

Table des matières

Introduction générale	1
Chapitre 1. Présentation du cadre de projet	2
1.1. Introduction.....	2
1.2. Organisme d'accueil.....	2
1.3. Présentation du sujet.....	3
1.4. Solution existante	4
1.5. Planification du projet	4
1.6. Conclusion	5
Chapitre 2. Spécification et étude conceptuelle	6
2.1. Introduction.....	6
2.2. Analyse et spécification des besoins	6
2.2.1. Identifications des acteurs	6
2.2.2. Spécification des besoins fonctionnels.....	7
2.2.3. Spécification des besoins non fonctionnels	7
2.3. Conception générale	8
2.3.1. Modèle MVC	8
2.3.2. Modèle client – serveur	9
2.4. Diagrammes des cas d'utilisation.....	10
2.4.1. Analyse du cas d'utilisation « S'authentifier »	10
2.4.2. Analyse du cas d'utilisation « Enregistrer des nouveaux utilisateurs »	13
2.4.3. Analyse du cas d'utilisation « Consulter le trafic Skype »	15
2.4.4. Analyse du cas d'utilisation « Consulter le trafic non VOIP »	17

2.4.5.	Analyse du cas d'utilisation « Consulter le trafic voix ».....	19
2.4.6.	Analyse du cas d'utilisation « Consulter le trafic non VOIP ».....	21
2.4.7.	Analyse du cas d'utilisation « Consulter le trafic international entrant vers fixe TT »	22
2.4.8.	Analyse du cas d'utilisation « Consulter le trafic international entrant vers mobile TT »	24
2.4.9.	Analyse du cas d'utilisation « Consulter le trafic international entrant vers Lycamobile »	26
2.5.	Diagramme de classes	28
2.6.	Diagramme d'activité	29
2.7.	Conclusion	31
Chapitre 3.	Réalisation	32
3.1.	Introduction.....	32
3.2.	Environnement de travail.....	32
3.2.1.	Etude et choix technologiques	32
3.2.2.	Architecture de l'application	33
3.2.2.1.	Développement en front end.....	34
3.2.2.2.	Développement en back end.....	34
3.2.2.3.	RESTful API.....	34
3.3.	Implémentation	35
3.3.1.	Environnement de développement.....	35
3.3.1.1.	Node.js.....	35
3.3.1.2.	Express	36
3.3.1.3.	MongoDB	37
3.3.1.4.	AngularJS	38
3.3.2.	Structure d'application.....	38
3.3.3.	Package.json	40

Table des matières

3.4. Interfaces de l'application.....	41
3.5. Conclusion	46
Conclusion générale	47

Liste des figures

Figure 1.1 : Planning de déroulement du projet	4
Figure 2.1: Diagramme des cas d'utilisation global	10
Figure 2.2 : Diagramme de cas d'utilisation « S'authentifier»	11
Figure 2.3 : Diagramme de séquence du cas d'utilisation « S'authentifier »	13
Figure 2.4: Diagramme de cas d'utilisation «Enregistrer des nouveaux utilisateurs»	14
Figure 2.5 : Diagramme de cas d'utilisation «Consulter le trafic Skype»	16
Figure 2.6 : Diagramme de séquence de « Consulter le trafic Skype »	17
Figure 2.7 : Diagramme de cas d'utilisation «Consulter le trafic Viber»	18
Figure 2.8: Diagramme de cas d'utilisation «Consulter le trafic voix»	20
Figure 2.9: Diagramme de cas d'utilisation «Consulter le trafic non VOIP»	21
Figure 2.10: Diagramme de cas d'utilisation «Consulter le trafic international entrant vers fixe TT»	23
Figure 2.11: Diagramme de cas d'utilisation «Consulter le trafic international entrant vers mobile TT»	25
Figure 2.12: Diagramme de cas d'utilisation « Consulter le trafic international entrant vers Lycamobile »	27
Figure 2.13 : Diagramme de classes	29
Figure 2.14 : Diagramme d'activité	30
Figure 3.1 : Architecture de mon application ^[1]	33
Figure 3.2 : Vérification de la version de Node.js	36
Figure 3.3: Structure du dossier d'application	39
Figure 3.4 : Fichier « package.json »	40
Figure 3.5 : Interface d'authentification	41
Figure 3.6: Page d'accueil de l'application	42

Figure 3.7 : Page d'enregistrement.....	42
Figure 3.8 : Les options data.....	43
Figure 3.9 : Les options du trafic international entrant	43
Figure 3.10: Page montrant le trafic international entrant vers fixe TT	44
Figure 3.11: Interface montrant une courbe du trafic international entrant vers fixe TT sur un mois	44
Figure 3.12: Affichage d'un message d'erreur.....	45
Figure 3.13 : page permettant de consulter le trafic Skype	45
Figure 3.14 : Interface montrant une courbe du trafic Skype sur une semaine	46

Liste des tableaux

Tableau 2.1 : : Description textuelle du cas d'utilisation « S'authentifier »	12
Tableau 2.2: Description textuelle du cas d'utilisation « Enregistrer des nouveaux utilisateurs»	15
Tableau 2.3: Description textuelle du cas d'utilisation « Consulter le trafic Skype»	16
Tableau 2.4: Description textuelle du cas d'utilisation « Consulter le trafic Viber»	19
Tableau 2.5: Description textuelle du cas d'utilisation « Consulter le trafic voix»	20
Tableau 2.6: Description textuelle du cas d'utilisation « Consulter le trafic non VOIP»	22
Tableau 2.7: Description textuelle du cas d'utilisation « Consulter le trafic international entrant vers fixe TT»	24
Tableau 2.8: Description textuelle du cas d'utilisation « Consulter le trafic international entrant vers mobile TT »	26
Tableau 2.9: Description textuelle du cas d'utilisation « Consulter le trafic international entrant vers Lycamobile»	28

Liste des abréviations

A

API : Application Program Interface

B

BSON : Binary Structured Object Notation

C

CLI : Character User Interface

CNOC : Customer Network Operation Center

CSS : Cascading Style Sheets

D

DERAF : Direction Exécutive Revenue Assurance et Fraude

H

HTML : Hypertext Markup Language

J

JSON : JavaScript Object Notation

K

KPI : Key Performance Indicator

L

LAMP : "Linux, Apache, MySQL, and PHP"

M

MEAN: "MongoDB, ExpressJS, AngularJS and Node.js"

O

ORM: Object Relational Mapping

P

PHP: Hypertext Preprocessor

R

REST: Representational State Transfer

T

TT: Tunisie Télécom

Introduction générale

Pour vérifier les progrès accomplis dans la réalisation de leurs objectifs, toutes les entreprises utilisent différents indicateurs et outils de mesures.

Les indicateurs clés de performance sont des outils de pilotage particulièrement importants pour les dirigeants d'entreprises. Ils permettent de bien comprendre la performance de l'entreprise, de prendre des décisions dans les meilleures conditions et de positionner un projet sur le chemin du succès.

En effet, les tableaux de bord constituent un élément central permettant aux utilisateurs d'accéder à, d'interagir et d'analyser des informations actualisées afin de prendre des décisions plus intelligentes et basées sur des données. Ils permettent ainsi de surveiller et de mesurer les performances et les mesures en temps réel et en déplacement.

De plus, visualiser et analyser les données et concentrer sur les indicateurs de performance clés de l'ensemble de l'entreprise sur un tableau de bord, assure aux utilisateurs l'obtention des informations précieuses et la prise des décisions rapides et précises.

C'est dans ce contexte que s'intègre notre projet, qui a pour objectif la mise en place d'une solution assurant le suivi d'un ensemble des indicateurs clés de performance dont le but d'aider les agents de la Direction Exécutive Revenu Assurance et Fraude à bien évaluer les situations et à prendre une décision adaptée.

Ce rapport s'articulera donc, autour de trois chapitres. Le premier chapitre comportera une description de l'organisme d'accueil et mettra l'accent sur les catégories de mes clients. Il spécifiera également les besoins fonctionnels et les besoins non fonctionnels.

Dans le deuxième chapitre, nous présenterons la conception en détaillant quelques diagrammes utiles pour bien modéliser les besoins fonctionnels de mon application.

Le troisième chapitre portera sur la réalisation de notre solution, il détaillera nos choix technologiques et l'architecture de notre système et il présentera quelques interfaces de l'application.

Chapitre 1. Présentation du cadre de projet

1.1. Introduction

L'étude préalable constitue une étape préliminaire pour la réalisation d'une application. En effet, elle permet de mettre notre travail dans son contexte général.

Tout d'abord, nous voulons commencer par une présentation de l'entreprise d'accueil. Puis, nous décrivons le sujet autour duquel s'articule notre projet. Et enfin, nous montrerons le planning de déroulement du projet.

1.2. Organisme d'accueil

Le " **Customer Network Operations Center** " ou " **CNOC** " est l'un des services techniques de Tunisie Télécom. Il assure la maintenance, la surveillance, le contrôle et la supervision des réseaux des clients Business (clients Entreprises ou Grands Comptes).

Il a pour missions :

- Centre de Relation Client (CRC) dans le but d'anticiper l'insatisfaction du client.
- La maintenance corrective N1 et N2, côté client (support N1 et N2).
- La maintenance préventive (sauvegarde périodique des configurations Clients, test périodique des liaisons backup, etc.)
- La supervision des indicateurs SLAs.
- Développement des outils de supervision aux autres directions de Tunisie Télécom.
- La modification de la configuration des clients existants.
- Le soutien aux chefs de projets lors des activités de déploiement ou de mise à niveau.
- La gestion des outils et systèmes relevant du CNOC (OSS, postes de travail, videowall, réseau LAN/WAN, etc.).

- Le « Reporting » (Rapport d'activité, Rapports d'incident, Escalade managériale, etc.).

1.3. Présentation du sujet

Un KPI est un acronyme pour « Key Performance Indicator ». C'est un indicateur de performance clé qui doit permettre de mesurer le progrès réalisé ou à réaliser pour atteindre un objectif stratégique.

Un indicateur doit être associé à un objectif précis et mesurable. La mesure de l'évolution d'un KPI devrait entraîner une décision. Donc ces indicateurs doivent être matérialisés sous différentes formes suivant leurs usages : boussole, barre de progression, histogramme, cartes, courbes ...

D'où le besoin de créer une application permettant de suivre les KPIs de l'entreprise. Elle ressemble en fait à un portail présentant une vue globale et en temps réel de l'état de santé de l'entreprise, en affichant des informations statistiques essentielles.

C'est dans ce contexte, que nous étions appelés à concevoir et à développer une application web aux agents de la DERAf. Cela est dans le but de simplifier la tâche de suivi des indicateurs de performance clé.

En effet, la Direction Centrale des Finances est l'une des directions opérationnelles de Tunisie Télécom dont la principale mission est de gérer la politique financière de la société en vue d'assurer l'équilibre financier et de présenter des états financiers fiables et crédibles conformément à la réglementation en vigueur.

L'une de ses divisions est la DERAf, Direction Exécutive Revenu Assurance et Fraude. Elle a pour principale mission la détection de :

- Fuites entre les système TT : la DRAF assure les contrôles quotidiens et en temps quasi réel des flux des systèmes TT afin de prévenir et détecter toute fuite pouvant engendrer une perte de revenue à l'entreprise.
- Fraudes : Elle assure la prévention et la détection de tout type de fraude que ce soit interne ou externe via des contrôles et des systèmes en temps quasi réel.

Ainsi la Direction Exécutive Revenue Assurance et Fraude assure :

- La fourniture des rapports et des statistiques sur demande au TOP management de TT ainsi qu'aux différentes structures TT ayant besoin d'une information précise vu que la DERAf contrôle la majorité des flux TT.
- Le fait de rechercher et d'être à jour dans l'évolution des techniques des fraudes à l'échelle international et de l'appliquer à l'échelle national.

1.4. Solution existante

Pour évaluer les données actuelles, la DERAf utilise l'Excel pour créer des rapports permettant de suivre les différents indicateurs.

Donc actuellement, la tâche de suivi des indicateurs est réalisée manuellement.

1.5. Planification du projet

La planification est parmi les phases d'avant-projet les plus importantes. Elle consiste à déterminer et à ordonnancer les tâches du projet et à estimer leurs charges respectives.

Parmi les outils de planification des projets, nous avons utilisé le diagramme de GANTT. C'est un outil qui permet de planifier le projet et de rendre plus simple le suivi de son avancement. Ce diagramme permet aussi de visualiser l'enchaînement et la durée des différentes tâches durant le stage comme illustré par la figure ci-dessous.

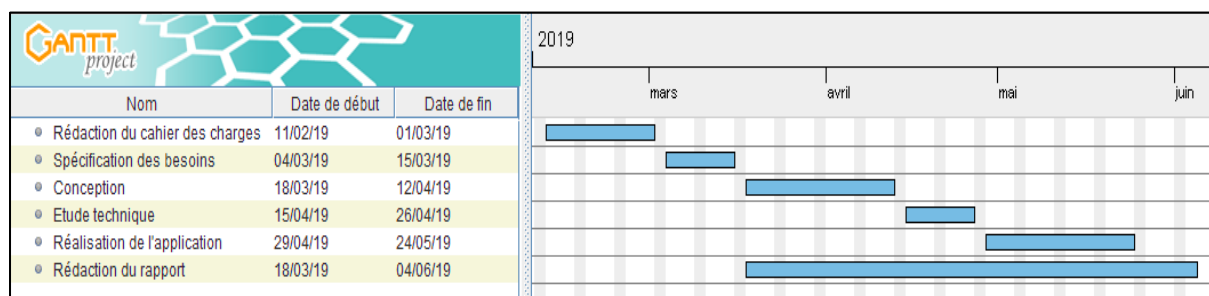


Figure 1.1 : Planning de déroulement du projet

1.6. Conclusion

Tout au long de ce chapitre, nous avons pu situer le cadre général de notre projet, à savoir la société d'accueil et la présentation de sujet.

Dans ce qui suit nous voulons entamer la phase de la conception de notre application.

Chapitre 2. Spécification et étude conceptuelle

2.1. Introduction

Penser avant d'agir, faire des plans avant de construire, concevoir d'abord et ensuite développer est la démarche qui doit être suivie lors du développement d'une application.

En effet, la conception d'un système informatique est une étape très importante qui va influencer la qualité et la fiabilité de toute application.

Dans le cadre de ce chapitre, nous nous intéressons alors à la spécification des besoins des différents acteurs dont le but d'identifier les différentes fonctionnalités de l'application.

Aussi nous détaillerons la conception de notre application en présentant le diagramme de cas d'utilisations, en détaillant ces différents cas et en montrant le diagramme de classes.

2.2. Analyse et spécification des besoins

Une exigence peut être définie comme la description de la fonction qu'un système doit fournir. Donc la spécification des besoins et des exigences de l'utilisateur du système est l'une des phases les plus importantes dans le cycle de vie d'un logiciel.

Pour cela, il est nécessaire de faire une description détaillée et claire des attentes des utilisateurs et des fonctionnalités que va fournir le système pour y répondre.

2.2.1. Identifications des acteurs

Dans le cadre de notre analyse, les acteurs que nous avons pu identifier sont :

- **L'agent de la DERA** : C'est celui qui utilise l'application pour bénéficier de tous les services proposés privé de la possibilité d'ajouter des nouveaux utilisateurs.
- **L'administrateur** : qui est un utilisateur possédant tous les droits et les privilèges.

2.2.2. Spécification des besoins fonctionnels

Au cours de cette section, nous exposons les différents besoins fonctionnels auxquels devraient répondre notre application.

Les besoins fonctionnels et les attentes vis-à-vis de notre application dépendent d'un acteur à un autre. Pour cela, nous avons décrit pour chaque acteur les besoins fonctionnels qui lui sont associés.

Le système doit permettre :

- L'authentification des utilisateurs par un login et un mot de passe pour accéder aux différentes fonctionnalités.
- Le contrôle d'accès des utilisateurs aux différentes fonctionnalités, en tenant compte des privilèges.

Le système doit permettre à l'administrateur et à l'agent de la DERAf de :

- Consulter le trafic Skype des utilisateurs.
- Consulter le trafic Viber des utilisateurs.
- Consulter le trafic voix des utilisateurs.
- Consulter le trafic non VOIP des utilisateurs.
- Consulter le trafic international entrant vers fixe TT.
- Consulter le trafic international entrant vers mobile TT.
- Consulter le trafic international entrant vers Lycamobile.

Le système doit permettre à l'administrateur de :

- Enregistrer des nouveaux utilisateurs.

2.2.3. Spécification des besoins non fonctionnels

Ce sont les besoins techniques décrivant toutes les contraintes auxquelles est soumise l'application pour sa réalisation et son bon fonctionnement.

La nature du notre projet exige certaines règles à respecter qui se résument dans les points suivants :

- **L'ergonomie des interfaces** :

Les interfaces de notre application doivent être ergonomique et conviviale. Aussi, elles doivent être accessible par tous les utilisateurs, quelles que soient leurs caractéristiques et leurs moyens d'accès à l'information.

- **La fiabilité** :

Les résultats apportés par l'application doivent être fiables et reflètent effectivement l'état de la base au moment de son interrogation, c'est-à-dire lors de la mise à jour des données.

- **La disponibilité** :

Notre application doit être disponible à tout instant pour être utilisée par n'importe quel utilisateur, et doit être facilement accessible via n'importe quel ordinateur.

- **La sécurité** :

Notre application comporte des informations personnelles et sensibles, donc elle doit respecter les règles relatives à la sécurité des systèmes informatiques.

2.3. Conception générale

2.3.1. Modèle MVC

Modèle – Vue – Contrôleur (MVC) est un modèle architectural couramment utilisé pour développer des interfaces utilisateur qui divise une application en trois parties interconnectées. Ceci est fait pour séparer les représentations internes d'informations des manières dont les informations sont présentées à l'utilisateur et acceptées par l'utilisateur.

Utilisée traditionnellement pour les interfaces utilisateur graphiques de bureau, cette architecture est devenue populaire pour la conception d'applications Web.

Les trois parties constituant le modèle MVC sont:

- **Modèle**

Le composant central du motif. C'est la structure de données dynamique de l'application, indépendante de l'interface utilisateur. Il gère directement les données, la logique et les règles de l'application.

- **Vue**

Toute représentation d'informations telles qu'un graphique, un diagramme ou un tableau. Plusieurs vues de la même information sont possibles, telles qu'un graphique à barres pour la gestion et une vue tabulaire pour les comptes.

- **Contrôleur**

Accepte l'entrée et la convertit en commandes pour le modèle ou la vue.

2.3.2. Modèle client – serveur

Le modèle client – serveur est une structure d'application distribuée qui partitionne les tâches ou les charges de travail entre les fournisseurs d'une ressource ou d'un service, appelés serveurs, et les demandeurs de service, appelés clients. Souvent, les clients et les serveurs communiquent via un réseau informatique sur un matériel distinct, mais le client et le serveur peuvent résider dans le même système. Un hôte serveur exécute un ou plusieurs programmes serveur qui partagent leurs ressources avec des clients. Un client ne partage aucune de ses ressources, mais demande le contenu ou la fonction de service d'un serveur. Les clients lancent donc des sessions de communication avec les serveurs qui attendent les demandes entrantes.

Donc notre application utilise le modèle client – serveur.

2.4. Diagrammes des cas d'utilisation

Le diagramme de cas d'utilisation décrit l'interaction entre le système et les acteurs en spécifiant les besoins des utilisateurs et tout ce que doit faire le système pour ces utilisateurs. Ci-dessous le diagramme de cas d'utilisation de notre application :

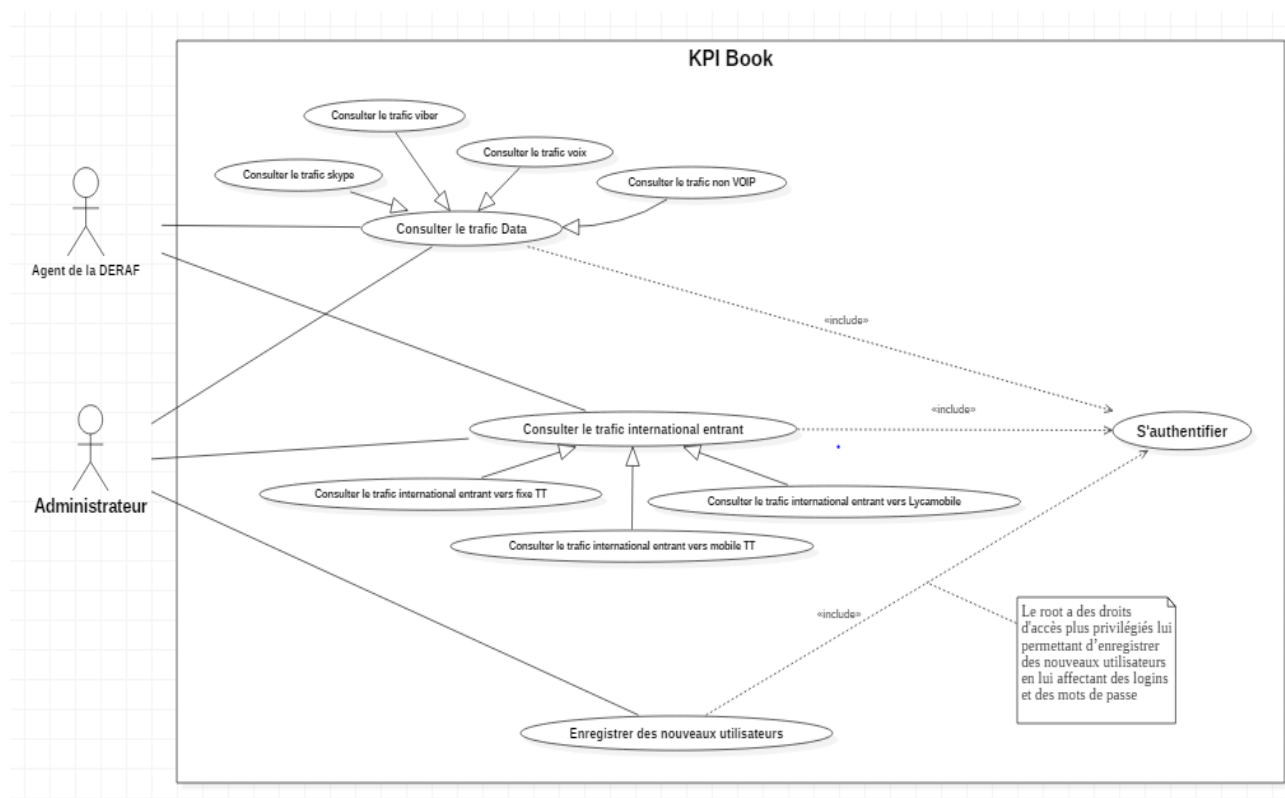


Figure 2.1: Diagramme des cas d'utilisation global

Pour mieux comprendre les cas d'utilisation exprimés dans le diagramme des cas d'utilisation global, nous voulons les détailler dans ce qui suit.

2.4.1. Analyse du cas d'utilisation « S'authentifier »

L'authentification est le processus de reconnaissance de l'identité d'un utilisateur. Elle s'exécute toujours au début de l'application, avant que les contrôles d'autorisation et de limitation ne soient effectués.

Différents systèmes peuvent nécessiter différents types d'informations d'identification pour déterminer l'identité d'un utilisateur. Les utilisateurs de mon application sont identifiés par un login et un mot de passe.

La figure ci-dessous explique mieux le cas d'utilisation « S'authentifier ».

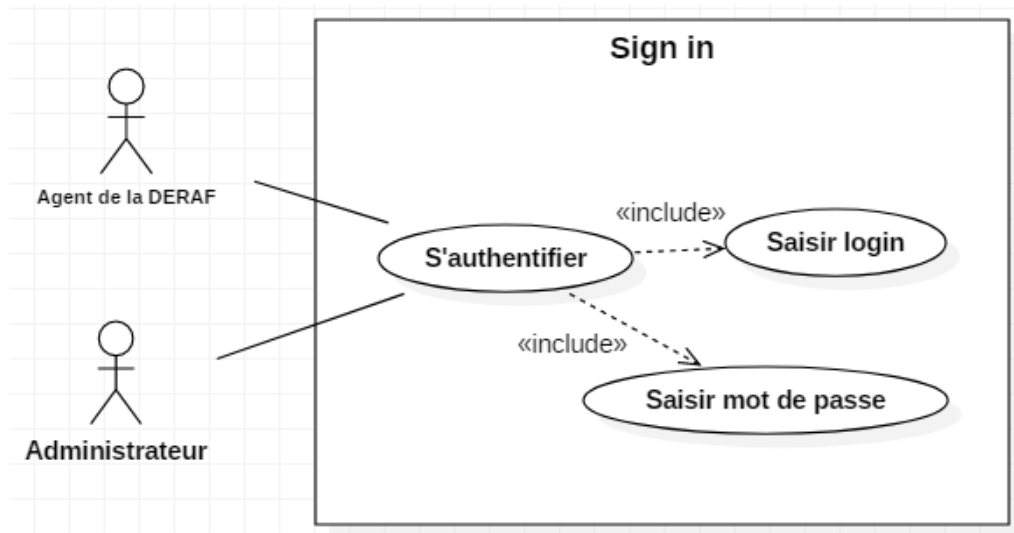


Figure 2.2 : Diagramme de cas d'utilisation « S'authentifier »

Le tableau 3.1 décrit textuellement le cas d'utilisation « S'authentifier ».



Tableau 2.1 : : Description textuelle du cas d'utilisation « S'authentifier »

Titre	« S'authentifier »
But	Identification et autorisation d'accès.
Résumé	L'utilisateur introduit son login et son mot de passe pour avoir accès au système.
Acteurs	Agent de la DERAf / Administrateur
Préconditions	
<ul style="list-style-type: none"> - L'application doit être installée sur l'ordinateur de l'agent de la DERAf / administrateur. - L'agent de la DERAf / administrateur doit être connecté au réseau de l'entreprise. 	
Post-conditions	
<ul style="list-style-type: none"> - Accès aux diverses fonctionnalités de l'application. 	
Scénario Nominal	
<ol style="list-style-type: none"> 1. L'agent de la DERAf / administrateur lance l'application. 2. L'application affiche le formulaire d'authentification (champs login et mot de passe). 3. L'agent de la DERAf / administrateur saisit son login et son mot de passe. 4. Le système contrôle les champs et les vérifie. 5. Si l'agent de la DERAf / administrateur est identifié, le système affiche l'interface d'accueil. 	
Enchaînement alternative	
<p>E1 : L'agent de la DERAf / administrateur introduit des paramètres d'authentifications Incorrectes.</p> <ol style="list-style-type: none"> 1. L'utilisateur est non authentifié et l'accès au système est refusé. 	

Et pour mieux décrire le cas d'utilisation « S'authentifier », voici ci-dessous son diagramme de séquence.

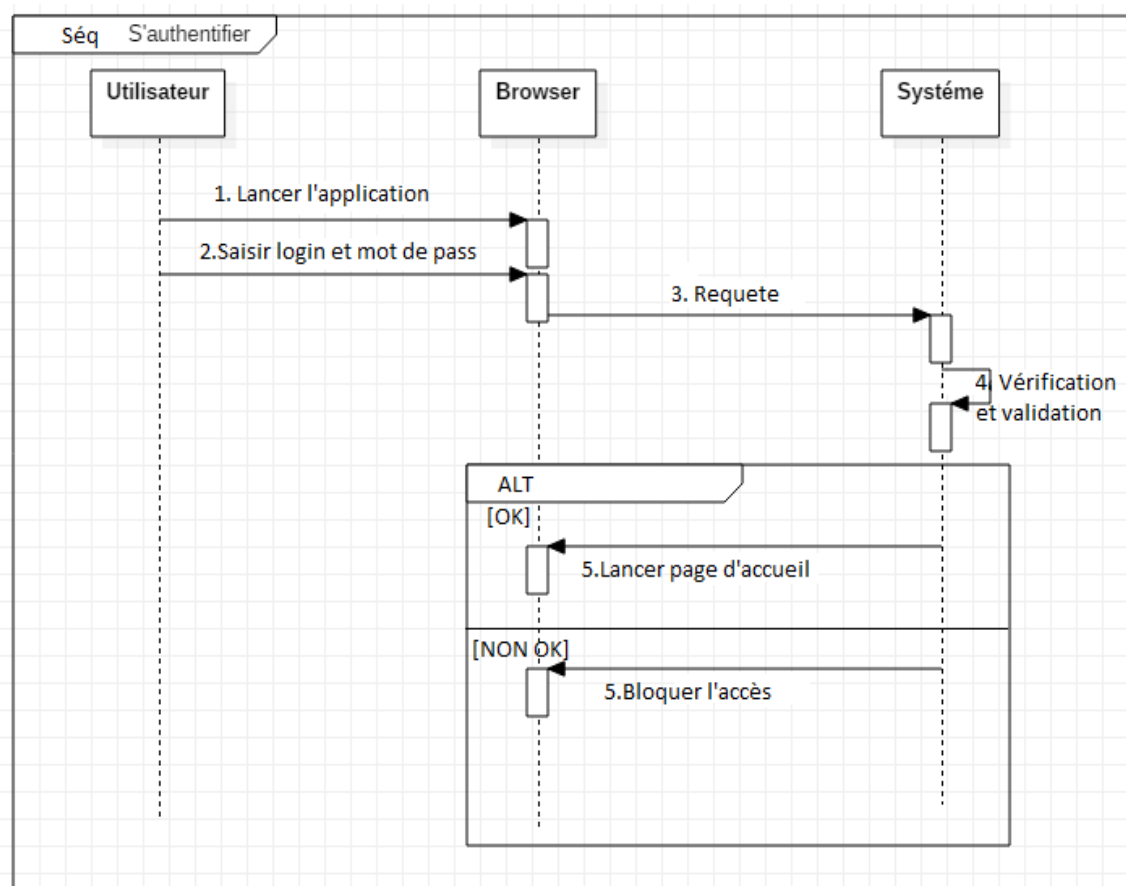


Figure 2.3 : Diagramme de séquence du cas d'utilisation « S'authentifier »

2.4.2. Analyse du cas d'utilisation « Enregistrer des nouveaux utilisateurs »

Pour avoir un accès à l'application, un agent de la DERAf doit être inscrit par l'administrateur.

La figure ci-dessous montre mieux le cas d'utilisation « Enregistrer des nouveaux utilisateurs ».

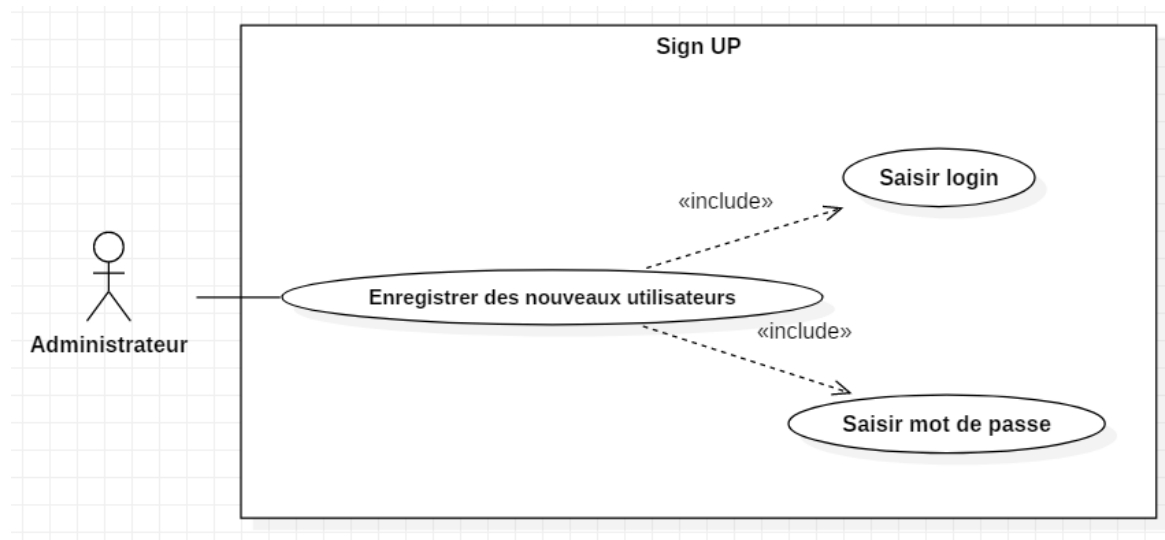


Figure 2.4: Diagramme de cas d'utilisation «Enregistrer des nouveaux utilisateurs»

Le tableau 2.2 décrit textuellement le cas d'utilisation « Enregistrer des nouveaux utilisateurs ».

Tableau 2.2: Description textuelle du cas d'utilisation « Enregistrer des nouveaux utilisateurs »

Titre	« Enregistrer des nouveaux utilisateurs »
But	Autoriser les nouveaux utilisateurs à accéder au système.
Résumé	L'administrateur associe un login et un mot de passe à chaque nouvel utilisateur pour avoir accès au système.
Acteurs	Administrateur
Préconditions	
- L'administrateur doit accéder à l'application.	
Post-conditions	
- Nouvel utilisateur inscrit.	
Scénario Nominal	
<ol style="list-style-type: none"> 1. L'administrateur consulte la barre de menu pour sélectionner « Register » 2. Le système affiche l'interface d'enregistrement des nouveaux utilisateurs. 3. L'administrateur saisit le login et le mot de passe à associer au nouvel utilisateur. 4. L'administrateur valide les données saisis en cliquant sur « Register » 5. Le système sauvegarde les nouvelles informations fournies. 	
Enchaînement alternative	
<p>E1 : Champs obligatoires non valides et/ou vides</p> <ol style="list-style-type: none"> 1. Le scénario reprend de 2. <p>E2 : Login existe dans la base de données</p> <ol style="list-style-type: none"> 1. Le scénario reprend de 2. 	

2.4.3. Analyse du cas d'utilisation « Consulter le trafic Skype »

Un utilisateur autorisé peut visualiser des courbes lui présentant le trafic Skype des clients de l'entreprise.

La figure ci-dessous montre mieux le cas d'utilisation « Consulter le trafic Skype ».

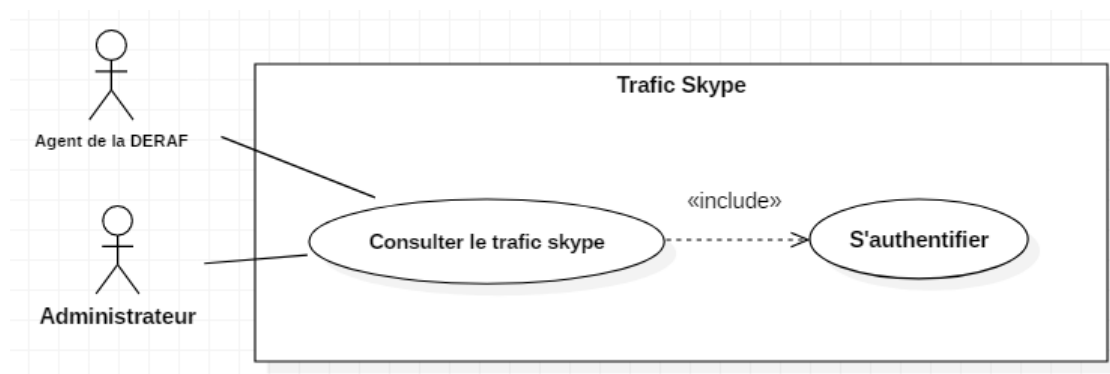


Figure 2.5 : Diagramme de cas d'utilisation « Consulter le trafic Skype »

Le tableau 2.3 décrit textuellement le cas d'utilisation « Consulter le trafic Skype ».

Tableau 2.3: Description textuelle du cas d'utilisation « Consulter le trafic Skype »

Titre	« Consulter le trafic Skype »
But	Visualiser des courbes présentant le trafic Skype des clients de l'entreprise dans une période donnée.
Résumé	En fonction du période précisé par l'agent de la DERAf / administrateur, des courbes montrant le trafic Skype des clients seront affichées.
Acteurs	Agent de la DERAf / Administrateur
Préconditions	
- Agent de la DERAf / administrateur authentifié.	
Post-conditions	
- Des données concernant le trafic Skype des clients de l'entreprise seront visualisées.	
Scénario Nominal	
<ol style="list-style-type: none"> 1. L'agent de la DERAf / administrateur consulte la barre de menu pour sélectionner « Data → Trafic Skype » 2. Le système affiche l'interface où les données seront visualisées. 3. L'administrateur choisit la période dont il veut visualiser les données associées. 4. Le système affiche les courbes montrant le trafic Skype des clients sur la période précisée par l'agent de la DERAf / administrateur. 	
Enchaînement alternative	
<p>E1 : Période non précisée</p> <ol style="list-style-type: none"> 1. Le scénario reste bloqué à 2. 	

Et pour mieux expliquer le cas d'utilisation « Consulter le trafic Skype », voici ci-dessous son diagramme de séquence.

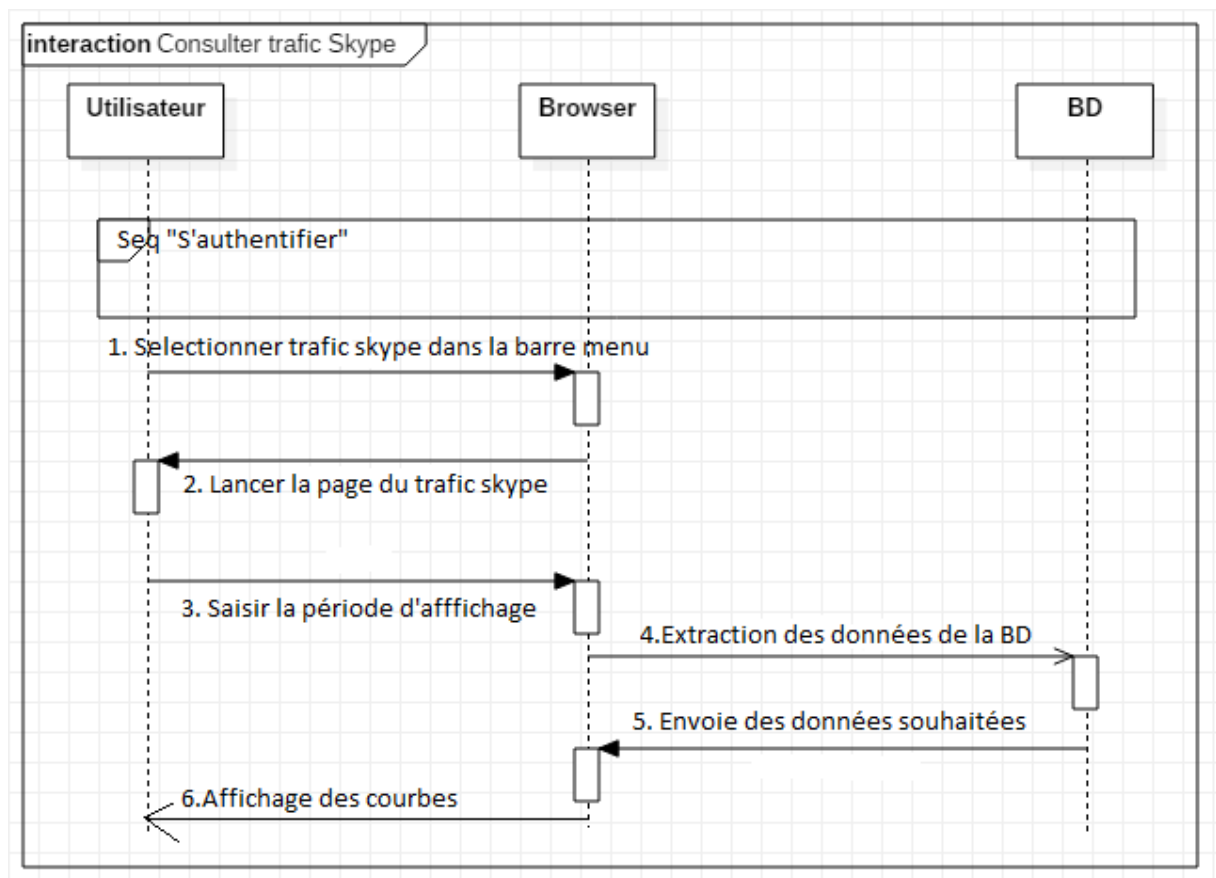


Figure 2.6 : Diagramme de séquence de « Consulter le trafic Skype »

2.4.4. Analyse du cas d'utilisation « Consulter le trafic non VOIP »

Un utilisateur autorisé peut visualiser des courbes lui présentant le trafic Viber des clients de l'entreprise.

La figure ci-dessous montre mieux le cas d'utilisation « Consulter le trafic Viber ».

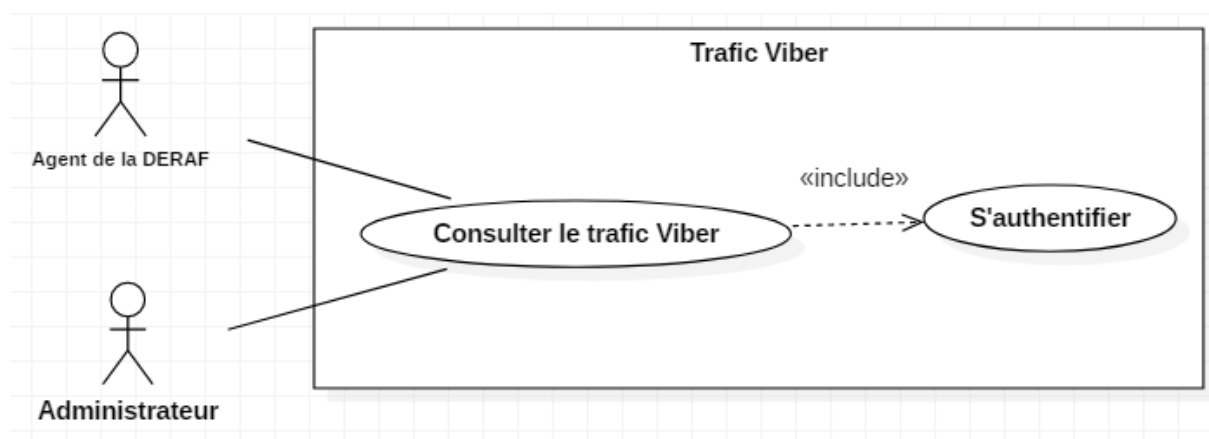


Figure 2.7 : Diagramme de cas d'utilisation «Consulter le trafic Viber»

Le tableau 2.4 décrit textuellement le cas d'utilisation « Consulter le trafic Viber ».

Tableau 2.4: Description textuelle du cas d'utilisation « Consulter le trafic Viber »

Titre	« Consulter le trafic Viber »
But	Visualiser des courbes présentant le trafic Viber des clients de l'entreprise dans une période donnée.
Résumé	En fonction du période précisé par l'agent de la DERAf / administrateur, des courbes montrant le trafic Viber des clients seront affichées.
Acteurs	Agent de la DERAf / Administrateur
Préconditions	
- Agent de la DERAf / administrateur authentifié.	
Post-conditions	
- Des données concernant le trafic Viber des clients de l'entreprise seront visualisées.	
Scénario Nominal	
<ol style="list-style-type: none"> 1. L'agent de la DERAf / administrateur consulte la barre de menu pour sélectionner « Data → Trafic Viber » 2. Le système affiche l'interface où les données seront visualisées. 3. L'administrateur choisit la période dont il veut visualiser les données associées. 4. Le système affiche les courbes montrant le trafic Viber des clients sur la période précisée par l'agent de la DERAf / administrateur. 	
Enchaînement alternative	
<p>E1 : Période non précisée</p> <ol style="list-style-type: none"> 1. Le scénario reste bloqué à 2. 	

2.4.5. Analyse du cas d'utilisation « Consulter le trafic voix »

Un utilisateur autorisé peut visualiser des courbes lui présentant le trafic voix d'un client donné.

La figure ci-dessous montre mieux le cas d'utilisation « Consulter le trafic voix ».

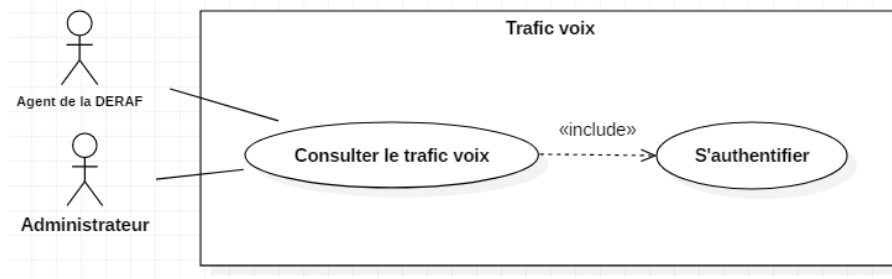


Figure 2.8: Diagramme de cas d'utilisation « Consulter le trafic voix »

Le tableau 2.5 décrit textuellement le cas d'utilisation « Consulter le trafic voix ».

Tableau 2.5: Description textuelle du cas d'utilisation « Consulter le trafic voix »

Titre	« Consulter le trafic voix »
But	Visualiser des courbes présentant le trafic voix d'un client précisé dans une période donnée.
Résumé	En fonction de la période précisée par l'agent de la DERAf /administrateur, des courbes montrant le trafic voix d'un client précisé seront affichées.
Acteurs	Agent de la DERAf / Administrateur
Préconditions	
- Agent de la DERAf / administrateur authentifié.	
Post-conditions	
- Des données concernant le trafic voix d'un client précisé seront visualisées.	
Scénario Nominal	
<ol style="list-style-type: none"> 1. L'agent de la DERAf / administrateur consulte la barre de menu pour sélectionner « Data → Trafic voix » 2. Le système affiche l'interface où les données seront visualisées. 3. L'administrateur saisit le numéro d'abonné dont il veut visualiser ses données. 4. Le système affiche les courbes montrant le trafic voix du client mentionné sur la période précisée par l'agent de la DERAf / administrateur. 	
Enchaînement alternative	
<p>E1 : L'agent de la DERAf / administrateur introduit un numéro inexistant.</p> <ol style="list-style-type: none"> 1. Le scénario reprend de 2. 	

2.4.6. Analyse du cas d'utilisation « Consulter le trafic non VOIP »

Un utilisateur autorisé peut visualiser des courbes lui présentant le trafic non VOIP des clients de l'entreprise.

La figure ci-dessous montre mieux le cas d'utilisation « Consulter le trafic non VOIP ».

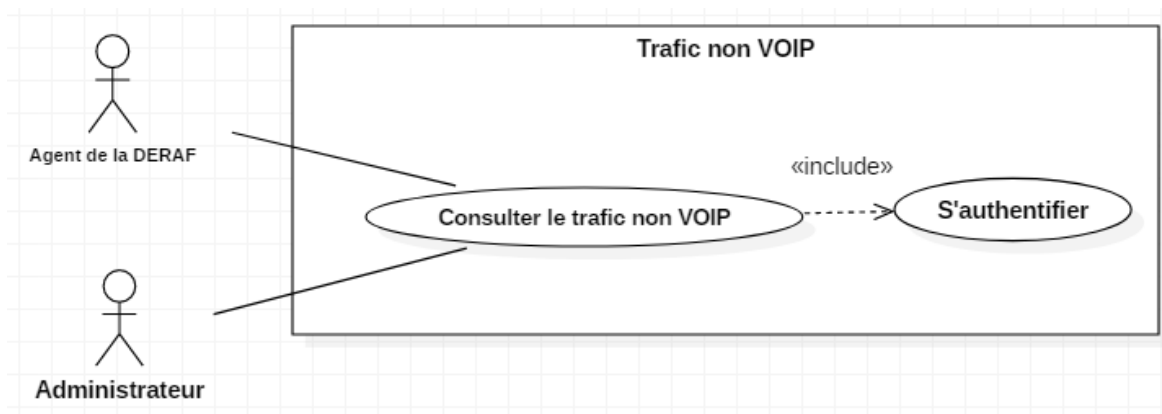


Figure 2.9: Diagramme de cas d'utilisation « Consulter le trafic non VOIP »

Le tableau 2.6 décrit textuellement le cas d'utilisation « Consulter le trafic non VOIP ».



Tableau 2.6: Description textuelle du cas d'utilisation « Consulter le trafic non VOIP »

Titre	« Consulter le trafic non VOIP »
But	Visualiser des courbes présentant le trafic non VOIP des clients de l'entreprise dans une période donnée.
Résumé	En fonction du période précisé par l'agent de la DERAf / administrateur, des courbes montrant le trafic non VOIP des clients seront affichées.
Acteurs	Agent de la DERAf / Administrateur
Préconditions	
- Agent de la DERAf / administrateur authentifié.	
Post-conditions	
- Des données concernant le trafic non VOIP des clients de l'entreprise seront visualisées.	
Scénario Nominal	
<ol style="list-style-type: none"> 1. L'agent de la DERAf / administrateur consulte la barre de menu pour sélectionner « Data → Trafic non VOIP » 2. Le système affiche l'interface où les données seront visualisées. 3. L'administrateur choisit la période dont il veut visualiser les données associées. 4. Le système affiche les courbes montrant le trafic non VOIP des clients sur la période précisée par l'agent de la DERAf / administrateur. 	
Enchaînement alternative	
<p>E1 : Période non précisée</p> <ol style="list-style-type: none"> 1. Le scénario reste bloqué à 2. 	

2.4.7. Analyse du cas d'utilisation « Consulter le trafic international entrant vers fixe TT »

Un utilisateur autorisé peut visualiser des courbes lui présentant le trafic international entrant vers fixe TT.

La figure ci-dessous montre mieux le cas d'utilisation « Consulter le trafic international entrant vers fixe TT ».

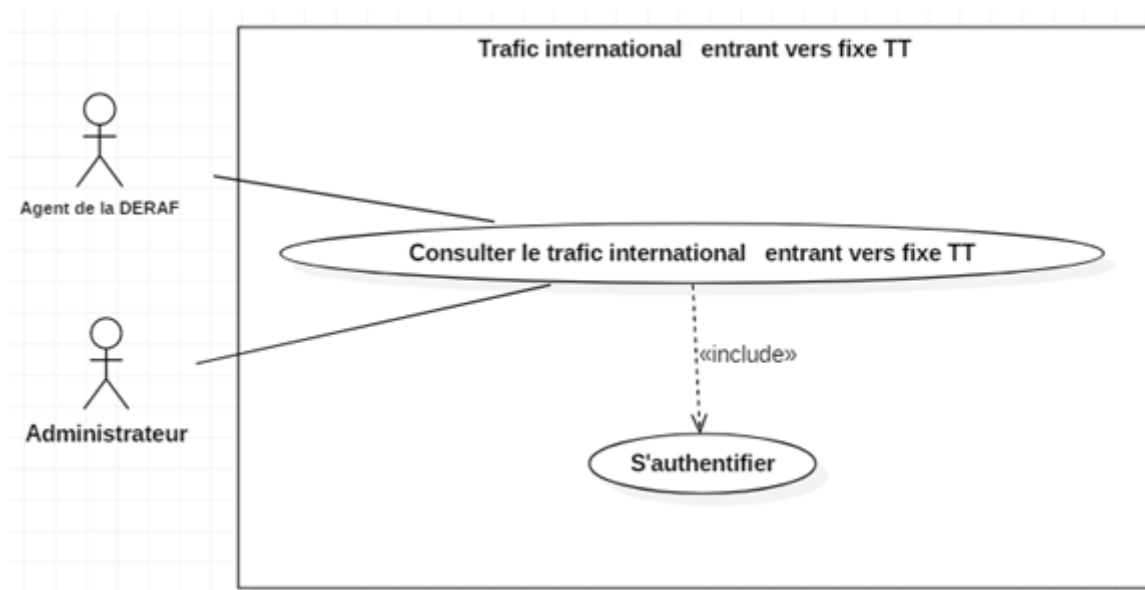


Figure 2.10: Diagramme de cas d'utilisation «Consulter le trafic international entrant vers fixe TT»

Le tableau 2.7 décrit textuellement le cas d'utilisation « Consulter le trafic international entrant vers fixe TT ».

Tableau 2.7: Description textuelle du cas d'utilisation « Consulter le trafic international entrant vers fixe TT »

Titre	« Consulter le trafic international entrant vers fixe TT »
But	Visualiser des courbes présentant le trafic international entrant vers fixe TT dans une période donnée.
Résumé	En fonction du période précisé par l'agent de la DERAf /administrateur, des courbes montrant le trafic international entrant vers fixe TT seront affichées.
Acteurs	Agent de la DERAf / Administrateur
Préconditions	
- Agent de la DERAf / administrateur authentifié.	
Post-conditions	
- Des données concernant le trafic international entrant vers fixe TT seront visualisées.	
Scénario Nominal	
<ol style="list-style-type: none"> 1. L'agent de la DERAf / administrateur consulte la barre de menu pour sélectionner « Data → Trafic international entrant vers fixe TT » 2. Le système affiche l'interface où les données seront visualisées. 3. L'administrateur choisit la période dont il veut visualiser les données associées. 4. Le système affiche les courbes montrant le trafic international entrant vers fixe TT sur la période précisée par l'agent de la DERAf / administrateur. 	
Enchaînement alternative	
<p>E1 : Période non précisée</p> <ol style="list-style-type: none"> 1. Le scénario reste bloqué à 2. 	

2.4.8. Analyse du cas d'utilisation « Consulter le trafic international entrant vers mobile TT »

Un utilisateur autorisé peut visualiser des courbes lui présentant le trafic international entrant vers mobile TT.

La figure ci-dessous montre mieux le cas d'utilisation « Consulter le trafic international entrant vers mobile TT ».

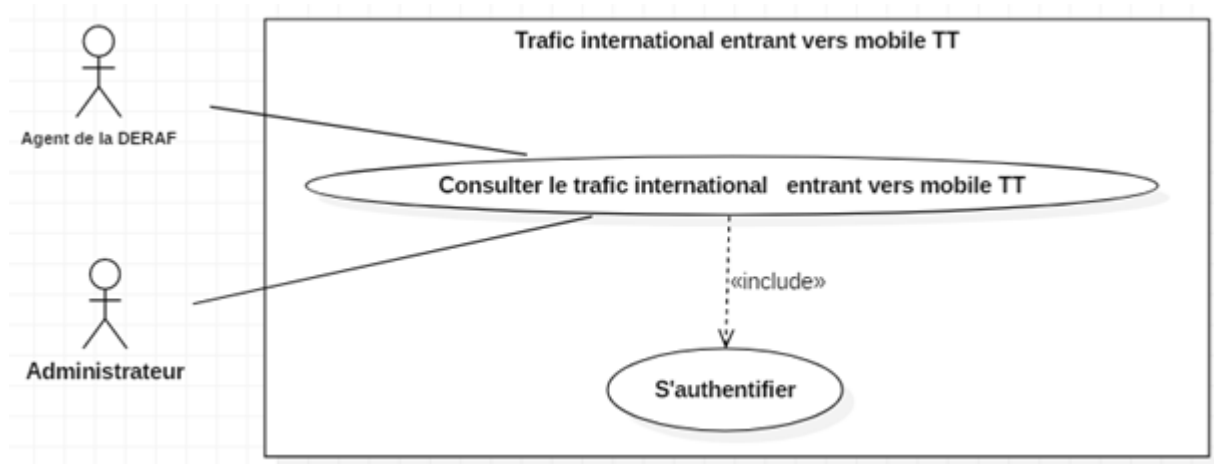


Figure 2.11: Diagramme de cas d'utilisation «Consulter le trafic international entrant vers mobile TT»

Le tableau 2.8 décrit textuellement le cas d'utilisation « Consulter le trafic international entrant vers mobile TT ».

Tableau 2.8: Description textuelle du cas d'utilisation « Consulter le trafic international entrant vers mobile TT »

Titre	« Consulter le trafic international entrant vers mobile TT »
But	Visualiser des courbes présentant le trafic international entrant vers mobile TT dans une période donnée.
Résumé	En fonction du période précisé par l'agent de la DERAf / administrateur, des courbes montrant le trafic international entrant vers mobile TT seront affichées.
Acteurs	Agent de la DERAf / Administrateur
Préconditions	
- Agent de la DERAf / administrateur authentifié.	
Post-conditions	
- Des données concernant le trafic international entrant vers fixe TT seront visualisées.	
Scénario Nominal	
<ol style="list-style-type: none"> 1. L'agent de la DERAf / administrateur consulte la barre de menu pour sélectionner « Data → Trafic international entrant vers mobile TT » 2. Le système affiche l'interface où les données seront visualisées. 3. L'administrateur choisit la période dont il veut visualiser les données associées. 4. Le système affiche les courbes montrant le trafic international entrant vers mobile TT sur la période précisée par l'agent de la DERAf / administrateur. 	
Enchaînement alternative	
<p>E1 : Période non précisée</p> <ol style="list-style-type: none"> 1. Le scénario reste bloqué à 2. 	

2.4.9. Analyse du cas d'utilisation « Consulter le trafic international entrant vers Lycamobile »

Un utilisateur autorisé peut visualiser des courbes lui présentant le trafic international entrant vers Lycamobile.

La figure ci-dessous montre mieux le cas d'utilisation « Consulter le trafic international entrant vers Lycamobile ».

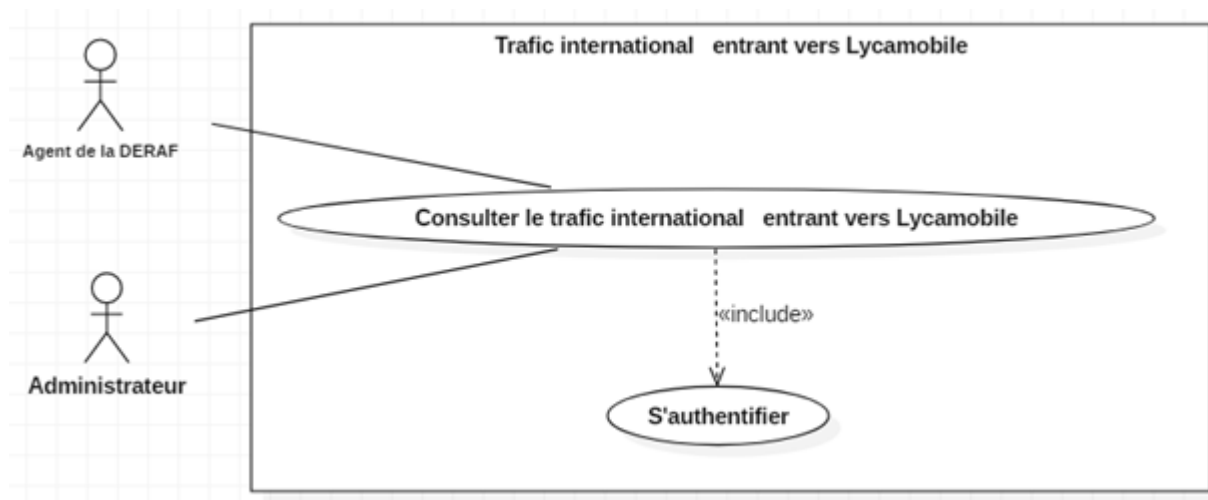


Figure 2.12: Diagramme de cas d'utilisation « Consulter le trafic international entrant vers Lycamobile »

Le tableau 2.9 décrit textuellement le cas d'utilisation « Consulter le trafic international entrant vers Lycamobile ».

Tableau 2.9: Description textuelle du cas d'utilisation « Consulter le trafic international entrant vers Lycamobile »

Titre	« Consulter le trafic international entrant vers Lycamobile »
But	Visualiser des courbes présentant le trafic international entrant vers mobile TT dans une période donnée.
Résumé	En fonction du période précisé par l'agent de la DERAf / administrateur, des courbes montrant le trafic international entrant vers Lycamobile seront affichées.
Acteurs	Agent de la DERAf / Administrateur
Préconditions	
- Agent de la DERAf / administrateur authentifié.	
Post-conditions	
- Des données concernant le trafic international entrant vers Lycamobile seront visualisées.	
Scénario Nominal	
<ol style="list-style-type: none"> 1. L'agent de la DERAf / administrateur consulte la barre de menu pour sélectionner « Data → Trafic international entrant vers Lycamobile ». 2. Le système affiche l'interface où les données seront visualisées. 3. L'administrateur choisit la période dont il veut visualiser les données associées. 4. Le système affiche les courbes montrant le trafic international entrant vers Lycamobile sur la période précisée par l'agent de la DERAf / administrateur. 	
Enchaînement alternative	
<p>E1 : Période non précisée</p> <ol style="list-style-type: none"> 1. Le scénario reste bloqué à 2. 	

2.5. Diagramme de classes

Le diagramme de classes constitue l'un des types de diagrammes les plus utiles du langage UML. Il s'agit d'un type de diagramme de structure statique qui décrit la structure d'un système en affichant ses classes, ses attributs, ses opérations et ses relations entre les objets.

La figure 2.13 montre le diagramme de classe de mon application. Il admet six classes : utilisateur, compte, interface, courbe, trafic et abonné.

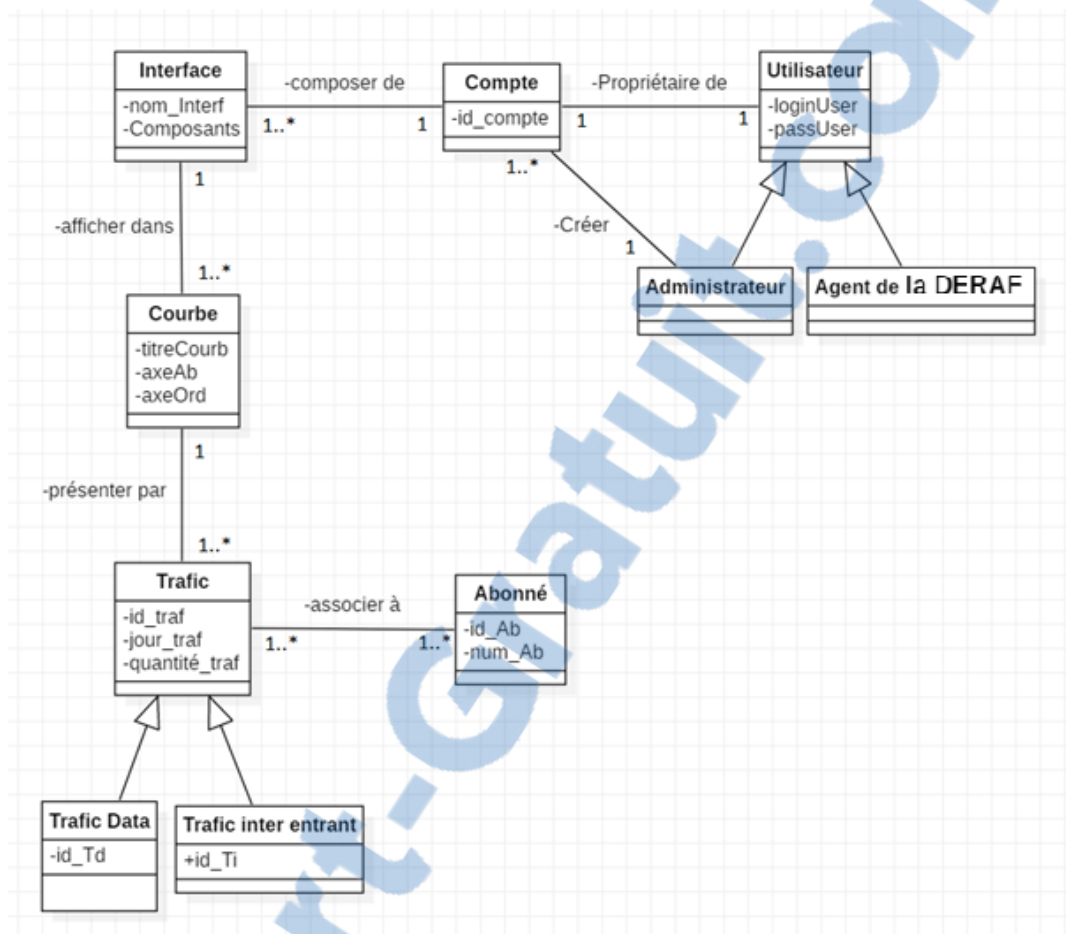


Figure 2.13 : Diagramme de classes

2.6. Diagramme d'activité

Le diagramme d'activité est un autre diagramme important en UML. Il décrit les aspects dynamiques du système.

Il s'agit en effet d'un organigramme essentiel pour représenter le flux d'une activité à une autre.

La figure ci-dessous présente le diagramme d'activité générale de notre application.

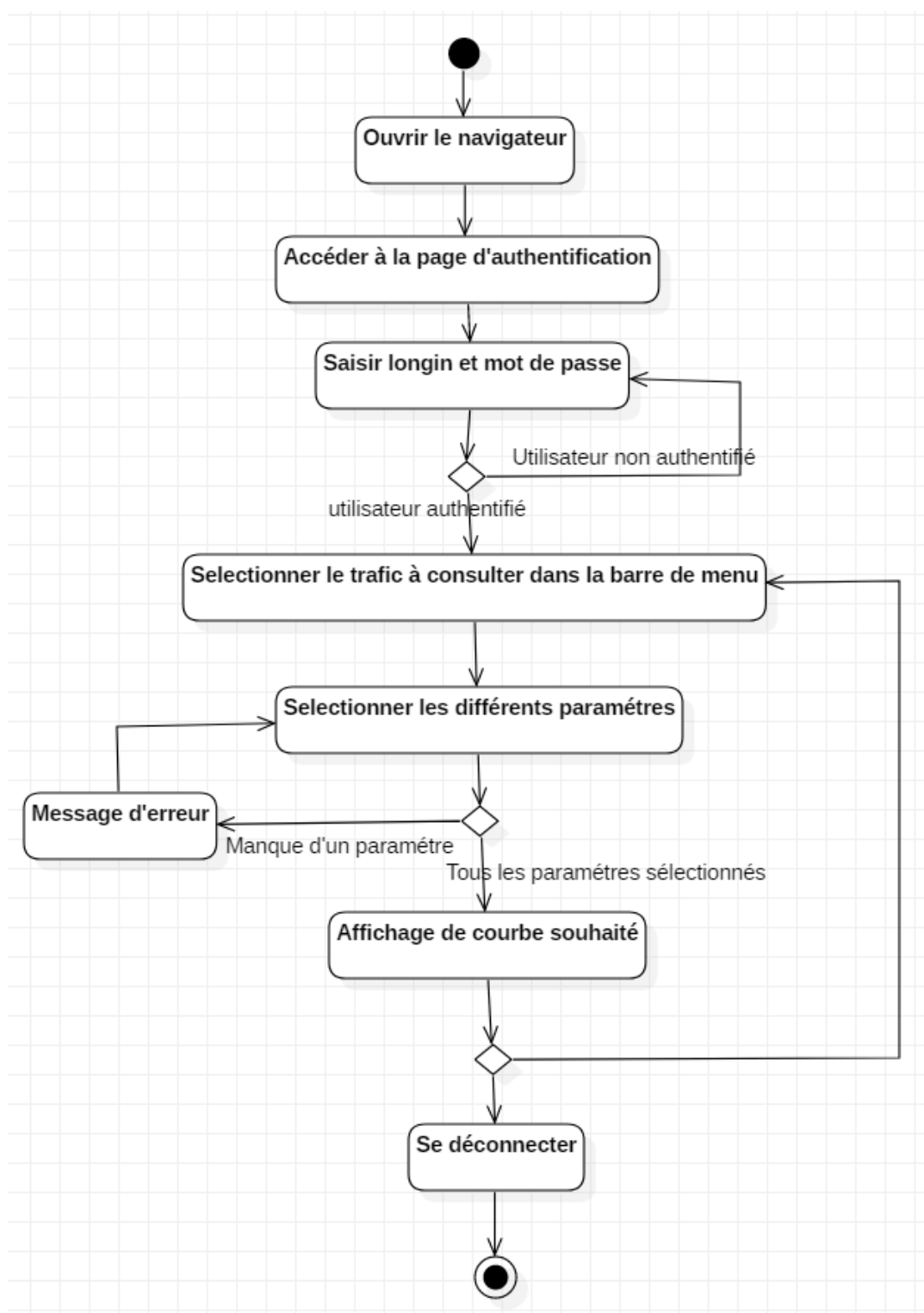


Figure 2.14 : Diagramme d'activité

2.7. Conclusion

Ce chapitre nous a permis de détailler la conception de notre application en présentant le diagramme de cas d'utilisation général, les diagrammes de cas d'utilisation de chaque cas, son description textuelle ainsi que le diagramme de classes.

Le chapitre suivant mettra en évidence, les différentes étapes de réalisation de notre application.

Chapitre 3. Réalisation

3.1. Introduction

Après avoir détaillé la conception adaptée à notre application, nous allons consacrer le dernier chapitre de ce rapport à la partie réalisation. Pour cela nous allons présenter dans un premier lieu l'environnement de développement et par la suite nous décrivons la phase d'implémentation en nous basant sur quelques interfaces.

3.2. Environnement de travail

3.2.1. Etude et choix technologiques

Avec l'évolution récente des technologies web, les développeurs et les technologues du monde entier n'ont jamais connu une période plus excitante pour la conception d'applications web modernes.

Le développement web ne se limite plus au HTML pur, au CSS et au JavaScript sur le front-end et à PHP / Perl sur le back-end. Il existe une pléthore de nouveaux langages, de structures web et d'outils à choisir pour tout développement d'application web. Alors que l'essor des technologies web a facilité le processus de développement des applications, les développeurs ont eu du mal à choisir la technologie parfaite.

Pour tout développement d'applications web, il est important de choisir la pile de technologies appropriée qui permet un prototypage rapide, une itération constante, la réutilisation du code, une efficacité et une robustesse maximales. Il est également important que la pile technologique soit facile à apprendre et à comprendre par les développeurs frontaux et principaux. Ainsi, le concept de JavaScript Full Stack a été développé. A l'origine, le développement web était basé sur la pile LAMP (Linux, Apache, MySQL, PHP / Perl) et Java (Java EE, Spring), constituée de différents langages de programmation. JavaScript a résolu ce

paradigme multilingue en introduisant une pile MEAN (MongoDB, Express, AngularJS, Node.js) basée sur un seul langage "JavaScript".

L'objectif de ce qui suit est alors d'étudier la pile MEAN, de présenter une analyse des forces et des faiblesses de cette pile pour mieux expliquer les raisons qui nous ont mené à l'utiliser et de fournir également l'architecture à suivre lors du développement de notre application web de pile MEAN.

3.2.2. Architecture de l'application

Le développement d'applications web est la combinaison du développement en front-end et en back-end. La figure 3.1 illustre l'architecture de notre application qui est basée sur l'ensemble des exigences étudiés. Il existe une séparation complète entre la logique côté serveur et la logique côté client. Le modèle est implémenté sur le serveur et la vue et le contrôleur sont implémentés dans le client.

De plus, l'application ne communique pas directement avec le serveur, mais utilise les API REST pour alimenter les données. L'application utilise les routes créées dans AngularJS pour naviguer dans les pages web.

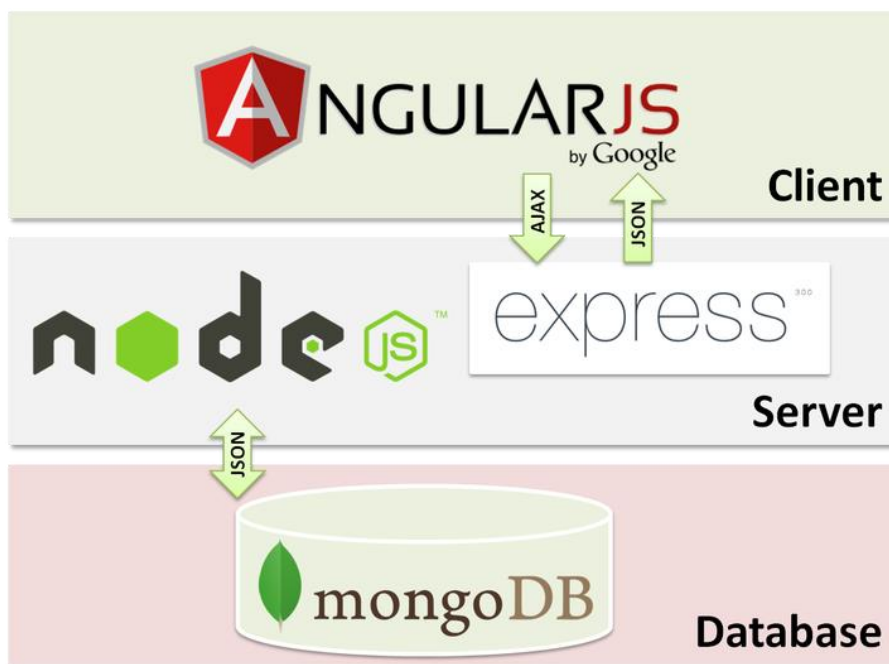


Figure 3.1 : Architecture de mon application^[1]

3.2.2.1. Développement en front end

Le développement web front-end, également appelé développement côté client, implique la création d'une interface utilisateur graphique pour les clients afin qu'ils puissent interagir avec l'application. Cela implique l'utilisation de technologies web et d'outils principaux tels que HTML, CSS et JavaScript.

- **HTML** est un langage dit de « marquage », de « structuration » ou de « balisage » dont le rôle est de formaliser l'écriture d'un document avec des balises de formatage. Les balises permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents.
- **CSS**, par contre, est un langage de feuille de style qui apporte des améliorations visuelles et stylistiques aux documents écrits en HTML.
- **JavaScript** est le langage le plus avancé parmi ces technologies. Il effectue une manipulation HTML DOM pour fournir une interface dynamique aux utilisateurs. De plus, il fournit une interface interactive aux utilisateurs en créant des messages contextuels, en validant les entrées de formulaire et en modifiant la disposition en fonction d'événements tels que les clics de souris ou les entrées de l'utilisateur.

Toutes ces technologies sont contrôlées par le navigateur pour fournir une interface web front-end.

3.2.2.2. Développement en back end

Le développement web en back-end, également appelé développement côté serveur, implique le développement de programmes informatiques et de bases de données pour servir le client. Les technologies d'arrière-plan se composent généralement des langages de programmation tels que Node.js.

3.2.2.3. RESTful API

REST est un style architectural utilisé dans le développement web pour créer des services web. Il définit uniquement les principes sur lesquels un service web est développé pour la

communication avec le serveur de clients. Tous les services web ou API (interfaces de programmation d'application) conçus avec l'architecture REST sont appelés API RESTful ou uniquement API REST.

REST fournit de bonnes performances, évolutivité et fiabilité dans un système informatique distribué.

3.3. Implémentation

Plusieurs programmes et outils ont été téléchargés et configurés pour créer un environnement pour notre application MEAN.

3.3.1. Environnement de développement

L'écosystème Node.js, avec l'ajout d'AngularJS sur le serveur front-end, Express comme backend et MongoDB comme base de données, a créé un nouveau terme dans le développement d'applications web appelé MEAN Stack.

3.3.1.1. Node.js

Node.js est le premier programme à télécharger car il fournit la plate-forme sur laquelle une application MEAN est créée.

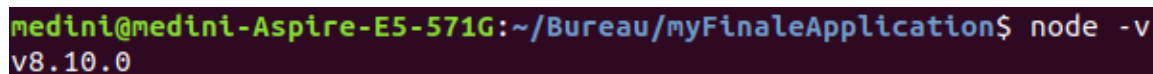
Node.js est une plate-forme de développement open source permettant d'exécuter du code JavaScript côté serveur. Node est utile pour développer des applications nécessitant une connexion permanente entre le navigateur et le serveur. Il est souvent utilisé pour des applications en temps réel telles que le chat, les flux d'actualités et les notifications push Web.

Node.js est destiné à fonctionner sur un serveur HTTP dédié et à utiliser un seul thread avec un processus à la fois. Les applications Node.js sont basées sur des événements et s'exécutent de manière asynchrone. Le code construit sur la plate-forme Node ne suit pas le modèle traditionnel de réception, traitement, envoi, attente, réception. Au lieu de cela, Node traite les demandes entrantes dans une pile d'événements constante et envoie de petites requêtes les unes après les autres sans attendre les réponses.

Node.js a un éco-système de développement riche avec plusieurs bibliothèques compatibles et gestionnaires de paquets. L'un des principaux gestionnaires de paquets fournis avec Node.js lors de l'installation est npm. Npm est exécuté à partir de l'interpréteur de ligne de commande (CLI) et il gère toutes les dépendances pour les applications Node.js. Au lieu de télécharger et de configurer manuellement les modules JavaScript pour les utiliser dans l'application, npm offre une alternative plus simple. Les noms des modules et des dépendances peuvent être inclus avec leurs numéros de version dans le fichier "package.json" dans la structure des dossiers, et les modules sont téléchargés en émettant une simple commande "npm install" depuis la CLI.

Le npm gère automatiquement tout le processus de téléchargement et enregistre tous les modules nommés dans le dossier « node-modules ». Les modules peuvent être mis à jour en modifiant leur numéro de version dans le fichier « package.json », et peuvent être facilement distribués en téléchargeant un fichier unique « package.json » sur le serveur au lieu de télécharger le grand dossier « node-modules ».

Après le téléchargement et l'installation, il est possible de le vérifier en émettant la commande « *node -v* » depuis le terminal, comme illustré à la figure 3.2.



```
medini@medini-Aspire-E5-571G:~/Bureau/myFinaleApplication$ node -v
v8.10.0
```

Figure 3.2 : Vérification de la version de Node.js

3.3.1.2. *Express*

Après l'installation de Node.js, un répertoire racine est créé et, à l'intérieur du répertoire, express est installé en émettant une commande à partir du terminal « *npm install -g express* ». Tous les modules et dépendances de nœud ont été installés à l'aide de la commande *npm install*.

Express est un module de Node qui fournit un cadre minimal et flexible pour les applications Web Node.js. En outre, il fournit des fonctions robustes et propres à ajouter aux modules Node dont le but de faciliter considérablement le développement de l'application Node.js ^[2].

Un serveur Express est composé de trois éléments constitutifs : router, routes et middleware.

La fonctionnalité principale d'un serveur web dépend de ses excellentes méthodes de routage. Dans une communication avec un serveur de clients, un client demande des ressources au serveur, le serveur localise les ressources et répond en envoyant les ressources au client. Il s'agit de la fonctionnalité de base d'un serveur web et il nécessite d'excellentes méthodes de routage pour répondre à la demande.

Express facilite grandement ce travail fastidieux en permettant aux développeurs de créer des itinéraires dans une structure simple. Une route dans Express est une combinaison d'un verbe HTTP et d'un chemin. Le verbe HTTP est généralement l'une des quatre méthodes HTTP : GET, POST, PUT et DELETE, et le chemin correspond à l'emplacement de la ressource (URI).

3.3.1.3. MongoDB

MongoDB est installé à partir du site web officiel.

Il s'agit en fait d'une base de données de documents open-source, non relationnelle. MongoDB dévie de la nécessité de créer un mapping objet relationnel (ORM), pour un développement rapide des applications. Contrairement aux bases de données relationnelles, il ne contient pas de colonnes et de lignes. Le format dans lequel MongoDB stocke les données est appelé BSON, qui signifie JSON binaire. Puisque JSON est le moyen JavaScript de stocker les données, MongoDB fonctionne parfaitement avec les applications créées avec la pile JavaScript ^[3].

Mongoose est une modélisation de données MongoDB pour Node.js. Il a été créé pour résoudre le problème de la validation de données complexes, de la conversion et de la logique métier dans MongoDB. Il fournit une fonctionnalité de modélisation de données simple et élégante à l'application.

En termes techniques, Mongoose fournit des règles de validation des données, des fonctions de création de requêtes et une logique métier aux données de l'application. De plus, il fournit un ensemble complet de nouvelles fonctionnalités à utiliser sur MongoDB. Il peut gérer les connexions à la base de données MongoDB, ainsi que lire, écrire et enregistrer des données. Il permet également d'enregistrer uniquement des données valides dans MongoDB en fournissant des règles de validation au niveau du schéma.

Mongoose a été installé en utilisant la commande : « *npm install mongoose* ».

3.3.1.4. AngularJS

AngularJS est un framework JavaScript permettant de créer des applications Web frontales. Il est conçu pour créer une application à page unique avec l'introduction de l'architecture MVC (Model View Controller) sur le front-end. Le framework AngularJS étend les fonctionnalités HTML en fournissant différents éléments et attributs, ce qui permet de créer facilement une grande application web. Il étend HTML avec ng-directives, lie les contrôles d'entrée HTML aux données d'application avec la directive ng-app et lie les données d'application à la vue avec la directive ng-bind ^[4].

3.3.2. Structure d'application

Après l'installation et la configuration de tous les outils et programmes requis, le processus de développement a été lancé.

La figure ci-dessous illustre une vue d'ensemble de la structure des dossiers de notre application.

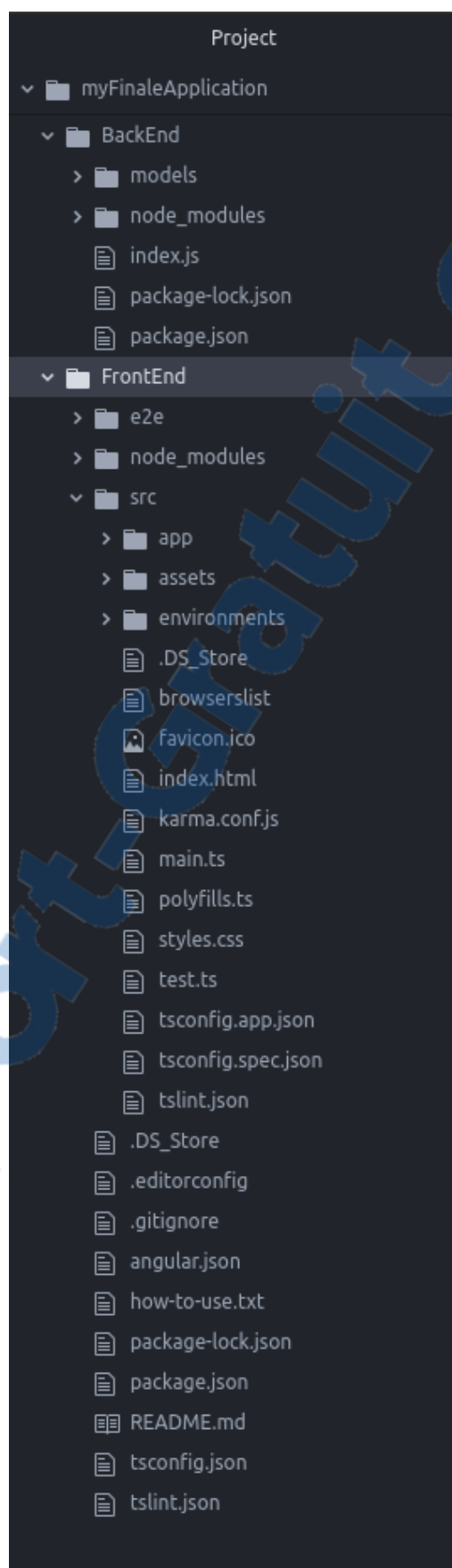


Figure 3.3: Structure du dossier d'application

Le fichier « *package.json* » contient toutes les dépendances requises pour exécuter l'application.

Le fichier « *app.js* » contient tous les middleware Express et les routes API de l'application. Il existe une division claire entre la logique côté serveur et la logique côté client. Les modèles de données sont placés côté serveur dans le répertoire « *BackEnd* », tandis que les vues et les contrôleurs sont placés côté client. Toutes les dépendances côté serveur sont installées dans le dossier « *node_modules* » et toutes les dépendances côté client, y compris AngularJS, sont installées dans le dossier « *FrontEnd* ».

3.3.3. *Package.json*

Le fichier « *package.json* » contient toutes les informations importantes sur l'application et les dépendances requises pour générer et exécuter l'application.

La figure 3.4 illustre une partie du contenu de ce fichier. Il contient le nom de l'application, la version et toutes les dépendances.

```
1 {  
2   "name": "my-first-app",  
3   "version": "0.0.0",  
4   "scripts": {  
5     "ng": "ng",  
6     "start": "ng serve",  
7     "build": "ng build",  
8     "test": "ng test",  
9     "lint": "ng lint",  
10    "e2e": "ng e2e"  
11  },  
12  "private": true,  
13  "dependencies": {  
14    "@angular/animations": "^6.0.0",  
15    "@angular/cdk": "^6.4.7",  
16    "@angular/common": "^6.0.0",  
17    "@angular/compiler": "^6.0.0",  
18    "@angular/core": "^6.0.0",  
19    "@angular/forms": "^6.0.0",  
20    "@angular/http": "^6.0.0",  
21    "@angular/material": "^6.4.7",  
22    "@angular/platform-browser": "^6.0.0",  
23    "@angular/platform-browser-dynamic": "^6.0.0",  
24    "@angular/router": "^6.0.0",  
25    "bootstrap": "3.3.7",  
26    "chart.js": "^2.7.2",  
27    "core-js": "^2.5.4",  
28    "firebase": "^4.6.0",  
29    "map": "^1.0.1",  
30    "mongoose": "^5.2.12",  
31    "rxjs": "^6.0.0",  
32    "rxjs-compat": "^6.1.0",  
33    "zone.js": "^0.8.26"  
34  },  
35  "devDependencies": {
```

Figure 3.4 : Fichier « *package.json* »

3.4. Interfaces de l'application

Dans ce qui suit, nous présentons le fruit de notre travail durant toute la période de notre stage en exposant quelques interfaces de l'application.

La figure 3.5 illustre l'interface d'authentification de l'application disposant de deux zones de texte (mail et mot de passe) et un bouton de connexion.

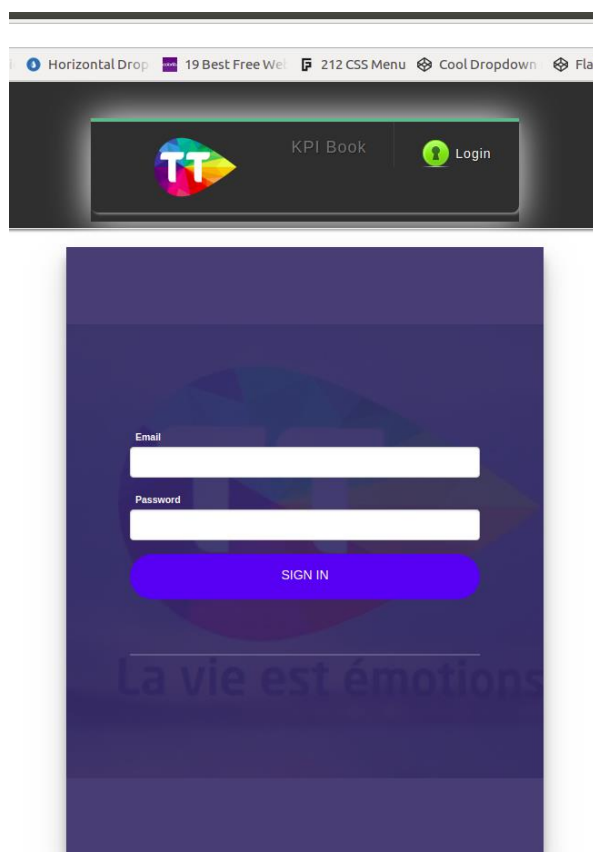


Figure 3.5 : Interface d'authentification

Après l'authentification, l'utilisateur peut visualiser l'interface de la figure ci-dessous qui présente la page de d'accueil d'un compte d'un administrateur.

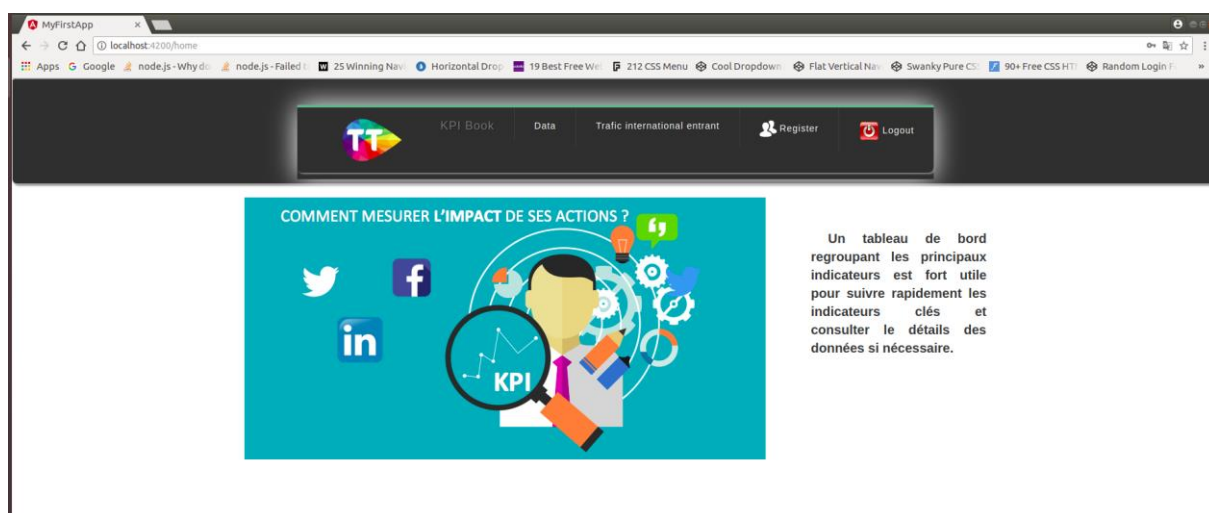


Figure 3.6: Page d'accueil de l'application

On peut remarquer qu'un menu s'affiche à l'entête du page. En consultant le bouton « S'inscrire », l'application affiche une page permettant à l'administrateur d'enregistrer des nouveaux utilisateurs tel qu'il est présenté par la figure ci-dessous.

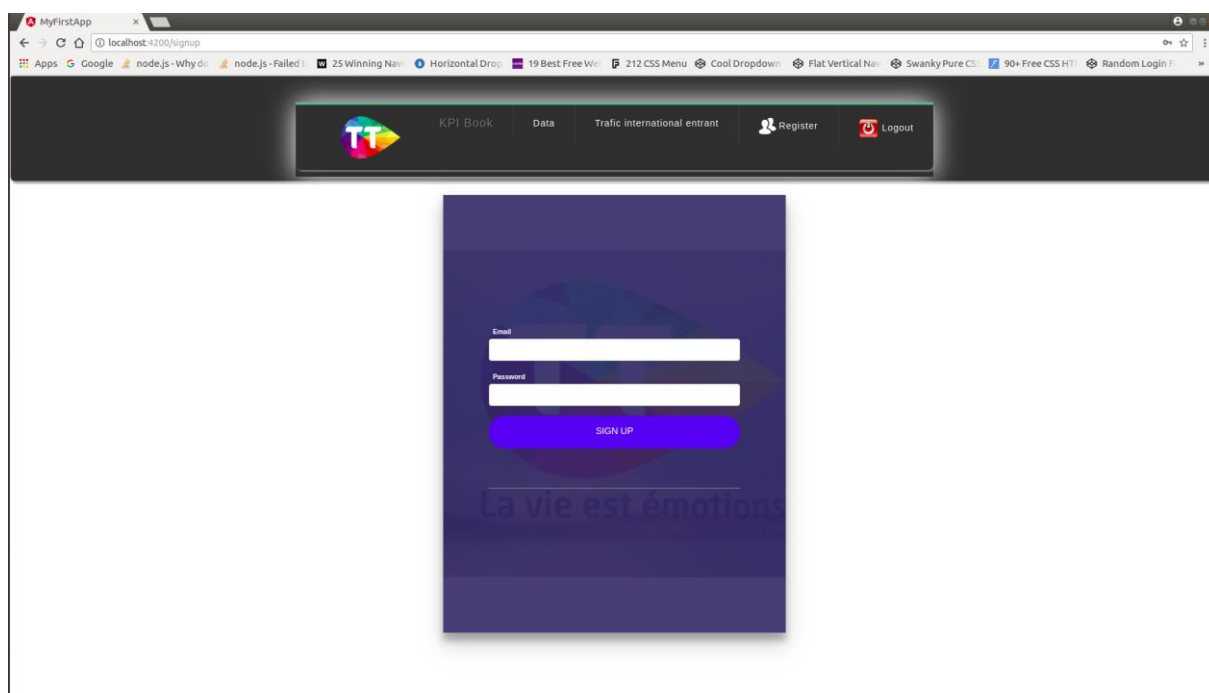


Figure 3.7 : Page d'enregistrement

En cliquant sur « Data » l'administrateur peut consulter le trafic Skype, Viber, autre VOIP ou non VOIP comme illustré par la figure ci-dessous.

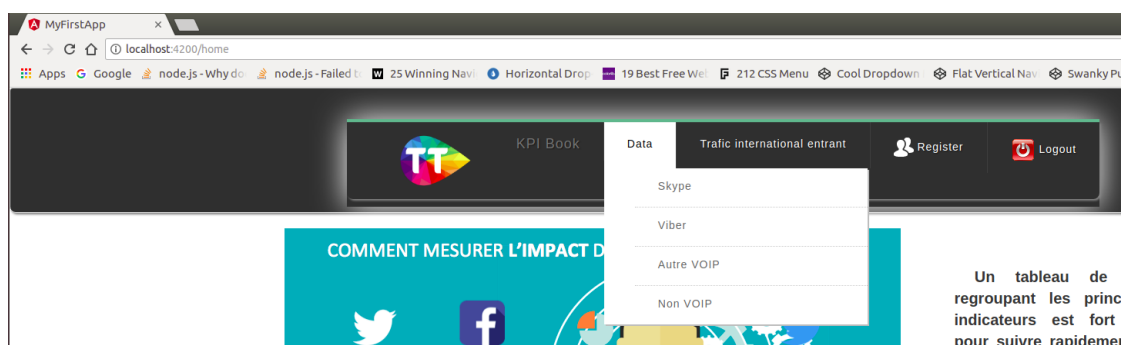


Figure 3.8 : Les options data

En cliquant sur « Trafic international entrant », l'utilisateur peut consulter les types de trafic indiquées par la figure ci-dessous.

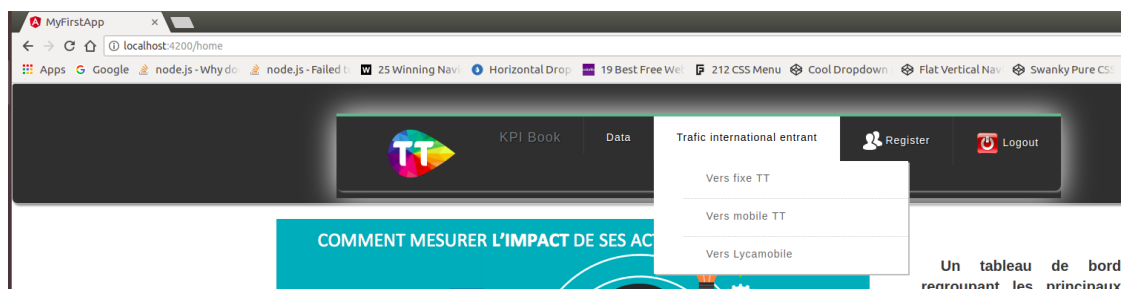


Figure 3.9 : Les options du trafic international entrant

En choisissant par exemple de consulter le trafic international entrant vers fixe TT, la page illustrée par la figure ci-dessous s'affiche. Elle contient une liste roulante à plusieurs choix permettant à l'utilisateur de préciser la période (une semaine, deux semaines, trois semaines ou un mois) sur laquelle il veut visualiser les données et en cliquant sur le bouton valider une courbe s'affiche.

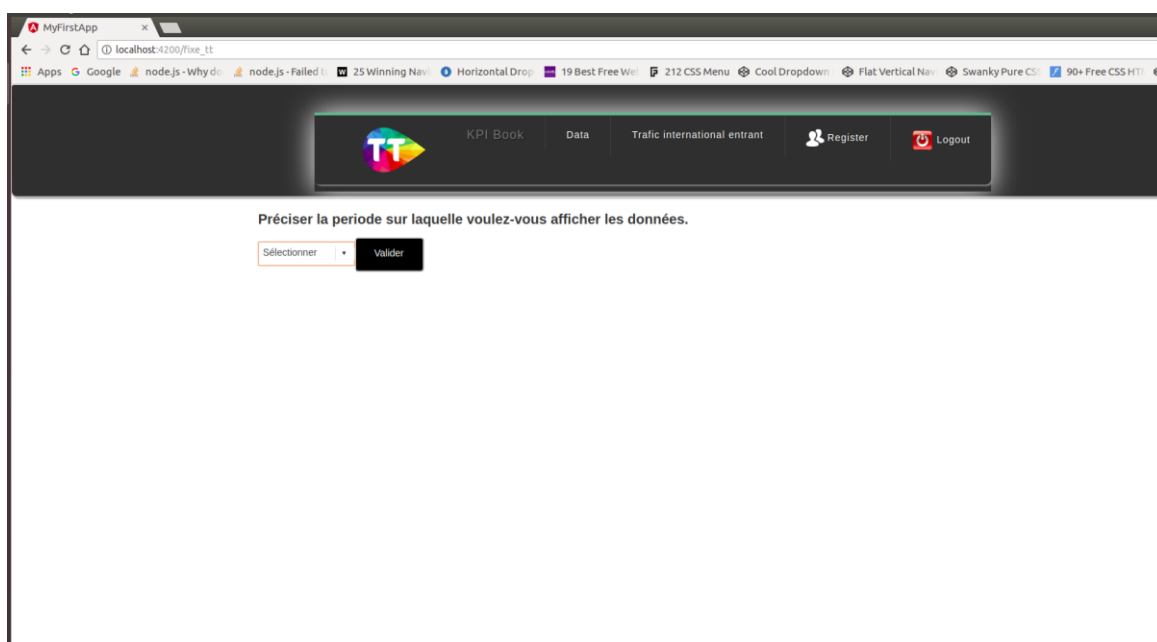


Figure 3.10: Page montrant le trafic international entrant vers fixe TT

La figure ci-dessous montre par exemple une courbe contenant des données d'un mois comme demandé par l'utilisateur.

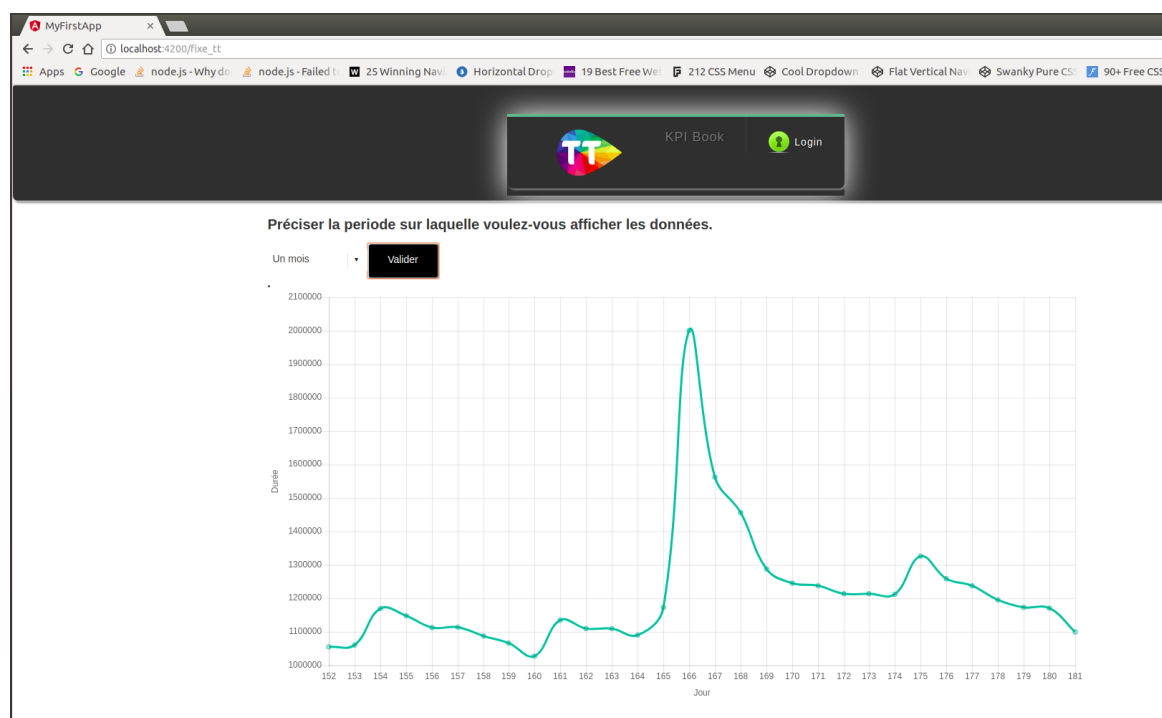


Figure 3.11: Interface montrant une courbe du trafic international entrant vers fixe TT sur un mois

Si l'utilisateur n'a rien coché (ou il a coché « Sélectionnez »), un message d'erreur s'affiche comme illustré par la figure ci-dessous.

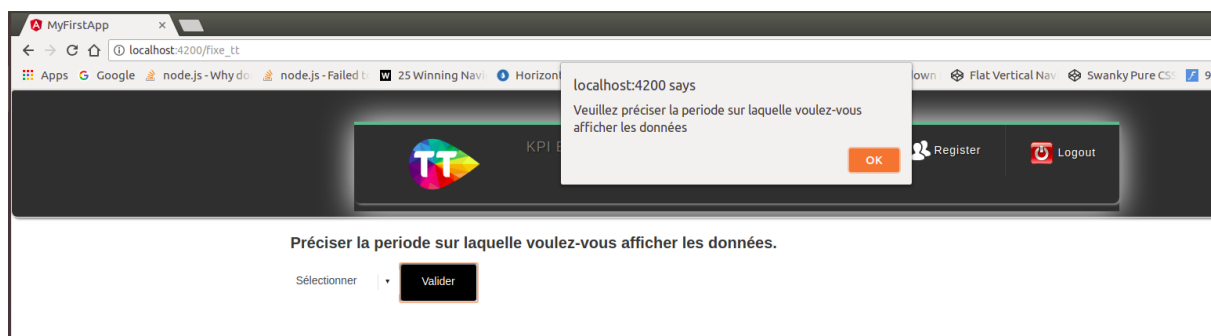


Figure 3.12: Affichage d'un message d'erreur

Si maintenant l'utilisateur veut consulter le trafic Skype, la page illustrée par la figure ci-dessous s'affiche en le choisissant dans la barre de menu.

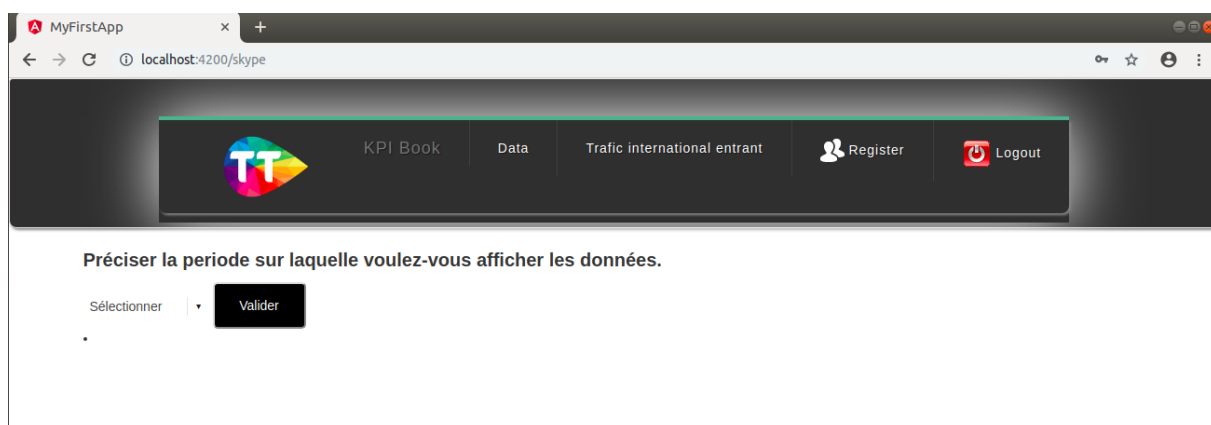


Figure 3.13 : page permettant de consulter le trafic Skype

Cette interface contient une liste roulante à plusieurs choix permettant à l'utilisateur de préciser la période (une semaine, deux semaines, trois semaines ou un mois) sur laquelle il veut visualiser les données et en cliquant sur le bouton valider une courbe s'affiche.

La figure ci-dessous montre une courbe correspondante au trafic Skype d'une semaine.

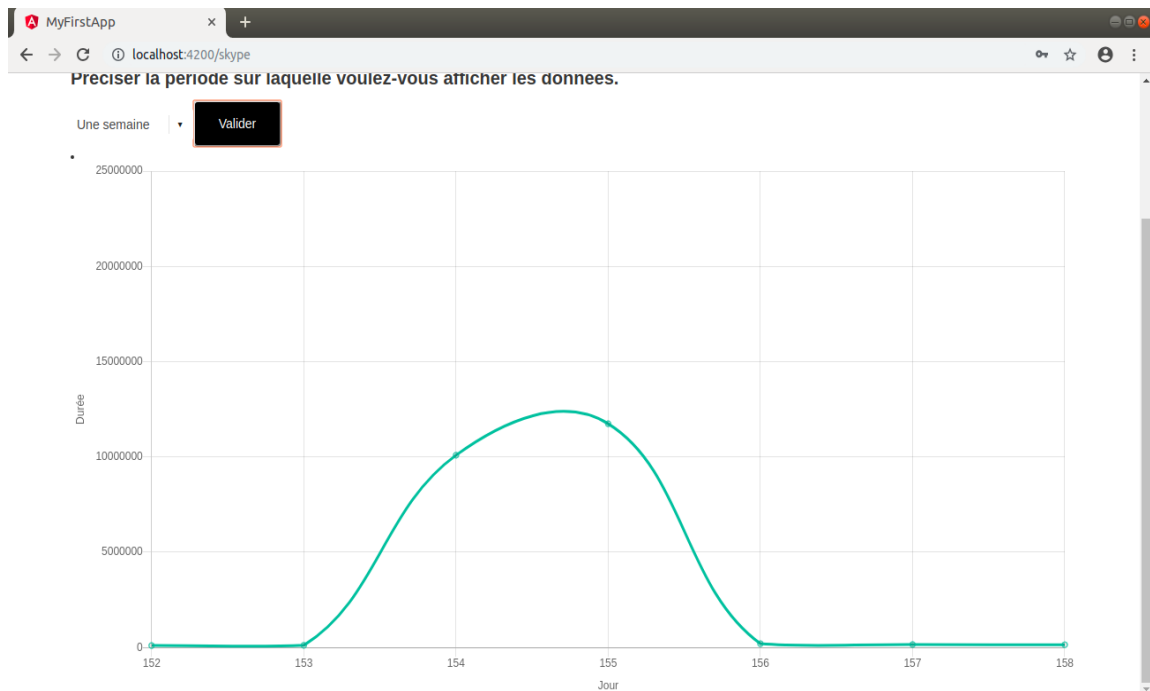


Figure 3.14 : Interface montrant une courbe du trafic Skype sur une semaine

3.5. Conclusion

A ce stade, notre projet atteint sa fin. Tout au long de ce chapitre, nous avons abordé notre environnement de travail et justifié nos choix technologiques. Par la suite nous avons entamé la présentation des principales fonctionnalités implémentées suite au développement par des captures écrans présentant les principales interfaces de notre application.

Conclusion générale

Pour certaines organisations, le processus de mesure des données est un exercice important comme il les permet d'atteindre des objectifs stratégiques. En effet, utilisés les KPI devient un élément important de la stratégie d'une entreprise.

C'est dans ce cadre que s'intègre notre projet qui a pour but la conception et le développement d'une application web assurant le suivi d'un ensemble d'indicateurs clés de performance.

Le présent rapport détaille toutes les étapes par lesquelles nous avons passé pour arriver au résultat attendu. Nous avons commencé par la compréhension du contexte général du projet et l'identification des différentes exigences du futur système. Nous avons préparé, par la suite, un planning de travail qui nous a aidé à réussir la réalisation de la totalité de l'application.

Ainsi à travers ce rapport nous avons présenté une analyse des forces et des faiblesses de la pile MEAN pour mieux expliquer les raisons qui nous ont mené à l'utiliser et nous avons montré également l'architecture suivant laquelle nous avons développé notre application web de pile MEAN. Ce choix nous a permis de réaliser une application complète répondant aux attentes de l'entreprise.

Finalement, et comme perspectives, ça sera intéressant si le suivi d'autre indicateurs de performance sera intégré dans l'application. Ces indicateurs peuvent par exemple correspondent au suivi de la quantité du trafic ADSL, SDSL ou VDSL.

Webographie

- [1] [En ligne], Disponible sur: <https://hackernoon.com/mean-stack-development-for-developers-4d88c40c4103> , [Consulté en Mai 2019]

- [2] [En ligne], Disponible sur: <https://openclassrooms.com/fr/courses/1056721-des-applications-ultra-rapides-avec-node-js/1057503-le-framework-express-js> , [Consulté en Mai 2019]

- [3] [En ligne], Disponible sur: <https://medium.freecodecamp.org/introduction-to-mongoose-for-mongodb-d2a7aa593c57> , [Consulté en Mai 2019]

- [4] [En ligne], Disponible sur: <http://www.journaldunet.com/web-tech/developpeur/1153790-angularjs-les-dessous-techniques-du-framework-javascript/> , [Consulté en Mai 2019]

Résumé

Un indicateur de performance clé est une mesure quantifiable qui démontre l'efficacité avec laquelle une organisation atteint ses objectifs commerciaux clés. Ces indicateurs sont des mesures commerciales utilisées par les dirigeants d'entreprise et les gestionnaires pour suivre et analyser les facteurs considérés comme essentiels au succès d'une entreprise. Donc ils constituent une partie importante des informations nécessaires pour déterminer et expliquer comment une entreprise progressera pour atteindre ses objectifs commerciaux et marketing. Notre travail consiste alors à concevoir et à mettre en place une solution Web assurant le suivi d'un ensemble des indicateurs clés de performance au profit de l'une des directions de Tunisie Télécom : La Direction Exécutive Revenu Assurance et Fraude. Cette solution aide à bien évaluer les situations et à prendre des décisions adaptées.

Mots Clés : KPI, Web, pile MEAN, performance, décision

Abstract

A key performance indicator is a quantifiable measure of how effectively an organization achieves its key business objectives. These indicators are business measures used by business leaders and managers to track and analyze the factors considered essential to the success of a business. So they are an important part of the information needed to determine and explain how a business will progress to achieve its business and marketing goals. Our job is then to design and implement a web-based solution monitoring a set of key performance indicators for the benefit of one of the divisions of Tunisie Telecom: The Executive Directorate Revenue Assurance and Fraud. This solution helps to properly assess situations and make appropriate decisions.

Key words: KPI, Web, MEAN stack, performance, decision