

MEMOIRE

DE STAGE DE FIN D'ETUDES

Pour l'obtention du

«Mastère professionnel en Nouvelles Technologies des
Télécommunications et Réseaux (N2TR)»

Présenté par :

Abir Ben Hadid

Titre

Conception et développement d'un outil de
facturation de voix sur IP : « BILLPRO »

Soutenu le :12-07-2018

Devant le jury :

Président : Mr. GHORBEL Khaled

Encadreur : Mr. SAKOUHI Mohamed

Rapporteur : Mme. Hela Boucetta

Année Universitaire : 2017 / 2018

Dédicaces

Je dédie ce travail à

Mes parents,

Aucun mot, aucune dédicace ne peut exprimer mon respect et ma considération pour les sacrifices et tous les efforts que vous avez consentis pour mon éducation ainsi que mon bien être. Trouvez dans ce travail le fruit de votre dévouement, de votre patience et l'expression de ma gratitude et mon profond amour. Puisse Dieu, vous accorder santé, bonheur et longue vie et faire en sorte que jamais je ne vous déçoive.

Ma chère sœur Nada

Je te souhaite une vie pleine de bonheur et de succès

Toute ma famille, Tous mes amis,

En leur souhaitant le succès dans leur vie aussi bien professionnelle que familiale.

Abir ben hadid

Remerciements

Je tiens à adresser mes sincères remerciements :

A **Monsieur Mohamed Sakouhi** et **Monsieur Mohamed Aziz Ferchichi** pour avoir bien voulu m'encadrer, pour tout le temps qu'ils m'ont octroyés et pour tous les conseils qu'ils m'ont prodigués. Qu'ils trouvent ici l'expression de ma profonde gratitude.

A tous les membres du jury :

Mr. GHORBEL Khaled & Mme. Hela Boucetta

A tous ceux qui, directement ou indirectement, ont aidé à la finalisation de ce travail. Merci à toutes et à tous.

Merci à toutes et à tous

Table des matières

Contenu

Introduction Générale.....	6
Chapitre 1 : Présentation générale.....	8
I. Introduction	8
II. Présentation de l'organisme d'accueil	8
1. Présentation générale.....	8
2. Organigramme	8
III. Notions on VoIP.....	9
IV. Présentation du sujet	10
V. Etude de solutions existantes	11
VI. Critique des solutions existantes:.....	12
VII. Les objectifs du nouveau système et solution proposée.....	12
VIII. Méthodologie et formalise adoptés.....	12
1. Choix de processus de développement.....	12
2. Formalisme adopté :.....	14
IX. Conclusion.....	15
Chapitre II : Etude préliminaire	16
Introduction :.....	16
I. Identifications des acteurs du système	16
1. Identification des acteurs	16
II. Exigences fonctionnelles	16
III. Besoins non fonctionnels :.....	18
IV. Conclusion.....	19
Chapitre III : Capture des besoins fonctionnels.....	20
I. Introduction	20
II. Diagramme de cas d'utilisation de l'application "BILLPRO"	20
III. Raffinement et description détaillée des cas d'utilisation.....	21
IV. Conclusion.....	27
Chapitre IV : Capture des besoins Techniques	28
I. Introduction	28
II. Spécification de l'architecture système.....	28
1. Environnement de développement.....	29

III. Conclusion :	33
Chapitre V: Analyse et Conception	34
I. Introduction	34
II. Architecture globale	34
1. L'architecture de General BILLPRO	34
2. Diagrammes de déploiement	35
2. Diagramme de paquet	36
3. Conception of BillPro	37
III. Diagramme de séquence	40
IV. Conclusion	42
Chapitre VI : Réalisation	43
I. Introduction	43
II. Environnement de travail :	43
1. Environnement matériel	43
2. Environnement logiciels	43
III. Etude de risques	44
IV. Gestion du projet	44
V. Principaux scénarios :	45
1. Launching application	45
2. Page d'authentification	45
3. L'interface Dashboard	46
4. Gestion IPBX	46
5. Gestion de la facture	47
6. Page Consultation statistique	49
7. Statistique par opérateur	50
8. Gestion de la méthode d'imposition	50
9. Interfaces de gestion des utilisateurs téléphoniques :	54
VI. Conclusion	55
Conclusion et futur travail	56
Nétographie	57
Annexe	58
Annexe 1: Server log line	58
Annexe 2: Ajouter une page supérieure	58
Annexe 3: Page de consommation par ligne	59

Table des Figures

Figure 1:organigramme de l'agence Vioservices	8
Figure 2: Contraintes fonctionnelles et techniques.....	14
Figure 3:Diagramme de cas d'utilisation générale	20
Figure 4 : Raffinement pour le cas d'utilisation " gérer journal"	21
Figure 5 : Raffinement pour le cas d'utilisation " gérer Facture"	22
Figure 6: Raffinement pour le cas d'utilisation "gérer méthode d'imposition"	23
Figure 7: Diagramme de gestion de cabine téléphonique étendu	25
Figure 8: Gestion étendue des utilisateurs de téléphone.....	26
Figure 9:MVT modèle.....	28
Figure 10:Django logo	30
Figure 11:MySQL logo	30
Figure 12:Python logo	32
Figure 13:Architecture générale de l'application	34
Figure 14:diagrammes de déploiement.....	35
Figure 15:diagramme de paquet	36
Figure 16 : diagramme de class	39
Figure 17:Créer des méthodes de facturation	40
Figure 18:Exporter les méthodes de facturation.....	41
Figure 19:Extraire facture	41
Figure 20:diagramme de Gantt	44
Figure 21; lancement du serveur Application.....	45
Figure 22:page d'authentification	45
Figure 23:tableau de bord de l'application	46
Figure 24:Page de configuration IPBX.....	47
Figure 25:page de consommation de cabine téléphonique.....	48
Figure 26:page de consommation d'un utilisateur de téléphone	48
Figure 27:Page statistique personnalisable.....	49
Figure 28:page des Statistiques par opérateur	50
Figure 29:page de gestion des méthode d'imposition.....	51
Figure 30:Page de création des méthodes d'imposition.....	51
Figure 31:Erreur de suppression des méthodes de taxation	52
Figure 32:changer les méthodes d'imposition	52
Figure 33: suppression réussite	53
Figure 34 : page de mise à jours	53
Figure 35: importation de csv.....	54
Figure 36:page de gestion d'utilisateur.....	54
Figure 37:formulaire d'ajout d'utilisateur.....	55

Introduction Générale

Depuis quelques années, la technologie VoIP commence à intéresser les entreprises, surtout celles de service comme les centres d'appels. La migration des entreprises vers ce genre de technologie n'est pas pour rien. Le but est principalement présenté comme suit : minimiser le coût des communications ; utiliser le même réseau pour offrir des services de données, de voix, et d'images ; et simplifier les coûts de configuration et d'assistance.

Le réseau VoIP est généralement basé sur deux équipements : le PABX ou l'IPBX et l'IPPhone. Le plus important est l'IPBX; c'est un équipement capable de gérer le réseau interne de données et de voix d'une entreprise et de le confier à un fournisseur de services téléphoniques externe.

Comme la technologie a un avantage, elle a aussi un défaut. Dans notre cas, le défaut se présente lorsque l'IPBX ne peut pas gérer la facturation; il envoie juste une ligne de journal ou génère un fichier journal. L'entreprise cliente a besoin d'un logiciel pour consulter et contrôler sa taxation téléphonique et ses appels sortants.

Actuellement, il existe de nombreux logiciels de facturation ; chacun a ses spécifications et ses cas d'utilisation qui varient selon les besoins de l'entreprise.

Notre projet consiste à contrôler les appels sortants et la consommation, et rendre la gestion plus facile et flexible.

Pour cela, nous devons utiliser le modèle et les méthodes de travail les plus simples et efficaces, et nous devons nous concentrer sur la flexibilité et l'évolutivité de l'application pour qu'elle s'adapte à toutes les tailles et à tous les types d'environnement.

Le contenu de notre rapport sera structuré en six chapitres qui reflètent la démarche adoptée pour la réalisation de ce projet :

Le premier chapitre présente le cadre général, le sujet de notre application et propose la démarche de développement qui a été adoptée dans la réalisation de l'application.

Le deuxième chapitre expose l'étude préliminaire, qui a permis d'identifier les besoins fonctionnels et opérationnels et de définir le périmètre du travail.

Les deux chapitres suivants donnent une vision plus détaillée du système à travers la capture des besoins fonctionnels et techniques.

Le cinquième chapitre détaillera la partie analyse et conception.

Enfin, le **dernier chapitre** décrira l'étape de réalisation, avec le choix de l'environnement de développement, et les interfaces des deux applications.

Et nous clôturons notre rapport par une **conclusion** qui offre une synthèse du travail réalisé et présente les perspectives.

Chapitre 1 : Présentation générale

I. Introduction

Nous nous proposons dans ce chapitre introductif de mettre le projet dans son cadre général. Nous commencerons par une présentation brève de l'organisme d'accueil. Par la suite, nous exposerons l'objectif du travail qui nous a été confié. Après, nous présenterons quelques solutions similaires ayant des objectifs en commun avec notre présent projet. Finalement, nous exposerons la méthodologie et les langages adoptés.

II. Présentation de l'organisme d'accueil

1. Présentation générale

Vioservice est une agence maîtrisant les dernières technologies web (Drupal, Symfony, Magento, WordPress, Yii...) et mobile (iOS, Android,...). Vioservice accompagne ces clients dans la conception et le développement des leurs projets web et mobile.

L'Agence Vioservices réalise aussi les installations des réseaux informatiques quelle que soit la topologie des locaux en conformité avec les normes internationales.

2. Organigramme

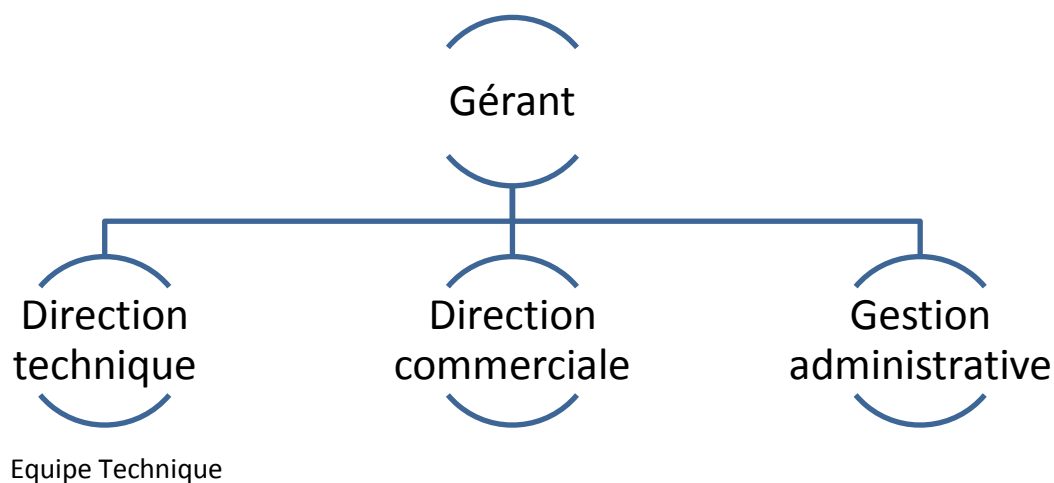


Figure 1:organigramme de l'agence Vioservices

III. Notions on VoIP

✓ PABX ou Private Automatic Branch eXchange

C'est un système électronique indépendant, il s'agit d'un autocommutateur téléphonique privé. Ce système est destiné aux petites, moyennes et grandes entreprises qui prennent en charge la gestion et la distribution de tous les appels, que ce soient entrants ou sortants. Il sert principalement à assurer la liaison entre des lignes internes et des lignes externes. Plus précisément, un PABX relie les postes téléphoniques d'une entreprise et le réseau de téléphonie public.

Ce système électronique permet de mettre en œuvre plusieurs fonctions comme autoriser les émissions d'appel entre les postes internes sans pour autant passer par le réseau public, proposer des différents services téléphoniques comme les transferts d'appels ou conférences, fournir des services de couplage téléphonie-informatique, gérer les terminaux téléphoniques que ce soient des postes numériques ou analogiques, etc.

On peut distinguer deux types de PABX : les PABX traditionnels et les PABX-IP. Du point de vue fonctionnement, un PABX peut fonctionner tout seul, en réseau homogène, en réseau hétérogène ou encore en réseau mixte.

Pour finir, il existe également des PABX Virtuels qui gomment les contraintes liées aux PABX physiques tout en assurant une téléphonie de haute qualité tout en réduisant les coûts pour les entreprises.

✓ IPBX :

Un système téléphonique VoIP/IPBX est constitué d'un ou de plusieurs téléphones SIP / VoIP, d'un serveur IPBX et facultativement d'une passerelle VoIP. Le serveur IPBX est similaire à un serveur proxy : les clients SIP sont soit des softphones ou des téléphones matériels, enregistrés auprès d'un serveur IPBX. Au cas du besoin de passer un appel, ils demandent à l'IPBX d'établir la connexion. L'IPBX possède un registre de tous les téléphones/usagers et de leur adresse SIP respectives et il peut ainsi connecter un appel

interne ou router un appel externe soit par le biais d'une passerelle VoIP ou d'un opérateur de service VoIP

✓ Voix sur ip

La Voix sur IP, raccourci de voix par le protocole internet, est aussi connue sous le terme de VoIP. La voix sur IP se réfère à la diffusion du flux de la voix sur les réseaux Internet, au lieu des réseaux téléphoniques RTC traditionnels. Le protocole Internet (IP) a été conçu à l'origine pour la gestion de réseaux de données puis après son succès, le protocole a été adapté à la gestion de la voix, en transformant et transmettant l'information en paquet de données IP. La VoIP est à présent disponible sur de nombreux smartphones, ordinateurs et tablettes.

La voix sur IP (VoIP) peut faciliter des tâches et fournir des services qu'il serait difficile ou coûteux de mettre en œuvre en utilisant le réseau RTC traditionnel:

Plus d'un appel téléphonique peut être transmis sur la même ligne téléphonique haut-débit. De cette manière, la voix sur IP peut faciliter l'ajout de lignes téléphoniques à l'entreprise, sans avoir besoin de lignes téléphoniques matérielles supplémentaires.

Les fonctionnalités qui sont habituellement facturées par les sociétés de télécom, tels que le transfert d'appel, l'ID d'appelant ou la composition automatique, sont simples avec la technologie de la voix sur IP.

IV. Présentation du sujet

L'agence Vioservice m'a confié la conception et l'implémentation de la solution de facturation de voix ip «**Billpro** » Suite à la demande du client qui utilise un logiciel de taxation limité, consomme de nombreuses ressources, périmés et non personnalisables. Le but de notre application est donc de rendre le processus de facturation plus facile et flexible. En fait, nous devons concevoir et développer une application qui améliorerait certaines anciennes fonctionnalités logicielles, à connaître les utilisateurs et la gestion hiérarchique ..., et inclurait de nouvelles fonctionnalités permettant à l'administrateur de consulter plus d'informations sur la consommation et de l'application ainsi que rendre la configuration plus facile.

V. Etude de solutions existantes

Dans cette partie, nous allons effectuer une étude d'un échantillon de solutions similaires existantes sur le marché national et international qui adressent les mêmes objectifs que notre présent travail. Dans ce qui suit nous étudierons les 2 solutions suivantes :

✓ **Wintax**

Wintax ou GCT est un logiciel de facturation par téléphone et d'analyse de la qualité de la réception téléphonique pour les PME et les hôtels. Il permet aux entreprises de surveiller de manière indépendante leur trafic téléphonique et de contrôler leur consommation par groupe d'utilisateurs, pour des services ou des sites. [N1]

Ses principales caractéristiques sont :

- Statistiques des appels entrants
- Efficacité par heure
- Balance Bilans synthétiques de la consommation téléphonique
- Ventilation de la consommation par article ou groupe d'articles selon le bottin téléphonique de l'entreprise ou de l'hôtel
- États statistiques multi-opérateurs
- Tableaux de bord sur la qualité de la réception téléphonique

✓ **Vigie**

C'est une solution efficace et sécurisée pour vos besoins de gestion, de suivi et d'analyse des flux d'appels qui détermine le taux exact de consommation d'appels téléphoniques pour chaque personne pour chaque unité au sein de l'entreprise. Il joue un rôle important pour limiter l'abus de l'utilisation du téléphone. ABS Computer a été créé pour y parvenir dans sa nouvelle VIGIE, qui se distingue par sa grande capacité pour le calcul des appels téléphoniques de taux de toute nature. Il dispose d'un numéro de téléphone électronique 80000 pour les entreprises publiques et privées que vous pouvez mettre à jour en ajoutant vos numéros de préférence avec un moteur de recherche facile à utiliser. [N2]

VI. Critique des solutions existantes:

Voici les anomalies que nous avons décelées dans les solutions informatiques existantes que nous venons de présenter :

- ✓ **Compatibilité:** les solutions présentées ne sont pas compatibles avec le type de journal SMDR, qui est utilisé, dans la nouvelle version IPBX.
- ✓ **Complexité:** tableau de bord complexe avec une interface moins ergonomique.
- ✓ **Wintax** consomme beaucoup de ressources au sein du serveur, ce qui entraîne un gaspillage de ressources.

VII. Les objectifs du nouveau système et solution proposée

Pour répondre aux besoins de l'entreprise cliente, nous proposons de créer une application qui : facilite l'accès aux différentes configurations et consultations via un tableau de bord permettant de transformer les lignes de données reçues en informations plus significatives. Il doit également être souple et compatible avec tout environnement choisi, et gère les différentes parties. Cette application doit intégrer les frameworks Django et importer des utilisateurs via des fichiers de sauvegarde. Il doit également être extensible et facile à utiliser.

VIII. Méthodologie et formalise adoptés

Dans le but de répondre aux objectifs cités précédemment, nous sommes amenés à suivre 2TUP comme méthodologie de modélisation connue en génie logiciel tout en adoptant UML2 comme langage de modélisation. Nos choix sont justifiés dans ce qui suit.

1. Choix de processus de développement

Processus	Description	Avantages	Inconvénients
RUP	RUP est à la fois une méthodologie de conduite et de développement de projets et un outil prêt à l'emploi.	Propose des modèles de documents, et des canevas pour des projets types.	Coûteux à personnaliser Très axé processus, au détriment du développement : peu de place pour le code et la technologie.
V	Chaque phase en	Préparation des	Obligation de définir

	amont de la production du logiciel prépare la phase correspondante de vérification en aval de la production du logiciel.	phases de vérification au moment de l'Analyse et de la Conception.	la totalité des besoins au départ Validation fonctionnelle tardive.
XP XP	Tout l'accent est mis sur les bonnes pratiques de la programmation avec un déroulement par itérations courtes et gérées collectivement.	Adaptée à des projets de petites tailles et sur de petites équipes. Pousse à l'extrême les principes simples de programmation.	Difficultés lorsque la maintenance doit être faite par une autre équipe Difficultés de planification, d'estimation de charge, d'implication de l'équipe.
2TUP	S'articule autour de l'architecture	Fait une large place à la technologie et à la gestion du risque	Ne propose pas de documents types

En analysant le tableau ci-dessus nous remarquons que :

- ✓ RUP propose plutôt un cadre complet pour la conduite de projet, mais n'attache pas trop d'importance pour le développement lui-même.
- ✓ Le cycle en V quant à lui, permet de maîtriser le développement à travers les vérifications à la fin des phases, mais il n'offre toujours pas de gestion de la complexité technique.
- ✓ Quant à XP elle est beaucoup plus adaptée à des projets de petites taille et de plus des activités telles que l'estimation, la planification, s'avèrent être très difficiles à réaliser ce qui est non acceptable pour un projet tel que Printing-Platform.
- ✓ Il s'avère donc incontestablement que c'est effectivement 2TUP (Two Track Unified Process) qui est le plus approprié à notre projet :
 - Seul 2TUP propose un intérêt vif pour la gestion de la complexité technique. Or notre projet fait lui aussi large place à technologie ce qui implique l'élimination du risque technologique,
 - Son aspect itératif et incrémental au vu du temps imparti au projet et
 - Une meilleure gestion des risques.

Le 2TUP propose un cycle de développement qui sépare les aspects techniques des aspects fonctionnels (un processus en Y) et propose une étude parallèle des deux branches : fonctionnelle (étude de l'application) et la technique (étude de

l'implémentation). Illustré dans la Figure 6, le processus 2TUP s'articule autour de trois branches [N3]

- Une branche technique,
- Une branche fonctionnelle et
- Une branche de conception et réalisation.

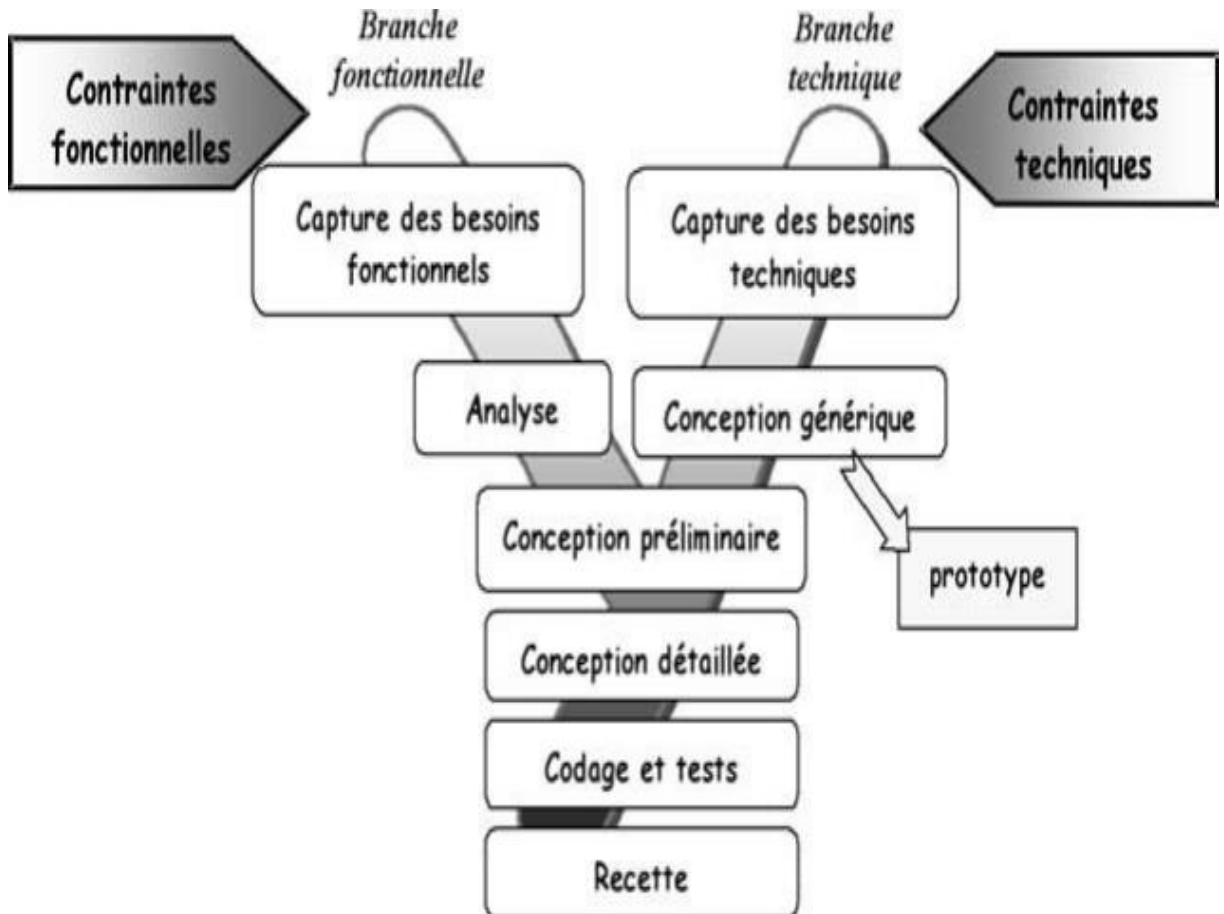


Figure 2: Contraintes fonctionnelles et techniques

Il convient aussi de noter que notre rapport sera adjacent à la méthodologie 2TUP. Nous retrouverons la plupart des phases de la figure 7.

2. Formalisme adopté :

Merise et UML sont deux grands principes de « traduction » ou modélisation d'un système. Le choix de l'un ou de l'autre se fait selon trois axes à savoir l'accessibilité, la précision et l'exploitabilité. Pour le premier axe (accessibilité) MERISE présente l'intérêt d'avoir des modèles logiques moins détaillés facilement compréhensibles. Tandis qu'UML conçu pour s'adapter à n'importe quel langage de programmation orientée objet (POO), présente plusieurs modèles (diagrammes) dont leurs compréhensions nécessitent une grande attention. En ce qui concerne le deuxième critère (précision), MERISE est moins préférable. Malgré sa clarté, il

manque une précision du fait qu'elle est éloignée du langage donc difficile à implémenter alors qu'UML intègre les éléments communs des différents langages, sa volonté est d'être fidèle à la réalisation finale. Elle est beaucoup plus complète avec ses différents diagrammes. Pour en finir avec l'exploitabilité, MERISE est une méthode plus généraliste. Elle donne une vue globale de la solution sans autant entrer dans les petits détails. Contrairement à UML qui est conçu pour l'implémentation objet avec ses différents détails et sa portabilité (s'adapte à n'importe quelle plateforme) elle est donc plus exploitable [N4].

Dans le cadre de notre étude nous avons opté pour UML qui se base sur l'approche objet. Cette dernière appréhende le système comme un ensemble d'objets qui interagissent entre eux. En conclusion, nos choix pour le développement sont résumés aux :

- Processus 2TUP comme méthodologie de développement.
- UML comme langage de modélisation.

IX. Conclusion

Dans ce premier chapitre nous avons présenté le cadre général de notre projet ainsi que la démarche nécessaire pour sa réalisation, nous avons cité encore quelques exemples d'applications développées comparable à la nôtre en dégageant leurs différents inconvénients Nous allons identifier dans le chapitre suivant les besoins fonctionnels et opérationnels du système à développer.

Chapitre II : Etude préliminaire

Introduction :

A partir de ce chapitre nous commençant à appliquer la méthodologie de conception (2TUP) que nous avons choisi d'adopter. Il est nécessaire en ce sens de commencer par définir les frontières de notre application. Cette étude nous permettra de dégager les objectifs et les besoins fonctionnels auxquels il faut répondre pour déterminer exactement ce qui est demandé, les différents acteurs et les messages émis par chacun d'eux. Nous allons préciser aussi les besoins non fonctionnels. Des diagrammes de contexte statiques et dynamiques seront utilisés..

I. Identifications des acteurs du système

1. Identification des acteurs

Dans ce paragraphe, nous nous proposons de détailler la liste des acteurs susceptibles d'interagir avec le système et ses différentes parties présentées dans la section qui précède. Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

- Le «Personnel », présente les utilisateurs qui peuvent créer, mettre à jour ou supprimer les paramètres, mettre à jour leurs valeurs de facturation, consulter et extraire des factures et consulter le tableau de bord.
- Le "Administrateur ", qui est un personnel avec les mêmes autorisations. Cependant, la différence entre eux est qu'un acteur avec un profil "administrateur " avait toutes sortes d'autorisations pour gérer l'application dans son ensemble.

II. Exigences fonctionnelles

Notre application doit remplir les fonctionnalités suivantes, classées par ordre d'importance.

- **Authentification:** Notre application comprend deux types d'acteur, à savoir « personnel » et « l'administrateur ». On utilise l'authentification pour savoir quelle permission doit être accordée.

• **Gestion de la communication IPBX:** notre application permet à l'administrateur de configurer les communications entre ses services et IPBX. Cela dépend de la façon dont l'IPBX envoie les journaux, s'il s'agit d'un SMDR (Station Messaging Detail Reporting), donc il nécessite l'adresse IP et le port utilisé, même si l'administrateur doit insérer l'adresse IP, le port utilisé, le Nom d'utilisateur et mot de passe IPBX, et le chemin CDR.

Dans le second cas, les champs Port et Chemin CDR ont une valeur par défaut permettant la communication avec le matériel AVAYA.

• **Réception des journaux:** la réception est effectuée par un service configuré pour démarrer au démarrage du système et commence à écouter sur le port configuré. Les lignes reçues seront scindées et les informations importantes seront sauvegardées dans la base de données. La facturation sera effectuée immédiatement après l'extraction de la durée de l'appel et de l'appelant.

• **Gestion des factures:** notre application permet à l'administrateur de suivre la facturation par les utilisateurs, le stand, le service et les lignes. Il peut toujours extraire la facture au format pdf et il peut personnaliser la facture et extraire d'autres détails comme la consommation du service par stand ou par sous-service. La facture peut être réinitialisée. Une date de réinitialisation et une date d'annulation sont sauvegardées à chaque modification. Seul le super utilisateur peut réinitialiser la facture personnalisée et il peut toujours réinitialiser toutes les factures en passant par la vérification du mot de passe.

• **Consultation des logs:** notre application affiche dans le tableau de bord la liste des logs reçus triés par date d'appel, elle permet à l'utilisateur de réorganiser la liste en utilisant l'entête du tableau et filtrée en utilisant les entrées du tableau.

• **Booth management:** notre application permet à l'administrateur de gérer le numéro de stand. Il lui permet d'exporter et d'importer un stand via un fichier csv ce qui facilite la sauvegarde et le processus de restauration.

• **Gestion de l'utilisateur:** notre application permet à l'administrateur de gérer les utilisateurs. Il lui permet d'exporter et d'importer un stand via un fichier csv ce qui facilite la sauvegarde et le processus de restauration.

- **Gestion des services:** notre application permet à l'administrateur de gérer les services, elle lui permet d'exporter et d'importer un stand via un fichier csv ce qui rend la sauvegarde et le processus de restauration plus faciles.
- **Gestion des notifications:** notre application permet à l'administrateur de gérer les notifications, elle lui permet d'exporter et d'importer un stand via un fichier csv ce qui rend la sauvegarde et le processus de restauration plus faciles.
- **Gestion des méthodes de facturation:** notre application permet à l'administrateur de gérer les méthodes de facturation, elle lui permet d'exporter et d'importer le stand via le fichier csv ce qui facilite la sauvegarde et la restauration.
- **Gestion de ligne:** notre application permet à l'administrateur de gérer les lignes, elle lui permet d'exporter et d'importer des stands via un fichier csv ce qui rend la sauvegarde et le processus de restauration plus faciles.
- **Gestion des codes de pays:** notre application permet à un super utilisateur d'importer, d'exporter, de consulter et de modifier le code de pays, et à un seul utilisateur de consulter et de modifier uniquement le code de pays.
- **Gestion de l'administrateur:** notre application permet à un super utilisateur d'ajouter, de supprimer, de consulter la liste des administrateurs et de consulter et modifier ses propres informations, et permet aussi à un seul utilisateur de simplement consulter et modifier ses propres informations.

III. Besoins non fonctionnels :

Les besoins non fonctionnels représentent les besoins qui caractérisent le système. Ce sont des besoins exprimés en termes de performance, de type de matériel ou le type de conception. Ces besoins doivent être pris en compte tout au long du développement de l'application. Les besoins non fonctionnels relatifs à notre système sont les suivants :

- **Ergonomie des interfaces :** Aucune application ne connaîtra le succès si elle n'est pas également facile à utiliser. Il faut donc minimiser la densité d'éléments sur les écrans, bien choisir la disposition et le flux et bien choisir les couleurs des interfaces. Cela garantira l'obtention de l'information avec un minimum d'efforts par l'utilisateur et avec moins de risque d'erreur.

- **La Sécurité** : Lors de sa demande de connexion à l'application « BILLPRO » chaque utilisateur doit être reconnu du système par un login (@mail ou pseudo) et un mot de passe; Les fonctionnalités offertes à chaque utilisateur doivent être restreintes à celles qui lui sont autorisées.

L'application doit garantir aussi à tous ses utilisateurs connectés l'intégrité et la confidentialité de leurs données.

- **La réutilisabilité** : L'application doit garantir la réutilisabilité de composantes, du code, du design et même d'exigences dans d'autres systèmes
- **Performance** : L'application doit être rapide en termes de temps de réponse, de temps de chargement des pages, d'ouverture d'écran et de délais de rafraîchissement.
- **Disponibilité** : Le système doit être présent en temps réel aux utilisateurs pour que chacun puisse appliquer la tâche souhaitée ou demandée.
- **Modificabilité**: L'application doit permettre l'ajout de nouvelles fonctionnalités.
- **Multi-utilisateur** : L'application doit être accessible simultanément par un nombre important d'utilisateurs sans régression notable des performances.
- **La maintenance** : Les différents modules de l'application doivent être faciles à maintenir, par conséquent, le code doit être lisible et bien structuré.
- **La consistance** : L'application doit toujours donner des résultats corrects à l'utilisateur en se référant au contexte de l'application.

Pour récapituler, notre application doit être facile à utiliser, avec une bonne ergonomie, elle doit garantir un temps de réponse court, et elle doit être paramétrable.

IV. Conclusion

Ce chapitre a fait l'objet de l'étude préliminaire du projet à développer. Dans le quelle nous avons détaillé les besoins fonctionnels et non fonctionnels de l'application.

Chapitre III : Capture des besoins fonctionnels

I. Introduction

Ce chapitre sera consacré à la modélisation des besoins fonctionnels qui représente la première étape de la branche fonctionnelle du processus de développement 2TUP. Elle permet de compléter le recueil initial des besoins effectués pendant l'étude préliminaire. Pour ce faire nous utiliserons la notion de Cas d'utilisation. Chaque Cas d'utilisation sera identifié, décrit, organisé et classé en fonction de son importance dans le projet.

II. Diagramme de cas d'utilisation de l'application "BILLPRO"

Notre système présente deux types d'acteurs, "Administrateur" et "Utilisateur simple", qui héritent tous deux des mêmes autorisations qu'un "Personnel" avec plus de fonctionnalités accordées pour le L'administrateur.

En tant que "Personnel", l'acteur peut consulter les journaux et les factures. Cet utilisateur peut gérer tous les paramètres, mettre à jour leurs valeurs ou les supprimer.

«L'administrateur» bénéficie d'une autorisation supplémentaire : ajout et suppression des utilisateurs. Il peut également gérer les factures en les réinitialisant.

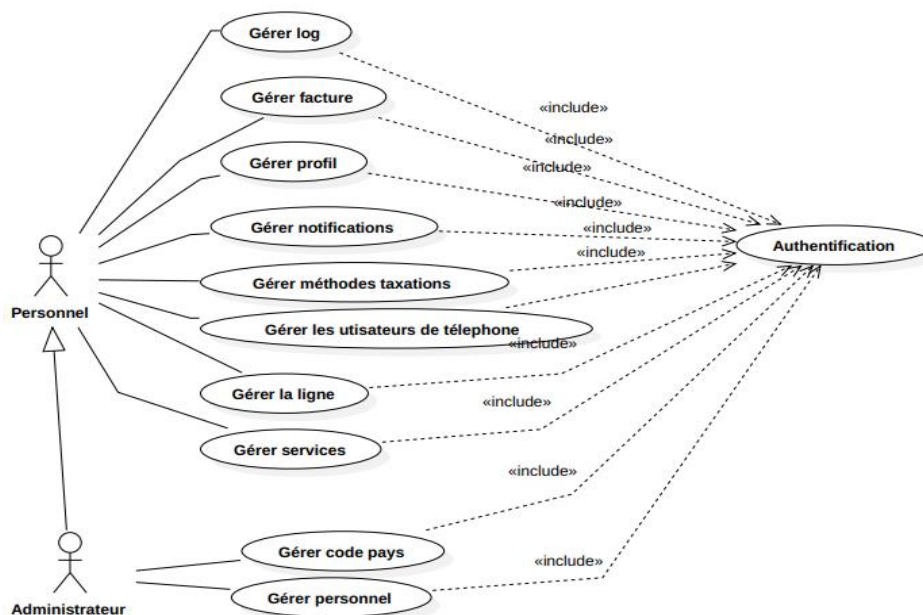


Figure 3: Diagramme de cas d'utilisation générale

III. Raffinement et description détaillée des cas d'utilisation

- ❖ Raffinement pour le cas d'utilisation " gérer journal"

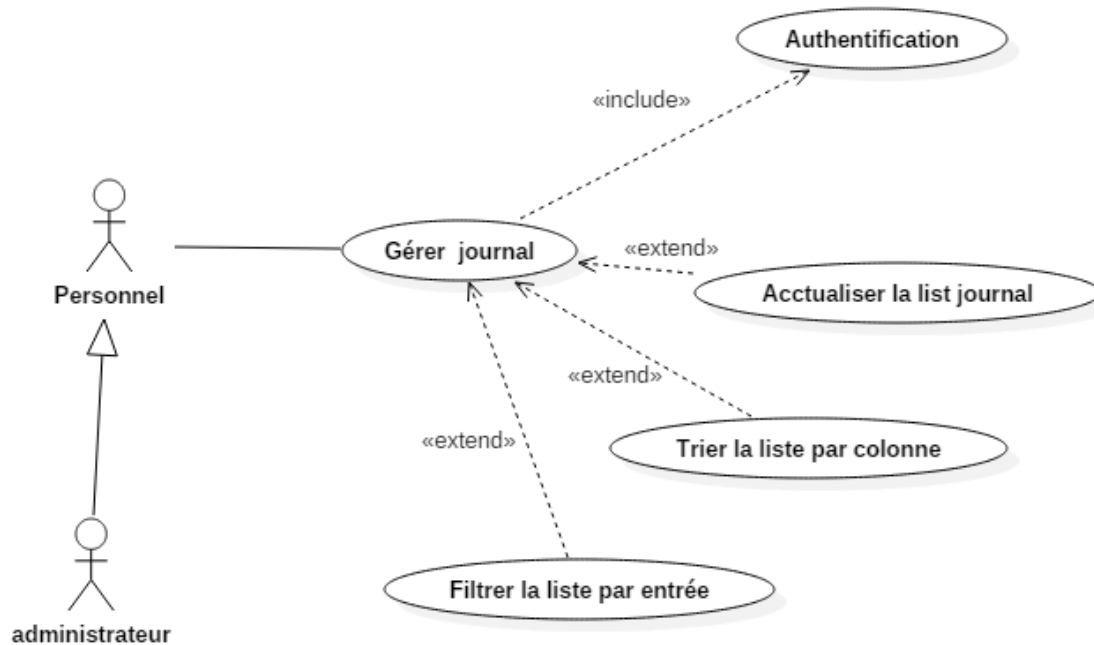


Figure 4 : Raffinement pour le cas d'utilisation " gérer journal"

Etape	Description
Abstract	Acteur : Administrateur, Personnel Titre: Gérer journal Description: La gestion de journal représente le tableau de bord de l'application. Elle contient le journal des appels.
Précondition	<ul style="list-style-type: none"> • L'administrateur doit être authentifié. • Il doit accéder à l'interface du tableau de bord
Scénario nominale	<ol style="list-style-type: none"> 1. Consulter la liste journal 2. Actualiser la list journal 3. Trier la liste par colonne 4. Filtrer la liste par entrée
Scenario alternative	----
Post-condition	<ul style="list-style-type: none"> • Tableau de bord actualisé avec succès • Entrée triée avec succès • Entrée filtrée avec succès

❖ Raffinement pour le cas d'utilisation “gérer facture”

Le processus de facturation est effectué par un autre service en arrière-plan pour séparer l'imposition du processus de gestion en cas de défaillance Gérer les factures, y compris: trier les factures, consulter les factures, vérifier les statistiques, il peut également extraire les factures au format PDF.

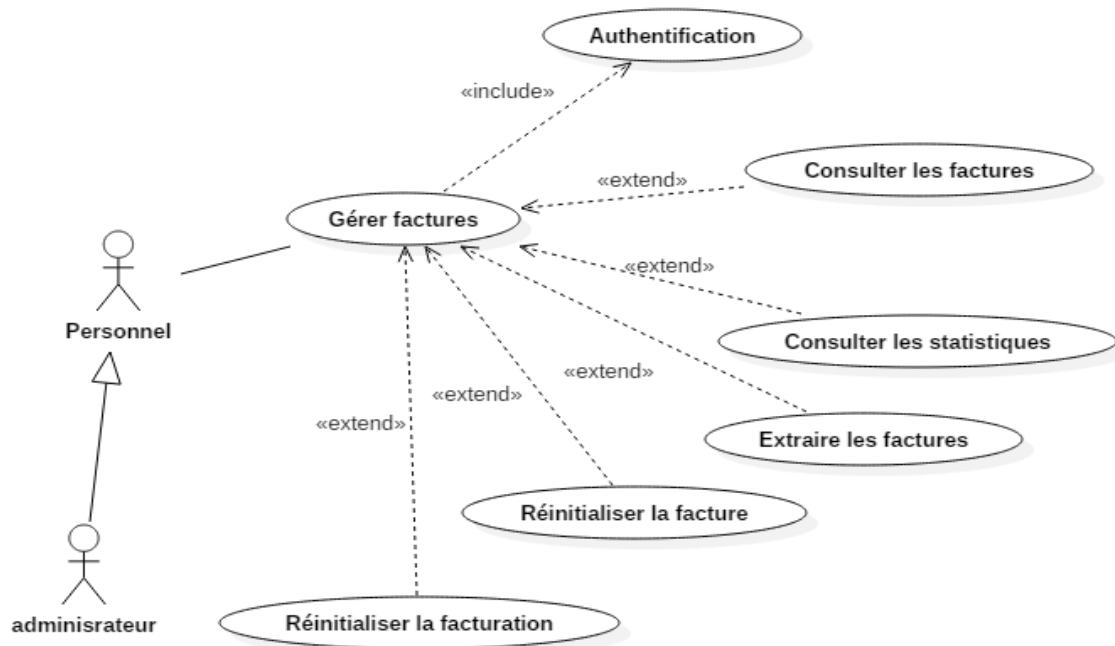


Figure 5 : Raffinement pour le cas d'utilisation “ gérer Facture”

Step	Description
Abstract	Acteur: Administrateur, Personnel Titre: Gérer factures Description: la gestion des factures permet à l'administrateur de contrôler la consommation.
Précondition	<ul style="list-style-type: none"> • L'administrateur doit être authentifié. • Il doit accéder à l'interface de consommation
Scénario nominale	<ol style="list-style-type: none"> 1. Consulter les factures 2. Consulter les statistiques 3. Trier les factures 4. Extraire les factures 5. Réinitialiser la facture 6. Réinitialiser la facturation

Scenario alternative	
Post-condition	<ul style="list-style-type: none"> • Factures triées • Les factures extraites • Bills réinitialisés

❖ Refinement pour le cas d'utilisation “gérer les méthode de fiscalité”

L'administrateur doit être capable de gérer les méthodes d'imposition en ajoutant, en supprimant, en consultant et en modifiant. Comme le montre la figure (6) ci-dessous, l'administrateur peut exporter et importer une liste de méthodes via un fichier au format .csv qui n'est pas autorisé par le personnel. Nous considérons les méthodes d'imposition comme les parties les plus importantes dans l'application. De plus, pour les protéger de la suppression par faute, nous avons fait une méthode de suppression sécurisée, permettant à l'administrateur de remplacer celui supprimé par un autre. La méthode d'export est utilisée pour créer un fichier de protection contenant une liste des méthodes de taxation avec son identifiant pour faciliter la réinitialisation des paramètres d'application en cas de défaut. La méthode import effectue une vérification sur chaque ligne du fichier .csv, puis l'insère en toute sécurité dans la base de données.

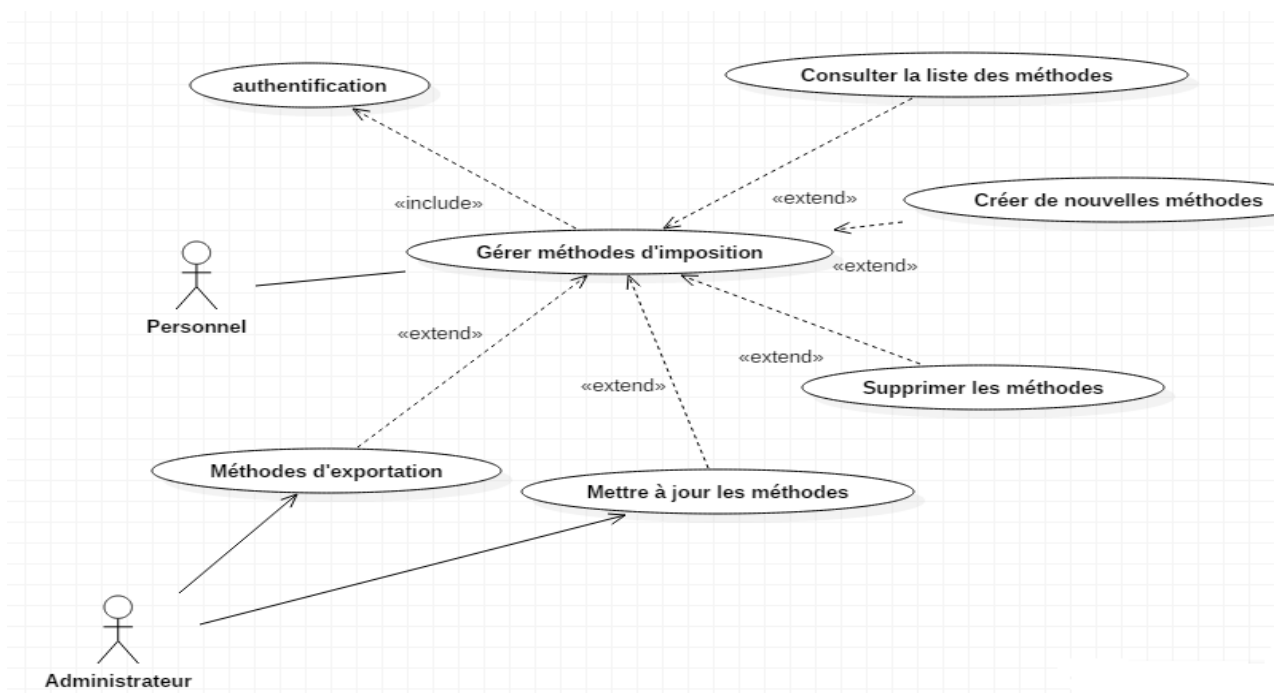


Figure 6: Raffinement pour le cas d'utilisation “gérer méthode d'imposition”

Etape	Description
Abstract	Acteur: Administrateur, Personnel Titre: gérer les méthodes d'imposition Description: il permet à l'administrateur de gérer les différentes méthodes d'imposition utilisées lors de la facturation.
Précondition	L'administrateur doit être authentifié. Il doit accéder à l'interface de gestion des méthodes d'imposition
Scénario nominale	<ol style="list-style-type: none"> 1) Consulter la liste des méthodes 2) Créer de nouvelles méthodes 3) Supprimer les méthodes 4) Méthodes de tri 5) Mettre à jour les méthodes 6) Méthodes d'exportation 7) Méthodes d'importation
Scenarion alternative	<ul style="list-style-type: none"> - L'opération a échoué - Entrée existante - Entrée utilisée
Post-condition	Succès de l'opération

❖ Diagramme de gestion de cabine téléphonique étendu

Le responsable de la cabine téléphonique doit permettre à l'administrateur de gérer complètement le téléphone via son interface. La gestion du stand téléphonique est la deuxième partie la plus importante de notre application, elle doit être insérée dans une phase finale, après avoir commencé les autres parties des applications, car elle est basée sur d'autres informations (méthodes de taxation, l'utilisateur et le service) utilisé sur le processus de facturation. Les méthodes d'importation et d'exportation sont également incluses pour faciliter la réinitialisation des informations de la cabine téléphonique dans le même ordre après vérification de la redondance:

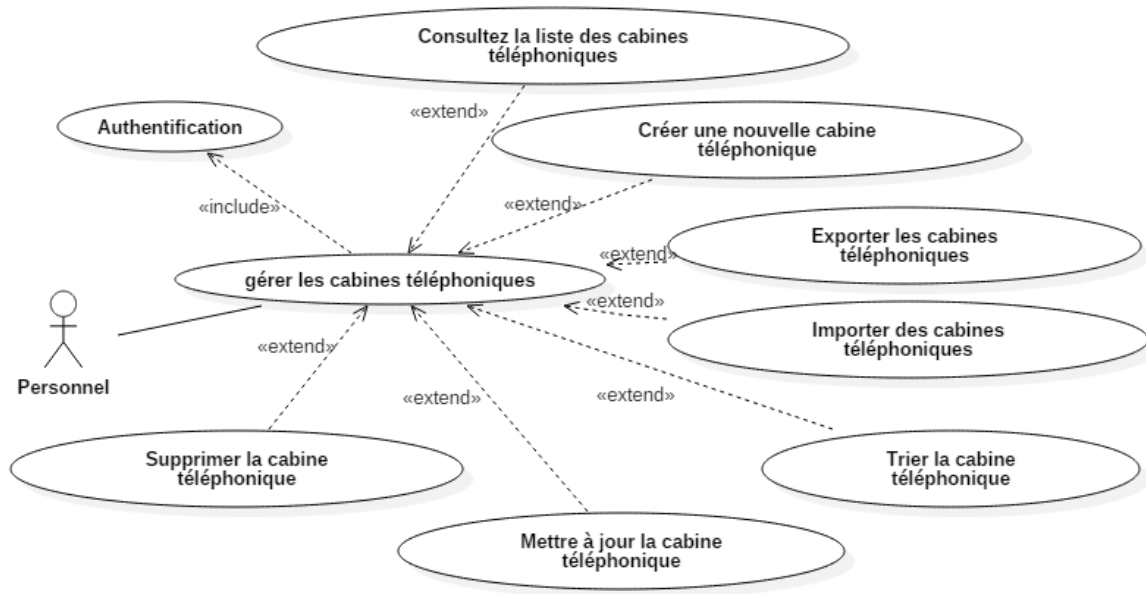


Figure 7: Diagramme de gestion de cabine téléphonique étendu

Étape	Description
Abstract	Acteur : Administrateur, personnel Titre: gérer les cabines téléphoniques Description: cela permet à l'administrateur de gérer les différentes cabines téléphoniques
Précondition	L'administrateur doit être authentifié. Il doit accéder à l'interface de gestion des cabines téléphoniques
Scénario nominale	1) Consultez la liste des cabines téléphoniques 2) Créer une nouvelle cabine téléphonique 3) Supprimer la cabine téléphonique 4) Trier la cabine téléphonique 5) Mettre à jour la cabine téléphonique 6) Exporter les cabines téléphoniques 7) Importer des cabines téléphoniques
Scenario alternative	- L'opération a échoué - Entrée existante - Entrée utilisée
Post-condition	- Succès de l'opération

❖ Refinement pour le cas d'utilisation "Gestion étendue des utilisateurs de téléphone"

La gestion étendue des utilisateurs permet à l'administrateur de gérer le nombre d'utilisateurs.

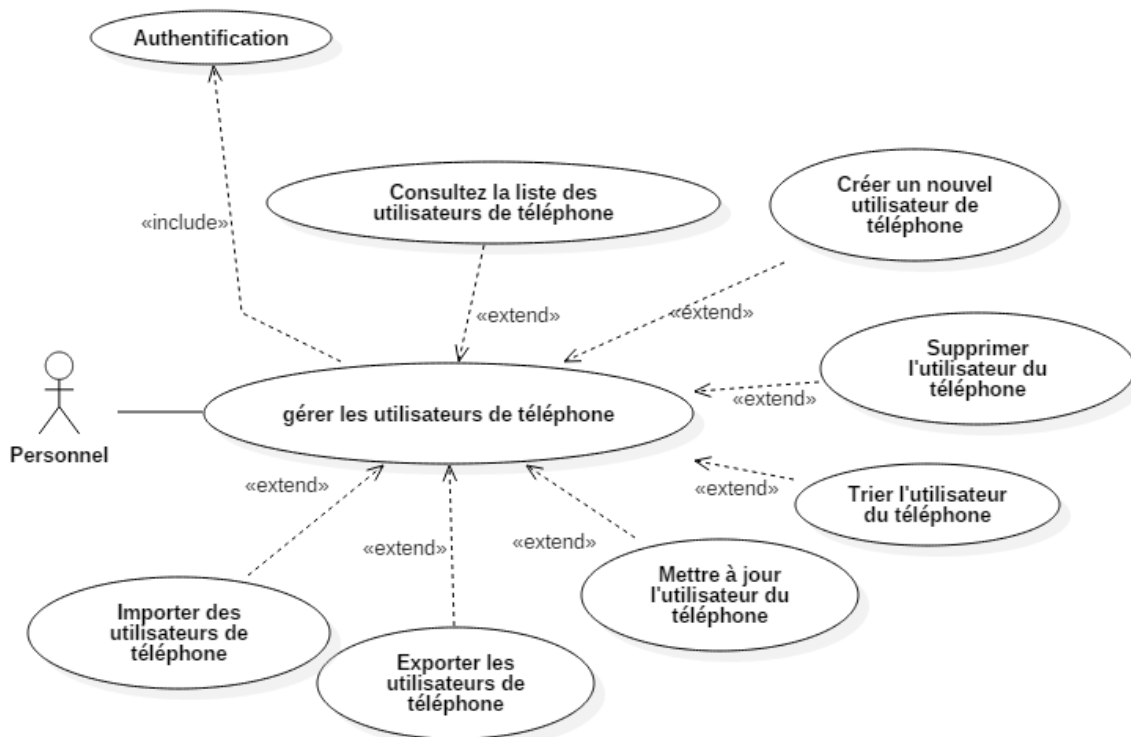


Figure 8: Gestion étendue des utilisateurs de téléphone

Etape	Description
Abstract	Acteur : Administrateur Titre : Gestion étendue des utilisateurs de téléphone Description: permet à l'administrateur de gérer les utilisateurs de l'application
Précondition	L'administrateur doit être authentifié. Il doit accéder à l'interface de gestion de l'administrateur
Scénario nominale	1) Consulter la liste des administrateurs 2) Créer un nouvel utilisateur de téléphone 3) Supprimer l'utilisateur du téléphone 4) Trier l'utilisateur du téléphone
Scenario alternative	- L'opération a échoué - Entrée existante
Post-condition	- Succès de l'opération

IV. Conclusion

Tout au long de ce chapitre, nous avons analysé et clarifié les exigences que notre application devrait fournir au futur utilisateur, et nous avons donné le cas d'utilisation et le raffinement de la partie la plus importante de notre projet. Le chapitre suivant vise à aller plus loin dans le processus de développement de l'application via la présentation du design de l'application

Chapitre IV : Capture des besoins Techniques

I. Introduction

Cette partie traite de manière générale la capture des besoins techniques. Cette phase sert de complément à la capture des besoins fonctionnels. L'idée vectrice à ce niveau tourne autour de la relève des différentes contraintes qui ne sont ni descriptives du métier des utilisateurs, ni descriptives d'un point de vue applicatif ou d'un quelconque processus métier. Nous pouvons résumer cette partie à travers les questions qui vont suivre :

- ✓ Quel sont les moyens technologiques dont nous disposons pour afin de mettre en œuvre l'application ?
- ✓ Avons-nous suffisamment de bagages techniques afin d'assurer la totalité des besoins fonctionnels énoncés ?

II. Spécification de l'architecture système

Tout d'abord il nous faut indiquer notre choix du style d'architecture à adopter pour notre application. Nous allons privilégier l'architecture en niveaux. Cela s'explique de faite que la configuration géographique du système, les prérequis techniques, la configuration matérielle ainsi que le nombre illimité de postes favorisent le choix de développement d'une solution client/serveur à trois niveaux Cette architecture (à trois niveaux) représente les avantages suivants :

- ✓ Une plus grande flexibilité/souplesse,
- ✓ Une sécurité accrue car la sécurité peut être définie indépendamment pour chaque service, et à chaque niveau et
- ✓ De meilleures performances, étant donné le partage des tâches entre les différents serveurs. Ainsi l'architecture logique du système est divisée en trois

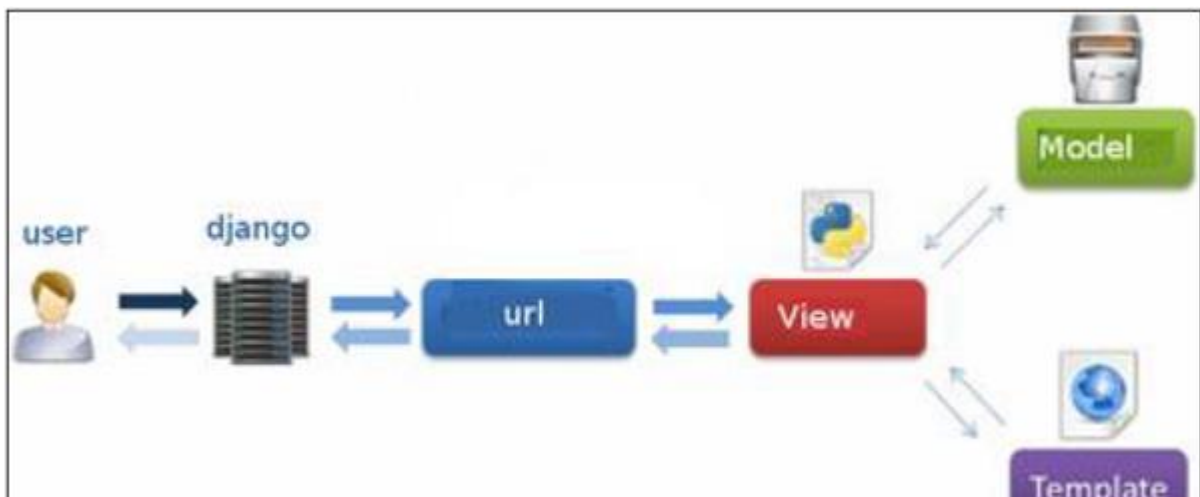


Figure 9:MVT modèle

Django supporte le pattern MVC.

➤ **MVC Pattern**

Lorsque nous parlons d'applications qui fournissent une interface utilisateur (Web ou de bureau), nous parlons généralement de l'architecture MVC. Et comme son nom l'indique, le pattern MVC est basé sur trois composants: Model, View et Controller.

➤ **DJANGO MVC - MVT Pattern**

Le Model-View-Template (MVT) est légèrement différent de MVC. En fait, la principale différence entre les deux modèles est que Django s'occupe lui-même de la partie Contrôleur (Code Logiciel qui contrôle les interactions entre le Modèle et la Vue), nous laissant avec le modèle. Le modèle est un fichier HTML mélangé avec Django Template Language (DTL).

Le diagramme suivant illustre comment chaque composant du modèle MVT interagit les uns avec les autres pour répondre à une requête de l'utilisateur développeur fournit le modèle, la vue et le modèle puis le mappe juste à une URL et Django fait la magie pour le servir à l'utilisateur. [N5]

1. Environnement de développement

a. Environnement logiciels

Nous allons dans ce qui suit présenter les différents outils et les divers langages pour la réalisation de notre application

➤ **Django (version 1.8):**

Django est un framework Web Python de haut niveau qui encourage un développement rapide et un design propre et pragmatique. Construit par des développeurs expérimentés, il prend en charge une grande partie des tracas du développement Web, de sorte que vous pouvez vous concentrer sur l'écriture de votre application sans avoir besoin de réinventer la roue. C'est gratuit et open source. [N6]



Figure 10:Django logo

Pour installer Django, nous avons utilisé les commandes suivantes:

```
Apt install python-pip  
Pip install django==1.8
```

➤ **Vi editor:**



Figure 11:MySQL logo

Pour installer MySQL-module, nous avons tapé la commande ci-dessous:

```
Apt install mysql-server  
apt install python-MySQLdb
```

Les autres programmes

➤ **Paramiko**

Paramiko est une implémentation Python (2.7, 3.4+) du protocole SSHv2 [1], fournissant à la fois des fonctionnalités client et serveur. Bien qu'il exploite une extension Python C pour la cryptographie de bas niveau (cryptographie), Paramiko lui-même est une pure interface Python autour des concepts de mise en réseau SSH. [N7]

```
Pip install cryptography  
Pip install paramiko
```

➤ **Python-ReportLab**

Ceci est une bibliothèque de logiciels qui nous permet de créer directement des documents en format PDF (Portable Document Format) en utilisant le langage de programmation Python. Il crée également des graphiques et des graphiques de données dans divers formats bitmap et vectoriels ainsi que PDF.

La bibliothèque ReportLab crée directement un PDF basé sur nos commandes graphiques. Il n'y a pas d'étapes intermédiaires. Nos applications peuvent générer des rapports extrêmement rapidement - parfois des ordres de grandeur plus rapidement que les outils traditionnels d'écriture de rapports.

```
Apt install python-reportlab
```

➤ **Django_chartit**

django Chartit est une application Django qui peut être utilisée pour créer facilement des graphiques à partir des données de votre base de données. Les graphiques sont rendus à l'aide des bibliothèques Highcharts et jQuery JavaScript. Les données de notre base de données peuvent être tracées sous forme de graphiques linéaires simples, de graphiques à colonnes, de graphiques à aires, de diagrammes de dispersion et de nombreux autres types de graphiques. Les données peuvent également être tracées en tant que graphiques croisés dynamiques où les données sont groupées et / ou pivotées par des colonnes spécifiques. Pour ajouter cette application, nous avons utilisé ces lignes de commande:

```
Pip install simplejson  
Pip install Django chartit
```

b. Langage de programmation

➤ **Python**

Python est un langage de programmation objet, multi-paradigme et multiplateformes. Il favorise la impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-

miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

Le langage Python est placé sous une licence libre proche de la licence BSD et fonctionne sur la plupart des plates-formes informatiques, des supercalculateurs aux ordinateurs centraux, de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et aussi avec Java ou encore .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser



Figure 12:Python logo

➤ **ORM Django**

En informatique, ORM Django est une technique de programmation permettant de convertir des données entre des systèmes de types incompatibles dans des langages de programmation orientés objet. Cela crée en effet une «base de données d'objets virtuelle» qui peut être utilisée depuis le langage de programmation. Les systèmes ORM (Object Relational Mapping) interagissent beaucoup plus facilement avec une base de données SQL, mais ils ont la réputation d'être inefficaces et plus lents que le SQL

➤ **HTML5 (HTML, CSS, JavaScript, jQuery, highcharts, Bootstrap)**

Conclusion : Au cours de ce chapitre nous avons présenté les différentes technologies et Framework utilisés avec des explications claires à propos de leur utilisation au niveau de notre projet. Enfin, nous avons détaillé les besoins techniques en présentant l'environnement matériel et logiciel du système pour se préparer à la prochaine phase celle d'analyse et de conception.

III. Conclusion :

Au cours de ce chapitre nous avons présenté les différentes technologies et Framework utilisés avec des explications claires à propos de leur utilisation au niveau de notre projet. Enfin, nous avons détaillé les besoins techniques en présentant l'environnement matériel et logiciel du système pour se préparé à la prochaine phase celle d'analyse et de conception.

Chapitre V: Analyse et Conception

I. Introduction

Après avoir identifié les besoins fonctionnels et techniques de notre système, le présent chapitre détaille la phase d'analyse et de conception

II. Architecture globale

1. L'architecture de General BILLPRO

La figure ci-dessous présente l'architecture globale de notre application et les différents modules qui la composent.

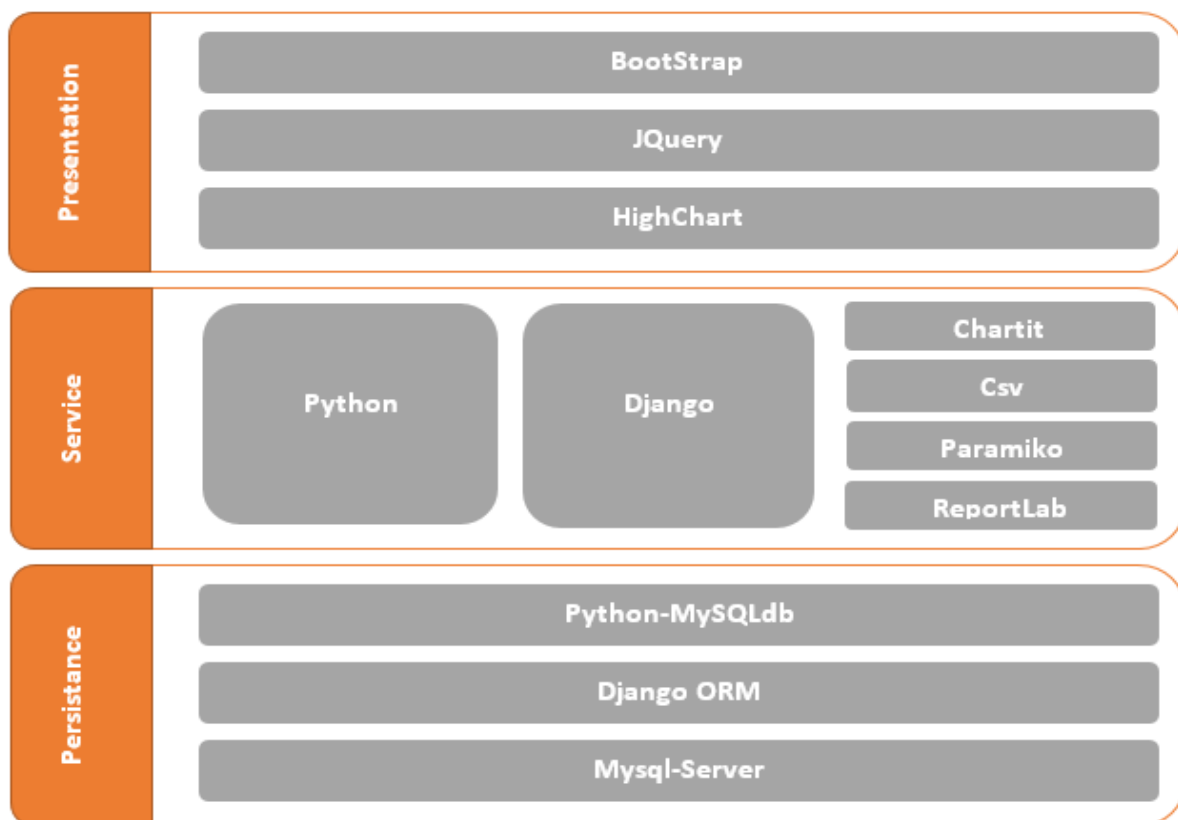


Figure 13: Architecture générale de l'application

Le diagramme d'architecture montre les différentes couches de notre application. La couche présentation est composée d'une liste de « framework » utilisée lors du développement des templates. La couche « service » contient la structure et le module utilisés dans le cœur de l'application, et enfin, la couche persistance dans laquelle nous définissons les modules liant notre application à la base de données.

2. Diagrammes de déploiement

Le diagramme de déploiement spécifie un ensemble de constructions pouvant être utilisées pour définir les systèmes d'architecture d'exécution qui représentent l'affectation d'artefacts logiciels aux nœuds. Les nœuds sont connectés via des chemins de communication pour créer des systèmes de réseau de toute complexité. Les nœuds sont généralement définis de manière imbriquée et représentent des périphériques matériels ou des environnements d'exécution logicielle. Les artefacts représentent des éléments concrets dans le monde physique, ils sont le résultat d'un processus de développement.

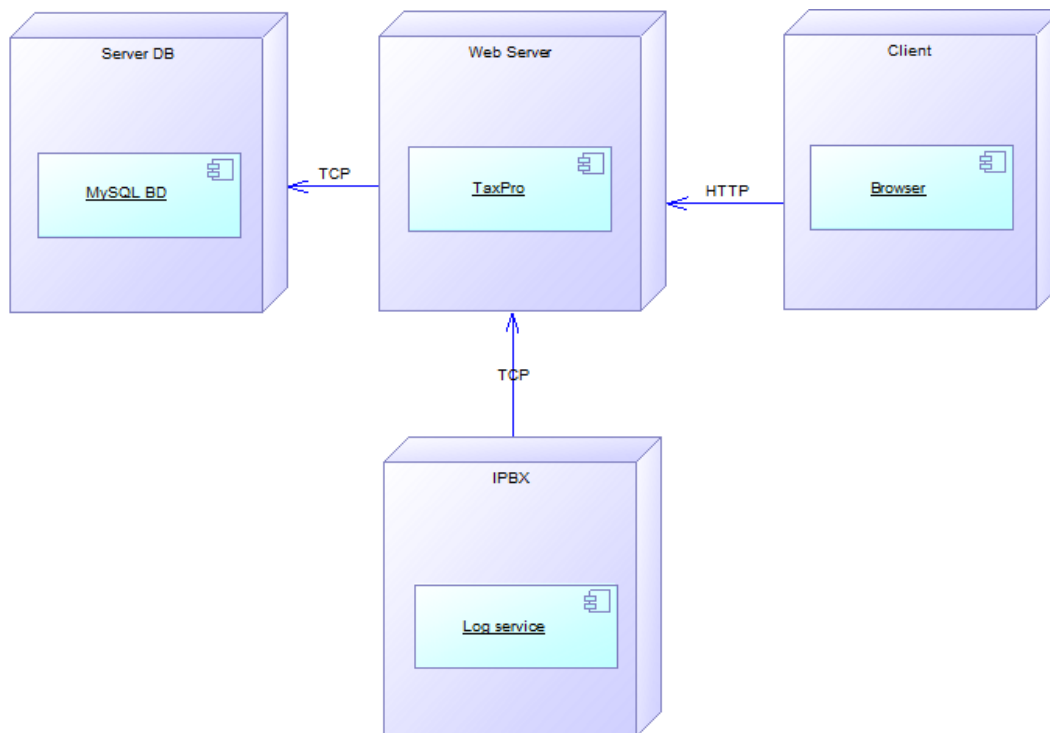


Figure 14: diagrammes de déploiement

La division du module de notre application nous permet de contrôler la complexité du système et de faciliter la compréhension. Les paquets correspondant aux différentes couches, qui interviennent sur le fonctionnement de l'opération BILLPRO. Nous identifierons le paquet différent et le décrirons. La division de BILLPRO met en évidence trois couches principales présentées ci-dessous:

Tableau 1: « BILLPro » description de catégorie

Couche de présentation	Ce paquet contient les modèles et les balises de modèle de notre application. Il contient également le CSS et les frameworks JS Bootstrap et jQuery.
Couche de service	Ce paquet contient les vues et les urls, qui sont lié aux Template.
Calque de modèle	Ce paquet contient la configuration des formulaires et les modèles.

2. Diagramme de paquet

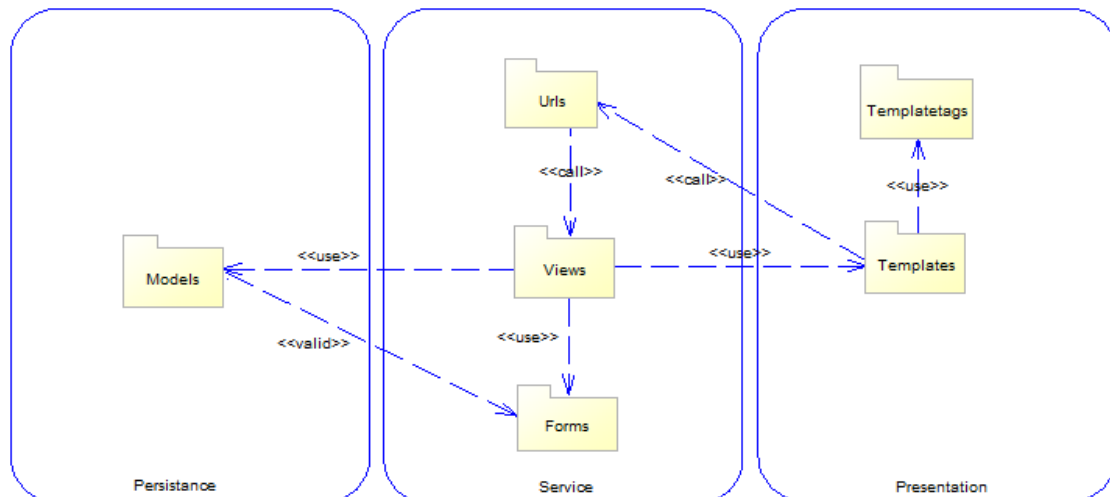


Figure 15:diagramme de paquet

À partir du diagramme de la figure (15), nous notons la séparation entre les 3 couches d'application : « présentation, service and persistence ». Nous présentons les différents packages existants dans la solution :

- **Template**: ce package regroupe tous les Template web utilisés dans l'application
- **Template tags** : ce paquet contient les balises utilisées dans les gabarits
- **Views** : il comprend les différentes vues avec l'affichage des modèles,
- **Urls**: ce paquet lie l'IHM aux vues,

- Forms :il clarifie la structure du modèle pour faciliter le processus de validation des demandes sur les vues,
- Modèles: ce paquet contient les différents modèles utilisés dans les applications

3. Conception of BillPro

a. Développement du modèle statique

➤ Diagramme de classe

Smcdr ;

La table de reporting de la messagerie et des détails de l'appel contient la liste des logs SMDR et CDR reçus de l'IPBX, ses attributs sont "id_log" qui sert à trier le résultat listé, "date_ap" regroupe la date et l'heure d'appel, "duree" Contient la durée de l'appel, "appelant" et "appele" contient l'appelant et le numéro appelé, "direct" reçoit la direction de l'appel (Entrée, Sortie ou interne), et finalement "ligne" contient la ligne utilisée en cas d'appel externe .

Operator : Il contient la totalité de la ligne externe liée à l'entreprise et leurs identifiants dans la configuration IPBX. Ses attributs sont "id_op" utilisé comme identifiant de ligne dans la table Operator et "nom_op" comme identifiant de ligne pour l'administrateur, "prix_op" qui contient le coût d'appel par minute, "num_op" et "pref_op" contiennent le numéro de la ligne liée à l'opérateur et à l'IPBX, et "desc_op" décrit la ligne.

IPBX Cette table conserve la configuration utilisée pour se connecter à l'IPBX. Ses attributs sont les identifiants "id_pbx", "type_pbx" qui conserve le type d'entrée de journalisation (SMDR et CDR), "user_os", "passwd", "ip_adr", "cdr_path" utilisé dans le cas de la connexion CDR et "ip_adr", "Port" utilisé dans le cas d'une connexion SMDR, et desc_pbx décrit les détails de la connexion.

- **Application manager :**

Serv : Cette table conserve la liste des services, départements et directions de l'entreprise. Ses attributs sont "id_serv" en tant qu'identifiant, "nom_serv" contient le nom du service et desc_serv en tant que champ de description.

Relation : Cette table est utilisée pour décrire l'architecture hiérarchique de l'entreprise détenant le service inférieur "ser_inf" au début de sa hiérarchie supérieure "ser_sup".

Poste : Cette table contient la liste du numéro de téléphone utilisé dans le réseau interne VoIP. Ses champs sont comme d'habitude l'identifiant "id_poste", le numéro de la station est contenu dans "num_poste", "serv" et "meth" comme clés étrangères et contenant le service associé et la méthode utilisée dans la taxation, et enfin, le "desc_poste" décrivant la station

Taxation : la table des méthodes d'imposition contient les méthodes utilisées pour la taxation avec des détails comme le nom de la méthode "meth_tax", le prix "prix_tax" et l'augmentation d'impôt "maj_tax" et une description dans le champ "desc_tax".

Prefixe : cette table contient la liste du code de pays international "prefixe" et son pays "pays" suivis de leur prix d'appel "prix_pref" et d'une description "desc_pref".

Utilisateur : Cette table conserve les informations relatives aux utilisateurs de la station, ces utilisateurs sont identifiés par un "id_user", les autres informations sont généralement des informations de contact, telles que "nom_utilisateur", "tel_1", "email" et "adresse".

User : Cette table contient la liste des administrateurs qui différencient le super-utilisateur et le personnel des champs "is_superuser" et "is_staff", et associe un nom d'utilisateur "login" et un mot de passe "password", identifié par "first_name", "last_name" et "email". Il y a également la date de création et les derniers champs de connexion pour consigner l'accès administrateur.

Notif : la table de notification maintient la liste des notifications personnalisables par l'administrateur. Cette table contient l'identifiant "id_notif" suivi de 3 champs; le nom "nom_notif", "rate" et la description "desc_notif".

- Facturation table:
- Hist_cons, hist_poste, hist_serv, hist_line: ces tables conservent la facture des différents composants de l'entreprise.

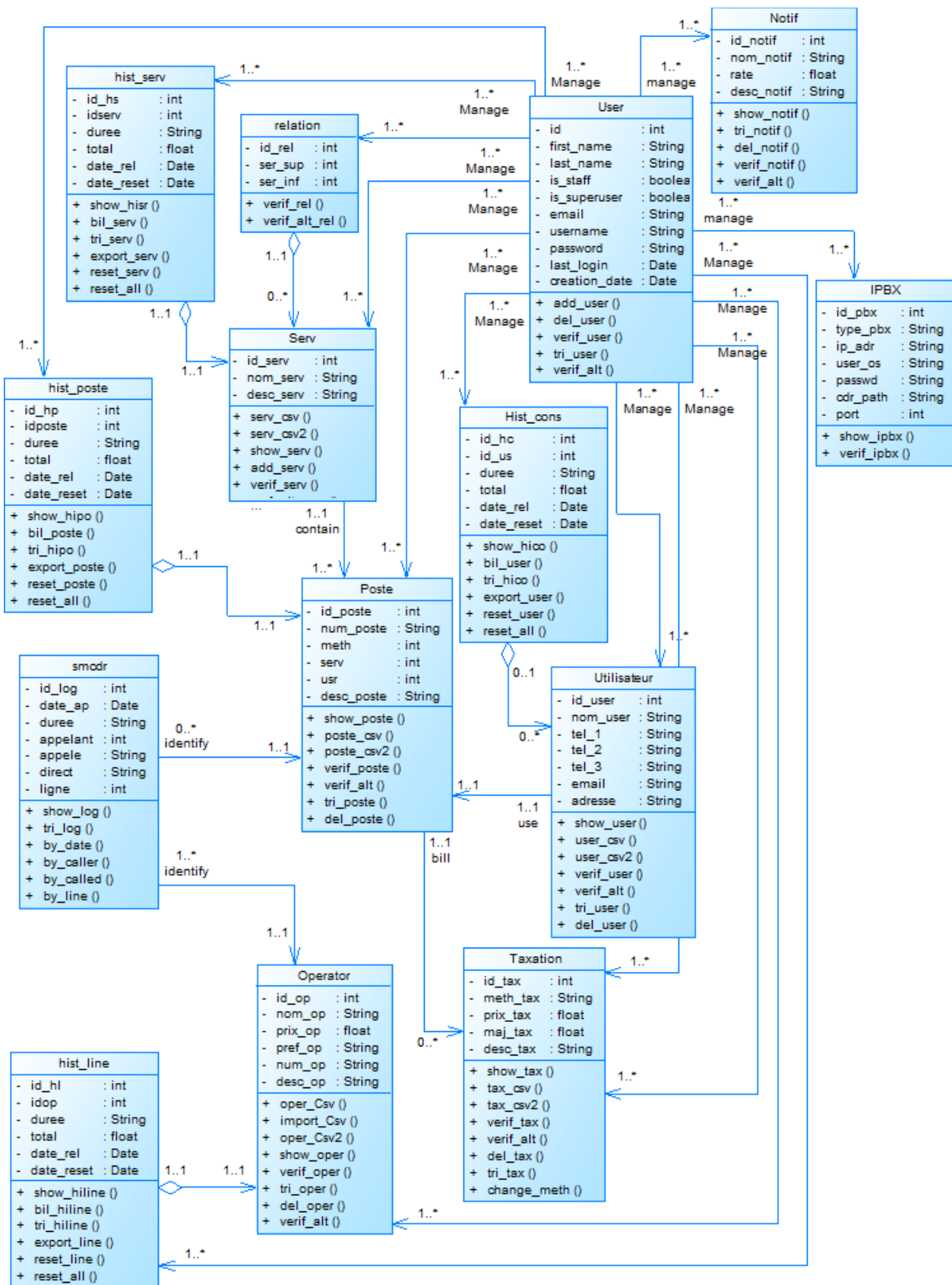


Figure 16 : diagramme de class

III. Diagramme de séquence

Dans cette partie, nous allons décrire le scénario le plus important de notre application pour donner un aperçu des détails d'exécution de la tâche différente.

- Créer des méthodes de facturation

Ce diagramme de séquence illustre le processus de création de méthodes d'facturation. . Lorsque l'utilisateur de l'application tente de créer une nouvelle méthode, il doit accéder à la page de création qui contient le formulaire de création, chaque requête doit passer par le fichier urls.py qui le redirige vers la vue correspondante, après quoi la vue valide la requête puis le traite en insérant la nouvelle valeur dans la base de données et envoie la réponse à l'utilisateur.

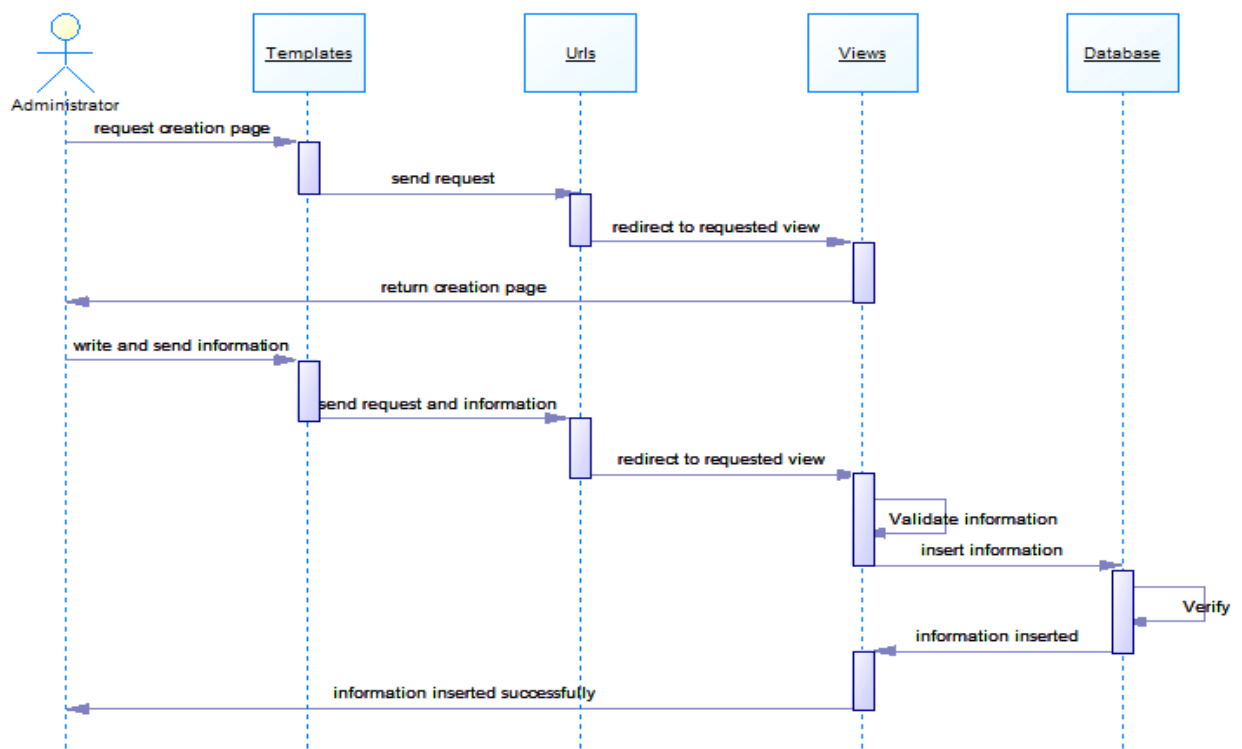


Figure 17:Créer des méthodes de facturation

- Exporter les méthodes de facturation

Le processus d'exportation le rend plus facile que toute autre sauvegarde, il insère toutes les méthodes existantes dans un fichier csv et l'envoie en réponse.

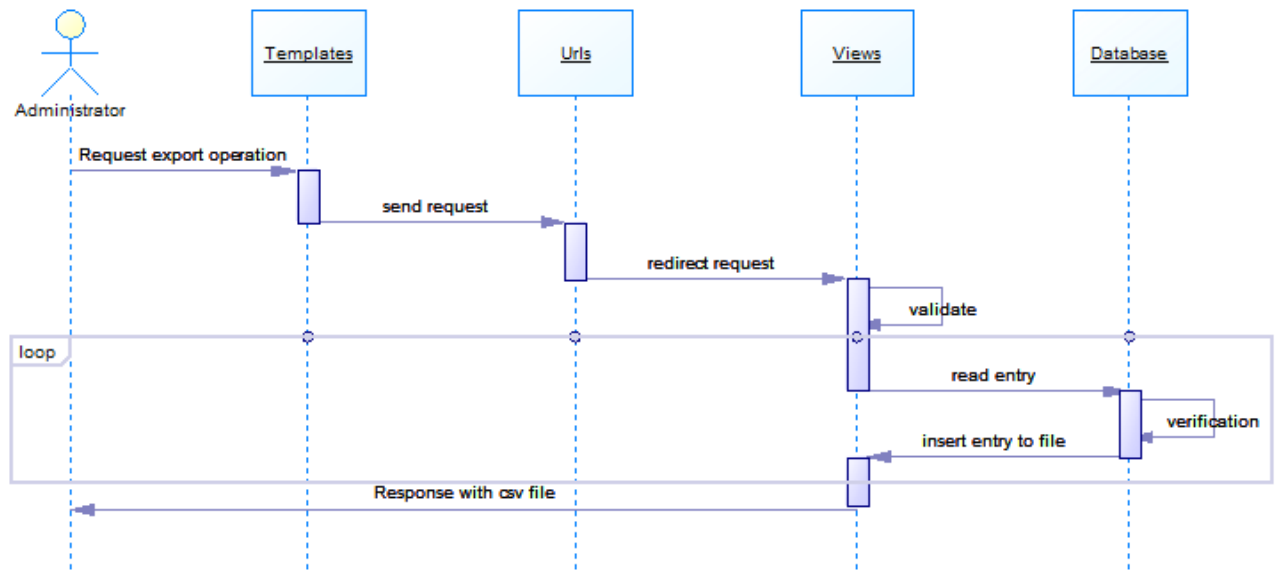


Figure 18: Exporter les méthodes de facturation

- Extraire facture

Le processus d'extraction de factures permet à l'administrateur d'extraire une facture au format pdf, ce qui en fait la partie la plus importante de l'application.

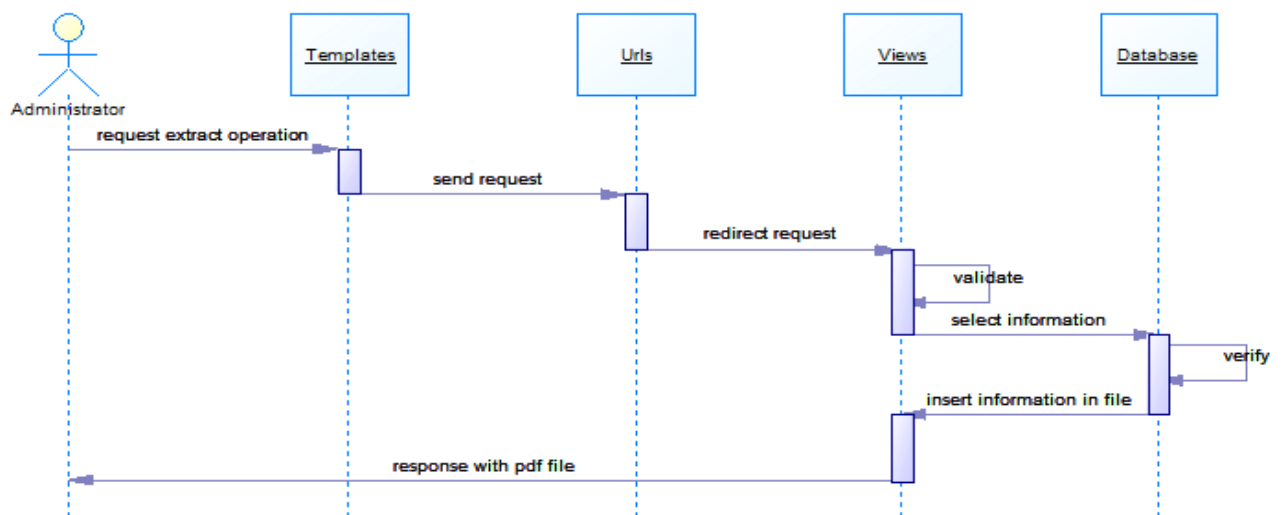


Figure 19: Extraire facture

IV. Conclusion

Au cours de ce chapitre, nous avons détaillé l'étape analyse qui nous a permis de déterminer les différentes catégories et aussi bien de mettre en avant la dynamique dans notre projet, et grâce à l'activité de conception, nous pensons maintenant que l'architecture est suffisamment stable. Maintenant, nous avons un modèle prêt à coder, l'implémentation de ce modèle sera l'objectif de l'étape suivante et le sujet du prochain chapitre où nous présenterons la réalisation qui doit être conforme à la conception déjà établie.

Chapitre VI : Réalisation

I. Introduction

Le présent chapitre constitue le dernier volet de notre rapport dont l'objectif est d'exposer le travail réalisé. Pour cela, nous commencerons par la gestion du projet et nous enchaînons, ensuite, avec la présentation du travail réalisé à travers quelques exemples d'illustration.

II. Environnement de travail :

La spécification technique est une activité primordiale pour la conception de l'architecture. Une bonne connaissance du matériel et des connexions physiques existantes va nous permettre d'éviter le risque d'avoir un développement non adéquat avec le matériel existant ou une architecture incapable de répondre aux exigences fonctionnelles. Dans ce qui suit nous présentons la configuration matérielle et logicielle existante:

1. Environnement matériel

Tout au long de ce projet, le travail a été réalisé sur un environnement matériel comportant des machines ayant les configurations suivantes : Un ordinateur avec la configuration suivante

- Fabricant : Dell
- Processeur : Intel®Core™ i5-3210M CPU @2.50GHz, Smart Cache 3MB
- RAM : 8 Go
- Graphique : NVIDIA GeForce GT 630M
- Système d'exploitation : Windows 10, 64 bits

2. Environnement logiciels

System d'exploitation	Linux Ubuntu Desktop 14.04 64bits LTS
SGBD	MySQL and Python-MySQLdb
Outils de modélisation	Star UML
Langage de développement et framework	Django (version 1.8) ; Python HTML5 (HTML, CSS, JavaScript, jQuery, highcharts, Bootstrap)

III. Etude de risques

Avant de se lancer dans la conception, nous avons déterminé les principaux risques mettant en danger la réalisation du projet, afin de les éviter et assurer le maximum de stabilité pour le système. Ainsi, nous avons fait face à deux types de risques :

- Des problèmes d'ordre techniques
- La non-maîtrise de certains Frameworks.
- Des problèmes d'ordre général

Ces problèmes se résument essentiellement au délai de livraison. En effet, nous étions contrariées par l'ampleur de notre projet d'une part, et par le délai de l'achèvement du travail demandé, d'autre part. On nous a fixé une durée de stage de 4 Mois en tout.

IV. Gestion du projet

La mise en place d'un projet nécessite une planification des tâches à effectuer pour la réalisation d'un projet dans le but de rendre plus simple le suivi de son avancement. Pour mener à bien notre projet et préciser ses différentes étapes à faire, nous avons utilisé le diagramme de Gantt, qui permet de représenter visuellement l'état d'avancement d'un projet en modélisant les différentes tâches qui constituent le projet. Gantt qui est un outil permettant de modéliser la planification de tâches nécessaires à la réalisation d'un projet. Pour mieux expliquer les répartitions des tâches nous avons réalisé le diagramme ci-dessous

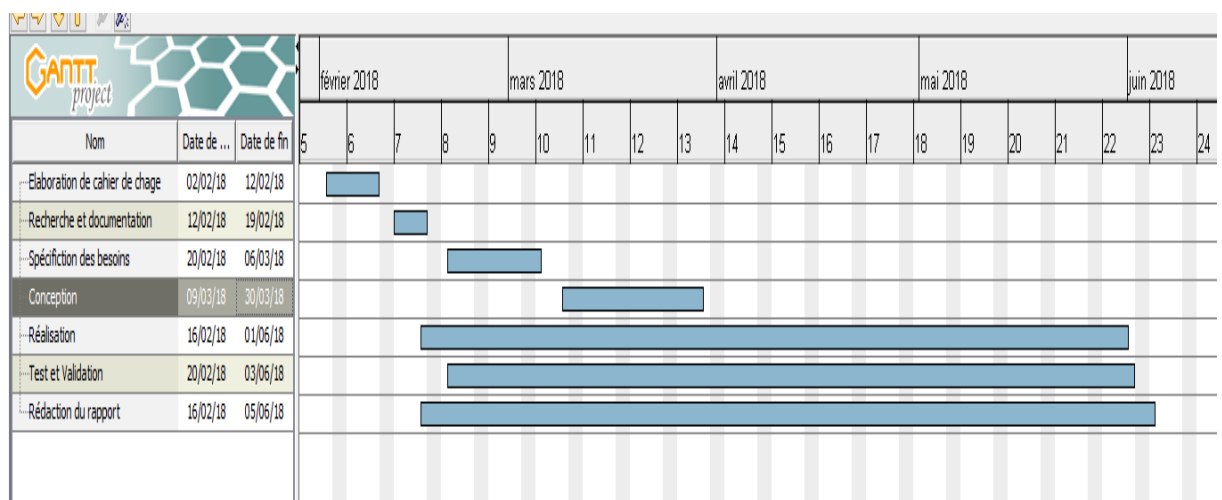


Figure 20:diagramme de Gantt

V. Principaux scénarios :

1. Launching application

Pour lancer un projet dans Django, l'utilisateur doit accéder au serveur puis exécuter la ligne de commande suivante:

```
root@Django:~/TaxPro# ./manage.py runserver 192.168.10.160:8080
Performing system checks...

System check identified no issues (0 silenced).
May 03, 2016 - 20:36:10
Django version 1.8, using settings 'TaxPro.settings'
Starting development server at http://192.168.10.160:8080/
Quit the server with CONTROL-C.
```

Figure 21; lancement du serveur Application

Après l'exécution du serveur et l'accès à l'application via le navigateur Web, le serveur affiche le journal de la transaction Web effectuée entre celui-ci et la machine cliente (voir l'Annexe 2).

2. Page d'authentification

Comme la montre la figure ci-dessous, la page d'authentification est composée uniquement de 3 éléments qui sont le login, le mot de passe et le bouton de connexion. Lorsque l'utilisateur a réussi la connexion, il sera redirigé vers la page du tableau de bord, sinon il sera renvoyé à la page de connexion

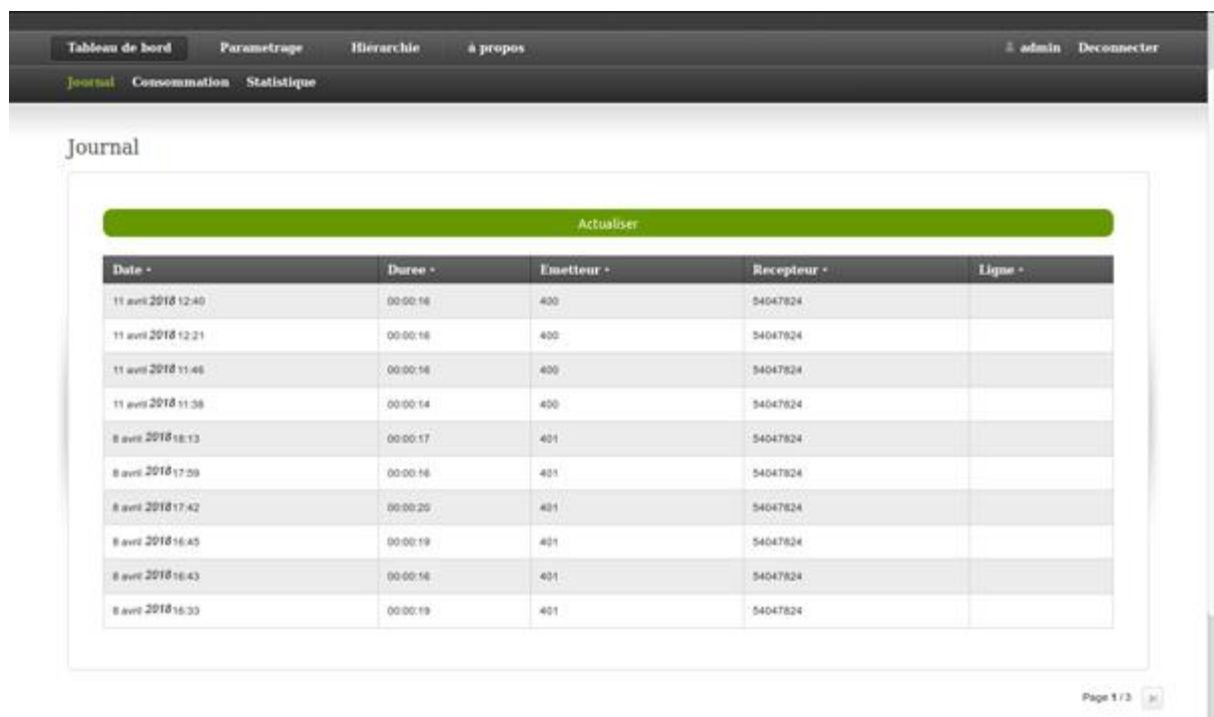


Figure 22:page d'authentification

3. L'interface Dashboard

La figure suivante fait une vue abstraite de l'application, elle montre en haut à gauche le raccourci des composants (tableau de bord, paramètres, hiérarchique ...), et en haut à droite les composants du profil (gestion du profil, déconnexion), au milieu, nous trouvons la liste des journaux et le bouton d'actualisation. Les journaux sont répertoriés dans l'ordre décroissant que les nouveaux journaux apparaissent en premier.

Pour activer un filtre, l'utilisateur doit cliquer sur la valeur du filtre distingué.



Date	Duree	Emetteur	Recepteur	Ligne
11 avr 2018 12:40	00:00:16	400	54047824	
11 avr 2018 12:21	00:00:16	400	54047824	
11 avr 2018 11:46	00:00:16	400	54047824	
11 avr 2018 11:38	00:00:14	400	54047824	
8 avr 2018 16:13	00:00:17	401	54047824	
8 avr 2018 17:59	00:00:16	401	54047824	
8 avr 2018 17:42	00:00:20	401	54047824	
8 avr 2018 16:45	00:00:19	401	54047824	
8 avr 2018 16:43	00:00:16	401	54047824	
8 avr 2018 16:33	00:00:19	401	54047824	

Figure 23:tableau de bord de l'application

4. Gestion IPBX

Au début, l'administrateur doit configurer les paramètres de connexion qui permettent au service d'application de communiquer avec l'IPBX et de recevoir les informations des journaux. La figure ci-dessous clarifie les différents paramètres nécessaires:

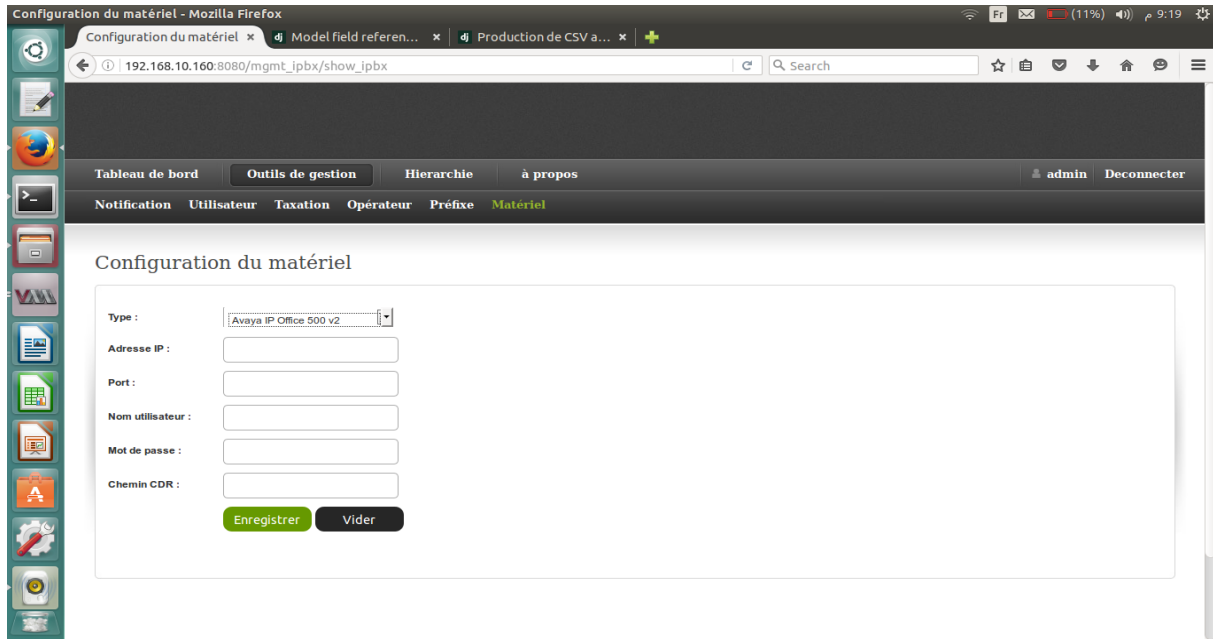


Figure 24:Page de configuration IPBX

5. Gestion de la facture

Comme nous l'avons dit, la consommation est la partie la plus importante de l'application; il permet à l'administrateur de contrôler le téléphone en fonction des taux de consommation, de l'utilisateur et de la ligne de service.

Dans la colonne d'opération, nous avons deux boutons, le premier doit récupérer la facture et le second est de réinitialiser la facturation de la ligne sélectionnée, il apparaît juste comme l'administrateur de se connecter en tant qu'utilisateur root. Pour passer d'une liste de facturation à l'autre, l'administrateur doit utiliser la liste déroulante située en haut de la table.

Lors de l'accès à la page de consommation en tant que super-utilisateur, un bouton de lien appelé tout réinitialiser, est ajouté à la liste de gestion des profils, il permet à l'administrateur de réinitialiser la facturation pour toute l'application.

Lorsque nous accédons à la page de consommation en tant que super utilisateur, un bouton de lien appelé tout réinitialiser est ajouté à la liste des profils de gestion, il permet à l'administrateur de réinitialisation de la facturation pour toutes les applications.

Pour trier les valeurs par colonne, l'utilisateur doit cliquer dans l'en-tête de la colonne souhaitée.

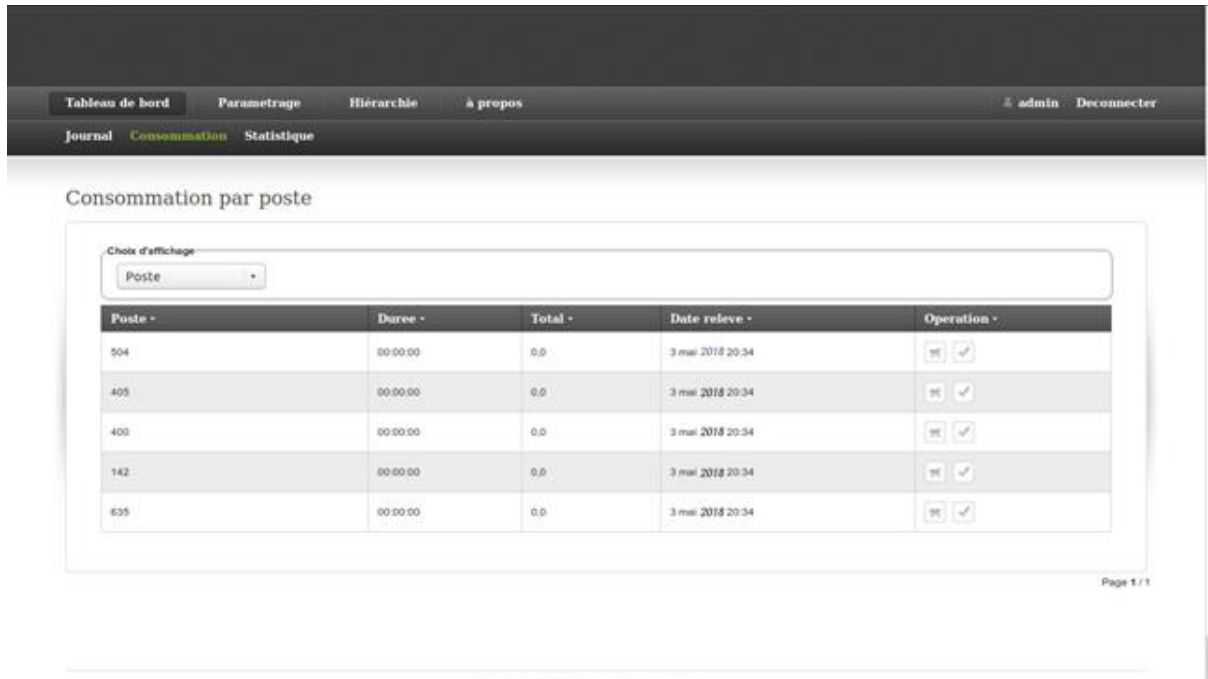
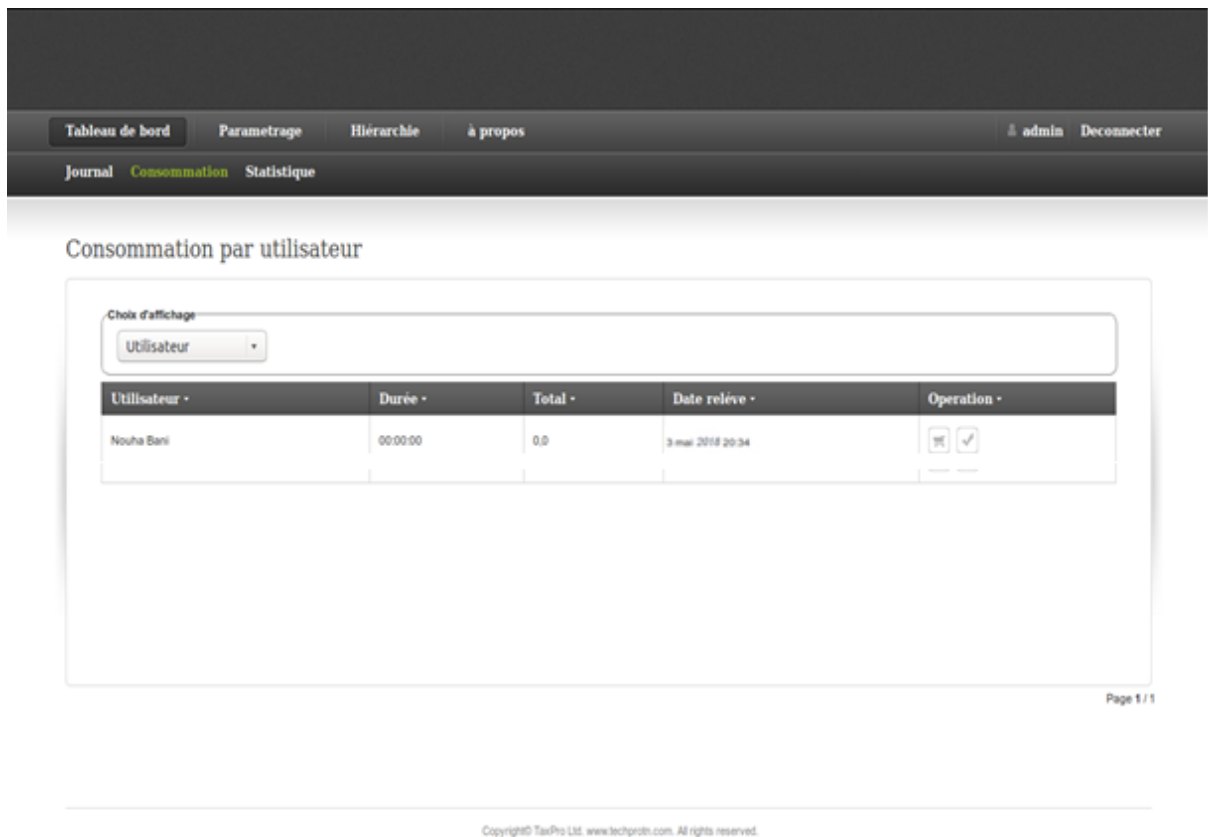


Figure 25:page de consommation de cabine téléphonique



3 mai 2018 20:34

Figure 26:page de consommation d'un utilisateur de téléphone

6. Page Consultation statistique

Pour consulter les statistiques, l'administrateur doit cliquer sur l'entrée souhaitée. La page statistique est composée de:

- Paramètres personnalisés: la liste déroulante composée contient toutes les entrées, la date de début, la date d'arrêt, le prix et la taxe d'augmentation.
- Informations générales: il comprend des détails de consommation (durée, total ..).
- Statistique: ce champ affiche une statistique en camembert.
- Journaux: cette partie répertorie les journaux pour les entrées sélectionnées.

Les factures personnalisées peuvent être extraites après la personnalisation de l'option statistique.

The screenshot displays the 'Bilan utilisateur' page. At the top, there is a navigation bar with 'Tableau de bord', 'Parametrage', 'Hiérarchie', and 'à propos'. The user is logged in as 'admin' and can click 'Deconnecter'. The main content area is titled 'Bilan utilisateur' and includes a 'Personnalise' section with a dropdown for 'Utilisateur' (set to 'acem Jhine'), date range (Debut: 3 Mai 2016, Fin: 1 Jan 2027), and price (0.1). Below this are 'Informations generales' and 'Statistique' sections. The 'Informations generales' section lists details for the user 'Utilisateur', including service, poste, and call statistics. The 'Statistique' section is currently empty, showing 'Chart will be rendered here'. There are 'Collecter' and 'Extraire' buttons.

Figure 27:Page statistique personnalisable

7. Statistique par opérateur

Statistique par opérateur Le bouton de statistiques sur la page du tableau de bord redirige l'administrateur d'une page de consommation par l'opérateur.

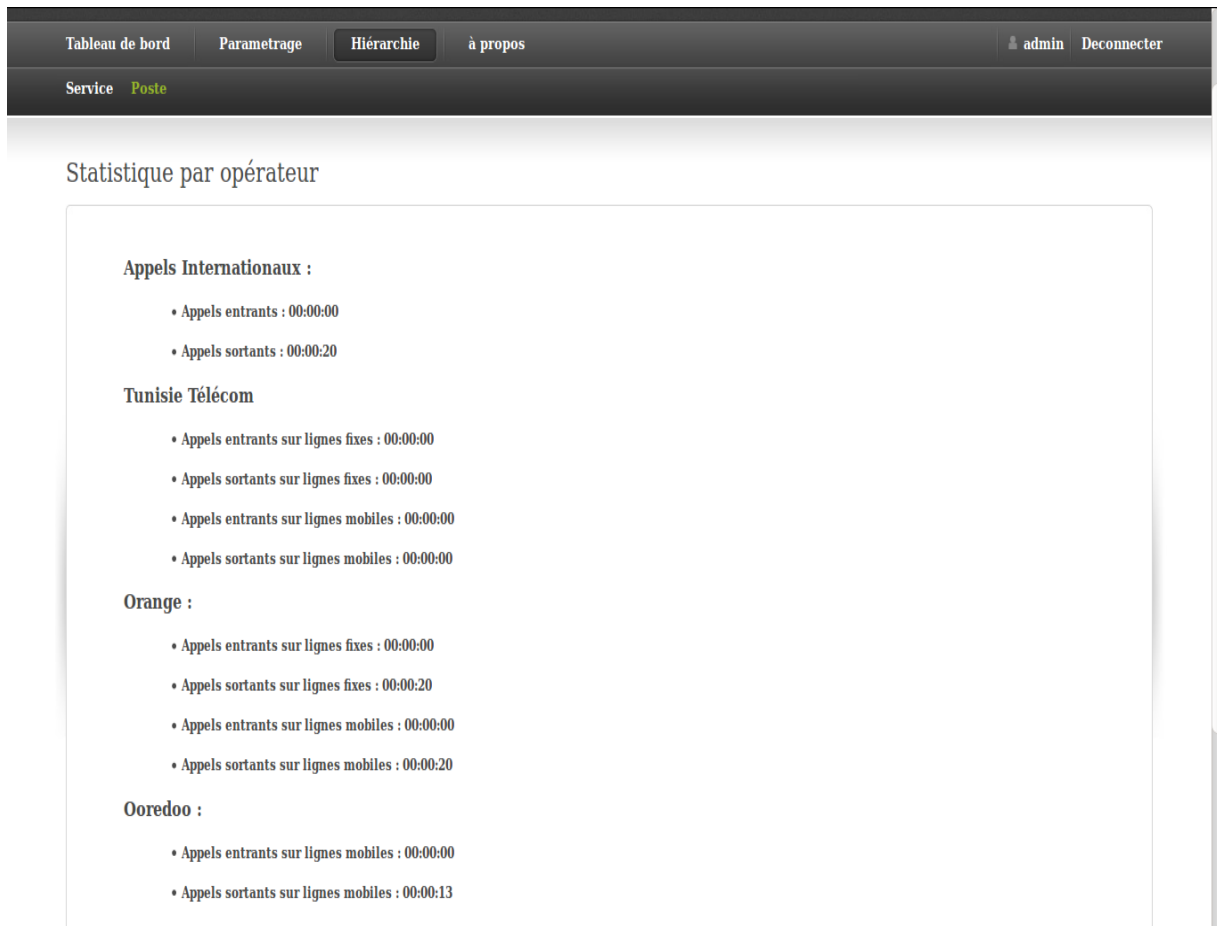


Figure 28:page des Statistiques par opérateur

8. Gestion de la méthode d'imposition

La gestion des entrées des méthodes d'imposition est composée de:

- ✓ La page Gestion: Elle contient la liste des méthodes de taxation suivie par le bouton de mise à jour et de suppression. La liste des boutons d'action représente les autres actions qui peuvent être effectuées, par exemple créer, exporter et importer.

Les méthodes de taxation

Actions

Méthode	Prix(m)	Majoration	Description	Opération
methode 3	255,0	14,0	nk	
methode 1	125,0	255,0	hhd	
methode 4	25,0	236,0	2mf	
methode 7	200,0	0,0	00	

Page 1 / 1

192.168.10.160:8080/taxameth/show_tax

Copyright© TaxPro Ltd. www.techprotn.com. All rights reserved.

Figure 29:page de gestion des méthode d'imposition

- ✓ La Page de création : L'utilisateur doit écrire les paramètres des méthodes de taxation et l'identifier avec un titre pour le différencier des autres méthodes. Le service de création dispose d'une page spéciale que tout administrateur doit associer à un supérieur hiérarchique (voir annexe 3).

Ajouter méthode

Titre :

Prix(m) :

Majoration :

Description :

Page 1 / 1

Copyright© TaxPro Ltd. www.techprotn.com. All rights reserved.

Figure 30:Page de création des méthodes d'imposition

- ✓ Bouton Supprimer:

Les méthodes d'imposition sont protégées contre la suppression lorsqu'elles sont utilisées par une ou plusieurs cabines téléphoniques. Si tel est le cas, le gestionnaire envoie une notification avec un lien vers le changement de page, informe l'administrateur et le laisse changer facilement et en toute sécurité.

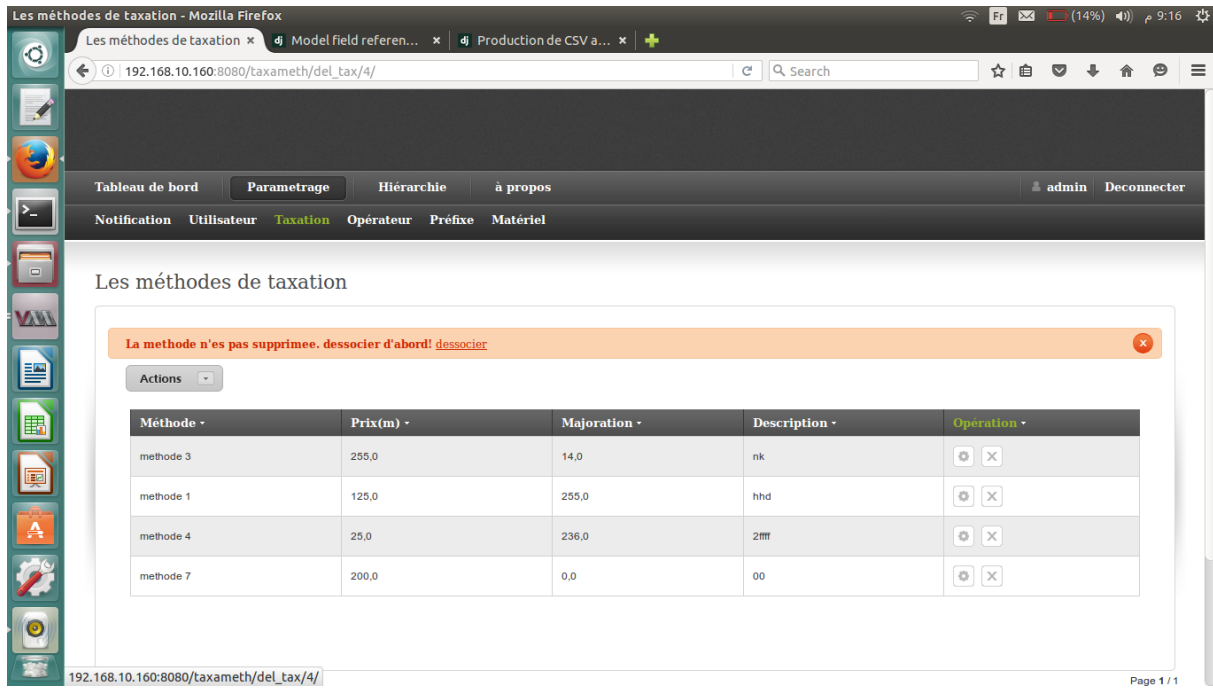


Figure 31:Erreur de suppression des méthodes de taxation

- ✓ Modifier la page de gestion des méthodes: elle n'existe que dans la méthode du gestionnaire. Il protège l'application en permettant à l'administrateur de changer de méthode s'il est utilisé par un ou plusieurs numéros de téléphone lors de la suppression

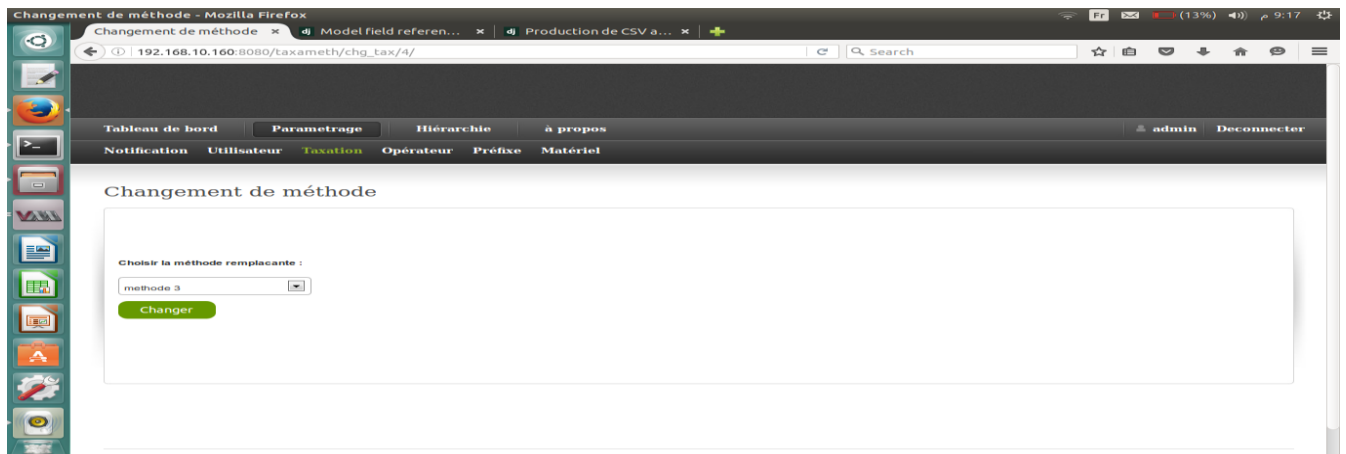


Figure 32:changer les méthodes d'imposition

* Si les méthodes sont supprimées, une notification est affichée à l'administrateur pour confirmer la suppression.

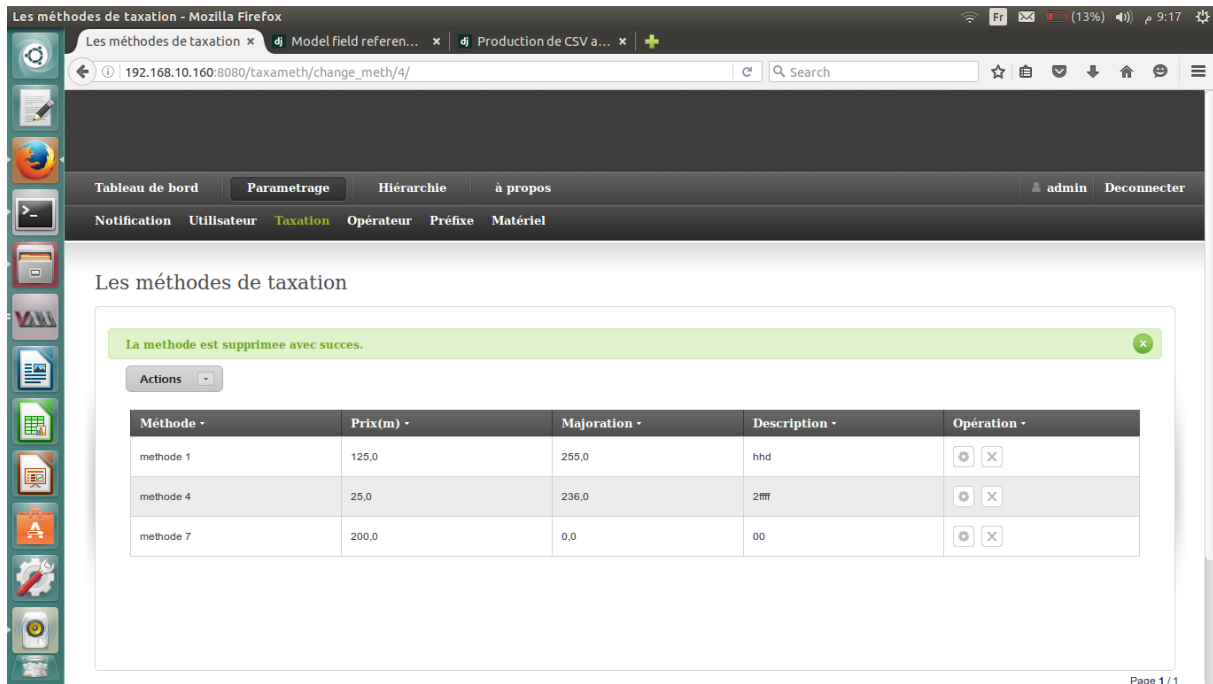


Figure 33: suppression réussite

- ✓ La page de mise à jour: cette page permet à l'utilisateur de l'application de mettre à jour les méthodes de taxation. Il récupère les paramètres des méthodes pour faciliter le processus de mise à jour.

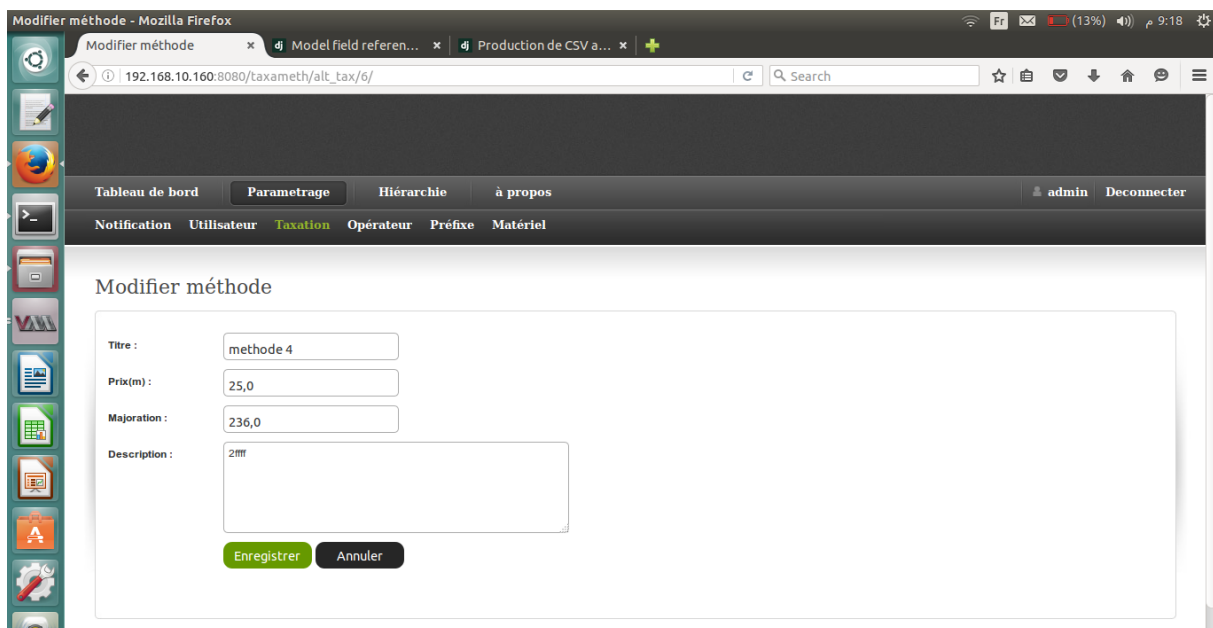


Figure 34 : page de mise à jours

- ✓ Page Import: cette page est utilisée pour sélectionner le fichier csv qui contient la liste des méthodes déjà exportées.

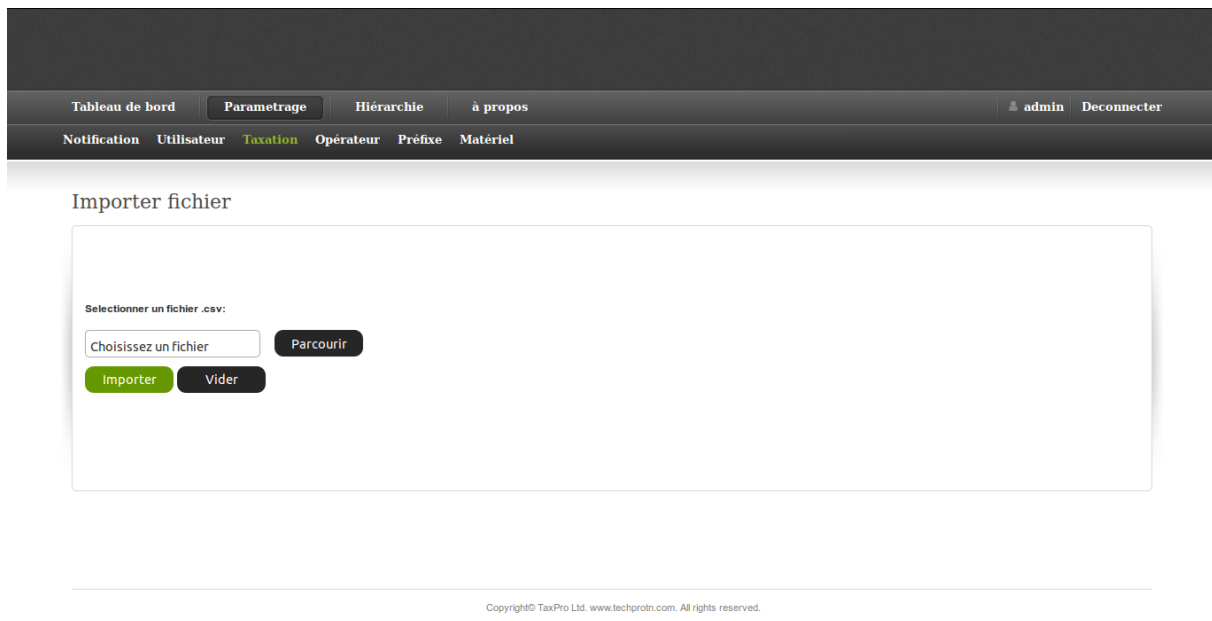


Figure 35: importation de csv

9. Interfaces de gestion des utilisateurs téléphoniques :

Le gestionnaire des utilisateurs du téléphone affiche dans sa première carte la liste de tous les utilisateurs suivis, dans chaque rangée, par les informations (téléphone personnel, email, adresse). Lorsque l'administrateur supprime une entrée associée au numéro de téléphone, la ligne correspondante de la liste des numéros de téléphone est supprimée pour sécuriser le processus de facturation.

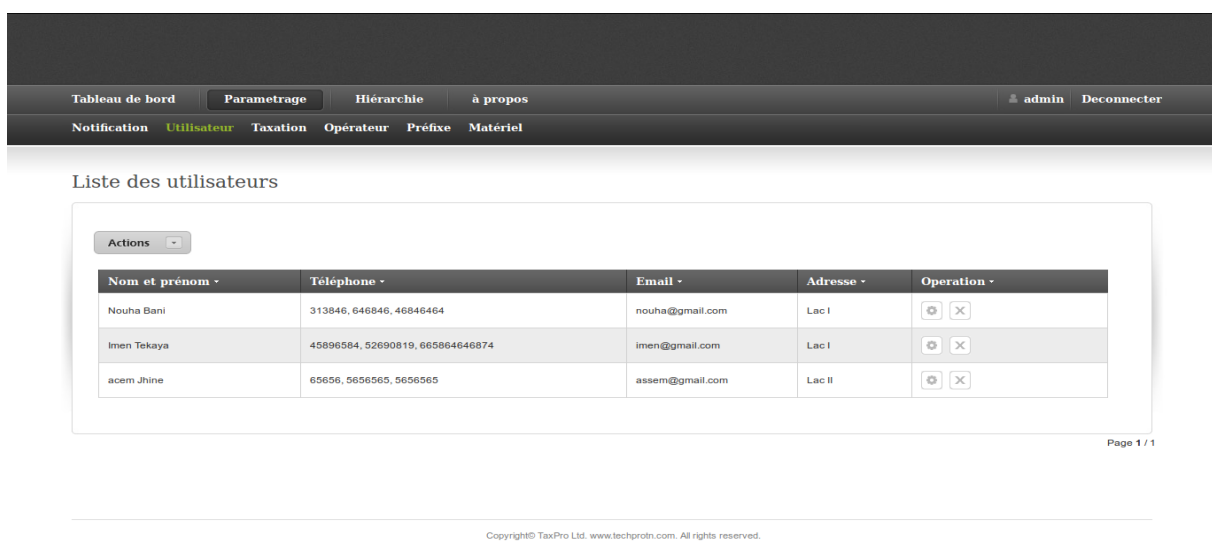


Figure 36:page de gestion d'utilisateur

Les utilisateurs de téléphone créent une page "permet à l'administrateur d'ajouter une nouvelle entrée en remplissant le texte entré par les informations de contact.

The screenshot shows a web application interface for adding a user. The top navigation bar contains the following elements: 'Tableau de bord', 'Parametrage', 'Hiérarchie', and 'à propos'. On the right side of the navigation bar, there is a user profile 'admin' and a 'Deconnecter' link. Below the navigation bar, there is a secondary menu with 'Notification', 'Utilisateur', 'Taxation', 'Opérateur', 'Préfixe', and 'Matériel'. The main content area is titled 'Ajouter Utilisateur' and contains a form with the following fields: 'Nom et prénom', 'Téléphone 1', 'Téléphone 2', 'Téléphone 3', 'Email', and 'Adresse'. At the bottom of the form, there are two buttons: 'Enregistrer' (green) and 'Vider' (black). At the bottom of the page, there is a small copyright notice: 'Copyright© TaxPro Ltd. www.techprotn.com. All rights reserved.'

Figure 37:formulaire d'ajout d'utilisateur

VI. Conclusion

Au cours de ce chapitre, nous avons présenté les technologies qui ont été choisies pour mettre en œuvre nos projets. Nous avons terminé en affichant les principales fonctionnalités et interfaces de notre application.

Conclusion et futur travail

Dans ce projet, dédié à l'obtention du diplôme national de mastère professionnelle en nouvelles technologies des réseaux et télécommunications à UVT, nous avons conçu et développé une application qui aide l'administrateur à contrôler les appels sortants et à gérer les paramètres de facturation au sein de l'entreprise.

La revue de la littérature et l'étude de l'état de l'art nous ont permis d'offrir notre solution aux entreprises pour lesquelles BILLPro travaille.

Nous avons adopté une présentation semi-formelle des exigences de l'administrateur. Par conséquent, nous avons utilisé un ensemble de diagrammes de cas d'utilisation en utilisant UML.

Pendant les différentes phases du développement de notre application nous avons travaillé avec différentes technologies, langages de programmation et frameworks, par exemple python comme langage de programmation backend, Django ORM et MySQL comme framework de persistance, HTML, bootstrap et jQuery pour les interfaces frontend ;

En fait, le choix des frameworks avait une grande influence sur la conception de notre application, nous en avons donc fait une application web avec plusieurs niveaux. De plus, pour mieux implémenter l'application, nous avons suivi les modèles de conception Django et MVC.

En bref, comme cela a été mentionné, notre application visait à rendre le processus de facturation plus efficace, à savoir la «Gestion des paramètres» et les «Gestion des factures» et «Contrôle des appels sortants avec un tableau de bord».

Néanmoins, la gestion VoIP est un domaine plus large que ces fonctionnalités.

Donc, pour améliorer notre application, nous proposons d'ajouter des modules qui reflètent d'autres aspects du système de gestion de la fiscalité, par exemple les "statistiques avancées", les "paramètres avancés" et "contrôle des appels directionnels". Cela sera possible grâce à la nature extensible de Django et du MVT.

Nétographie

[N1] <http://www.ipexia.com/telephonieip-ipbx-voip/> 18/05/2016

[N2] http://www.madeintunisia.com/detail_produit-30202.html 30/03/2016

[N3] <https://fr.scribd.com/doc/49697489/Processus-de-Developpement-Y-Processus-2TUP>

[N4] https://www.memoireonline.com/04/12/5729/m_Analyse-et-developpement-d-un-logiciel-de-gestion-des-donneurs-de-sangcas-du-CNTS6.html

[N5] https://www.tutorialspoint.com/django/django_overview.htm

[N6] <https://www.djangoproject.com/>

[N7] <http://www.paramiko.org/>

Annexe

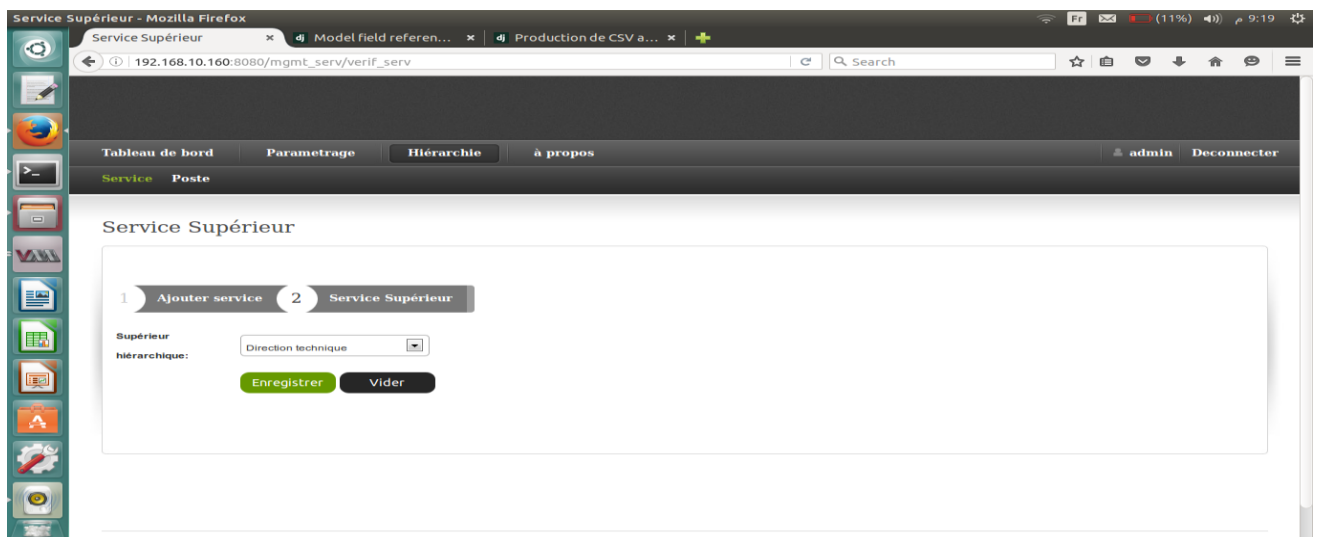
```

root@django:~/TaxPro# ./manage.py runserver 192.168.10.160:8080
Performing system checks...

System check identified no issues (0 silenced).
May 03, 2016 - 20:36:10
Django version 1.8, using settings 'TaxPro.settings'
Starting development server at http://192.168.10.160:8080/
Quit the server with CONTROL-C.
[03/May/2016 20:36:52]"GET /static/images/shared/nav/menus/back_1.gif HTTP/1.1" 404 1720
[03/May/2016 20:36:53]"GET /taxameth/show_tax HTTP/1.1" 302 0
[03/May/2016 20:36:53]"GET /?next=/taxameth/show_tax HTTP/1.1" 200 1945
[03/May/2016 20:36:59]"POST / HTTP/1.1" 302 0
[03/May/2016 20:37:00]"GET /callog/show_log HTTP/1.1" 200 18758
[03/May/2016 20:37:00]"GET /callog/css/pro_dropline_ie.css HTTP/1.1" 404 4970
[03/May/2016 20:37:00]"GET /static/images/shared/nav/menus/back_1.gif HTTP/1.1" 404 1720
[03/May/2016 20:37:00]"GET /static/images/shared/nav/pro_line_0.gif HTTP/1.1" 404 1714
[03/May/2016 20:37:01]"GET /aboutus/show_us HTTP/1.1" 200 13204
[03/May/2016 20:37:02]"GET /static/images/shared/nav/pro_line_0.gif HTTP/1.1" 404 1714
[03/May/2016 20:37:03]"GET /static/images/shared/nav/menus/back_1.gif HTTP/1.1" 404 1720
[03/May/2016 20:37:04]"GET /taxameth/show_tax HTTP/1.1" 200 15448
[03/May/2016 20:37:04]"GET /static/images/shared/nav/pro_line_0.gif HTTP/1.1" 404 1714
[03/May/2016 20:37:04]"GET /static/images/shared/nav/menus/back_1.gif HTTP/1.1" 404 1720
[03/May/2016 20:37:04]"GET /static/images/shared/nav/menus/back_1.gif HTTP/1.1" 404 1720

```

Annexe 1: Server log line







Annexe 2: Ajouter une page supérieure

Tableau de bord Parametrage Hiérarchie à propos admin Deconnecter

Journal **Consommation** Statistique

Consommation par ligne

Choix d'affichage
Ligne ▾

Ligne ▾	Durée ▾	Total ▾	Date relève ▾	Operation ▾
74589654	00:00:00	0,0	3 mai 2016 20:34	 
32155438	00:00:00	0,0	3 mai 2016 20:34	 

Page 1 / 1

Copyright© TaxPro Ltd. www.techprofn.com. All rights reserved.

Annexe 3: Page de consommation par ligne