

# Réalisation professionnelle 3 :

## Helios formation

Compétences mises en œuvre :

- Recenser et identifier les ressources numériques
- Exploiter des référentiels, normes et standards adoptés par le prestataire informatique
- Collecter, suivre et orienter des demandes
- Traiter des demandes concernant les applications
- Planifier les activités
- Réaliser les tests d'intégration et d'acceptation d'un service
- Accompagner les utilisateurs dans la mise en place d'un service

### Contexte



L'application existante est utilisée par l'administration pour de gérer les formations proposées et les étudiants inscrits.

Elle permet de :

- Lister les formations proposées
- Créer, modifier et supprimer une formation
- Afficher le détail d'une formation
- Lister les étudiants de toutes les formations
- Inscrire un étudiant à une formation
- Afficher le détail des informations d'un étudiant

La version de l'application actuelle est composée de deux pages principales et quatre pages secondaires qui sont ouvertes suite à des actions sur les pages principales.

## Recenser et identifier les ressources numériques

Pour concevoir au mieux l'application j'ai utilisé de multiples ressources numériques. À savoir :



### Eclipse

Eclipse est un environnement de développement intégré libre extensible, universel et polyvalent, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Il est distribué sous licence Eclipse qui est une licence libre à copyleft faible utilisée à l'origine par l'EDI Eclipse. L'Eclipse Public License est conçue pour être plus favorable aux entreprises souhaitant faire du logiciel propriétaire.



### Mamp

Le logiciel MAMP (abréviation de Macintosh, Apache, MySQL and PHP) permet de lancer un serveur web local sur une machine fonctionnant sur le système d'exploitation Mac OS X.



### Java

Java est un langage de programmation orienté objet qui doit être compilé et une plate-forme informatique. Le compilateur Java produit du code intermédiaire dont les fichiers ont l'extension class. Ce code intermédiaire est appelé bytecode. Le bytecode n'est compréhensible que par la machine virtuelle, qui va traduire à la volée (interpréter) les instructions exécutées en code compréhensible par la machine physique. Il est distribué sous la licence publique générale GNU.



### phpMyAdmin

phpMyAdmin est une application Web de gestion pour les systèmes de gestion de base de données MySQL et MariaDB, réalisée principalement en PHP et distribuée sous licence GNU GPL. C'est un outil qui nous permet de visualiser rapidement l'état de notre base de données et de la modifier, sans avoir à écrire de requêtes SQL.



### GitLab

GitLab est un logiciel Open source et collaboratif, distribué sous licence MIT (licence de logiciel pour logiciels libres et open source). Il permet d'héberger les projets web, la gestion de versions des codes sources, permet une collaboration simple entre les collaborateurs sur un même projet. La version de base qui est déjà très complète est gratuite.



### Hibernate

Hibernate est un framework open source gérant la persistance des objets en base de données relationnelle, il est distribué sous la Licence publique générale limitée GNU.



### JDBC

La technologie JDBC (Java DataBase Connectivity) est une API fournie avec Java (depuis sa version 1.1) permettant de se connecter à des bases de données, c'est-à-dire que JDBC constitue un ensemble de classes permettant de développer des applications capables de se connecter à des serveurs de bases de données (SGBD).

## Explications :

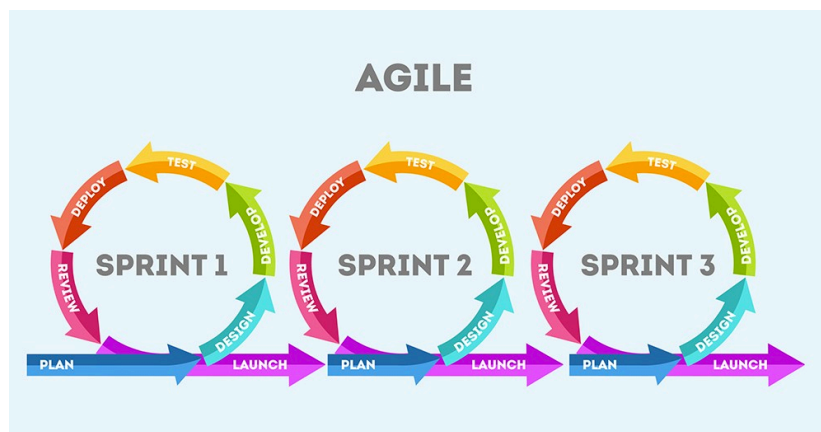
Pour faire évoluer au mieux l'application, sur mon poste de travail j'utilise Eclipse pour la réalisation des ajouts de fonctionnalités en Java. Afin de voir tester en local mon code, je lance Mamp, ma base de donnée est stockée sur PhpMyAdmin. L'organisation des tâches et issues est réalisée sur Gitlab. Enfin je versionne mon code sur Gitlab.

## Exploiter des référentiels, normes et standards adoptés par le prestataire informatique

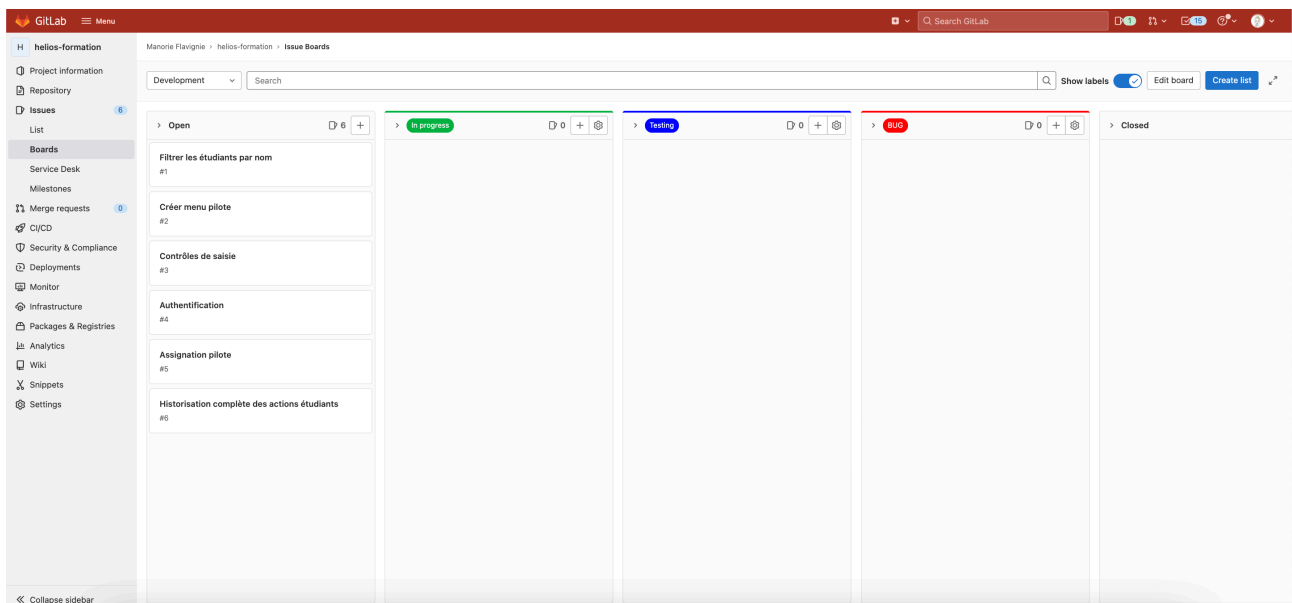
## Collecter, suivre et orienter des demandes

## Traiter des demandes concernant les applications

## Planifier les activités



Pour ce projet j'ai utilisé la méthode agile. La méthode Agile recommande de se fixer des objectifs à court terme. Le projet est donc divisé en plusieurs sous-projets. Une fois l'objectif atteint, on passe au suivant, et ce jusqu'à l'accomplissement de l'objectif final. Cette approche est plus flexible. On entend par sprint des itérations de courtes durées décomposant un processus de développement souvent complexe afin de le rendre plus simple et plus facile à concevoir.

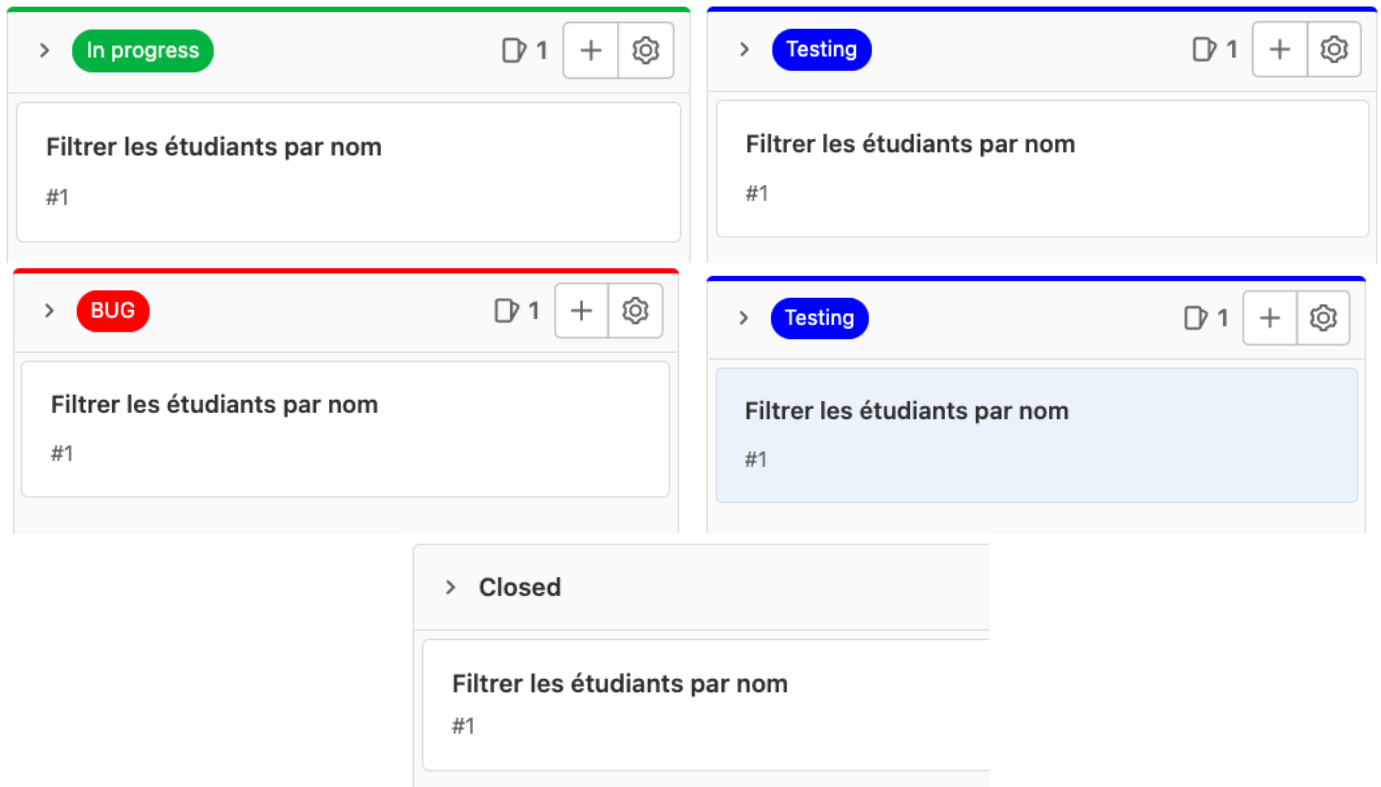


## Sprint 1

Lors du premier sprint j'ai eu 2 expressions de besoins à mettre en oeuvre. A savoir :

1. La création d'un filtre par nom des étudiants
2. Un menu pilote

### 1 - La création d'un filtre par nom des étudiants



Les premiers tests, le code que j'avais conçu n'était pas pertinent ainsi la fonctionnalité ne fonctionner pas, j'ai donc déplacé mon issue au sein de la colonne « BUG », après plusieurs tentatives j'ai enfin réussi à concevoir le besoin demandé.

```
MonJPanel filtrePanel = new MonJPanel();
filtrePanel.setLayout(new FlowLayout());

filtreField = new MonJTextField(20);
filtrePanel.add(filtreField);

filtreBtn = new MonJButton("Filtrer");
filtreBtn.addActionListener(this);
filtreBtn.setFont(Constance.texteFont);
filtrePanel.add(filtreBtn);

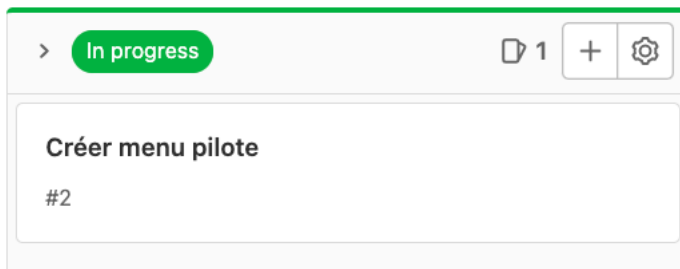
panTabPlusBtns.add(filtrePanel, BorderLayout.NORTH);
```

```
else if(e.getSource() == filtreBtn){
    System.out.println("filtrer étudiant");
    controleur.filtrerParNom(filtreField.getText());
    filtreField.setText("");
}
```

```
@Override
public List<Etudiant> readByNom(String nom) {
    EntityManager entityManager = EntityManagerUtil.getEntityManager();
    Query query = entityManager.createQuery("select e from Etudiant e where lower(e.nom) like :nom");
    query.setParameter("nom", "%" + nom + "%");
    List<Etudiant> etudiants = query.getResultList();
    entityManager.close();

    return etudiants;
}
```

## 2 - Un menu pilote



J'ai commencé la création du menu, pilote, pour le moment j'ai un tableau qui affiche la liste des formations, j'ai créé le bouton permettant d'afficher toutes les informations de la formation néanmoins il n'est pas opérant.

```
piloteItem = new MonJMenuItem ("Pilote" );
administration.add(piloteItem) ;
piloteItem.addActionListener (this) ;
```

```
else if(source == piloteItem) {
    System.out.println ("** Action option pilote") ;
    ModelePiloteFormationTab modelePiloteFormationTab = new ModelePiloteFormationTab();
    ControleurPilote controleurPilote = new ControleurPilote(service, modelePiloteFormationTab);
    VuePilote vuePilote = new VuePilote(controleurPilote, modelePiloteFormationTab);
    panneauCourant = vuePilote;
}
```

```
1 package ihm.admin.pilote;
2
3 import java.awt.BorderLayout;
4
5 /**
6  * Vue principale "Pilote". Administrer les formations proposées par hélios.
7  * Afficher tableau pilote (action supprimer/afficher/modifier) et formulaire de création.
8  *
9  * @author tophe
10  * @version 1.0
11  */
12 public class VuePilote extends MonJPanel implements ActionListener {
13     private ControleurPilote controleur;
14     // private PanFormulairePilote panFormulairePromotion;
15     private JTable table;
16     private ModelePiloteFormationTab model;
17     private ModeleEtudiantPilote modeleEtudiant;
18     private MonJButton btnSupprimer;
19     private MonJButton btnAfficher;
20     private MonJButton btnModifier;
21
22     public VuePilote(ControleurPilote controleurPilote, ModelePiloteFormationTab model) {
23         this.controleur = controleurPilote;
24         this.model = model;
25         setBorder(new EmptyBorder(10, 10, 10, 10));
26         setLayout(new BorderLayout(10, 10));
27         MonJPanel panTitrePlusForm = new MonJPanel();
28         panTitrePlusForm.setLayout(new BorderLayout(10, 10));
29
30         JLabel titre = new JLabel("PILOTE - FORMATIONS", SwingConstants.CENTER);
31         titre.setFont(Constance.titre2Font);
32         panTitrePlusForm.add(titre, BorderLayout.NORTH);
33
34         MonJPanel panTabPlusBtns = new MonJPanel();
35         panTabPlusBtns.setLayout(new BorderLayout(10, 10));
36
37         table = new MonJTable(model);
38         JScrollPane tabScroll = new JScrollPane(table);
39         tabScroll.setForeground(Color.white);
40         add(tabScroll, BorderLayout.CENTER);
41
42         MonJPanel panneauBtn = new MonJPanel();
43         panneauBtn.setLayout(new FlowLayout());
44         add(panneauBtn, BorderLayout.SOUTH);
45         panneauBtn.setFont(Constance.textFont);
46
47         btnAfficher = new MonJButton("Afficher les informations de la formation");
48         panneauBtn.add(btnAfficher);
49         btnAfficher.addActionListener(this);
50     }
51
52     @Override
53     public void actionPerformed(ActionEvent e) {
54         System.out.println("actionPerformed - Action boutons afficher");
55         int index = table.getSelectedRow();
56
57         if (index != -1) {
58             if (e.getSource() == btnAfficher) {
59                 System.out.println("Afficher les informations de la formation");
60
61                 Formation formation = model.getFormationByIndex(index);
62                 ModelePilote modelePilote = new ModelePilote();
63                 Formation formation2 = controleur.readAvecEtudiantEtModule(formation.getAcronyme());
64                 controleur.ihmPiloteAfficherInfosFormation(formation2, modelePilote);
65             } else {
66                 JOptionPane.showMessageDialog(null, "Vous devez sélectionner une formation.");
67             }
68         }
69     }
70 }
71
72
73
```

## Sprint 2

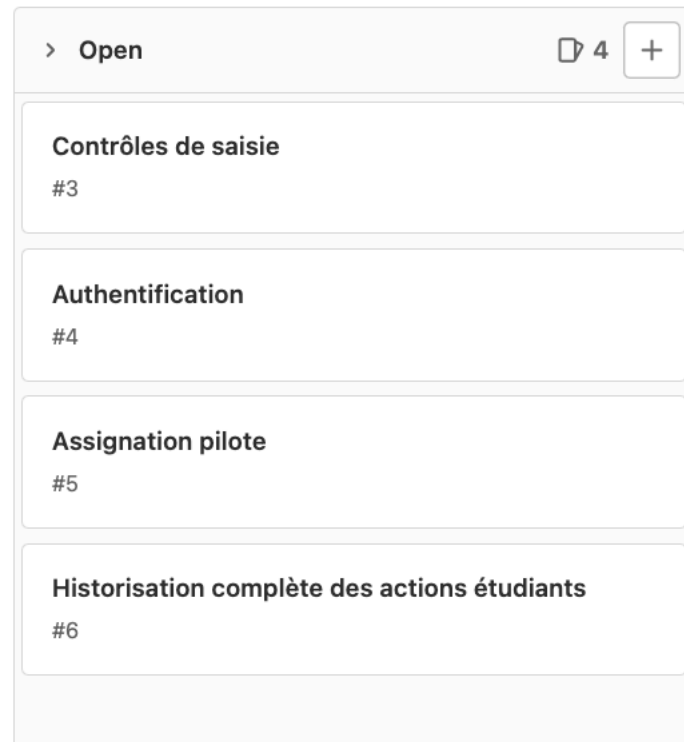
Lors du deuxième sprint j'ai eu 2 expressions de besoins à mettre en oeuvre. A savoir :

1. La mise en place d'un contrôle de saisie
2. L'authentification

## Sprint 3

Lors du troisième sprint j'ai eu 2 expressions de besoins à mettre en oeuvre. A savoir :

1. L'assignation pilote
2. L'historisation complète des actions étudiants



La deuxième partie du sprint 1 n'étant pas fini je n'ai pas pu continuer les sprints suivants.

**Réaliser les tests d'intégration et d'acceptation d'un service**

**Accompagner les utilisateurs dans la mise en place d'un service**

Video pour aide à l'utilisation du service