



# Livrable 1

---

Cours: GTI525

Session: Hiver 2025

Groupe: 01

Équipe: 11

Remis à: Marcos Dias de Assuncao

## Membres de l'équipe:

- Franck Patrick Assan
  - Monclés Junior Hérissé
  - Anthony Tremblay
- 

## Évaluation de la participation

Nom de l'étudiant	Facteur multiplicatif
Franck Patrick Assan	1
Monclés Junior Hérissé	1
Anthony Tremblay	1

---

# Table des matières

- [Introduction](#)
  - [Architecture logicielle utilisée](#)
    - \* [Diagramme de contexte](#)
    - \* [Diagramme de conteneurs](#)
    - \* [Diagramme de classes](#)
  - [Triage des tableaux](#)
  - [Subdivision des tâches](#)
  - [Conclusion](#)
- 

## Introduction

L'application PédaMap est une application web développée dans le cadre du cours GTI525 à l'hiver 2025. C'est une application qui va permettre aux adeptes de vélo de facilement consulter les itinéraires et les informations sur le réseau cyclable de la belle ville de Montréal. Le projet est divisé en plusieurs livrables. Pour ce premier livrable, nous utiliserons différentes technologies web dans le but de construire la partie statique (présentation) de l'application. Tout d'abord, nous verrons l'architecture que notre équipe a décidé d'utiliser pour développer cette solution. Ensuite, nous exposerons les techniques et outils utilisés pour traiter les données fournies pour ce livrable et, finalement, nous verrons comment les tâches liées au projet ont été assignées aux différents membres de l'équipe.

---

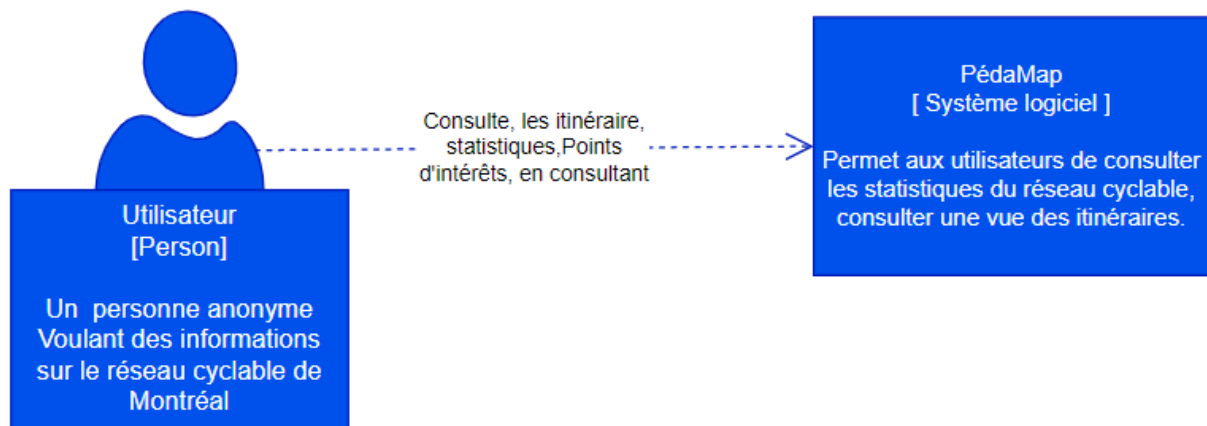
## Architecture logicielle utilisée

Étant donné que nous devons concevoir une application web, nous avons choisi le patron architectural Client-Serveur. Nous avons décidé d'utiliser

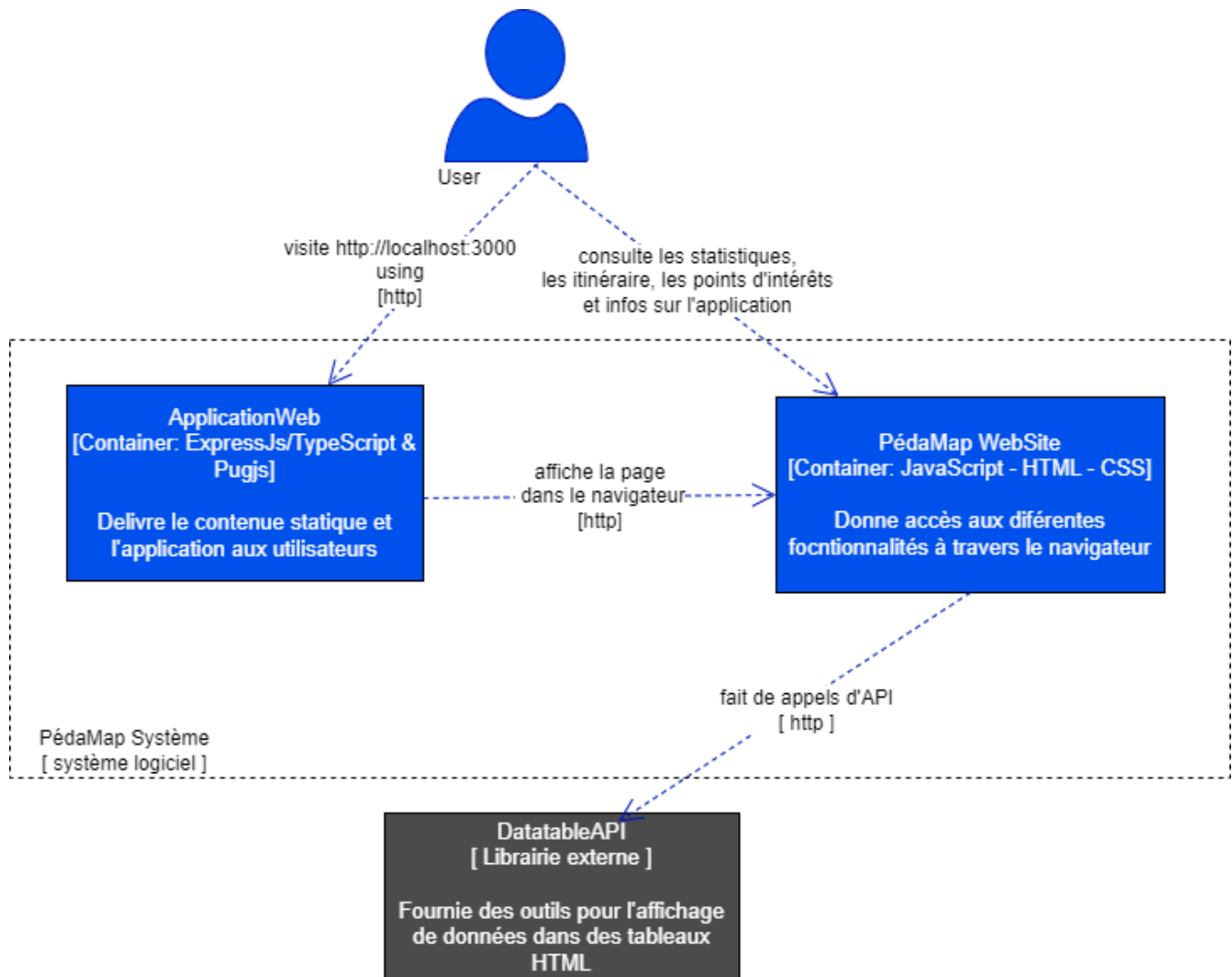
Express.js, un framework minimaliste mais facilement extensible en fonction de nos objectifs. Pour la délivrance du contenu statique de la page, Express.js nous permet de générer le contenu HTML et CSS du côté serveur.

Nous allons voir quelques diagrammes de l'architecture utilisée :

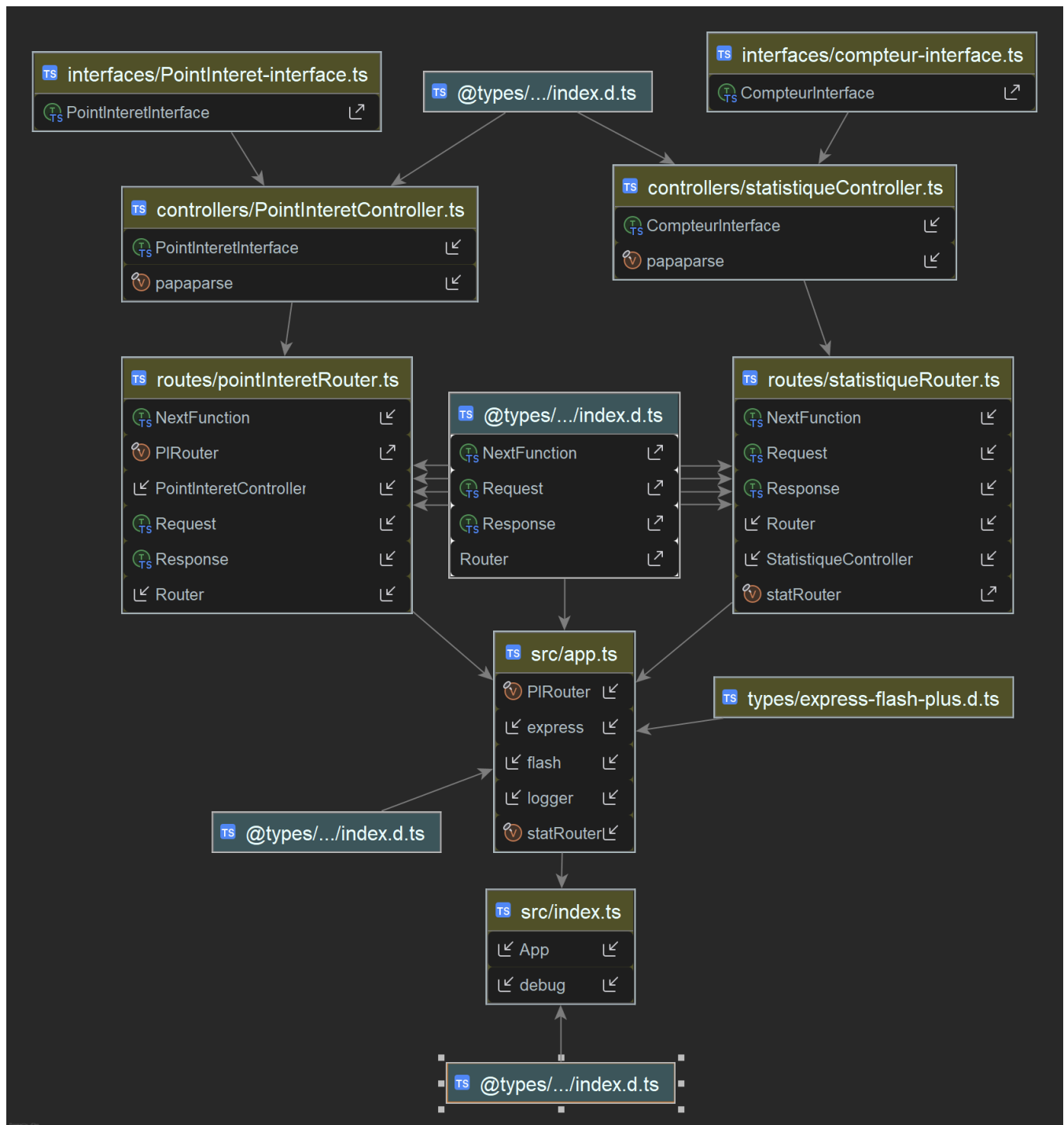
## Diagramme de contexte



# Diagramme de conteneurs



# Diagramme de classes



# Triage des tableaux

Pour le triage des tableaux nous avons utilisé une librairie qui s'occupe de trier les tableaux pour nous.

La librairie s'appelle `dataTable`. Tu peux cliquer sur une colonne et ça va trier la table pour cette colonne.

Tu peux aussi faire une recherche pour trouver une valeur et tu peux faire une pagination pour le tableau

---

## Subdivision des tâches

pour la subdivision des tâches, on a divisé le travail par page. Chacun de nous avait deux ou trois pages à faire.

La page principal et la page d'intérêt a été fait par Moncles Junior Herissé. Ensuite la page des statistiques

et la page des itinéraires a été fait par Franck Patrick Assan. Finalement la page de description de l'équipe et la page du

description du projet a été fait par Anthony Tremblay. Nous avons aussi implementé un côté serveur avec `nodeJs` en `typescript`. La personne qui s'est occupée de créer le côté serveur était Franck Patrick Assan

---

## Conclusion

En conclusion, on a fait les pages statiques de notre application pour le premier livrable. On s'est

occupée d'afficher les tableaux et de les trier. Nous avons utilisé plusieurs technologies pour pouvoir construire

la partie statique du projet. On a aussi pris de l'avance en implémentant une partie serveur. Cette avance va nous aider à faire

le livrable 2 plus facilement.