**REPUBLIQUE TOGOLAISE**

TRAVAIL-LIBERTE-PATRIE

**SOMM**

**SUPERVISEUR**

**Gestion de la Cité du Port : Affectation des Immobiliers**

Mr AGBEBE Kwami Ange

Mr MODEDZI Julien

**MAITRE DE STAGE**

Période du : 15 Juillet 2024 au 1er septembre 2024

Rédigé et présenté par : ALABI Isaac Adeyinka

Etudiant en deuxième année, Tronc Commun

Année Académique : 2023-2024

RAPPORT DE STAGE PRATIQUE EN ENTREPRISE

TYPE DE STAGE : PROGRAMMATION

**Direction Générale Du Port Autonome**

**De Lomé**

Tél: (+228) 22 27 26 27 / 22 23 78 00

E-mail: [togoport@togoport.tg](mailto:)

01 BP 1225 Lomé 01 Togo

Site web : www.togo-port.net

**Institut Africain d’Informatique Représentation du TOGO (IAI-TOGO)**

Tel : (+228) 22 20 47 00

Email : [iaitogo@gmail.com](mailto:iaitogo@gmail.com)

07 BP 12456 Lomé 07, TOGO

Site web : www.iai-togo.com

**SOMMAIRE**

REMERCIEMENTS

En premier lieu, je souhaite exprimer ma profonde gratitude envers Dieu Tout-Puissant pour la santé et la force qu’Il m’a accordées tout au long de cette année et pour Sa présence réconfortante durant cette période de stage.

Je tiens à adresser mes sincères remerciements à :

* **M. AGBETI Kodjo**, Représentant Résident de l’IAI-Togo, pour sa supervision attentive et ses conseils éclairés tout au long de ce projet.
* **M. AMEYIKPO Kossi**, Représentant national de l’IAI-Togo, pour avoir rendu ce stage possible et pour le soutien précieux qu'il a apporté.
* **M. DAVON Essé**, Enseignant à IAI-TOGO, pour l’encadrement technique et pour nous avoir guidé, encouragé et conseillé ;
* **M. MODEDZI**, Mon superviseur, pour l’encadrement minutieux et l’aide précieuse apportée pendant la préparation de ce travail.
* **M. AGBEBE**, Mon maître de stage, pour son accompagnement constant, ses encouragements et ses conseils avisés.

Ma famille, et en particulier mon père, dont le soutien financier et psychologique ainsi que les conseils judicieux ont été d’une aide inestimable.

Enfin, un grand merci à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce projet.

INTRODUCTION

La gestion efficace des biens immobiliers est essentielle pour le bon fonctionnement des activités dans un port, notamment pour assurer un environnement de travail optimal pour le personnel. Dans ce contexte, le Port Autonome de Lomé, principal hub maritime du Togo, joue un rôle stratégique dans l'économie du pays. La cité portuaire, qui abrite le personnel indispensable aux opérations portuaires, nécessite une gestion rigoureuse et transparente de ses biens immobiliers.

Face à ces enjeux, il est impératif de mettre en place un système informatisé qui permettra d’optimiser l’affectation et la gestion des propriétés au sein de la cité portuaire. Le projet intitulé **"Gestion de la Cité du Port : Affectation des Immobiliers"** a pour objectif de développer une solution numérique adaptée aux besoins spécifiques de cette gestion, en vue de rationaliser les processus et d'améliorer l'efficacité opérationnelle.

Ce rapport de stage vise à présenter en détail le développement de ce système informatisé. Il décrit les étapes du projet, depuis l’analyse des besoins jusqu’à la mise en œuvre de la solution, en mettant en lumière les défis rencontrés, les solutions apportées, et les résultats obtenus.

Partie 1 CAHIER DES CHARGES

1.1 Présentation du sujet

Accueillir le personnel dans la cité portuaire du Port Autonome de Lomé nécessite une gestion efficace des logements et des biens immobiliers. Actuellement, cette gestion est réalisée de manière manuelle, ce qui peut entraîner des inefficiences et des erreurs.

Le sujet de notre étude, **"Gestion de la Cité du Port : Affectation des Immobiliers"**, vise à améliorer cette gestion. Nous allons d’abord analyser les méthodes existantes pour identifier leurs limites. Ensuite, nous développerons un système informatisé centralisé pour optimiser l’affectation des logements, le suivi des biens, et la maintenance, afin de rendre la gestion plus efficace et transparente.

1.2 Problématique du sujet

Actuellement, la gestion des biens immobiliers dans la cité portuaire du Port Autonome de Lomé est réalisée de manière manuelle et décentralisée, ce qui entraîne des difficultés majeures. La recherche et l'affectation des logements, le suivi des biens et la gestion des demandes sont souvent laborieux et sujet à des erreurs. Cette méthode traditionnelle, basée sur des documents papier, ne répond plus aux exigences de précision et d'efficacité modernes.

Pour une gestion optimale et transparente des propriétés, il est crucial d'adopter un système informatisé. Ce système doit non seulement faciliter l'affectation des logements mais aussi assurer un suivi rigoureux des biens immobiliers et de leur maintenance.

Les questions suivantes se posent concernant l'organisation et le fonctionnement d'un tel système :

* Comment centraliser et automatiser la gestion des logements ?
* Comment suivre efficacement l'état et l'occupation des biens ?
* Comment réduire les erreurs et les inefficiences dans la gestion ?
* Comment améliorer la transparence et l'accessibilité des informations liées aux biens immobiliers ?

Pour répondre à ces préoccupations, notre projet vise à développer un système informatisé qui permettra d'optimiser la gestion des biens immobiliers au sein de la cité portuaire, en apportant une solution moderne et efficace aux défis actuels.

**1.3 Intérêt du sujet**

1.3.1 Objectifs

Le développement d'un système de gestion pour la cité portuaire du Port Autonome de Lomé présente plusieurs objectifs essentiels pour améliorer l'efficacité et la qualité de la gestion immobilière au sein de la cité. Ces objectifs se déclinent comme suit :

* **Centralisation des informations :** Créer une base de données centralisée et accessible contenant toutes les informations relatives aux biens immobiliers et aux résidents.
* **Optimisation de l'affectation des logements :** Mettre en place des critères clairs et automatisés pour l'attribution des logements, réduisant ainsi les erreurs et les inégalités.
* **Suivi et gestion de la maintenance :** Développer un système pour planifier, suivre et prioriser les tâches de maintenance afin d'assurer un environnement sûr et bien entretenu pour les résidents.
* **Transparence et traçabilité :** Améliorer la transparence des processus de gestion des biens immobiliers et garantir la traçabilité des opérations, facilitant ainsi les audits et la résolution des litiges.
* **Rapports et statistiques :** Fournir des outils pour générer des rapports et des statistiques, aidant à la prise de décision et à l'optimisation continue des processus.

1.3.2 Résultats attendus

Les résultats attendus de ce projet sont multiples et variés, touchant plusieurs aspects de la gestion des biens immobiliers et de la satisfaction des résidents :

* **Efficacité opérationnelle accrue :** Grâce à la centralisation et à l'automatisation des processus, les tâches administratives seront réalisées plus rapidement et avec moins d'erreurs.
* **Satisfaction des résidents améliorée :** Un système de gestion efficace et réactif contribuera à une meilleure satisfaction des résidents, qui bénéficieront d'un service de maintenance plus rapide et d'une affectation plus équitable des logements.
* **Meilleure prise de décision :** Les rapports et statistiques générés par le système fourniront des données précises et pertinentes pour prendre des décisions éclairées concernant la gestion des biens immobiliers.
* **Transparence et responsabilisation :** La traçabilité des opérations et la transparence accrue permettront de renforcer la confiance des résidents et de faciliter les audits internes et externes.

PARTIE 2 PRE-PROGRAMMATION

Avant la conception de l’application proprement dite, il est nécessaire de comprendre et de mieux cerner le sujet. La phase d’analyse étant primordiale dans la réalisation d’un projet informatique. Cette partie, nous permettra d’étudier le sujet et ainsi proposer les solutions au problème dont on fait face.

2.1. Etude de l’existant

Actuellement, l’affectation des villas au sein de la cité portuaire du Port Autonome de Lomé est gérée de manière manuelle et rudimentaire à l’aide de supports papier. Ce processus traditionnel présente plusieurs inefficiences et limitations qui impactent la gestion globale des demandes.

Le processus actuel se déroule comme suit :

* **Soumission de la Demande :** Un employé du port qui souhaite obtenir une villa doit soumettre une demande manuscrite adressée au Directeur Général (DG). Cette demande comprend des informations personnelles et des motifs pour l’acquisition d’un logement.
* **Évaluation Initiale :** Une fois la demande reçue, le DG la transmet au service concerné pour une première évaluation. Ce service vérifie la conformité des informations et évalue l'éligibilité de la demande en fonction des critères établis.
* **Validation par le Service :** Après l'évaluation initiale, le service concerné compile une liste des demandes validées et la renvoie au DG. Cette étape inclut la révision des demandes et la préparation d’un rapport sur l’état des demandes.
* **Décision Finale :** Le DG examine les demandes validées par le service et prend la décision finale d’accepter ou de rejeter chaque demande. Les décisions sont ensuite communiquées aux employés concernés.

2.2 Critique de L’existant

Ce processus présente plusieurs problèmes clés :

* **Manque de Traçabilité :** Le suivi des demandes et des décisions est difficile à gérer en raison de l'utilisation de documents papier, ce qui complique la traçabilité et l’historisation des demandes.
* **Efficience Limitée :** La gestion manuelle des demandes peut entraîner des erreurs, des pertes de temps et des retards dans le traitement des demandes.
* **Risque d’Erreur :** Le passage des documents entre différents acteurs du processus augmente le risque d’erreurs et de pertes d’informations.

Pour remédier à ces problèmes, il est crucial de mettre en place un système informatisé qui centralise et automatise le processus de demande et d’affectation des villas. Ce système devrait permettre une gestion transparente, efficace et facile à suivre de toutes les demandes, avec une traçabilité complète des actions et des décisions prises.

2.3. Planning prévisionnel de réalisation

Tableau 1:PLANNING PREVISIONNEL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Taches | Date de début | Date de fin | Durée |
| 1 | Intégration dans l’entreprise et prise en main du theme | 15/07/2024 | 19/07/2024 | 5 jours |
| 2 | Choix du Framework et début de l’élaboration du cahier de charges | 22/07/2024 | 24/07/2024 | 3 jours |
| 3 | Début de l’élaboration des diagrammes et poursuite du cahier de charges | 25/07/2024 | 30/07/2024 | 5 jours |
| 4 | Développement et révision continuelle des diagrammes | 31/07/2024 | 19/08/2024 | 20 jours |
| 5 | Test et finalisation du rapport de stage | 20/08/2024 | 29/08/2024 | 10 Jours |

2.4 ETUDE DETAILLE DE LA SOLUTION

2.4.1. Présentation de la méthode d’analyse

Pour notre étude, nous avons opté pour le langage UML (Unified Modeling Language) et la démarche 2TUP (2 Track Unified Process).

2.4.1.1- UML

Figure 1: logo uml

Logo d’UML Le langage de modélisation unifié, en anglais Unified Modeling Langage est un langage de modélisation graphique à base de pictogrammes (dessins figuratifs stylisés ayant fonction de signe) conçu pour fournir une méthode normalisée pour visualiser la conception d’un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

2.4.1.2 Les diagrammes UML

UML propose 14 types de diagrammes. Ces diagrammes sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d’un projet tout au long de son cycle de vie.

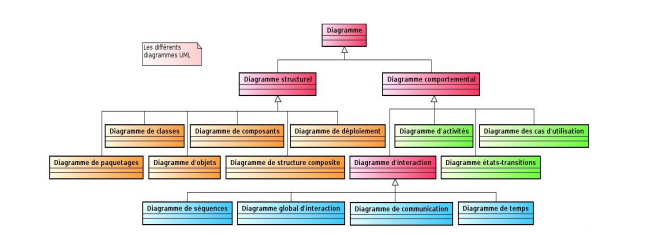


Figure 2 logo diagrammes:

❖ Les diagrammes structurels ou statiques (07) qui rassemblent :

• Le diagramme de classes (intervenant dans le système)

• Le diagramme d’objets (instances de classes)

• Le diagramme de composants (du point de vue physique)

• Le diagramme de déploiement,

• Le diagramme des paquetages (ensembles de définitions)

• Le diagramme de structure composite (relations entre les composants)

• Le diagramme de profils.

❖ Les diagrammes comportementaux (03) qui rassemblent :

• Le diagramme des cas d’utilisation (acteurs et interactions avec le système)

• Le diagramme d’états-transitions (description sous forme de machine à états finis

Du comportement du système et ses composants)

• Le diagramme d’activité (description sous forme d’activités du comportement du

Système et ses composants)

❖ Les diagrammes d’interaction ou dynamiques (04) qui rassemblent :

• Le diagramme de séquences

• Le diagramme de communication (échange de messages entre les objets)

• Le diagramme global d’interaction (variante du diagramme d’activité)

• Le diagramme de temps (variation d’une donnée au cours du temps)

2.4.2. Présentation de l’outil de modélisation : Draw io

L’outil retenu pour la modélisation est draw.io. C’est une application gratuite en ligne (éventuellement hors ligne), accessible via son navigateur, qui permet de dessiner des diagrammes ou des organigrammes à l’aide d’une interface simple et facile d’utilisation. Cet outil propose à l’utilisateur de concevoir toutes sortes de diagrammes, de dessins vectoriels, de les enregistrer puis de les exporter

raw io 'ion des diagrammes :

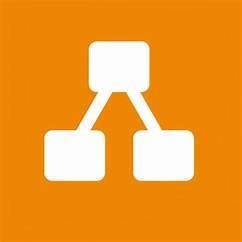


Figure 3:logo draw.io

2.4.3 Identification des acteurs et Tableau récapitulatif de leurs actions

* *Identification des acteurs*
* **Employé**
* **Rôle :** Soumet des demandes d'affectation de villa.
* **Responsabilités :** Remplir et soumettre le formulaire de demande de villa.
* **Service Concerné**
* **Rôle :** Évalue les demandes soumises par les employés.
* **Responsabilités :** Vérifier les informations fournies, évaluer l'éligibilité, et préparer une liste des demandes validées.
* **Directeur Général (DG)**
* **Rôle :** Prend la décision finale sur les demandes d'affectation.
* **Responsabilités :** Examiner les demandes validées par le service concerné, approuver ou rejeter les demandes, et communiquer les décisions aux employés.
* **Administrateur Système**
* **Rôle :** Gère et maintient le système informatisé de gestion des villas.
* **Responsabilités :** Configurer le système, gérer les utilisateurs, assurer le bon fonctionnement de l'application, et résoudre les problèmes techniques.
* **Développeur**
* **Rôle :** Développe et met en œuvre le système de gestion des villas.
* **Responsabilités :** Concevoir, coder et tester les fonctionnalités du système, intégrer les exigences des utilisateurs et s'assurer que le système répond aux besoins définis.
* **Responsable Maintenance**
* **Rôle :** Gère la maintenance et les réparations des villas.
* **Responsabilités :** Planifier et coordonner les travaux de maintenance, tenir à jour les dossiers de maintenance des villas, et informer les utilisateurs des travaux effectués.
* *Tableau récapitulatif des acteurs et leurs actions*

|  |  |  |
| --- | --- | --- |
| Cas d'Utilisation | Acteurs | Actions |
| Soumettre une demande d'affectation de villa | Employé | - Remplir et soumettre le formulaire de demande de villa. |
| Évaluer les demandes | Service Concerné | - Vérifier les informations fournies dans la demande  - Évaluer l'éligibilité  - Préparer une liste des demandes validées à transmettre au DG. |
| Prendre une décision sur les demandes | Directeur Général (DG) | - Examiner les demandes validées par le service concerné  - Approuver ou rejeter les demandes  - Communiquer les décisions aux employés. |
| Gérer le système informatisé | Administrateur Système | - Configurer et maintenir le système  - Gérer les utilisateurs  - Assurer le bon fonctionnement et résoudre les problèmes techniques. |
| Planifier et coordonner les travaux de maintenance | Responsable Maintenance | - Planifier les travaux de maintenance  - Coordonner les réparations  - Tenir à jour les dossiers de maintenance  - Informer les utilisateurs des travaux. |

Tableau 2:cas d'utilisation et acteur

2.4.4 Diagrammes des cas d’utilisation

Présentation des diagrammes :

* **Cas de l’employé :**

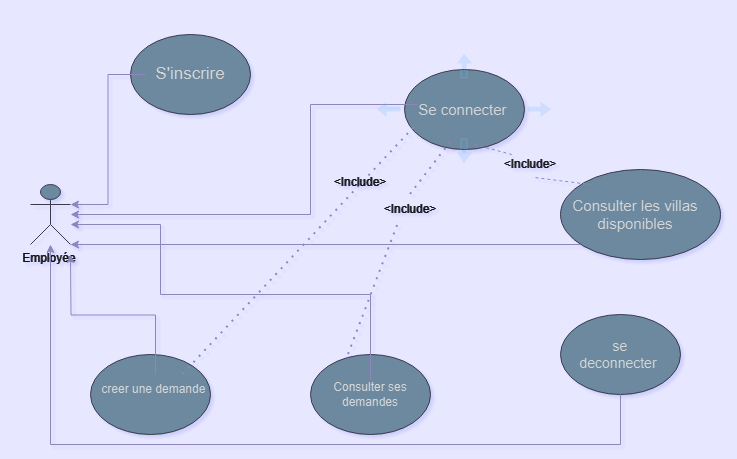


Figure 4: Cas d'utilisation employé

* **Cas du directeur**

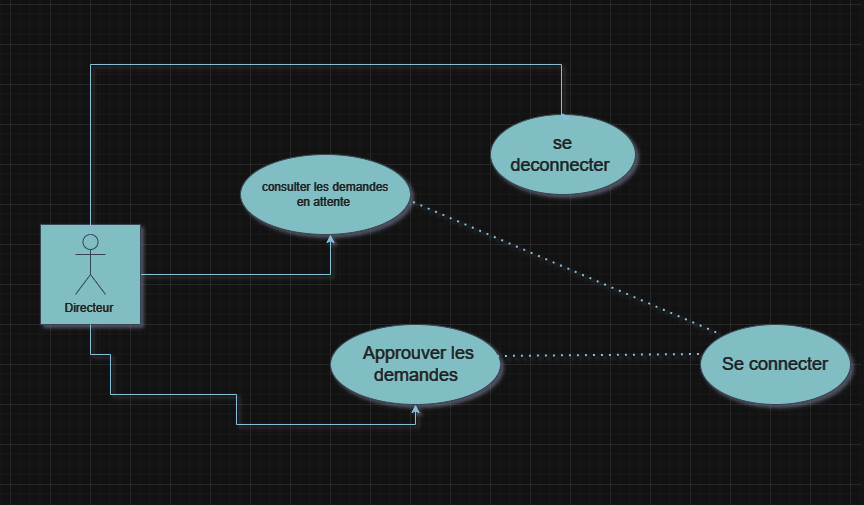


Figure 5:cas du Directeur

* **Cas de l’Admin**
* **Cas de L’admin**

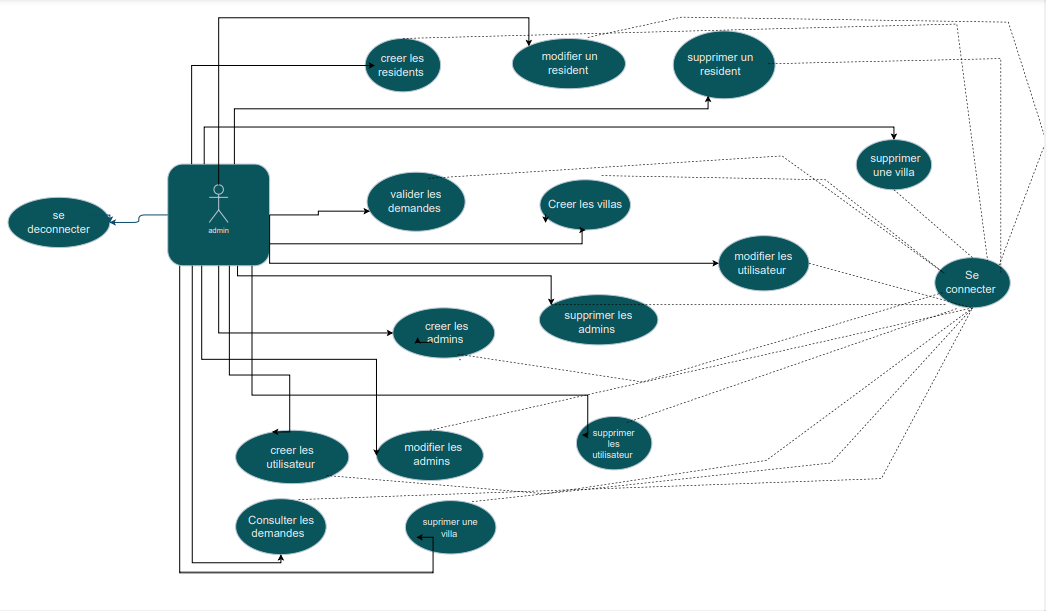


Figure 6:cas Admin

* **Cas du service concerné**

2.4.5 Diagrammes d’activité

* Cas d’utilisation «  **Se connecter**  »

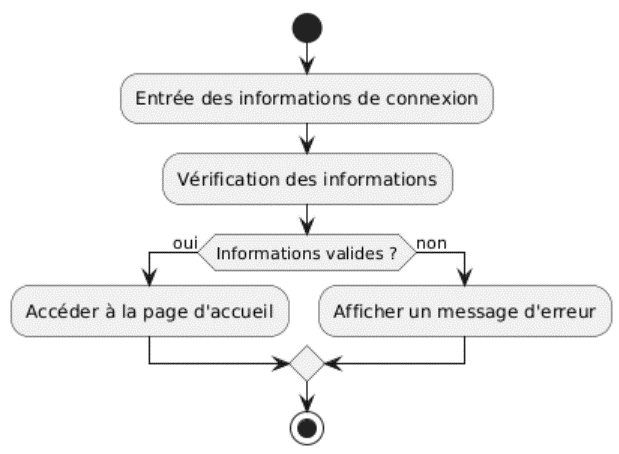


Figure 7:activé-Se connecter

* Cas d’utilisation « **Faire Une demande**  »

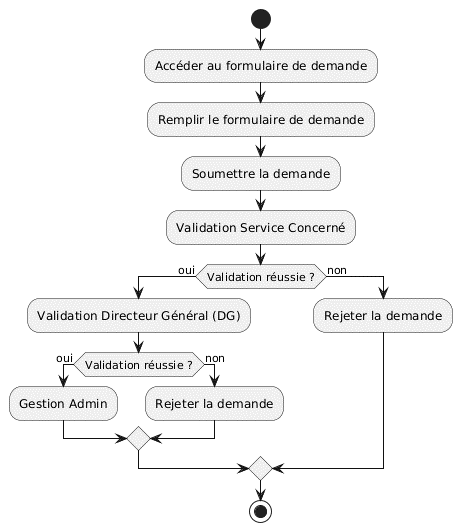


Figure 8: activité-demande

* Cas d’utilisation « **Ajouter Un Résident** »

Figure 9:Activité-residents

2.4.6 Diagrammes de séquence

Présentation des diagrammes :

* Cas d’utilisation « **S’authentifier** »

Figure 10: séquence authentifié

* Cas d’utilisation « **faire une demande**  »

Figure 11:sequence demande

* Cas d’utilisation « **Ajouter un résident**  »

Figure 12:séquence résident

2.4.7 Description textuelle des cas d’utilisation

1. Cas d’utilisation **s’authentifier**

Tableau 3:cas auth

|  |
| --- |
| SOMMAIRE D’IDENTIFICATION |
| Titre : **S’authentifier**  Résumé : Ce cas d’utilisation permet à tout utilisateur de s’identifier et d’accéder aux fonctionnalités de l’application.  Acteurs : Administrateur, Employé, Service, Directeur  Auteur : ALABI Isaac Adeyinka  Date : 15 Juillet 2024  Date de mise à jour : 15 Juillet 2024  Version : 1.0  Préconditions :   * Le système est opérationnel * La page d’authentification est accessible |
| DESCRIPTION DES ENCHAINNEMENTS |
| Scénario Nominal |
| 1. L'utilisateur saisit son email dans le champ email 2. L’utilisateur saisit son mot de passe dans le champ mot de passe 3. L'utilisateur clique sur le bouton <<Connexion>> 4. Le système effectue une vérification syntaxique (SA1) 5. Le système effectue une vérification de l’email et du mot de passe dans la base de données (SE1) 6. Le système redirige l’utilisateur vers la page d’accueil et affiche le menu correspondant à son profil utilisateur |
| Scénario Alternatif |
| **Scénario alternatif SA1 : Les paramètres de connexion sont incorrects**   * -Le SA1 se déclenche au point 4 du scénario nominal. Le système affiche un message <<champ invalide>> * -Le système renvoie un message qui indique que la connexion a échoué * -Les enchainements reprennent au point 2 du scenario nominal |
| **Scénarios d’exception** |
| **Scenario d'exception SE1 : Les paramètres de connexions sont incorrects pour 3ème fois (pas de correspondance avec les données de la base de données)**  Ce scénario est déclenché au point 5, lorsque l’utilisateur saisit des informations incorrectes 3 fois d’affilée   * le système affiche à l’utilisateur qu’il a effectué sa troisième tentative * le compte de l’utilisateur est verrouillé et le scénario nominal se termine par un échec |
| **Post condition :**  L’utilisateur est connecté et a accès au menu correspondant à son profil Utilisateur |

1. Cas d’utilisation **« créer une demande  »**

Tableau 4:cas demande

|  |
| --- |
| SOMMAIRE D’IDENTIFICATION |
| Titre : Créer une Demande Résumé : Ce cas d’utilisation permet aux utilisateurs de soumettre une demande pour acquérir une maison. Acteur : Employé  Auteur : ALABI Isaac Adeyinka Date de création : 15 Juillet 2024 Date de la dernière modification : 15 Juillet 2024 Version : 1.0  Préconditions : Accéder au formulaire de création de demande |
| DESCRIPTION DES ENCHAINNEMENTS |
| **Scénario nominal :** |
| 1. L’utilisateur se connecte à l'application et accède à la page de création de demande. 2. Le système lui affiche le formulaire de demande. 3. L’utilisateur charge la lettre de demande 4. Le système vérifie les informations et enregistre la demande avec un statut initial de "en attente" (SD2, SD3). 5. Le système affiche un message de confirmation indiquant que la demande a été créée avec succès. |
| Scénario alternatif : |
| 1. SD1 : Les champs obligatoires ne sont pas remplis   Le formulaire affiche que les données sont obligatoires. Le déroulement reprend au point 2 du scénario nominal   1. SD2 : Les informations fournies sont incorrectes   La page le signale à l’utilisateur. Le déroulement reprend au point 2 du scénario nominal.   1. SD3 : Le fichier de la lettre de demande est invalide ou corrompu   Le système signale l’erreur et demande de télécharger un fichier valide. Le déroulement reprend au point 2 du scénario nominal. |
| **Scénario d’exception :**   * **SE1 :** L’utilisateur a interrompu la soumission  Il ferme la fenêtre ou quitte la page. Le cas d’utilisation se termine en erreur. |

2.4.8 Diagrammes de classe

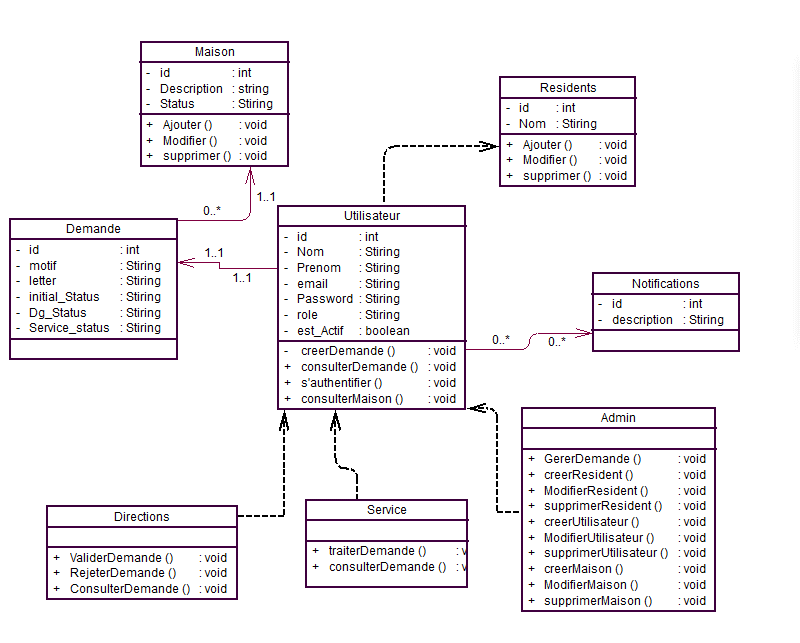


Figure 13: classe diagramme

Partie 3 : REALISATION ET MISE EN ŒUVRE

3.1 Matériels et logiciels utilisés

3.1.1 Matériels

Pour la réalisation de notre projet de stage nous avons utilisé un ordinateur portable dont les caractéristiques sont les suivantes :

Tableau 5:caracteristique matérielles

|  |  |  |
| --- | --- | --- |
| N° | Rubriques | Caractéristiques |
| 1 | Fabricant | Lenovo |
| 2 | Processeur | Intel(R) Core(TM) i5-5300UCPU@ 2.30GHz 2.30 GHz |
| 3 | Ram | 8Go |
| 4 | Disque dur | 350Go SSD |
| 5 | Système  d’exploitation | Windows 10 64 bits, processeur x64 |

3.1.2 Logiciels

Pour mener à bien le projet, nous avons utilisé plusieurs logiciels clés, chacun offrant des fonctionnalités spécifiques pour faciliter le développement et la gestion de notre application.

* **XAMPP**

XAMPP est un environnement de serveur local gratuit et open-source, développé par Apache Friends. Il permet de créer et de gérer un serveur web local sur un PC, ce qui est essentiel pour le développement et les tests de projets web. XAMPP inclut :

* **Serveur HTTP Apache** : Permet de servir les fichiers web au navigateur.
* **Base de données MariaDB** : Fournit un système de gestion de bases de données relationnelles.
* **Interpréteurs PHP et Perl** : Permettent d'exécuter des scripts PHP et Perl sur le serveur.

XAMPP est prisé pour sa simplicité d'installation et sa configuration, ainsi que pour ses fonctionnalités complètes qui répondent aux besoins de nombreux développeurs.



Figure 14:logo de Xampp

* Visual Studio Code

Visual Studio Code est un éditeur de code gratuit crée par Microsoft. Il peut être utilise avec une variété de langage programmation et il les prend immédiatement en charge.



Figure 15:logo Vs code

Plusieurs langages sont inclus par défaut, par exemple JavaScript, Type Script, CSS et HTML, mais d’autres extensions de langage peuvent être trouvées et téléchargées gratuitement à partir de VS Code Marketplace. Il inclut aussi des flux travails guidés pour GitHub Actions.

Les raisons de choix de cet éditeur :

* Ses fonctionnalités (couleurs de syntaxe, marques d’erreurs et d’avertissements, visualiseur de structure et numéros de ligne) servent de guide et de support lors du codage ;
* Sa compatibilité avec GitHub qui permet faciliter des dépôts ;
* Une recherche de référence dans le code qui est très avancée et qui est d’une grande aide.

 **PHPMyAdmin**

PHPMyAdmin est une application web de gestion pour de base de données pour les systèmes de base de données MySQL réalisée en PHP et distribuée sous licence GNU GPL. Elle offre de grands atouts pour l’administration de la base de données.



Figure 16:PhpMyAdmin

Nous avons eu à l’utiliser du fait qu’elle est d’accès facile et elle nous a permis la manipulation de la base données, la possibilité d’effectuer et de vérifier nos requêtes SQL.

 **Langage PHP**

PHP (HyperText Préprocesseur) est un langage de script à usage populaire qui est particulièrement adapté au développement web. Ses syntaxes sont largement inspirées du langage C, de Java et de Perl, avec des améliorations spécifiques. Le but du langage est d’écrire rapidement des pages HTML dynamiques. Ce qui distingue PHP des langages de script comme le JavaScript, est que le code est exécuté sur le serveur, générant ainsi le HTML, qui sera ensuite envoyé au client.



Figure 17:php Logo

Notre choix c’est porté sur le langage PHP car de base nous réalisons une plateforme web et ce langage fait partie des meilleurs pour ce domaine. De par sa facilité d’apprentissage, de manipulation et d’intégration facile au Framework Symfony elle a été d’une aide très précieuse. Il existe d’autres langages comme Python, Java, Rust ; PHP fut notre choix car elle a une très grande communauté et une très bonne compatibilité avec le front-end HTML/CSS.

* **Github**

Figure 18:logo de github

GitHub est une plateforme de gestion de versions basée sur Git, utilisée pour le contrôle de version et la collaboration dans le développement logiciel. GitHub offre plusieurs fonctionnalités clés pour le développement de projets :

* **Gestion des versions** : Permet de suivre les modifications apportées au code source, de revenir à des versions antérieures et de gérer les différentes branches de développement.
* **Collaboration** : Facilite le travail en équipe avec des fonctionnalités telles que les pull requests, les commentaires sur le code, et les revues de code.
* **Intégration continue** : Prend en charge les workflows d'intégration continue et de déploiement continu (CI/CD) avec des outils comme GitHub Actions.
* **Suivi des problèmes** : Inclut un système de suivi des problèmes et des tâches, ce qui aide à gérer les bugs, les demandes de fonctionnalités et les tâches en cours.

GitHub est largement utilisé dans la communauté de développement logiciel en raison de sa robustesse, de son interface conviviale et de sa capacité à gérer des projets de toutes tailles, des projets personnels aux projets open source de grande envergure.

* **Laravel**

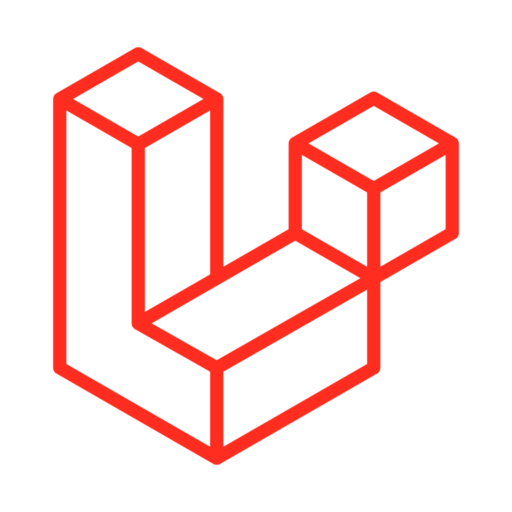


Figure 19:logo laravel

Laravel est un framework PHP open-source conçu pour simplifier le développement d'applications web. Il offre une structure élégante et expressive avec des outils puissants pour la gestion des bases de données, l'authentification des utilisateurs, et la création d'interfaces utilisateur. Laravel facilite le développement rapide avec ses fonctionnalités intégrées telles que l'ORM Eloquent, le système de routage, et la gestion des tâches planifiées. Il est également doté d'un système de migration de base de données et d'une protection contre les menaces courantes comme les injections SQL et les attaques CSRF. Ideal pour créer des applications web robustes et sécurisées.

* **Bootstrap**

Bootstrap est un framework web qui facilite la création des sites internet et d’applications web. Il contient des modèles HTML et CSS qui permettent de créer rapidement des formulaires et des boutons, des outils de navigation et autres.



Figure 20:logo Bootstrap

3.2 Sécurité de l’application

Pour assurer la sécurité de notre application Laravel, nous avons mis en place les mesures suivantes :

* **Authentification et Autorisation** : Utilisation de l'authentification Laravel avec mot de passe hashé (bcrypt) et gestion des rôles via middlewares pour contrôler l'accès.
* **Protection contre les Attaques** : Prévention des injections SQL grâce à Eloquent ORM, protection contre XSS et CSRF avec échappement des entrées et tokens CSRF.
* **Sécurisation des Données** : Cryptage des données sensibles et sécurisation des sessions avec cookies chiffrés.
* **Validation des Données** : Validation rigoureuse des entrées utilisateur pour éviter les données malveillantes.
* **Sécurisation des Routes** : Utilisation de middlewares pour protéger les routes sensibles et contrôle d'accès basé sur les rôles.
* **Mises à jour et Surveillance** : Application régulière des mises à jour de sécurité et surveillance des activités suspectes.

3.3 Evaluation Financière

Tableau 6 evaluation financiere:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N°** | **Désignation** | **Description** | **Coût Horaire** | **Nombre d’heures** | **Montant (CFA)** | **Sources** |
| 1 | Chef de projet | Ingénieur Logiciel pour la gestion du projet | 7 000 | 336 | 2 352 000 | Salaire moyen au Togo (source locale) |
| 2 | Développeur | Analyste Programmeur pour le développement | 2 200 | 300 | 660 000 | - |
| 3 | Web Designer | Conception des interfaces utilisateur | 1 800 | 140 | 252 000 | - |
| 4 | Formation | Formation des utilisateurs sur l'utilisation de l'application | 5 500 | 6 | 33 000 | - |
| 5 | Ordinateur | Ordinateur de bureau pour l’installation de l’application | - | - | 400 000 | HP Pro 300 G6 MT Intel Core i3 10è Génération 4Go / 1To, Écran 22 Pouces - Fournisseur local au Togo |

3.4 Présentation de l’application

3.4.1 Mise en place de la base de données

J’ai eu à utiliser comme base de données MySql sous PHPMyAdmin .

Le script qui m’a permis dele créer est : Create database Imobilliers .

* **Script de connexion :**

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=imobiliers

DB\_USERNAME=root

DB\_PASSWORD=

* **Creation des tables :**

--

-- Base de données : `imobilliers`

--

CREATE TABLE `houses` (

`id` bigint(20) UNSIGNED NOT NULL,

`description` text DEFAULT NULL,

`status` enum('available','occupied','maintenance') NOT NULL,

`image` varchar(255) DEFAULT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

CREATE TABLE `requests` (

`id` bigint(20) UNSIGNED NOT NULL,

`user\_id` bigint(20) UNSIGNED NOT NULL,

`house\_id` bigint(20) UNSIGNED NOT NULL,

`service\_status` enum('en attente','approved','rejected') NOT NULL DEFAULT 'en attente',

`dg\_status` enum('en attente','approved','rejected') NOT NULL DEFAULT 'en attente',

`initial\_status` enum('en attente','approved','rejected') NOT NULL DEFAULT 'en attente',

`motif` varchar(255) DEFAULT NULL,

`letter` varchar(255) NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

CREATE TABLE `residents` (

`id` bigint(20) UNSIGNED NOT NULL,

`user\_id` bigint(20) UNSIGNED NOT NULL,

`nom` varchar(255) NOT NULL,

`prenom` varchar(255) NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

CREATE TABLE `users` (

`id` bigint(20) UNSIGNED NOT NULL,

`Nom` varchar(255) NOT NULL,

`Prenom` varchar(255) NOT NULL,

`email` varchar(255) NOT NULL,

`email\_verified\_at` timestamp NULL DEFAULT NULL,

`password` varchar(255) NOT NULL,

`role` enum('user','service','Dg','admin') NOT NULL DEFAULT 'user',

`remember\_token` varchar(100) DEFAULT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL,

`is\_active` tinyint(1) NOT NULL DEFAULT 0

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

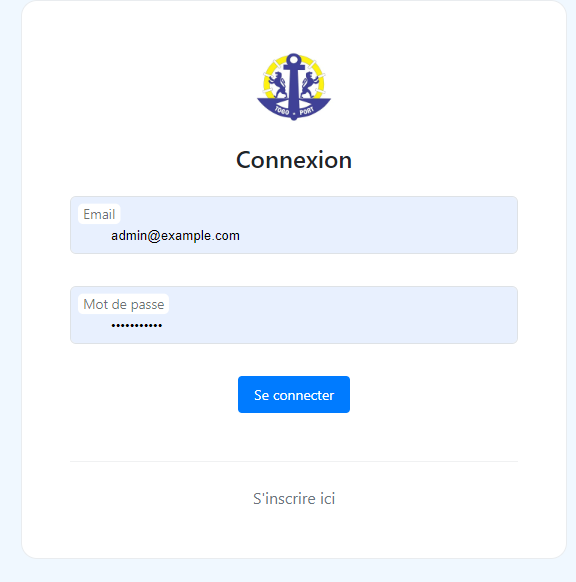
3.4.2 Plan de Navigation

3.4.3 Quelques masques de saisies

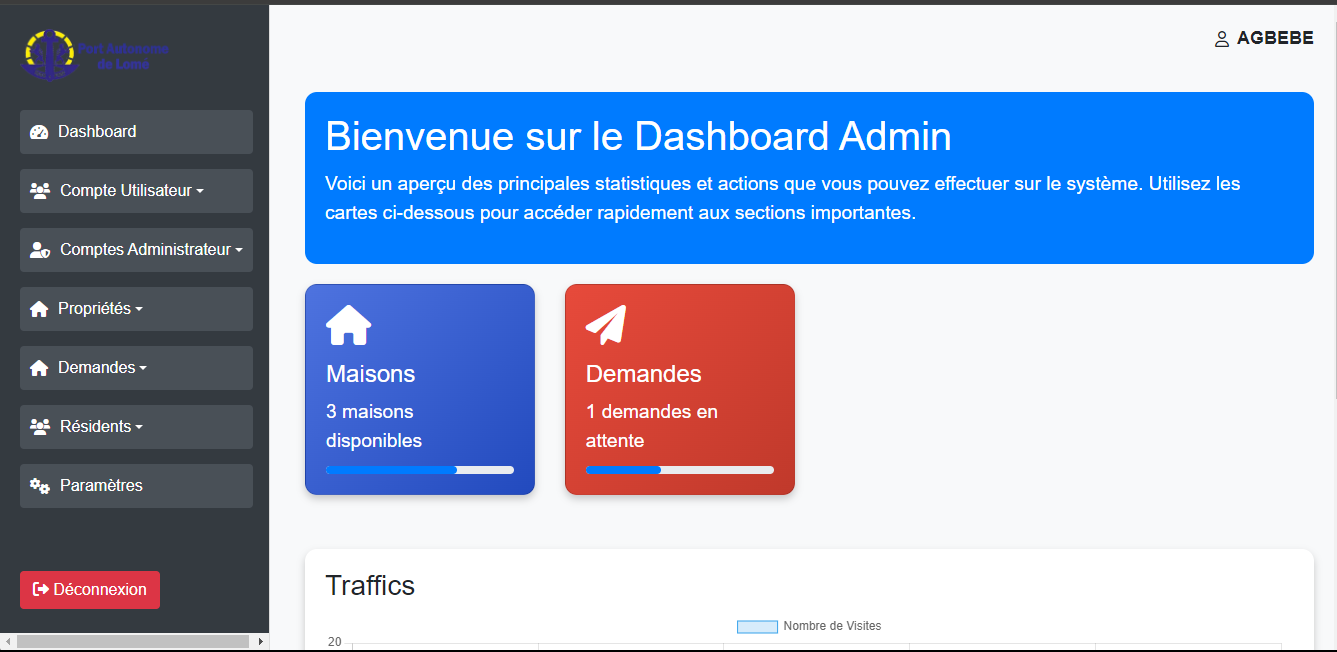
* **Page d’accueil du site**



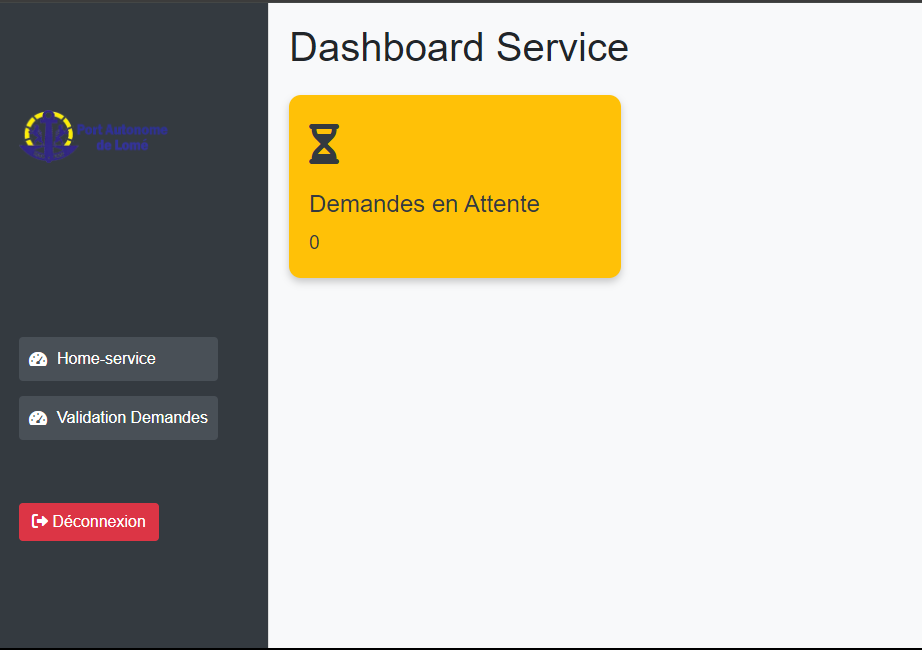
* **Page d’authentification des utilisateurs du site**



* **Page d’acceuil de l’administrateur**



* **Page d’acceuil du service concerné**



* **Page d’acceuil du service concerné**