

Rapport de projet

Bachelier en informatique, orientation développement  
d'application

Bloc 3

# Tinder

**Falconi Franco**

**Smaali Zakaria**

**Rinaldi Alessio**

**Année académique 2023 – 2024**

# Tables des matières

1. Étude préliminaire .....	2
1.1 Contexte .....	2
1.2 Analyse des Besoins .....	3
2. Architecture technique .....	4
2.1 Base de données .....	4
2.2 Architecture logicielle .....	6
2.2.1. Description détaillée des composants logiciels .....	6
2.2.2. Interactions détaillées entre les composants .....	7
2.2.3. Découpage en couches .....	7
2.2.4. Choix technologiques.....	7
2. Choix des frameworks.....	9
3.1. ASP.NET .....	9
3.2. Angular .....	10
4. Mini-guide .....	11
5. Glossaire .....	18
6. Bibliographie .....	19

# 1. Étude préliminaire

## 1.1 Contexte

Le but principal de ce projet était de concevoir et de développer une application de rencontre innovante. Pour ce faire, nous avons choisi d'utiliser les technologies Angular version 16 pour le frontend et ASP.NET pour le backend. Ces technologies ont été choisies pour leur robustesse, leur flexibilité et leur capacité à répondre aux exigences de performance et de sécurité de notre application.

L'étude préliminaire a débuté par une analyse approfondie des besoins des utilisateurs. Nous avons étudié les fonctionnalités clés des applications de rencontre existantes. Nous avons également pris en compte les exigences non fonctionnelles telles que la sécurité et la performance, qui sont essentielles pour garantir une expérience utilisateur optimale.

Dans notre parcours pour créer une application qui répond aux attentes des utilisateurs tout en apportant quelque chose de nouveau et d'excitant, nous nous sommes beaucoup inspirés des géants du secteur. Nous avons étudié de près l'application Tinder, qui est largement reconnue comme la référence en matière d'applications de rencontre. Notre objectif était de fournir une expérience qui se rapproche au plus près des standards actuels, tout en apportant notre propre touche unique. Nous voulions créer une application qui serait immédiatement familière aux utilisateurs, tout en leur offrant de nouvelles fonctionnalités et une expérience utilisateur améliorée.

## 1.2 Analyse des Besoins

L'analyse des besoins a été une étape cruciale dans le développement de notre application. Elle nous a permis de comprendre ce que les utilisateurs attendent d'une application de rencontre et comment nous pouvons répondre à ces attentes.

Nous avons identifié la nécessité d'une section dédiée à la connexion et à l'inscription. Cette fonctionnalité permet aux utilisateurs d'accéder à l'interface de l'application et de parcourir les profils des autres utilisateurs. Nous avons conçu cette section pour être simple et intuitive, afin que les utilisateurs puissent commencer à utiliser l'application dès qu'ils l'ont téléchargée.

Une fois connectés, les utilisateurs peuvent parcourir les profils et exprimer leur intérêt pour certains profils grâce à la fonctionnalité de "Match". Cette fonctionnalité permet aux utilisateurs de signaler leur intérêt pour un profil en le "likant". Si l'intérêt est réciproque, c'est-à-dire si les deux utilisateurs se sont mutuellement "likés", alors un "match" est créé.

En cas de match, les utilisateurs ont la possibilité de répondre à des questions spécifiques, préalablement renseignées dans leur profil. Ces questions permettent aux utilisateurs de mieux se connaître et de déterminer s'ils sont compatibles. Si les deux profils obtiennent chacun une note au moins égale à la moyenne, ils peuvent alors débiter une discussion entre eux. Sinon, cette fonctionnalité ne sera pas disponible. Cette approche garantit que seuls les utilisateurs qui sont potentiellement compatibles peuvent entrer en contact, ce qui améliore l'expérience utilisateur et augmente les chances de succès des rencontres.

## 2. Architecture technique

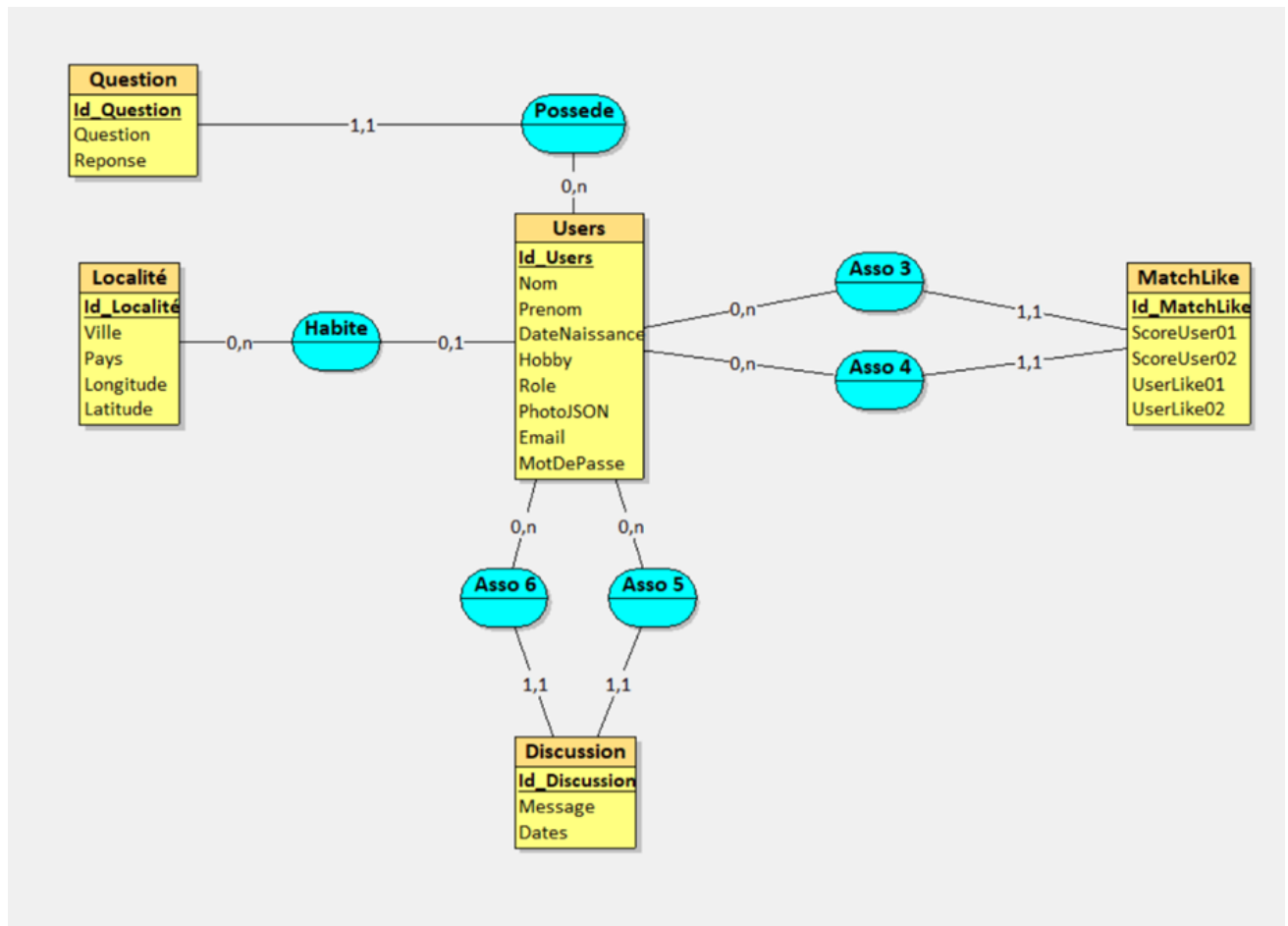
### 2.1 Base de données

Pour notre projet, nous avons opté pour une architecture de base de données relationnelle, qui est le standard proposé par Microsoft SQL Server. Pour rappel, nous utilisons ASP.NET pour le backend, ce qui signifie que nous travaillons également avec l'ensemble des outils Microsoft.

Pour structurer notre base de données, nous avons choisi d'utiliser cinq tables principales :

- La table Users : Cette table symbolise les utilisateurs de notre application. Elle contient les champs de base d'un utilisateur, tels que le nom, l'adresse e-mail, le mot de passe, etc. De plus, elle contient également une liste de photos, ce qui permet à l'utilisateur de poster plusieurs photos sur son profil.
- La table Question : Cette table permet à chaque utilisateur de renseigner ses questions ainsi que les réponses à ces questions. Cela permet aux autres utilisateurs de pouvoir répondre à ces questions en cas de match. C'est une fonctionnalité clé de notre application qui favorise l'interaction entre les utilisateurs.
- La table Localite : Cette table permet aux utilisateurs de renseigner leur localisation. Cela permet de matcher et de rencontrer d'autres profils près de leur résidence. Cela permet également de mettre en place des filtres de recherche basés sur la localisation, ce qui améliore l'expérience utilisateur.
- La table Discussion : Cette table permet de stocker les messages échangés entre deux utilisateurs. Chaque fois qu'un match est créé et qu'une discussion est initiée, les messages sont stockés dans cette table. Cela permet une récupération rapide et efficace des discussions précédentes.
- La table MatchLike : Cette table permet de sauvegarder les matchs si deux personnes décident de se matcher. Elle stocke également le score des réponses aux questions des deux profils. Si le score est supérieur à la moyenne, la fonctionnalité de discussion est débloquée. Sinon, cette fonctionnalité reste inaccessible. Cette approche garantit que seuls les utilisateurs qui sont potentiellement compatibles peuvent entrer en contact, ce qui améliore l'expérience utilisateur et augmente les chances de succès des rencontres.

En somme, notre base de données a été conçue pour soutenir efficacement les fonctionnalités clés de notre application de rencontre, tout en garantissant une performance optimale et une sécurité robuste. Nous avons veillé à ce que chaque table joue un rôle spécifique et contribue à l'objectif global de fournir une expérience utilisateur exceptionnelle.



## 2.2 Architecture logicielle

### 2.2.1. Description détaillée des composants logiciels

Votre système est composé de cinq modules principaux, chacun ayant un rôle spécifique et essentiel dans le fonctionnement global de l'application :

**Contrôleurs** : Ce module joue un rôle crucial en tant que point d'entrée de votre application. Il est chargé de réceptionner les requêtes des utilisateurs, de les traiter et de renvoyer les réponses appropriées. Il agit comme un intermédiaire entre l'utilisateur et le système, assurant une communication fluide et efficace.

**Data** : Ce module est au cœur de la gestion des données de votre application. Il symbolise les liens entre les tables de votre modèle de données et est responsable de leur organisation dans la base de données. Il assure la cohérence, l'intégrité et la sécurité des données tout au long de leur cycle de vie dans l'application.

**Migration** : Ce module est essentiel pour maintenir la structure de votre base de données à jour. Il symbolise les migrations de votre base de données, permettant de gérer les modifications de la structure de la base de données au fil du temps et d'assurer que la base de données est toujours en phase avec les besoins de l'application.

**Modèles** : Ce module contient vos classes principales, qui symbolisent également vos tables de base de données. Ces classes définissent la structure des données qui seront stockées dans la base de données, assurant ainsi que chaque élément de données est correctement représenté et géré.

**Propriétés** : Ce module contient des fichiers de configuration qui définissent les paramètres de votre application. Ces paramètres peuvent inclure des informations telles que les détails de la connexion à la base de données, les paramètres de sécurité, etc.

### 2.2.2. Interactions détaillées entre les composants

Les interactions entre ces composants sont essentielles pour le fonctionnement de votre système. Par exemple, lorsque le module Controllers reçoit une requête, il peut interagir avec le module Models pour récupérer ou modifier les données nécessaires. Ensuite, le module Data peut être utilisé pour effectuer les opérations de base de données correspondantes. Si une modification de la structure de la base de données est nécessaire, le module Migration peut être utilisé. Ces interactions garantissent que chaque requête est traitée de manière efficace et précise, assurant une expérience utilisateur optimale.

### 2.2.3. Découpage en couches

Votre système est structuré en plusieurs couches, chacune ayant une responsabilité spécifique. Par exemple, vous pourriez avoir une couche de présentation (représentée par le module Controllers), une couche métier (représentée par les modules Models et Data), et une couche de données (représentée par le module Migration). Cette structure en couches permet une séparation claire des préoccupations, améliorant la maintenabilité et l'évolutivité de l'application.

### 2.2.4. Choix technologiques

Vous avez fait plusieurs choix technologiques clés pour votre application. Vous utilisez ASP.NET pour le backend, qui est un framework de développement web gratuit et open-source de Microsoft. Il est connu pour sa flexibilité, sa robustesse et sa facilité d'utilisation, ce qui en fait un excellent choix pour le développement d'applications web modernes.

Pour la gestion de la base de données, vous utilisez Microsoft SQL Server, un système de gestion de base de données relationnelle robuste et fiable. Il offre une grande flexibilité et de nombreuses fonctionnalités pour la gestion des données, ce qui en fait un excellent choix pour votre application.

Pour gérer les sessions, vous utilisez JWT (JSON Web Token). JWT est une norme ouverte qui définit un moyen compact et autonome de transmettre des informations entre les parties sous forme d'un objet JSON. Cette information peut être vérifiée et approuvée car elle est signée numériquement. JWT est utilisé pour créer des tokens signés qui peuvent garder vos informations et gérer les rôles. C'est un choix technologique judicieux pour garantir la sécurité et l'intégrité des données de votre application.



Enfin, pour gérer les CORS (Cross-Origin Resource Sharing) policy, vous utilisez ASP.CORS. CORS est une technologie qui permet aux ressources d'un site web d'être demandées à partir d'un domaine différent de celui du site d'origine. ASP.CORS est une implémentation de cette technologie pour le framework ASP.NET, permettant une gestion efficace et sécurisée des requêtes cross-origin dans votre application.

## 2. Choix des frameworks

Pour le développement de notre application, nous avons choisi d'utiliser deux frameworks principaux : ASP.NET pour le backend et Angular pour le frontend.

### 3.1. ASP.NET

C'était la première fois que nous utilisions ASP.NET, et nous avons été agréablement surpris par sa rapidité et sa facilité d'utilisation pour générer un CRUD (Create, Read, Update, Delete). ASP.NET embarque de nombreux outils de Microsoft, ce qui rend la création et la mise en route d'un projet ASP.NET extrêmement rapide. De plus, de nombreux outils se greffent de base et ne nécessitent pas de configuration supplémentaire, comme Microsoft SQL Server ou même Copilot. Étant très familiers avec Java et C#, nous avons rapidement compris la logique de ASP.NET et avons pu commencer à travailler avec le framework sans difficulté. Cependant, nous avons remarqué qu'il ne faut pas considérer la génération du CRUD comme un produit fini, mais plutôt comme une base à partir de laquelle nous devons modifier et adapter notre projet.

## 3.2. Angular

Nous avons choisi d'utiliser Angular pour notre projet pour plusieurs raisons clés. Premièrement, nous voulions approfondir nos connaissances et notre expertise avec ce framework. Angular a été un outil précieux pour nous lors de notre projet de deuxième année, et nous avons constaté que sa flexibilité et sa robustesse nous ont permis de développer des solutions efficaces et innovantes.

De plus, dans le cadre de notre cours de projet 3 cette année, l'utilisation d'Angular a été fortement encouragée. Nous avons saisi cette opportunité pour renforcer notre maîtrise de ce framework, en nous familiarisant davantage avec ses fonctionnalités avancées et en apprenant à surmonter les défis qu'il peut présenter.

En somme, le choix d'Angular pour notre projet a été motivé par notre désir d'acquérir une expérience pratique plus approfondie avec ce framework, tout en tirant parti de son potentiel pour faciliter le développement de solutions logicielles de haute qualité. Nous sommes convaincus que cette expérience nous sera bénéfique dans nos futurs projets et carrières.

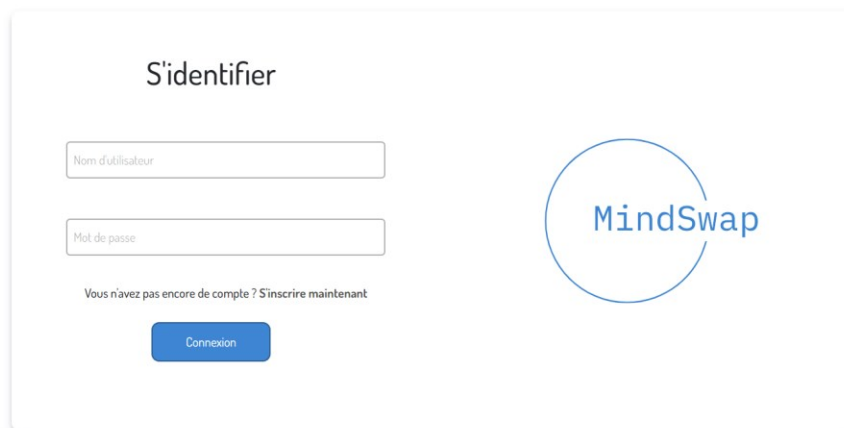
## 4. Mini-guide

Une fois arrivé sur le site, il vous sera demandé de vous connecter. Dans le cas où vous n'avez pas de compte, appuyez sur "S'inscrire maintenant".

### **Connexion**

Connectez-vous à l'aide de votre nom d'utilisateur et de votre mot de passe.

---



The screenshot shows a login form titled "S'identifier". It contains two input fields: "Nom d'utilisateur" and "Mot de passe". Below the password field is a link that says "Vous n'avez pas encore de compte ? S'inscrire maintenant". At the bottom is a blue button labeled "Connexion". To the right of the form is the MindSwap logo, which consists of the word "MindSwap" in blue text next to a blue circle.

### **Inscription**

Renseignez un nom d'utilisateur, votre adresse email ainsi qu'un mot de passe qui doit contenir au minimum 8 caractères.



The screenshot shows a registration form titled "S'inscrire" under the "MindSwap" header. It contains four input fields: "Nom d'utilisateur", "Email", "Mot de passe", and "Confirmer mot de passe". Below the "Mot de passe" field is a small note: "\*8 caractères minimum". At the bottom is a blue button labeled "Inscription". Above the button is a link that says "Vous avez déjà un compte ? Se connecter".

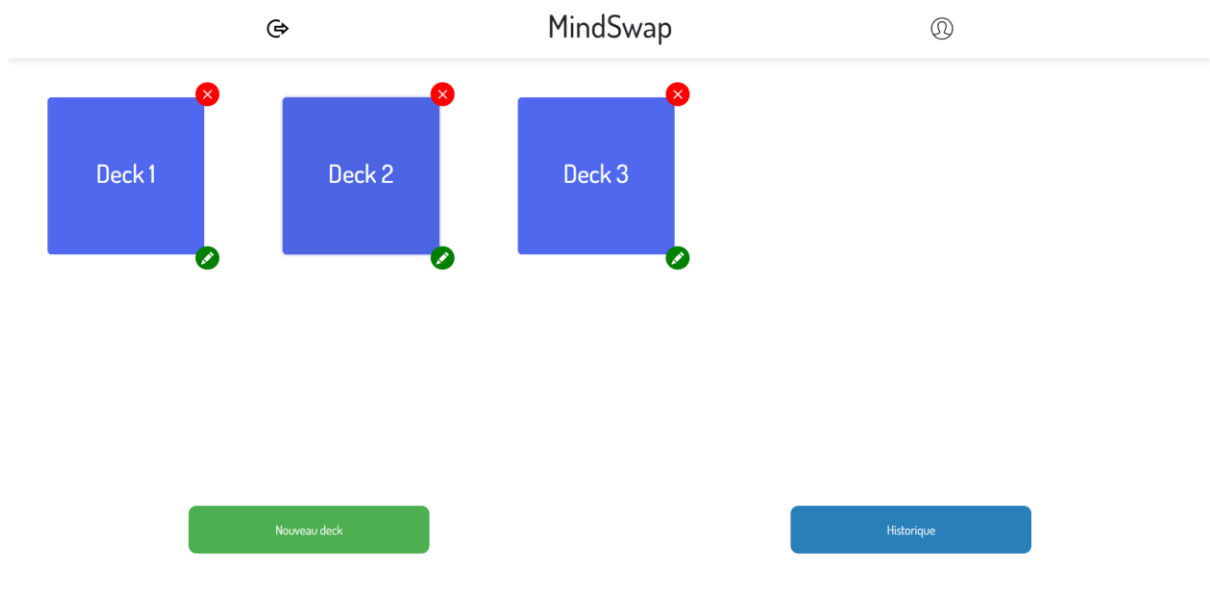
## **Page d'accueil**

Une fois connecté à votre compte, vous êtes redirigé vers la page d'accueil où se trouvent deux boutons :

- Nouveau deck : ce bouton vous permet de créer un deck en lui donnant le nom que vous souhaitez, par exemple : mathématiques.
- Historique : ce bouton vous permet d'afficher l'historique de vos parties avec la date, le nom du deck et votre score afin de pouvoir constater votre progression.

Une fois le deck créé, il apparaît sur la page d'accueil sous forme de carte.

- Pour supprimer un deck, il suffit de cliquer sur la croix en haut à droite de la carte.
- Pour modifier le nom du deck, il vous suffit de cliquer sur l'icône du crayon en bas à droite de la carte.



### **Ajout de carte dans un deck**

Commencez par cliquer sur le deck pour lequel vous souhaitez ajouter des cartes. Cliquez sur “Nouvelle carte”. Il vous sera demandé de fournir une question qui sera affichée sur la face avant de la flash card ainsi qu’une réponse qui se trouvera au dos de celle-ci, ensuite cliquez sur “Envoyer”.

Votre carte apparaît à l’écran, si vous souhaitez ajouter une image pour cette carte, appuyez sur “Modifier” et sélectionnez une photo pour l’avant et l’arrière ou uniquement pour une des deux faces au choix.

Pour supprimer une carte, il vous suffit de cliquer sur “Supprimer”.

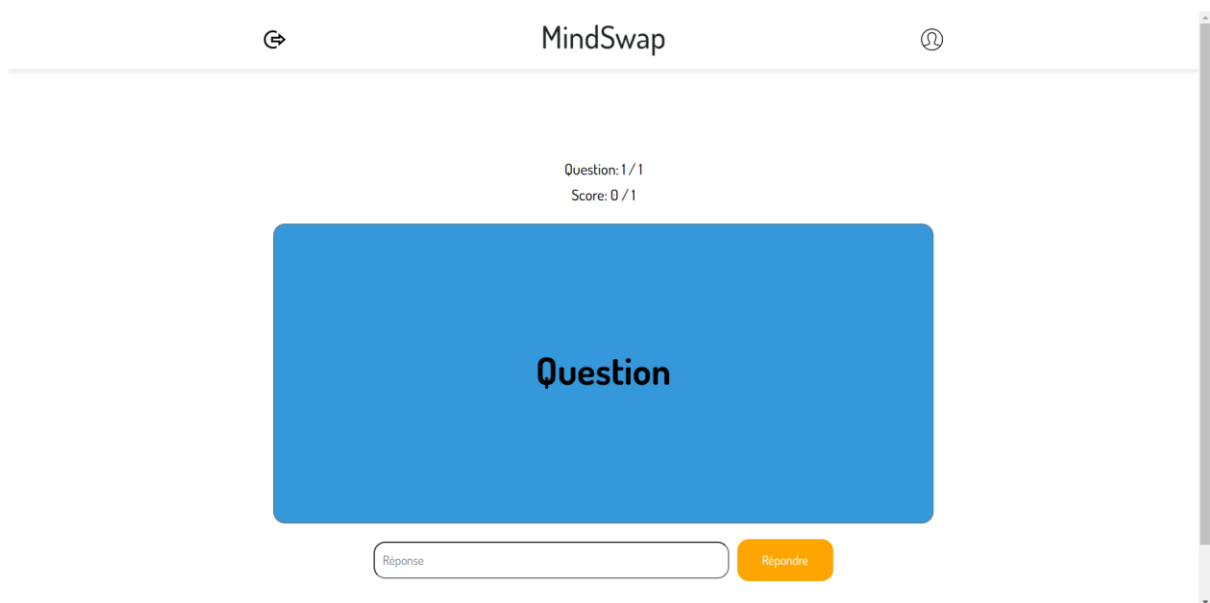


## Jeu

Une fois vos cartes créées, il est possible de les réviser sous forme de test, pour cela appuyez sur le bouton “Jouer”.

La page de jeu vous affiche une carte, si vous souhaitez connaître la réponse en cas d’oubli, il vous suffira de cliquer sur la carte pour la retourner et de cliquer à nouveau pour cacher la réponse.

Une zone de texte située en dessous de la carte vous permettra d’y introduire la réponse, ensuite en cliquant sur “Répondre” vous passerez à la prochaine carte. Sur le dessus est affichée la position dans le questionnaire afin de savoir où vous vous situez dans le jeu. En dessous de la question est affiché le score, celui-ci augmente à chaque bonne réponse, à la fin du jeu les données sont enregistrées et un écran vous affiche le score, il sera possible de consulter son historique, rejouer la partie ou revenir à l’accueil.



The screenshot displays the MindSwap game interface. At the top, there is a navigation bar with a home icon, the text "MindSwap", and a user profile icon. Below the navigation bar, the text "Question: 1 / 1" and "Score: 0 / 1" is displayed. The main content area features a large blue rectangular card with the word "Question" centered on it. Below the card, there is a text input field with the placeholder text "Réponse" and an orange button labeled "Répondre".

### **Barre de navigation**

Sur le dessous de l'écran se situe la barre de navigation qui permet 3 choses :

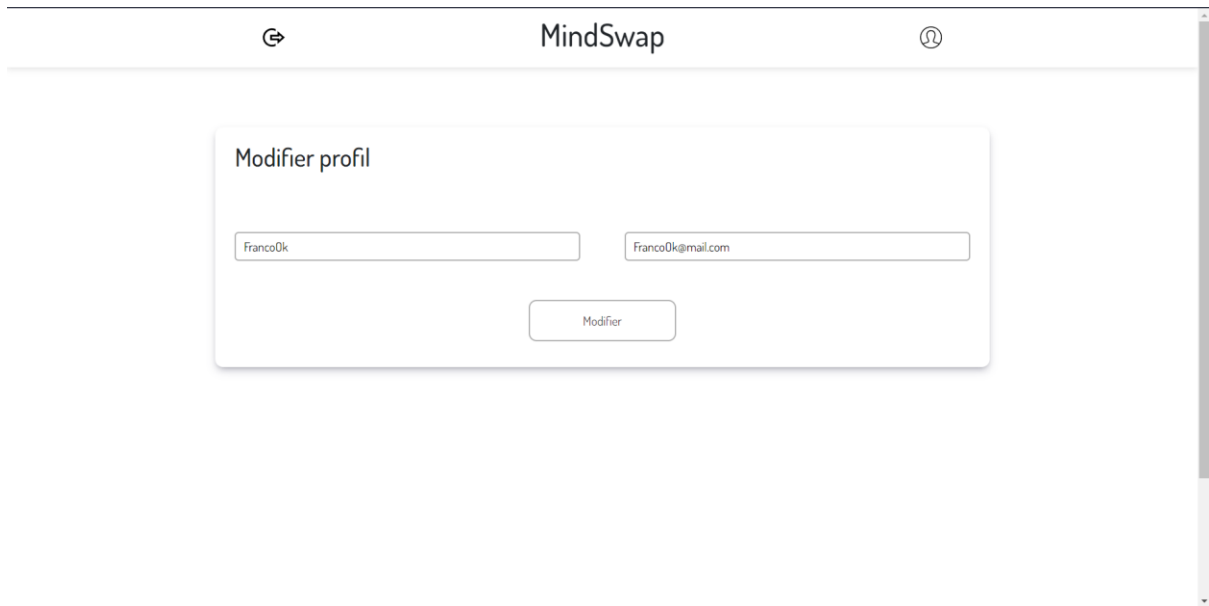
- Se déconnecter à l'aide du bouton de gauche.
- Revenir à la page d'accueil en appuyant sur le nom du site.
- Modifier votre profil avec le bouton "Profil" qui se situe à droite.







## **Modifier son profil**

Vous aurez la possibilité de modifier votre nom d'utilisateur et l'adresse mail.



The screenshot shows a web browser window with the MindSwap logo in the center of the header. On the left of the header is a circular icon with a right-pointing arrow, and on the right is a circular icon with a person silhouette. The main content area features a light gray rounded rectangle titled 'Modifier profil'. Inside this rectangle, there are two text input fields: the first contains 'Franco0k' and the second contains 'Franco0k@mail.com'. Below these fields is a 'Modifier' button.

 MindSwap 

### Modifier profil

Modifier

## **Historique**

Sur cette page, nous pouvons retrouver l'historique de nos parties.



MindSwap



### Historique

Date	Nom du deck	Score
24/12/2023	Deck 1	0 / 1
24/12/2023	Deck 1	0 / 2

## 5. Glossaire

Tinder	Application de rencontre
Match	Quand deux personnes aiment le profil de l'autre
Frameworks	Outils de développement
JWT	JSON Web Token, suite de caractères en base64 contenant différentes informations signées.
CRUD	Créer, Lire, Modifier, Supprimer, ce sont les opérations de base d'une API.
API	Interface de Programmation d'Applications, est un ensemble de règles et de protocoles qui permet à différents logiciels de communiquer entre eux.
ASP.NET	Outils de programmation pour C#, orientés vers le développement serveur.
Angular	Outils de programmation en TypeScript, HTML, CSS, orientés vers le développement frontal.

## 6. Bibliographie

Microsoft ASP.NET : <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-8.0>

Angular : <https://angular.io/docs>

Medium : <https://medium.com/>

StackOverflow : <https://stackoverflow.com/>

Reddit : <https://www.reddit.com/>

Github Topics : <https://github.com/topics/help>