

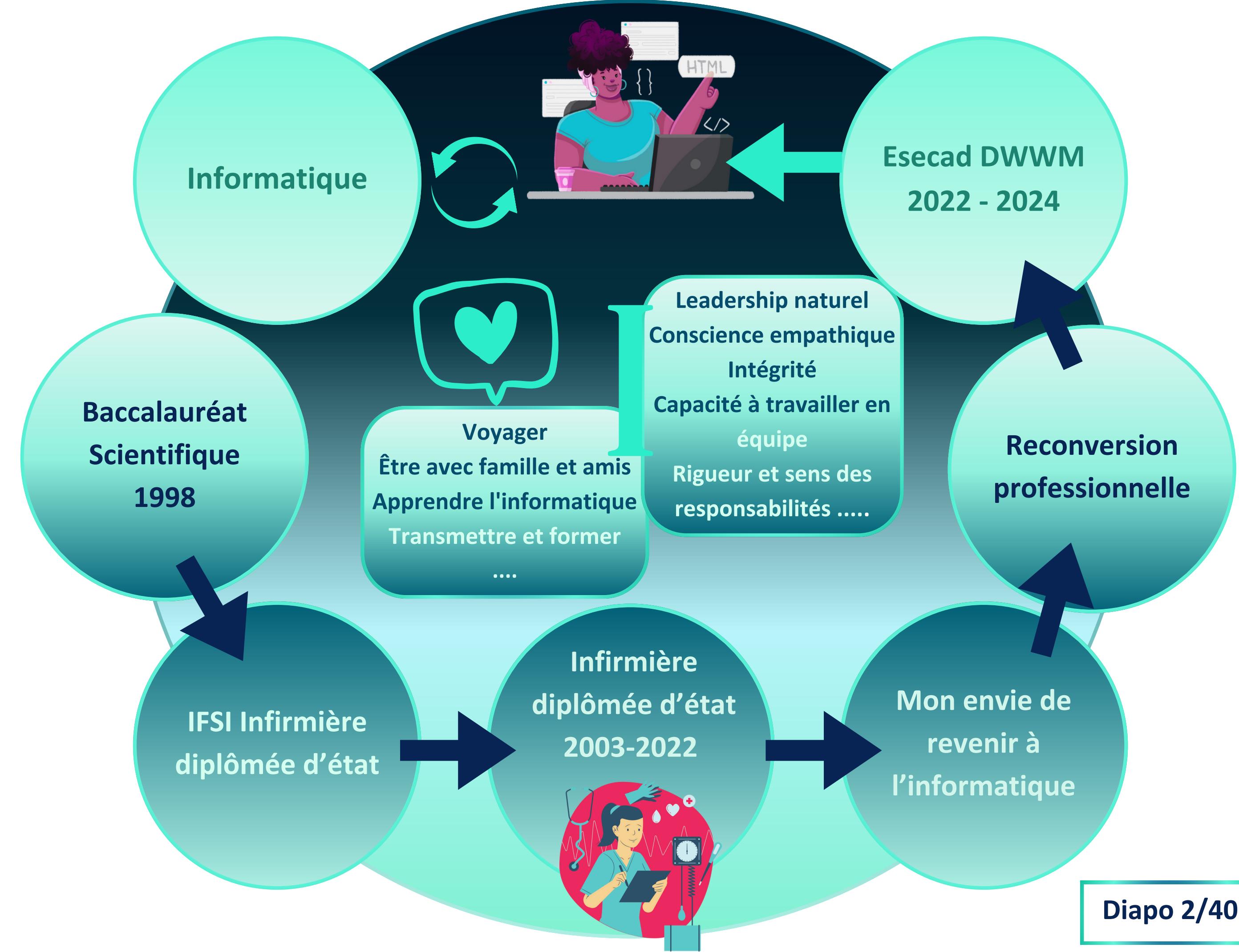


TITRE PROFESSIONNEL DÉVELOPPEUR WEB ET WEB MOBILE

SUPPORT DE PRÉSENTATION

Latifa ATTAR - studee

QUI SUIS JE ?



PRÉSENTATION DU PROJET



studee,
par les étudiants,
pour les étudiants.

Sky is the limit!

Seul on va plus vite,
ensemble on va plus
loin.

Ton stage sera ce
que tu veux en faire.

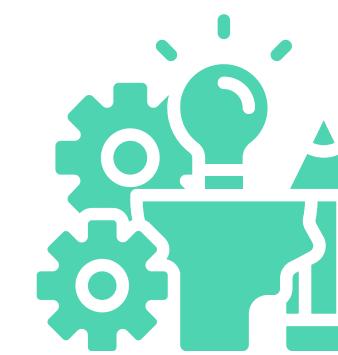
LE PROJET STUDEE



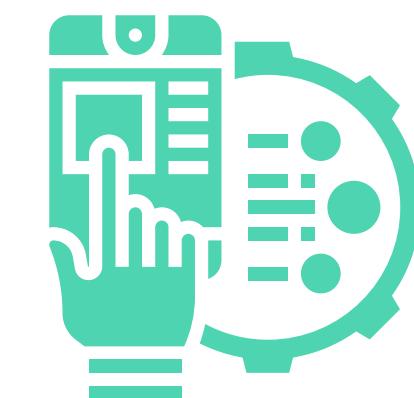
SOMMAIRE



PRÉSENTATION DU PROJET



CONCEPTION ET DÉVELOPPEMENT



INTERFACE UTILISATEUR



SÉCURITÉ



TEST UNITAIRES



RECHERCHE SUR SITE ANGLOPHONE



SYNTHESE ET CONCLUSION

Thank You

REMERCLEMENT

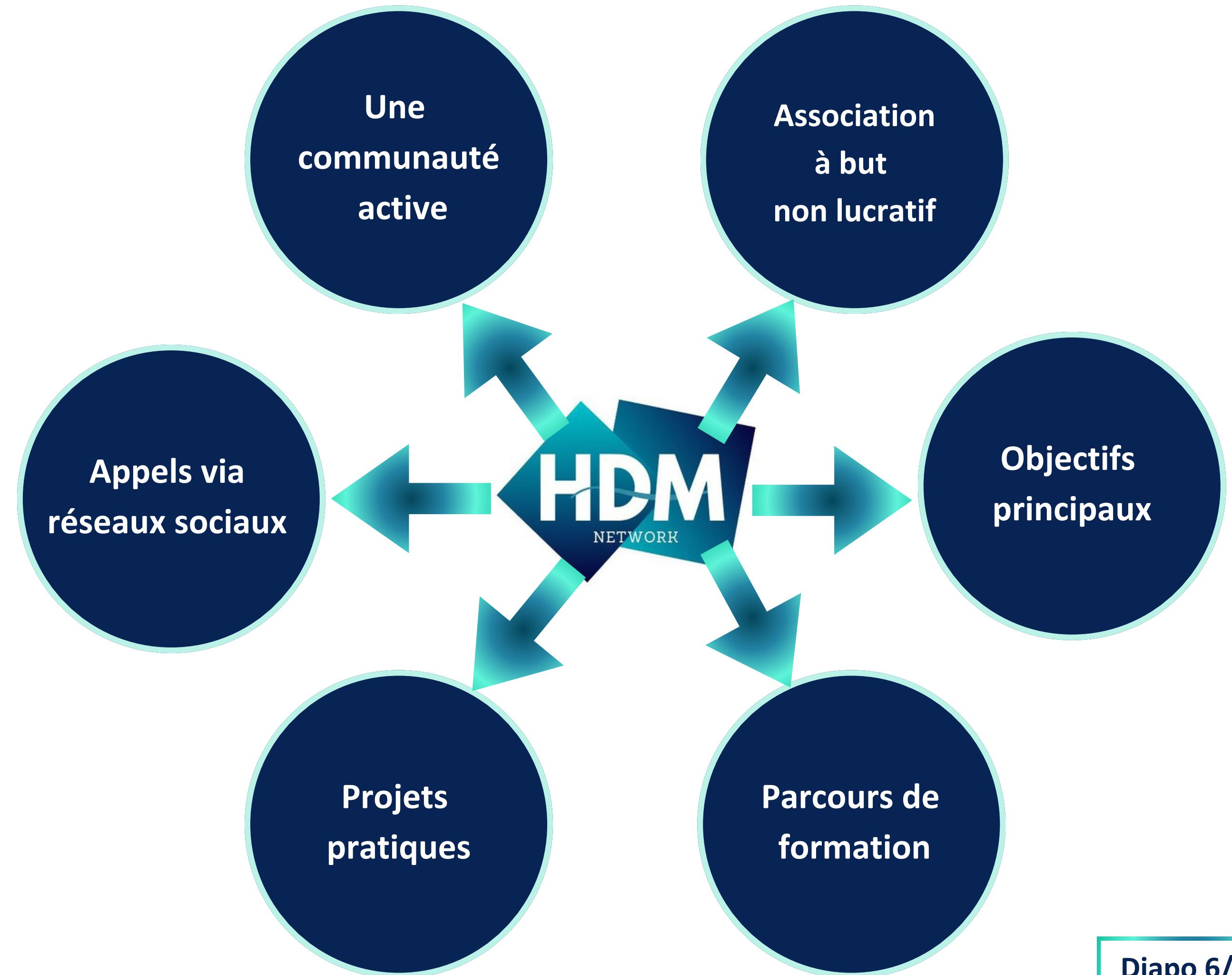
PRÉSENTATION DE L'ENTREPRISE



PHILOSOPHIE HDM



PRÉSENTATION DE L'ENTREPRISE



PRÉSENTATION DE L'ENTREPRISE



David VAN GOIDTSNOVEN

Quentin MOUSSET

Ressource Humaines

Recrutement

Placement

Business developpers/
Commerciaux (B2B)

SEO/Journalistes

Graphistes

WEB Developers

CONTEXTE DU PROJET



CAHIER DES CHARGES



CONTEXTE DU PROJET



ENVIRONNEMENT HUMAIN COLLABORATION ET METHODOLOGIE DE TRAVAIL

Attar Latifa
DWWM 2022-2024



- ◆ Outils de communication
- ◆ Résolution Collective des Problèmes



- ◆ Sprint planning, Daily Scrum, Sprint Review, Sprint Retro
 - ◆ Push quotidien du code développé
 - ◆ Reporting quotidien sur le Dashboard HDM

CONTEXTE DU PROJET



ENVIRONNEMENT HUMAIN LA TEAM STUDEE

Attar Latifa

DWWM 2022-2024




Développement de l'interface étudiant
Mme Attar Latifa


Développement de l'interface entreprise
Mr Sacré Alexandre


Développement de l'interface chat
Bagayoko Yankhoba

CONTEXTE DU PROJET

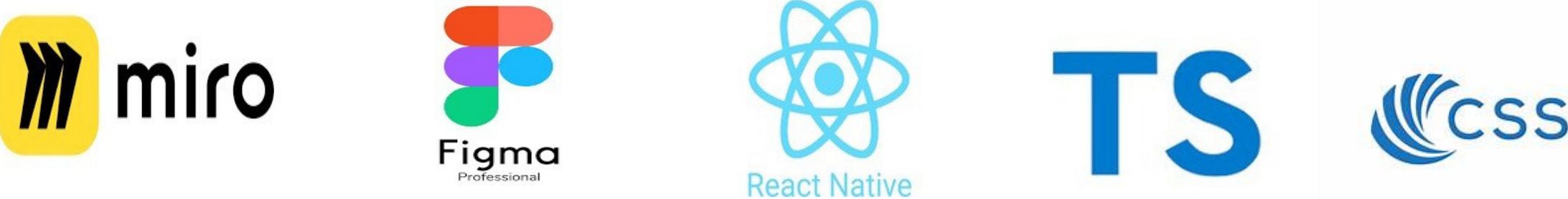


ENVIRONNEMENT TECHNIQUE

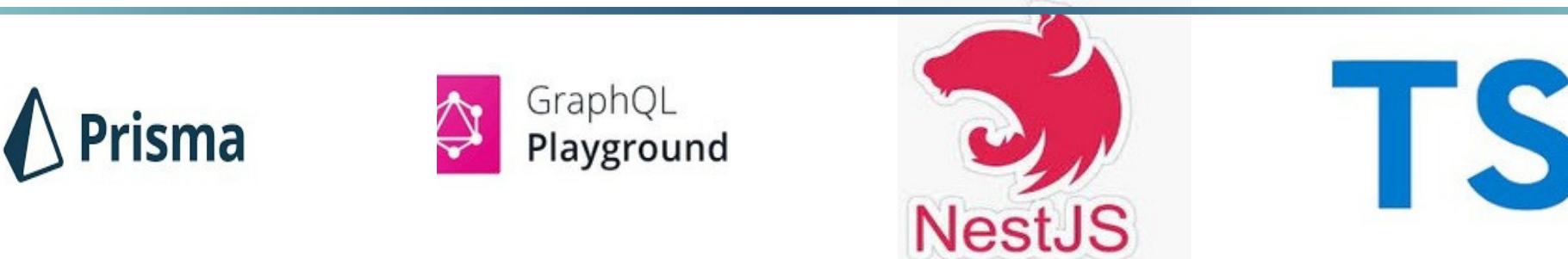
Environnement de Développement, Installation et Configuration



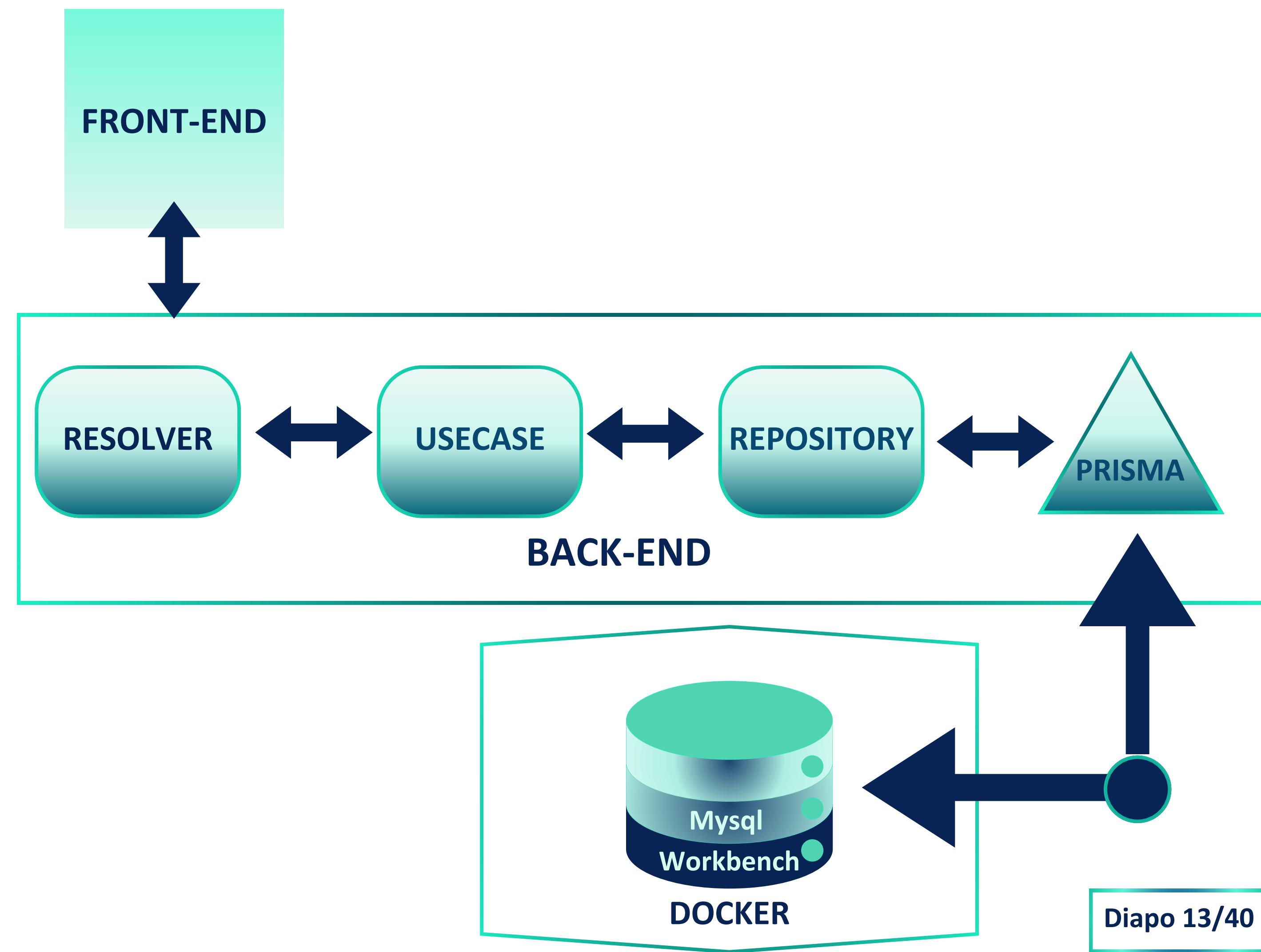
Partie Front-end



Partie Back-end



ARCHITECTURE DE L'APPLICATION



CONCEPTION ET DÉVELOPPEMENT DES COMPOSANTS FRONT-END



MAQUETTE

Attar Latifa

DWWM 2022-2024

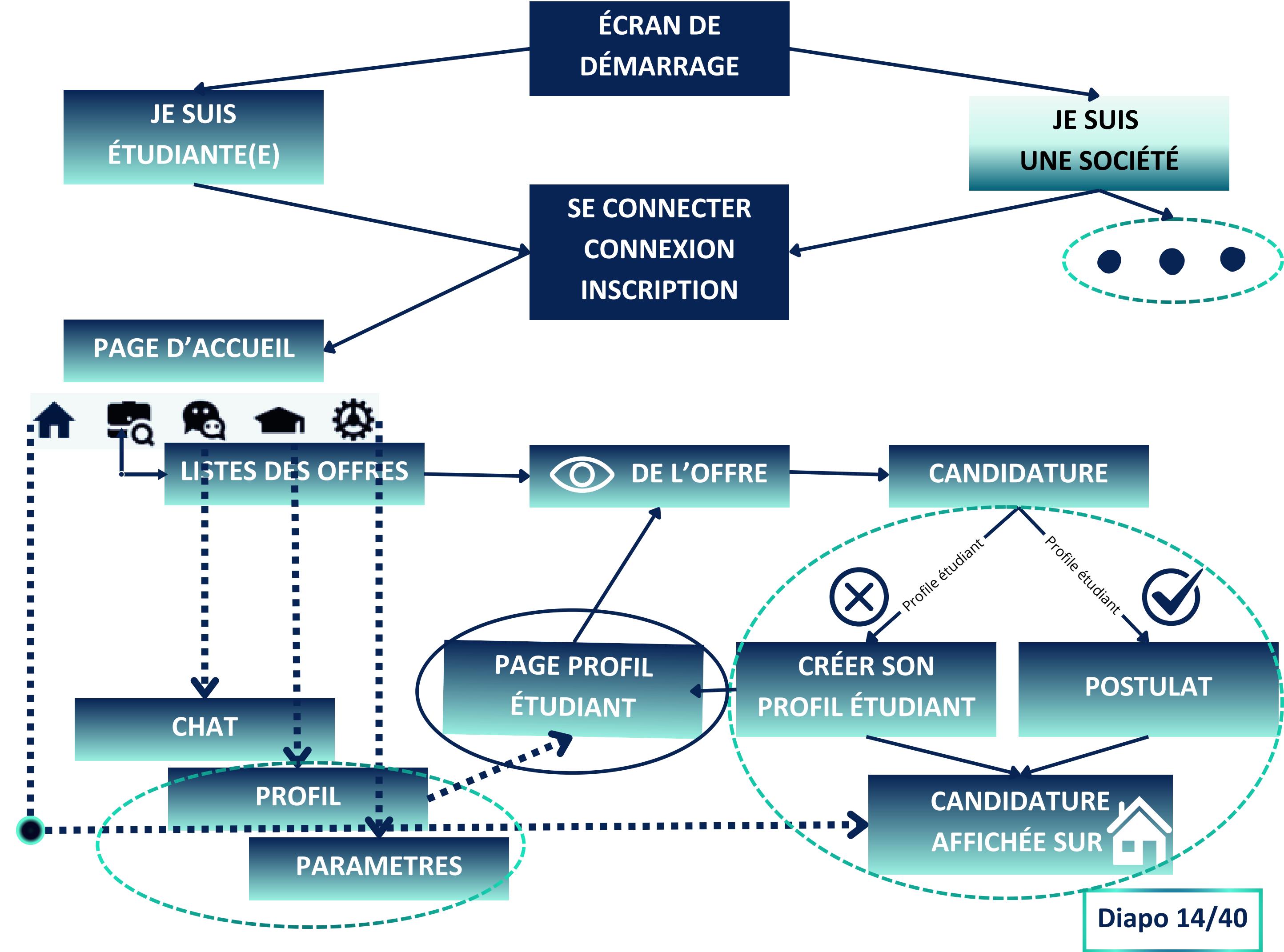
Diapo 14/40

CONCEPTION ET DÉVELOPPEMENT DES COMPOSANTS



Attar Latifa

DWWM 2022-2024



CONCEPTION ET DÉVELOPPEMENT DES COMPOSANTS FRONT-END



MAQUETTE STUDENTPROFILE

Attar Latifa
DWWM 2022-2024

Maquette Studee: StudentProfile refaite par mes soins

Diapo 15/40

CONCEPTION ET DÉVELOPPEMENT DES COMPOSANTS FRONT- END



STRUCTURE DU PROJET

Attar Latifa
DWWM 2022-2024

Github Récupéré
**hdmnetwork/studee-mobile-
app repository**

COMPOSANT ENFANT

COMPOSANT PARENT

Project Structure:

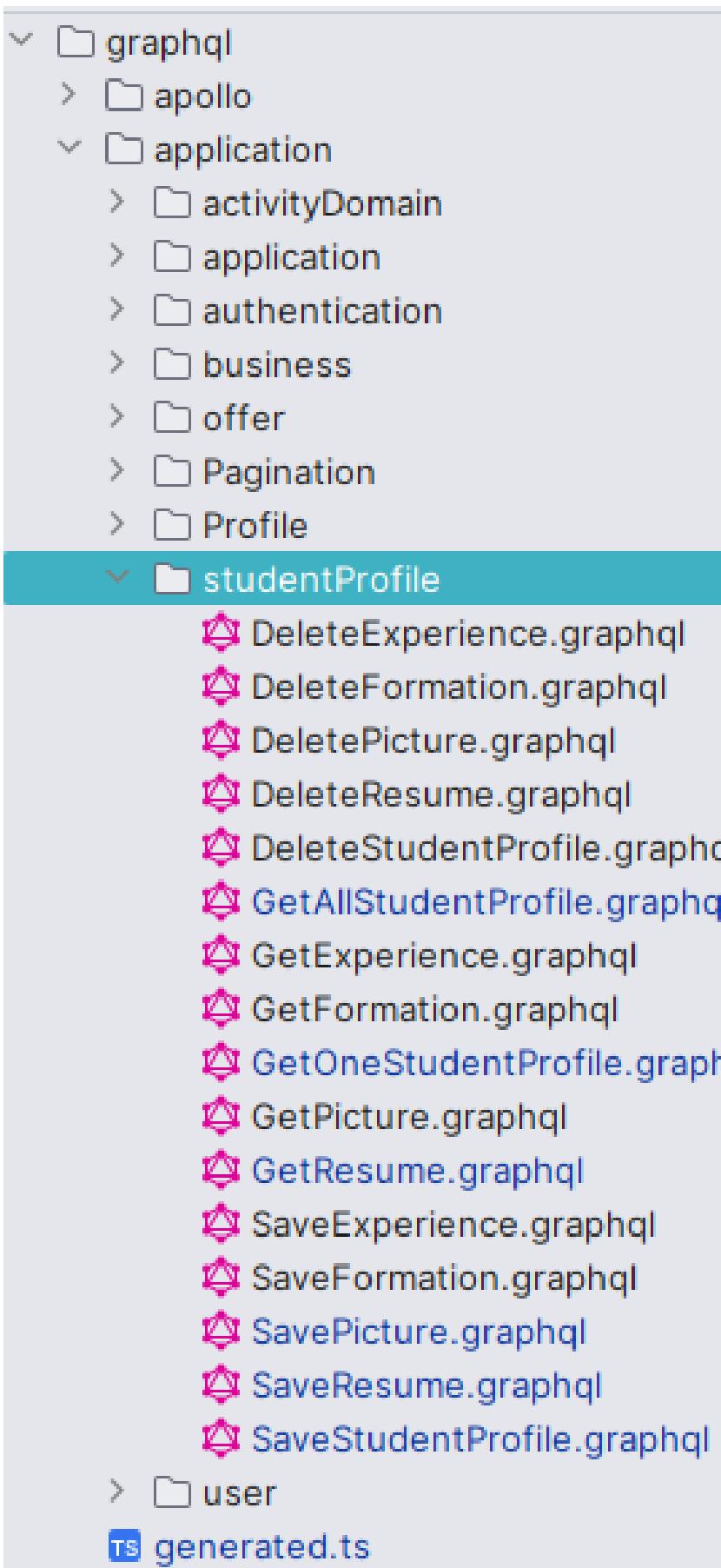
- studee-mobile-app C:\Users\...
 - External Libraries
 - Scratches and Consoles
 - node_modules library root
 - src
 - components
 - graphql
 - helpers
 - hooks
 - .eslintignore
 - .eslintrc.json
 - .gitignore
 - .yarnrc
 - .yarnrc.yml
 - app.json
 - babel.config.js
 - codegen.yaml
 - graphql.config.yaml
 - jestSetupFile.js
 - metro.config.js
 - package.json
 - README.md
 - schema.graphql
 - tsconfig.json
 - yarn.lock
 - assets
 - node_modules library root
 - src
 - components
 - providers
 - screens
 - online
 - business
 - student
 - student
 - app
 - business
 - online
 - business
 - student
 - Business
 - Offer
 - assets
 - node_modules library root
 - src
 - StudentProfile
 - [id].tsx
 - _layout.tsx
 - Chat.tsx
 - HomeOffers.tsx
 - index.tsx
 - MarketPlace.tsx
 - Profile.tsx
 - Settings.tsx
 - StudentProfile.tsx
 - _layout.tsx
 - student
 - _layout.tsx
 - Home.tsx
 - index.tsx
 - assets
 - node_modules library root
 - src
 - components
 - screens
 - online
 - business
 - student
 - student

CONCEPTION ET DÉVELOPPEMENT DES COMPOSANTS FRONT-END

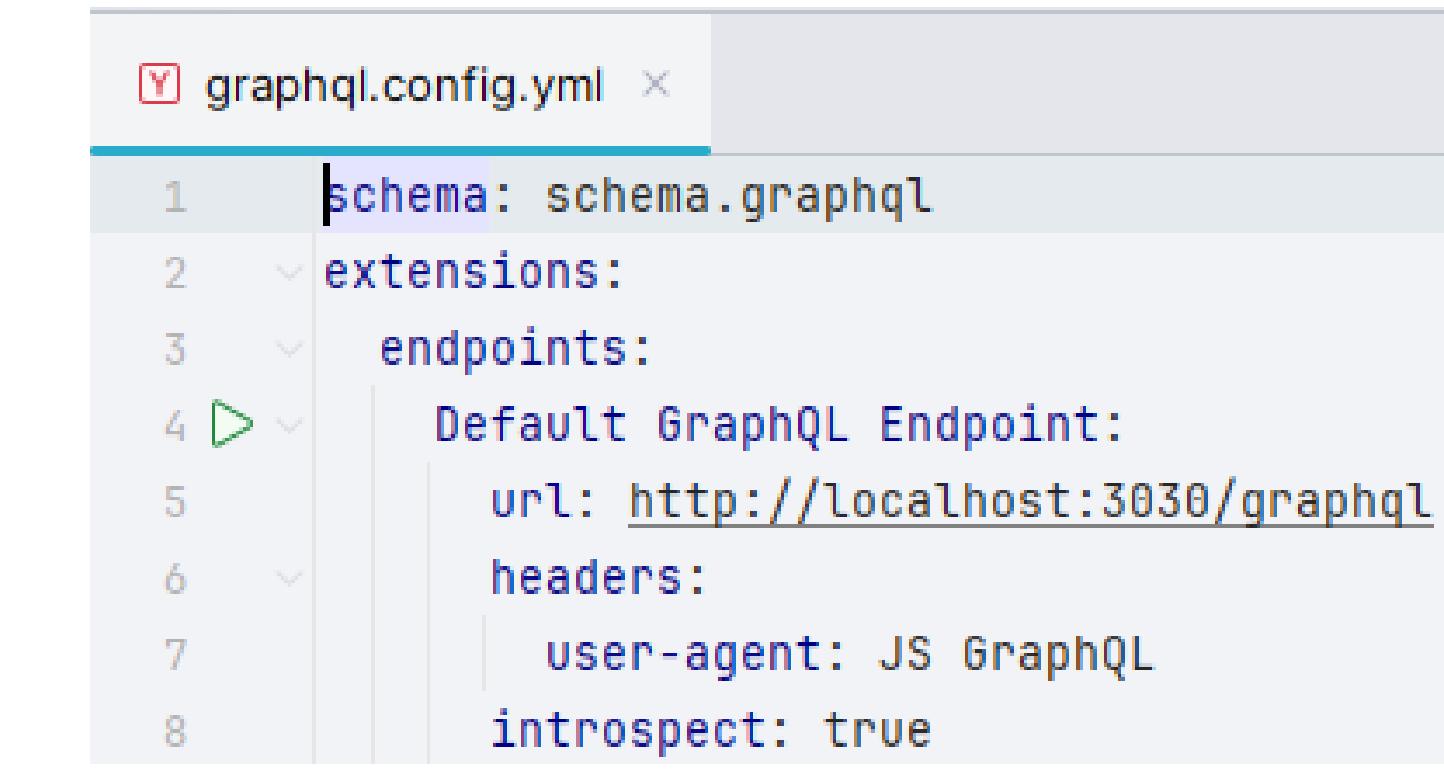


Intégrer les requêtes GraphQL

- Intégrer les requêtes dans le front-end.
exemple: `src/graphql/StudentProfile/SaveStudentProfile.graphql`



- Générer le schéma GraphQL avec `graphql.config.yml`.



- Utiliser `yarn generate` pour créer les types TypeScript.



- Intégrer les requêtes/mutations dans les composants front-end.

CONCEPTION ET DÉVELOPPEMENT DES COMPOSANTS FRONT- END



STUDENTPROFILESCREEN

StudentProfileScreen.tsx

```
import {  
  useGetLoggedUserLazyQuery, useGetOneStudentProfileLazyQuery,  
  useSaveStudentProfileMutation,  
} from '...../graphql';  
  
const StudentProfileScreen = () => { Show usages ± Latifa *  
  const { theme } = useTheme();  
  const { route } = useProfileType();  
  const styles = globalBusinessPageStylesMobile;  
  const { id, returnToOffer } = useLocalSearchParams<{ id: string, returnToOffer: string }>();  
  const studentProfileIdFromUrl = parseInt(id!, 10);  
  
  /** Mutation gql pour la création ou modification du studentProfile */  
  const [saveStudentProfile] = useSaveStudentProfileMutation();  
  
  const handleValidateResumeRef = useRef<() => Promise<void>>(async () => {});  
  const handleValidatePictureRef = useRef<() => Promise<void>>(async () => {});  
  const handleValidateExperienceRef = useRef<() => Promise<void>>(async () => {});  
  const handleValidateFormationRef = useRef<() => Promise<void>>(async () => {});  
  const handleValidateActivityDomainRef = useRef<() => Promise<void>>(async () => {});
```

CONCEPTION ET DÉVELOPPEMENT DES COMPOSANTS FRONT- END



STUDENTPROFILESCREEN

```
return (
  <ScrollView>
    {/* Affichage de la photo */}
    <PictureComponent
      handleValidateRef={handleValidateResumeRef}
      studentProfileId={studentProfileId}
      pictureStudent={formDataStudentProfile.picture}
    />
```

JSX

Composant
Enfant

```
<View style={[styles.buttonContainer, { zIndex: -1 }]}>
  <UiButton
    disabled={isButtonDisabled}
    style={styles.buttons}
    onPress={() => handleSave()}
    bgColor={theme.button}
    mt={8}
  >
    <View style={styles.TextContainer}>
      <UiText
        color={theme.lightText}
        fontSize={20}
        type="QuicksandBold"
        style={styles.textInButton}
      >
        Valider
      </UiText>
    </View>
    <View style={styles.arrowContainer} />
  </UiButton>
</View>
```

Bouton Validation

```
const globalBusinessPageStylesMobile = StyleSheet.create({
  buttonContainer: {
    marginTop: 25,
    marginBottom: 40,
    height: 54,
  },
  buttons: {
    width: 261,
    display: 'flex',
    flexDirection: 'row',
    justifyContent: 'center',
    alignItems: 'center',
    borderRadius: 10,
    maxHeight: 54,
  },
  TextContainer: {
    width: 220,
    display: 'flex',
    flexDirection: 'row',
    justifyContent: 'center',
    height: 54,
  },
  arrowContainer: {
    width: 42,
    display: 'flex',
    flexDirection: 'row',
    justifyContent: 'center',
  },
})
```

CSS

CONCEPTION ET DÉVELOPPEMENT DES COMPOSANTS BACK-END



STRUCTURE DU PROJET

Attar Latifa
DWWM 2022-2024

Github Récupéré
hdmnetwork/studee-api
repository

```
prisma
  migrations
  schema.prisma
  seed.ts
src
  Api
    ActivityDomain
    Application
    Business
    File
    Message
    Offer
  Profile
    Entity
    Repository
    Resolver
    Service
    UseCase
    ProfileModule.ts
  User
    ApiModule.ts
    enums.ts
    types.ts
  Core
  Workers
    filelogger.ts
    index.d.ts
    main.ts
  .env
  .gitignore
  nest-cli.json
  package.json
  README.md
  schema.gql
  tsconfig.build.json
  tsconfig.json
```

```
Profile
  Entity
    BusinessProfile.ts
    Experiences.ts
    Formations.ts
    Picture.ts
    Profile.ts
    Resume.ts
    StudentProfile.ts
  Repository
    ExperiencesRepository.ts
    FormationRepository.ts
    PictureRepository.ts
    ProfileRepository.ts
    ResumeRepository.ts
    StudentProfileRepository.ts
  Resolver
    ProfileResolver.ts
    StudentProfileResolver.ts
  Service
  UseCase
    Experience
    Formation
    Picture
    Profile
    Resume
  StudentProfile
    DeleteStudentProfileUseCase.ts
    GetAllStudentProfileUseCase.ts
    GetOneStudentProfileUseCase.ts
    SaveStudentProfileUseCase.ts
    CreateBusinessProfileDto.ts
    SaveExperienceDto.ts
    SaveFormationDto.ts
    SavePictureResumeDto.ts
    SaveResumeDto.ts
    SaveStudentProfileDto.ts
    StudentProfileUseCaseFactory.ts
    ProfileModule.ts
```

CONCEPTION ET DÉVELOPPEMENT DES COMPOSANTS BACK-END



CRÉATION DE LA BASE DE DONNÉES

Github récupéré hdmnetwork/msql5.7repository

```
≡ .env x  
1  COMPOSE_PROJECT_NAME=mysql57  
2  
3  ## MYSQL ##  
4  MYSQL_DATABASE=studee-db  
5  MYSQL_USER=mysql57  
6  MYSQL_PASSWORD=studee_mdp  
7  MYSQL_ROOT_PASSWORD=studee_mdp
```

...webstorm/Studee/mysql57/.env

```
M+ INSTALL.md x  
1  # How to install Mysql on your local machine  
2  
3  1. Update .env file by adding user & root DB passwords  
4  2. Install docker on your machine  
5  3. Execute `docker network create mysql57`  
6  4. Execute `docker volume create mysql57`  
7  5. Execute `docker compose -f stack.local.yml up -d`  
8  6. Execute `docker ps` and check if the container is running  
9  7. Enjoy
```

Terminal : → ...webstorm/Studee/mysql57/

```
≡ .env x  
1  APP_ENV=local  
2  JWT_SECRET=62y$10$4zxKLip851WEYeVDDpsckeCRw6.t8WT7bXu0KCKW6dAhTC9988  
3  DATABASE_URL="mysql://${MYSQL_USER}:${MYSQL_PASSWORD}@localhost:3306/${MYSQL_DATABASE}"  
4  API_PORT=3030  
5  API_BASE_URL=http://localhost:3030  
6  API_REST_URL=http://192.168.65.5:3030  
7  API_BASE_GRAPHQL_URL=http://localhost:3030/graphql  
8  MOBAPP_LOCAL_STORAGE_KEYNAME=studee  
9  WORKERS_PORT=3032  
10  GRAPHQL_PLAYGROUND_ENABLED=true  
11  CDN_ACCESS_KEY_ID=SCWKTW7PF6PED59TKA4A  
12  CDN_ACCESS_KEY=432f22f6-8173-4bbe-8ae2-274edeb04989  
13  CDN_PUBLIC_URL=https://hdmnetwork-cdn.s3.fr-par.s3.cloud.studee  
14  MYSQL_DATABASE=studee_db  
15  MYSQL_USER=root  
16  MYSQL_PASSWORD=studee_mdp  
17  MYSQL_ROOT_PASSWORD=studee_mdp
```

...webstorm/Studee/studee-api/src/.env



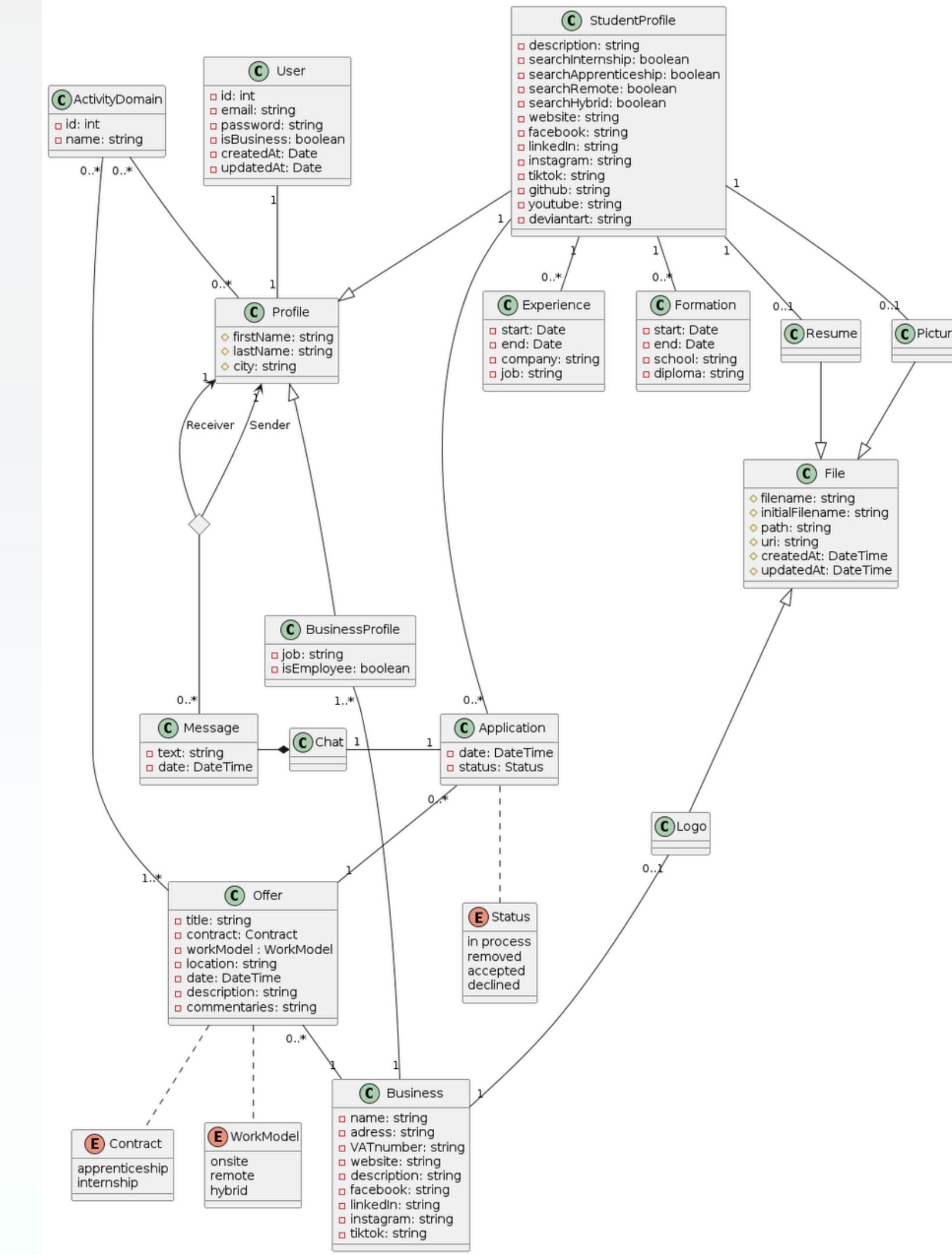
Aller sur Mysql
workbench ->
MySQL Connection
->Name : Studee-db
->
Paramètres : port :
3306

CONCEPTION ET DÉVELOPPEMENT DES COMPOSANTS BACK-END



MODELE CONCEPTUEL DE DONNÉES STUDEE

Attar Latifa
DWWM 2022-2024



CONCEPTION ET DÉVELOPPEMENT DES COMPOSANTS BACK-END



INTÉRATION AVEC LA BASE DE DONNÉES CRÉÉES

```
model Profile {
    id      Int      @id @default(autoincrement())
    firstName String  @db.VarChar(50)
    lastName String  @db.VarChar(50)
    city    String  @db.VarChar(50)
    createdAt DateTime @default(now())
    updatedAt DateTime @default(now()) @updatedAt

    user          User      @relation(fields: [id], references: [id], onDelete: Cascade)
    // chaque instance de Profile est associée à une instance correspondante de User via l'ID.
    // Si user Supprimer Son Profile est supprimer aussi
    receivedMessages Message[]    @relation("Receiver") // Relation inverse pour le champ receiver
    sentMessages   Message[]    @relation("Sender") // Relation inverse pour le champ sender
    studentProfile StudentProfile[] // un Profile peut ou non être lié à plusieurs instance de StudentProfile.
    businessProfile BusinessProfile? // un Profile peut être lié ou non à une instance de BuisnessProfile.
    activityDomains ActivityDomain[] // Relation plusieurs à plusieurs avec ActivityDomain
}
```

...\\WebstormProjects\\Studee\\stud
ee-api\\prisma\\schema.prisma

npx prisma migrate dev : Génère et applique les migrations basées sur le schéma Prisma

```
-- CreateTable
CREATE TABLE `Profile` (
    `id` INTEGER NOT NULL AUTO_INCREMENT,
    `firstName` VARCHAR(50) NOT NULL,
    `lastName` VARCHAR(50) NOT NULL,
    `city` VARCHAR(50) NOT NULL,
    `createdAt` DATETIME(3) NOT NULL DEFAULT CURRENT_TIMESTAMP(3),
    `updatedAt` DATETIME(3) NOT NULL DEFAULT CURRENT_TIMESTAMP(3),

    PRIMARY KEY (`id`)
) DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

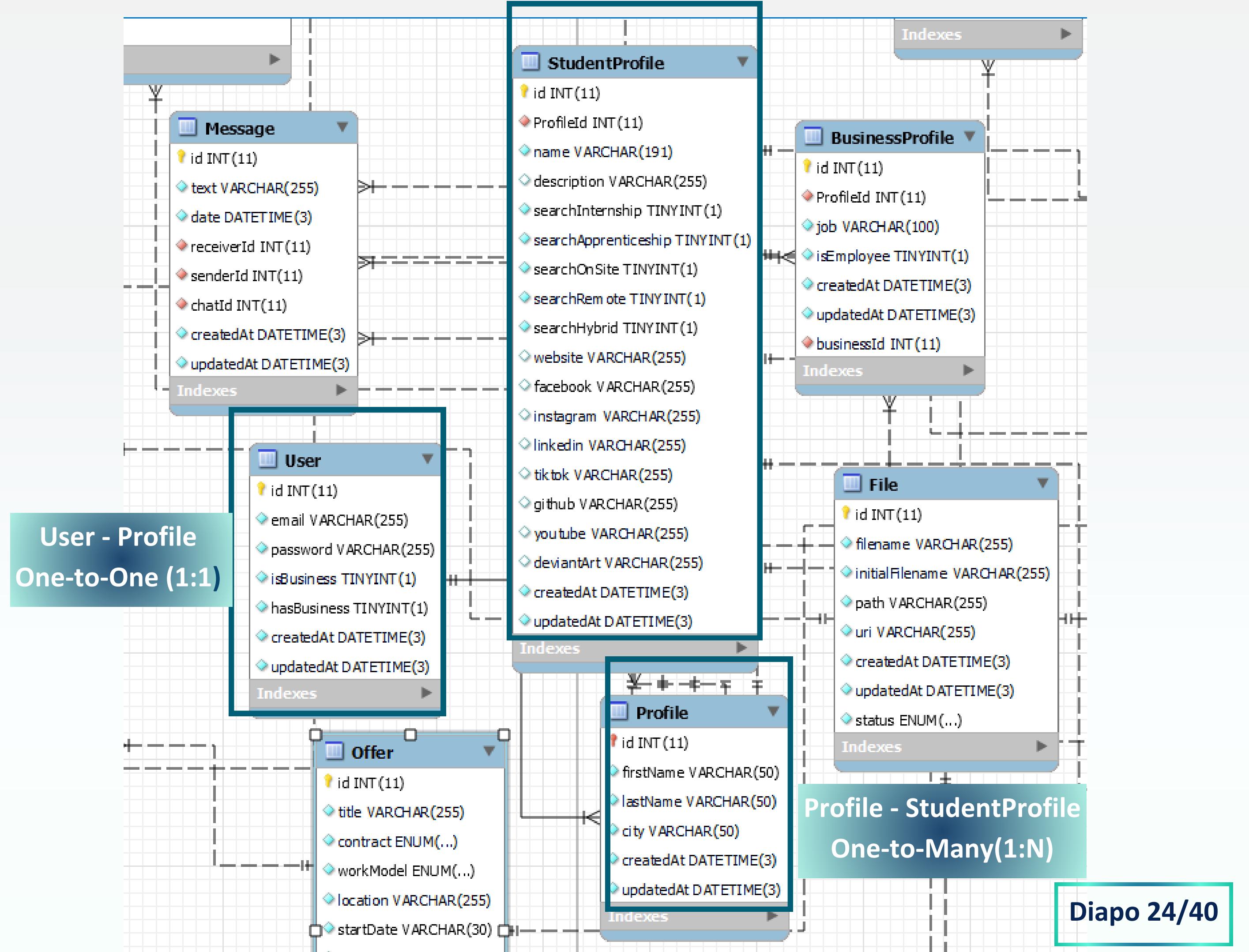
npx prisma generate : Génère le Prisma Client pour interagir avec la base de données..

CONCEPTION ET DÉVELOPPEMENT DES COMPOSANTS BACK-END



BASE DE DONNÉES STUDEE_DB

Attar Latifa
DWWM 2022-2024



CONCEPTION ET DÉVELOPPEMENT DES COMPOSANTS BACK-END



Modèle de Données Entity et DTO - **StudentProfile**

Attar Latifa
DWWM 2022-2024

```

@ObjectType() Show usages ± Latifa +1* ✓
export default class StudentProfile { ± Latifa
  @Field() nullable: true
  id: number;

  @Field()
  name: string;

  @Field({ nullable: true })
  description: string | null;

  @Field()
  searchInternship: boolean;

  @Field()
  searchApprenticeship: boolean;

  @Field()
  searchOnSite: boolean;

  @Field()
  searchRemote: boolean;

  @Field()
  searchHybrid: boolean;

  @Field({ nullable: true })
  website: string | null;

  @Field({ nullable: true })
  facebook: string | null;

  @Field({ nullable: true })
  linkedin: string | null;
}

10  export default class StudentProfile { ± Latifa
47 @Field({ nullable: true }) tiktok: string | null;
48
49
50 @Field({ nullable: true }) github: string | null;
51
52
53 @Field({ nullable: true }) youtube: string | null;
54
55
56 @Field({ nullable: true }) deviantArt: string | null;
57
58
59 @Field(() => [Experiences]) experiences?: Experiences[];
60
61
62 @Field(() => [Formations]) formations?: Formations[];
63
64
65 @Field(() => Picture, { nullable: true }) picture?: Picture | null;
66
67
68 @Field(() => Resume, { nullable: true }) resume?: Resume | null;
69
70
71 @Field(() => [Application]) application?: Application[];
72
73
74 @Field({ nullable: true }) ProfileId: number;
75
76
77 context?: ContextualGraphqlRequest;
78
79 }

```

StudentProfile.ts

```

!InputType() Show usages ± Latifa * ✓
export default class SaveStudentProfileDto { ± Latifa
  @Field(() => Int, { nullable: true }) id?: number;

  @Field()
  name: string;

  @Field({ nullable: true }) description: string | null;

  @Field({ nullable: true }) searchInternship: boolean | null;

  @Field({ nullable: true }) searchApprenticeship: boolean | null;

  @Field({ nullable: true }) searchOnSite: boolean | null;

  @Field({ nullable: true }) searchRemote: boolean | null;

  @Field({ nullable: true }) searchHybrid: boolean | null;

  @Field({ nullable: true }) website: string | null;

  @Field({ nullable: true }) facebook: string | null;

  @Field({ nullable: true }) instagram: string | null;
}

8  export default class SaveStudentProfileDto { ± Latifa *
42 @Field({ nullable: true }) linkedin: string | null;
43
44
45 @Field({ nullable: true }) tiktok: string | null;
46
47
48 @Field({ nullable: true }) github: string | null;
49
50
51 @Field({ nullable: true }) youtube: string | null;
52
53
54 @Field({ nullable: true }) deviantArt: string | null;
55
56
57 @Field(() => [SaveExperienceDto]) experiences?: SaveExperienceDto[];
58
59
60 @Field(() => [SaveFormationDto]) formations?: SaveFormationDto[];
61
62
63 @Field(() => [SaveApplicationDto]) application?: SaveApplicationDto[];
64
65
66 @Field(() => SaveResumePictureDto, { nullable: true }) picture?: SaveResumePictureDto | null;
67
68
69 @Field(() => SaveResumePictureDto, { nullable: true }) resume?: SaveResumePictureDto | null;
70
71
72
73 @Field({ nullable: true }) ProfileId: number;
74
75 }

```

SaveStudentProfileDto.ts

```

export type ContextualGraphqlRequest = {
  userId?: number;
  email?: string;
  roles?: string[];
  password?: string;
  isBusiness?: boolean;
  args?: Record<string, any>;
  password?: string;
  profil: {
    ...
  };
};

```

ContextualGraphqlRequest

Diapo 25/40

CONCEPTION ET DÉVELOPPEMENT DES COMPOSANTS BACK-END



StudentProfileRepository

```
export default class StudentProfileRepository { * Latifa *
  constructor(private readonly prisma: PrismaService) {} no usages * Latifa

  async delete(id: number) { Show usages * Latifa
    return this.prisma.studentProfile.delete({
      where: { id }
    });
  }

  async findById(id: number): Promise<StudentProfile> { Show usages * Latifa
    return this.prisma.studentProfile.findUnique({
      where: { id }
    });
  }

  async findAllByProfileId(ProfileId: number) { Show usages * Latifa
    return this.prisma.studentProfile.findMany({
      where: { ProfileId },
    });
  }

  async save(data: Prisma.XOR<Prisma.StudentProfileCreateInput, Prisma.StudentProfileUncheckedCreateInput> | Prisma.XOR<Prisma.StudentProfileUpdateInput, Prisma.StudentProfileUncheckedUpdateInput>) {
    if (!data.id) {
      return this.prisma.studentProfile.create({ data: data as Prisma.XOR<Prisma.StudentProfileCreateInput, Prisma.StudentProfileUncheckedCreateInput> });
    }

    return this.prisma.studentProfile.update({
      where: {
        id: data.id as number,
      },
      data: data as Prisma.XOR<Prisma.StudentProfileUpdateInput, Prisma.StudentProfileUncheckedUpdateInput>,
    });
  }
}
```

StudentProfileRepository.ts

CONCEPTION ET DÉVELOPPEMENT DES COMPOSANTS BACK-END



SaveStudentProfileUseCase

Attar Latifa

DWWM 2022-2024

```
@Injectable() Show usages  ± Latifa *
export default class SaveStudentProfileUseCase implements UseCase<Promise<StudentProfile>, [dto: SaveStudentProfileDto]> {  ±
  constructor(  no usages  ± Latifa
    private readonly studentProfileRepository: StudentProfileRepository,
    private readonly eventEmitter: RequestEventEmitter,
  ) {}

  async handle(context: ContextualGraphqlRequest, dto: SaveStudentProfileDto ) : Promise<StudentProfile> {  Show usages  ± Latifa *
    try {
      return { ...(await this.studentProfileRepository.save({
        ...(dto.id ? { id: dto.id } : {}),
        name: dto.name,
        description: dto.description,
        searchInternship: dto.searchInternship,
        searchApprenticeship: dto.searchApprenticeship,
        searchOnSite: dto.searchOnSite,
        searchRemote: dto.searchRemote,
        searchHybrid: dto.searchHybrid,
        website: dto.website,
        facebook: dto.facebook,
        instagram: dto.instagram,
        linkedin: dto.linkedin,
        tiktok: dto.tiktok,
        github: dto.github,
        youtube: dto.youtube,
        deviantArt: dto.deviantArt,
        ProfileId: dto.ProfileId,
      )), context};
    } catch (error) {
      this.eventEmitter.emit('SaveStudentProfileUseCase::failed', { context: context, error: error.message });
      throw new BadRequestException(error.message);
    }
  }
}
```

SaveStudentProfileUseCase.ts

Diapo 27/40

CONCEPTION ET DÉVELOPPEMENT DES COMPOSANTS BACK-END



StudentProfileResolver

```
@Resolver(StudentProfile) Show usages ± Latifa *
export default class StudentProfileResolver { ± Latifa *
  constructor( no usages ± Latifa *
    private readonly serviceFactory: StudentProfileUseCaseFactory,
    @UseGuards(GraphqlAuthGuard) Show usages ± Latifa
    @Mutation(() => StudentProfile)
  async saveStudentProfile(
    @ContextualRequest() context: ContextualGraphqlRequest,
    @Args('dto') dto: SaveStudentProfileDto,
  ): Promise<StudentProfile> {
    return (await this.serviceFactory.create(SaveStudentProfileUseCase)).handle(context, dto);
  }
}
```

StudentProfileResolver.ts

Démarrage du serveur avec `yarn start:dev`.

Tester les résolveurs dans **GraphQL Playground**.<http://localhost:3030/graphql>

PRÉSENTATION DES ÉLÉMENTS LES PLUS SIGNIFICATIFS



UiButton

Attar Latifa
DWWM 2022-2024

```
interface Props { Show usages
  text?: string
  to?: string,
  disabled?: boolean,
  bgColor?: string,
  onPress?: () => void, .
const UiButton = ({ Show usage
  text,
  to = '',
  disabled = false,
  bgColor = '#ABF6f9',
  onPress,
}: Props) => (
  <PressableHStack
    onPress={() => {
      if (disabled) {
        return;
      }

      if (to && to !== '') {
        router.push(to);
      }

      return;
    }

    onPress?.();
  }>
  <UiText
    color={disabled ? '#1a1d1e' : textColor}
    fontSize={15}
    style={_textStyle}
    type={font}
  >
    { text }
  </UiText>
  </PressableHStack>
);

export default UiButton; S
```



Page d'accueil

Diapo 29/40

PRÉSENTATION DES ÉLÉMENTS LES PLUS SIGNIFICATIFS



UiTextField

```
interface Props { Show usages ± Quentin +4 *  
  label?: string,  
  placeholder?: string,  
  secureTextEntry?: boolean,  
  onChange?: (value: string) => void,  
  keyboardType?: KeyboardTypeOptions,  
  
const UiTextField = (props: Props) => { Show usages ± Quentin  
  return (  
    <View  
      style={{width: '96%'...}}  
    >  
    {  
      props.label && (  
        <UIText  
          textAlign="left"  
          color="black"  
          mb={2}  
          w="100%"  
          fontSize={15}  
          ...>  
      )  
      <HStack  
        style={{flexDirection: 'row'...}}  
      >  
      { props.icon }  
      <TextInput  
        keyboardType={props.keyboardType}  
        placeholder={props.placeholder}  
        secureTextEntry={props.secureTextEntry ?? false}  
        onChangeText={(text) => {  
          setValue(text);  
          props.onChange?.(text);  
        }}  
      />  
      { props.button }  
    </HStack>  
  </View>  
);  
  
export default UiTextField; Show usages ± Quentin
```

Créer mon compte

Remplissez vos coordonnées ou
continuez avec les réseaux sociaux

👤 prénom

✉️ email

🔒 mot de passe

🏡 Ville

Entrez un secteur d'activité

J'accepte les [conditions générales d'utilisation](#)

J'accepte la [politique de confidentialité](#)

S'inscrire

— Ou continuer avec —

G

f

Vous avez déjà un compte ? [Se connecter](#)

Page de création de compte

Diapo 30/40

SÉCURITÉ DE L'APPLICATION



TEST UNITAIRES



Objectif

Vérifier que la méthode `saveStudentProfile` du `StudentProfileResolver` crée correctement un profil étudiant.

Jeu d'essai

Définition des données d'entrée sous la forme d'un objet `SaveStudentProfileDto`.
Détermination des résultats attendus : création réussie d'un profil étudiant.

Configuration

Utilisation de `Test.createTestingModule` pour créer un module de test isolé.
Importation de `ApiModule`.
Inclusion des dépendances : `StudentProfileResolver` et `SaveStudentProfileUseCase`.

Exécution

Appel de la méthode `saveStudentProfile` du résolveur avec les données d'entrée définies.

Validation

Vérification des résultats obtenus par rapport aux résultats attendus en utilisant `expect`.

TEST UNITAIRES



```
describe('StudentProfileResolver', () => {
  let resolver: StudentProfileResolver;
  // Mettre en place le module de test
  beforeEach(async () => {
    const module: TestingModule = await Test.createTestingModule({
      imports: [ApiModule]
    }).compile();
    resolver = module.get<StudentProfileResolver>(StudentProfileResolver);
  });
  // Test pour la fonction saveStudentProfile
  describe('saveStudentProfile', () => {
    it('should create a new student profile', async () => {
      // Données en entrée
      const testDto: SaveStudentProfileDto = {
        name: "dwwm", ...
        ProfileId: 3 // ID du profil (lié à un autre modèle)
      };
      // Contexte GraphQL simulé
      const mockContext: ContextualGraphqlRequest = {
        userId: 3, // ID de l'utilisateur authentifié
        profil:{}
      };
      const savedProfile = await resolver.saveStudentProfile(mockContext,
      testDto);
      // Vérification des résultats obtenus
      expect(savedProfile).toBeDefined();
    });
  });
});
```

TEST UNITAIRES



```
▶ C:\Users\Latifa\AppData\Roaming\npm\yarn.cmd test
🔧 yarn run v1.22.21
warning ..\..\..\package.json: No license field
$ jest
PASS  src/Api/Profile/Test/studentProfileResolver.spec.ts (18.616 s)
  StudentProfileResolver
    saveStudentProfile
      ✓ should create a new student profile (678 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        18.996 s, estimated 20 s
Ran all test suites.
Done in 22.00s.
```

Terminal: Résultat du test

Result Grid	Filter Rows:	Edit:	Ex
id	ProfileId	name	description
1	3	DwWM	Je suis développeur Web et web mobile

Mysql Workbench - Création du studentProfile via le test effectué

RECHERCHE SUR SITES ANGLOPHONES



graphql vs useguards

Tous Vidéos Images Actualités Livres Finance Web

 **NestJS**
<https://docs.nestjs.com/graphql> · Traduire cette page

GraphQL + TypeScript - Other features
You can use standard guards, interceptors, filters and pipes in the same fashion with GraphQL as with any RESTful application.

 **Stack Overflow**
<https://stackoverflow.com/questions/how-to-implement-user-guards-in-nestjs-graphql> · Traduire cette page

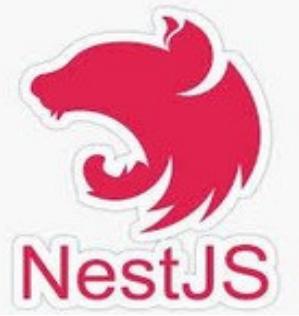
how to implement user guards in nestjs graphql
I'm trying to get the current user but in the resolver I get undefined, in the jwt strategy I get the user object using the token but in the resolver the user ...

 **GitHub**
<https://github.com/nest/Issues> · Traduire cette page

Authentication with GraphQL using @Guard · Issue #403
7 févr. 2018 — I'm using guards and this works fine - but I have no clue how to set context in graphqlExpress. In my old implementation I could just do:

 **hashnode.dev**
<https://paelinjanja.hashnode.dev/build-a-graphql-api-with-nestjs-and-mongodb-in-a-modern-way> · Traduire cette page

Build a GraphQL API with NestJs and MongoDB, in a modern way
27 févr. 2023 — **@UseGuards**: Used to apply one or more guards to a controller or route handler method. Guards can be used to protect routes or to perform ...



NestJS
GitHub



Everything you need to start
blogging as a developer!

SYNTHÉSE DU PROJET STUDEE



Attar Latifa
DWWM 2022-2024

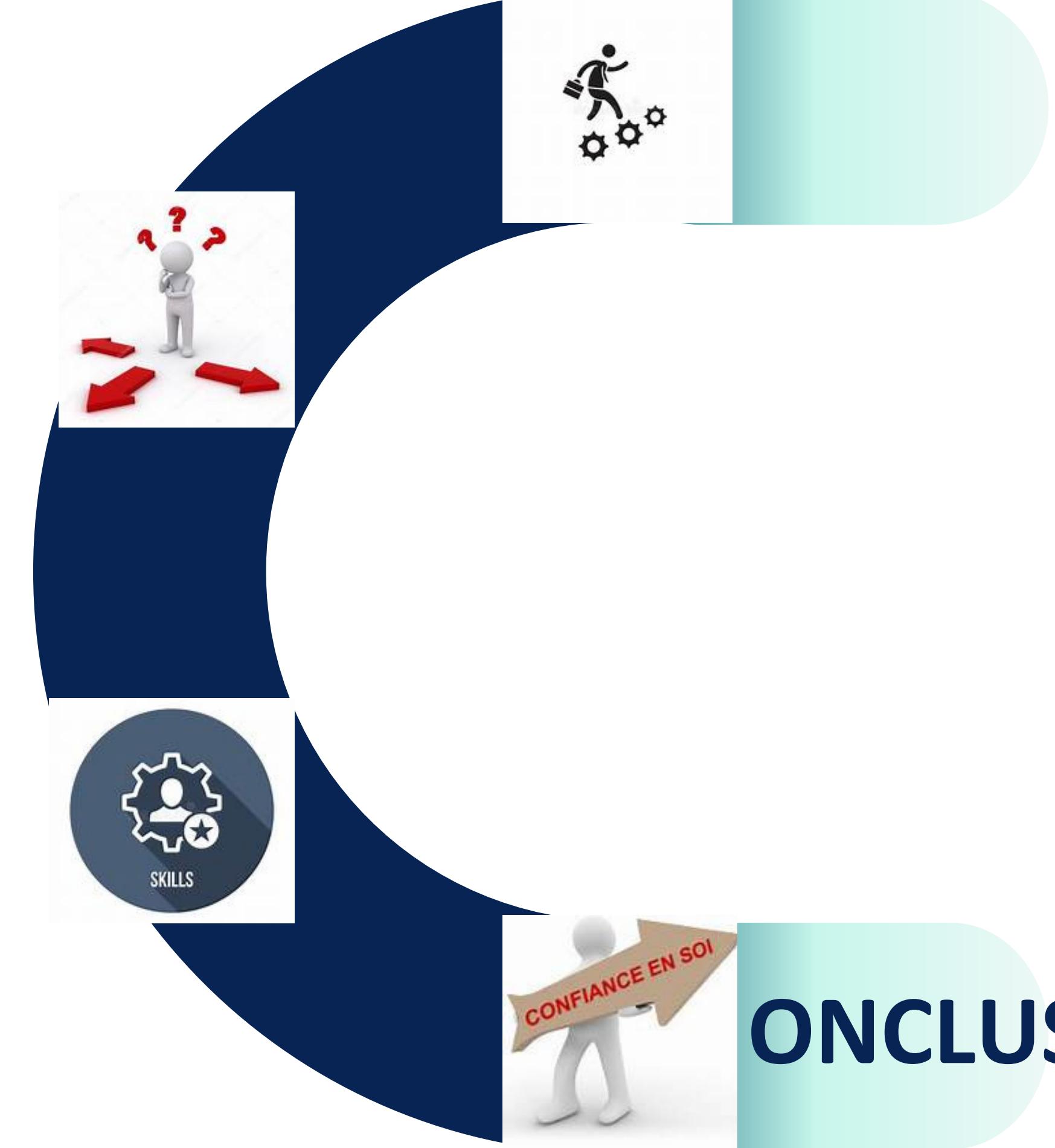


CONCLUSION



Attar Latifa

DWWM 2022-2024



Diapo 37/40

GRILLE D'ÉVALUATION DE STAGE



Attar Latifa
DWWM 2022-2024

Grille d'évaluation de stage



Nom du stagiaire : **Latifa ATTAR**

Nom du superviseur : **Quentin MOUSSET**

Objectifs du stage : **Validation d'acquis de formation en milieu professionnel**

Échelle d'évaluation :

1	Insuffisant aux attentes	2	Inférieur aux attentes	3	Moyennement conforme aux attentes	4	Conforme aux attentes	5	Dépasse les attentes
---	--------------------------	---	------------------------	---	-----------------------------------	---	-----------------------	---	----------------------

		Note
Connaissances	Maîtrise des équipements requis pour le travail (logiciels, outils, etc.)	4
	Connaissances théoriques ou techniques liées au poste	4
	Mise en place de stratégies pour acquérir les connaissances manquantes	5

Compétences professionnelles (savoir-faire)	Capacité à comprendre les objectifs et respecter les consignes	5
	Capacité d'exécuter les tâches demandées	4
	Capacité et volonté d'apprendre et de se perfectionner	5
	Capacité à travailler en équipe et/ou de façon indépendante	5
	Communication orale et/ou écrite	5
	Sens de l'organisation et gestion des priorités	4

Compétences personnelles (savoir-être)	Motivation et persévérance	5
	Ponctualité et assiduité	5
	Respect des règles de l'entreprise	5
	Autonomie et débrouillardise	4
	Relations interpersonnelles (avec le superviseur et les collègues)	5
	Gestion du stress	4

Commentaires :

Latifa s'est très bien intégrée au sein de son pôle mais aussi dans l'ensemble de l'association et des différents pôles. Elle a fait preuve des savoirs être essentiels à la vie d'une entreprise, communication, implication et esprit d'équipe. Latifa doit apprendre à se faire plus confiance sur le côté technique, l'expérience l'aidera en ce sens. Sa curiosité et sa persévérance l'aideront grandement dans sa future carrière.

Signature : **Quentin MOUSSET**

Date : **19/05/2024**

HDM NETWORK ASBL
38 rue Alfred Tenret
6030 Charleroi Belgique
Tél. : +32 (0)4 99 17 91 47
BCE/TVA : BE0769.344.711
RPM Charleroi

LETTRE DE RECOMMENDATION



Attar Latifa
DWWM 2022-2024



Responsable/Co-owner : Mousset Quentin

Adresse :

Rue Alfred Tenret, 38

6030 Charleroi, Belgique

Mail : rh@hdmnetwork.com



Objet : Lettre de recommandation

Madame, Monsieur,

Mme Latifa ATTAR a effectué son stage chez HDM Network ASBL au sein du pôle Web Développement du 19/02/2024 au 17/05/2024.

Au cours de son stage, Mme Latifa ATTAR, à travers son travail a su nous démontrer une maîtrise assez diverse du langage informatique tel que JS, HTML5, CSS3, PHP et MYSQL ainsi que Node JS, Divers Librairie et pour finir l'utilisation de Framework comme React et Symfony. Durant son stage c'est avec une certaine créativité et un investissement important que Mme Latifa ATTAR a pu réaliser les missions qui lui ont été attribuées tel que :

- La création de base de données
- La réalisation d'API
- La réalisation de site web/refonte
- Le Scraping de données
- La réalisation d'application web front-end et back-end

Dans chacune de ses missions Mme Latifa ATTAR a effectué son travail avec professionnalisme, rigueur et de manière organisée. Au sein de notre stage, le travail s'effectue en autonomie mais également en équipe, Mme Latifa ATTAR a parfaitement su s'adapter en faisant preuve de respect ainsi que d'un bon sens de communication ce qui a été très apprécié par l'ensemble de son pôle.

Pour toutes ces raisons, je ne peux que vous recommander Mme Latifa ATTAR persuadé qu'elle sera être un atout majeur au sein de votre entreprise.

Pour de plus amples renseignements, n'hésitez pas à nous contacter.

Cordialement.

Fait à Charleroi, le 19/05/2024

Mousset Quentin

HDM NETWORK ASBL

38 rue Alfred Tenret
6030 Charleroi Belgique
Tél : +32 (0)4 99 17 91 47
BCE/TVA : BE0769.344.711
HDM Charleroi

REMERCIEMENT



Thank You

