



Toureasy

Étudiants

BRANCATI Silvio, FRACHE Nicolas, MOITRIER Arthur, ZAPP Rémi

Tuteurs

P.A. GUÉNÉGO, I. DEBLED-RENNESON

Année 2020 - 2021



Table des matières

Introduction	2
Objet du document	2
Présentation du projet	2
Présentation de l'équipe	3
Rôle de chaque membre	4
Planning de déroulement du projet	5
Analyse	6
Etude de l'existant	6
Découpage fonctionnel du projet (initialement prévues)	7
Modèles UML utilisés	8
Evolutions par rapport à l'étude préalable de Décembre	11
Réalisation	13
Technologies utilisées	13
Fonctionnalités réalisées	14
Difficultés rencontrées	20
Conclusion	21
Lexique	22

Introduction

Objet du document

Ce document a pour but de retracer l'entièreté de l'avancement de notre projet tutoré — Toureasy. Ce projet s'étend sur le troisième et quatrième semestres du DUT informatique. Il a commencé le 28 septembre 2020 et s'est terminé le vendredi 9 avril 2021. Il aura donc duré 6 mois et 12 jours.

Présentation du projet

Initialement, le sujet du projet proposé par Pierre-André Guénégo avait pour intitulé *Easy Forest*. L'objectif étant de concevoir une application Web permettant à un propriétaire forestier de géolocaliser les arbres qu'il souhaite faire abattre par un bûcheron sans que ceux-ci ne se voient simultanément. Nous avons choisi de d'adapter le sujet initial.

Toureasy se présente sous la forme d'une application web multi-plateforme permettant à tout type d'utilisateur de géolocaliser des monuments (autour d'eux, dans une autre ville) afin de leur faire découvrir de nouveaux monuments avec quelques informations supplémentaires qui peuvent être ajoutées comme leur histoire, leurs particularités...

Ce projet est donc destiné aux personnes ayant un intérêt pour la culture souhaitant notamment découvrir un monument avec quelques précisions. Mais aussi aux vacanciers souhaitant visiter des monuments sur leurs lieux de vacances. Les personnes ciblées sont donc les touristes mais aussi les personnes curieuses en termes de culture.

La particularité de ce projet est que sa base de données sera nourrie par les utilisateurs. En effet, les utilisateurs inscrits peuvent proposer, via un formulaire, d'ajouter un monument avec éventuellement des informations sur celui-ci avec son nom, sa localisation et une ou plusieurs photos afin de nourrir la quantité de monuments recensés sur l'application web.

L'application web se présente principalement sous la forme d'une carte navigable sur laquelle seront présents des marqueurs — chacun correspondant à un monument. L'utilisateur pourra donc naviguer sur la carte et sélectionner par exemple un marqueur afin d'obtenir les différentes informations renseignées, partager ce monument ou encore proposer une modification sur le monument.

L'un des objectifs de ce projet est de le rendre accessible à tous, il doit donc être ergonomique et facile d'utilisation afin que n'importe quel type d'utilisateur puisse utiliser sans aucune difficulté l'application web. C'est pour cela que l'application web sera accessible sans inscription préalable afin de faciliter l'accès. Cependant il sera tout de même possible de s'inscrire. Ce qui débloquent à l'utilisateur des fonctionnalités en plus comme l'ajout de monuments dans ses favoris, l'accès à l'historique des monuments visités auparavant ainsi que la possibilité d'ajouter/modifier des monuments à la base de données.

Présentation de l'équipe

L'équipe de ce projet tutoré est composée de 4 étudiants :

- Rémi Zapp,
- Nicolas Frache,
- Arthur Moitrier,
- Silvio Brancati.

Nous sommes quatre étudiants en seconde année de DUT informatique

Rôle de chaque membre

Rémi Zapp : Rémi s'est principalement occupé de la carte. Il a donc traité l'affichage de la carte sur le site Web et les interactions possibles entre cette dernière et l'utilisateur.

Nicolas Frache : Nicolas s'est occupé de la partie base de données du site Web qu'il a mis en place et met à jour si besoin. Il a configuré en grande partie le serveur VPS qui héberge l'application et a également traité le système d'amis sur le site Web.

Arthur Moitrier : Arthur s'est occupé du site Web. Il a donc mis en place un prototype d'upload d'images. Il a réalisé les maquettes et développé les pages HTML correspondant aux fonctionnalités. Il a également optimisé l'affichage du site Web sur l'ordinateur.

Silvio Brancati : Silvio s'est occupé du site Web. Il a mis en place le système de Modèle Vue et Contrôleur (MVC). Il s'est occupé du système d'identification et des fonctionnalités en lien avec la base de données. Il a également optimisé l'affichage du site Web sur téléphone et sur ordinateur avec entre autres des animations CSS.

Planning de déroulement du projet

Planning initial :

Itération 1 :

- ☐ Maquettes du site web
- ☐ Mise en place de prototypes séparés :
 - ☐ upload de fichiers,
 - ☐ récupération des données GPS,
 - ☐ utilisation de l'API Mapbox,
 - ☐ structuration d'une base de données,
 - ☐ structuration de l'application web

Itération 2 :

- ☐ Fusion des prototypes dans l'application web
- ☐ Optimisation de la carte
- ☐ Ajout de fonctionnalités sur la carte
- ☐ Mise en place d'un système d'identification
- ☐ Développement des fonctionnalités de création de monuments/listes
- ☐ Mise en place d'un système d'amis
- ☐ Début de mise en page CSS
- ☐ Mise à jour de la structure de la base de données suite aux nouvelles fonctionnalités

Itération 3 :

- ☐ Rendre le site web responsive (CSS)
- ☐ Installation du site web sur un serveur
- ☐ Fin de la mise en place du système d'amis
- ☐ Mise en place d'un système de partage de liste/monument
- ☐ Optimisation et ajout de dernières fonctionnalités sur la map

Réalisation :

- ☐ Itération 1 : réalisée comme prévu
- ☐ Itération 2 : report de certaines optimisation de la map non essentiel ainsi que report du début de mise en page CSS à l'itération 3
- ☐ Itération 3 : réalisée comme prévu

Analyse

Étude de l'existant

Lors de notre analyse, nous avons trouvé diverses applications qui se rapprochent de notre projet. Chacune possède tout de même des différences mais le rapprochement se trouve dans le fonctionnement de l'application.

Pokémon Go :

Des pokémons sont situés à certains endroits sur la map du jeu, l'utilisateur se déplace et sa position est modélisée (et actualisée en temps réel) sur une carte avec une boussole indiquant sa direction. (fonctionnalités similaires en termes de géolocalisation).

Système similaire avec les *PokéStop*, lieux indiqués sur la carte, et lorsque l'on est assez proche on peut débloquent des bonus ainsi que voir une photo du lieu-dit. Cependant pas d'informations détaillées sur le lieu-dit (n'étant pas toujours un monument) en question contrairement à notre proposition.

Géocaching :

Un loisir qui consiste à utiliser la technique du géopositionnement par satellite (GPS) pour rechercher ou dissimuler des « caches » ou des « géocaches », dans divers endroits à travers le monde. L'application possède une map avec divers points, un pour savoir où l'on se situe et d'autres pour savoir où se situent les géocaches. On peut donc choisir le géocache que l'on souhaite trouver et s'y rendre à l'aide d'une boussole et de la distance qu'il nous reste à parcourir. Un utilisateur peut donc chercher des géocaches mais aussi en créer. Pour cela, il doit indiquer sa position GPS.

Ainsi, le fonctionnement et la manière d'utiliser l'application sont semblables à notre projet. La seule différence notable est le but de l'application. Ici le géocaching et pour notre projet, le tourisme.

Découpage fonctionnel du projet (initialement prévues)

1 - Gestion compte

- Accès à l'application en mode anonyme sans compte
- Création d'un compte
- Connexion à un compte
- Déconnexion d'un compte / changer de compte
- Modification des informations d'un compte
- Possibilité de rester connecté d'une session à l'autre (cookie d'authentification)
- Suppression d'un compte et de l'ensemble des informations associées. Les contributions apportées par ce compte sont alors anonymisées.

2 - Utilisation normale

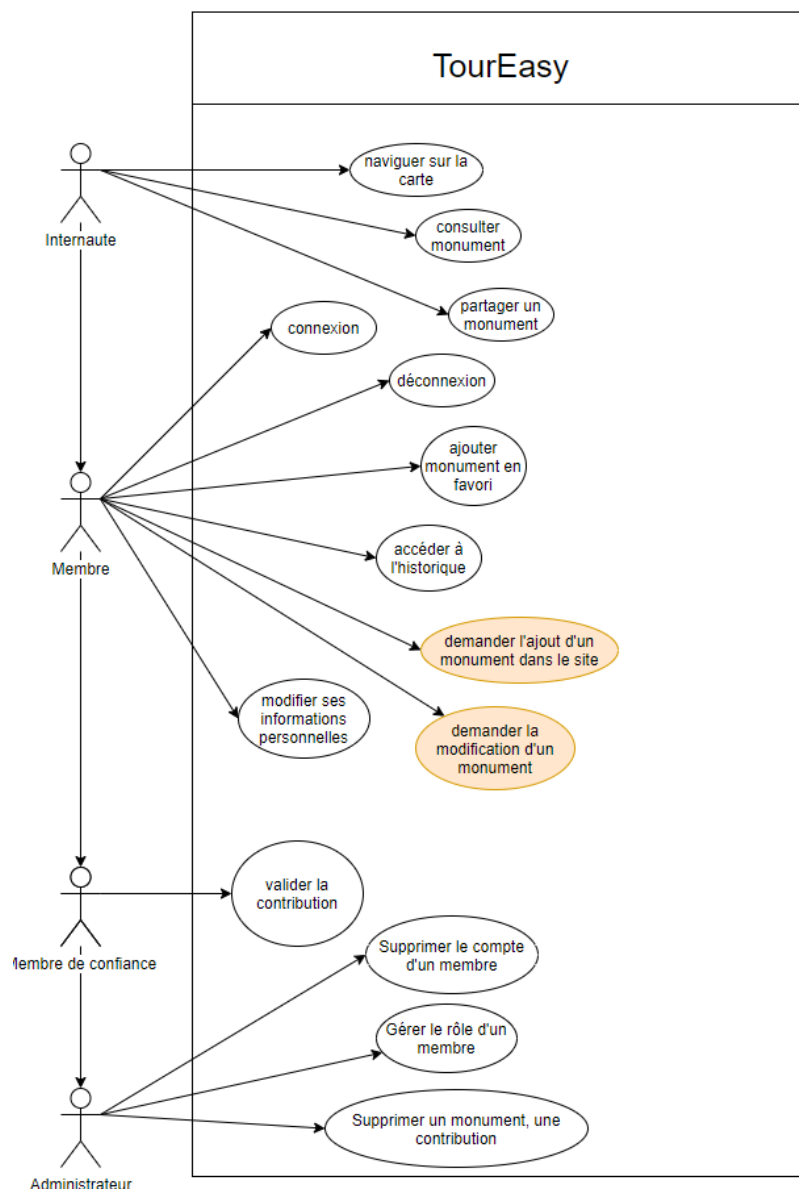
- Affichage de la carte avec implémentation des monuments enregistrés
- Suivi des déplacements de l'utilisateur
- Partager un monument
- Consulter son profil
- Rechercher un monument sur wikipédia (pour plus de détails si disponible)
- Historique des monuments visités, recherchés ajoutés et modifiés
- Consulter les statistiques du nombre de connexions, d'utilisateurs (certifiés) etc...
- (rôle utilisateur lambda) : Se voir attribuer le rôle d'utilisateur de confiance après un certain nombre de contributions.
- (rôle utilisateur de confiance seulement) : Modération des monuments proposés par d'autre utilisateur
- Ajouter/retirer un monument à ses favoris
- Consulter la liste de ses monuments favoris
- Être notifié lorsqu'un de nos monuments favoris est modifié

3 - Ajout ou modification de monument

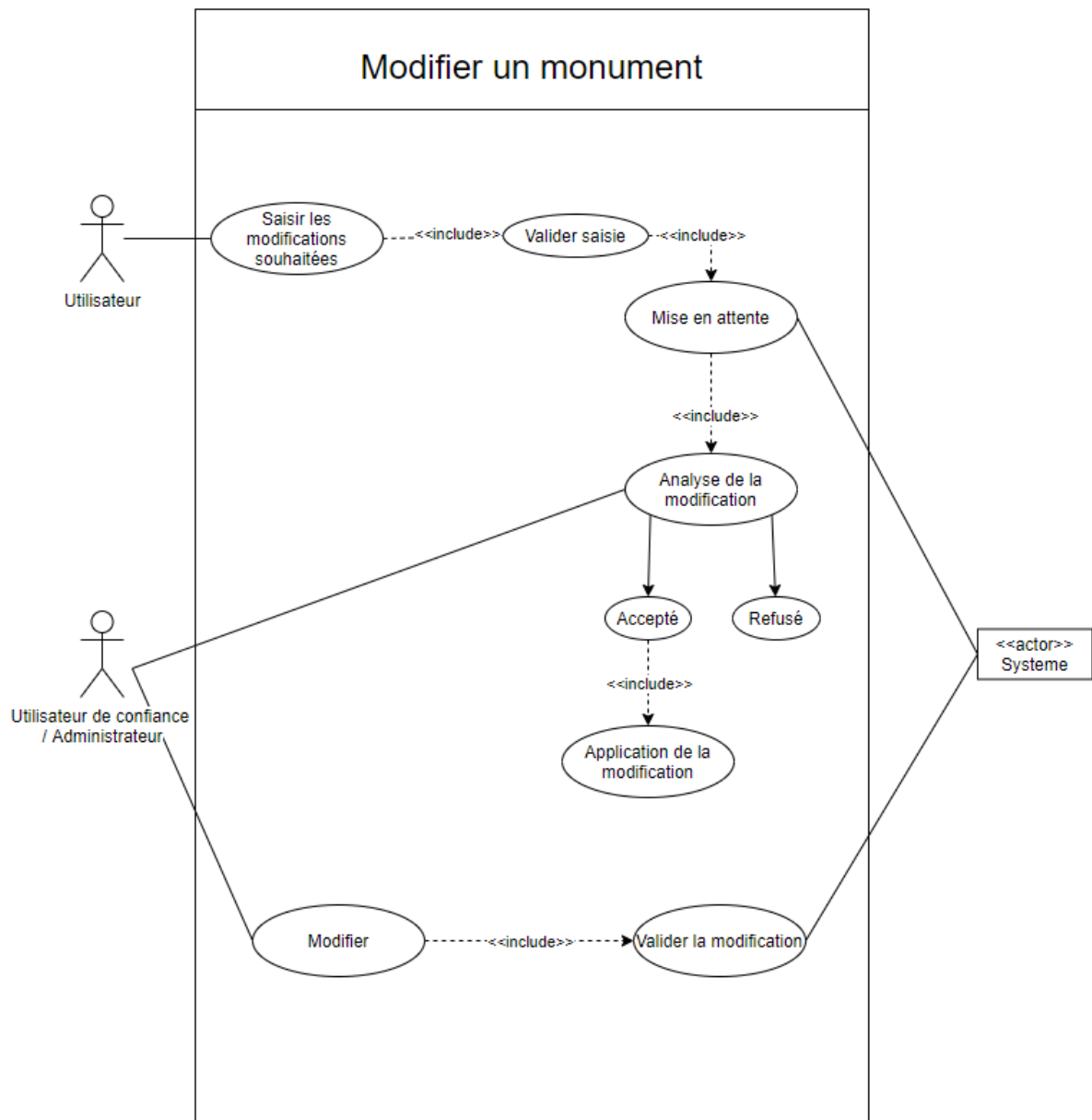
- Ajout d'un monument
- Ajout d'une description
- Ajout d'une ou plusieurs photos (choix entre prendre la photo ou sélectionner dans photothèque)
- Localiser le monument lors de son ajout
- Modification de la description d'un monument déjà existant
- Ajouter une photo à un monument déjà existant

Modèles UML utilisés

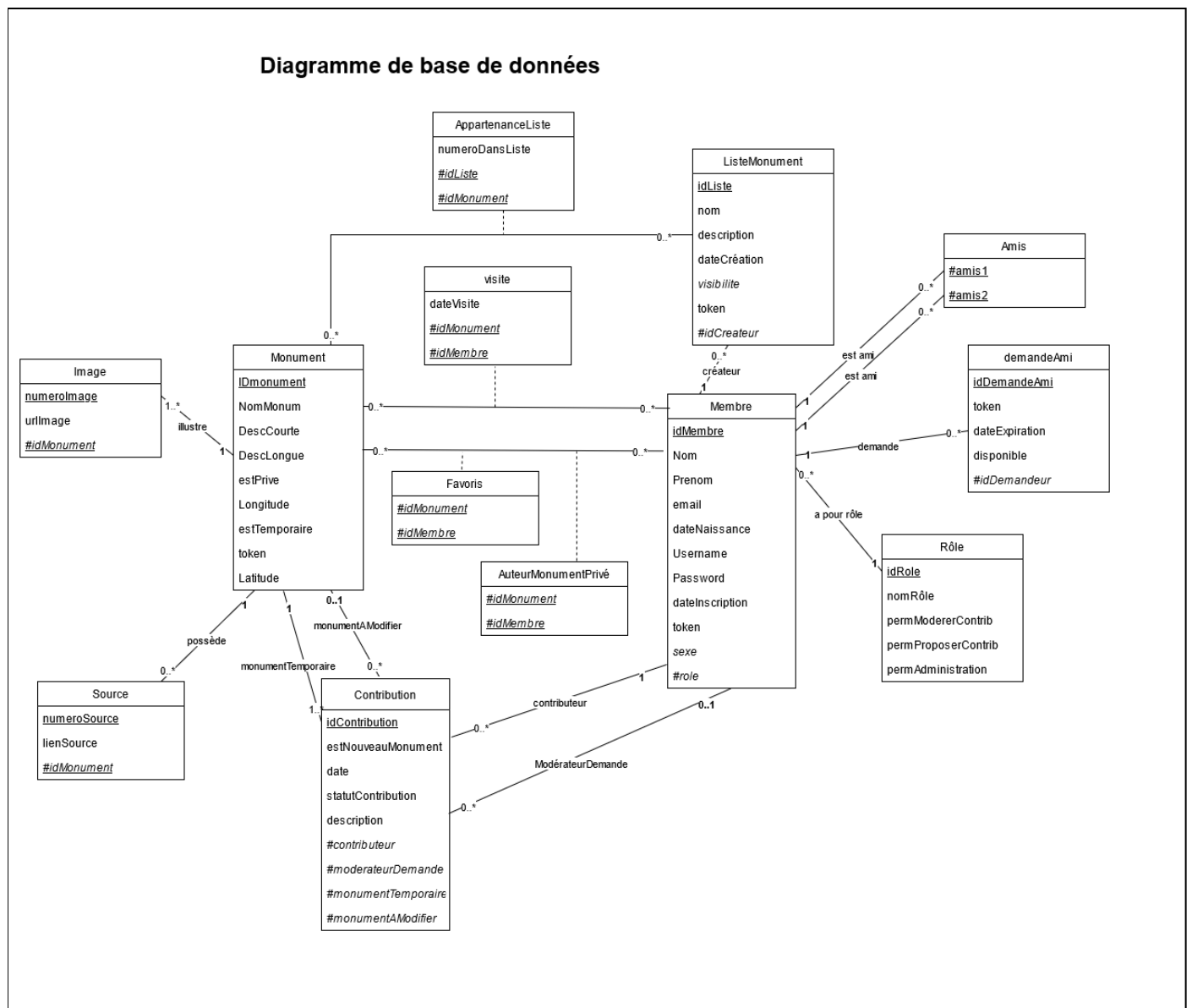
Initialement, Toureasy proposait diverses fonctionnalités orientées sur l'utilisateur. Certaines ont été modifiées pour permettre un usage beaucoup plus simple. Un utilisateur qui arrive pour la première fois sur le site peut naviguer sur la carte, consulter un monument et le partager. Il peut devenir membre en s'inscrivant pour pouvoir accéder à d'autres fonctionnalités comme le fait de pouvoir se connecter et se déconnecter, modifier ses informations personnelles, ajouter un monument aux favoris ou encore d'accéder à son historique et demander un ajout ou proposer une modification d'un monument. Le membre peut évoluer en membre de confiance pour pouvoir valider la proposition. Le membre de confiance peut évoluer en administrateur pour pouvoir valider la proposition.



Il est possible de modifier un monument. Un utilisateur peut saisir les modifications souhaitées qu'il devra ensuite valider pour pouvoir être mis en attente. Cette proposition de modifications sera ensuite analysée par un utilisateur de confiance. Soit elle est acceptée et dans ce cas la modification prend effet, soit elle est refusée. Un membre de confiance peut directement modifier un monument.



La base de données permet d'enregistrer toutes les données nécessaires au bon fonctionnement de l'application. On distingue deux tables principales, *Monument* et *Membre*. Respectivement, elles permettent d'enregistrer les données des monuments et des membres. Ensuite, d'autres tables ont été ajoutées pour permettre de répondre à différentes possibilités comme l'ajout d'un monument en favori ou création d'une liste de monuments...



Itération 3:

Lors de la mise en place de la base de données MySQL¹ sur notre serveur fonctionnant sur Centos 6 nous avons dû modifier certaines parties de script SQL pour des raisons de compatibilité de version avec MySQL. Notamment au niveau des types et des déclencheurs.

Maquettes :

Les maquettes ont été conçues avec la connaissance qu'on avait acquises lors du premier semestre en HTML-CSS. Notre vision était donc vraiment simpliste et pas très attirante.

Deux membres de l'équipe ont donc décidé de chercher un style particulier qui se prêterait bien à l'application. Pour cela, nous nous sommes quelque peu éloignées du style des maquettes originelles.

Réalisation

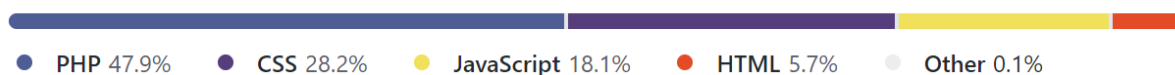
Technologies utilisées

Toureasy n'étant pas un projet Web statique, nous avons fait le choix d'utiliser le langage PHP² afin de pouvoir mettre en place les fonctionnalités désirées. En plus des langages HTML³ et CSS⁴ pour la création des pages web, nous avons aussi utilisé JavaScript⁵ qui nous a servi pour quelques animations ainsi que pour la gestion totale de la map fonctionnant avec ce langage.

En plus de ces langages nous avons décidé d'utiliser des Frameworks⁶ et APIs⁷ afin d'implémenter toutes les fonctionnalités souhaitées. Voici la liste des technologies exploitées :

- ❑ Slim 3 : micro-framework pour la structure de l'application web
- ❑ Mapbox : Complétant OpenStreetMap⁸ formant une API libre de droit d'une qualité et d'une efficacité irréprochable.
- ❑ Eloquent ORM : gestionnaire de relation entre la base de données et l'application web

Langages



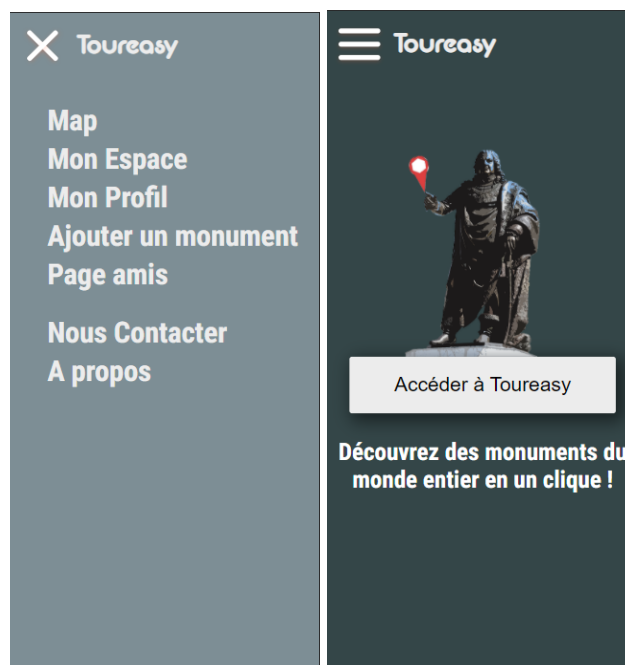
Fonctionnalités réalisées

Différentes fonctionnalités composent le projet. Voici les fonctionnalités qui ont été réalisées :

1 - Gestion compte

- Accès à l'application en mode anonyme sans compte
- Création d'un compte
- Connexion à un compte
- Modification des informations d'un compte
- Possibilité de rester connecté d'une session à l'autre (cookie d'authentification)
- Consulter son profil

Un utilisateur arrivera en premier sur la page d'accueil où il pourra appuyer sur "Accéder à Toureasy" ou utiliser le menu en haut à gauche pour naviguer plus rapidement.



L'utilisateur arrive sur une page de connexion afin de rentrer son code d'authentification s'il en a un. Sinon il peut demander un code en appuyant sur "Obtenir un nouveau code d'identification". Il sera alors sauvegardé dans les cookies afin de se connecter automatiquement. Après ça, il aura accès à la page "Mon Profil" où il pourra changer ses informations personnelles qui seront enregistrées.

The image displays three sequential screenshots of the Toureasy application interface, all featuring a dark teal background and a white hamburger menu icon in the top left corner.

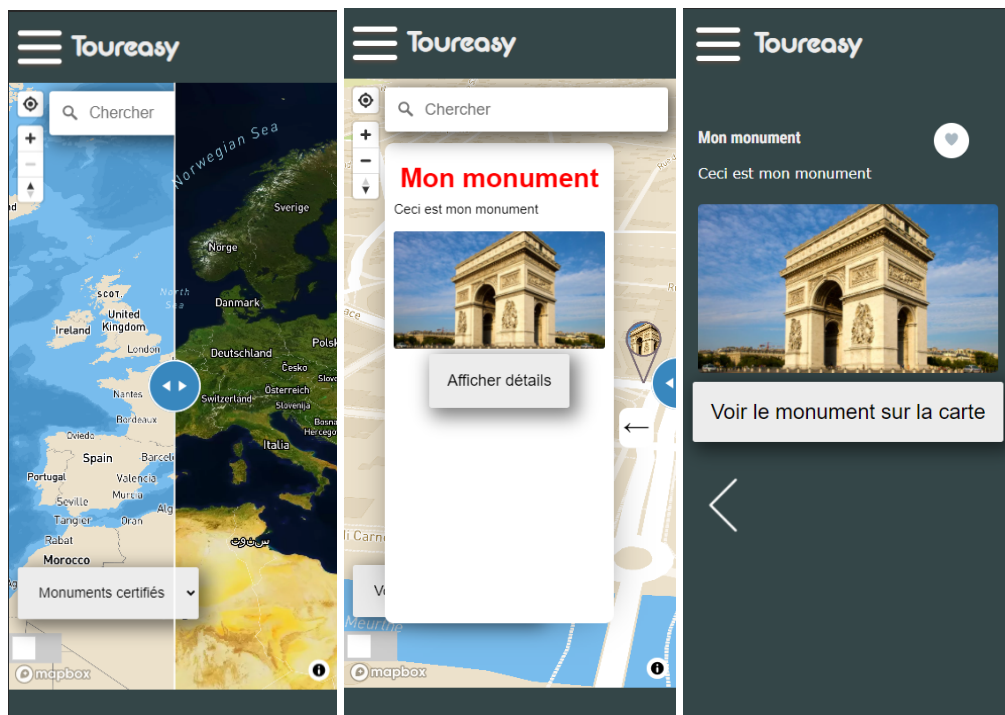
- First Screenshot:** The screen shows the login options. At the top, it says "J'ai un code d'identification" (I have an identification code) above a white input field and an "OK" button. Below this, it says "C'est ma première fois" (It's my first time) above a button labeled "Obtenir un code d'identification" (Get an identification code).
- Second Screenshot:** This screen displays a confirmation message: "Votre code d'identification est 8ea234ac12" (Your identification code is 8ea234ac12), with an "Ok" button centered below it.
- Third Screenshot:** This screen shows the "Mon Profil" (My Profile) section. It contains several form fields: "Prénom" (First Name), "Nom" (Last Name), "Sexe" (Gender) with a dropdown menu currently set to "Non-renseigné" (Not specified), "Date de naissance" (Date of Birth) with a date picker showing "jj / mm / aaaa", and "Code d'identification" (Identification Code) which is pre-filled with "8ea234ac12".

2 - Utilisation normale

- Affichage de la carte avec implémentation des monuments enregistrés (selon des listes privées / contributions / publiques)
- Visualiser la description d'un monument sur la carte
- Visualisation simultanée de la carte satellites et plan
- Possibilité d'un affichage 3D des deux cartes (avec bâtiments en relief)
- Rechercher un lieu
- Page de chargement de la map
- Partager un monument
- Ajouter/retirer un monument à ses favoris
- Consulter la liste de ses monuments favoris

L'utilisateur peut accéder à la carte qui affiche les monuments suivant la liste choisie (Monuments certifiés, Liste privée, contribution). La page "mon espace" permet de gérer les listes et les monuments de l'utilisateur. Sur la carte, on retrouve une barre qui sépare la carte en deux pour avoir une partie plan avec les fonctionnalités comme recherche, centrée la position... et une partie satellite vide qui permet de mieux se repérer. Il est possible d'activer la 3D grâce au bouton en bas à gauche.

Les monuments sont représentés par une punaise. On pourra accéder à son descriptif et sa page. Dans cette dernière, l'utilisateur peut ajouter le monument en favori ou le retirer.



3 - Ajout ou modification de monument

- Ajout d'un monument
- Ajout d'une description
- Ajout d'une ou plusieurs photos (choix entre prendre la photo ou sélectionner dans phototheque)
- Localiser le monument lors de son ajout
- Modification de la description d'un monument déjà existant
- Ajouter une photo à un monument déjà existant

Un utilisateur connecté peut créer un monument en saisissant les informations de ce dernier. La position du monument sera automatiquement enregistrée. L'ajout du monument sera mis en attente afin d'être validé ou non par un administrateur. Un autre utilisateur devra suggérer une modification qui devra être validée ou non.

The image displays two side-by-side screenshots of a mobile application interface for adding or editing a monument. Both screens have a dark blue background.

Left Screen (Initial State):

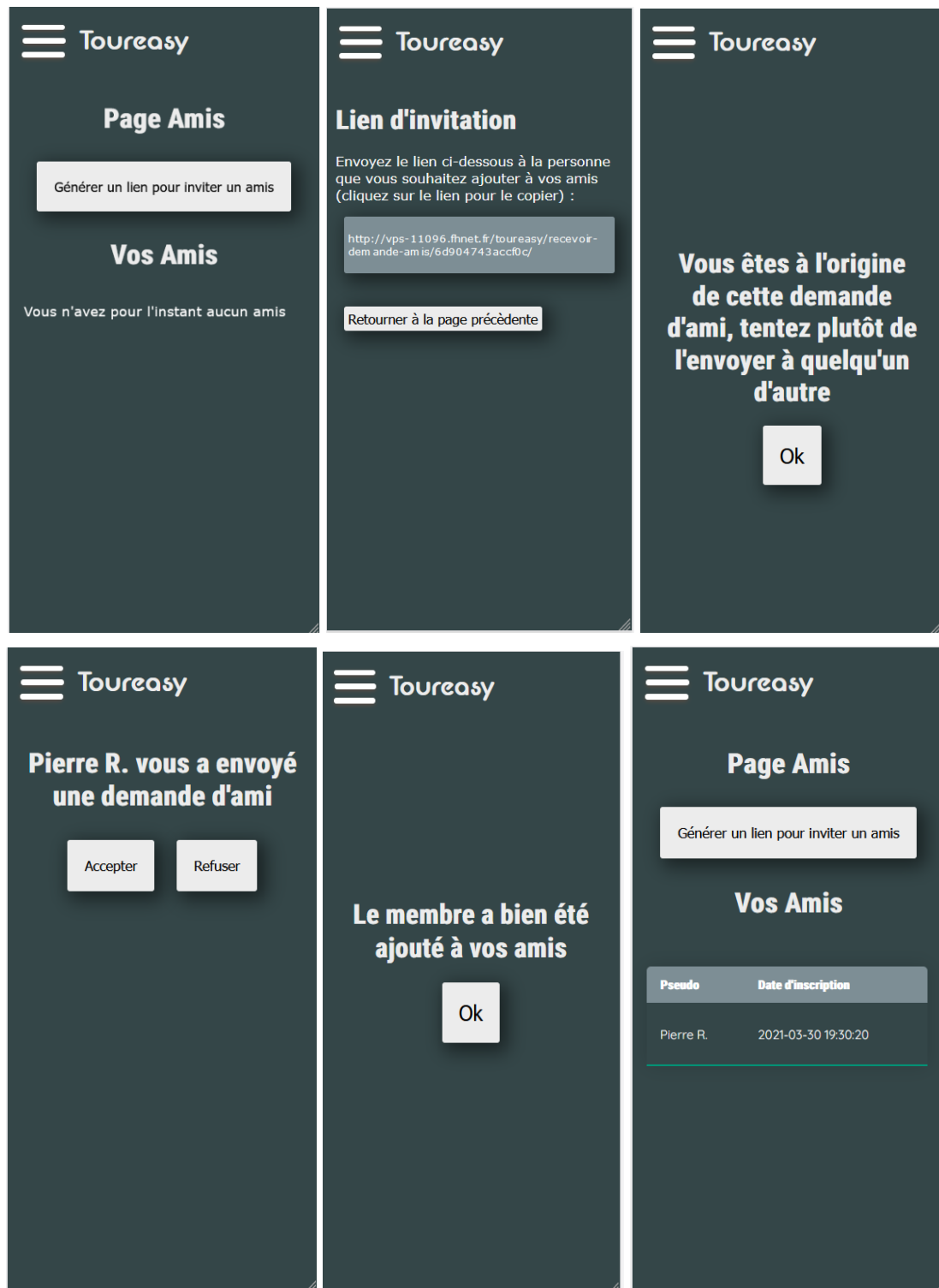
- Nom du monument :** A text input field containing the placeholder text "Mon monument".
- Privé / Public:** Two radio buttons, with "Privé" selected.
- Image Placeholder:** A white icon of a mountain range with a plus sign inside a circle.
- Image Preview:** A small image of the Arc de Triomphe in Paris.
- Description:** A text input field containing the placeholder text "Ceci est mon monument".

Right Screen (After Photo Selection):

- Nom du monument :** A text input field containing the placeholder text "Mon monument".
- Image Placeholder:** A white icon of a mountain range with a plus sign inside a circle.
- Image Preview:** A small image of the Arc de Triomphe in Paris, with a red circle containing a white 'X' in the top right corner, indicating a change or error.
- Description:** A text input field containing the placeholder text "Ceci est mon monument".
- Buttons:** At the bottom, there are two buttons: "Valider" (Validate) and "Annuler" (Cancel).

4 - Système d'amis

- Visualiser la liste de ses amis sous forme de tableaux
- Générer une nouvelle demande d'amis
- Accepter ou refuser une demande d'amis qu'on a reçu sous forme de lien



L'utilisateur peut accéder la page amis par l'intermédiaire du menu latéral. Sur cette page il peut consulter la liste de ses amis et connaître leur date d'inscription. Appuyer sur le bouton en évidence de cette page redirige vers une page avec un lien de demande d'amis. Cliquer dans la zone de texte copie directement le lien dans le presse-papier de l'utilisateur. Si celui-ci tente d'y accéder il verra un message d'erreur le redirigeant (de même si l'utilisateur est déconnecté, déjà ami avec l'émetteur de la demande, ou si la demande d'amis a déjà été utilisée). Un second utilisateur à qui on envoie le lien peut alors accepter ou refuser de devenir ami avec l'émetteur de la demande.

Difficultés rencontrées

Lors de l'avancement de notre projet, beaucoup de difficultés ont été rencontrées mais une s'est démarquée par rapport aux autres :

Carte :

- Leaflet est une API proposant de placer des marqueurs sur une carte. Nous l'avons exploité au maximum et nous avons été très limité au niveau des fonctionnalités (enlever des icônes par défaut sur la carte par exemple) donc nous sommes reparti de 0 et nous avons choisi l'API Mapbox qui propose énormément de fonctionnalité personnalisable — ainsi que des tutoriels officiel pour nous apprendre à utiliser l'API.
- Le problème des monuments qui se superposent : Après nombres recherches, nous avons trouvé une solution. Ce sont des *clusters* (des groupes) qui permettent d'afficher *un groupe* à la place de plusieurs monuments lorsqu'ils sont rapprochés sur la carte. Avec ce système nous avons pu ajouter des animations différentes lorsqu'on clique dessus.
- Afficher l'image du monument dans un rond afin de le placer sur un pointeur sur la carte. Rogner une image selon sa taille, selon son format puis ajouter des traits de couleur pour simuler un effet de pointeur de lieu. Ceci fut vraiment compliqué. Deux choix s'offraient à nous : utiliser une API de dessin (la plupart en Node JS⁹ et trop compliqué) ou utiliser en brut la balise canvas HTML (solution choisie).
- Le problème le plus récent était d'animer les curseurs des monuments lorsqu'on clique dessus (afin de l'agrandir par exemple). Cela n'a pas pu être fait. Les monuments qui sont affichés sur la carte ne sont pas accessibles en CSS et l'API Mapbox ne propose pas des fonctions d'animation, il fallait donc passer par des paramètres JSON¹⁰ à placer à chaque création de monument. Nos essais s'étant révélés infructueux nous avons finalement abandonné bien qu'une solution aurait certainement était d'utiliser le langage SVG¹¹.

Conclusion

Le sujet que nous avons initialement choisi — EasyForest — était déjà relativement complet, alors notre décision de le modifier pour l'appliquer au secteur du tourisme n'a que renforcé cet aspect. En effet, au-delà d'un noyau imposé nous avons une grande liberté à la fois dans les fonctionnalités à mettre en place et dans les technologies à utiliser pour y parvenir.

Le projet Toureasy aura donc permis à chacun de nous de se spécialiser dans un domaine particulier, nous aurons eu la possibilité d'approfondir certaines compétences bien au-delà de ce que nous avons vu jusqu'ici durant notre cursus.

Toureasy fut notre première expérience de travail sur un projet de cette envergure. Il nous a permis de prendre la mesure de l'importance des outils de travail collaboratifs tels que Git ou Trello qui ont facilité grandement l'organisation entre les membres de notre équipe. En comparant la liste initiale de fonctionnalités prévues et celles implémentées nous avons observé notre difficulté de prévoir le temps de réalisation du projet, et en particulier nous avons eu tendance à surestimer le nombre de fonctionnalités que nous pouvions faire par itération.

En l'état actuel notre application est parfaitement fonctionnelle mais beaucoup de fonctionnalités pourraient encore être ajoutées pour la compléter. C'est dans cette optique que nous avons conçu et documenté notre projet afin qu'une autre équipe qui déciderait de travailler dessus puisse le faire sans difficulté.

Enfin nous aimerions remercier Pierre-André Guénégo qui a accepté que nous remaniions le sujet initial afin de l'adapter au secteur du tourisme et de rendre son usage destiné au grand public. Il a suivi notre avancée et a répondu à nos questions avec une grande disponibilité. Nous remercions également Isabelle Debled-Rennesson pour le suivi hebdomadaire de notre projet.

1. **MySQL** : MySQL est un système de gestion de bases de données relationnelles. Il fait partie des logiciels de gestion de base de données les utilisés mondialement. Ses principaux concurrents sont Oracle, PostgreSQL et Microsoft SQL Server.
2. **HTML** : L'HyperText Markup Language, HTML, désigne un type de langage informatique descriptif. Il s'agit plus précisément d'un format de données utilisé dans l'univers d'Internet pour la mise en forme des pages Web.
3. **CSS** : Cascading Style Sheets (CSS) est un langage de feuille de style utilisé pour décrire la présentation d'un document écrit en HTML ou en XML (on inclut ici les langages basés sur XML comme SVG ou XHTML). CSS décrit la façon dont les éléments doivent être affichés à l'écran, sur du papier, en vocalisation, ou sur d'autres supports.
4. **Javascript (JS)** : JavaScript (qui est souvent abrégé en « JS ») est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web.
5. **PHP** : Le PHP, pour Hypertext Preprocessor, désigne un langage informatique, ou un langage de script, utilisé principalement pour la conception de sites web dynamiques.
6. **API** : Une API est un ensemble de définitions et de protocoles qui facilite la création et l'intégration de logiciels d'applications. API est un acronyme anglais qui signifie « Application Programming Interface », que l'on traduit par interface de programmation d'application.
7. **Framework** : Un Framework (ou infrastructure logicielle en français) désigne en programmation informatique un ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application.
8. **OpenStreetMap** : OpenStreetMap est un projet collaboratif de cartographie en ligne qui vise à constituer une base de données géographiques libres du monde, en utilisant le système GPS et d'autres données libres.
9. **Node JS** : Node.js est un environnement d'exécution JavaScript open-source et inter-plateforme. C'est un outil populaire pour pratiquement tous les types de projets.

10. **JSON** : JSON (*JavaScript Object Notation*) est un langage léger d'échange de données textuelles. Pour les ordinateurs, ce format se génère et s'analyse facilement. ... JSON permet de représenter des données structurées (comme XML par exemple).
11. **SVG** : Le format de fichier SVG (Scalable Vector Graphics) est le format de dessin vectoriel pris en charge par le logiciel MediaWiki, moteur de la Wikipédia. Il est basé sur XML et permet la création d'images sous forme de graphiques vectoriels.