Hofmann Jonny

Documentation projet module 6

Division Technique du CEJEF

ILBSYS



**Table des matières**

[Introduction 2](#_Toc49443044)

[1 But de l’application 2](#_Toc49443045)

[2 Esquisse 2](#_Toc49443046)

[2.1 Cahier des charges fonctionnel 2](#_Toc49443047)

[2.2 Esquisse 2](#_Toc49443048)

[2.3 Gestion des fichiers 2](#_Toc49443049)

[2.4 Planification 2](#_Toc49443050)

[3 Maquette 3](#_Toc49443051)

[3.1 Use case UML 3](#_Toc49443052)

[3.2 Maquette 3](#_Toc49443053)

[4 Gestion du projet 3](#_Toc49443054)

[4.1 Versionning et code source 3](#_Toc49443055)

[4.2 Remontée et suivis des bogues 3](#_Toc49443056)

[5 Version 1.0 4](#_Toc49443057)

[5.1 Maquette 4](#_Toc49443058)

[5.2 Diagramme de classe 4](#_Toc49443059)

[6 Sprint 1 4](#_Toc49443060)

[6.1 Fonctionnalités 4](#_Toc49443061)

[6.1.1 Uptime 4](#_Toc49443062)

[6.1.2 Modification de serveurs 5](#_Toc49443063)

[6.1.3 Suppression de serveurs 6](#_Toc49443064)

[7 Sprint 2 6](#_Toc49443065)

[7.1 Fonctionnalité 6](#_Toc49443066)

[7.1.1 Accès depuis l’extérieur 6](#_Toc49443067)

[7.1.2 Jauges pour les pourcentages 6](#_Toc49443068)

|  |
| --- |
| ILBSYS |
| Projet module 6 |

# Introduction

Dans le cadre du module 6 nous devons réaliser une application Android avec Android studio. Ce module a été décaler en avant parce qu’il est plus que de faire un cours sur les microcontrôleurs à distance.

# But de l’application

Le but de l’application sera d’afficher des informations et des statistiques de différents postes. Ces informations sont stockées dans une base de données InfluxDB qui est nourrie avec Telegraf. Pour adapter le projet à la durée du projet, les fonctionnalités ont été réduite. De ce fait ce logiciel doit plus tôt être vu en tant que proof of concept au lieu d’une application complète.

# Esquisse

## Cahier des charges fonctionnel

* Ajout de serveur
* Connection automatique au serveur
* Choix des serveurs avec des noms définit à l’ajout
* Afficher l’utilisation CPU
* Afficher l’utilisation de la RAM

## Esquisse

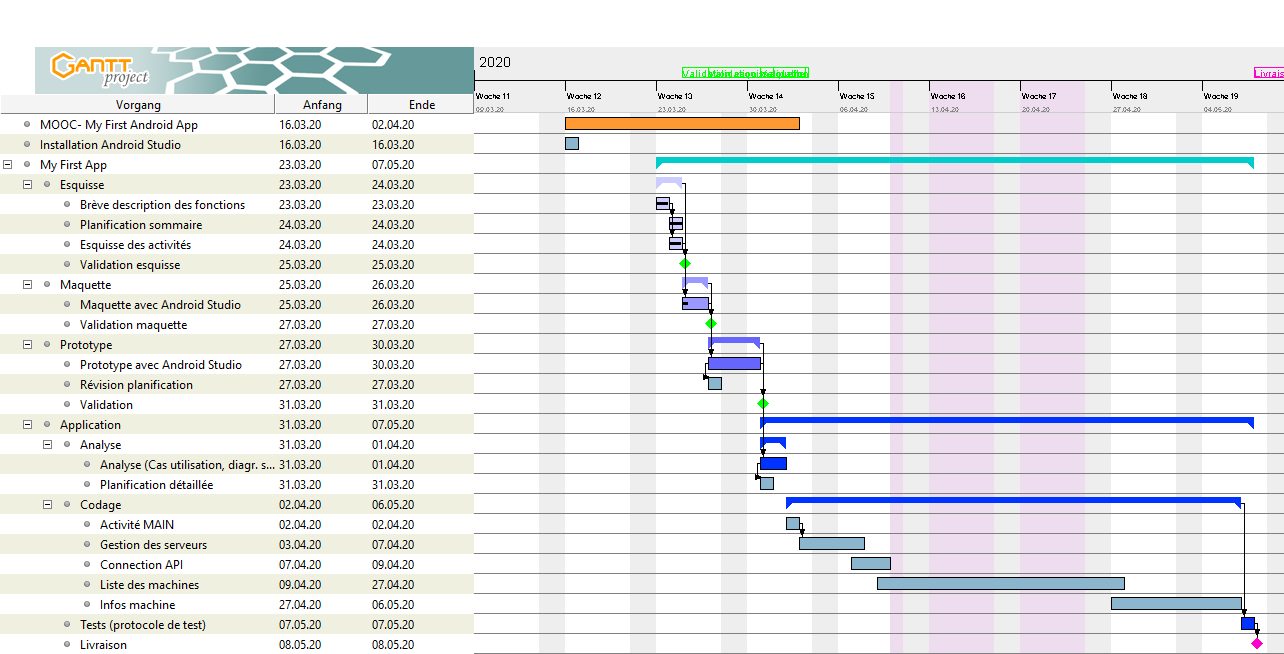
Ceci est la première Esquisse faite pour avoir une vue globale sur la structure fonctionnelle qu’aura l’application.

## Gestion des fichiers

Tous les fichiers sont sauvegardés sur mon GitHub personnel. [ILikeBananas/ILBSYS\_android](https://github.com/ILikeBananas/ILBSYS_android)

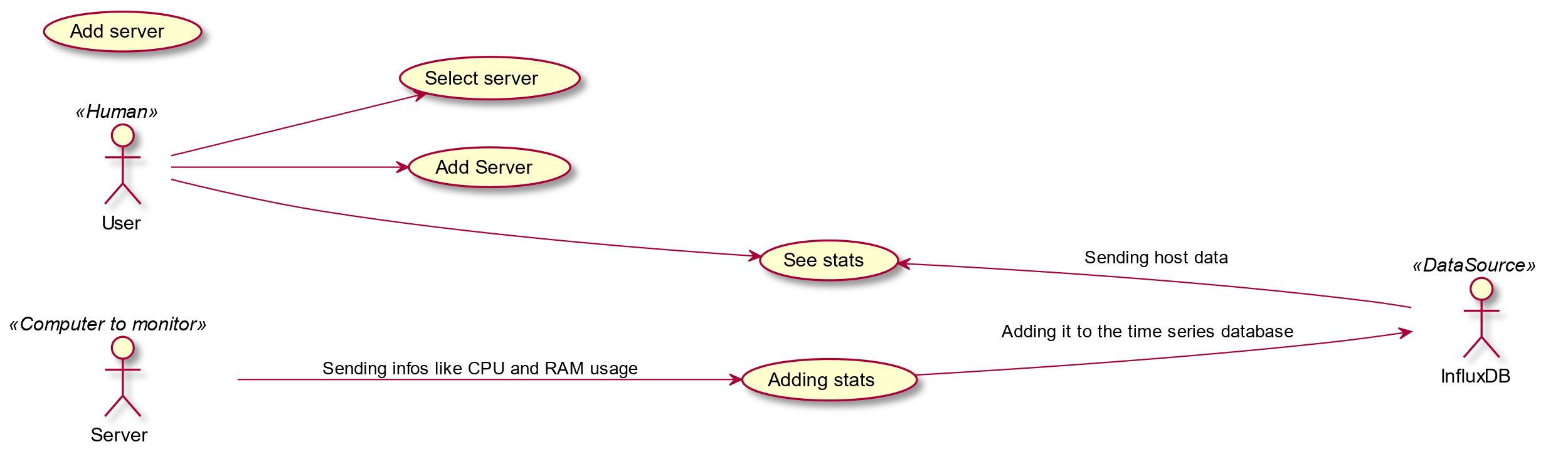
## Planification

La planification se situe dans le dossier « doc » sur GitHub.



# Maquette

## Use case UML



Ce use case est également disponible sous « doc/diagrams/out/useCases.svg » dans le repo GitHub.

## Maquette

La maquette est faite avec le logiciel Balsamiq Mokup et le fichier se trouve dans le dossier « doc » sur GitHub. Elle a été fortement modifiée et n’est plus à jour.

# Gestion du projet

## Versionning et code source

Les versions et le code source sont gérés dans Github ce qui permet de voir tous les changements, quand elles ont été faites et pour quelle cause elles ont été faites.

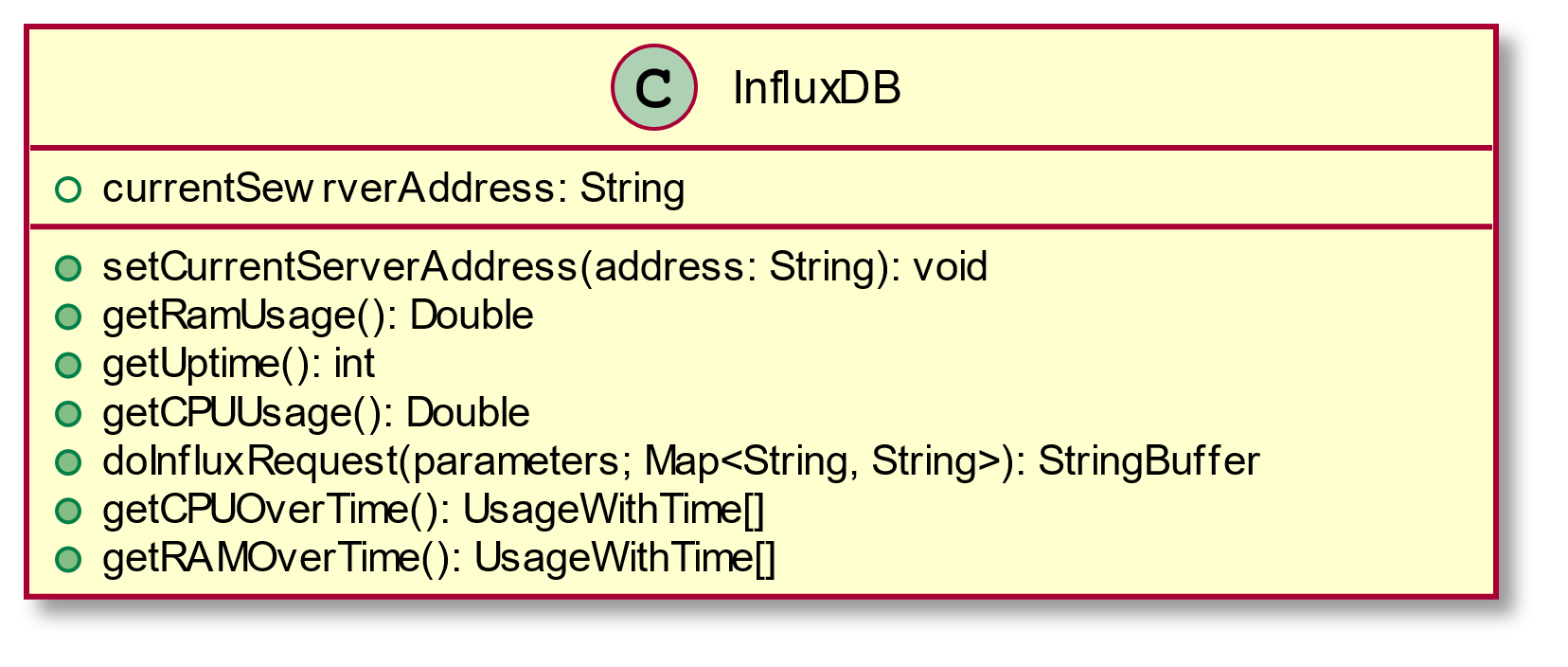
## Remontée et suivis des bogues

Les bogues trouvés durant les tests qui ne sont pas corrigeable directement sont noté et documenté sur GitHub dans la partie Issues. Ceci permet d’avoir un suivit et de lier le commit qui corrige l’erreur.

# Classes

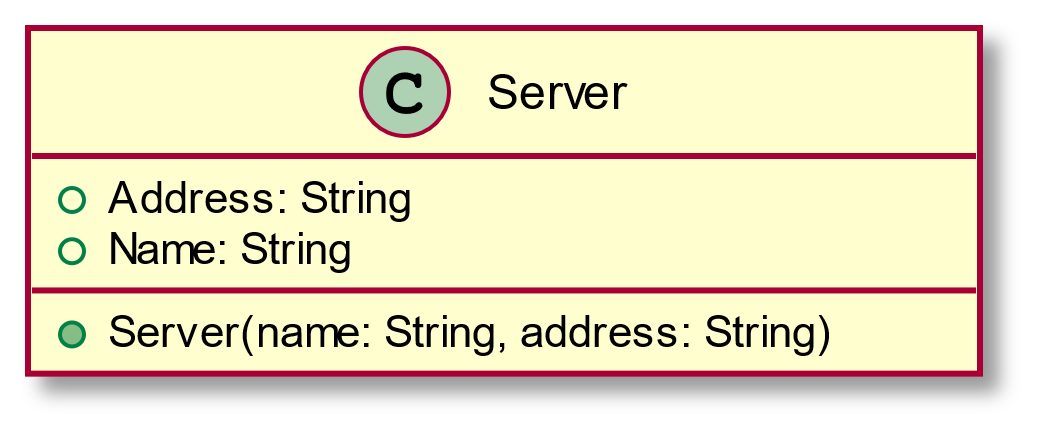
## InfluxDB

La classe Influx gère toues les requêtes vers le serveur InfluxDB.



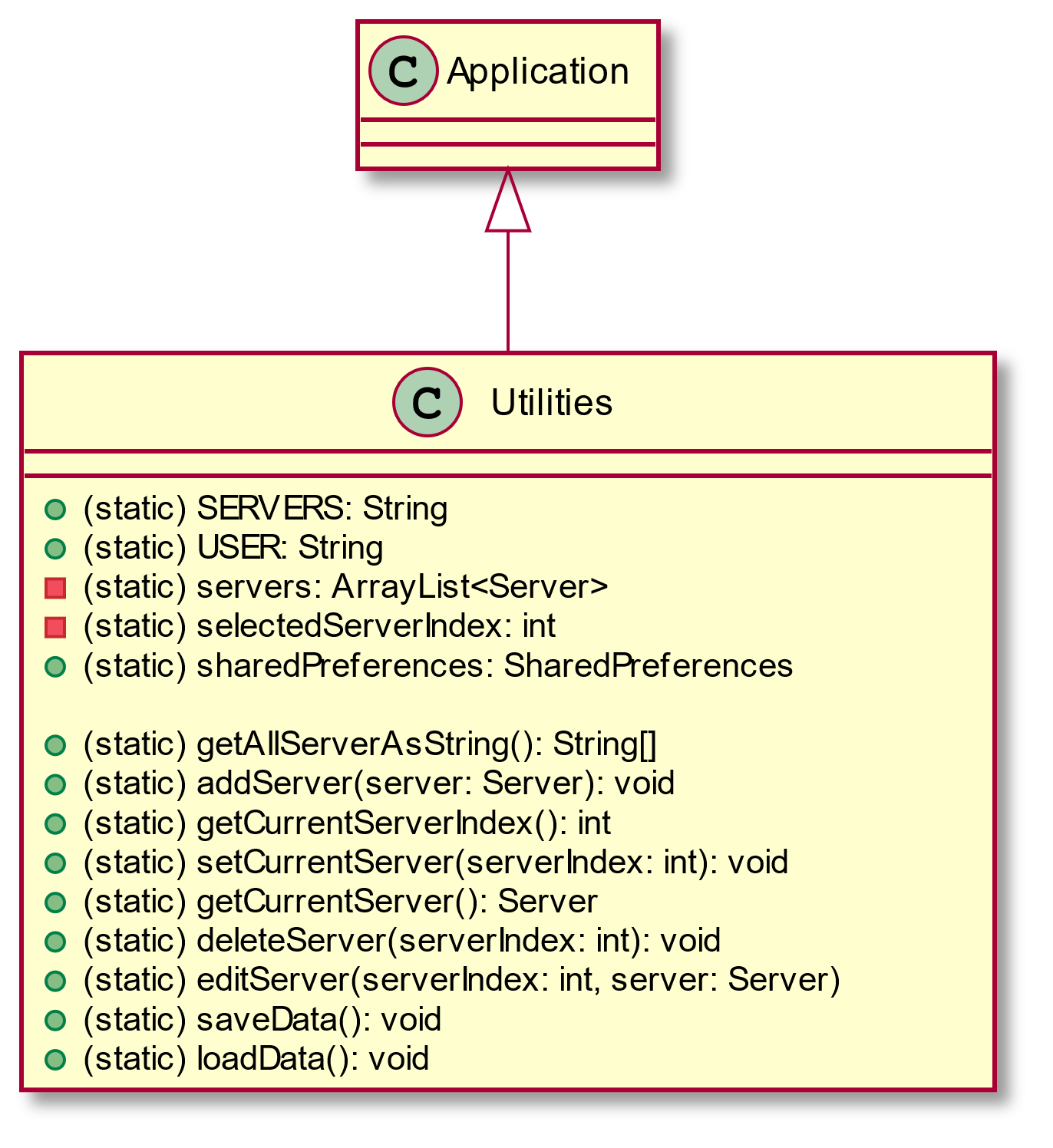
## Server

Ceci est juste une petite classe pour pouvoir stocker de manière facile les données d’un serveur. Au début une structure simple aurait été voulue mais java, ce langage magnifique ne supporte pas ça.



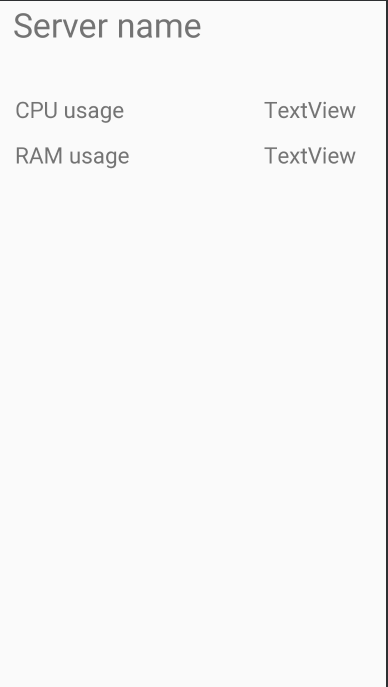
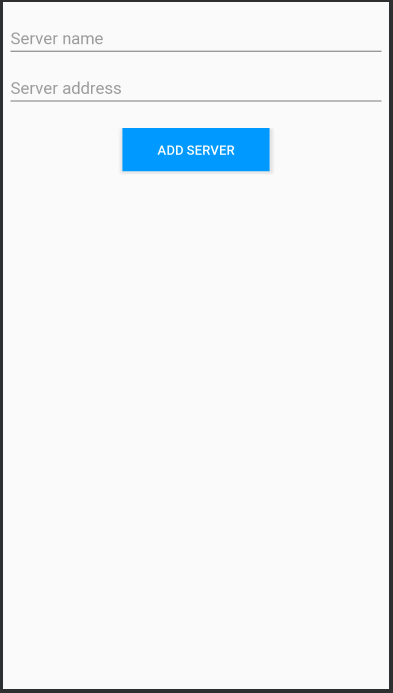
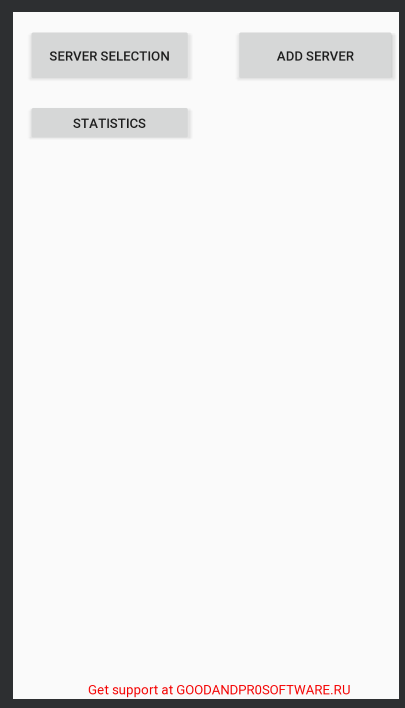
## Utilities

Cette class static gère les données qui doivent être passé d’une activité à l’autre et gère également le stockage de donnée dans les shared preferences.

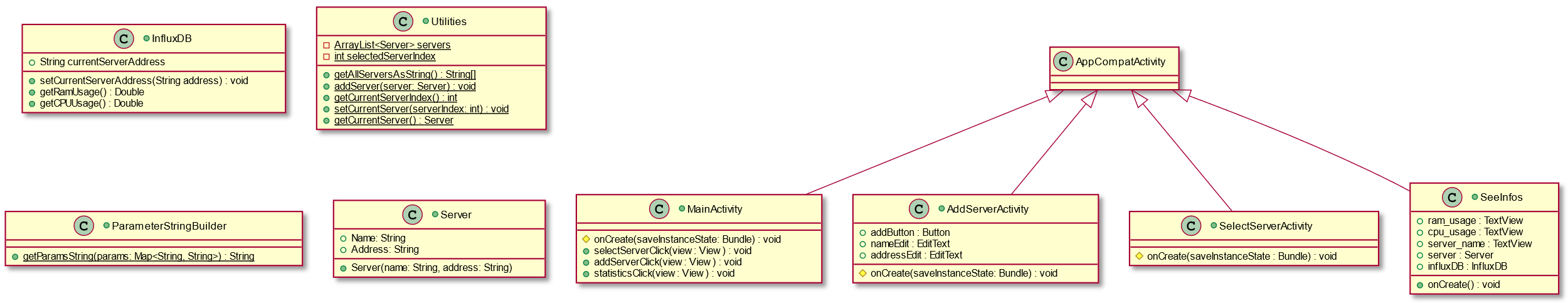


# Version 1.0

## Maquette



## Diagramme de classe

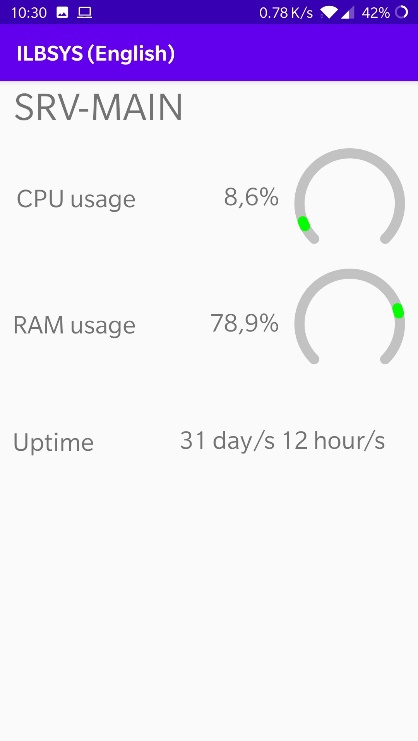


# Sprint 1

## Fonctionnalités

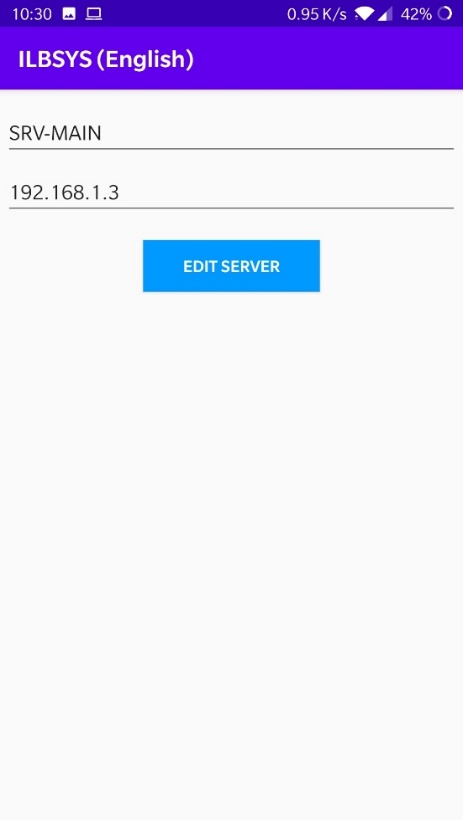
### Uptime

Ajout du temps en ligne depuis le dernier redémarrage est désormais dans la vue des informations serveur. La base de données rend l’uptime en secondes depuis le démarrage.



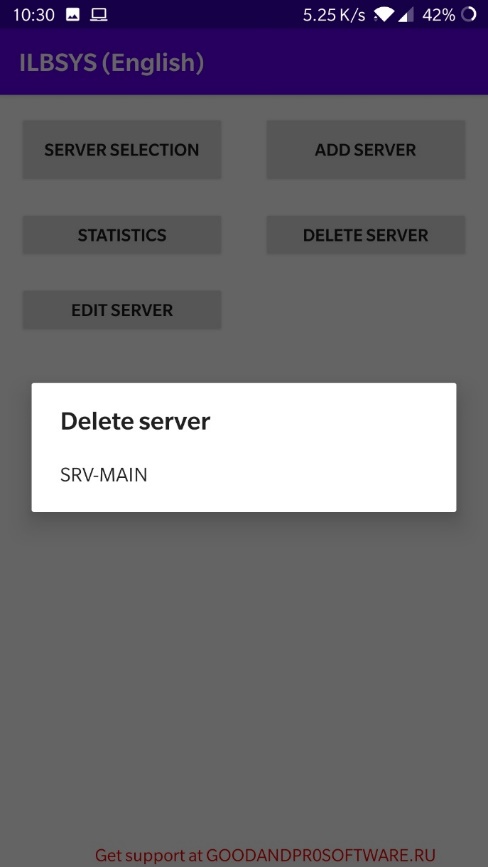
### Modification de serveurs

Modifier les données d’un serveur après coup est très utile dans le cas d’une erreur ou d’un changement d’adresse. Il suffit d’aller sur l’activité de modification de serveur, changer les entrées et ensuite cliquer sur sauvegarder.



### Suppression de serveurs

Ouvre une modale comme pour la sélection d’un serveur mais le supprime quand il est choisi.



# Sprint 2

## Fonctionnalité

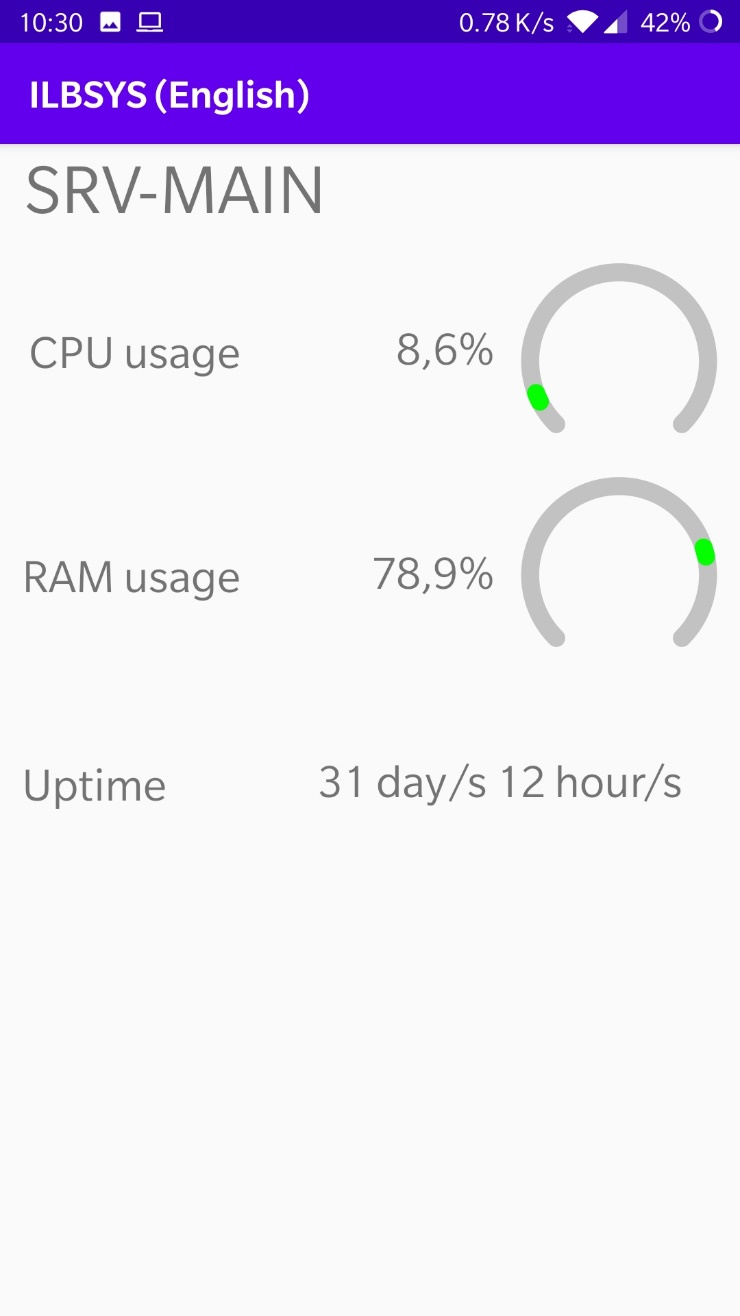
### Accès depuis l’extérieur

Jusqu’à maintenant le serveur InfluxDB était seulement accessible depuis l’intérieur de mon réseau. Une redirection de port a été créer pour pouvoir y accéder depuis l’extérieur. Il s’agit du port 8086 qui est le port par défaut pour InfluxDB.

L’adresse depuis l’extérieur pour faire fonctionner l’application est 185.97.136.203.

### Jauges pour les pourcentages

Pour avoir un affichage plus parlant pour les pourcentages comme pour l’utilisation du CPU et de la RAM des jauges ont été rajoutés. Plusieurs librairies ont dû être testé avant de trouver une qui est compacte et optimal pour les besoins de ce projet.



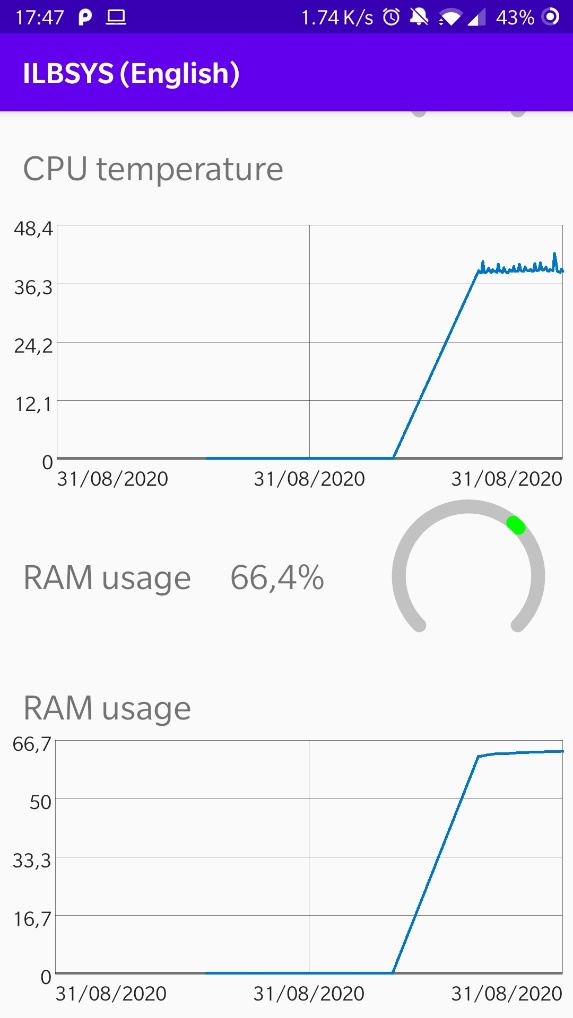
# Sprint 3

## Issue #2

Correction de l’erreur que l’application crashait si aucun serveur était dans les shared preferencies.

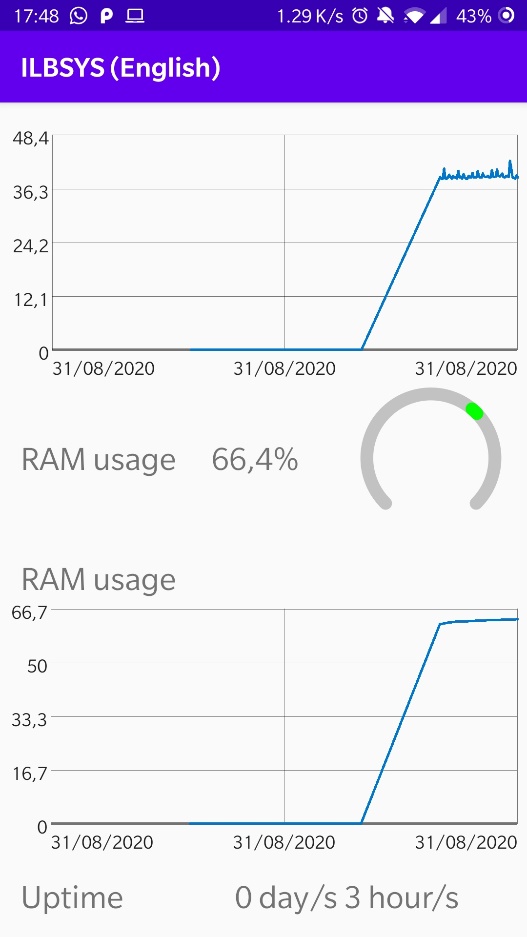
<https://github.com/ILikeBananas/ILBSYS_android/issues/2>

## Graphique pour la température du processeur

Un graphique a été ajouté sur la page des informations qui affiche les températures du processeur sur les derniers 6 heures. Pour cela la librairie « [jjoe64](https://github.com/jjoe64)» a été utilisé, celle-ci fonctionne bien même s’il n’existait pas de la documentation pour tout ce qui a été utilisé.

## Graphique utilisation RAM

Comme pour le graphique des températures CPU, la librairie de [jjoe64](https://github.com/jjoe64)a été utilisée.



## Réorganisation de la page d’informations

L’affichage et l’organisation des éléments sur la page d’informations ont été refaits à neuf pour permettre un affichage plus propre, moins de valeurs fixe et une gestion du mode paysage correcte. Une scrollview a été utilisée pour permettre d’afficher plus de données que la hauteur (ou largeur en mode paysage) du périphérique.

