

*Projet CIR2*

# *AIDE À LA NAVIGATION*

## *Evaluation*

Mis à jour le 01/06/2016

[michael.aron@isen-bretagne.fr](mailto:michael.aron@isen-bretagne.fr)  
[yann.le-ru@isen-bretagne.fr](mailto:yann.le-ru@isen-bretagne.fr)



**ISEN** | école  
d'ingénieurs


**Brest**

Projet CIR2  
Description

| 1

## Les deadlines...

Livrables	Limite
Présentation orale : Maquettes Web + Base De Données + algo	23/05 à 08:30
Vérification plateforme de tests	25/05 08:30
MCD + fichier SQL	07/06 17:30
Sources du projet	09/06 08:00



Les livrables seront à poster sur l'intranet  
Tout retard sera sanctionnée (l'heure du réseau fait foi)  
Les fichiers au mauvais format ou mauvais nommage seront pénalisés

Brest



## *Livrable Projet*

- Une archive à déposer sur l'ENT (infos sur la norme à respecter sur le dépôt) contenant :
  - Fichier README indiquant les étapes à effectuées pour que le site fonctionne sur n'importe quel serveur (les informations suivantes doivent être facilement paramétrables : accès à la BDD , URL du site, ...)
  - Fichier SQL contenant la structure de la BDD, ainsi qu'un jeu de données permettant d'effectuer les tests.
  - MCD format PowerAMC
  - Code source :
    - PHP, HTML, Javascript, programme JAVA, scripts shell, ressources (images, ...), serveur temps réel si besoin, ...





## *Informations*

- Les barèmes indiqués dans ce document sont donnés à titre indicatif
- Le but de ce document est d'indiquer les fonctionnalités attendues et de donner un ordre de priorité dans la réalisation des tâches





## Barème prévisionnel général

Partie	Sous-partie	Barème
Réunion client	MCD	2
	Maquette Web	1,5
	Présentation Java	1,5
Test plateforme (monôme)	-	De +0 à -0.5pts
Web	Fonctionnalités (recette)	9
Java	Fonctionnalités (recette)	6
	Total	20



## *Informations Web*

- Se concentrer sur la partie cliente !!!
- Répartition probable :
  - Client : 80%
  - Admin : 20%
- Concernant la partie cliente, la gestion de plusieurs plans (pour des étages par exemple) n'est pas à faire en priorité. La majorité des points pourront être validés sur un plan unique.





## Barème prévisionnel web : client

#	Intitulé		Barème
1	Structure générale	MVC	0
		Ergonomie	
		Navigation	
		Mobile	
		Configuration Appli (BDD, ...)	
2	Localisation	Accéder lecteur QR-code	
		Lecteur QR-code	
		Enchainement lecture QR-code	
		Visualiser position	
		Position sur une carte	
	Total		



## *Barème prévisionnel web : client*

#	Intitulé		Barème
3	Points d'intérêts	Liste des points	
		Détail d'un point d'intérêt	
		Affichage des points sur une carte	
		Gestion des différents plans	
	Total		





## Barème prévisionnel web : client

#	Intitulé		Barème
4	Itinéraire	Choix d'un point d'intérêt (liste ou carte)	
		Confirmation position actuelle ?	
		Enchaînement (logique)	
		Enchaînement (ergonomie)	
		Affichage itinéraire (texte, numéros)	
		Affichage itinéraire (carte, numéros)	
		Affichage itinéraire (carte, tracés) => Bonus	
		Gestion des différents plans	
		Lancement du programme Java	
	Total		



## *Barème prévisionnel web : admin*

#	Intitulé		Barème
1	Plans	Ajout	
		Modification	
		Suppression	
	Total		





## *Barème prévisionnel web : admin*

#	Intitulé		Barème
2	Points d'intérêts	Ajout	
		Modification	
		Suppression	
		Lien avec plan	
		Infos (titre, texte, image, ...)	
		Gestion des voisins	
		Positionnement sur la carte	
		Génération QR-Code	
	Total		





## *Barème prévisionnel web : admin*

#	Intitulé		Barème
3	Authentification	Authentification	
	Total		





## Barème prévisionnel Java

#	Intitulé		Barème
1	Fonctionnalités	Exécution paramétrable (pt entrée et pt de sortie)	
		Récupération du graphe de la BDD	
		Calcul du plus court chemin	
		Ecriture du résultat	
		Accès BDD paramétrable	
	Total		





## *Barème prévisionnel Java*

#	Intitulé		Barème
2	Code	Séparation code BDD/calcul	
		Commentaires compatibles JAVADOC	
		Compilation code source	
		...	
	Total		



# *VALIDATION DES FONCTIONNALITÉS*



## Processus

- Séparation de la validation des différentes parties
  - Interface Web cliente
    - Entrées : informations venant de la BDD (points d'intérêts, cartes, ...) ou de fichiers (programme de routage)
    - Sorties : aucune
  - Programme Java
    - Entrées : paramètres, fichiers, ...
    - Sorties : fichier, BDD
  - Interface Web admin
    - Entrées : informations venant de la BDD
    - Sorties : aucune







## *BDD*

- Un fichier SQL contenant :
  - La structure de la BDD
  - Des données permettant de tester les fonctionnalités de l'application Web cliente et du programme Java (présence de points d'intérêts, de points de passage, ...)





## *Identification des points*

- QR-codes :
  - Imprimer les QR-codes correspondant aux informations présentes dans la BDD



## *Mise en garde !*

- Les informations précédentes sont à lire **attentivement**
- Une fonctionnalité ne pouvant pas être testée aisément ne sera pas validée (pas d'opérations manuelles à effectuer lors de la recette – ajout dans la BDD, modifications dans le code source...)



# *PROCESSUS D'ÉVALUATION*



## *Préparation recette (BDD)*

- **Etape 1** : Les **fichiers sources** doivent être postées sur l'intranet **avant l'heure indiquée** sous format compressé (\*.zip ou \*.rar)
- **Etape 2** : L'enseignant importe le fichier SQL sur un serveur MySQL de test (attention, le nom de la BDD ainsi que l'utilisateur sont choisis par l'enseignant → paramétrables)
  - Si erreur lors de l'import : repassage (3 points en moins)



## Recette Web

- **Étape 1** : L'enseignant **télécharge** l'archive **sur son propre PC** dans un **répertoire de test choisi par l'enseignant**
- **Étape 2** : L'enseignant modifie la configuration permettant d'accéder à la BDD de test (un seul emplacement) + éventuellement modification d'un autre fichier de configuration (racine du site par exemple...) en suivant les instructions décrites dans le fichier README
  - Si autres actions à réaliser pour que le site soit opérationnel : repassage (3 points en moins)





## *Recette Web*

- **Étape 3:** L'enseignant vérifie avec les étudiants les différents points de la recette
  - Si impossibilité de faire les tests : repassage (3 points en moins)



## Recette Web

- **Étape 4** : La qualité du code est analysé par un outil de qualimétrie logicielle ([SONAR](#))
  - Intégrité,
  - Modularité
  - Nombres de variables locales et globales
  - Commentaires, ...
- **Étape 5 (à posteriori)** : Les codes de chacun des binômes sont comparés entre eux avec un outil anti-plagiat ([Baldr](#))







## Recette Java

- **Étape 1** : L'enseignant télécharge l'archive sur son propre PC dans un répertoire de test
- **Étape 2** : L'enseignant modifie la configuration permettant d'accéder à la BDD de test (BDD importée lors de la recette Web)
  - Si autres actions à réaliser pour que le logiciel soit opérationnel : repassage (3 points en moins)
- **Étape 3** : Les étudiants et l'enseignant compilent ensemble le projet téléchargé
  - Si erreur de compilation : repassage (3 points en moins)





## *Recette Java*

- **Étape 4** : L'enseignant vérifie avec les étudiants les différents points de la recette
  - Si impossibilité de faire les tests : repassage (3 points en moins)



## Recette Java

- **Étape 5** : La qualité du code est analysé par un outil de qualimétrie logicielle (SONAR)
  - Intégrité,
  - Modularité
  - Nombres de variables locales et globales
  - Commentaires, ...
- **Étape 6 (à posteriori)** : Les codes de chacun des binômes sont comparés entre eux avec un outil anti-plagiat (Baldr)



***FIN***

*Des questions?*

