

UNIVERSITÉ MOHAMMED V de Rabat
Faculté des Sciences



Département d'Informatique

Filière Licence fondamentale
en Sciences Mathématiques et Informatique

PROJET DE FIN D'ÉTUDES

intitulé :

Développement d'une application web pour le partage
d'écran et l'accès à distance

Présenté par :

HAJI ISSAM ET JEBRANE ISSAM

soutenu le 2 Juillet 2021 devant le Jury

M. Prénom Nom	Professeur à la Faculté des Sciences - Rabat	<i>Président</i>
M. Prénom Nom	Professeur à la Faculté des Sciences - Rabat	<i>Encadrant</i>
Mme Prénom Nom	Professeur à la Faculté des Sciences - Rabat	<i>Examineur</i>

Année universitaire 2020-2021

Remerciements

Nous remercions notre créateur ALLAH Grand et Miséricordieux tout puissant de nous avoir donné le courage, la volonté et la patience de mener à terme le présent travail. Au terme de ce travail, nous tenons à exprimer notre profonde gratitude et nos sincères Remerciements à notre encadrant Professeur ALI OUACHA pour son aide, ses orientations, sa patience et sa disponibilité , pour tout le temps qu'il nous a consacré, ses conseils précieux, et pour la qualité de son suivi durant toute la période de notre projet.

Nous exprimons aussi notre gratitude à tous les membres du jury d'avoir accepté d'examiner et d'évaluer ce travail.

Avec beaucoup d'égard, nous ne manquerons pas d'exprimer notre grande reconnaissance à tous les enseignants et administrateurs de la Faculté des Sciences Rabat.

Nos remerciements vont aussi à toute personne qui a contribué de près ou de loin à l'élaboration de ce travail.

Résumé

L'objectif principal de ce projet de fin d'études consiste à développer une application web pour le protocole de bureau à distance base sur la plateforme NodeJS et Electron . nous avons essayé de donner à chaque utilisateur l'option de suivre le partage d'écran d'un autre ordinateur et aussi de pouvoir le contrôler, tout cela à distance .

Mots clés : NodeJS, Electron, php,xampp,app

Abstract

The purpose of this project consists of implementing a web app that allows to share computer's screen with others and the ability to have a complete access of a device based on NodeJs and Electron platforms.

Keywords : NodeJS, Electron, php,xampp,app

Table des matières

Dédicaces.....	i
Remerciements.....	ii
Résumé.....	iii
Abstract.....	iv
INTRODUCTION GÉNÉRALE.....	viii
1 LA PRÉSENTATION GÉNÉRALE	2
1.1 Cadre du travail.....	2
1.2 Présentation du projet.....	2
1.2.1 Objectifs visés.....	3
1.2.2 Approches du travail.....	3
1.2.3 Phase de recherche.....	3
1.2.4 Phase de conception et de développement.....	3
1.3 Le partage d'écran.....	3
1.3.1 définition.....	4
1.3.2 les outils de partage d'écran à distance.....	5
1.4 l'accès à distance.....	8
1.4.1 définition.....	8
1.4.2 mode d'affichage.....	9
1.4.3 fonctionnement.....	9
1.5 conclusion.....	10
2 L'ANALYSE ET SPÉCIFICATION DES BESOINS.....	10
2.1 Introduction.....	10
2.2 Spécification des besoins fonctionnels.....	10
2.3 Spécification des besoins non fonctionnels.....	10
2.4 Spécifications techniques.....	11
2.5 Diagramme de cas d'utilisations.....	11
2.5.1 Raffinement des cas d'utilisations.....	12
2.6 Conclusion.....	13
3 LA CONCEPTION.....	14
3.1 Introduction.....	14
3.2 Architecture Client / Server.....	14
3.3 Diagramme d'activités.....	15
3.3.1 Diagramme d'activités pour le cas d'utilisation "créer un compte".....	16

3.3.2 Diagramme d'activités pour le cas d'utilisation "s'authentifier"	17
3.4 Diagramme de séquence.....	18
3.5 Conclusion.....	18
4 L'ENVIRONNEMENT DE TRAVAIL.....	19
4.1 Introduction.....	19
4.2 Environnement matériel.....	19
4.3 Environnement logiciel.....	20
4.3.1 Les outils utilisés.....	21
4.4 Langages de programmation.....	24
4.5 Conclusion.....	25
5 REALISATION ET TEST.....	27
5.1 Introduction	26
5.2 architecture d'application	27
5.3 Présentation des interfaces	27
5.4 Conclusion.....	30
Conclusion	31

Table des figures

Figure 3.2.1 - join-me	5
Figure 3.2.2-teamviewer	6
Figure 3.2.2-anymeeting	6
Figure 3.2.3-skype	7
Figure 3.2.4 - zipcat	7
Figure 3.2.4 - acces a distance	8
// Figure 2.5.1 - Diagramme de cas d'utilisation	
.	11
Figure 3.2 - server /client	15
Figure 3.3.1 - Diagramme d'activités pour le cas d'utilisation "créer un compte"	16
Figure 3.3.2 - Diagramme d'activités pour le cas d'utilisation "s'authentifier"	17
Figure 3.4.1 - Diagramme de séquence pour le cas d'utilisation "s'authentifier"	18
Figure 4.3.1 - Les technologies utilisées	20
Figure 4.3.1.1 - Les technologies utilisées	21
Figure 4.3.1.2 - Les technologies utilisées	21
Figure 4.3.1.3 - Les technologies utilisées	22
Figure 4.3.1.4 - Les technologies utilisées	22
Figure 4.3.1.5 - Les technologies utilisées	23
Figure 4.3.1.6 - Les technologies utilisées	23
Figure 4.4.1 - L'angage de programmation	24
Figure 4.4.2 - L'angage de programmation	25
Figure 5.3.1 - home page	
.	27
Figure 5.3.2 - login page	28
Figure 5.3.3 - option page	29
Figure 5.3.4 - server page	30
Figure 5.3.5 - client page web	30
Figure 5.3.5 - client app	31
Figure 5.3.6 - client app	31

Introduction

Pendant ces dernières années, on a assisté à un développement fulgurant et une prolifération d'applications spécialisées pour réseau dans la transmission de données par Internet. Chaque jour apparaissent de nouvelles applications qui se déroulent à distance pour :

vidéoconférence, helpdesk, enseignement à distance, maintenance, reconfiguration, télétravail, réparation, aide ...etc. La liste est longue et ne cesse de grandir. Les conditions de service associées à ces applications diffèrent de celles des applications dites « élastiques » (email, web, partage de fichier,...) car les exigences de service de ces applications multimédia reposent sur deux axes : la synchronisation et la tolérance à la perte de données. Le bureau à distance (ou Remote Control en anglais) fait partie de ces applications multimédia. Aujourd'hui, la grande majorité des responsables informatiques ont pris conscience de l'intérêt des dispositifs de bureau distant. En effet, pour que les entreprises répondent à leurs défis surtout en ce qui concerne la continuité de l'activité et la rentabilité, elles doivent dorénavant s'orienter vers cette approche stratégique qui répondra à la demande croissante des utilisateurs, soutient les initiatives stratégiques et garantit la réactivité informatique, indispensable à toute organisation efficace.

Le bureau distant est une solution puissante garantissant la sécurité de l'accès, la mobilité des utilisateurs et la mise à disposition des applications à tout moment et à n'importe quel endroit. Dans ce contexte, notre objectif est de réaliser une application de bureau à distance qui permet aux étudiants de passer les examens à distance et permettent aussi au professeur de suivre les étudiants tous le long de ce passage d'examen, et ce par le biais de l'internet comme si vous étiez à sa place.

Chapitre 1

LA PRÉSENTATION GÉNÉRALE

Introduction :

Ce chapitre est consacré à la présentation, la précision du cadre du projet et la problématique puis l'énumération des étapes à suivre pour le réaliser.

1.1 Cadre du travail

Ce travail s'inscrit dans le cadre du projet de fin d'études (PFE) pour l'obtention du diplôme de licence Mathématique et Informatique et cela au terme du cursus de la formation qui s'est étalée sur trois ans au sein de la faculté des Sciences Rabat. Le sujet est intitulé "Développement d'une application web pour le partage d'écran et l'accès à distance".

1.2 Présentation du projet

Notre projet consiste à mettre en place une application de bureau à distance pour le passage des examens en ligne, qui s'exécutent à distance via le réseau local ou mondial tel que la télémaintenance, la téléintervention et la formation à distance.

L'application que nous essayons de faire est une application qui a le même principe que les autres applications connues comme zoom, teamviewer, anydesk, etc. sauf que pendant la réalisation de ce projet nous avons insisté sur des objectifs majeurs : faciliter la création d'une réunion et partager l'écran et ainsi la possibilité d'avoir l'accès simultanément aux écrans des utilisateurs participant à la réunion. L'utilisateur doit éventuellement s'inscrire pour qu'il puisse accéder aux différentes fonctionnalités, comme par exemple la création d'une réunion.

Nous avons modélisé cette application à l'aide d'un outil UML : Modelio en présentant dans le chapitre conception quelques diagrammes de vue statique et celles de vue dynamique.

Nous avons partagé les utilisateurs de notre application en deux types :

type 1 : les professeur son role est de creer une reunion pour pouvoire acceder au bureau d'un autre utilisateur qui rejoin cette reunion .

type 2 : ensuite on a un autre type d'utilisateur qui sont les etudiant qui vont essayer de rejoindre la reunion creer par leur professeur pour passer leur examen .

1.2.1 Objectifs visés

Ce projet a pour but de réaliser une application web fonctionnelle et esthétiquement attrayante, de navigation fluide et efficace, facile et simple d'utilisation qui permet au professeur d'examiner leurs étudiants à distance en utilisant le partage d'écran et l'accès au bureau des étudiants.

1.2.2 Approches du travail

Le projet comprend deux phases : la première est la recherche d'une solution convenable pour réaliser l'application, et la deuxième, la phase de conception et de développement.

1.2.3 Phase de recherche

C'est l'étape qui inclut l'étude bibliographique, dans laquelle nous devons saisir les différentes notions et technologies à utiliser dans le projet, les architectures, etc. Aussi elle renferme les tests des différentes solutions mises en hypothèse, et les outils nécessaires pour la réalisation du projet.

1.2.4 Phase de conception et de développement

C'est une étape, dans laquelle, nous spécifions les besoins fonctionnels et nous modélisons le système à réaliser pour clarifier les tâches à accomplir dans la partie développement. Cette phase se termine par une partie qui comprend la programmation et les tests de validation.

1.3 Le partage d'écran

De nos jours, les écrans sont partout. Cependant, parfois la multiplicité des appareils nuit à la pédagogie ou à la collaboration, pouvoir partager le même écran serait une idée

judicieuse.

1.3.1 définition

Les gros systèmes IBM, DEC ou bien Unisys mentent d'emblée en œuvre le partage d'écrans ; puisqu'un système centralisé est équipé de terminaux passifs qui ne contiennent aucune information. En se connectant avec son login sur n'importe quel terminal, l'utilisateur peut consulter son environnement personnalisé et sécurisé.

Avec le développement de la micro-informatique, chacun était capable d'exploiter son propre système, c'était la fin des écrans partagés. On échangeait des disquettes et on parlait de partage de fichier. Ce dernier s'est étendu avec le développement des réseaux locaux, de la messagerie, puis de l'internet. L'explosion de la micro-informatique, cela a amené beaucoup d'autres problèmes de gestion de parc et de partage d'information.

- Le partage de fichier n'est pas toujours efficient, car il est soumis à la contrainte d'avoir les mêmes versions de logiciels entre ceux qui se partagent les fichiers.
- Chacun étant capable d'installer des programmes dans son environnement, cela peut compromettre le bon fonctionnement des ordinateurs (apparition de virus informatiques, détérioration des performances des systèmes, ...).

Dans le monde des applications Windows, on a l'application Windows Terminal Server qui permet d'utiliser des applications sur un mode équivalent à celui d'un Terminal passif sur un serveur central. Une réponse a été apportée par le développement de l'Internet, et la mise à disposition d'applications partagées et sécurisées avec authentification (parfois appelées extranet). On reconduit ainsi en partie le modèle des systèmes centralisés.

Pour les outils de type bureautique, une réponse est avancée par les applications qui s'exécutent sur Internet avec des fichiers eux aussi sur Internet (applications Cloud). Dans ce cas, la dépendance à une application installée sur le PC devient caduque. Par exemple, le géant Google met à disposition une suite bureautique très riche, Google applications, qui est utilisable à partir d'un simple navigateur.

Mais il y a des moments pour résoudre des problèmes techniques sur un pc en prenant le contrôle à distance de celui-ci ; on parle ici des applications comme VNC ou TeamViewer.

1.3.2 les outils de partage d'écran à distance

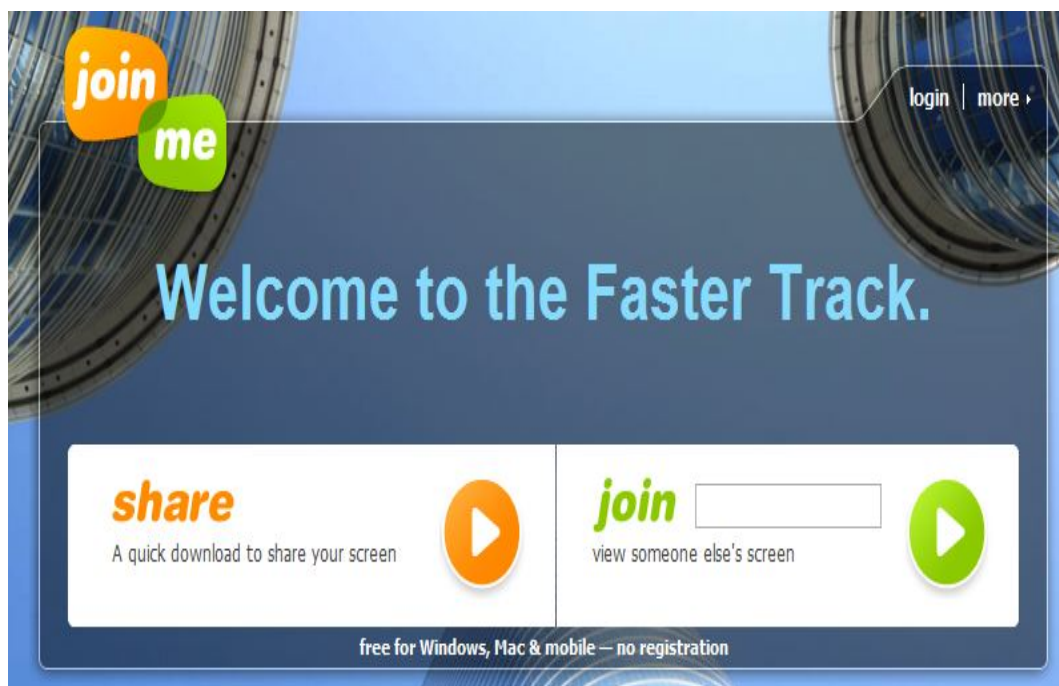


Figure 3.2.1 – join-me

Avec join.me, vous pouvez changer d'écran à partager. Autrement dit, soit vous choisissez de partager votre écran avec d'autres participants, à ce moment-là, vous cliquez sur "Share" et vous obtenez un code d'accès. Vous partagez ce code d'accès avec les participants, qui ont ensuite accès à votre écran d'ordinateur. Soit, vous choisissez de visionner l'écran d'un des participants. À ce moment-là, ce participant doit procéder de la même façon pour partager avec vous son écran. Une fois que vous avez téléchargé le logiciel, vous y avez accès via un raccourci sur votre bureau. Vous avez notamment des options pour obtenir des numéros d'appels conférences, vous avez des options de "chat" et vous pouvez voir le nombre de participants.

2 – Team Viewer



Figure 3.2.2 – teamviewer

TeamViewer ressemble un peu à join.me. Lorsqu'un individu partage son écran d'ordinateur avec vous, vous pouvez notamment utiliser votre souris pour effectuer des manipulations sur le fichier que l'autre individu partage avec vous.

3 – AnyMeeting



Figure 3.2.2 – anymeeting

AnyMeeting offre de nombreuses options intéressantes dont celle de pouvoir enregistrer vos présentations à distance et celle d'intégrer un outil pour gérer les abonnements à vos

webinaires (formulaires d'inscription, paiement par cartes de crédit ou paypal).

4 – Skype



Figure 3.2.3 – skype

Vous utilisez probablement déjà la fonctionnalité d'appels de Skype, mais sachez que Skype vous offre également la possibilité de partager votre écran à distance.

5 – Zipcast



Figure 3.2.4 – zipcat

Zipcast est l'outil de partage d'écran du célèbre SlideShare. Il vous permet notamment de partager vos présentations à distance ainsi que des vidéos. Vous pouvez céder à

l'avance vos rencontres à distance. L'outil dispose d'une zone de "chat" et offre aussi une intégration avec Facebook et Twitter.

1.4 l'accès à distance¹

1.4.1 définition



Figure 3.2.4 – acces a distance

En informatique, l'accès à distance¹, la commande à distance ou encore le contrôle à distance sont des méthodes qui permettent, depuis un ordinateur éloigné et sans limite théorique de distance, de prendre le contrôle d'un autre ordinateur en affichant l'écran de celui-ci et en manipulant les fonctions d'un périphérique d'entrée comme un clavier. Cet accès peut être effectué vers des postes de travail ou des serveurs informatique en fonction des possibilités du logiciel utilisé.

Un avantage de ces solutions à distance est qu'elles permettent au débutant en informatique de vivre l'expérience en temps réel : il voit et vit en direct sur son propre écran ce que fait un technicien distant comme si ce dernier était à ses côtés.

L'accès à distance est une évolution importante dans l'informatique : les utilisateurs peuvent être dépannés facilement et rapidement sans déplacement géographique d'un technicien. Selon Sun Microsystems, 80

1.4.2 mode d'affichage

Deux modes d'affichage du poste distant sont actuellement possible : le mode texte, qui affiche l'interface distante en ligne de commande, souvent appelé par abus de langage terminal distant et le mode graphique qui est appelé bureau à distance.

1.4.3 fonctionnement

La prise en main à distance a principalement vu le jour avec l'amélioration des connexions Internet et l'apparition du Haut-débit dans les foyers. Son fonctionnement est relativement complexe mais arrive aujourd'hui à maturité, notamment grâce aux solutions d'éditeurs, robustes, sécurisées et fiables.

- L'utilisateur et le technicien se connectent à Internet.
- Un petit logiciel s'exécute, et éventuellement chiffre, puis transmet une image du bureau de l'utilisateur au technicien.
- Le technicien voit alors exactement l'écran de l'utilisateur.
- À chaque opération d'un côté ou de l'autre – clic, ouverture de dossier ou de programme, etc – l'image se met à jour.

Pour se dérouler dans les meilleures conditions, une session d'assistance à distance doit mettre en œuvre un logiciel adéquat et une connexion d'un débit suffisant : l'Accès à internet à haut débit, par exemple l'ADSL, est préférable. En effet, lors de l'usage habituel de l'Internet, c'est le débit « descendant », c'est-à-dire la vitesse de transfert des données vers l'ordinateur de l'utilisateur, qui conditionnera la vitesse de travail de ce dernier. Alors qu'ici c'est le débit « montant » qui sera le plus utile, de l'ordinateur de la personne dépannée vers l'Internet.

1.5 conclusion

Au terme de ce premier chapitre, nous avons introduit le contexte de notre projet dont les étapes de réalisation seront décrites d'une manière détaillée dans les chapitres qui suivent. Le chapitre suivant sera réservé à la spécification détaillée et l'analyse des différents besoins de notre application.

Chapitre 2

L'ANALYSE ET SPÉCIFICATION DES BESOINS

2.1 Introduction

La phase d'analyse et spécification des besoins présente une étape primordiale dans le cycle de développement d'un projet. En effet, elle permet de mieux comprendre le travail demandé en dégagant les besoins des différents utilisateurs que le système doit accomplir.

2.2 Spécification des besoins fonctionnels

le client peut creer une reunion est acceder aux differents machines des autres personnes qui sont dans sa reunion comme il peut tous simplement rejoindre une reunion est partager l'acces de sa machine au personne qui a creer la reunion

2.3 Spécification des besoins non fonctionnels

A part les besoins fondamentaux, notre système doit répondre aux critères suivants :
la rapidité de traitement : En effet, vu le nombre important des transactions quotidiennes, il est impérativement nécessaire que la durée d'exécution des traitements s'approche le plus possible du temps réel.

La performance : Un site doit être avant tout performant c'est-à-dire à travers ses fonctionnalités, répond à toutes les exigences des usagers d'une manière optimale.

La convivialité : Le site doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales c'est-à-dire simples, ergonomiques et adaptées à l'utilisateur.

2.4 Spécifications techniques

- Il faut que toute interface de notre site soit homogène, en effet, les différentes interfaces doivent suivre le même modèle de représentation (couleurs, images, textes défilants, etc.).
- Le code doit être extensible et maintenable pour faciliter toute opération d'amélioration ou d'optimisation

2.5 Diagramme de cas d'utilisations

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure une relation entre l'utilisateur et les objets que le système met en œuvre. Il représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Pour ce qui est de notre projet, nous l'utilisons lors des étapes de capture des besoins fonctionnels et techniques dans notre processus de développement. Cette procédure est efficace car elle permet de cerner tous les besoins de l'utilisateur avant la réalisation du projet (Figure 2.5.1).

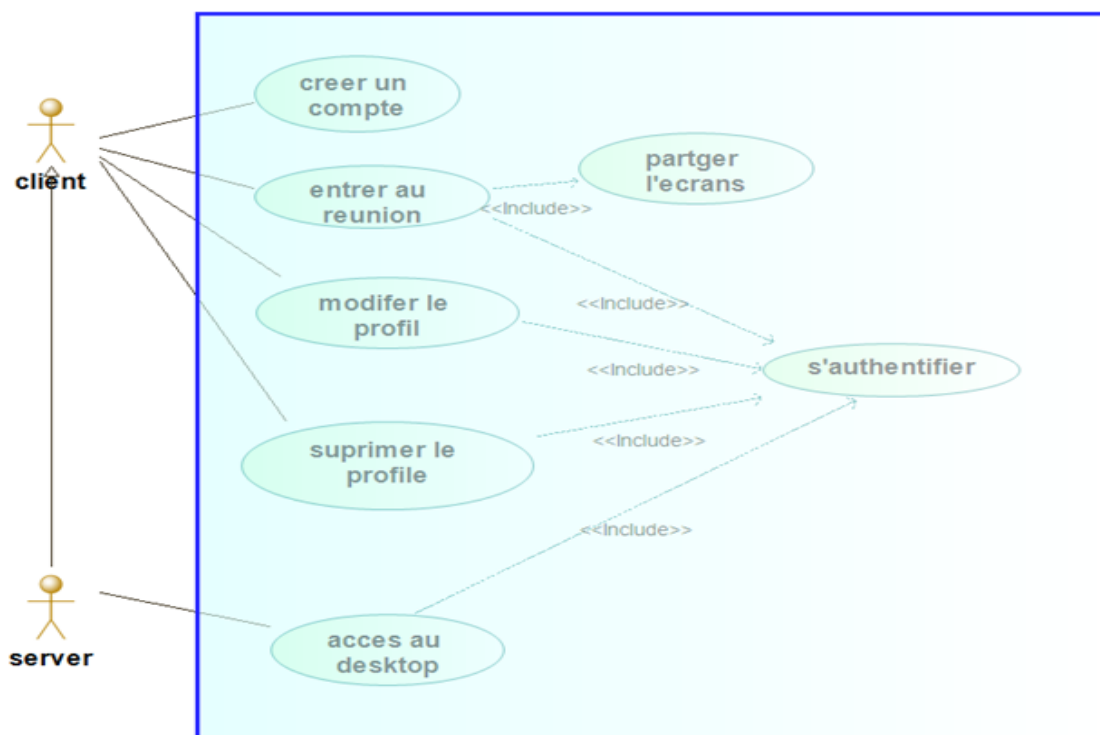


Figure 2.5.1 – Diagramme de cas d'utilisation

2.5.1 Raffinement des cas d'utilisations

Raffinement de cas d'utilisation "Créer un compte"

Cas d'utilisation :	Créer un compte (s'inscrire)
Acteurs :	client
Pré-conditions :	le client lance le site
post-condition :	le client est ajouté dans la base de données
Scénario :	1.Le client clique sur"register(s'inscrire). 2.Le système affiche le formulaire d'inscription. 3.Le visiteur remplit le formulaire par ses informations. 4.Le système vérifie la cohérence et l'unicité des données saisies.

TABLE 2.1 – "Créer un compte"

Raffinement de cas d'utilisation "partager l'écrans"

Cas d'utilisation :	Partager l'écrans
Acteurs :	client
Pré-conditions :	Le site fonctionne et le client est connecté
post-condition :	Le client a rejoint la réunion
Scénario :	1.Le client clique sure 'join room' (rejoindre le réunion). 2.le système affiche une interface qui propose au client l'ouverture d'une nouvelle fenêtre broad-cast-client ou se trouve l'option share screen

TABLE 2.2 – " Partager l'écrans"

Raffinement de cas d'utilisation "acces au desktop"

Cas d'utilisation :	Acces au desktop
Acteurs :	client
Pré-conditions :	Le site fonctionne et le client est connecté
post-condition :	Le client contrôle l'écrans
Scénario :	1.le client clique sure 'get access to desktop'(acces au desktop). 2.Le systeme ouvre une pompt alert qui demande au utilisateur d'entrer id du client

TABLE 2.3 – "Acces au desktop"

Raffinement de cas d'utilisation "Se connecter" (s'authentifier)

ET

Raffinement de cas d'utilisation "Supprimer son profil"

Cas d'utilisation :	Se connecter
Acteurs :	client serveur
Pré-conditions :	Le système fonctionne
post-condition :	le client est connecté
Scénario :	1.L'utilisateur clique sur " log in "(se connecter) 2.Le système affiche le formulaire d'authentification 3.L'utilisateur saisit son email et mot de passe. 4.Le système ouvre une session à l'utilisateur en donnant un accès a l'application.

TABLE 2.4 – "Se connecter"

Cas d'utilisation :	Supprimer son profile
Acteurs :	client
Pré-conditions :	Le site fonctionne et le client est connecté
post-condition :	Le profil est supprimé
Scénario :	1.Le client appuie sur "your profil". 2.Le système affiche une " alert dialog ". 3.L'utilisateur appuie sur " delete ". 4.Le système supprime le client de la base de données et toutes ses annonces. 5. Un message de confirmation s'affiche

TABLE 2.5 – "Supprimer son profile"

2.6 conclusion

Dans ce chapitre, nous avons énuméré les différents besoins fonctionnels et non fonctionnels de notre application. Ensuite, nous avons fait une étude des différents cas d'utilisation de notre solution. Ce chapitre a été d'une importance cruciale surtout pour la compréhension des besoins et attentes du client.

Chapitre 3

LA CONCEPTION

3.1 Introduction

Dans ce chapitre nous abordons la partie conception du projet, dans laquelle, nous détaillons les différents éléments de conception, à savoir les diagrammes de séquences et les diagrammes d'activités.

3.2 Architecture Client / Server

Avons d'abord les phases de conception nous consacrons une partie pour définir l'architecture client/serveur qu'on a utilisée dans notre application de partage à distance . nous avons devisé notre projet en deux partie client et partie serveur chacune joue un rôle précise, pour le côté serveur il attend l'envoi des images du côté client et il les s'affichent dans l'interface serveur ou le côté client il attend une demande d'image de la parts du serveur lorsque la demande et faite il envoie. pour bien comprendre cette notion on va se bien ses approfondire , Le protocole ou environnement client–serveur désigne un mode de transaction (souvent à travers un réseau) entre plusieurs programmes ou processus : l'un, qualifié de client, envoie des requêtes ; l'autre, qualifié de serveur, attend les requêtes des clients et y répond. Le serveur offre ici un service au client. Par extension, le client désigne souvent l'ordinateur sur lequel est exécuté le logiciel client, et le serveur, l'ordinateur sur lequel est exécuté le logiciel serveur. Les machines serveurs sont généralement dotées de capacités supérieures à celles des ordinateurs personnels en ce qui concerne la puissance de calcul, les entrées-sorties et les connexions réseau, afin de pouvoir répondre de manière efficace à un grand nombre de clients. Les clients sont souvent des ordinateurs personnels ou terminaux individuels (téléphone, tablette), mais pas systématiquement. Un serveur peut répondre aux requêtes de plusieurs clients. Parfois le client et le serveur peuvent être sur la même machine.

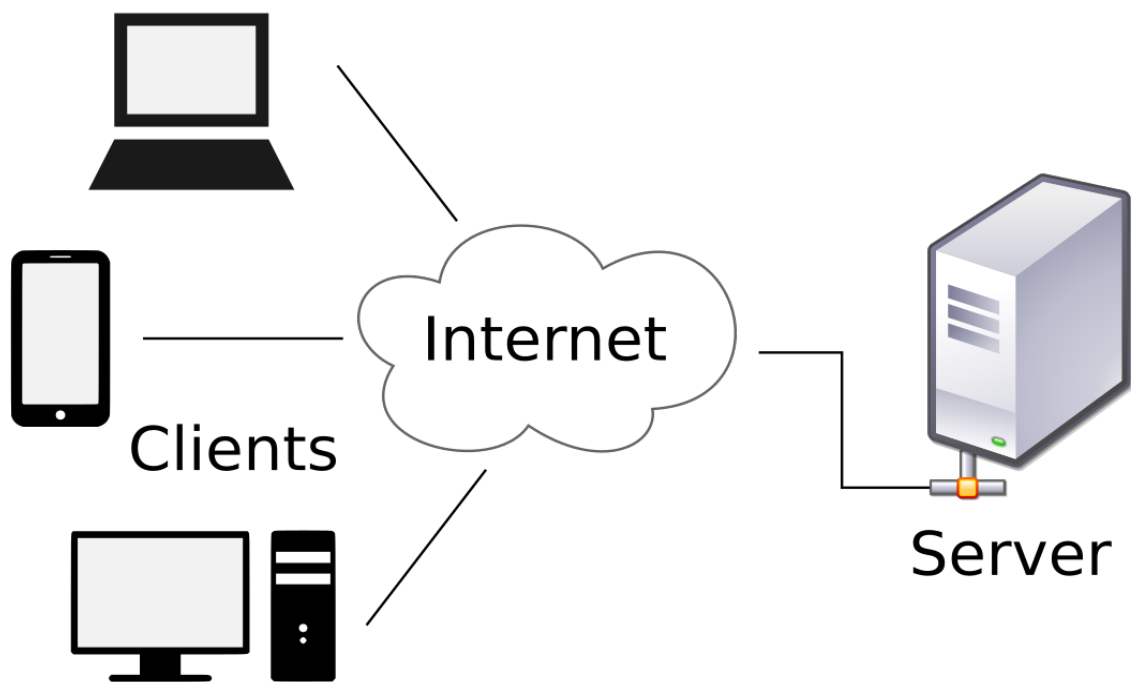


Figure 3.2 – server /client

3.3 Diagramme d'activités

Le diagramme d'activités permet de décrire sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants.

3.3.1 Diagramme d'activités pour le cas d'utilisation "créer un compte"

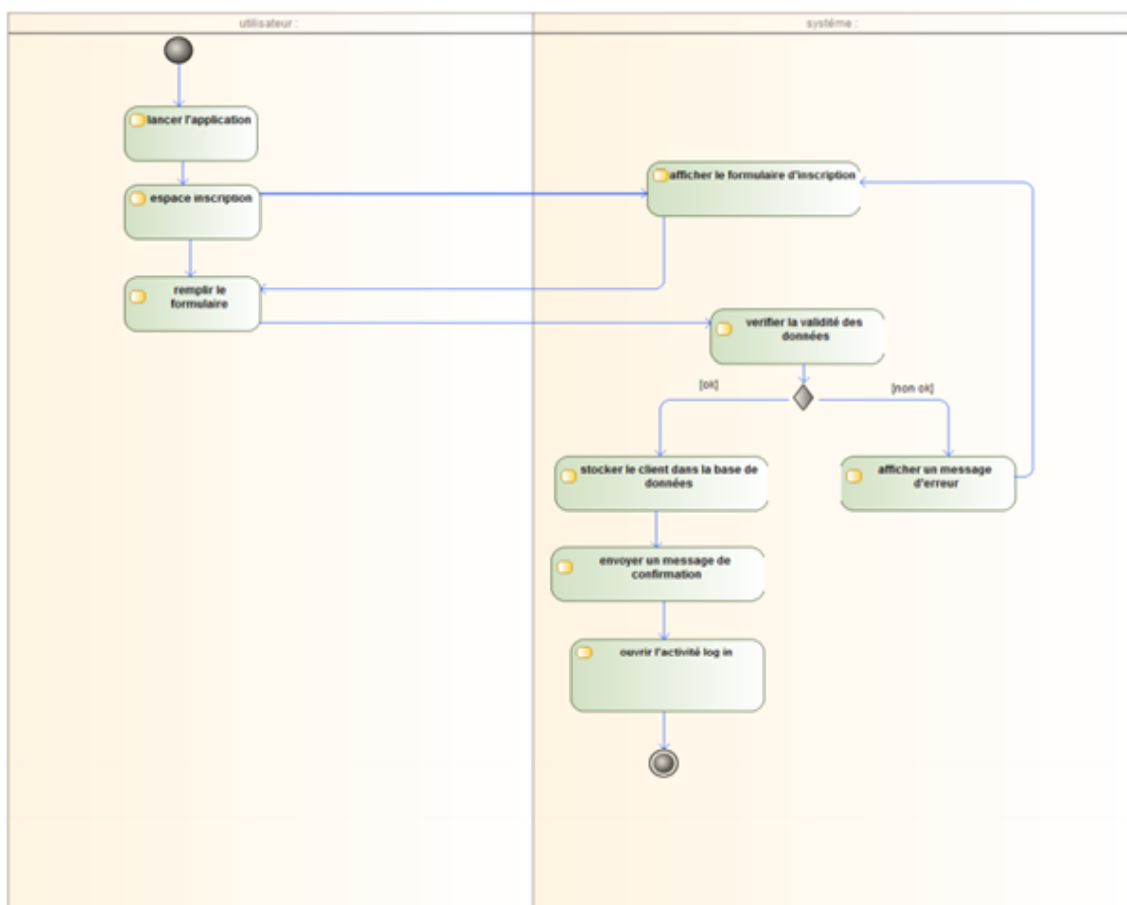


Figure 3.3.1 – Diagramme d'activités pour le cas d'utilisation "créer un compte"

3.3.2 Diagramme d'activités pour le cas d'utilisation "s'authentifier"

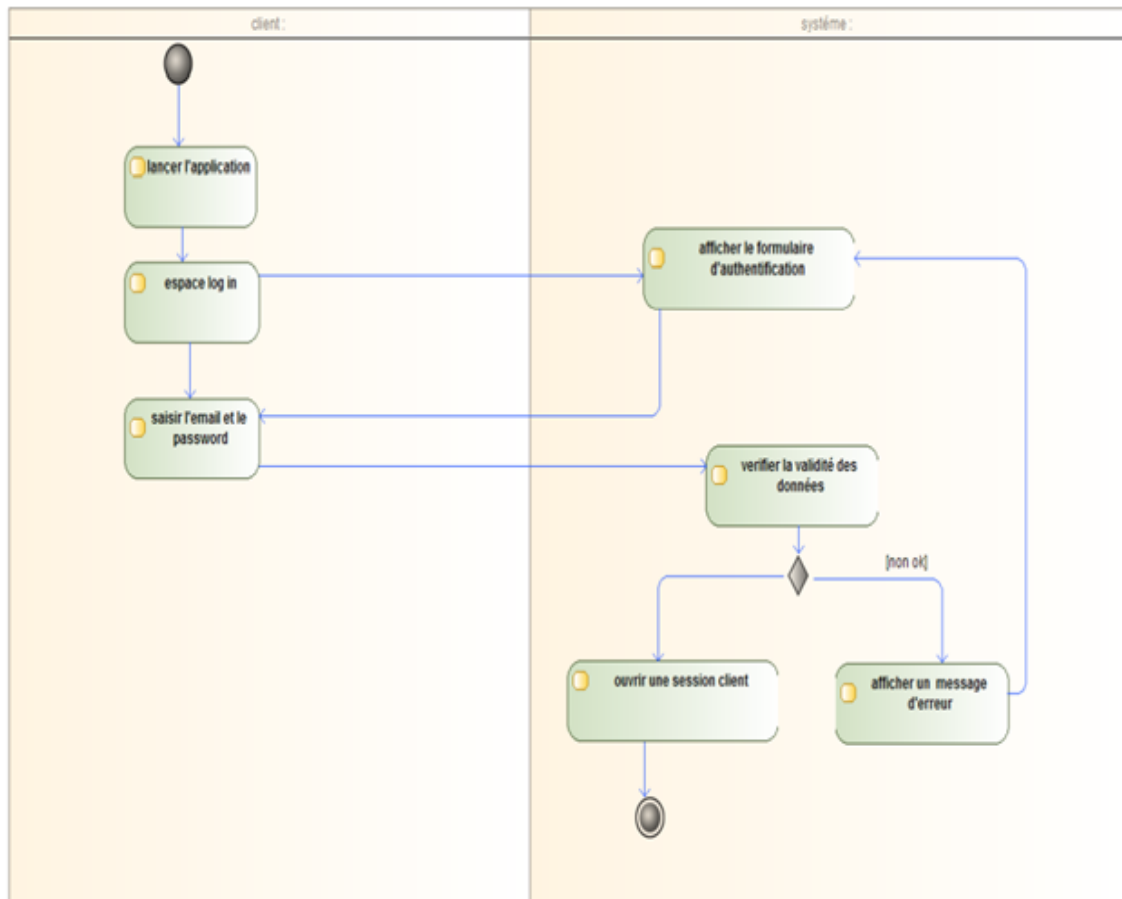


Figure 3.3.2 – Diagramme d'activités pour le cas d'utilisation "s'authentifier"

3.4 Diagramme de séquence

Le diagramme de séquence décrit l'aspect dynamique du système. Il modélise les interactions entre les objets ou entre utilisateur et objet, en mettant l'accent sur la chronologie des messages échangés.

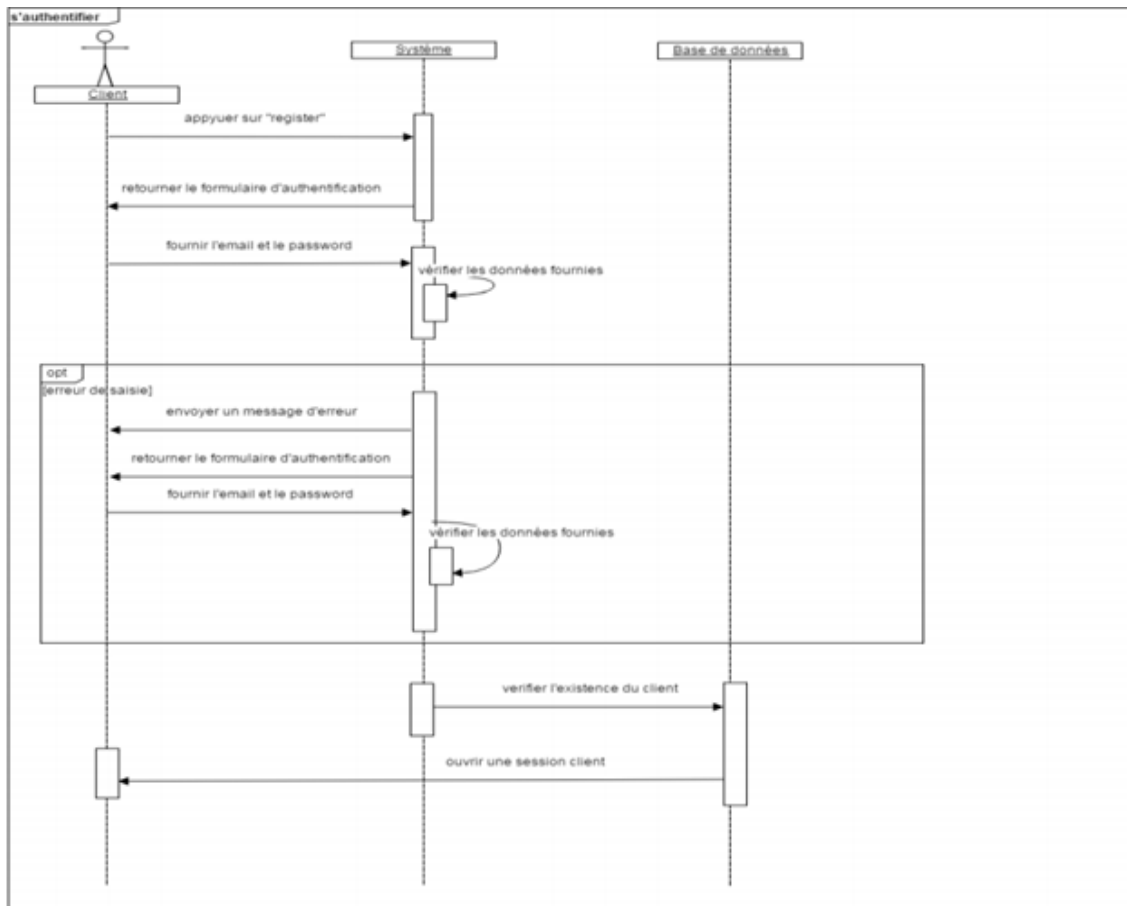


Figure 3.4.1 – Diagramme de séquence pour le cas d'utilisation "s'authentifier"

3.5 Conclusion

Dans ce chapitre nous avons détaillé la phase de conception, ainsi nous avons fait la description des diagrammes de classe et d'activité afin de délimiter le cadre de notre travail et de préparer pour la prochaine étape.

Chapitre 4

L'ENVIRONNEMENT DE TRAVAIL

4.1 Introduction

Ce chapitre est consacré à la présentation de l'environnement matériel et logiciel utilisés pour le développement de la solution proposée, nous expliquerons éventuellement nos choix techniques relatifs aux langages de programmation et des outils utilisés.

4.2 Environnement matériel

L'application a été développée sur des machines possédant les caractéristiques suivantes, les deux machines ont été utilisées pour coder :

Un pc portable HP :

- Système d'exploitation Windows 10,
- Processeur Intel CORE i7 5th Gen,
 - 8 Go de RAM,
- La taille du disque dur 256 Go.

Un pc portable HP :

- Système d'exploitation Windows 10,
- Processeur Intel CORE i5 9th Gen,
 - 16 Go de RAM,
- La taille du disque dur 256 Go.

4.3 Environnement logiciel

Nous avons utilisé les outils suivants tout au long du développement de notre projet de fin d'étude :



Figure 4.3.1 – Les technologies utilisées

4.3.1 Les outils utilisés

Pour réaliser notre application nous avons eu recours aux logiciels suivants :

1-XAMPP : est une plateforme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement les bases de données.



Figure 4.3.1.1 – Les technologies utilisées

2-Visual studio code : est un éditeur de code source léger mais puissant qui s'exécute sur votre bureau et est disponible pour Windows, macOS et Linux. Il est livré avec une prise en charge intégrée de JavaScript, TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages (tels que C++, C, Java, Python, PHP, Go) et des environnements d'exécution (tels que .NET et Unity) .

Vs code est développé avec le Framework Electron et conçu principalement pour développer des projets avec Javascript, Node.js. Facile à installer, à comprendre, à utiliser bien qu'il est rapide, il dispose d'une interface graphique responsive et customisable via des thèmes déjà installés.

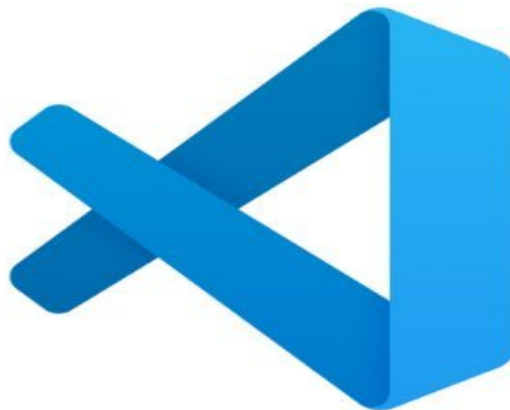


Figure 4.3.1.2 – Les technologies utilisées

3-NODE.JS : Node.js est une plateforme de développement JavaScript intégrant un serveur http, il est un environnement d'exécution single-thread, open-source et multi-

plateforme permettant de créer des applications rapides et évolutives côté serveur et en réseau. Il fonctionne avec le moteur d'exécution JavaScript V8 et utilise une architecture d'E / S non bloquante ce qui le rend efficace et adapté aux applications en temps réel.



Figure 4.3.1.3 – Les technologies utilisées

3- ELECTRON.JS : est un environnement permettant de développer des applications multi-plateformes de bureau avec des technologies web. L'infrastructure est codée en node.js, et l'interface est bâtie sur les outils Chromium, la partie open source de Google Chrome.

Le principe de ce Framework Electron vous permet d'écrire des applications de bureau multiplateformes en utilisant JavaScript, HTML et CSS. Il est basé sur Node.js et Chromium, et il est utilisé par l'éditeur Atom, mais également de nombreuses autres applications comme Visual Studio Code, l'éditeur de code open source développé par Microsoft.



Figure 4.3.1.4 – Les technologies utilisées

5- PHP : est le langage utilisé pour l'écriture de nos scripts, c'est un langage de script HTML, exécuté côté serveur. 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.



Figure 4.3.1.5 – Les technologies utilisées

6- PHPmyadmin : est une application Web de gestion pour les systèmes de gestion de base de données MySQL et MariaDB, réalisée principalement en PHP et distribuée sous licence GNU GPL. on a utilisé cette application pour stocker les données des utilisateurs.



Figure 4.3.1.6 – Les technologies utilisées

4.4 Langages de programmation

Pour la mise en forme des interfaces web on a utiliser les langage html et css ensuite pour que nos interface s'affiche en fonction de la demande de l'utilisateur (l'internaute, le client), là on utiliser l'angage de programmation plus évolué que le simple code HTML de construction de page Web qu'il faut . Le langage PHP permet cela, il permet de faire construire des pages web dynamiquement par le Serveur Web (construction de la page Web à la demande). Les pages Web construite dynamiquement par le Serveur Web sont dites sécurisées car les informations qui auront servies à construire ces pages Web dynamiques ne sont pas visible par l'utilisateur (l'internaute, le client). Bien souvent quand on utilise la technologie des pages Web dynamiques, les informations écrites (textes, ...) se trouvent stockées dans une base de données et ce n'est qu'au moment où la page Web est demandée à être créée, que l'on va demander de déclencher l'action (par le langage PHP) d'aller puiser dans cette base de données (à l'aide du langage SQL) pour n'en extraire que les informations souhaitées.

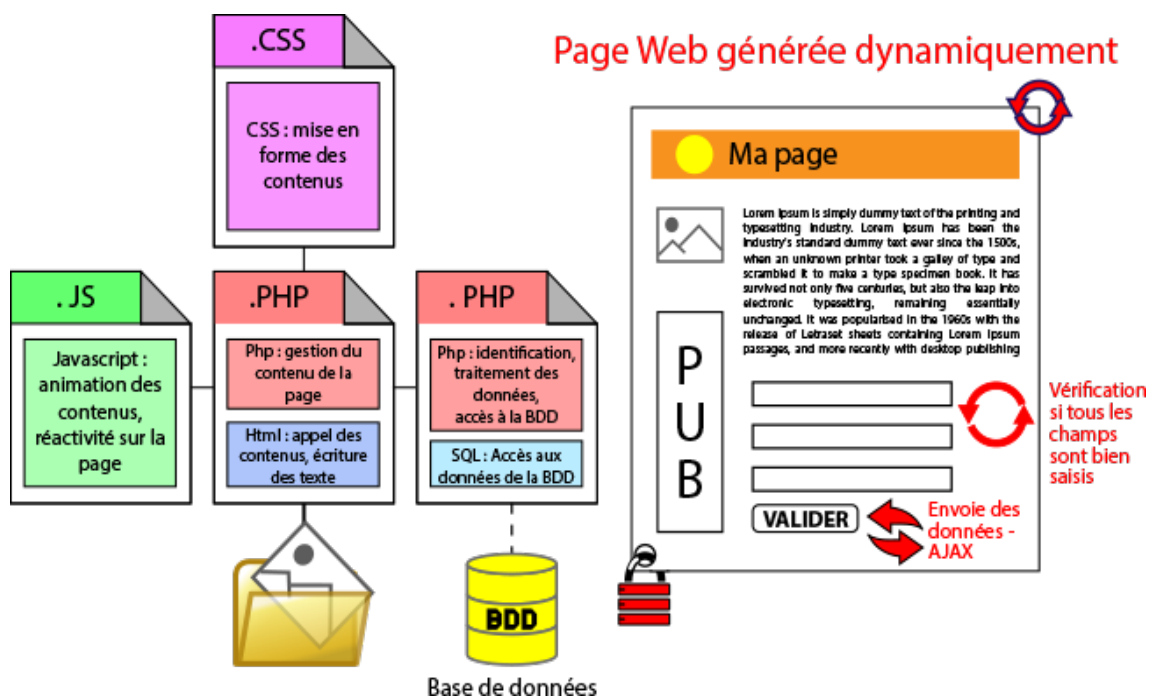


Figure 4.4.1 – L'angage de programmation

Ensuite on a deviser notre application en deux couche : couche server et couche client . pour le cote server on utilises langage nodejs et pour cote client on a travaille avec electronjs . Pour la communication entre ces deux couche (client et server) on a utilise web socket protocole .

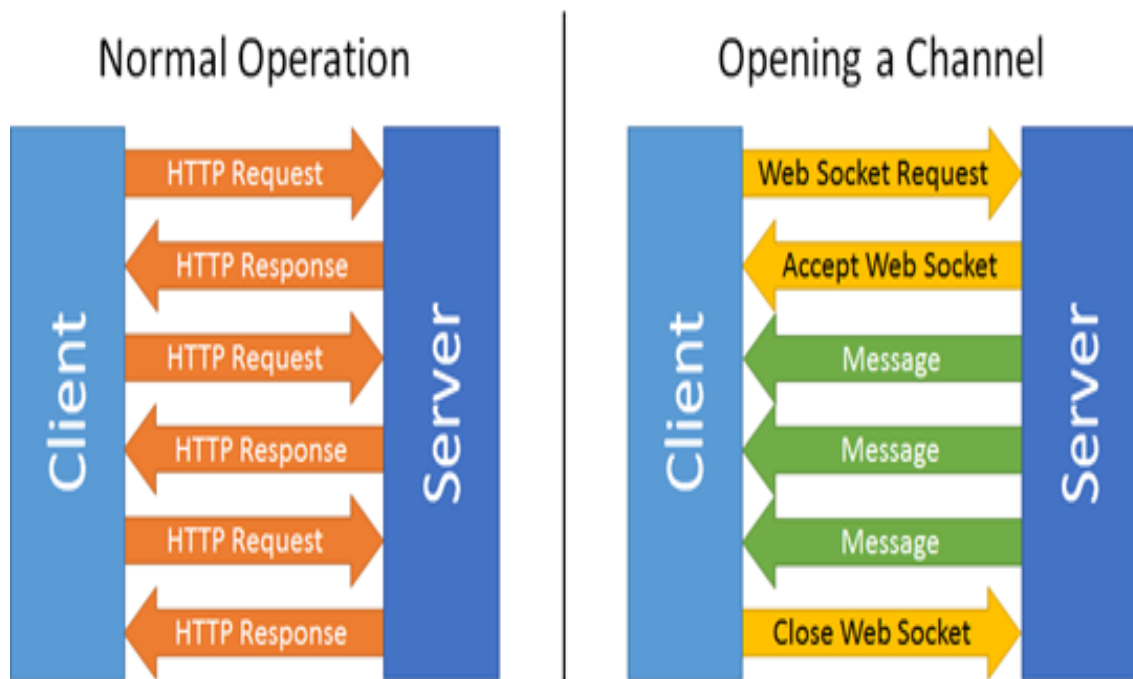


Figure 4.4.2 – L'angage de programmation

4.5 Conclusion

Dans ce chapitre nous avons détaillé les technologies utilisées pour la réalisation de notre projet ainsi que les différents logiciels ,et langages de programmation .

Chapitre 5

REALISATION ET TEST

5.1 Introduction

Dans ce chapitre, nous nous intéressons à la description de quelques interfaces du système implémentés dans le cadre de quelques scénarios d'utilisation par des imprimés écrans.

5.2 Architecture d'application

Nous avons partager notre travaille en deux parties :

partie 1 : Partager un écran avec un utilisateur distant

le partage d'écran c'est juste une vidéo de communication entre deux machines et la vidéo c'est une série continue d'images du bureau de votre ordinateur donc si on fait des captures d'écran continu de votre ordinateur et les envoyes vers un autre ordinateur cela rassemblent exactement au partage d'écran .

Donc c'est ça qu'on a essayé de faire dans la première partie on va prendre des series de capture d'écran d'un ordinateur et les partages vers un autre ordinateur on utilisent web socket .

partie 2 : Effectuer une tâche à partir d'un écran partagé

dans cette partie nous avons essayé d'effectuer l'emplacement à distance de l'événement clé et souris, il existe beaucoup de cle event comme (key up, key down, key presse.....,etc.) et aussi pour souris event (cursor move, left click, right click, scroll, ...,etc.) .

donc lorsque utilisateur position son souris dans n'importe qu'elle place dans son image partage l'événement du souris va se produire au bureau de la personne qui partage leur écran .

5.3 Présentation des interfaces

1. Le premier interface de notre application (One Touche). cette interface affiche la page d'accueil qui a une seule option intéressante c'est le button login..

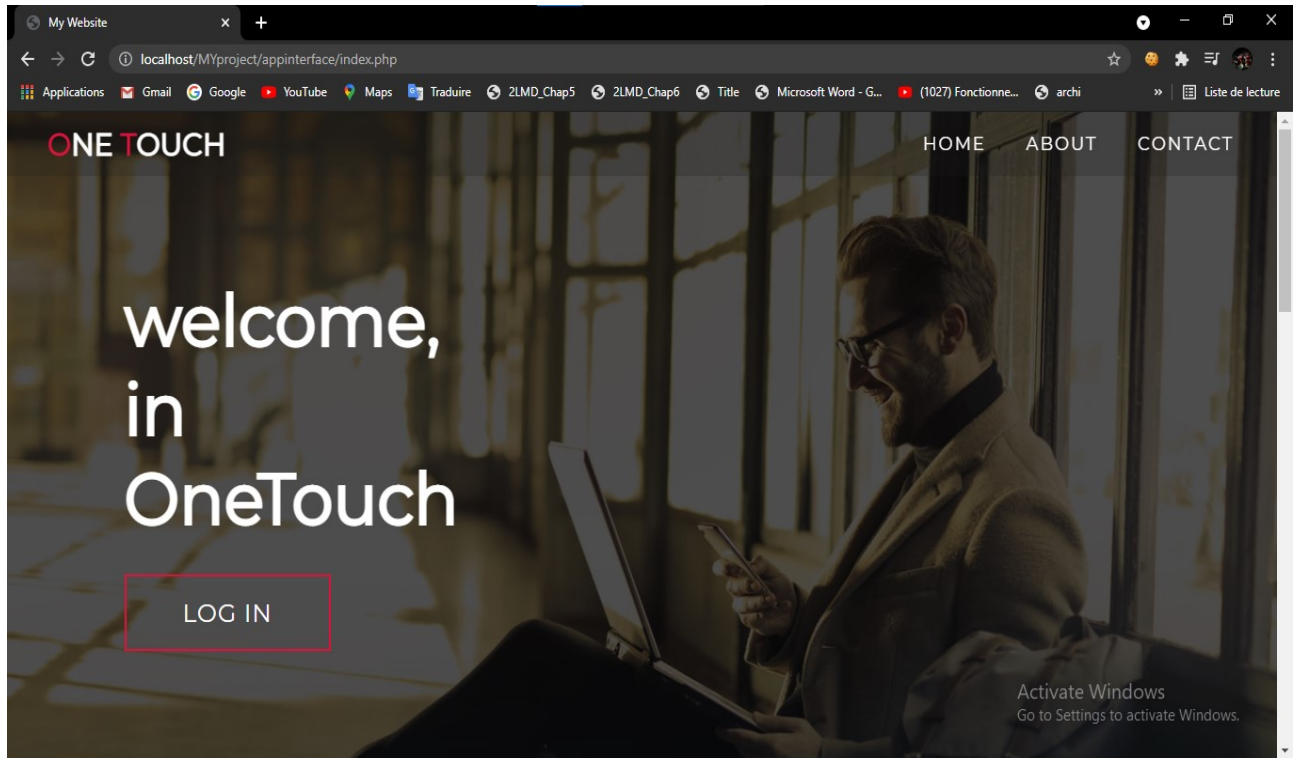


Figure 5.3.1 – home page

2. la deuxième interface s'affiche lorsque vous cliquez sur le bouton login, elle contient un formulaire d'inscription si vous êtes nouveaux ou tout simplement un formulaire qui vous demande d'entrer votre mot de passe et email .

Sign up

name

username

email

passowrd

passowrd confirm

sign up

Already have an account? [login](#)

Sign in

username or email

passowrd

Login

New? [signup](#)

Logout

Figure 5.3.2 – login page

3. la 3e interface contient deux options soient getes accès To desktop qui vos dreiges vers l'interface serveur ou deuxième option share my screen qui vos dreiges vers l'interface client.

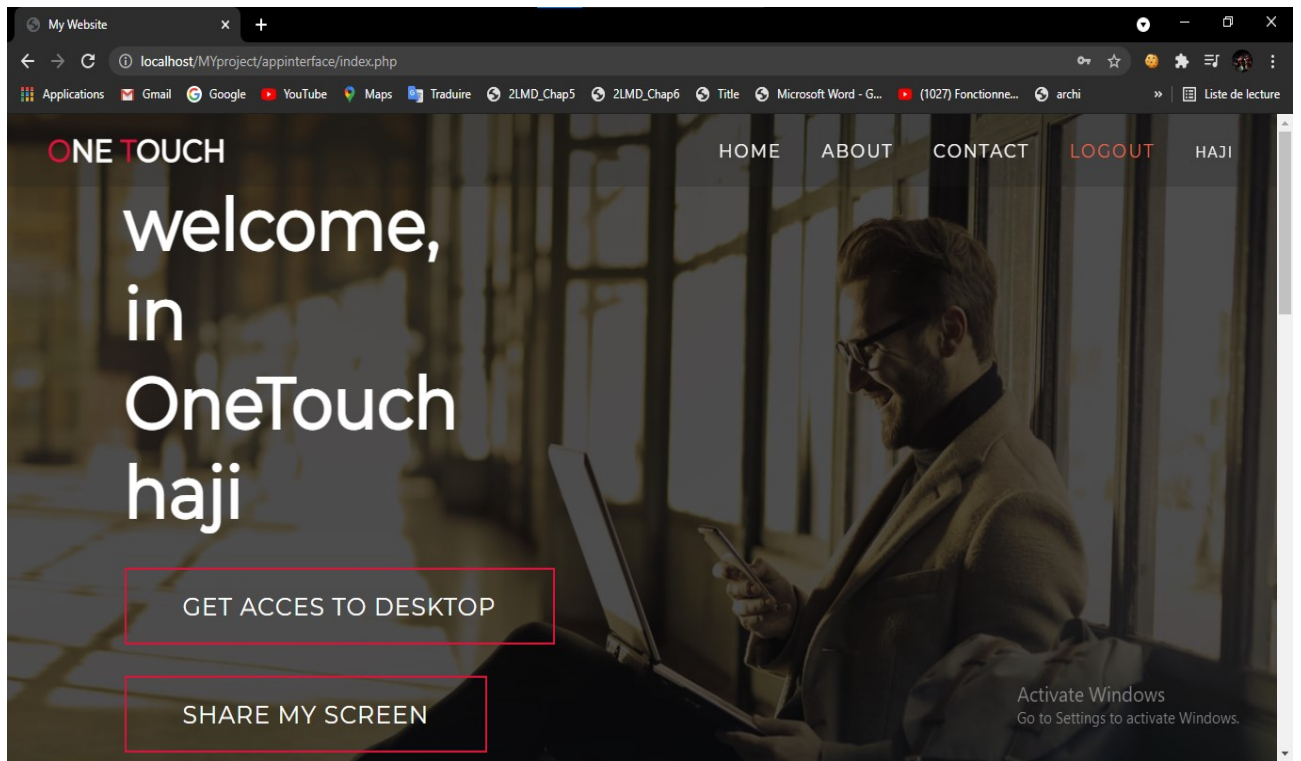


Figure 5.3.3 – option page

4.interface serveur c'est l'interface ou il va y avoir le partage d'écrans, une pompe va vous demander d'entrer l'id du client qui a choisi la 2e option. Lorsque vous entrer l'id correct l'écran du client va s'afficher et vous pouvez contrôler son ordinateur.

!!! Mais avant ça pour pouvoir accéder à cette interface il faut que vous entrer la commande node index.js sur votre terminale pour que le lien marche . puisqu'on est en train de travailler sur un réseau local . le lien est "http ://"+ votre adresse Ip + " : 5000/views".

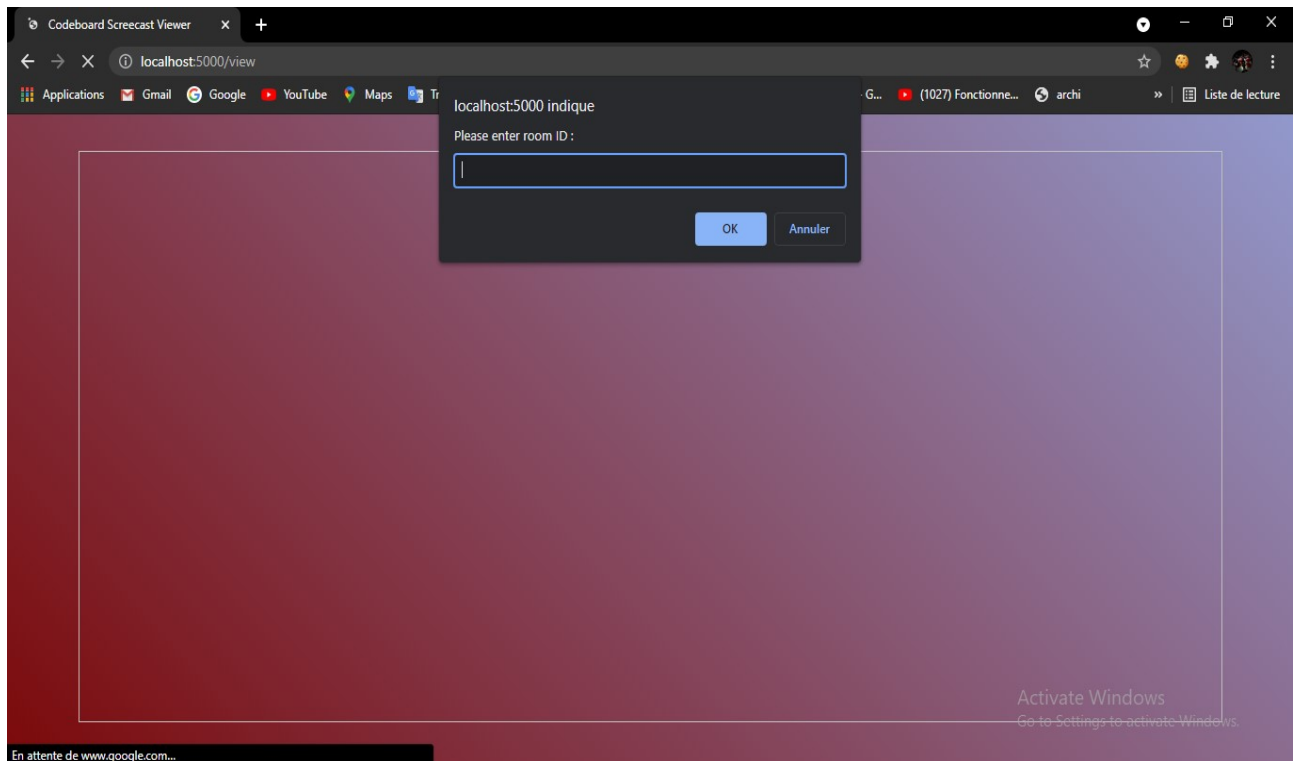


Figure 5.3.4 – server page

5. pour l'interface client c'est juste une simple page qui vous informe d'ouvrier votre application logicielle .

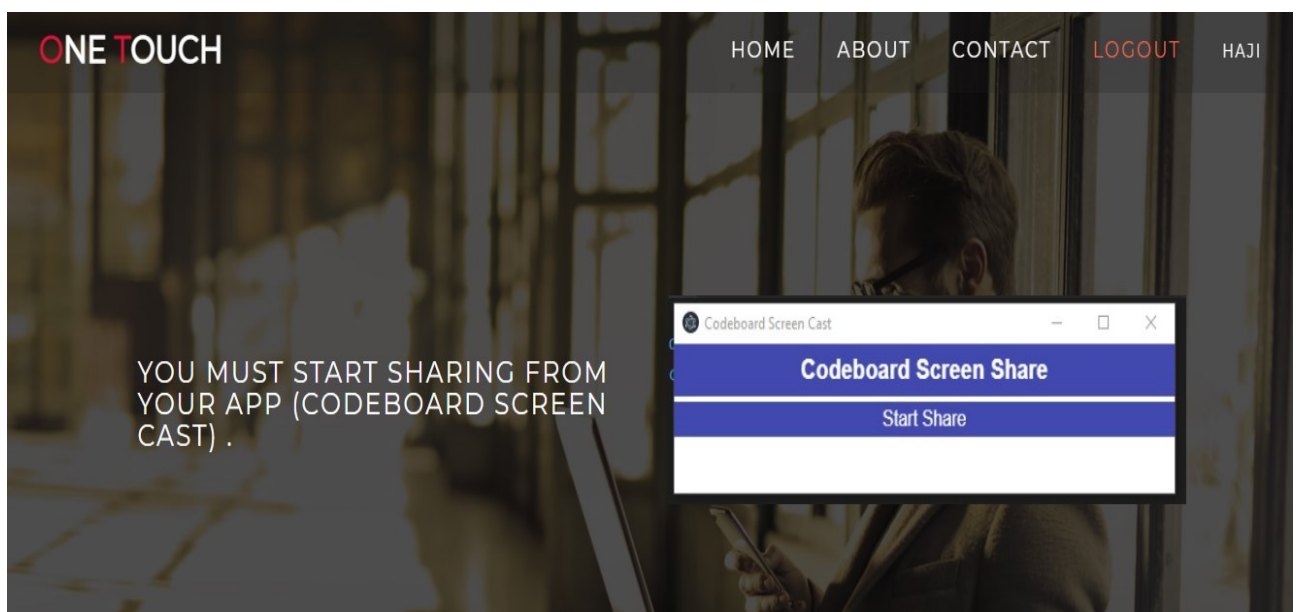


Figure 5.3.5 – client page web

6.pour l'application logiciel elle est tres simple une button pour activer le partage d'ecran et une button pour deactiver le partage .

!!! pour ouvrir l'application vous devez entrer la commande npm start .



Figure 5.3.5 – client app

7. lorsque vous cliquez sur le bouton start share un id va s'afficher, cette id est celui qu'on va mettre dans la page serveur. Et si vous voulez arrêter le partage vous utilisez le bouton stop share.

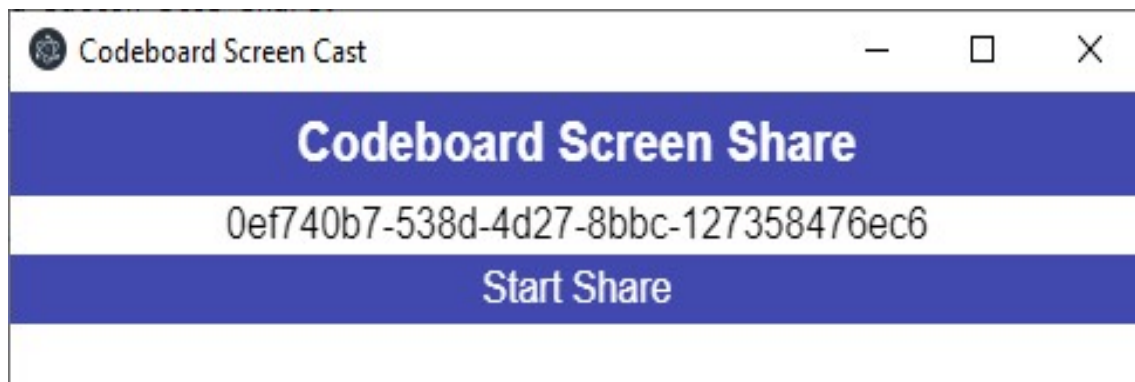


Figure 5.3.6 – client app

5.4 Conclusion

Nous sommes parvenus au terme de ce chapitre dont l'objectif principal était de présenter le résultat final obtenu, à l'aide de quelques imprimés-écrans.

Conclusion Générale

Au bout de notre cursus en licence informatique, nous avons été chargés de réaliser un projet de fin d'études. Ceci nous a amené à découvrir une nouvelle plateforme de développement et enrichir notre savoir et notre expérience.

Il s'agissait au cours du projet d'étudier et de développer une application pour le partage d'écran et l'accès à distance, qui est un des plus grands challenges de cette décennie car il s'articule autour de plusieurs points notamment le contrôle, la maintenance, la sécurité et l'évolutivité. Ces différents points montrent l'importance du contrôle à distance dans un réseau étendu composé de plusieurs réseaux locaux. Au niveau professionnel, le bureau à distance paraît un outil indispensable pour maintenir la rapidité, l'efficacité et la sécurité d'un réseau étendu.

webBibliographie

1. [https ://www.electronjs.org/](https://www.electronjs.org/)
2. [http ://robotjs.io/](http://robotjs.io/)
3. [https ://nodejs.org/en/](https://nodejs.org/en/)
4. [https ://www.python.org/downloads/release/python-395/](https://www.python.org/downloads/release/python-395/)
5. [https ://code.visualstudio.com/](https://code.visualstudio.com/)
6. [https ://www.electronjs.org/](https://www.electronjs.org/)
7. [https ://fr.reactjs.org/tutorial/tutorial.html](https://fr.reactjs.org/tutorial/tutorial.html)
8. [https ://www.npmjs.com/package/shortid](https://www.npmjs.com/package/shortid)
9. [https ://socket.io/docs/v3/client-api/index.html](https://socket.io/docs/v3/client-api/index.html)
10. [https ://expressjs.com/fr/](https://expressjs.com/fr/)
11. [https ://www.npmjs.com/package/screenshot-desktop](https://www.npmjs.com/package/screenshot-desktop)

Bibliographie

- [1] [https ://nodejs.org/en/](https://nodejs.org/en/)
- [2] [https ://www.electronjs.org/](https://www.electronjs.org/)
- [3] [http ://robotjs.io/](http://robotjs.io/)