

# Projet R3st0.fr

## Contexte

### Table des matières

<b>I - Présentation .....</b>	<b>2</b>
<b>II - Fonctionnalités .....</b>	<b>2</b>
<b>III - Types de missions .....</b>	<b>3</b>
<b>III.1 Traiter les tickets de maintenance .....</b>	<b>3</b>
<b>III.2 Ajouter des fonctionnalités.....</b>	<b>3</b>
<b>III.3 Déployer l'application sur le serveur .....</b>	<b>3</b>
<b>IV - Méthode de gestion de projet .....</b>	<b>4</b>
<b>IV.1 Les équipes .....</b>	<b>4</b>
<b>IV.2 Les itérations.....</b>	<b>4</b>
<b>IV.3 La planification .....</b>	<b>4</b>
<b>V - Outils .....</b>	<b>5</b>
<b>V.1 E.D.I. ....</b>	<b>5</b>
<b>V.2 Dépôts de code .....</b>	<b>5</b>
<b>V.3 Planification des tâches.....</b>	<b>5</b>

## I - Présentation

Le site r3st0.fr est construit à l'image des nombreux sites de critique existant sur le marché : la fourchette, tripadvisor, etc.

L'objectif de ce site est de construire une base de données d'un grand nombre de restaurants, de stocker leurs caractéristiques, de recueillir les avis des consommateurs, et de diffuser ces informations aux visiteurs pour leur permettre de mieux choisir où manger.

## II - Fonctionnalités

Deux types d'utilisateurs ont accès au site : les visiteurs, et les utilisateurs authentifiés.

### Visiteur

En tant que visiteur non authentifié, l'utilisateur a accès à :

- un module de recherche lui permettant de trouver des restaurants par nom, adresse, et multi-critères.
- un module d'inscription
- sur la fiche d'un restaurant, aux caractéristiques du restaurant, et aux évaluations laissées par les utilisateurs enregistrés

### Utilisateur authentifié

Un visiteur enregistré a accès à plus de fonctionnalités, il peut ainsi :

- personnaliser son profil,
- aimer des restaurants,
- critiquer des restaurants et noter (de 1 à 5) des restaurants.



## **III - Types de missions**

Vous êtes chargés de faire évoluer l'existant. Chaque mission sera réalisée lors d'une itération d'une ou deux séances.

### **III.1 Traiter les tickets de maintenance**

La résolution de certains incidents ou de petites demandes d'évolutions sur la version initiale vous sont soumises.

### **III.2 Ajouter des fonctionnalités**

Il s'agira de répondre à des demandes d'évolution de l'application, notamment la création d'un accès administrateur pour gérer les contenus du site web.

### **III.3 Déployer l'application sur le serveur**

L'application web et sa base de données seront installées sur le serveur d'application web de la section SIO.

## IV - Méthode de gestion de projet

Les projets doivent être conduits en équipes, suivant une méthode itérative, empruntant quelques éléments de méthodes agiles.

Les maîtres d'œuvre sont les enseignants d'A.P..

### IV.1 Les équipes

Une équipe de projet est constituée de 2 ou 3 membres qui doivent participer à toutes les tâches au cours du projet (gestion de projet, codage, tests, documentation, déploiement, ...).

Un animateur d'équipe (le « **scrum master** ») est choisi. Il est chargé de rappeler le respect de la méthodologie, d'animer les réunions de séances, et de faire le lien avec les maîtres d'œuvre.

### IV.2 Les itérations

Une itération est une étape de courte durée (1 à 2 séances) qui conduit à la fourniture d'un **livrable** : une nouvelle version exécutable est « poussée » sur le dépôt de code de l'équipe et un **compte-rendu d'itération** est produit, qui constituera la documentation technique du projet.

Chaque séance débute par une courte réunion d'équipe (le « **daily scrum** ») permettant de :

- faire le bilan de la séance précédente
- planifier le travail de la séance à venir

La trace de ces « daily - scrum » doit être conservée et être intégrée au compte-rendu d'itération.

Sur son dépôt de code, chaque équipier ouvre une nouvelle branche pour traiter son ticket. Lorsque la tâche est terminée et testée, il pousse sa branche sur le dépôt distant. Il faudra ensuite fusionner cette branche avec la branche principale.

### IV.3 La planification

Les **tâches** doivent être définies et affectées en réunion.

Chaque tâche doit être affectée à un ou plusieurs membres de l'équipe.

Le **statut** de la tâche doit évoluer de « A faire », « En cours » puis à « Terminé »

Elle doit être **priorisée** (Peu prioritaire/Priorité normale/Prioritaire).

Pendant tout le projet, chaque membre de l'équipe est responsable de ses tâches et de leur **avancement**.

## V - Outils

### V.1 E.D.I.

NetBeans v.17 ou plus

### V.2 Dépôts de code

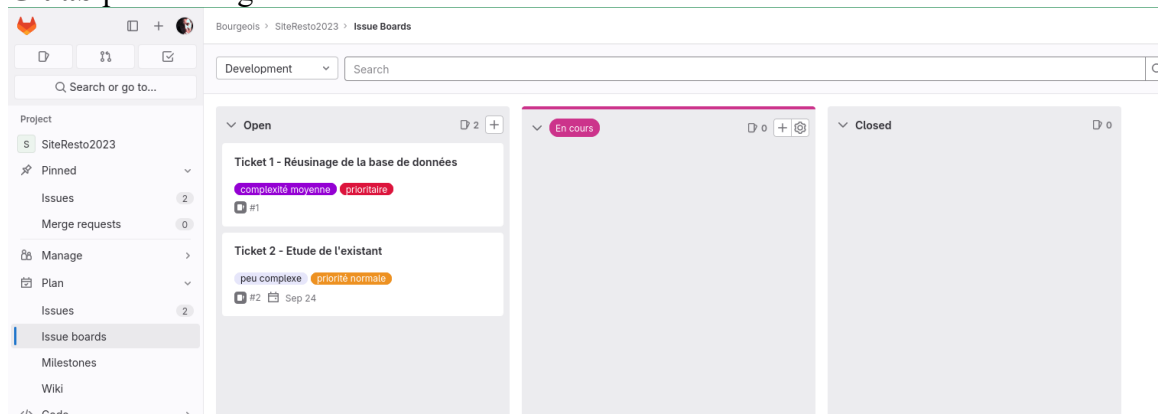
Utiliser **gitlab.com** à synchroniser avec le dépôt local de chaque équipier par l'intermédiaire de NetBeans.

### V.3 Planification des tâches

L'équipe de développement partagera un tableau Kanban qui contiendra au moins les éléments suivants :

- trois colonnes : « A faire » (Open), « En cours », « Terminé » (Closed) ;
- trois labels de complexité : « peu complexe », « complexité normale », « complexe » ;
- trois labels de priorité : « peu prioritaire », « priorité normale », « prioritaire »

**Gitlab** permet de gérer un tel tableau : Plan / Issue boards

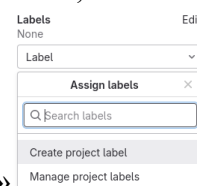


**Présentation des éléments :**

- « Issue » : c'est un ticket l'équipe de développement ; tâche de maintenance ou d'évolution de l'application ; un ticket en cours est affecté à un ou plusieurs membres de l'équipe.
- « Label » (étiquette de couleur) : caractérise un ticket et/ou une colonne du tableau (« list »). Les « labels » sont à définir avant les « issues ».

**Pour créer :**

- un nouveau label : menu Manage / Labels , puis bouton « New label », ou bien « Create project



label » colonne de droite sur l'écran de définition d'une « issue »

- une nouvelle colonne bouton « Create list » en haut à droite de la page « Issue boards » ;
- un nouveau ticket : « New Issue » en haut à droite de la page « Issues », ou bien le bouton « + » en haut de chaque colonne du « Issue board » du projet.

