Bruno Théorêt, Étienne Robert, Nicolas Mathieu, Frank Della Amegnignon, Tom Tran, Joey Mallet

cahier des charges

Sprint 1

Contents

[Introduction 2](#_Toc132359369)

[Choix des technologies 2](#_Toc132359370)

[Backend (serveur) 2](#_Toc132359371)

[Frontend (client) 2](#_Toc132359372)

[CSS Framework 3](#_Toc132359373)

[Schéma de Base de Données 3](#_Toc132359374)

# Introduction

Dans ce sprint 1 nous avons choisi les technologies à utiliser et avons configuré le projet. Nous avons ensuite créé des binômes pour travailler en paire sur des fonctionnalités de l’application. Nous avons défini les épics, tâches et user stories dans JIRA, et assigné un temps d’exécution à chaque tâche en utilisant poker scrum online.

# Choix des technologies

## Backend (serveur)

Nous avons choisi d’utiliser ASP.NET Core pour créer une application Restful API qui se connecte à une base de données Microsoft SQL Server à l’aide d’Entity Framework Core.

Nous avons choisi ASP.NET à la place de SpringBoot / Java parce que nous avons éprouvé des difficultés à utiliser SpringBoot avec le REST API, alors qu’avec un autre cours nous avons constaté la facilité de programmer des API Web avec C# et ASP.NET.

## Frontend (client)

On voulait un client / site web façade qui soit facile à coder et aussi maintenable. Comme notre autre cours avec Hugo est en React, nous avons choisi cette technologie afin de pouvoir transférer les acquis et connaissances à ce projet d’intégration.

On a considéré NextJS (<https://nextjs.org/> ) afin de faciliter le codage des routes (utilisation de « folder based routes » à la place d’utiliser React Router 6), mais on a finalement opté pour React 18 .

Lien : <https://blog.logrocket.com/next-js-vs-react-developer-experience/>

Comme React est une librairie et non un Framework, on a dû choisir les packages / utilitaires qui nous permettraient de faire les tâches nécessaires à la bonne exécution de l’application :

Requêtes réseau : Axios

Navigation : React-Router-Dom@6

Gestion des cookies : React-Cookies

### CSS Framework

Choix de Tailwind CSS à la place de React-Bootstrap.

Pourquoi : on a déjà utilisé React-Bootstrap et on voulait en essayer un nouveau. Tailwind CSS n’inclus pas de composants JavaScript comme Bootstrap Navbar ou Accordéon (Composants UI), mais seulement des classes CSS utilitaires prédéfinies.

Liens :

<https://blog.logrocket.com/comparing-tailwind-css-bootstrap-time-ditch-ui-kits/>

<https://prismic.io/blog/tailwind-vs-bootstrap#final>

# Schéma de Base de Données

Diagram

Description automatically generated

# Projet JIRA

## Backlog 1

Graphical user interface, text, application

Description automatically generated

## Backlog 2

Graphical user interface, text, application

Description automatically generated

# Roadmap

Graphical user interface, application, Teams

Description automatically generated

## Roadmap / Epic / inscription et Authentication

Graphical user interface, application

Description automatically generated

## Tache : création d’un formulaire login

A screenshot of a computer

Description automatically generated