

Portail AWE

Point de contrôle – 3/3

COOLS Aurélie GENIN Émilie GAUTHIER Inès
LERAT Jean-Sébastien ROMBAUX Michael
ROBERTS François VANDEN DRIES Virginie

FPMS

2 mai 2019

Plan

- | | | | |
|---|--|---|------------------|
| 1 | Introduction | 3 | Modèle |
| 2 | Mise à jour du PID | 4 | Tests/Validation |
| | <ul style="list-style-type: none">• Diagramme de Gantt• Gestion des retards | 5 | Backend |
| | | 6 | Conclusion |

Introduction – cahier des charges

- ➊ **Portail Web** gérant l'accès aux données
- ➋ Démontrer l'**interconnexion** de sources de **données** diverses
- ➌ **Visualiser** les données (tableaux, graphiques)
- ➍ Respect **RGPD** (suppression des données personnelles)
- ➎ **Authentification** utilisateur

Environnement *Proof of concept*

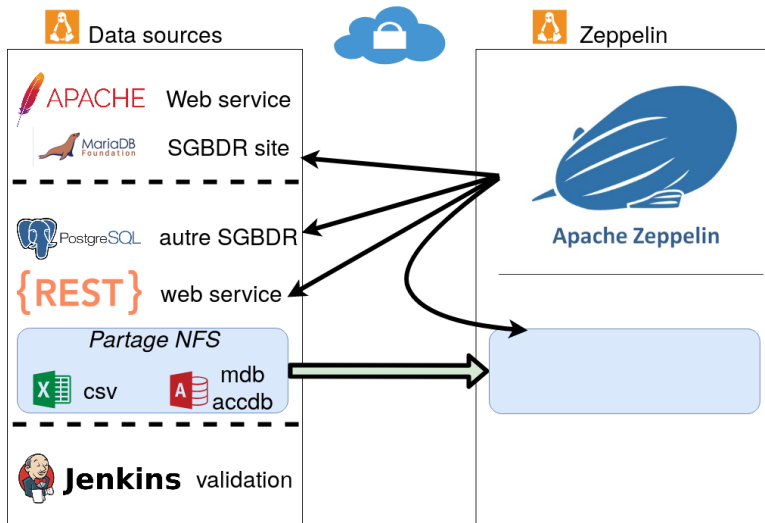


Diagramme de Gantt – Depuis le *checkpoint 2*

10 0	▢ ◦ Histoire 4 - données	19/03/19	02/04/19
107	◦ Ajout d'un fichier de données (backend)	19/03/19	02/04/19
110	◦ Test de l'ajout d'un fichier de données	19/03/19	02/04/19
111	◦ Ajout d'un fichier de données (frontend)	19/03/19	02/04/19
114	◦ Intégration des sources fichiers	19/03/19	02/04/19
11 0	▢ ◦ Histoire 5 - graphiques	02/04/19	16/04/19
116	◦ Gestion des graphiques dans Zeppelin	02/04/19	16/04/19
118	◦ Intégration des graphiques (backend)	02/04/19	16/04/19
119	◦ Intégration des graphiques (frontend)	02/04/19	16/04/19
120	◦ Validation (tests) des graphiques	02/04/19	16/04/19
12 0	▢ ◦ Histoire 6 - scientifique	16/04/19	01/05/19
115	◦ Ajout d'une source REST (implémentation serveur REST) + Jenkins	16/04/19	30/04/19
121	◦ Création d'un projet scientifique et requête de droits (backend)	16/04/19	30/04/19
123	◦ Création d'un projet scientifique et requête de droits (frontend)	16/04/19	30/04/19
124	◦ Tests des droits de projets scientifiques	16/04/19	30/04/19
140	◦ Recherche d'une source de données	30/04/19	01/05/19
139	◦ Recherche d'un utilisateur	30/04/19	01/05/19

Diagramme de Gantt – Suite

13	3,8,10,11,12	◻ ◦	Histoire 7 - Webservice	02/05/19	16/05/19
126		◦	Implémentation API Backend REST d'accès aux données	02/05/19	16/05/19
127		◦	Sortie frontend REST d'accès aux données	02/05/19	16/05/19
129		◦	Intégration de la source REST	02/05/19	16/05/19
131		◦	Tests de l'API REST	02/05/19	16/05/19
14	3,8,10,11,12	◻ ◦	Histoire 8 - responsive	14/05/19	28/05/19
132		◦	Adaptation responsive des graphiques	14/05/19	28/05/19
136		◦	Adaptation responsive de l'upload de fichiers	14/05/19	28/05/19
133		◦	Adaptation responsive des frontend	14/05/19	28/05/19
137		◦	Tests d'intégration responsive	14/05/19	28/05/19

Équipe

Équipe	Membres
Gestion	Virginie, Jean-Sébastien
Tests	Aurélie, Inès
BDD	Émilie, Virginie
BI/Infrastructure	Jean-Sébastien
Site <i>frontend</i>	Émilie, Virginie
Site <i>backend</i>	François, Michael ¹
Site <i>responsive</i>	François, Michael

1. Participation de Jean-Sébastien (interaction avec Zeppelin)

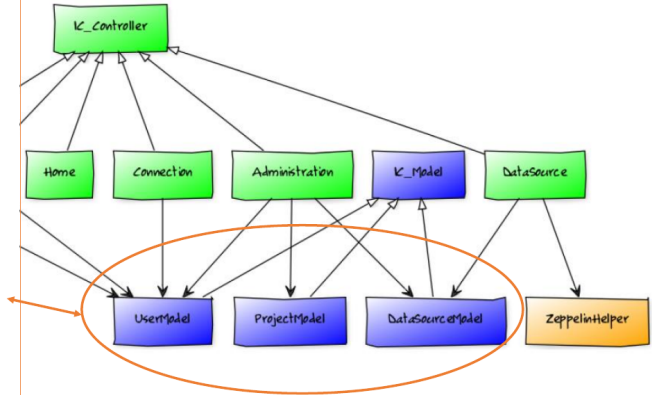
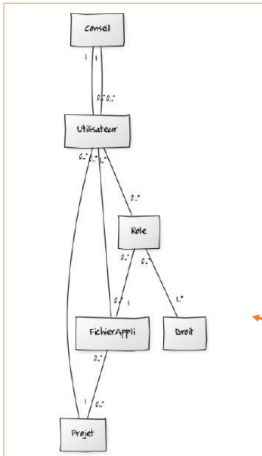
Équipe

Équipe	Description
Gestion	coordination, respect <i>scrum</i>
Tests	unitaires, fonctionnels (intégration continue)
BDD	base de données du site (droits, utilisateurs, ...)
BI/Infrastructure	matériel, configuration, interfaçage des données
Site <i>frontend</i>	visuel du site
Site <i>backend</i>	fonctionnalité du site
Site <i>responsive</i>	visuel mobile du site et compatibilité

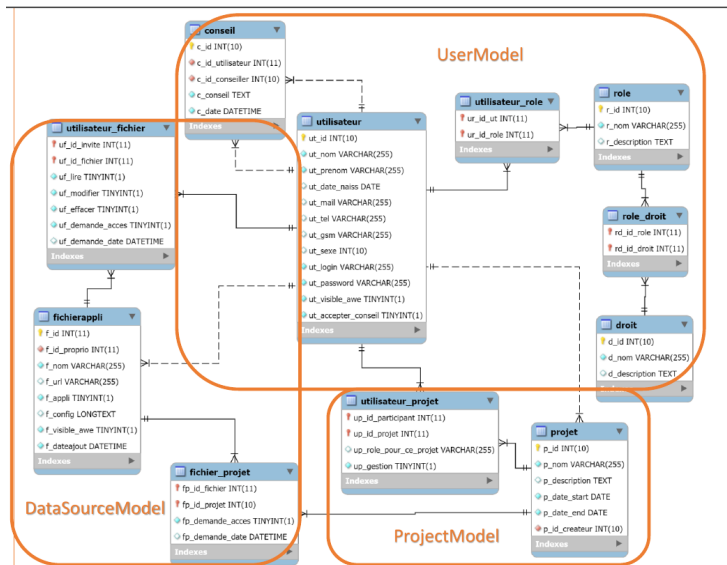
Gestion des retards

Plus de retard, en accord avec la planification

Modèle – Conception



Modèle – Conception



Modèle – Implémentation

• Types de fonctions :

- insert
- delete
- update
- Get

1 / table ou 1 / certains attributs (ex. : changement de l'accessibilité d'une personne à une source de données)

- ~ 1 / attributs provenant d'une ou plusieurs tables,
- introduction de filtres de recherches

→ En évolution avec l'évolution des fonctionnalités

Modèle – Implémentation

- Types de fonctions :

- add
- delete
- update
- get

```
public function addRole($name, $description = NULL)
{
    if(empty($name)) return false;

    $sql="INSERT IGNORE INTO role (r_nom, r_description) VALUES (?,?)";
    if( ! $this->db->query($sql, array($name, $description)) ) return false;

    return $this->db->insert_id();
}
```

Modèle – Implémentation

• Types de fonctions :

- add
- **delete**
- update
- get

```
public function deleteUserProject($userID, $projID)
{
    if (empty($userID))
        return false;
    if (empty($projID))
        return false;

    $sql = "DELETE FROM utilisateur_projet WHERE up_id_participant=? AND up_id_projet=?";
    if (! $this->db->query($sql, array(
        intval($userID),
        intval($projID)
    )))
        return false;

    return true;
}
```

Modèle – Implémentation

• Types de fonctions :

- add
- delete
- **update**
- get

```
public function updateDataSourceProject($dataSourceID, $projectID, $askAccess)
{
    if(is_null($dataSourceID)) return false;
    if(is_null($projectID)) return false;

    $first=true;
    $params = array();
    $sql="UPDATE fichier_projet SET ";

    if(isset($askAccess))
    {
        if($first) $first=false;
        else $sql.=' , ';

        $sql.=' fp_demande_acces = ? ';
        $params[] = intval($askAccess);
    }

    $sql.= " WHERE fp_id_projet= ".intval($projectID)." AND fp_id_fichier = ".intval($dataSourceID);

    if($first) return false;

    if( ! $this->db->query($sql, $params) ) return false;
    else return true;
}
```

Modèle – Implémentation

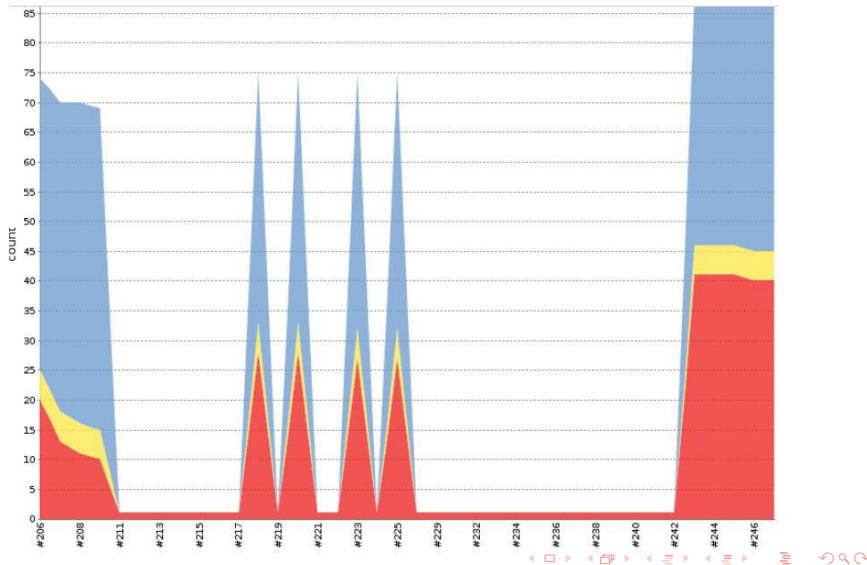
• Types de fonctions :

- add
- delete
- update
- **get**

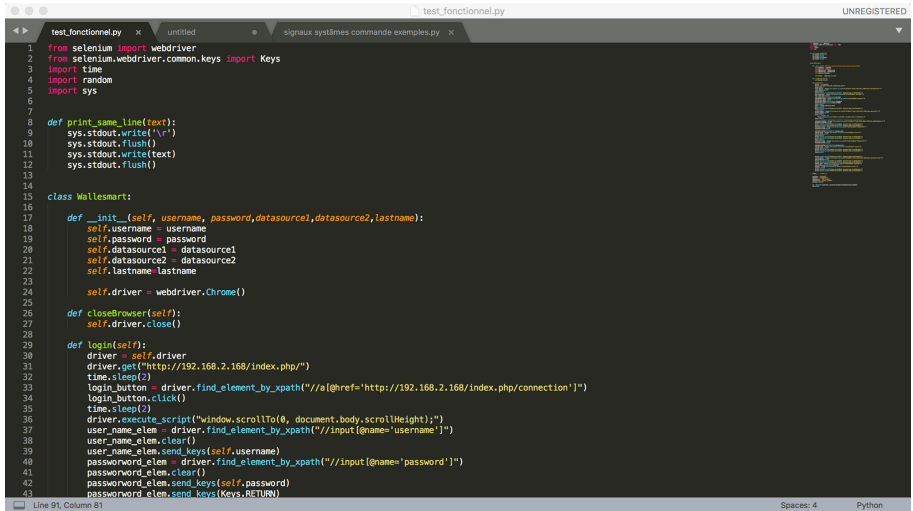
```
public function getRight($rightid)
{
    if(is_null($rightid)) return NULL;
    $sql="SELECT
                                d_id AS id,
                                d_nom AS name,
                                d_description AS description
        FROM droit
        JOIN role_droit ON rd_id_droit=d_id
        WHERE rd_id_role=?
        ORDER BY d_nom";
    $query = $this->db->query($sql, array($rightid));
    $droit=$query->row_array();

    return $droit;
}
```


Tests/Validation



Tests/Validation



```

1 from selenium import webdriver
2 from selenium.webdriver.common.keys import Keys
3 import time
4 import random
5 import sys
6
7
8 def print_same_line(text):
9     sys.stdout.write('\r')
10    sys.stdout.flush()
11    sys.stdout.write(text)
12    sys.stdout.flush()
13
14
15 class Wallesmart:
16
17     def __init__(self, username, password,datasource1,datasource2,lastname):
18         self.username = username
19         self.password = password
20         self.datasource1 = datasource1
21         self.datasource2 = datasource2
22         self.lastname=lastname
23
24         self.driver = webdriver.Chrome()
25
26     def closeBrowser(self):
27         self.driver.close()
28
29     def login(self):
30         driver = self.driver
31         driver.get("http://192.168.2.168/index.php/")
32         time.sleep(2)
33         login_button = driver.find_element_by_xpath("//a[@href='http://192.168.2.168/index.php/connection']")
34         login_button.click()
35         time.sleep(2)
36         driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
37         user_name_elem = driver.find_element_by_xpath("//input[@name='username']")
38         user_name_elem.clear()
39         user_name_elem.send_keys(self.username)
40         password_elem = driver.find_element_by_xpath("//input[@name='password']")
41         password_elem.clear()
42         password_elem.send_keys(self.password)
43         password_elem.send_keys(Keys.RETURN)

```

Line 91, Column 81

Spaces: 4 Python

Backend

Register – Création du compte

- Test du login
- Traitement des champs via des règles définies (`$this→form_validation→set_rules(...)`)
 - Required
 - min/max length
 - password matches
 - valid_email

Backend

Profil

- Profil
- Modifier le mot de passe
- Modifier les droits d'accès
- Suppression du compte

Backend

Datasource

- Ajout de nouvelle source de données
- Affichage des données
- Gérer ses sources et accès

Conclusion – État

- Retard rattrapé
- Prototypes testés
- Histoires 7 et 8 : responsive et Webservice