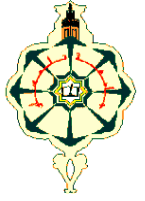


REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTRE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE ABOU BAKR BELKAID-TLEMCEM
FACULTE DE TECHNOLOGIE
DEPARTEMENT DU GENIE ELECTRIQUE ET ELECTRONIQUE

MEMOIRE DE FIN D'ETUDE POUR L'OBTENTION DE DIPLOME DE MASTER
EN
TELECOMMUNICATIONS
OPTION
RESEAUX ET SYSTEMES DE TELECOMMUNICATIONS

THEME

***Serveur intranet pour la diffusion des
requêtes GSM USSD***

Soutenu le 23 Juin 2013

Présenté par :

Mr. OULD AHMED Ismail

Devant les membres du jury composés de :

Dr. DJENNAS Sidi Ahmed

Président

Dr. BAHRI Sidi Mohammed

Examineur

Dr. KAMECH Samir

Examineur

Dr. BENADDA Belkacem

Encadreur

Année universitaire: 2012 – 2013

Dédicace

Dédicace

Je dédie cet humble travail à ceux qui me sont les plus chers :

- *Ma chère mère Oumeyma FADEL, pour sa tendresse, ses gâteries, et ses sacrifices dont elle me comblait.*
- *Mon cher père Ahmed khadil, pour ses aides, ses encouragements et ses suivis permanents.*
- *Mes frères Ahmed Djebri Abdallah, Med Lemine Hawbette, Med Laghdaf ould Ahmed et mon cher petit frère Abdelkader ould Ahmed.*
- *Mes deux sœurs, Amal Hawbette et Aicha Ahmed khadil.*
- *Mes cousins et mes cousines et à toute ma famille.*
- *Mes chères Amis Mauritanien à Tlemcen, Med Kaber ould el heiba, Ahmed BEDDY, Ahmed MENANE, Sidi ould ely, Med yahya ould Taleb Abderrahmane, Med Ahmed Aida, El hadi ould Bouh, Abderahman ould yasser.*
- *Mes amis Algériens, Abdesslam BELKHITER, Yasmine SABOUNDJI, Bilal BELDJILALI, Imane SELADJI, Adila Nemich, Mokhtar KORTI, Sara FEROU, Youcef BENDIAF, Houcine HAMEL, Assi BENFKEIR, Leil HADJAZE, Med BARKA, Youssef Bendiabdallah, Brahim Benemostefa, Tarek Gourari, et à toute la promotion de RST 2013.*
- *A tous ceux qui m'aiment et qui ont cru en moi !*
- *Tous ceux qui vont admirer mon travail.*

Avec toute la chaleur de mon affection et ma sincérité.

ISMAIL OULD AHMED



Sommaire

Sommaire

Sommaire.....	I
Liste des figures.....	IV
RESUME.....	VI
INTRODUCTION GENERALE.....	1
Chapitre I Etude du protocole USSD.....	2
I.1-Introduction	2
I.2-Le réseau GSM.....	2
I.2.1- Présentation.....	2
I.2.2-Architecture GSM.....	3
I.2.2.1- Sous système radio.....	3
I.2.2.2- Sous système réseau ou d'acheminement.....	4
I.2.2.3- Sous système d'exploitation et maintenance	5
I.3-Mode de fonctionnement USSD.....	5
I.4- Le protocole MAP (Mobile Application Part).....	6
I.5- Procédure des sessions USSD [2].....	7
I.6-Les dialogues USSD dans le réseau GSM.....	8
I.6.1- Services USSD initiés par le mobile.....	8
I.6.2- Services USSD initiés par le réseau.....	9
I.6.3-Horloge USSD (Timers USSD).....	9
I.6.3.1- ProcessUSSDRequest Invoke Timer	9
I.6.3.2-USSDRequest Invoke Timer.....	10
I.6.4- Les dialogues multipl.....	10
I.7-Aspects d'adressage.....	10
I.8-Les commande GSM normalisée.....	10
Conclusion.....	12
Chapitre II Réseaux internet -intranet.....	13
Introduction.....	13
II.1- Modèle de référence OSI (Open Systems Interconnexion).....	13
II.2-Modèle TCP/IP (Transmission Control Protocol / Internet Protocol).....	15
II.2.1- Historique des protocoles TCP/IP.....	15
II.2.2- Avantages des protocoles TCP/IP.....	15

Sommaire

II.2.2.1-Couche application	16
II.2.2.2-Couche transport.....	17
II.2.2.3- Couche Internet	17
II.2.2.4- Couche Accès réseau	18
II.3-Comparaison TCP/IP-OSI.....	18
II.4-Adressage IP	19
II.5- DNS (Domain Name System)	20
II.6-Protocoles logiciels	21
II.7-Intranet	22
Conclusion	23
Chapitre III Présentation des outils.....	24
Introduction	24
III.1-Serveur	24
III.1.1-Serveur Web.....	24
III.1.2-Serveur Apache	24
III.2-le navigateur Internet	25
III.2.1-Protocole HTTP	25
III.2.1.1- Requêtes HTTP	26
III.3-Langage HTML	26
III.4-Feuilles de style en cascade (CSS).....	27
III.5- Langage PHP	28
III.5.1- Naissance de PHP	28
III.5.2-Fonctionnement de PHP.....	29
III.5.3- Pages dynamiques et PHP.....	30
III.5.4- les garanties d'utilisation de PHP	30
III.6-MySQL	30
III.6.1-MySQL Administrator	31
Conclusion	31
Chapitre IV configuration des serveurs et implémentation de la base des données	32
Introduction	32
IV.1-Installation et configuration du serveur Apache sous Windows	32
IV.2-Installation de PHP	33
IV.3- Installation et lancement de MySQL sous Windows	34
IV.4-Conception de la base de données	37

Sommaire

IV.4.1-Table inscription	37
IV.4.2-Table historique	38
Conclusion	38
Chapitre V Application	39
Introduction	39
V.1.Les scripts de création et l'ouverture d'une session	39
V.1.1.Script de création d'une session (création d'un compte)	39
V.1.2.Script d'ouverture d'une session	40
V.2.Présentation des interfaces graphiques de l'application	42
V.2.1.Page d'accueil	42
V.2.2.Interface graphique de la création d'une session	42
V.2.3.Interface graphique de l'ouverture d'une session	43
Conclusion	48
CONCLUSION GENERALE	49
BIBLIOGRAPHIE	50

Introduction générale

Les principaux services supplémentaires offerts par un opérateur mobile sont : SMS, MMS et USSD (Unstructured Supplementary Services Data), Le protocole USSD est l'une des méthodes préférées pour le rechargement de crédit prépayé. Quelques applications de transaction emploient également ce protocole. Peu d'applications USSD ont été développées car elles ne sont pas aussi souples que les SMS. En outre, l'information ne peut ni être partagée ni sauvegardée car les menus et les textes USSD ne sont pas stockés sur le téléphone ni sur la carte SIM.

Ce présent projet a pour objectif de concevoir et de réaliser un serveur permettant d'implémenter un nouveau service basé sur l'USSD sur un serveur, le rechargement du crédit à partir d'un ordinateur vers un téléphone mobile via un modem GSM ou un téléphone mobile jouant le même rôle que le modem GSM a été utilisé comme validation de la solution élaborée. Conçue pour un réseau Intranet, cette application pourra faire l'objet d'une extension vers Internet.

Ce projet est composé de cinq chapitres :

Le premier chapitre est consacré à l'étude du protocole USSD et son fonctionnement et à l'étude du réseau GSM.

Le deuxième chapitre fait l'objet de l'étude du réseau Internet-intranet dans sa structure et son fonctionnement et éventuellement de l'étude de la famille du protocole TCP/IP.

Le troisième chapitre est consacré à la présentation des différents outils utilisés pour la conception des interfaces graphiques du site.

Le quatrième chapitre aborde la configuration des serveurs et l'implémentation de la base de données qui permettront aux utilisateurs de pouvoir naviguer sur le site.

Enfin, le cinquième chapitre portant sur la partie application, décrit le fonctionnement du site en présentant les différents scénarios et scripts PHP.

I.1-Introduction

L'USSD (*Unstructured Supplementary Service Data*), qui peut se traduire en « Données de services supplémentaires peu structurés » est une fonctionnalité des réseaux GSM ou 3G. Il est généralement associé aux services de téléphonie de type temps réel ou de messagerie instantanée. Sur la station mobile il n'a aucune possibilité d'enregistrement et de transfert qui sont une caractéristique des messages courts SMS classiques. Les temps de réponse pour des services basés USSD interactifs sont généralement plus rapides que ceux utilisés pour SMS. USSD permet par exemple de demander le suivi de consommation. L'USSD permet également une gestion d'appel à des services par menus, en une forme de navigation par sollicitation, générant un nouvel USSD jusqu'à obtention du service final.

Le protocole USSD est une des méthodes préférées pour le rechargement de crédit prépayé. Quelques applications de transaction emploient également ce protocole. Peu d'applications USSD ont été développées car elles ne sont pas aussi souples que les SMS. En outre, l'information ne peut ni être partagée ni sauvée car les menus et les textes USSD ne sont pas stockés sur le téléphone ni sur la carte SIM. On pourrait cependant fournir des flashes d'actualités par l'intermédiaire d'USSD. Le coût de transmission de données via cette plate-forme est nettement inférieur à celui des SMS. Dans ce chapitre nous allons présenter les notions primordiales de ce service.

La fonction **USSD** est une capacité de base de tous les mobiles qu'ils soient GSM, GPRS ou UMTS. Elle utilise les canaux de signalisation pour transmettre des informations bidirectionnelles entre le mobile et l'application définie par l'opérateur. Les messages USSD sont importants pour notifier, fournir des menus, déclencher un service mobile ou configurer un abonné de manière rapide et directe

I.2-Le réseau GSM [1]

I.2.1- Présentation

Le GSM est un système cellulaire et numérique de télécommunication mobile. Il apparaît au début des années 90 en remplacement du Groupe Spécial Mobile, créé en 1982 par l'**ETSI** (**E**uropean **T**elecommunications **S**tandards **I**nstitut) pour élaborer les normes de communication mobile en Europe.

Ce système a été très vite adopté en Europe puis dans le reste du monde, on dénombre aujourd'hui plus de 210 pays utilisant le GSM et plus de 3 milliards d'utilisateurs équipés d'une solution GSM.

La norme GSM existe sous trois formes : le GSM 900 fonctionnant dans la bande de fréquences aux alentours des 900Mhz, le GSM 1800 fonctionnant dans la bande de fréquences aux alentours des 1800Mhz et enfin le GSM 1900 (aux Etats Unis notamment) qui fonctionne dans la bande de fréquences aux alentours des 1900Mhz.

I.2.2-Architecture GSM

Le GSM est composé de trois sous systèmes : Sous système radio, Sous système réseau ou d'acheminement et Sous système d'exploitation et maintenance.

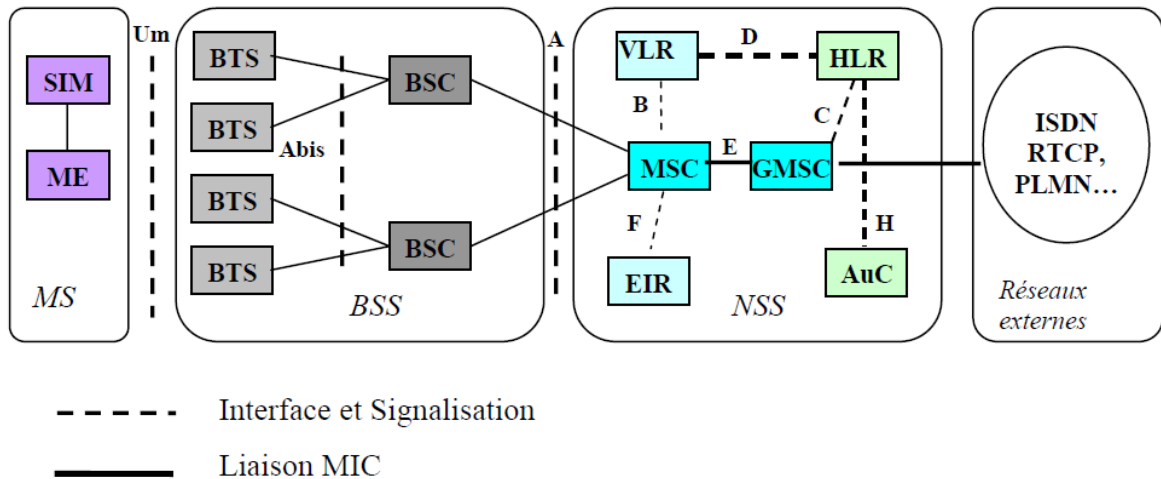


Figure I.1- Architecture GSM

I.2.2.1- Sous système radio

Le sous-système radio gère la transmission radio. Il est constitué de plusieurs entités dont le mobile, la station de base (BTS) et un contrôleur de station de base (BSC)

➤ BTS (Base Transceiver Station)

C'est un ensemble d'émetteur/récepteur pilotant une ou plusieurs cellules. Ces émetteurs/récepteurs sont dans la nature des antennes de forme vertical, déployés sur les toits des immeubles et les piliers. Elle réalise les fonctions semblables à celles de la couche physique et de la couche liaison de données du modèle OSI. En cas de besoin, on peut exploiter une station de base localement ou par télécommande à travers son contrôleur de station de base.

➤ BSC (Base Station Controler)

Ce maillon de la chaîne qui gère une ou plusieurs BTS remplit différentes fonctions tant au niveau de la communication (Concentration et transfert des communications vers un centre unique, commutation, gestion des ressources radios tels le handover) qu'au niveau de l'exploitation.

I.2.2.2- Sous système réseau ou d'acheminement

Il joue un rôle essentiel dans un réseau mobile notamment la prise en charge de toutes les fonctions de contrôle et d'analyse d'informations contenues dans des bases de données nécessaires à l'établissement de connexion utilisant une ou plusieurs des fonctions suivantes : chiffrement, authentification ou roaming. Le NSS (Network Sub System) est constitué des entités MSC, HLR, AuC, VLR et EIR.

➤ MSC (Mobile Switching Centre)

Son rôle principal est d'assurer la commutation entre les abonnés du réseau mobile et les réseaux externes (téléphonie mobile, téléphonie fixe, Internet). Il participe à la fourniture des différents services aux abonnés tels que la téléphonie, les services supplémentaires et les services de messagerie. Il permet encore de mettre à jour les différentes bases de données (HLR et VLR) qui donnent toutes les informations concernant les abonnés et leur localisation dans le réseau.

Les commutateurs MSC d'un opérateur sont reliés entre eux pour la commutation interne des informations.

➤ HLR (Home Location Register)

Il s'agit d'une base de données avec des informations essentielles pour les services de téléphonie mobile et avec un accès rapide de manière à garantir un temps d'établissement de connexion aussi court que possible. Le HLR contient :

- Toutes les informations relatives aux abonnés : le type d'abonnement, la clé d'authentification **Ki** cette clé est connue d'un seul HLR et d'une seule carte SIM, les services souscrits, le numéro de l'abonné (IMSI).
- Un certain nombre de données dynamiques telles que la position de l'abonné dans le réseau en fait, son VLR et l'état de son terminal (allumé, éteint, en communication, libre).

Les données dynamiques sont mises à jour par le MSC. Cette base de données est souvent unique pour un réseau GSM.

➤ VLR (Visitor Location Register)

Cette base de données ne contient que des informations dynamiques et est liée à un MSC comme le montre la figure1. Il y en a donc plusieurs dans un réseau GSM. Elle contient des données dynamiques qui lui sont transmises par le HLR avec lequel elle communique lorsqu'un abonné entre dans la zone de couverture du centre de commutation mobile auquel elle est rattachée. Lorsque l'abonné quitte cette zone de couverture, ses données sont transmises à un autre VLR, les données suivent l'abonné en quelque sorte.

➤ AuC (Authentication Center)

C'est une base de données permettant de remplir la fonction de protection des communications dans un réseau GSM. Pour ce faire GSM prévoit deux mécanismes :

- Le chiffrement des transmissions radios. Remarquons qu'il s'agit d'un chiffrement faible, qui ne résiste pas longtemps à la crypto-analyse
- L'authentification des utilisateurs du réseau au moyen d'une clé **Ki**, qui est à la fois présente dans la station mobile et dans le centre d'authentification.

➤ EIR (Equipment Identity Register)

Permet d'enregistrer les codes **IMEI** (International Mobile Equipment Identity) des téléphones mobiles. Tous les opérateurs n'implémentent pas une telle base de données en raison du coût de l'équipement.

I.2.2.3- Sous système d'exploitation et maintenance

Cette partie du réseau regroupe trois activités principales de gestion : la gestion administrative, la gestion commerciale et la gestion technique. Le réseau de maintenance technique s'intéresse au fonctionnement des éléments du réseau. Il gère notamment les alarmes, les pannes, la sécurité. Ce réseau s'appuie sur un réseau de transfert de données, totalement dissocié du réseau de communication GSM.

I.3-Mode de fonctionnement USSD

L'USSD est une technologie de communication GSM utilisée pour échanger les messages entre un téléphone mobile et une application, les informations USSD sont directement transmises du mobile de l'utilisateur vers l'équipement de l'application.

Typiquement ce service est utilisé comme déclencheur pour invoquer les appels de services indépendants qui n'exigent pas les dépenses d'utilisation aériennes et complémentaires d'un **SMSC (Short Message Service Center)**, comme un service de rappel de service (un suivi de consommation, crédit restant sur votre compte), ou le service de menu interactif (par exemple des cours de la bourse, des résultats sportifs).

Les codes USSD se présentent sous la forme : (%) (%)%XY (*text)#

Où :

- () : indique un élément optionnel.
- % vaut « # » ou « * ».
- X et Y sont deux chiffres (0 à 9).
- text est un élément alphanumérique de longueur variable.

Après émission d'un code USSD, la réponse de l'opérateur est reçue dans les secondes suivantes. USSD permet par exemple de demander un suivi de consommation.

Les deux types de messages (USSD et SMS) sont transmis sur les canaux de signalisation du GSM. A la différence du SMS l'USSD nécessite l'établissement d'une session qui reste ouverte jusqu'à ce que l'un des partenaires la libère.

Les messages USSD sont des messages texte d'une longueur maximale de 182 caractères. L'intérêt de l'USSD est de proposer des services de données à faible bande passante qui permet aux utilisateurs d'envoyer des informations sur un réseau GSM sans connexion à Internet

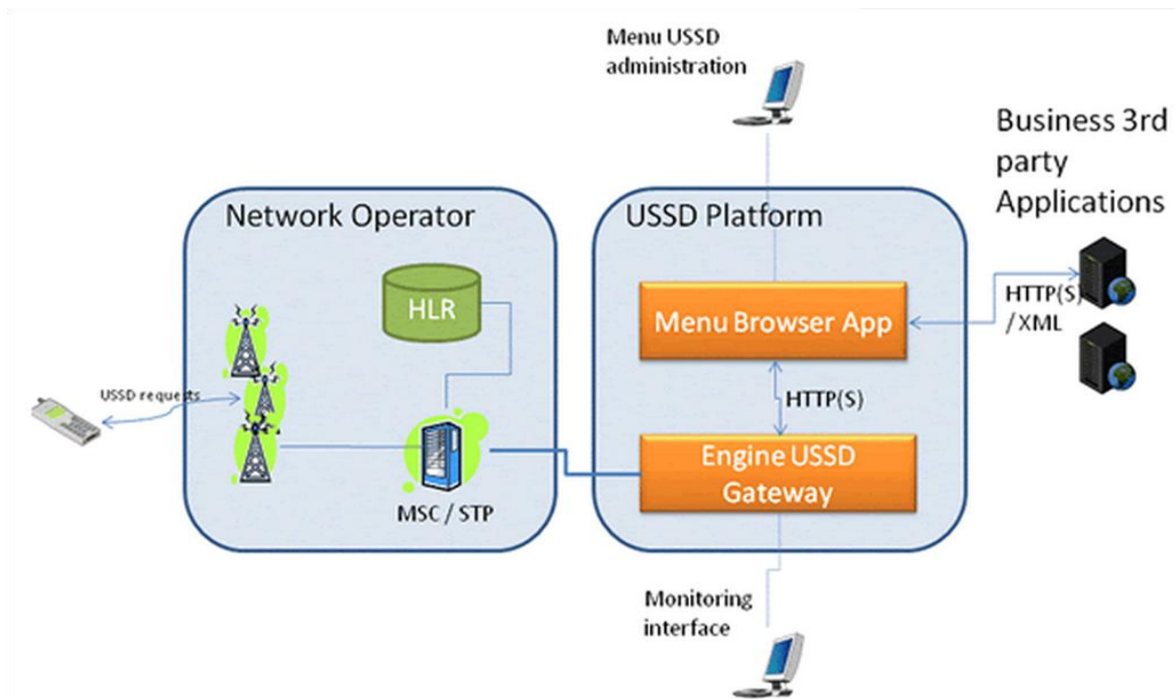


Figure I.2-Architecture USSD

I.4- Le protocole MAP (Mobile Application Part)

Le protocole MAP (Mobile Application Part) régit l'ensemble des échanges entre équipements du réseau mobile (NSS, Network Subsystem). Il offre les fonctions de signalisation nécessaires à un service de communication voix ou données dans un réseau mobile. Le protocole MAP concerne les dialogues entre différentes entités du réseau mobile notamment, MSC/VLR, MSC Server, SGSN, HLR, EIR, SMSC, etc.

Les messages MAP relatifs aux services USSD sont :

- **MAP_PROCESS_UNSTRUCTURED_SS_REQUEST** : Ce service est utilisé entre le VLR et le HLR, entre le HLR et le Gateway USSD afin d'échanger des informations pour la réalisation de services USSD.

Le service MAP_PROCESS_UNSTRUCTURED_SS_REQUEST est un service confirmé. Le Gateway USSD traduit cette requête en une requête spécifique au serveur d'application qui fournit le service.

- **MAP_UNSTRUCTURED_SS_REQUEST** : Ce service est utilisé entre le Gateway USSD et le HLR, et entre le HLR et le VLR lorsque l'entité qui invoque ce service demande des informations de l'utilisateur mobile en relation avec la prise en charge d'un service USSD. Le service MAP_UNSTRUCTURED_SS_REQUEST est un service confirmé.
- **MAP_UNSTRUCTURED_SS_NOTIFY**: Ce service est utilisé entre le Gateway USSD et le HLR et entre le HLR et le VLR lorsque l'entité qui invoque ce service demande l'envoi d'une notification à l'utilisateur mobile, en relation avec la prise en charge d'un service USSD. Le service MAP_UNSTRUCTURED_SS_NOTIFY est un service confirmé.

I.5- Procédure des sessions USSD [2]

Quand un mobile veut accéder à un service USSD, il demande un canal de signalisation, la demande contient en fait le code du service demandé qui est délimité généralement par les caractères «*» et «#». Un tel changement est nécessaire car le mobile peut changer de zone de localisation qui est contrôlé par un autre MSC/VLR et mise à jour à partir du HLR. C'est le MSC qui décide si la demande peut être transmise au HLR ou non. En cas d'erreur (comme la non validité du service demandé ou lorsque l'abonné n'est pas inscrit dans le réseau) un message d'erreur est retourné au MSC.

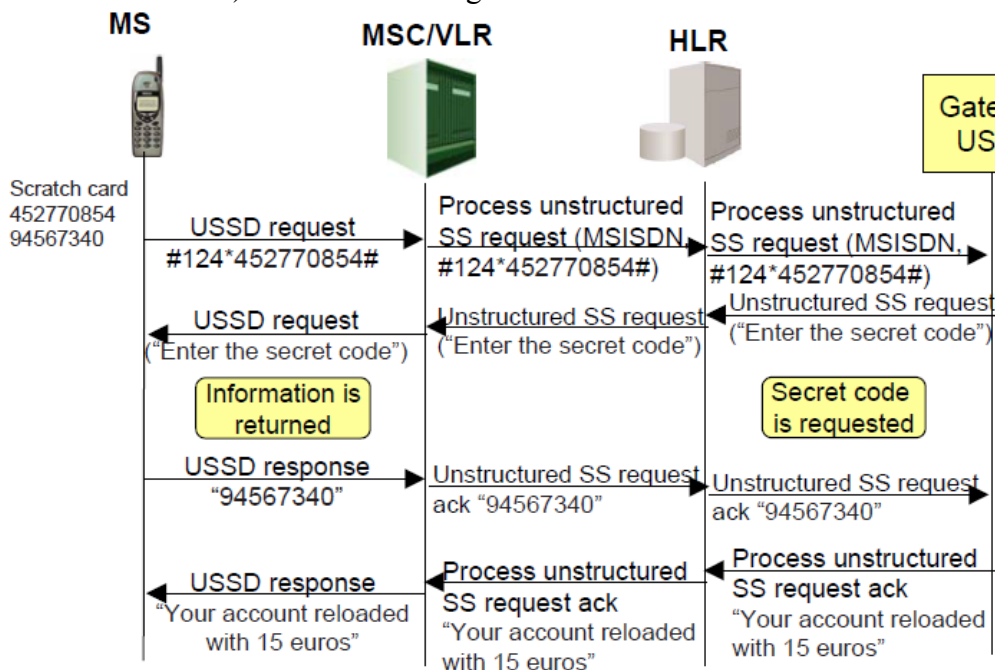


Figure I.3 Service USSD de recharge de compte prépayé

Considérons le service USSD de rechargement de compte prépayé via une carte à gratter (**Figure I.3**):

L'utilisateur introduit le numéro de la carte à savoir 452770854 préfixé par le code #124*.

Ce code correspond au service de recharge. Le MSC/VLR reçoit la demande, reconnaît qu'il s'agit d'une demande de service USSD et génère le message :

MAP_PROCESS_UNSTRUCTURED_SS_REQUEST, émis au HLR de l'utilisateur invoquant le service.

Cette demande qui permet l'établissement d'un dialogue USSD contient aussi le numéro MSISDN de l'utilisateur. Le HLR sait que ce code correspond à un service USSD qui doit être pris en charge par un Gateway USSD. Le HLR relaye donc ce message MAP au Gateway USSD, Le Gateway se charge de traduire cette demande en une requête à émettre au système de recharge des comptes prépayés.

Le système de recharge requiert des informations supplémentaires pour prendre en compte la demande de recharge du client, notamment le code secret présent sur la carte une fois grattée. Cette demande est envoyée au Gateway USSD, qui la traduit en un message

MAP_UNSTRUCTURED_SS_REQUEST, envoyée au HLR qui la relaye au MSC/VLR.

La demande est reçue par le mobile. L'utilisateur répond en fournissant le code demandé.

Le MSC/VLR retourne la réponse **MAP_UNSTRUCTURED_SS_REQUESTAck** contenant le code secret. Cette réponse est relayée au HLR qui l'achemine au Gateway USSD. Ce dernier la traduit et envoie le message résultant au serveur de recharge.

Ce dernier peut à présent recharger le compte de l'utilisateur et retourne une réponse qui confirme le rechargement au client et qui permet par ailleurs de fermer la session USSD.

La réponse MAP correspondante retournée par le Gateway USSD au HLR puis au MSC/VLR est **MAP_PROCESS_UNSTRUCTURED_SS_REQUESTAck**.

I.6-Les dialogues USSD dans le réseau GSM

Un dialogue USSD est l'établissement d'une connexion USSD entre le mobile et le réseau. Il existe deux types de dialogues USSD

I.6.1- Services USSD initiés par le mobile

Cette catégorie de services USSD est initiée en introduisant une commande USSD avec le clavier de son mobile.

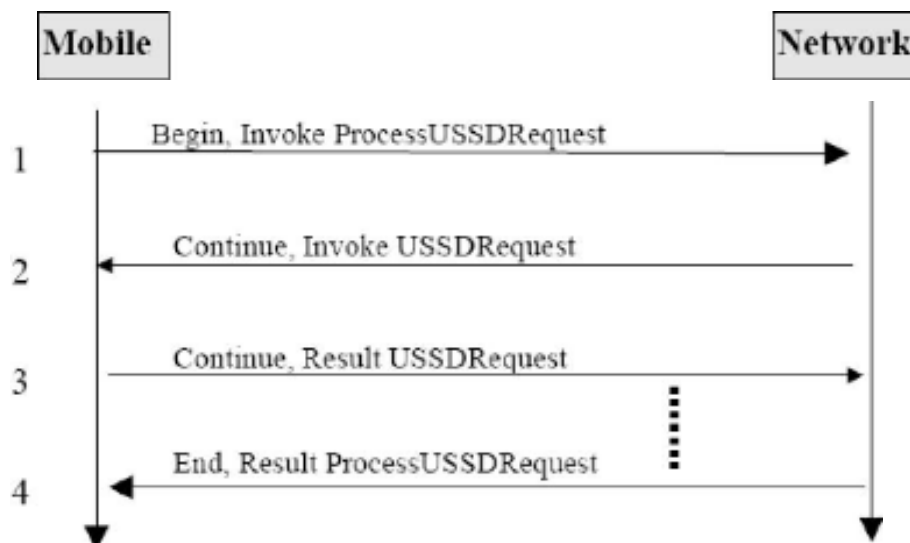


Figure I.4- Le Mobile initie le dialogue USSD

Le mobile initie le dialogue par invocation de la méthode **ProcessUSSDRequest**. L'opérateur (Network) peut répondre par invocation de la méthode **USSDRequest** ou bien libère le dialogue en retournant le résultat via la méthode **ProcessUSSDRequest**.

Si l'opérateur décide d'invoquer la méthode **USSDRequest**, le mobile répond grâce à la méthode **USSDRequest** et ainsi de suite.

I.6.2- Services USSD initiés par le réseau

Le HLR ou le VLR peuvent à tout moment initier une session de service USSD avec le mobile (le mobile est toujours le destinataire des sessions initiées par le réseau). Le réseau inclut un code de service USSD au message USSD envoyé au mobile. Le mobile exécute alors le service USSD requis. Il n'y a pas de rangée de valeur particulière puisque le mobile est toujours le destinataire.

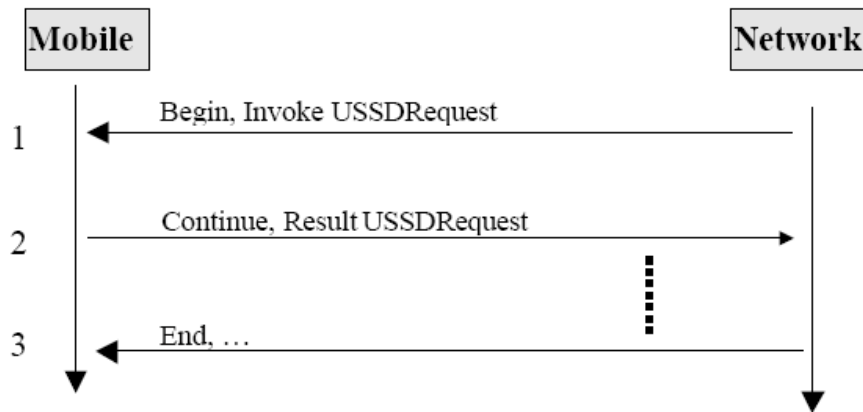


Figure I.5- Le réseau initie le dialogue USSD

L'opérateur initie le dialogue par invocation de la méthode **USSDRequest**. Le mobile répond en retournant le résultat via l'opération **USSDRequest**. Les deux entités s'échangent des messages jusqu'à ce que l'opérateur décide de mettre un terme au dialogue en retournant un END.

I.6.3-Horloge USSD (Timers USSD)

Pour superviser les dialogues USSD et éviter les dialogues interrompus, il a été défini une politique de gestion de la durée des opérations dans le réseau.

I.6.3.1- ProcessUSSDRequest Invoke Timer

L'horloge est démarrée lorsque la méthode **ProcessUSSDRequest** est reçue par le réseau (Dialogue initié par le MS). Il est stoppé lorsque le MS reçoit le résultat de sa requête (Libération du dialogue.). Cette horloge limite la durée totale de dialogue, sa valeur est comprise entre 1 et 10 minutes.

I.6.3.2-USSDRequest Invoke Timer

L'horloge est démarrée lorsque la méthode **USSDRequest** est reçue par le réseau (Dialogue initié par le MS). Il est stoppé à la réception du résultat de la routine **USSDRequest** envoyé au MS (Le dialogue est libéré).

Cette horloge met une restriction à l'application MS qui traite le temps. Pour certaines applications, cela peut inclure l'obtention d'une réponse de l'utilisateur. Cette horloge limite la durée totale du dialogue, sa valeur est comprise entre 1 et 10 minutes.

I.6.4- Les dialogues multiples

Dans les spécifications GSM USSD, seul un dialogue entre le MS et le réseau est permis. Une fois le dialogue établi entre le MS et un nœud du réseau GSM, un autre dialogue ne peut être établi en parallèle. Cela veut dire qu'un hôte occupé qui ne peut pas être atteint par le nœud de la fin avec lequel le dialogue est établi, ne peut pas être atteint à tous sans avortement du dialogue établi en premier. Ainsi, l'établissement d'un nouveau dialogue vers un nœud différent permettra que le terminal puisse être atteint.

I.7-Aspects d'adressage

USSD a été conçu pour les dialogues entre le MS et une application USSD dans le MSC, VLR ou HLR. Le MSISDN est transporté dans la partie du dialogue du message TCAP. Par exemple, quand un dialogue initié par le MS est établi vers une application dans le HLR, le MSISDN et l'adresse HLR sont inclus. Pour un mobile initiant un dialogue, l'application USSD dans le HLR n'est pas probablement la dernière.

Elle travaillera comme un relais seulement, et laissera passer des opérations USSD entre le réseau GSM et le nœud externe.

I.8-Les commande GSM normalisées :

Il s'agit de commandes standardisées au niveau mondial, et donc utilisables sur tous les réseaux GSM / UMTS et sur tous les mobiles.

La quasi-totalité de ces codes n'a plus d'utilité sur les mobiles récents qui comportent toutes les commandes équivalentes dans leur menu. Toutefois, il arrive que certaines commandes ou paramètres manquent. Par exemple, les Samsung Galaxy ne permettent pas de régler le délai de renvoi sur non-réponse. Enfin, sur certains mobiles, les menus sont tellement touffus qu'il est parfois plus rapide de composer la commande au clavier plutôt que de s'entêter à la trouver dans les menus.

➤ Obtenir le numéro IMEI de son mobile :

Chaque terminal mobile GSM possède un numéro d'identification appelé **IMEI** (**I**nternational **M**obile station **E**quipment **I**ntity). Ce numéro qui est indiqué sur l'appareil (souvent sous la batterie) est aussi stocké dans les mémoires du terminal.

***#06#** pour afficher le numéro **IMEI** du mobile.

➤ Gérer les codes de la carte SIM :

Changer le code PIN : ****04*ancienPIN*nouveauPIN*nouveauPIN#**

Débloquer le code PIN : ****05*PUK*nouveauPIN*nouveauPIN#**

Le code PIN est le mot de passe permettant d'utiliser la carte SIM sur le mobile. Sur le réseau Orange, il est fixé par défaut à 0000 lors de la livraison de la carte. Le code PUK est fourni sur le support de la carte à la livraison ou dans l'espace client Web.

Changer le code PIN 2 : ****042*ancienPIN2*nouveauPIN2*nouveauPIN2#**

Débloquer le code PIN 2 : ****052*PUK2*nouveauPIN2*nouveauPIN2#**

Le code PIN2 sert à accéder aux services supplémentaires de la carte SIM, principalement les restrictions d'appels. Sur le réseau Orange, le code PIN2 n'est fourni que si vous avez souscrit à des options de restriction d'appels spécifiques.

➤ Gérer le signal d'appel :

Activer : ***43#**

Désactiver : **#43#**

Consulter l'état : ***#43#**

Le signal d'appel permet de gérer simultanément une communication établie et un appel entrant. Mais attention ! Il a aussi pour effet de désactiver le renvoi d'appel sur occupation qui est remplacé par le renvoi d'appel sur non-réponse.

➤ Masquer son numéro de ligne lors des appels sortants :

Le numéro de l'appelant est par défaut toujours affiché, le masquage du numéro est possible en composant le code USSD **#31#** numéro appelé.

Conclusion

Cette étape nous a permis de comprendre la prépondérance du protocole USSD dans sa structure, son utilisation et surtout son fonctionnement dans le réseau GSM ainsi que le mode de fonctionnement de ce dernier.

Dans l'application de ce projet, un terminal mobile servant de modem sera connecté au serveur. C'est dans cette perspective que nous allons aborder dans le chapitre suivant les réseaux Internet-Intranet et éventuellement l'étude du protocole TCP/IP.

Introduction

Les réseaux Internet-intranet sont des réseaux qui utilisent l'architecture TCP/IP dans un cadre privé. Dans ce chapitre on va représenter les deux modèles OSI et TCP/IP dans leurs généralités puis introduire les caractéristiques des réseaux Internet-Intranet.

II.1- Modèle de référence OSI (Open Systems Interconnexion) [3]

Les données à transmettre d'une machine à une autre sont fragmentées à l'émission en petit blocs de quelques centaines d'octets munis de l'adresse du destinataire, envoyées sur le réseau et réassemblées à la réception pour reproduire les données d'origine.

Ce concept facilite le partage des possibilités physiques des réseaux (bande passante) et est parfaitement adapté pour une implémentation sur machines séquentielles travaillant en temps partagé (plusieurs communications peuvent alors avoir lieu simultanément et sur une même machine).

Partant de ce concept, un modèle d'architecture pour les protocoles de communication a été développé par l'ISO (International Standards Organisation) entre 1977 et 1984. Ce modèle sert souvent de référence pour décrire la structure et le fonctionnement des protocoles de communication, mais n'est pas une contrainte de spécification.

Ce modèle se nomme OSI comme "Open Systems Interconnection Reference Model".

Les constituants de ce modèle sont si largement employés qu'il est difficile de parler de réseaux sans y faire référence.



Figure II.1- Modèle OSI

Chapitre II Réseaux Internet-intranet

Niveau 7: couche application, gère le transfert des informations entre programmes.

Niveau 6: couche présentation, s'occupe de la mise en forme des données, éventuellement l'en cryptage et la compression des données, par exemple mise en forme des textes, images et vidéo.

Niveau 5: la couche session, s'occupe de l'établissement, de la gestion et coordination des communications.

Niveau 4: la couche transport, gère la remise correcte des informations (gestion des erreurs), utilise notamment l'**UDP** (User Datagram Protocol) et le **TCP/IP** (Transmission Control Protocol / Internet Protocol).

Niveau 3: la couche réseau, détermine les routes de transport et s'occupe du traitement et du transfert de messages: gère IP et ICMP.

Niveau 2: la couche liaison de données, définit l'interface avec la carte réseau: hubs, switch, etc....

Niveau 1: la couche physique, gère les connections matérielles, définit la façon dont les données sont converties en signaux numériques.

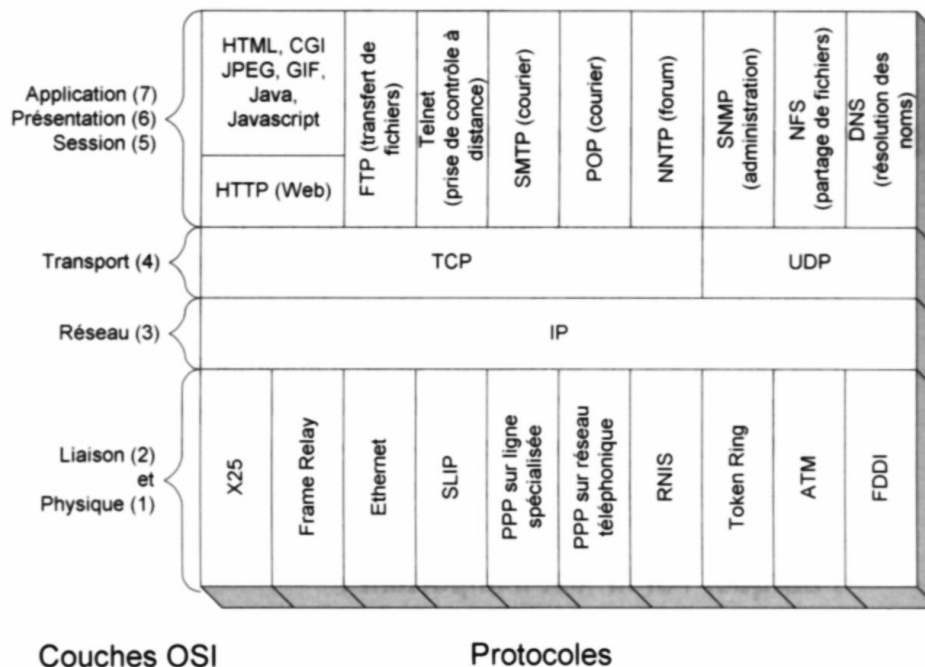


Figure II.2- Les couches du modèle OSI et les principaux protocoles

II.2-Modèle TCP/IP (Transmission Control Protocol / Internet Protocol) [4]

II.2.1- Historique des protocoles TCP/IP

Au milieu des années 70, les travaux de DARPA (Defence Advanced Research Project Agency), Débutent pour développer un réseau à commutation de paquets afin de relier ses centres de recherches, partager des équipements informatiques et échanger données et courriers. Le réseau devait résister à des attaques militaires.

Il fallait donc protéger les points névralgiques dont la destruction entraînerait la paralysie du réseau. Ainsi, dès le départ le réseau ARPANET est-il conçu sans nœud particulier le dirigeant, de telle sorte que, si une voie de communication venait à être rompue, le réseau continuerait d'acheminer les informations en empruntant un autre chemin.

Le DARPA établit donc un certain nombre de normes, spécifiant les principes et conventions de communications entre ordinateurs, qui sont souvent référencées sous l'appellation TCP/IP.

C'est vers 1980 qu'apparaît le réseau Internet tel qu'on le connaît maintenant.

La DARPA commence à faire évoluer les ordinateurs de ses réseaux de recherche vers les nouveaux protocoles TCP/IP et elle se met à subventionner les travaux de l'université de Berkeley pour qu'elle intègre TCP/IP à son système d'exploitation Unix. Ainsi, la quasi-totalité des départements d'informatique des universités américaines commencent-ils à se doter de réseaux locaux qui, sous l'impulsion de la NSF (National Science Foundation), seront interconnectés, en Quelques années. Les normes Internet sont publiées sous forme de fichiers texte, appelés RFC (Request for Comments) et téléchargeables gratuitement. Chaque RFC reçoit un numéro unique et définitif, donc non modifiable. De ce fait, plusieurs RFC peuvent traiter du même sujet, mais certaines d'entre elles étant obsolètes ou simplement complémentaires.

Même si, dès son origine Internet comprenait des sociétés privées, celles-ci étaient plus ou moins liées à la recherche et au développement, mais depuis l'arrivé du Web, en 1993, les activités commerciales s'y multiplient considérablement.

II.2.2- Avantages des protocoles TCP/IP

Les protocoles TCP/IP présentent certains avantages. Nous en retenons les plus significatifs:

- ✓ TCP/IP est indépendant de tout matériel ou constructeur. Il peut fonctionner sur une grande variété de technologies, utilisant une unité de transmission nommée datagramme, spécifiant la façon de transmettre les informations sur un type de réseau donné.
- ✓ Chaque machine connectée via TCP/IP reçoit une adresse unique et toutes les paires de machines peuvent communiquer entre elles.

Les nœuds intermédiaires utilisent les adresses contenues dans les datagrammes pour prendre les décisions de routage des paquets.

- ✓ TCP/IP assure un acquittement direct entre machine source et destination, même dans l'hypothèse où les machines ne sont pas reliées sur le même réseau physique.
- ✓ Outre les protocoles de transport, TCP/IP inclut différents protocoles d'application, par exemple, pour la messagerie électronique, le transfert de fichiers et la connexion à distance.

Le modèle TCP/IP reprend l'approche modulaire (utilisation de modules ou couches) mais en contient uniquement quatre :

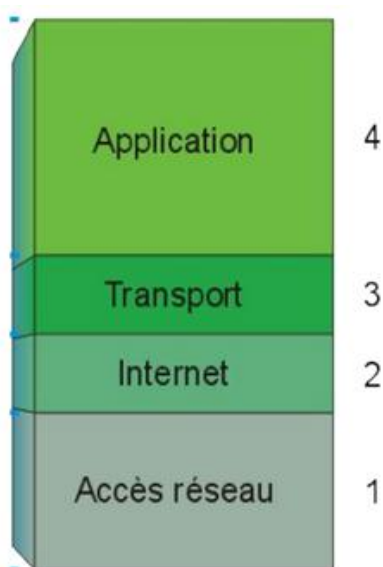


Figure II.3-Modèle TCP/IP

II.2.2.1-Couche application

La Couche Application englobe les applications standards du réseau:

- ✓ **SMTP** (Simple Mail Transport Protocol) : gestion des mails.
- ✓ **Telnet**: protocole permettant de se connecter sur une machine distante (serveur) en tant qu'utilisateur.
- ✓ **FTP** (File Transfert Protocol) : protocole permettant d'échanger des fichiers via Internet et d'autres moins courants.

II.2.2.2-Couche transport

La Couche transport assure l'acheminement des données et les mécanismes permettant de connaître l'état de la transmission. Les protocoles des couches suivantes permettent d'envoyer des informations d'une machine à une autre. La couche transport permet d'identifier les applications qui communiquent. Pour faciliter la communication, on a défini non pas des noms d'applications, mais des ports de communication spécifiques à chaque application.

La couche transport gère deux protocoles de livraison des informations, indépendamment du type de réseau emprunté:

- ✓ **TCP (Transmission Control Protocol):** un protocole orienté connexion (assure le control de transmission de données), et responsable de l'établissement de la connexion.
- ✓ **UDP (User Datagram Protocol):** archaïque et non orienté connexion, n'assure aucun contrôle de transmission des données.

II.2.2.3- Couche Internet

La couche Internet est chargée de fournir le paquet des données. Elle définit les datagrammes et gère la décomposition / recomposition des segments.

La couche Internet contient 5 protocoles (les 3 premiers sont les plus importants):

- ✓ **Le protocole IP (Internet Protocol):** gère les destinations des messages, adresse du destinataire
- ✓ **Le protocole ARP (Adresse Resolution Protocol):** gère les adresses des cartes réseaux. Chaque carte a son propre adresse d'identification codée sur 48 bits.
- ✓ **Le protocole RARP (Reverse Address Resolution Protocol) :** gère l'adresse IP pour les équipements qui ne peuvent s'en procurer une par lecture d'information dans un fichier de configuration. En effet, lorsqu'un PC démarre, la configuration réseau lit l'adresse IP qu'elle va utiliser. Ceci n'est pas possible dans certains équipements qui ne possèdent pas de disques durs (terminaux essentiellement).
- ✓ **Le protocole ICMP (Internet Control Message Protocol) :** gère les informations relatives aux erreurs de transmission. ICMP ne corrige pas les erreurs, mais signale aux autres couches que le message contient des erreurs.
- ✓ **Le protocole IGMP (Internet Group Management Protocol) :** permet d'envoyer le même message à des machines faisant partie d'un groupe. Ce protocole permet également à ces machines de s'abonner ou de se désabonner d'un groupe. Ceci est utilisé par exemple dans la vidéo conférence à plusieurs machines, envoi de vidéos, ... Ce protocole permet de regrouper des stations.

II.2.2.4- Couche Accès réseau

La couche Accès réseau spécifie la forme sous laquelle les données doivent être acheminées, quel que soit le type de réseau utilisé. Elle prend en charge les notions suivantes:

- Acheminement des données sur la liaison
- Coordination de la transmission de données (synchronisation)
- Format des données
- Conversion des signaux (analogique/numérique) pour les modems RTC
- Contrôle des erreurs à l'arrivée.

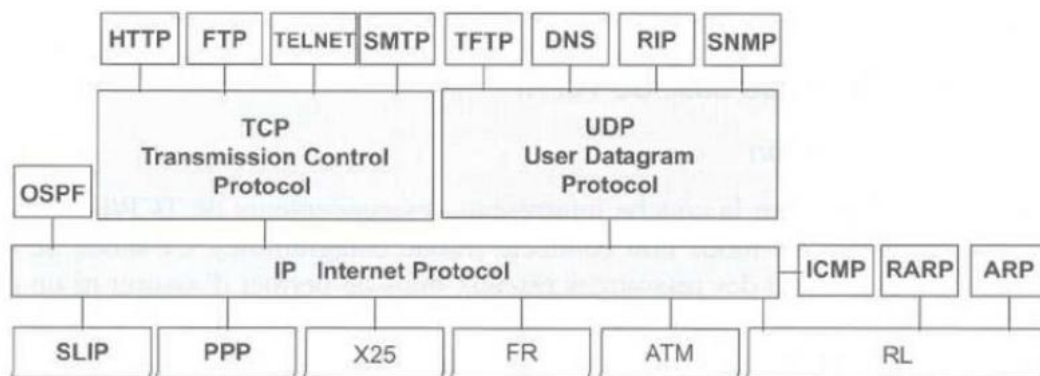


Figure II.4-Protocoles et application TCP/IP

II.3-Comparaison TCP/IP-OSI

La suite de protocoles désignée par TCP/IP, ou encore “ pile ARPA ”, est construite sur un modèle en couches moins complet que la proposition de l’ISO. Quatre couches sont suffisantes pour définir l’architecture de ce protocole.

- 4 Couche Application (Application layer).
- 3 Couche Transport (Transport layer).
- 2 Couche Internet (Internet layer).
- 1 Couche interface réseau (Network access layer).

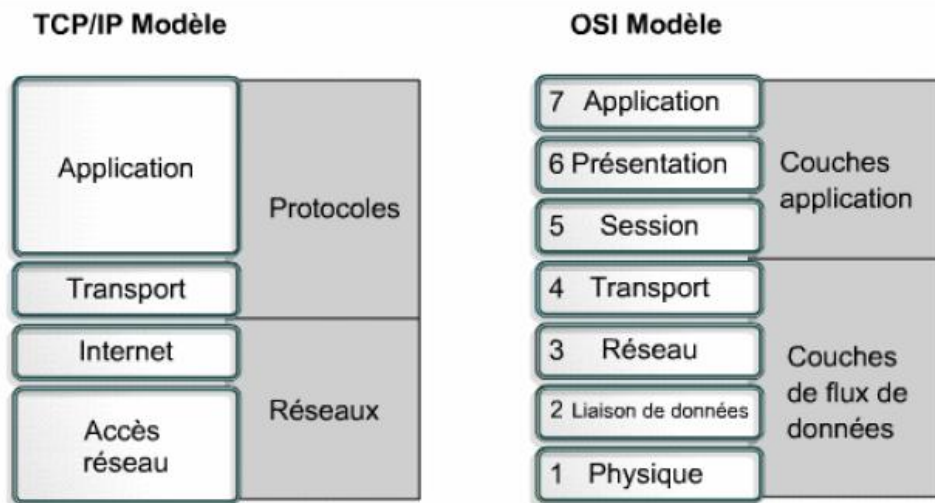


Figure II.5- Comparaison OSI-TCP/IP

II.4-Adressage IP [5]

IP signifie "**I**nternet **P**rotocol", protocole Internet. Il représente le protocole réseau le plus répandu. Il permet de découper l'information à transmettre en paquets, de les adresser, de les transporter indépendamment les uns des autres et de recomposer le message initial à l'arrivée.

Chaque ordinateur qui utilise les protocoles TCP/IP doit disposer d'au moins une adresse Internet (adresses IP). Une adresse IP est constituée de 32 bits, regroupés pour des raisons de visibilité en quatre octets. Ces valeurs sont généralement exprimées sous forme de 4 chiffres décimaux séparés chacun par un point. Ce mode de représentation est appelé **DDN** (**D**otted **D**ecimal **N**otation).

Une adresse IP est constituée de deux parties:

- L'identifiant réseau (Net-ID).
- L'adresse de l'ordinateur (Host-ID).

Ce découpage a été réalisé pour faciliter l'adaptation aux unités d'organisation

L'identifiant réseau permet de désigner l'unité d'organisation, l'adresse de l'ordinateur et de préciser de quel ordinateur il s'agit à l'intérieur de l'unité. Pour créer ces unités, des classes de réseau ont été définies, pour offrir aux administrateurs systèmes la possibilité d'affecter des adresses à des unités organisationnelles. Les adresses Internet sont réparties dans des classes d'adresses de A à E, dont seules les classes de A à C sont disponibles pour l'adressage normal. Les parties de l'adresse séparées par un point n'indiquent pas quelle partie de l'adresse constitue l'identifiant réseau et laquelle désigne l'adresse de l'ordinateur. Les logiciels réseau identifient la classe d'adresses au contenu des 4 premiers bits du premier octet de l'adresse IP.

II.5- DNS (Domain Name System)

Chaque ordinateur directement connecté à internet possède au moins une adresse IP propre. Cependant, les utilisateurs ne veulent pas travailler avec des adresses numériques du genre *194.153.205.26* mais avec un nom de domaine ou des adresses plus explicites (appelées adresses FQDN) du type [www.univ-tlemcen.dz].

Ainsi, il est possible d'associer des noms en langage courant aux adresses numériques grâce à un système appelé **DNS** (*Domain Name System*).

On appelle *résolution de noms de domaines* (ou *résolution d'adresses*) la corrélation entre les adresses IP et le nom de domaine associé.

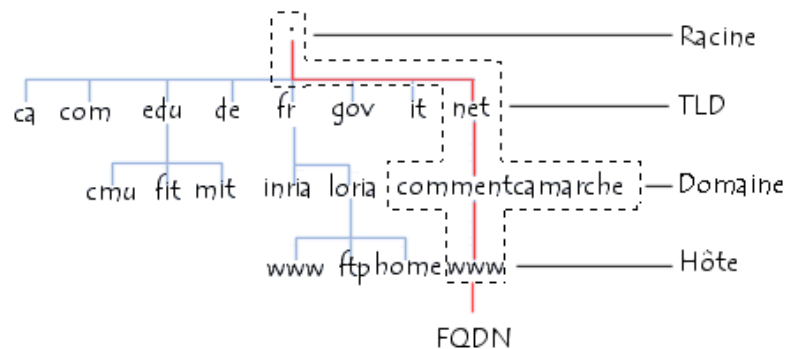


Figure II.8-Structure hiérarchique de la DNS

- ✓ **Racine** : elle est sous-entendue et n'est jamais indiquée dans les FQDN, mais il existe, au même titre que le "\" dans l'arborescence Windows ou le "/" dans celle d'Unix.
- ✓ **TLD (Top Level Domains)** : Ce sont les domaines de niveau supérieurs, caractérisant principalement les pays qui sont indiqués par deux lettres, comme Fr pour la France, et les organisations fonctionnelles comme Gouv ou net.
- ✓ **Domaine** : Contient les noms de domaines réservés par les gestionnaires de sites. Ces noms de domaines doivent être déposés auprès de l'organisme responsable du domaine.
- ✓ **Hôte** : Correspond à une machine ou une entité du réseau.
- ✓ **FQDN (Fully Qualified Domain Name)**: en français « nom de domaine complètement qualifié » Il constitue le nom absolu correspondant à l'ensemble des étiquettes des nœuds d'une arborescence, séparées par des points, et terminé par un point final. L'adresse **FQDN** permet de repérer de façon unique une machine sur le réseau des réseaux. par exemple *www.univ-tlemcen.dz* représente une adresse FQDN.

II.6-Protocoles logiciels

Les protocoles logiciels utilisés sur internet sont les conventions structurant les échanges d'informations nécessaires au transfert des contenus applicatifs pour l'utilisateur final. Ils permettent notamment d'identifier les interfaces (et donc les machines), de s'assurer de la réception des données envoyées, et de l'interopérabilité.

Internet fonctionne suivant un modèle en couches, similaire au modèle **OSI**. Les éléments appartenant aux mêmes couches utilisent un protocole de communication pour s'échanger des informations.

Un protocole est un ensemble de règles qui définissent un langage afin de faire communiquer plusieurs ordinateurs. Ils sont définis par des normes ouvertes, les RFC.

Chaque protocole a des fonctions propres et, ensemble, ils fournissent un éventail de moyens permettant de répondre à la multiplicité et à la diversité des besoins sur Internet.

Les principaux sont les suivants, classés selon leur couche (**IP**, **TCP** et **UDP**), couches applicatives :

- ✓ **IP** (*Internet Protocol*) aussi appelé IPv4 : protocole réseau qui définit le mode d'échange élémentaire entre les ordinateurs participants au réseau en leur donnant une adresse unique sur le réseau.
Mais en vu du nombre d'internaute qui est de plus en plus grand une nouvelle norme voit le jour appelé IPv6 qui permet d'accueillir un plus grand nombre d'utilisateurs.
- ✓ **TCP** : responsable de l'établissement de la connexion et du contrôle de la transmission. C'est un protocole de remise fiable. Il s'assure que le destinataire a bien reçu les données, au contraire d'UDP.
 - ✓ **HTTP** (*HyperText Transfer Protocol*) : protocole mis en œuvre pour le chargement des pages web.
 - ✓ **FTP** (*File Transfer Protocol*) : protocole utilisé pour le transfert de fichiers sur Internet.
 - ✓ **SMTP** (*Simple Mail Transfer Protocol*) : mode d'échange du courrier électronique en envoi.
 - ✓ **POP3** (*Post Office Protocol Version 3*) : mode d'échange du courrier électronique en réception.
 - ✓ **IMAP** (*Internet Message Access Protocol*) : un autre mode d'échange de courrier électronique.
 - ✓ **IRC** (*Internet Relay Chat*) : protocole de discussion instantanée.
 - ✓ **NNTP** (*Network News Transfer Protocol*) : protocole de transfert de message utilisé par les forums de discussion.

- ✓ **SSL** ou **TLS** : protocoles de transaction sécurisée, utilisés notamment pour le paiement sécurisé.
- ✓ **UDP (User Datagram Protocol)** : permet de communiquer, de façon non fiable mais légère, par petits datagrammes.
- ✓ **DNS (Domain Name System)** : système de résolution de noms Internet.
- ✓ **ICMP (Internet Control Message Protocol)** : protocole de contrôle du protocole IP.

II.7-Intranet

Un **intranet** est un ensemble de services internet (par exemple un serveur web) internes à un réseau local, c'est-à-dire accessibles uniquement à partir des postes d'un réseau local, ou bien d'un ensemble de réseaux bien définis, et invisibles (ou inaccessibles) de l'extérieur. Il consiste à utiliser les standards client-serveur de l'internet (en utilisant les protocoles TCP/IP), comme par exemple l'utilisation de navigateurs internet (client basé sur le protocole HTTP) et des serveurs web (protocole HTTP), pour réaliser un système d'information interne à une organisation ou une entreprise.

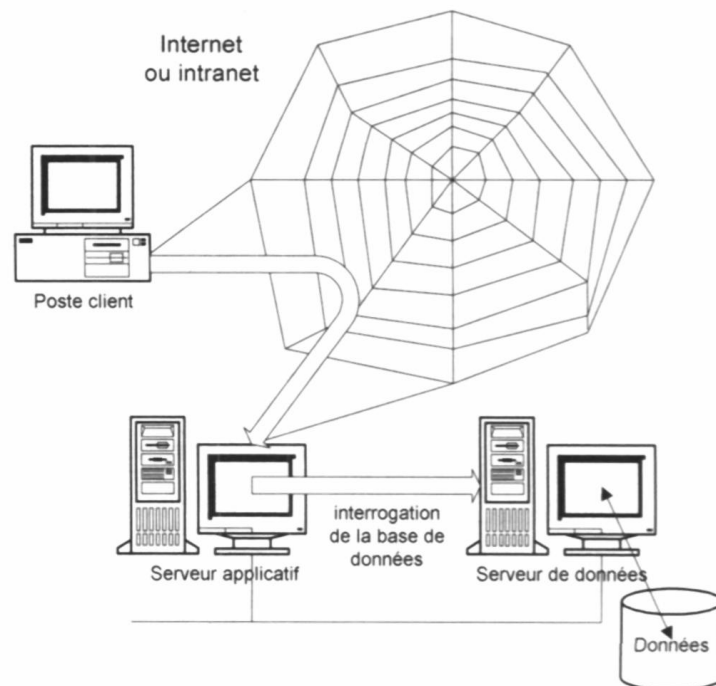


Figure II.9-Architecture intranet

Un intranet repose généralement sur une architecture à trois niveaux, composée :

- de clients (navigateur internet généralement).
- D'un ou plusieurs serveurs d'application (middleware): un serveur web permettant d'interpréter des scripts CGI, PHP, ASP ou autres, et les traduire en requêtes SQL afin d'interroger une base de données.
- D'un serveur de bases de données.

Conclusion

L'étude de ce chapitre nous a permis de comprendre les principes de base de fonctionnement du protocole TCP/IP qui sont primordiaux dans les réseaux Internet-intranet. L'acquis de ces notions nous permet d'aborder en mieux l'implémentation des serveurs dans le prochain chapitre.

Introduction

Dans le cadre de notre projet, plusieurs outils logiciels et langages de programmation ont été exploités allant du langage HTML au langage PHP, le gestionnaire des bases de données MySQL Administrator et le serveur web Apache. Ce chapitre, est consacré à la présentation de ces différents outils que nous avons exploités.

III.1-Serveur [5]

Un serveur est une machine généralement très puissante en termes de capacités d'entrée-sortie, qui fournit aux ordinateurs fonctionnant sur son environnement des services. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, etc.

III.1.1-Serveur Web

Un serveur Web est un logiciel permettant aux clients d'accéder à des pages Web, c'est-à-dire en réalité des fichiers au format HTML à partir d'un navigateur (aussi appelé browser) installé sur leur ordinateurs distant.

Les principaux serveurs Web sur le marché sont entre autres :

- Apache (qui a l'avantage d'être gratuit)
- Microsoft IIS (Internet Information Serveur)
- Microsoft PWS (Personal Web Server)
- Xitami
- Netscape Entreprise Server
- Borland IntraBuilder

Les serveurs web peuvent remplir plusieurs autres fonctionnalités :

- Générer des pages PHP en HTML.
- Servir une page en ajoutant des entêtes appropriés.
- Reconnaître les entêtes de requêtes et servir des pages différentes suivant le client ou ses préférences.
- Répondre différemment aux différentes machines grâce aux adresses IP et aux ports de celles-ci.

III.1.2-Serveur Apache

Apache est le type de serveur le plus utilisé au monde. C'est un logiciel libre, qui permet de construire et de tester un site sur son propre ordinateur, sans hébergement extérieur.

Apache est le serveur le plus répandu sur Internet. Il s'agit d'une application fonctionnant à la base sur les systèmes d'exploitation de type Unix, mais il a désormais été porté sur de nombreux systèmes, dont Microsoft Windows.

Apache est le fruit d'une multitude de correctifs logiciels afin d'en faire une solution très sûre. Ainsi, dès qu'un bug ou une faille de sécurité est décelée, ceux-ci sont rapidement corrigés par le serveur et une nouvelle version de l'application est éditée.

Apache possède désormais de nombreuses fonctionnalités dont la possibilité de définir une configuration spécifique à chaque fichier ou répertoire partagé, ainsi que de définir des restrictions d'accès grâce aux fichiers .htaccess.

III.2-le navigateur Internet

Le rôle d'un navigateur Internet est de permettre à un utilisateur de consulter le contenu d'un serveur Web. Pour ce faire, le navigateur communique avec le serveur Web, charge une copie de son contenu en local puis interprète les pages au format HTML.

La communication s'établit selon le protocole TCP/IP qui est chargé du routage de l'information d'un site à un autre, et le transit des informations (principalement des pages HTML) s'effectue selon le protocole HTTP qui définit comment les données sont transmises entre le client et le serveur via TCP/IP.

Le Web est donc un ensemble de serveurs connectés à l'Internet et proposant des ressources. L'utilisateur qui accède à ces ressources utilise en général un type particulier de programme client, le navigateur. Les deux principales tâches d'un navigateur consistent à :

- ✓ Dialoguer avec un serveur.
- ✓ Afficher à l'écran les documents transmis par un serveur.

Les navigateurs les plus utilisés sont Internet Explorer, Google chrome et Mozilla Firefox.

III.2.1-Protocole HTTP

Les navigateurs et les serveurs Web utilisent le protocole HTTP (**H**yper**T**ext **T**ransfer **P**rotocol, ou protocole de transfert hypertexte), qui définit précisément comment un navigateur doit formater et envoyer une requête à un serveur Web. Le navigateur client envoie une requête de document constitué d'une ligne de caractères ASCII. Une requête de document se compose du mot GET, suivi d'un espace typographique, puis d'une adresse spécifiée par rapport au répertoire racine du serveur Web. Ce répertoire racine est désigné lors de la configuration du serveur (ou du site) Web par les répertoires du système de fichiers de l'ordinateur hôte. C'est ce répertoire qui permet de retrouver les documents. HTTP est un protocole dit en mode non connecté car, dès que la requête est effectuée puis satisfaite, la connexion entre le client et le serveur est terminée. C'est le serveur qui rompt la connexion une fois que tout le document a été transféré. De son côté le client peut interrompre le transfert en cours de route en mettant fin à la connexion. HTTP s'exécute au-dessus du protocole TCP mais pourrait le faire au-dessus de tout autre service de connexion.

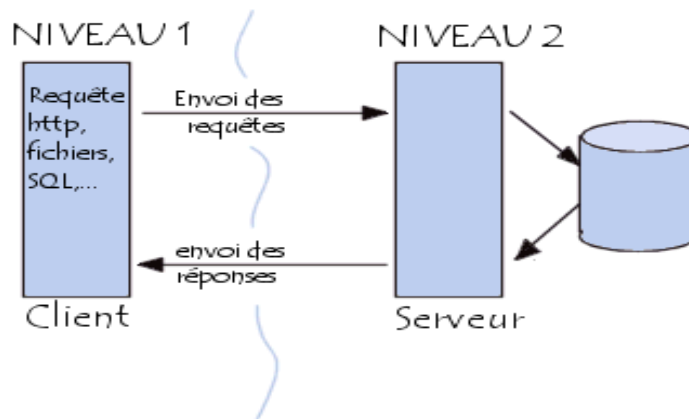


Figure III.1 : Schéma du protocole HTTP

III.2.1.1- Requêtes HTTP

La requête se compose d'une ligne initiale qui correspond à la demande du client et cette ligne est composée de trois parties :

- ✓ Méthodes
- ✓ URL
- ✓ Protocole utilisé (en général http/1.0 ou http/1.1).

Méthodes Elles indiquent le type de la requête, les méthodes les plus utilisées sont GET et POST.

La méthode **GET** est utilisée pour obtenir une ressource, elle est plus simple car le corps du message est vide.

La méthode **POST** permet d'envoyer des informations au serveur dans le corps d'une requête HTTP.

URL (Uniform Ressource Locator) : localisateur uniforme de ressource syntaxe utilisé par le World Wide Web pour spécifier la localisation physique d'un fichier ou d'une ressource sur Internet.

Protocole HTTP : permet de visualiser les pages web

III.3-Langage HTML

Le **HTML (Hyper Text Mark-Up Language)** est un langage 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.

Le langage HTML permet notamment la lecture de documents sur Internet à partir de machines différentes, grâce au protocole HTTP, permettant d'accéder via le réseau à des documents repérés par une adresse unique, appelée URL (Uniform Ressource Locator).

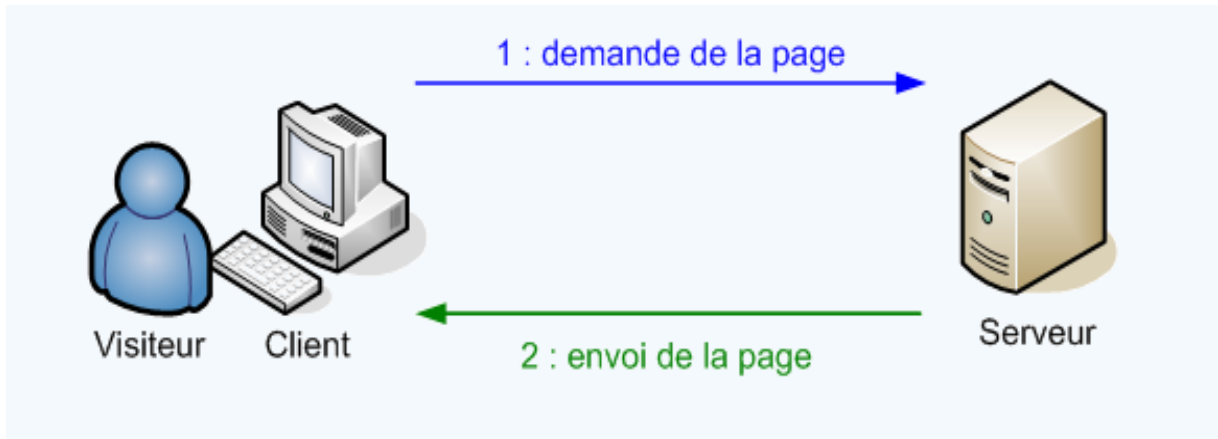


Figure III.2- Requêtes et réponses avec HTML

III.4-Feuilles de style en cascade (CSS)

Les feuilles de styles (en anglais "*Cascading Style Sheets*", abrégé CSS) sont un langage qui permet de gérer la présentation d'une page Web. Le langage CSS est une recommandation du World Wide Web Consortium (W3C), au même titre que HTML ou XML.

Les styles permettent de définir des règles appliquées à un ou plusieurs documents HTML. Ces règles portent sur le positionnement des éléments, l'alignement, les polices de caractères, les couleurs, les marges et espacements, les bordures, les images de fond, etc.

Le but de CSS est de séparer la structure d'un document HTML et sa présentation. En effet, avec HTML, on peut définir à la fois la structure (le contenu et la hiérarchie entre les différentes parties d'un document) et la présentation. Mais cela pose quelques problèmes. Avec le couple HTML/CSS, on peut créer des pages web où la structure du document se trouve dans le fichier HTML tandis que la présentation se situe dans un fichier CSS.

Avec CSS on peut par exemple définir un ensemble de règles stylistiques communes à toutes les pages d'un site internet. Cela facilite ainsi la modification de la présentation d'un site entier. CSS permet aussi de définir des règles différentes pour chaque support d'affichage (un navigateur classique, une télévision, un support mobile, un lecteur braille...). CSS permet aussi d'améliorer l'accessibilité des documents web.

De plus, CSS ajoute des fonctionnalités nouvelles par rapport à HTML au point de vue du style. En effet, HTML permet une gestion assez sommaire du style des documents.

Les feuilles de style permettent notamment :

- d'obtenir une présentation homogène sur tout un site en faisant appel sur toutes les pages à une même définition de style.

- de permettre le changement de l'aspect d'un site complet entier par la seule modification de quelques lignes.
- une plus grande lisibilité du HTML, car les styles sont définis à part.
- des chargements de page plus rapides, pour les mêmes raisons que précédemment.
- un positionnement plus rigoureux des éléments.

III.5- Langage PHP [6]

PHP est un langage interprété (un langage de script) exécuté du côté serveur et non du côté client. PHP est un acronyme récursif, qui signifie "*PHP: Hypertext Preprocessor*" : c'est un langage de script HTML. Sa syntaxe est empruntée aux langages C, Java et Perl, et est facile à apprendre. Le but de ce langage est de permettre aux développeurs web d'écrire des pages dynamiques rapidement.

III.5.1- Naissance de PHP

Initialement, PHP était une bibliothèque Perl rédigée par Rasmus Lerdorf en 1995.

Il s'en servait pour mettre à disposition son curriculum vitae sur Internet. Au fur et à mesure qu'il ajoutait des fonctionnalités Rasmus a transformé la bibliothèque Perl en une implémentation C. Il décida par la suite de partager son code pour que tout le monde puisse en profiter, ce qui attira les premiers contributeurs.

PHP/FI est le sigle de Personal Home Page/Form Interpreter. Cette première version Possédait déjà une syntaxe similaire à celle que nous connaissons aujourd'hui.

PHP/FI 2 publiée en novembre 1997, c'est la deuxième refonte en langage C de PHP.

À ce moment là, plusieurs milliers de personnes dans le monde utilisent déjà PHP et environ 50000 noms de domaines indiquaient utiliser PHP. La durée de vie de cette version aura été très courte avec l'arrivée de PHP 3.

PHP 3 fut une refonte complète initiée en 1997 par deux développeurs : Zeev Suraski et Andi Gutmans. La plate-forme telle que nous la connaissons actuellement est issue de cette version. Dans un effort de coopération et de compatibilité avec les anciennes versions de PHP/FI, Zeev, Andi et Rasmus décidèrent d'annoncer PHP 3 comme étant le successeur de PHP/FI.

À partir de ce moment, PHP change de nom. Il devient PHP Hypertext Preprocessor (acronyme récursif). Ce fut le signal de la publication d'une nouvelle plateforme qui n'est plus le projet d'un seul homme, mais d'une communauté.

En 1998, plusieurs dizaines de milliers d'utilisateurs et plusieurs centaines de milliers de sites font déjà confiance à PHP. PHP 3 a couvert environ 10 % du parc mondial de serveurs web.

PHP 4 a été initié durant l'hiver 1998. Andi Gutmans et Zeev Suraski décidèrent de réécrire

Le moteur interne de PHP afin d'améliorer les performances et la modularité du code.

Le nouveau moteur, baptisé Zend Engine (Zend est une combinaison de Zeev et Andi), a atteint haut la main son objectif.

La première version de PHP 4 a été publiée officiellement en mai 2000. Des performances plus élevées, une compatibilité avec de nombreux serveurs et plusieurs nouvelles fonctionnalités utiles ont contribué au succès de cette version.

PHP 4 a battu des records de popularité. Plusieurs millions de sites web indiquent qu'ils sont installés avec PHP et des centaines de milliers d'utilisateurs à travers le monde lui font confiance.

PHP 5 est sorti en juillet 2004. Il propose un nouveau moteur, Zend Engine II, optimisé pour les nouvelles fonctionnalités que nous lui connaissons, notamment l'approche objet.

III.5.2-Fonctionnement de PHP

La consultation de l'adresse d'un site (URL) depuis un browser permet au serveur de renvoyer une page. Lorsqu'il s'agit d'une page PHP se passe différemment car les scripts PHP sont directement exécutés sur le serveur, et la page renvoyée contient uniquement le code HTML.

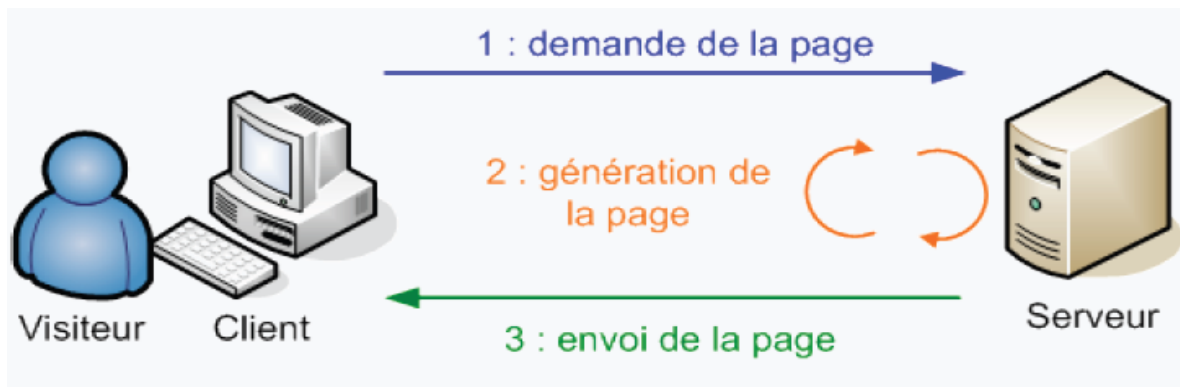


Figure III.3 Requêtes et réponses avec PHP

Voici les différentes étapes :

1. Le client demande à voir une page PHP. Pour lui il n'y a aucune différence. Il demande la page au serveur.
2. Mais là, il y a une étape très importante, qui fait toute la différence en PHP. Le serveur n'envoie pas de suite la page au client. Il la génère. En effet, le client n'est pas capable de lire une page PHP (seul le serveur sait faire ça). Le client ne peut lire que des pages HTML.
Ce que fait le serveur est simple : il va transformer la page PHP en page HTML, pour que le client puisse la lire
3. En fin une fois que la page est générée, elle ne contient plus que du code HTML. Le serveur peut l'envoyer au client.

III.5.3- Pages dynamiques et PHP

Lors du chargement d'une page PHP, c'est le serveur qui va lire, interpréter et exécuter le code. Puis il renvoie le résultat, généralement sous la forme de code HTML au navigateur. Ainsi le navigateur et l'utilisateur ne voient jamais le véritable code PHP exécuté. De plus le résultat étant une page web classique en HTML, pas besoin d'installer sur le client des composants spécifiques. Il n'y a donc pas de notion de vitesse d'exécution du côté du client, mais celle du serveur est prépondérante.

III.5.4- les garanties d'utilisation de PHP

PHP n'a pas de support technique commercial. En revanche, il dispose d'une alternative très efficace : une communauté ouverte et réactive de nombreux passionnés répartis à travers le monde.

Cette communauté est à l'écoute de tous les problèmes et de toutes les attentes des utilisateurs.

Si PHP 5 a su gagner son pari de combler les principaux reproches que l'on faisait à PHP 4 dans le monde professionnel, c'est grâce à la mobilisation de cette communauté de contributeurs.

À juste titre, les professionnels ont tendance à ne pas faire confiance aux programmes Open Source pour leur manque de garanties contractuelles. Mais il existe quelques applications d'exception dans lesquelles on a naturellement confiance. Le serveur Apache et PHP en font partie.

III.6-MySQL [7]

Le Système de Gestion de Bases de Données utilisé pour la gestion et le stockage des données dans le cadre de la mise en œuvre du prototype final est MySQL dans sa version 5.1. Le logiciel MySQL est un serveur de base de données SQL très rapide, multi-thread, multi utilisateurs et robuste.

MySQL est la base de données open source la plus populaire au monde. Bien qu'elle soit avant tout connue pour son utilisation par des sociétés Web, telles que Google, Facebook et Yahoo!, MySQL est également une base de données embarquée très populaire. Plus de 3000 éditeurs de logiciels et fabricants de matériel lui font confiance, parmi lesquels sept des dix plus grandes entreprises logicielles au monde.

Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble.

Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête.

Le SQL dans ``MySQL" signifie ``Structured Query Language" : le langage standard pour les traitements de bases de données.

Les principaux concurrents de MySQL sont : **PostgreSQL, Microsoft SQL Server, et Oracle**. Ainsi, le choix de ce Serveur de Bases de Données a été particulièrement orienté par un certain nombre d'avantages qu'il offre aux développeurs. En effet, par rapport aux autres SGBD cités, MySQL est un logiciel intégrant un haut degré de portabilité, de sécurité et constitue un système de sauvegarde assez évolué avec utilisation optimale de ressources. En plus c'est un outil fourni en open source.

III.6.1-MySQL Administrator :

MySQL Administrator est un utilitaire permettant de faciliter la gestion des bases de données MySQL sur un serveur web. Il permet d'effectuer toutes les tâches nécessaires à la mise en place et à la maintenance d'une base de données en l'occurrence MySQL :

- ✓ Ajout de tables et de données
- ✓ Modification de tables et de données
- ✓ Suppression de tables et de données
- ✓ Fonctionnalités dédiées aux sauvegardes et à l'optimisation
- ✓ Possibilité d'exécuter des requêtes SQL
- ✓ Tout cela avec un affichage simple et clair sous forme de listes et de tableaux.

Pour cette application, la version 5 MySQL Administrator a été adoptée.

Conclusion

Les différents outils logiciels présentés dans ce chapitre sont d'une grande importance pour l'implémentation des interfaces de notre serveur USSD. Le chapitre suivant sera consacré à l'intégration d'un tel serveur, nous allons représenter l'installation et la configuration de ces outils qui permettront aux utilisateurs de pouvoir naviguer sur le site.

Introduction

Dans ce chapitre on a focalisé sur la configuration et l'installation des serveurs : Serveur WEB, un gestionnaire CGI, Gestionnaire des bases de données, la mise en œuvre du portail web dédié a notre problème. On va terminer le présent chapitre par le modèle de la base de données proposée.

IV.1-Installation et configuration du serveur Apache sous Windows [8]

Dans ce projet, le serveur **Apache 2.2.9** a été adopté comme serveur WEB. L'installation d'Apache est facile, il suffit de suivre les instructions affichées sur l'écran.

Une fois l'installation achevée, il suffit de taper dans le navigateur, localement l'adresse <http://localhost/>, ou l'adresse IP à partir d'un hôte connecté au réseau, pour visualiser les pages web installées dans le serveur.

Pour la configuration d'Apache, il suffit de modifier le fichier **httpd.conf**.



Figure IV.1-Page d'accueil *du site*

IV.2-Installation de PHP [8]

Pour notre application la version **php-5.2.8** a été adoptée comme CGI (Common Gateway Interface), PHP a été installé là où est installé le serveur web Apache.

Après l'installation il est nécessaire de configurer le serveur Apache en y ajoutant les lignes suivantes :

```
LoadModule php5_module "C:/Program Files/PHP/php5apache2_2.dll"
```

```
AddType application/x-httpd-php .php
```

```
PHPIniDir "C:/Program Files/PHP"
```

Après cette étape il faudra configurer le fichier **php.ini** du CGI PHP en modifiant les lignes suivantes :

```
extension_dir = "./" En extension_dir="C:\Program Files\PHP\ext"
```

```
;upload_tmp_dir = En upload_tmp_dir = C:/ Program Files\PHP \uploadtemp
```

```
;session.save_path ="/tmp" En session.save_path="C:/Program Files\PHP \sessionsave"
```

Enfin enlever le commentaire de la ligne suivante :

```
;extension=php_mysql.dll En extension=php_mysql.dll
```

Pour vérifier que l'installation et la configuration de PHP ont réussi, il suffit de créer un fichier **info.php** comme page web. Le fichier **info.php** contient la ligne suivante :

```
<html> <body> <?php phpinfo(); ?> </body></html>
```

On obtient la page suivante :

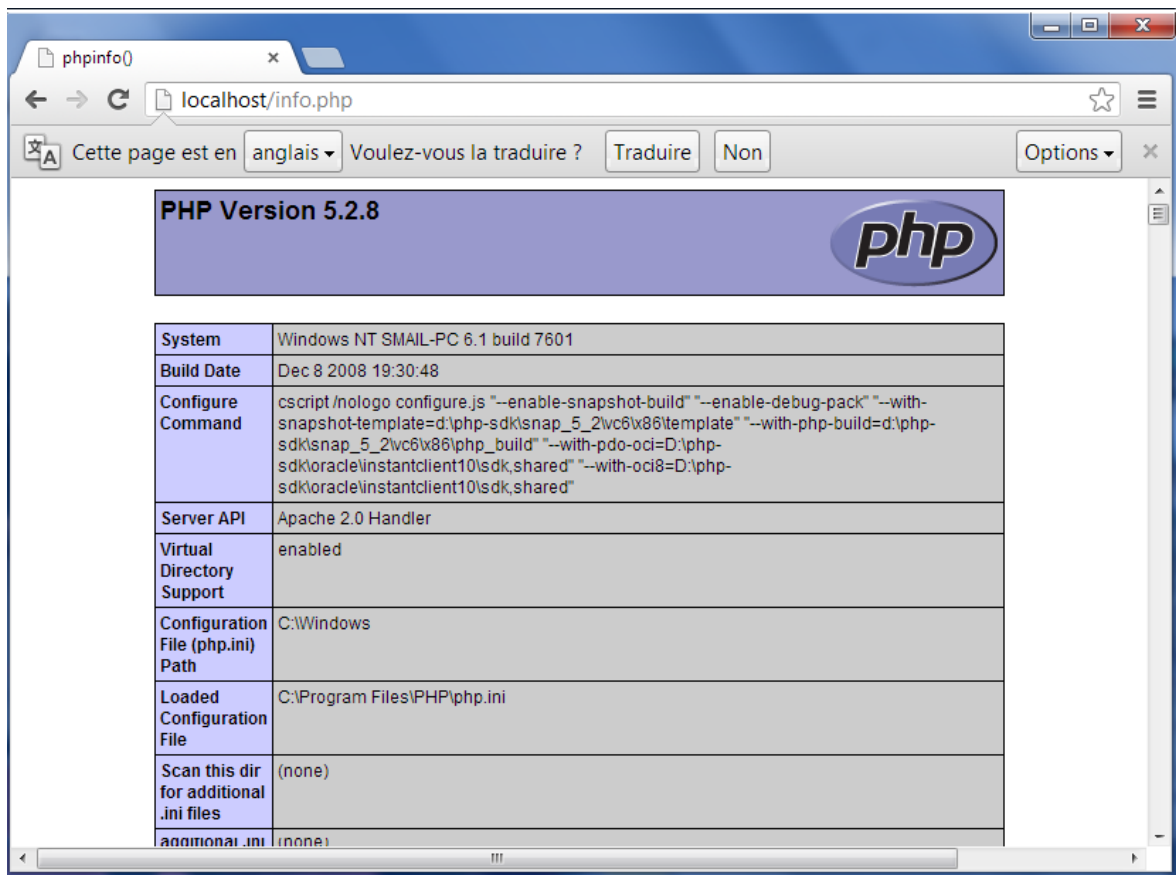


Figure IV.2- Page de configuration de PHP

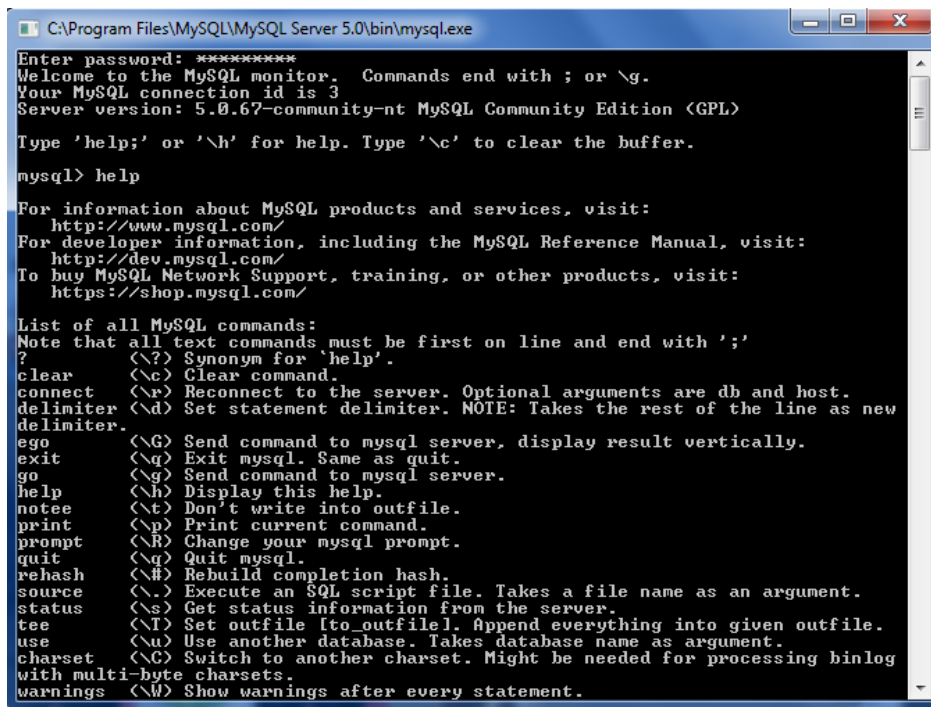
Cette page confirme la bonne prise en compte de PHP par Apache.

IV.3- Installation et lancement de MySQL sous Windows [8]

MySQL pour Windows est disponible gratuitement en téléchargement sur le site officiel *mysql.com* et son installation n'est pas compliquée, il suffit de lire les instructions à l'écran. Pour notre application, c'est la version **mysql-essential-5.0.67** qui a été utilisée. Le serveur MySQL ayant été installé en tant que service, ce dernier doit être dès cet instant en action.

Pour vérifier que le serveur MySQL fonctionne bien, on peut se connecter au Shell de ce dernier.

En tapant `help mysql> help` on obtient la liste des commandes sur l'écran ci-dessous :



```
C:\Program Files\MySQL\MySQL Server 5.0\bin\mysql.exe
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.0.67-community-nt MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

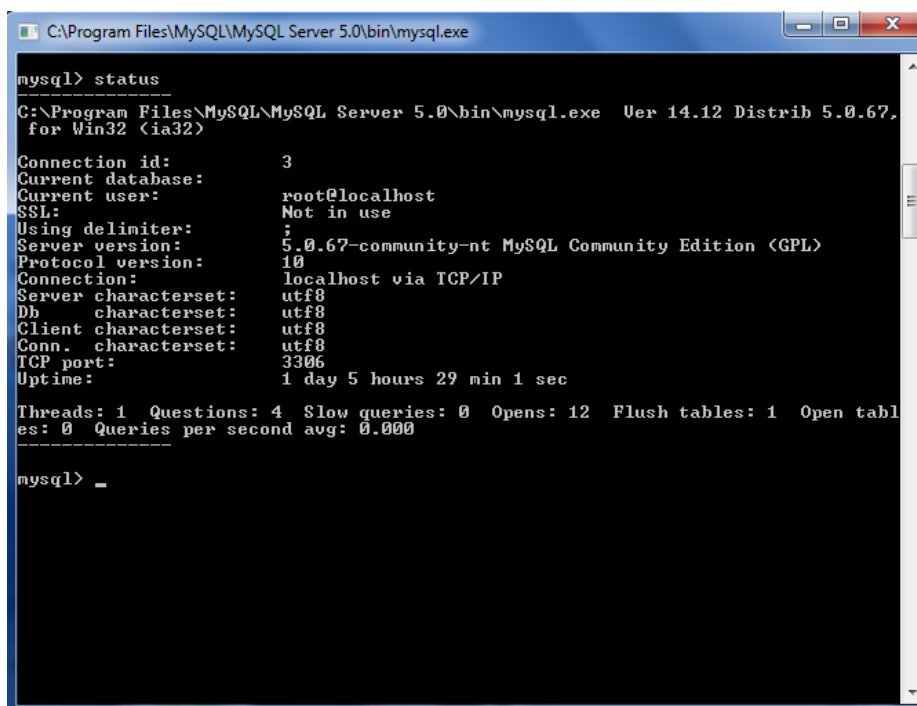
mysql> help

For information about MySQL products and services, visit:
  http://www.mysql.com/
For developer information, including the MySQL Reference Manual, visit:
  http://dev.mysql.com/
To buy MySQL Network Support, training, or other products, visit:
  https://shop.mysql.com/

List of all MySQL commands:
Note that all text commands must be first on line and end with ';'
?          (\?) Synonym for 'help'.
clear      (\c) Clear command.
connect    (\r) Reconnect to the server. Optional arguments are db and host.
delimiter (\d) Set statement delimiter. NOTE: Takes the rest of the line as new
delimiter.
ego        (\G) Send command to mysql server, display result vertically.
exit       (\q) Exit mysql. Same as quit.
go         (\g) Send command to mysql server.
help       (\h) Display this help.
notee     (\t) Don't write into outfile.
print      (\p) Print current command.
prompt     (\R) Change your mysql prompt.
quit       (\q) Quit mysql.
rehash     (\#) Rebuild completion hash.
source     (\.) Execute an SQL script file. Takes a file name as an argument.
status     (\s) Get status information from the server.
tee        (\T) Set outfile [to_outfile]. Append everything into given outfile.
use        (\u) Use another database. Takes database name as argument.
charset    (\C) Switch to another charset. Might be needed for processing binlog
with multi-byte charsets.
warnings   (\W) Show warnings after every statement.
```

Figure IV.3- Ecran d'aide MySQL

Par exemple la commande **status**, fournit quelques informations concernant l'état du serveur MySQL.



```
C:\Program Files\MySQL\MySQL Server 5.0\bin\mysql.exe

mysql> status
-----
C:\Program Files\MySQL\MySQL Server 5.0\bin\mysql.exe  Ver 14.12 Distrib 5.0.67,
for Win32 (ia32)

Connection id:          3
Current database:
Current user:           root@localhost
SSL:                    Not in use
Using delimiter:       ;
Server version:         5.0.67-community-nt MySQL Community Edition (GPL)
Protocol version:      10
Connection:            localhost via TCP/IP
Server characteraset:  utf8
Db characteraset:      utf8
Client characteraset:  utf8
Conn. characteraset:   utf8
TCP port:              3306
Uptime:                1 day 5 hours 29 min 1 sec

Threads: 1  Questions: 4  Slow queries: 0  Opens: 12  Flush tables: 1  Open tabl
es: 0  Queries per second avg: 0.000
-----

mysql> _
```

Figure IV.4- Ecran MySQL Command Line Client résultat commande status

On lance sur MySQL Administrator pour accéder à notre gestionnaire de base de données.

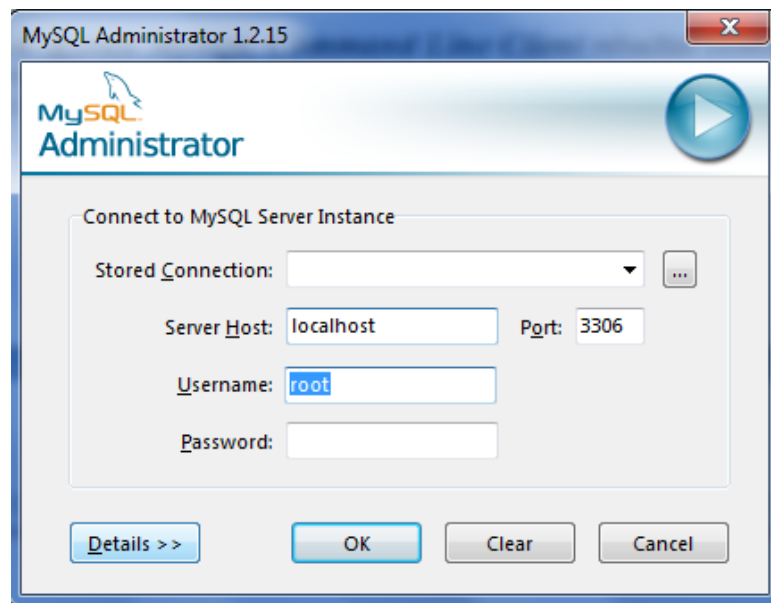


Figure IV.5-Page d'authentification de MySQL Administrator

Une fois entré le nom d'utilisateur (root par exemple) et le mot de passe, l'interface d'administration du serveur de données ci-dessous s'affiche :

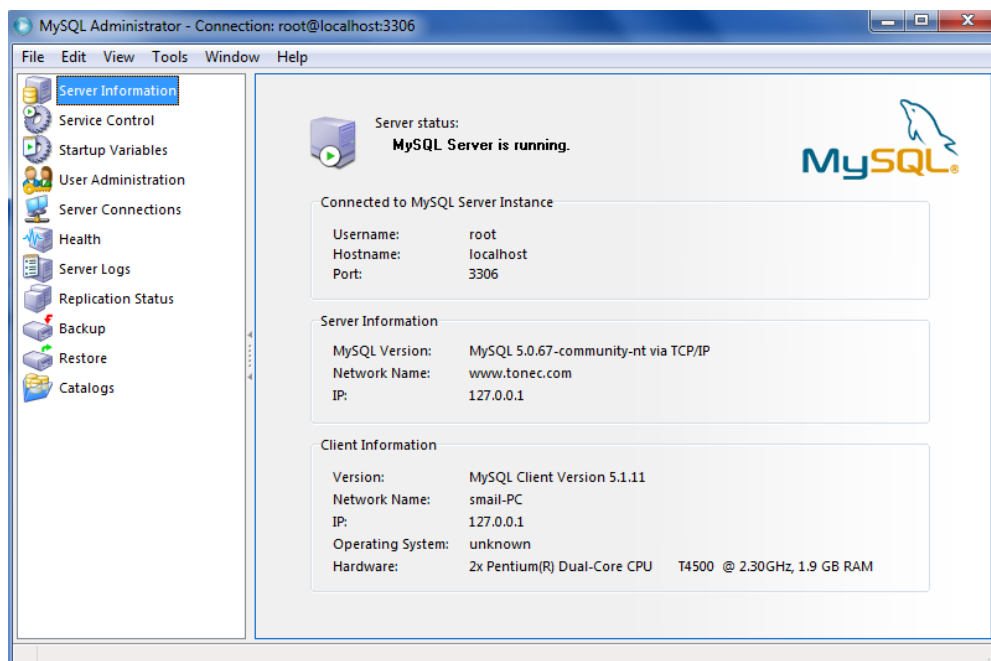


Figure IV.6-Page principale de MySQL Administrator

Cette interface, nous permet de réaliser toutes les opérations d'administration de maintenance sur notre serveur de données, entre autres, création/suppression d'utilisateur, création/suppression de bases, de tables...etc.

Après avoir installé et configuré le serveur de base de données MySQL et l'interface graphique d'administration *MySQL Administrator*, la conception de la base de donnée peut être entamée.

IV.4-Conception de la base de données

Afin de rendre nos services facilement accessibles à nos abonnés, nous avons créé une base de données nommée **ussd** composée de deux tables (inscription, historique.) structurées comme suit :

IV.4.1-Table inscription

La table inscription contient toutes les informations d'identité de nos abonnés. Elle est constituée de cinq champs et est structurée comme suit :

```
CREATE TABLE inscription (  
  'Id_user' INTEGER NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  'nom' VARCHAR (45) NOT NULL,  
  'email' VARCHAR (45) NOT NULL,  
  'password' VARCHAR (45) NOT NULL, PRIMARY KEY ('password'),  
  'portable' VARCHAR (10) NOT NULL,
```

Ce qui donne comme résultat sur l'interface MySQL Administrator :

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
Id_user	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	
nom	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL	
email	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL	
password	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL	
portable	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL	

Figure IV.8- Structure de la table inscription sur MySQL Administrator

IV.4.2-Table historique

La table historique contient l'historique de rechargement de crédit effectué par chaque abonné.

Elle est constituée de cinq champs créés avec les requêtes SQL suivantes :

```
CREATE TABLE historique (  
  `id` INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
  `nom` VARCHAR(45) NOT NULL,  
  `numero` VARCHAR(45) NOT NULL,  
  `credit` BIGINT(20) UNSIGNED NOT NULL,  
  `date` DATETIME NOT NULL,
```

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
id	INTEGER	✓	✓	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	
nom	VARCHAR(45)	✓		<input type="checkbox"/> BINARY	NULL	
numero	VARCHAR(45)	✓		<input type="checkbox"/> BINARY	NULL	
credit	BIGINT(20)	✓		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	
date	DATETIME	✓			NULL	

Figure IV.9- Structure de la table historique sur MySQL Adminstrator

Conclusion

Cette partie sur l'installation des serveurs et l'implémentation de la base de données MySQL constitue la première phase avant l'implémentation finale du projet qui est celle de l'interface services USSD, dans ce chapitre on a appris l'installation et la configuration des outils représentés dans le chapitre précédent.

Introduction

Notre solution se présente sous forme d'un portail WEB. Pour cela un développement basé sur un comportement dynamique via l'interpréteur PHP a été exploité.

Dans ce chapitre, nous allons expliquer le fonctionnement de différents scripts PHP qu'on a utilisé pour la création et l'ouverture d'une session, la présentation des interfaces graphiques de l'application et enfin le rechargement de crédit (envoi d'une requête USSD) à partir de la page d'un utilisateur.

V.1. Les scripts de création et l'ouverture d'une session

V.1.1. Script de création d'une session (création d'un compte)

Un utilisateur n'étant pas abonné, est obligé de créer une session pour accéder aux services.

Les scénarios ci-dessous illustrent le fonctionnement détaillés de cette création

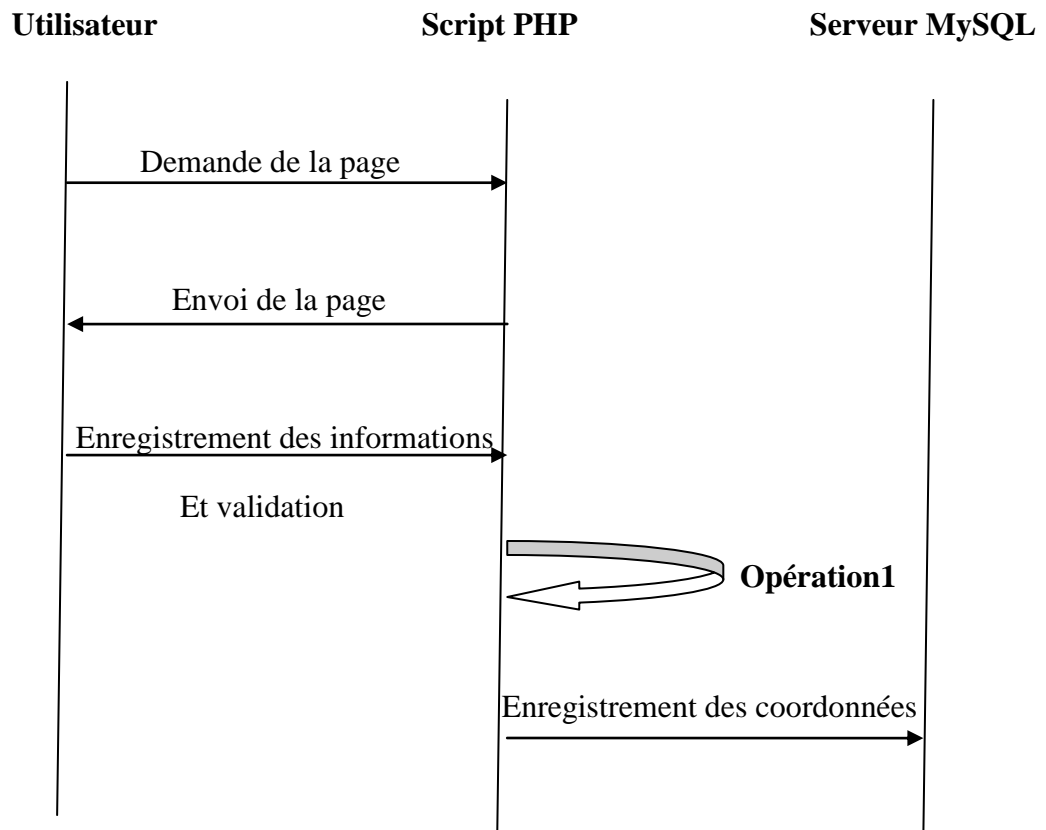
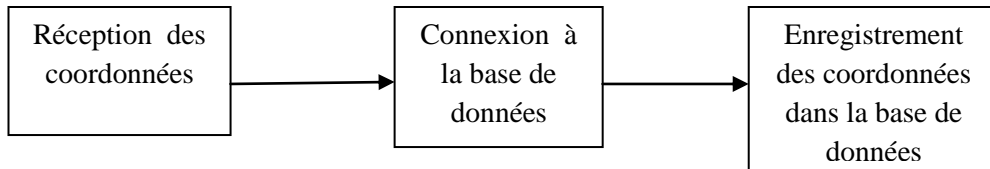


Figure V.1 scénario de création d'un compte utilisateur

Explication des scénarios

L'**opération1** fait appel à un ou plusieurs scripts qui sont détaillés dans les organigrammes suivants :



V.1.2.Script d'ouverture d'une session

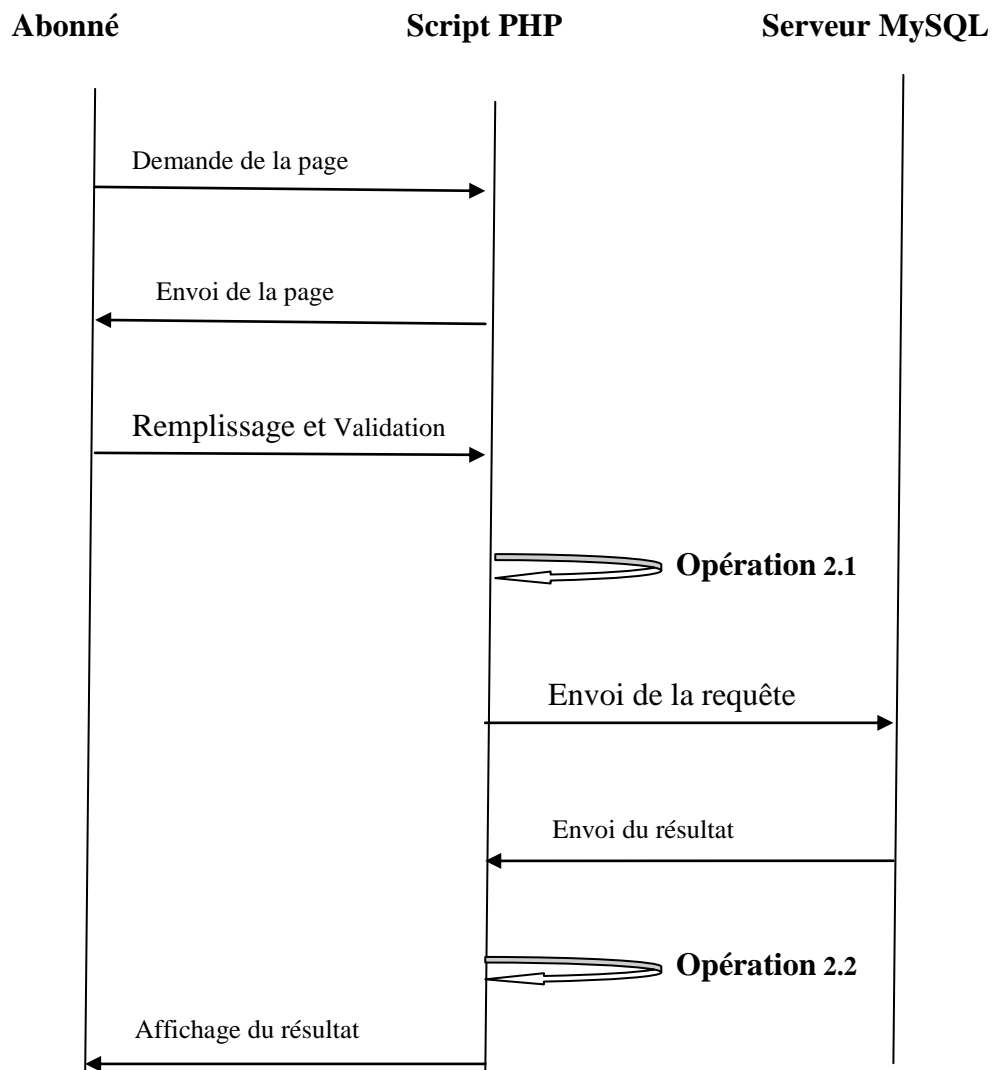
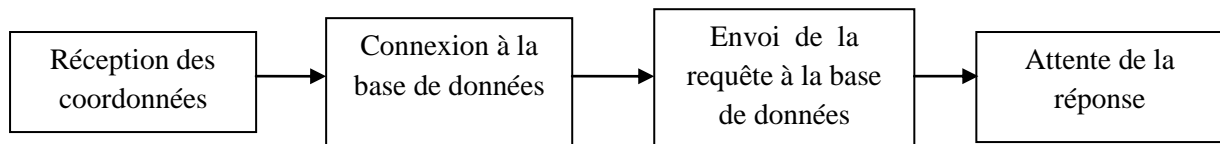


Figure V.2 scénario d'ouverture d'une session utilisateur

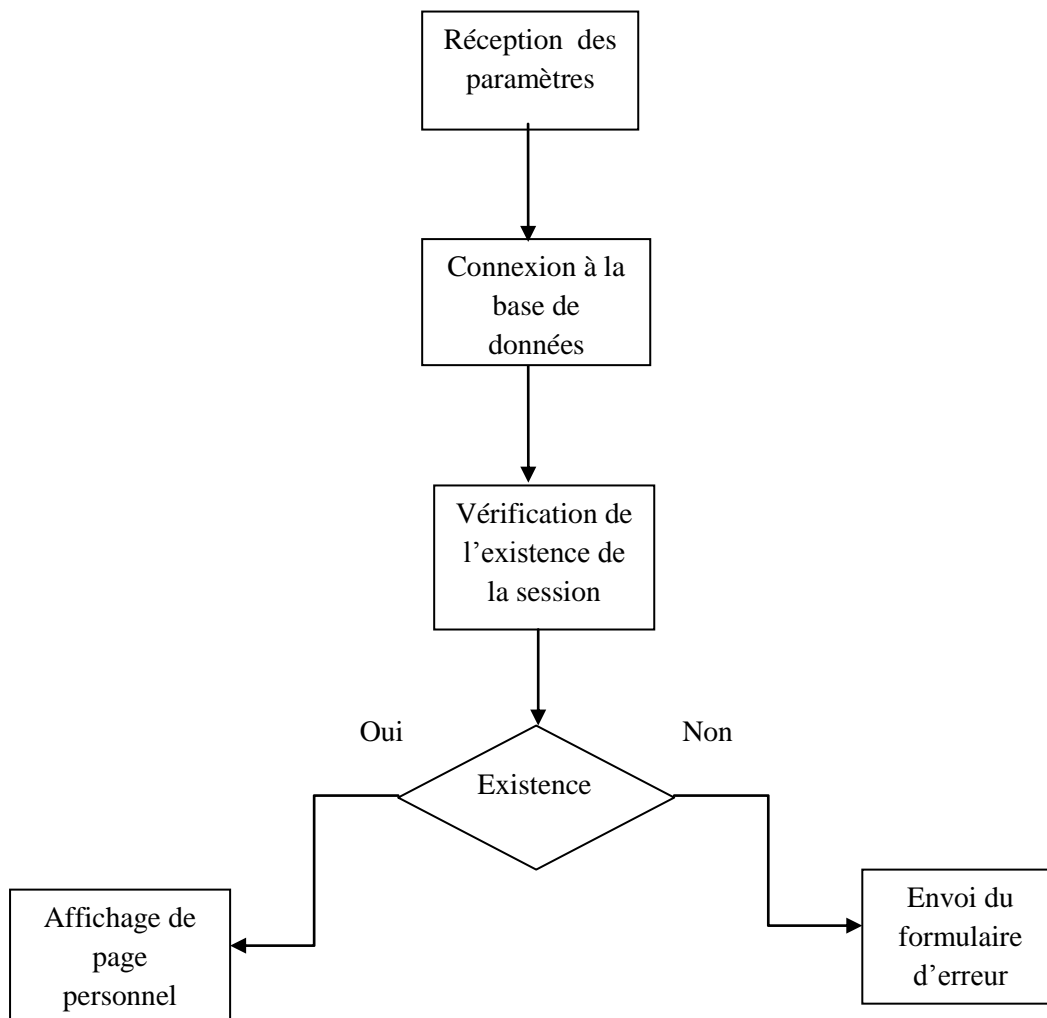
Explication des scénarios

Les méthodes de traitements des scripts PHP diffèrent de celles expliquées dans les scripts de création des comptes.

Opération 2.1



Opération 2.2



V.2.Présentation des interfaces graphiques de l'application

V.2.1.Page d'accueil

Après le saisis de l'URL du site dans le navigateur par l'utilisateur, la page d'accueil ci-après s'affiche :



Figure V.1-Page d'accueil du portail web

V.2.2.Interface graphique de la création d'une session

Avant de bénéficier de nos services l'utilisateur doit créer un compte c'est-à-dire créer une session en s'enregistrant son nom, son e-mail, son mot de passe, et son numéro de portable.

En cliquant sur le bouton **créer une session**, la page suivante s'affiche :



Page d'inscription x
localhost/inscription.php

Veillez remplir ce formulaire pour vous inscrire

Nom:

Email:

password:

portable:

Valider

Figure V.2-Page de création d'un compte

V.2.3.Interface graphique de l'ouverture d'une session

Pour ouvrir une session, l'abonné doit s'identifier en saisissant son nom et son mot de passe après avoir cliqué sur le bouton **ouvrir une session**.



identification x
localhost/identification.php

Bienvenue sur Mon site USSD

*****Veillez entrer vos coordonnées*****

Nom :

password:

Valider

Figure V.3- Page d'authentification des abonnés

Chapitre V Application

Si les coordonnées saisies sont erronées, un formulaire d'erreur que voici s'affiche :

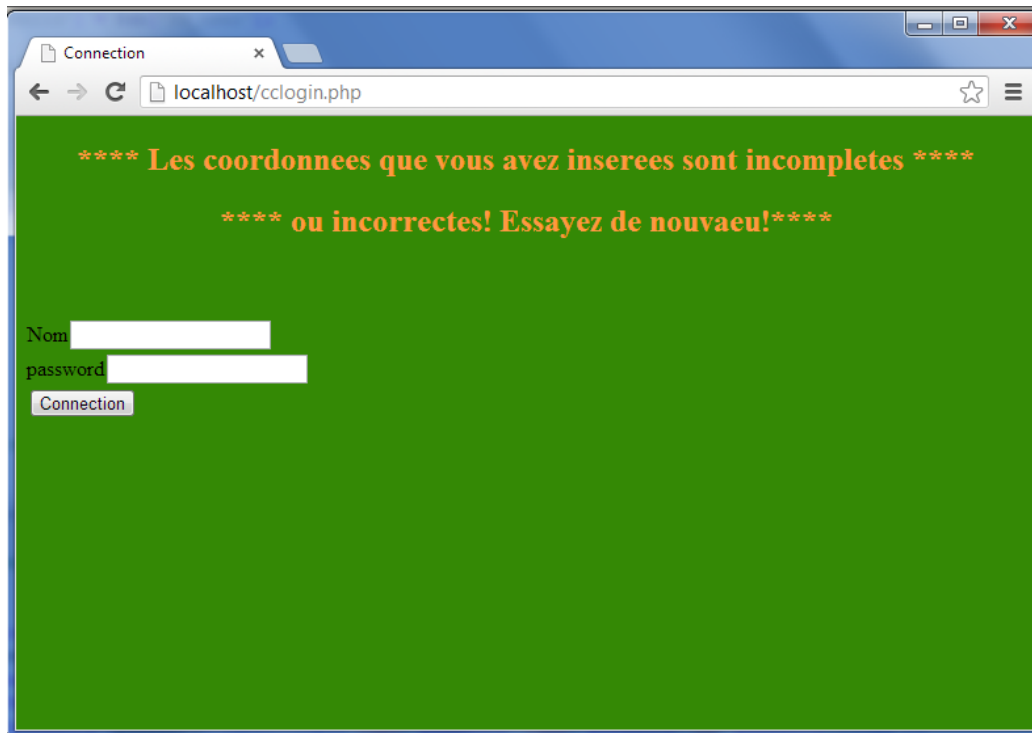


Figure V.4-Page d'affichage de message d'erreur

Sinon la page personnelle de l'abonné s'affiche. L'exemple suivant, illustre la page personnelle de l'abonné ismail oudahmed.



Figure V.5 : Page personnel de l'abonné Ismail

Cette interface présente un menu qui contient les services USSD implémentés. On s'est limité au service de rechargement de crédit pour les trois opérateurs **NEDJMA**, **MOBILIS**, et **DJEZZY**, et un service historique de rechargement qui permet à l'utilisateur de savoir les rechargements de crédit effectués à partir de son compte.

En cliquant sur le lien **Rechargement de crédit NEDJMA**, on obtient la page suivante :



Figure V.6-Page de rechargement de crédit NEDJMA

En cliquant sur le lien **Rechargement de crédit MOBILIS**, on obtient la page suivante :



Figure V.7-Page de rechargement de crédit MOBILIS

En cliquant sur le lien **Rechargement de crédit DJEZZY**, on obtient la page suivante :



Figure V.8-Page de rechargement de crédit DJEZZY

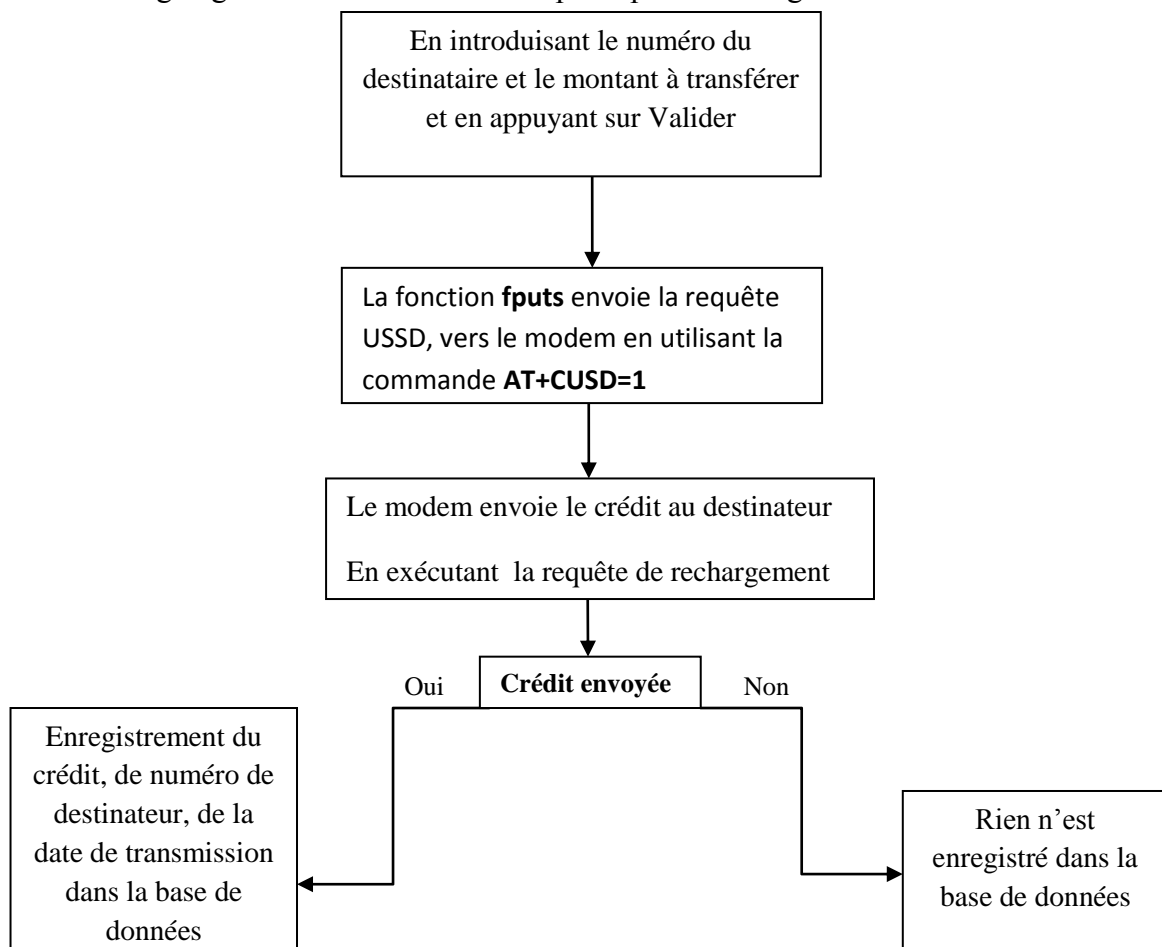
Chapitre V Application

En appuyant sur **Valider** après le remplissage de deux champs le numéro du destinataire et le montant à transférer, le destinataire reçoit le montant dans les quelques secondes qui suivent.

Ici quelques commandes AT et deux fonctions PHP ont été utilisées dans le programme PHP qui permet à l'utilisateur de transférer du crédit via un modem GSM ou un téléphone portable jouant le rôle d'un modem. Les scripts utilisent les fonctions si dessous :

- La fonction ***fputs*** () : renvoie la chaîne de caractères qui permet le rechargement de crédit par exemple (***fputs*** (*115*\$numero*\$credit#)) pour le rechargement de crédit NEDJMA.
- La fonction ***Sleep*** () : Arrête l'exécution durant quelques secondes
- La commande **AT+CUSD=1**: On l'utilise pour que la chaîne de caractère de rechargement de crédit s'envoie vers le modem.
- **AT+CMGF=1**: cette commande spécifie au modem que c'est le mode texte qui a été choisi, on choisit le mode texte pour l'envoi des codes USSD et SMS.

L'organigramme suivant résume le principe de rechargement de crédit effectué



Extrait du script PHP effectuant la gestion du modem :

```
// Identification du port série virtuel associé avec le modem GSM
    $port="com10: ";
// Test d'existence du port série
    if ($fh=fopen("com10:", "w+")) {
//écriture de la requette USSD vers le modem GSM
    fputs($fh, "AT\r");
        sleep(1);
        fputs($fh, "AT+CMGF=1\r");
    fputs($fh, "AT+CUSD=1, \ "*115*$numero*$credit#\ "\r");
        sleep(7);
        fputs($fh, "at+cusd=1, \ "1\ "\r");
    }
```

Conclusion

Dans ce chapitre on a résumé d'une façon pratique le travail effectué tout au long de ce projet. Le site dynamique qu'on a créé peut fonctionner dans un réseau intranet. L'objectif du projet est atteint car en saisissant le numéro du destinataire et le crédit à transfère depuis l'interface graphique « **Rechargement de crédit** », le crédit est directement envoyé via le réseau GSM. Ce chapitre prouve que l'envoi des requêtes USSD à partir d'un site dynamique via un réseau GSM a été un succès.

CONCLUSION GENERALE

L'objectif de notre projet de fin d'études était la conception et la réalisation d'un serveur intranet à partir duquel un abonné au site peut exécuter des commandes USSD, le rechargement du crédit à partir d'un ordinateur vers un téléphone mobile via un modem GSM ou un téléphone mobile jouant le même rôle que le modem GSM a été utilisé comme validation de la solution élaborée.

Dans ce projet plusieurs serveurs ont été implantés tels qu'Apache, MySQL. En utilisant des scripts PHP, un portail WEB a été créé pour intégrer le protocole USSD et simplifier les services de ce dernier. En effet, les utilisateurs vont exploiter un simple navigateur WEB pour exécuter une commande USSD. Un exemple d'utilisation permettant le rechargement des crédits prépayés des utilisateurs enregistrés a été effectué.

Durant l'élaboration de ce modeste travail, nous avons acquis plusieurs connaissances qui s'avèrent bénéfiques pour notre formation et qui sont un complément essentiel pour la consolidation de plusieurs données théoriques acquises tout long de notre formation académique, Nous avons également gagné en expériences dans l'étude des concepts de télécommunication tels que, GSM, USSD ..., avec lesquels nous n'étions pas très familiers par le passé.

Loin de rendre ce travail parfait, nous tenons à signaler que plusieurs difficultés ont été rencontrées et ont nécessités du courage et de la patience pour en venir à bout.

Liste des figures

Figure I.1- Architecture GSM.....	3
Figure I.2-Architecture USSD.....	6
Figure I.3 Service USSD de recharge de compte prépayé.....	7
Figure I.4- Le Mobile initie le dialogue USSD	8
Figure I.5- Le réseau initie le dialogue USSD.....	9
Figure II.1- <i>Modèle OSI</i>	13
Figure II.2- Les couches du modèle OSI et les principaux protocoles	14
Figure II.3-Modèle TCP/IP	16
Figure II.4-Protocoles et application TCP/IP.....	18
Figure II.5- Comparaison OSI-TCP/IP.....	19
Figure II.8-Structure hiérarchique de la DNS.....	20
Figure II.9-Architecture intranet.....	22
Figure III.1 : Schéma du protocole HTTP.....	26
Figure III.2- Requêtes et réponses avec HTML.....	27
Figure III.3 Requêtes et réponses avec PHP.....	29
Figure IV.1-Page d'accueil <i>du site</i>	32
Figure IV.2- Page de configuration de PHP	34
Figure IV.3- Ecran d'aide MySQL	35
FigureIV.4- Ecran MySQL Command Line Client résultat commandestatus	35
Figure IV.5-Page d'authentification de MySQL Administrator.....	36
Figure IV.6-Page principale de MySQL Administrator	36
Figure IV.8- Structure de la table inscription sur MySQL Administrator	37

Liste des figures

Figure IV.9- Structure de la table historique sur MySQL Adminstrator.....	38
Figure V.1 scénario de création d'un compte utilisateur.....	39
Figure V.2 scénario d'ouverture d'une session utilisateur.....	40
Figure V.1-Page d'accueil du portail web.....	42
Figure V.2-Page de création d'un compte.....	43
Figure V.3- Page d'authentification des abonnés.....	43
Figure V.4-Page d'affichage de message d'erreur	44
Figure V.5 : Page personnel de l'abonné Ismail.....	44
Figure V.6-Page de rechargement de crédit NEDJMA	45
Figure V.7-Page de rechargement de crédit MOBILIS.....	46
Figure V.8-Page de rechargement de crédit DJEZZY	46

Bibliographie

[1] www.memoireonline.com, PFE « Point sur l'internet et la téléphonie mobile au Cameroun » pour l'obtention de Diplôme d'ingénieur en télécommunications, Université de Yaoundé, Cameroun.

[2] Rapport de Projet de fin d'études, Année universitaire : 2006/2007, « Modélisation de la charge des nœuds « Cœur » de réseau GSM du Tunisie Télécom », Ecole supérieur de communication de Tunis.

[3] **Cours d'introduction à TCP/IP**, François Laissus, Version du 25 février 2009.

[4] **PFE** « Développement d'un outil d'aide à la gestion de la sécurité dans les réseaux TCP/IP » pour l'obtention de Diplôme d'ingénieur en Télécommunications, Ecole supérieur de communication de Tunis.

[5] **PFE** « conception et réalisation d'un serveur SMS », pour l'obtention de Diplôme d'ingénieur en Télécommunications, université de Tlemcen.

[6] Livre Best practices PHP 5, Auteur : Guillaume Ponçon, chapitre1, pages 7, 8,9.

[7] <http://www.mysql.fr>

[8] Maisse, S. 2003 - Installation et configuration d'un serveur web. 3^e édit. [En ligne : <http://tutorial.monserveurperso.com>].

RESUME

RESUME

L'objectif de ce projet est de concevoir et réaliser un serveur qui permet de diffuser des requêtes USSD (Unstructured supplementary Service data). Un test effectuant le rechargement de crédit à partir d'un ordinateur, connecté au réseau internet, vers un téléphone mobile a été utilisé comme référence.

Pour la mise en place de notre serveur, plusieurs étapes sont abordées, allant de l'étude du protocole USSD à l'élaboration du pilote permettant la gestion et l'administration du serveur connecté à un modem GSM, en passant par le développement d'un site web dynamique avec le langage PHP et la mise en service à travers le serveur web Apache et le serveur de base de données MySQL.

Mots clés : **USSD, PHP, MySQL, Apache, Base de données, TCP/IP.**