

Conception d'un site web de base pour la gestion d'un garage automobile

L'objectif de ce projet est le développement d'une plate forme très simpliste pour la gestion d'un garage automobile avec le service de base de prises de rdv pour diverses interventions. On proposera un modèle sous forme d'un site web dynamique.

Un garage est composé d'un directeur d'atelier, de mécaniciens et d'agents d'accueil. Le garage offre aux clients la possibilité de payer les interventions en différé, c'est à dire plus tard (cf. paragraphe gestion financière).

Votre site proposera une page d'accueil qui demande la saisie de l'identifiant et du mot de passe de l'employé. Une fois ces informations validées, une page dépendant de la catégorie de l'employé s'affiche (la page ne dépend pas de l'identité de l'employé, mais de sa catégorie). On aura donc trois interfaces différentes : agent, mécanicien et directeur.

Pages web des agents

Un agent d'accueil peut effectuer les opérations suivantes :

- saisir et modifier les informations du client : nom, prénom, adresse, num de tél, mail, etc.
- consulter la synthèse d'un client,
- effectuer le paiement d'une intervention cash, ou la mettre en différé,
- effectuer le remboursement d'un ou plusieurs différés,
- prendre des rdv avec un mécanicien pour une intervention X.

Pour la gestion financière : lorsque le client se présente à la caisse, on saisi son identifiant et on clique sur valider. La dernière intervention « en attente de paiement » ainsi que toutes celles « en différé » s'affichent et sont chacune précédées d'un checkbox. Le client a le choix entre :

- Payer la dernière intervention cash. Elle passe alors de l'état « en attente de paiement » à « payée ». L'agent encaisse le client et valide le paiement.
- Demander un différé pour la dernière intervention. Elle passe alors de l'état « en attente de paiement » à « en différé ».
- Rembourser le différé d'une ou plusieurs interventions. Pour cela, on coche les checkbox des interventions et on valide tous le paiements d'un coup (et non pas paiement par paiement). Elles passent alors de l'état « en différé » à l'état « payée ».

Chaque client dispose d'un montant maxi de différé autorisé (éventuellement nul) qui est fixé par l'agent d'accueil. Ce montant peut varier dans le temps et c'est à la charge de l'agent de gérer sa valeur selon le sérieux de son client. Attention, un client ne peut pas obtenir de différé si le paiement d'une intervention en différé lui fait dépasser son maximum autorisé. Ainsi si le maxi autorisé de Mr Dupont est de 1000 euros et qu'il a déjà 3 interventions de 300 euros en différé, alors aucune intervention supplémentaire dont le montant est supérieur à 100 euros ne pourra être mise en différé.

Pour la synthèse client : il faut insérer l'id client puis appuyer sur le bouton « synthèse client ». On affichera alors l'identité complète du client, la liste des interventions réalisées, le nom du mécanicien qui a réalisée chaque intervention, le montant de chaque intervention ainsi que son état : « en attente de paiement », « payée », « en différé ». On affiche également le montant total des différés en cours ainsi que le montant disponible pour des différés ultérieurs. Sur l'exemple ci-dessus on affichera 900 euros comme montant total de différé et 100 euros comme montant

disponible pour d'autres différés. Pour simplifier on suppose que chaque intervention a un code, un nom (vidange, forfait plaquette,...), un montant, et une date.

L'agent d'accueil doit également disposer d'une interface qui permet de retrouver l'id d'un client à partir de son nom + sa date de naissance. Attention à la gestion des homonymes.

Pour la gestion des rdv : un agent d'accueil choisit un mécanicien « X », affiche le planning de « X » sur une semaine « Y ». Cette affichage montrera les plages libres et les plages occupées de X durant la semaine Y. L'agent propose alors une date et une heure libre (on fonctionne par tranche horaire d'une heure). Une fois que le client a approuvé la date et l'heure du rdv, l'agent doit le sauvegarder dans son interface et préciser le nom de l'intervention (vidange, forfait plaquette, etc). Toutes les interventions possibles sont chargées dynamiquement dans une liste déroulante, ce qui permet à l'agent de simplement cliquer dessus pour la choisir. Le site web affiche alors la liste des éléments à apporter pour le jour de l'intervention (écrou anti vol, code alarme, etc). C'est le directeur qui décide de la liste des éléments à fournir pour chaque intervention. Il disposera d'une interface web qui lui permet de noter pour chaque intervention la liste des éléments que le client doit amener avec lui le jour du rdv. Pour simplifier le projet on considère qu'un rdv contient une et une seule intervention. Ainsi pour effectuer deux interventions consécutives, il faut prendre deux rdv consécutifs.

Une fois que l'agent a validé sa saisie de rdv, l'état de la facture de ce rdv passe en « attente de paiement ». Le jour du rdv, le client se présente alors au garage, effectue sa réparation et repasse à la caisse pour définir son mode de paiement qui fera passer l'état de la facture vers l'état «payé » ou « en différé ».

Pages web des mécaniciens

Un mécanicien doit disposer d'une page pour visualiser le planning de la journée de n'importe quel mécanicien. Il peut également visualiser le planning sur une date X. D'autre part, un mécanicien Y doit pouvoir bloquer certains créneaux horaires afin d'accomplir des formations. Un agent d'accueil ne pourra donc pas prendre de rdv à un client avec le mécanicien Y dans ces créneaux horaires.

Lorsque le mécanicien visualise son planning du jour, il doit pouvoir cliquer sur chaque rdv et visualiser la synthèse du client ainsi que l'objet du rdv et la liste des éléments nécessaires pour ce rdv.

Pages web du directeur

Le directeur doit disposer de plusieurs interfaces pour :

- créer, supprimer et modifier les logins et mot de passe pour les trois interfaces : agent, mécanicien et directeur.
- créer, supprimer et modifier la liste des interventions possibles que propose la garage. C'est à dire que s'il décide que son garage peut réaliser des « changement de distribution » alors il l'ajoute à la base, ce qui permettra aux agents d'accueil de le visualiser dans leur liste déroulante d'interventions possibles lors d'une prise de rdv.
- créer, supprimer et modifier la liste des éléments à fournir par le client pour chaque intervention. C'est à dire que s'il décide d'exiger le code de l'alarme et la carte grise pour changer un système d'anti-démarrage codé, alors l'agent d'accueil visualisera ceci lors de l'affichage des éléments à fournir sur une prise d'un rdv pour une intervention « changement système anti-démarrage».

Travail à réaliser par trinômes

- Proposez un site web dynamique en utilisant OBLIGATOIREMENT une architecture MVC et les langages suivants : HTML5, CSS3, PHP, PDO et RIEN D'AUTRES.
- Avant de commencer la programmation, définissez toutes les tables de la base de données. Pour cela, réfléchissez bien à toutes les opérations nécessaires au fonctionnement du site web, et comment ces opérations vont lire ou modifier les tables de la base de données.
- Une présentation CSS simple et efficace : de belles pages web aérées et agréables à consulter. De beaux formulaires clairs et propres. Ne surchargez pas votre CSS.
- Soignez les contrôles Php : dépassement du montant différé max lors du paiement d'une intervention, ouvrir deux fois le même compte pour un client...etc. Proposez un maximum de contrôles en Php pour que votre modèle soit le plus proche possible de la réalité.
- Proposez une architecture MVC propre, claire et lisible.
- Les mini soutenances se feront la première ou deuxième semaine de janvier (date à préciser plus tard).
- Amenez le jour de votre soutenance un descriptif de toutes vos tables expliquant le rôle de chaque table dans votre modélisation.
- Remplissez vos tables avec des informations avant le jour de la soutenance.

Notation des trinômes

Css : 1

Contrôles Php : 1

Architecture Php : 2

L'agent se connecte et :

crée un client avec un découvert : 0.25

recherche un client : 0.25

modifie directement un client une fois la recherche effectuée : 0.25

visualise le planning d'un mécanicien : 1

essaye de prendre un rdv sur un créneau où le mécanicien a déjà une intervention avec un autre client : 1

essaye de prendre un rdv sur un créneau non disponible cause formation du mécanicien : 1

sauvegarde un rdv sur un créneau disponible avec visualisation automatique d'une liste déroulante d'interventions possibles et une fois la sauvegarde validée affichage des pièces à apporter : 4

visualise le rdv (pris dans le paragraphe ci-dessus) dans le planning du mécanicien : 1

visualise les interventions (dont le rdv pris dans le paragraphe ci-dessus) en attente de paiement : 1

fait payer un client et ce paiement s'affiche dans la synthèse client : 0.5

essaye de mettre en différé une intervention mais visualise un blocage de la mise en différé

car son montant dépasse le montant autorisé restant : 1
met en différé une intervention d'un client dont le montant ne dépasse pas le montant autorisé restant : 1
augmente ou baisse le montant de découvert d'un client : 0.25
affiche le montant de différé encore possible à effectuer pour un client X avant et après avoir payer une intervention : 1

Le mécanicien se connecte et :

visualise son planning sur une date : 0.25
bloque un créneau pour formation : 0.5
visualise la synthèse d'un client et l'objet d'un rdv en cliquant sur le planning : 0.5

Le directeur se connecte et :

crée/modifie une intervention avec un prix et les pièces à fournir et prise en compte de ses informations lors de la prise de rdv et paiement de l'intervention: 1
modifie les logins et mots de passe : 0.25

Chaque trinôme doit donc équitablement se répartir le travail entre Html/CSS/Php et des questions précises sur le code source seront posées en détail à chaque étudiant sur les parties dont il a été le concepteur. Amenez également avec vous le jour de la soutenance un document (une page recto/verso maxi) expliquant avec détail et précision « qui a fait quoi dans le projet ». L'architecture MVC vous facilitera le développement dans le sens où chacun développe ses propres fonctions.