

Billetterie du Louvre

PRÉSENTATION

Marion BATTISTA

battista.marion@gmail.com




Table des matières

1. SUIVI DES VERSIONS	3
2. Présentation	4
2.1. Le projet	4
2.2. Les enjeux	4
3. Type de billet	5
4. Date de réservation	6
4.1. Spécifications fonctionnelles	6
4.2. Spécifications techniques	6
5. Email	7
5.1. Spécifications fonctionnelles	7
5.2. Spécifications techniques	7
6. Utilisateurs	8
6.1. Spécifications fonctionnelles	8
6.2. Spécifications techniques	8
6.2.1. Base de données	8
6.2.2. Création	8
7. Paiement	10
7.1. Spécifications fonctionnelles	10
7.2. Spécifications techniques	10
8. Annexes	11

1. SUIVI DES VERSIONS

Cette section met en évidence les évolutions du document, et quelles parties ont été mises à jour pour faciliter la relecture.

➔ V.1.0, 28 Mai 2017

2. Présentation

2.1. Le projet

Le musée du Louvre nous a missionné pour un projet ambitieux : créer un nouveau système de réservation et de gestion des tickets en ligne pour diminuer les longues files d'attente et tirer parti de l'usage croissant des smartphones.

2.2. Les enjeux

L'interface doit être accessible aussi bien sur ordinateur de bureau que tablettes et smartphones, et utiliser pour cela un design responsif.

L'interface doit être fonctionnelle, claire et rapide avant tout. Le client ne souhaite pas surcharger le site d'informations peu utiles : l'objectif est de permettre aux visiteurs d'acheter un billet rapidement.

3. Type de billet

Le musée propose plusieurs types de tarifs :

- ➔ Un tarif « normal » à partir de 12 ans à 16 €
- ➔ Un tarif « enfant » à partir de 4 ans et jusqu'à 12 ans, à 8 € (l'entrée est gratuite pour les enfants de moins de 4 ans)
- ➔ Un tarif « senior » à partir de 60 ans pour 12 €
- ➔ Un tarif « réduit » de 10 € accordé dans certaines conditions (étudiant, employé du musée, d'un service du Ministère de la Culture, militaire...)

Les différents types de billets sont stockés dans MbTicket.php qui permet :

- ➔ De récupérer le type de billet en fonction de l'âge du / des participants
`getKindFromBirthday($date, $isReduced)`
- ➔ De récupérer le type de billet en fonction de l'âge du / des participants
`getPriceFromBirthday($date, $isReduced)`
- ➔ De récupérer le nom du billet en fonction du type
`getNameFromKind($kind)`
- ➔ De récupérer le prix depuis le type de billet
`getPriceFromKind($kind)`

4. Date de réservation

4.1. Spécifications fonctionnelles

Il existe 2 types de billets : le billet « Journée » et le billet « Demi-journée » (il ne permet de rentrer qu'à partir de 14h00). Le musée est ouvert tous les jours sauf le mardi et le dimanche. Le musée est fermé pour les jours fériés suivants : le 1er mai, le 1er novembre et le 25 décembre.

Le client précise qu'il n'est pas possible de réserver pour les jours où plus de 1000 billets ont été vendus en tout pour ne pas dépasser la capacité du musée.

Quel jour souhaitez-vous venir ?

26/05/2017

☐ journée

☐ demi-journée

4.2. Spécifications techniques

Les utilisateurs peuvent sélectionner le jour de leur réservation grâce au plugin jQuery UI DatePicker.

November 2017

L	M	M	J	V	S	D
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

17/11/2017

November 2017

L	M	M	J	V	S	D
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Puis sélectionner la durée de la visite :

- ☐ journée
- ☐ demi-journée

5. Email

5.1. Spécifications fonctionnelles

Pour chaque commande, une adresse mail doit être récupérée afin d'envoyer un mail de confirmation de réservation et de paiement à l'utilisateur. Ce mail fera office de billet d'entrée pour le jour de la visite.



Votre adresse mail

xxxx@xxx.xxx

5.2. Spécifications techniques

Pour l'envoi de mail, Symfony dispose de sa propre librairie : Swift Mailer. Le service MailerManagementService.php instancie Swift Mailer et permet d'envoyer un modèle de mail créé à partir d'un template Twig [registration.html.twig](#) une fois le paiement et la réservation effectués.

Dans le mail envoyé doit apparaître :

- ➔ Le nom et le logo du musée
- ➔ La date de la réservation
- ➔ Le tarif
- ➔ Le nom de chaque visiteur
- ➔ Le code de la réservation (un ensemble de lettres et de chiffres)



Le Louvre vous remercie d'avoir réservé vos places !

Détails de votre commande :

Vous venez de commander 3 billets pour la journée du 15/06/2017

Numéro de ticket	Prénom	Nom	Prix
n°LVMBCS000004	Battista	Mairon	16 €
n°LVMBCS000005	senior	senior	12 €
n°LVMBCS000006	reduit	reduit	10 €
TOTAL			38 €

Vous avez réglé la somme de 38 €, nous vous attendons le 15/06/2017 pour la visite du Louvre

À bientôt !

6. Utilisateurs

6.1. Spécifications fonctionnelles

Pour chaque billet, l'utilisateur doit indiquer son nom, son prénom, son pays et sa date de naissance. Elle déterminera le tarif du billet.

Si la personne dispose du tarif réduit, elle doit simplement cocher la case « Tarif réduit ». Le site doit indiquer qu'il sera nécessaire de présenter sa carte d'étudiant, militaire ou équivalent lors de l'entrée pour prouver qu'on bénéficie bien du tarif réduit.

6.2. Spécifications techniques

6.2.1. Base de données

```

Id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
firstname VARCHAR(255) NOT NULL
lastname VARCHAR(255) NOT NULL
birthday DATE NOT NULL
country VARCHAR (255) NOT NULL
ticket_id INT NOT NULL
isReduced SMALLINT NOT NULL

```

6.2.2. Création

Le formulaire a été généré par Symfony depuis l'entité MbUser. L'ajout et la suppression des participants sont directement gérés avec du JavaScript. Enfin, lorsque l'utilisateur valide le formulaire de réservation, le formulaire de paiement apparaît en dessous. Le panier de

l'utilisateur apparaît également et affiche les noms et prénoms de tous les participants, le prix individuel du billet et le prix total.

Le prix du billet de chaque participant est calculé en JavaScript à partir de la fonction `updateBasket()` qui permet :

- ➔ De faire apparaître la div de paiement,
- ➔ De faire apparaître le panier de l'utilisateur,
- ➔ De stocker les valeurs du formulaire de réservation au format JSON
- ➔ De calculer les différents prix
- ➔ De construire le panier avec les données à jour grâce à la fonction `getUsersPriceAction(Request, $request)` depuis le `DefaultController.php`

7. Paiement

7.1. Spécifications fonctionnelles

Le visiteur doit pouvoir payer avec la solution Stripe par carte bancaire.

Le site doit gérer le retour du paiement. En cas d'erreur, il invite à recommencer l'opération. Si tout s'est bien passé, la commande est enregistrée et les billets sont envoyés au visiteur. Stripe fournit des environnements de test pour simuler la transaction, afin de ne pas avoir besoin de rentrer notre propre carte bleue.

- ➔ Numéro de carte bleue : 4242 4242 4242 4242,
- ➔ Date d'expiration : n'importe quelle date,
- ➔ CVC : 123



The screenshot shows a payment details form titled "Détails de paiement" on a dark background. The form contains four input fields: "N° de carte" with the value "4242424242424242", "Mois" with the value "12", "Année" with the value "18", and "CVC" with the value "123". At the bottom right, there are two buttons: "Modifier votre commande" (orange) and "Réserver" (white).

7.2. Spécifications techniques

Le service [StripeManagementService.php](#) la connexion à l'API de Stripe. Le service est appelé dans le [DefaultController.php](#) après que le formulaire de réservation est été rempli. Si le paiement est refusé alors l'utilisateur est redirigé vers les informations de paiement pour recommencer l'opération.

8. Annexes

CONTROLLER

DefaultController
indexAction(Request \$request)
getUsersPriceAction(Request \$request)

SERVICES

MailerManagementService.php
getMailerService(\$order, \$emailBody)

TicketLimitManagementService
getOnlyHalfDayService(\$order)
getNumberMaxOfTicketService(\$totalTickets)

StripeManagementService.php
getStripeService(\$order)

ENTITIES

