

11/12/2024

NotaDomus

Bilan de projet

par TeapotStudio

S3.01: Création d'une application web pour valoriser le patrimoine culturel français

Equipe 16 :

Emilie DUBIEF, Flavien RIONDET,
Aline ROSTAGNAT, Maxime RASTELLI,
Eddy FRANCOU, Nils RAYOT, Ali KUŞ

Sommaire

Retour sur le projet	1
Contexte	1
Objectifs	1
Suivi des risques	1
Qualité du projet	2
Mesures organisationnelles	2
Mesures techniques	3
Synthèse technique	3
Modélisation	3
Nos choix techniques	4
Bilan Technique	5
Bilan général	5
Annexe	5
Readme	5
Cahier de tests	7
Ballade cognitive	7
Tests unitaires	7
Tests utilisateurs - Think Aloud	7
Traitements ETL	10
Structure du code	11
Sécurité	12
Pour les Serveurs	12
Pour l'Application	12
Autres documents	12
Formulaire de consentement	12
Tâches des participants	13

Retour sur le projet

Contexte

En tant qu'équipe attachée à la richesse du patrimoine français, nous avons constaté, grâce à un questionnaire d'étude réalisé en amont, que le label *Maison d'illustres* n'était pas suffisamment reconnu à sa juste valeur. C'est pourquoi nous avons décidé, en groupe de sept personnes, de développer une application web dédiée à la valorisation du patrimoine culturel de la France. Après avoir défini le cadrage de notre projet et les détails de la conception, nous avons entamé la phase de développement dont les détails sont exposés dans ce document.

Pour illustrer ce rapport vous pouvez accéder à notre application avec le lien suivant: <https://9173-193-55-51-9.ngrok-free.app/>

Objectifs

Nos objectifs pour ce projet étaient de proposer une application web accessible à tous sous forme de carte interactive avec un côté social afin de mettre en avant les maisons d'illustre françaises.

Pour la carte interactive, nous voulions pouvoir afficher les différentes maisons d'illustres et les filtrer en fonction de différents paramètres. Ces paramètres sont les favoris de l'utilisateur, l'époque de l'illustre, la localisation de la maison (région et département) et la note de la maison. Nous souhaitons également rechercher les maisons selon une œuvre d'un illustre, un nom d'illustre ou bien l'adresse.

Pour le côté social, nous souhaitons mettre en place la possibilité d'avoir des amis et de pouvoir leur recommander des maisons d'illustres. De plus, nous voulions que les utilisateurs puissent poster des commentaires. Nous voulions également proposer un site respectueux grâce à la possibilité pour les utilisateurs de signaler des commentaires, et la présence d'un modérateur qui prendra connaissance des signalement et supprimera les commentaires problématiques.

Nous voulions aussi permettre aux utilisateurs d'être acteur du patrimoine en leur permettant de proposer des lieux qui pourraient devenir des maisons d'illustres. Pour ce faire nous voulions ajouter un formulaire qui envoie un mail à une adresse mail appartenant à l'application. Cela permettrait à un responsable de l'application de valider ou non les propositions afin de les transmettre au label *Maison d'illustre*.

Pour l'interface, notre objectif est de proposer quelque chose de clair et de compréhensible, accessible à tous les publics. Cela s'illustre par des pop-up clair et des messages concis à chaque action réalisée par l'utilisateur.

Suivi des risques

Dès la phase de cadrage du projet, nous avons repéré plusieurs contraintes et en avons extrait les risques associés. Certains se sont réalisés, mais la plupart ont eu moins d'impact que ce qui était prévu.

Concernant le choix des technologies utilisées, il s'est révélé suffisant pour notre projet. Nous aurions cependant pu gagner du temps en utilisant des frameworks. Malgré ces ralentissements, les fonctionnalités les plus importantes ont été implémentées,

contrairement à celles dont la priorité était plus faible. Nous ne pouvons donc pas dire qu'il y a eu un mauvais choix de technologie.

Concernant les salles de travail, nous avons la plupart du temps une salle adaptée à nos besoins. Il est arrivé une fois seulement que nous devions travailler dans une salle sans aucun ordinateur. Certains d'entre nous n'ayant pas amené leur ordinateur personnel, nous en avons profité pour planifier certains aspects de notre projet comme par exemple ce qui restait à faire. Nous avons donc réussi à transformer ce problème en élément bénéfique pour le projet.

Afin de pouvoir observer en temps réel la mise à jour de l'application, l'utilisation de la base de données nous contraignait à faire de nombreux commit sur notre serveur puisque le serveur public_html bloquait l'accès à toute base de données.

Certains d'entre nous ont dû prendre en main l'IDE PhpStorm afin de pouvoir travailler en équipe efficacement à l'aide de la fonctionnalité "Code with me". Pour finir, les tâches ont été réparties afin d'utiliser au mieux les spécialités de chacun.

Qualité du projet

Mesures organisationnelles

Pour les deux premières phases de notre projet (cadrage et conception), nous avons mis en place un Trello afin de bien visualiser l'avancée du projet. Nous devions régulièrement mettre à jour les tâches réalisées. Cette méthodologie a été très pratique puisqu'il se passait plusieurs jours entre les séances du projet, le trello était donc utile pour se rappeler de l'avancée des étapes.

Pour la phase de développement, nous n'avons pas utilisé trello, nous avons beaucoup communiqué directement pour suivre l'avancée des différentes tâches. Le logiciel qui a été le plus utile pour cette étape a été GitLab, nous nous sommes assurés dès le début du développement que tout le monde savait l'utiliser. Nous avons également mis en place la rédaction d'un journal de bord qui a été rempli chaque jour à 17h (11H30 les jeudis). Ce journal nous a permis de bien visualiser l'avancée du projet mais aussi celle personnelle pour chaque membre du groupe.

Une fois cela fait nous avons constitué plusieurs équipes, une pour le modèle, une pour les vues et une pour la finalisation du serveur et de la base de données. La communication concernant l'avancée des équipes se faisait régulièrement entre les plages du matin et de l'après-midi. L'équipe du serveur et de la base de données a fini ses tâches avant les autres et a donc rejoint l'équipe du modèle. Une fois ce dernier terminé, il a fallu le tester avant de l'utiliser dans les contrôleurs.

Une fois les vues et le modèle terminés, nous avons réalisé des back-up des vues afin d'avoir des ressources si les contrôleurs nous posaient problème. Ensuite, toute l'équipe a entamé le développement des contrôleurs. Pour ce faire nous devions communiquer le plus possible afin de définir les tâches de chacun. Nous avons créé des sessions avec la fonction code-with-me de PhpStorm pour chaque modules à développer.

Nous avons ensuite réalisé des tests utilisateur de notre application, par manque de temps à l'IUT nous les avons réalisés chez nous. Par la suite, nous avons pris un moment après les passations pour faire un débrief et prioriser les remarques à mettre en place.

Pour tous les documents hors du code, nous les avons partagés sur un Google Drive afin que tout le monde puisse y avoir accès et les modifier. Cela a notamment été utile lors

de la dernière semaine pour les tests utilisateurs, nous avons pu partager facilement à tous les membres de l'équipe tous les documents nécessaires à une passation de test utilisateur. Nous avons aussi fait le bilan de ces tests sur ce drive afin que tout le monde puisse consulter les modifications à mettre en place et choisir lesquels faire.

Enfin, tout au long de ce projet, nous avons utilisé un serveur discord bien organisé sur lequel nous pouvions communiquer efficacement.

Mesures techniques

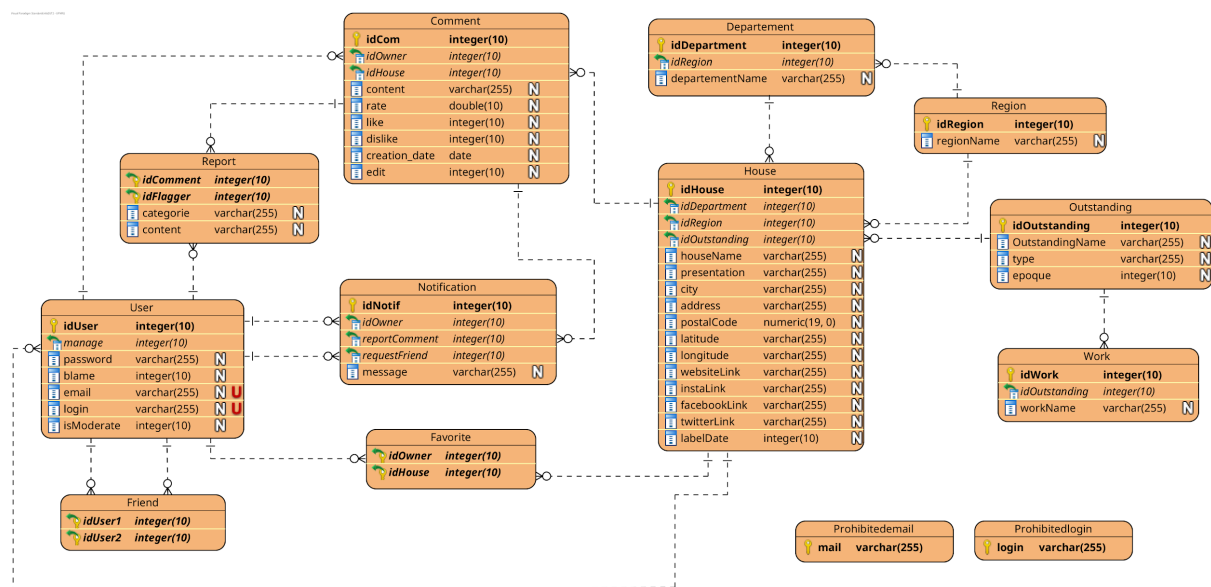
Lors de la phase de développement, nous avons principalement utilisé deux IDE, Visual Studio Code pour les vues et les scripts, et PhpStorm pour le modèle, les contrôleurs et lorsque nous avons besoin de faire du pair programming. Nous avons bien entendu relié nos environnements de programmation à GitLab afin de pouvoir partager le code entre tous.

Nous avons aussi mis en place un script sur le serveur Apache pour pouvoir faire des tests directement sur le serveur. En effet, le serveur réalisait un "git pull" toutes les minutes afin d'être toujours en cohésion avec le code du serveur Gricad du projet. Cette mesure était nécessaire pour nous puisque pour accéder à la base de données, et donc tester nos contrôleurs, il nous fallait accéder à la base de données uniquement disponible sur le serveur.

De plus, afin de réaliser les tests unitaires sans affecter l'avancée des contrôleurs, nous avons créé une copie de notre base de données. Les tests modifiaient donc des données extérieures à l'application.

Synthèse technique

Modélisation



Légende:

N = potentiellement null

U = Unique.

flèche verte = clef étrangère

Par rapport au SEA de la phase 2 nous avons fait certaines modification:

- Nous avons changé certains types d'arguments pour qu'ils correspondent mieux au fonctionnement réel de l'application.
- Nous avons aussi ajouté certains arguments pour créer des fonctionnalités comme par exemple creation_date et edit dans la table commentaire, qui permettent d'indiquer si un commentaire à été modifié et de trier les plus récents
- Nous avons aussi ajouté une table report qui stocke plus d'informations sur les signalements pour donner du contexte au modérateur.



Dans ce diagramme qui correspond à notre code nous avons ajouté une classe “report” qui sert pour signaler un commentaire. Elle est composée de deux attributs : une catégorie pour savoir si le commentaire est offensant ou incorrect, etc. Et l’attribut content qui est la raison du signalement de l’utilisateur. Tous les commentaires signalés vont apparaître dans la vue du modérateur pour pouvoir trier les demandes.

Nous avons aussi changé certaines multiplicités d’attributs comme les maisons qui peuvent avoir maintenant qu’un seul illustre (outstanding).

De plus, la classe work a été transformée en attribut et contient une liste de string.

Nous avons aussi enlevé la classe image car nous n’avions pas de données qui y correspondait les images sont prises à partir de l’API de google qui prend des image de “street view”.

Nous avons aussi ajouté un grand nombre de fonctions que nous n’avions pas prévu dans la phase de modélisation.

Nos choix techniques

Pour coder cette application, nous avons choisi d’utiliser PHP en version 8.2, car c’est le langage de programmation qui nous semblait le plus adapté pour développer le backend d’une application web. De plus, c’est celui qui nous a été enseigné en cours de développement web. Pour des raisons similaires, nous avons choisi de développer l’interface utilisateur en HTML/CSS et JavaScript.

En ce qui concerne le déploiement de l’application, il est réalisé grâce à deux machines virtuelles (VM) sur le serveur Assr mis à notre disposition par l’IUT. La première

héberge l'application grâce à Apache et la rend disponible via ngrok. La deuxième gère notre base de données grâce à PostgreSQL en version 15.

Nous avons effectué la gestion des versions avec Git et hébergé le code sur Gricad, le serveur GitLab également mis à disposition par l'IUT.

Pour communiquer, nous avons choisi d'utiliser Discord, car c'est sur cette plateforme que nous répondons le plus rapidement. De plus, la création de différents salons sur notre serveur de SAE nous permet de séparer facilement les discussions.

Bilan Technique

En conclusion de cette synthèse technique, nous pouvons constater que nos choix techniques ont été globalement pertinents. Les langages, les logiciels, et même l'API Google Maps n'ont pas posé de problèmes majeurs au cours de cette SAE. Même si le choix d'un framework nous aurait permis d'implémenter plus de fonctionnalités.

En revanche, concernant les choix de conception, nous avons moins bien anticipé les besoins nécessaires à la réalisation des fonctionnalités. Cela nous a conduits à modifier de manière significative les diagrammes à plusieurs reprises. Et nous a fait perdre quelques heures au début du projet.

Bilan général

Pour finir, nous pouvons retenir que notre projet a traversé trois phases clefs : le cadrage, la conception et la réalisation.

Lors de la phase de cadrage, nous avons su identifier les besoins et les objectifs de notre application ainsi que les contraintes et risques potentiels. Nous avons également établi un rythme et une méthode de travail efficace nous servant de base solide pour les phases suivantes.

La phase de conception nous a permis de modéliser et de formaliser les besoins fonctionnels et non-fonctionnels de l'application. Nous avons créé des maquettes, des diagrammes, des modèles et un plan du site. Nous avons également validé nos choix technologiques. Ces éléments nous ont permis de mieux visualiser l'application et de préparer au mieux sa mise en œuvre.

Enfin, la phase de réalisation a permis de concrétiser notre application par le développement de nos principales fonctionnalités, atteignant notre objectif de créer une application reliant le patrimoine français et ses utilisateurs. Les différents tests techniques et utilisateurs nous ont permis d'optimiser l'ergonomie et la sécurité de l'application. Malgré nos choix de conception parfois remis en doute, nous avons terminé la création de notre application, ce qui marque la fin de notre projet.

Au terme de ces trois phases, nous pouvons considérer que notre projet a été un succès. Nous sommes fiers du résultat obtenu.

Annexe

Readme

Équipe Teapot Studio

Cheffe projet
DUBIEF Émilie

Membres

- RASTELLI Maxime
 - KUS Ali
 - RAYOT Nils
 - RIONDET Flavien
 - FRANCOU Eddy
 - ROSTAGNAT Aline
-

Notadomus

NotaDomus est une application web permettant de naviguer parmi les différentes **Maisons d'illustres** réparties à travers la France. Ce sont principalement des bâtiments ayant reçu ou ayant un quelconque rapport avec des artistes, auteurs, inventeurs, etc.

Ceci consiste en une carte interactive utilisant l'API Google Maps. Il est aussi possible de filtrer les maisons et de les sélectionner en utilisant le panneau de gauche. Enfin, la dimension sociale est accessible à partir du panneau de droite. Celui-ci permet de voir sa liste d'amis, d'en ajouter, de les supprimer...

Dans la barre de navigation située en haut, il est possible d'effectuer une recherche par mots-clés. Il est possible de retrouver une Maison d'illustre à partir du nom de l'illustre en question, ou même avec le nom d'une de ses œuvres.

Ce dépôt est organisé comme suit :

rendus

```
|— controller
|— data
|— docs
|— framework
|— log
|— model
|— public
|   |— api
|   |— design
|   |   |— img
|— test
|   |— nonUnit
|— view
|   |— back-up
|   |— script
```


Il reprend la structure **MVC** avec routeur. C'est-à-dire que le seul fichier dont doit faire objet une requête est `index.php`, situé à la racine du projet. Un nom de contrôleur est spécifié dans la query-string. `index.php` va charger le script de la partie `controller` correspondant à ce nom.

Le contrôleur récupère les valeurs nécessaires dans le tableau `$_POST` ou `$_GET`. Ensuite, il appelle des méthodes de la partie `model` pour calculer la valeur des données à caractère dynamique qu'il faudra afficher. Enfin, il charge une vue pertinente de la partie `view`. Les vues sont des fichiers composés quasi-exclusivement de HTML. Les seuls "morceaux" de code PHP permettent d'intégrer les données produites par le contrôleur en amont, ou d'afficher des structures en prenant en compte une condition.

Certaines vues intègrent des fichiers Javascript. Ceux-ci ajoutent un comportement qui s'exécutera sur le client à la suite du chargement du fichier ou de l'appui sur un bouton.

Cahier de tests

Ballade cognitive

Lors de la phase de conception, nous avons réalisé une balade cognitive afin de tester nos maquettes, elle nous a permis de remarquer certains problèmes.

Dans un premier temps, le guidage n'était pas assez explicite. En effet, lors de la réalisation de la fonctionnalité de filtrage des maisons, le bouton destiné à cet effet n'était pas assez mis en avant. Cela a conduit le testeur à vouloir entrer les critères de filtrage directement dans la barre de recherche de l'application.

De plus, l'utilisateur a rencontré un problème pour accéder à ses amis. En effet, son premier réflexe a été de cliquer sur l'icône utilisateur en haut à droite de l'écran. C'est pourquoi, nous avons déplacé les informations utilisateur au-dessus des amis. Cela permet de mieux respecter la signification des codes et dénominations puisque l'icône utilisée pour les amis représente aussi le compte de l'utilisateur dans beaucoup d'applications.

Cette ballade cognitive a été réalisée par un utilisateur qui a ensuite participé aux tests utilisateurs en think aloud détaillé plus bas. Il a ainsi pu observer l'amélioration de l'application et nous faire de nouveaux retours.

Tests unitaires

Nous avons utilisé un framework du nom de PHPUnit afin d'automatiser les tests unitaires et d'avoir des assertions qui vérifient efficacement nos conditions. Ainsi, la garantie de la réussite ou non des tests dépend du fonctionnement du modèle et non de celui de la base de données. Nous avons par la suite conçu une base de données contenant les informations nécessaires pour servir de pré-condition aux tests. Cela a aussi pour but de ne pas impacter l'application avec les tests longs et répétitifs. Enfin nous avons choisi de ne pas contrôler les getters simples qui n'ont pas de risque d'erreur.

Tests utilisateurs - Think Aloud

L'objectif de ce test est de vérifier l'utilisabilité de notre application de manière qualitative. Nous avons opté pour cette approche puisque nous avons réalisé une

application sociale, par conséquent nous voulions observer la façon dont les utilisateurs vont utiliser notre service.

Nous avons sélectionné nos testeurs selon deux critères. Premièrement, nous avons choisi des personnes ayant répondu au formulaire réalisé lors du cadrage du projet. Deuxièmement, nous les avons choisis selon nos persona identifiés :

- Une étudiante de 22 ans qui aime utiliser les réseaux sociaux pour partager sa passion;
- Un père de famille qui cherche à faire des visites le week-end.

Afin de garantir le bon déroulement de nos tests, nous devons nous assurer que l'application était prête, nous avons donc démarré ces derniers à la fin de la deuxième semaine de développement. Il était important pour nous d'avoir le temps de mettre en place les retours des utilisateurs.

Pour réaliser ces tests, nous avons préparé le nécessaire à chaque passation :

- un formulaire de consentement disponible [ici](#);
- un questionnaire que les utilisateurs doivent remplir avant de faire le test, il nous permet de récupérer les informations suivantes :
 - l'âge du participant;
 - l'utilisation de cartes interactives;
 - l'utilisation des réseaux sociaux.
- les tâches que les utilisateurs doivent effectuer, elles sont disponibles [ici](#).

Nous avons réalisé six passations, les utilisateurs nous ont fait remonter des bugs et des améliorations pour l'application. Nous avons classé ces remarques dans un tableau en indiquant dans la colonne "Application ?" notre volonté à mettre en place les remarques, et dans la colonne "Etat" l'avancée des modifications.

Type	Modifications	nbParticipants concernés	Application ?	Etat
Amélioration	grouper les points sur la carte (ou les afficher au fur et à mesure)	1	NON	-
Amélioration	déplacer les pop-up de la carte en haut à droite	1	NON	-
Amélioration	ouvrir la page des maisons dans une autre fenêtre	1	NON	-
Amélioration	baissier le pop-up des propositions par défaut	1	NON	-
Amélioration	zoomer sur le bon département si il est sélectionné	1	NON	-
Amélioration	afficher les commentaires de l'utilisateur d'une autre couleur	1	OUI	FAIT
Amélioration	accéder à la page des maisons depuis les marqueurs de la carte	1	OUI	FAIT
Amélioration	cacher le texte du pop-up de propositions quand il est baissé, agrandir la flèche	1	OUI	FAIT
Amélioration	afficher des noms de maisons quand on écrit dans la barre de recherche	1	OUI	FAIT
Amélioration	inverser la connexion et l'inscription sur la page d'accueil	3	OUI	FAIT

Amélioration	mettre le logo dans l'onglet de la fenêtre	1	OUI	FAIT
Amélioration	ajouter une barre de recherche pour ajouter des amis	1	OUI	FAIT
Amélioration	recentrer sur la maison sélectionnée dans les filtres	3	OUI	FAIT
Amélioration	pouvoir voir son mot de passe lors de la connexion	1	OUI	FAIT
Amélioration	pouvoir accéder au commentaire signalé depuis la notification	3	OUI	FAIT
Amélioration	agrandir les notifications	1	OUI	FAIT
Amélioration	ajouter l'accès à ses propres commentaires dans la vue des amis	2	OUI	FAIT
Amélioration	mettre le marqueur sélectionné en rouge sur la carte	4	OUI	FAIT
Amélioration	ajouter un hover textuel pour les boutons filtre et amis	1	OUI	FAIT
Amélioration	mettre en avant le bouton filtrer par	1	OUI	FAIT
Amélioration	rendre cliquable la photo de la maison dans les filtres	1	OUI	FAIT
Amélioration	mettre en avant le bouton filtrer dans le formulaire des filtres	1	OUI	FAIT
Amélioration	ignorer les accents dans la barre de recherche	2	OUI	FAIT
Amélioration	déplacer le formulaire pour poster un commentaire dans la page des maisons	2	OUI	FAIT
Amélioration	réécrire les commentaires et les étoiles pour la modification	2	OUI	FAIT
Amélioration	mettre un bouton retour arrière explicite sur la plage des maisons	1	OUI	FAIT
Amélioration	mettre une flèche pour le pop-up de proposition de maison	2	OUI	FAIT
Amélioration	recentrer la carte selon le marqueur sélectionné	2	OUI	FAIT
Amélioration	rendre plus accessible les dep d'outre-mer	1	OUI	FAIT
Amélioration	ajouter un message lorsque les filtres ne correspondent à aucune maison	1	OUI	FAIT
Amélioration	mettre plus de critères dans la barre de recherche (villes, départements, régions...)	1	OUI	FAIT
Amélioration	réactualiser les 30 maisons par défaut à chaque rechargement	1	OUI	FAIT
Amélioration	mettre une recherche textuelle des régions et des départements dans le formulaire des filtres	2	OUI	PLUS TARD
Amélioration	afficher ses départements quand une région est sélectionnée dans le formulaire des filtres	1	OUI	PLUS TARD
Amélioration	empêcher de proposer une maison déjà existante	1	OUI	PLUS TARD

Problème	problème de liens mort pour les sites de certaines maisons	2	NON	-
Problème	problème de confusion avec le bouton ovale des filtres	1	OUI	FAIT
Problème	problème d'affichage des amis supprimés	1	OUI	FAIT
Problème	problème de bouton remettre à zéro dans les filtres	1	OUI	FAIT
Problème	problème de filtre favoris	1	OUI	FAIT
Problème	problème de "?" et de "1/2" dans les textes de certaines maisons	3	OUI	PLUS TARD
Problème	problème d'image pour certaines maisons	2	OUI	PLUS TARD

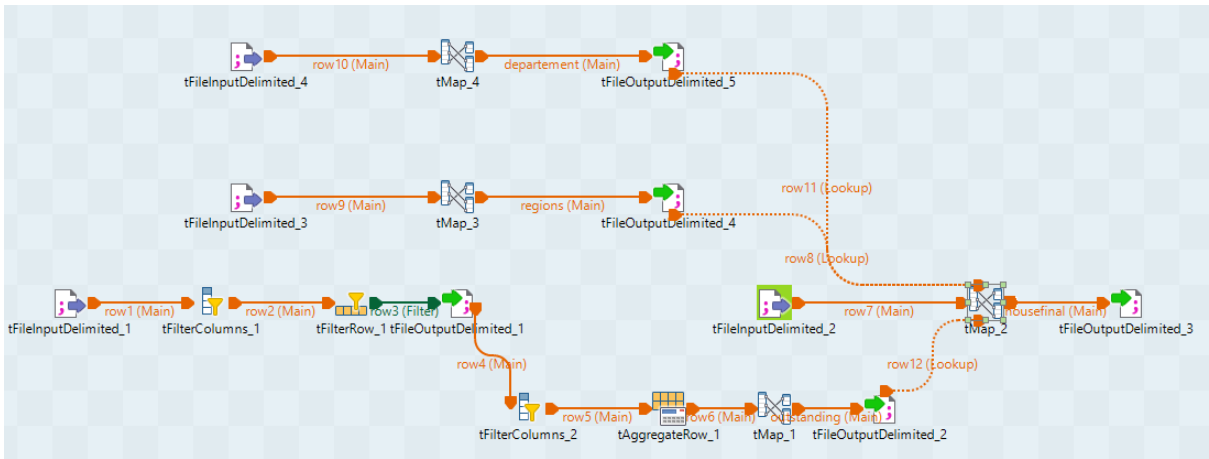
Traitements ETL

Le flux ETL mis en place pour constituer notre base de données se repose sur trois jeu de données disponible sur le site data.gouv:

- le jeu de données des maisons d’illustre françaises
- le jeu de données des régions françaises
- le jeu de données des départements français

Lors de la précédente phase, nous avons établi un schéma de ce flux afin de rendre sa réalisation plus efficace sur Talend.

Le flux final est représenté ci-dessous, il s’agit d’une capture d’écran du logiciel Talend. Le détail de certains composants est renseigné plus bas.



La réalisation de ce flux a nécessité la création de métadonnées générées à partir des jeux de données du gouvernement :

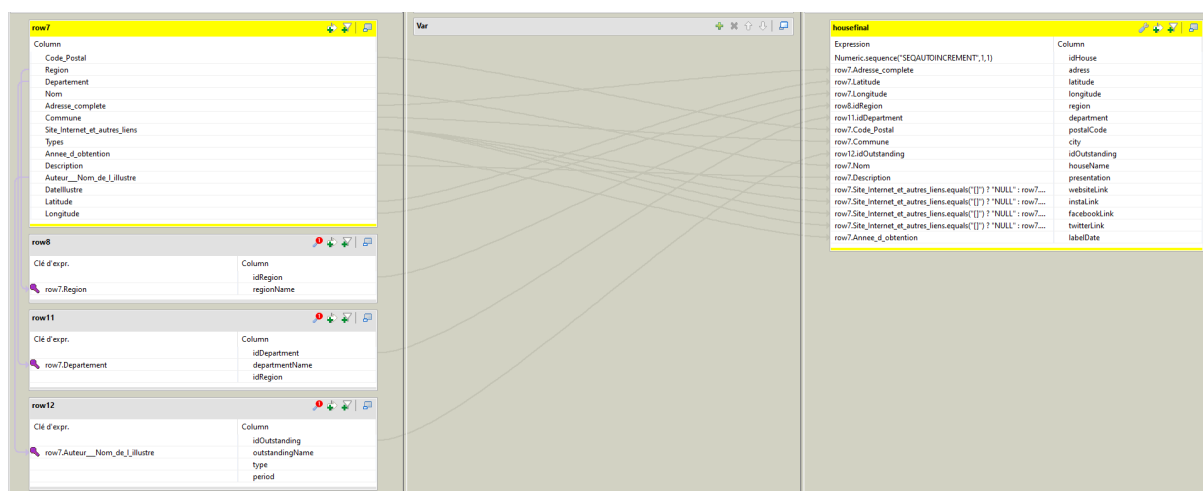
- department 0.1 (contenu dans `tFileInputDelimited_4`) qui se base sur la liste des départements français et qui renseigne pour chaque département son nom, son code et le nom de sa région;
- region 0.1 (contenu dans `tFileInputDelimited_3`) qui se base sur la liste des régions françaises et qui renseigne pour chaque région son nom et son code;

- house 0.1 (contenu dans tFileInputDelimited_1) qui se base sur la liste des maisons d'illustre françaises et qui renseigne pour chaque maison une trentaine de critères;

Tous ces fichiers ont subi des transformations afin de correspondre à notre schéma de base de données, les fichiers de sorties associés sont :

- tFileOutputDelimited_5 renseigne le nom, le code et le code de la région de chaque départements français;
- tFileOutputDelimited_4 renseigne le nom et le code de chaque régions françaises;
- tFileOutputDelimited_1 est une version nettoyée de la base des maisons d'illustre, ce fichier a donné lieu à la création de métadonnées contenues dans tFileInputDelimited_2 afin de simplifier le flux;
- tFileOutputDelimited_2 renseigne les informations relative aux illustres afin de les séparer des maisons;

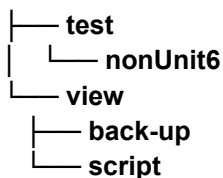
Le composant tMap_2 nous a permis de faire la jointure finale entre tous ces fichiers de sorties afin d'obtenir la table des maisons d'illustre complète. Nous avons utilisé les départements, les régions, les données simplifiées des maisons et les illustres. Le détail de ce composant est le suivant :



Structure du code

Nous avons opté pour une structure similaire aux TPs, à savoir une structure MVC avec routeur (uniquement côté serveur) :

```
./
├── controller
├── data
├── docs
├── framework
├── log
├── model
├── public
├── design
└── img
```



Sécurité

Pour les Serveurs

Pour assurer la sécurité du serveur nous avons en premier lieu automatisé les mises à jour de sécurité de nos serveurs grâce à l'outil unattended-upgrade. Nous avons aussi mis en place un système qui nous envoie des mails à chaque mise à jour pour que nous puissions vérifier qu'elles s'intègrent correctement à notre environnement.

Pour que notre application web soit accessible de partout nous sommes passés par le logiciel ngrok qui nous donne un lien vers notre serveur depuis l'adresse privée. Ce logiciel sécurise aussi notre site avec un certificat signé par let's encrypt, une autorité assez importante.

Pour la sécurité des communications entre nos serveurs et nos clients, nous avons mis en place des certificats signés grâce à ngrok. Cet outil nous sert à rendre notre site web accessible et nous permet de faire des certificats validés par Let's Encrypt et crypte avec du SHA-256.

Pour notre DB nous avons uniquement autorisé le serveur apache à communiquer avec lui grâce au fichier pg_hba.conf .

Pour l'Application

Nous laissons à plusieurs endroits les utilisateurs agir sur notre application. Nous avons donc mis en place les solutions suivantes :

Pour éviter les Injection SQL nous faisons tous nos accès à notre base de donnée grâce à des requêtes préparées

Concernant les attaques de persistance XSS, nous empêchons l'inscription de caractères spéciaux dans notre base de donnée grâce à des filtres dans le code php lors de la récupération des données.

Pour les attaques CSRF nous n'avons pour l'instant pas de solution. Nous avons préféré nous concentrer sur d'autres types d'attaques plus impactantes. En effet nous n'avons pas de formulaire de paiement, ni de traitement d'autres données sensibles. Dans le futur notre application permettra de faire des donations pour aider les maisons d'illustres, mais le code de l'application pourra être modifié pour empêcher cette attaque.

Enfin pour plus de sécurité nous ne stockons pas les mots de passe en clair mais avec du hachage et du salage, grâce à la méthode argon 2.

Autres documents

Formulaire de consentement

Formulaire de consentement

Le studio de développement TeapotStudio est en voie de déployer une nouvelle application web nommée NotaDomus. Elle désire donc rencontrer des utilisateurs et recueillir leurs ressentis vis-à-vis de l'utilisation de cette application. Le but de cette étude est donc de recueillir des informations afin de faire évoluer l'application de façon à ce qu'elle soit adaptée aux besoins des utilisateurs.

Le recueil de ces informations est obtenu par l'usage des deux méthodes suivantes :

- Un questionnaire écrit afin de recueillir les informations des participants utile à l'amélioration de l'application (âge, compétence en informatique)
- Un test utilisateur durant lequel le participant interagit avec l'application web.

L'enregistrement des données se fera par une notation directe des paroles du participant lors de son test.

TeapotStudio s'engage par la présente à ce que les données recueillies soient exploitées de manière à ce que l'identité du Participant ne figure pas dans les résultats, uniquement les données recueillies dans le questionnaire seront associées. Les données recueillies seront détruites à compter de deux mois après la fin du test utilisateur et ne seront pas divulguées en dehors du studio.

En apposant sa signature, le Participant déclare avoir pris connaissance des conditions de réalisation de l'étude et accepte d'y participer :

(Signatures des deux parties)

Tâches des participants

Tâches des participants aux tests utilisateurs

1. Se connecter à l'application à l'aide des informations suivantes:
 - login: *à changer en fonction du participant*

- mot de passe: *à changer en fonction du participant*
- 2. Consulter vos notifications
- 3. Afficher sur la carte toutes les maisons du 19e siècle en Bretagne
- 4. Commenter la Maison Ernest Renan
- 5. Ajouter cette maison en favoris
- 6. Consulter la page de cette maison et modifier le commentaire
- 7. Signaler un commentaire sur la page d'une autre maison
- 8. Proposer une nouvelle maison avec les caractéristiques de votre choix
- 9. Demander en ami l'utilisateur suivant :
 - Login : aluso
- 10. Rechercher la maison de l'auteur de *La liberté guidant le peuple*
- 11. Accéder à la page facebook de cette maison
- 12. Explorer librement le site