1.Model de données :

(a) Tables

1. Une table d’utilisateurs « user » avec une clé primaire auto-incrémentée « id », et deux colonnes « name » et « password » contenant du texte.
2. Une table de la location du parking «parkingLocation » avec une clé primaire « parkingName », et deux colonnes « numberofFloors » et « capacity » contenant des nombres entiers .
3. Une table de places réservées « reservedSlots » avec une clé primaires auto-incrémentée « reservation\_Id » contenant du texte et cinq colonnes « userId » qui contient le id du client a qui est la réservation, « parkingName » et « SlotNumber » qui contiennent le nom du parking et la place réserver dans le parking, « dateStart » la date de début de la réservation et « dateEnd » la date de fin de la réservation.

(b) Fonctions d’accès

1. login(user, password) : retourne la clé id associée à l’utilisateur dans la table user si l’utilisateur existe et si le mot de passe correspond à celui enregistré ; retourne -1 sinon.
2. user\_exists(user) : retourne true si l’utilisateur existe et false sinon.
3. new\_user(user, password) : si l’utilisateur existe déjà retourne -1 ; sinon ajoute un utilisateur et retourne la clé id associée dans la table user.
4. get\_reservations(parkingName, floorNumber , dateStart , dateEnd) : retourne un tableau de toute les réservations effectuer en fonction du cadre de temps et de la location du parking misent en paramètre .
5. add\_reservation (parkingName, floorNumber , dateStart , dateEnd): appel valid\_reservation et ajoute la réservation à la table reservedSLots si vrai en retournant l’Id de réservation « reservation\_Id » . Ne modifie pas la table et retourne « -1 » si une réservation ne peut pas être effectuer (en fonction des parametres).
6. delete\_reservation(resevation\_Id) : retourne “true” si la réservation a été enlever et « false » sinon.
7. valid\_reservation(parkingName, floorNumber , dateStart , dateEnd) : retourne « true » si la réservation peut être faite à la date et la location misent et retourne « false » sinon.
8. reservation\_ended() : supprime les réservations où la date de fin des réservations a dépassé le temps actuelle.
9. Modifie\_reservation(resevation\_Id, parkingName, floorNumber , dateStart , dateEnd) : permet de modifier le temps et la place de la réservation faites par l’utilisateur.

2.Vues

(a) Un en-tête header, utilisé sur toutes les pages, avec notamment une barre de navigation. Si l’utilisateur est connecté à son compte personnel, la barre de navigation affiche un lien vers les réservations effectuer (et valide) par l’utilisateur, un lien pour faire d’autres réservations et un lien pour se déconnecter de son compte. Si l’utilisateur n’est pas connecté, la barre de navigation affiche un lien pour se connecter à un compte existant et un lien pour créer un nouveau compte.

(b) Un bas de page footer, utilisé sur toutes les pages

(c) Une page d’accueil index, affichant un message de bienvenue et une image

(d) Une page de connection login, contenant un formulaire pour se connecter à l’aide de son nom d’utilisateur et de son mot de passe

(e) Une page de création de compte « new\_user », affichant un formulaire permettant de choisir un nom d’utilisateur et un mot de passe et optionnellement un message pour signaler que le nom d’utilisateur choisi est déjà pris

(f) Une page de connection « login », contenant un formulaire pour se connecter à l’aide de son nom d’utilisateur et de son mot de passe

(g) Une page « reservation », pour afficher la demande de réservation d’un utilisateur connectée. Elle affiche :

1. Un Drop down menu qui permet de choisir la location. Un calendrier pour choisir le jour et un formulaire pour choisir l’intervalle d’heure pendant lequel l’utilisateur souhaite faire la demande.
2. Un bouton « place reservation » qui permet de placer la réservation aux critères choisies.
3. Un bouton qui affiche la liste des heures de réservations effectuées au jour choisi par l’utilisateur pour lui faciliter de trouver une place (temps) libre.

(h) Une page « modifie\_reservation » contenant un formulaire contenant les détails de la réservation fait par l’utilisateur qui lui permet de de modifier sa réservation.

(h) Une page « user\_reservation » pour afficher les réservations faites par l’utilisateur avec l’id de réservation où pour chaque réservation un bouton pour supprimer la réservation.

3.Routes et logiques de contrôle :

(a) Sur la route GET / on affiche la page d’index

(b) Sur les routes GET /login, GET /new\_user on affiche respectivement les pages de connection et de création de compte

(c) La route POST /login essaye de connecter l’utilisateur à partir de l’identifiant et du mot de passe fournis et redirige l’utilisateur vers son profil personnel si la connection réussit (c’est à dire si l’identifiant et le mot de passe correspondent à un utilisateur enregistré sur le site) et vers la route GET /login sinon

(d) La route POST /new\_user essaye de créer un nouveau compte avec l’identifiant et le mot de passe fournis et connecte et redirige l’utilisateur vers son profil personnel si la création réussit (c’est à dire si le nom d’utilisateur n’est pas déjà pris) et vers la route GET /new\_user sinon, avec le message « User already exists ».

(e) La route GET /logout déconnecte l’utilisateur et le redirige vers la route GET /

(f) La route GET /resrvation/ :user\_name n’est accessible qu’aux utilisateurs connectés à leur compte personnel. Selon l’utilisateur connecté et le paramètre de la route elle mène à la vue user\_reservation.

(g) La route GET /user\_reservation n’est accessible qu’aux utilisateurs connectés à leur compte personnel. Elle mène à la vue user\_reservation.

(h) La route GET/modifie\_resrvation n’est accessible qu’aux utilisateurs connectés à leur compte personnel. Elle mène à la vue modifie\_reservation.

(h) La route POST /reservation n’est accessible qu’aux utilisateurs connectés à leur compte personnel. Elle essaie d’enregistrer la demande de réservation de l’utilisateur. Si la demande est effectuer elle redirige l’utilisateur vers la vue user\_reservation . Si la réservation ne peux pas être effectuer elle envoie la liste des réservations faites au jour et location choisient par l’utilisateur pour lui faciliter le choix. C’est à dire l’utilisateur pourra retenter à placer la réservation avec différents critères (temps, location).

(i) la route POST/modifie\_reservation n’est accessible qu’aux utilisateurs connectés à leur compte personnel. Elle sert à modifier la réservation faite par l’utilisateur et redirige vers user\_reservation.