Uber Cut

Cahier des charges & Expression des besoins

Wassim Bacha

Table des matières

[Personal presentation 3](#_Toc169662750)

[About me 3](#_Toc169662751)

[About my job 3](#_Toc169662752)

[ADventori 3](#_Toc169662753)

[DCO 3](#_Toc169662754)

[Mon rôle 4](#_Toc169662755)

[Une de mes missions 4](#_Toc169662756)

[Conclusion 4](#_Toc169662757)

[Présentation du projet 4](#_Toc169662758)

[Cas d’un utilisateur 5](#_Toc169662759)

[Cas d’un coiffeur 5](#_Toc169662760)

[Cahier des charges 5](#_Toc169662761)

[Objectifs 5](#_Toc169662762)

[Cibles 6](#_Toc169662763)

[Exigences fonctionnelles 6](#_Toc169662764)

[Exigences ‘Front-end’ 6](#_Toc169662765)

[Exigences ‘Back-end’ 7](#_Toc169662766)

[L’utilisateur 7](#_Toc169662767)

[Confidentialité 8](#_Toc169662768)

[Droits d’accès 8](#_Toc169662769)

[Exigences & choix techniques 9](#_Toc169662770)

[Exigences 9](#_Toc169662771)

[Choix 9](#_Toc169662772)

[Définition du MVP (Minimum Viable Product) 9](#_Toc169662773)

[Contrainte de temps 10](#_Toc169662774)

[Méthodologie & Organisation 11](#_Toc169662775)

[Conception de l’interface graphique 14](#_Toc169662776)

[Zoning 14](#_Toc169662777)

[Charte graphique 14](#_Toc169662778)

[Les couleurs 15](#_Toc169662779)

[Le logo 15](#_Toc169662780)

[Le Wireframe 15](#_Toc169662781)

[Maquettage 16](#_Toc169662782)

[Conception de la base de données 17](#_Toc169662783)

[Modèle Conceptuel de Données (MCD) 17](#_Toc169662784)

[Modèle Logique de Données (MLD) 19](#_Toc169662785)

[Conception de l’application 21](#_Toc169662786)

[Diagramme de Cas d'Utilisation (Use Case) 21](#_Toc169662787)

[Conception multicouche MVC 23](#_Toc169662788)

[Le Modèle 23](#_Toc169662789)

[La Vue 26](#_Toc169662790)

[Le Contrôleur 27](#_Toc169662791)

[Communication entre les 3 composants 30](#_Toc169662792)

[L’architecture 3 tiers 30](#_Toc169662793)

[Sécurité 30](#_Toc169662794)

[Les attaques XSS (Cross-Script Scripting) 30](#_Toc169662795)

[Les injections SQL 30](#_Toc169662796)

[Les attaques CSRF (Cross-Site Request Forgery) 30](#_Toc169662797)

[Politique de test 30](#_Toc169662798)

[Les tests unitaires 30](#_Toc169662799)

[Les tests d’intégration 30](#_Toc169662800)

[Veille Technologique 30](#_Toc169662801)

[Veille globale 30](#_Toc169662802)

[Veille sécurité 30](#_Toc169662803)

[Difficultés rencontrées 30](#_Toc169662804)

[Conclusion 30](#_Toc169662805)

[Remerciments 30](#_Toc169662806)

[Annexe 30](#_Toc169662807)

[Cahier des charges 30](#_Toc169662808)

# Personal presentation

## About me

Hello, I'm Wassim BACHA, currently 22 years old. I'm in my third year of full-stack development studies at EFREI while working at ADventori.

Although I currently reside in Villejuif, I've had the opportunity to move quite frequently, exploring different cities and opportunities. Last year, I worked as a Computer Engineer at Hanover Displays in Nice.

I'm deeply passionate about software development and have a keen interest in video games. After completing my baccalauréat in Sciences, I pursued a multidisciplinary academic path in mathematics, computer science, and physics, followed by two years in a BTS SIO SLAM program in alternating work-study format. Today, I'm nearing the completion of my third year in a bachelor's program specializing in web and application development, still in alternating work-study format.

The project I'm presenting today is called Uber Cut. It's an innovative platform for scheduling appointments with local hairdressers.

## About my job

### ADventori

ADventori est une start-up créée en 2009 par Pierre-Antoine Durgeat, anciennement créateur de l’application Mappy.

Nous travaillons dans le domaine du marketing digital où nous sommes leader en France des bannières de publicité dynamiques (DCO).

### DCO

La DCO (Dynamic Creative Optimisation) est une technologie publicitaire qui permet de personnaliser en temps réel les informations présentes sur les bannières publicitaires.

Le contenu d’une bannière publicitaire s’adapte à l’utilisateur et à son contexte de navigation (profil, géolocalisation, flux de produits, météo, etc…).

Les clients peuvent diffuser une infinité de publicités à partir d’une seule bannière publicitaire.

Tout cela est possible grâce à du code, principalement en Groovy (Java) & JavaScript, sur lesquels j’ai été formé cette année.

### Mon rôle

Au sein d’ADventori, je suis Développeur Fullstack Junior et mon but est de m’occuper des tickets sur lesquels je suis attribué.

Cela me permet de pouvoir travailler sur une large sélection de sujets avec des stack et des technologies différentes.

Mes missions varient et consistent principalement à réparer du code, ajouter des fonctionnalités ou encore optimiser du code.

### Une de mes missions

Une de mes missions à été de développer avec mon tuteur une Creative Management Platform (CMP) pour le client Renault.

Il s’agit d’un dashboard permettant au client de monitorer, déployer et modifier ses bannières publicitaires.

Le langage de développement était Groovy (Java) et la mission nous à pris entre 5 et 10 jours à compléter.

### Conclusion

ADventori est un très bon tremplin pour mon apprentissage dans le développement fullstack, j’ai appris énormément de méthodologies de travail et forgé de solides bases en développement.

Mon but pour la suite serait donc de monter en compétences et de pouvoir exercer plus de responsabilités au sein de l’entreprise.

Par la suite, je pourrais donc étudier en mastère de développement Fullstack & Manager tout en passant à l’étape supérieure et pouvoir approfondir mes connaissances en développement, car notre solution est nourrie de plus de 10 ans de développement dans tous types de langages et toutes sortes de technologies.

# Présentation du projet

A l’origine, Uber Cut était une idée que j’ai eu en 2020 lors de la crise sanitaire du Covid-19, date à laquelle il était très compliqué de trouver un rendez-vous chez le coiffeur.

Uber Cut est donc une application web permettant à des clients de réserver des prestations à domicile.

Il existe 2 cas principaux d’utilisation.

## Cas d’un utilisateur

Dans le cas d’utilisation d’un client, celui-ci à la possibilité de se créer un compte et de réserver un coiffeur à domicile en remplissant un formulaire renseignant les diverses informations sur la coupe voulue.

À la suite du paiement, un rendez-vous est créé.

Le coiffeur effectue son travail et le client à ensuite la possibilité d’envoyer une note sur 5 étoiles accompagnée d’un commentaire.

## Cas d’un coiffeur

Dans le cas d’utilisation d’un coiffeur, celui-ci à la possibilité de se créer un compte, de renseigner ses diplômes, ajouter une photo de profil, une description et de se marquer comme ‘prêt à coiffer’ sur la plateforme dédiée sur l’application web.

Lorsqu’un client réserve un rendez-vous avec lui, le coiffeur à le choix de refuser ou d’accepter et d’aller à l’adresse du client pour lui faire sa prestation.

Ensuite, le coiffeur recevra son paiement.

# Cahier des charges

Voici quelques points importants de la partie fonctionnelle de l’application :

## Objectifs

Uber Cut est une plateforme web gratuite permettant de mettre en relation les clients et les coiffeurs. Le but est de permettre à n’importe qui de profiter du confort de se faire coiffer à domicile.

Uber Cut est aussi conçue afin d’être une opportunité de travail pour les coiffeurs amateurs et professionnels.

Les clients pourront réserver des coiffures à domicile et paieront plus ou moins cher dépendant de la coupe demandée et des compétences du coiffeur choisi.

Un système de notation permettra aux coiffeurs talentueux d’évoluer en étant mis en avant et en profitant de taxes réduites.

## Cibles

Les cibles de l’application Uber Cut sont répartis en 2 groupes :

Les clients :

* Les particuliers souhaitant bénéficier de services de coiffure à domicile.
* Les personnes à mobilité réduite ne pouvant pas se déplacer facilement jusqu’à un salon de coiffure.

Les Coiffeurs :

* Les professionnels de la coiffure afin d’arrondir leur fin de mois en offrant des services à domicile.
* Les débutants de la coiffure souhaitant apprendre tout en se faisant de l’argent de poche.

## Exigences fonctionnelles

Ici seront présentées les exigences fonctionnelles de l’application Uber Cut.

### Exigences ‘Front-end’

#### Page d’accueil

La page d'accueil doit être présente pour fournir un aperçu des services de l'application et permettre une navigation facile vers les autres sections.

#### Page Profil

La page profil doit permettre aux utilisateurs et aux coiffeurs de visualiser et de modifier leurs informations personnelles, telles que leurs coordonnées ainsi que leur historique de commandes et leur facture.

#### Page Administration

La page d'administration doit être disponible pour que les administrateurs puissent gérer les utilisateurs, les coiffeurs, les réservations et autres aspects de la plateforme.

#### Page Dashboard Coiffeur

La page dashboard dédiée aux coiffeurs doit leur permettre de gérer leur disponibilité, d'accepter ou de refuser des rendez-vous et de modifier leur vitrine de coiffeur (description & coupes disponibles).

#### Page Dashboard Client

La page dashboard dédiée aux clients doit leur permettre d’effectuer une recherche de coiffeurs dans les environs à partir d’un formulaire.

Ils peuvent ainsi sélectionner un coiffeur et créer un rendez-vous.

#### Page Historique des commandes

La page historique des commandes doit lister toutes les réservations passées des utilisateurs, avec les détails des prestations et la facture correspondante.

#### Page Mentions légales

La page "Mentions légales" doit être présente pour fournir les informations juridiques et réglementaires relatives au site.

### Exigences ‘Back-end’

#### Gestion des utilisateurs

Implémenter un système sécurisé d'inscription et d'authentification pour les utilisateurs et les coiffeurs tout en gardant la possibilité de modifier ou supprimer son profil.

#### Gestion des réservations

Permettre aux utilisateurs de créer des rendez-vous et aux coiffeurs de mettre à jour leur disponibilité en temps réel.

#### Evaluation & Commentaires

Permettre aux utilisateurs de laisser des notes et des commentaires sur les prestations des coiffeurs.

#### Dashboard Administrateur

Gérer les droits d'accès et les rôles des différents utilisateurs (administrateurs, modérateurs, coiffeurs, clients).

#### Sécurité

Assurer la protection des données personnelles des utilisateurs et des coiffeurs.

### L’utilisateur

Une fois que l’utilisateur à créé son compte, il doit avoir accès au contenu du site et à la page de modification du profil.

Les fonctionnalités disponibles aux utilisateurs sont :

* Consultation/Modification/Suppression de son compte
* Recherche de coiffeurs dans les environs
* Réservation de Séances
* Consultation de ses Réservations
* Évaluation des Prestations
* Consultation des Factures

Si l’utilisateur est un coiffeur, à la création de son compte, il devrait avoir accès à la page de modification du profil ainsi qu’au dashboard de coiffeur.

Les fonctionnalités disponibles aux coiffeurs sont :

* Consultation/Modification/Suppression de son compte
* Gestion de la Disponibilité
* Gestion des Réservations
* Consultation de son Historique de Prestations
* Mise à Jour de la Vitrine

### Confidentialité

Il est essentiel de protéger l'accès en lecture ou en écriture aux informations "privées" de chaque utilisateur. Les mesures suivantes doivent être mises en place :

* **Protection des Informations Privées :** Les informations privées des utilisateurs, telles que leurs réservations, leurs évaluations, et leurs données personnelles, doivent être strictement accessibles uniquement par eux-mêmes.
* **Suppression de Compte** : Les utilisateurs doivent pouvoir supprimer leur compte à tout moment, par exemple en cas de dysfonctionnement.
* **Visibilité et Modification des Données Privées** : Les réservations, commentaires, et évaluations créées par un utilisateur doivent être protégés de la vue ou de la modification par les administrateurs du site ou d'autres utilisateurs qui ne sont pas directement concernés. Seuls les coiffeurs impliqués dans une réservation ou les utilisateurs spécifiques désignés par le créateur du contenu (par exemple, ajoutés à une liste de favoris ou d'amis) peuvent avoir accès à ces données privées.

### Droits d’accès

L'accès aux différentes fonctionnalités et informations du site sera contrôlé en fonction des rôles du profil de l'utilisateur.

Les droits d'accès (lecture, écriture) doivent être déterminés en fonction de la catégorie d'informations et de l'identité de l'utilisateur.

Ces règles garantissent que les droits d'accès sont correctement gérés et que les informations sensibles des utilisateurs sont protégées tout en permettant une gestion efficace de la plateforme Uber Cut.

## Exigences & choix techniques

### Exigences

Afin de pouvoir atteindre le plus largement possible les cibles du projet, il conviendra que l’application soit fonctionnelle sur les principaux navigateurs internet, à savoir : Chrome, Firefox et Edge à minima.

En plus d’être adaptée aux écrans d’ordinateur, l’application devra être responsive, avec un accent particulier sur les appareils de type smartphone. Cette décision découle de la cible spécifique de l'application, qui vise principalement les utilisateurs mobiles.

Il est crucial de garantir une expérience utilisateur optimale sur les smartphones, principaux dispositifs utilisés pour accéder aux services de coiffure à domicile.

Le focus sera donc mis sur le développement d’une interface conviviale et adaptée aux smartphones, afin de permettre une utilisation aisée et intuitive pour les utilisateurs ciblés.

### Choix

Afin de faciliter la mise en place du design responsive, on utilisera la librairie CSS Bootstrap permettant de créer des designs rapidement et efficacement.

Du point de vue des langages de programmation, on optera pour le famework Symfony PHP pour le Back-end.

Pour le Back-end (Symfony PHP) :

* La structure et l’organisation de Symfony à laquelle je suis déjà habitué, l’organisation y est claire et organisée facilitant le développement, la maintenance et la scalabilité du projet.
* Les fonctionnalités de sécurité intégrées telles que l’authentification, la gestion des droits d’accès et la protection contre les attaques courantes (CSRF, XSS).
* La large communauté de développeurs et la grande quantité de documentation, rendant plus facile la résolution des problèmes et l’accès à certaines informations.

Pour le Front-end (Twig & Bootstrap CSS) :

* Bootstrap permet de créer des designs responsifs rapidement et efficacement, assurant une interface utilisateur conviviale sur tous les types d'appareils.
* L'intégration de Twig avec Bootstrap permet de séparer clairement la logique du backend et la présentation, simplifiant ainsi le développement et la maintenance du front-end.

## Définition du MVP (Minimum Viable Product)

Le MVP de l’application nous permettra :

* Inscription via un formulaire dans la page /register
* Connexion via un formulaire dans la page /login
* Effectuer une recherche de coiffeurs
* Effectuer une recherche de clients à coiffer
* Accès à la page d’administration pour les utilisateurs munis du rôle `USER\_ADMIN`
* Modifier ses informations personnelles
* Consulter l’historique des commandes ainsi que leur facture.

A diagram of a website

Description automatically generated

## Contrainte de temps

L’application est en cours de développement et devrait sortir publiquement d’ici janvier 2025.

Ce deadline me permettra ainsi de développer et peaufiner chaque élément présent sur le Diagramme de Use Case.

Cependant, étant donné le démarrage tardif du projet et les dates de rendu prévues du CDA, je me dois de proposer une application remplissant à minima les fonctionnalités du MVP.

A hand holding a phone

Description automatically generated

# Méthodologie & Organisation

Dans le cadre de ce projet, j’ai choisi la méthode agile pour la simple raison qu’il s’agit d’un projet 100% autonome et que je me suis familiarisé avec cette méthode de travail au sein de mon alternance chez ADventori.

La méthode Agile est un ensemble de pratiques et de principes pour la gestion de projets et le développement de logiciels qui mettent l'accent sur la flexibilité et la satisfaction du client. Cette méthode se distingue par son approche itérative et incrémentale, permettant de livrer des parties fonctionnelles du produit à intervalles réguliers, d'incorporer les retours et de s'adapter rapidement aux changements.

A diagram of a process

Description automatically generated

Pour l’organisation de mes tâches dans le projet, je me suis servi de Trello où j’ai réparti les différents points d’amélioration de l’application sous forme de tickets rangés par ordre d’importance.

Cette méthode permet d’avoir une vue claire sur l’avancement du projet.

A screenshot of a computer

Description automatically generated

Pour le versionning du code j’ai utilisé un repository GitHub (Git) me permettant de garder à jour mon environnement de travail même lorsque je change de machine.

GitHub me permet aussi de garder un œil sur les versions précédentes du code afin de revenir en arrière si un problème survient.

Le lien du repository est disponible à l’adresse suivante :

https://github.com/abwii/HubertCut

A screenshot of a computer

Description automatically generated

Voici un aperçu de mon historique de « Commits » sur GitHub, chaque ligne correspond à des changements dans le code.

A screenshot of a computer

Description automatically generated

# Conception de l’interface graphique

La conception de l’interface utilisateur (UI) et de l’expérience utilisateur (UX) occupent un rôle primordial dans le développement d’une application ayant pour but d’être rapide, efficace et utilisable depuis des appareils à plusieurs formats d’écran.

Ce chapitre se concentre donc sur la phase de conception graphique.

L’aspect visuel, l’ergonomie et l’expérience utilisateur ont été les points les plus soigneusement conçus lors du développement de cette application.

Le public est habitué à des applications rapides, fluides et intuitives.

Dans l’objectif de démocratiser Uber Cut, il est donc essentiel de proposer une expérience satisfaisante aux utilisateurs.

## Zoning

## Charte graphique

Afin de proposer une application visuellement agréable, je me suis grandement inspiré des applications Uber et Uber Eats dans les couleurs, la police d’écriture et l’agencement du site.

### Les couleurs

La couleur principale du site est : #1f1f1f



La couleur secondaire du site est un blanc pur : #ffffff



### Le logo

Pour le logo, il fallait quelque chose de simple et facilement reconnaissable, j’ai donc fait un cercle noir avec un contour blanc sur lequel j’ai placé une paire de ciseaux en blanc.

A white logo with a spoon and knife in a black circle

Description automatically generated

## Le Wireframe

Un wireframe est une esquisse ou un schéma visuel qui représente la structure et la disposition d'une page web ou d'une application. Il s'agit d'un outil de conception essentiel utilisé au début du processus de développement pour définir la structure et le contenu des pages, sans se concentrer sur les détails visuels ou graphiques.

Voici les wireframes de l’application Uber Cut :

A sketch of a computer screen

Description automatically generated with medium confidence

## Maquettage

Le maquettage est une étape du processus de conception d'un site web et d'une application. Contrairement aux wireframes, qui se concentrent sur la structure et la disposition, les maquettes fournissent une représentation visuelle détaillée du produit final, incluant les couleurs, la typographie, les images et les éléments graphiques.

Pour Uber Cut, les maquettes garantiront une interface utilisateur intuitive et esthétiquement plaisante, offrant une expérience utilisateur optimale sur tous les appareils.

Voici la maquette de l’application Uber Cut :

A screenshot of a computer

Description automatically generated

# Conception de la base de données

Structurer la base de données d'une application est un processus complexe mais essentiel pour éviter un développement à l'aveugle. En effet, cette structuration permet de comprendre les données que l'application utilisera. Elle aide à modéliser et représenter les entités, les relations et les attributs clés nécessaires au bon fonctionnement de l'application.

Pour cette étude, j'ai utilisé la méthode Merise, une approche de conception et de modélisation des systèmes d'information. La méthode Merise comprend plusieurs étapes clés : l'analyse des besoins, la modélisation conceptuelle des données (MCD), la modélisation logique des données (MLD) et la modélisation physique des données (MPD). Grâce à cette méthode, il est possible de concevoir des bases de données de manière rigoureuse et structurée.

## Modèle Conceptuel de Données (MCD)

Le MCD, ou Modèle Conceptuel de Données, est une représentation abstraite des concepts et des relations entre les entités d'un système d'information. Ce schéma conceptuel décrit les principales entités, leurs attributs et les associations entre elles, tout en étant indépendant de toute considération technique ou de mise en œuvre.

Le MCD utilise des symboles graphiques tels que des rectangles pour représenter les entités et des lignes pour représenter les relations entre ces entités. Les attributs des entités sont spécifiés à l'intérieur des rectangles, et les cardinalités des relations sont indiquées avec des notations appropriées.

L'objectif principal du MCD est de fournir une vue claire et abstraite des données du système, en mettant l'accent sur la structure logique et les relations entre les entités. Il permet de comprendre les besoins métier, d'identifier les entités clés, d'organiser les données de manière cohérente et de préparer la modélisation logique des données pour les étapes ultérieures de développement.

Voici le diagramme MCD de l’application Uber Cut :

A diagram of a computer program

Description automatically generated

## Modèle Logique de Données (MLD)

Le MLD, ou Modèle Logique de Données, est, quant à lui, une représentation structurée et formelle des données d’un système d’information.

Il s’agit d’une étape intermédiaire entre le MCD (Mocèle Conceptuel de Données) et le MPD (Modèle Physique de Données).

Voici le MLD de l’application Uber Cut :

A diagram of a computer

Description automatically generated

# Conception de l’application

Pour la conception de Uber Cut, je me suis appuyé sur le langage UML (Unified Modeling Language) pour représenter le fonctionnement de l’application. Le langage UML est un langage graphique standard utilisé pour modéliser et représenter visuellement les systèmes logiciels. Il permet de capturer les aspects essentiels d'un système, de sa structure à son comportement, en utilisant une notation graphique compréhensible par les développeurs, les concepteurs et les parties prenantes.

UML offre un ensemble de diagrammes qui permettent de décrire différents aspects d'un système, tels que les cas d'utilisation, les classes, les séquences, les activités, les composants, les déploiements, etc. Ces diagrammes servent de supports de communication pour analyser, concevoir, documenter et visualiser un système logiciel, facilitant ainsi la compréhension et la collaboration entre les membres de l'équipe de développement et les parties prenantes. Le langage UML est largement utilisé dans l'industrie du développement logiciel pour améliorer la modélisation, la conception et la documentation des systèmes complexes.

Parmi les schémas UML que j'ai élaborés pour Uber Cut, deux d'entre eux jouent un rôle essentiel dans la compréhension des fonctionnalités et de la structure de mon système : le diagramme de cas d'utilisation (use case) et le diagramme de classes. Ces schémas UML offrent une vision claire et structurée de mon application, permettant ainsi de visualiser les interactions entre les utilisateurs et le système, ainsi que la structure et les relations entre les différentes entités.

## Diagramme de Cas d'Utilisation (Use Case)

Le diagramme de cas d'utilisation décrit les interactions entre les utilisateurs (clients, coiffeurs, administrateurs) et le système. Il identifie les différentes fonctionnalités offertes par l'application, telles que la création de compte, la réservation de séances de coiffure, la gestion de la disponibilité des coiffeurs, et l'administration de la plateforme.

Voici le diagramme de Use Case de l’application Uber Cut :

A diagram of a diagram

Description automatically generated

# Conception multicouche MVC

Le schéma MVC (Modèle-Vue-Contrôleur) est un concept fondamental en développement logiciel, largement utilisé dans la conception d'applications web et d'autres systèmes interactifs. Il offre une structure claire et organisée pour séparer les différentes responsabilités d'une application et favoriser une meilleure maintenabilité, extensibilité et réutilisabilité du code. Voici une présentation des composants de ce schéma dans le cadre du Framework Symfony.

## Le Modèle

Le Modèle représente la couche responsable de la gestion des données et de la logique métier. Il est chargé d'interagir avec la base de données, de récupérer et de stocker les informations pertinentes pour l'application Uber Cut.

En utilisant Symfony, j'ai pu bénéficier de l'ORM (Object Relational Mapping) Doctrine, qui facilite la création et la manipulation des entités en se basant sur les modèles de données définis. L'ORM, ici Doctrine, sert d’intermédiaire entre les classes PHP et les tables de la base de données SQL.

Le Modèle dans Symfony se compose donc de mes entités et de leurs relations, ainsi que des classes de gestion des requêtes et des opérations de validation et de persistance des données. Grâce à la clarté et à la flexibilité du Modèle dans Symfony, j'ai pu organiser mes données de manière cohérente et les manipuler de manière efficace, en garantissant la stabilité et l'intégrité de mon application.

Voici l’organisation des Entités dans le projet.

A screenshot of a computer

Description automatically generated

Exemple avec l’entité User Cette entité est divisée en 3 parties, la déclaration des variables de l’entité, les getters et les setters :

A screen shot of a computer

Description automatically generated

Exemples de getters et setters de l’entité User :

A screen shot of a computer program

Description automatically generated

## 

## La Vue

La Vue est responsable de la présentation des informations aux utilisateurs et de l'interaction avec eux. Grâce au moteur de templates Twig fourni par Symfony, j'ai pu concevoir des vues dynamiques en utilisant des modèles réutilisables et des composants.

Les vues dans Symfony sont construites à partir des données fournies par le Contrôleur et sont responsables de l'affichage des résultats de manière structurée et conviviale. J'ai pu créer des interfaces utilisateur esthétiques et réactives en utilisant les fonctionnalités de Twig telles que les boucles, les conditions et les filtres. De plus, Symfony offre des outils pratiques pour gérer les formulaires et les interactions utilisateur, ce qui facilite la collecte de données et les actions de l'utilisateur au sein de l'application.

Voici l’organisation des fichiers Twig :

A screenshot of a computer

Description automatically generated

Pour cet exemple, voici le fichier Twig responsable de l’affichage de la page login (connexion) :

A screen shot of a computer program

Description automatically generated

Le code présenté est un template Twig pour une page de connexion (login.html.twig) dans une application Symfony. Il étend un template de base (base.html.twig) et utilise les fonctionnalités de Twig pour créer une interface utilisateur pour la connexion.

Dans ce cas précis, l’utilisation du `{% extends ‘base.html.twig’ %}` permet d’hériter de la barre de navigation ainsi qu’à toutes les librairies CSS et JS permettant le bon fonctionnement de toutes les pages de l’application.

## Le Contrôleur

Le Contrôleur joue un rôle crucial dans la gestion des actions de l'utilisateur et l'orchestration des interactions entre le Modèle et la Vue. Dans mon projet Symfony, le Contrôleur agit comme un intermédiaire entre les requêtes entrantes et les opérations à effectuer.

Le rôle du contrôleur est de recevoir les données saisies par l'utilisateur sous forme de requête et les transmet au Modèle pour effectuer des opérations telles que la création, la modification ou la suppression de données. Une fois que le Modèle a effectué les opérations nécessaires, le Contrôleur met à jour la Vue pour refléter les changements et renvoie la réponse appropriée à l'utilisateur.

Symfony fournit un système de routage qui associe les URL aux actions des Contrôleurs, facilitant ainsi la gestion des différentes fonctionnalités de l'application. Chaque action d'un contrôleur correspond à une méthode qui traite une requête spécifique.

Voici l’organisation des contrôleurs de l’application :

A screenshot of a computer program

Description automatically generated

Pour cet exemple, voici le contrôleur de la page Dashboard destiné aux coiffeurs :

A screen shot of a computer program

Description automatically generatedIl est important de noter que les erreurs rouges du code ne sont que des bugs de lecture de mon IDE, la déclaration ligne 5 ainsi que les utilisations de la fonction getCutterStatus & setCutterStatus fonctionnent bien.

## Communication entre les 3 composants

## L’architecture 3 tiers

# Sécurité

## Les attaques XSS (Cross-Script Scripting)

## Les injections SQL

## Les attaques CSRF (Cross-Site Request Forgery)

# Politique de test

## Les tests unitaires

## Les tests d’intégration

# Veille Technologique

## Veille globale

## Veille sécurité

# Difficultés rencontrées

# Conclusion

# Remerciments

# Annexe

## Cahier des charges