



Gestion de performances de coureurs

Quentin Le Guennec
<http://github.com/AureoleMoka/gpc.git>

Lycée Pierre Méchain Laon

8 mars 2014

1 Présentation

- Sommaire
- Introduction
- Travail effectué

2 Analyse UML

- Use cases
- Class
- Sequence

3 Code source

- Classe dbDial
- Classe Performance (serveur)

4 Conclusion

- Fonctionnalités présentes
- Difficultés rencontrés
- Avantages du système
- Améliorations possibles

Objectif du système

Objectifs

- Gérer le déroulement d'une course automatiquement
- Enregistrer les performances des coureurs
- Présenter sur un support de son choix les performances d'un coureur

Naming utilisé

Naming

dbDial Classe qui gère les dialogues avec la base de données

CodeReader Lecteur de code barre pour identifier un adhérent grâce à sa carte.

RFIDReader Lecteur de badges par RFID pour attribuer à un coureur un ID

Portique Lecteur de badges par RFID qui permet de notifier le passage d'un coureur

Michel Désigne un participant, Adhérent ou non

Prérequis

Software nécessaire

Un browser

Pour afficher les performances

Un serveur de base de données

Pour stocker les informations des adhérents

Le client

Actuellement développé en C#

Hardware nécessaire

Deux lecteur RFID

Un pour attribuer un badge à un participant et un portique RFID

Un lecteur de code barre

Pour identifier les adhérents grâce à leur carte

Participants au projet

Quentin Le Guennec

site web et de la base de données, aide générale
Classes CodeReader, dbDial

Bastien Kopka

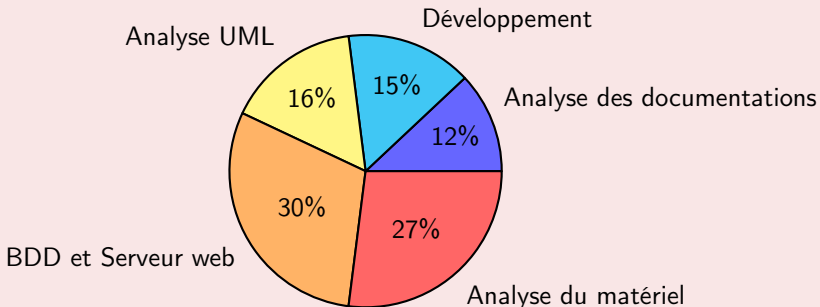
design du site web, analyse des documentations
Classes Portique, RFIDReader

Léo David

analyse des documentations
Classes Portique, RFIDReader

Répartition du temps

Temps



Au total :

Environ 70 heures

Sommaire

1 Présentation

- Sommaire
- Introduction
- Travail effectué

2 Analyse UML

- Use cases
- Class
- Sequence

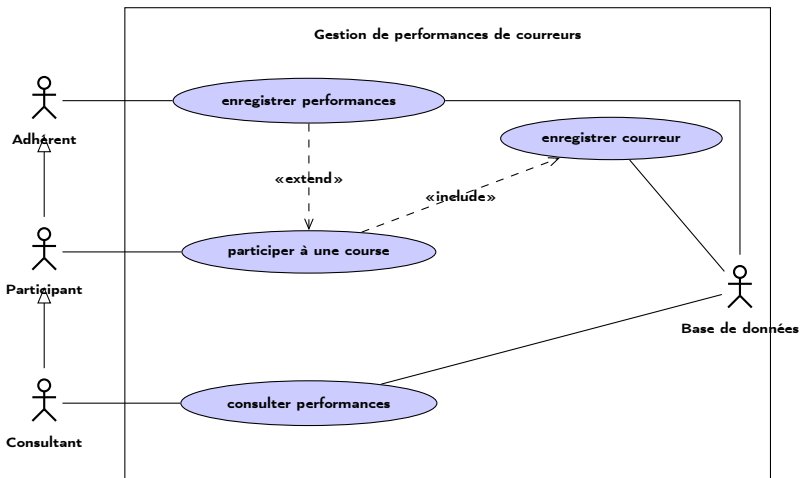
3 Code source

- Classe dbDial
- Classe Performance (serveur)

4 Conclusion

- Fonctionnalités présentes
- Difficultés rencontrés
- Avantages du système
- Améliorations possibles

Diagramme de cas d'utilisation



Gestion de performances de coureurs

Gestion des coureurs

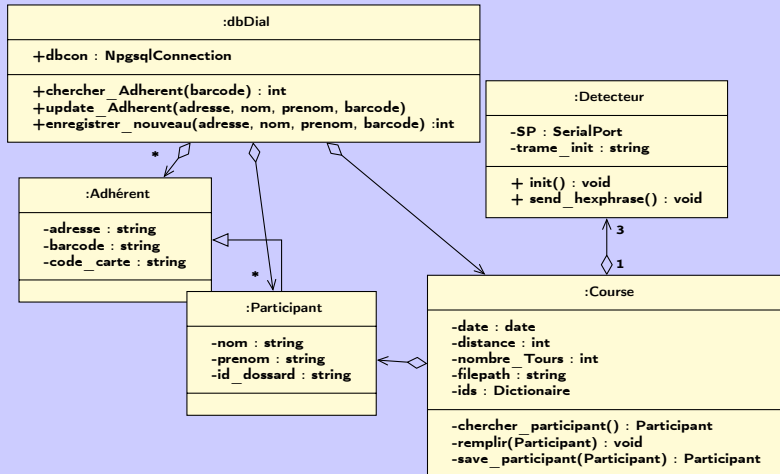
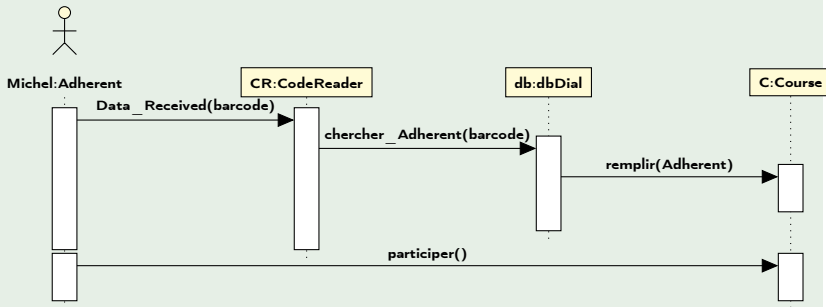


Diagramme de séquence

Michel est un adhérent



Sommaire

1 Présentation

- Sommaire
- Introduction
- Travail effectué

2 Analyse UML

- Use cases
- Class
- Sequence

3 Code source

- Classe dbDial
- Classe Performance (serveur)

4 Conclusion

- Fonctionnalités présentes
- Difficultés rencontrés
- Avantages du système
- Améliorations possibles

Connection à une DB postgresql en C#

Classe dbDial - Méthode constructeur

```
public dbdial(string server, string port, string
    database, string user, string password) {

    dbcon = new NpgsqlConnection(
        "Server=" + server + ";" +
        "Port=" + port + ";" +
        "Database=" + database + ";" +
        "User Id=" + user + ";" +
        "Password=" + password + ";Pooling=False"
    );

    liste_adherents = new List<Adherent>();
    remplir_liste_adherents();
}
```

Framework web Django - Classe Performances

Calcul des performances en python

```
def _get_vitesse(self , num_tour):  
    return self.course.distance / self.tours[  
        num_tour].temps  
  
def _get_fulltime(self):  
    return sum ([T.temps for T in self.tours.  
        all()])  
  
def get_vitesse_generale(self):  
    return self.course.distance / self.  
        _get_fulltime()
```

Sommaire

1 Présentation

- Sommaire
- Introduction
- Travail effectué

2 Analyse UML

- Use cases
- Class
- Sequence

3 Code source

- Classe dbDial
- Classe Performance (serveur)

4 Conclusion

- Fonctionnalités présentes
- Difficultés rencontrés
- Avantages du système
- Améliorations possibles

Fonctionnalités présentes

Fonctionnalités

- Identification des adhérents sur le serveur
- Détection des puces RFID
- Détection des codes barre
- Consultation des performances

Difficultés rencontrés

Difficultés

- Documentation peu claire
- Pour une raison inconnue, le lecteur de badges répondait très lentement
- L'ordinateur ARM sur lequel était installé le serveur calculait lentement
- Le portique semblait défectueux

Avantages du système

Avantages

- Le système est multi-plateforme
- Le système est entièrement modulable
- Facilités de déploiement

Améliorations possibles

Améliorations

- Le serveur présente des failles
- La fonctionnalité qui permet de notifier un passage n'est pas terminée