 kelly.montana@epfl.ch montana +41 12 111 23 44

Figure 5: 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. Cependant ils figurent parmi la liste des améliorations futures dans le rapport.*

## Affichage des résultats

Cette fonctionnalité consiste en l'affichage des informations concernant la personne recherchée. Les informations présentées doivent être claires et lisibles afin de simplifier leur lecture par les utilisateurs finaux. Pour y parvenir, il est nécessaire d'effectuer quelques recherches ainsi que de créer une liste exhaustive de toutes les informations qui doivent y figurer dans le cadre de ce TPI.

Il est important de garder en tête que des informations supplémentaires pourront être rajoutées par la suite sur cette page et que donc, des mesures doivent être prises pour simplifier l'ajout de nouvelles informations à l'avenir. Ces mesures seront présentées en détails dans la partie réalisation.

## Informations générales

Cette section regroupe les informations principales de l'individu recherché. Elles sont présentées sous la forme d'un tableau liste au style conforme à *Element*, la charte graphique officielle de l'EPFL. Le code servant de template à l'élément en question est disponible à cette adresse :

<https://epfl-si.github.io/elements/#/organisms/contact>

Les champs présents à titres d'exemples sont remplacés les suivants :

prénom, nom, numéro sciper, numéro de téléphone, adresse email, nom d'utilisateur gaspar, genre.

Informations générales
Prénom Nom : Gary Lescargot
Sciper : 123456
Téléphone : +41 12 345 67 89
Email : gary.lescardot@epfl.ch
Gaspar : lagrenouille
Genre : hermaphrodite

Figure 6: Maquette informations générales

## Menus déroulants unités

Cette section regroupe l'ensemble des unités dont la personne fait partie. Elle contient autant de menus déroulants que d'unités auxquelles le candidat appartient. Le nom de l'unité ainsi que la fonction occupée par l'individu sont inscrits sur le menu. Le premier est ouvert par défaut. La présentation est similaire à celle que l'on peut retrouver sur une page personne du site [people.epfl.ch](http://people.epfl.ch)

Lorsque les menus en question sont cliqués, ils révèlent des informations supplémentaires d'informations utiles détaillées ci-dessous :

### Adresse

Présentée sur plusieurs lignes, le numéro qui accompagne chaque fragment d'adresse correspond simplement à son numéro de la ligne.

### Numéro(s) de téléphone

Le ou les numéros de téléphones correspondants au post occupé dans l'unité en question.

### Bureau

Le nom du bureau accompagné d'un lien cliquable renvoyant sur le site [plan.epfl.ch](http://plan.epfl.ch) avec le bureau présélectionné.

### Hierarchie unité

Une représentation visuelle de l'emplacement de l'unité par rapport au reste de la hiérarchie.

**Menus déroulants unités**

Unité 1 Δ

adresse 1	téléphone(s)
adresse 2	bureau
adresse 3	
adresse 4	hierarchie unité

Unité 2 ▽

Unité 3 ▽

Figure 7: menus déroulants unités

## Stratégie de test

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

Voici les tests à effectuer :

Description	Marche à suivre	Résultat attendu
Serveur keycloak	Se rendre à la racine du projet et taper la commande «docker-compose up»	Le container du serveur Keycloak démarre
La page d'accueil	Ouvrir un navigateur et se rendre à l'adresse « <a href="http://localhost:3000">http://localhost:3000</a> »	La page d'accueil du site s'affiche
Recherche partielle	Une fois loggé, saisir quelques caractères dans la barre de recherche	Les résultats incluant les caractères renseignés sont suggérés
Affichage des détails de la personne	Sélectionner un résultat dans le champ de recherche	La page de détails de la personne concernée s'affiche

# Réalisation

## Liens dépôts

Lien du dépôt ATARI : <https://github.com/epfl-si/ATARI>

Lien du dépôt de la documentation : [https://github.com/SaphireVert/ATARI\\_documentation](https://github.com/SaphireVert/ATARI_documentation)

## Recherche de personnes

La recherche de personnes est une fonctionnalité difficile à implémenter car elle doit pouvoir en temps réel effectuer une recherche sur un ou plusieurs champs parmi des dizaines de milliers d'entrées. Cela doit être fait sans surcharger le réseau ni la base de données.

### Approches possibles

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.

Cette approche présente l'avantage de ne rien stocker en local sur le navigateur car la recherche est effectuée sur une base de données distante avant d'envoyer des suggestions de résultats au navigateur, ce qui permet de réduire le temps de chargement initial de la page. Cependant, cette approche nécessite de limiter la quantité de requêtes effectuées par le client afin de ne pas surcharger le réseau et la base de données.

Une autre approche consiste à charger localement uniquement les champs sur lesquels on souhaiterait effectuer une recherche, puis, lorsque l'utilisateur est sélectionné, charger les éléments restants. Cette approche a l'avantage de très peu solliciter la base de données car en dehors du chargement initial, les informations de une seule personne seront recherchées dans la base de données

Après validation avec le chef de projet, la deuxième approche a été choisie.

### Création composant

L'API, au moment où le composant a été intégré, n'est pas encore connectée à la base de données 'CADI\_HELPDESK'. Pour palier à ce problème, un bouchon [https://fr.wikipedia.org/wiki/Bouchon\\_\(informatique\)](https://fr.wikipedia.org/wiki/Bouchon_(informatique)) a dû être créé pour fournir des données temporaires au champ de recherche de personnes. Il s'agit d'un substitut de l'API qui n'existe pas encore.

Une recherche a été effectuée sur le site web de la librairie de composants officielle de React, material-ui <https://mui.com/> car il est probable qu'il existe déjà un composant permettant de faire de l'autocomplétion sur une liste. Il existe un composants se nommant autocomplete, <https://mui.com/material-ui/react-autocomplete>, qui semble correspondre aux besoins de la fonctionnalité en cours d'implémentation.



## Optimisation des performances

Il est possible d'y intégrer une librairie tierce, nommée react-window <https://github.com/bvaughn/react-window>, qui permet d'effectuer une virtualisation le dom de la liste, ce qui permet de grandement limiter les ressources nécessaires au rendu de celle-ci lorsqu'elle est mise à jour (à chaque fois qu'un caractère est rajouté ou enlevé dans le champ de recherche). Cet aspect est d'autant plus important lorsque l'on travaille avec des grandes listes, ce qui sera le cas dans ce projet qui, lorsqu'il sera déployé en prod, devra être capable de supporter une liste de 30 000 personnes. React-window permet également de réduire la quantité de mémoire utilisée en évitant d'allouer trop de noeuds du DOM. Tout cela va nous permettre d'avoir une application TRÈS réactive, et donc agréable à utiliser pour l'utilisateur final. Un exemple d'implémentation de la fonctionnalité avec une liste de 10 000 entrées générées aléatoirement, est disponible ici : <https://mui.com/material-ui/react-autocomplete/#virtualization>. Cet exemple a été réutilisé dans le projet. La fonctionnalité est disponible à l'essai sur le site codesandbox à l'adresse suivante <https://codesandbox.io/s/r43d57?file=/demo.tsx>. Le site codesandbox permet de collaborer, partager et essayer du code en ligne.

## Test de réactivité

Toujours sur codesandbox, une comparaison a été faite sur une liste de 30 000 entrées entre la méthode classique et une liste virtualisée, et le constat est sans appel : alors qu'une liste "classique" nécessite plusieurs secondes à se mettre à jour, l'utilisation d'une liste virtualisée affiche les résultats instantanément.

## Types de recherche

Le champ doit s'adapter aux types de données saisies. Pour ce faire, des regex ont été choisies. 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.

## Difficultés rencontrées

De nombreuses difficultés ont été rencontrées lorsqu'il s'agissait d'intégrer React-window dans cette partie était de trouver quels étaient les bons éléments de material-ui qui pouvaient être utilisés et où ils étaient sur le site et de trouver les bonnes props à utiliser, elles sont disponibles sur <https://mui.com/material-ui/api/autocomplete/>, notamment pour récupérer la saisie de l'utilisateur et comment intégrer tout ça dans le code du projet. Beaucoup de débogage a dû être fait et il y a eu quelques problèmes de types typescript. Une difficulté supplémentaire était d'adapter le code d'exemple de la virtualisation de la liste pour le faire fonctionner avec des données du projet au lieu d'une liste générée aléatoirement.

## Résultats

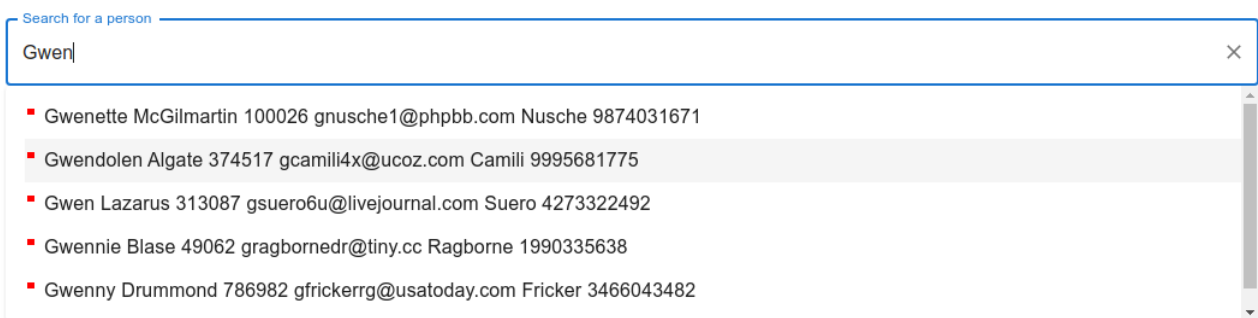
Les résultats obtenus sont des champs à l'apparence esthétique et au design épuré.



Search for a person X

Figure 8: Champ de recherche lorsqu'il n'est pas sélectionné

Les résultats de recherche partielle s'affichent lorsqu'une chaîne de caractères est saisie.



Search for a person X

- Gwenette McGilMartin 100026 gnusche1@phpbb.com Nusche 9874031671
- Gwendolen Algate 374517 gcamili4x@ucoz.com Camili 9995681775
- Gwen Lazarus 313087 gsuer06u@livejournal.com Suero 4273322492
- Gwennie Blase 49062 gragbornedr@tiny.cc Ragborne 1990335638
- Gwenny Drummond 786982 gfrickerrg@usatoday.com Fricker 3466043482

Figure 9: Résultats recherche partielle

## Affichage des résultats

Pour l'implémentation de cette fonctionnalité, il était nécessaire d'utiliser des éléments de la charte graphique de l'EPFL, l'élément utilisé est disponible à l'adresse suivante :


<https://epfl-si.github.io/elements/#/organisms/contact>

## Général

Pour l'implémentation de cette section, le template disponible sur Element a été utilisé puis les valeurs ont été remplacées avec celles de la personne.

Search for a person

Daisy Rable



daisy.rable@epfl.ch

+41 1234567890

Copy

**Daisy Rable**

**Général**

Nom complet

Daisy Rable

Sciper

316897

Téléphone

+41 1234567890

Email

daisy.rable@epfl.ch

Nom d'utilisateur

rable

Genre

F

Sciper géré par

SAP

Figure 10: Affichage des informations d'une personne

## Unités

Les unités dont la personne fait partie sont affichées grâce à un composant fonctionnel React, nommé «Unit». Il inclut un menu ouvrable et refermable et qui permet d'afficher les informations correspondantes de l'unité en question. Il utilise les informations fournies par l'API de Meteor.

La principale difficulté de cette fonctionnalité est la création des composants d'affichage des unités car le composant doit permettre à la liste de s'ouvrir et de se fermer. Cette propriété est gérée par la présence d'un attribut de classe d'une balise HTML.

Ci-dessous se trouve un exemple du composant fini :

**Apprenti informaticien, Développement Full-Stack**

**EPFL VPO-SI ISAS-FSD**  
 INN 012 (Bâtiment INN)  
 Station 14  
 CH-1015 Lausanne

Unité: [EPFL](#) > [VPO-SI](#) > [ISAS](#) > [ISAS-FSD](#)  
 Statut: Personnel

Figure 11: Détails unité

Une personne peut appartenir à plusieurs unités, auquel cas une liste exhaustive de celles-ci sera affichée.

<b>Professeur titulaire, SMT - Enseignement</b>	▼
<b>Membre passif, AGEPoly - RoboPoly</b>	▼
<b>Professeur titulaire, EDRS - Enseignement</b>	▼
<b>Professeur titulaire, Groupe SCI STI FMO1</b>	▼

Figure 12: Lorsque la personne appartient à plusieurs unités, elles sont toutes affichées

## Outils

Afin d'ajouter les outils il a fallu générer dynamiquement une URL qui contient le sciper de la personne en paramètres, par exemple pour la page people : <https://people.epfl.ch/133134> le sciper à la fin de l'URL a été remplacé par la variable contenant le sciper.

**Tools**

Figure 13: Les outils couramment utilisés par le service desk sont inclus

## Responsive design

La conception réactive de l'application a été testée sur l'application Responsively et le test en question démontre clairement l'adaptation du site web sur les téléphones mobiles et tablettes. Une capture d'écran est disponible en annexe.

## Backend

Afin d'alimenter le frontend créé avec des vraies données, un backend doit être fourni à l'application. Celui-ci va permettre de récupérer et traiter l'information avant de l'envoyer au client.

### Communications avec DB

Afin de permettre à l'application de se connecter à la base de données, le module npm «mysql» a été utilisé.

### Interfaces

Afin de pouvoir communiquer des données au frontend, le framework Meteor utilise son propre protocole de communication : DDP. Ce protocole fonctionne grâce à un websocket et permet la transmission de données ainsi que leur mise à jour automatique. C'est à dire que si pour une raison quelconque, une donnée change dans le backend, ce changement sera automatiquement retransmis au client frontend sans que celui-ci ait besoin d'effectuer des actions. Pour mettre ceci en place, le système de souscription de meteor est utilisé.

### Contrôle d'accès

Afin de restreindre l'accès aux seules personnes/entités autorisées, trois mesures de sécurité ont été prises : La première est l'authentification du client en utilisant l'IdP tequila de l'EPFL. La deuxième est l'ajout d'un système de policy sur fonctions de souscription de Meteor qui permet de sécuriser l'accès aux données sensibles. La troisième mesure est la présence d'un système de rôles au sein de l'application. De ce fait, si un utilisateur est authentifié mais ne possède pas le bon rôle, il ne pourra pas accéder aux données des personnes.

# Résultats des tests

Voici les résultats des tests menés sur le projet :

Description	Marche à suivre	Résultat attendu	Résultat
Serveur keycloak	Se rendre à la racine du projet et taper la commande «docker-compose up»	Le container du serveur Keycloak démarre	Fonctionnel
La page d'accueil	Ouvrir un navigateur et se rendre à l'adresse «http://localhost:3000»	La page d'accueil du site s'affiche	Fonctionnel
Recherche partielle	Une fois loggé, saisir quelques caractères dans la barre de recherche	Les résultats incluant les caractères renseignés sont suggérés	Fonctionnel
Affichage des détails de la personne	Sélectionner un résultat dans le champ de recherche	La page de détails de la personne concernée s'affiche	Fonctionnel

## Organisation des résultats de travail

Afin de garantir l'intégrité des données dans toutes circonstances, l'entièreté du travail est sauvegardé chaque soir sur le stockage cloud de SWITCH (owncloud), le dépôt git dédié à la documentation et un disque dur externe.

Une copie dans une archive zip du travail en l'état est également disponible sur le dépôt GitHub.

Voici son url : [https://github.com/SaphireVert/ATARI\\_documentation](https://github.com/SaphireVert/ATARI_documentation)

L'utilisation de Git permet de retracer avec précision l'historique de chaque fichier dans l'état dans lequel il se trouve lorsque le commit a été effectué.

## Améliorations possibles

Le projet a été conçu avec une approche modulaire. Cela permet de simplifier l'ajout de nouvelles fonctionnalités.

Parmi les possibilités d'amélioration futures mises en place, il y a la possibilité de rajouter d'autres sources de données, comme par exemple la recherche d'informations concernant l'active directory. Pour l'instant, ces informations sont présentes sous la forme de bouchons dans la page d'affichage des résultats. Mais elles pourront être intégrées simplement en ajoutant une nouvelle fonction dans le fichier `server/cadidb/userDetailsCollection.ts`.

Une autre possibilité d'amélioration future serait la création d'une API REST en complément des méthodes de transmission de données propres à Meteor utilisées dans le cadre de ce projet. Cette amélioration peut être faite avec l'aide d'un paquet Meteor conçu à cet effet : <https://meteor-rest.readthedocs.io/en/rtd/rest/>

# Glossaire

## Accréditation

L'accréditation est le processus par lequel un individu obtient un ou plusieurs droits ou statuts officiellement reconnus par l'EPFL.

## Active directory

Active Directory est un service de gestion d'annuaire développé par Microsoft. Il est utilisé principalement dans les environnements Windows pour centraliser et gérer les informations relatives aux utilisateurs, aux groupes et aux ressources d'un réseau.

## API

Une API (Application Programming Interface ou Interface de Programmation Applicative), est une interface fournissant un moyen standardisé d'échange de données et d'interaction entre différentes applications via le protocole *HTTP*. Elle permet à une application d'accéder aux fonctionnalités ou aux données d'une autre application de manière sécurisée et structurée.

## Base de données

Une base de données est un système organisé pour stocker, gérer et récupérer des informations de manière structurée. Elle est conçue pour permettre un stockage efficace, la manipulation et la récupération des données de manière fiable et sécurisée. Les systèmes de gestion de bases de données les plus utilisés sont Oracle et MySQL.

## Docker

Docker est une plateforme open-source de virtualisation légère qui permet de créer, déployer et exécuter des applications de manière portable et isolée. Il utilise la notion de conteneurisation pour encapsuler une application et ses dépendances dans un environnement autonome appelé conteneur. Voici les principaux concepts et fonctionnalités de Docker :

1. Conteneurisation : Docker permet d'empaqueter une application et tous ses composants nécessaires, y compris les bibliothèques, les dépendances et les fichiers de configuration, dans un conteneur. Un conteneur est une instance isolée et portable qui peut être exécutée de manière cohérente sur n'importe quel système d'exploitation prenant en charge Docker.
2. Image : Une image Docker est un modèle immuable qui sert de base à la création de conteneurs. Elle contient un système de fichiers en lecture seule représentant l'application, ainsi que des paramètres de configuration. Les images sont créées à partir de fichiers appelés Dockerfiles, qui spécifient les étapes nécessaires pour construire l'image.
3. Registre : Docker Hub est le registre public officiel de Docker, où les utilisateurs peuvent télécharger et partager des images Docker préconstruites. Il existe également des registres privés pour stocker et distribuer des images internes à une organisation.



4. Docker Engine : C'est le moteur principal de Docker qui permet de créer et de gérer les conteneurs. Il fournit une API et un ensemble d'outils en ligne de commande pour interagir avec les conteneurs et les images.

5. Orchestration : Docker peut être utilisé en conjonction avec des outils d'orchestration tels que Docker Compose et Kubernetes. Ces outils permettent de gérer et de coordonner plusieurs conteneurs, permettant ainsi de déployer des applications complexes et de les mettre à l'échelle facilement.

6. Portabilité : Grâce à la conteneurisation, les applications Docker sont hautement portables. Elles peuvent être exécutées sur différents environnements, qu'il s'agisse d'un ordinateur local, d'un serveur distant ou d'un cloud public, à condition que Docker soit installé sur la machine hôte.

7. Isolation : Les conteneurs Docker sont isolés les uns des autres, ce qui signifie qu'ils s'exécutent de manière indépendante et ne peuvent pas interférer les uns avec les autres. Cela permet d'assurer la cohérence et la sécurité de l'environnement d'exécution des applications.

En résumé, Docker est une technologie de conteneurisation qui facilite la création, le déploiement et l'exécution d'applications dans un environnement isolé et portable. Grâce à Docker, les développeurs peuvent éviter les problèmes liés aux différences d'environnement entre les machines, ce qui simplifie le déploiement et la gestion des applications.

## Gaspar

Gaspar est un nom d'utilisateur, généralement constitué du nom de famille, attribué à chaque membre de l'EPFL. Contrairement au numéro sciper, celui-ci peut être changé ou repris par une autre personne.

## HTTP

Le protocole HTTP (Hypertext Transfer Protocol) est un protocole de communication utilisé pour transférer des données sur le Web.

## Identity provider

Un Identity Provider (IDP), également appelé fournisseur d'identité, est un service qui permet aux utilisateurs de s'authentifier et d'accéder à des systèmes ou des applications en ligne de manière sécurisée. Il agit comme un tiers de confiance chargé de vérifier l'identité des utilisateurs et de fournir des informations d'identification aux services qui en ont besoin.

## LDAP

Le protocole LDAP (Lightweight Directory Access Protocol) est un protocole de communication utilisé pour accéder et gérer des services d'annuaire en réseau. Il fournit un moyen standardisé d'interagir avec des annuaires qui contiennent des informations sur les utilisateurs, les groupes, les ressources et d'autres entités au sein d'un réseau.

## Meteor

Meteor est un framework de développement d'applications web open-source, basé sur JavaScript, qui permet de créer des applications web et mobiles de manière rapide et efficace. Il est souvent décrit comme un "framework complet" car il intègre à la fois le frontend et le backend du développement d'applications.

Voici les éléments clés du framework Meteor :

1. Full-Stack : Meteor permet de développer à la fois le frontend et le backend d'une application en utilisant JavaScript. Il utilise Node.js pour le backend et propose une couche de persistance de données intégrée.
2. Réactivité : La réactivité est une caractéristique centrale de Meteor. Les données sont automatiquement synchronisées entre le serveur et le client, ce qui signifie que les changements effectués côté serveur sont immédiatement reflétés côté client, sans nécessiter de rechargement de page.
3. Isomorphisme : Meteor permet d'écrire du code JavaScript à la fois côté client et côté serveur, ce qui facilite le partage de code entre les deux environnements. Cela permet de créer des applications plus rapides et réactives, car certaines tâches peuvent être exécutées côté client plutôt que d'avoir à faire des requêtes supplémentaires au serveur.
4. Packaging : Meteor utilise un système de gestion de packages qui facilite l'ajout de fonctionnalités à une application. Il existe une vaste bibliothèque de packages disponibles, ce qui permet aux développeurs de gagner du temps en réutilisant du code existant.
5. Déploiement simplifié : Meteor propose des outils intégrés pour le déploiement d'applications. Il est facile de déployer une application Meteor sur des plateformes comme Galaxy ou Heroku, ou de créer des bundles pour un déploiement personnalisé.
6. Écosystème : Meteor bénéficie d'une communauté active de développeurs et propose de nombreux packages, tutoriels et ressources pour faciliter le développement d'applications.

En résumé, Meteor est un framework puissant qui permet de créer des applications web et mobiles réactives de manière efficace en utilisant JavaScript. Sa combinaison de réactivité, d'isomorphisme et de gestion de packages en fait un choix populaire pour le développement d'applications modernes.

## React

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 :

1. Composants : React se base sur le concept de composants réutilisables. Les composants sont des blocs de construction autonomes qui encapsulent le HTML, le CSS et la logique de comportement d'une partie spécifique de l'interface utilisateur. Ils peuvent être imbriqués les uns dans les autres pour construire une interface plus complexe.
2. Virtual DOM : React utilise un Virtual DOM (Document Object Model virtuel). Plutôt que de manipuler directement le DOM réel (la structure HTML de la page), React crée une représentation virtuelle du DOM en mémoire. Lorsqu'un composant est mis à jour, React compare le Virtual DOM avec le DOM réel et effectue uniquement les modifications nécessaires pour refléter les changements, ce qui rend les mises à jour plus efficaces et rapides.
3. Unidirectionnalité des données : React suit un flux de données unidirectionnel, également connu sous le nom de "flux de données descendant". Les données sont transmises de haut en bas dans la hiérarchie des composants. Cela facilite le suivi des données et rend le code plus prévisible et maintenable.
4. JSX : React utilise une syntaxe spéciale appelée JSX (JavaScript XML) qui permet d'écrire du code JavaScript et du code HTML (ou XML) en même temps. Cela permet aux développeurs de décrire la structure de l'interface utilisateur de manière déclarative et intuitive.
5. Réactivité : React facilite la mise à jour dynamique de l'interface utilisateur en fonction des changements de données. Lorsqu'une donnée change, React met à jour de manière optimisée les parties concernées de l'interface utilisateur sans recharger la page entière.
6. Écosystème et composants tiers : React dispose d'une vaste communauté de développeurs et d'un écosystème riche en outils et en bibliothèques tiers. Il existe de nombreux composants et bibliothèques React préconstruits qui peuvent être utilisés pour accélérer le développement d'applications et ajouter des fonctionnalités supplémentaires.
7. Utilisation avec d'autres technologies : React peut être utilisé avec d'autres bibliothèques et frameworks JavaScript. Il est couramment utilisé avec des bibliothèques de gestion d'état telles que Redux ou MobX, ainsi qu'avec des frameworks de développement web tels que Next.js pour des fonctionnalités avancées.

En résumé, React est une bibliothèque JavaScript populaire utilisée pour construire des interfaces utilisateur interactives et réactives. Sa facilité d'utilisation, son efficacité grâce au Virtual DOM et son écosystème actif en font un choix privilégié pour le développement d'applications web et mobiles modernes.

## Sciper

Afin de simplifier les tâches d'administration, l'EPFL utilise des numéros, nommés « *sciper* ». Le numéro sciper est un numéro unique d'identification attribué à chaque collaborateur, étudiant et externe ayant besoin d'être *accédité* à l'EPFL, il est propre à chacun et immuable, c'est à dire qu'il ne peut pas être changé une fois attribué, même si le membre en question quitte l'EPFL.

# Conclusion

Les résultats obtenus ont été satisfaisants, quelques difficultés techniques ont été rencontrées et ont eu pour conséquence de provoquer du retard dans le projet. La quasi-totalité des fonctionnalités demandées ont cependant été implémentées et l'application est presque prête à être déployée.

## Bilan personnel

Ce TPI a été une expérience pleine de surprises et de rebondissements. Elle m'a enseigné à ne pas me précipiter sur la première solution que j'estime viable et à prendre le temps de considérer toutes les autres possibilités qui s'offrent à moi. En agissant ainsi, le recul accumulé me permet d'agir en connaissance de cause.

# **Annexes**

**Journal de travail**

**Procès verbaux**

**Test responsive design**

**Cahier des charges**

ATARI

## **Journal de travail**

**02.05.2023**

### **Visite de l'expert TPI n°1**

Durant cette séance, l'expert a expliqué le protocole du déroulement du TPI et fourni une copie du cahier des charges. Le cahier des charges a été signé par le candidat, l'expert et le chef de projet.

**Catégorie** : Communications avec les experts/chefs de projet

**Durée** : 30 min

### **Création planning initial**

Le cahier des charges a été analysé de manière approfondie afin d'avoir une vue d'ensemble du travail à réaliser. Celui-ci a ensuite été découpé en tâches qui ont été placées dans l'ordre le plus judicieux possible. La méthodologie de gestion de projet pour ce TPI a été choisie. Il s'agit de la méthode en cascade. Le planning initial a été créé en utilisant l'outil Instagantt.

**Catégorie** : Documentation

**Durée** : 5h

### **Rapport de TPI**

Creation des premiers éléments de structure de base du document. L'introduction a été rédigée. La rubrique "mise en contexte" a également été rajoutée.

**Catégorie** : Documentation

**Durée** : 2h30

### **Divers**

Aménagement de l'espace de travail afin de permettre un bon workflow.

Conception et réalisation d'un système de backup adapté au projet. Création d'un dépôt github qui servira pour la sauvegarde, l'archivage et la gestion de version de la documentation.

**Catégorie** : Divers

**Durée** : 2h

### **Communications par mail**

Envoi planning initial + rapport + journal de travail aux experts

**Catégorie** : Communications avec les experts/chefs de projet

**Durée** : 30min

**Bilan journée :**

La difficulté aujourd'hui était d'avoir une vision à la fois globale, détaillée et approfondie du projet car il s'agit d'un pré-requis pour concevoir un planning initial de qualité, qui sera le fil rouge à suivre pour le reste du TPI. Le respect des horaires du TPI aujourd'hui a échoué en raison d'une charge de travail plus grande qu'anticipée. Certains points mentionnés dans les critères d'évaluation (dont la gestion de version de la doc, utilisation correcte de git et de la méthodologie de gestion de projet, ergonomie de l'espace de travail) ont nécessité des dispositions et du temps d'analyse supplémentaire afin d'avoir de très bonnes bases établies dès le départ sachant qu'ils sont pour certains difficilement rattrapables, voire pas rattrapables du tout s'ils ne sont pas bien faits dès le départ du TPI.

**Total heures supplémentaires** : 2h

**03.05.2023**

## **Initialisation du dépôt**

Création du dépôt git du projet. Ce dernier est forké du dépôt react.starterkit (dépôt sert de base de développement pour les applications react conçues à l'EPFL pour l'EPFL). Les éléments "génériques" de l'application tels que le titre et les menus du bord sont remplacés par ceux du future IDP-EXOP.

**Catégorie** : Initialisation projet

**Durée** : 1h

## **Recherche de personnes**

L'API n'est pas encore connecté à la base de données 'CADIHELPDESK'. Il a fallu créer un bouchon [https://fr.wikipedia.org/wiki/Bouchon\(informatique\)](https://fr.wikipedia.org/wiki/Bouchon(informatique)) pour fournir des fausses données au champ de recherche de personnes. Il s'agit d'un substitut temporaire de l'API qui n'existe pas encore.

Une recherche a été effectuée sur le site web de la librairie material-ui de react car il est probable qu'il existe déjà un composant permettant de faire de l'autocomplétion sur une liste. Un des composants disponibles se nommait "Autocomplete", ce qui semble correspondre aux besoins du projet. Après recherche approfondir, il en ressort même qu'il est possible d'y



intégrer react-window, une librairie permettant de virtualiser le dom de la liste, ce qui permet de grandement limiter les ressources nécessaires au rendu de celle-ci lorsqu'elle est mise à jour (à chaque fois qu'un caractère est rajouté ou enlevé). Cet aspect est d'autant plus important lorsque l'on travaille avec des grandes listes, ce qui sera le cas dans ce projet. Il permet également de réduire la quantité de mémoire utilisée en évitant d'allouer trop de noeuds du DOM. Tout cela va nous permettre d'avoir une application TRÈS réactive, et donc agréable à utiliser pour l'utilisateur final. Un exemple d'implémentation de la fonctionnalité avec une liste de 10000 entrées générées aléatoirement, est disponible ici. Il a été réutilisé dans le projet. La fonctionnalité est disponible à l'essai dans un codesandbox. La principale difficulté dans cette partie était de trouver quels étaient les bons éléments de mui et où ils étaient sur le site et de trouver les bonnes props à utiliser, elles sont disponibles ici notamment pour récupérer la saisie de l'utilisateur et comment intégrer tout ça dans le code du projet. Beaucoup de débogage a dû être fait et il y a eu quelques problèmes de types typescript. Une difficulté supplémentaire était d'adapter le code d'exemple de la virtualisation de la liste pour le faire fonctionner avec des données du projet au lieu d'une liste générée aléatoirement.

**Catégorie :** Recherche de personnes

**Durée :** 6h

## Rapport de TPI

Avancement du rapport. La partie "méthodologie de gestion de projet" a été faite et la partie "analyse" a été commencée.

**Catégorie :** Documentation

**Durée :** 1h

## Bilan journée

Résultats obtenus très satisfaisants pour le projet. Beaucoup de marge avait été gardée pour l'initialisation du projet en cas d'imprévu, mais qui a pour finir été terminée en avance. Le temps précieux libéré a permis de travailler plus longtemps que prévu sur la fonctionnalité "recherche de personnes".

## 04.05.2023

Recherche de personnes

Implémentation de la fonctionnalité de "détection du type de saisie". La principale difficulté était

l'utilisation des Regex pour reconnaître des patterns. L'utilisation de stackoverflow pour en chercher des exemples, comme celui-ci pour détecter lorsque la chaîne de caractères ne contient que des chiffres ainsi que du site web Regex101 pour les tester, ont été d'une grande utilité.

**Catégorie :** Recherche de personnes

**Durée :** 4h

## Séance avec le chef de projet

Une séance a été faite avec le chef de projet pour faire le point sur l'état d'avancement du projet ainsi que d'autres éléments. (c.f. procès verbal 04.04.2023)

**Catégorie :** Séances

**Durée :** 1h

## Documentation

La documentation a été avancée :

Chaque élément du cahier des charges a été passé en revue, réinterprété et documenté dans le rapport.

Rédaction du procès verbal de la séance avec le chef de projet.

**Catégorie :** Documentation

**Durée :** 3h

## Bilan journée

Les résultats obtenus sont satisfaisants. Cependant, des difficultés ont été ressenties lorsqu'il a fallu utiliser des regex pour des reconnaissances de patterns.

## 09.05.2023

### Recherche de personnes

Avancement dans la fonction recherche de personnes. Le champ se remplit tout seul lorsque l'utilisateur fait des recherches dans le champ et qu'il ne reste qu'un seul résultat parmi les suggestions. Le composant déclenche automatiquement une fonction qui sera utilisée pour afficher l'utilisateur en question. Beaucoup de difficultés ont été rencontrées durant cette phase d'implémentations.

**Catégorie :** Recherche de personnes

**Durée :** 7h30

**10.05.2023****Recherche de personnes**

Le système d'autocomplétion a été terminé. La recherche peut être effectuée sur plusieurs champs (sciper, nom, prénom, etc...).

Une autre fonctionnalité a été implémentée, et qui permet de détecter lorsque l'utilisateur clique sur un résultat suggéré lors de la recherche, pour ensuite exécuter le `console.log` avec les infos de la personne sélectionnée.

**Catégorie :** Recherche de personne

**Durée :** 4h00

**Détails personne**

Création du design de la page "détails personne" en partant de zéro et en utilisant la technique du "nested layout". Le résultat n'a pas été au rendez-vous car inesthétique...

Catégorie : Détails personne

**Durée :** 3h00

**Séance avec le chef de projet**

Une séance a été faite avec le chef de projet pour faire le point sur l'état d'avancement du projet ainsi que d'autres éléments. (c.f. procès verbal 10.04.2023)

**Catégorie :** Séances

**Durée :** 30min

**Bilan journée**

Du côté de la fonctionnalité de recherche, il reste encore quelques détails à paufiner mais le gros du travail a été fait. Le code doit encore être refactorisé un peu avant d'être commité.

Quant à la page détails personne, les résultats n'ont pas été à la hauteur.

**11.05.2023****Détails personne**

Réimplémentation de la page "détails personne". De nombreuses difficultés ont été rencontrées lors du choix de la disposition des informations et lorsqu'il s'agissait de centrer correctement des div ou des éléments sur la page. Cependant, cette phase s'est conclue avec

des résultats satisfaisants et qui, de plus, respectent parfaitement la lettre la charte graphique de l'EPFL.

**Catégorie** : Détails personne

**Durée** : 6h00

## Modèle

Un moment a été pris pour analyser la situation et planifier les actions devant être entreprises pour mener à bien l'intégration de la fonctionnalité au sein de l'API. Au vu de la méthodologie de gestion de projet choisie, cela correspond aux phases d'analyse et conception.

Pour ce faire, une reconsidération approfondie des éléments constituant du cahier des charges a été faite.

**Catégorie** : Détails personne

**Durée** : 1h30

## Bilan journée

Le projet a pris un peu de retard par rapport au planning. Pour l'instant cela reste raisonnable mais s'il n'est pas rattrapé durant la journée de TPI suivante, il va impérativement falloir faire des heures supplémentaires afin d'éviter toute surcharge de travail vers la fin du TPI.

## 16.05.2023

Le active directory a besoin d'une connexion ldap, on le laisse de côté pour l'instant et on rajoutera à la fin du tpi s'il reste du temps

Il m'a fallu faire beaucoup de recherches dans la DB pour me rendre compte que les infos n'y étaient pas. En fait je dois faire une sorte d' "enum" moi même pour avoir la liste des fonctions correspondantes car il n'y a aucune table correspondante dans la base de données. J'ai pu trouver l'info dans ce fichier (le lien est protégé par des droits d'accès)

Une connexion a été établie pour la db en utilisant un tunnel ssh pour passer par le serveur "bastion". C'est lors d'une séance avec un de mes formateurs que j'ai appris ça.

Du coup j'arrive à récupérer les données nécessaires. J'ai préparé les requêtes sql dont je vais avoir besoin. Il me reste juste à les faire fonctionner correctement dans le back end et acheminer les information aux ayant droit et le tout sera joué !

Le travail a été refactorisé aussi au niveau des menus dépliant des unités. Ils sont chacun des composants maintenant !

**Catégorie :** Détails personne

**Durée :** 10h30

**Heures supplémentaire journée :** 3h

**Bilan journée**

Des heures supplémentaires ont été prises pour rattraper le retard accumulé, mais il reste encore du travail à faire, des heures supplémentaires devront être prises le jour suivant également.

**17.05.2023**

**API**

Les requêtes SQL ont été intégrées avec succès dans le code. Le code a également été refactorisé afin de simplifier son entretien sur le long terme

**Catégorie :** API

**Durée :** 7h00

**Bilan journée**

C'était un peu le rush aujourd'hui car je suis sur la dernière ligne droite.

**23.05.2023**

**API**

Le front-end a été amélioré. Le projet a été connecté à la DB de test de l'EPFL, les utilisateurs sont affichés correctement. On peut donc effectuer une recherche sur les personnes présentes dans la base de données de test. Le projet devrait à priori fonctionner sur la base de données de prod également à priori. Les performances de l'application ont également été optimisées en limitant le nombre de requêtes DDP, autrement dit en envoyant un gros tableau d'un coup au frontend (plus de détails viendront dans le rapport).

**Catégorie :** API

**Durée :** 8h00

**Bilan journée**

La journée a été intense, mais l'étendue du travail effectué est satisfaisante

**24.05.2023**

## **Documentation**

Avancement de la partie "conception" et création de schémas  
d'illustration

**Catégorie** : Documentation

**Durée** : 7h30

### **Bilan journée**

Avancement normal de la documentation

**25.05.2023**

## **Documentation**

Création définition glossaire + avancement dans la partie "réalisation"

**Catégorie** : Documentation

**Durée** : 7h30

### **Bilan journée**

Avancement un peu lent de la documentation aujourd'hui, il va falloir mettre un coup de boost la prochaine et dernière journée.

**30.05.2023**

## **Documentation**

Documentation terminée et paufinée

**Catégorie** : Documentation

**Durée** : 7h30

### **Bilan journée**

Dernière ligne droite, pas le temps de faire des pauses !

ATARI

## **Procès verbaux**

## **Procès verbal 02.04.2023**

### **Heure et Date**

02.05.2023, 17h45 - 18h

### **Participants**

Jérôme Cosandey

Nicolas Borboën

### **Lieu**

EPFL, salle INN013

### **Résumé**

L'état du projet a été évalué. Il a été conclu que le projet avance à vitesse suffisante. Pour le moment aucun retard ou élément portant préjudice au projet n'est à signaler.

### **Actions à entreprendre**

Aucune action spéciale n'est à entreprendre, il faut juste finir le travail de la journée et envoyer les différents rendus, à savoir le planning initial ainsi que le rapport de TPI et le journal de travail en l'état aux experts et au chef de projet.



## Procès verbal 04.04.2023

### Heure et Date

04.04.2023, 14h-15h

### Participants

Jérôme Cosandey

Nicolas Borboën

### Lieu

Appel vidéo Zoom

### Résumé

Durant cette réunion, l'état d'avancement du projet a été évalué avec le Chef de Projet.

Il a suggéré de se concentrer davantage sur la documentation durant la deuxième partie de la journée

La fonctionnalité de recherche a été présentée. L'affichage des suggestions de personnes ne lui convenait pas. Il a alors proposé une autre manière de présenter l'information.

L'accord du chef de projet a été demandé pour charger tous les utilisateurs dans le code javascript du navigateur, plutôt que faire une requête à l'API à chaque saisie de caractère. La proposition a été acceptée.

La disposition de certains éléments de la page a également été discutée et validée. Le chef de projet en a profité pour montrer un élément de la collection de la charte graphique de l'EPFL

qui va sans doute être utile pour la suite du projet :

<https://epfl-si.github.io/elements/#/pages/layout-demo>

### Actions à entreprendre

Corriger la présentation du champ de recherche.

Mettre en place la nouvelle méthode de recherche des utilisateurs.

Faire de la documentation la seconde partie de la journée

## Procès verbal 10.04.2023

### Date

10.04.2023 de 17h à 17h30

### Participants

Jérôme Cosandey

Nicolas Borboën

### Lieu

EPFL, salle INN013

### Résumé

Durant cette séance, l'aspect visuel de la page de détails a été examinée par le chef de projet.

Il a suggéré de s'inspirer d'autres sites déjà existants de l'epfl, tels que peopleet d'utiliser des exemples de code disponibles sur la charte graphique de l'EPFL pour la réalisation du code de la page.

### Actions à entreprendre

Recommencer la page de zéro en utilisant exclusivement des éléments de la charte graphique de l'EPFL et s'inspirer de la disposition des éléments sur d'autres sites de l'EPFL.





## **Test responsive design**



ATARI

## **Cahier des charges**

## 1 INFORMATIONS GÉNÉRALES

<b>Candidat</b>	Nom : <b>COSANDEY</b>	Prénom : Jérôme
	 <a href="mailto:jerome.cosandey@epfl.ch">jerome.cosandey@epfl.ch</a>	tél : +41 78 720 38 00
<b>Lieu de travail</b>	EPFL INN 014 (Bâtiment INN) CH-1015 Lausanne	
<b>Orientation</b>	88601 Développement d'applications (ordonnance 2014)	
<b>Chefs de projet</b>	Nom : <b>BORBOËN</b>	Prénom : Nicolas
	 <a href="mailto:nicolas.borboen@epfl.ch">nicolas.borboen@epfl.ch</a>	tél : +41 21 693 54 55
<b>Expert 1</b>	Nom : <b>CARREL</b>	Prénom : Xavier
	 <a href="mailto:xavier.carrel@eduvaud.ch">xavier.carrel@eduvaud.ch</a>	tél : +41 79 212 96 21
<b>Expert 2</b>	Nom : <b>GIRARDET</b>	Prénom : Alain
	 <a href="mailto:alain.girardet@eduvaud.ch">alain.girardet@eduvaud.ch</a>	tél : +41 21 316 56 46
<b>Période de réalisation</b>	Du 2 au 30 mai : sans les lundis et vendredis (le candidat est aux cours) et le jeudi de l'Ascension, soit les 2, 3, 4, 9, 10, 11, 16, 17, 23, 24, 25 et 30 mai	
<b>Horaires de travail</b>	<b>8:30 – 12:30, 13:30 – 17:30</b> Le candidat a le droit à 20 minutes de pause pour chaque matinée et 15 minutes pour chaque après-midi.	
<b>Nombre d'heures</b>	<b>89 heures</b>	
<b>Planning (en H ou %)</b>	Analyse : 20%	
	Implémentation : 40%	
	Tests : 10%	
	Documentation : 30%	

## 2 PROCÉDURE

- Le candidat réalise un travail personnel sur la base d'un cahier des charges reçu le 1er jour.
- Le cahier des charges est approuvé par les deux experts. Il est en outre présenté, commenté et discuté avec le candidat. Par sa signature, le candidat accepte le travail proposé.
- Le candidat a connaissance de la feuille d'évaluation avant de débiter le travail.
- Le candidat est entièrement responsable de la sécurité de ses données.
- En cas de problèmes graves, le candidat avertit au plus vite les deux experts et son CdP.
- Le candidat a la possibilité d'obtenir de l'aide, mais doit le mentionner dans son dossier.
- À la fin du délai imparti pour la réalisation du TPI, le candidat doit transmettre par courrier électronique le dossier de projet aux deux experts et au chef de projet. En parallèle, une copie papier du rapport doit être fournie sans délai en trois exemplaires (L'un des deux experts peut demander à ne recevoir que la version électronique du dossier). Cette dernière doit être en tout point identique à la version électronique.

---

### 3 TITRE

**ATARI** (*Annuaire Technique d'Attributs pour Responsables Informatiques*) — Modernisation de l'annuaire technique destiné au help-desk et aux administrateurs IT de l'EPFL.

---

### 4 MATÉRIEL ET LOGICIEL À DISPOSITION

Le matériel de travail habituel du candidat doit être utilisé :

- l'ordinateur portable fourni par l'EPFL,
- les logiciels de développement usuels,
- les accès habituels (keybase, GitHub, SSH, etc.),
- le code source de l'application actuelle.

Les accès à la base de données (CADI\_HELPDESK) seront communiqués en début de TPI.

---

### 5 PRÉREQUIS

- Très bonnes notions de TypeScript
- Compétences dans le développement frontend (React.js) et backend (Node.js)
- Connaissances du concept d'API REST (OAS<sup>1</sup>)
- Connaissances du système de droits, de rôles et de groupes (accred.epfl.ch et groupes.epfl.ch) de l'EPFL

---

### 6 DESCRIPTIF DU PROJET

---

#### 6.1 Contexte

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.

*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.*

---

<sup>1</sup> OAS, Open API Specification : <https://spec.openapis.org/oas/latest.html>

## 6.2 Détails de l'application actuelle

Identification des personnes

**Sciper/User/Email:** 123456 / username / prenom.nom@epfl.ch

**Nom/Prenom:** nom autocomplété

**Téléphone:** [+41 21 69] 11122

Chercher Effacer

Figure 1 — écran d'accueil de l'application actuelle

Kermit La Grenouille

**Tools:** Accreds (copy sciper 1st) Check AD (copy username 1st) Check LDAP ServiceNow

**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](#) - [VPO-SI](#) - [ISAS](#) - [ISAS-GE](#)  
**Fonction:** Consultant  
**Statut:** Hors EPFL
- 2. Middleware Services  
 ▲ [EPFL](#) - [VPO-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)



## 6.3 Objectif

L'application qui doit être réalisée par le candidat sera architecturée en deux parties ; la première est le frontend, permettant aux utilisateurs d'accéder à l'information de manière rapide et de façon conviviale. La seconde est l'API, qui fournira au frontend les différentes informations nécessaires. L'API doit se fournir de données véhiculées par différentes entités : autres APIs, LDAP, base de données, etc.

Il est attendu que l'accès aux données (via le frontend ou par l'API) soit authentifié.

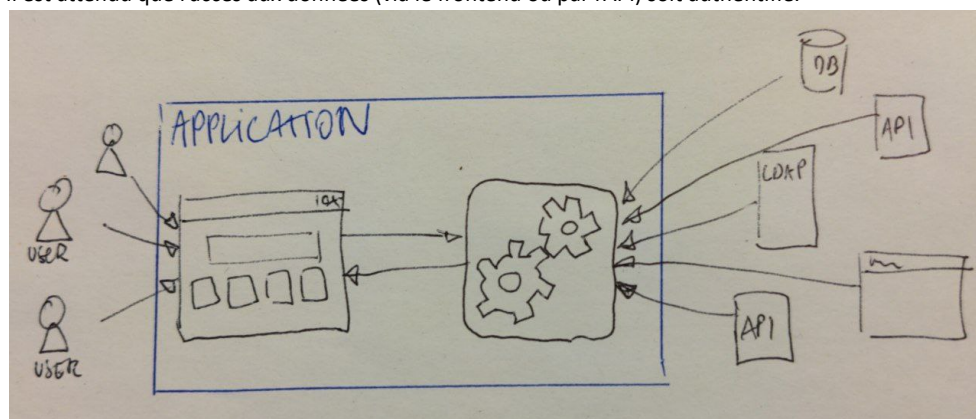


Figure 3 —schéma de fonctionnement basique

### 6.3.1 Frontend

L'interface de l'application doit respecter la charte graphique de l'EPFL, dont les différents éléments sont disponibles sur <https://epfl-si.github.io/elements>.

Le frontend doit être complètement séparé de l'API.

La page d'accueil de l'application est composée d'une section pour la recherche de personne et une autre pour l'affichage des résultats.

#### Recherche

La refonte de la fonctionnalité de recherche de personnes doit en faciliter l'utilisation. A l'opposé de ce qui est proposé actuellement, elle doit proposer un champ unique de recherche "intelligente".

Le champ de recherche doit proposer une fonctionnalité de recherche partielle pendant la saisie ("search-as-you-type"), qui présente des résultats de recherche qui correspondent (comme préfixe ou comme sous-chaîne) avec les chaînes partiellement saisies, selon l'heuristique suivante :

- si le champ de saisie contient uniquement des chiffres, alors la recherche se fait sur les numéros SCIPER<sup>2</sup> ou les numéros de téléphone connus du back-end ;
- si le champ de saisie contient uniquement des lettres, alors la recherche se fait par nom de login, nom, prénom ou adresse e-mail ;
- si un point ou qu'une arobase est saisie, alors la recherche se fait par adresse e-mail.

<sup>2</sup> Le numéro sciper est l'identifiant unique d'une personne à l'EPFL. Il est composé de 6 chiffres.

La recherche partielle s'adapte au rythme de saisie de l'utilisateur, afin de limiter la charge du back-end et d'éviter de présenter un trop grand nombre de résultats (nombre limité à 10 entrées).

### Résultats

Lorsque la recherche est restreinte à une seule personne (soit parce qu'il n'y a plus qu'une correspondance dans la recherche partielle; soit que l'utilisateur a choisi explicitement un résultat de recherche partielle), alors s'affiche une vue d'ensemble de la personne. Les mêmes informations qui sont disponibles actuellement doivent alors être présentées à l'utilisateur.

### Outils

Afin de faciliter le travail du service desk, des liens permettant de se rendre sur différents outils avec les détails de la personnes présélectionnés doivent être disponibles :

- Un lien vers <https://accres.epfl.ch/> ;
- Un lien vers <https://windows.epfl.ch/checkad/default.aspx> avec le champ pré-rempli avec le username ou le SCIPER ;
- Un lien vers une page de l'application présentant tous les champs présents dans le LDAP pour cette personne ;
- Un lien vers la page ServiceNow de cette personne (du type [https://it.epfl.ch/backoffice/sys\\_user.do?sysparm\\_query=user\\_name=133134](https://it.epfl.ch/backoffice/sys_user.do?sysparm_query=user_name=133134)) ;
- Un lien vers la page permettant de vérifier le status de l'email (du type <https://mailwww.epfl.ch/emailStatus.cgi?query=prenom.nom@epfl.ch>) ;
- Un lien vers la page de l'annuaire de cette personne (du type <https://people.epfl.ch/133134>).

### Disposition

La disposition des différents éléments sur la page est libre mais doit être discutée et validée avec le chef de projet.

---

## 6.3.2 API

L'API du projet permet de centraliser les informations provenant de différentes sources. De plus, elle permet de fournir les données de manière uniformisée au frontend. Elle permet aussi de créer, modifier ou supprimer différentes sources de données. Finalement, elle rend les évolutions du projet plus simples (par exemple ajouter un deuxième client à l'API).

Le candidat prendra les mesures nécessaires pour :

- que l'API soit complètement séparée du frontend ;
- que l'API ne soit pas librement accessible ;
- et donc que l'accès à l'API soit protégé (par exemple avec des Bearer Token).

Dans le cadre de ce TPI, l'API devra être connectée à la base de données. La connexion au LDAP ne nécessite pas d'authentification et peut être faite directement du serveur. La connexion à l'Active Directory ou à d'autres services (comme par exemple websrv.epfl.ch, une API permettant d'avoir des informations sur les rôles d'une personne) n'est pas nécessaire.

---

### 6.3.3 Contrôle d'accès

Le contrôle d'accès à l'application (frontend) doit pouvoir être fait par rôles, droits ou groupes. Ces différentes informations sont fournies par l'IdP<sup>3</sup> de l'EPFL, Tequila.

*NOTE : Il est probable que les informations présentées aux utilisateurs puissent varier en fonction de ces droits dans le futur. Par exemple, un utilisateur d'une faculté A ne pourra pas voir les informations des utilisateurs d'une faculté B. Le candidat n'a pas besoin d'implémenter cette fonctionnalité mais doit garder ceci en tête afin de préférer des choix qui permettront ceci plus tard.*

L'accès à l'API est pour le moment strictement limité à l'application (frontend), mais doit être authentifié.

*NOTE : l'évolution du système doit pouvoir offrir des accès à l'API à d'autres clients, toujours de manière authentifiée. Il n'est pas demandé de prévoir un système de gestion de token ou de « rate limit » lors de ce TPI mais il faut néanmoins garder ces fonctionnalités en tête afin de permettre des améliorations futures.*

---

## 7 LIVRABLES

Le candidat est responsable de livrer à son chef de projet et aux deux experts :

- Une planification initiale, livrée après la première journée du TPI
- Un rapport de projet, qui contiendra, entre-autres,
  - La planification initiale
  - La planification finale
  - Les différents points demandés dans ce document
- Un journal de travail
- Un dépôt Git public sur le GitHub <https://github.com/epfl-si/> avec les sources de l'application accompagné d'un fichier README.md
  - Ce dépôt doit être neutre en ce qui concerne ce TPI ; tout document concernant le TPI (rapport, journal de travail, etc.) ne doit pas s'y trouver.

---

<sup>3</sup> [https://en.wikipedia.org/wiki/Identity\\_provider](https://en.wikipedia.org/wiki/Identity_provider)

---

## 8 POINTS TECHNIQUES ÉVALUÉS SPÉCIFIQUES AU PROJET

La grille d'évaluation définit les critères généraux selon lesquels le travail du candidat sera évalué (documentation, journal de travail, respect des normes, qualité, ...).

En plus de cela, le travail sera évalué sur les 7 points spécifiques suivants (points A14 à A20) :

1. La qualité de l'utilisation de Git :
  - a. Le fichier README.md décrit brièvement le projet et explique aux futures développeurs comment faire fonctionner le système sur machine similaire à celle utilisée par le candidat ;
  - b. Les messages de commits concis et explicites, qui permettent de retracer l'historique du projet ;
  - c. L'utilisation de branche de features est visible dans le dépôt.
2. Le rapport contient un schéma de l'architecture de l'application commenté et listant les sources et provenances des différentes données utilisées.
3. Le candidat utilise Docker pour son environnement de développement et décrit les avantages de son utilisation dans son rapport.
4. La qualité du code, mesuré en termes de :
  - a. «Once and only once» : code exempt de sections copiées/modifiées (Don't Repeat Yourself<sup>4</sup>) ;
  - b. d'utilisation du paradigme orienté objet ;
  - c. typage statique (TypeScript) précis (pas de `any`);
  - d. présence de documentation interne au code.
5. Le candidat met en exergue dans son rapport trois mesures de sécurité qu'il a implémentées pour ce projet. Chaque mesure est décrite en détail et en quoi elle rend l'application plus sécurée.
6. La charte graphique («style guide») utilisée pour le frontend est celle de l'EPFL et le candidat démontre dans son rapport que l'interface utilisateur se décline proprement sur des écrans de différentes tailles (*responsive web design*).
7. Le candidat énumère différentes possibilités d'améliorations et d'évolutions du système qu'il a mis en place dans une section dédiée de son rapport.

---

<sup>4</sup> [https://en.wikipedia.org/wiki/Don't\\_repeat\\_yourself](https://en.wikipedia.org/wiki/Don't_repeat_yourself)

---

**9 VALIDATION**

	Lu et approuvé le :	Signature :
Candidat :		
Expert n°1 :		
Expert n° 2 :		
Chef de projet :		

ATARI