

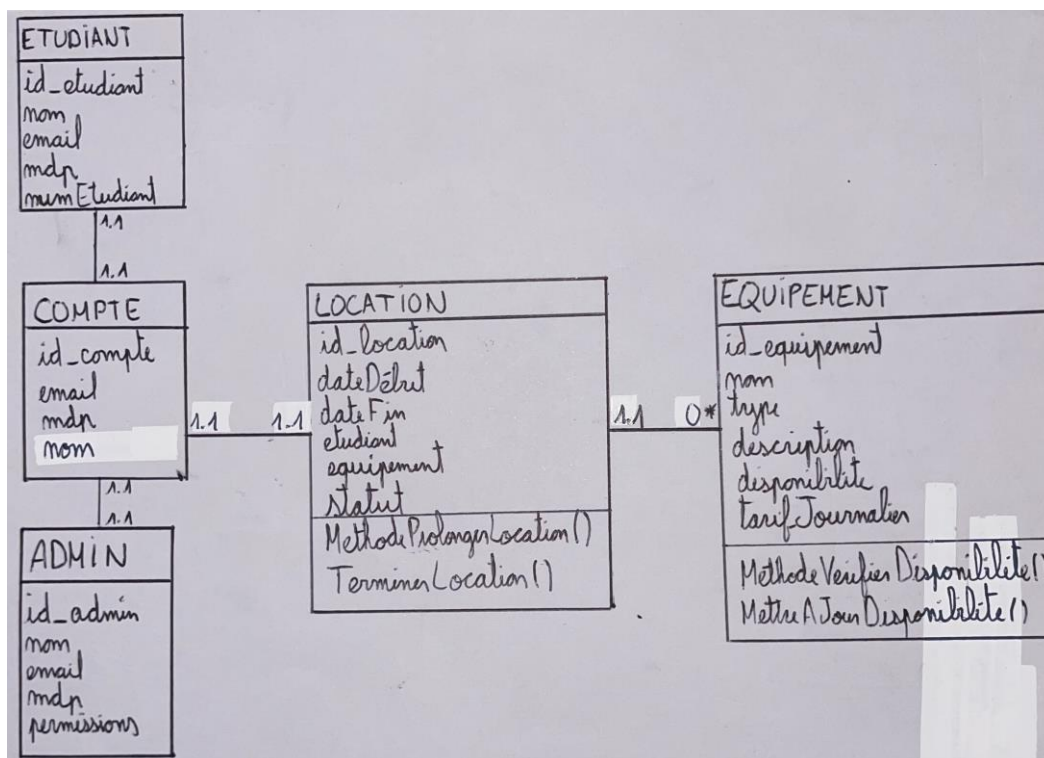
Cahier des Charges Technique

1. Présentation générale

- **Nom du projet** : Loc'Info
- **Objectif** : Offrir une solution de location de matériel informatique pour les étudiants, leur permettant de louer des ordinateurs, imprimantes, accessoires, etc., pour une période définie (en général tout le cursus ou sur une année par exemple).
- **Public cible** : Étudiants qui ont besoin de matériel informatique pour une durée limitée (ex : examens, projets).
- **Valeur ajoutée** : Facilite l'accès à du matériel de qualité sans avoir à investir dans un achat, et assure une gestion centralisée des équipements disponibles.

2. Schéma UML de l'application

- Schéma UML



- Diagramme de Cas d'Utilisation :

- **Acteurs :**
 - **Étudiant** : Peut parcourir les équipements, effectuer une location, prolonger une location, et consulter son historique.
 - **Administrateur** : Gère les équipements disponibles, suit les retours, gère les comptes utilisateurs.
- **Cas d'utilisation principaux :**
 - **Parcourir les équipements** : L'étudiant consulte le catalogue, applique des filtres de recherche, et examine la disponibilité.
 - **Réserver un équipement** : L'étudiant sélectionne l'équipement souhaité, vérifie la disponibilité, et choisit une période de location.
 - **Prolonger la location** : L'étudiant peut prolonger une location existante s'il en a besoin plus longtemps.
 - **Gérer les équipements** : L'administrateur ajoute, met à jour, ou supprime des équipements.
 - **Suivre les retours** : L'administrateur valide le retour des équipements et met à jour le statut dans le système.
- **Diagramme de Classes (exemple) :**
 - **Classes principales :**
 - **Équipement** : Attributs nom, type, description, disponibilité, tarifJournalier. Méthodes vérifierDisponibilité(), mettreÀJourDisponibilité().
 - **Location** : Attributs idLocation, dateDébut, dateFin, étudiant, équipement, statut. Méthodes prolongerLocation(), terminerLocation().
 - **Étudiant** : Attributs nom, email, motDePasse, numÉtudiant.
 - **Administrateur** : Attributs nom, email, motDePasse, permissions.
 - **Compte** : Lié aux classes Étudiant et Administrateur, gère l'authentification et la gestion des droits d'accès.
- **Diagramme de Séquence (exemple pour une réservation) :**
 - Étudiant consulte la liste d'équipements → Envoie une requête de réservation → Vérifie disponibilité → Confirme la réservation → Enregistrement de la réservation → Mise à jour de la disponibilité.

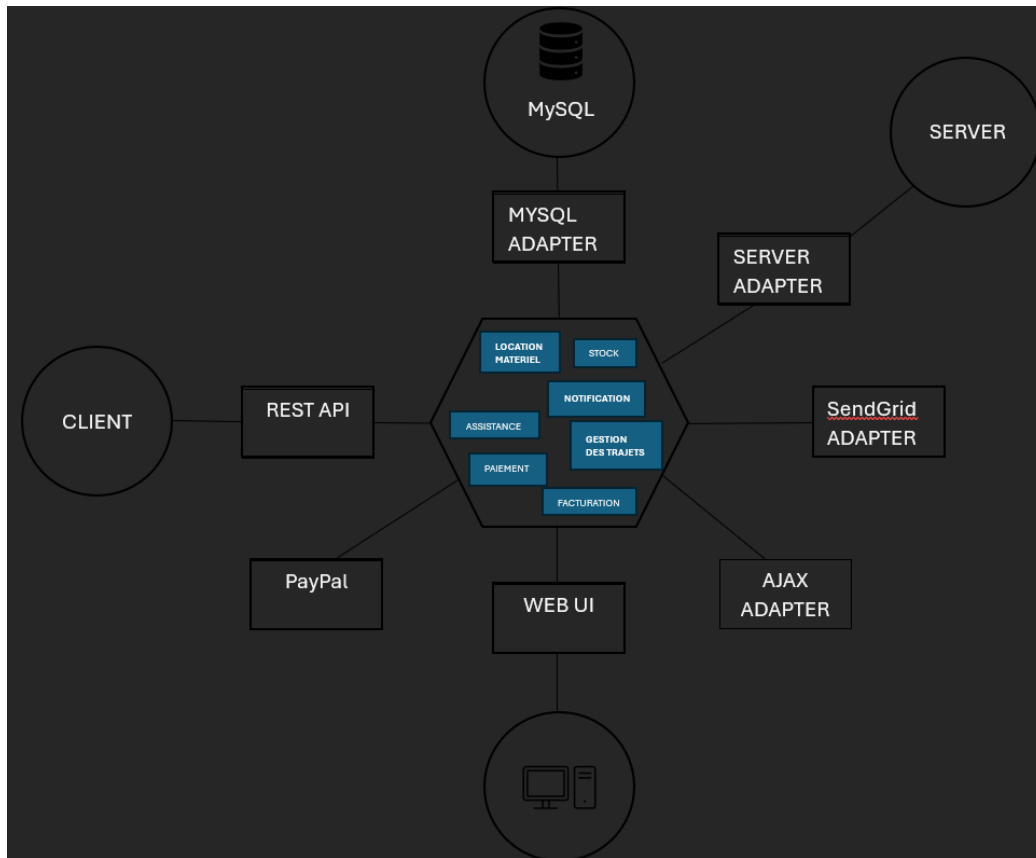
3. Choix Technologiques

- **Front-end** : **HTML** et **CSS** pour la structure et le style ; **JavaScript** pour des interactions utilisateur dynamiques.
- **Back-end** : **PHP** pour la logique métier, permettant de gérer les demandes de location, la base de données et les comptes utilisateurs.
- **Base de données** : **MySQL** ou **MariaDB** pour le stockage de données telles que les équipements, les utilisateurs, et les informations de location.
- **Serveur Web** : **Apache** pour héberger le site web et assurer la gestion des requêtes.
- **API de Notifications** : Utilisation d'une API (ex. : Twilio ou SendGrid) pour envoyer des rappels de retour et de prolongation.
- **Authentification** : **OAuth** ou système de jetons pour assurer une sécurité supplémentaire aux utilisateurs (surtout pour l'administration).

4. Projection dans une Architecture Client-Serveur

- **Client (côté utilisateur)** :
 - Interface web accessible depuis un navigateur, permettant aux étudiants de naviguer, réserver et consulter leurs locations.
 - **Communication avec le serveur** : Utilisation d'**AJAX** pour des requêtes asynchrones et fluides ; affichage dynamique pour les équipements disponibles.
- **Serveur (côté application)** :
 - Le serveur reçoit les requêtes du client (ex : réservation, prolongation), vérifie les disponibilités en base de données, et renvoie une réponse.
 - **API REST** pour standardiser les échanges entre le front-end et le back-end (ex. : /api/equipements, /api/reservations).
- **Architecture de sécurité** :
 - **HTTPS** pour sécuriser les échanges.
 - Authentification par session pour gérer les connexions utilisateur et administrateur, avec des droits d'accès adaptés.
- **Processus de Location (projection détaillée)** :
 - **Étape de consultation** : L'étudiant parcourt le catalogue, choisit un équipement, et vérifie sa disponibilité.
 - **Étape de réservation** : Le serveur enregistre la réservation et met à jour la disponibilité de l'équipement.
 - **Étape de rappel** : Avant la fin de la période de location, le serveur envoie une notification (SMS ou email) rappelant la date de retour.

- **Étape de retour** : Lors du retour, l'administrateur vérifie l'équipement, met à jour son statut, et clôture la location.



5. Fonctionnalités Techniques

- **Gestion des Équipements :**
 - Fonction de recherche et de filtres pour afficher les équipements disponibles par catégorie, nom ou tarif.
 - Aide au choix du matériels à l'aide d'informations disponible sur le site ou de personelles joignable directement sur le site pour un accompagnement par téléphone.
 - Mise à jour automatique en temps réel (à quelques minutes près) des disponibilités dès qu'un équipement est loué ou retourné.
- **Gestion des Locations :**
 - **Réservation** : Possibilité de réserver pour des périodes spécifiques, en vérifiant automatiquement la disponibilité en fonction des dates.
 - **Prolongation de la période de location** : Option de prolongation disponible sous certaines conditions.

- **Suivi des équipements retournés** : Mise à jour de l'état de chaque équipement pour assurer sa disponibilité pour d'autres étudiants.
- **Notifications et alertes** :
 - **Rappels de retour** : Envoi d'alertes avant la fin de la période de location.
 - **Alertes d'indisponibilité** : Notifications si un équipement est indisponible pour la période souhaitée, avec possibilité d'ajouter l'étudiant sur une liste d'attente.
- **Sécurité et Authentification** :
 - Sécurisation des comptes utilisateurs avec des mots de passe chiffrés et sessions sécurisées.
 - Restrictions pour les administrateurs afin de limiter les modifications aux équipements et aux comptes.
- **Rapports et statistiques** (pour l'administrateur) :
 - Génération de rapports sur les équipements les plus loués, périodes de forte demande, et historique des locations.
 - Suivi des utilisateurs les plus actifs, gestion des pénalités pour les retards.

6. Les trois visions

Vision opérationnelle :

L'objectif du site est de permettre à une clientèle avec des moyens limités d'accéder à une variété d'outils informatiques grâce au service de location proposé sur le site. A l'aide du site, des informations complémentaires sont disponibles pour aider l'utilisateur sur son choix et il peut également rentrer en contact téléphonique avec un professionnel du site, cela permet au client d'avoir deux styles de soutien différents pour les aidés dans leurs recherches.

Tout cela garantissant une expérience fluide et accessible.

Vision Fonctionnelle :

- **Catalogue d'Équipements** :
 - **Parcours et sélection** : Les étudiants consultent le catalogue, appliquent des filtres et vérifient la disponibilité des équipements (ordinateurs, imprimantes, accessoires).
 - **Détails et mises à jour** : Les équipements affichent des informations complètes, avec une mise à jour en temps réel de leur disponibilité.
- **Gestion des Comptes Utilisateurs** :

- Création de compte et connexion sécurisée : Les étudiants s'enregistrent et se connectent pour accéder aux services.
- Espace personnel : Chaque utilisateur peut voir son historique de location et prolonger ses réservations si nécessaire.
- Réservation et Suivi des Locations :
 - Réservation d'équipement : Sélection et réservation d'un matériel pour une période définie, avec possibilité de prolongation selon disponibilité.
 - Retour et mise à jour de l'état : L'administrateur valide les retours et vérifie l'état des équipements.
- Fonctions Administrateur :
 - Gestion des équipements : Ajout, modification, suppression des équipements et suivi des retours pour assurer la disponibilité des matériels en location.
 - Suivi de l'utilisation : Surveillance de l'état des équipements et identification des articles les plus demandés pour optimiser la gestion des stocks.
- Notifications et Support Client :
 - Rappels automatiques : Alertes de retour et prolongation envoyées par email ou SMS.
 - Assistance : Messagerie en ligne pour les questions, centre d'aide avec FAQ et guides.

Vision Organique

- **Serveur et Hébergement :**
 - **Serveur Web (Apache)** : Héberge le site, gère les requêtes utilisateurs, et met en place le site.
 - **Base de données (MySQL)** : Stocke toutes les données liées aux équipements, utilisateurs, et locations de manière sécurisée.
- **Front-end :**
 - **Interface utilisateur (HTML, CSS, JavaScript)** : Fournit une expérience fluide aux utilisateurs, avec des formulaires intuitifs pour la recherche, la réservation, et la gestion des comptes.
- **Back-end :**
 - **API REST** : Facilite les échanges entre le front-end et la base de données pour des actions comme les réservations, les retours, et les notifications.
 - **Gestion des sessions et authentification sécurisée** : Garantit la sécurité des accès, avec des droits spécifiques pour les étudiants et les administrateurs.
- **Sécurité et Protection des Données :**

- **Chiffrement des mots de passe et HTTPS** : Assure la protection des données personnelles et des transactions.
- **Double authentification (2FA)** : En plus de l'identifiant et du mot de passe, un second facteur, comme un code envoyé par SMS ou une application de sécurité, est requis pour accéder au système, renforçant la sécurité des comptes administrateurs et utilisateurs.
- **API de Notification** :
 - **Envoi de rappels par SMS ou email (Twilio ou SendGrid)** : Automatisation des rappels pour les retours et les prolongations, afin de garder les utilisateurs informés

6. Différentes couches et interactions

Couche Physique

- **Serveurs physiques ou virtuels** : Hébergent le site, les applications back-end et la base de données. Ces serveurs peuvent être internes à l'entreprise ou dans le cloud (ex. : AWS, Azure).
- **Équipements réseau** : Routeurs, pare-feu et switchs pour sécuriser et gérer le trafic entre le serveur et les utilisateurs.
- **Postes utilisateurs** : Ordinateurs, tablettes ou téléphones utilisés par les étudiants et administrateurs pour accéder à la plateforme.

Couche Réseau

- **Protocoles de communication (TCP/IP, HTTP/HTTPS)** : Assurent la transmission des données entre le client (navigateur de l'utilisateur) et le serveur.
- **HTTPS** : Protège les échanges entre le client et le serveur pour garantir la sécurité des données.
- **APIs de notifications (SendGrid)** : Utilisées pour envoyer des rappels ou alertes aux utilisateurs via email ou SMS.

Couche Logicielle

- **Serveur Web Apache** : Gère les requêtes des utilisateurs, exécute les scripts PHP pour les fonctionnalités du sites, et sert les pages HTML.
- **Base de données MySQL** : Stocke et gère les données utilisateurs, informations d'équipement et de location.
- **Environnement front-end** : HTML, CSS, JavaScript pour créer une interface utilisateur interactive et intuitive.

Couche Données

- **Gestion des données utilisateurs et équipements** : Stockées dans une base de données relationnelle (MySQL) et incluent les comptes utilisateurs, historiques de location, et disponibilités des équipements.
- **Sécurité des données** : Chiffrement des mots de passe et gestion des droits d'accès pour protéger les données sensibles et assurer leur intégrité.

Couche Applicative

- **Interface utilisateur** : Page de catalogue, recherche, réservation, et espace personnel pour que les étudiants puissent gérer leurs locations.
- **Logique de gestion des locations** : Gère les processus de réservation, prolongation, et retour d'équipement, avec vérification des disponibilités.
- **Fonctions administratives** : Permet aux administrateurs de gérer les équipements, superviser les retours, et mettre à jour le catalogue.
- **Notifications et rappels** : Automatisation de l'envoi d'alertes pour le rappel des retours ou des prolongations, ainsi que la gestion des comptes utilisateurs.

7. Interactions et indépendances entre les couches

1. Interaction entre la couche matérielle et la couche logicielle

Les serveurs hébergent les logiciels comme Apache, PHP, et MySQL. Ces logiciels fonctionnent grâce au matériel pour gérer les requêtes des utilisateurs et stocker les données.

2. Interaction entre la couche logicielle et la couche données

Les logiciels d'application comme PHP utilisent la base de données MySQL pour stocker et récupérer des informations sur les équipements, les comptes utilisateurs, et les locations.

3. Interaction entre la couche réseau et la couche applicative

Les utilisateurs accèdent au site via Internet. Les fonctionnalités du site, comme la réservation et la gestion des locations, utilisent le réseau pour envoyer et recevoir des informations entre le serveur et le navigateur des utilisateurs.

8. Avantage de notre architecture :

- **Modularité accrue** : L'architecture client-serveur permet de séparer le front-end et le back-end, facilitant la maintenance et les mises à jour sans affecter l'ensemble du système, contrairement aux systèmes monolithiques où chaque modification impacte l'ensemble du code.
- **Scalabilité** : Les composants peuvent être répartis sur plusieurs serveurs en cas d'augmentation du nombre d'utilisateurs ou de la charge, ce qui est difficile avec une architecture monolithique.
- **Performances optimisées** : En répartissant les traitements entre le client et le serveur, l'architecture réduit les temps de latence et améliore l'expérience utilisateur, offrant une navigation plus fluide.
- **Sécurité renforcée** : Le système est protégé par une authentification centralisée et un chiffrement HTTPS, assurant la confidentialité et la protection des données sensibles comme les informations de location et d'utilisateur.
- **Évolutivité** : Il est possible d'ajouter des fonctionnalités (comme des rappels automatiques) sans perturber l'ensemble du système, offrant une flexibilité que l'architecture monolithique ne permet pas aisément.

9. Pitch

E-Loc Informatique est un système innovant de location de matériel informatique conçu pour répondre aux besoins temporaires des étudiants. Sa structure client-serveur utilise une architecture claire, modulaire et sécurisée, permettant une réactivité et une gestion efficace des données sensibles. Les utilisateurs peuvent facilement parcourir le catalogue d'équipements, réserver des articles, et prolonger leur période de location grâce à une interface intuitive, tandis que les administrateurs gèrent les stocks et valident les retours avec fluidité.

L'architecture client-serveur garantit la sécurité et la rapidité d'accès, séparant les traitements côté client (navigation, réservation) de ceux du serveur (authentification, gestion des stocks et des locations). Comparée à une architecture monolithique, qui intègre toutes les fonctions sur un seul serveur, notre solution est plus évolutive et permet de répartir les charges entre plusieurs serveurs, réduisant les temps d'attente et les risques de panne.

De plus, cette modularité permet d'ajouter facilement de nouvelles fonctionnalités (comme des notifications automatisées pour les retours) sans impacter les autres

composants. E-Loc Informatique est donc un choix plus souple et robuste pour offrir un service performant, adapté à une audience étudiante en demande de flexibilité et d'accessibilité.