



Documents de conception

Tentative d'apprentissage des corrélations entre phases de la lune et du soleil et coefficients des marées

Auteurs :

Otmane EL ALOI, Abdelhafid SAOUD, Abdelhadi ZIANE

Encadrant :

Mr. Olivier ROUX

Option :
Informatique

Table des matières

1	Spécification détaillée	2
1.1	Architecture de l'outil de prédiction	2
1.2	Spécification relative au modèle de prédiction	4
1.2.1	Diagramme de classe	5
1.3	Spécification relative à l'interface du client	6
1.3.1	Diagrammes de cas d'utilisation (Rappel)	6
1.3.2	Diagramme de classe de l'interface graphique	7
1.3.3	Maquettes des interfaces	7
1.3.4	Diagramme de séquence	10

Introduction

Dans ce document, nous détaillons les architectures de conception adoptées afin de répondre au problème qui est la prédiction des coefficients de marées.

Nous commençons par définir l'architecture globale du projet en décrivant chacune des couches qui la compose. Nous détaillons ensuite les choix d'outils et les relations entre les classes qui interviennent dans chaque couche sous forme de diagrammes UML de classe. Enfin nous fournissons un diagramme de séquence de la page de prédiction.

1 Spécification détaillée

1.1 Architecture de l'outil de prédiction

Nous adoptons pour notre outil de prédiction des coefficients de marée le design pattern suivant : **Model-Service-Client** (MSC).

Le but de ce modèle est de capturer des principes communs pour construire un environnement de production d'un système ML. En fournissant des lignes directrices pour la séparation des préoccupations.

Il se compose de 3 couches, comme représenté dans la figure qui suit :

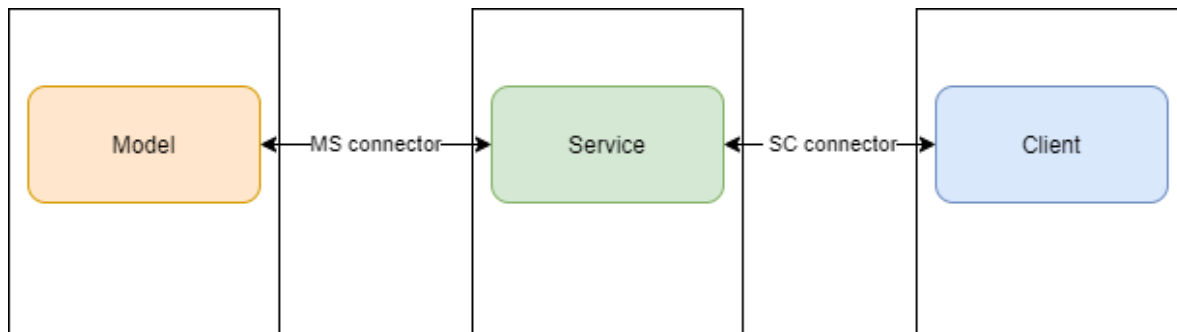


FIGURE 1 – Illustration de l'architecture de l'outil de prédiction

La première couche représente le modèle de machine learning qui sera développé. Et cette phase comporte toutes les étapes de développement d'un modèle de prédiction. Les détails sur les environnements et l'architecture de développement des modèles de prédiction seront fournis dans la section qui suit.

Le schéma ci-dessous détail le contenu de cette couche :

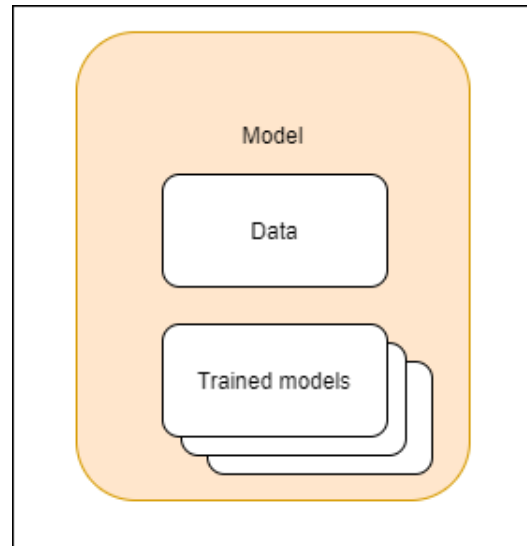


FIGURE 2 – Détail de la couche modèle

La deuxième couche concerne le service de déploiement. Elle fournit les plateformes et les outils pour répondre aux exigences telles que le traitement des requêtes des clients, la génération de réponses et la coordination avec le modèle de prédiction. Elle doit gérer aussi les exigences non-fonctionnelles de type : équilibrage de charge, sécurité... Le schéma ci-dessous détaille le contenu de cette couche :

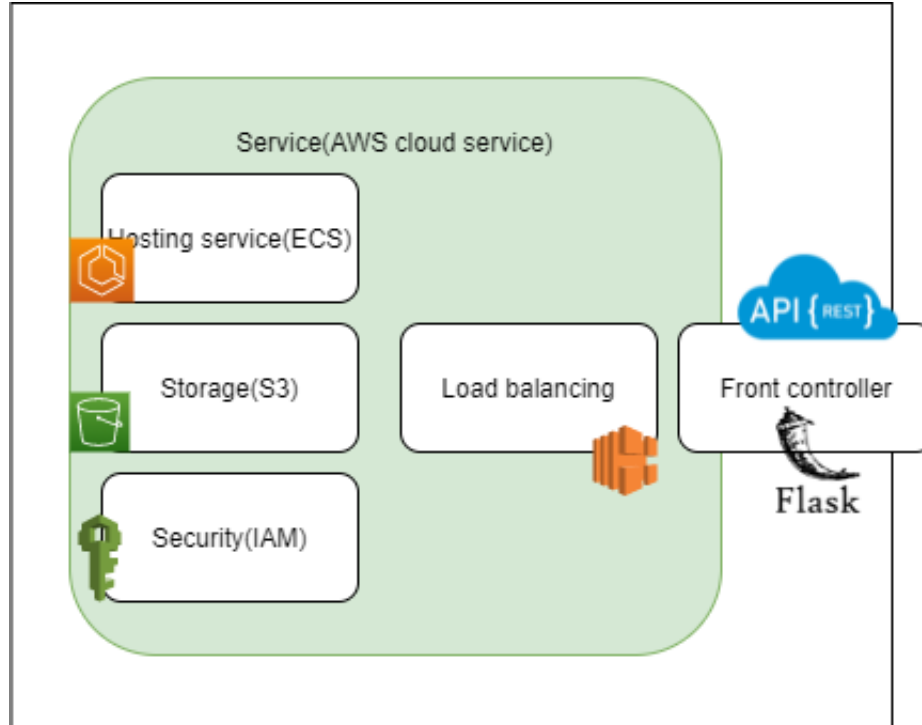


FIGURE 3 – Détail de la couche service

La troisième couche L'interface avec laquelle l'utilisateur interagira avec le sys-

tème. Dans notre conception. Elle sera principalement sous forme d'une page web.

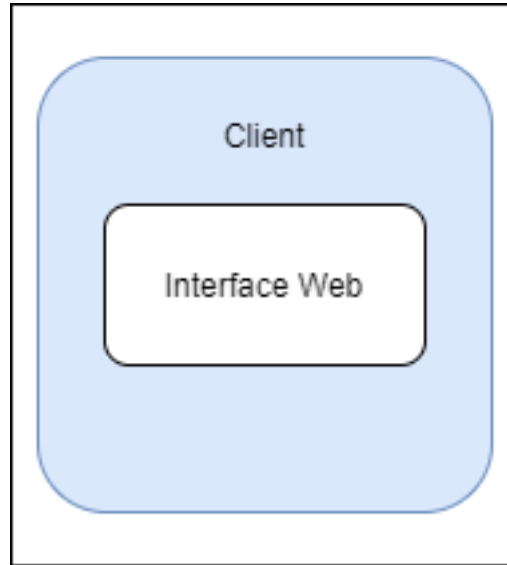


FIGURE 4 – Détail de la couche client

Ces trois couches sont reliées par des interfaces de communication :

- **MS connector** : Cette interface décrit les formats des données échangées entre les deux couches. Dans notre projet, afin de faciliter la réutilisation et l'agilité des algorithmes de prédiction développés, nous considérerons les modèles de prédiction comme des micro-services. Chaque micro-services sera exposée sous une image docker.
- **SC connector** : Cette interface décrit la manière d'exposition du modèle de prédiction par la couche service. Généralement les modèles de machine learning sont exposés sous forme d'une API RESTful. Dans notre projet nous gardons cette même pratique pour l'exposition de nos modèles. Le client doit ainsi invoquer le service en envoyant des requêtes HTTP et obtenir des réponses au format json.

Enfin avec ce paradigme de développement, nous pourrions séparer les responsabilités et améliorer l'efficacité en travaillant en parallèle sur des tâches non dépendantes.

1.2 Spécification relative au modèle de prédiction

Les modèles de prédiction seront développés sous python. Les bibliothèques et framework de développement sont représentées dans la figure suivante :



FIGURE 5 – Bibliothèques et framework de développement

1.2.1 Diagramme de classe

Dans cette partie, nous avons implémenté un diagramme de classe (*Figure 6*) initiale correspondant au modèle de prédiction. Cela en se basant dans un premier temps sur les classes implémentées jusqu'à maintenant, puis sur notre vision par rapport aux modèles que nous allons testés et les classes dont nous aurons besoin au fur et à mesure de l'avancement du projet.

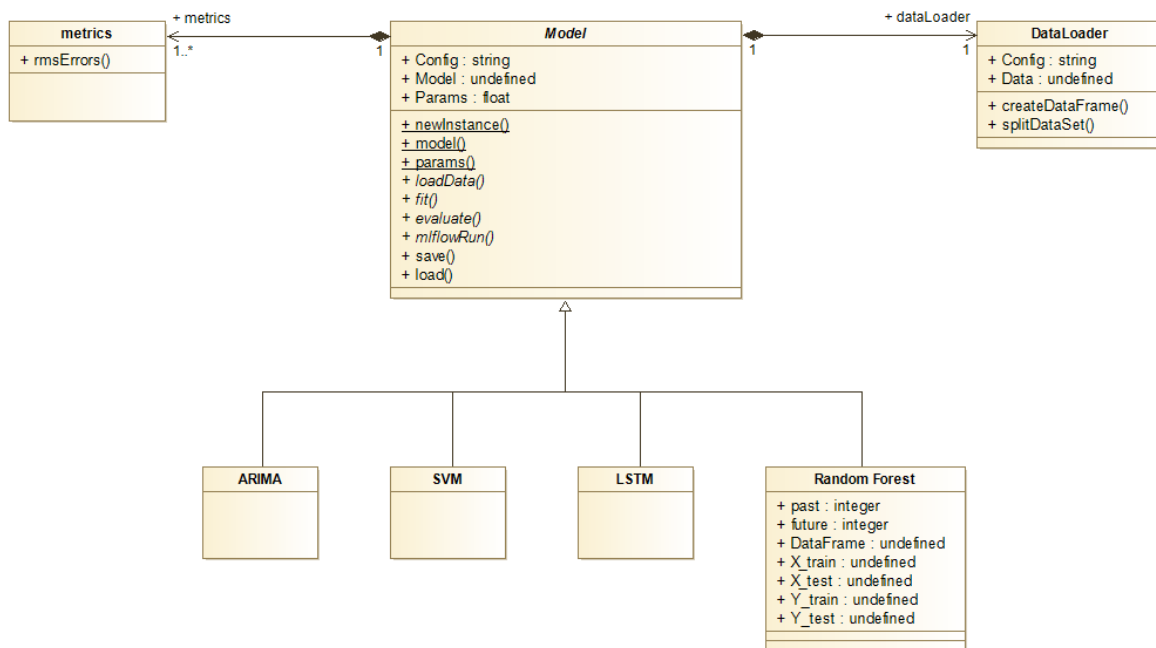


FIGURE 6 – Diagramme de classe du modèle de prédiction

La classe principale de notre modèle est `MODEL`. Elle se compose de deux classes : `DATALOADER` et `METRICS`. Cette classe regroupe les attributs et les méthodes nécessaires pour développer et tester n'importe quel type de modèle de prédiction. De cette super-classe hérite les classes des modèles que nous prévoyons développer pour notre projet et qui sont :

- **ARIMA** : Auto-Regressive Integrated Moving Average
- **SVM** : Support Vector Machine
- **LSTM** : Long Short-Term Memory
- **Random Forest**

`DATALOADER` est une classe regroupant les méthodes permettant la préparation des données selon les inputs/outputs nécessaires pour chaque modèle. Cela en divisant les données importées en séquence de plusieurs échantillons.

La classe `METRICS` a comme objectif surveiller et mesurer les performances du modèle en question (pendant l'entraînement et les tests).

1.3 Spécification relative à l'interface du client

1.3.1 Diagrammes de cas d'utilisation (Rappel)

Le diagramme de cas d'utilisation a pour but de donner une vision globale sur les interfaces de future application. C'est le premier diagramme UML constitué d'un ensemble d'acteurs qui agit sur des cas d'utilisation et qui décrit, sous la forme d'actions et des réactions, le comportement d'un système du point de vue utilisateur.

Acteur : un acteur est un utilisateur qui communique et interagit avec les cas d'utilisation du système. C'est une entité ayant un comportement comme une personne, système ou une entreprise.

Système : cet élément fixe les limites du système en relation avec les acteurs qui l'utilisent (en dehors de système) et les fonctions qu'il doit fournir (à l'intérieur du système). Analyse et spécification des besoins Projet.

Cas d'utilisation : un cas d'utilisation représente un ensemble de séquences d'actions à réaliser par le système et produisant un résultat observable intéressant pour un acteur particulier représenté par des ellipses et limité par un rectangle pour représenter le système.

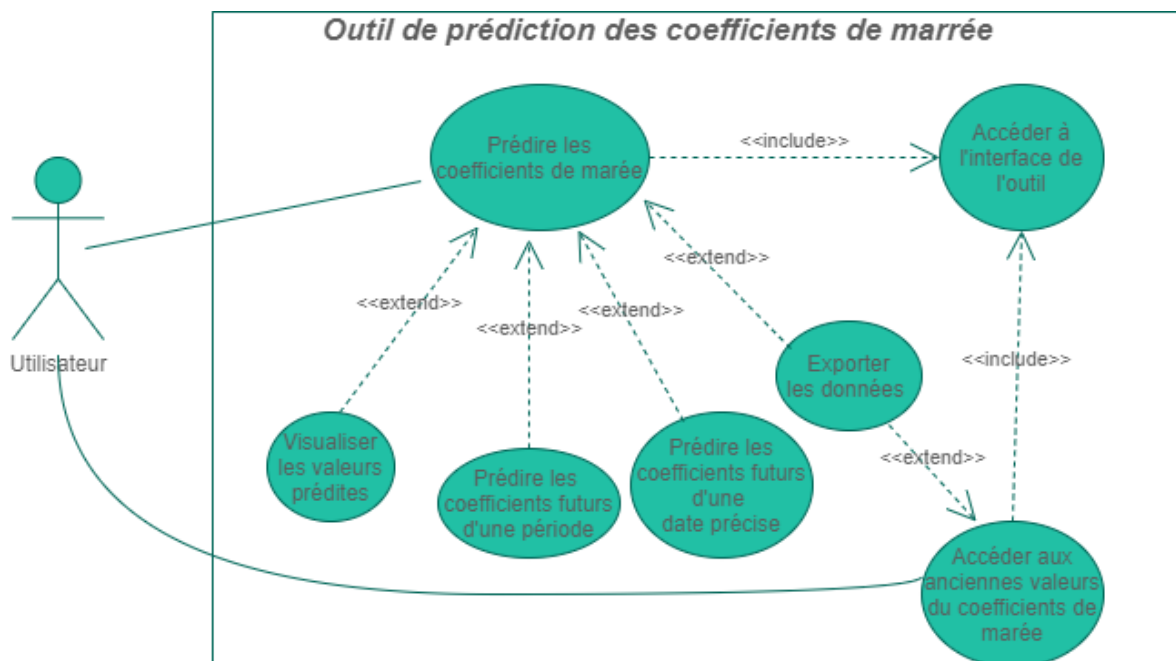


FIGURE 7 – Diagramme de cas d'utilisation

1.3.2 Diagramme de classe de l'interface graphique

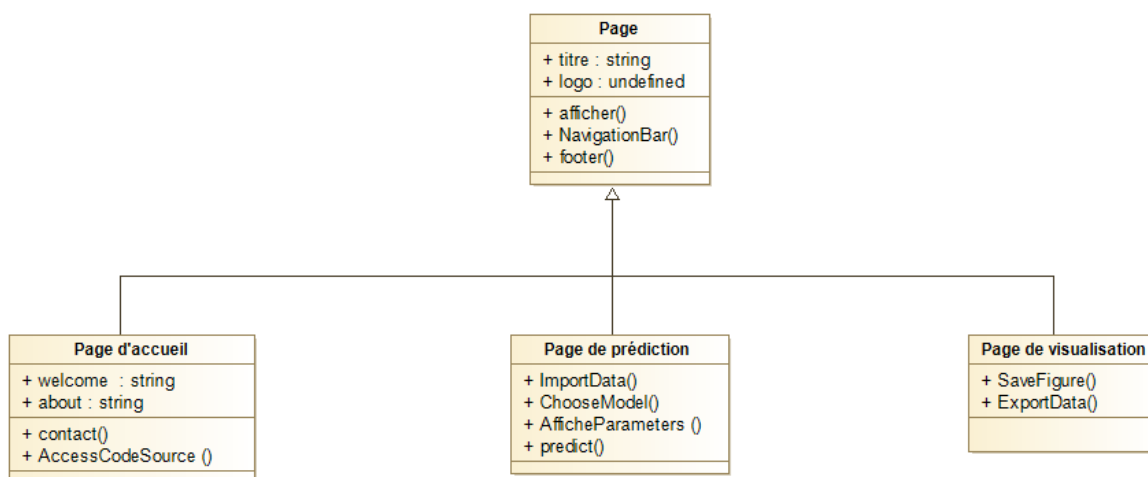


FIGURE 8 – Diagramme de classe de l'interface graphique

1.3.3 Maquettes des interfaces

Pour développer l'interface de notre outil de prédiction, nous allons se baser sur le modèle HTML suivant :

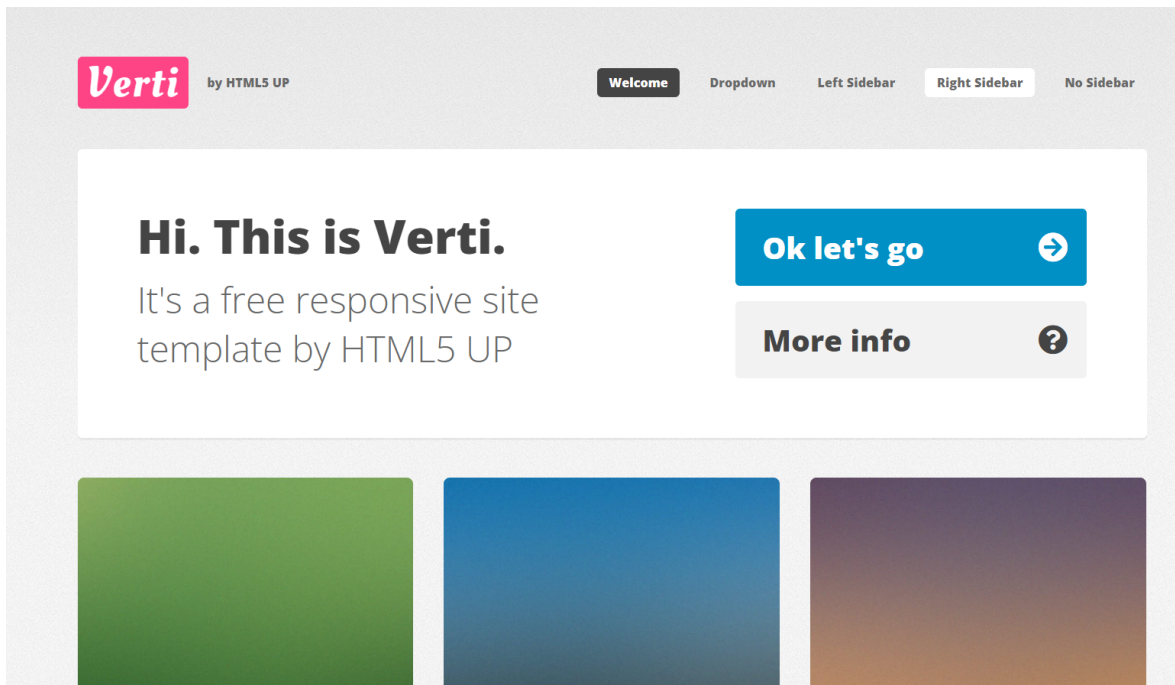


FIGURE 9 – Template HTML5 téléchargée depuis HTML5 UP

Nous allons implémenter les maquettes suivantes au dessus de la template précédente. Nous ajouterons ainsi du HTML, du CSS et du JS au besoin. Pour créer toutes les composantes manquantes et qui sont décrites dans les maquettes de conception ci-dessous.

Page d'accueil L'objectif de cette page est de présenter notre projet et décrire les éléments qui interviennent au cours de l'interaction de l'utilisateur avec le site.

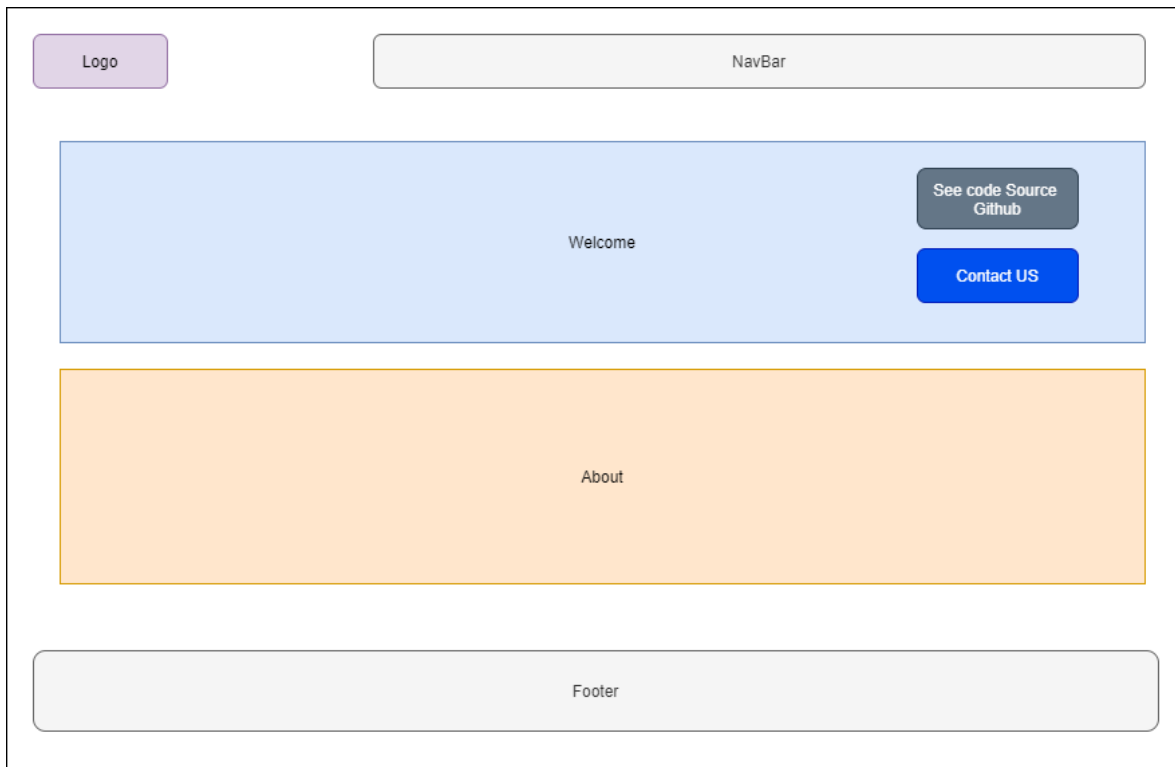


FIGURE 10 – Maquette de la page d'accueil

Page de prédiction pour une semaine Cette page présente une prédiction par défaut des coefficients de marée d'une semaine à venir.

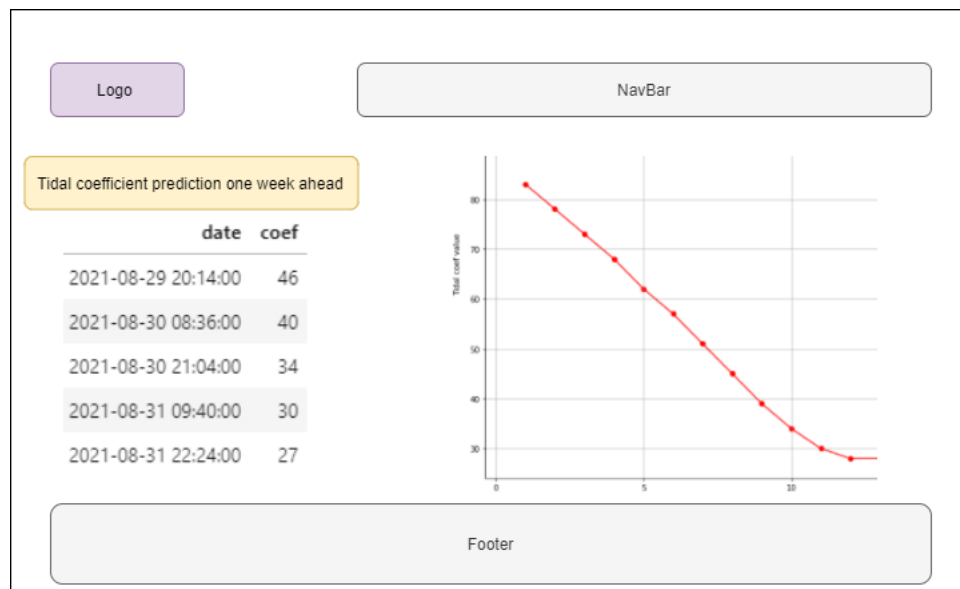


FIGURE 11 – Maquette de la page de prédiction pour une semaine

Api de prédiction Cette page interagit avec l'utilisateur et lui donne le choix de faire plusieurs tâches.

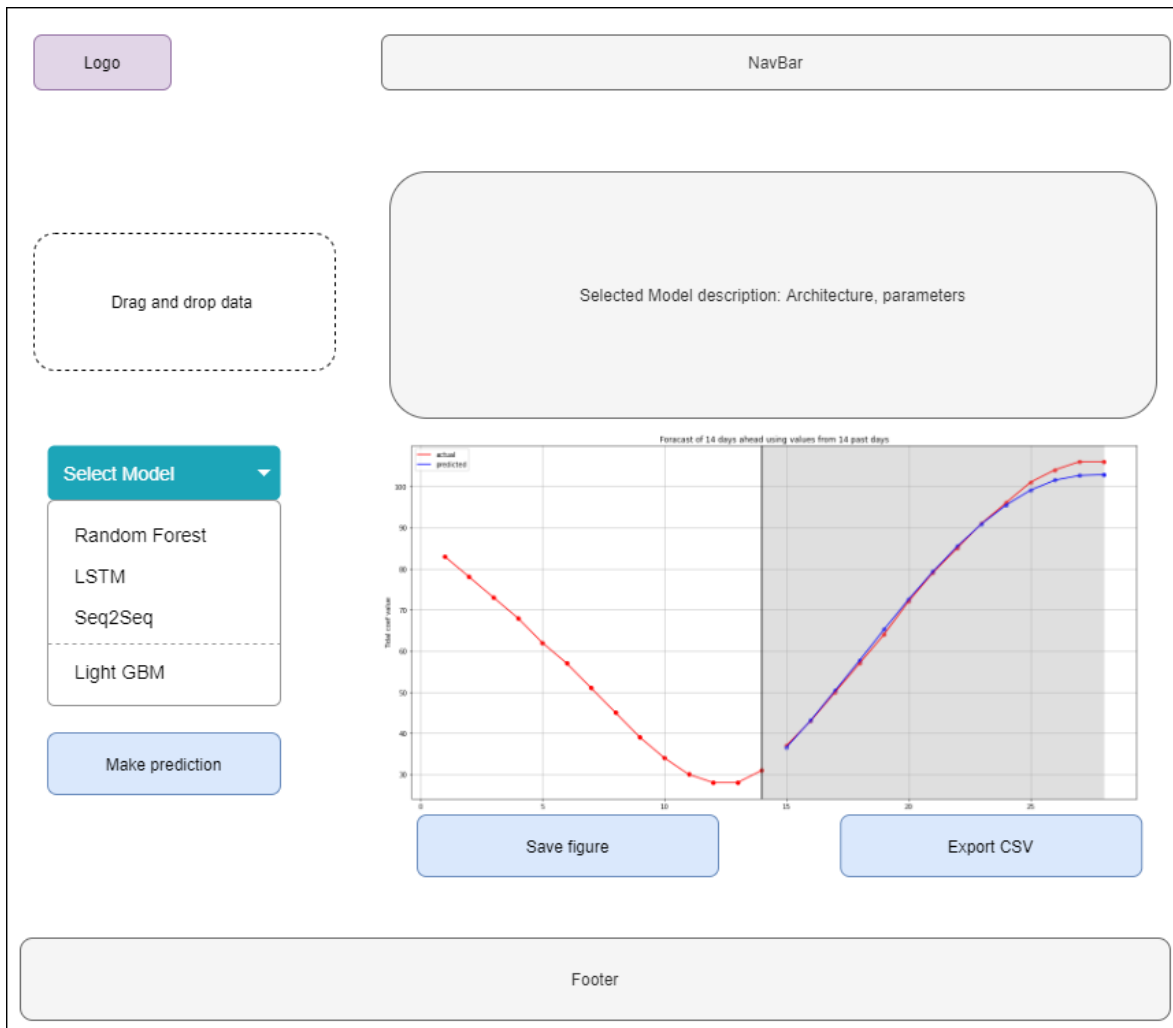


FIGURE 12 – Maquette de la page : api de prédiction

1.3.4 Diagramme de séquence

Les diagrammes de séquence UML sont des diagrammes d'interaction qui détaillent la manière dont les opérations sont exécutées. Ils capturent l'interaction entre les objets dans le contexte d'une collaboration.

Les diagrammes de séquence sont centrés sur le temps et montrent visuellement l'ordre de l'interaction en utilisant l'axe vertical du diagramme pour représenter le temps, les messages envoyés et le moment où ils le sont.

Ci-dessous, nous vous présentons le diagramme de séquence de la page Api de prédiction.

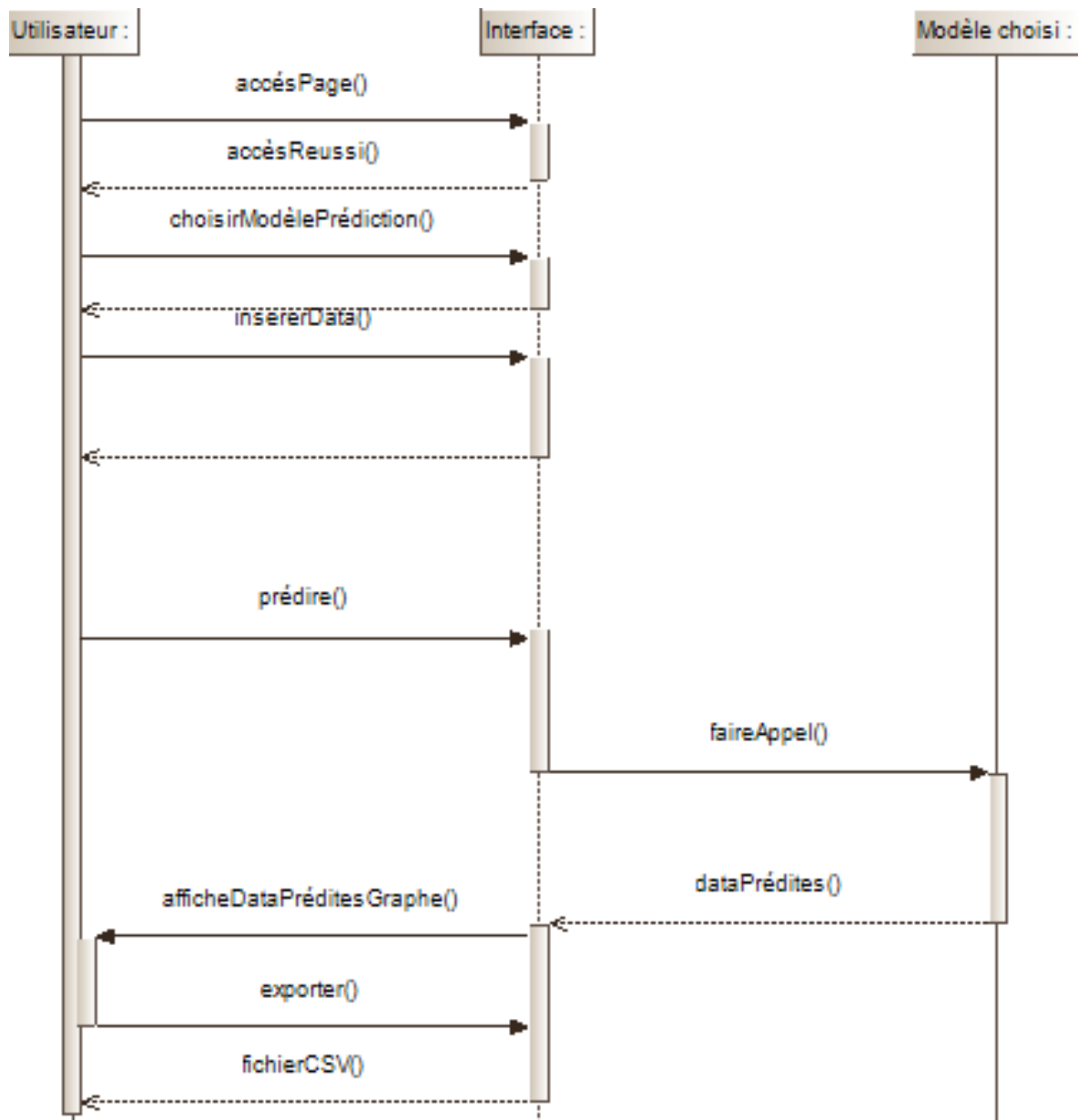


FIGURE 13 – Diagramme de séquence