



Projet Fil Rouge

CDA34 24-01

Botta Facundo



Sommaire

- [About me](#)
- [The project](#)
- [CDC](#)
- [Technologies utilisées](#)
- [Maquettage - Moadboard](#)
- [Maquettage - Zoning](#)
- [Maquettage - WireFrame](#)
- [Maquettage - Mockup](#)
- [Conception - Arborescence](#)
- [Conception - UML - UseCase](#)
- [Conception - UML - Activité](#)
- [Conception - UML - Séquence](#)
- [Conception - UML - Classe](#)
- [Conception - MCD](#)
- [Conception - MLD](#)
- [Architecture](#)
- [Fonctionnalité front](#)
- [Fonctionnalité back](#)
- [Tests](#)
- [Bonus](#)
- [Dépôt GitHub](#)

About me...

Facundo Botta

[Portfolio](#)

[GitHub](#)

[LinkedIn](#)

[Dev community](#)





The project: EventHub

⌚ What about?

- All-in-one platform for events & appointment management
- For professionals & regular users
- Simplifies organization & communication

⌚ Why this app?

- Options in the market
- First approach -> Appointment management system
- Finally -> More like a social network



CDC - Analyse du besoin

Fonctions du produit

- ⌚ Gérer commentaires, ponctuation et signalement des utilisateurs
- ⌚ Authentification avec compte Google et intégration avec son calendrier
- ⌚ Administrer les événements, notifier les utilisateurs
- ⌚ Système de messagerie et liste de tâches
- ⌚ Système de recherche (utilisateurs, événements, messages)



CDC - Analyse du besoin

Contraintes techniques

- ⌚ Base de données large et complexe
- ⌚ Sécurité du site et protection des données
- ⌚ Intégration sur mobile

CDC - SWOT

S

Strengths

FORCES

W

Weaknesses

FAIBLESSES

O

Opportunities

OPPORTUNITES

T

Theats

MENACES

- Faible concurrence
- Intuitif
- Gratuit
- Peu connu
- Confiance
- Limitations
- Innovation
- Evolution
- Conflits entre utilisateurs

CDC - Public visé



Nom
"Claude Martin"

- Age : 34
- Travail : Personal Trainer
- Ville : Montpellier

Bio

Il propose ses services de formation personnelle pour les groupes et les particuliers. L'application le permet de gérer efficacement son emploi du temps, d'accepter de nouveaux rendez-vous et de conserver un enregistrement organisé de ses clients et de ses sessions.

Personnalité

	Extrovert	Créatif	Temps libre	Organisé	Esprit d'équipe
Introverti					
Occupé					
Analytique					
Déssordonné					
Indépendant					



Nom
"Jean Pierre"

- Age : 68
- Travail : À la retraite
- Ville : Montpellier

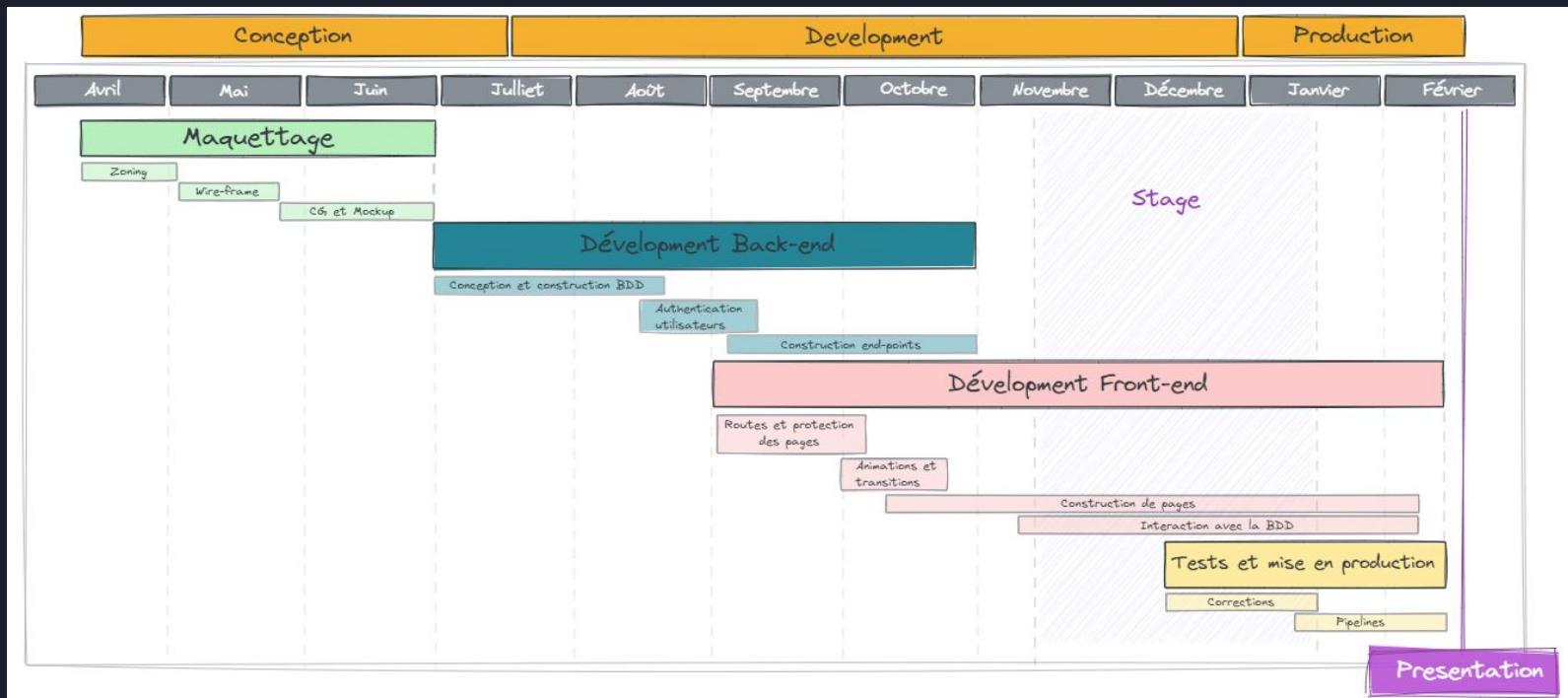
Bio

Bien que lui puisse bénéficier des services professionnels disponibles dans l'applapplication , il recherche spécifiquement un chirurgien certifié. Pour ses besoins, un site comme Doctolib serait plus approprié.

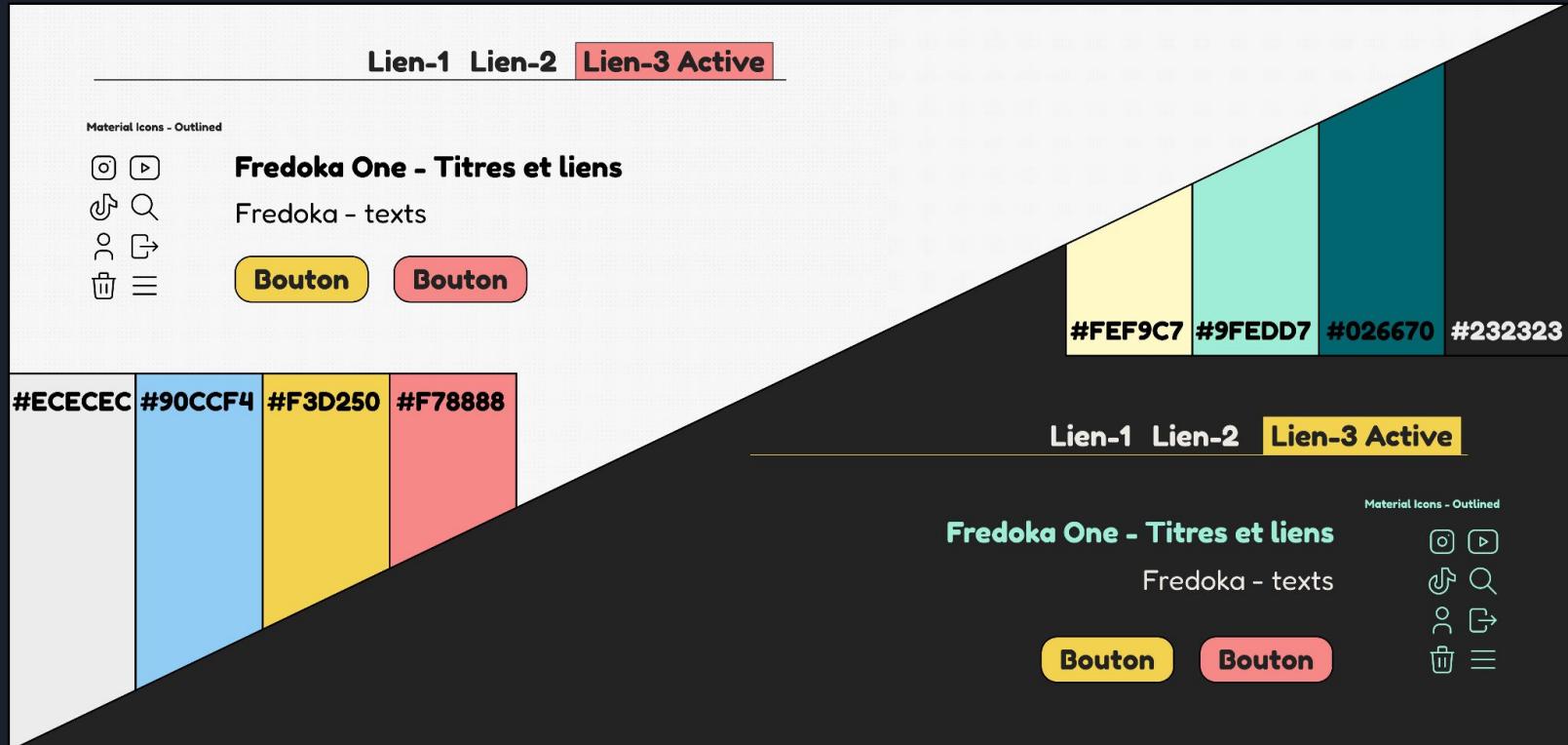
Personnalité

	Extrovert	Créatif	Temps libre	Organisé	Esprit d'équipe
Introverti					
Occupé					
Analytique					
Déssordonné					
Indépendant					

CDC - Gantt



CDC - Style tile



CDC - Sélection des technologies



JavaScript



CSS3



HTML5



MySQL



TypeScript



Tailwind CSS



React



Prisma



Zod

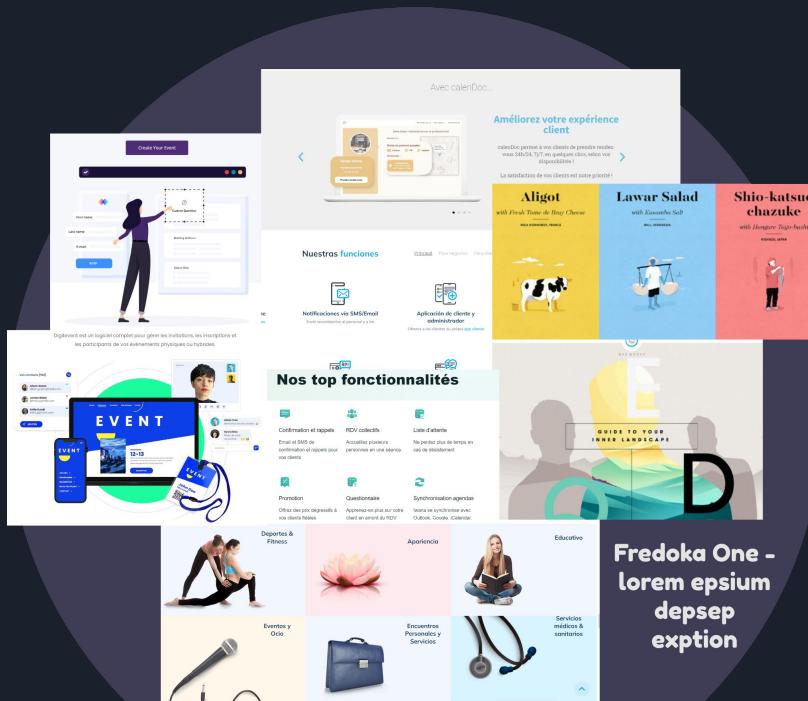


Next.js



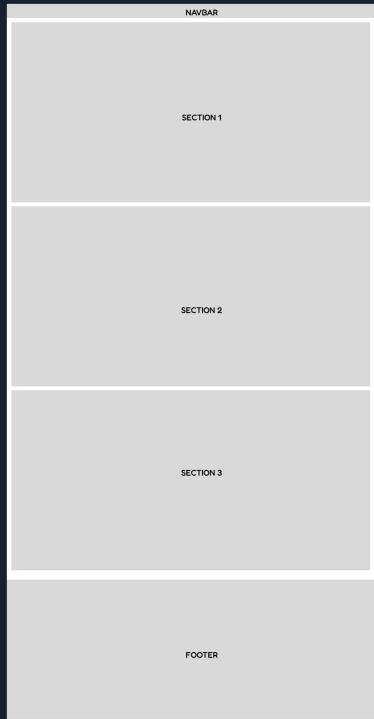
Auth.js

Maquettage - MoodBoard

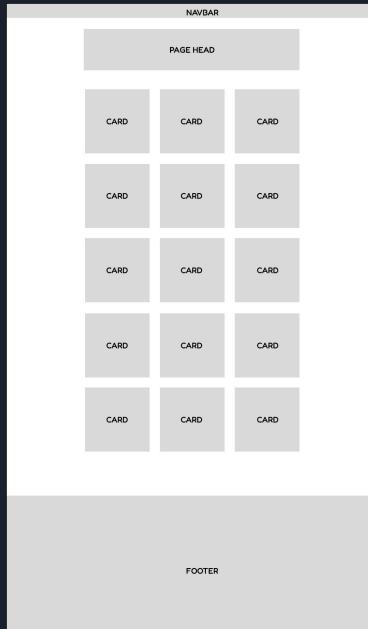


Maquettage - Zoning

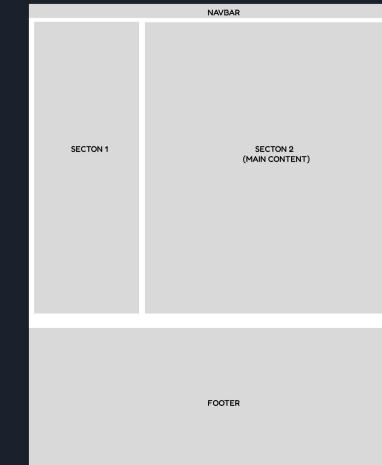
Accueil



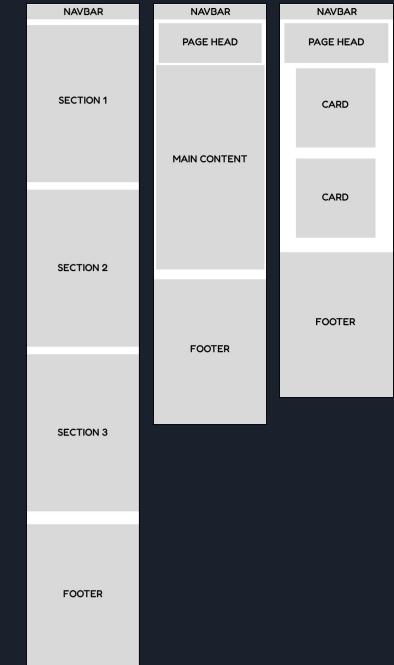
Communauté /Événements



Messagerie/Profil

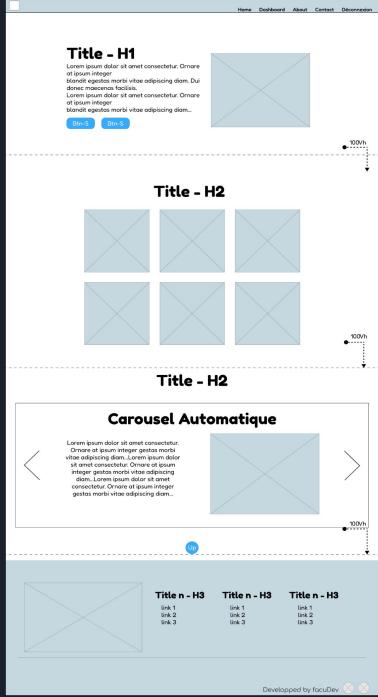


Petits écrans

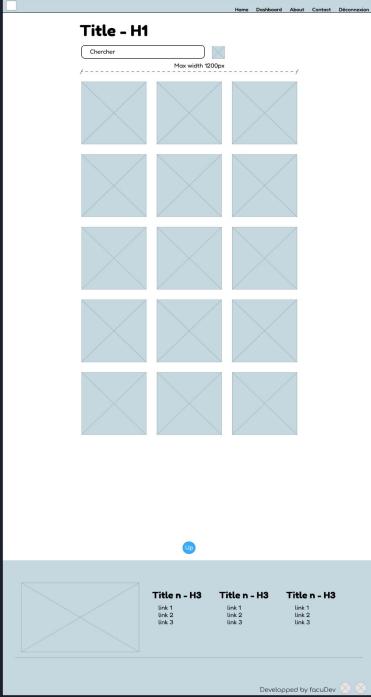


Maquettage - WireFrame

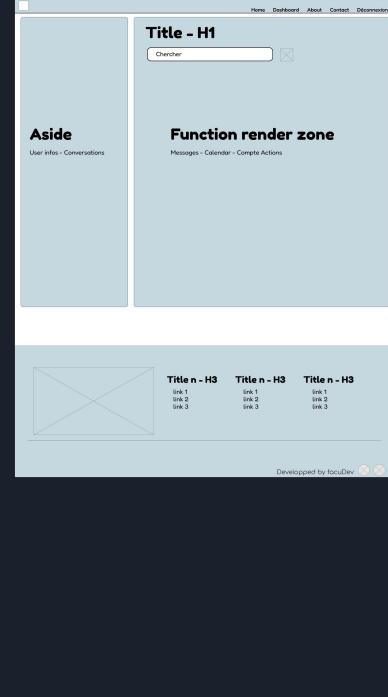
Accueil



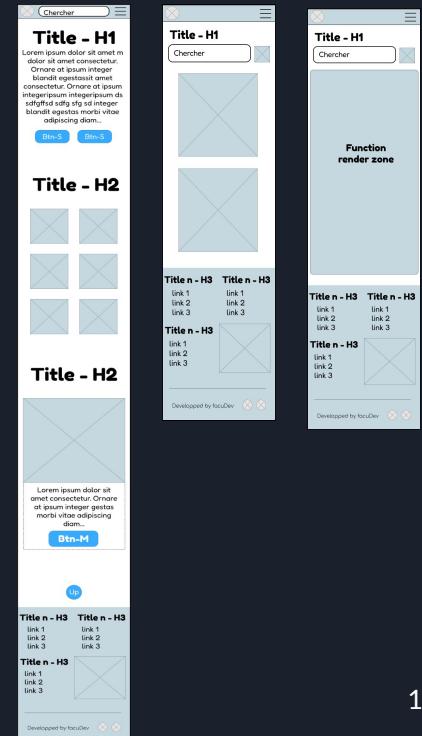
Communauté/Événements



Messagerie/Profil



Petits écrans

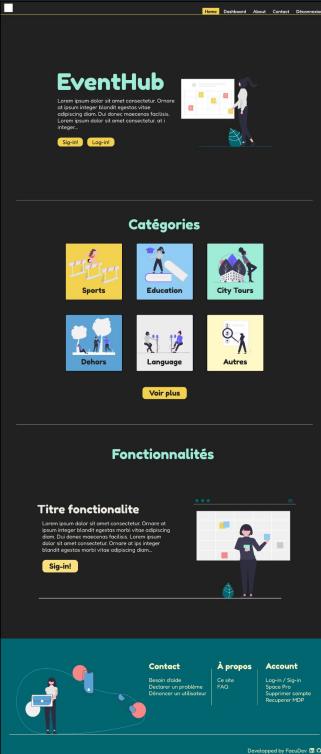


Maquettage - Mockup - 1440px

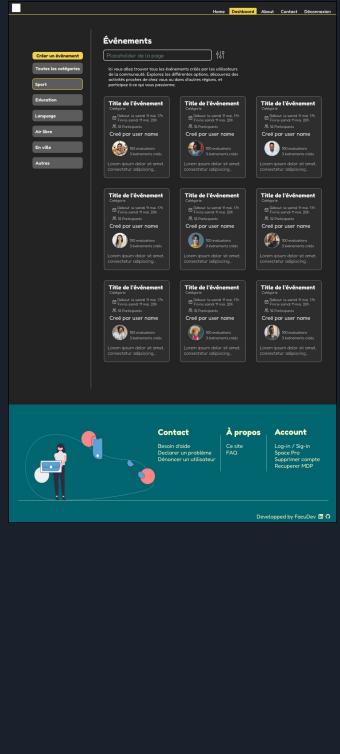
Accueil



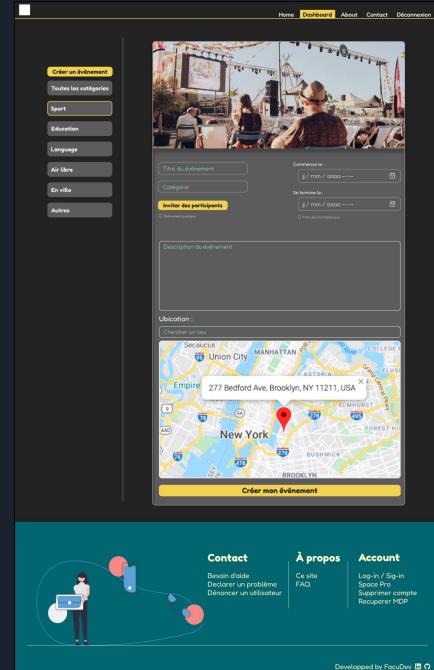
Événements



Événement/:id



Événement/création



Maquettage - Mockup - 360px

ACCUEIL

EventHub
Lorem ipsum dolor sit amet consectetur. Ornare et ipsum integer euismod diam non eros id est. Sed donec mosemias facilis. Lorem ipsum dolor sit amet consectetur et integer...

Catégories

- Sports
- Education

Fonctionnalités

- Envoyer un message à l'autre utilisateurs
- Envoyer un message à l'autre utilisateurs
- Envoyer un message à l'autre utilisateurs

À propos

Ce site
FAQ

Account

Log-in / Sign-up
S'inscrire Pro
Supprimer compte
Récupérer MDP

COMMUNAUTÉ

Communauté
Chercher

- User Name ★★★★★ 10 évaluations 3 événements créés
- User Name ★★★★★ 100 évaluations 3 événements créés
- User Name ★★★★★ 100 évaluations 3 événements créés

Contacter

À propos de moi
Lorem ipsum dolor sit amet consectetur adipisicing elit. Sed non nisi, temporibus sed, dolorum, dolores.

Mes événements

Titre de l'événement
Catégorie :
Début le vendredi 10 mai 20h
Fin le vendredi 11 mai 20h
12 personnes

Mes notes

Mes événements
Créer un événement

PROFIL/ID

Home
Profile
Communauté
Événements
Messagerie

Niko Huter
User Name ★★★★★ 10 évaluations 5 événements créés

À propos de moi
Lorem ipsum dolor sit amet consectetur adipisicing elit. Sed non nisi, temporibus sed, dolorum, dolores.

Mes événements

Titre de l'événement
Catégorie :
Début le vendredi 10 mai 20h
Fin le vendredi 11 mai 20h
12 personnes

Mes notes

Mes événements
Créer un événement

MON COMPTE

Mon profile
Chercher

Niko Huter
User Name ★★★★★ 10 évaluations 5 événements créés

Bio
Lorem ipsum dolor sit amet consectetur adipisicing elit. Sed non nisi, temporibus sed, dolorum, dolores.

Mes notes

Mes événements
Créer un événement

Titre de l'événement
Catégorie :
Début le vendredi 10 mai 20h
Fin le vendredi 11 mai 20h
12 personnes

Dates :

Ubication :
Paque très loin
60 Avenue de Versailles
MANHATTAN
Ford Ave, Brooklyn, NY 11
New York

Créé par user name
User Name ★★★★★ 100 évaluations 3 événements créés

Mes avis

Mes événements
Créer l'événement

ÉVÉNEMENT

Titre de l'événement
Catégorie :
Début le vendredi 10 mai 20h
Fin le vendredi 11 mai 20h
12 personnes

Dates :

Ubication :
Paque très loin
60 Avenue de Versailles
MANHATTAN
Ford Ave, Brooklyn, NY 11
New York

Créé par user name
User Name ★★★★★ 100 évaluations 3 événements créés

Mes avis

Mes événements
Créer l'événement

ÉVÉNEMENTS

Créer un événement

Événements
Chercher

Trier par
Plus récent

Créer par user name
User Name ★★★★★ 100 évaluations 3 événements créés

Titre de l'événement
Catégorie :
Début le vendredi 10 mai 20h
Fin le vendredi 11 mai 20h
12 personnes

Dates :

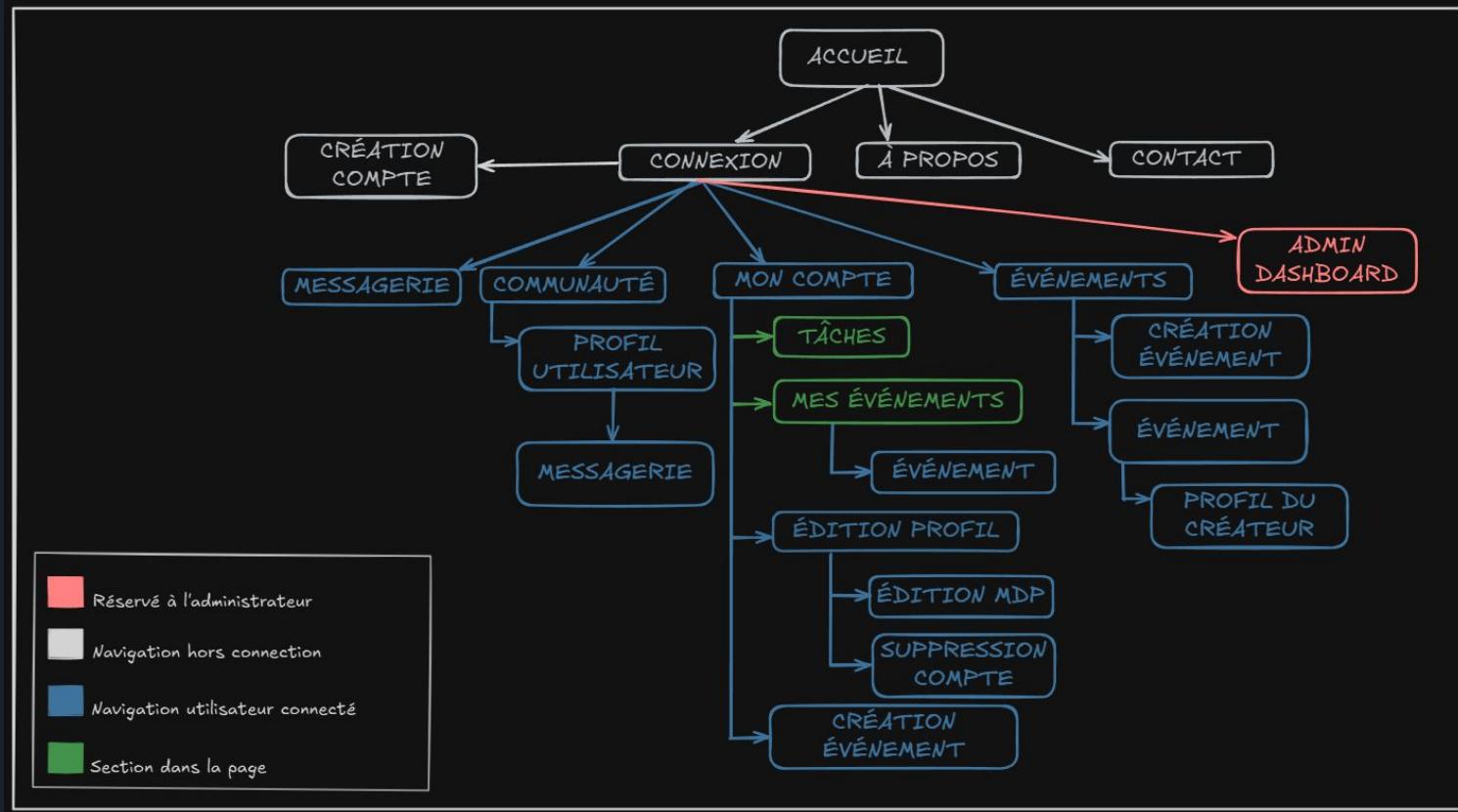
Ubication :
Paque très loin
60 Avenue de Versailles
MANHATTAN
Ford Ave, Brooklyn, NY 11
New York

Créé par user name
User Name ★★★★★ 100 évaluations 3 événements créés

Mes avis

Mes événements
Créer l'évenement

Conception - Arborescence du site

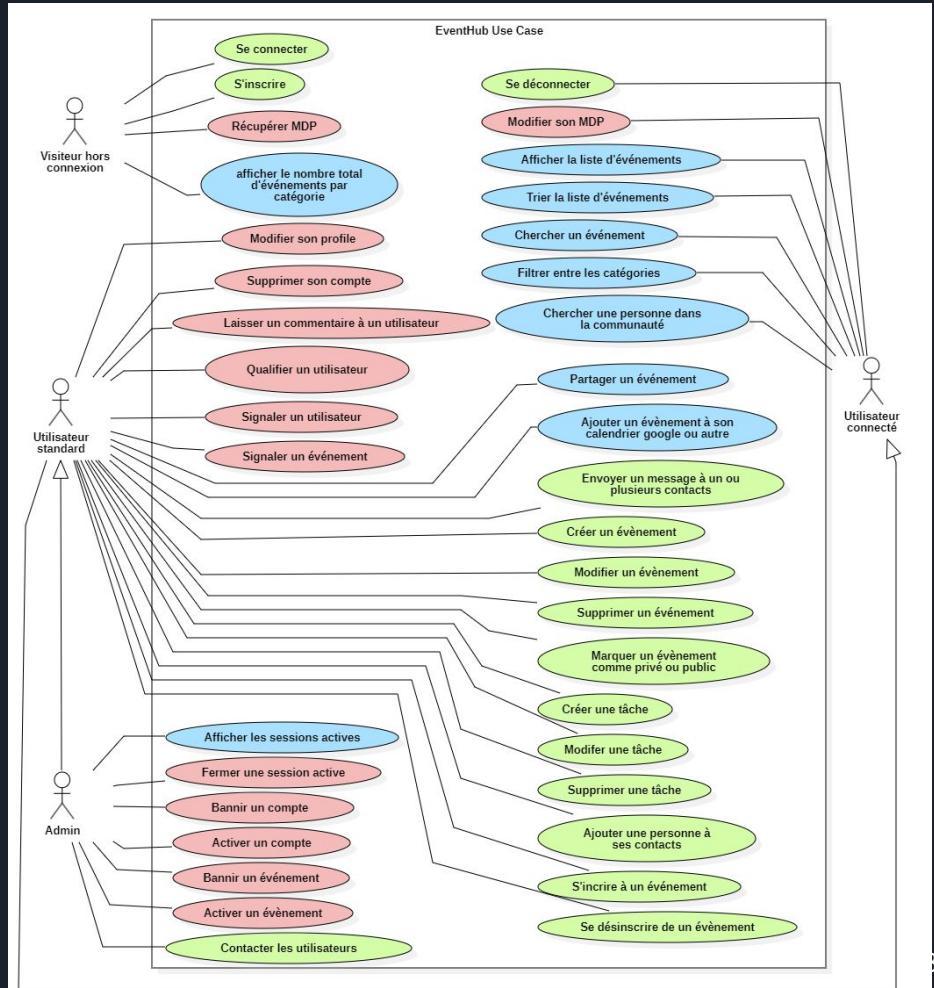


Conception - UML - Use Case

Actions ayant un impact sur les comptes utilisateurs.

Actions sans création d'éléments dans la base de données.

Actions avec un impact sur la base de données.



Conception - UML - Activité - Création événement

Appels asynchrones

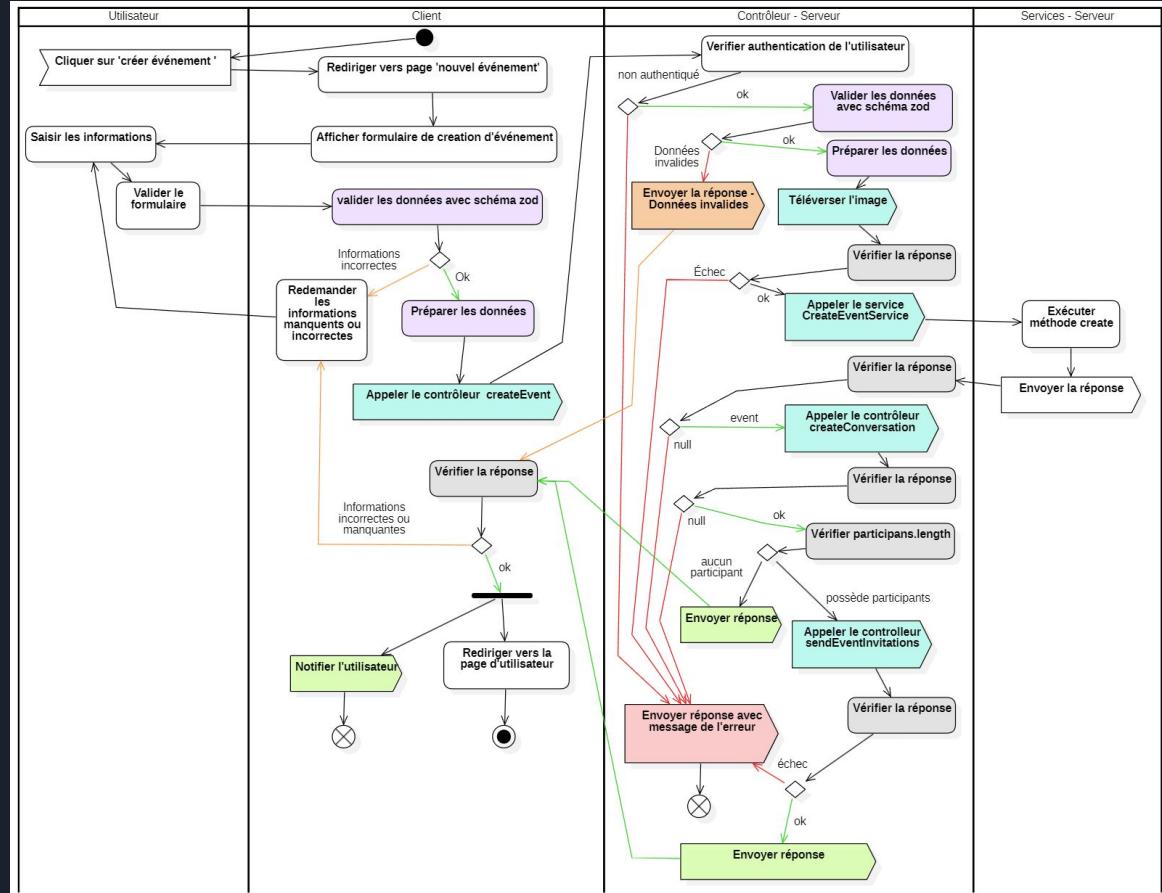
Manipulation des données

Validation des données

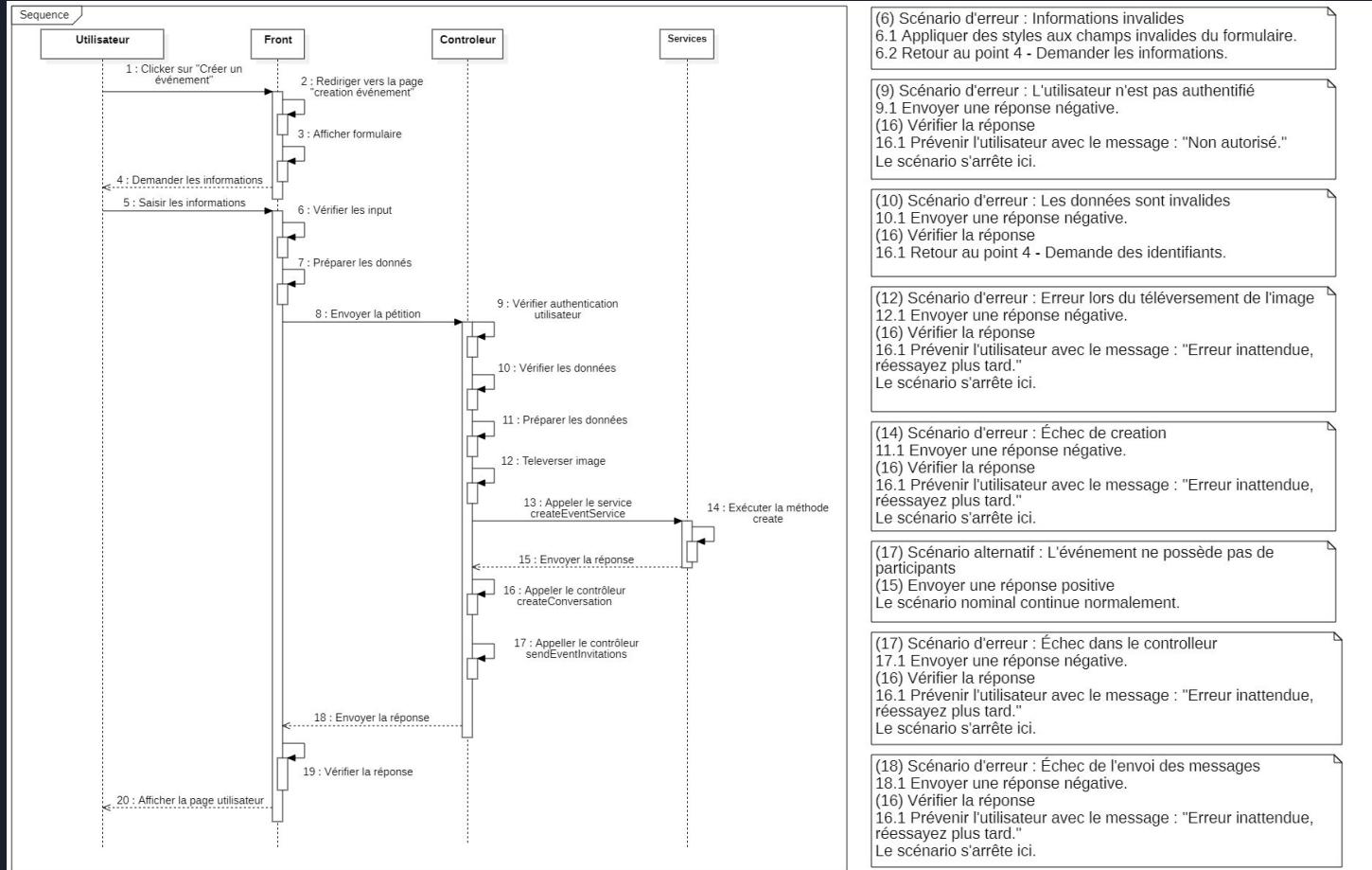
Réponse négative sans arrêter le flow

Réponse négative arrêtant le flow

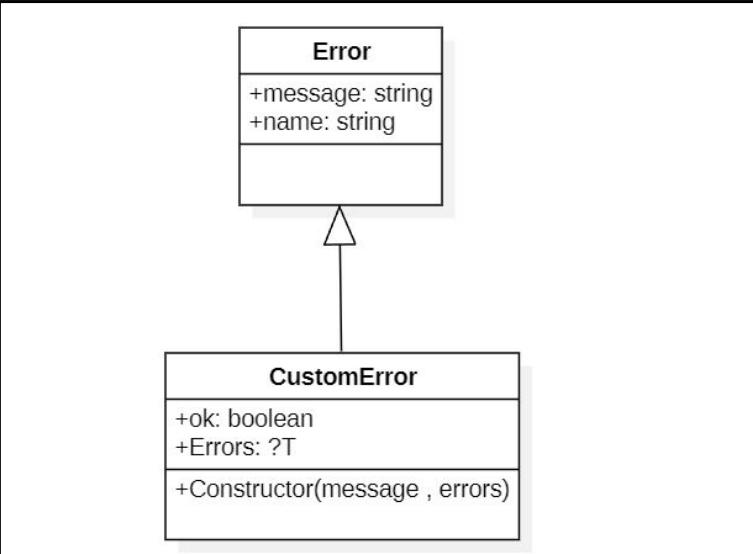
Réponse positive



Conception - UML - Séquence - Crédit événement

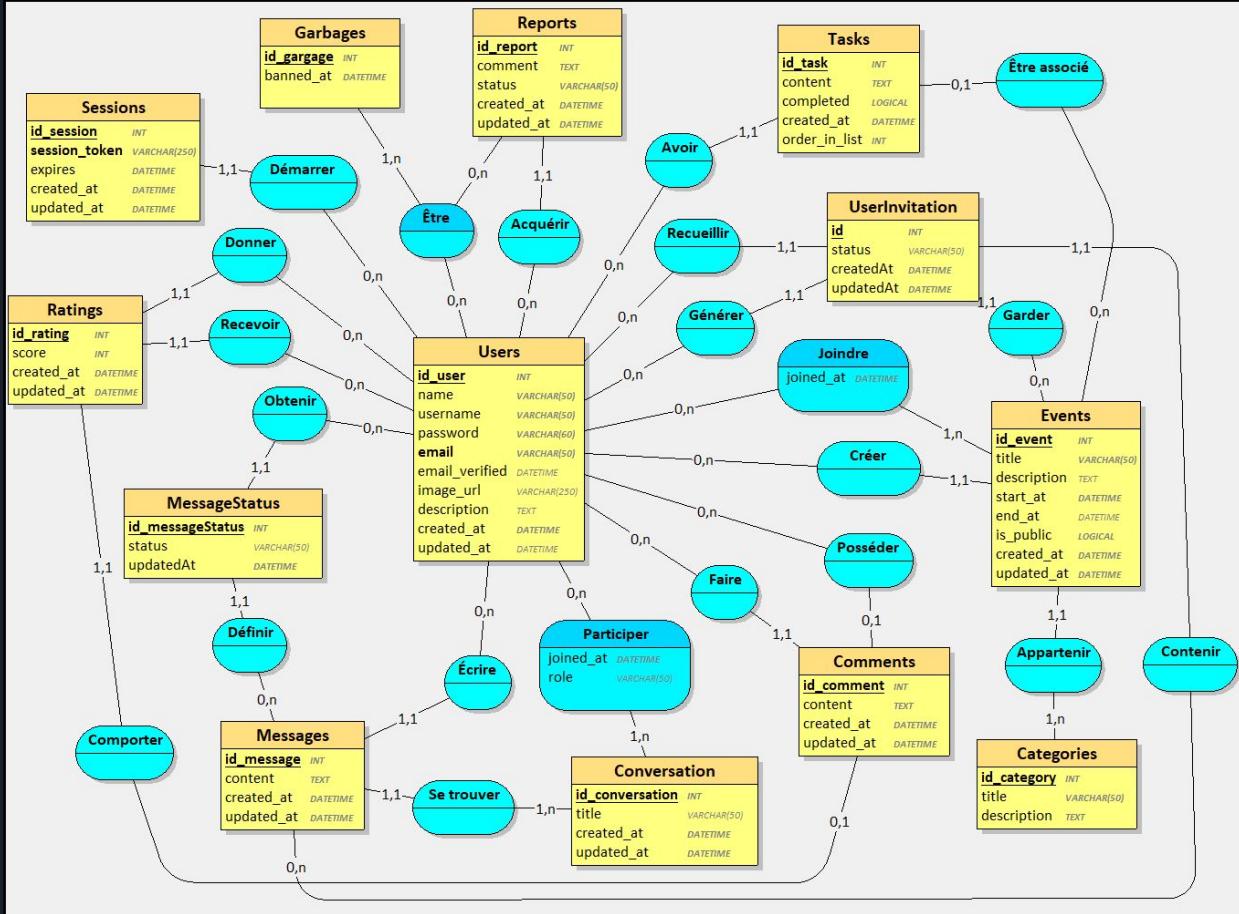


Conception - UML - Classe - CustomError



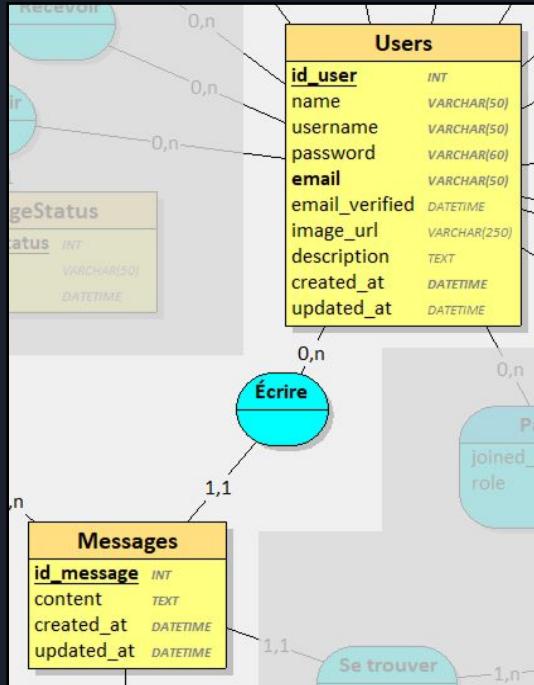
```
/**  
 * Represents a custom error that extends the built-in 'Error' class.  
 * This class allows for additional structured error information.  
 *  
 * @template T - The type of additional error details (e.g., validation errors).  
 */  
class CustomError<T = unknown> extends Error {  
    /**  
     * Indicates that the operation was not successful.  
     * Always 'false' for instances of 'CustomError'.  
     */  
    public ok: boolean;  
  
    /**  
     * A error message describing the issue.  
     */  
    public message: string;  
  
    /**  
     * Additional error details, such as validation errors or metadata.  
     * This field is optional.  
     */  
    public errors?: T;  
  
    /**  
     * Creates an instance of 'CustomError'.  
     *  
     * @param {string} message - A descriptive error message.  
     * @param {T} [errors] - Optional additional error details.  
     */  
    constructor(message: string, errors?: T) {  
        super(message);  
        this.name = this.constructor.name;  
        this.ok = false;  
        this.message = message;  
        this.errors = errors;  
    }  
}  
export default CustomError;
```

Conception - MCD

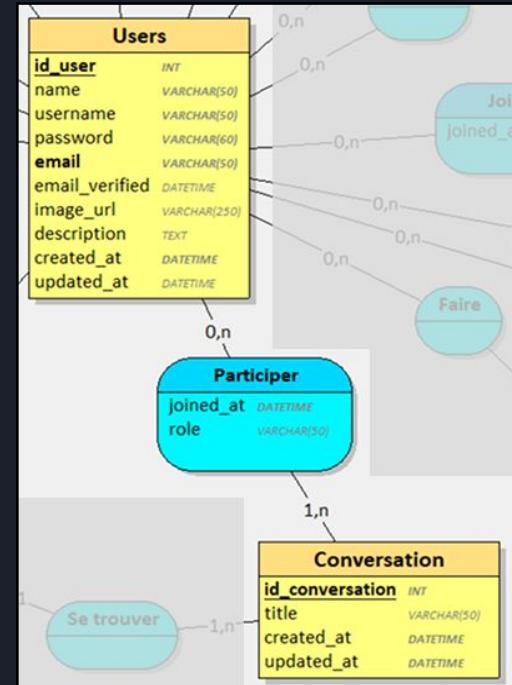


Conception - MCD

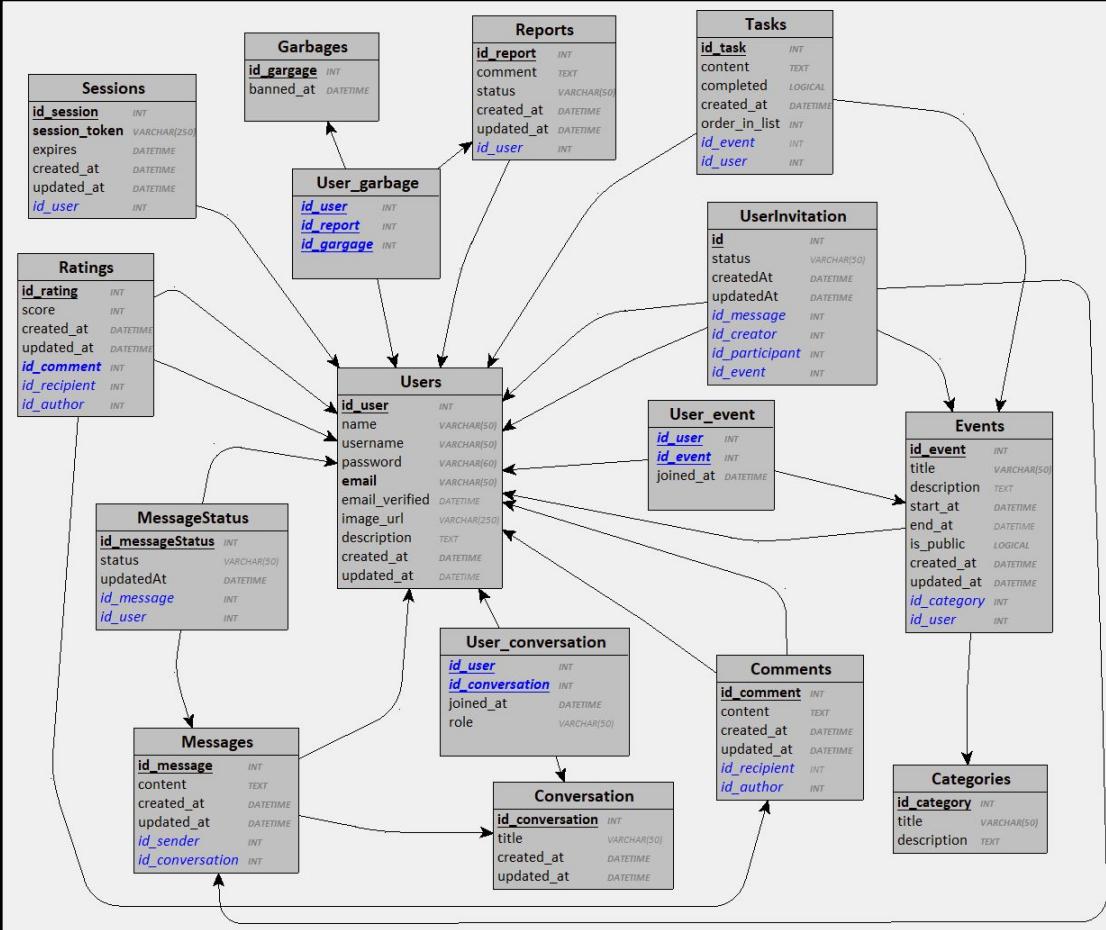
Contrainte d'Intégrité Fonctionnelle (CIF)



Contrainte d'Intégrité Multiple (CIM)

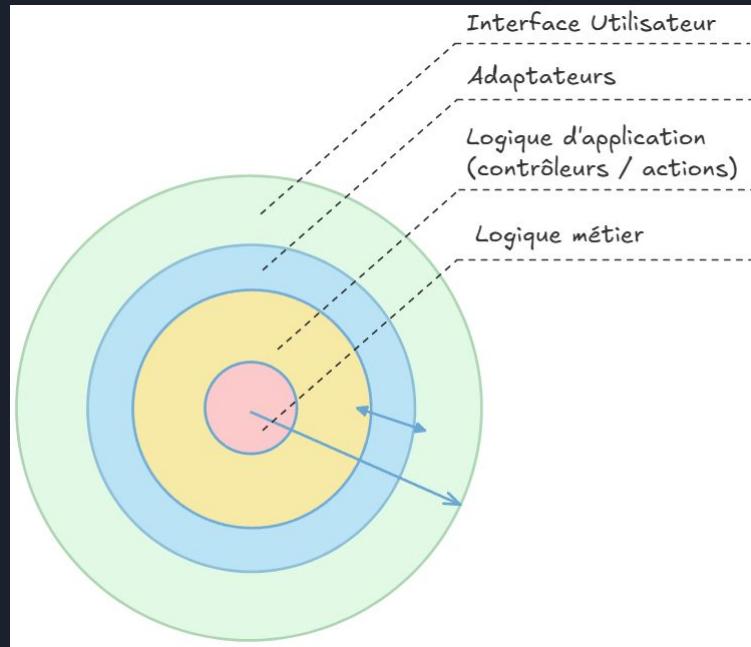


Conception - MLD



Architecture - Clear Architecture

```
/app
  /about
  /contact
  /login
  etc...
  layout.tsx
  not-found.tsx
  page.tsx
/components
  /ui
  /layout
  /forms
/actions
  /authServerActions
    checkIsAuthenticated.ts
    etc...
    authActions.test.ts
  /userServerActions
    selectUserById.ts
    etc...
    userActions.test.ts
/lib
  /prisma
    /migrations
    prisma.ts
    prisma_schema.prisma
  /auth
    authConfig.ts
    next-auth.ts
    emailTemplate.ts
  /zod
    zodSchemas.ts
    handleErrors.ts
    handleError.test.ts
/services
  eventService.ts
  event.test.ts
  etc...
/types
  appTypes.ts
  servicesTypes.ts
/utils
  adapters/
    userAdapters.ts
    userAdapters.test.ts
    etc...
    FormatDates.ts
    getFormDataValues.ts
    calculateUserScore.ts
    etc...
  middleware.ts
  tailwind.config.ts
  tsconfig.json
  package.json
  next.config.ts
  .env
```



Fonctionnalité Front - Menu de navigation avec 0 JavaScript

The screenshot displays a web application interface with a dark background. At the top, there is a horizontal navigation bar with the following items: Home (highlighted in yellow), Dashboard, Events, About, Contact, and Déconnexion. Below the navigation bar, the word "EventHub" is displayed in a large, bold, light green font. To the right of the title, there is a white rectangular area containing a grid of colored squares (yellow and red) representing event icons. A cartoon illustration of a person with purple hair and a white shirt is standing next to the grid, pointing at one of the squares. Below the title and grid, there is a block of text describing the service, followed by a yellow button labeled "Voir le tableau de bord".

EventHub est votre solution tout-en-un pour la gestion d'événements et de RDV. Simplifiez l'organisation de vos activités avec notre interface moderne et intuitive. Vous pouvez facilement synchroniser vos événements avec Google Calendar, gérer vos tâches, et communiquer avec vos collègues. Que vous planifiez un petit rassemblement ou un grand événement, EventHub vous offre les outils nécessaires pour réussir !

Voir le tableau de bord

Fonctionnalité Front - Menu de navigation avec 0 JavaScript

```
8  export default async function Header() {
9
10    return (
11      <>
12        <input type="checkbox" id="menu-toggle" className="hidden peer" />
13        <header className="sticky z-50 top-0 min-w-full sm:max-h-none sm:items-end flex justify-between overflow-hidden max-h-[44px] transition-[max-height] duration-500 ease-in-out peer-checked:max-h-[500px] select-none px-5 border-b-[1px] border-dark-bg dark:border-light-yellow bg-light-ciel/95 dark:bg-dark-bg/95">
14          <ThemeSwitcher />
15          <nav>
16            <div className="flex flex-col items-end">
17              <label
18                id="menu-label"
19                htmlFor="menu-toggle"
20                className="cursor-pointer sm:hidden p-2 active:scale-95">
21                <NavMenuButton />
22              </label>
23            </div>
24            <ul className="mr-[-1.25rem] sm:mr-0 gap-4 font-bold text-dark-bg dark:text-dark-grey sm:flex sm:flex-row">
25              </ul>
26            </div>
27          </nav>
28        </header>
29      </>
30    );
31  }
32
```

Fonctionnalité Front - Menu de navigation avec 0 JavaScript

```
8  export default async function Header() {
9
10    return (
11      <>
12        <input type="checkbox" id="menu-toggle" className="hidden peer" />
13        <header className="sticky z-50 top-0 min-w-full sm:max-h-none sm:items-end flex justify-between overflow-hidden max-h-[44px] transition-[max-height] duration-500 ease-in-out peer-checked:max-h-[500px] select-none px-5 border-b-[1px] border-dark-bg dark:border-light-yellow bg-light-ciel/95 dark:bg-dark-bg/95">
14          <ThemeSwitcher />
15          <nav>
16            <div className="flex flex-col items-end">
17              <label
18                id="menu-label"
19                htmlFor="menu-toggle"
20                className="cursor-pointer sm:hidden p-2 active:scale-95">
21                <NavMenuItem />
22              </label>
23            </div>
24            <ul className="mr-[-1.25rem] sm:mr-0 gap-4 font-bold text-dark-bg dark:text-dark-grey sm:flex sm:flex-row">
25              <li>
26                <NavMenuItem />
27              </li>
28            </ul>
29          </div>
30        </nav>
31      </header>
32    </>
33  );
34}
35
36
37
38
```

Fonctionnalité Front - Menu de navigation avec 0 JavaScript

```
8  export default async function Header() {
9
10    return (
11      <>
12        <input type="checkbox" id="menu-toggle" className="hidden peer" />
13        <header className="sticky z-50 top-0 min-w-full sm:max-h-none sm:items-end flex justify-between overflow-hidden max-h-[44px] transition-[max-height] duration-500 ease-in-out peer-checked:max-h-[500px] select-none px-5 border-b-[1px] border-dark-bg dark:border-light-yellow bg-light-ciel/95 dark:bg-dark-bg/95">
14          <ThemeSwitcher />
15          <nav>
16            <div className="flex flex-col items-end">
17              <label
18                id="menu-label"
19                htmlFor="menu-toggle"
20                className="cursor-pointer sm:hidden p-2 active:scale-95">
21                <NavMenuItem />
22              </label>
23            </div>
24            <ul className="mr-[-1.25rem] sm:mr-0 gap-4 font-bold text-dark-bg dark:text-dark-grey sm:flex sm:flex-row">
25              <li>
26                <NavMenuItem />
27              </li>
28            </ul>
29          </div>
30        </nav>
31      </header>
32    </>
33  );
34}
35
```

Fonctionnalité Front - Menu de navigation avec 0 JavaScript

```
8  export default async function Header() {
9
10    return (
11      <>
12        <input type="checkbox" id="menu-toggle" className="hidden peer" />
13        <header className="sticky z-50 top-0 min-w-full sm:max-h-none sm:items-end flex justify-between overflow-hidden max-h-[44px] transition-[max-height] duration-500 ease-in-out peer-checked:max-h-[500px] select-none px-5 border-b-[1px] border-dark-bg dark:border-light-yellow bg-light-ciel/95 dark:bg-dark-bg/95">
14          <ThemeSwitcher />
15          <nav>
16            <div className="flex flex-col items-end">
17              <label
18                id="menu-label"
19                htmlFor="menu-toggle"
20                className="cursor-pointer sm:hidden p-2 active:scale-95">
21                <NavMenuItem />
22              </label>
23            </div>
24            <ul className="mr-[-1.25rem] sm:mr-0 gap-4 font-bold text-dark-bg dark:text-dark-grey sm:flex sm:flex-row">
25              <li>
26                <NavMenuItem />
27              </li>
28            </ul>
29          </div>
30        </nav>
31      </header>
32    </>
33  );
34}
35
```

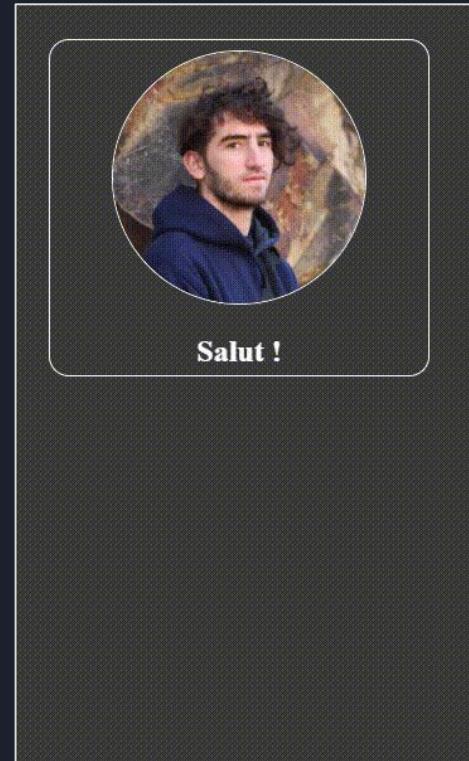
Fonctionnalité Front - Menu de navigation avec 0 JavaScript

```
html {  
    background-color: #rgb(59, 59, 59);  
    color: white;  
    padding: 2rem;  
}  
article img {  
    width: 200px;  
    height: 200px;  
    border: solid 1px white;  
    border-radius: 100%;  
}  
article {  
    text-align: center;  
    height: 250px;  
    width: 300px;  
    border: solid 1px white;  
    border-radius: 1rem;  
    padding: .5rem 0;  
    overflow: hidden;  
    transition: height .5s;  
}  
article:hover {  
    height: auto;  
}
```



Fonctionnalité Front - Menu de navigation avec 0 JavaScript

```
html {  
    background-color: #rgb(59, 59, 59);  
    color: white;  
    padding: 2rem;  
}  
article img {  
    width: 200px;  
    height: 200px;  
    border: solid 1px white;  
    border-radius: 100%;  
}  
article {  
    text-align: center;  
    height: 250px;  
    width: 300px;  
    border: solid 1px white;  
    border-radius: 1rem;  
    padding: .5rem 0;  
    overflow: hidden;  
    transition: height .5s;  
}  
article:hover {  
    height: 500px;  
}
```



Fonctionnalité Front - Menu de navigation avec 0 JavaScript

```
html {  
    background-color: □rgb(59, 59, 59);  
    color: ■white;  
    padding: 2rem;  
}  
article img {  
    width: 200px;  
    height: 200px;  
    border: solid 1px ■white;  
    border-radius: 100%;  
}  
article {  
    text-align: center;  
    height: 250px;  
    width: 300px;  
    border: solid 1px ■white;  
    border-radius: 1rem;  
    padding: .5rem 0;  
    overflow: hidden;  
    transition: height .5s;  
    interpolate-size: allow-keywords;  
}  
article:hover {  
    height: auto;  
}
```



Fonctionnalité Front - Menu de navigation avec 0 JavaScript

Interpolate-size et sa compatibilité...

	Chrome	Edge	Firefox	Opera	Safari	Chrome / Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
interpolate-size 	✓ 129	✗ No	✗ No	✗ 115	✗ No	✗ 129	✗ No	✗ No	✗ No	✗ No	✓ 129	✗ No
allow-keywords 	✓ 129	✗ No	✗ No	✗ 115	✗ No	✗ 129	✗ No	✗ No	✗ No	✗ No	✓ 129	✗ No
numeric-only 	✓ 129	✗ No	✗ No	✗ 115	✗ No	✗ 129	✗ No	✗ No	✗ No	✗ No	✓ 129	✗ No

Fonctionnalité Front - Animation scroll

The screenshot shows the homepage of a web application named "EventHub". The header features a navigation bar with tabs: Accueil (highlighted in yellow), Profil, Communauté, Événements, and Messages. Below the header, the main content area has a dark background with a light gray grid pattern. On the left, the "EventHub" logo is displayed in a large, bold, teal font. To the right of the logo is a descriptive text block:

EventHub est votre solution tout-en-un pour la gestion d'événements et de RDV. Simplifiez l'organisation de vos activités avec notre interface moderne et intuitive. Vous pouvez facilement synchroniser vos événements avec Google Calendar, gérer vos tâches, et communiquer avec vos collègues. Que vous planifiez un petit rassemblement ou un grand événement, EventHub vous offre les outils nécessaires pour réussir !

Below this text is a yellow button labeled "Voir le tableau de bord". To the right of the text, there is a white rectangular area containing a calendar-like grid with colored boxes (yellow and red) representing events. A stylized illustration of a woman with purple hair, wearing a white shirt and dark pants, is standing next to the grid, pointing at it with her right hand. The bottom of the screen features a teal decorative graphic element.

Fonctionnalité Front - Animation scroll

```
export default async function Home() {
  return (
    <main className="flex min-h-screen flex"
      <section className="animate-scroll flex flex-col items-center justify-center">
        <h1>Frontend
        <p>Frontend
      </section>
      <section className="animate-scroll flex flex-col items-center justify-center">
        <h2>Backend
        <p>Backend
      </section>
      <section className="animate-scroll flex flex-col items-center justify-center">
        <h3>Database
        <p>Database
      </section>
    </main>
  );
}

/* ===== Scroll animation ===== */
/* Can't use this animation with Tailwind because animation-timeline is not supported by Tailwind */
@keyframes appear {
  from {
    opacity: 0;
    transform: scale(0.8);
  }
  to {
    opacity: 1;
  }
}
/* Animation applied only for large screens because the range is too big for small screens */
@media screen and (min-width: 768px) {
  .animate-scroll > h1,
  .animate-scroll > div,
  .animate-scroll > a {
    animation: appear linear;
    animation-timeline: view();
    animation-range: entry 0% cover 11%;
    animation-fill-mode: both;
    animation-duration: 1ms; /* Firefox requires this to apply the animation */
  }
}
```

Fonctionnalité Front - Animation scroll

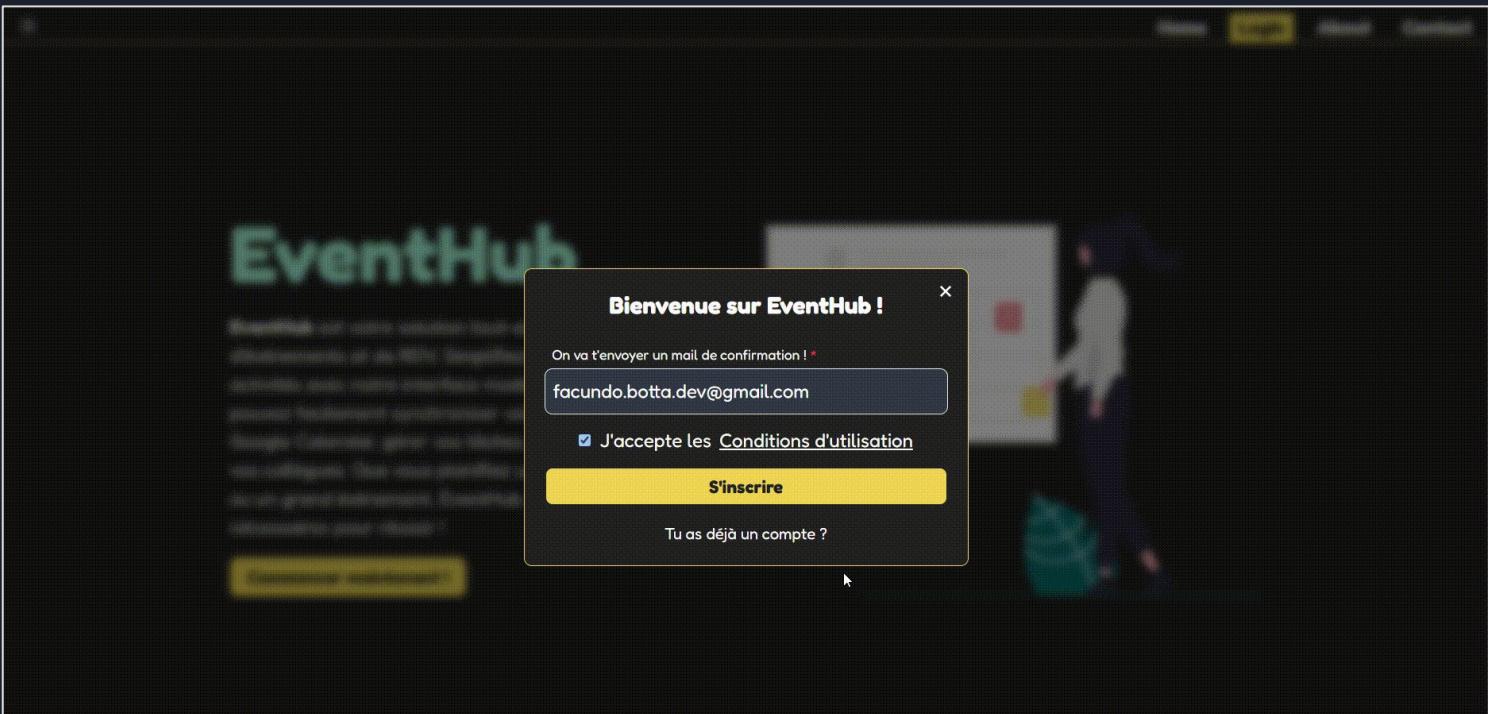
view() et sa compatibilité...

Browser compatibility

[Report problems with this compatibility data on GitHub](#)

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
view()	✓	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗
	115	115	114	101	No	115	No	77	No	23.0	115	No

Fonctionnalité Back - Crédation de compte avec e-mail



Fonctionnalité Back - Création de compte avec e-mail

```
components > forms > LogForm.tsx > LogForm
15  export default function LogForm() {
152
153    return (
154      <form
155        className="relative flex flex-col items-center gap-3 w-full max-w-md p-5 mx-5 border-light-yellow dark:bg-dark-bg"
156        onSubmit={formType === 'Sign-Up' ? handleSignUpSubmit : handleLogInSubmit}
157      >
158        <Icon
159          type="plus" ...
160        />
161        <h2 className="text-2xl mb-3 font-bold text-center"> ...
162        </h2>
163        <Input
164          className={inputClasses}
165          required={true}
166          label={
167            formType === 'Sign-In'
168            ? 'Email'
169            : 'On va te envoyer un mail de confirmation !'
170          }
171          type="email"
172          id="Email"
173          name="email"
174          placeholder="Email"
175          disabled={isPending}
176          autoComplete="email"
177          errorStyles={{ color: 'red', fontSize: '1rem' }}
178          error={{ message: error.mail?.message, value: error.mail?.value }}
179        />
180      </form>
181    );
182  }
183
```

```
const handleSignUpSubmit = (e: React.FormEvent<HTMLFormElement>) => {
  e.preventDefault();
  setError({
    mail: { message: '', value: false },
    password: { message: '', value: false },
    conditions: { message: '', value: false },
  });
  if (!conditionsAccepted) {
    setError({
      ...error,
      conditions: {
        message: "Veuillez accepter les conditions d'utilisation",
        value: true,
      },
    });
    return;
  }

  try {
  } catch (error) {
  }
};


```

Fonctionnalité Back - Crédit de compte avec e-mail

```
const handleSignUpSubmit = (e: React.FormEvent<HTMLFormElement>) => {
  e.preventDefault();
  setError({ ... });
};

if (!conditionsAccepted) { ... }

try {
  const formData = new FormData(e.currentTarget);
  const email = getFormDataStringValue(formData, 'email');
  // validate email
  emailSchema.parse({ email });

  startTransition(async () => {
    const result = await emailSignInServerAction(email);

    if (!result?.ok) {
      setError({
        ...error,
        mail: { message: result?.message as string, value: true },
      });
    } else if (result.ok) {
      setEmailSent(true);
    }
  });
} catch (error) {
  const handledError = handleError(error);
  setError((prevError) => ({
    ...prevError,
    mail: { message: handledError.message, value: true },
  }));
}
};
```

```
export const emailSchema = z.object({
  email: z
    .string()
    .email({ message: 'Veuillez utiliser une adresse e-mail valide' }),
});
```

```
15 export default function LogForm() {
16   const Router = useRouter();
17   const [formType, setFormType] = useState<string>('Sign-In');
18   const [isPending, startTransition] = useTransition();
19   const [conditionsAccepted, setConditionsAccepted] = useState(false);
20   const [error, setError] = useState({
21     mail: { message: '', value: false },
22     password: { message: '', value: false },
23     conditions: { message: '', value: false },
24   });
};
```

Fonctionnalité Back - Crédation de compte avec e-mail

Bienvenue sur EventHub !

On va t'envoyer un mail de confirmation ! *

Veuillez utiliser une adresse e-mail valide

J'accepte les [Conditions d'utilisation](#)

S'inscrire

Tu as déjà un compte ?

Bienvenue sur EventHub !

On va te envoyer un mail de confirmation ! *

Cet email est déjà associé à un compte

J'accepte les [Conditions d'utilisation](#)

S'inscrire

Vous Avez déjà un compte ?

Log handleError :

```
▼ handledError: CustomError: Veuillez utiliser une adresse e-mail valide at handleError (webpack-intern
  ▼ errors: Array(1)
    ▼ 0:
      message: "Veuillez utiliser une adresse e-mail valide"
      ▶ path: ['email']
      ▶ [[Prototype]]: Object
      length: 1
      ▶ [[Prototype]]: Array(0)
      name: "CustomError"
      ok: false
      message: "Veuillez utiliser une adresse e-mail valide"
      stack: "CustomError: Veuillez utiliser une adresse e-mail valide\n      at handleError (webpack-intern
      ▶ [[Prototype]]: Error
      ▶ [[Prototype]]: Object
```

Log result.ok == false

LogForm.tsx:124

```
▶ {ok: false, message: 'Cet email est déjà associé à un compte'}
```

Fonctionnalité Back - Créeation de compte avec e-mail

```
'use server';

import { signIn } from '@lib/auth/authConfig';
import { handleError } from '@lib/zod/handleError';
import { emailSchema } from '@lib/zod/zodSchemas';
import { selectUserByEmailService } from '@services/userServices';

/**
 * Handles the email sign-in process.
 *
 * This function sends an email to the provided email address with a link to sign in.
 *
 * @param {string} email - The email address to sign in with.
 * @returns {Promise<{ ok: boolean, message?: string }>} An object indicating the result of the
 * sign-in process.
 */
export async function emailSignInServerAction(
  email: string
): Promise<{ ok: boolean; message: string }> {
  try {
    // Validate the email using Zod schema
    emailSchema.parse({ email });

    // Check if the user already exists
    const user = await selectUserByEmailService({ email });
    if (user) {
      return {
        ok: false,
        message: 'Un compte existe déjà avec cet email',
      };
    }

    // Proceed with the sign-in process
    await signIn('nodemailer', {
      email,
      callbackUrl: '/profile',
      redirect: false,
    });

    return { ok: true, message: 'Email envoyé' };
  } catch (error) {
    return handleError(error);
  }
}
```

```
export const selectUserByEmailService = async ({  
  email,  
}: {  
  email: string;  
}): Promise<Partial<User> | null> => {  
  try {  
    return prisma.user.findUnique({  
      where: {  
        email,  
      },  
      select: {  
        id: true,  
        role: true,  
        password: true,  
        hasPassword: true,  
        email: true,  
      },  
    });  
  } catch (error) {  
    console.error('selectUserByEmailService', error);  
    return null;  
  }  
};
```

Fonctionnalité Back - Crédation de compte avec e-mail

```
lib > auth > TS authConfig.ts > ...
...
12 export const { handlers, auth, [signIn, signOut] } = NextAuth({
13   trustHost: true, // This is required for NextAuth to work properly in localhost
14   adapter: PrismaAdapter(prisma),
15   secret: process.env.AUTH_SECRET,
16   session: { ... },
17 },
18   pages: { ... },
19   callbacks: { ... },
20   providers: [
21     Google({ ... }),
22     Nodemailer({
23       server: {
24         host: process.env.EMAIL_SERVER_HOST,
25         port: parseInt(process.env.EMAIL_SERVER_PORT!, 10),
26         auth: {
27           user: process.env.EMAIL_SERVER_USER,
28           pass: process.env.EMAIL_SERVER_PASSWORD,
29         },
30       },
31       from: process.env.EMAIL_FROM,
32       sendVerificationRequest: async (params) => {
33         const { identifier, url, provider } = params;
34         const transport = createTransport(provider.server);
35         const result = await transport.sendMail({
36           to: identifier,
37           from: provider.from,
38           subject: 'Bienvenue sur EventHub !',
39           text: text({ url }), // This is the text version of the email
40           html: html({ url }), // This is the HTML version of the email
41         });
42         const failed = result.rejected.concat(result.pending).filter(Boolean);
43         if (failed.length) {
44           throw new Error(`Email(s) ${failed.join(', ')} could not be sent`);
45         }
46       },
47     }),
48   ],
49   Credentials({ ... }),
50 },
51 );
52 
```

```
.env
1 ######
2 ## MySQL Database configuration #
3 ######
4
5 DATABASE_URL="mysql://admin:password@localhost:3306/filrouge2024?schema=public"
6 DB_NAME="filrouge2024"
7 DB_USER="admin"
8 DB_PASSWORD="password"
9 SERVER_PORT=3306
10 NEXT_AUTH_URL="http://localhost:3000"
11
12 ######
13 ## IDENTITY ACCESS AND MANAGEMENT #
14 ######
15
16 AUTH_GOOGLE_ID="534357650281-nklhinq8om34ht4v7hfd1gpd4hu0gm6a.apps.googleusercontent.com"
17 AUTH_GOOGLE_SECRET="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"
18 NEXTAUTH_URL="http://localhost:3000"
19 # generated with openssl rand -base64 33
20 AUTH_SECRET="g8JdX+OWEqRHcjjjI2YNJPiHxT6c2D6fdwDe6x1yIX6"
21
22 ######
23 ## EMAIL SERVER CONFIGURATION #
24 ######
25 EMAIL_FROM="noreply@eventhub.com"
26 EMAIL_SERVER_HOST="smtp.gmail.com"
27 EMAIL_SERVER_PORT=587
28 EMAIL_SERVER_USER="nutuestampados@gmail.com"
29 # Temporary password for adrar
30 EMAIL_SERVER_PASSWORD="mkmv kpbq lsnm dahp"
```

Fonctionnalité Back - Créeation de compte avec e-mail

```
lib > auth > TS authConfig.ts > ...
...
12  export const { handlers, auth, signIn, signOut } = NextAuth({
13    trustHost: true, // This is required for NextAuth to work properly in localhost
14    adapter: PrismaAdapter(prisma),
15    secret: process.env.AUTH_SECRET,
16  > session: { ... },
17  > pages: { ... },
18  > callbacks: { ... },
19  > providers: [
20    Google({ ... }),
21  ],
22  > Nodemailer({
23    server: {
24      host: process.env.EMAIL_SERVER_HOST,
25      port: parseInt(process.env.EMAIL_SERVER_PORT!, 10),
26      auth: {
27        user: process.env.EMAIL_SERVER_USER,
28        pass: process.env.EMAIL_SERVER_PASSWORD,
29      },
30    },
31    from: process.env.EMAIL_FROM,
32    sendVerificationRequest: async (params) => {
33      const { identifier, url, provider } = params;
34      const transport = createTransport(provider.server);
35      const result = await transport.sendMail({
36        to: identifier,
37        from: provider.from,
38        subject: `Bienvenue sur EventHub !`,
39        text: text({ url }), // This is the text version of the email
40        html: html({ url }), // This is the HTML version of the email
41      });
42      const failed = result.rejected.concat(result.pending).filter(Boolean);
43      if (failed.length) {
44        throw new Error(`Email(s) ${failed.join(', ')} could not be sent`);
45      }
46    },
47  > Credentials({ ... }),
48  > ],
49});
```

```
lib > TS emailTemplate.ts > ...
...
1  export function html(params: { url: string }) {
2    const { url } = params;
3
4    const color = {
5      background: '#f9f9f9',
6      text: '#444',
7      mainBackground: '#fff',
8      buttonBackground: '#F3D250',
9      buttonBorder: 'black',
10     buttonText: 'black',
11   };
12
13   return `...
14   `;
15 }
16
17 // Email Text body (fallback for email clients that don't render HTML, e.g. feature phones)
18 export function text({ url }: { url: string }) {
19   return `Bienvenue sur EventHub !\nNous sommes ravis de vous accueillir. Pour continuer, cliquez sur le lien suivant pour vous connecter:${url}\n\nSi vous n'avez pas demandé cette connexion, vous pouvez ignorer cet email.\n`;
20 }
```

Bienvenue sur EventHub !

Nous sommes ravis de vous accueillir. Pour continuer, cliquez sur le bouton ci-dessous pour commencer à utiliser votre compte !.

Se connecter

<http://localhost:3000/api/auth/callback/nodemailer?callbackUrl=http%3A%2F%2Flocalhost%3A3000%2Flogin&token=bdf582b70c049b817a28dabc40ea3a0cc5bcad3a46e9844cdf80d2ac489e93c5&email=facundo.botta.dev%40gmail.com>

Fonctionnalité Back - Modification de compte - mdp

```
import { getToken } from 'next-auth/jwt';
import { NextRequest, NextResponse } from 'next/server';

export async function middleware(req: NextRequest) {
  const token = await getToken({ req, secret: process.env.AUTH_SECRET });
  const isAuthenticated = !!token;
  const hasPassword = token?.hasPassword === true;
  const isAdmin = token?.role === 'admin';
  const isAdminPage = req.nextUrl.pathname.startsWith('/admin');

  if (isAdminPage) {
    if (isAuthenticated && isAdmin) {
      return NextResponse.next();
    } else {
      return NextResponse.redirect(new URL('/unauthorized', req.url));
    }
  }

  if (req.nextUrl.pathname.startsWith('/login') && isAuthenticated) {
    return NextResponse.redirect(new URL('/profile', req.url));
  }

  if (
    req.nextUrl.pathname.startsWith('/profile') ||
    req.nextUrl.pathname.startsWith('/messages') ||
    req.nextUrl.pathname.startsWith('/events') ||
    req.nextUrl.pathname.startsWith('/communaute')
  ) {
    if (!isAuthenticated) {
      return NextResponse.redirect(new URL('/login', req.url));
    }

    if (!hasPassword) {
      return NextResponse.redirect(new URL('/set-password', req.url));
    }
  }
}

return NextResponse.next();
}

export const config = {
  matcher: [
    '/login',
    '/admin/:path*',
    '/profile/:path*',
    '/messages/:path*',
    '/events/:path*',
    '/communaute/:path*',
  ],
};
```

Log jeton de session

```
{
  token: {
    name: null,
    email: 'facundo.botta.dev@gmail.com',
    picture: null,
    sub: 'cm7lyag1h0002rgq8m8fimjpi',
    id: 'cm7lyag1h0002rgq8m8fimjpi',
    role: 'user',
    hasPassword: false,
    iat: 1740576482,
    exp: 1741181282,
    jti: '780a1e7f-3a77-4f06-85f5-7982c2d1c534'
  }
}
```

Fonctionnalité Back - Modification de compte - mdp

http://localhost:3000/set-password

Bienvenue sur EventHub !

Vous devez d'abord créer un mot de passe pour pouvoir commencer.

Nouveau mot de passe *

Confirmer le mot de passe *

Créer un mot de passe

```
const passwordRegex = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[$!%*?&])[A-Za-z\d@$!%*?&]{8,30}$/;

export const passwordSchema = z.object({
  password: z
    .string()
    .min(8, { message: 'Le mot de passe doit être de 8 caractères minimum' })
    .max(30, { message: 'Le mot de passe doit être de 30 caractères maximum' })
    .regex(passwordRegex, {
      message: 'Le mot de passe doit contenir au moins 8 caractères, une majuscule, une minuscule, un chiffre et un caractère spécial',
    }),
});
```

```
/** 
 * Updates or create the user's password.
 */
@param {FormData} formData - The FormData object containing the user ID and new password.
@returns {Promise<{ ok: boolean, message?: string }>}
*/
export const updatePassword = async (formData: FormData) => {
  const session = await auth();

  const id = getFormDataStringValue(formData, 'id');
  const password = getFormDataStringValue(formData, 'password');

  if (!session || session.user.id !== id) {
    return {
      ok: false,
      message: 'Non connecté',
    };
  }

  async function hashPassword(password: string): Promise<string> {
    const saltRounds = 10;
    const hashedPassword = await bcrypt.hash(password, saltRounds);
    return hashedPassword;
  }

  try {
    // Validate the password using Zod schema
    passwordSchema.parse({ password });

    // Hash the password
    const hashedPassword = await hashPassword(password);

    // Update the user's password
    await updateUserService({
      id,
      data: {
        password: hashedPassword,
        hasPassword: true,
      },
    });

    return { ok: true, message: 'Mot de passe mis à jour avec succès' };
  } catch (error) {
    return handleError(error);
  }
}
```

Fonctionnalité Back - Modification de compte - mdp

```
export const updateUserService = async ({  
  id,  
  data,  
}: {  
  id: string;  
  data: Pick<  
    Prisma.UserUpdateInput,  
    'image' | 'description' | 'username' | 'hasPassword' | 'password'  
  >;  
}) => {  
  try {  
    return prisma.user.update({  
      where: {  
        id,  
      },  
      data,  
    });  
  } catch (error) {  
    console.error('updateUserService', error);  
    return null;  
  }  
};
```

```
export type UserUpdateInput = {  
  id?: StringFieldUpdateOperationsInput | string  
  name?: NullableStringFieldUpdateOperationsInput | string | null  
  username?: NullableStringFieldUpdateOperationsInput | string | null  
  password?: NullableStringFieldUpdateOperationsInput | string | null  
  hasPassword?: BoolFieldUpdateOperationsInput | boolean  
  email?: NullableStringFieldUpdateOperationsInput | string | null  
  emailVerified?: NullableDateTimeFieldUpdateOperationsInput | Date | string | null  
  image?: NullableStringFieldUpdateOperationsInput | string | null  
  description?: NullableStringFieldUpdateOperationsInput | string | null  
  role?: NullableStringFieldUpdateOperationsInput | string | null  
  createdAt?: DateTimeFieldUpdateOperationsInput | Date | string  
  updatedAt?: DateTimeFieldUpdateOperationsInput | Date | string  
  Session?: SessionUpdateManyWithoutUserNestedInput  
  Account?: AccountUpdateOneWithoutUserNestedInput  
  Authenticator?: AuthenticatorUpdateManyWithoutUserNestedInput  
  CommentsAuthored?: CommentsUpdateManyWithoutAuthorNestedInput  
  CommentsReceived?: CommentsUpdateManyWithoutRecipientNestedInput  
  Reports?: ReportsUpdateManyWithoutUserNestedInput  
  Reported?: ReportsUpdateManyWithoutReportedNestedInput  
  Trash?: TrashUpdateManyWithoutUserNestedInput  
  Ratings?: RatingsUpdateManyWithoutUserNestedInput  
  EventsCreated?: EventsUpdateManyWithoutUserNestedInput  
  EventsJoined?: UserEventsUpdateManyWithoutUserNestedInput  
  Tasks?: TasksUpdateManyWithoutUserNestedInput  
  MessagesSent?: MessageUpdateManyWithoutSenderNestedInput  
  UserConversation?: UserConversationUpdateManyWithoutUserNestedInput  
  MessageStatus?: MessageStatusUpdateManyWithoutUserNestedInput  
  UserInvitations?: UserInvitationsUpdateManyWithoutUserNestedInput  
}
```

Fonctionnalité Back - Modification de compte - mdp

Déconnexion



facundo.botta.dev@gmail.com

★★★★★

Bio

Il semble que tu n'aises pas encore ajouté ta bio ! Prends un moment pour compléter ton profil et accéder à toutes les fonctionnalités de l'application. Nous avons hâte de mieux te connaître !

Mon profil

Un mot clé, un contact, une tâche ?

Mes notes

Nouvelle tâche

Mes événements créés

Créer un événement

Mes événements à venir

Découvrir les événements

Mes avis

Aucun avis pour le moment

Tests - updateUserService

```
services > ts userServices.test.ts > ...
1 import { PrismaClient } from '@prisma/client';
2 import { mockDeep } from 'jest-mock-extended';
3 import { expect } from '@jest/globals';
4 import {
5   selectAllUsersService,
6   selectUserByEmailService,
7   selectUserByIdService,
8   updateUserService,
9 } from './userServices';
10 import {
11   getAllUsersServiceSchema,
12   getUserServiceSchema,
13 } from '@lib/zod/zodSchemas';
14
15 jest.mock('@lib/prisma', () => ({
16   __esModule: true,
17   default: mockDeep<PrismaClient>(),
18 }));
19
20 const prismaMock = jest.requireMock('@lib/prisma').default;
21
22 const mockUser = {
23   id: '1',
24   email: 'user@example.com',
25   username: 'testuser',
26   password: 'secretpassword',
27   image: null,
28   description: null,
29   _count: { Ratings: 2, EventsCreated: 1 },
30 };


```

```
describe('User Services', () => {
  afterEach(() => {
    jest.clearAllMocks();
  });
  describe('UpdateUserService', () => {
    it('should update a user successfully', async () => {
      const newData = {
        username: 'newUserName',
        description: 'new description',
      };
      prismaMock.user.update.mockResolvedValue({
        ...mockUser,
        ...newData,
      });
      const result = await updateUserService({
        id: '1',
        data: newData,
      });
      expect(prismaMock.user.update).toHaveBeenCalledWith({
        where: { id: '1' },
        data: newData,
      });
      expect(result).toEqual({
        ...mockUser,
        ...newData,
      });
    });
  });
  describe('selectUserByIdService', () => {...});
  describe('selectUserByEmailService', () => {...});
  describe('selectAllUsersService', () => {...});
});
```

Tests - Log des tests

```
● $ npm run test

> filrouge@0.1.0 test
> jest --verbose

  PASS  services/userServices.test.ts
    User Services
      UpdateUserService
        ✓ should update a user successfully (4 ms)
      selectUserIdService
        ✓ should select a user by id successfully (zod schema ok and password not returned) (3 ms)
        ✓ should return null when user is not found
      selectUserByEmailService
        ✓ should select a user by email successfully
        ✓ should return null when user is not found
      selectAllUsersService
        ✓ should select all users successfully (1 ms)
        ✓ should return null when there is an error

  PASS  lib/zod/handleError.test.ts
    handleError function
      ✓ should handle zod validation errors correctly (2 ms)
      ✓ should handle generic errors correctly (1 ms)

  PASS  utils/customError.test.ts
    CustomError class
      ✓ should create an instance with a message and no errors (1 ms)
      ✓ should create an instance with a message and errors (1 ms)
      ✓ should set the name property to the constructor name
      ✓ should inherit from the built-in Error class (1 ms)

Test Suites: 3 passed, 3 total
Tests:       13 passed, 13 total
Snapshots:   0 total
Time:        1.218 s
Ran all test suites.
```

Conclusion

À faire 

Bugs 

À venir 

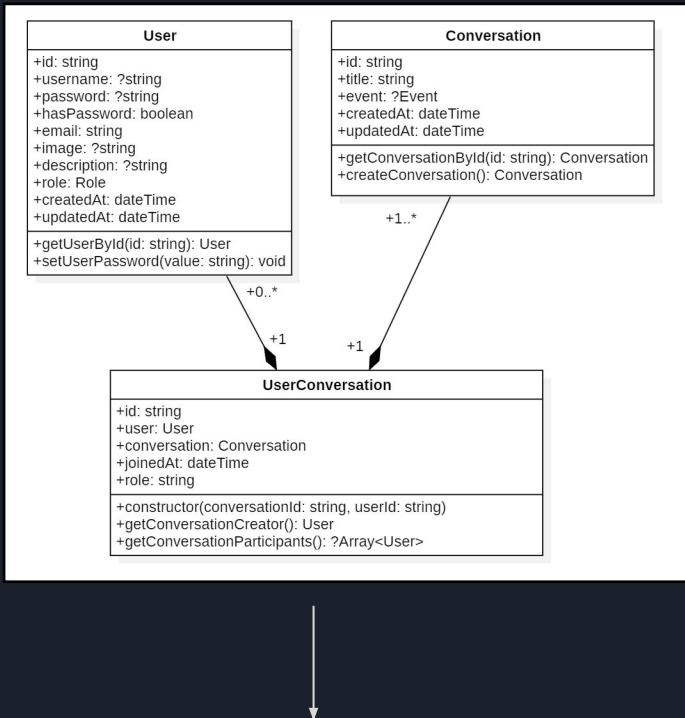
Liens :

- [Figma](#)
- [Mémoire](#)
- [Portfolio](#)
- [Site développé en freelance](#)



Merci ! 😊

Bonus - Diagramme de classe de exemple



Approche fonctionnelle - Services

```
export const getConversationParticipantsService = async (
  conversationId: string
) => {
  try {
    const participants = await prisma.userConversation.findMany({
      where: { conversationId },
      select: {
        userId: true,
      },
    });
    return participants;
  } catch (error) {
    console.error('getConversationService: error', error);
    throw new Error('Service error: getConversationService');
  }
};
```

```
const creator = new UserConversation(<<conversationId>>, <<userId>>).getConversationCreator();
```

Bonus - handleError

```
import CustomError from '@/utils/customError';
import { ZodError } from 'zod';

// This interface is used by the handleError function to type the error object
export interface ValidationError {
  message: string;
  path: (string | number)[];
}

/**
 * Handles errors and converts them into a standardized `CustomError` instance.
 * Supports both generic errors and `ZodError` validation errors.
 */
/**
 * @param {unknown} error - The error to handle. This can be any type of error, including validation errors from Zod.
 * @returns {CustomError<ValidationError[]>} A structured `CustomError` instance containing:
 *   - `message`: A general error message, which may describe validation errors or a generic issue.
 *   - `errors`: An array of validation errors if the error is a `ZodError` ; otherwise, it is undefined.
 */
/**
 * @example
 * try {
 *   schema.parse(data);
 * } catch (error) {
 *   return handleError(error);
 * }
 */

export function handleError(error: unknown): CustomError<ValidationError[]> {
  if (error instanceof ZodError) {
    // Extract validation errors from Zod
    const validationErrors: ValidationError[] = error.errors.map(
      ({ message, path }) => ({
        message,
        path,
      })
    );

    // Create an instance of CustomError with the validation error details
    const message = validationErrors.map((e) => e.message).join(', ');
    return new CustomError<ValidationError[]>(message, validationErrors);
  }

  // For other errors, generate a CustomError with a generic message
  return new CustomError<ValidationError[]>('Une erreur est survenue');
}
```

C:\Users\User\Docume

```
▼ {errorHandled: CustomError: Le mot de passe doit être de 8 caractères minimum, une majuscule, une minuscule, une minuscule, un chiffre et un caractère spécial}
  ▼ errorHandled: CustomError: Le mot de passe doit être de 8 caractères minimum, une majuscule, une minuscule, un chiffre et un caractère spécial
    ▼ errors: Array(2)
      ▼ 0:
        ▶ message: "Le mot de passe doit être de 8 caractères minimum"
        ▶ path: ['password']
        ▶ [[Prototype]]: Object
      ▼ 1:
        ▶ message: "8 caractères, une majuscule, une minuscule, un chiffre et un caractère spécial"
        ▶ path: ['password']
        ▶ [[Prototype]]: Object
        ▶ length: 2
        ▶ [[Prototype]]: Array(0)
        ▶ name: "CustomError"
        ▶ ok: false
        ▶ message: "Le mot de passe doit être de 8 caractères minimum, 8 caractères, une majuscule, une minuscule, une minuscule, un chiffre et un caractère spécial"
        ▶ stack: "CustomError: Le mot de passe doit être de 8 caractères minimum, 8 caractères, une majuscule, une minuscule, une minuscule, un chiffre et un caractère spécial"
        ▶ [[Prototype]]: Error
        ▶ [[Prototype]]: Object
```