

Analyse Préliminaire U-Quetigny

Université de Bourgogne
Systèmes d'Informations Orientés Objets



JALLAL BOUDABZA
YACINE MAGHEZZI

Encadrant : Mr. Kokou Yetongnon

- 1 Le projet U-Quetigny
- 2 Cahier des charges
- 3 Organisation et Spécifications
- 4 Base de données
- 5 Conclusion

1 - Le projet U-Quetigny

Analyse
Preliminaire
U-Quetigny

Le projet
U-Quetigny

Cahier des
charges

Organisation
et
Spécifications

Base de
données

Conclusion

- 1 Le projet U-Quetigny
- 2 Cahier des charges
- 3 Organisation et Spécifications
- 4 Base de données
- 5 Conclusion

Analyse
Préliminaire
U-Quetigny

Le projet
U-Quetigny

Cahier des
charges

Organisation
et
Spécifications

Base de
données

Conclusion



Application libre

- ▶ Application de gestion avec BDD



Application libre

- ▶ Application de gestion avec BDD
- ▶ Application Web

2 - Cahier des charges

Analyse
Préliminaire
U-Quetigny

Le projet
U-Quetigny

Cahier des
charges

Organisation
et
Spécifications

Base de
données

Conclusion

- 1 Le projet U-Quetigny
- 2 Cahier des charges
- 3 Organisation et Spécifications
- 4 Base de données

Contraintes

- ▶ Projet en Binôme
- ▶ Date limite : 16 Janvier 2014
- ▶ Application avec Base de donnée

3 - Organisation et Spécifications

Analyse
Préliminaire
U-Quetigny

Le projet
U-Quetigny

Cahier des
charges

Organisation
et
Spécifications

Logiciels

Langages

Architecture

Librairies

Modules

Base de
données

Conclusion

- 1 Le projet U-Quetigny
- 2 Cahier des charges
- 3 Organisation et Spécifications**
 - Logiciels
 - Langages
 - Architecture
 - Librairies
 - Modules
- 4 Base de données

Analyse
Preliminaire
U-Quetigny

Le projet
U-Quetigny

Cahier des
charges

Organisation
et
Spécifications

Logiciels

Langages

Architecture

Librairies

Modules

Base de
données

Conclusion



Logiciels

- ▶ IDE : Netbeans 7.4
- ▶ Performant pour la compilation des application web

Analyse
Préliminaire
U-Quetigny

Le projet
U-Quetigny

Cahier des
charges

Organisation
et
Spécifications

Logiciels

Langages

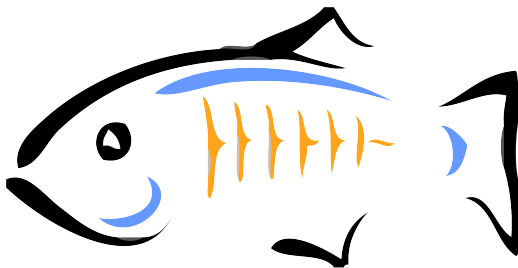
Architecture

Librairies

Modules

Base de
données

Conclusion



Logiciels

- ▶ Serveur : Glassfish
- ▶ Bonne compatibilité avec Netbeans

Analyse
Preliminaire
U-Quetigny

Le projet
U-Quetigny

Cahier des
charges

Organisation
et
Spécifications

Logiciels

Langages

Architecture

Librairies

Modules

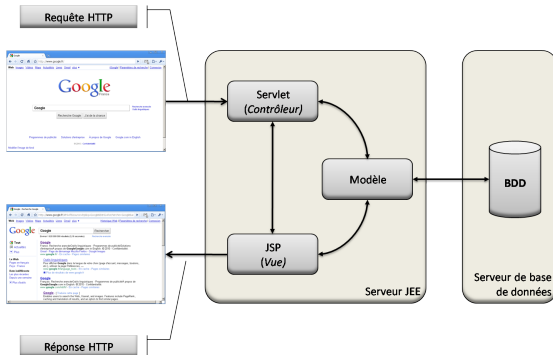
Base de
données

Conclusion



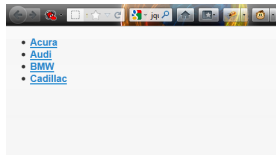
Langages

- ▶ Utilisation de Java et JSP
- ▶ Application Java orientée Web

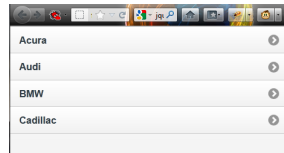


Modèle

- ▶ Pattern MVC
- ▶ Implémentations des modules séparément



(a) Sans jQuery



(b) Avec jQuery

Utilité de jQuery Mobile

- ▶ Rapide, simple, performant et optimisé pour les plateformes mobiles
- ▶ Design attrayant

Interface d'interrogation

- ▶ Java \longrightarrow MongoDB

Schémas et système dynamique

- ▶ Java \Longleftrightarrow MongoDB

Persistance des objets

- ▶ \implies MongoDB

Mashup et Intégration de Données

- ▶ Faire communiquer des applications Web : \longrightarrow ??

3 Organisation et Spécifications

4 Base de données

- Choix du SGBD
- Diagramme de classe
- Diagramme ODMG
- ODL
- OQL
- Insertion de tuples

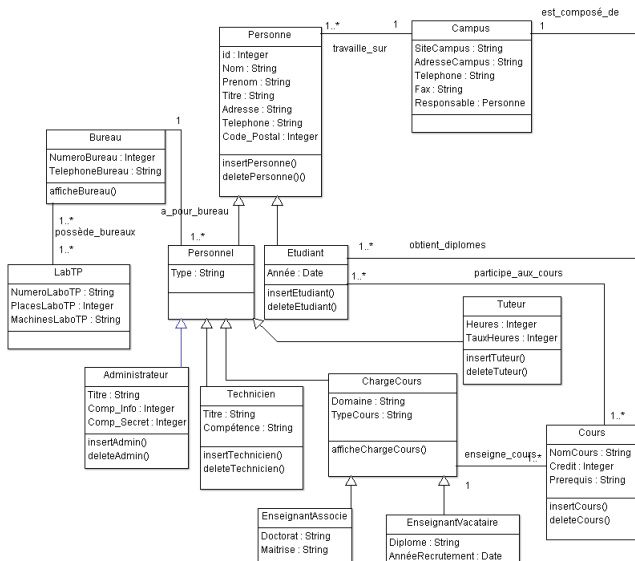
5 Conclusion

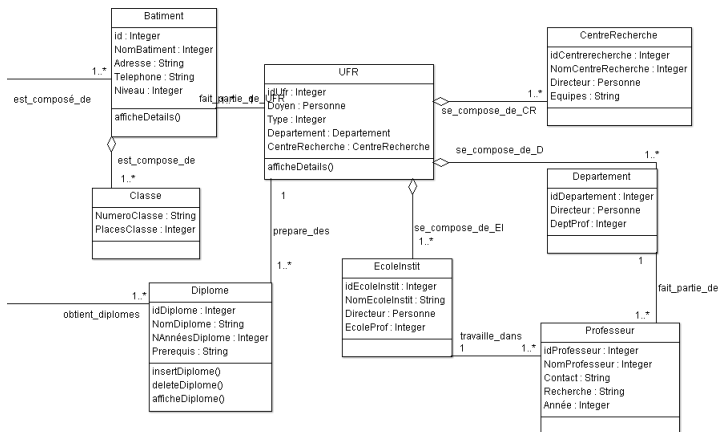


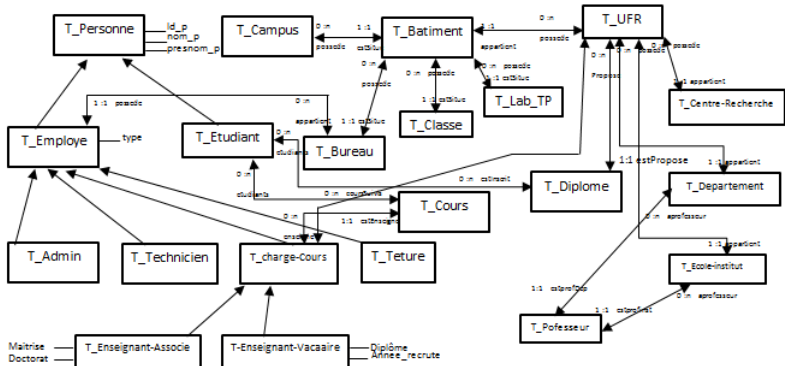
mongoDB

Pourquoi ?

- ▶ Documentation riche
- ▶ Le + utilisé des SGBD NoSQL
- ▶ Drivers Java/MongoDB nombreux







```

1  class Personne
    (extent LesPersonnes key idPersonne)
3  {
    attribute int id;
5    attribute string Nom;
    attribute string Prenom;
7    attribute string Titre;
    attribute string Adresse;
9    attribute string Telephone;
    attribute string Code_Postal;
11
    relationship List<Campus> travaille_sur inverse
        Campus:: a_pour_professeur;
13
    void insertP(idPersonne);
15    void deleteP(idPersonne);
    }

```

```
class Personnel : Personne
2 {
    attribute string TypePersonnel;
    4 relationship List<Bureau> a_pour_bureau inverse
        Bureau::appartien_a;
}
```

```
1 class Etudiant : Personne
  {
3     attribute date Annee;

5     relationship Set<Diplome> obtient_diplomes
        inverse Diplome::est_obtenu_par;
    relationship Set<Cours> suit_cours inverse Cours
        ::est_suivi_par;

7     void insertEtudiant();
9     void deleteEtudiant();
  }
```

```
class Administrateur : Personnel
2 {
    attribute string Titre;
    attribute int Comp_Info;
    attribute int comp_secret;
    void insertAdmin();
    void deleteAdmin();
8 }
```

```
1 class Technicien : Personnel
  {
3     attribute string Titre;
     attribute string Competence;
5
     void insertTechnicien();
7     void deleteTechnicien();
  }
```



```
class Tuteur : Personnel
2 {
    attribute int Heures;
    attribute int TauxHeures;
4
    void insertTuteur();
    void deleteTuteur();
6
8 }
```

```
class ChargeCours : Personnel
2 {
    attribute string Domaine;
    attribute string TypeCours;
4
    relationship Cours enseigne_cours inverse Cours
6         :: est_enseigne_par;

    void afficheChargeCours();
8 }
}
```

```
1 class EnseignantAssocie : ChargeCours
  {
3     attribute string Doctorat;
     attribute string Maitrise;
5  }
```

```
1 class EnseignantVacataire : ChargeCours
  {
3     attribute string Diplome;
     attribute date AnneeRecrutement;
5
     void insertEnseignantVacataire();
7     void deleteEnseignantVacataire();
  }
```

```

class Campus
{
    attribute string SiteCampus;
    attribute string AdresseCampus;
    attribute string Telephone;
    attribute string Fax;
    attribute Personne Responsable;

    relationship Personne est_frequente_par inverse
        Personne::frequente;
    relationship Batiment est_compose_de inverse
        Batiment::fait_partie_du_campus;
}

```

```

1  class Cours
    (extent LesCours key idCours)
3  {
    attribute int idCours;
5   attribute string NomCours;
    attribute int Credit;
7   attribute string Prerequis;

9   relationship List<ChargeCours> est_enseigne_par
        inverse ChargeCours::enseigne_cours;
    relationship Set<Etudiant> est_suivi_par inverse
        Etudiant::suit_cours;

11  void insertCours();
13  void deleteCours();
    }

```

```

1  class Diplome
2      (extent LesDiplomes key idDiplome)
3  {
4      attribute int idDiplome;
5      attribute string NomDiplome;
6      attribute int NAnneesDiplome;
7      attribute string Prerequis;
8
9      relationship Set<Etudiant> est_obtenu_par
10         inverse Etudiant::obtient_diplomes;
11     relationship List<UFR> est_prepare_dans inverse
12         UFR::prepare;
13
14     void insertDiplome();
15     void deleteDiplome();
16     void afficheDiplome();
17 }
    
```

```

1  class UFR
    (extent LesUFR key idUFR)
3  {
    attribute int idUFR;
5    attribute Personne Doyen;
    attribute int TypeUFR;
7    attribute Departement DepartementUFR;
    attribute CentreRecherche CentreRechercheUFR;
9
    relationship Batiment est_compose_de_batiments
        inverse Batiment::fait_partie_de_UFR;
11   relationship Diplome prepare_diplomes inverse
        Diplome::est_prepare_dans;
13
    relationship EcoleInstit se_compose_de_decole
        inverse EcoleInstit::ecole_fait_partie_de;
    relationship CentreRecherche
        se_compose_de_centre inverse CentreRecherche
        ::centre_fait_partie_de;

```



```

1  class CentreRecherche
    (extent LesCentresRecherche key
      idCentreRecherche)
3  {
    attribute int idCentreRecherche;
    attribute string NomCentreRecherche;
    attribute Personne DirecteurCentreRecherche;
    attribute string Equipes;

    relationship List<UFR> centre_fait_partie_de
      inverse UFR::se_compose_de_centre;
9  }

```

```

class Departement
2   (extent LesDepartement key idDepartement)
3   {
4       attribute int idDepartement ,
5       attribute Personne DirecteurDepartement ;
6       attribute int DeptProf ;
7
8       relationship List<UFR> dep_fait_partie_de
9           inverse UFR::se_compose_de_dep ;
10      relationship Professeur dep_a_pour_professeur
11          inverse Professeur::fait_partie_de
12  }
  
```

```

class EcoleInstit
    (extent LesEcolesInstit key idEcoleInstit)
{
    attribute int idEcoleInstit;
    attribute string NomEcoleInstit;
    attribute Personne DirecteurEcoleInstit;
    attribute int EcoleProf;

    relationship List<UFR> ecole_fait_partie_de
        inverse UFR::se_compose_decole;
    relationship Professeur ecole_a_pour_professeur
        inverse Professeur::travaille_dans
}
    
```

```

1  class Professeur
    (extent LesProfesseurs key idProfesseur)
3  {
    attribute int idProfesseur;
5    attribute string NomProfesseur;
    attribute string Contact;
7    attribute string Recherche;
    attribute int Annee;
9
    relationship List<EcoleInstit> travaille_dans
        inverse EcoleInstit::ecole_a_pour_professeur
        ;
11   relationship List<Departement>
        fait_partie_de_dep inverse Departement::
        dep_a_pour_professeur;
    }

```

```
1 SELECT struct (p.nom, p.prenom)
2 FROM p in Personne
3 GROUP BY (Dijonnais: p.codepostal = 21000 ,
4 Autre : p.codepostal != 21000);
```

Requête n°1

- Explication : Affichage des de toutes les personnes provenant de Dijon et des alentours.

```
2 SELECT c.* FROM etudiant e, cours c WHERE e.id = c.
    idCours
    AND e.nom = "Maghezzi" AND e.prenom = "Yacine";
```

Requête n°2

- Explication : Affichage des cours de l'étudiant Maghezzi Yacine.

Insertion dans mongoDB

```
2 etu_temp = db.etu.findOne({nom : « Maghezzi »,
    prenom : « Yacine »})
    cours = db.cours.find({idCours : etu_temp.idCours})
```

```
db.personne.insert( { nom: "Boudabza", prenom: "Jallal", Titre: "Monsieur", Adresse: "Chemin de st Apo" .... }
```

Insertion

- Explication : MongoDB insertion dans une base *DB* et classe *Personne*

5 - Conclusion

Analyse
Préliminaire
U-Quetigny

Le projet
U-Quetigny

Cahier des
charges

Organisation
et
Spécifications

Base de
données

Conclusion

- 2 Cahier des charges
- 3 Organisation et Spécifications
- 4 Base de données
- 5 Conclusion**

- ▶ Analyse des spécifications du projet
- ▶ Choix préliminaires
- ▶ Gain de temps et meilleures chances de réussite dans le projet

Jallal Boudabza
jallalboudabza21@gmail.com

Yacine Maghezzi
yacine1991@gmail.com

Questions ?