



DEPARTEMENT INFORMATIQUE

RAPPORT DE PROJET DE FIN D'ETUDES

En vue de l'obtention du :
Diplôme National d'Ingénieur Informatique
Option : Informatique Industrielle

Conception et réalisation d'une application de présentation configurable

Réalisé à : eNano-Tech



Elaboré par : MAJDOUB Baha Eddine

Soutenu le 03 / 10 / 2020 devant le jury :

Président	:	M. ACHOUR Sami	ISSAT de Sousse
Examineur	:	Mme FEKI Souheila	ISSAT de Sousse
Encadrant	:	M. BEN AHMED Sofiene	ISSAT de Sousse
Encadrant Industriel	:	M. HABACHI Mahmoud	eNano-Tech

Année Universitaire : 2019 / 2020

Dédicaces

Je dédie ce travail

A mes chers parents

Pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de nos études

A mon cher frère

Pour ses encouragements permanents, et son soutien moral

A mes chères sœurs

Pour leurs appuis et leurs encouragements

A mes professeurs

A tous mes amis

Et à tous ceux qui me sont proches...

Remerciements

Je tiens à exprimer mes vifs remerciements à mon encadrant M. SOFIEN BEN AHMED pour avoir dirigé ce travail, pour m'avoir soutenu ainsi que pour ses précieux conseils et pour sa bonne humeur.

Mes vifs remerciements s'adressent aussi à M. Mahmoud HABACHI, mon encadrant professionnel qui a facilité mon intégration au sein de eNano-Tech et qui m'a aidé tout au long de ce rapport.

Je voudrais aussi exprimer ma gratitude à tous les personnels d'eNano-Tech et en particulier à ceux qui travaillent dans le département web et mobile pour tout le soutien et toute l'aide qu'ils m'ont apporté durant les mois du stage.

Mes vifs remerciements s'adressent aussi à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail, à ma famille et à tous mes amis qui m'ont apporté un soutien moral durant ce travail.

Je voudrais remercier aussi les membres du jury pour l'honneur qu'ils nous ont fait en acceptant de juger ce modeste travail.

Sommaire

INTRODUCTION GÉNÉRALE	1
CHAPITRE I: CONTEXTE GENERAL DU PROJET.....	3
1- PRESENTATION DE L'ENTREPRISE	4
2- PROBLEMATIQUE, CAHIER DES CHARGES ET OBJECTIFS DU PROJET	7
3- PLAN DE TRAVAIL, METHODES ET OUTILS	8
3.1- Plan de Travail.....	8
3.2- Méthodes de conception	8
3.3- Moyens (logiciels)	8
4- ETUDE DE L'EXISTANT.....	9
4.1- Analyse de l'existant	9
4.2- Critique de l'existant.....	10
4.3- Proposition de différentes solutions.....	11
CHAPITRE II: CONCEPTION DE LA SOLUTION.....	13
1- SPECIFICATION DES BESOINS.	14
1.1- Les besoins attendus de l'application	14
1.2- Les besoins fonctionnels.....	14
1.3- Les besoins non fonctionnels.....	15
2- LE LANGAGE UML:	15
2.1- L'intérêt d'UML.....	15
2.2- Définition d'un modèle	16
2.3- Les différents types diagrammes UML	16
2.4- Identification des acteurs.....	16
2.5- Diagramme de contexte du système à réaliser.....	17
3- LES DIAGRAMMES DES CAS D'UTILISATION :	17
3.1- Le diagramme global des cas d'utilisation:.....	17
3.2 -Les différents cas d'utilisation:	19
4- LE DIAGRAMME DE CLASSE:	26
4.1- Définition d'un diagramme de classe.....	26
4.2- Diagramme de classe de l'application.....	27
5- LE DIAGRAMME DE SEQUENCE:	28
5.1- Définition d'un diagramme de séquence.....	28

5.2- Les différents diagrammes de séquence	29
CHAPITRE III: REALISATION	36
I- ENVIRONNEMENT DU TRAVAIL :.....	37
1- ENVIRONNEMENT MATERIEL :.....	37
1.1- Choix de l'architecture de l'application	37
1.2- Elaboration du diagramme de déploiement	39
1.3- Choix de type de l'application	39
1.4- Processus de synchronisation	41
2- ENVIRONNEMENT LOGICIEL:	42
2.1- Les Frameworks.....	42
2.2- Les langages	43
2.3- Les outils	44
2.4- Les technologies utilisées.....	46
II- REALISATION DES INTERFACES DE L'APPLICATION :.....	48
1- INTRODUCTION:	48
2- REALISATION.....	48
2.1- Sur PC, Tablette, SmartTV.....	48
2.2- Sur Smartphone	65
CONCLUSION GÉNÉRALE	73
WEBOGRAPHIE.....	74

Liste des Figures

Figure 1 : Logo de société	4
Figure 2 : Organigramme général d'eNano-Tech	6
Figure 3 : Logo de Power Point	9
Figure 4 : Logo de Google Slides	9
Figure 5 : Logo d'Adobe Captivate	9
Figure 6 : Logo de Presi.....	10
Figure 7 : Les diagrammes UML	16
Figure 8 : Diagramme de contexte du système à réaliser	17
Figure 9 : Diagramme global des cas d'utilisation.....	19
Figure 10 : Diagramme du cas d'utilisation Authentification.....	21
Figure 11 : Diagramme du cas d'utilisation Inscription.....	22
Figure 12 : Diagramme du cas d'utilisation Déconnexion.....	22
Figure 13 : Diagramme du cas d'utilisation Créer un projet.....	23
Figure 14 : Diagramme du cas d'utilisation Modifier le profil	24
Figure 15 : Diagramme du cas d'utilisation Modifier un projet.....	25
Figure 16 : Diagramme du cas d'utilisation Synchronisation	26
Figure 17 : Diagramme de classes	28
Figure 18 : Diagramme de séquences relatif à « S'authentifier ».....	30
Figure 19 : Diagramme de séquences relatif à « S'inscrire »	31
Figure 20 : Diagramme de séquences relatif à « Modifier Profil ».....	32
Figure 21 : Diagramme de séquences relatif à «Créer un projet»	33
Figure 22 : Diagramme de séquences relatif à «Modifier Projet».....	34
Figure 23 : Diagramme de séquences relatif à « Synchronisation »	35
Figure 24 : Architecture 3-tiers.....	38
Figure 25 : Diagramme de Déploiement.....	39
Figure 26 : Architecture de service Worker	40
Figure 27 : Processus de synchronisation	42
Figure 28 : Logo Angular.....	42
Figure 29 : Logo NestJS	43
Figure 30 : Logo HTML	43
Figure 31 : Logo TypeScript	43
Figure 32 : Logo Sass.....	44
Figure 33 : Logo mongoDB.....	44

Figure 34 : Logo IndexedDB.....	45
Figure 35 : Logo Visual Code	45
Figure 36 : Logo StarUML	45
Figure 37 : Logo Bitbucket	46
Figure 38 : Exemple JSON.....	46
Figure 39 : REST API.....	47
Figure 40 : Logo Facebook.....	47
Figure 41 : Logo Google Plus.....	48
Figure 42 : L'entête de landing page.....	49
Figure 43 : Le corps de landing page	49
Figure 44 : Le bas de landing page	50
Figure 45 : Interface d'accueil	50
Figure 46 : menu de navigation (accueil)	51
Figure 47 : Interface de création de projet.....	51
Figure 48 : menu de navigation (constructeur).....	51
Figure 49 : Structure des composants.....	52
Figure 50 : Menu des éléments.....	53
Figure 51 : Menu des configurations de navigateur	53
Figure 52 : Arbre des éléments	53
Figure 53 : breadcrumb	54
Figure 54 : Propriété Graphique.....	54
Figure 55 : Ajouter action.....	54
Figure 56 : Alignement rapide	55
Figure 57 : Barre des outils de texte	55
Figure 58 : Exemple d'un élément Texte	55
Figure 59 : Configuration de carrousel	56
Figure 60 : Exemple d'un élément Carrousel	56
Figure 61 : Navigateur.....	57
Figure 62 : Configuration de Navigateur.....	57
Figure 63 : Etiquette de pastille.....	57
Figure 64 : Exemple d'une action.....	58
Figure 65 : Propriétés des onglets.....	58
Figure 66 : Exemple d'un élément Tabs.....	59
Figure 67 : Exemple d'un élément Galerie	59
Figure 68 : Photo sélectionnée de l'élément Galerie	60

Figure 69 : Grid des photos	60
Figure 70 : Menu Project.....	61
Figure 71 : Enregistrer le projet	61
Figure 72 : Modifier la résolution de panneaux.....	61
Figure 73 : Boîte d'alerte	62
Figure 74 : Interface d'accueil	62
Figure 75 : Carte d'un projet	63
Figure 76 : Modal d'authentification.....	63
Figure 77 : Modal d'inscription	63
Figure 78 : Menu de navigation (Après l'authentification)	64
Figure 79 : Modal de profil	64
Figure 80 : Interface d'authentification	65
Figure 81 : Interface d'inscription.....	66
Figure 82 : Interface d'accueil	67
Figure 83 : Interface Modifier le profil.....	68
Figure 84 : Synchronisation.....	69
Figure 85 : Interface d'accueil	69
Figure 86 : Interface de présentation.....	69
Figure 87 : Mot de passe oublié.....	70
Figure 88 : Alerte email envoyé	71
Figure 89 : Email de mot de passe	71

Liste des Tableaux

Tableau 1: Comparaison des solutions existantes	10
Tableau 2 : Le formalisme de description des cas d'utilisation	20
Tableau 3 : Description du cas d'utilisation Authentification	20
Tableau 4 : Description du cas d'utilisation Inscription/Enregistrement	21
Tableau 5 : Description du cas d'utilisation Déconnexion	22
Tableau 6 : Description du cas d'utilisation Créer un projet	23
Tableau 7 : Description du cas d'utilisation Modifier le profil	24
Tableau 8 : Description du cas d'utilisation Modifier un projet	25
Tableau 9 : Description du cas d'utilisation Synchronisation	26

Liste des acronymes

UML : Unified Modeling Language

API : Application Programming Interface

HTTP : HyperText Transfer Protocol

REST : Representational State Transfer

JSON : JavaScript Object Notation

SGBD : Système de gestion de base de données

PWA : Progressive Web Application

DOM : Document Object Model

MVC : Modèle-vue-contrôleur

Introduction générale

L'univers informatique est devenu une composante clé dans notre quête d'informations. Mais si le savoir est aujourd'hui accessible en quelques clics, des changements structurels fondamentaux sont à prévoir dans notre manière d'enseigner, d'apprendre et d'administrer les dispositifs éducatifs. Une véritable révolution est en marche pour réinventer la pédagogie de demain.

Aujourd'hui, la technologie a évolué de façon que tout le monde utilise des équipements informatiques. Pour cette raison, de nombreuses entreprises cherchent à moderniser leurs communications en s'équipant d'applications web et mobile qui ont l'avantage d'être multiplateformes et compatibles avec les Smartphones, les tablettes, les smart TV et les ordinateurs, sans nécessiter pour autant d'un financement abusif.

Ainsi, l'information bien présentée est un des leviers les plus efficaces pour engager l'audience. Pour cela, on doit choisir un outil de création de présentations qui permet de transmettre efficacement le message, d'afficher et d'illustrer des informations concernant des campagnes de marketing, ou à des fins éducatives, entre autres. Ces présentations peuvent être attrayantes et plus interactives lorsqu'elles sont utilisées conjointement avec les systèmes de réponse du public et les outils de conférence Web.

Constatant la grande nécessité des outils de présentation, eNano-Tech, entreprise tunisienne spécialisée dans les solutions informatiques, a décidé de développer une application multiplateforme avec des critères concurrentiels, qui permet à l'utilisateur de réaliser un pas de plus vers les nouvelles technologies et de gérer des présentations attrayantes et interactives même en mode hors ligne.

Ce présent rapport sera donc réparti en trois chapitres. Dans le premier chapitre, nous allons introduire le contexte général du projet en présentant la société eNano-Tech, la problématique ainsi que les objectifs et l'étude d'existant. Le deuxième chapitre sera consacré à la conception qui est une étude préalable de notre projet, tout en présentant les besoins de notre système. La réalisation de notre application fera l'objet du chapitre trois, où nous allons présenter les outils ainsi que les langages de programmations utilisés et expliquer le

fonctionnement de notre application en utilisant quelques captures d'écran. Nous concluons ce rapport par exposer l'ensemble des connaissances acquises au cours de la réalisation de cette application, tout en illustrant quelques perspectives.

CHAPITRE I :

Contexte général du projet

Introduction

Dans ce premier chapitre, nous allons présenter la société eNano-Tech au sein de laquelle j'ai effectué mon stage, la problématique, l'objectif et l'étude d'existant de notre travail.

1- Présentation de l'entreprise

eNano-Tech est une agence spécialisée dans des solutions informatiques et de communications novatrices et à forte valeur ajoutée. L'expérience acquise au service de ses clients, lui a permis de se positionner rapidement sur le marché des nouvelles technologies de l'information et de la communication.

eNano-Tech devient une division du groupe Pixelraise (Pixelraise est une agence digitale spécialisée dans le conseil et le déploiement des solutions business et techniques pour une transformation digitale aboutie, elle est située en France, Tunisie, Suisse et Canada), elle propose à ses clients une relation d'accompagnement et de conseil pour une réalisation réussie de leurs projets, à travers une large panoplie de services couvrant l'ensemble du cycle de vie d'un projet informatique, allant des études des besoins, à la spécification fonctionnelle et technique, au développement et intégration des solutions choisies, et au transfert de savoir-faire en passant par l'assistance à la mise en œuvre des systèmes fournis. [1]

❖ La Carte d'identité

Nom commercial : eNano-Tech

Gérant : M. Amine BOUSEMA

Secteur d'activité : Développement informatique

Téléphone : +216 70 035 774

Siège social : 08 Rue Sidi Sarrag, Monastir 5000

Date Création: Avril 2015

E-mail : amine@pixelraise.com

Les effectifs employés: 30



Figure 1 : Logo de société

❖ Produits et services

eNano-Tech est une société spécialisée dans la Développement informatique :

- **Identité Corporate:** Une forte identité visuelle est d'une importance vitale pour le succès de toute entreprise. L'équipe de designers vous offrira l'opportunité de créer une forte image de marque pour votre société.
- **Digital Marketing:** Leader dans le marché, eNano-Tech est capable de gérer votre campagne sur Google Adword, Facebook ads ou tout autre support et cela approfondie des perspectives d'évolution de votre marque.
- **IT Consulting :** L'équipe IT vous apporte un accompagnement régulier tout au long de la vie de votre projet digital : conseil, assistance et suivi constant.
- **Web Design :** eNano-Tech ne crée pas seulement votre site, elle vous aide à livrer une expérience utilisateur unique, agréable et enrichissante, ainsi elle participe à la fidélisation de vos clients et à la croissance de votre activité.
- **Développement Web :** Pour tous les nouveaux sites qu'elle conçoit, la vision mobile first elle a permis de concevoir vos portails d'abord pour le mobile, mettant ainsi en avant le contenu le plus pertinent pour qu'il soit directement accessible sur les appareils mobiles.
- **Développement Mobile :** Bénéficiez des avantages des plates-formes mobiles populaires pour vendre, promouvoir ou élargir votre présence en ligne avec les Smartphones Android et iOS grâce aux services de développement d'applications.
- **E-commerce :** L'agence eNano-Tech s'est spécialisée dans la création des sites e-commerce. De par cette expertise, elle propose des solutions de vente en ligne puissantes.
- **Référencement SEO :** eNano-tech référence votre site web selon stratégie sémantique structurée et une optimisation de contenu professionnelle.

❖ Différents services d'eNano-Tech

La figure 2 représente l'organigramme de l'eNano-Tech. Cette société comprend : Service administratif, Service financière, Service GRH, Service Informatique.

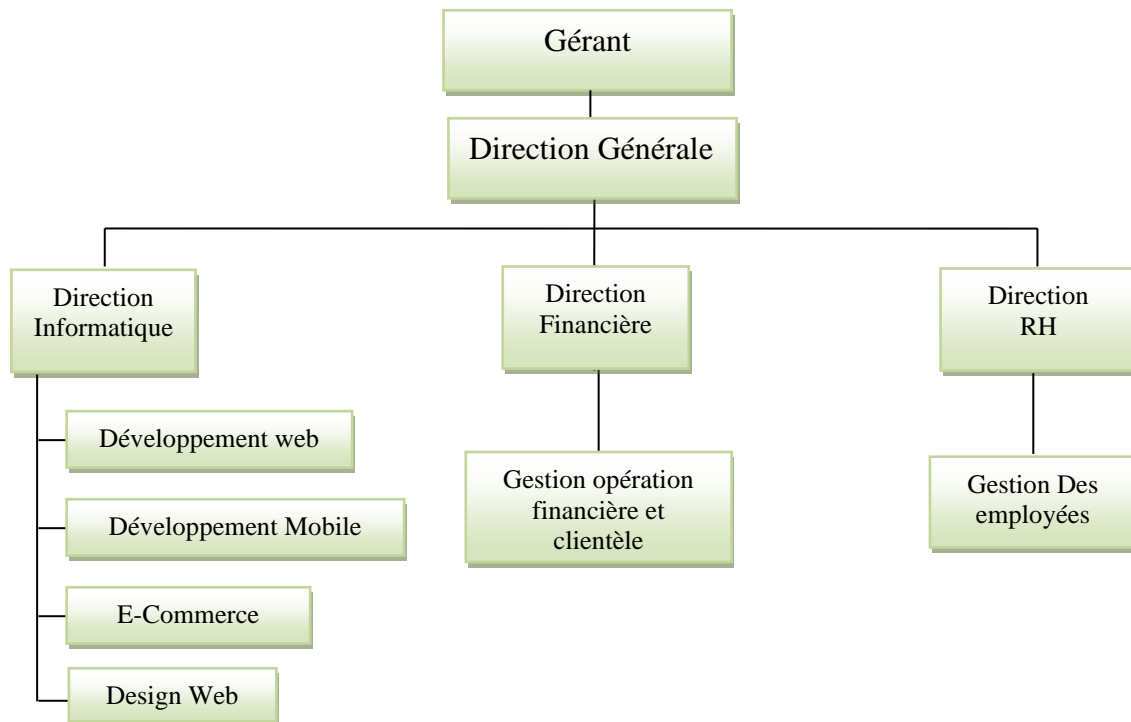
L'organigramme de la société :

Figure 2 : Organigramme général d'eNano-Tech

L'organigramme d'eNano-Tech se présente comme suit : (Figure 2)

eNano-Tech se compose de plusieurs directions :

- **Direction Générale** : Le gérant, qui est le propriétaire d'eNano-Tech, définit la stratégie de l'entreprise et sa politique et supervise les différentes directions et services afin de s'assurer de la mise en œuvre de cette stratégie.
- **Direction informatique** : Elle veille au bon fonctionnement des systèmes d'informations, c'est la direction de production et le secteur d'activité principal de la société.
- **Direction Ressources Humaines** : La principale mission de cette direction consiste à renforcer, mobiliser et dynamiser le capital humain de l'entreprise.
- **Direction Financière** : Elle règle les transactions financières et tient la comptabilité de la société. Elle contrôle les revenus et les coûts, interprète les ratios et analyse les résultats obtenus au cours des exercices comptables.

2-Problématique, cahier des charges et objectifs du projet

Les applications responsive est actuellement en pleine croissance grâce aux succès de Smartphones, de tablettes et de Smart TV ceux qui tournent sous Android et IOS. Cette tendance offre aux entreprises de formidables opportunités pour créer de nouveaux services ou élargir ces supports de communication. Dans cette perspective, la société eNano-Tech souhaite exploiter cette opportunité dans leurs produits et elle nous demande de créer une application web progressive.

L'application web progressive est un site web qui apparaît aux mobinautes de la même manière qu'une application mobile native.

Depuis quelques années, les gens se plaignent du manque de facilité et des interactivités des outils de présentation pour présenter leurs idées et pour faire convaincre leurs interlocuteurs, leurs clients et leurs prospects.

C'est pour cette raison, la problématique de notre projet consiste à développer une application de présentation pour répondre à leur besoins.

L'objectif de notre projet est de réaliser ces besoins en développant une application web progressive qui permet à l'utilisateur de réaliser un pas de plus vers les nouvelles technologies et de gérer des interfaces utilisateur attrayantes et des présentations interactives pour tout système d'information en libre-service ou écran tactile. Du bout des doigts, il intègre des présentations de produits, des applications de commerce électronique, des projets d'éducation, des informations sur les clients ou des publicités animées à présenter lors de salons et d'expositions, dans les espaces clients, les universités, les espaces publics ou les points de vente.

Dans cette optique, et afin de satisfaire les spécifications du cahier des charges, notre mission dans ce projet consiste à :

- Réaliser la conception de l'application avec la méthode UML,
- Développer la couche métier et les web services nécessaires dans l'application,
- Développer une interface de dialogue Homme-Machine à utiliser par les utilisateurs,
- Valider l'étude ainsi réalisée par des tests.

3. Plan de travail, méthodes et outils

3.1. Plan de travail

Le plan de travail adapté à ce projet se présente comme suit :

1. Conception du projet.
2. Elaboration des différents diagrammes (cas d'utilisation, séquences, classes).
3. Choix de la configuration logiciel.
4. Développement des services web nécessaires pour assurer le fonctionnement de notre application en local avec la base des données locale de web IndexedDB.
5. Développement d'une interface homme-machine répondant aux spécifications du cahier des charges en utilisant le Framework Angular.
6. Développement de la couche métier et les services web nécessaires dans notre application pour assurer la synchronisation des données de local vers le serveur avec le Framework NestJs et la base des données MongoDB.
7. Validation de l'application développée par des différents tests.

3.2. Méthodes de conception

La méthode de conception utilisée dans ce projet est le langage de Modélisation Unifié **UML**. Nous avons utilisé essentiellement ces trois diagrammes :

- Le diagramme de cas d'utilisations.
- Le Diagramme de classes.
- Le diagramme de séquence.

3.3. Moyens (Logiciels)

Les logiciels utilisés dans la réalisation de ce projet sont :

- L'éditeur de Text Visual code.
- Le système de gestion de base de données MongoDB.

4. Etude de l'existant :

Cette section a pour objectif d'étudier les outils de présentation les plus connus sur le marché. Cette étude permet de dégager les points forts et les points faibles de chacun de ces outils. Dans ce qui suit, nous allons présenter l'analyse de l'existant, puis nous allons détailler la critique de l'existant.

4.1. Analyse de l'existant :

Aujourd'hui, l'utilisation d'un support papier pour présenter nos idées peut sembler dépassée. Ce type de présentation présente beaucoup d'inconvénients. Dans le but de résoudre ces inconvénients, plusieurs outils ont été créés à base des nouvelles technologies. Parmi lesquels nous pouvons citer :

- ❖ **Power Point** : Microsoft PowerPoint est un logiciel de présentation édité par Microsoft. Il fait partie de la suite Microsoft Office. Microsoft PowerPoint fonctionne sous Windows et Mac OS. [2]



Figure 3 : Logo de Power Point

- ❖ **Google Slides** : Google Slides est un programme de présentation inclus dans le cadre d'une suite bureautique logicielle offerte par Google au sein de son service Google Drive. Il met nos idées en valeur grâce à des centaines de polices et des vidéos intégrées. [3]



Figure 4 : Logo de Google Slides

- ❖ **Adobe Captivate** : Adobe Captivate est un outil payant de création utilisé pour créer des contenus d'apprentissage en ligne tels que des démonstrations de logiciels et des simulations de logiciels. [4]



Figure 5 : Logo d'Adobe Captivate

- ❖ **Prezi** : Prezi est un logiciel de présentation édité par la société hongroise éponyme, créé et lancé à Budapest en 2009. [5]



Figure 6 : Logo de Prezi

4.2 Critique de l'existant

Le tableau ci-dessous récapitule les avantages et les inconvénients de tous les outils traités, mentionnés dans le paragraphe précédent.





	Power Point	Google Slides	Prezi	Adobe Captivate
Prix	19,5 DT/mois	116 DT/mois	22,76 DT/mois	93,93 DT/mois
Plateformes				
Stockage locale	Oui	Non	Oui	Non
Hors-ligne	Oui	Non	Oui (sauf stockage)	Oui
Synchronisation des données	Non	Oui	Non	Non
Diapos	Oui	Oui	Non	Oui
Usage	Simple	Simple	Complicé	Complicé
Interactivité	Non	Non	Non	faible
Fonctionnalités	Transitions Objets 3D audio Vidéo	Transitions Objets 3D audio Vidéo	Transitions audio Vidéo	Transitions Objets 3D audio Vidéo

Tableau 1: Comparaison des solutions existantes

Comme le montre clairement le tableau ci-dessus, les outils existants de présentation sur le marché proposent de différentes fonctionnalités de base. Cependant, ils possèdent plusieurs des inconvénients tels que :

- Le paiement des outils.
- La portabilité.
- La complexité d'utilisation.
- Tous les outils fonctionnent avec le principe de diapositive.
- Presque tous les outils n'assurent pas la synchronisation des données.
- Tous les outils n'assurent pas l'interactivité pour tous les systèmes d'information.
- Le stockage et l'utilisation de certains outils nécessite une connexion à l'internet.

Notre étude a montré donc que les outils du marché n'offrent pas toutes les fonctionnalités nécessaires. Pour cette raison, notre application tente d'être parmi les premiers outils qui intègrent ces différentes fonctionnalités et confrontent les inconvénients des solutions existantes.

4.3. Proposition de différentes solutions

L'étude de l'existant nous a permis de dégager plusieurs anomalies que nous avons détaillées dans la section précédente. Pour faire recours à ces anomalies, nous proposons de concevoir et d'implémenter une application web qui regroupe toutes les fonctionnalités de présentation.

Dans notre solution, nous envisageons que :

- ✓ Notre application doit être basée sur une interface web responsive, ce qui évitera les problèmes de compatibilité avec le système d'exploitation de l'utilisateur.
- ✓ Notre application doit présenter des interfaces utilisateur tactiles attrayantes, des présentations multimédias parfaitement intégrées, des applications tactiles conviviales ainsi que de nombreuses applications interactives. Grâce à une architecture logicielle flexible, même des projets complexes peuvent être mis en œuvre sans planification en quelques jours.
- ✓ Notre application doit être basée sur une architecture logicielle flexible, ce qui évitera les problèmes de mettre en œuvre des projets complexes.

- ✓ Notre application doit être rapide et sécurisée. Étant donné le nombre de fonctionnalités importantes, elle devra aussi offrir une simplicité d'utilisation et surtout une ergonomie d'interface.
- ✓ Notre application doit assurer la synchronisation des projets sur tous les appareils et le fonctionnement en mode hors ligne.
- ✓ Notre application doit regrouper les fonctionnalités de toutes les applications présentées dans l'étude de l'existant.

Conclusion

Dans ce chapitre, nous avons introduit le contexte général du projet en présentant la société eNano-Tech, la problématique, les objectifs du projet ainsi que l'étude d'existant. Dans le chapitre suivant, nous allons présenter l'étude de conception de l'application.

Chapitre II : Conception de la solution

Introduction

La phase de conception est la première étape dans la réalisation d'un projet, elle doit d'écrire de manière non ambiguë le fonctionnement futur du système, afin d'en faciliter la réalisation. Pour cela, différentes méthodes existent permettant de formaliser les étapes préliminaires du développement.

Dans ce chapitre, nous allons présenter les objectifs de notre application tout en identifiant les possibilités du système et les besoins des utilisateurs à l'aide des diagrammes de cas d'utilisations globaux et détaillés.

1. Spécification des besoins

1.1. Les besoins attendus de l'application

L'application envisagée doit satisfaire les besoins fonctionnels qui seront exécutés par le système et les besoins non fonctionnels qui perfectionnent la qualité logicielle du système.

1.2. Les besoins fonctionnels

Les besoins fonctionnels ou besoins métiers représentent les actions que le système doit exécuter, il ne devient opérationnel que s'il les satisfait. Cette application doit couvrir principalement les besoins fonctionnels suivants :

- Enregistrement des utilisateurs.
- Authentification des utilisateurs.
- Création des projets.
- Consultation des projets et démarrage de présentation.
- Modification et/ou suppression d'un projet.
- Importation et/ou exportation d'un projet.
- Fonctionnement en mode hors-ligne et enligne.
- Synchronisation des données de local vers le serveur.
- Modification de profil.
- Déconnexion (quitter l'application).

1.3. Les besoins non fonctionnels

Ce sont des exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt identifient des contraintes internes et externes du système. Les principaux besoins non fonctionnels de notre application se résument comme suit :

- Le code doit être clair pour permettre des futures évolutions ou améliorations.
- L'ergonomie : l'application offre une interface conviviale et facile à utiliser.
- La sécurité : l'application doit respecter la confidentialité des données.
- Garantir l'intégrité et la cohérence des données à chaque mise à jour et à chaque insertion.

2. Le langage UML

UML (Unified Modeling Language), se définit comme un langage de modélisation graphique et textuel destiné à comprendre et à définir des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML modélise l'ensemble des données et des traitements en élaborant des différents diagrammes. [6]

En clair, il ne faut pas designer UML en tant que méthode (Il y manque la démarche) mais plutôt comme une boîte d'outils qui sert à améliorer les méthodes de travail.

2.1. L'intérêt d'UML

UML est un langage semi-formel et normalisé qui :

- Permet le gain de précision.
- Gagne de stabilité.
- Encourage l'utilisation d'outils.

UML est un support de communication performant, puisqu'il cadre l'analyse, facilite la compréhension de représentations abstraites complexes. Son caractère polyvalent et sa souplesse en font un langage universel. [7]

2.2. Définition d'un modèle

Un modèle est une vue subjective mais pertinente de la réalité. Un modèle définit une frontière entre la réalité et la perspective de l'observateur. Ce n'est pas "la réalité", mais une vue très subjective de la réalité. Bien qu'un modèle ne représente pas une réalité absolue, un modèle reflète des aspects importants de la réalité, il en donne donc une vue juste et pertinente [8].

2.3. Les différents types diagrammes d'UML:

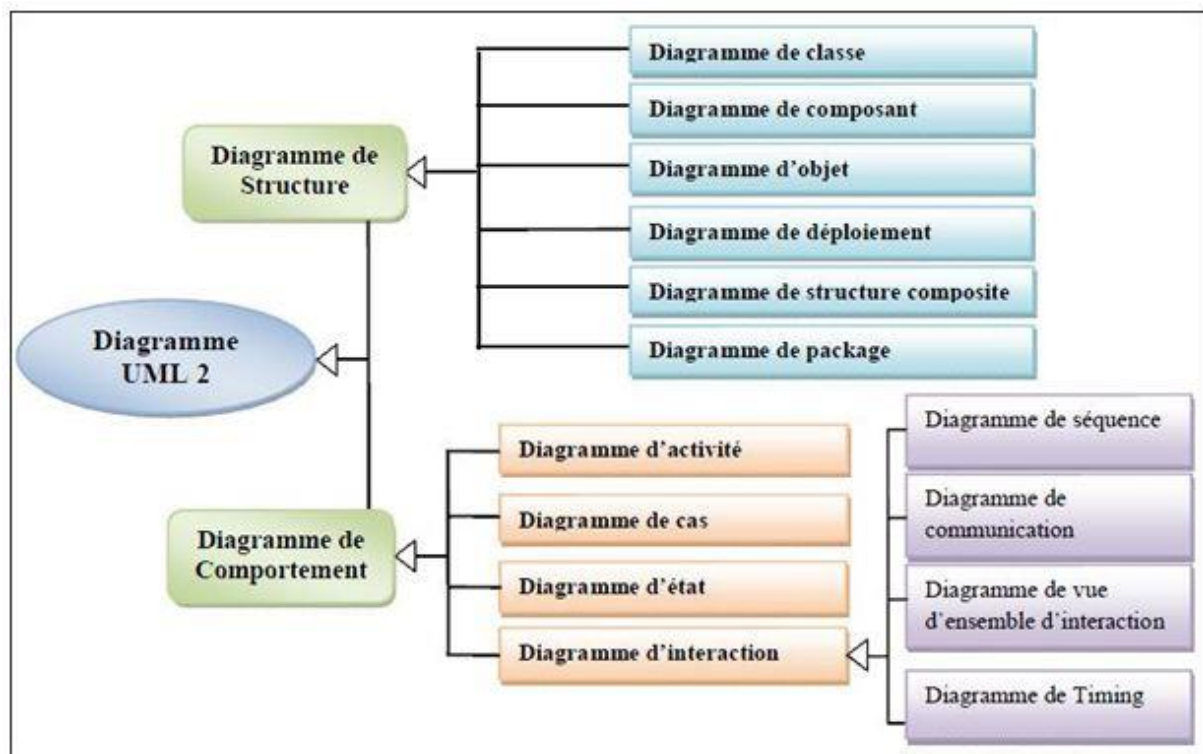


Figure 7 : Les diagrammes UML

2.4. Identification des acteurs

Un acteur représente un rôle joué par un utilisateur humain ou un autre système qui interagit directement avec le système étudié. Un acteur participe à au moins un cas d'utilisation.

Un acteur peut consulter et/ou modifier directement l'état du système, en création et/ou en regardant des projets d'être porteurs de données.

Dans notre cas, nous avons un acteur qui est l'utilisateur à un accès au système via un contrôle d'accès (login et mot de passe). Les opérations qu'il peut les effectuer sont :

- Créer des projets.
- Afficher la liste des ses projets.
- Démarrer une présentation.
- Modifier et/ou supprimer un projet.
- Importer et/ou Exporter un projet.
- Synchroniser ses projets avec le serveur.
- Modifier son profil.
- Se déconnecter.

2.5. Diagramme de contexte du système à réaliser

La figure ci-dessous montre l'interaction entre un acteur qui est l'utilisateur et le système que nous allons mettre en place :

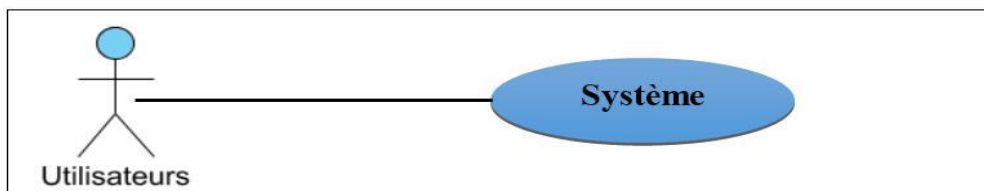


Figure 8 : Diagramme de contexte du système à réaliser

3. Les Diagrammes des cas d'utilisation

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre. [9]

3.1. Le diagramme global des cas d'utilisations

La figure 9 représente le diagramme de cas d'utilisation générale qui fait preuve d'un acteur appelé utilisateur, qui admet les fonctionnalités suivantes :

- **Création d'un projet** : L'utilisateur peut créer un projet à travers l'interface accueil.
- **Accès au projet** : L'utilisateur peut accéder à ses projets.
- **Démarrage d'un projet** : L'utilisateur peut démarrer une présentation d'un projet.

- **Importation/Exportation d'un projet :** L'utilisateur peut exporter un projet en un fichier XML ou importer un fichier XML déjà exporté.
- **Inscription :** L'utilisateur doit s'inscrire à l'application qui présente une interface dont il doit introduire son login et son mot de passe afin qu'il puisse synchroniser ses projets.
- **Authentification :** L'authentification doit être obligatoire pour la synchronisation des données.
- **Synchronisation :** Une fois l'utilisateur est authentifié, il peut synchroniser ses projets de la base locale vers le serveur.
- **Accès au profil :** L'utilisateur peut accéder à son profil pour consulter et/ou modifier ses informations personnelles.
- **Réception des notifications :** L'utilisateur peut recevoir une notification quand l'application détecte une connexion internet.
- **Déconnexion.**

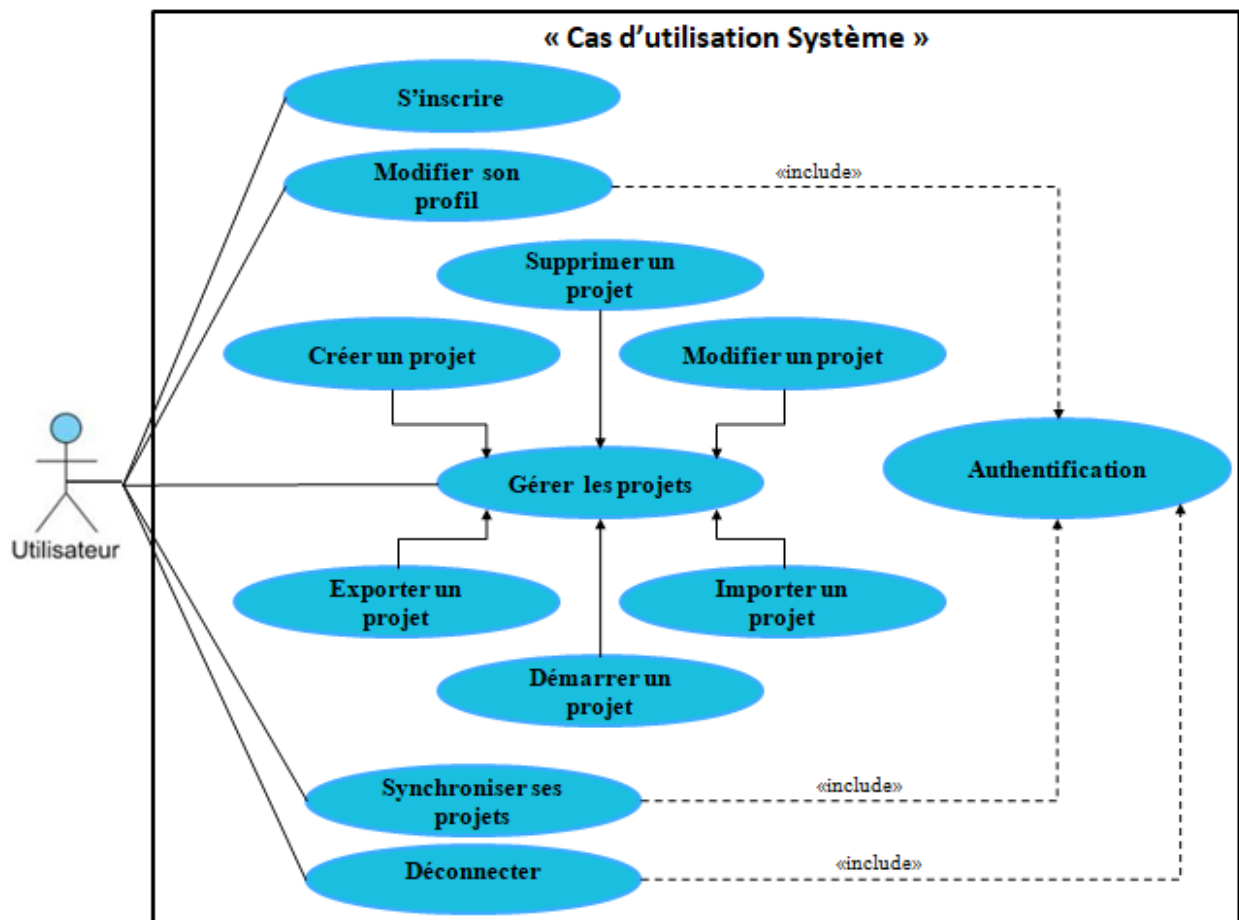


Figure 9 : Diagramme global des cas d'utilisation

3.2. Les différents cas d'utilisation

L'étude de cas d'utilisation a pour objectif de déterminer ce que chaque utilisateur attend du système. La détermination du besoin est basée sur la représentation de l'interaction entre l'acteur et le système.

A chaque cas d'utilisation, doit être associée une description textuelle des interactions entre l'acteur et le système et les actions que le système doit réaliser en vue de produire les résultats attendus par les acteurs. Pour exprimer les cas d'utilisations de notre système, nous avons choisi le formalisme suivant :

Numéro de cas d'utilisation	Nom de cas d'utilisation.
Résumé	But de cas d'utilisation.
Acteur	Acteurs participants au cas d'utilisation.

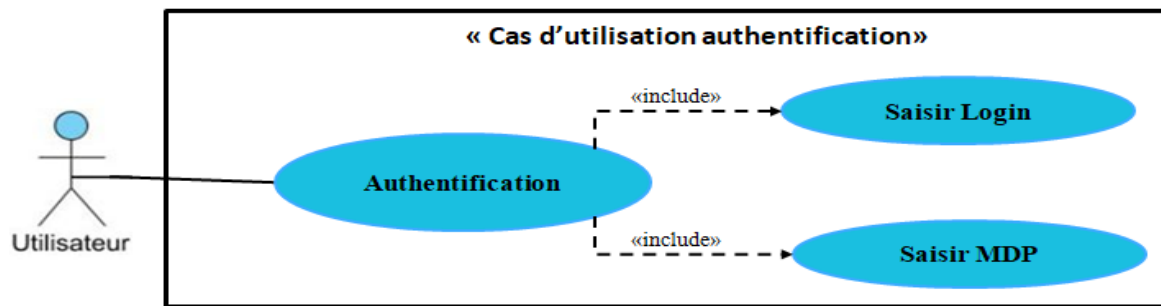
Précondition	Condition qui doit être remplie avant le début du cas d'utilisation.
Scénario nominal	Séquence d'action normale associée au cas d'utilisation.
Alternative Exception	Séquence d'action alternative pouvant conduire également un à un succès. Séquence d'action conduisant à un échec.

Tableau 2 : Le formalisme de description des cas d'utilisation**3.2.1. Le cas d'utilisation Authentification :**

Numéro de cas d'utilisation	Authentification
Résumé	Vérification de l'identité des utilisateurs (Login et mot de passe).
Acteur	Utilisateur
Précondition	L'utilisateur doit avoir un compte.
Scénario nominal	<p>[Début]</p> <ul style="list-style-type: none"> • Demande de connexion; • Le système affiche le formulaire d'authentification (demande le login et le mot de passe); • L'utilisateur saisit son login et son mot de passe puis valide; • Le système vérifie la conformité des informations fournies A1¹. • Le système donne l'accès à l'interface correspondante. <p>[Fin]</p>
Alternative Exception	Dans le cas où les informations fournies incomplète ou incorrectes le système réaffiche le formulaire d'authentification et attend que l'utilisateur ressaisisse ses informations.

Tableau 3 : Description du cas d'utilisation Authentification

¹ A1 (alternative) est une étiquette utilisée dans le cas où les informations saisies sont incomplètes ou incorrectes pour que le système réaffiche le formulaire d'authentification.

➤ **Diagramme du cas d'utilisation Authentification :****Figure 10 : Diagramme du cas d'utilisation Authentification****3.2.2. Le cas d'utilisation Inscription / Enregistrement:**

Numéro de cas d'utilisation	Inscription / Enregistrement
Résumé	Permet à l'utilisateur de s'inscrire.
Acteur	Utilisateur
Précondition	Aucune
Scénario nominal	[Début] <ul style="list-style-type: none"> • Demande de formulaire d'inscription; • Le système affiche le formulaire d'inscription (demande le mot de passe, nom, email); • L'utilisateur saisit ses informations puis valide; • Le système vérifie la conformité des informations fournies A1. [Fin]
Alternative Exception	Dans le cas où les informations fournies incomplète ou incorrectes le système réaffiche le formulaire d'inscription et attend que l'utilisateur ressaisisse ses informations.

Tableau 4 : Description du cas d'utilisation Inscription/Enregistrement

➤ Diagramme du cas d'utilisation Inscription :

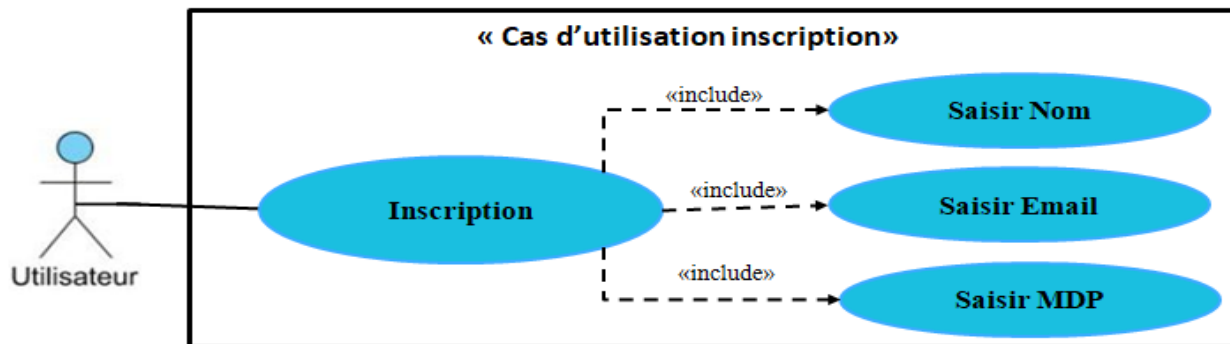


Figure 11 : Diagramme du cas d'utilisation Inscription

3.2.3. Le cas d'utilisation Déconnexion :

Numéro de cas d'utilisation	Déconnexion
Résumé	Permet à l'utilisateur de quitter son compte.
Acteur	Utilisateur
Précondition	Authentification
Scénario nominal	[Début] <ul style="list-style-type: none"> • Accès à l'application ; • Accès à mon profile ; • Cliquer sur le bouton « LogOut ». [Fin]

Tableau 5 : Description du cas d'utilisation Déconnexion

➤ Diagramme du cas d'utilisation Déconnexion :

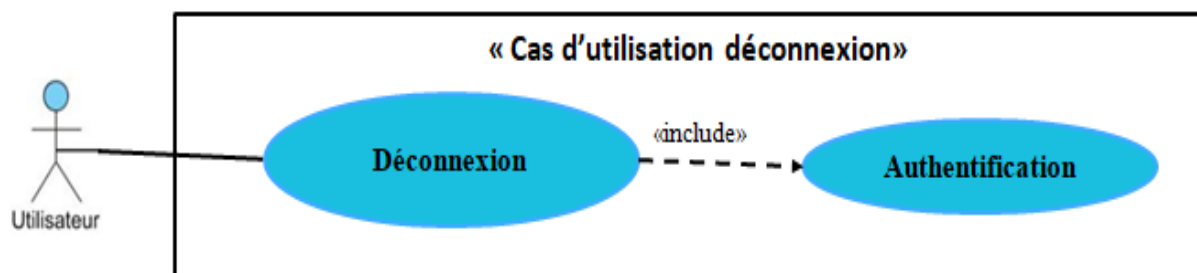


Figure 12 : Diagramme du cas d'utilisation Déconnexion

3.2.4. Le cas d'utilisation Créer un projet :

Numéro de cas d'utilisation	Créer un projet
Résumé	L'utilisateur a le privilège de créer des nouveaux projets.
Acteur	Utilisateur
Précondition	Aucune
Scénario nominal	<p>[Début]</p> <ul style="list-style-type: none"> • Accéder a l'application; • Cliquer sur le bouton "New Project" • Le système s'affiche l'interface de création de projet; • L'utilisateur créer son projet puis « Save » ; • Le système s'affiche le champ de nom de projet; • L'utilisateur saisir le nom de son projet puis « Save » ; <p>[Fin]</p>

Tableau 6 : Description du cas d'utilisation Créer un projet

➤ Diagramme du cas d'utilisation Créer un projet:

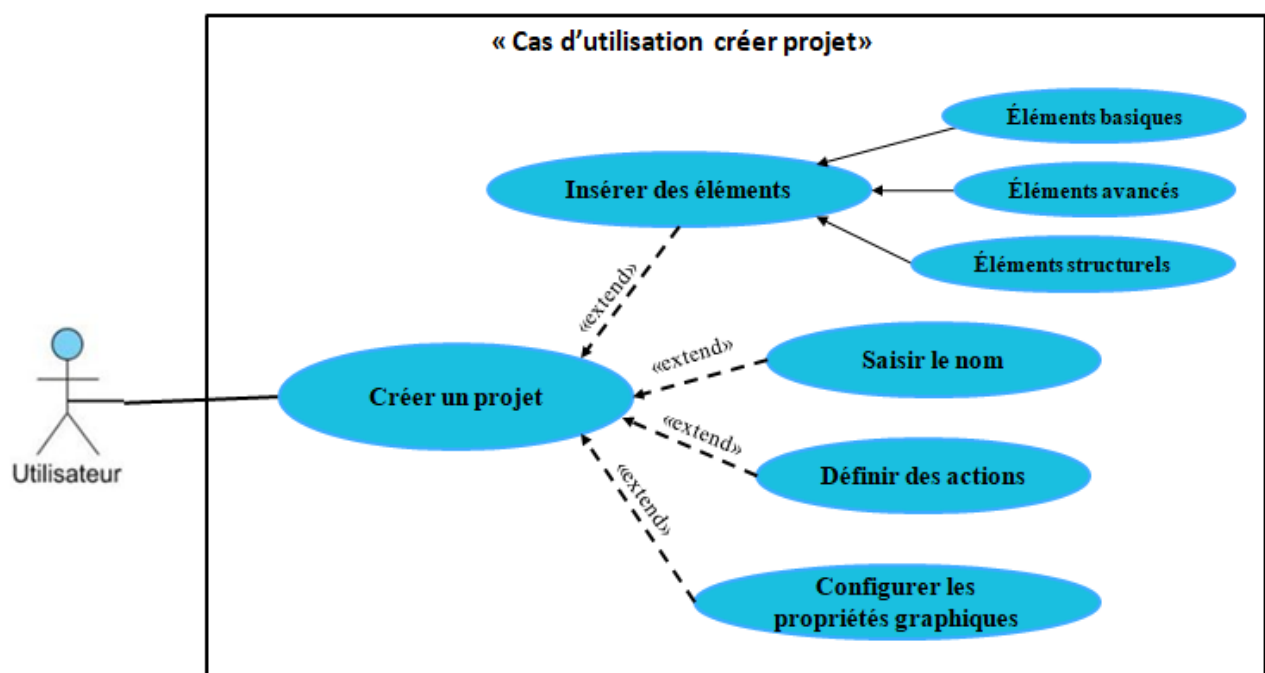


Figure 13 : Diagramme du cas d'utilisation Créer un projet

3.2.5. Le cas d'utilisation Modifier le profil :

Numéro de cas d'utilisation	Modifier le profil
Résumé	Permet à l'utilisateur de modifier son profile.
Acteur	Utilisateur
Précondition	Authentification
Scénario nominal	<p>[Début]</p> <ul style="list-style-type: none"> • Accès à l'application ; • Accès au « Menu » de l'application ; • Cliquer sur le bouton "connecter"; • L'application renvoie automatiquement l'interface; • L'utilisateur saisit les modifications puis valide; • Le système vérifie la conformité des informations fournies A1. <p>[Fin]</p>
Alternative Exception	Dans le cas où les informations fournies incomplète ou incorrectes le système réaffiche le formulaire et attend que l'utilisateur ressaisisse ses informations.

Tableau 7 : Description du cas d'utilisation Modifier le profil

➤ **Diagramme du cas d'utilisation modifier le profil :**

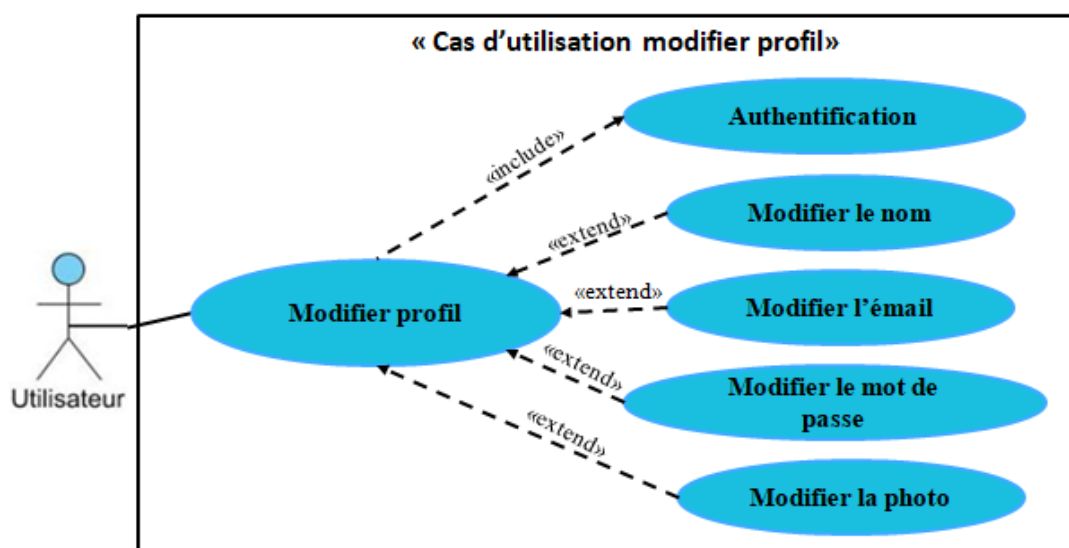


Figure 14 : Diagramme du cas d'utilisation Modifier le profil

3.2.6. Le cas d'utilisation Modifier un projet:

Numéro de cas d'utilisation	Modifier un projet
Résumé	Permet à l'utilisateur de modifier son projet.
Acteur	Utilisateur
Précondition	Aucune.
Scénario nominal	<p>[Début]</p> <ul style="list-style-type: none"> • Accès à l'application ; • Choisir le projet à modifier puis cliquer sur le bouton «Modifier »; • L'application renvoie automatiquement l'interface; • L'utilisateur saisit les modifications ; • puis cliquer sur le bouton « Save ». <p>[Fin]</p>

Tableau 8 : Description du cas d'utilisation Modifier un projet

➤ Diagramme du cas d'utilisation modifier un projet:

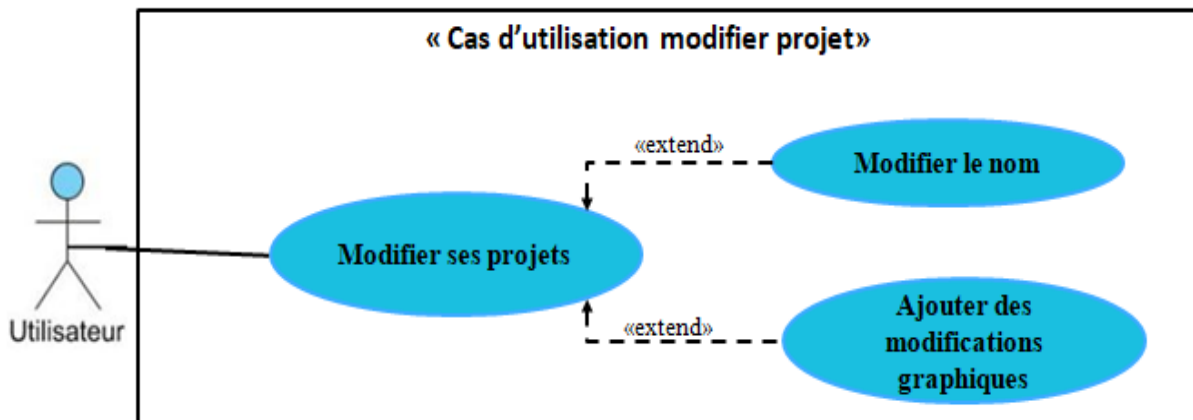


Figure 15 : Diagramme du cas d'utilisation Modifier un projet

3.2.7. Le cas d'utilisation Synchronisation :

Numéro de cas d'utilisation	Synchronisation
Résumé	Permet à l'utilisateur de synchroniser ses projets avec le serveur.
Acteur	Utilisateur
Précondition	Authentification
Scénario nominal	<p>[Début]</p> <ul style="list-style-type: none"> • Accès à l'application ; • Accès au « Menu » de l'application ; • Cliquer sur le bouton "Synchroniser"; • Le système va automatiquement synchroniser les projets avec le serveur. <p>[Fin]</p>

Tableau 9 : Description du cas d'utilisation Synchronisation

➤ Diagramme du cas d'utilisation Synchronisation:

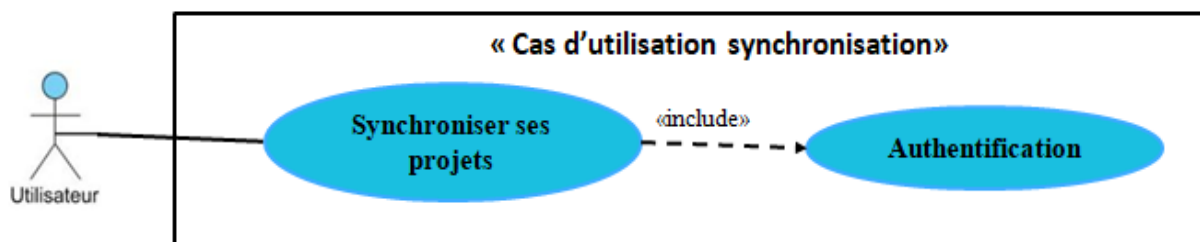


Figure 16 : Diagramme du cas d'utilisation Synchronisation

4. Diagramme de classes

4.1. Définition de diagramme de classes

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Il est important de noter qu'un même objet peut très bien intervenir dans la

réalisation de plusieurs cas d'utilisation. Les cas d'utilisation ne réalisent pas une partition des classes du diagramme de classes.

Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application. Chaque langage de Programmation orienté objet donne un moyen spécifique que d'implémenter le paradigme objet (pointeurs ou pas, héritage multiple ou pas, etc.), mais le diagramme de classes permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier.

Les principaux éléments de cette vue statique sont les classes et leurs relations : association, généralisation et plusieurs types de dépendances, telles que la réalisation et l'utilisation. [10]

4.2. Diagramme de classes de l'application

Comme le montre la figure 17 ci-dessous, le diagramme de classes de l'application admet six classes qui sont User, Project, Configuration, Element, Property et ActionInterface.

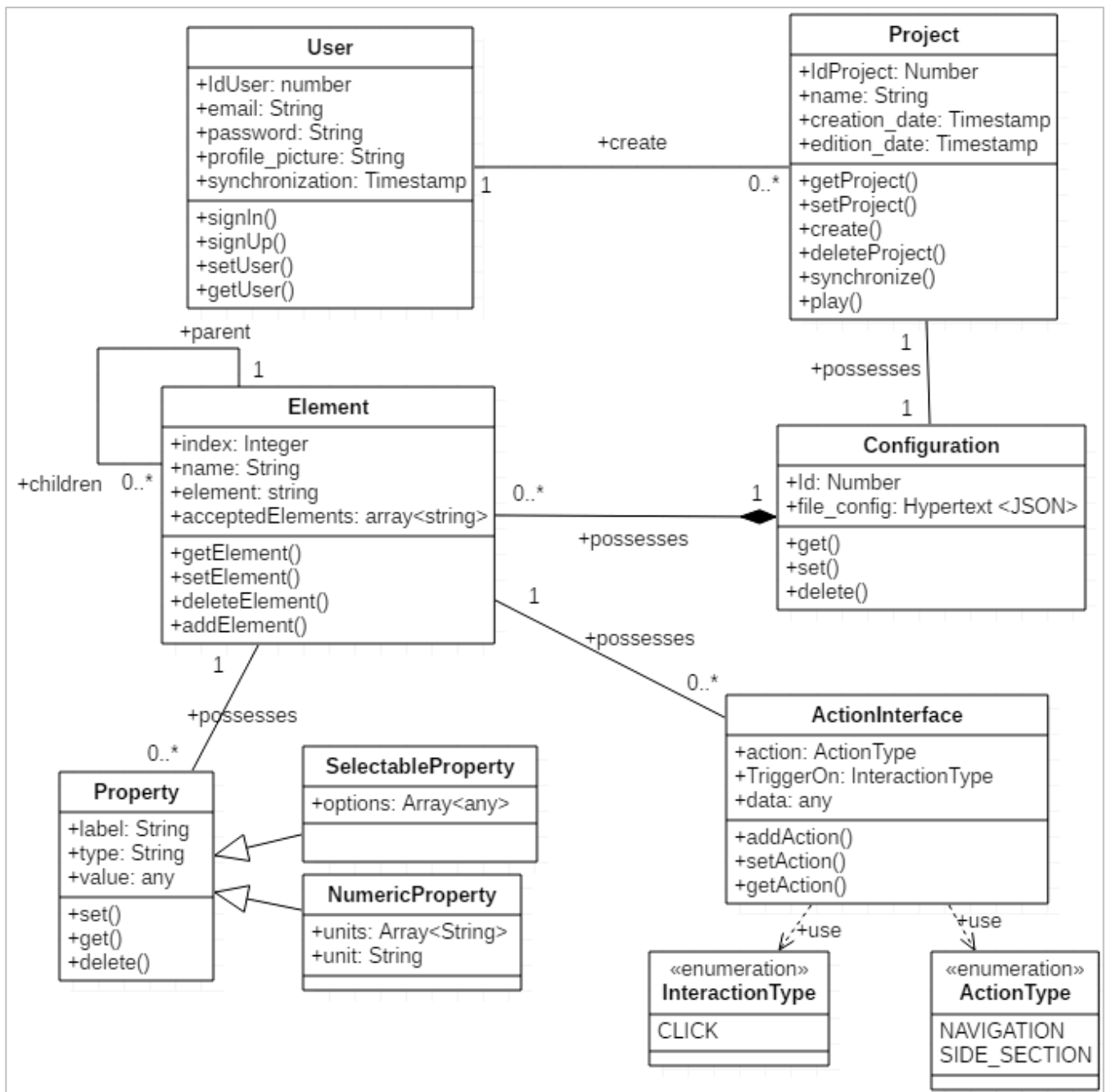


Figure 17 : Diagramme de classes

5. Diagrammes de séquences

5.1. Définition de diagrammes de séquences

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML. [11]

Dans ce qui suit, nous présentons le diagramme de séquence selon le diagramme de cas d'utilisation dans notre système.

5.2. Les différents diagrammes de séquences

5.2.1. Diagramme de séquences relatif à « S'authentifier »

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation «S'authentifier». Dans un premier lieu, l'utilisateur remplit le formulaire d'authentification et clique sur le bouton « Sign In ». Ensuite le système vérifie s'il s'agit bien d'un utilisateur ou pas à travers le web service « Authentification ».

La figure 18 ci-dessous illustre une description détaillée du scénario relatif au cas d'utilisation «S'authentifier ».

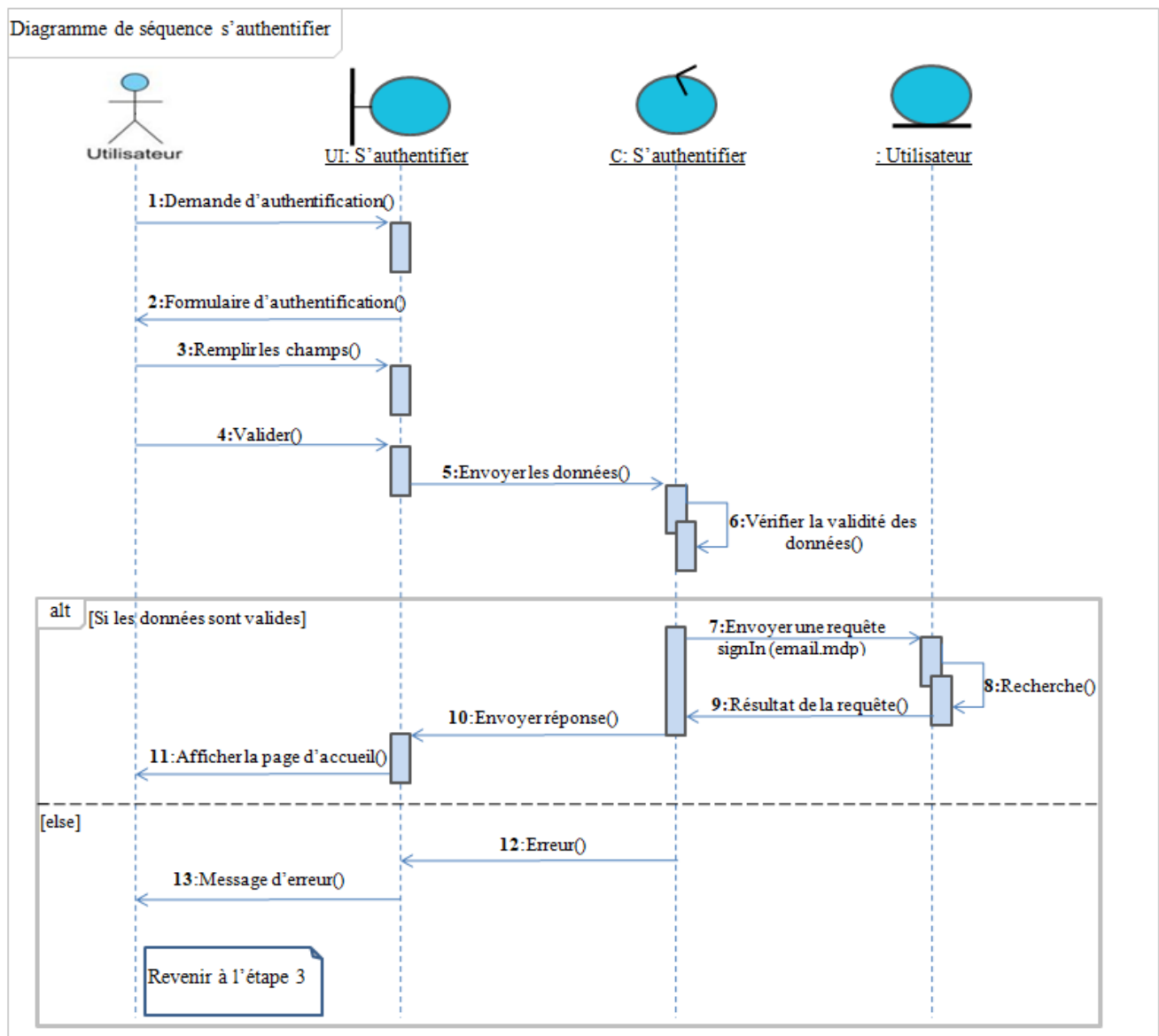


Figure 18 : Diagramme de séquences relatif à « S'authentifier »

5.2.2. Diagramme de séquences relatif à « S'inscrire »

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation « S'inscrire ».

Dans un premier lieu, l'utilisateur demande au système de faire une inscription. Une fois le formulaire est affiché, il remplit les champs de saisies puis il enregistre ses données. Le système s'assure d'abord que les champs obligatoires n'ont pas la valeur NULL. Ensuite, il enregistre les informations entrées dans la base de données.

La figure 19 ci-dessous illustre une description détaillée du scénario relatif au cas d'utilisation « S'inscrire ».

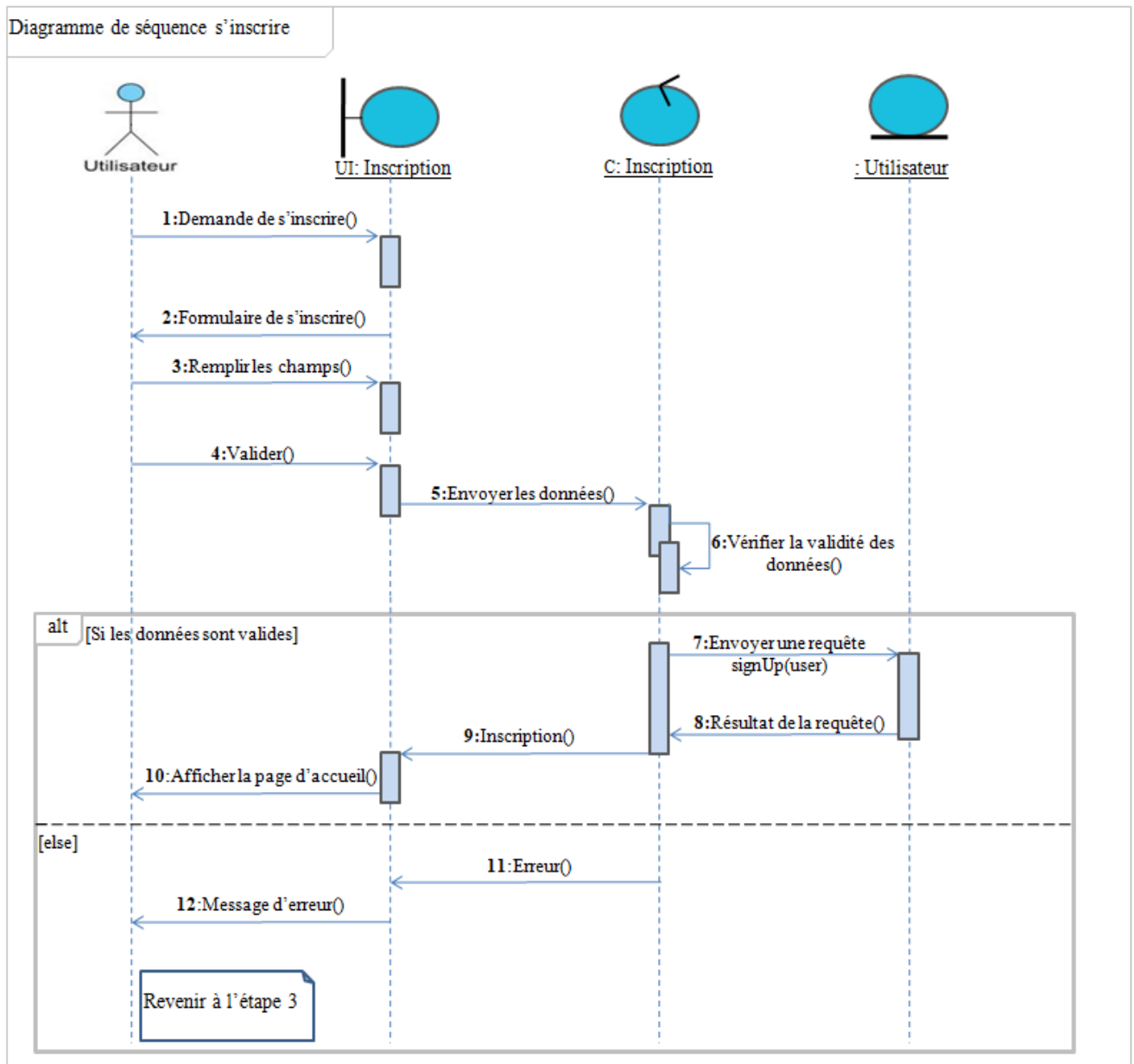


Figure 19 : Diagramme de séquences relatif à « S'inscrire »

5.2.3. Diagramme de séquences relatif à « Modifier Profil » :

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation « Modifier Profil ». Dans un premier lieu, l'utilisateur remplit le formulaire de modification profil. Ensuite, le système vérifie la saisie et enregistre les nouvelles données à travers le web service «updateProfile».

La figure 20 ci-dessous illustre une description détaillée du scénario relatif au cas d'utilisation «Modifier Profil».

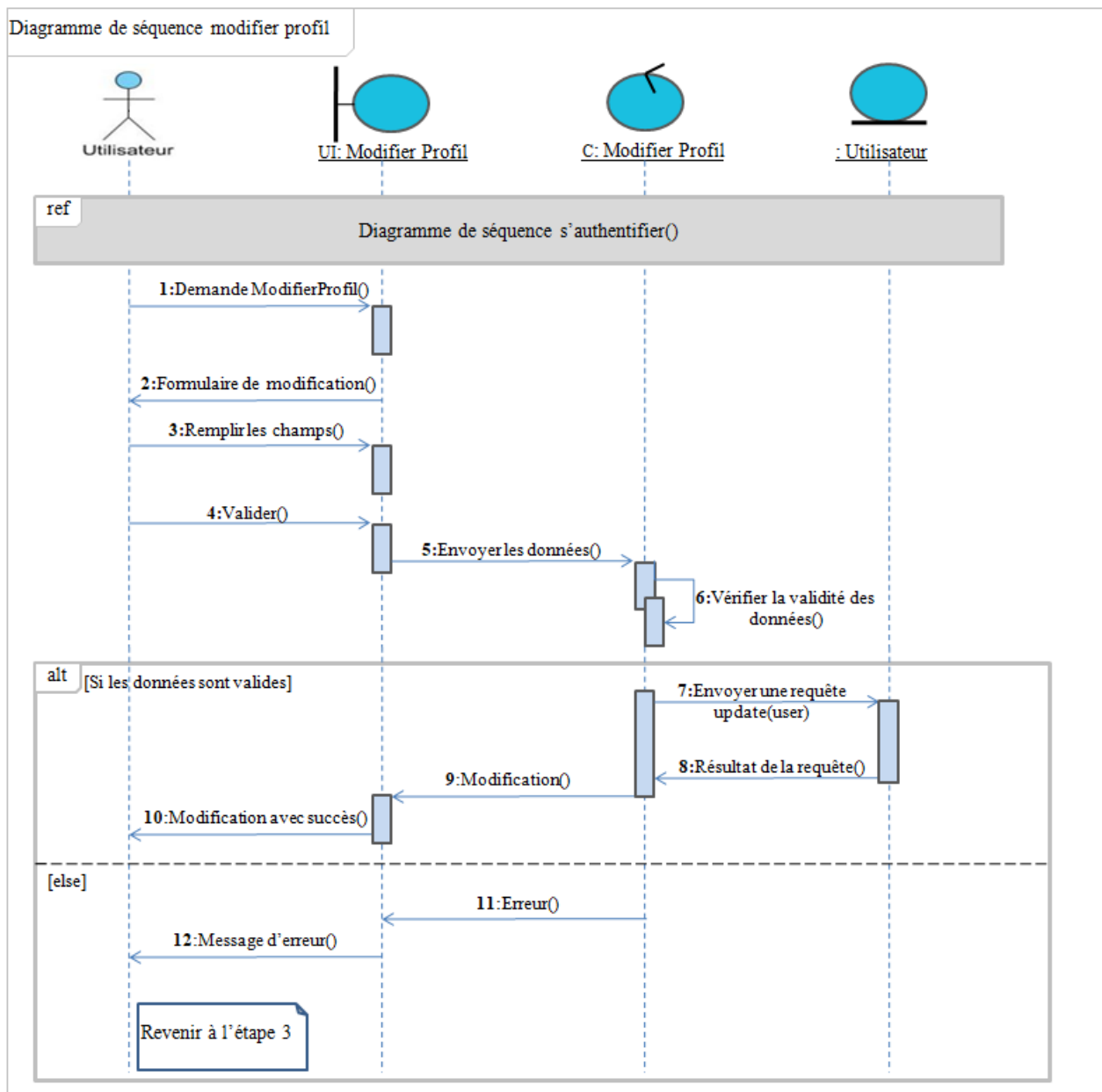


Figure 20 : Diagramme de séquences relatif à « Modifier Profil »

5.2.4. Diagramme de séquences relatif à « Créer un Projet »

A travers ce diagramme, nous allons décrire les scénarios du cas d'utilisation «Créer un Projet». Dans un premier lieu, l'utilisateur clique sur l'item « New Project » à partir du menu principal depuis l'interface graphique, alors l'application renvoi l'interface de création de projet.

La figure 21 ci-dessous illustre une description détaillée du scénario relatif au cas d'utilisation «Créer un Projet».

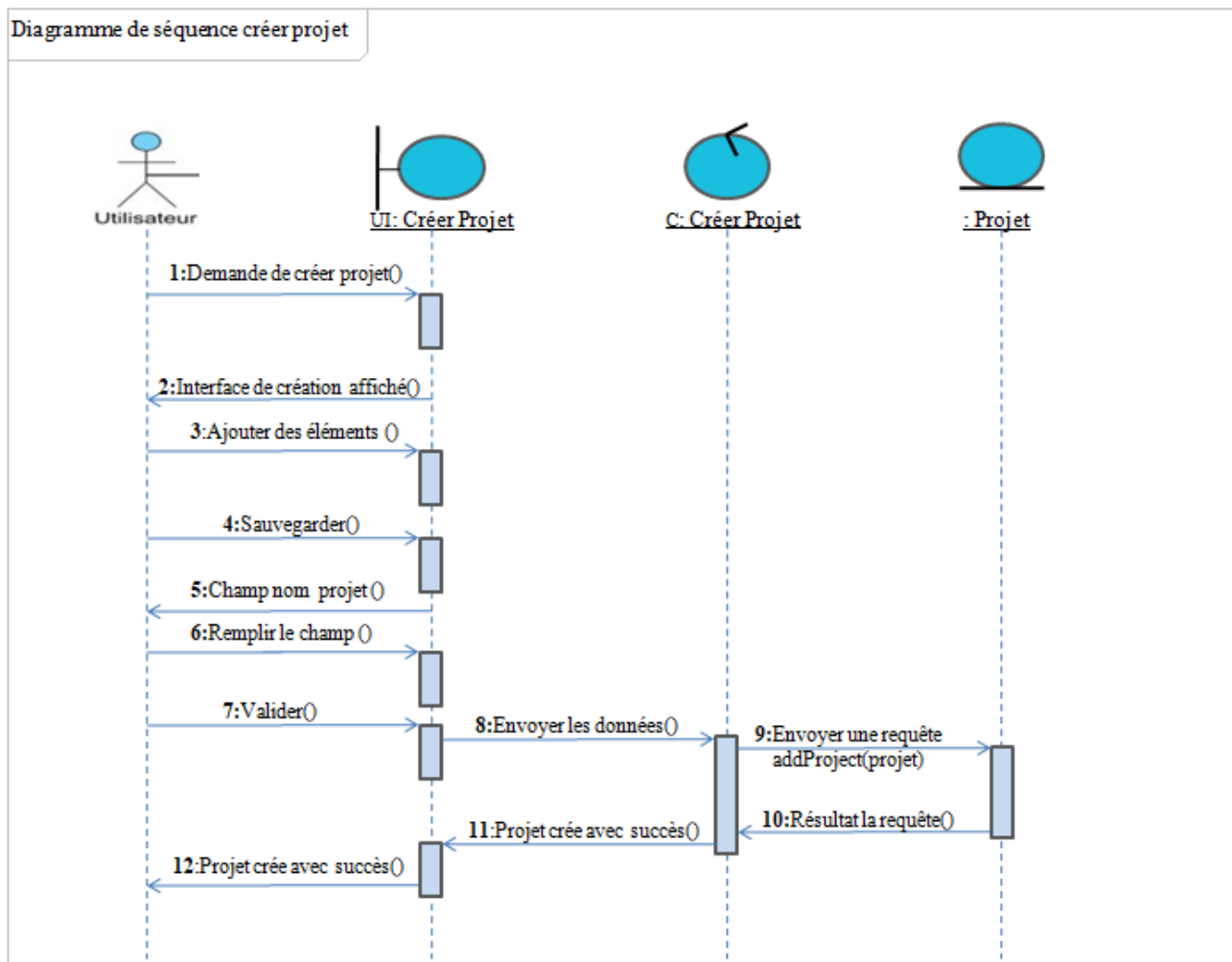


Figure 21 : Diagramme de séquences relatif à «Créer un projet»

5.2.5. Diagramme de séquences relatif à « Modifier Projet»

A travers ce diagramme, nous allons décrire les scénarios du cas d'utilisation «Modifier Projet». Dans un premier lieu, l'utilisateur choisit le projet à modifier à partir de l'interface d'accueil. Puis, il clique sur le bouton « Modify ». Ainsi, l'application renvoie l'interface de modification de projet. Par la suite, l'utilisateur ajoute les modifications et appuie sur le bouton « Save ».

La figure 22 ci-dessous illustre une description détaillée du scénario relatif au cas d'utilisation «Modifier Projet».

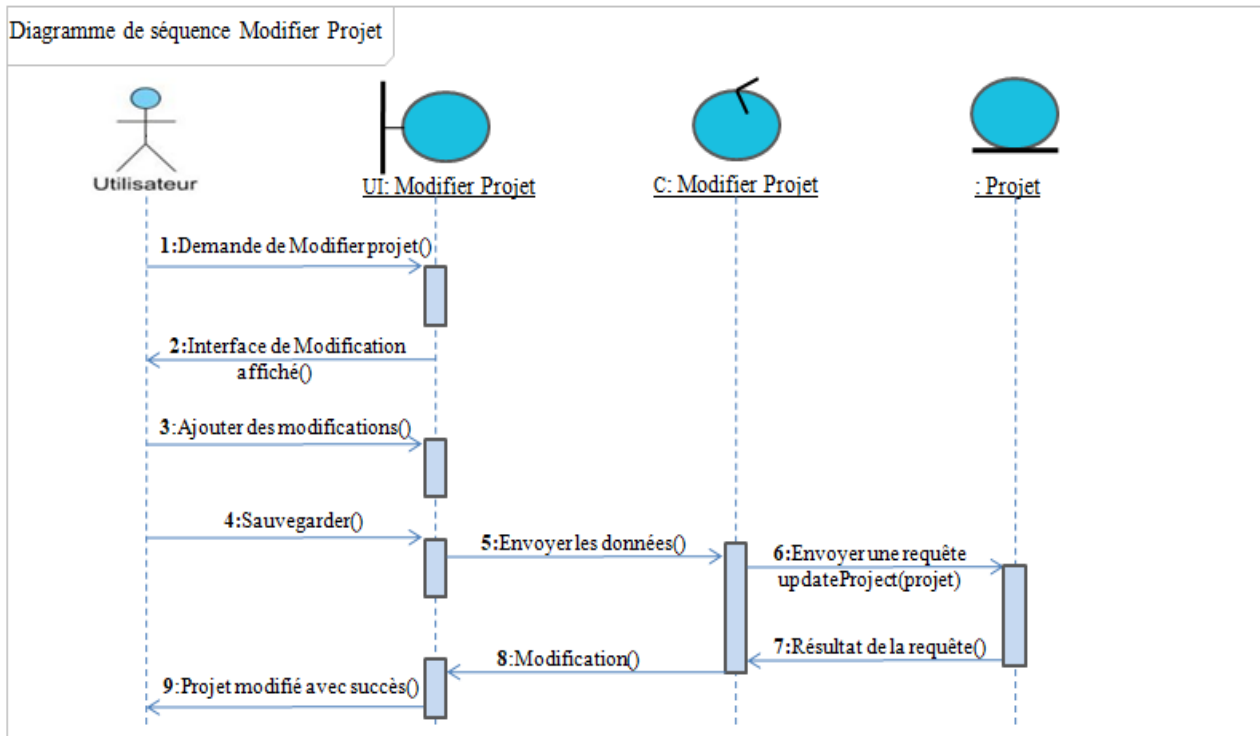


Figure 22 : Diagramme de séquences relatif à «Modifier Projet»

5.2.6. Diagramme de séquences relatif à « Synchronisation »

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation «Synchronisation». Dans un premier lieu, l'utilisateur clique sur l'item «Synchronisation » à partir du menu principal depuis l'interface graphique, alors le système synchronise les données locales avec le serveur.

La figure 23 ci-dessous illustre une description détaillée du scénario relatif au cas d'utilisation «Synchronisation».

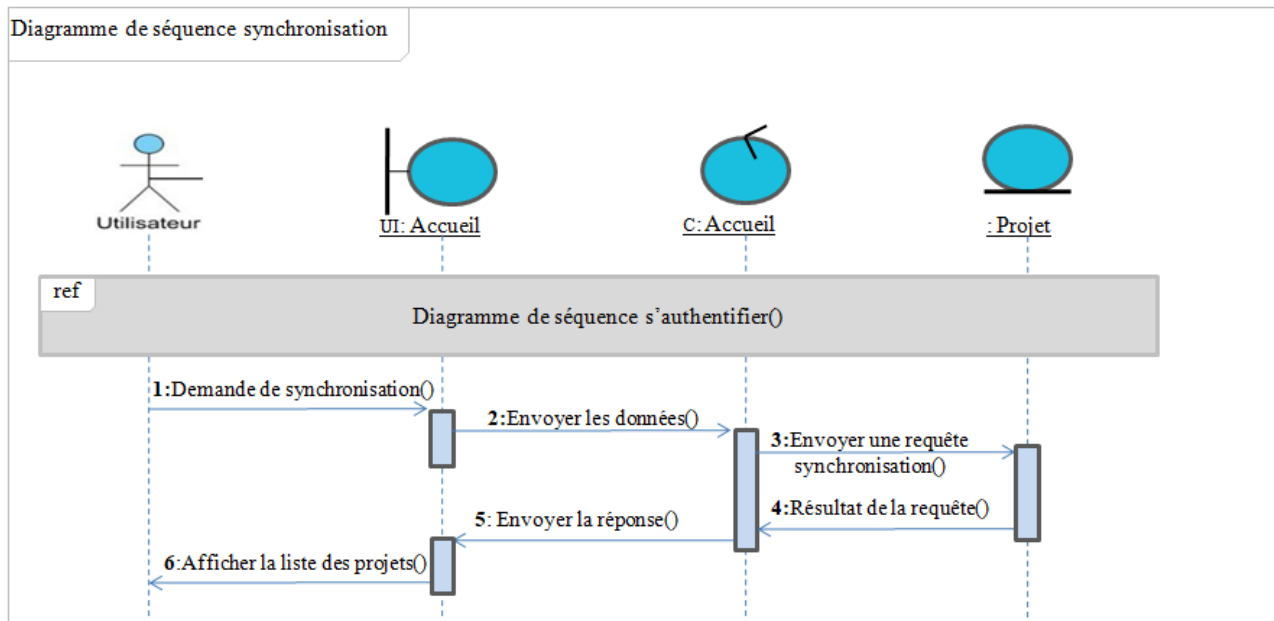


Figure 23 : Diagramme de séquences relatif à « Synchronisation »

Conclusion :

Tout au long de ce chapitre, nous avons détaillé la conception de notre application à travers le diagramme de cas d'utilisation et le diagramme de classes ainsi que le diagramme de séquences, afin que la phase de la réalisation et la mise en place de l'application soient plus souples et plus aisées. Le chapitre suivant mettra en évidence, le fruit de ce passage et les différents résultats du développement de l'application demandée.

Chapitre III :

Réalisation

Introduction

Pour pouvoir mener à bien un projet informatique, il est nécessaire de choisir des technologies permettant de simplifier sa réalisation. Pour cela, après avoir complété l'étude conceptuelle dans le chapitre précédent, nous allons aborder la partie implémentation dans ce qui suit. Nous allons commencer par présenter l'environnement matériel et logiciel, et ensuite, l'état de réalisation.

I. Environnement du travail

Le choix de système Windows et les programmes de développement est requis pour réaliser notre application. Ce chapitre couvre les prés requis matériels et logiciels qui sont utilisés pour la réalisation de celle-ci.

1. Environnement matériel

Pour la réalisation de notre application, nous avons utilisé :

- Un pc pour le développement ayant les caractéristiques suivantes :
 - Intel core i5 2.40 GHz.
 - 8 Go de mémoire vive.
 - Windows 10 - 64 bits
- Un Smartphone Condor A8, un Mac et une Tablette pour réaliser les tests.

1.1. Choix de l'architecture de l'application

L'architecture 3-tiers est un modèle logique d'architecture applicative qui s'est imposé lors de la phase de conception, non seulement grâce à ses nombrables avantages, mais notamment parce qu'elle est adaptée à notre cas de figure. La séparation, claire et nette entre les trois couches logicielles d'une même application ou système, aide à avoir une idée globale en ce qui concerne les principaux package de l'outil à concevoir. Comme décrit précédemment, cette architecture nous permet de diviser l'application en 3 sous-parties, qui sont :

- **Présentation des données :** La couche « Présentation des données » est la partie qui est immédiatement visible pour l'utilisateur. Elle prend en charge tout ce qui concerne l'interface graphique et l'interaction homme machine. Cette interface peut prendre de

multiples facettes sans changer la finalité de l'application. La couche présentation relaie les requêtes de l'utilisateur à destination de la couche métier, et en retour lui présente les informations renvoyées par les traitements de cette couche. Il s'agit donc ici d'un assemblage de services métiers et applicatifs offerts par la couche inférieure.

- **Logique métier** : La couche « Logique métier » est le cœur de l'application. Elle correspond à la partie fonctionnelle de l'application et décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche présentation.

Les différentes règles de gestion et de contrôle du système sont mises en œuvre dans cette couche. La couche métier offre des services applicatifs et métier à la couche présentation. Pour fournir ces services, elle peut s'appuyer sur les données du système, accessibles au travers la couche accès aux données.

- **Accès aux données** : La couche « Accès aux données » : gère la permanence des données et a pour but de conserver une quantité plus ou moins importantes de données de façon structurée. On peut utiliser pour cette partie des systèmes de bases de données relationnelles permettent le partage des données qu'ils contiennent via un réseau. Pour résumer notre choix en matière d'architecture système qui vient suite à une étude stratégique, on peut résumer et dire que l'architecture système 3-tiers a été choisie pour :

- Avoir les meilleures performances, étant donné le partage des tâches entre les différents serveurs.
- Avoir une aisance lors de la conception. Une conception qu'on veut claire et efficace.
- Plus de flexibilité dans l'allocation des ressources et dans les requêtes du client vers le serveur. Le client qui n'a donc que des fonctions d'affichage ne fait que des requêtes vers le serveur, aucun calcul n'est effectué par le client. Les résultats de ses requêtes sont ensuite affichés. [12]

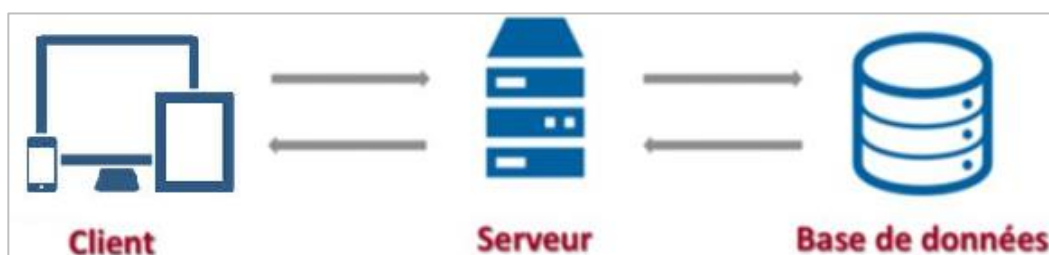


Figure 24 : Architecture 3-tiers

1.2. Elaboration du diagramme de déploiement

Les diagrammes de déploiement modélisent l'architecture physique d'un système et affichent les relations entre les composants logiciels et matériels, d'une part, et la distribution physique du traitement, d'autre part. [13]

L'utilisateur de l'application demande une ressource auprès de son équipement. Ceci se traduit par une requête HTTP GET, POST, DELETE ou PUT ou qui sera envoyée à travers des web services au serveur de la base de données. La réponse finale est renvoyée au serveur de l'application sous forme d'un fichier JSON.

Le diagramme de déploiement de notre application est représenté ci-dessous :

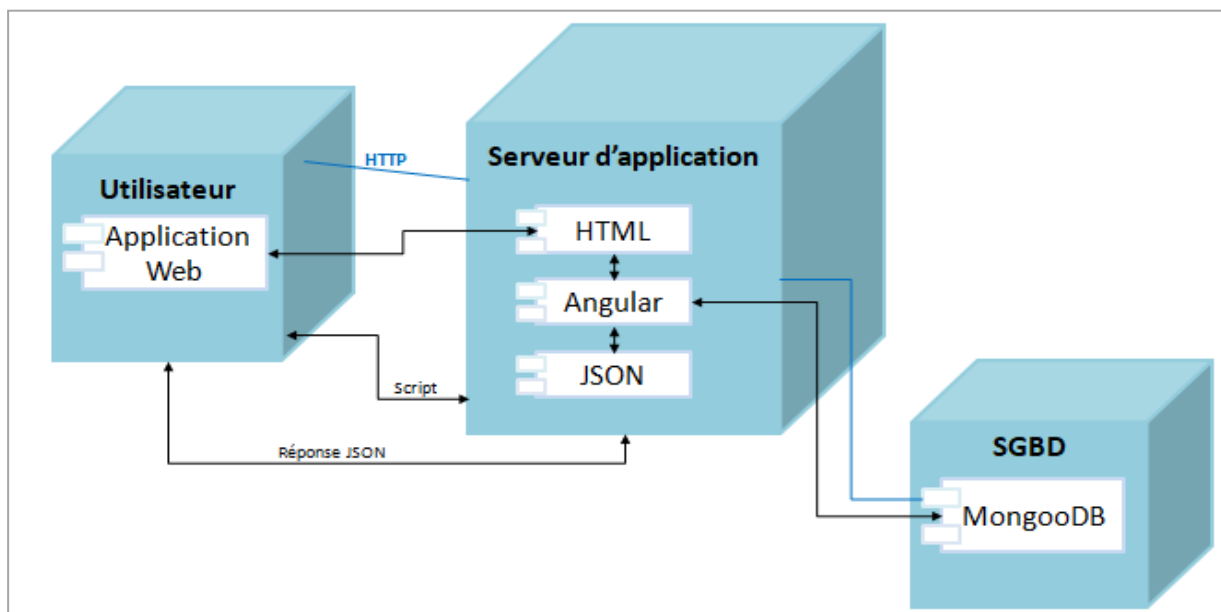


Figure 25 : Diagramme de Déploiement

1.3. Choix de type de l'application

Les Applications Web Progressives (PWA) peuvent être exécutées sur n'importe quel terminal, et ont la capacité de s'adapter totalement à l'écran sur lequel elles seront lues. Ce fonctionnement présente deux avantages majeurs.

Le premier est évidemment que la PWA est développée avec les technologies du web. L'offre de compétences dans les technologies web est supérieure à la demande. Cela coûte globalement moins cher de développer une PWA qu'une application native, d'autant que si on

veut cibler iOS, Android et desktop, il faudra réaliser trois développements natifs distincts. Le deuxième avantage repose sur le fait que les PWAs s'exécutent dans une application universelle, le navigateur web. Tous les équipements connectés à internet en possèdent un, aussi bien mobile, tablette que desktop. Encore une fois, en ne développant plus qu'une seule application, et cela permet de rationaliser les coûts.

Vu les avantages précités des PWA, nous avons utilisé le Service Worker qui est un Proxy programmable asynchrone, dans le navigateur. Il intercepte toutes les requêtes afin de les exploiter et d'y appliquer des règles de caches, exécutées dans un thread séparé. En schématisant, cela donne :

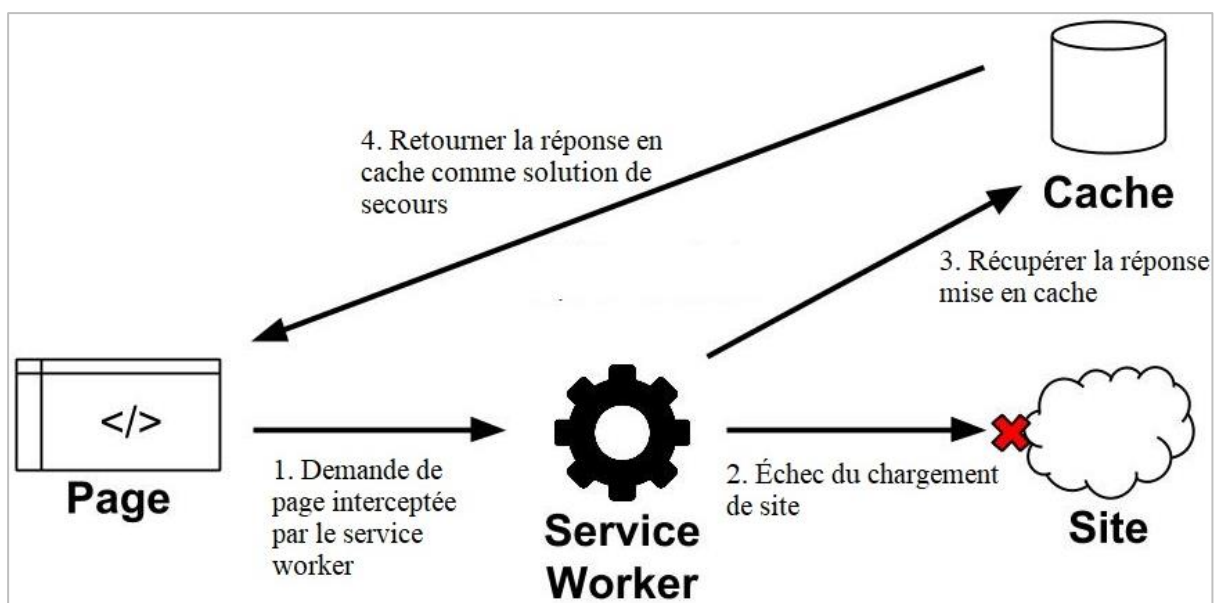


Figure 26 : Architecture de service Worker

Le service Worker va souvent prendre la forme d'un petit fichier JavaScript et joue le rôle de proxy entre notre application, le réseau et le navigateur. C'est un peu le chef d'orchestre de la synchronisation des données entre ces trois éléments. Aujourd'hui, son développement est quasi-uniquement concentré sur le mobile. En effet, c'est sur mobile que l'on va retrouver des connexions à vitesse variable et surtout des déconnexions intempestives.

Il va par exemple intercepter les requêtes réseau et choisir quelles sont les actions à réaliser en fonction de la connexion. S'il n'y a pas de réseau, la requête demandée par l'utilisateur, comprendre l'affichage de la page par exemple, va être exécutée en local avec les données disponibles sur l'appareil.

C'est aussi lui va gérer le pré-chargement des informations essentielles. Utile pour assurer une réponse à l'utilisateur en cas de perte de réseau par exemple. Le service Worker va donc devenir indispensable pour concevoir des applications « offline first ».

1.4. Processus de synchronisation

La synchronisation est une action ou une étape qui consiste à dupliquer les données stockées en local sur un serveur dans le but de les faire correspondre au fil du temps.

Ainsi, toutes informations sont constamment mises à jour, et nous pourrons y accéder avec tous les outils que nous utilisons à condition d'être reliés au même serveur par internet généralement.

La synchronisation se présente comme un processus qui permet de faire correspondre, c'est-à-dire conserver à l'identique, les contenus de deux ou plusieurs emplacements de stockage. Ainsi, lorsqu'un utilisateur ajoute, supprime ou modifie un projet depuis un ordinateur, une Smart TV, une tablette ou encore un Smartphone, tous ces ajouts, suppressions et modifications seront directement répercutés sur les autres supports gérés par le processus de synchronisation, après qu'il passe nécessairement par l'étape d'authentification à l'application.

Concrètement comme indique la figure 27 ci-dessous, notre application fonctionne toujours en mode local avec lequel les données seront stockées dans la base de données locale IndexedDB. Mais, sauf à l'étape d'authentification, d'inscription et de synchronisation, l'application sera fonctionnée en mode distant. Dans ce cas, le service store va envoyer toutes les modifications de la base locale vers la base des données distante MongoDB à l'aide de l'api de synchronisation, à condition de présence d'une connexion internet.

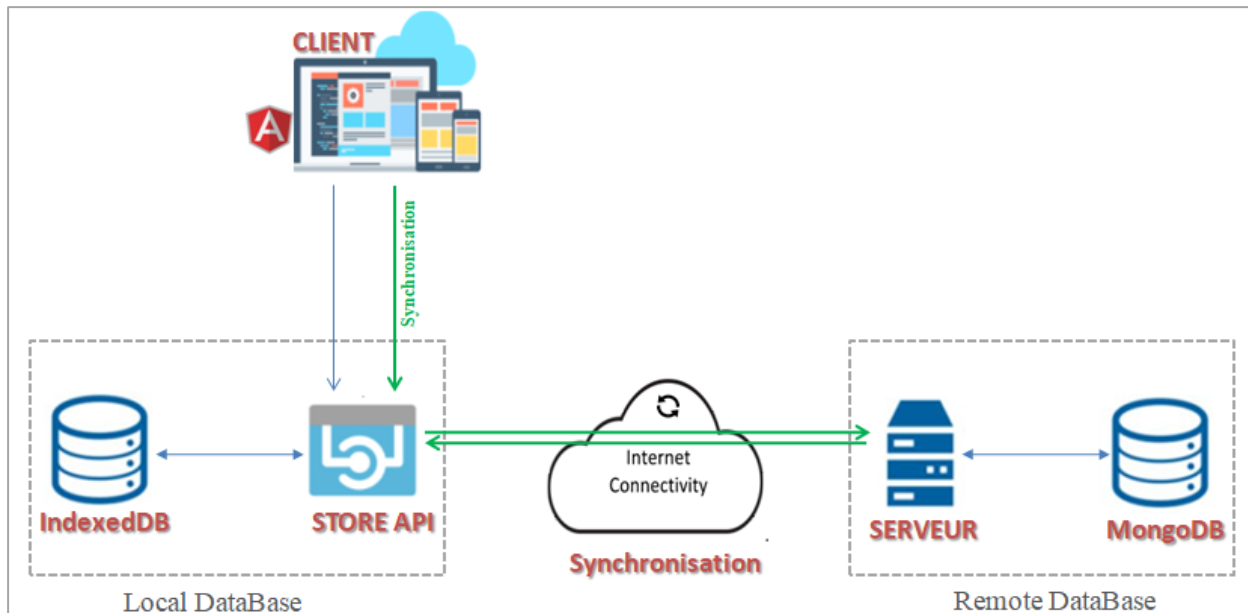


Figure 27 : Processus de synchronisation

2. Environnement logiciel

2.1. Les Framework

Angular 9 : Angular est un framework JavaScript Open Source développé par Google.

Il utilise l'architecture MVC. Cela va permettre de structurer son code et bien séparer la vue (l'interface) des modèles (fonctionnement). Il est considéré comme un langage « côté client », ceux-ci permettent de gérer l'interface utilisateur de chaque page (affichage, interactions...) de façon dynamique et viennent en complément aux langages côté serveur. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Angular donne aussi la possibilité de développer des programmes pour téléphones portables. [14]



Figure 28 : Logo Angular

- **NestJs** : Nestjs est un framework basé sur nodejs et qui nous permet de créer notre backend capable d'accepter n'importe quel client comme Angular et Vue. Nestjs est 100% Typescript et apporte donc le typage, la notion de classe et la programmation

orientée objet dans javascript. Comme la plupart des frameworks pour le backend , Nestjs utilise Expressjs comme serveur web. [15]



Figure 29 : Logo NestJS

2.2. Les langages

- **HTML** : L'HyperText Markup Language, généralement abrégé HTML, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS). [16]



Figure 30 : Logo HTML

- **TypeScript** : est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. C'est un sur-ensemble de JavaScript (c'est-à-dire que tout code JavaScript correct peut être utilisé avec TypeScript). [17]



Figure 31 : Logo TypeScript

- **Sass** : (Syntactically Awesome Stylesheets) est un langage dynamique de génération de feuilles de style en cascade initialement développé par Hampton Catlin et Nathalie Weizenbaum. Sass est un préprocesseur CSS. C'est un langage de description compilé en CSS. SassScript est un langage de script pouvant être utilisé à l'intérieur du code Sass. Deux syntaxes existent. La syntaxe originale, nommée « syntaxe indentée », est proche de Hamlet. La nouvelle syntaxe se nomme SCSS. Elle a un formalisme proche de CSS. il peut être étoffé avec Compass (pratique pour les préfixes des différents navigateurs). [18]



Figure 32 : Logo Sass

2.3. Les outils

- **MongoDB** : est une base de données orientée documents NoSQL, à l'inverse de MySQL. Ce système de gestion fonctionne différemment en matière de mémorisation des données, bien que la structure de base présente quelques similitudes :
- Les données sont mémorisées dans des classeurs (collections) et non dans des tableaux.
 - Les documents en format BSON remplacent les lignes et les champs définis dans ces documents et les colonnes de requêtes SQL.
 - Un champ est toujours composé d'une valeur et d'un nom de champ. Cette valeur peut indiquer des listes de mots ou de chiffres, un texte ou un dossier complet.
 - Un document MongoDB est donc un ensemble de paires valeur/clés, tout comme avec MySQL. [19]



Figure 33 : Logo mongoDB

- **IndexedDB** : IndexedDB est un système de gestion de base de données transactionnel, similaire à un SGBD relationnel basé sur SQL. Cependant contrairement aux SGBD relationnels, qui utilisent des tables avec des colonnes fixes, IndexedDB est une base de données orienté objet pour JavaScript. IndexedDB vous permet de stocker et de récupérer

des objets qui sont indexés avec une clef; tout objet supporté par l'algorithme de clonage structuré peut être stocké. Vous devez spécifier le schéma de la base de données, y ouvrir une connexion, puis récupérer et mettre à jour des données dans une série de transactions. Alors que le stockage DOM (Web Storage) est utile pour stocker de petites quantités de données, il est moins utile pour stocker de grandes quantités de données structurées. IndexedDB répond à ce besoin. Cette page est le point d'entrée pour tout ce qui concerne IndexedDB sur MDN. [20]



Figure 34 : Logo IndexedDB

- **Visual Code** : est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS, il est présenté lors de la conférence des développeurs Build d'avril 2015 comme un éditeur de code multiplateforme, open source et gratuit, supportant une dizaine de langages. [21]



Figure 35 : Logo Visual Code

- **StarUML** : est un logiciel de modélisation UML open source qui peut remplacer dans bien des situations des logiciels commerciaux et coûteux comme Rational Rose1 ou Together2 . Étant simple d'utilisation, nécessitant peu de ressources système, supportant UML 2, ce logiciel constitue une excellente option pour une familiarisation à la modélisation. Cependant, seule une version Windows est disponible. [22]



Figure 36 : Logo StarUML

- **Bitbucket** : est un service web d'hébergement et de gestion de développement logiciel utilisant le logiciel de gestion de versions Git. Il s'agit d'un service freemium dont la version gratuite permet déjà de créer jusqu'à un nombre illimité de dépôts privés, accessibles par cinq utilisateurs au maximum. [23]



Figure 37 : Logo Bitbucket

2.4. Les technologies utilisées

- **JSON** : « JSON (JavaScript Object Notation – Notation Objet issue de JavaScript) est un format léger d'échange de données. Il est aisément analysable ou générable par des machines. Il est basé sur un sous-ensemble du langage de programmation JavaScript. JSON est un format texte complètement indépendant de tout langage. » La figure 38 représente un exemple qui lit les données JSON à partir d'un serveur. [24]

```
{
  "project": [],
  "_id": "5e9f10d407f7a811e8e76410",
  "name": "Majdoub Baha Eddine",
  "login": "majdoubb53@gmail.com",
  "password": "GOOGLE101300174106920356302",
  "__v": 0
}
```

Figure 38 : Exemple JSON

- **REST API** : REST est l'acronyme de Representational State Transfer. REST décrit un style d'architecture logicielle permettant de construire une application devant fonctionner sur des systèmes distribués, typiquement internet. Quand un web service utilise l'architecture REST, on l'appelle « REST API ». Le principe de REST est d'utiliser HTTP pour l'implémentation d'un Web Service, non seulement comme simple protocole de transport, mais également pour définir l'API de chaque service, c'est à dire la définition même des messages entre clients et serveur. Pour interagir avec la base de données MongoDB, nous avons besoin premièrement de construire une API REST. L'intérêt d'une API REST est d'obtenir la demande du client, d'interagir avec la base de données et enfin donner la réponse au client. Une API effectue la démarche décrite ci-dessous :

- Accepter les demandes dans les méthodes GET ou POST ou PUT ou DELETE.
- Interagir avec la base de données en insérant ou en récupérant des données.
- Enfin donner une réponse de retour au format JSON.

La figure 39 ci-dessous explique cette démarche.

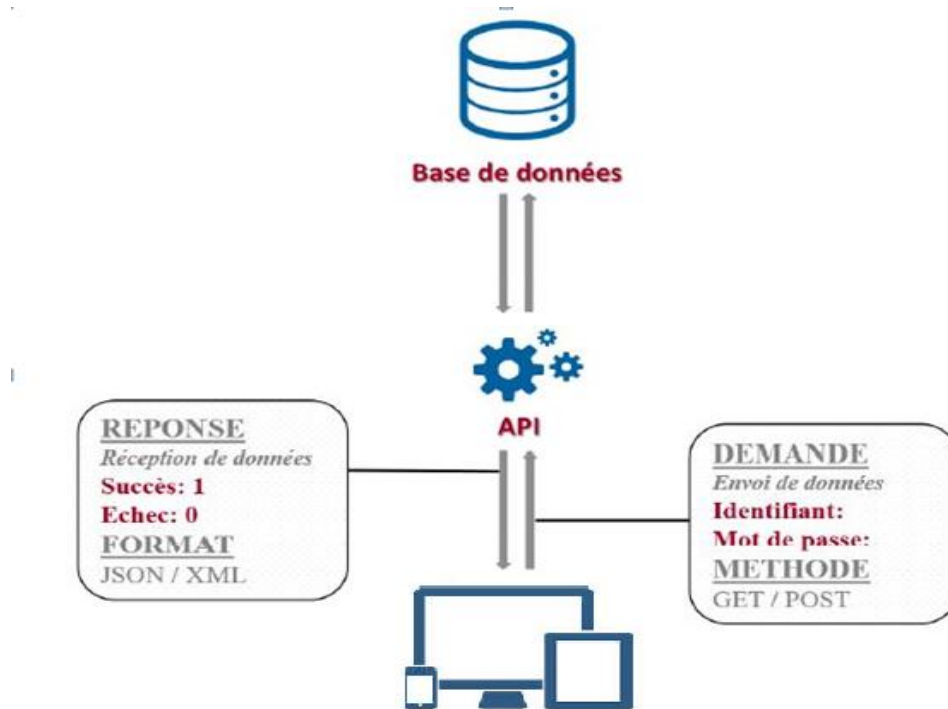


Figure 39 : REST API

API Facebook Login : Avec Facebook Login, les personnes peuvent rapidement et facilement créer des comptes et se connecter à votre app sur plusieurs plates-formes. Cette option est disponible pour les apps iOS, Android, web et les apps de bureau, ainsi que sur les appareils tels que les téléviseurs intelligents et Internet des objets. Facebook Login active deux scénarios, l'authentification et la demande d'autorisations pour l'accès aux données des personnes (au format JSON ou XML). Vous pouvez utiliser Facebook Login simplement pour l'authentification ou à la fois pour l'authentification et l'accès aux données. [25]



Figure 40 : Logo Facebook

API Google Plus : Pour cette nouvelle API, Google a choisi de continuer le choix de OAuth2, utilisé de manière unifiée dans la gestion des autorisations d'accès des APIs aux données de votre compte Google. Une fois l'API key obtenue grâce à l'APIs Console, le contenu lors des appels à l'API est quant à lui renvoyé au client au format JSON. [26]



Figure 41 : Logo Google Plus

II. Réalisation des interfaces de l'application

1. Introduction

Dans ce qui suit, nous allons présenter le fruit de notre travail durant toute la période du stage tout en exposant les différentes interfaces de l'application que nous allons réussir à les mettre en place.

2. Réalisation

2.1. Sur PC, Tablette et Smart TV

Lorsque l'application se démarre, une page d'atterrissage (landing page) comme présentée sur les figures 42, 43 et 44 ci-dessous se lance. Cette page sert à :

- donner une idée sur notre application
- attirer l'attention de différents utilisateurs et les encourager à utiliser notre application
- guider l'utilisateur pour l'aider à utiliser notre application.

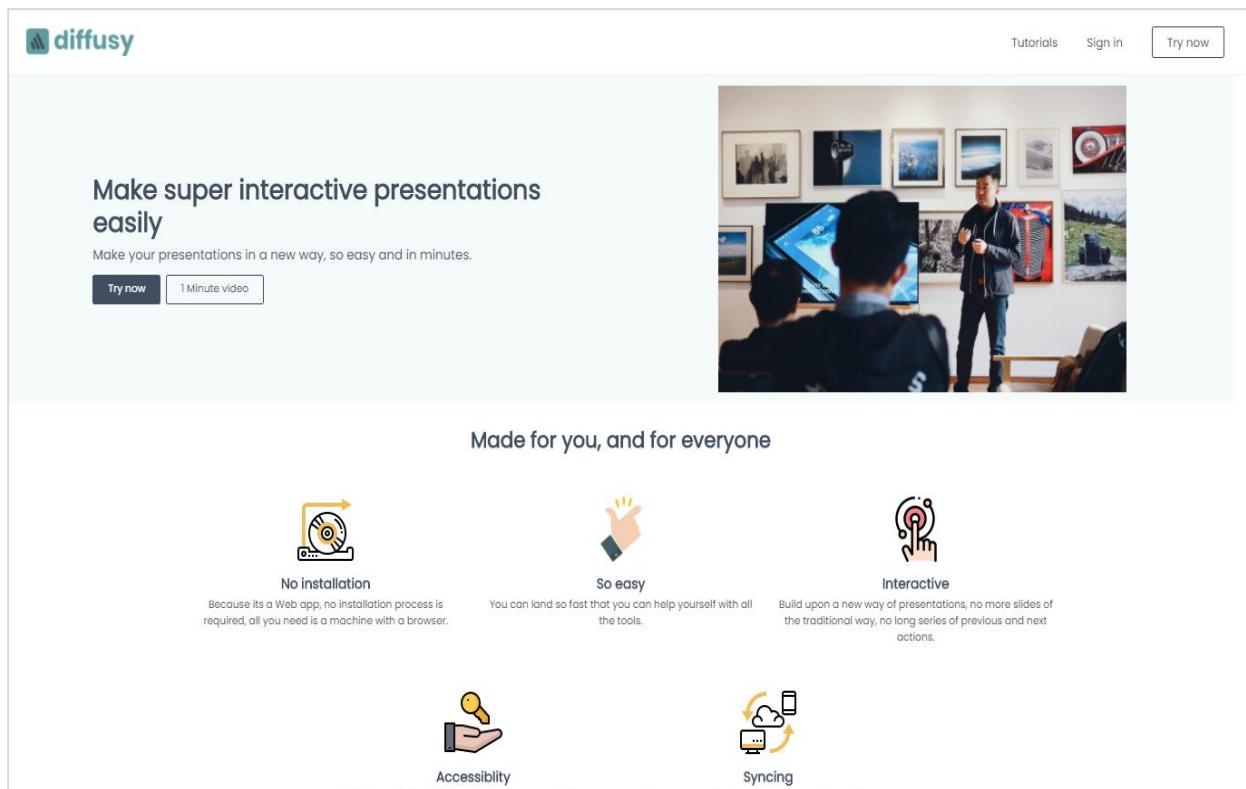


Figure 42 : L'entête de landing page

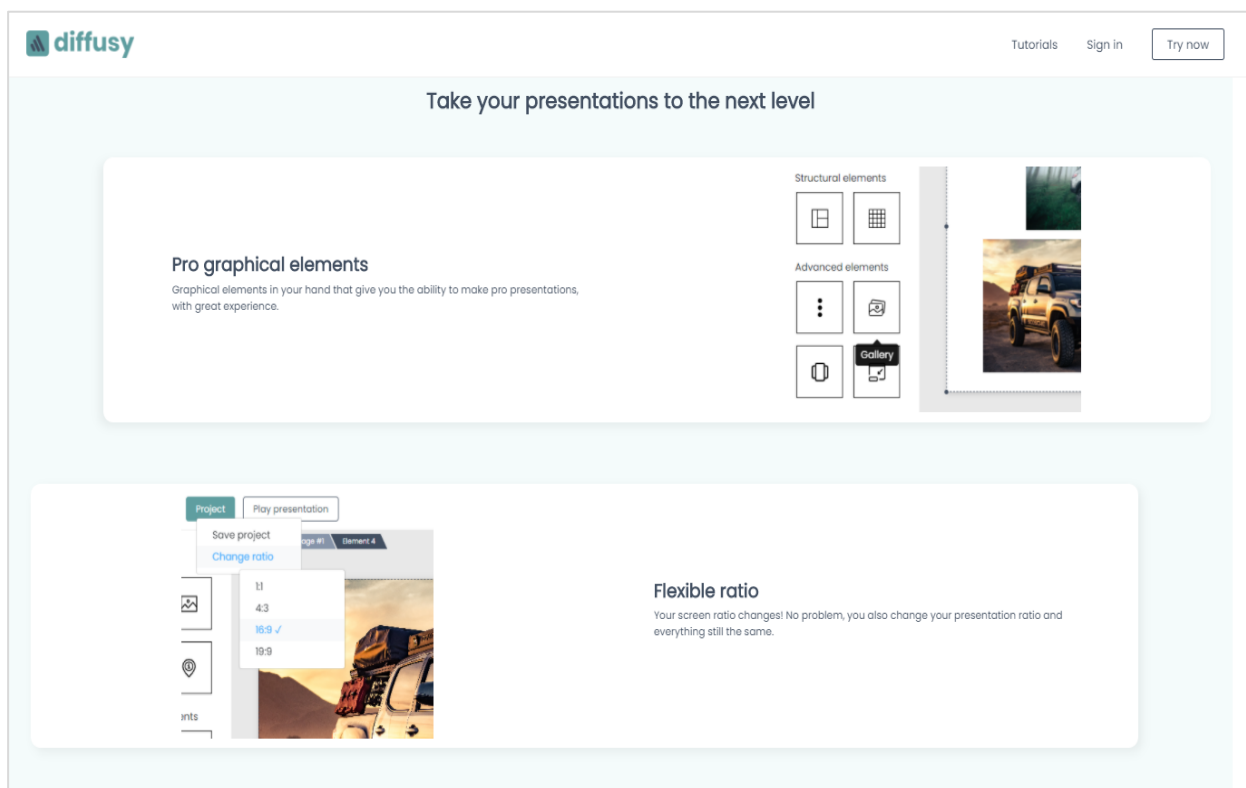


Figure 43 : Le corps de landing page

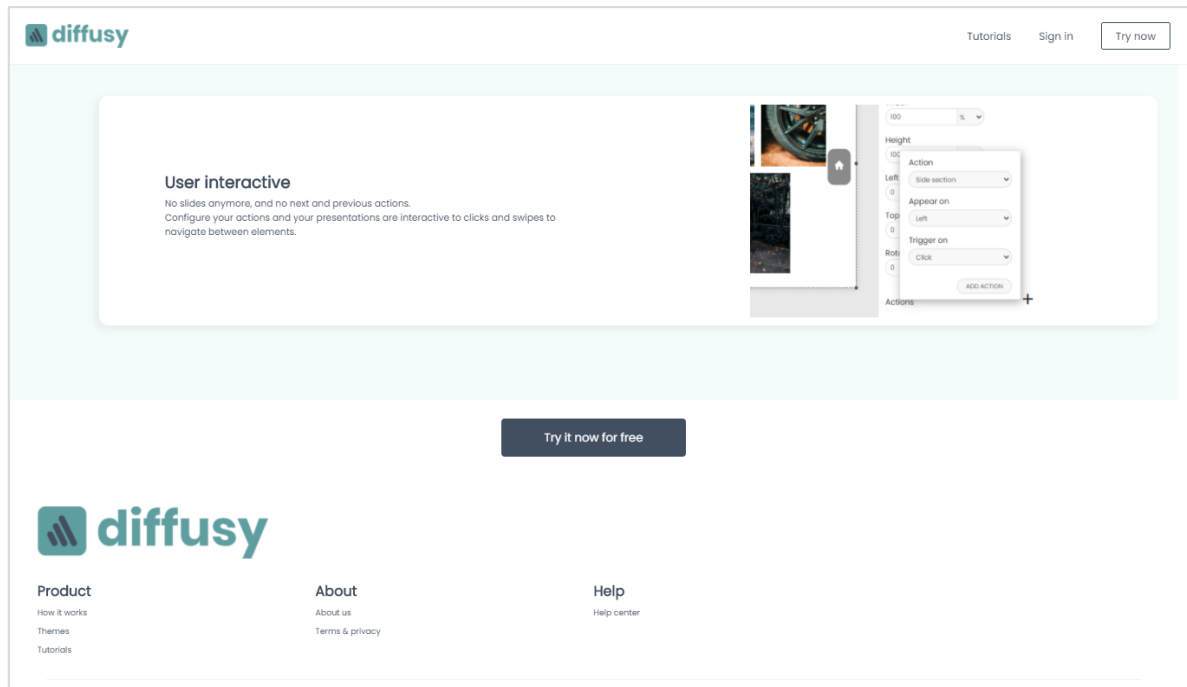


Figure 44 : Le bas de landing page

Lorsque l'utilisateur veut accéder à l'application sans passer par l'étape d'authentification, il doit cliquer sur le bouton « Try Now ». Une interface d'accueil comme se présente dans la figure 45 ci-dessous s'affiche. Notre application est dotée d'un menu de navigation (navbar) comme le montre la figure 46 qui englobe toutes les fonctionnalités offertes par l'interface d'accueil.

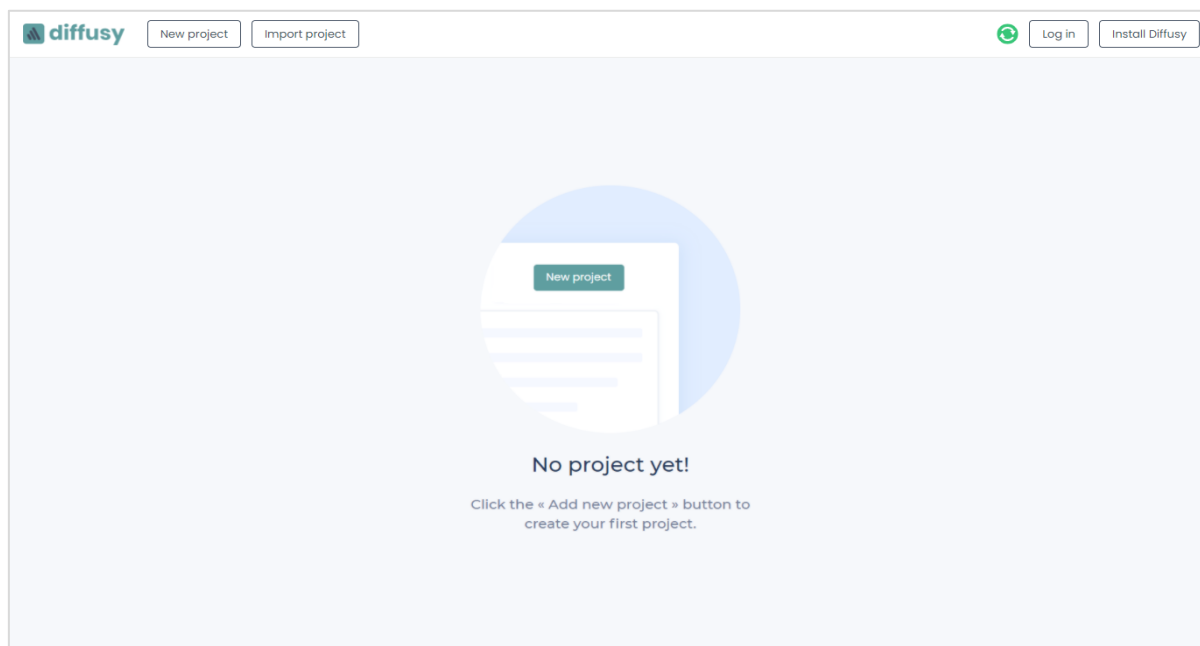


Figure 45 : Interface d'accueil

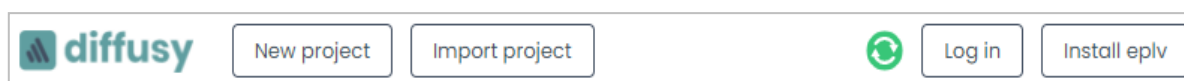


Figure 46 : menu de navigation (accueil)

Si l'utilisateur veut installer l'application sur son équipement, il doit cliquer sur l'item « Install Diffusy ». Une fois l'application est installée, cet item est disparu.

Et si l'utilisateur veut créer un nouveau projet, il doit cliquer sur l'item «New Project». Une interface de création de projet s'ouvre comme indiquée sur la figure 47 ci-dessous et le contenu de menu se change comme illustré dans la figure 48 ci-dessous.

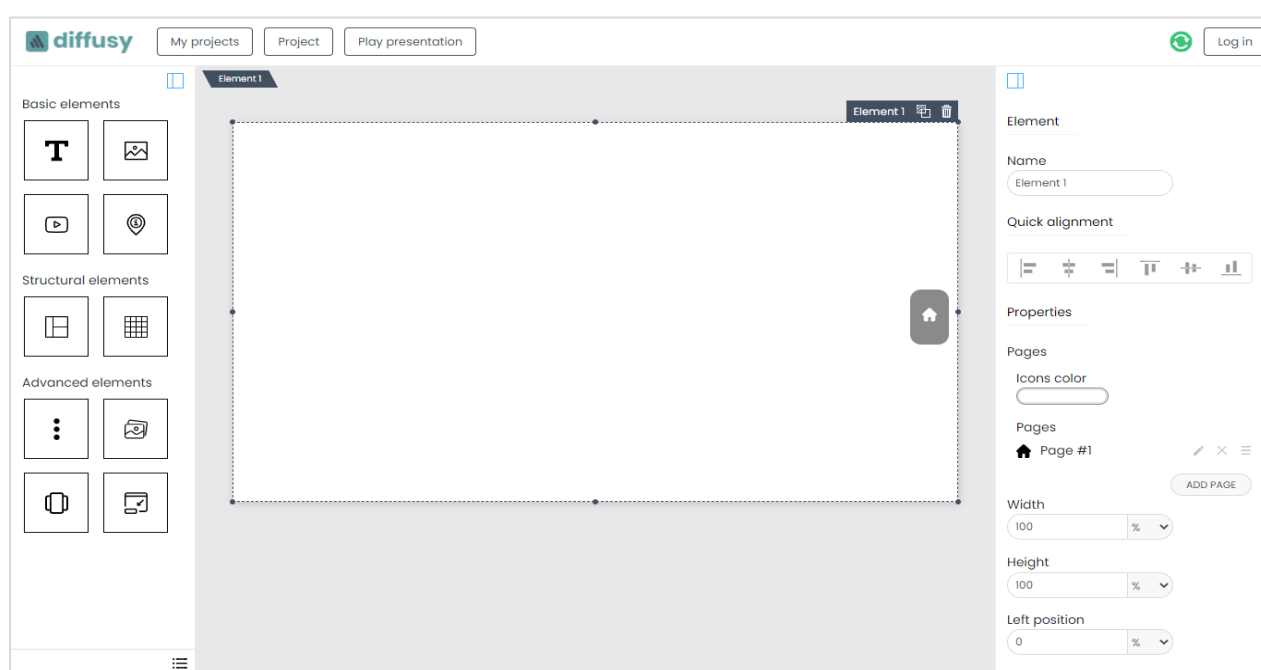


Figure 47 : Interface de création de projet

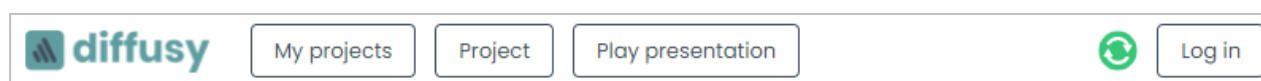


Figure 48 : menu de navigation (constructeur)

Avant de passer aux détails de l'interface de création de projet, nous allons présenter sur la figure 49 ci-dessous la structure de ses cinq composants suivants :

- 1- Menu de navigation (NavBar).
- 2- Menu des éléments.
- 3- Arbre des éléments (breadcrumb).

- 4- Panneaux de création.
- 5- Menu de configuration.

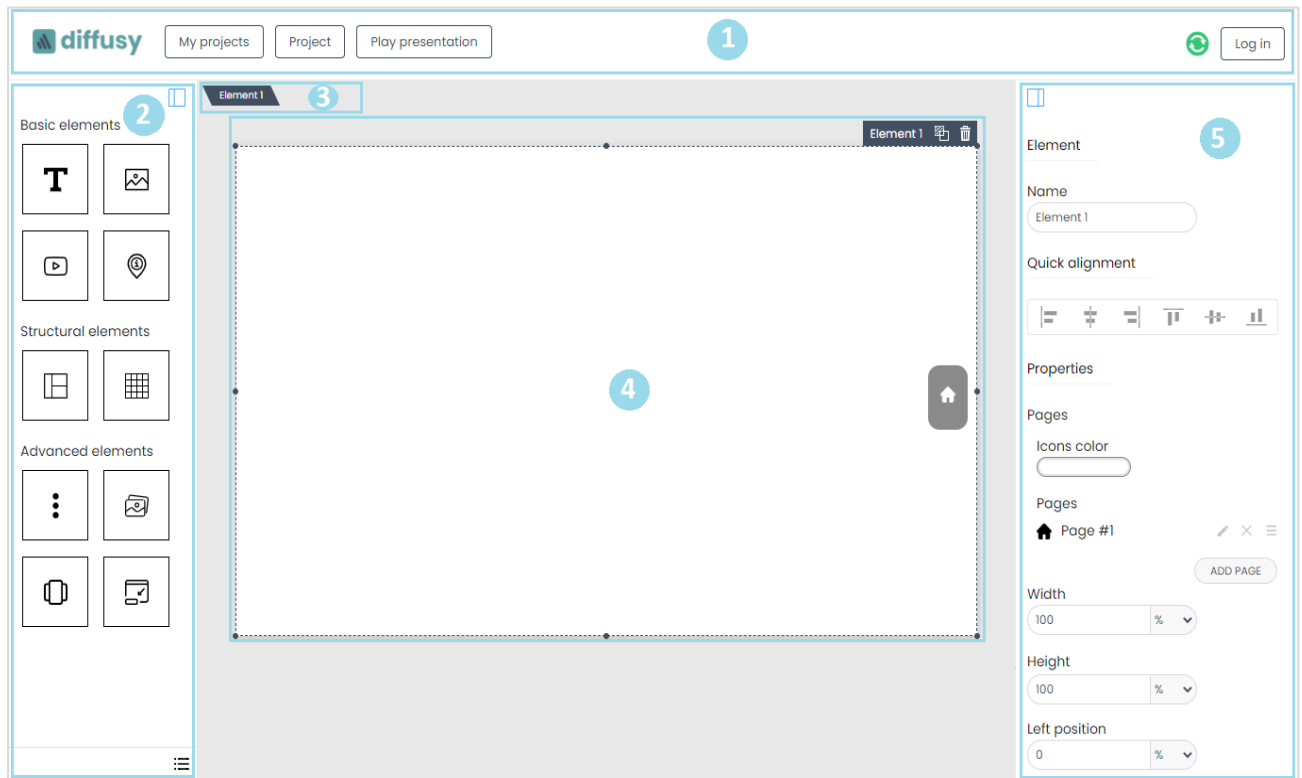


Figure 49 : Structure des composants

Si l'utilisateur veut ajouter un élément, il doit le glisser à partir de menu des éléments comme indiqué dans la figure 50 ci-dessous et de le déposer dans le panneau qui se trouve au milieu de l'interface de création de projet comme illustré dans la figure 49.

Notre menu des éléments se répartie en trois catégories, qui sont :

- Les éléments basiques : Texte, Image, Vidéo et Pastille.
- Les éléments structurels : Page et Grille (Grid).
- Les éléments avancés : Navigateur, Galerie, Carrousel et Onglets.

Et chaque élément possède des configurations comme le changement de position, le changement de couleur, l'ajout des actions sur des éléments, l'ajout de sous pages, l'ajout des diapos ou des onglets, etc. L'utilisateur peut alors configurer ses éléments à partir de menu de configuration comme présenté dans la figure 51 ci-dessous. Le contenu de ce menu se change à chaque sélection d'un élément, car chaque élément possède sa propre propriété.

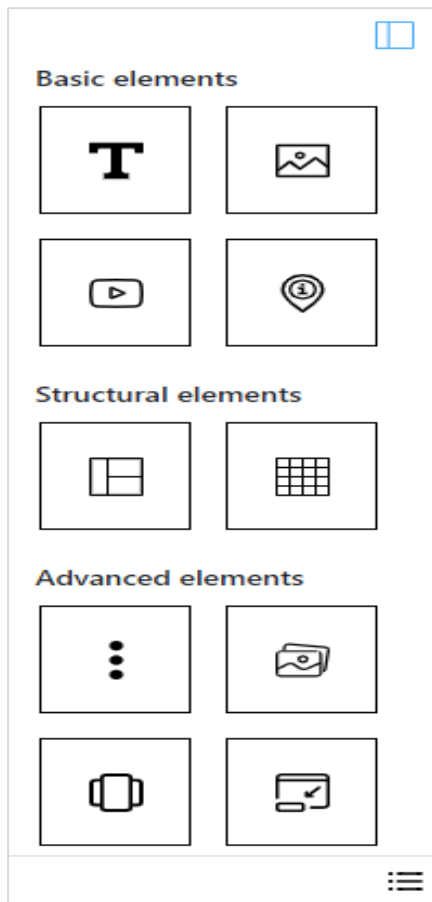


Figure 50 : Menu des éléments

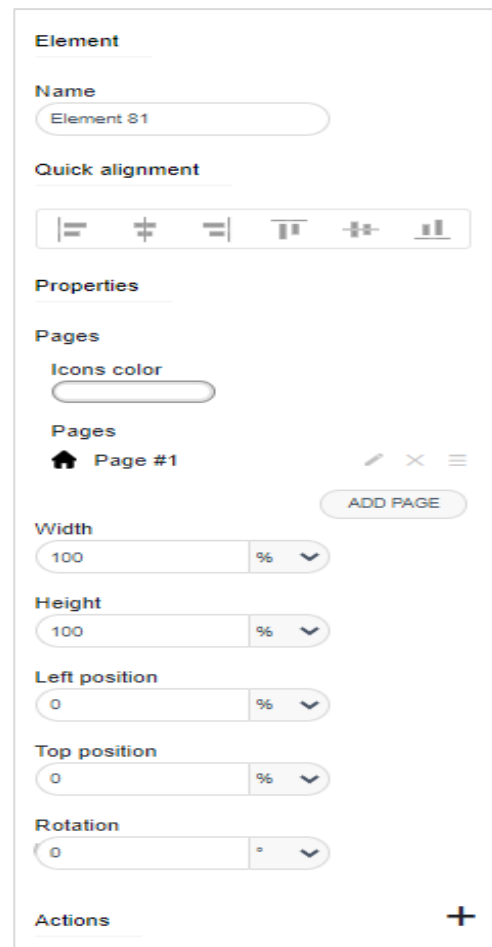


Figure 51 : Menu des configurations de navigateur

Et à partir de menu des éléments mentionné précédemment dans la figure 46 et par le un simple clique sur l'icône « ≡ », une boîte qui contienne l'arbre des éléments de projet s'affiche comme illustré dans la figure 52 ci-dessous. Ce qui facilite la navigation de l'utilisateur entre ses éléments ajoutés.

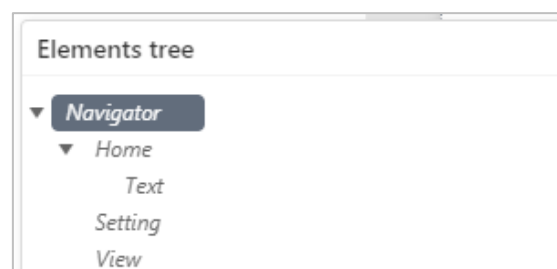


Figure 52 : Arbre des éléments

Il existe aussi une breadcrumb qui se trouve au dessus de panneaux de création comme indiquée dans la figure 53, et qui permet aussi l'utilisateur de naviguer entre les éléments d'un élément sélectionné sans passer par l'arbre des éléments.

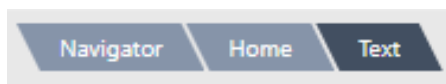


Figure 53 : breadcrumb

Nous allons présenter dans les figures ci-dessous quelques captures des configurations des éléments.

- La configuration commune pour tous les éléments :

L'utilisateur peut utiliser la fonctionnalité « alignement rapide » comme la montre la figure 56 pour aligner son élément sélectionné et il peut aussi changer la position, la taille et la rotation d'un élément à travers les fonctionnalités comme présentées dans la figure 54 ci-dessous. Il peut faire toutes ces fonctionnalités avec l'option «Drag and Drop» sans utiliser le menu de configuration.

De plus, s'il veut ajouter une action d'ouvrir une section ou une demi-section sur un élément, il doit cliquer sur l'item « Actions ». Une petite interface s'apparaît comme indiquée sur la figure 55 ci-dessous.

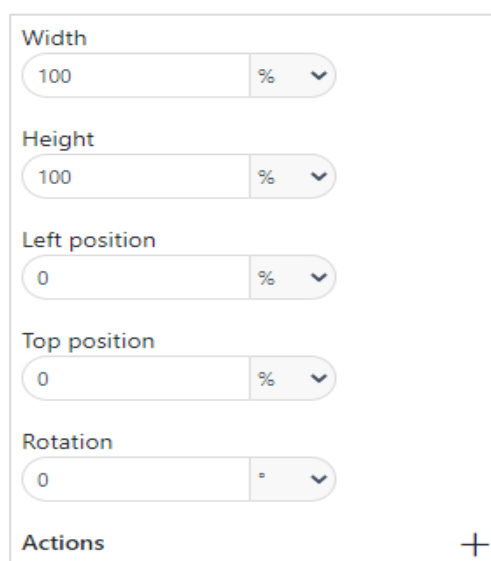


Figure 54 : Propriété Graphique

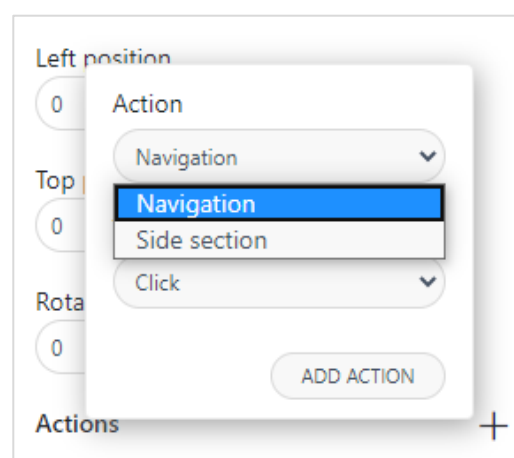


Figure 55 : Ajouter action

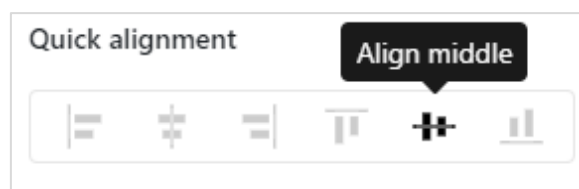


Figure 56 : Alignement rapide

Concernant les configurations spécifiques pour quelques éléments, nous allons commencer par l'élément « Texte » qui est un élément basique comme l'image, la vidéo et la pastille. Dans le cas où l'utilisateur désire modifier les propriétés de son texte, il doit appuyer sur la zone de texte. Une barre des outils s'apparaît comme la présente la figure 57 ci-dessous.

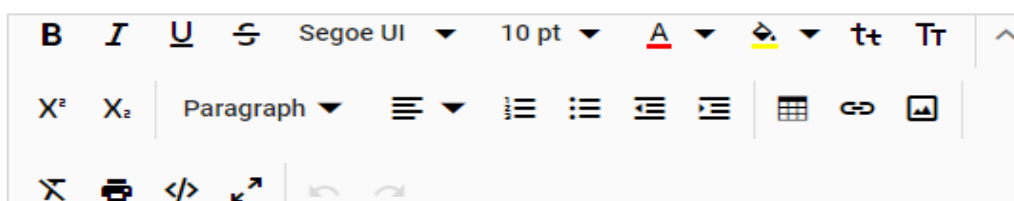


Figure 57 : Barre des outils de texte

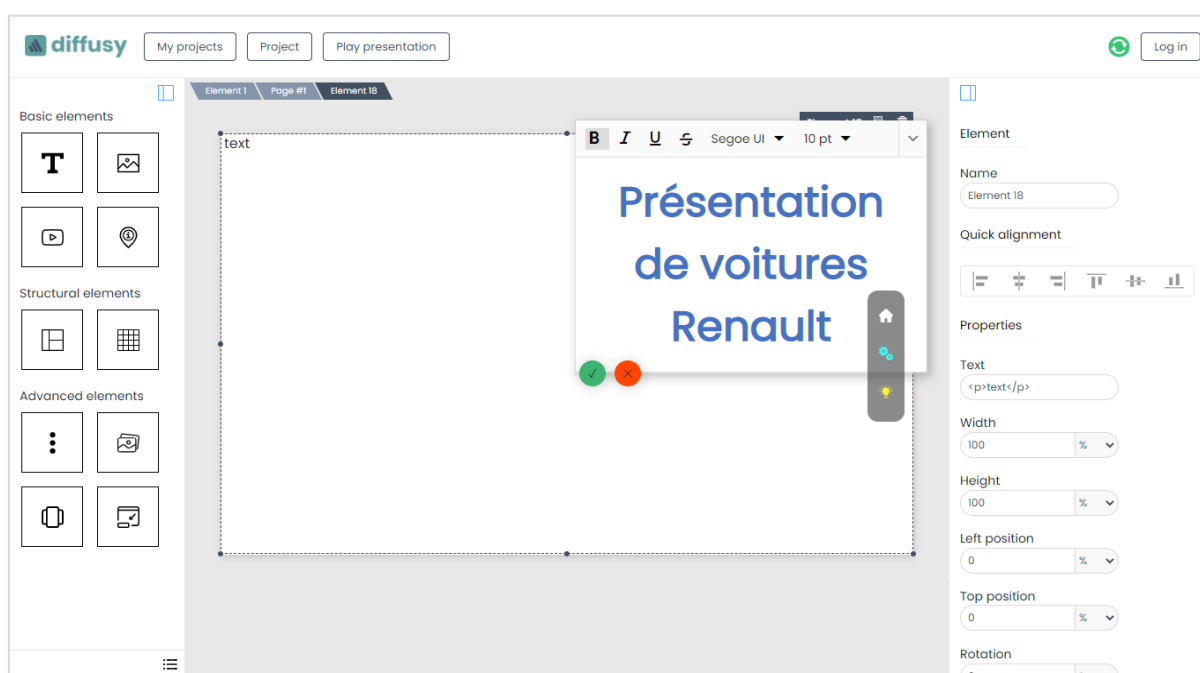


Figure 58 : Exemple d'un élément Texte

Pour l'élément carrousel, l'utilisateur peut ajouter des diapos et changer les configurations comme le passage automatique des diapos, le contrôle avec le clavier, l'apparition des flèches de navigation, etc. Toutes ces configurations sont présentées dans la figure 59 ci-dessous.

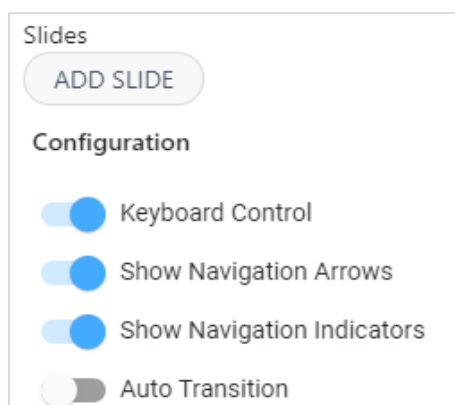


Figure 59 : Configuration de carrousel

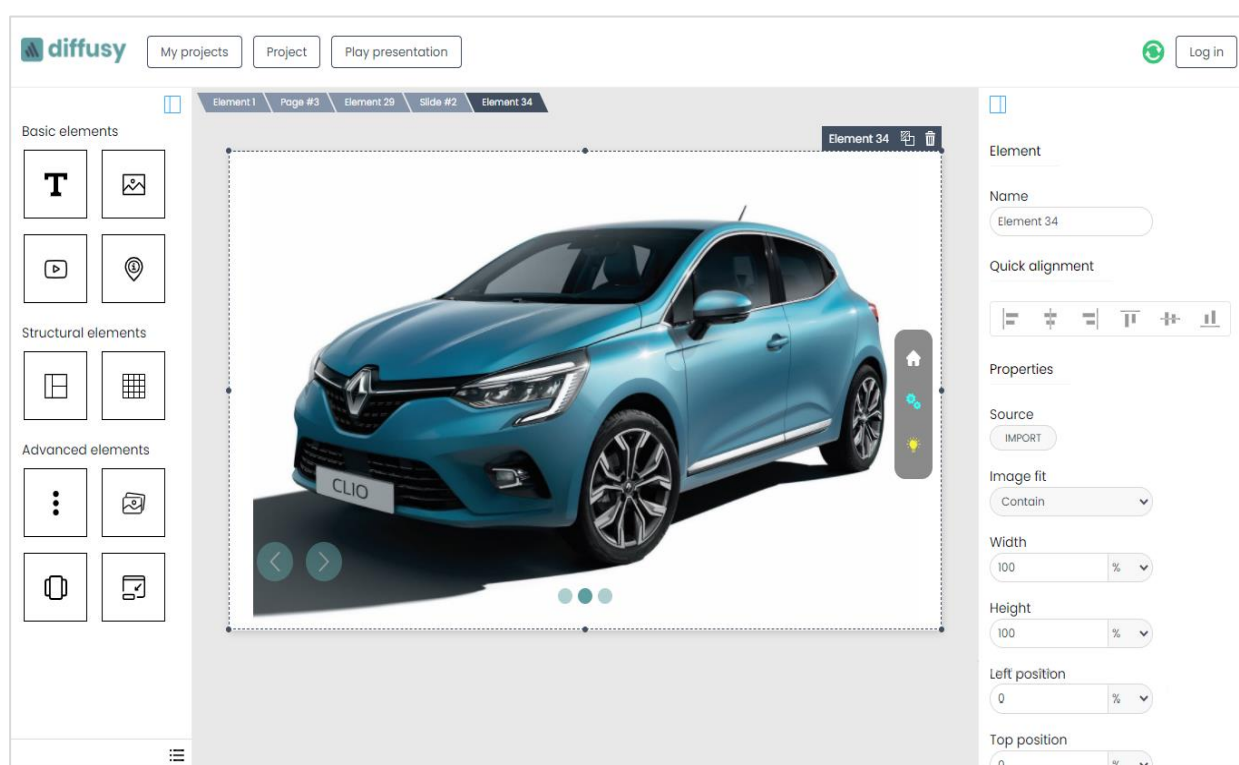
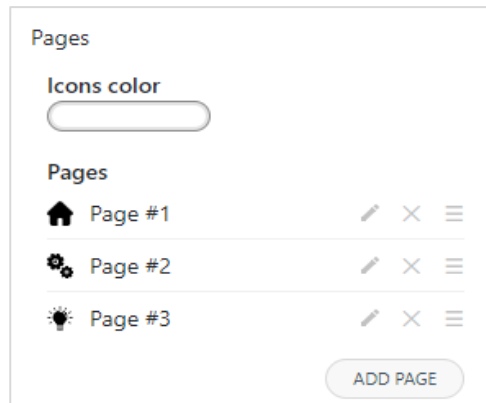
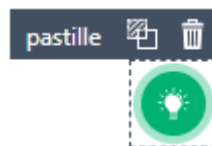


Figure 60 : Exemple d'un élément Carrousel

A propos l'élément « Navigateur » comme le présente la figure 61 ci-dessous, il sera créé par défaut lors de la création de nouveau projet. Et pour l'ajout d'une nouvelle page, l'utilisateur doit appuyer sur le bouton « ADD PAGE » de menu de configuration, et il peut aussi configurer les pages en changeant la couleur de leurs icônes et/ou les triées selon son besoin comme le montre la figure 62 ci-dessous.

**Figure 61 : Navigateur****Figure 62 : Configuration de Navigateur**

A la sélection de chaque élément ajouté, une étiquette s'affiche au dessus de l'élément comme la présente la figure 63. Cette étiquette nous permet de supprimer l'élément sélectionné ou de le mettre au premier plan.

**Figure 63 : Etiquette de pastille**

L'utilisateur peut ajouter des actions sur n'importe quel élément (pastille, texte, image, ...), pour interagir avec sa présentation comme présente la figure 64 ci-dessous. Une action par la pastille sur le capot de la voiture pour ouvrir la demi-section qui contienne les caractéristiques du moteur de voiture.

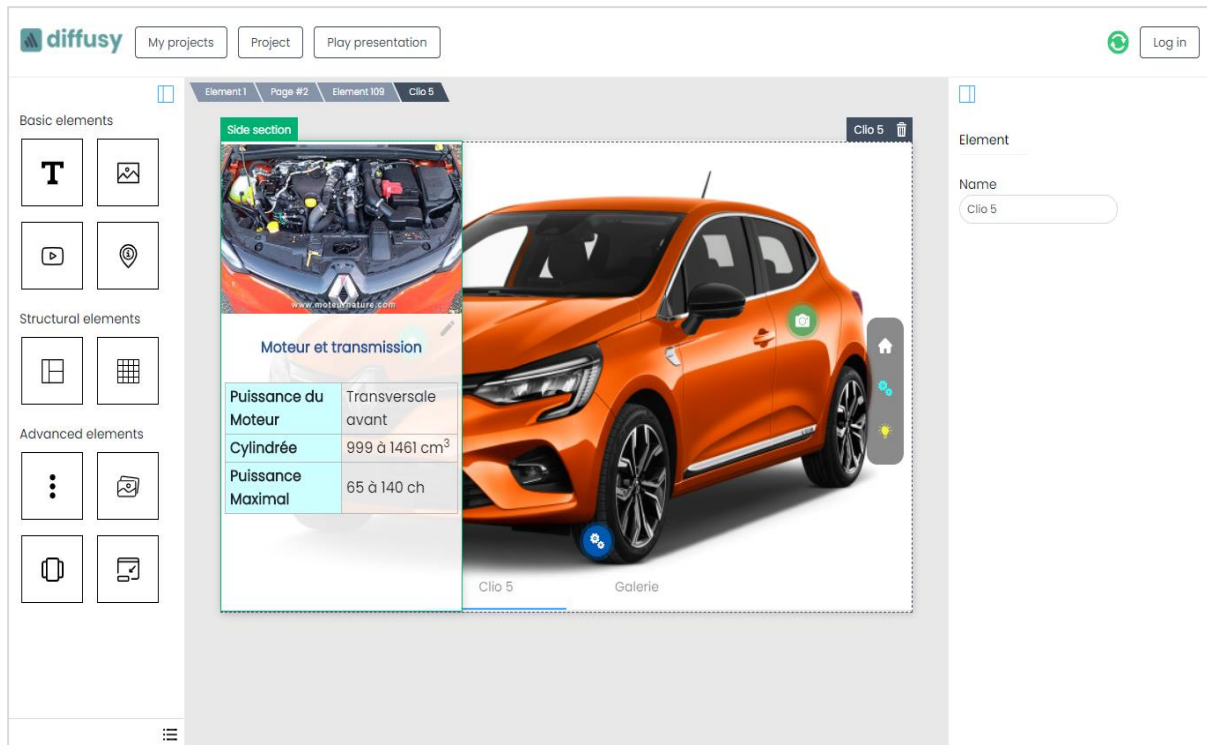


Figure 64 : Exemple d'une action

Pour l'élément Tabs, l'utilisateur peut ajouter des onglets et changer les propriétés comme le triage des onglets, à l'aide de l'espace de propriétés qui se trouve dans le menu de configuration comme indiqué dans la figure 65 ci-dessous.

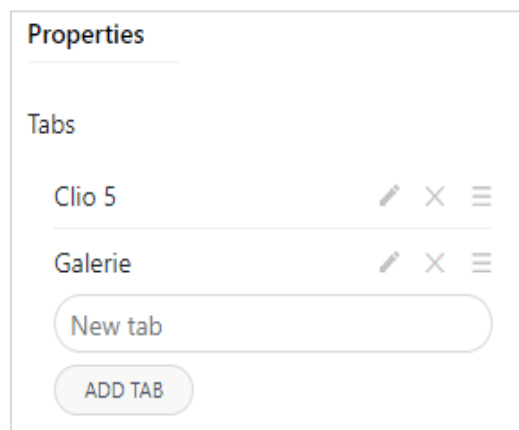


Figure 65 : Propriétés des onglets

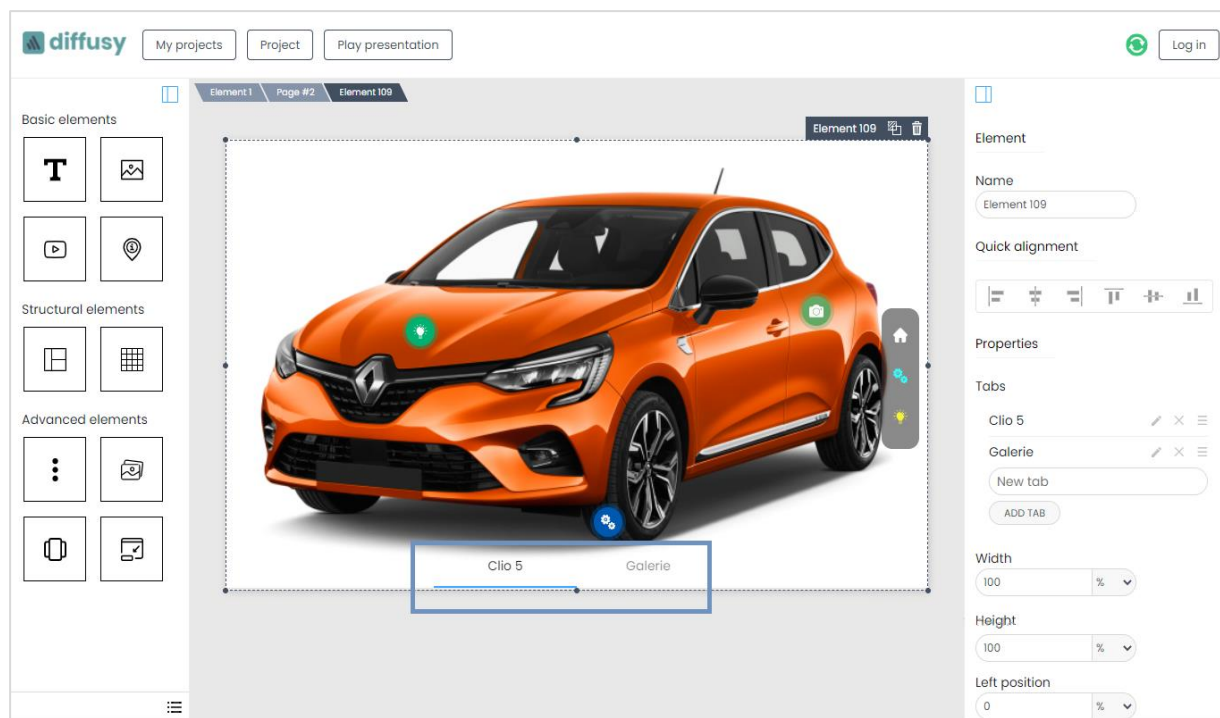


Figure 66 : Exemple d'un élément Tabs

Si l'utilisateur veut ajouter des photos, il existe un élément comme présenté sur la figure 67 ci-dessous appelé galerie qui le permet d'ajouter un ensemble des images. En mode de présentation, il peut naviguer entre ses photos avec le même principe de l'élément carrousel.

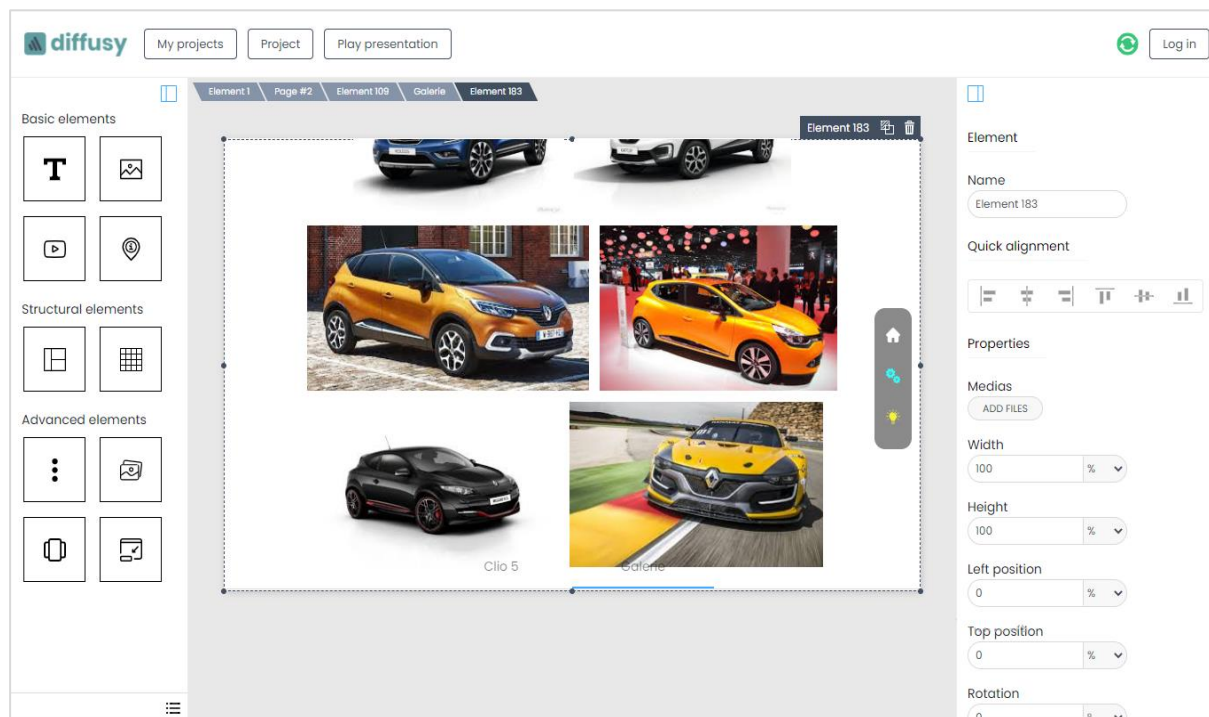


Figure 67 : Exemple d'un élément Galerie

Lors de sélection d'une image de galerie, deux boutons s'affichent comme montre la figure 68 ci-dessous. Un bouton « CHANGE » pour le changement de la photo et une icône de poubelle pour la suppression.



Figure 68 : Photo sélectionnée de l'élément Galerie

La grille dynamique comme présente la figure 69 ci-dessous, permet à l'utilisateur d'organiser des éléments (texte, image, vidéo,...) dans une seule interface. L'utilisateur peut les configurer en modifiant le nombre de lignes et de colonnes et en allouant des cellules pour chaque élément à l'aide de l'espace de propriétés qui se trouve dans le menu de configuration.

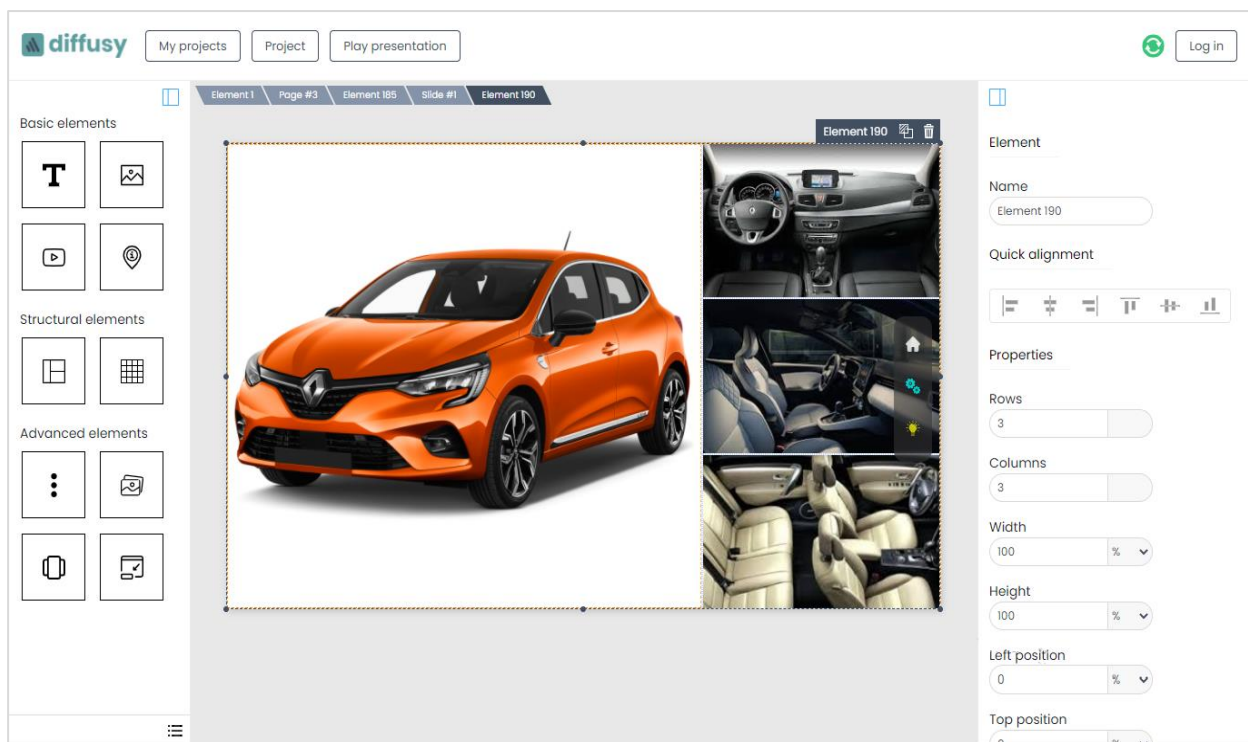


Figure 69 : Grid des photos

Si l'utilisateur désire enregistrer son projet, il doit cliquer sur le bouton « Project ». Un menu déroulant s'ouvre comme présenté dans la figure 70 ci-dessous. Il doit dans ce cas appuyer sur l'item « Save Project ». Un input s'apparaître pour saisir le nom de projet comme indiqué sur la figure 71 ci-dessous. Après l'introduction de nom, l'utilisateur doit cliquer sur le bouton « SAVE ».

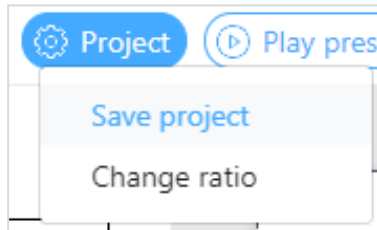


Figure 70 : Menu Project

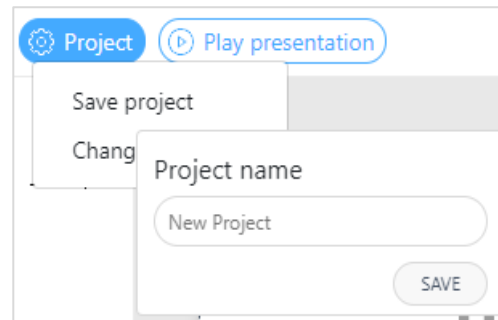


Figure 71 : Enregistrer le projet

Et à travers ce menu déroulant, il peut aussi changer la résolution de panneaux de création tout en cliquant sur l'item « Change ratio ». Une liste des tailles s'ouvre comme indiquée sur la figure 72 ci-dessous. Et à partir de cette liste, il peut choisir la résolution qui correspond à son besoin.

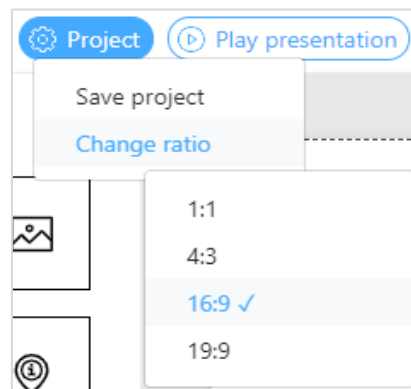


Figure 72 : Modifier la résolution de panneaux

Si l'utilisateur veut démarrer sa présentation, il doit cliquer sur le bouton « Play presentation » qui se trouve dans le menu de l'application comme indiqué dans la figure 48 vue précédemment. Son projet s'ouvre en plein écran et en mode diaporama.

Et si l'utilisateur veut retourner à la page d'accueil pour consulter ses projets, il doit cliquer sur le bouton « My Projects » qui se trouve dans le menu de navigation. Dans ce cas, si

l'utilisateur oubliera de sauvegarder son travail, une boîte d'alerte s'ouvre comme présentée dans la figure 73 ci-dessous. Cette boîte s'affiche aussi dans le cas où l'utilisateur ne sauvegarde pas son projet et veut quitter l'application ou le navigateur.

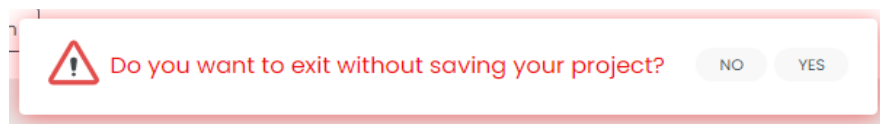


Figure 73 : Boîte d'alerte

Après l'appui sur le bouton « My Projects », l'interface d'accueil s'ouvre avec la liste des projets comme la présente sur la figure 74 ci-dessous. L'utilisateur peut aussi filtrer ses projets à l'aide de champ de recherche.

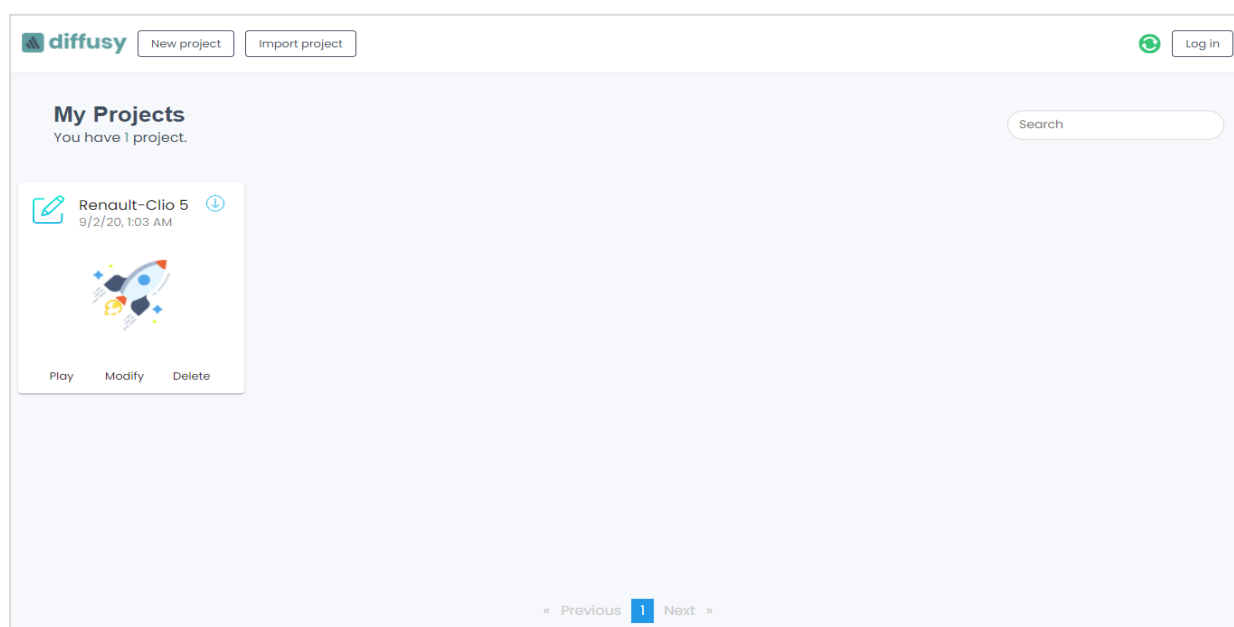




Figure 74 : Interface d'accueil

Si l'utilisateur désire importer un projet à partir de son équipement, il doit cliquer sur l'item « Import Project » du menu d'application comme présenté dans la figure 46.

Chaque projet sera représenté sous la forme d'une carte comme indiqué sur la figure 75 ci-dessous. Pour la modification d'un projet, l'utilisateur doit cliquer sur le bouton « Modify » de la carte qui lui amène à l'interface de construction. Il existe aussi un bouton « Delete » pour la suppression de projet, un bouton « Play » pour démarrer la présentation. Et pour la modification de nom de projet, il existe une icône de stylo «  » et une icône d'exportation «  » pour l'exportation de projet.

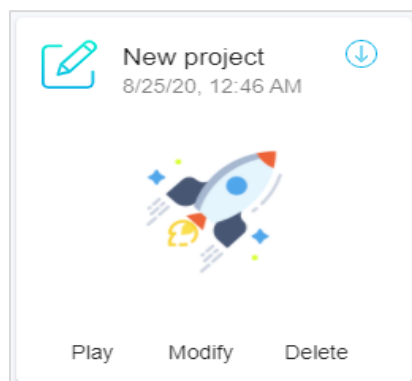


Figure 75 : Carte d'un projet

Si l'utilisateur veut synchroniser ses projets, il doit connecter avec le bouton « login » qui se trouve dans le menu de navigation comme indiqué dans la figure 46 vue précédemment, ou à travers le bouton « Sign In » qui se trouve dans le menu de la page d'atterrissage. Un modal s'ouvre comme le présente dans la figure 76 ci-dessous.


A travers ce modal, l'utilisateur peut se connecter ou s'inscrire. Lors de l'appui sur le bouton « Sign Up », le contenu de modal se change comme l'indique la figure 77 ci-dessous.

Le modal "SignIn/SignUp" est configuré pour l'authentification. Les onglets "Sign In" et "Sign Up" sont présents, avec "Sign In" sélectionné. Les champs de formulaire incluent "Email *", "Password *", et "Remember Me!". Un lien "Forgot password?" est visible. Les boutons "Sign In" et "or connect with:" sont présents, suivis des boutons de connexion sociale "G+" et "f".

Figure 76 : Modal d'authentification

Le modal "SignIn/SignUp" est configuré pour l'inscription. Les onglets "Sign In" et "Sign Up" sont présents, avec "Sign Up" sélectionné. Les champs de formulaire incluent "User Name *", "Email *", et deux champs "Password *". Le bouton "Sign Up" est visible, suivi des boutons de connexion sociale "G+" et "f".

Figure 77 : Modal d'inscription

Après l'authentification, l'utilisateur doit cliquer sur l'icône «  » pour compléter l'étape de synchronisation de ses projet.

Si l'utilisateur veut accéder ou modifier son profil, il doit cliquer sur le bouton « Profile » qui s'affiche dans le menu d'application à la place de bouton « login » juste après l'étape d'authentification comme le présente sur la figure 78 ci-dessous. Alors, le modal de profil s'ouvre comme indiqué sur la figure 79 ci-dessous. Et à travers ce modal, l'utilisateur peut changer ses informations professionnelles et peut aussi quitter l'application par l'appui sur le bouton « LogOut ».



Figure 78 : Menu de navigation (Après l'authentification)

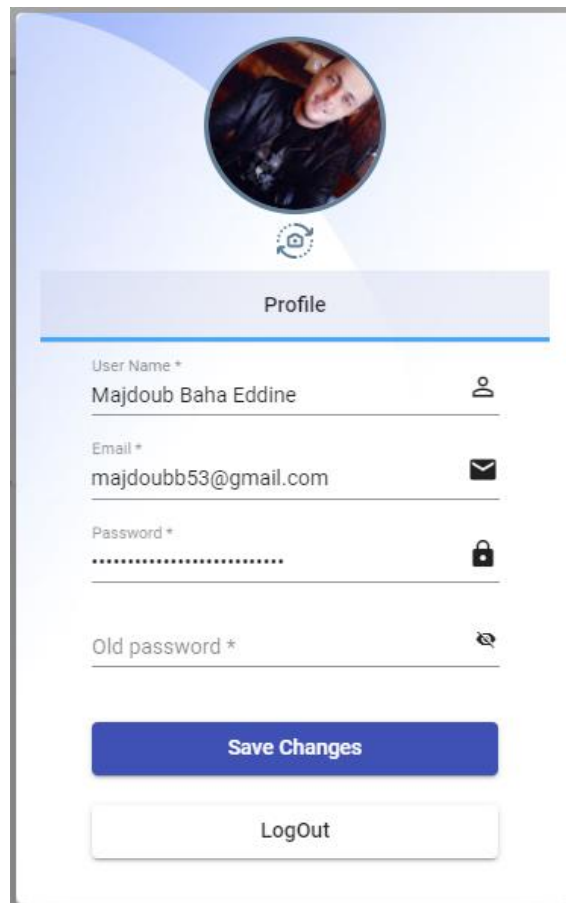


Figure 79 : Modal de profil

2.2. Sur Smartphones

La version mobile est uniquement destinée à la suppression et à la consultation des projets. Alors, lorsque l'application se lance, une interface comme présentée sur la figure 80 ci-dessous s'affiche et invite l'utilisateur à se connecter en introduisant son login et son mot de passe ou à s'inscrire avec les réseaux sociaux tels que FaceBook et Google Plus ou avec l'interface d'inscriptions s'il ne possède pas encore un compte sur l'application.

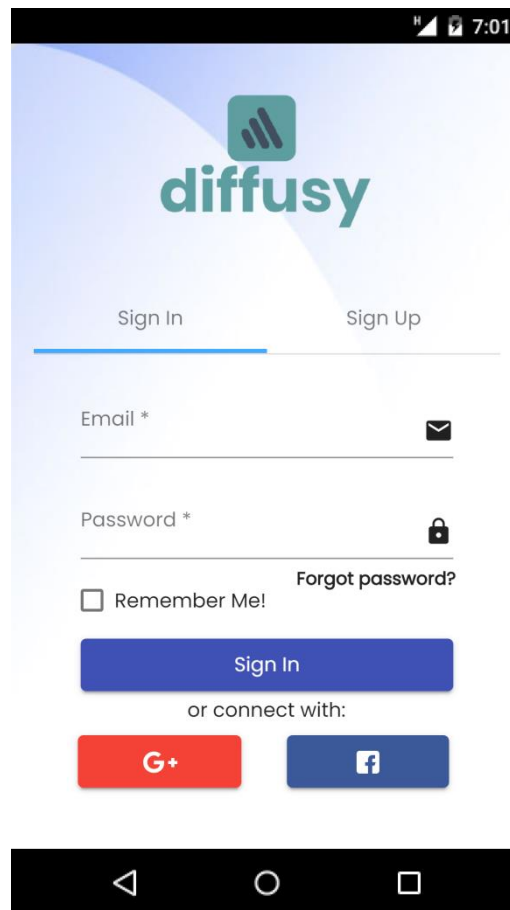


Figure 80 : Interface d'authentification

Lorsque l'utilisateur clique sur le bouton "Sign Up", un formulaire d'inscription se lance pour lui donner la main à introduire un ensemble d'informations afin d'avoir un compte sur l'application comme indiqué sur la figure 81 ci dessous.

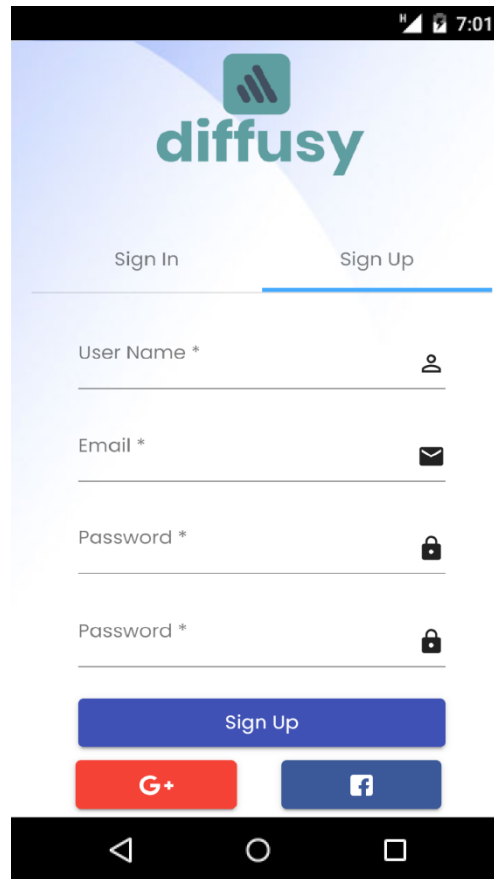


Figure 81 : Interface d'inscription

Après l'étape d'authentification ou d'inscription, l'interface d'accueil de notre application s'ouvre directement comme montre la figure 82. Notre application est donc dotée d'un menu intuitif qui englobe toutes les fonctionnalités offertes.

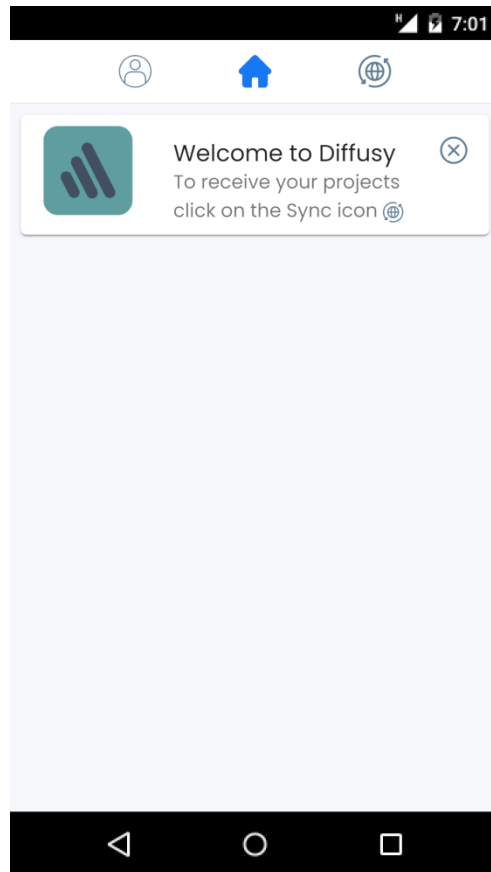




Figure 82 : Interface d'accueil

Si l'utilisateur veut installer l'application sur son Smartphone, il doit cliquer sur l'item «  ». Et une fois l'application s'installe, cette icône sera disparue. Lorsque l'utilisateur veut consulter et modifier ses informations personnelles, il doit cliquer sur l'item «  ». L'ensemble de ses informations s'affiche comme indiqué sur la figure 83 ci-dessous. A l'aide des inputs, il peut changer ses informations et les enregistrer en appuyant sur le bouton "Save Changes" tout en utilisant son ancien mot de passe. Comme il peut quitter l'application par l'appui sur le bouton "LogOut".

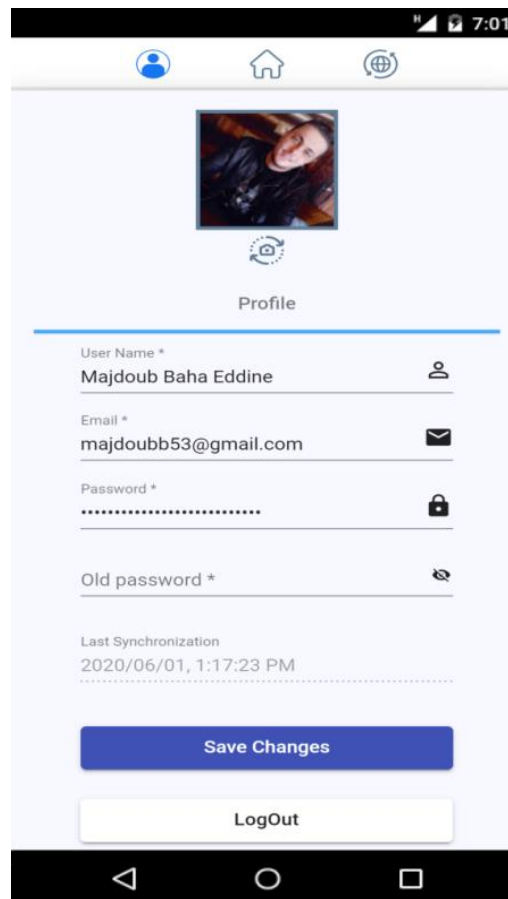



Figure 83 : Interface Modifier le profil

Si l'utilisateur veut retourner à l'interface d'accueil, il doit appuyer sur l'icône « Home ». Et lorsqu' il veut synchroniser ses projets, il doit cliquer sur l'item «  ». L'ensemble de ses projets qui sont déjà stockés dans le serveur, seront récupérés et sauvegardés dans la base locale comme l'indique la figure 85 ci-dessous.

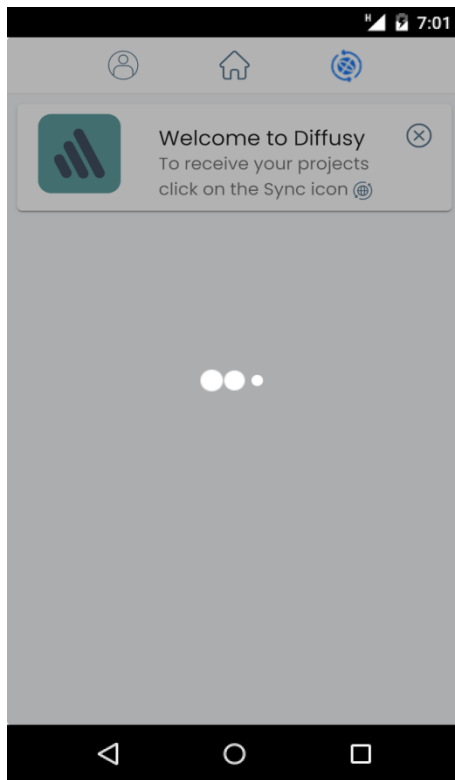


Figure 84 : Synchronisation

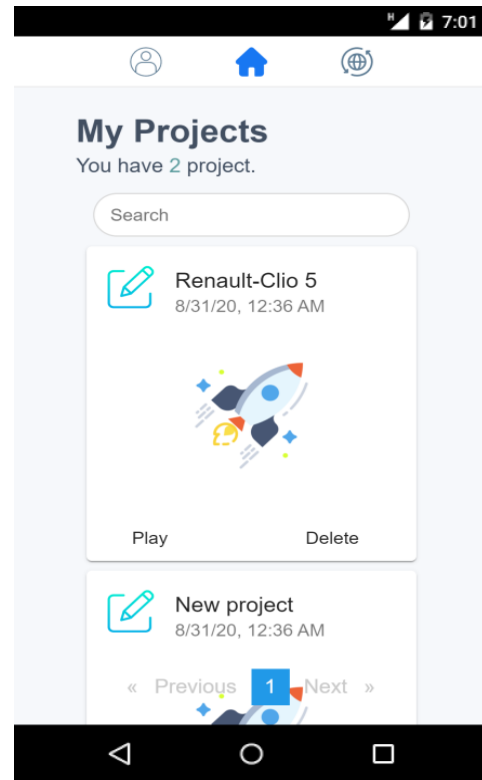


Figure 85 : Interface d'accueil

Lorsque l'utilisateur veut supprimer un projet, il doit cliquer sur le bouton « Delete ». Le projet ainsi sera supprimé mais seulement de la base locale. Pour le retirer aussi de la base distante, l'utilisateur doit cliquer sur l'item de synchronisation. Et pour démarrer une présentation, ce dernier doit appuyer sur le bouton « Play ». La présentation de projet se démarre en plein écran comme indique la figure 86 ci-dessous.

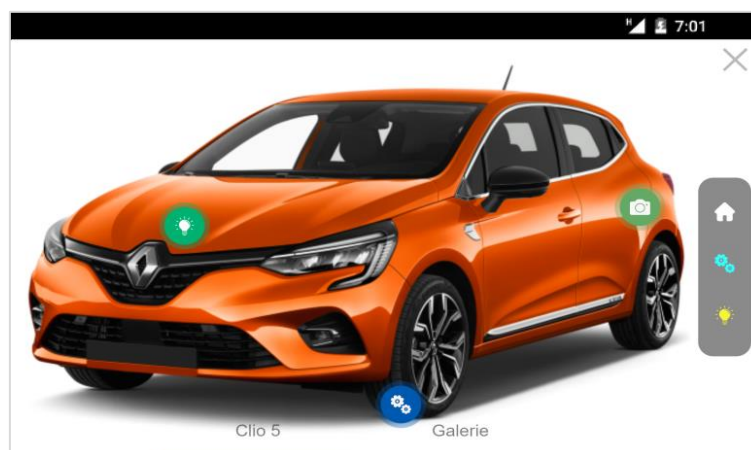


Figure 86 : Interface de présentation

Dans le cas où l'utilisateur oublie son mot de passe, il doit cliquer sur le bouton « Forgot password » qui se trouve dans l'interface d'authentification comme se présente dans la figure 80 précédente. Une boîte s'affiche contenant un champ d'email comme indique la figure 87 ci-dessous.

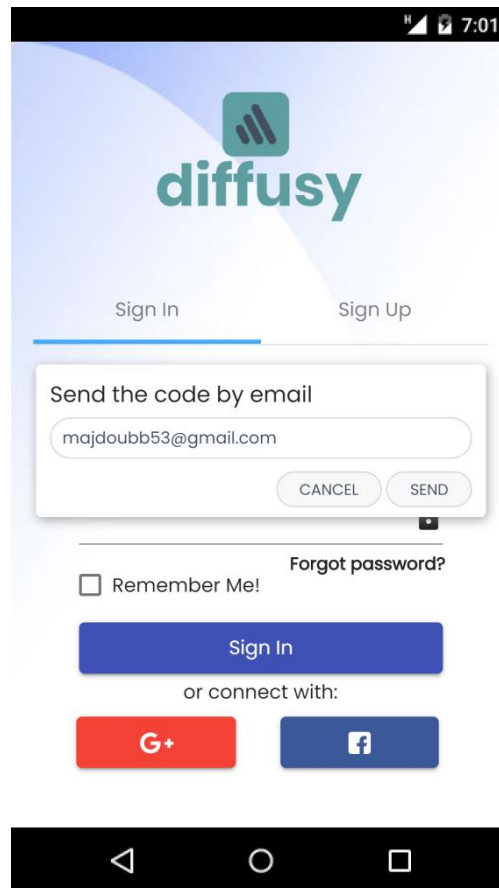


Figure 87 : Mot de passe oublié

Pour récupérer son mot de passe déjà oublié, l'utilisateur doit entrer son adresse email utilisée lors de la création de son compte. Puis, il doit cliquer sur le bouton « SEND ». Si l'email est valide, une alerte s'affiche en bas de l'interface comme illustre la figure 88 ci-dessous.

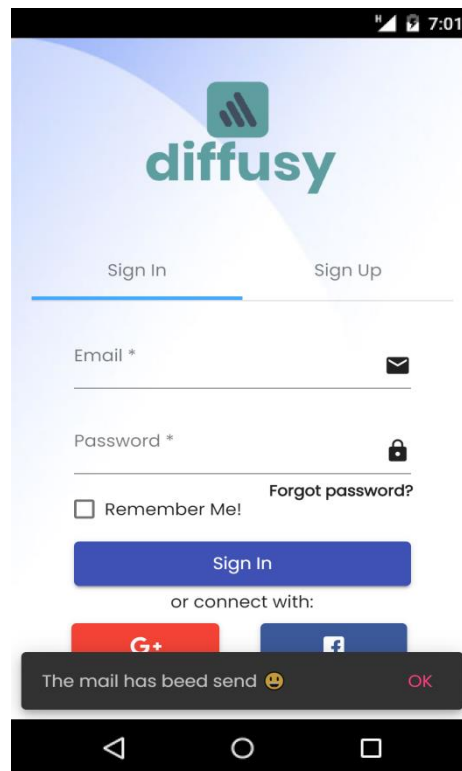


Figure 88 : Alerte email envoyé

Un email sera envoyé à l'utilisateur qui contient son mot de passe oublié, comme indiqué sur la figure 89 ci-dessous.

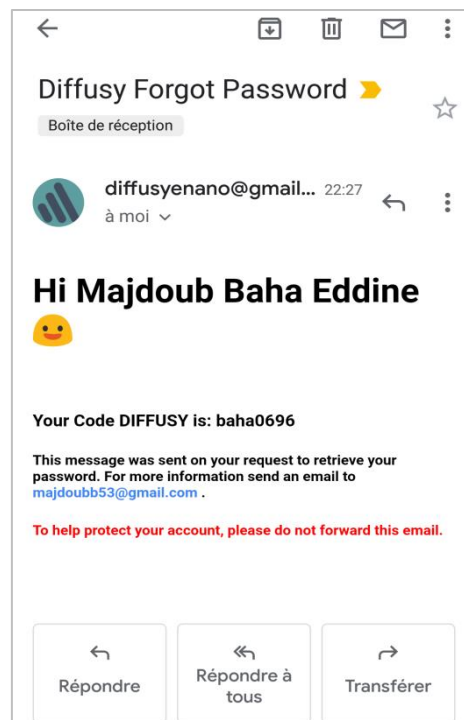


Figure 89 : Email de mot de passe

Conclusion

Au cours de ce chapitre, nous avons décrit la dernière étape de notre projet qui est la mise en œuvre de l'application. Nous avons commencé par une présentation de l'environnement matériel et logiciel du travail accompagné des définitions des choix effectués. Nous avons également présenté la méthode de développement appliquée avec un aperçu de quelques interfaces de l'application.

Conclusion générale

Pour conclure, notre projet, au sein de la société eNano-Tech, a porté sur la conception et la réalisation d'une application web progressive. Nous exposons ici et dans l'ordre chronologique une synthèse des grandes étapes qui ont marqué le déroulement de notre projet.

L'objectif de notre projet était le développement d'un outil de présentation interactive et multiplateforme qui permet à plusieurs utilisateurs de produire et de gérer des interfaces attrayantes. Au cours de la phase de réalisation de notre application, nous avons élaboré une étude préalable sur les applications web progressives et leur importance sur le plan social afin de préciser le but principal pour la future application. Cette phase a constitué le point de départ pour l'étape d'analyse et de spécification des besoins. Une fois nos objectifs sont fixés, nous avons enchaîné avec la conception afin de mener à bien ce projet. Nous avons procédé à la phase de réalisation au cours de laquelle nous sommes familiarisés avec le langage de programmation.

A l'issue de la réalisation de ce travail, je peux affirmer que ce projet m'a été d'une grande utilité dans la mesure où il m'a permis de me familiariser avec le travail sur des nouveaux environnements de développement, à savoir le deux Frameworks Angular 9 et NestJs, le système de gestion de base de données MongoDB et les techniques de stockage en local et de la synchronisation des données. Non seulement, les bénéfices ont été réalisés sur le plan technique mais aussi sur le plan social. J'ai pu avoir un aperçu autour du travail au sein d'une boîte de développement. L'intégration dans une équipe de travail a été une expérience qui marquera la période de réalisation du projet au sein de la société eNano-Tech, hôte pour mon stage.

Enfin, le travail réalisé ne se limite pas à ce niveau. En effet, plusieurs perspectives s'offrent à ce projet. Parmi les fonctionnalités que nous pouvons envisager pour notre solution sont l'ajout des animations aux plusieurs éléments, l'ajout d'un lien de partage sur les réseaux sociaux et l'insertion de l'application « DIFFUSY » sur Android market, App Store et Web Store.

Webographie

- [1] : <https://enano-tech.com/>
- [2] : https://fr.wikipedia.org/wiki/Microsoft_PowerPoint
- [3] : <https://www.google.com/intl/fr/slides/about/>
- [4] : <https://www.adobe.com/fr/products/captivate.html>
- [5] : <https://prezi.com/p/cgpqpkclrqop/le-logiciel-prezi/>
- [6] : [https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique))
- [7] : <http://uml.free.fr/index-cours.html>
- [8] : http://www.siloged.fr/cours/STI2D_sysml/index.html?Diagrammedecasdutilisation.html
- [9] : <https://sites.google.com/site/developpementdelinformatique/uml>
- [10] : <http://www.laurent-audibert.developpez.com/cours-uml>
- [11] : <https://www.lri.fr/~longuet/Enseignements/16-17/Et3-UML/Et3-5DiagSequence.pdf>
- [12] : <https://www.cairn.info/theorie-des-organisations--9782130441632-page-97.htm>
- [13] : https://www.ibm.com/support/knowledgecenter/fr/SS4JE2_7.5.5/com.ibm.xtools.modeler.doc/topics/cdepd.html
- [14] : <http://www.code4corsica.com/blog/technique/quest-ce-que-angular>
- [15] : <https://docs.nestjs.com/>
- [16] : https://fr.wikipedia.org/wiki/Hypertext_Markup_Language
- [17] : [https://fr.wikipedia.org/wiki/TypeScript#:~:text=TypeScript%20est%20un%20langage%20de,peut%20%C3%AAtre%20utilis%C3%A9%20avec%20TypeScript\)](https://fr.wikipedia.org/wiki/TypeScript#:~:text=TypeScript%20est%20un%20langage%20de,peut%20%C3%AAtre%20utilis%C3%A9%20avec%20TypeScript))
- [18] : [https://fr.wikipedia.org/wiki/Sass_\(langage\)](https://fr.wikipedia.org/wiki/Sass_(langage))
- [19] : <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/mongodb-presentation-et-comparaison-avec-mysql/>
- [20] : https://developer.mozilla.org/fr/docs/Web/API/API_IndexedDB
- [21] : https://fr.wikipedia.org/wiki/Visual_Studio_Code
- [22] : <https://inf1410.telug.ca/telugDownload.php?file=2014/01/INF1410PresentationStarUML.pdf>
- [23] : <https://fr.wikipedia.org/wiki/Bitbucket>
- [24] : <http://www.json.org/jsonfr.html>
- [25] : <https://developers.facebook.com/>
- [26] : <http://www.inextcom.fr/sortie-de-lapi-google-plus/>