

Projet Fil-Rouge

Jalon 1

Application Web

*{ Gestion d'incident d'un parc informatique
entre des usagers et un service informatique }*

*Projet de mise en situation professionnelle dans le
cadre du RNCP 31678 de niveau 6*

Concepteur Développeur d'application

ESRP 2isa Millau



PATRICK NARDI

CDA07 | EPCF 1

27/06/2023

2isa

UNIVERSITÉ D'ARTS ET DE MÉTIERS
Recherche professionnelle

Dans la formation CDA, ce projet de gestion des incidents du parc informatique de l'association AMIO vise à faciliter la maintenance du parc matériel, en simplifiant la communication entre les utilisateurs et le Service Informatique. Il se compose de deux applications : une application web destinée aux utilisateurs et une application desktop exclusivement pour le SI.

L'application web de ce Jalon un, permet de signaler les incidents en ouvrant des tickets et de dialoguer de ces incidents avec le SI dans un espace sécurisé.

Les premières étapes ont impliqué l'identification des besoins via le QQQQCP¹ et la définition des fonctionnalités par des User Stories. À chaque étape, pour l'amélioration en continu de mon code, j'ai employé le cycle PDCA².

Dans un deuxième temps, le maquettage a été réalisé avec l'outil Figma³ en m'inspirant du site de l'association, avec comme objectif une IHM⁴ conviviale. En parallèle, j'ai rédigé les scripts de création de la base de données en SQL et je l'ai déployé sur le SGBD⁵ MariaDB avec un jeu de données correspondant.

Après validation de la maquette par le Product Owner, j'ai développé la logique métier en PHP avec le Framework Laravel et le pattern MVC⁶. Avec la sémantique HTML combinée avec le CSS et le JavaScript, j'ai conçu une interface pour une expérience utilisateur optimale. Enfin, en suivant mes plans de validation, j'ai testé les fonctionnalités, la robustesse et la sécurité de mon code de manière itérative, cela avec une veille technologique sur les bonnes pratiques et l'intégration des nouvelles connaissances de la formation.⁷

Accès au site web : lab004.s1.2isa.org ;

login usager : 133@amio.com ou 111@amio.com,

login SI : 115@amio.com ;

mot de passe : +1Mot@Passe

1 Le QQQQCP (Qui, Quoi, Où, Quand, Comment, Pourquoi) est une méthode d'analyse contextuelle

2 La méthode PDCA (Plan-Do-Check-Act) ou Roue de Deming est un cycle d'amélioration continue

3 Figma© outil de design collaboratif

4 L'IHM (Interface Homme-Machine) est le point de contact entre un utilisateur et un système informatique

5 Système de Gestion de Base de Données

6 Le MCV (Modèle-Vue-Contrôleur) est un modèle de conception logicielle

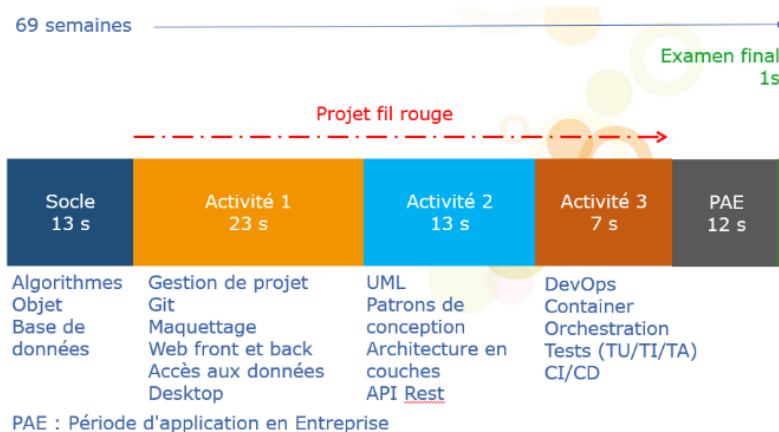
7 L'ensemble de ce projet est réalisé avec la participation de ChatGPT©

Table des matières

1.	Liste des compétences du référentiel couvertes par le projet	3
2.	Méthodologie de travail.....	4
3.	Analyse du cahier des charges, et expression du besoin	4
3.1.	Qui Quoi Où Quand Comment Pourquoi	4
3.1.	User stories	8
3.2.	Planification	11
4.	Spécifications techniques et de sécurité	12
4.1.	Phase de maquettage	12
4.2.	Phase de création de la base de données.....	13
4.3.	Phase de Conception et développement avec le Framework PHP Laravel	14
4.4.	Phase de déploiement	19
5.	Réalisations du candidat comportant les extraits de code les plus significatifs, et en les argumentant, y compris pour la sécurité	19
5.1.	Phase de maquettage	19
5.1.1.	Exemple d'IHM du maquettage	20
5.1.2.	Charte graphique et model de Vue.....	21
5.2.	Phase de création de la base de données.....	21
5.3.	Phase de Conception et développement avec le Framework PHP Laravel	23
5.3.1.	Conception et développement de l'espace sécurisé	23
6.	Plan de tests et de déploiement	30
6.1.	Checklist des fonctionnalités demandées.....	30
6.2.	Checklist des contraintes techniques.....	30
7.	Veille, effectuée sur les vulnérabilités de sécurité	31
8.	Description d'une situation de travail ayant nécessité une recherche	31
9.	Conclusion.....	32
10.	Annexes.....	34
10.1.	Annexe 1, code du fichier, resources/views/tickets.blade.php :.....	34
10.2.	Annexe 2, Schéma du pattern MVC pour fonctionnalité de connexion	36
10.3.	Annexe 3, Analyse préliminaire pour les User-Stories :.....	38

Le Fil-Rouge représente la première mise en situation professionnelle de notre groupe CDA07 dans le cadre de la formation de Concepteur Développeur d'Applications (CDA). Au cours des sept derniers mois passés au sein de l'ESRP 2isa, j'ai pu développer des compétences professionnelles transférables en milieu d'entreprise. Ce projet Fil Rouge me permet de mettre en pratique les différentes compétences acquises ces derniers mois, ainsi que celles issues de mes expériences professionnelles antérieures. Il s'intègre parfaitement dans mon parcours de formation actuel et vise à valider mes aptitudes en matière de maquettage, de développement avec PHP, HTML, CSS, JavaScript, SQL, ainsi que l'utilisation du Framework Laravel avec son module backend Fortify dédié à la gestion des authentifications et le pattern MVC.

Ce projet est divisé en deux parties distinctes : une application web et une autre desktop. La partie concernant l'application web est celle dont je me suis occupé jusqu'à présent, la partie desktop sera abordée ultérieurement, couvrant ainsi la formation jusqu'au stage PAE. Les réalisations que j'ai effectuées reposent sur les connaissances transmises par les formateurs tout au long du socle et de l'activité une. Le projet a démarré le 14 avril 2023 avec une date de livraison prévue pour le 26 mai 2023.



1. Liste des compétences du référentiel couvertes par le projet

Le concepteur développeur d'applications est chargé de concevoir et développer des services numériques en respectant les normes et standards de l'industrie. Il doit avoir des compétences en conception d'interfaces utilisateur, développement de bases de données et applications multicouches réparties, tout en s'adaptant aux évolutions technologiques et en respectant les recommandations de sécurité.

Pour valider ce titre je dois par mes livrables attester de mon savoir-faire correspondant aux blocs de compétences définis dans le Répertoire national des certifications professionnelles pour le RNCP31678⁸. A savoir pour ce jalon :

- Maquetter une application.
- Développer des composants d'accès aux données.
- Développer la partie front-end d'une interface utilisateur web.
- Développer la partie back-end d'une interface utilisateur web.
- Mettre en place une base de données.
- Développer des composants dans le langage d'une base de données.
- Concevoir une application.

8 .RNCP31678 - TP - Concepteur développeur d'applications - France Compétences (francecompetences.fr)

- Développer des composants métier.

2. Méthodologie de travail

J'ai utilisé la méthode du cycle PDCA ou Roue de Deming (Planifier, Faire, Vérifier, Agir) lors des étapes de conception et de développement du projet afin d'assimiler les nouvelles connaissances acquises grâce à la formation ainsi que celles provenant de ma veille concurrentielle et de les déployer sur le projet. Cette approche m'a permis :

- Une amélioration en continu de la qualité et du respect des bonnes pratiques dans mon code.
- Une logique de développement avec des tests en continu sur les réponses fonctionnelles, les attendus sécuritaires et l'expérience utilisateur.

Roue de Deming, PDCA :« Le cycle PDCA sert à transformer une idée en action et l'action en connaissance »⁹ (Pour plus d'information sur la méthode : [Wikipédia](#))

3. Analyse du cahier des charges, et expression du besoin

J'ai utilisé les méthodes QQQQCP pour analyser le cahier des charges et mettre en évidence les attentes du projet. Cet acronyme représentant une technique d'analyse contextuelle visant à comprendre une situation donnée en posant les bonnes questions (Qui, Quoi, Où, Quand, Comment, Pourquoi). Principalement utilisée lors de la phase de cadrage d'un projet, cette méthode m'a aidé à mieux appréhender cette problématique spécifique en obtenant des informations précises et pertinentes. Cela m'a permis d'apporter des solutions plus efficaces plutôt que de me précipiter dans la résolution du problème. En favorisant ma compréhension du projet, elle a également contribué à maintenir mon attention sur les objectifs à atteindre.

3.1.Qui Quoi Où Quand Comment Pourquoi

Pour plus d'information sur la méthode : [Wikipédia](#) et [Appvizer.fr](#)

Qui ?	Qui est concerné par ce projet ?	<ul style="list-style-type: none"> • Le personnel de l'association AMIO (client fictif), spécialisée dans le domaine de la formation professionnelle en informatique découpé en deux rôles (utilisateurs et service informatique) • Les stagiaires du groupe CDA07 de la formation de concepteur développeur d'application • Les Product Owner (formateurs)
Quoi ?	Quelle est la problématique à résoudre ?	<p>Automatiser la gestion du parc informatique, en permettant de :</p> <ul style="list-style-type: none"> • Fluidifier la maintenance du parc informatique. • Faciliter la gestion des incidents matériels. • Assurer une traçabilité des problèmes (historique des messages, historique des évolutions de l'avancement, historiques des intervenants, pas sur le type et le sujet de l'incident) • Permettre une communication efficace entre les utilisateurs et le service informatique. • Former les utilisateurs du service informatique aux deux applications • Tracer les incidents • Attribuer une référence à l'incident • Résoudre les problèmes matériels

9 .Gerald J. Langley, Ronald Moen, Kevin M. Nolan, Thomas W. Nolan, Clifford L. Norman, Lloyd P. Provost, The Improvement Guide : A Practical Approach to Enhancing Organizational Performance, JOSSEY-BASS, 2009, 490 p. (ISBN 978-0-470-19241-2), p. 97

	Quelles sont les fonctionnalités ?	<p>La première application devra :</p> <ul style="list-style-type: none"> • Se dédier aux usagers de la structure. • Être un portail web. • Permettre d'ouvrir un ticket pour signaler des incidents techniques. • Permettre de faire la liaison entre l'incident en cours et le service informatique via le ticket. • Permettre à l'utilisateur de suivre l'évolution de la résolution de son problème. • Créer un espace sécurisé pour chaque ticket. Cet espace devra : donner la possibilité d'envoyer des précisions ou des demandes d'informations complémentaires sur l'incident en temps réel. • Avoir une interface simple et intuitive pour une expérience utilisateur optimale. <p>La deuxième application (phase 2 du projet), destinée aux professionnels de la maintenance du parc informatique de l'entreprise, devra :</p> <ul style="list-style-type: none"> • S'installer sur les postes des professionnels du service de maintenance informatique. • Permettre de faire le suivi et l'inventaire du matériel. • Faciliter le suivi des contrats de maintenance sur certains matériels actuellement effectués par des entreprises externes.
	Quels livrables sont demandés ?	<ul style="list-style-type: none"> • Une base de données • Un jeu de test cohérent et représentatif • Une application Laravel • Un résumé du projet • Un rapport • Une présentation avec son support
Où ?	Où les livrables doivent être déposés ?	Plateforme AMIO-FIT : Cours : Fil Rouge (2isa.eu)
	Où le projet doit être déployé ?	<p>Host Lab004</p> <p>HostName lab004.2isa.org</p> <p>Port 22004</p> <p>User lab004</p> <p>ForwardAgent yes</p> <p>Mot de passe : 3265lab004</p>
	Où le projet doit-être présenté ?	Dans les locaux d'AMIO
Quand ?	Début	Vendredi 14 avril 2023
	Évaluation individuelle	Semaine du 26 juin 2023
	Fin	Vendredi 23 juin 2023 au plus tard
Comment ?	Quelles contraintes techniques ?	<ul style="list-style-type: none"> • Code de qualité • Portail web : <ul style="list-style-type: none"> ○ Langage backend : PHP ○ Framework : LARAVEL ○ IHM sous : HTML5, CSS3 et JavaScript. ○ Protocole réseau : HTTP2.0 ○ Déploiement sur un serveur de production fourni par l'équipe pédagogique • Application lourde : <ul style="list-style-type: none"> ○ Elle se découpera en deux parties (client et serveur) <ul style="list-style-type: none"> ▪ Application serveur, API : <ul style="list-style-type: none"> - Langage : C# - Framework : .NET

		<ul style="list-style-type: none"> - Une première application développée avec le framework .NET en C# pour faire l'API. - Permettre la communication avec la deuxième application client ▪ Client lourd : <ul style="list-style-type: none"> - Langage : C# - Framework : .NET - IHM sous : Winforms
	Quels formats des documents demandés ?	<ul style="list-style-type: none"> • Tous les livrables devront être fournis au format numérique (PDF et Word) • Un résumé du projet : <ul style="list-style-type: none"> ○ 200 à 250 mots (pour donner une vision du projet à un jury) ○ Fourni au format Word (docx) et Pdf. ○ Il donnera une vision de son projet à un jury qui ne le connaît pas. ○ Le résumé doit être pertinent, bien présenté et rédigé sans faute d'orthographe. • Le fichier devra être nommé : <ul style="list-style-type: none"> ▪ PRENOM_NOM_Projet_fil_rouge_Jalon1_Resume.zip ▪ PRENOM_NOM_Projet_fil_rouge_Jalon1_Resume.docx ▪ PRENOM_NOM_Projet_fil_rouge_Jalon1_Resume.pdf • Un dossier projet fourni sous la forme d'une archive (zip). • Respecte du plan type : <ol style="list-style-type: none"> 1. Liste des compétences du référentiel qui sont couvertes par le projet 2. Cahier des charges, expression des besoins 3. Spécifications fonctionnelles du projet 4. Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité 5. Réalisations du candidat comportant les extraits de code les plus significatifs, et en les argumentant, y compris pour la sécurité 6. Plan de tests et de déploiement 7. Description de la veille, effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité 8. Description d'une situation de travail ayant nécessité une recherche, effectuée par le candidat durant le projet, à partir de sites francophones ou anglophones • Support de présentation orale au format type Powerpoint.
	Quelles sont les contraintes rédactionnelles ?	<ul style="list-style-type: none"> • Description de la veille sur les vulnérabilités de sécurité : <ul style="list-style-type: none"> ○ Décrire comment vous avez effectué la veille de sécurité : les sites et les mots clés utilisés et indiquer les vulnérabilités trouvées et éventuellement les failles potentielles corrigées. • Description de la situation de travail, ayant nécessité une recherche basée sur un ou des sites anglophones. La description doit concerner un problème technique ou une nouvelle fonctionnalité mise en œuvre, dans le cadre du projet. : <ul style="list-style-type: none"> ○ Décrire le besoin d'information, et indiquer comment vous avez effectué la recherche : les mots clés des recherches utilisés et la liste des sites retournés. ○ Pour finir précisez les critères de sélection du (ou des) site(s) et indiquer la solution trouvée et si elle a pu être mise en œuvre. • Longueur du dossier de projet hors annexes : 30 pages, soit environ 48750 caractères espaces non compris.
	Quelles sont les contraintes temporelles de la présentation orale ?	<p>La présentation orale se déroulera en deux parties :</p> <ol style="list-style-type: none"> 1. 40 minutes pour la présentation de la réalisation du projet 2. 30 minutes d'entretien technique avec le jury.
	Quelles sont les contraintes de la présentation orale ?	<ul style="list-style-type: none"> • Un support de présentation • Une présentation professionnelle • Objectifs et les compétences visées annoncées. • Contexte situé. • Reformulation des contours du projet et de ses objectifs fonctionnels par type d'utilisateurs • Détail de l'architecture employée • Mise en évidence des principales étapes du travail réalisé

		<ul style="list-style-type: none"> • Présentation des résultats les plus pertinents pour démontrer les compétences en jeu • Argumenterez les choix • Exposez les difficultés rencontrées et les solutions apportées • Présentez l'organisation de la gestion du projet • Conclure en faisant le point sur l'état d'avancement et dresser le bilan à cette étape
Pourquoi ?	Quels buts pédagogiques recherche le projet ?	<p>Attester des compétences 1 et 2 du RNCP 31678 par un projet individuel :</p> <ol style="list-style-type: none"> 1. Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité : <ol style="list-style-type: none"> a. Maquetter une application. b. Développer une interface utilisateur de type desktop. c. Développer des composants d'accès aux données. d. Développer la partie frontend d'une interface utilisateur web. e. Développer la partie backend d'une interface utilisateur web. 2. Concevoir et développer la persistance des données en intégrant les recommandations de sécurité <ol style="list-style-type: none"> f. Concevoir une base de données. g. Mettre en place une base de données. h. Développer des composants dans le langage d'une base de données.

3.1. User stories

Les User-Stories m'ont permis de décrire les besoins fonctionnels de l'application de gestion des incidents de l'association AMIO, en me mettant à la place des utilisateurs finaux. Même si je n'ai pas utilisé la Méthode AGILE dans sa totalité, les User-Stories en tant qu'artéfact sont une solution pertinente pour un projet digital. J'ai identifié six User-Stories pour les utilisateurs et six autres pour le personnel du service informatique, que j'ai nommé TechHotline. (Voir [Anx3](#) pour ma préanalyse).

Pour plus d'information sur la méthode : [Atlassian.com](https://atlassian.com) et kanbios.fr

N°	Qui	Quoi	Pourquoi	Comment
US001	En tant que personnel d'AMIO	je souhaite me connecter au portail web de gestion des incidents informatiques	afin de d'accéder comme usager à l'espace sécurisé de gestion des incidents matériels (connexion & accès aux fonctionnalités)	en utilisant une interface web
US002	En tant que personnel d'AMIO connecté au portail web de gestion des incidents informatiques,	je souhaite me déconnecter du portail web de gestion des incidents informatiques	afin de quitter l'espace sécurisé de gestion des incidents matériels	
US003	En tant que personnel d'AMIO connecté au portail web de gestion des incidents informatiques,	je souhaite ouvrir un nouveau ticket	afin que le SI résolve mon incident	en donnant un sujet à l'incident, en typant l'incident suivant une liste de type, en communiquant avec le SI sur l'incident via un échange de message texte
US004	En tant que personnel d'AMIO connecté au portail web de gestion des incidents informatiques,	je souhaite dialoguer sur un ticket avec le SI	afin que d'échanger des informations sur l'incident avec le SI	
US005	En tant que personnel d'AMIO connecté au portail web de gestion des incidents informatiques,	je souhaite visualiser l'état d'avancement de mon incident matériel	afin d'être informé de sa résolution	

N°	Qui	Quoi	Pourquoi	Comment
US006	En tant que personnel d'AMIO connecté au portail web de gestion des incidents informatiques,	je souhaite accéder à la liste de mes incidents matériels	afin d'accéder à mon incident matériel	
US107	En tant que personnel du service SI	je souhaite me connecter au portail web de gestion des incidents informatiques	afin de d'accéder comme personnel du service informatique à l'espace sécurisé de gestion des incidents matériels (connexion & accès aux fonctionnalités)	en utilisant une interface web
US108	En tant que personnel du service SI connecté au portail web de gestion des incidents informatiques,	je souhaite me déconnecter du portail web de gestion des incidents informatiques	afin de quitter l'espace sécurisé de gestion des incidents matériels	
US109	En tant que personnel du service SI connecté au portail web de gestion des incidents informatiques,	je souhaite clôturer un incident matériel	afin d'informer l'auteur de la finalisation de son incident	
US110	En tant que personnel du service SI connecté au portail web de gestion des incidents informatiques,	je souhaite écrire un message sur un incident matériel non clôturé	afin de communiquer avec son auteur	
US111	En tant que personnel du service SI connecté au portail web de gestion des incidents informatiques,	je souhaite accéder à la liste des incidents matériels	afin d'accéder à chacun de ces incidents même chose que la colonne précédente ?	en pouvant filtrer sur les états d'avancement, les dates, les utilisateurs, les matériels, les types de panne, les intervenants du SI
US212	En tant que personnel d'AMIO	je souhaite un portail web simple et intuitif	afin de ne pas faire d'erreurs entre avec des informations non pertinentes manque un mot ?	

N°	Qui	Quoi	Pourquoi	Comment
US213	En tant qu'association AMIO	je souhaite un portail web sécurisé aux attaques informatiques	afin de ne pas perdre, corrompre ou se faire voler les données	

3.2. Planification

Afin de préparer les différentes tâches et d'obtenir une réponse efficace aux besoins du client, j'ai utilisé Jira.com pour dresser la liste des tâches à réaliser, pour intégrer les fonctionnalités et les nouvelles connaissances dans l'application.

<p>Vue de la feuille de Route sur Jira</p>	 <p>The screenshot shows a Jira project backlog with the following tasks:</p> <ul style="list-style-type: none">PFRP1-1 Rédiger un rapport (Progress bar)PFRP1-5 QQQQCCP (Status: EN COURS)PFRP1-9 Introduction, présentation (Status: A FAIRE)PFRP1-10 Réception du CdC (Status: TERMINÉ(E))PFRP1-2 Déployer une base de données avec un jeu cohérent de données (Progress bar)PFRP1-11 script SQL de création suivant MPD (Status: TERMINÉ(E))PFRP1-12 Jeu de données (Status: A FAIRE)PFRP1-13 requêtes SQL (Status: A FAIRE)PFRP1-3 Développer une application laravel (Progress bar)PFRP1-8 User stories (Status: EN COURS)PFRP1-4 Rédiger un résumé du projetPFRP1-14 Maquettage (Progress bar)PFRP1-15 envoie au Product Owner (Status: A FAIRE)PFRP1-17 Réaliser la maquette (Status: EN COURS)PFRP1-18 Dev app (Progress bar)PFRP1-19 Dev éléments (Status: A FAIRE)
---------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Pour concevoir et développer ce projet de portail web, j'ai suivi cette planification générale :

1. J'ai élaboré le QQQQCP afin d'identifier les attentes du projet et ses objectifs.
2. J'ai construit les User-Stories pour spécifier chaque fonctionnalité et je les ai vérifiées avec le Product Owner pour leur pertinence et leur adéquation à la demande.
3. J'ai réalisé des maquettes en utilisant l'application web figma.com, en m'inspirant du site de l'association amio-millau.fr et en respectant les exigences de simplicité et de cohérence.
4. J'ai créé les vues et sous-vues en utilisant le moteur de vue Blade, en exploitant ses fonctionnalités pour les sous-vues, les formulaires et les directives, tout en respectant la sémantique HTML.
5. J'ai développé le style en suivant les maquettes, en adoptant une approche mobile-first et en découpant les fichiers en fonction du type de vue.
6. J'ai réalisé la base de données sous MariaDB en utilisant le SGBD, en incluant un jeu de données correspondant.
7. J'ai implémenté les fonctionnalités une par une en suivant la logique du Modèle-Vue-Contrôleur (MVC) dans le framework Laravel.
8. J'ai testé l'application avec le Product Owner pour recueillir ses critiques, identifier d'éventuelles améliorations, effectuer une revue de code et valider son déploiement en production.

4. Spécifications techniques et de sécurité

4.1.Phase de maquettage

Pour répondre à la demande du client et concevoir **une interface simple, intuitive et offrant une expérience utilisateur optimale**, j'ai abordé la phase du Jalon 1 en suivant les bonnes pratiques. J'ai effectué des recherches sur le web comme par exemple sur ionos.fr ou sur openedition.org, cela complété par des questions posées à ChatGPT et des interactions avec le Product Owner. Cette démarche m'a permis d'analyser le maquettage en fonction de quatre concepts clés. Pour chacun de ces concepts, j'ai défini des spécifications afin de répondre aux besoins **particuliers** de l'association AMIO.

Concept de Maquettage	Spécificité technique de réalisation du maquettage
Architecture de l'information	<ul style="list-style-type: none"> • Structure claire et logique pour le contenu en utilisant des catégories, des sous-catégories et un menu de navigation. • Hiérarchie visuelle claire.
Stratégie de contenus	<ul style="list-style-type: none"> • Informations pertinentes et utiles, sans informations superflues. • Langage simple et compréhensible, sans jargon technique autant que possible. • Structurer le contenu visuel : titres, sous-titres et paragraphes.
Convivialité, interactivité et accessibilité	<ul style="list-style-type: none"> • Eléments interactifs (boutons, liens, formulaires, etc.) positionnés de manière cohérente et facilement repérable. • Indicateurs visuels (changements de couleur, effets d'animation, etc.) pour signaler les interactions et les rétroactions. • Optimiser la navigation en fournissant des liens de retour à la page d'accueil (pas de page morte). • Normes d'accessibilité du Web.
Interface utilisateur, ergonomie et charte graphique	<ul style="list-style-type: none"> • Esthétique et cohérente, charte graphique harmonieuse (couleurs, typographie, icônes, etc.). • Page épurée et aérée, sans éléments superflus ou encombrants. • Des visuels, pour faciliter la reconnaissance rapide des fonctionnalités. • Assurer la réactivité de l'interface en la rendant compatible avec les différents appareils (responsive design).

4.2.Phase de création de la base de données

Pour répondre à la demande du Product Owner, j'ai rédigé les scripts de création de la base de données afin d'obtenir une structure **identique au modèle physique de données fourni dans le cahier des charges**. La base de données a été déployée sous le **système de gestion de base de données (SGBD) MariaDB**.

Dans le but de faciliter le développement lors de la phase de conception et de développement sous le framework Laravel, il était nécessaire que la base de données et son jeu de données puissent être facilement réinitialisés. De plus, le framework Laravel, grâce à sa **bibliothèque d'implémentation backend d'authentification Fortify**, implique plusieurs tables, dont la table "`users`". Cette dernière doit remplacer celle du modèle physique de données et je devais l'adapter pour qu'elle corresponde aux exigences en termes de types et de noms de colonnes.

Afin de répondre à ces contraintes tout en respectant les bonnes pratiques SQL et en assurant une qualité de code optimale, j'ai défini les spécifications techniques concernant la création de la base de données.

Attendus	Spécification de développement de la base de données
Écriture simple des scripts	Décomposition de la création de la base par étapes :
Intégration de la complexité est faite progressivement sur les liaisons entre tables	<ol style="list-style-type: none">1. Création de la base de données2. Migration des tables Fortify3. Adaptation la table <code>USERS</code>4. Création des tables avec typage et clé primaire5. Adaptation ordonnancée des tables aux clefs étrangères6. Intégration du jeu de données
Respecte strict du MPD	Respect des noms de tables, des noms des colonnes, de leurs types, des liaisons entre tables via les clefs étrangères
Réinitialisation complète réalisée aisément	Structuration et ordonnancement des scripts avec l'emploi de <code>IF EXISTS</code> sur l'ensemble des éléments devant être recréés
SGBD de déploiement : MariaDB avec langage SQL	Utilisation du SGBD MariaDB et du langage SQL
Le jeu de données respecte les contraintes entre tables	Rédaction d'un jeu de données avec les différents types d'utilisateurs. Les types de pannes ne sont pas représentatives.
La table des utilisateurs <code>USERS</code> est celle de Fortify	Utilisation de la table <code>USERS</code> de Fortify

4.3.Phase de Conception et développement avec le Framework PHP Laravel

Le Framework implique l'utilisation du pattern MVC (Modèle-Vue-Contrôleur). Chaque fonctionnalité est associée à une route, qui est un lien fonctionnel entre l'utilisateur et l'application, écrit en pseudocode PHP de Laravel. La logique métier est exécutée par un contrôleur écrit en PHP. Si nécessaire, le contrôleur interagit avec un modèle, également écrit en PHP, qui intègre des requêtes SQL via la façade DB pour accéder aux données du SGBD. Une fois que l'algorithme du contrôleur est finalisé, les résultats sont envoyés à la vue correspondante ou à un autre contrôleur. L'interface visuelle est développée en HTML, CSS et JavaScript, en utilisant le moteur de vue Blade et son pseudocode. Le développement suit le principe du mobile-first pour le responsive design.

Ce pattern MVC impose certaines contraintes en termes de codage, notamment l'utilisation du pseudocode PHP et la découpe des fichiers, dans le but d'améliorer la maintenabilité en limitant le champ d'action de chaque élément du code et en utilisant la programmation orientée objet.

En tenant compte des contraintes du pattern, des besoins fonctionnels du client et des bonnes pratiques des différents langages, j'ai établi les spécifications techniques de cette phase regroupées en Épopée¹⁰. Les épopées de 110 à 113 correspondent à des exigences sur les technologies utilisées.

Epics	Attendus CdC	Users stories	Spécification de conception et de développement sous LARAVEL
EP001	La solution est un portail web	US001	La solution est développée avec un langage de programmation web PHP et serveur de données MariaDB
EP002	La solution permet d'ouvrir un ticket pour signaler des incidents techniques	US003	<p>La solution comporte une page spécifique pour de signaler un incident technique avec l'ensemble des critères le spécifiant que l'utilisateur doit remplir :</p> <ul style="list-style-type: none">• Non de l'auteur du ticket de l'incident<ul style="list-style-type: none">○ Élément HTML type <code>INPUT</code>○ Variable type <code>string</code>• Sujet de l'incident<ul style="list-style-type: none">○ Élément HTML type <code>INPUT</code>○ Variable type <code>string</code>• Type de panne de l'incident par une liste déroulante dont les données sont issues de la base de données<ul style="list-style-type: none">○ Élément HTML type <code>SELECT OPTION</code>;

10 « les Epics sont des corpus de tâches qui peuvent être subdivisées en tâches spécifiques (appelées « user stories ») basées sur les besoins/demandes de clients ou d'utilisateurs finaux »
Atlassian

Epics	Attendus CdC	Users stories	Spécification de conception et de développement sous LARAVEL
			<ul style="list-style-type: none"> ○ Variable type <code>int</code>, récupération de l'<code>id</code> du type • Message descriptif de l'incident <ul style="list-style-type: none"> ○ Élément HTML type <code>TEXTAREA</code> ○ Variable type <code>string</code>
EP003	La solution permet de faire la liaison entre l'incident en cours et le service informatique via le ticket	US004	<p>La solution informe le SI des nouveaux incidents via une page spécifique nommé « TOUT LES INCIDENTS » comportant un tableau des incidents listant leurs informations, et triée suivant les plus récents. Il se compose des colonnes suivantes :</p> <ul style="list-style-type: none"> • La référence de l'incident • Le sujet de l'incident • Le type de panne de l'incident • L'avancement de l'incident (nombre de jour d'ouverture de l'incident) • Le statut de l'incident (ouvert, en cours, clôturé)
EP004	La solution permet à l'utilisateur de suivre l'évolution de la résolution de son problème	US005 US109	<p>La solution informe l'utilisateur de ses incidents via une page spécifique nommé « MES INCIDENTS » comportant un tableau des incidents listant leurs informations, et triée suivant les plus récents. Il se compose des colonnes suivantes :</p> <ul style="list-style-type: none"> • La référence de l'incident • Le sujet de l'incident • Le type de panne de l'incident • L'avancement de l'incident (nombre de jour d'ouverture de l'incident) <p>Le statut de l'incident (ouvert, en cours, clôturé)</p>
EP005	La solution crée un espace sécurisé pour accéder à l'espace de gestion des incidents.	US001 US002 US107 US108	<p>La solution comporte une page « CONNEXION » qui permet à l'utilisateur de se connecter en renseignant</p> <ul style="list-style-type: none"> • Son mail <ul style="list-style-type: none"> ○ Élément HTML type <code>INPUT</code> ○ Variable type <code>email</code> <p>La sécurisation est garantie par l'utilisation de la bibliothèque Fortify avec le respect de ses bonnes pratiques définies sur Laravel Fortify - Laravel - The PHP Framework For Web Artisans</p> <ul style="list-style-type: none"> • Son mot de passe.

Epics	Attendus CdC	Users stories	Spécification de conception et de développement sous LARAVEL
			<ul style="list-style-type: none"> ○ Élément HTML type <code>INPUT</code> ○ Variable type string : <p>Complexité du mot-de-passe avec Fortify. Paramétrage du MDP par la fonction <code>passwordRules</code> dans le namespace , <code>app/Actions/Fortify/PasswordValidationRules.php</code> et sa classe :</p> <pre>protected function passwordRules(): array { return ['required', 'string', (new Password()) ->requireUppercase() ->requireNumeric() ->requireSpecialCharacter(), 'confirmed', // <- NO]; }</pre>
EP006	La solution donne la possibilité d'envoyer des précisions ou des demandes d'informations complémentaires sur l'incident en temps réel	US004 US110	<p>La solution comporte une page de chat nommée « INCIDENT » qui permet l'échange de message en temps réel entre l'auteur du ticket de l'incident et le service informatique.</p> <ul style="list-style-type: none"> ○ Ils doivent écrire un message de type texte. <ul style="list-style-type: none"> ○ Élément HTML type <code>TEXTAREA</code> ○ Variable type <code>string</code>
EP107	La solution possède une interface simple et intuitive	US006 US211	<p>La solution respecte le maquetage validé par le Product Owner</p> <p>Le responsive design est respecté par une conception en mobile-first. Emploi de <code>@media</code> queries pour les écrans desktop et mobile</p>
EP108	La solution a comme Langage backend le PHP	US001	La solution est développée avec comme langage backend le PHP

Epics	Attendus CdC	Users stories	Spécification de conception et de développement sous LARAVEL
EP109	La solution est développée avec le Framework LARAVEL	US001 US213	La solution est développée avec le Framework Laravel
EP110	IHM sous : HTML5, CSS3 et JavaScript	US001	La solution comporte des IHM avec le moteur de Vue Blade de Laravel avec les langages HTML5, CSS3, JavaScript et le pseudocode de Blade avec les bonnes pratiques suivantes : Blade Templates - Laravel - The PHP Framework For Web Artisans
EP111	Respecter les bonnes pratiques de sécurité	US213	<p>Le site accessible par le protocole HTTPS</p> <p>La solution est protégée contre les attaques de type :</p> <ul style="list-style-type: none"> Faible XSS : <ul style="list-style-type: none"> Utilisation stricte des <code>{ }</code> et non de la fonction <code>echo()</code> Utilisation stricte de la fonction <code>htmlspecialchars()</code> lors de l'intégration de données type string Utilisation des fonction <code>isset()</code> et <code>empty()</code> lors de la réception de données de type <code>input</code> issues de l'utilisateur Injection SQL : <ul style="list-style-type: none"> Les requêtes SQL lors de <code>SELECT</code> doivent sélectionner uniquement les colonnes utiles à la fonction métier. Aucune requête SQL du type <code>SELECT *</code> Utilisation de la Façade Laravel DatabaseManager : Illuminate\Database\DatabaseManager Laravel API Directory traversal : <ul style="list-style-type: none"> Emploi des chemins relatifs pour les fichiers images Validation et nettoyage des entrées utilisateur par les fonctions Laravel <code>validate</code> du namespace <code>Illuminate\Foundation\Validation</code> à l'entrée des contrôleurs suite à un POST d'un formulaire <p><i>Fortify intègre différentes protections pour mon site web : Cross-Site Request Forgery (CSRF), Cross-Site Scripting (XSS), Brute-Force, Protection des mots de passe par hachage, Sécurisation des sessions.</i></p>

Epics	Attendus CdC	Users stories	Spécification de conception et de développement sous LARAVEL
			<p><i>Règle d'emploi de la bibliothèque Fortify :</i></p> <ul style="list-style-type: none"> Utilisation des features <code>registration()</code>, <code>resetPassword()</code> et <code>updatePassword()</code> dans le fichier <code>config/fortify.php</code> Pour les Routes : <ul style="list-style-type: none"> Strict emploi des routes nommées en GET et POST de la bibliothèque : <ul style="list-style-type: none"> <code>home</code> en GET pour l'accueil d'un utilisateur enregistré <code>login</code> en GET pour l'appel du formulaire de connexion <code>login</code> en POST pour la soumission du formulaire de connexion <code>register</code> en GET pour l'appel du formulaire d'enregistrement <code>register</code> en POST pour la soumission du formulaire d'enregistrement Strict emploi de <code>middleware('guest')</code> sur l'ensemble des Routes pour les visiteurs non connectés Strict emploi de <code>middleware('auth')</code> sur l'ensemble des Routes pour l'utilisateur connecté
EP112	La solution respecte le pattern MVC	US213	<ul style="list-style-type: none"> Le fichier <code>routes/web.php</code> ne contient que le routage des URI via le pseudocode Laravel Les fichiers Contrôleur dans <code>app/http/Controller</code> ne contiennent que du code PHP et ne contiennent pas la logique métier Les fichiers Model dans <code>app/Models</code> ne contiennent que du code en PHP et l'utilisation de langage SQL via la façade PDO liée à la façade DB pour exécuter des requêtes dans SGBD MariaDB. Aucune logique algorithmique n'est admise dans les Models. Les fichiers de Vue dans <code>ressources/views</code> sont découpés en éléments simples (<code>head</code>, <code>nav</code>, <code>footer</code>....) et utilisent le pseudocode Blade et ses directives <code>@include</code>, <code>@yield</code>, <code>@extends</code>, <code>@section</code> Chaque Vue hors celles accessibles par un visiteur non authentifié sont construite suivant une template : <code>ressources/views /template.blade.php</code>. <ul style="list-style-type: none"> Celles accessibles par les visiteurs sont des sigles pages sans utilisation des directives sauf pour le footer.
EP113	La solution est écrite en respectant les standards rédactionnels		<p>HTML et CSS : Vue d'ensemble des standards d'accessibilité du W3C Web Accessibility Initiative (WAI) W3C</p> <p>SQL : [SQL] Bonnes pratiques pour le nommage des tables et colonnes</p> <p>PHP et Laravel : PHP Standards Recommendations - PHP-FIG</p>

4.4.Phase de déploiement

Le développement est réalisé directement sur le serveur afin d'éviter les problématiques de compatibilité liées au transfert d'une application Laravel développée en local. Cette phase ne se déroule pas en dehors de la connexion au SGBD.

Attendus	Spécification de conception et de développement sous LARAVEL
Protocole réseau : HTTP2.0	RAS
Déploiements sur Lab004	Connection à la base de données via le fichier <code>.env</code> <ul style="list-style-type: none">Paramétrage :<pre>DB_CONNECTION=mysql DB_HOST=lab004.2isa.org DB_PORT=33004 DB_DATABASE=GESTION_TICKETS DB_USERNAME=root DB_PASSWORD=3265lab004</pre>

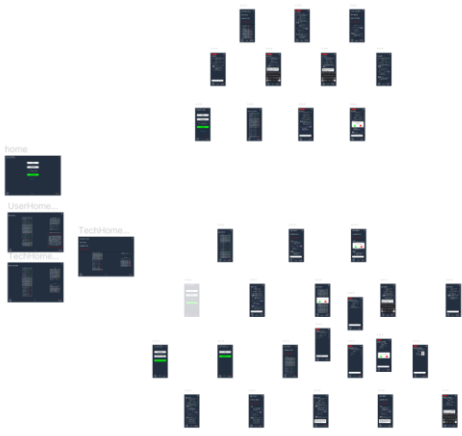
5. Réalisations du candidat comportant les extraits de code les plus significatifs, et en les argumentant, y compris pour la sécurité

5.1.Phase de maquettage

Pour le maquettage de l'application, j'ai utilisé l'outil de design collaboratif, Figma. Cette application web m'a permis de créer des éléments simples que je pouvais combiner pour obtenir une interface utilisateur visuellement manipulable. Grâce à Figma, j'ai pu réaliser deux scénarios d'utilisation de l'application :

- Maquettage pour un utilisateur type personnel d'AMIO : [figma.user](#)
- Maquettage pour un utilisateur type service informatique (seul les spécificités des SI sont représentés) : [figma.si](#)

Validé par le PO le 2 mai 2023.

Ensemble des interfaces réalisées	
-----------------------------------	--------------------------------------------------------------------------------------

Ensemble des éléments simples réalisés



5.1.1. Exemple d'IHM du maquettage

Connexion

Accueil après
connexion d'un
TecHotline

Un incident

Clôturer un
incident

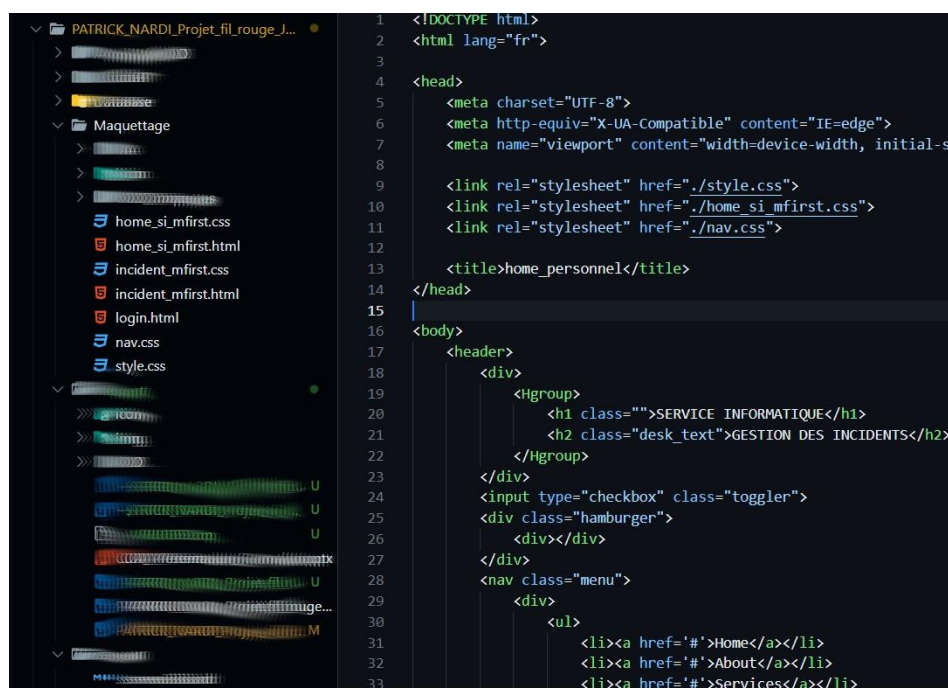
Menu



5.1.2. Charte graphique et model de Vue

Pour définir ma charte graphique, je me suis inspiré du site de l'association [AMIO \(amio-millau.fr\)](http://amio-millau.fr), en récupérant les codes couleurs utilisés sur leur site internet. Ensuite, j'ai effectué des recherches sur les CSS pour les boîtes de dialogue et les tableaux, en utilisant l'éditeur de code en ligne codepen.io, afin de trouver des styles correspondants à mon design. J'ai également étudié les bonnes pratiques en matière de design de sites web et de codage en CSS comme celles définies sur le site alticreation.com. Ils expliquent aussi la méthodologie BEM pour structurer son code CSS, je me suis inspiré sans la pratiquer intégralement.

À partir de là, j'ai créé chaque page en utilisant le HTML avec une attention particulière portée sur sa sémantique et CSS, anticipant ainsi leur intégration ultérieure dans le Framework Laravel avec le moteur de vue Blade.



5.2.Phase de création de la base de données

La réalisation de la base de données en suivant le modèle physique de données fourni par le Product Owner dans le cahier des charges m'a permis de créer une première base en respectant les contraintes des clés primaires et étrangères, en utilisant un jeu de données correspondant. Par la suite, avec l'intégration de la bibliothèque Fortify de Laravel pour la gestion des identifications, j'ai dû adapter mon script aux tables induites par cette bibliothèque. Cela impliquait de remplacer ma table d'utilisateurs par celle de Fortify et de réécrire mon script en conséquence.

Pour installer ou réinitialiser la base de données, vous devez suivre les étapes suivantes dans le SGBD MariaDB (en utilisant le logiciel DBeaver, par exemple) :

1. Sous DBeaver : script001_create_DataBase pour créer la base de données,
2. Sous VSCode : installer le package `laravel/fortify` et migration le sur la base de données avant toutes créations des autres tables
3. Sous VSCode : dans le dossier de site web sur le serveur faire : `php artisan migrate,`
4. Sous DBeaver : script002 ALTER_TABLE_USER.sql pour adapter la table users de Fortify au besoin de l'application,

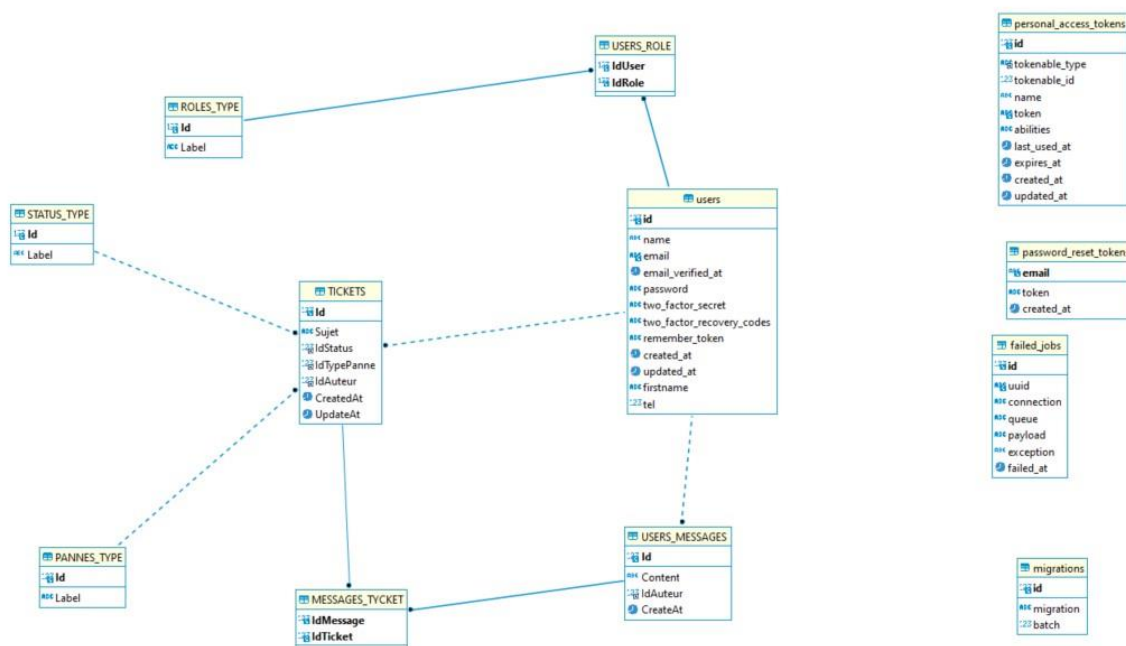
5. Sous DBeaver : script003_CREATE_TABLES.sql pour créer les tables et leurs liens via les clefs primaires et étrangères,
6. Sous DBeaver : fichier cvs de 01 à 08 pour intégrer le jeu de données dans la base.

Fichier pour l'installation ou la réinitialisation de la base de données. Voir dossier annexe

```

Part01 > PATRICK_NARDI_Projet_fil_rouge_Jalon1_DossierProjet > DataBase > dataSet > script003
70  CREATE table ROLES_TYPE (
71  -- Id integer(5) NOT NULL DEFAULT nextval(seq_ROLE_TYPE_id),
72  Id integer(5) not null,
73  Label varchar(50) not null CHECK (Label <> ''),
74  constraint PK_ROLE_TYPE primary key (Id)
75  );
76
77  CREATE table USERS_MESSAGES (
78  -- Id integer NOT NULL DEFAULT nextval(seq_USER_MESSAGE_id),
79  Id integer(5) not null,
80  Content text not null CHECK (Content <> ''),
81  IdAuteur bigint(20) unsigned NOT NULL,
82  CreateAt Datetime default sysdate() auto_increment NULL,
83  constraint PK_USER_MESSAGE primary key (Id)
84  );
85
86  CREATE table MESSAGES_TYCKET (
87  IdMessage integer(5) not null,
88  IdTicket integer(5) not null,
89  constraint PK_MESSAGE_TYCKET primary key (IdMessage, IdTicket)
90  );
91
92  CREATE table TICKETS (
93  -- Id integer NOT NULL DEFAULT nextval(seq_TICKET_id),
94  Id integer(5) not null,
95  Sujet varchar(100) not null CHECK (Sujet <> ''),
96  IdStatus integer(5) not null,
97  IdTypePanne integer(5) not null,
98  IdAuteur bigint(20) unsigned NOT NULL,
99  CreatedAt Datetime default sysdate(),
100  UpdateAt Datetime default sysdate(),
101  constraint PK_TICKET primary key (Id)
102  );
  
```

Modèle physique des données pour l'application réalisée sur le SGBD MariaDB



5.3.Phase de Conception et développement avec le Framework PHP Laravel

J'ai choisi d'écrire le code backend en suivant les fonctionnalités exigées, en respectant les demandes du client et les spécificités du chapitre 4. J'ai utilisé une méthode proche du Plan-Do-Check-Act (PDCA), un cycle d'amélioration continu pour planifier la réalisation de cette phase. Cela en fonction des attendus fonctionnels et des spécifications techniques. J'ai effectué une veille technologique sur les bonnes pratiques et la sécurité, puis intégré les améliorations nécessaires. Étape par étape après la fin du premier cycle qui pour une fonctionnalité représenté sa juste intégration, je me suis attaché à l'améliorer, une amélioration après l'autre.

Je ne détaillerai pas l'ensemble de mes réalisations sur ce projet, je me concentrerai sur deux aspects en particulier : la fonction de connexion (US001) détaillé ci-dessous et la création d'un nouvel incident (US03).

Vous trouverez en [Anx2](#) une représentation schématique de mon intégration du pattern MVC pour la fonction de connexion avec Fortify.

Il est important de noter que comme toutes les autres Épopées, celles de 110 à 112, doivent être prises en compte dans la réalisation de chacune des fonctionnalités.

5.3.1. Conception et développement de l'espace sécurisé

L'Épopée EP005, intitulée "La solution crée un espace sécurisé pour accéder à l'espace de gestion des incidents", regroupe plusieurs User Stories. Le Product Owner demande de permettre, grâce à la bibliothèque Fortify, la création d'un nouvel utilisateur de type usager, ainsi que la possibilité pour l'utilisateur de modifier son mot de passe.

Ainsi, l'Épopée EP005 doit permettre les fonctionnalités suivantes : la connexion, la déconnexion, l'enregistrement et la modification du mot de passe.

L'utilisation de Fortify implique l'utilisation des classes de cette bibliothèque dans les espaces de noms (namespaces) App\Actions\Fortify et Vendor\laravel\fortify. J'ai dû modifier la classe CreateNewUser en ajoutant l'attribution du rôle lors de la création d'un nouvel utilisateur. La gestion des erreurs est prise en charge par la bibliothèque elle-même.

- Remarque sur la modification du mot de passe :

Cette fonctionnalité nécessite un serveur SMTP pour fonctionner. J'ai utilisé [ethereal.email](#), un service SMTP factice. Il s'agit d'un service de courriels anti-transactionnels entièrement gratuit, où les messages ne sont jamais réellement envoyés. Il permet de simuler la réception d'e-mails de demande de modification du mot de passe envoyés par Fortify via la vue `auth.forgot-password`.

Paramétrage du serveur SMTP dans le fichier `.env`

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.ethereal.email
MAIL_PORT=587
MAIL_USERNAME=cristina.lakin32@ethereal.email
MAIL_PASSWORD=AaQZwa3bDnDduhF4U1
MAIL_ENCRYPTION=STARTTLS
MAIL_FROM_ADDRESS="no-reply@monsite.com"
MAIL_FROM_NAME="{APP_NAME}"
```


Graphise	Code	Remarques
	<pre> <!DOCTYPE html> <html lang="fr"> <head> <meta charset="UTF-8"> <meta http-equiv="X-UA-Compatible" content="IE=edge"> <meta name="viewport" content="width=device-width, initial-scale=1.1"> <link rel="stylesheet" href="{{ Vite::asset('resources/css/style.css') }}"> <title>Elements du site</title> </head> <body> <header> <nav class="nav"> </nav> </header> <div id="flex_part"> <main> <hgroup> <h1 class="text_center">SERVICE INFORMATIQUE</h1> <h2 class="text_center">GESTION DES INCIDENTS</h2> </hgroup> <article> <form method="POST" action="{{ route('login') }}"> @csrf <fieldset> <legend>Connectez-vous</legend> <label for="email" class="hidden"></label> <input class="input bt_value" type="email" id="email" name="email" value="" placeholder="Entrer votre email" required> <label for="password" class="hidden">Entrer votre mot-de-passe</label> <input class="input bt_value" type="password" id="password" name="password" placeholder="Entrer votre mot-de-passe" required> <button type="submit" class="input bt_ok pointer" name="button">Se connecter</button> Mot-de-passe oublier ? S'enregistrer </fieldset> </form> </article> </main> <footer> @include('templates.footer') </footer> </div> </body> </html> </pre>	<p>Act001 : connexion étape 1</p> <p>Route de Connexion à l'application</p> <p>En POST activé par le submit du form. Vers la route 'login' du fichier : vendor/laravel/fortify/routes/routes.php</p> <p>Act002 : Route de mise-à-jour du mot de passe</p> <p>Renvoie vers l'URI : /forgot-password</p> <p>Act003 : Route de mise-à-jour du mot de passe</p> <p>Renvoie vers la route en GET 'register' du fichier : vendor/laravel/forti</p>

Etapas de connexion	Commentaires	Code
<p>Etape 2. Routage</p>	<p>La route renvoie vers le Controller de Fortify <code>AuthenticatedSessionController</code> et sa classe <code>create</code>. est l'interface d'entrée de la boîte noire Fortify.</p> <p>Son interface de sortie est l'URI <code>'home'</code> qui doit être codé avec le pseudocode Laravel dans le fichier <code>routes/web.php</code>.</p> <p>Cette route en <code>GET</code> exécute la méthode <code>choiseHomePageToRoleAndSetSession</code> de la classe du contrôleur <code>app/Http/Controllers/MyUserController.php</code> avec l'identifiant de l'utilisateur provenant de Fortify, <code>Auth::id()</code></p>	<pre>Route::get('home', function(Request \$request){ return MyUserController::choiseHomePageToRoleAndSetSession(Auth::id()); })->middleware('auth')->name('choisehome');</pre>
<p>Etape 3. Redirection.</p>	<p>Cette méthode me permet router l'utilisateur en fonction de son rôle.</p> <ul style="list-style-type: none"> • Soit vers de la listes de tous les tickets , c'est un TecHotline par la redirection vers la route nommée <code>'tickets'</code> • Soit vers la liste de tous les tickets d'un usager spécifique via son identifiant par la redirection vers la route nommée <code>'mytickets'</code> 	<pre>public static function choiseHomePageToRoleAndSetSession(int \$idUser){ session(['idUser' => \$idUser]); if(self::UserisTecHoline(\$idUser)){ session(['IsTecHotline' => true]); return redirect()->route('tickets'); }else{ session(['IsTecHotline' => false]); return redirect()->route('mytickets',['iduser' => \$idUser]); } }</pre>

Etapas de connexion	Commentaires	Code
<p>Etape 4 pour les usagers.</p> <p>Routage</p>	<p>Cette route paramétrée renvoie vers la méthode de classe <code>getMyTickets</code> du contrôleur <code>app/Http/Controllers/TicketController.php</code> avec comme paramètre l'identifiant de l'utilisateur, <code>'idUser'</code> par affectation de l'élément du tableau associatif <code>sessions</code> de Laravel :</p> <pre>session()->get('idUser')</pre>	<pre>Route::get('/mesincidents/{idUser}', [TicketController::class, 'getMyTickets', ['idUser' => session()->get('idUser')]])->where('idUser', '^[\d]{5}')->middleware('auth')->name('mytickets');</pre>
<p>Etape 5 pour les usagers.</p> <p>Algorithmie métier</p>	<p>Cette méthode réalise la logique métier pour obtenir la liste des tickets de l'utilisateur. Cela en instanciant le Model <code>app/Models/Ticket.php</code> pour exécuter sa méthode de classe <code>getMyTickets</code> afin de récupérer un tableau associatif renvoyé à la Vue <code>'tickets'</code></p>	<pre>public function getMyTickets() { Controller::forgetItemsSession(); \$data=[]; if(!empty(session()->get('idUser'))) { \$db = new Ticket(); \$data = \$db->getMyTickets(session()->get('idUser')); if (empty(\$data)) { session(['noticket' => "Aucun incident"]); } }else{ session(['errordb' => "Vous ne devriez pas être ici"]); } return view('tickets', ['data' => \$data]); }</pre>

Etapas de connexion	Commentaires	Code
<p>Etape 5bis pour les usagers.</p> <p>Interrogation du SGBD</p>	<p>Cette méthode paramétrée a pour but d'interroger la base de données en exécutant la requête SQL via la façade PDO et de retourner les données brutes récupérées en fonction de l'identifiant utilisateur, <code>\$id_user</code>.</p>	<pre> public function getMyTickets(int \$id_user) { return DB::select("SELECT TICKETS.Id AS 'id_ticket', TICKETS.Sujet AS 'sujet', STATUS_TYPE.Id AS 'id_status', STATUS_TYPE.Label AS 'status_label', PANNES_TYPE.Id AS 'id_type_de_panne', PANNES_TYPE.Label AS 'type_de_panne', users.id AS 'id_user', CONCAT(users.name, ' ', LEFT(users.firstname, 1), '.') AS 'nom', DATEDIFF(TICKETS.UpdateAt, TICKETS.CreatedAt) AS 'avancement', TICKETS.CreatedAt AS 'date_de_creation', TICKETS.UpdateAt AS 'date_de_maj', (SELECT COUNT(*) FROM MESSAGES_TYCKET WHERE MESSAGES_TYCKET.IdTicket = TICKETS.Id) AS 'nb_de_message' FROM TICKETS JOIN PANNES_TYPE ON TICKETS.IdTypePanne = PANNES_TYPE.Id JOIN users ON TICKETS.IdAuteur = users.id JOIN STATUS_TYPE ON TICKETS.IdStatus = STATUS_TYPE.Id WHERE users.id = ? ORDER BY 'date_de_creation' ASC", [\$id_user]); } </pre>

Etapas de connexion	Commentaires	Code
étape 6. Restitution de la vue	La Vue resources/views/tickets.blade.php est la même pour les usagers et les TecHotline. Et en fonction adapte le tableau sur ses colonnes.	Code, voir Anx1
	Rendu de la Vue pour les TecHotline authentifié	Page d'accueil d'un usager authentifié

Etapes de connexion	Commentaires	Code																																																	
	<div><div>SERVICE INFORMATIQUE</div><div>TOUS LES INCIDENTS</div><table><thead><tr><th>n°</th><th>Auteur</th><th>Avcmt</th><th>msg</th></tr></thead><tbody><tr><td>99106</td><td>Vincent A.</td><td>34j </td><td>4</td></tr><tr><td>99206</td><td>Rice C.</td><td>34j </td><td>8</td></tr><tr><td>99306</td><td>Kirby D.</td><td>1j </td><td>1</td></tr><tr><td>99402</td><td>Kirby D.</td><td>12j </td><td>3</td></tr><tr><td>99406</td><td>Marc D.</td><td>0j </td><td>1</td></tr><tr><td>99407</td><td>Patrick B.</td><td>0j </td><td>1</td></tr></tbody></table></div>	n°	Auteur	Avcmt	msg	99106	Vincent A.	34j 	4	99206	Rice C.	34j 	8	99306	Kirby D.	1j 	1	99402	Kirby D.	12j 	3	99406	Marc D.	0j 	1	99407	Patrick B.	0j 	1	<div><div>SERVICE INFORMATIQUE</div><div>TOUS MES INCIDENTS</div><table><thead><tr><th>n°</th><th>Avcmt</th><th>Date de création</th></tr></thead><tbody><tr><td>99106</td><td>34j </td><td>2023-05-03 11:56:30</td></tr><tr><td>99102</td><td>0j </td><td>2023-02-03 11:56:30</td></tr><tr><td>99103</td><td>83j </td><td>2023-03-15 11:56:30</td></tr><tr><td>99104</td><td>2j </td><td>2023-04-15 11:56:30</td></tr><tr><td>99105</td><td>0j </td><td>2023-05-03 11:56:30</td></tr><tr><td>99101</td><td>142j </td><td>2023-01-03 11:56:30</td></tr></tbody></table></div>	n°	Avcmt	Date de création	99106	34j 	2023-05-03 11:56:30	99102	0j 	2023-02-03 11:56:30	99103	83j 	2023-03-15 11:56:30	99104	2j 	2023-04-15 11:56:30	99105	0j 	2023-05-03 11:56:30	99101	142j 	2023-01-03 11:56:30
n°	Auteur	Avcmt	msg																																																
99106	Vincent A.	34j 	4																																																
99206	Rice C.	34j 	8																																																
99306	Kirby D.	1j 	1																																																
99402	Kirby D.	12j 	3																																																
99406	Marc D.	0j 	1																																																
99407	Patrick B.	0j 	1																																																
n°	Avcmt	Date de création																																																	
99106	34j 	2023-05-03 11:56:30																																																	
99102	0j 	2023-02-03 11:56:30																																																	
99103	83j 	2023-03-15 11:56:30																																																	
99104	2j 	2023-04-15 11:56:30																																																	
99105	0j 	2023-05-03 11:56:30																																																	
99101	142j 	2023-01-03 11:56:30																																																	

6. Plan de tests et de déploiement

J'ai effectué une vérification pas à pas de mon code en échangeant régulièrement avec les Product Owners, en effectuant un débogage continu à l'aide des fonctions `dd()`, `var_dump()` et `print_r()`, tout en intégrant les exigences spécifiées. Les checklists ci-dessous représentent ma vision simpliste de test d'intégration et de test navigateur.

Afin de réaliser des tests visant à valider chaque fonctionnalité dans différents scénari, il aurait été nécessaire pour moi d'acquérir de nouvelles connaissances. J'aurais pu acquérir ces connaissances en échangeant avec les formateurs, en consultant le chapitre de la documentation [Laravel](#), en suivant des tutoriels tels que celui de [Grafikart](#), ou encore en lisant des blogs de développeurs comme celui de [Jason McCreary](#). Cependant, étant donné que l'élaboration des plans de test ne fait pas partie des compétences à valider, j'ai décidé de ne pas l'inclure dans ma démarche de travail à ce stade.

6.1.Checklist des fonctionnalités demandées

Items à vérifier	Interprétation validée avec le Product Owner	Remarques	Vérifications	Remarques
Se dédie aux usagers de la structure		Le SI doit pouvoir y accéder aussi		
Est un portail web				
Permet d'ouvrir un ticket pour signaler des incidents techniques				
Permet de faire la liaison entre l'incident en cours et le service informatique via le ticket d'un incident				
Permet à l'utilisateur de suivre l'évolution de la résolution de son problème				
Crée un espace sécurisé pour chaque incident		Connexion au site = espace sécurisé		
Cet espace donne la complémentaire sur l'incident en temps réel				
A une interface simple et intuitive				
Permet une expérience optimale				

6.2.Checklist des contraintes techniques

Items à vérifier	Vérifications	Remarques
Code de qualité		
Langage backend : PHP		
Framework : LARAVEL		

IHM sous : HTML5, CSS3 et JavaScript.			
Protocole réseau : HTTP2.0			
Déploiements sur Lab004			
Fonctionnel sur Firefox	Desktop		
	Mobil		
Fonctionnel sur Edge	Desktop		
	Mobil		
Fonctionnel sous Chrome	Desktop		
	Mobil		
Fonctionnel sous Safari	Desktop		
	Mobil		

7. Veille effectuée sur les vulnérabilités de sécurité

Durant ce projet Laravel, j'ai accordé une grande importance à la veille sécuritaire. Le framework Laravel offre des fonctionnalités robustes en matière de sécurité, telles que l'utilisation de façades pour les requêtes SQL, le langage pseudo-PHP dans les vues avec le moteur Blade, la gestion des sessions et les bibliothèques telles que Fortify. Cependant, il restait des failles potentielles liées à mon code que je devais combler.

Pour assurer la sécurité de l'application, j'ai effectué des recherches sur des sites tels que laravel-france.com, parasoft.com et ChatGPT. J'ai identifié les failles courantes à éviter avec des consultations des recommandations de [bonnes pratiques de l'ANSI](#), en particulier celles liées à la [sécurité d'un site web](#) avec cela le [Top 10 de owasp](#) qui définit les 10 vulnérabilités les plus courantes des application web.

Cependant, étant donné que je ne possède pas toutes les connaissances nécessaires pour une compréhension approfondie de ces recommandations, il est essentiel qu'elles fassent partie intégrante de la prochaine évolution du projet, en tant que première exigence, afin de garantir la sécurité de l'application. Le niveau d'exigence de sécurité pour une version ultérieure, devra être discuté avec le Product Owner.

8. Description d'une situation de travail ayant nécessité une recherche

En début de projet, j'ai effectué des recherches sur codepen.io afin de trouver des éléments en HTML et CSS qui correspondaient aux demandes du projet, tels que des [tableau](#), le [menu](#) ou le « [chat](#). » J'y ai découvert des solutions facilement adaptables à mon projet, ce qui m'a permis d'obtenir une interface agréable et intuitive pour les utilisateurs. Ces inspirations ont été précieuses pour la conception de l'interface du projet.

9. Conclusion

Durant ma formation en CDA, j'ai entrepris ce projet dans le but de simplifier la gestion des incidents du parc informatique de l'association AMIO. Dans cette première étape, j'ai développé une application web qui permet aux utilisateurs authentifiés de signaler leurs incidents de manière rapide et facile. De leur côté, les membres du service informatique disposent d'un outil efficace pour gérer et résoudre ces problèmes rapidement et efficacement au travers d'un portail web de communication.

Pour répondre aux besoins d'AMIO, qui souhaitait disposer d'un espace sécurisé pour faciliter la gestion et la traçabilité des incidents de son parc, j'ai conçu et développé plusieurs fonctionnalités clés. J'ai utilisé la bibliothèque Fortify pour mettre en place un système d'authentification sécurisé. J'ai également mis en place un formulaire permettant aux utilisateurs de signaler un nouvel incident, ainsi qu'un « chat » pour faciliter la communication entre l'utilisateur et le service informatique sur un ticket spécifique. Toutes ces fonctionnalités ont été conçues en tenant compte des besoins exprimés par le client, en privilégiant des interfaces simples et intuitives. Après avoir effectué plusieurs tests d'utilisation en collaboration avec le Product Owner, les fonctionnalités et leurs interfaces ont été validées. L'application que j'ai développée répond de manière efficace aux exigences du client.

Le choix du Framework PHP Laravel pour cette solution garantit un développement sécurisé en suivant les meilleures pratiques définies par l'OWASP¹¹. Malgré le grand nombre de fichiers qu'il peut comporter, ce Framework reste accessible grâce à une documentation claire et une communauté très active. L'implémentation du pattern MVC dans Laravel permet d'obtenir un code lisible et une séparation claire des fonctionnalités métier. Grâce à son architecture et à sa gestion efficace des appels aux bases de données, Laravel offre une expérience agréable lors de sa pratique.

Mon processus de développement m'a permis de mettre en œuvre progressivement les nouvelles connaissances acquises lors de ma formation, ainsi que celles obtenues grâce à ma veille sur les solutions existantes répondant aux besoins. J'ai rencontré plusieurs défis, tels que la compréhension de la boîte noire Fortify et la gestion des erreurs entre les contrôleurs et les formulaires avec le moteur de vues Blade. Ces défis ont contribué à mon évolution dans l'application des bonnes pratiques de développement en PHP, HTML, CSS, JavaScript et SQL. Ma première expérience avec Laravel se conclut par la création d'une application conviviale et efficace.

Cette application offre de nombreuses possibilités d'amélioration du côté de l'expérience utilisateur, comme la mise en place d'un indicateur de nouveaux messages pour se rapprocher d'une expérience de vrai chat, d'un compteur de nouveaux tickets, de filtre sur les colonnes des tableaux, une page des nouveaux incidents pour le service informatique ou encore d'un mode jour-nuit. De plus, l'amélioration de la gestion des media queries permettrait une meilleure adaptation aux différents types d'écrans. Du côté backend, la transformation des données provenant des requêtes SQL en instances de classe serait un avantage si de nouvelles fonctionnalités devaient être intégrées dans une future API¹².

J'ai développé une solution conviviale pour automatiser la gestion du parc informatique de l'association AMIO. Cette solution se distingue par sa réactivité, que ce soit pour la création de nouveaux tickets ou pour les échanges de messages. Son développement sous le Framework PHP Laravel, en suivant le modèle MVC, lui confère une modularité, une extensibilité et une sécurité accrues. L'utilisation de la bibliothèque Fortify pour l'authentification garantit une

11 [Open Web Application Security Project](#), liste des meilleures pratiques en matière de sécurité des API

12 Interface de programmation applicative

protection solide et fiable du site. Grâce à cette application web épurée, l'association bénéficiera d'une amélioration significative dans la résolution des incidents informatiques.

Ce projet, réalisé dans le cadre de ma formation de Concepteur Développeur d'applications, a été une expérience enrichissante, me permettant de plonger de manière agréable dans le développement de solutions web. Sa pertinence et ses objectifs clairs m'ont offert une direction bien définie, tandis que les échanges avec les formateurs ont guidé mes réflexions. Grâce à cette expérience, j'ai renforcé ma conviction quant à mon choix de reconversion professionnelle.

10. Annexes

10.1. Annexe 1, code du fichier, resources/views/tickets.blade.php :

```
@extends('template')
@vite('resources/js/liste_incidents.js')
@section('title')
    Service informatique AMIO - Gestion des incidents
@endsection

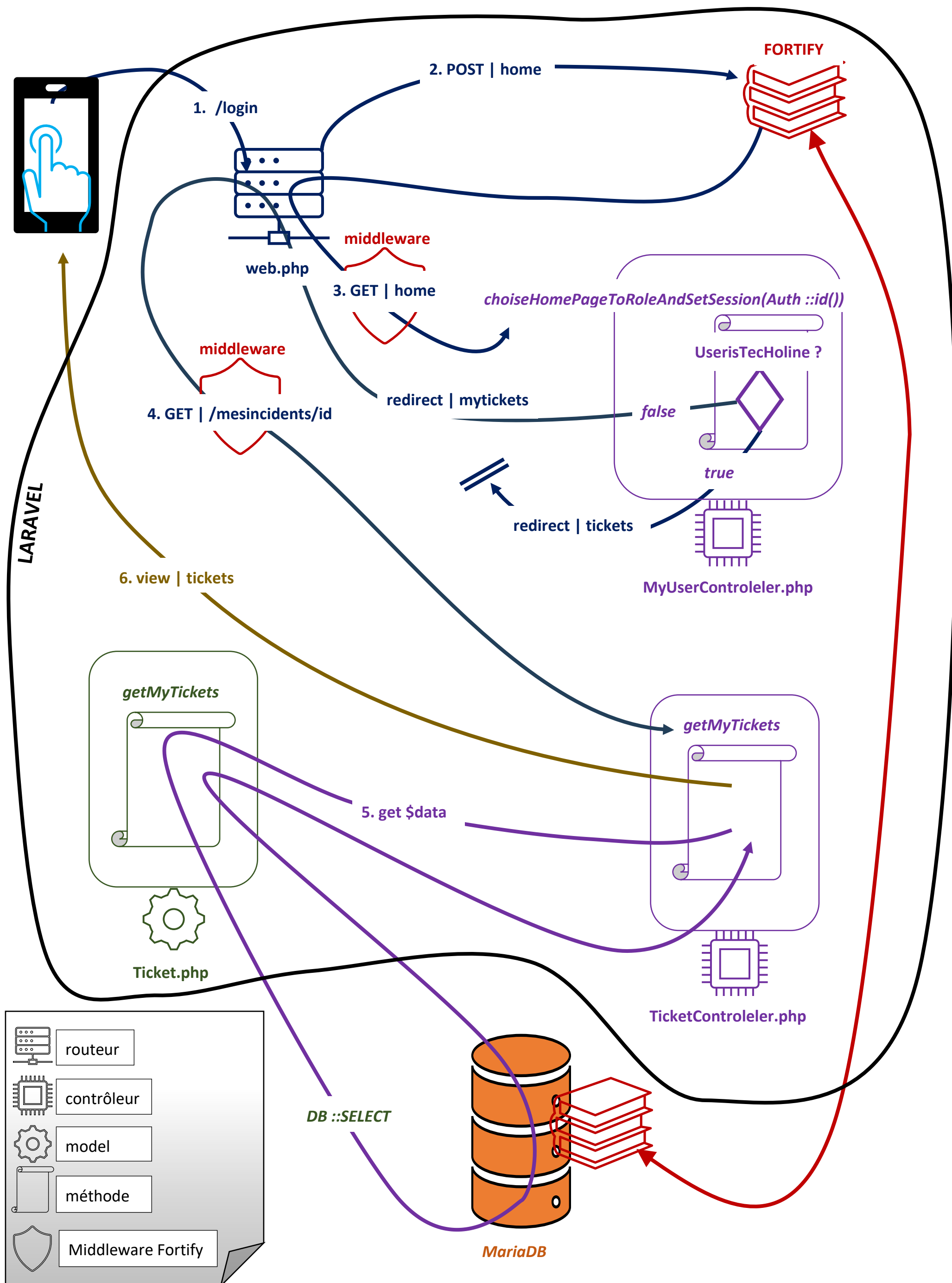
@section('contenu')
    <article class="table_fix_head">
        <div class="tableau">
            <table class="container">
                @if (!empty($IsTecHotline) && $IsTecHotline)
                    <caption class="fixed">TOUS LES INCIDENTS</caption>
                @else
                    <caption class="fixed">TOUS MES INCIDENTS</caption>
                @endif
                <thead class="fixed">
                    <th>
                        <span class="mobil_text">n°</span>
                        <span class="desk_text">Référence</span>
                    </th>
                    <th>Sujet</th>
                    <th>
                        <span class="mobil_text">Panne</span>
                        <span class="desk_text">Type de panne</span>
                    </th>
                    @if (!empty($IsTecHotline) && $IsTecHotline)
                        <th>Auteur</th>
                    @endif
                    <th>
                        <span class="mobil_text">Avcmt</span>
                        <span class="desk_text">Avancement</span>
                    </th>
                    <th>Date de création</th>
                    <th>Date de mise-à-jour</th>
                    <th>
                        <span class="mobil_text">msg</span>
                        <span class="desk_text">nombre de message</span>
                    </th>
                </thead>
                <tbody>
                    @if (!session()->get('noticket'))
                        @foreach ($data as $RowData)
                            <tr onclick="goToIncident('{{ route('ticket', ['nb' =>
$RowData->id_ticket]) }}')">
                                <td><a
```

```

                                href="{{ route('ticket', ['nb' => $RowData->id_ticket]) }}">{{ $RowData->id_ticket }}
                            </td>
                            <td>{{ $RowData->sujet }}

```

10.2. Annexe 2, Schéma du pattern MVC pour fonctionnalité de connexion



10.3. Annexe 3, Analyse préliminaire pour les User-Stories :

(Personnel AMIO)			
	Personnel	SI	
Log in	X	X	Product Owner arbitrage : 17/04/23 Il est choisi d'attribuer les rôles par affectation suivant une liste : <ul style="list-style-type: none"> • Usager • Service informatique Contrainte technique de l'entreprise 21/04/23 Le personnel du SI peut travailler sur tous les tickets Il n'y a pas d'attribution de ticket
Log out	X	X	
Ouvrir un ticket	X*		
Clôturer un ticket		X*	
Écrire un message sur un ticket		X*	
Écrire un message sur un de mes tickets	X* **		
Définir le type de panne	X* **	X*	
Attribuer un matériel à un ticket	X* **	X*	
Afficher les tickets		X*	
Afficher ses Tickets	X*	X*	
Afficher un ticket existant	X* **	X*	

* Être Logué

** Être Le propriétaire du