INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

RAPPORT PDLA

Application d'aide aux personnes vulnérables

LARGE Sophie

GIMOND Elise

4A IR-A

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

RAPPORT PDLA

Application d'aide aux personnes vulnérables

Lien dépôt git : https://github.com/sophielrge/SEE.git

Jira :https://elianboaglio.atlassian.net/jira/software/projects/PDLA/boards/2

LARGE Sophie

GIMOND Elise

4A IR-A

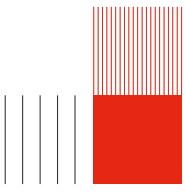
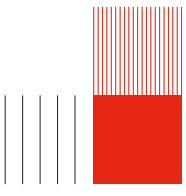


Table des matières



1. Contexte du projet

De nos, jours, un grand nombre de personnes se trouvent isolées, que ce soit à cause d'un éloignement familial ou de problèmes de santé invalidants. Ces situations créent des besoins spécifiques, souvent impossibles à satisfaire sans aide extérieure. Il peut s'agir par exemple de récupérer un colis pour une personne âgée ou d'assurer le lavage de vêtements durant une hospitalisation, des tâches qui requièrent un soutien extérieur fiable.

Au cours des deux derniers mois, nous avons cherché à concevoir une application destinée au bénévolat. Cette dernière permettrait aux personnes en difficulté de publier leurs demandes, qui pourraient alors être prise en charge par des bénévoles.

Pour organiser ce projet, nous avons utilisé Jira afin de définir et ordonner les besoins du client grâce aux user stories en suivant l'avancement des tâches. Ceci tout en gérant les deadlines à l'aide des sprints de manière agile. Côté développement, nous avons choisi Java comme langage principal. Nous avons également utilisé GitHub pour le contrôle de version tout au long de la réalisation du projet, facilitant la répartition du travail au sein notre équipe.

2. Application de la méthode agile

2.1 Méthode Scrum

Avant de se lancer dans le développement, nous avons cherché à définir les acteurs concernés par l'application, ainsi que les fonctionnalités les plus importantes en se basant sur le cahier des charges. Il s'agit ici de trois types de personnes, celles en situation de vulnérabilité, les bénévoles, et des personnes responsables, dont l'approbation est indispensable pour permettre à un volontaire de répondre à un service. Pour s'organiser au mieux, nous avons planifié notre travail en suivant la méthode Scrum, à l'aide d'un Jira.

2.2 Gestion des user stories

Lors de nos premières réunions, nous avons réalisé le backlog de notre projet. Pour cela, nous avons exprimé toutes les fonctionnalités nécessaires sous la forme d'user stories. Par exemple "En tant que bénévole, je veux pouvoir créer un compte pour répondre à des demandes d'aide". Nous avons ensuite ordonné ces tâches à l'aide de niveau de priorité afin de respecter les demandes du client.

2.3 Planification et suivi des sprints

Nous avons ensuite organisé notre avancement en sprints. Chaque semaine dans l'idéal, nous avons choisis deux ou trois user stories réalisable par les membres de notre équipe dans le temps imparti. Les réunions nous ont permise de discuter ensemble des sous-tâches nécessaires à concevoir, programmer et tester afin d'implémenter chaque nouvelle fonctionnalité. La semaine suivante, nous terminions le sprint en récapitulant ce qui fonctionnait ou non, puis planifions le suivant.

2.4 Répartition des tâches

Nous nous sommes réparti les tâches de cette manière :

- Ensemble : création des tables
- Sophie : gestion de la base de données, des tests de celle-ci, et des requêtes
- Elise : gestion du menu et du traitement de texte, des tests de celui-ci, et des types de comptes.

3. Organisation du Git

3.1 Structure du dépôt

Afin de collaborer facilement et de pouvoir travailler à distance, nous avons utilisé un dépôt Git. Chaque membre a ainsi pu travailler de son côté en synchronisant le développement à l'aide des commandes git tout en respectant certaines conditions. Lors de la création du projet, nous avons commencé par ajouter un fichier .gitignore comportant les fichiers à ne pas ajouter sur le dépôt lors des commit (par exemple les fichiers .class) afin d'éviter des conflits inutils. Chaque nouveau fichier a dû être ajouté au dépôt avec git add. Tout au long du développement, il nous a été nécessaire de bien répartir le travail et de communiquer pour être sûres que l'on ne modifiait pas les mêmes classes au même moment. Nous avons régulièrement réalisé des commit, avec des noms clairs de la forme "Fichier_modifié : modifications_apportées" pour y revenir facilement en cas de besoin, et des tags lors de moments clefs.

3.2 Amélioration possible

Malgré nos précautions, nous avons parfois eu des conflits que nous avons heureusement pu résoudre dans le merge editor. Afin d'éviter ce genre de problème la prochaine fois, il faut en fait travailler sur différentes branches, principale et secondaires. La branche main contient les versions stables, et on créé une branche secondaire pour chaque fonctionnalité en cours d'implémentation. On s'assure qu'une branche est stable avant de la merge avec la branche principale. Cette méthode de travail permet d'éviter tout conflit.

4. Mise en place de l'intégration continue (DevOps)

Pour ce projet, nous avons choisi GitHub Actions pour automatiser le pipeline CI/CD. Cet outil nous a permis de simplifier le processus de développement en automatisant les étapes essentielles comme la compilation, les tests, et le déploiement. Grâce à cela, nous sommes alertées par mail lorsque ce que nous ajoutons ne compile pas.

4.1 Déroulement du pipeline

Le pipeline se compose de plusieurs étapes :

- Compilation avec Maven : Nous avons utilisé Maven pour compiler le code source et garantir qu'il est conforme aux exigences.

- Exécution des tests unitaires : Les tests JUnit ont été automatisés pour s'assurer que chaque fonctionnalité fonctionne comme prévu. En cas d'échec, le pipeline s'interrompt automatiquement.
- Génération d'un fichier .jar : Une fois le code validé, un artefact exécutable est produit pour faciliter le déploiement.

4.2 Difficultés rencontrées

Nous avons rencontré plusieurs problèmes lors de la mise en place de ce pipeline. Par exemple, le déploiement était difficile à configurer en raison de permissions incorrectes et de problèmes avec les secrets. Nous avons dû reconfigurer nos fichiers pom.xml et maven.yml, en faisant attention à la syntaxe (ex : bien respecter les minuscules ou les majuscules).

5. Conclusion

Ce projet nous a permis de développer une application en Java avec la manipulation de base de données, tout en nous familiarisant avec des méthodes de travail indispensables pour le travail en équipe pour des clients. Nous avons aussi pu remarquer l'importance de l'utilisation des outils disponibles pour l'automatisation du développement tels que Maven et JUnit. Le PDLA nous a ainsi apporté une expérience non négligeable dans la réalisation de projet autant par ses aspects techniques, qu'organisationnels.

6. Table des annexes

```
SEE/
├── .github/
│   └── workflows/
├── .vscode/
├── SEE/
│   ├── src/
│   │   ├── main/java/org/pdla/assistance_app/
│   │   │   ├── accounts/
│   │   │   │   ├── User.java
│   │   │   │   ├── Applicant.java
│   │   │   │   ├── Validator.java
│   │   │   │   └── Volunteer.java
│   │   │   ├── structures/
│   │   │   │   ├── BDD.java
│   │   │   │   ├── Menu.java
│   │   │   │   ├── Request.java
│   │   │   │   └── Traitement_texte.java
│   │   │   └── Main.java
│   │   └── test/java/org/pdla/assistance_app/structure
│   │       ├── BDDTest.java
│   │       └── MenuTest.java
│   ├── target/
│   ├── Makefile
│   ├── pom.xml
│   └── .gitignore
├── BD_SEE.mwb
├── README.md
└── Rapport.md
```

Annexe 1 : Structure

```
-------------------------------------------
|               MENU SEE                  |
|0 - Quit                                 |
|1 - Sign in                              |
|2 - Log in                               |
-------------------------------------------
1
-------------------------------------------
|0 - Go back                              |
|1 - I am applicant                       |
|2 - I am volunteer                       |
|3 - I am validator                       |
-------------------------------------------
1
-------------------------------------------
|Enter your name                          |
Jean
|Enter your age                           |
32
|Enter your dpt                           |
28
|Enter a password                         |
****
-------------------------------------------
|Applicant account created with success |
|Your id is: 3 remember it
-------------------------------------------
```

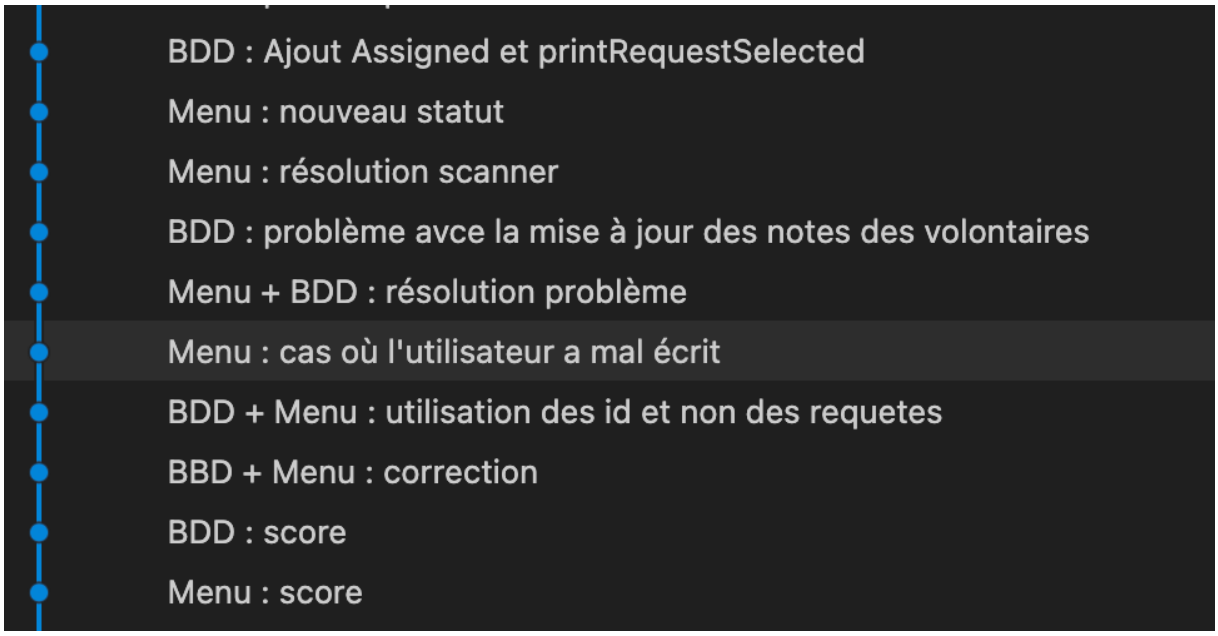
Annexe 2: Menu et création d'un compte appliquant

```
-----------------------------------------
|Welcome Anna                    |
-----------------------------------------
-----------------------------------------
|0 - Go back                            |
|1 - Add a new request                  |
|2 - View my requests                   |
-----------------------------------------
2
|Your requests                     :    |
-----------------------------------------
Request n°1
Subject: Récuperer mes courses
Date: 2024-12-12
Status: Completed
-----------------------------------------
Request n°2
Subject: Promener mon chien
Date: 2024-12-13
Status: Assigned
-----------------------------------------
Request n°3
Subject: M'amener chez le médecin
Date: 2024-12-14
Status: Completed
-----------------------------------------
Request n°4
Subject: Faire une lessive
Date: 2024-12-13
Status: Completed
-----------------------------------------
Request n°5
Subject: Faire du ménage
Date: 2024-12-15
Status: Completed
-----------------------------------------
Request n°6
Subject: déposer mes lettres à la poste
Date: 2024-12-18
Status: Pending
-----------------------------------------
```

Annexe 3: Connexion et affichage des requêtes

```
-----------------------------------------
|Welcome Mathias                 |
-----------------------------------------
-----------------------------------------
|0 - Go back                            |
|1 - Print pending requests             |
|2 - View my requests                   |
|3 - View my score                      |
-----------------------------------------
3
-----------------------------------------
|Score: 3.25
|Number of scores: 4
-----------------------------------------
```

Annexe 4 : Connexion volontaire et affichage du score



Annexe 5 : Exemples de commit

id int(11)	subj varchar(100)	id_volunteer int(11)	id_applican int(11)	id_validator int(11)	statut varchar(	helpda date	motif varchar(10	date_creation timestamp
1	Récuperer mes courses	1	1	1	C	2024-12-12	(NULL)	2024-12-10 15:06:28
2	Promener mon chien	3	1	(NULL)	S	2024-12-13	(NULL)	2024-12-10 15:07:06
3	M'amener chez le médecin	1	1	1	C	2024-12-14	(NULL)	2024-12-10 15:11:42
4	Faire une lessive	1	1	1	C	2024-12-13	(NULL)	2024-12-10 15:15:49
5	Faire du ménage	1	1	1	C	2024-12-15	(NULL)	2024-12-10 15:24:02
6	déposer mes lettres à la poste	(NULL)	1	(NULL)	P	2024-12-18	(NULL)	2024-12-10 15:26:38
7	Laundry	(NULL)	3	(NULL)	P	2024-12-20	(NULL)	2024-12-18 15:57:43

Annexe 6 : Table des requêtes

id int(11)	nom varchar(10C	age int(11)	dpt int(11)	note float	nb_avis int(11)	psw varchar(10C
1	Mathias	20	83	3.25	4	mathias
2	Gaël	21	12	(NULL)	0	gael
3	Laza	20	81	(NULL)	0	laza

Annexe 7 : Table des volontaires