**Dédicaces**

*A mes très chers parents, qui m’ont permis de devenir ce que je suis aujourd’hui,*

*A mes frères, qui ont été toujours à mes côtés*

*A mes collègues, qui ont rendu la période de stage agréable,*

*A mes amis,*

*Je dis merci et je leur dédie ce travail qui est l’aboutissement de plusieurs années de travail.*

**Wael MRABET**

**Remerciements**

*En tout premier lieu, je remercie le bon Dieu, tout puissant, de m’avoir donné la force pour survivre, ainsi que l’audace pour dépasser toutes les difficultés.*

*Je tiens à adresser mes plus sincères remerciements à Monsieur JGUIRIM Nabil, mon encadrant à l’école supérieur privée d’ingénierie et de technologies, pour le temps qu’il m’a consacré, pour son aide précieuse et l’encadrement sérieux dont il a fait preuve ainsi qu’à nos enseignants de l’ESPRIT à qui je dois ma formation.*

*Je tiens à exprimer ma profonde gratitude à* ***Monsieur KAMMOUN Slim****, mon tuteur, pour m’avoir intégré rapidement au sein de l’entreprise et m’avoir accordé toute sa confiance pour le temps qu’il m’a consacré tout au long de cette période et de répondre à toutes mes interrogations sans oublier sa participation aux réalisations de ce rapport.*

*Je tiens à remercier tout particulièrement et à témoigner toute ma reconnaissance aux personnes suivantes, pour l’expérience enrichissante et pleine d’intérêt qu’elles m’ont fait vivre durant ces six mois au sein de* ***AHEADIT****.*

*Enfin, j’adresse mes remerciements aux membres du jury pour m’avoir honoré en acceptant d’évaluer ce travail.*

*Sans oublier tous ceux qui ont participé de près ou de loin à l’accomplissement de ce travail.*

**Table des matières**

[Chapitre I :](#_Toc95772694) [Cadre général du projet 1](#_Toc95772695)

[1. Organisme d’accueil 2](#_Toc95772696)

[1.1- Présentation de AHEADIT 2](#_Toc95772697)

[1.2- Expertise 2](#_Toc95772698)

[2. Etat de l’art 2](#_Toc95772699)

[Chapitre II :](#_Toc95772700) [Analyse](#_Toc95772701) et specification des besoins

[1. Démarche suivie : Processus 2TUP 5](#_Toc95772702)

[1.1- Le processus 2TUP 6](#_Toc95772703)

[1.2- Pourquoi 2TUP 6](#_Toc95772704)

[2. Cahier de charge 7](#_Toc95772705)

[3. Modélisation du contexte 12](#_Toc95772706)

[3.1- Identification des acteurs 12](#_Toc95772707)

[3.2- Identification des messages 12](#_Toc95772708)

[4. Capture des besoins fonctionnels 15](#_Toc95772709)

[4.1- Identification de cas d’utilisation 15](#_Toc95772710)

[4.2- Description détaillée des cas d’utilisation 17](#_Toc95772711)

[5. Analyse 22](#_Toc95772712)

[5.1- Modèle statique 22](#_Toc95772713)

[5.2- Modèle dynamique 23](#_Toc95772714)

[Chapitre III :](#_Toc95772715) [Etude technique du projet 25](#_Toc95772716)

[1. Choix des technologies et l’environnement de travail 26](#_Toc95772717)

[1.1- Technologies 26](#_Toc95772718)

[1.2- Environnement de travail 26](#_Toc95772719)

[3. Patron de conception 29](#_Toc95772723)

[Chapitre IV :](#_Toc95772724) [Conception](#_Toc95772725)

[1. Hiérarchie 31](#_Toc95772726)

[1.1- Hiérarchie de l’application front end (Angular) 31](#_Toc95772727)

[1.2- Hiérarchie de l’application Back End (dot net core) 32](#_Toc95772728)

[Conclusion générale 41](#_Toc95772729)

[Perspective 41](#_Toc95772730)

**Liste des figures**

[**Figure 1 :**  Le processus 2TUP 6](#_Toc95772830)

[**Figure 2 :** Diagramme de cas d’utilisations global 15](#_Toc95772831)

[**Figure 3 :** Diagramme de cas d’utilisation gérer campagne publicitaire 16](#_Toc95772832)

[**Figure 4 :** Diagramme de cas d’utilisation gérer campagne business 16](#_Toc95772833)

[**Figure 5 :** Diagramme de classe global 22](#_Toc95772834)

[**Figure 6 :** Diagramme de séquence du cas d’utilisation Créer campagne 23](#_Toc95772835)

[**Figure 7 :** Diagramme de séquence du cas d’utilisation modifier état business 24](#_Toc95772836)

[**Figure 8 :** Diagramme de déploiement 27](#_Toc95772837)

[**Figure 9 :** Hiérarchie de l’application front end (Angular) 31](#_Toc95772838)

[**Figure 10 :** Hiérarchie de l’application Back End (dot net core) 32](#_Toc95772839)

[**Figure 11 :** Page d’accueil de système 33](#_Toc95772840)

[**Figure 12 :** Formulaire d’ajout d’un produit publicitaire. 34](#_Toc95772841)

[**Figure 13 :** Formulaire d’ajout d’un client et la liste des derniers client ajoutés dans l’application 35](#_Toc95772842)

[**Figure 14 :** Formulaire de création de campagne 36](#_Toc95772843)

[**Figure 15 :** Liste des campagnes crées avec ses états actuels 37](#_Toc95772844)

[**Figure 16 :** Interface modifier la campagne 38](#_Toc95772845)

[**Figure 17 :** Liste de tous les business de la campagne 39](#_Toc95772846)

[**Figure 18 :** Pop-up de modification business 40](#_Toc95772847)

|  |  |
| --- | --- |
| Chapitre I : | Cadre général du projet |

Ce chapitre aura comme objectif la présentation de l’organisme d’accueil **<<Ahead IT>>** et l’état de l’art.

# Organisme d’accueil

## Présentation de AHEADIT

La société AHEADIT est une entreprise de service et du numérique. AHEADIT a été créé en 2008, elle est née d’une grande passion par les nouvelles technologies et du Challenge. Elle a pour mission de concevoir et mettre en œuvre les meilleures solutions visant à améliorer la productivité, la rentabilité des entreprises sur leur marché.

## Expertise

AHEADIT a construit son expertise à l’aide d’une équipe solide et compétente dans divers domaines tel que :

**Système d’information :** plusieurs systèmes d’information sont réalisés bases sur les technologies JEE et .Net.

**Web :** AHEADIT a réalisé pour le compte de son partenaire un CMS riche est réussie dans le développement web. Les clients sont dans divers domaines tels que la cosmétique, de grands laboratoires pharmaceutiques ou des spécialistes de l’e-commerce.

**Finance :** La principale réalisation de AHEADIT dans ce domaine est RIZLAB

**Publicité :** AHEADIT a décidé de lancer un nouveau projet qui sert à digitaliser le business d’une agence de publicité française qui avait besoin d’une application de création et de suivi des campagnes publicitaires géolocalisé.

# Etat de l’art

La publicité au sein d’une entreprise permet de contribuer à atteindre des objectifs commerciaux. Elle a un pouvoir d’influence sur les consommateurs, elle peut et doit susciter, chez les destinataires, un nouveau besoin.

De nos jours, la publicité est omniprésente dans notre société. Il ne s’agit pas seulement des spots publicitaires que nous avons l’habitude de regarder tous les soirs, mais surtout toutes les campagnes de promotion auxquelles notre œil est soumis chaque jour dans notre quotidien.

Aujourd’hui la publicité est partout avec le marketing direct, les promotions, la presse, la radio, la télévision, les événements les annuaires et guide et évidement à l’internet.

La publicité vise principalement trois résultats :

* Promouvoir une marque, gagner en notoriété, concurrencer le marché.
* Promouvoir un produit ou un service.
* Convertir, c’est-à-dire inciter la cible à acheter.

Vue l’importance de la publicité, chaque entreprise doit donc avoir un budget dédié à sa communication notamment la publicité.

Aujourd’hui, il y en a pas mal de type de publicité Télévisée, radiodiffusée, affichage…

Ces types de publicité demandent souvent de gros budgets et parfois ils ne visent pas bien la cible demandée. Par exemple si Ralph Lauren ouvre une nouvelle boutique sur Paris et en crée une campagne publicitaire Télévisée. Le coût de cette compagne sera très élevé et la cible n’est pas bien cadrée c’est-à-dire les Marseillais, par exemple, n’ont pas besoin de savoir cette action.

Pour résoudre tous ces problèmes et répondre à la demande d’une agence de publicité française, AHEADIT à décider de créer, une application web de gestion et de suivi des campagnes publicitaires géolocalisées qui permet d’une part de digitaliser le bousinées de son client et de lui permettre de facilité la création de la campagne publicitaire et la fixation de la compagne qui correspond au budget client et de bien déterminer la cible client en se basant sur les points d’intérêts googleMaps d’autre part.

|  |  |
| --- | --- |
| Chapitre II : | Analyse et spécification des besoins |

Lors de cette étude je vais présenter la démarche suivie, le cahier de charge, la modélisation du contexte du système ainsi que l’étude fonctionnelle.

# Démarche suivie : Processus 2TUP

Pour la réalisation de mon projet, j’ai suivi le processus 2TUP.

Le 2TUP est processus de développement qui implémente le processus unifié.

Ce processus propose un cycle en Y. Il dissocie les aspects fonctionnels des aspects techniques permettant de répondre au changement continu imposé sur le système. Il s’articule sur 3 branches :

* **La branche fonctionnelle :** Ses principales étapes sont comme suit :
* **Etape de captures de besoins fonctionnels :** produit le modèle des besoins focalisés sur le métier des utilisateurs. Elle qualifie, au plus tôt le risque de produire un système qui ne répond pas aux besoins utilisateurs.
* **Etape d’analyse :** consiste à étudier les spécifications des besoins fonctionnels afin d’obtenir une idée de ce que le système va réaliser en terme de métier.
* **La branche Techniques :** Capitalise un savoir-faire technique indépendant du fonctionnel. Cette branche comporte les 2 étapes suivantes :
  + La capture des besoins techniques qui spécifie les contraintes, les besoins non fonctionnels et les choix conditionnant la conception du système.
  + La conception générique consiste à construire l’architecture technique du système qui doit être le moins dépendant de l’aspect fonctionnels. Construire le squelette du système en décrivant les composants nécessaires et leurs interactions. Cette étape se concrétise par la production d’un prototype.
* **La phase de réalisation :** c’est la branche au milieu, à l’issue des évolutions du modèle fonctionnel et l’architecture technique, la réalisation du système consiste à fusionner les résultats des 2 branches. Elle comporte les étapes suivantes :
* La conception préliminaire qui consiste à intégrer le modèle d’analyse dans l’architecture technique.
* La conception détaillée qui définit comment chaque composant
* Le codage et les tests des composants.

## Le processus 2TUP

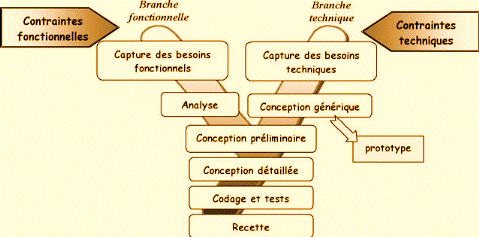
****

Figure  : Le processus 2TUP

## Pourquoi 2TUP

Pourquoi 2TUP et pas d’autre processus unifié, on a choisi pour les raisons suivantes :

* Implémente le processus unifié UP (Itératif et incrémentale)
* Répond aux contraintes de changement continu sur notre système
* Renforce la capacité d’évolution et de correction de notre système
* Notre système est développé par un seul développeur et un consultant

# Cahier de charge

Après un premier tour d’horizon des besoins exprimés par la société j’ai établi le cahier des charges ci-après qui consiste en la réalisation d’une application web qui permet l’implémentation des taches suivantes :

* **Gestion des produits publicitaires**
* **A l’administrateur**
* D’ajouter un produit publicitaire : l’administrateur du système peut ajouter un nouveau produit à partir de son espace admin.
* Editer un produit publicitaire : l’administrateur peut modifier les détails du produit publicitaire.
* Consulter la liste des produits publicitaires : le système permet à l’administrateur de consulter la liste des produits publicitaire depuis son espace admin.
* Consulter détails produit publicitaire : l’application permet à l’administrateur de consulter les détails d’un produit.
* **Gestion des clients**
* **A l’administrateur**
* D’ajouter un client : le système permet aux administrateurs d’ajouter un nouveau client depuis l’espace administrateur.
* Consulter la liste des clients : l’application permet de consulter la liste des clients à partir de l’espace administrateur.
* Modifier détails client : l’administrateur peut modifier les détails d’un client bien déterminé.
* Consulter détails Client : l’administrateur peut aussi consulter les détails d’un client sélectionné.
* **Gestion des Utilisateurs**
* **A l’administrateur**
* D’ajouter un utilisateur : le système permet aux administrateurs d’ajouter un nouvel utilisateur depuis l’espace administrateur.
* Modifier détails utilisateur : le système permet aux administrateurs de modifier les détails d’un utilisateur.
* Activer un utilisateur : l’administrateur peut activer un utilisateur depuis l’espace administrateur.
* Désactiver un utilisateur : l’administrateur peut désactiver un utilisateur depuis l’espace administrateur.
* Consulter liste utilisateur : l’administrateur peut consulter la liste des utilisateurs de l’application.
* **Gestion des campagnes publicitaires**
* **A l’administrateur**
* Création d’une campagne publicitaire : le système permet à l’administrateur de créer une nouvelle campagne publicitaire en spécifiant toutes les caractéristiques de ce derniers (budget, titre, description, date de lancement, la région, les villes, le client, les types de produits publicitaires de la compagne, les types de business (station de métro, boulangerie…) ainsi que le taux de pénétration.
* Editer campagne publicitaire : l’application permet aux administrateurs de modifier la campagne créer afin de le fixer selon le besoin du client la zone et les villes cibles tout en respectant le budget du client.
* Consulter liste campagne publicitaire : le système permet d’afficher la liste des campagnes publicitaires.
* Lancer une campagne publicitaire : une fois le devis de la compagne est généré et validé par le client, l’administrateur peut lancer la réalisation de la campagne.
* Consulter détails campagne : le système permet d’afficher les détails de la campagne via l’espace administrateur.
* Dupliquer campagne publicitaire : le system permet à l’administrateur de dupliquer une campagne à partir de son espace administrateur.
* Clôturer campagne : l’administrateur peut clôturer une campagne publicitaire depuis son espace administrateur.
* **Gestion des produit campagne**
* **A l’administrateur**
* Ajouter supprimer un type de produit campagne : tant que la campagne n’est pas est toujours dans l’état brouillon, le système permet à l’administrateur de rajouter ou supprimer les types de produit de la campagne afin que le cout de celle-ci être adapté au budget client.
* Consulter liste des types de produit campagne : le système permet de consulter la liste des types de produit publicitaire affectés à une campagne bien déterminée.
* Rechercher un type de produit campagne : le système permet de rechercher un type de produit affecté à une campagne bien déterminée
* Modifier un type de produit campagne : le système permet à l’utilisateur de modifier le prix unitaire de produit ainsi que le nombre de produit dans chaque business campagne
* **Gestion des villes cibles campagne**
  + **A l’administrateur**
* Ajouter ou supprimer une ville cible : pour que le client puisse avoir une meilleure flexibilité dans la définition de la compagne, le système permet d’ajouter ou supprimer des villes des villes de la campagne tout en mettant à jour le cout total de la campagne.
* Consulter détails ville cible : l’application permet à l’administrateur de consulter les détails d’une ville cible dans un écran contenant la répartition des business de la campagne et tout information qui concerne la ville de la campagne.
* Rechercher une ville campagne : le système permet de rechercher une ville affectée à une campagne bien déterminée.
* Consulter liste des villes campagne : le système permet d’afficher la liste des villes d’une campagne.
* **Gestion des types de business campagne**
  + **A l’administrateur**
* Ajouter supprimer un type de business : le système permet de supprimer ou ajouter un type de business a la campagne tant que son état est brouillon.
* Consulter liste des types business de campagne : l’application permet de consulter la liste des types de business d’une campagne bien déterminé.
* Recherche type de business compagne : l’application permet de rechercher un type de business campagne d’une campagne bien déterminée.
* **Gestion des business campagne**
  + **A l’administrateur**
* Inclure et exclure un business de la campagne : En se basant sur POI de google maps le système permet d’exclure ou inclure un business dans l’exécution de la campagne tant que l’état de la campagne est brouillon.
* Consulter liste des campagne business : l’application permet de consulter la liste des business d’une campagne bien déterminée.
* Modifier état campagne business : l’administrateur peut modifier l’état d’une campagne business (A faire, en cours, fini ou bien annulé).
* Affecter photo a un business campagne : le système permet d’affecter une photo de réalisation a un business campagne.
* Rechercher campagne business : le system permet d’effectuer une recherche multicritère sur les business d’une campagne publicitaire.
* **Gestion des devis**
  + **A L’administrateur**
* Générer devis campagne : après avoir créé une campagne dont l’état brouillon, le système permet de générer un devis contenant le cout total de la campagne.
* Consulter liste devis : l’administrateur peut consulter la liste des devis des campagnes publicitaires depuis l’espace administrateur.
* Consulter détails devis : l’administrateur peur consulter le devis d’une campagne publicitaire.
  + **Au client**
* Valider devis : l’application permet au client de valider le devis campagne via son espace client.
* Consulter devis : l’application permet au client de consulter seulement les devis de ses campagnes publicitaire.
* Consulter détails devis : le client peut consulter les détails du devis de son campagne publicitaire depuis l’espace client.
* **Gestion des factures**
  + **A l’administrateur**
* Génération de facture : lorsque le client valide le devis d’une campagne une facture est généré automatiquement par le système
* Consultation de la liste facture : le système permet aux administrateurs de consulter la liste des factures
* Consulter détails facture : le système permet de consulter les détails d’une facture

# Modélisation du contexte

## Identification des acteurs

Un acteur représente l’abstraction d’un rôle joué par des entités externes (Utilisateur, dispositif matériel ou bien un autre système) qui interagissent directement avec le système.

Un acteur peut interagir directement avec le système, en émettant ou recevant des messages porteurs de données

Notre système comporte 4 principaux acteurs :

* **Le Client :** Les taches que le client peut effectuer se résument dans la création d’un campagne publicitaire, consultation de ses campagnes, consulter le détails campagne, il est aussi autorisé d’éditer les critères de la compagne ainsi que la consultation et la validation du devis campagnes.
* **L’administrateur :** c’est notre acteur principal, il est autorisé de tout faire, il s’occupe de la gestion des utilisateurs, des clients, des types de produit publicitaire, des campagnes publicitaires

## Identification des messages

Un message représente la spécification d’une communication unidirectionnelle entre les objets qui transporte l’information afin de déclencher une activité chez le récepteur.

Un message est normalement associé à deux occurrences d’évènements : un évènement de d’envoie et un évènement de réception.

Cette action de message est applicable pour décrire les interactions de plus haut niveau entre les acteurs et le système.

* **Message émis par le système**
  + Gestion des types de produit
    - Liste des types de produit
    - Détails type de produit
  + Gestion des clients
    - Liste des clients
    - Détails Client
  + Gestion des utilisateurs
    - Liste des utilisateurs
    - Détails utilisateur
  + Gestion des campagnes publicitaires
    - Liste des campagnes publicitaires
    - Détails campagne publicitaire
  + Gestion des devis
    - Liste des devis
    - Détails devis
  + Gestion des factures
    - Liste des factures
    - Détails facture
* **Message reçus par le système**
  + Gestion des types de produit
    - Création d’un type de produit
    - Modification d’un type de produit
  + Gestion des clients
    - Création d’un client
    - Modification d’un client
  + Gestion des utilisateurs
    - Création d’un utilisateur
    - Modification d’un utilisateur
  + Gestion des campagnes publicitaires
    - Création d’une campagne pub
    - Modification d’une campagne
    - Ajout ou suppression d’un type de produit d’une campagne
    - Ajout ou suppression d’une ville d’une campagne
    - Ajout ou suppression d’un type de business d’une campagne
    - Lancement d’exécution d’une campagne
    - Clôture d’une campagne publicitaire
    - Changement d’état d’un campagne business
  + Gestion de devis
    - Validation de devis d’une campagne publicitaire

# Capture des besoins fonctionnels

## Identification de cas d’utilisation

Un cas d’utilisation est un ensemble de séquences d’actions réalisées par le système et produise un résultat observable intéressant pour un acteur particulier

Un cas d’utilisation modélise un service rendu par le système. Il exprime les interaction acteur/système et apporte une valeur ajoutée à l’acteur concerné.

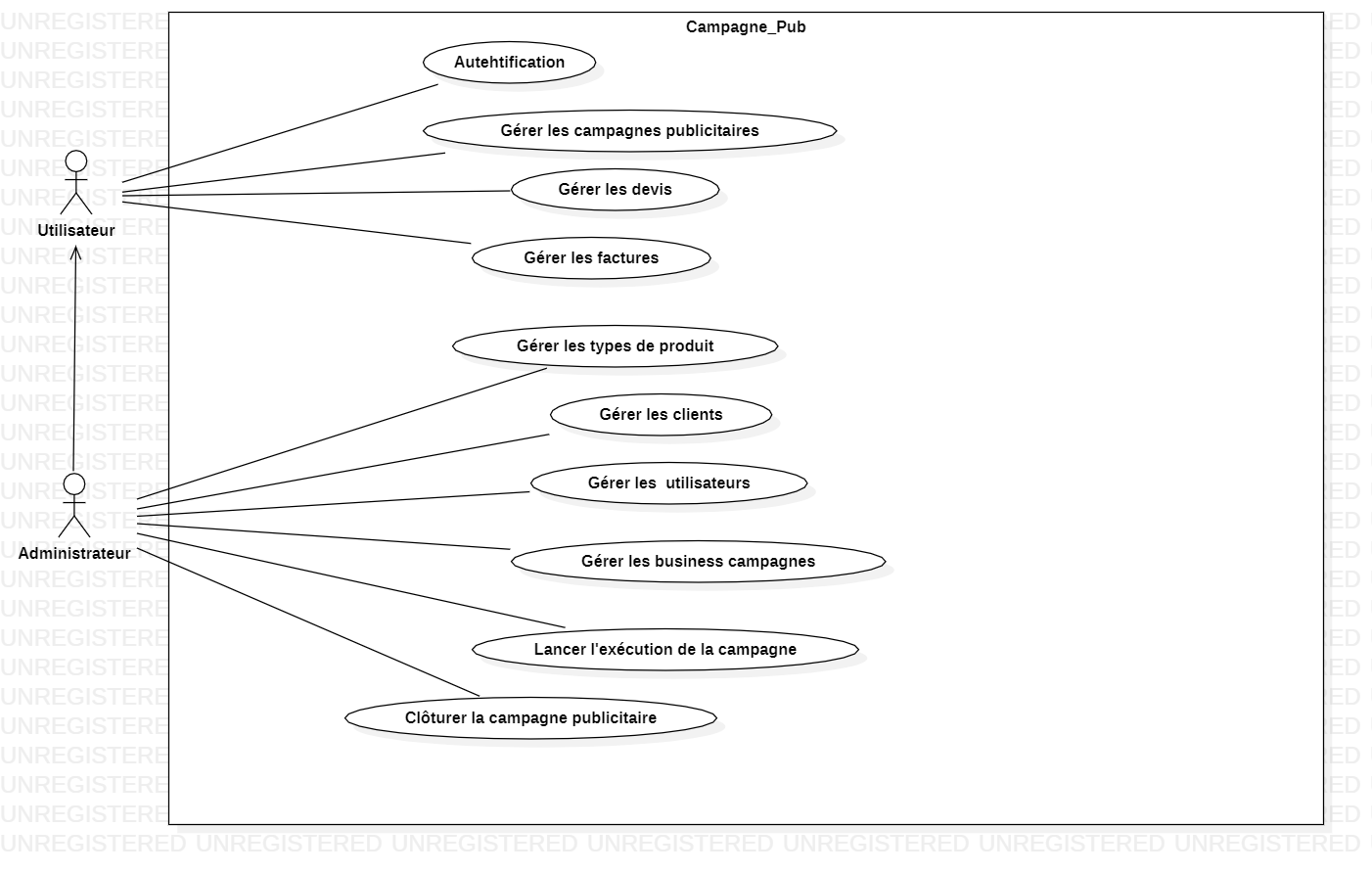


Figure  : Diagramme de cas d’utilisations global

Le diagramme de cas d’utilisation globale permet de décrire les différentes fonctionnalités implémentées par notre système. Il décrit aussi les droits de chaque acteur qui peut interagir avec notre système. Dans notre cas, le système permet aux utilisateurs de s’authentifier de gérer les campagnes, de gérer les devis et les factures. L’administrateur de notre système hérite toutes les fonctionnalités de l’utilisateur et il permit d’effectuer d’autres opérations sur le système tels que la gestion des utilisateurs, la gestion des types de produit publicitaires, le lancement d’exécution et le clôture des campagnes publicitaire et ainsi de suite.

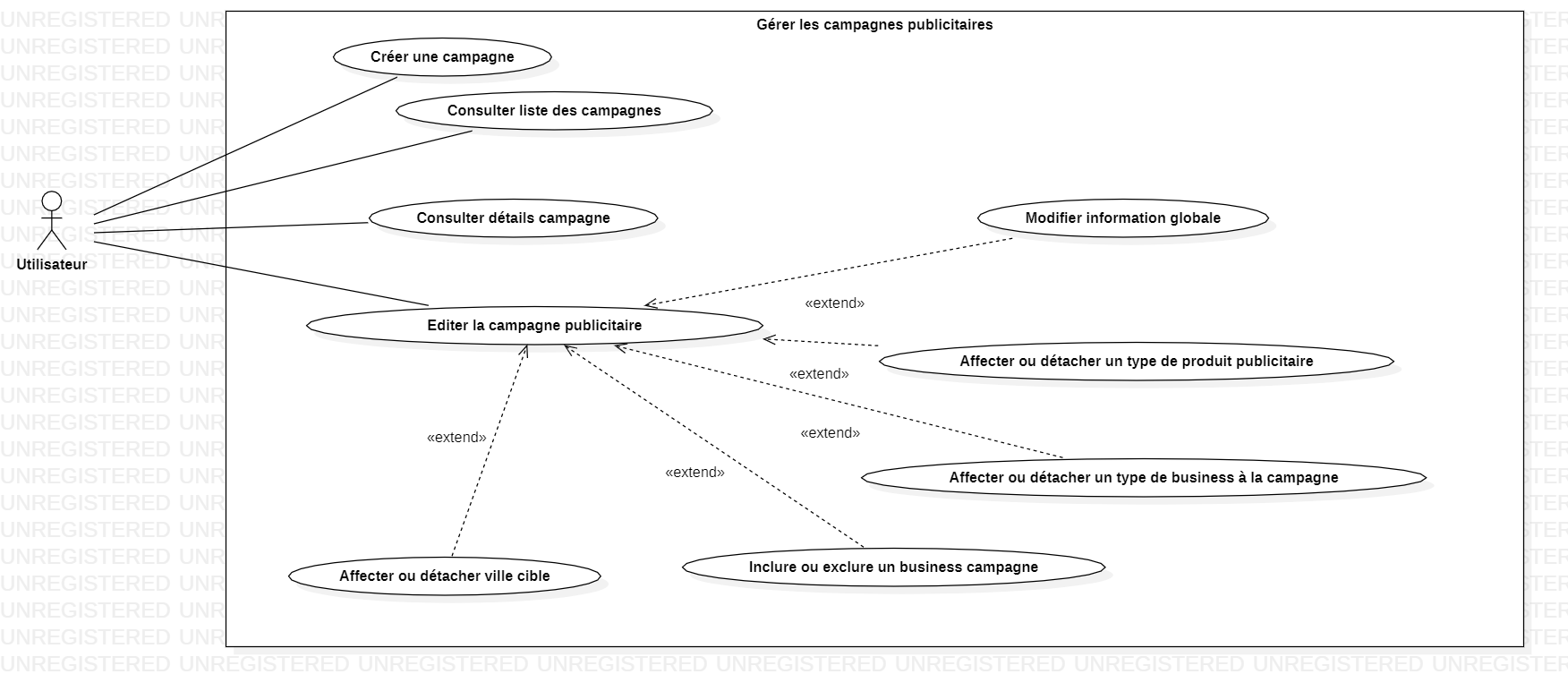
****

Figure  : Diagramme de cas d’utilisation gérer campagne publicitaire

Le diagramme de cas d’utilisation ci-dessus est une représentation graphique détaillé de la fonctionnalité générale présentée dans le diagramme de cas d’utilisation globale ou bien générale. Dans notre cas, le cas d’utilisation gérer les campagnes publicitaires est une fonctionnalité de bas de notre système, cette fonctionnalité inclus les fonctionnalités suivantes : Créer campagne, consulter liste des campagnes, consulter détails campagne, et finalement, éditer campagne.

Le cas d’utilisation éditer campagne englobe les fonctionnalités de modification de la campagne lorsqu’elle l’état de la campagne est brouillon, en effet, l’édition d’une campagne se fait en réalisant une des opérations suivantes : l’affectation et le détachement d’une ville cible, inclure ou exclure un type de business campagne à la campagne…

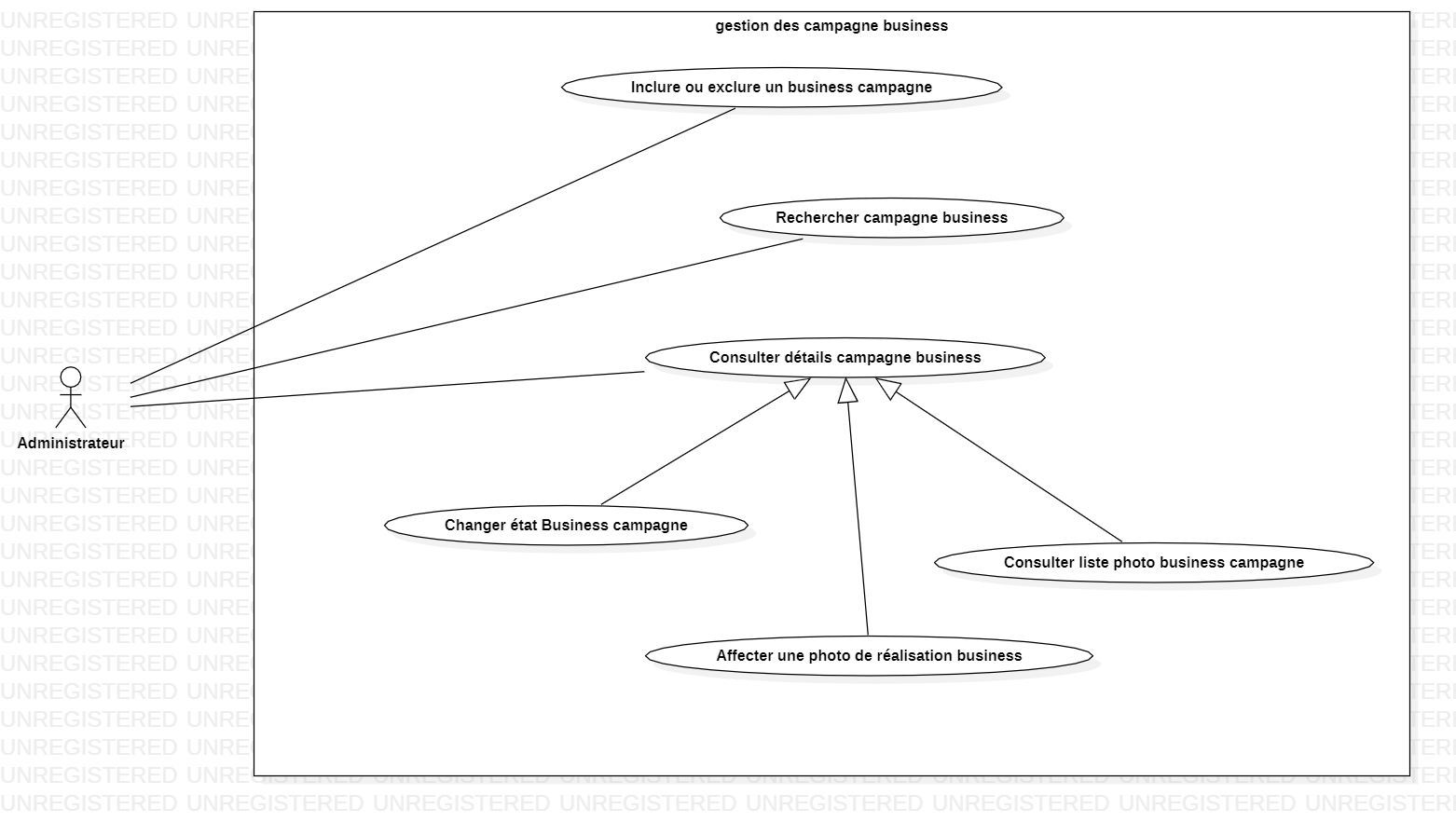
****

Figure  : Diagramme de cas d’utilisation gérer campagne business

Le diagramme de cas d’utilisation ci-dessus est le raffinement du cas d’utilisation gérer campagne business. En effet, c’est un diagramme détaillé de ce cas.

Dans ce cadre, le système permet à l’administrateur d’inclure ou exclure un business campagne lors de la réalisation de la campagne c’est-à-dire quand l’état de la campagne est en cours. Le système aussi permet à l’administrateur de rechercher un business avec un mot clé, de consulter les détails d’un business et ensuite lui permet changer son état, affecter des photos qui décrit la réalisation de publicité dans ce business…

## Description détaillée des cas d’utilisation

La description détaillée consiste à recenser toutes les interactions entre le système et les acteurs de façon textuelle. Le cas d’utilisation doit avoir un début et une fin clairement identifié. Il doit préciser quand ont lieu les interactions entre acteurs et système, et quels sont les messages échangés. Il faut également préciser les variantes possibles, telles que les différents cas nominaux, les cas alternatifs, les cas d’erreurs, tout essayant d’ordonner séquentiellement les descriptions, afin d’améliorer la lisibilité. Chaque unité de description de séquences d’actions est appelée enchainement. Un scenario représente une succession particulière d’enchainements, qui s’exécute du début a la fin du cas d’utilisation.

Dans cette partie nous allons nous limiter au cas d’utilisation Gestion des campagnes publicitaires.

Sommaire d’identification :

|  |
| --- |
| **Titre :** Créer une campagne publicitaire  **But :** Créer une campagne publicitaire  **Résumé :** permettre à l’utilisateur de créer une campagne publicitaire  **Acteur :** Utilisateur |

Description d’enchaînements :

|  |
| --- |
| **Préconditions :** L’utilisateur doit être authentifié.  **Postconditions :** Campagne publicitaire créé.  **Enchainement nominal :** Ce cas d’utilisation commence quand l’utilisateur demande au système de créer une campagne publicitaire.   1. L’utilisateur demande au système de créer une campagne 2. Le système affiche le formulaire d’ajout de la campagne 3. L’utilisateur rempli le formulaire en saisissant le titre, le budget, l’objectif et la description et la date de réalisation souhaitée 4. L’utilisateur choisit la région cible connu par google map 5. Le système chercher et renvoi la liste des villes de la région 6. L’utilisateur cherche et choisi les villes cible par le code postal et ou par le nom de la ville 7. L’utilisateur choisi les types de business cible (boulangerie. Epicerie, station de bus…) 8. L’utilisateur valide les détails de la nouvelle campagne 9. Le système vérifie les données obligatoires 10. Le système enregistre la nouvelle campagne avec état brouillon   **Enchainements alternatifs :**  **A1 :** Champs invalides  L’enchainement A1 démarre au point (9) du scenario nominal   1. Le système demande à l’utilisateur de vérifier les champs obligatoires 2. Le scénario nominal reprend au point (8) |

Sommaire d’identification :

|  |
| --- |
| **Titre :** Consulter liste des campagnes publicitaires  **But :** Consulter liste des campagnes publicitaires  **Résumé :** permettre à l’utilisateur de consulter la liste des campagnes publicitaires  **Acteur :** Utilisateur |

Description d’enchaînements :

|  |
| --- |
| **Préconditions :** L’utilisateur doit être authentifié.  **Postconditions :** Liste des campagnes publicitaire affichées.  **Enchainement nominal :** Ce cas d’utilisation commence quand l’utilisateur demande au système de consulter la liste des campagnes publicitaires.   1. L’utilisateur demande au système de consulter la liste des campagnes 2. Le système affiche la liste des campagnes paginé   **Enchainements alternatifs :**  Aucun enchainement alternatif à signaler. |

Sommaire d’identification :

|  |
| --- |
| **Titre :** Consulter détails campagne d’une campagne publicitaire  **But :** Consulter détails d’une campagne publicitaire  **Résumé :** permettre à l’utilisateur de consulter les détails d’une campagne publicitaire  **Acteur :** Utilisateur |

Description d’enchaînements :

|  |
| --- |
| **Préconditions :** L’utilisateur doit être authentifié.  **Postconditions :** afficher les détails d’une campagne publicitaire.  **Enchainement nominal :** Ce cas d’utilisation commence quand l’utilisateur demande au système de consulter les détails d’une campagne publicitaire.   1. L’utilisateur demande au système de consulter la liste des campagnes 2. Le système affiche la liste des campagnes 3. L’utilisateur cherche une campagne bien déterminée est clique sur consulter détails 4. Le système renvoi un écran qui affiche la campagne demandée contenant tous les détails de la campagne   **Enchainements alternatifs :**  Aucun enchainement alternatif à signaler. |

Sommaire d’identification :

|  |
| --- |
| **Titre :** Modifier une campagne  **But :** Consulter détails d’une campagne publicitaire  **Résumé :** permettre à l’utilisateur de consulter les détails d’une campagne publicitaire  **Acteur :** Utilisateur |

Description d’enchaînements :

|  |
| --- |
| **Préconditions :**  L’utilisateur doit être authentifié.  Etat de la campagne doit être brouillon.  **Postconditions :** Modifier les détails de la campagne afin d’adapter le cout a son budget.  **Enchainement nominal :** Ce cas d’utilisation commence quand l’utilisateur demande au système de consulter la liste des campagnes publicitaires.   1. L’utilisateur demande au système de consulter la liste des campagnes 2. Le système affiche la liste des campagnes 3. L’utilisateur cherche une campagne bien déterminée est clique sur modifier campagne 4. Le système renvoi un écran qui permet de modifier la campagne demandée 5. L’utilisateur peut modifie les paramètres globaux de la compagne 6. L’utilisateur valide la modification effectuée 7. Le système vérifie la modification 8. Le système recalcule le cout total de la campagne 9. Le système enregistre les modifications 10. Le système affiche de nouveau la dernière version de la campagne   **Enchainements alternatifs :**  **A1 :** Champs invalides  L’enchainement A1 démarre au point (7) du scenario nominal   1. Le système demande à l’utilisateur de vérifier les champs obligatoires 2. Le scenario nominal reprend au point (6) |

# Analyse

L’objectif de l’analyse est de détailler les spécifications réalisées lors de la capture des besoins fonctionnels. Elle consiste à modéliser ces spécifications selon les vues statique et dynamique.

5.1 Modèle **dynamique**

Le modèle dynamique est une vision microscopique du fonctionnement du système. Il sert à mettre en évidence les interactions entre les objets du système. Il intervient après la définition du modèle statique.

Je vais présenter les diagrammes de séquences pour les cas d’utilisation : Créer campagne et modifier état de business.

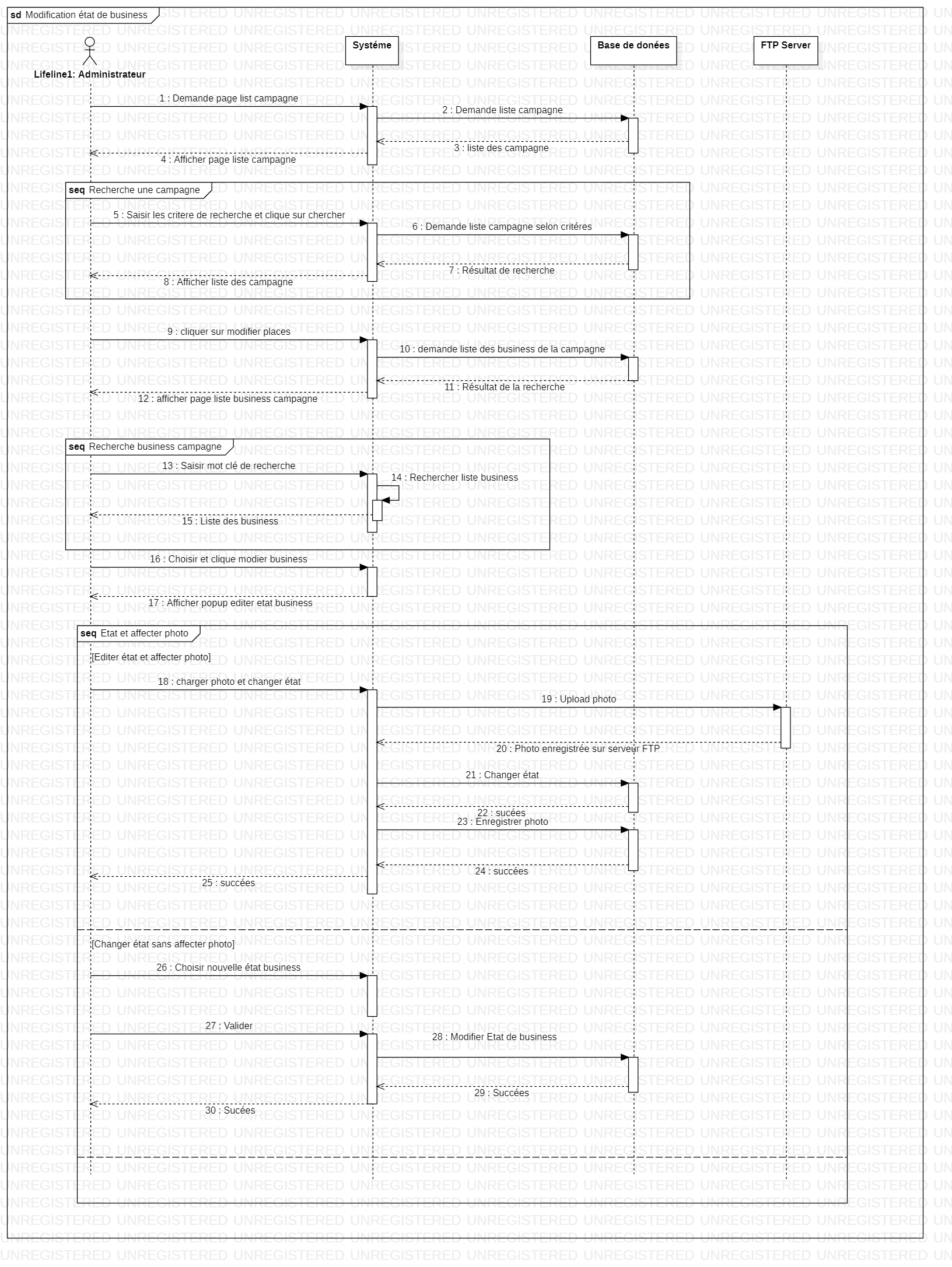
**Diagramme de séquence du cas d’utilisation Créer campagne**

**Une image contenant texte

Description générée automatiquement**

**Figure 6 :** Diagramme de séquence du cas d’utilisation Créer campagne

**Diagramme de séquence du cas d’utilisation modifier état business**

****

**Figure 7 :** Diagramme de séquence du cas d’utilisation modifier état business

|  |  |
| --- | --- |
| Chapitre III : | *Etude technique du projet* |

Dans cette étape, je vais présenter l’étude technique du projet. Le choix technique et l’environnement du travail d’abord. Ensuite, je présente les patrons de conception utilisés dans ce sujet.

# Choix des technologies et l’environnement de travail

## Technologies

**Net Core** : c’est un Framework gratuit qui permet de développer plusieurs type d’application Web, Web Api, Windows Forms … Il est multiplateforme et il prend en charge trois langage de programmation tels que C#, F# et Visual Basic VB.

**Angular** : c’est un Framework frontal open source créée pour les applications web moderne et puissante. Angular combine des modèles déclaratifs, une injection de dépendance, des outils de bout en bout et des modules intégrées afin de résoudre les problèmes de développement.

**SQL Server** : c’est notre système de gestion de base de données relationnel qui est a été créé par Microsoft.

## Environnement de travail

**Visual Studio 2019 :** c’est un environnement de travail intégré (IDE), il est un ensemble complet d’outils de développement permettant de générer des applications web, bureautiques ou bien mobile.

**Visual Code :** c’est un éditeur de code extensible créé par Microsoft, il permet de développer des applications de diffèrent technologie. Dans notre cas je l’ai utilisé lors du développement de la partie frontend.

# Patron de conception

En génie logiciel, les patrons de conception appelés design patterns en anglais, ce sont des solutions reproductibles au problèmes courant dans la conception de logiciels, un patron de conception n’est pas un modèle fini il peut être transformer directement dans le code. Ils sont aussi des modèles de solution aux problèmes fréquentes dans plusieurs types de projets.

Voici la liste des patrons de conception utilisés :

* + **Repository pattern :**

Ce design pattern répond à besoin un besoin d’accès aux données stockées en base de données. L’objectif principal est de séparer la couche d’accès aux données de la couche métier. Il expose plusieurs méthodes s’appuyant sur le modèle CRUD (Create, Read, Update, Delete). Un repository est cantonné à la manipulation d’une entité spécifique. Pour cela on besoin de coder un repository par entité gérée.

* + **Generic repository :**

Ce design pattern permet d’accélérer le process de création de la couche accès aux données. Il définit des méthodes génériques qui permet de réaliser la plupart des opérations commune tels que Save, Read, Update, Delete… Il est simple, flexible et il permet de développer une grande partie de la couche accès aux données sans écrire beaucoup de code.

* + **Generic Service :**

Il ressemble beaucoup au design pattern Generic Repository mais il est implémenté dans la couche métier. Il définit des méthodes génériques qui permet de réaliser la plupart des opérations commune tels que Create, Read, Update, Delete…

* + **DTO**

DTO (data transfer objects) c’est un patron de conception qui permet de de simplifier le transfert des données entre les sous-systèmes d’une application logiciel. Il est très important de l’utiliser lorsqu’il s’agit d’une application web api.

* + **Dependency Injection**

Ce patron de conception nous permet de développer des composent logiciel faiblement couplée. En autre termes, nous pouvons dire que ce patron de conception permet de réduire le couplage fort entre les composants logiciels. Par conséquence, nous pouvons facilement gérer les modifications futures et autres complexités de notre application.

|  |  |
| --- | --- |
| Chapitre IV : | *Conception* |

Dans ce Chapitre, je vais présenter la conception de notre systéme en se basant sur le diagramme de classe, l’architecture physique et les architectures logiques coté FrontEnd et coté Back End.

# Diagramme de classe

En se basant sur l’étude préliminaire et de la description détaillées des cas d’utilisation j’ai élaboré le diagramme de classe suivant :

Une image contenant texte, mots croisés, tableau de points

Description générée automatiquement

**Diagramme de classe**

# Architecture physique

L’architecture physique décrit les composants logiciel déployés sur les composants matériels.



**Diagramme de déploiement**

**Architecture logique**

L’architecture logique est la décomposition logique du projet en couche.

* + 1. **Architecture logique Back End**
* **La couche présentation Web Api** :

C’est la couche de liaison entre les applications client est notre système backend. Cette couche contient les contrôleur api qui reçoivent les requetés http et répond par les réponses http elle contient aussi les dtos (data Transfer Object) et les mapper qui sert de convertie de modèle externe connu par les clients en modèle interne connu que par l’application backend.

* **La couche Service BLL :**

C’est la couche métier de notre système, cette couche contient tous les traitement métier de notre système, cette couche ne peut pas accéder à la base de données qu’à travers la couche data.

* **La couche Data :**

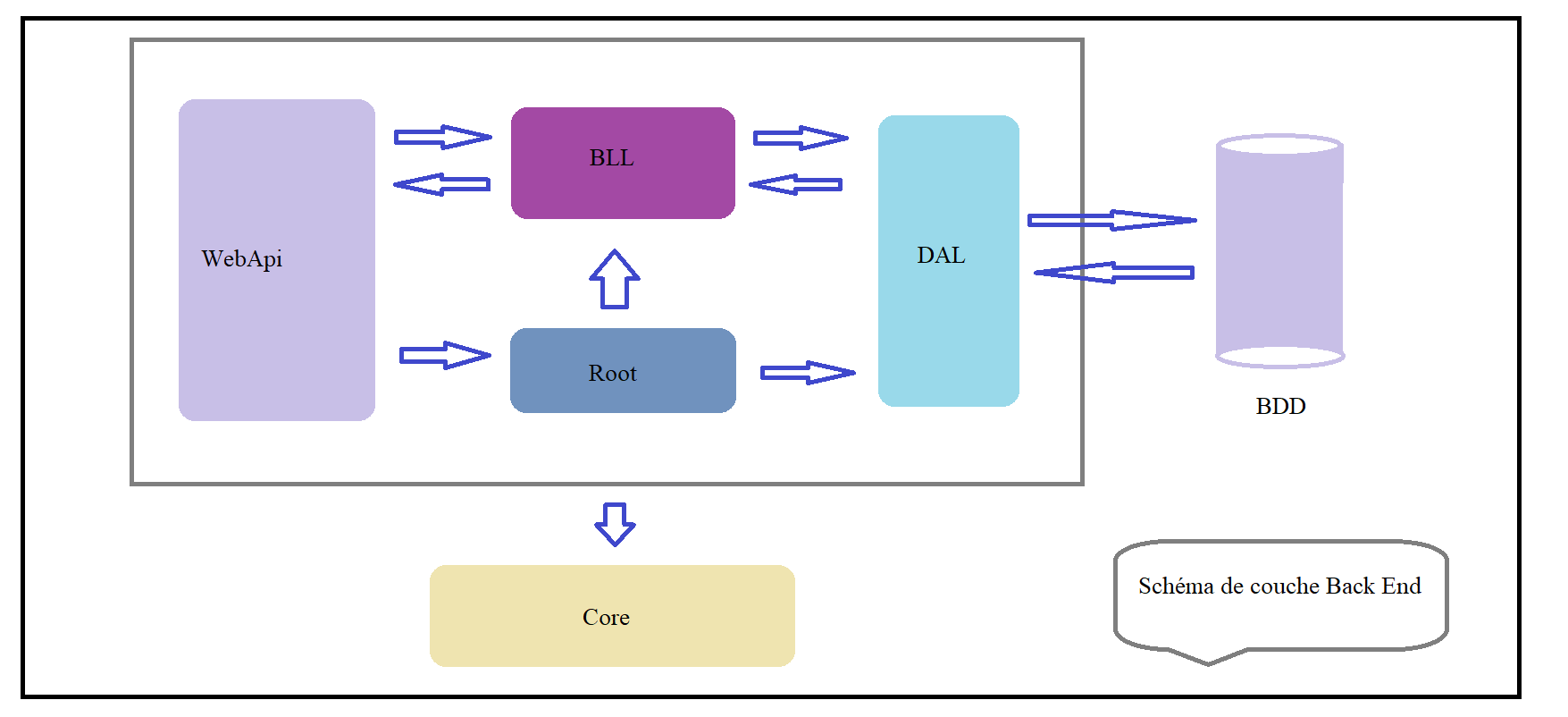
C’est la couche d’Access aux données. L’accès a la base de données ce fait que par cette couche, elle contient le contexte de base de données, les conventions les configurations et les scripts de migration base de données.

* **La couche Core :**

C’est le core de notre application, on peut l’appeler aussi la couche domaine. Elle contient les modelés, les Énumération, les types complexes et les exceptions spécifiques.

* **La couche Root :**

C’est une couche intermédiaire qui sert à faire l’injection des dépendances (le contexte de base de données, les repositories, et les services).

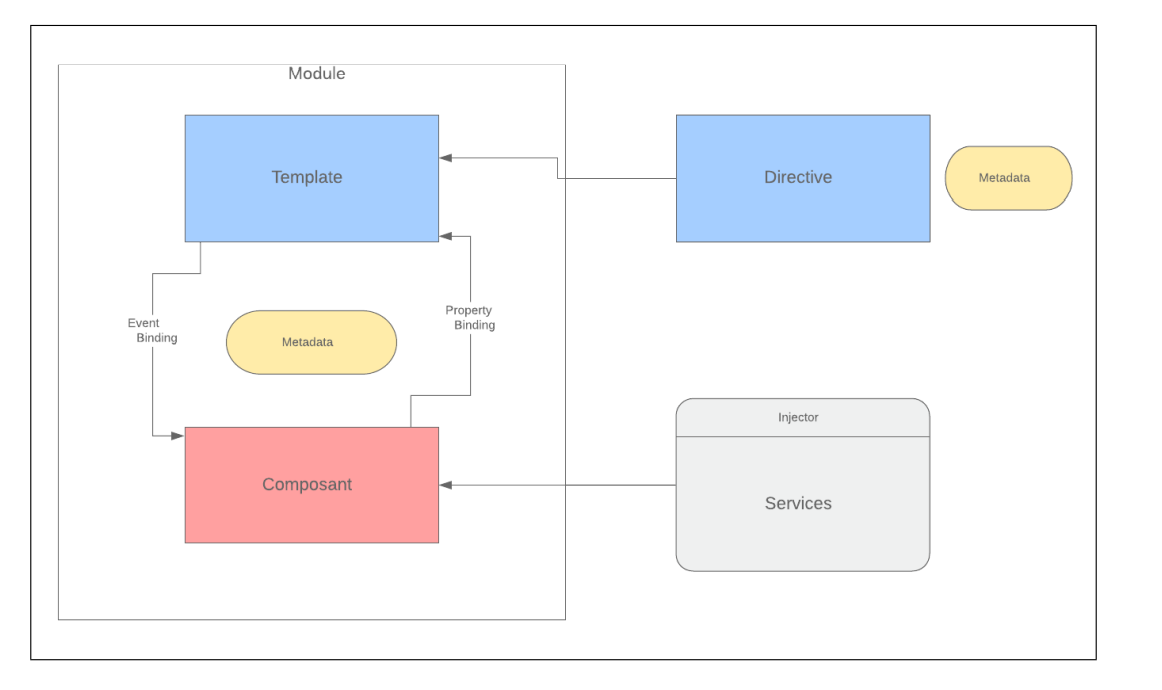


**Architecture logique du BackEnd**

* + 1. **Architecture logique Front End**
    - **La couche helpers :**

Cette couche contient des classes de service qui sert à nous aider à réaliser de résoudre des complexités techniques de développement. Elle contient le authGuard.ts qui sert à faciliter gestion d’accès aux différents pages du système, elle contient aussi le jwtInterceptor.ts qui permet d’intercepter chaque requête http et de la modifier puis la laisser continuer son chemin.

* + **La couche components : C**ette couche contient tous les composants qui seront utilisé dans notre application. Chaque composant est présenté dans un dossier qui contient 4 fichiers le TS, Html, SCSS, et le Spec.ts.
  + **La couche service :** cette couche contient tous les services client http de notre application.
  + **La couche modèle :** cette couche contient les modèles équivalents aux Dtos de la partie backend.



**Architecture logique du FrontEnd**

|  |  |
| --- | --- |
| Chapitre V : | *Réalisation* |

Dans cette dernière partie, tout d’abord je vais présenter les hiérarchies des sous-systèmes de notre application.

Ensuite, je montrerai quelques imprimes écran qui décrivent quelques fonctionnalités de base notre système.

Enfin, je vais finir par une conclusion et une la partie perspective.

# Hiérarchie

## Hiérarchie de l’application front end (Angular)

Une image contenant texte

Description générée automatiquement

Figure  : Hiérarchie de l’application front end (Angular)

## Hiérarchie de l’application Back End (dot net core)

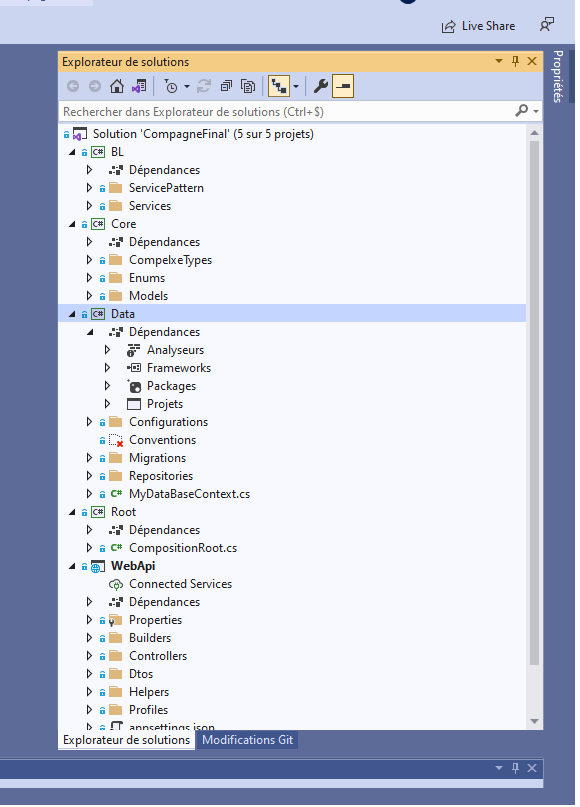
****

Figure  : Hiérarchie de l’application Back End (dot net core)

Fonctionnalités et imprimes écran

* **Page Accueil**

L’image ci-dessous montre la page d’accueil de notre système. A gauche, on trouve les menus qui regroupe les différents modules de notre système, en haut, c’est la barre de navigation et au milieu on trouve une image qui signifie le monde de publicité.

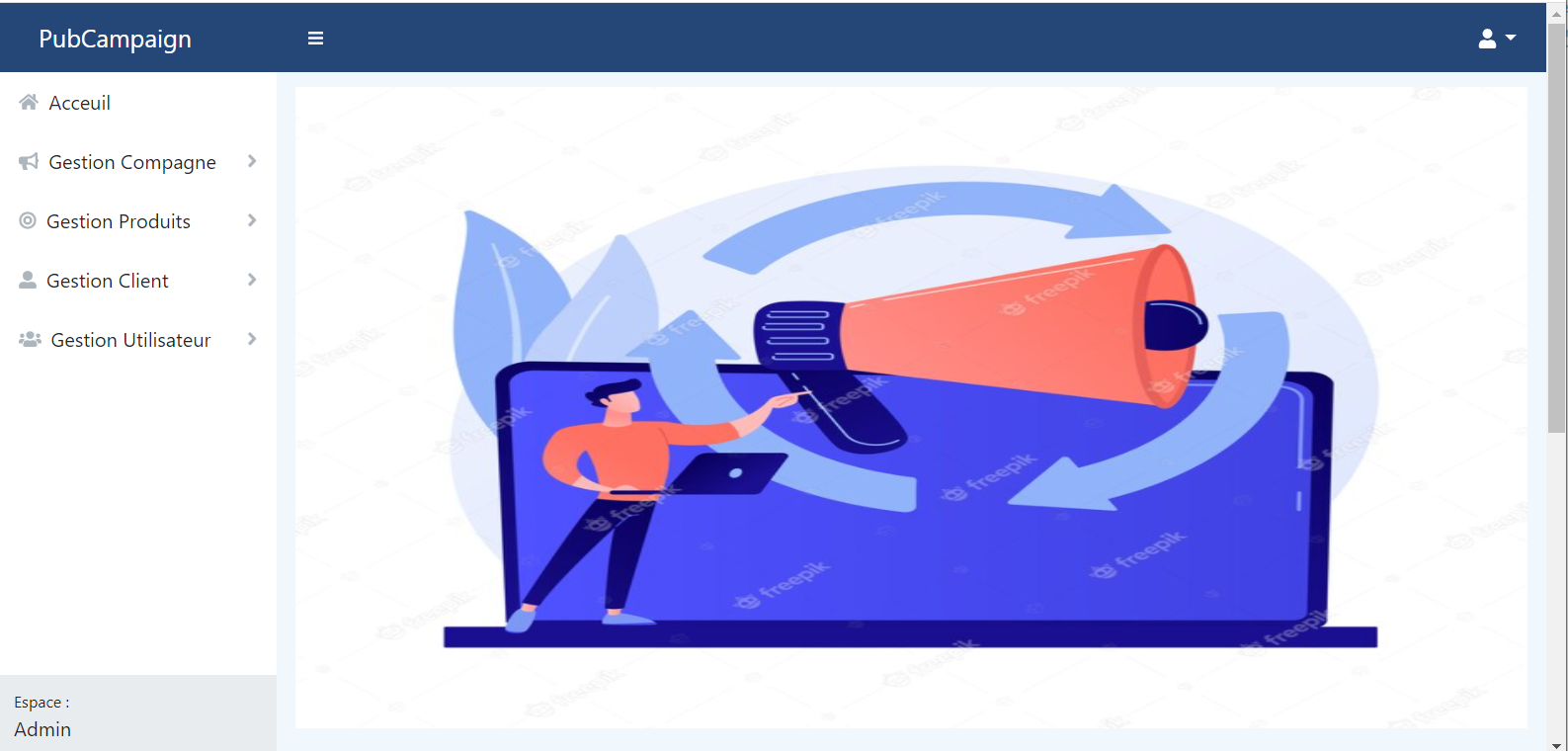


Figure  : Page d’accueil de système

* **Fonctionnalité Ajout type de produit publicitaire**

Ci-dessous une image qui décrit le formulaire d’ajout d’un produit publicitaire.

Une image contenant texte

Description générée automatiquement

Figure  : Formulaire d’ajout d’un produit publicitaire.

* **Fonctionnalité Ajout client**

Une fois l’administrateur clique sur nouveau client l’application affiche un écran qui contient un formulaire d’ajout d’un client et la liste des derniers client ajoutés dans l’application.

Une image contenant texte

Description générée automatiquement

Figure  : Formulaire d’ajout d’un client et la liste des derniers client ajoutés dans l’application

* **Fonctionnalité Création campagne publicitaire**

L’utilisateur clique sur nouveau campagne, le système affiche un formulaire de création de campagne, l’utilisateur remplit les champs du formulaire et choisit, le client, le type de produit, la région et les villes cible et clique sur valider pour créer une nouvelle campagne publicitaire.

Une image contenant texte

Description générée automatiquement

Figure  : Formulaire de création de campagne

* **Fonctionnalité liste des campagnes publicitaire**

Une fois l’utilisateur clique sur List des campagnes, le système le redirige vers un écran qui présente la liste des campagnes crées avec ses états actuels.

Cet écran permet à l’utilisateur d’effectuer une rechercher multicritère sur une campagne bien déterminer, de consulter les détails d’une campagne, consulter le devis d’une campagne ainsi que lui rediriger vers un écran de modification campagne lorsqu’il est en état brouillon de consulter et enfin, modifier la liste des business d’une campagne en cours de réalisation.

Une image contenant table

Description générée automatiquement

Figure  : Liste des campagnes crées avec ses états actuels

* **Fonctionnalité modifier campagne publicitaire état brouillon**

Une fois l’utilisateur clique sur détails campagne à état brouillon, le système affiche un écran permettant de modifier la campagne. Cet écran permet de modifier la région cible, les villes cible, les type de business cible ainsi que les type de produit publicitaire qui seront utiliser lors de la réalisation de la campagne.

Finalement, Cet écran permet aussi de consulter le devis d’une campagne en cliquant sur le bouton consulter devis.

Une image contenant texte

Description générée automatiquement

Figure  : Interface modifier la campagne

* **Fonctionnalité rechercher campagne business**

Lorsque la réalisation de la campagne est déclenchée, l’administrateur peut consulter la liste des business de la campagne en cliquant sur liste des places dans l’écran liste des campagnes.

Cet écran montre la liste de tous les business de la campagne et permet aussi de rechercher un business spécifique et finalement de le modifier.

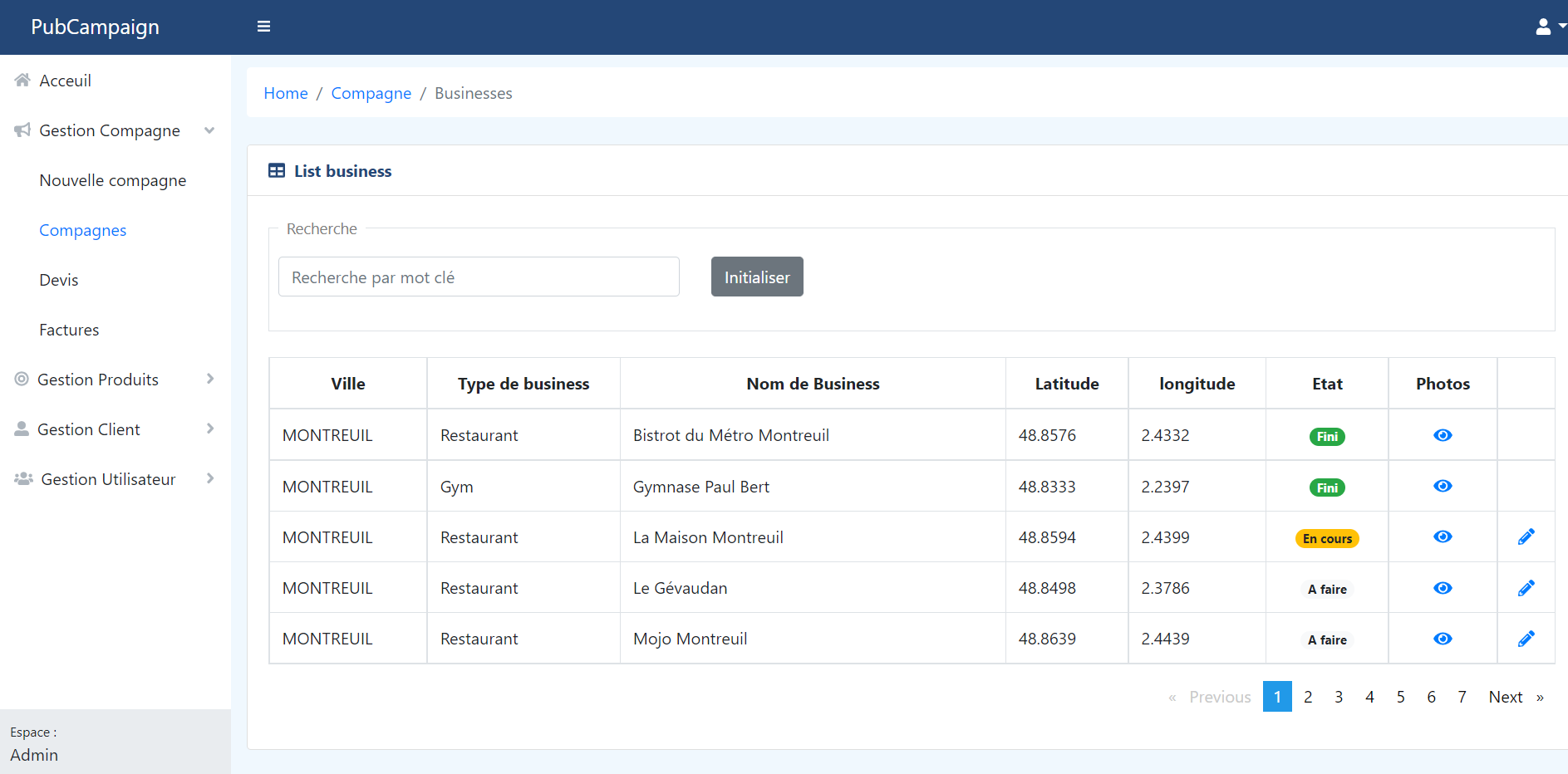


Figure  : Liste de tous les business de la campagne

* **Fonctionnalité édit campagne business**

Lorsque l’administrateur clique sur modifier business représente par une icône d’édition un pop-up de modification business s’affiche pour lui permettre de modifier le business.

Une image contenant table

Description générée automatiquement

Figure  : Pop-up de modification business

Conclusion générale

Mon stage effectué au sein de **AheadIT** m’a donné non seulement l’opportunité de mettre en pratique mes acquis théoriques mais aussi il a contribué à l’enrichissement de mes acquis et de développer mes compétences techniques.

Le travail a consisté, dans un premier temps, à l’étude du contexte du projet et l’analyse des besoins fonctionnels et non fonctionnels que le système devrait assurer l’exécution. Dans un second temps, nous avons conçu et mis en place une architecture clean qui doit nous permettre d’avoir un logiciel de qualité fiable, maintenable, facile à utiliser… pour cela nous avons utilisé plusieurs technologies. Pour l’analyse, la conception et le développement des différents modules du projet nous avons adopté le formalisme UML, la démarche 2TUP, et les frameworks Angular pour la partie front-end et dotnet core pour le back-end. Concernant la base de données nous avons utilisé SQL Server comme système de gestion de base de données relationnelle.

Perspective

Aujourd’hui, pour indiquer qu’un business d’une campagne est réalisé, l’administrateur reçoit un mail d’un agent de publicité contenant la liste des images qui montre la réalisation de ce business par exemple une image d’un panneau de publicité installé devant une boulangerie. Ensuite, l’administrateur cherche et modifie l’état et affecte les photos de réalisation de ce business.

Pour réduire ce processus, On souhaite qu’on développe une application mobile avec un espace agent de publicité qui lui permet de lui donner accès de modification de business campagne et d’affecter les photos de réalisation depuis son téléphone mobile.