

UNIVERSITÉ DE LORRAINE
IUT DE METZ

SAE 3.01 : Développement d'une Application
Communicante

SPARKING

Solution de Stationnement Urbain Intelligent

Auteurs :

Enzo LÉONARD-RUBECK
Alexandre BARRE
Thomas BOGUET
Yoann DARCIAUX

Encadrants :

Équipe Pédagogique
BUT Informatique

Année Universitaire 2024 – 2025

Remerciements

Nous tenons à remercier sincèrement l'équipe pédagogique de l'IUT de Metz pour leur encadrement et leur disponibilité tout au long de ce semestre. Ce projet SAE 301 a été une opportunité précieuse de consolider nos compétences en développement full-stack, en administration système et en réseau dans un contexte quasi-professionnel.

Nous remercions également l'Université de Lorraine pour la mise à disposition des infrastructures techniques (serveurs, réseaux) nécessaires au déploiement et aux tests grandeur nature de notre solution **Sparking**.

Table des matières

Introduction	4
1 Gestion et Conception du Projet	5
1.1 Analyse des besoins	5
1.2 Maquettage et UX/UI Design	5
1.2.1 Charte Graphique "Dark Mode"	5
1.2.2 Approche Mobile First	6
1.3 Modélisation des Données	6
2 Infrastructure Système et Réseau	7
2.1 Architecture Réseau (Packet Tracer)	7
2.2 Services Hébergés	7
3 Développement Web et Algorithmique	8
3.1 Architecture Technique	8
3.2 Le Cœur du Système : La Cartographie	8
3.2.1 Clustering et Lisibilité	8
3.2.2 Géolocalisation et Fallback	9
3.3 Fonctionnalités Avancées	9
3.3.1 Algorithme "Le plus proche"	9
3.3.2 Rayon de Recherche	9

3.4	Internationalisation	9
4	Développement Mobile (Android)	10
4.1	Adaptation de la solution	10
4.2	Ergonomie Mobile	10
5	Bilan et Perspectives	11
5.1	Difficultés rencontrées	11
5.2	Perspectives d'amélioration (V2)	11
6	Gestion de Projets en mode Agile/Scrum	12
6.1	Capacité d'effort de l'équipe	12
6.2	Carnet de produit (Product Backlog)	12
6.3	Estimation du MVP (Minimum Viable Product)	13
6.4	Carnets de Sprint (Sprint Backlogs)	14
6.4.1	Sprint 1 : Expansion Géographique	14
6.4.2	Sprint 2 : Utilisateurs et Données	14
6.4.3	Sprint 3 : Finitions et Mobilité	15
6.5	Suivi des Sprints (Trello)	15
6.6	Livraison Finale	17

Introduction

La mobilité urbaine représente l'un des défis majeurs des villes modernes. Avec l'augmentation constante du parc automobile, la recherche d'une place de stationnement est devenue une source quotidienne de stress, de congestion du trafic et de pollution atmosphérique. C'est dans ce contexte problématique que naît le projet **Sparking**.

Sparking est une solution applicative communicante complète (Web et Mobile) conçue pour fluidifier l'expérience de stationnement en ville. Elle permet aux utilisateurs :

- De géolocaliser en temps réel les parkings disponibles ;
- De consulter leurs caractéristiques détaillées (tarification, places PMR, bornes électriques) ;
- D'initier un itinéraire optimisé vers le lieu choisi.

Au-delà du simple stationnement, Sparking intègre une dimension **multimodale** innovante en proposant automatiquement les arrêts de transports en commun situés dans un rayon de 500 mètres autour du lieu de stationnement, favorisant ainsi l'intermodalité (voiture + bus/métro).

Ce rapport détaille la réalisation technique de ce projet dans le cadre de la SAE 301. Il couvre la conception, l'architecture réseau simulée, le développement web et mobile, ainsi que les choix algorithmiques qui font la force et la robustesse de notre solution.

1 | Gestion et Conception du Projet

1.1 Analyse des besoins

Le projet a été conçu pour répondre aux attentes de deux acteurs principaux, identifiés lors de la phase d'analyse :

L'automobiliste urbain Il cherche avant tout à optimiser son temps de trajet. Son besoin principal est la rapidité d'information : *"Où est le parking le plus proche ?"*
"Reste-t-il de la place ?"

Le touriste Il ne connaît pas la topologie de la ville (Metz ou Londres). Il a besoin de filtres spécifiques (Hauteur max pour son véhicule, proximité du centre-ville) et d'une interface multilingue pour se guider sans friction.

1.2 Maquettage et UX/UI Design

L'identité visuelle de Sparking a été pensée pour une utilisation en situation de mobilité (souvent au volant, à l'arrêt, ou en marchant).

1.2.1 Charte Graphique "Dark Mode"

L'interface privilégie des tons sombres (gris anthracite) avec des accents vifs (orange/jaune). Ce choix n'est pas uniquement esthétique : il assure une lisibilité maximale sans éblouir l'utilisateur, notamment lors d'une utilisation nocturne ou dans un habitacle de voiture sombre.

1.2.2 Approche Mobile First

Bien que l'application soit accessible via un navigateur web de bureau, la conception est résolument *Mobile First*. Les boutons d'action (Recherche, Itinéraire, Annulation) sont dimensionnés et positionnés dans une zone accessible avec le pouce, pour faciliter les interactions tactiles.

1.3 Modélisation des Données

La persistance des données est assurée par une base de données relationnelle **MySQL** hébergée sur le serveur de l'IUT. Le schéma s'articule autour de plusieurs entités clés :

- **Utilisateurs** : Stockage sécurisé des identifiants (mots de passe hachés via `password_hash`) et des préférences (ville par défaut).
- **Parkings** : Cette table centrale contient les données statiques (Latitude, Longitude, Tarifs, PMR) et dynamiques (Places libres mises à jour pour Metz).
- **Villes** : Permet l'extensibilité de l'application. Actuellement, Metz est la ville par défaut, mais l'architecture permet d'ajouter Londres.

2 | Infrastructure Système et Réseau

L'application Sparking ne repose pas uniquement sur du code applicatif, mais sur une infrastructure réseau simulée et validée via **Cisco Packet Tracer**.

2.1 Architecture Réseau (Packet Tracer)

Le fichier `SAE301_MVP.pkt` joint au projet matérialise l'infrastructure nécessaire au déploiement de Sparking. L'architecture a été pensée pour garantir la sécurité et la disponibilité des services.

- **Topologie** : Nous avons mis en place une architecture client-serveur classique segmentée.
- **Routage et Commutation** : Des routeurs assurent l'interconnexion entre le réseau local (LAN) de l'entreprise fictive et l'accès extérieur (Internet simulé), permettant aux clients Web et Android de communiquer avec l'API.

2.2 Services Hébergés

Pour rendre l'application fonctionnelle, plusieurs services ont été configurés sur les serveurs de la topologie :

Serveur Web (HTTP/HTTPS) Il héberge les fichiers sources (PHP, JS, CSS) et traite les requêtes des clients. C'est le point d'entrée unique de l'application.

Service DNS Il assure la résolution des noms de domaine (ex : `sparking.iut`) vers les adresses IP des serveurs, facilitant l'accès utilisateur sans avoir à retenir d'adresses IP.

Serveur de Base de Données Isolé pour des raisons de sécurité, il répond uniquement aux requêtes SQL envoyées par le serveur Web (Backend PHP), empêchant les accès directs depuis l'extérieur.

3 | Développement Web et Algorithmique

3.1 Architecture Technique

L'application Web repose sur une stack standard, robuste et maîtrisée :

- **Frontend** : HTML5, CSS3, JavaScript ES6. L'utilisation de JS asynchrone (Fetch API) est cruciale pour le chargement des données de parkings sans recharger la page.
- **Backend** : PHP 7+ gère la logique serveur, les sessions utilisateurs et les interactions sécurisées avec la base de données (PDO).

3.2 Le Cœur du Système : La Cartographie

L'élément central de Sparking est la carte interactive, propulsée par la bibliothèque open-source **Leaflet.js**.

3.2.1 Clustering et Lisibilité

L'affichage de tous les parkings simultanément rendrait la carte illisible (surcharge cognitive). Nous avons implémenté une logique de **Clustering** (grappes de marqueurs).

- Lorsque l'utilisateur dézoome, les parkings proches se regroupent en un cercle de couleur indiquant leur nombre.
- Au clic sur un cluster, la carte zoome automatiquement pour révéler les marqueurs individuels (*Cluster Spiderfy*).

3.2.2 Géolocalisation et Fallback

L'application utilise l'API `navigator.geolocation` du navigateur.

- **Cas nominal** : La position GPS réelle est récupérée pour centrer la carte sur l'utilisateur.
- **Fallback (Gestion d'erreur)** : Si l'utilisateur refuse la géolocalisation ou si le signal est absent, l'application positionne par défaut l'utilisateur au centre de la ville sélectionnée (Metz ou Londres) pour garantir l'utilisabilité immédiate de l'outil.

3.3 Fonctionnalités Avancées

3.3.1 Algorithme "Le plus proche"

Un bouton dédié déclenche un algorithme de calcul de distance (basé sur la méthode géométrique via Leaflet ou Haversine) qui :

1. Compare la position utilisateur avec les coordonnées de chaque parking chargé en mémoire.
2. Identifie la distance la plus faible.
3. Trace automatiquement un itinéraire via `Leaflet Routing Machine`.

3.3.2 Rayon de Recherche

Pour favoriser la mobilité douce, Sparking affiche les arrêts de transports en commun. Un algorithme de filtrage spatial est appliqué : seuls les arrêts situés dans un rayon de **500 mètres** autour du parking sélectionné sont affichés (représentés par des cercles bleus dynamiques), évitant de noyer la carte sous des informations inutiles.

3.4 Internationalisation

L'application supporte nativement le Français et l'Anglais. Le choix de la langue est stocké en variable de session PHP (`$_SESSION['lang']`), ce qui permet de conserver la préférence de l'utilisateur tout au long de sa navigation, même après reconnexion.

4 | Développement Mobile (Android)

4.1 Adaptation de la solution

L'application Android a été générée via **Android Studio**. Elle permet de porter l'expérience Sparking sur les terminaux mobiles natifs. L'architecture de l'APK (disponible dans le dossier `app/build/outputs/apk/debug`) repose sur l'intégration des technologies Web au sein d'un conteneur natif (WebView), assurant une cohérence parfaite entre la version navigateur et la version mobile tout en facilitant la maintenance (une seule base de code principale).

4.2 Ergonomie Mobile

L'interface a été spécifiquement adaptée pour l'usage tactile :

- **Menu Latéral (Burger)** : Permet d'accéder au profil, à l'aide et aux paramètres sans empiéter sur la surface utile de la carte.
- **Popup Responsive** : Les informations des parkings s'affichent dans des modales qui s'adaptent à la largeur de l'écran (mobile portrait).

5 | Bilan et Perspectives

5.1 Difficultés rencontrées

Asynchronisme et Données La gestion du chargement des données de parkings (notamment ceux de Londres) a nécessité une gestion de l'API Overpass pour éviter que la carte ne s'affiche vide au démarrage. Nous avons pensé à mettre dans le readme du github du projet, de dire aux utilisateurs de recharger la carte si jamais les parkings de Londres n'apparaissent pas directement.

Précision GPS La gestion des permissions navigateur pour la géolocalisation a demandé une attention particulière pour offrir une expérience fluide (via le Fallback Metz/Londres) même en cas de refus de l'utilisateur.

5.2 Perspectives d'amélioration (V2)

Le projet Sparking est fonctionnel et répond au cahier des charges MVP (*Minimum Viable Product*), mais plusieurs axes d'évolution sont envisagés pour une V2 :

- **Païement in-app** : Intégrer une passerelle de paiement (ex : Stripe) pour permettre aux utilisateurs de réserver et payer leur place à l'avance.
- **Données Temps Réel Londres** : Connecter l'application à l'API live de *Transport for London* (TfL) pour avoir les places disponibles en temps réel, comme c'est actuellement le cas pour les parkings de Metz.

[a4paper, 12pt]report

[utf8]inputenc [T1]fontenc [french]babel lmodern geometry graphicx [table]xcolor enumitem hyperref titlesec fancyhdr

a4paper, left=2.5cm, right=2.5cm, top=3cm, bottom=3cm

6 | Gestion de Projets en mode Agile/Scrum

Cette partie détaille l'organisation de l'équipe et le suivi temporel du projet, basé sur la méthode Scrum adaptée aux contraintes universitaires.

6.1 Capacité d'effort de l'équipe

L'équipe est composée de **4 membres**. L'unité d'effort utilisée est le *Story Point* (SP), où 1 point correspond à une journée de travail pour une personne.

- **Capacité journalière** : 4 points / jour (1 point \times 4 membres).
- **Sprint 1 (10 jours)** : $10 \times 4 = 40$ points max théoriques (Vélocité réelle : 34 points).
- **Sprint 2 (14 jours)** : $14 \times 4 = 56$ points max théoriques (Vélocité réelle : 56 points).
- **Sprint 3 (4 jours)** : $4 \times 4 = 16$ points max théoriques (Vélocité réelle : 16 points).

6.2 Carnet de produit (Product Backlog)

Le tableau suivant regroupe l'ensemble des User Stories (US) développées au cours du projet, classées par priorité.

lightGrey ID	Intitulé de la User Story	Priorité	SP
US01	Consulter les parkings disponibles à Londres	Haute	3
US03	Connexion à l'API de Londres	Haute	5
US02	Guidage dans une nouvelle ville	Haute	5
US05	Création de compte et connexion	Haute	13
US08	Intégration du JSON dans la base de données	Haute	8
US07	Filtrage des parkings selon les préférences	Moyenne	8
US12	Accès aux services de mobilité à proximité	Moyenne	5
US06	Accès au profil utilisateur	Moyenne	5
US09	Amélioration de l'interface et ergonomie	Basse	5
US11	Modification du profil utilisateur	Basse	3
US10	Restructuration de l'arborescence du code	Basse	13
US14	Renseignement de la motorisation du véhicule	Basse	3
US13	Débogage final	Basse	1
US04	Débogage continu (Sprint 1, 2 et 3)	Basse	8

6.3 Estimation du MVP (Minimum Viable Product)

Le MVP s'est concentré sur les besoins critiques pour la ville de Metz : Géolocalisation, Identification du parking le plus proche et Guidage. L'effort total estimé est de **55 Story Points**.

Socle Technique (8 SP) Configuration de l'environnement, Git, et architecture réseau.

Module Géolocalisation (13 SP) Implémentation du service GPS HTML5.

Consommation API Metz (13 SP) Récupération et traitement des données JSON.

Algorithme de calcul (8 SP) Logique de tri pour trouver la distance minimale.

Interface de guidage (13 SP) IHM pour afficher l'itinéraire.

6.4 Carnets de Sprint (Sprint Backlogs)

6.4.1 Sprint 1 : Expansion Géographique

Durée : 10 jours | **Vélocité** : 34 Points

Objectif

Étendre l'application à de nouvelles villes (Londres) et valider la structure API.

Backlog du Sprint

- **US01 (13 pts)** : Consultation des parkings de Londres.
- **US03 (8 pts)** : Connexion technique à l'API londonienne.
- **US02 (5 pts)** : Guidage dans une zone non-locale.
- **US04 (8 pts)** : Débogage et stabilisation.

Bilan

L'objectif est atteint : l'application est capable de changer de contexte géographique dynamiquement. L'architecture est validée pour l'ajout futur de nouvelles villes.

6.4.2 Sprint 2 : Utilisateurs et Données

Durée : 14 jours | **Vélocité** : 56 Points

Objectif

Structurer l'application techniquement (MVC) et introduire la gestion des utilisateurs (BDD).

Backlog du Sprint

- **US05 (18 pts)** : Système complet d'inscription/connexion.
- **US07 (18 pts)** : Filtrage des parkings (PMR, Électrique...).

- **US08 (8 pts)** : Migration des données JSON vers MySQL.
- **US06 (5 pts)** : Page de profil utilisateur.
- **US09 (5 pts)** : Amélioration UX/UI.
- **US10 (2 pts)** : Refactoring de l'arborescence code.

Bilan

C'est le sprint le plus dense. La mise en place de la base de données permet désormais une persistance des préférences utilisateurs. L'application quitte le stade de prototype statique pour devenir dynamique.

6.4.3 Sprint 3 : Finitions et Mobilité

Durée : 4 jours | **Vélocité** : 16 Points

Objectif

Dernière ligne droite : fonctionnalités "bonus" (favoris, mobilité) et nettoyage du code.

Backlog du Sprint

- **US11 (5 pts)** : Services de mobilité (Bus) à proximité.
- **US13 (4 pts)** : Gestion des favoris.
- **US12 (3 pts)** : Modification du profil.
- **US14 (3 pts)** : Gestion de la motorisation.
- **US15 (1 pt)** : Recette finale.

Bilan

Le sprint, bien que court, a permis d'ajouter la dimension multimodale (Bus) qui enrichit considérablement l'offre de valeur de Sparking.

6.5 Suivi des Sprints (Trello)

Voici l'évolution de notre tableau Kanban au fil des itérations.

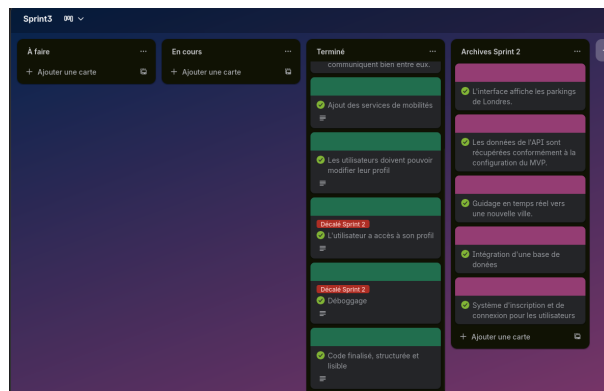


FIGURE 6.1 – Etat du Trello à la fin du Sprint 1

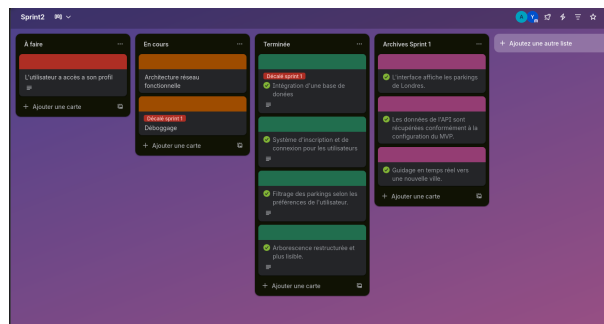


FIGURE 6.2 – Etat du Trello à la fin du Sprint 2

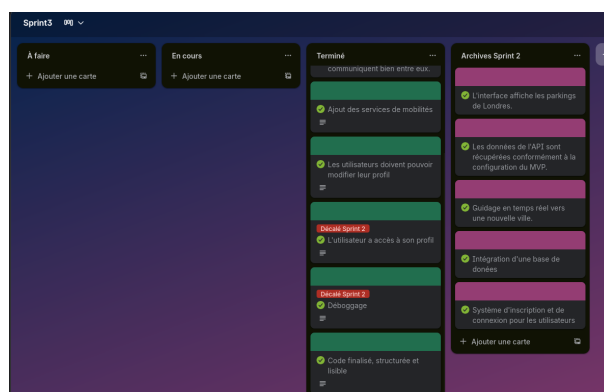


FIGURE 6.3 – Etat du Trello à la fin du Sprint 3

6.6 Livraison Finale

Le Sprint 3 s'est achevé avec succès le **12 janvier**. Toutes les tâches critiques sont terminées. La livraison du produit final (Code source + APK + Documentation) est fixée au **16 janvier 2026**. Le calendrier a été respecté grâce à une vélocité constante et une bonne gestion des imprévus lors du Sprint 2.

Conclusion

La SAE 301 nous a permis de mener un projet complet et transversal, allant de la conception de l'architecture réseau à la réalisation logicielle concrète. **Sparking** répond aux objectifs fixés : c'est une application utile, robuste, sécurisée et agréable à utiliser. Elle démontre notre capacité à intégrer des technologies hétérogènes (Base de Données, Web, Mobile, Réseau) au service d'un besoin utilisateur concret : la simplification de la mobilité urbaine.