



Rapport de stage

BTS SIO 2ème année SLAM

RATP INFRASTRUCTURES

RATP - Site Val Bienvenue, 11 Av. Louison Bobet, 94120 Fontenay-sous-Bois

Tuteur entreprise : Joel GALLEGO

Tuteur académique : M.MARECHAL

Période du stage : 01/12/2025 au 23/01/2026

SOMMAIRE

Page de couverture	1
Sommaire	2
Remerciements	3
Introduction	4
Présentation de l'entreprise	5
Activités réalisées	7
Compétences	19
Problèmes rencontrés	20
Conclusion	21
Annexes	22

REMERCIEMENTS

Avant tout mon développement sur cette expérience professionnelle, j'aimerais commencer ce rapport de stage par des remerciements, à ceux qui m'ont beaucoup appris au cours de ce stage.

Je remercie mon tuteur de stage Mr.Gallego qui m'a formé et accompagné tout au long de ce stage avec beaucoup de patience et de pédagogie. Enfin, je remercie l'ensemble de l'équipe U2D et l'unité ISI pour les conseils qu'ils ont pu me prodiguer au cours de ce stage.

INTRODUCTION

Dans le cadre du Brevet de Technicien Supérieur Services Informatiques aux Organisations (BTS SIO), option Solutions Logicielles et Applications Métiers (SLAM), j'ai effectué un stage de deuxième année au sein de la RATP INFRASTRUCTURES .

Ce stage s'est déroulé du 1er Décembre au 23 Janvier 2026, au sein de l'unité ISI (industrialisation des systèmes d'information) et dans l'entité U2D (Urbanisation Développement et Data) . J'ai intégré l'équipe en charge du support technique et de développements de solutions logicielle, où j'ai pu m'impliquer dans des missions liées au développement d'application.

Tout au long de cette période, j'ai eu l'opportunité de participer à des tâches variées, qui m'ont permis de renforcer mes compétences techniques, de découvrir les réalités du travail en environnement professionnel, et de mieux comprendre les différents métiers liés à l'informatique.

Ce rapport propose une synthèse des connaissances acquises au cours de ce stage. Il présentera dans un premier temps l'environnement de travail et les missions réalisées, avant de développer une étude de cas détaillant un projet significatif auquel j'ai contribué.

Enfin, une conclusion viendra clore ce rapport en dressant un bilan global de cette expérience en milieu professionnel.

PRÉSENTATION DE L'ENTREPRISE



La **RATP INFRASTRUCTURES** est une filiale du groupe **RATP** qui assure la gestion d'infrastructures du réseau historique en Île-de-France. Elle garantit aussi la gestion technique, la surveillance, l'entretien et la réparation nécessaires au fonctionnement et à la sécurité du réseau.

ORGANIGRAMME

L'organisation de RATP Infra repose sur une structure en cascade, segmentée en pôles d'expertise et unités spécialisées. Mon stage s'est déroulé au sein du pôle DI3D (Direction de l'Innovation, du Digital et du Développement Durable).

Au sein de cette direction, j'ai rejoint l'unité ISI (Industrialisation des Systèmes d'Information), chargée de la conception et du déploiement des outils technologiques. Plus précisément, j'ai été intégré à l'entité U2D (Urbanisation, Développement et Data). Cette entité se distingue par des compétences variées : des chefs de projets digitaux qui assurent le pilotage, des administrateurs d'infrastructure SI garantissant la robustesse des systèmes, ainsi que des équipes de développeurs et de MOE (Maîtrise d'Œuvre) locale. Cette pluridisciplinarité permet à l'U2D de couvrir l'intégralité du cycle de vie des projets, de l'urbanisation des données jusqu'au développement technique de proximité. (Organigramme voir Annexes)

ACTIVITÉS RÉALISÉES

Application : GAP (Gestion de l'Activité et du Pointage)

Le contexte métier : L'application GAP (Gestion de l'Activité et du Pointage) est un outil dédié à l'entité GIGP (Gestionnaire d'Infrastructure Grand Paris). Cette structure de la RATP joue un rôle important dans la modernisation des transports franciliens.

L'enjeu du Jour-Homme (J.H) : Le cœur de l'application repose sur la saisie et le suivi des Jours-Hommes (J.H). Dans le cadre de projets d'envergure comme le Grand Paris, le J.H est l'unité de mesure permettant de quantifier l'effort de travail et de piloter les budgets alloués aux infrastructures. Une erreur de saisie ou un bug dans le calcul de ces unités peut entraîner des écarts financiers significatifs dans la comptabilité analytique de la RATP.

Objectif de ma mission : Mon rôle a été d'intervenir sur l'application en cours. Ma mission consistait à valider les nouvelles fonctionnalités et à assurer la non-régression du système avant sa validation définitive. (voir Annexes).

Infrastructure et Sécurité

Environnement de virtualisation et cloud privé : L'application GAP est hébergée sur une infrastructure composée de VM (Machines Virtuelles). Ces serveurs ne sont pas présents physiquement dans les bureaux mais sont centralisés dans les Data Centers propriétaires de la RATP en Île-de-France. Cette architecture en Cloud privé répond à trois enjeux majeurs :

- La haute disponibilité : Garantir que les ingénieurs du GIGP puissent pointer à tout moment.
- La scalabilité : Pouvoir augmenter les ressources du serveur (RAM, CPU) selon les besoins de l'application.
- La sécurité : Isolation totale des données sensibles par rapport au réseau internet public.

Accès via le Réseau Privé d'Entreprise (RPE) : Contrairement à une application web classique accessible via Internet, l'accès à l'interface de GAP nécessite d'être connecté au réseau local de la RATP.

- Flux réseau : Mon poste de travail communiquait avec un serveur de test.

Habilitations : L'accès à l'URL de l'application est protégé par une authentification liée à l'annuaire d'entreprise. Cela garantit que seul le personnel autorisé (développeurs, testeurs, utilisateurs GIGP) peut atteindre l'interface, ajoutant une couche de sécurité périphérique avant même la connexion à l'application.

3. Particularité : La Recette en Environnement de Production

La recette en environnement de test : Pour garantir la fiabilité de GAP, j'ai utilisé un environnement de test isolé. Cela m'a permis d'injecter des jeux de données sans risque pour l'activité réelle.

- Tests d'intégration réelle : À partir d'un fichier Excel regroupant ces matricules, je me suis connecté à chaque profil pour vérifier le bon fonctionnement des fonctionnalités. Cette approche "pas à pas" était indispensable pour m'assurer que l'application réagissait correctement pour chaque utilisateur.
- Contrôle de la suppression de masse : Pour valider les fonctions de suppression, j'ai créé mes propres jeux de données fictifs. En les supprimant manuellement, j'ai pu confirmer que l'outil était efficace .

Étude de cas n°1 : Analyse du bug des actions de masse

Lors de mes tests sur l'application GAP, j'ai remarqué que si l'application permettait d'ajouter des activités sur plusieurs jours simultanément, elle ne proposait pas la même flexibilité pour la suppression. Un utilisateur souhaitant corriger une erreur de saisie sur une semaine entière devait répéter l'opération de suppression jour après jour.

Ma démarche : Plutôt que de considérer cela comme un simple oubli, j'ai analysé cette limitation comme un frein à la productivité des agents du GIGP. J'ai donc pris l'initiative de suggérer une évolution fonctionnelle au développeur : aligner les capacités de suppression sur les capacités d'ajout.

- Objectif : Permettre la suppression groupée d'activités, même lorsqu'elles sont réparties sur plusieurs dates différentes.
- Le but : Réduire le nombre de clics et fluidifier la gestion du calendrier de pointage.

Étude de faisabilité technique : J'ai discuté avec le développeur des modalités d'implémentation de cette fonctionnalité :

- Traitement des données : Ma suggestion impliquait que le script Backend (PHP) puisse recevoir une liste d'identifiants hétérogènes (multi-dates) et les traiter dans une transaction SQL unique via une clause WHERE IN.
- Sécurité : Nous avons dû vérifier que cette souplesse n'augmentait pas le risque de suppression accidentelle de données réelles sur la production.

Bilan de la modification : Le développeur a accueilli favorablement ma proposition. Cette expérience a été très enrichissante car elle m'a permis de :

- Participer activement à l'évolution d'une application métier.
- Comprendre le cycle : Identification d'un besoin -> Suggestion technique -> Implémentation -> Validation.
- Démontrer ma capacité à proposer des solutions pour améliorer le confort des utilisateurs finaux.

Étude de cas n°2 : Intégrité des données et robustesse du système

Détection d'une anomalie de logique métier : Au-delà des tests fonctionnels classiques, j'ai réalisé des "tests aux limites" pour vérifier la robustesse de l'application GAP face à des comportements utilisateurs imprévus. J'ai ainsi découvert une faille critique dans les règles de gestion : l'application n'imposait aucun plafond de saisie. Techniquement, un utilisateur pouvait enregistrer un nombre illimité d'activités sur une seule journée, cumulant par exemple 10 J.H (soit 10 jours de travail théoriques) sur une date unique de 24 heures.

Le GIGP gérant des projets d'infrastructure de grande ampleur, cette absence de contrôle présentait des risques sérieux :

- Corruption des indicateurs : Les rapports de productivité exportés vers la direction auraient été faussés par des erreurs de saisie (ex: un zéro de trop).
- Incohérence de la base de données : Une base de données métier doit refléter la réalité physique. Permettre de dépasser 1 J.H par jour calendaire constitue une aberration logique qui fragilise la fiabilité du Système d'Information.

Correctifs techniques : Nous avons procédé à la résolution des anomalies bloquant le bon fonctionnement de l'application. Cette étape a permis de corriger les erreurs de traitement et de stabiliser l'application, garantissant ainsi que les fonctionnalités de gestion soient pleinement opérationnelles pour les agents du Grand Paris.

Validation côté Client (Frontend - JavaScript/Bootstrap) :

Rôle : Offrir un retour immédiat à l'utilisateur.

Action : Empêcher l'envoi du formulaire si le cumul des J.H dépasse le seuil autorisé sur la page. Cela améliore l'expérience utilisateur (UX) en évitant un rechargement de page inutile.

Limite : Cette sécurité peut être contournée (en désactivant le JS), elle n'est donc pas suffisante.

Validation côté Serveur (Backend - PHP/SQL) :

Rôle : Sécurité absolue et intégrité des données.

Action : Avant chaque requête INSERT ou UPDATE, le script PHP doit interroger la base de données pour calculer la somme des J.H déjà existants pour cet agent à cette date. Si la nouvelle saisie fait dépasser le total de 1 J.H, le serveur rejette la transaction et renvoie un message d'erreur.

Bilan : Cette étude de cas démontre l'importance du rôle du testeur : il ne doit pas seulement vérifier que le logiciel "marche", mais aussi s'assurer qu'il ne permet pas de faire des actions "impossibles" ou "dangereuses" pour l'entreprise.

Documentation de la mission GAP

La rédaction du dossier de Recette pour que mon travail soit exploitable par le développeur, j'ai formalisé mes tests sous forme d'un dossier de Recette. Ce document ne se contente pas de lister les erreurs, il sert de preuve de conformité. (voir en annexes).

- Le cahier de tests : Pour chaque fonctionnalité (Ajout, Duplication, Suppression multi-dates), j'ai rempli un tableau de suivi comprenant :
 - Le scénario (ex: "Dupliquer la journée du lundi vers le mardi").
 - Le résultat attendu (ex: "Les données sont copiées avec un nouvel ID").
 - Le résultat obtenu et le statut (Validé / Échoué).
- Preuves de test : J'ai joint les captures d'écran de l'interface (voir Annexes) pour illustrer les comportements anormaux avant correction.

Détail du cycle de traitement Chaque dysfonctionnement identifié sur l'application GAP a été traité selon les quatre étapes suivantes :

- Étape 1 Identification : Lors de l'utilisation de l'interface sur le réseau interne, je détectais des comportements s'écartant des besoins métier (ex: échec de suppression ou saisie de J.H incohérente).
- Étape 2 Qualification et Analyse : C'est l'étape la plus technique. Je cherchais à isoler la cause racine. Par exemple, déterminer si le problème venait d'une erreur de logique dans le code PHP (problème de traitement de tableau) ou d'un manque de règles de gestion dans la base de données.

- Étape 3 : Je formalisais mes conclusions par un compte-rendu écrit, suivi d'un échange oral avec le développeur. Cette communication directe permettait d'expliquer précisément le scénario de reproduction du bug, évitant ainsi les allers-retours inutiles.
- Étape 4 : Une fois le correctif déployé par le développeur directement sur l'environnement de production, je réalisais un nouveau passage complet. Ce retest servait à confirmer la résolution, mais aussi à effectuer des tests de non-régression pour s'assurer qu'aucune autre partie de l'application n'avait été impactée par la modification.

Validation de la persistance : Un point clé de ma validation finale a porté sur le module de Duplication de journée. Vérifier que les métadonnées (Contrat , Horaires ...) étaient intégralement conservées. Cette fonctionnalité est critique car elle évite la double saisie et réduit drastiquement le risque d'erreur humaine pour les collaborateurs.

Bilan sur le projet GAP : Cette première mission au sein de la RATP m'a permis de comprendre qu'un projet informatique n'est pas qu'une affaire de code. La qualité logicielle est le dernier rempart avant l'utilisateur final.

Cette autonomie sur l'environnement de production m'a préparé à la seconde grande mission de mon stage : le projet OMEGA.

Application : Nouveau outil de ticketing “OMEGA”

Contexte et Objectifs : Le projet OMEGA est né d'une nécessité de modernisation au sein de la RATP. L'outil de gestion d'incidents (ticketing), *osTicket*, présentait des limites en termes d'évolutivité et d'interface. L'objectif d'OMEGA est de proposer une solution transverse, capable de gérer les flux de support non plus pour une seule entité, mais pour l'ensemble des départements de la RATP INFRASTRUCTURES.

Choix Technologique : Le Framework Laravel L'application a été développée avec Laravel, un framework PHP réputé pour sa robustesse. Ce choix technique est stratégique pour plusieurs raisons :

- Architecture MVC (Modèle-Vue-Contrôleur) : Séparation claire entre la logique métier, la base de données et l'interface utilisateur. (voir schémas dans Annexes)
- Sécurité : Protection native contre les injections SQL, les failles XSS et CSRF (essentiel pour la sécurité des données internes RATP).
- Eloquent ORM : Facilite la manipulation de la base de données via des objets PHP plutôt que des requêtes SQL complexes.

Mon rôle dans le projet contrairement à GAP où j'étais focalisé sur la recette, mon intervention sur OMEGA a porté sur l'ingénierie des données et la structuration fonctionnelle. J'ai agi comme le garant de la cohérence des données avant le déploiement partielle. (voir l'interface en annexes).

La réussite d'un logiciel de support dépend avant tout de la qualité et de la structure de son annuaire. Ma mission a consisté à préparer l'intégration de centaine d'agent dans le nouvel écosystème OMEGA.

Le traitement des flux CSV : Le point de départ était un export massif issu d'*osTicket*, l'ancien outil de ticketing de la RATP. Ce fichier contenait des données brutes qu'il a fallu transformer pour les rendre exploitables par le framework Laravel.(voir Annexes)

- Nettoyage des données : J'ai procédé à la suppression des doublons et à la normalisation des formats (noms, prénoms, casse). Une vérification des matricules a été effectuée, car ils constituent la clé primaire permettant l'identification unique de chaque agent au sein du SI.
- Segmentation et Mapping : J'ai réalisé une correspondance technique entre les rubriques d'aides de *osTicket* et le catalogue d'OMEGA. Ce travail garantit que les flux de tickets soient dirigés vers les bons formulaires.

Le cœur de ma mission sur OMEGA a été de structurer la table des intervenants. Contrairement aux utilisateurs "clients" qui ne font que poser des questions, les intervenants sont les techniciens qui résolvent les incidents.

- Association directe : J'ai dû configurer l'association entre chaque intervenant et son service de rattachement (ex: Jean Dupont -> Service RH).
- Gestion du cloisonnement : Cette étape est le garant de la sécurité du système. En associant un intervenant à un service précis, on définit son périmètre de visibilité. Un intervenant "Infrastructure" ne pourra jamais consulter les tickets du service "RH", assurant ainsi une étanchéité totale des données sensibles.

Enjeux de la migration : Ce travail de préparation en amont dans les fichiers CSV était indispensable pour automatiser l'importation via Laravel. Une erreur dans ce fichier aurait paralysé le support technique :

- Risque d'erreur : Un intervenant mal affecté recevrait des tickets qui ne le concernent pas.
- Solution : L'utilisation de fonctions de recherche avancées (RECHERCHEV/XLOOKUP) pour croiser les fichiers d'intervenants avec les codes services de la RATP.

Pendant que je préparais les données, j'ai pu observer la phase de codage de l'application. Le choix de Laravel a permis une gestion simplifiée des relations entre les intervenants et les services.

L'IA comme accélérateur de développement (Cursor)

Le développeur utilisait Cursor, un IDE intégrant une intelligence artificielle. J'ai constaté que cet outil était particulièrement efficace pour :

- Générer les scripts d'import : Créer rapidement le code PHP capable de lire des fichiers CSV pour alimenter la base de données SQL.
- Vérification de cohérence : L'IA permettait de détecter des anomalies de typage dans le code avant même l'exécution des tests.
- Correction lors de blocages : J'ai observé que le développeur utilisait l'IA comme un partenaire de résolution de problèmes. Lorsqu'il était bloqué par une erreur complexe ou un bug logique, Cursor analysait le code et les logs pour proposer une solution immédiate, permettant de lever les verrous techniques beaucoup plus rapidement qu'une recherche classique.

Le projet OMEGA a été l'occasion pour moi de découvrir une méthode de travail au sein de l'entité U2D (Urbanisation de Développement et Data) : l'utilisation de l'IDE Cursor, un environnement de développement orienté vers l'Intelligence Artificielle.

Cursor n'est pas un simple éditeur de texte ; c'est un "fork" de Visual Studio Code qui intègre nativement des modèles de langage (LLM) avancés. Contrairement à un outil externe (comme ChatGPT), Cursor possède une connaissance contextuelle de l'intégralité du projet OMEGA :

- Il peut lire et comprendre l'architecture du framework Laravel.
- Il analyse les relations entre les fichiers (Models, Controllers, Migrations).
- Il suggère des corrections en fonction des spécificités du code déjà écrit.

Applications concrètes sur OMEGA

J'ai pu observer comment l'IA a permis d'accélérer des phases critiques du projet :

- Maintenance et Débogage (Aide au déblocage) : C'est lors des phases de blocage technique que l'outil s'est montré le plus utile. Lors de l'apparition d'erreurs pendant les tests de cloisonnement, l'IA analysait les logs d'erreurs et proposait instantanément une correction logique (ex: problème de liaison entre l'ID de l'intervenant et le service). Elle permettait au développeur de lever des verrous techniques rapidement lorsqu'une erreur complexe ralentissait sa progression.
- Documentation du code : L'IA a facilité la génération de commentaires techniques, rendant le code d'OMEGA plus maintenable pour les futurs développeurs de l'unité.

- Le rôle du dev face à l'IA : Cette expérience m'a permis de comprendre que l'IA ne remplace pas le développeur, mais agit comme un "Assistant". L'expertise humaine reste indispensable pour :
 1. La Sécurité : Vérifier que le code généré ne crée pas de failles (ex: accès non autorisé aux tickets RH).
 2. La Logique Métier : S'assurer que les suggestions de l'IA respectent scrupuleusement le fonctionnement des services de la RATP.
 3. La Validation : L'IA propose, mais le développeur valide et teste en environnement réel.

COMPÉTENCES

Avant le début de ce stage, mon établissement nous a demandé d'identifier et de compléter certaines compétences à partir d'un catalogue précis. Au cours de cette période, j'ai eu l'opportunité de développer et valider la quasi-totalité de ces compétences grâce aux missions confiées sur les applications GAP et OMEGA.

Réalisations en milieu professionnel en cours de seconde année						
Maintenance corrective sur l'application GAP	01/12/2025 au 23/01/2026	X			X	
Migration et Mapping de données (osTicket vers OMEGA)	01/12/2025 au 23/01/2026	X			X	
Gestion et organisation d'annuaire (Projet OMEGA)	01/12/2025 au 23/01/2026	X	X			
Recette fonctionnelle et Assurance Qualité (OMEGA)	01/12/2025 au 23/01/2026				X	X
Veille technologique et observation (Cursor IA)	01/12/2025 au 23/01/2026					X

PROBLÈMES RENCONTRÉS

Au cours de mon stage, j'ai été confronté à des défis qui m'ont permis de développer ma capacité d'adaptation et technique. Voici les principaux obstacles rencontrés :

Adaptation à un écosystème complexe

Lors de mon intégration, la multiplicité des services (ISI, U2D, GIGP) et la diversité des interlocuteurs rendaient la compréhension des circuits de décision difficile. Identifier précisément le rôle de chaque entité au sein de la RATP était un défi pour être rapidement opérationnel.

Cette phase d'immersion m'a permis de comprendre les flux de travail et de devenir autonome plus rapidement dans mes missions de support et de test.

Pollution et incohérence des données sources (Projet OMEGA)

Lors de la préparation de la migration des données, j'ai constaté que l'export CSV de l'ancien outil (osTicket) contenait de nombreuses erreurs : des agents étaient rattachés à des services obsolètes ou n'existant plus. Importer ces données telles quelles aurait provoqué des erreurs critiques de routage des tickets dans le nouvel outil Laravel.

J'ai mis en place un processus de curation (nettoyage) de données. À l'aide de fonctions Excel avancées (Recherche X, filtres), j'ai croisé les données du fichier avec le référentiel actuel des services RATP. J'ai ensuite sollicité mon tuteur pour valider les corrections et les rattachements d'agents.

Le Résultat : Ce travail de nettoyage a garanti l'intégrité de la base de données.

CONCLUSION

Ce stage au sein de l'équipe de développement a été une étape clé pour consolider mes compétences en BTS SIO option SLAM. À travers le projet GAP, j'ai appris que le métier de développeur exige une grande rigueur pour diagnostiquer des bugs et maintenir des applications existantes.

Le projet OMEGA m'a ensuite permis de comprendre l'importance de la gestion des données et des tests lors d'un déploiement pour une structure comme le Groupe RATP. En travaillant sur le mapping des rubriques et l'organisation des intervenants, j'ai pris conscience que la réussite d'un logiciel dépend autant de sa préparation que de son code.

Enfin, les outils comme Laravel et Cursor m'ont donné une vision concrète des méthodes de travail actuelles. Cette expérience me donne envie de poursuivre mon parcours pour aller plus loin dans la compréhension de l'informatique. Mon objectif est d'acquérir des bases plus larges et plus théoriques, afin de pouvoir, à terme, participer à la conception de projets encore plus ambitieux.

ANNEXES

RATP INFRASTRUCTURES

Jean-Louis Houpert
Stéphanie Vallée Assistante



Organigramme sur l'ensemble des différents pôles



L'équipe U2D (l'équipe dans laquelle j'étais)

Gestion Avancée du Pointage

GIGP / GAP by U2D



BIENVENUE DANS GAP

POINTAGE | CONSULTATION

ADMIN | REF | USER

GAP EN QUELQUES CHIFFRES



Interface du GAP

POINTAGE | CONSULTATION

ADMIN | REF | USER

JANVIER 2026 | Presse-papier | Presse papier vide | Effacer

MON POINTAGE

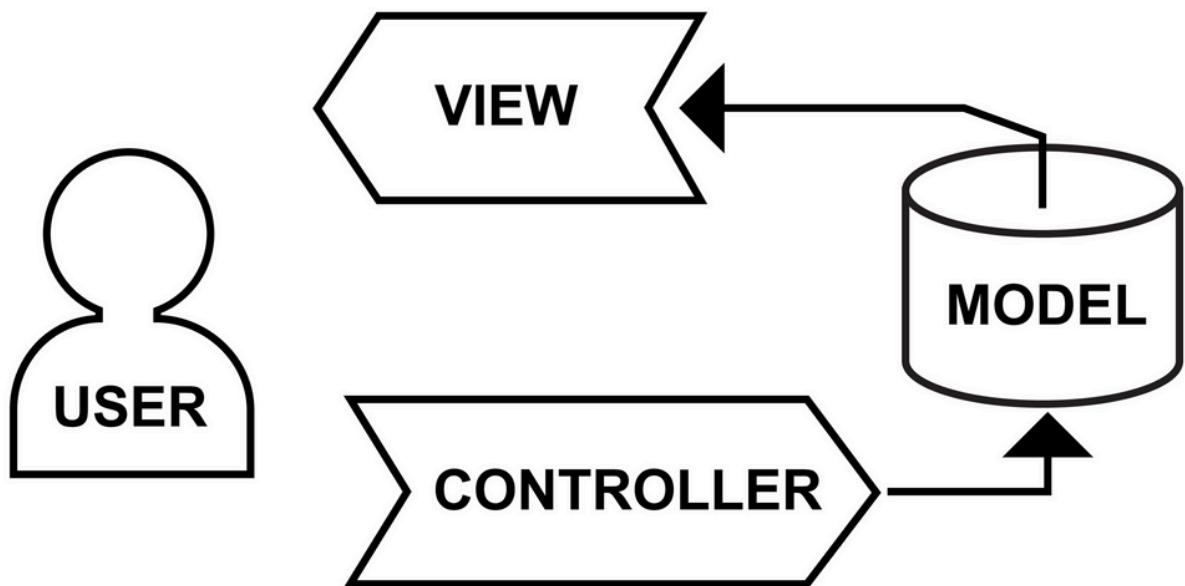
DIMANCHE	LUNDI	MARDI	MERCREDI	JEUDI	VENDREDI	SAMEDI
28	29	30	31	01 Férié	02	03
04	05	06	07	08	09	10
11	12	13	14	15	16	17



Pointage GAP

Fonctionnalité	Scénario de test	Statut	Commentaire technique
Calcul J.H	Saisie de 0.5 + 0.5	OK	Somme correcte (1.0 J.H)
Suppression Multiple	Sélectionner 3 lignes + Delete	KO	Conflit Array vs ID simple en PHP
Duplication	Bouton "Copier vers demain"	OK	Persistante des données vérifiée

Extrait du cahier de tests



MODEL - VIEW - CONTROLLER PATTERN

Schémas MVC (modèle - vue - contrôleur)

Titre	Description	Status	Dernière mise à jour
#423	IMPORTANT Accès à des ressources bureautiques	Fermé	Il y a 2 jours
#422	Dossier très important !! Accès à des ressources bureautiques	Fermé	Il y a 2 jours
#421	Bonjour Accès à des ressources bureautiques	En cours	Il y a 2 jours
#420	test du ticket pour omega Assistance OMEGA	En cours	Il y a 5 jours

Services favoris

- Assistance AKUITEO
- Service de Assistance AKUITEO
- Assistance FNCA
- Service de Assistance FNCA
- Assistance MESURE
- Service de Assistance MESURE
- Assistance OMEGA
- Service de Assistance OMEGA
- Assistance ORQUESTR
- Service de Assistance ORQUESTR
- Photographie
- Service de Photographie

Compteur de tickets: 4

Interface de OMEGA

```

// Lire l'en-tête (première ligne)
$header = str_getcsv(string: array_shift(array: &$amp;lines), separator: ';');
if (empty($header)) {
    $this->command->error("Le fichier CSV est vide ou invalide.");
    return;
}

$processedCount = 0;
$skippedCount = 0;
$errorCount = 0;
$totalLines = 0;

$this->command->info('Traitement des lignes du CSV...');

// Traiter chaque ligne
foreach ($lines as $line) {
    if (empty(trim(string: $line))) {
        continue;
    }

    $row = str_getcsv(string: $line, separator: ';');
    $totalLines++;
    if (count(value: $row) < 3) {
        $skippedCount++;
        continue;
    }
}

```

Script d'importation

GLOSSAIRE TECHNIQUE

Infrastructure et Environnement RATP

RATP INFRASTRUCTURES : Filiale spécialisée dans la maintenance, la gestion technique et la surveillance du réseau de transport historique.

ISI (Industrialisation des Systèmes d'Information) : Unité informatique chargée de garantir la robustesse, la sécurité et la mise en production des applications du groupe.

GIGP (Gestionnaire d'Infrastructure Grand Paris) : Entité de la RATP dédiée au pilotage des projets d'infrastructure liés au nouveau réseau de transport francilien.

RPE (Réseau Privé d'Entreprise) : Réseau informatique interne et sécurisé de la RATP, non accessible depuis l'internet public sans habilitations spécifiques.

VM (Virtual Machine) : Environnement informatique qui simule un ordinateur physique à l'intérieur d'un serveur, permettant la centralisation et la scalabilité des ressources.

Gestion de Projet et Qualité (QA)

J.H (Jour-Homme) : Unité de mesure quantifiant la charge de travail d'une personne durant une journée. Elle sert au pilotage budgétaire et à l'analyse de productivité.

Recette / Homologation : Phase de vérification durant laquelle on teste l'application pour s'assurer qu'elle répond strictement au cahier des charges.

Tests aux limites (Edge Cases) : Tests consistant à pousser le logiciel dans ses retranchements en saisissant des données extrêmes ou illogiques.

Non-régression : Tests effectués après une modification du code pour s'assurer que les fonctionnalités existantes n'ont pas été dégradées par la mise à jour.

Bug Tracking : Processus de détection, de signalement et de suivi de la résolution des anomalies logicielles.

Développement et Données (Projets GAP & OMEGA)

CSV : Format de fichier texte tabulaire permettant d'échanger des données entre des outils différents (ex: d'osTicket vers Excel).

Mapping : Opération de mise en correspondance de données entre deux systèmes (ex: lier un ancien rubrique d'osTicket au nouveau service OMEGA).

Laravel : Framework PHP moderne facilitant le développement d'applications web grâce à une structure organisée et sécurisée.

MVC (Modèle-Vue-Contrôleur) : Architecture logicielle séparant les données (Modèle), l'interface (Vue) et la logique de traitement (Contrôleur).

IDE (Integrated Development Environment) : Logiciel de travail des développeurs (comme Cursor ou VS Code) regroupant les outils d'écriture, de test et de débogage du code.

LLM (Large Language Model) : Intelligence Artificielle (comme celle intégrée à Cursor) capable de comprendre et de générer du code informatique pour assister le développeur.

