

# 26 février 2021

## Backend :

Alexandre FROEHLICH  
Erwan AUBRY

## Frontend :

Guillaume LEINEN  
Ayrwan GUILLERMO  
Jean Noël CLINK

## User stories

- Création d'un SecurityManager
- Création d'une classe Configuration
- Création d'une instance permanente de serveur sur une carte RPi dédiée
- Création d'une BDD Sqlite
- Implémentation d'un ORM : ORMLite
- Création d'un contrôleur pour dispatcher les requêtes
- Refactorisation code serveur TCP concurrent
- Définition des structures manquantes (Groupes, Users, ...)
- Changement des requêtes texte en requêtes JSON et réponses en JSON
- Création des entités (modèles) des entrées BDD
- Créer une classe protocole par commande
- Ébauche de maquette inerte sur FIGMA de l'application client
- Commencement de la page de login client

Pour rappel l'ensemble des user stories est disponible sur le Zenhub : <https://app.zenhub.com/workspaces/projet-conception-logicielle-602f6ef86e99540010214bae/> ou encore dans les issues du repository github : <https://github.com/Projet-Java-ENSTA-Bretagne/Projet-Conception-Logicielle/issues>

The screenshot shows the ZenHub interface for a project named 'Projet Conception Logicielle'. The board is organized into columns representing different stages of the workflow: 'Icebox', 'Backlog', 'In Progress', 'Done', and 'Closed'. Each column contains a list of issues (user stories) with their titles, IDs, and assigned sprints. The 'Icebox' column has 3 issues, 'Backlog' has 5, 'In Progress' has 4, 'Done' has 12, and 'Closed' has 1. The interface includes a sidebar with navigation options like 'Board', 'Reports', 'Roadmap', 'Milestones', 'Notifications', 'Workflows', 'Create...', 'Edit Workspace', 'View Tutorials', 'Shortcuts', 'Support and training', and 'Changelog'. At the bottom, there are buttons for 'Invite your team' and 'NightlySide'.

# Résultats des tests unitaires

---

Pour l'instant nous n'effectuons pas de tests unitaires. Cette séance étant essentiellement dédiée à l'installation des logiciels et des dépendances.

Nous sommes déjà en train de mettre en place un coverage automatisé du code : <https://github.com/Projet-Java-ENSTA-Bretagne/Projet-Conception-Logicielle/issues/17>

## Cartes CRC

---

## Exécutables

---

Les fichiers exécutables se trouvent dans l'archive en pièce jointe. On fait bien la distinction client/serveur :

- Client : client-conception-logicielle-1.0-SNAPSHOT-jar-with-dependencies.jar
- Serveur : serveur-conception-logicielle-1.0-SNAPSHOT-jar-with-dependencies.jar

## Documentation

---

La documentation n'a pas été écrite pour ce premier sprint. Pour les prochains elle se fera générer par javadoc.

## Client

### Maquette inerte sur FIGMA

Le but de cette maquette est de visualiser l'apparence souhaité du client sans avoir à rédiger de code.

### Charte Graphique

Nous partons sur une charte graphique Material design pour des raisons esthétiques principalement

### Implémentation dans un projet javafx

Pour ajouter des composants Material-design dans Javafx, nous utilisons la bibliothèque Jfoenix. Elle s'intègre comme une dépendance Maven :

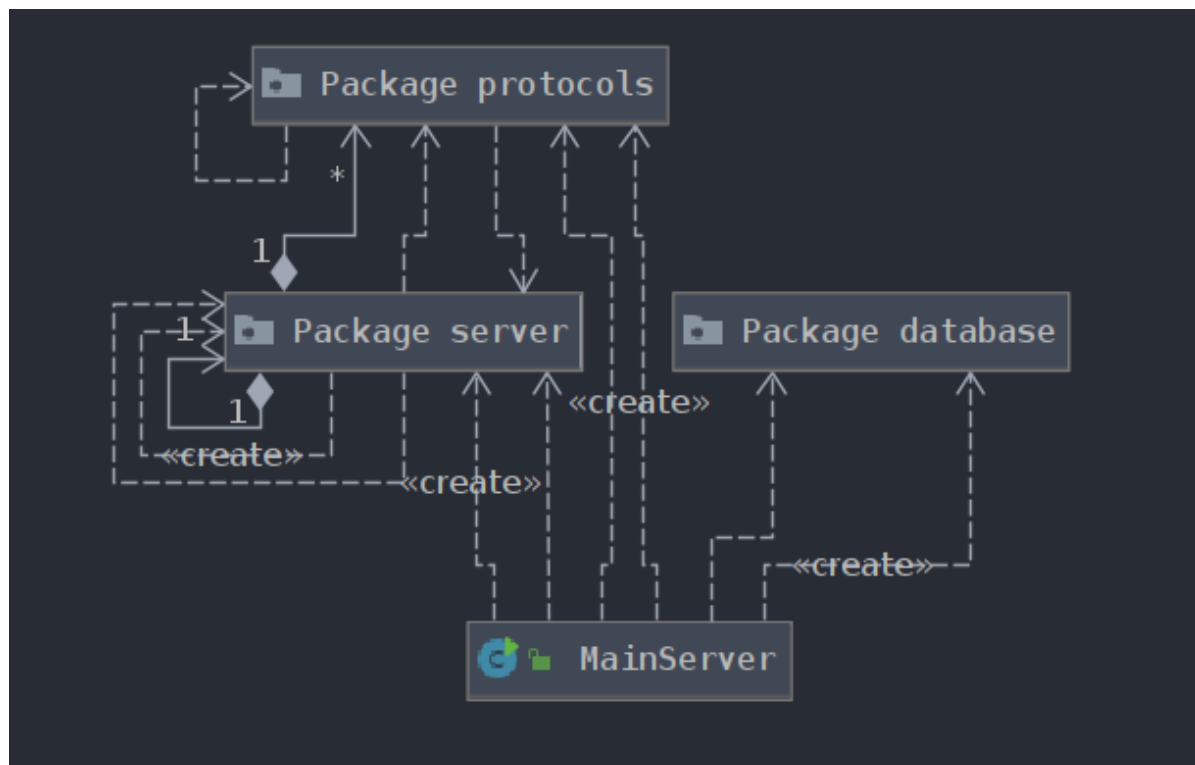
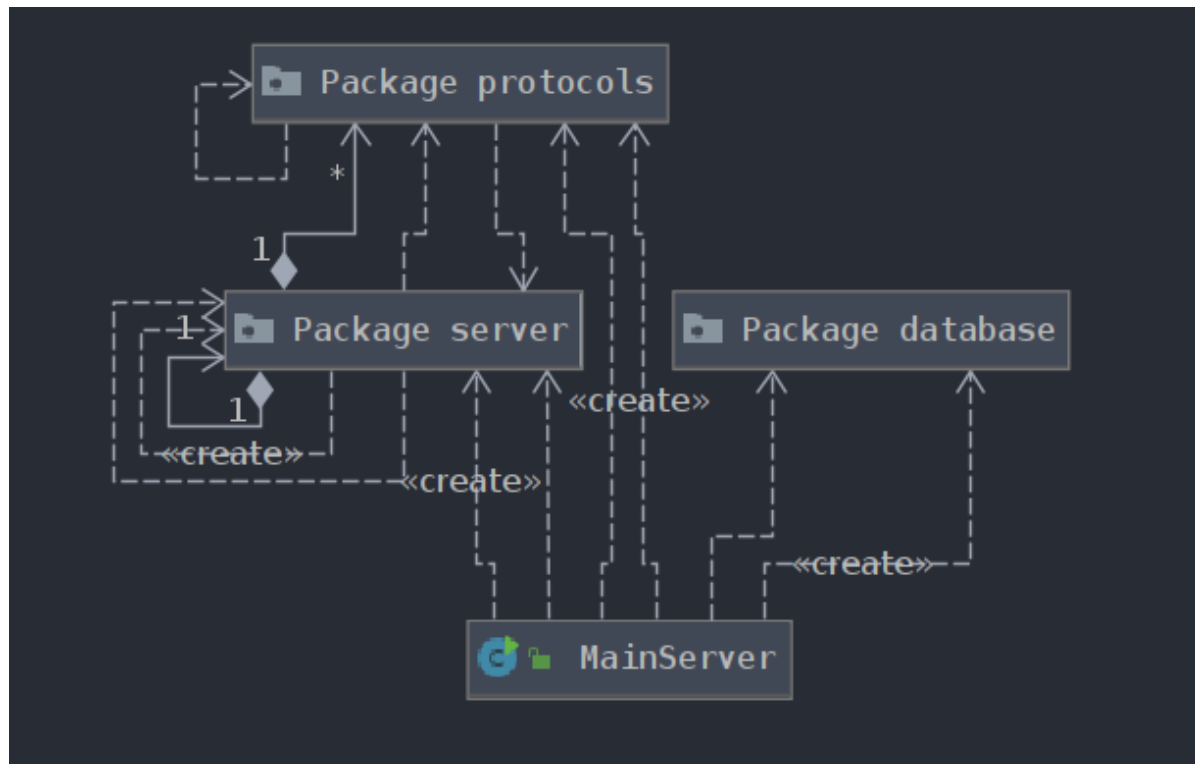
```
1 <dependency>
2   <groupId>com.jfoenix</groupId>
3   <artifactId>jfoenix</artifactId>
4   <version>9.0.10</version>
5 </dependency>
```

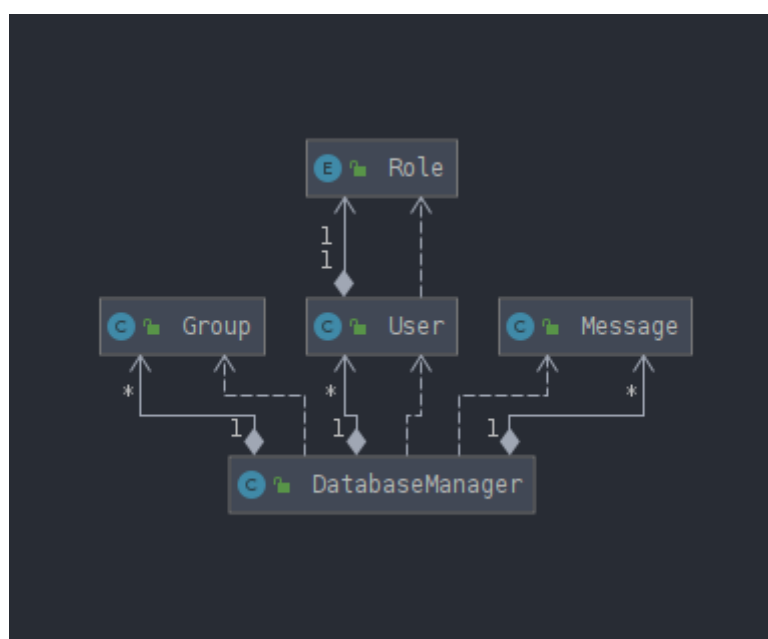
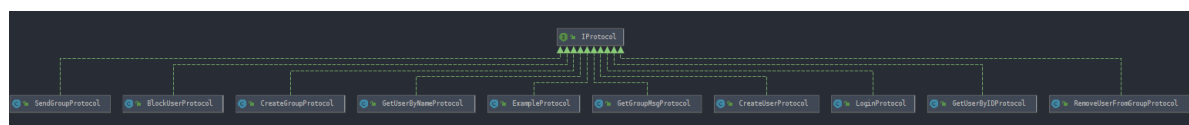
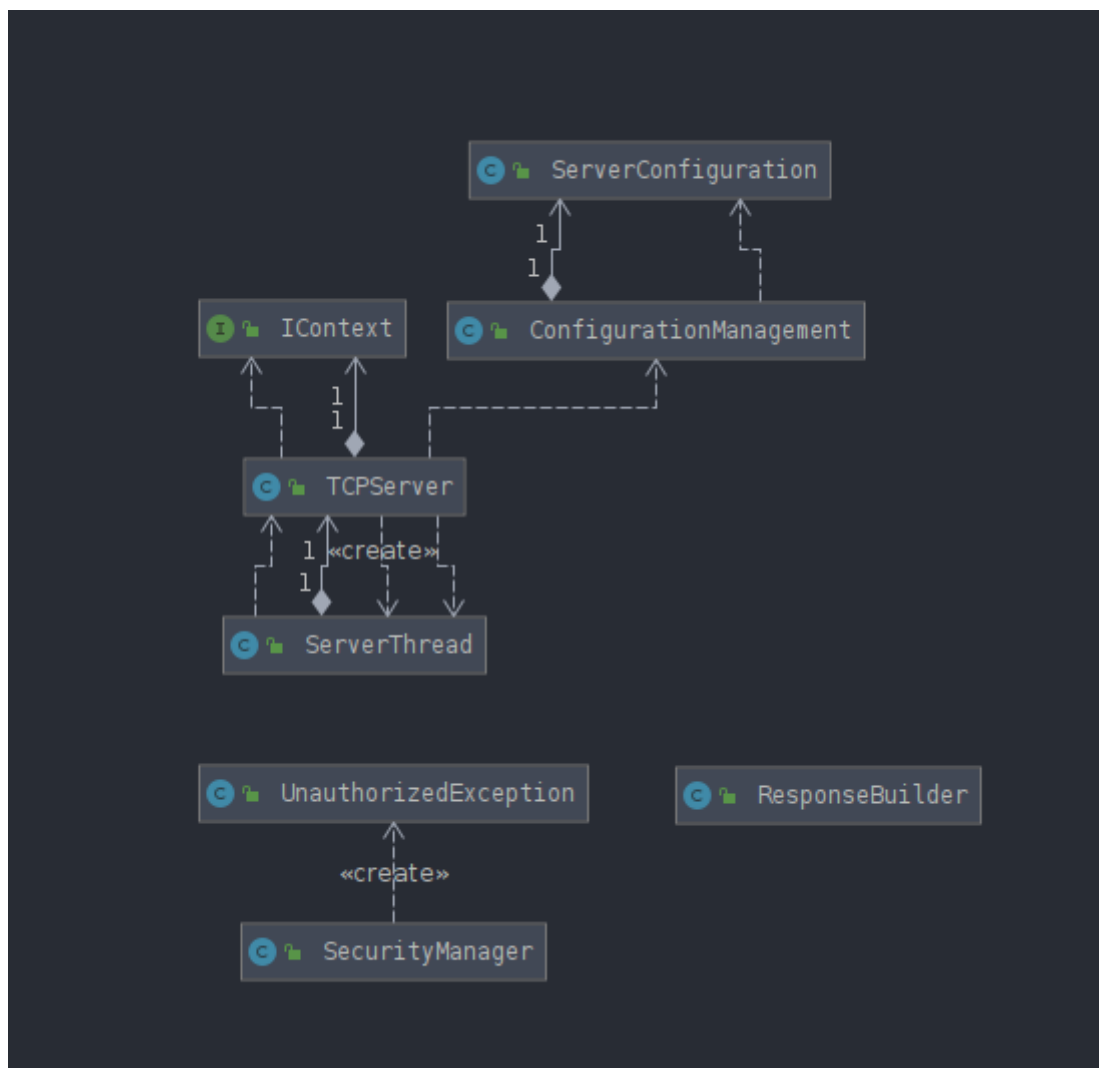
L'ajout dans Scenebuilder s'effectue via le téléchargement d'un Jar de Jfoenix et son import dans scenebuilder.

## Diagrammes de classe

---

## Serveur





## Client

