# LunchBuddy

Table des matières

[LunchBuddy 1](#_Toc404338701)

[Préparatifs 2](#_Toc404338702)

[Organisation 2](#_Toc404338703)

[Objectifs 2](#_Toc404338704)

[Analyse 2](#_Toc404338705)

[Maquette préliminaire 2](#_Toc404338706)

[Stratégie de tests 4](#_Toc404338707)

[Planification détaillée 4](#_Toc404338708)

[Conception 4](#_Toc404338709)

[Charte graphique 4](#_Toc404338710)

[Maquette graphique 4](#_Toc404338711)

[Conception base de donnée 4](#_Toc404338712)

[Interrogation 4](#_Toc404338713)

[La vie du projet 4](#_Toc404338714)

[Conclusion 4](#_Toc404338715)

[Annexes 4](#_Toc404338716)

## Préparatifs

### Organisation

Élève 1 : Mendez, Grégory, [gregory.mendez.rodriguez@gmail.com](mailto:gregory.mendez.rodriguez@gmail.com)

Élève 2 : Gindre, Ludovic, [lu.gindre@gmail.com](mailto:lu.gindre@gmail.com)

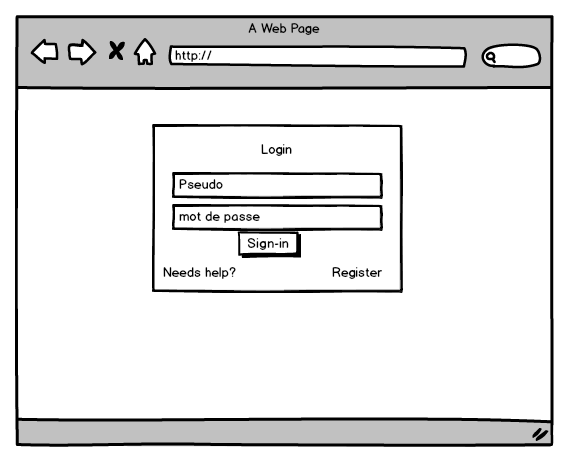
### Objectifs

* L’utilisateur peut se connecter au site
* L’utilisateur peut s’inscrire
* L’utilisateur peut modifier ses informations personnelles
* L’utilisateur voit sa position sur la carte
* L’utilisateur voit les personnes dans un rayon donné disponible le jour-même
* L’utilisateur peut donner un rendez-vous à une autre personne
* L’utilisateur peut visualiser les rendez-vous

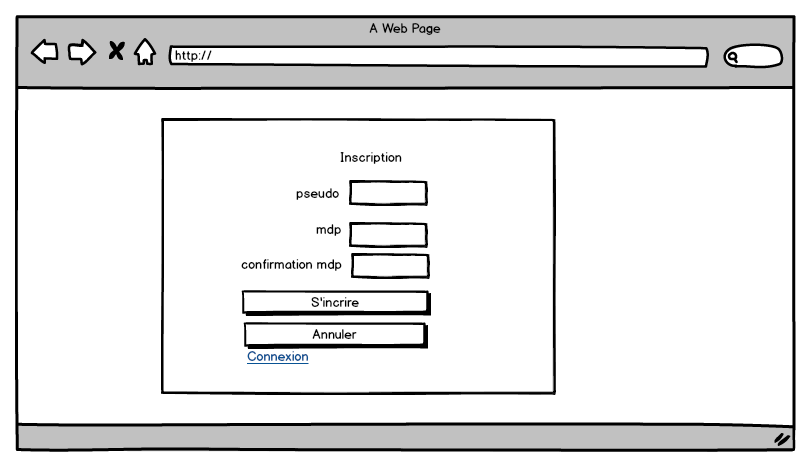
## Analyse

### Maquette préliminaire

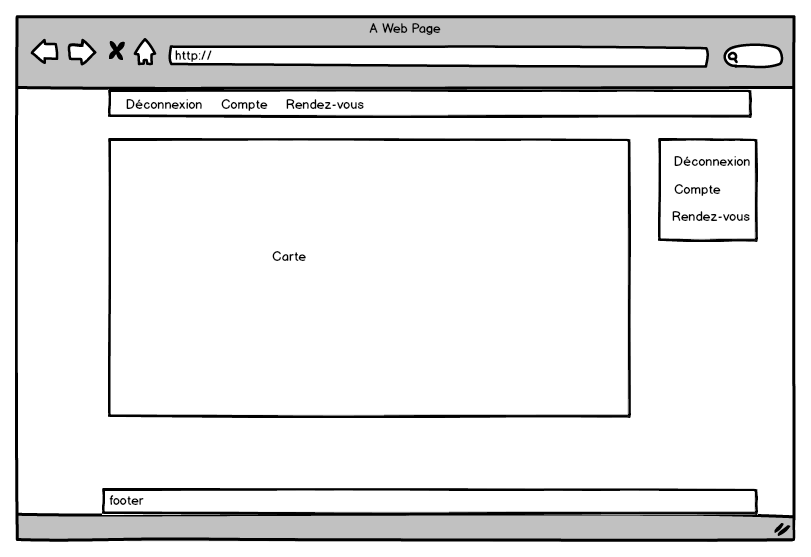
Login



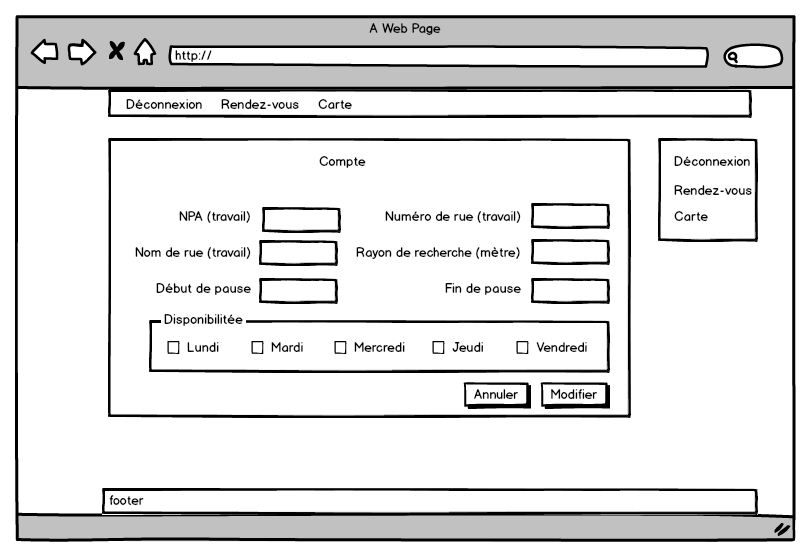
Inscription



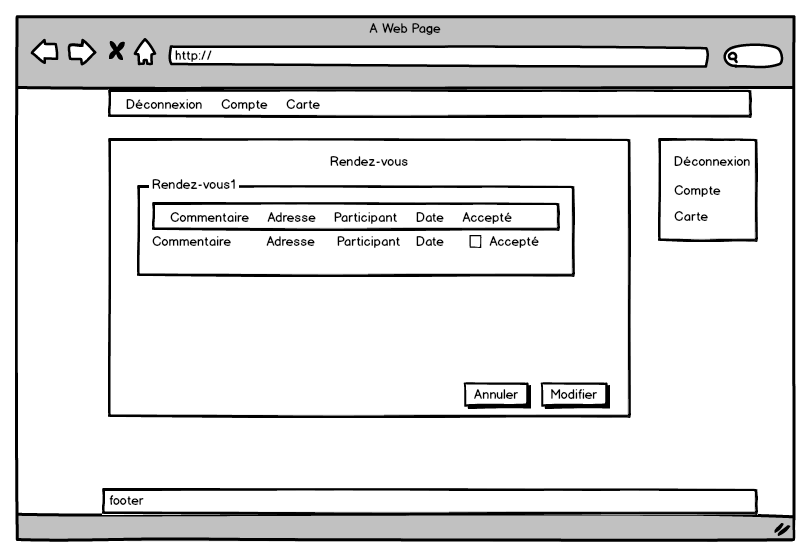
Carte



Compte



Rendez-vous



### Stratégie de tests

1. Inscription d’un utilisateur
   1. L’utilisateur doit à présent être inscrit et connecté au site
2. Connexion
   1. Si l’utilisateur entre ses informations et qu’elles sont correctes, il est connecté et peut accéder au site
   2. Si l’utilisateur entre des informations erronées, il n’est pas connecté et n’a donc pas accès au site
3. Accueil
   1. la carte doit apparaître.
4. paramètres
   1. les paramètres doivent apparaître dans les champs
   2. la validation gère les erreurs de saisie.
   3. les données sont enregistrée quand elle sont correctes et la saisie est redemandée le cas contraire.
5. créer un rendez-vous
   1. le rendez-vous est bien associé à la personne souhaitée.
6. Afficher les rendez-vous de l’utilisateur
   1. Les rendez-vous sont affichés et leurs informations sont correctes.
7. Editer un rendez-vous
   1. Les informations sont affichées et éditables.
   2. L’utilisateur peut changer les informations du rendez-vous et son statut.
8. Supprimer un rendez-vous
   1. Une confirmation est demandée
   2. Si l’utilisateur confirme, le rendez-vous est effectivement supprimé.
   3. Le cas contraire le rendez-vous est conservé.
9. Déconnexion
   1. L’utilisateur est déconnecté , il n’a plus accès au site

### Planification détaillée



## Conception

### Charte graphique

Barre de navigation

* Couleur du texte : #9d9
* d9d
* Police : Helvetica Neue,Helvetica,Arial,sans-serif
* Couleur de fond : #333

Menu latéral

* Couleur du texte : #505f70
* Police : Helvetica Neue,Helvetica,Arial,sans-serif
* Couleur de fond : #333

Contenu

1. titre

* Couleur du texte : #505f70
* Police : Helvetica Neue,Helvetica,Arial,sans-serif

1. Contenu

* Couleur du texte : #505f70
* Police : Helvetica Neue,Helvetica,Arial,sans-serif

### Conception base de donnée

Lunchbuddy

* utilisateurs :
  + idUtilisateur int Non null Clé primaire
  + nom varchar(25) Non null
  + prenom varchar(25) Non null
  + email varchar(25) Non null
  + mdp varchar(25) Non null
  + NPA varchar(4) Non null
  + nomRue varchar(25) Non null
  + numeroRue varchar(25) Non null
  + rayon integer Non null
  + lat decimal(65,30) null
  + lng decimal(65,30) null
  + debutPause time Non null
  + finPause time Non null
* groupes :
  + idGroupe int Non null Clé primaire
  + nom varchar(25) Non null
* jours :
  + idJour int Non null Clé primaire
  + nomJour varchar(25) Non null
* disponible :
  + idUtilisateur int Non null Clé primaire
  + idJour int Non null Clé primaire
* statuts :
  + idStatut int Non null Clé primaire
  + nomStatut varchar(25) Non null
* rendezvous :
  + idRdv int(11) Non null Clé primaire
  + date Rdv datetime Non null
  + commentaire longtext Non null
  + idGroupe int(11) Non null
* composer :
  + idUtilisateur int(11) Non null Clé primaire
  + idGroupe int(11) Non null Clé primaire
  + idStatut int(11) Non null Clé primaire

### Interrogation

* SELECT \* FROM Utilisateurs WHERE idUtilisateur = :idUtilisateur
* SELECT idUtilisateur, nom, prenom FROM Utilisateurs WHERE email = :email
* SELECT idUtilisateur, rayon FROM utilisateurs WHERE password = :password and email = :email
* insert into utilisateurs (email, password, nom, prenom) values (:email, :password, :nom, :prenom)
* SELECT nomRue, numeroRue, NPA, rayon, debutPause, finPause FROM utilisateurs WHERE idUtilisateur = :idUtilisateur
* update utilisateurs set nomRue = :adresse, numeroRue = :numeroRue, NPA = :NPA, rayon = :rayon, debutPause = :debutDispo, finPause = :finDispo where idUtilisateur = :idUtilisateur;
* select idUtilisateur from utilisateurs where email = :email
* DELETE FROM disponible WHERE idUtilisateur = :idUtilisateur
* INSERT INTO `disponible`(`idUtilisateur`, `idJour`) VALUES (:idUtilisateur,:idJour)
* SELECT idJour FROM disponible WHERE idUtilisateur = :idUtilisateur
* SELECT lat, long, rayon FROM utilisateurs WHERE idUtilisateur = :id
* SELECT idUtilisateur, nom, prenom, email, NPA, nomRue, numeroRue, lat, lng, debutPause, finPause FROM utilisateurs AS u NATURAL JOIN disponible AS d WHERE d.idJour = :jour
* insert into groupes (nom) values (:nomGroupe)
* SELECT idGroupe FROM rendezvous WHERE idRdv = :id
* SELECT r.idRdv, r.dateRdv, r.commentaire, r.idGroupe, s.nomStatut FROM rendezvous as r

join groupes as g on r.idGroupe = g.idGroupe

join composer as c on c.idGroupe = g.idGroupe

join statuts as s on c.idStatut = s.idStatut

where c.idUtilisateur = :idUtilisateur

* SELECT r.commentaire, s.idStatut, s.nomStatut FROM rendezvous as r

join groupes as g on r.idGroupe = g.idGroupe

join composer as c on c.idGroupe = g.idGroupe

join statuts as s on c.idStatut = s.idStatut

where c.idUtilisateur = :idUtilisateur

and r.idRdv = :idRdv

* SELECT s.nomStatut FROM statuts as s

join composer as c on c.idStatut = s.idStatut

WHERE c.idGroupe = :idGroupe

and c.idutilisateur = :idUtilisateur

* INSERT INTO `composer`(`idUtilisateur`, `idGroupe`, `idStatut`) VALUES (:idUtilisateur, :idGroupe, :idStatut)
* UPDATE composer SET idStatut = :statut WHERE idUtilisateur = :idUtilisateur
* UPDATE `rendezvous` SET commentaire = :commentaire WHERE idRdv = :idRdv
* insert into rendezvous (dateRdv, commentaire, idGroupe) values (:date, :commentaire, :idGroupe)
* DELETE FROM rendezvous WHERE idRdv = :idRdv
* DELETE FROM composer WHERE idUtilisateur= :idUtilisateur AND idGroupe = :idGroupe
* DELETE FROM groupes WHERE idGroupe = :idGroupe
* DELETE FROM composer WHERE idGroupe = :idGroupe
* select u.idUtilisateur, u.nom, u.prenom, u.email from composer as c join utilisateurs as u on c.idUtilisateur = u.idUtilisateur where c.idGroupe = :idGroupe

## La vie du projet

14.11.2014

* Création du MCD
* Création de la base de donnée
* Création de la page de login

21.11.2014

* Création page Inscription
* Création fonction connexion
* Création fonction inscription (pas fini)

28.11.2014

* Création fonction d’inscription
* Création page d’accueil
* Affichage de la carte google maps(Javascript)

05.12.2014

* Fonction modifier les infos de l’utilisateur
* Création de la page paramètre
* Récupération de la position de l’utilisateur

11.12.2012

* Récupération de la position des autres utilisateurs disponible le jour-même

Passage dans la récupération de la position de l’utilisateur connecté ne s’effectue pas à chaque chargement de la page ⇒ problème non résolus

12.12.2014

* Création fonction suppression des informations du compte
* Création fonction création des groupes
* Création fonction création des rendez-vous

18.12.2014

* Création fonction création des groupes
* Création page affiche rendez-vous
* Création page supprimer rendez-vous
* Fonction affichage des rendez-vous
* Fonction création groupe
* Fonction modification groupe
* Fonction suppression groupe

19.12.2014

* Création page modifier rendez-vous
* Fonction lecture groupe

## Conclusion

Au début du projet après analyse, nous nous sommes fixés des objectifs. Au terme de notre projet, nos objectifs sont tous atteints.

Le projet nous a permis d’approfondir nos connaissances dans le web. Il nous a aidé à comprendre et à utiliser la gestion de projet. En revanche, la quantité de travail était très importante et l’énoncé très vague.

Durant le projet, nous avons rencontrés certains problèmes. Certains sont réglés, d’autres font parties des suites possibles pour le projet. Parmi ces difficultés rencontrées il y a eu le passage des données du php dans le javascript, la gestion des interactions entre des utilisateurs et enfin les différents façons d’aborder le sujet du fait de l’énoncé.

Dès le début de notre projet nous avons pensé qu’il pourrait très bien être appliqué à une utilisation publique. C’est pourquoi il pourrait y avoir de nombreuses améliorations possibles. Tout d’abord nous pourrions gérer des groupes de personne et pas seulement deux personnes.

## Annexes

<http://www.w3schools.com/>

<http://www.mapanet.eu/fr/Resources/Script-Distance.htm>

<http://www.w3schools.com/googleAPI/>

<http://openclassrooms.com/courses/google-maps-javascript-api-v3>

Mme Travjnak (Ajax, Géolocalisation)