

UE : Développement Mobile

Développement d'application de monitoring et contrôle d'automate

2020-2021
Pittonet Benjamin

Table des matières

Table des matières	3
1. Introduction.....	4
2. Contexte	5
3. Cahier des charges.....	5
3.1. Gestion Utilisateurs	5
3.2. Paramètres de connexion.....	5
3.3. Lecture et écriture des données.....	6
4. Réalisation	6
4.1. Structure générale de l'application	6
4.2. Design	7
4.3. Base de données.....	7
4.4. Gestion utilisateurs.....	8
4.5. Connexion Automates	8
4.6. Envoie de données	9
5. Améliorations et conclusions	11
6. Table des figures.....	11

1. Introduction

De nos jours, nous utilisons de plus en plus d'automates programmables. Ces automates nous facilitent bien la vie puisqu'ils peuvent effectuer des tâches très variées et requérant parfois beaucoup de précision.

Connus désormais sous l'appellation l'industrie 4.0, ces automates deviennent, à l'instar de notre vie quotidienne, hyper connectés. Cette évolution nous permet d'obtenir de manière automatisée de précieuses informations quant à l'activité de ces automates.

Nombre de pièces produites, tri de produits défectueux à la volée, ou encore connaître l'état de la machine afin de prévenir des pannes, ne sont là que quelques exemples de l'utilisation de cette hyper connectivité.

Cette collecte de données nous a permis de mettre en relation plusieurs machines afin de toujours pousser plus loin l'automatisation des tâches industrielles. Ajoutons désormais à cela l'univers Mobile, rendu populaire par l'utilisation de Smartphones ou de tablettes qui caractérise désormais l'« Internet des Objets » et nous obtenons alors un moyen rapide et efficace de non seulement monitorer toutes ces données, mais aussi de pouvoir envoyer des instructions aux automates de manière mobile.

REVOLUTION INDUSTRIELLE

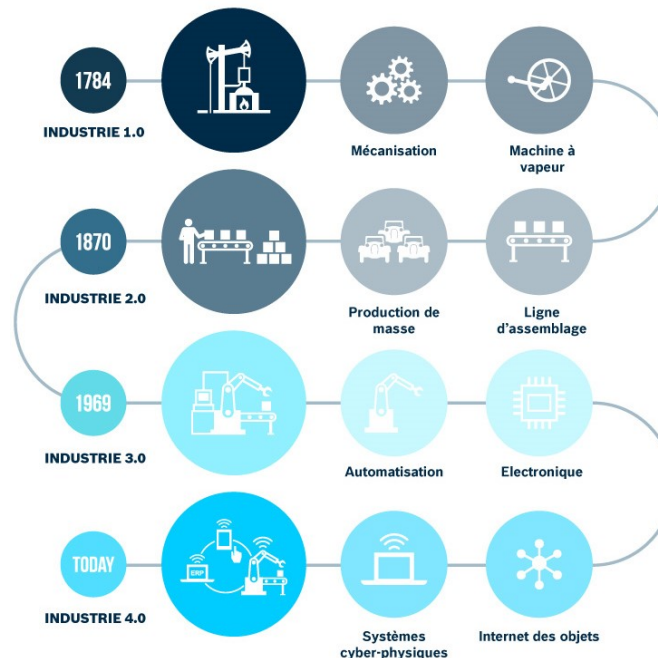


Figure 1: (R)Evolution Industrielle

2. Contexte

Ce projet, réalisé dans le cadre du cours de Développement Mobile, aura comme but le développement d'une application Android permettant d'interagir avec deux Automates industriels. L'application devra permettre la supervision des automates, mais également la commande à distance. Ces deux automates seront par ailleurs simulés via une machine virtuelle fournie par notre professeur.

3. Cahier des charges

Le cahier de charges détaillées ci après est également disponible dans le syllabus du cours de Développement Mobile.

L'objectif du projet est de réaliser une application permettant la supervision et la commande à distance de deux Automate :

- Automate S7-1516 2DPPN servant le conditionnement de comprimés
- Automate S7-1214C permettant l'asservissement de niveau de liquide

Le design général de l'application est laissé à la discrétion de l'élève.

3.1. *Gestion Utilisateurs*

Il est nécessaire de prévoir une gestion utilisateurs.

Un utilisateur sera représenté par plusieurs éléments :

- Son Nom
- Son Prénom
- Son Mail
- Son Mot de passe (4 caractères minimum)

La connexion à l'application se fera via le login de l'utilisateur (Mail) et son mot de passe.

Il existera deux types de profils utilisateurs :

- Les administrateurs : pourront lire les données des automates, envoyer des données aux automates, mais aussi gérer les utilisateurs
- Les utilisateurs « Basic » : pourront lire les données des automates

Au démarrage de l'application, un utilisateur Administrateur sera créé.

Tout autre utilisateur créé sera de type « Basic ».

Les permissions des autres utilisateurs seront gérées par l'administrateur qui pourra promouvoir ou rétrograder les utilisateurs.

3.2. *Paramètres de connexion*

Les paramètres de connexion devront être demandés lors de la connexion à un type d'automate.

Ainsi, lorsque l'utilisateur voudra se connecter, il devra renseigner :

- L'adresse IP
- Le rack
- Le slot

3.3. Lecture et écriture des données

Il est nécessaire de créer une interface visuelle soignée permettant l'affichage d'informations d'entrées et de sorties « machine » de façon claire et compréhensible.

Il est également nécessaire de prévoir un affichage permettant l'envoi de données vers l'automate reprenant le numéro de databloc et la variable au sein de ce dernier.

4. Réalisation

4.1. Structure générale de l'application

Lors de la première utilisation de l'application il sera demandé de créer un compte administrateur. Une fois l'administrateur créé, toute création d'utilisateur entraînera l'ajout d'utilisateurs « Basic ». Il est à noter qu'à ce stade, seul l'utilisateur administrateur initialement créé pourra modifier les autres utilisateurs.

Une fois authentifié, l'utilisateur arrivera sur le menu principal qui sera différent selon les permissions accordées à ce dernier. Un utilisateur basic ne verra que les options des automates et un bouton déconnexion alors qu'un administrateur aura aussi accès à la gestion utilisateurs.

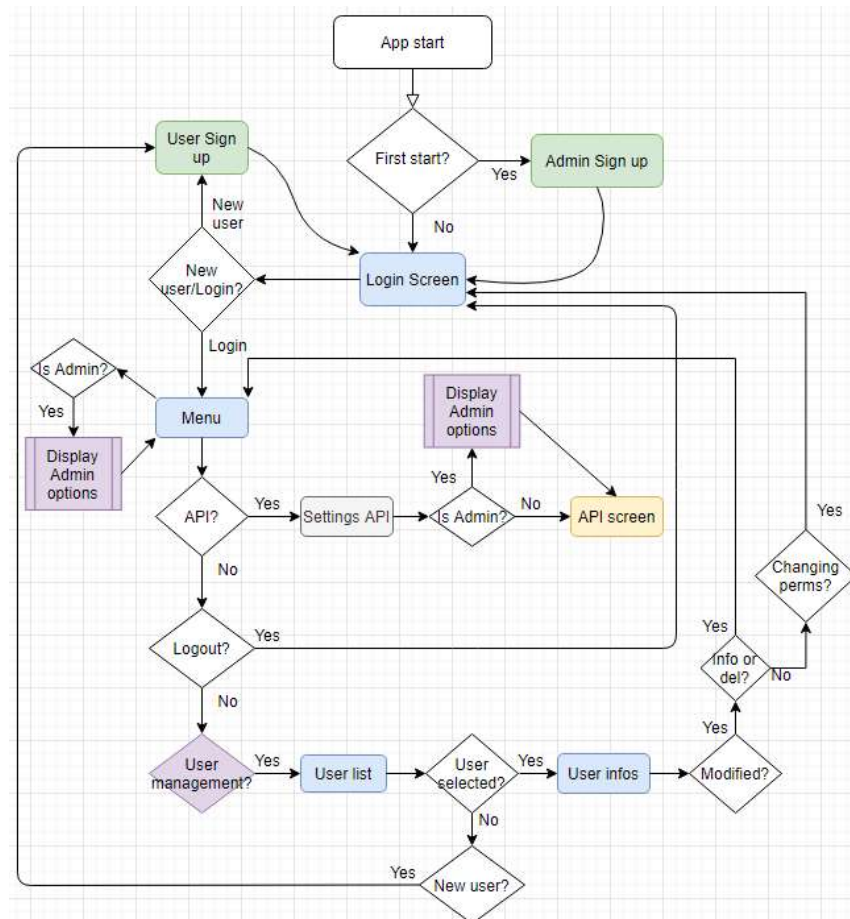


Figure 2: Algorithme général

Lors de la connexion aux automates, il sera demandé à l'utilisateur en cours de renseigner les informations de l'automate, à savoir : Adresse IP, Rack et Slot. Un utilisateur basic ne pourra que lire les états machines là ou un administrateur pourra lire et écrire des données dans les datablocs.

Lors de l'accès à la gestion utilisateurs par l'administrateur, une liste des utilisateurs sera affichée. L'administrateur pourra ensuite choisir quel utilisateur il souhaite gérer. Il pourra alors modifier les informations de l'utilisateur ainsi que ses permissions.

4.2. Design

Concernant le design général de l'application, je suis parti sur du bleu pour la couleur générale puis vert pour l'accentuation des boutons. J'ai choisi de rester simpliste dans le design, préférant quelque chose d'épuré mais efficace.

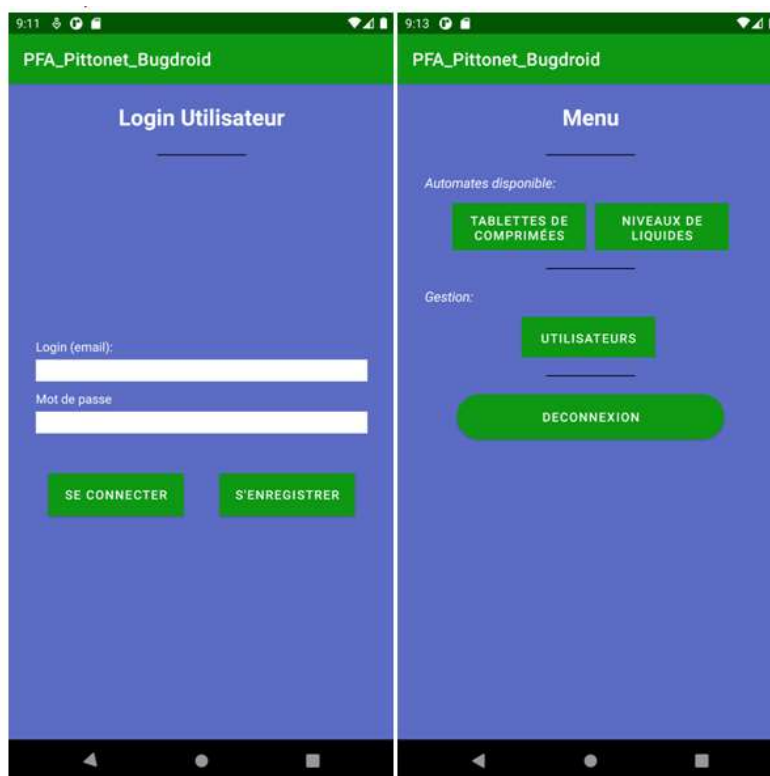


Figure 3: Design général

4.3. Base de données

La base de données gèrera les utilisateurs. Comme expliqué dans le cahier de charge, nous devons encoder quatre informations : le nom, prénom, mail et mot de passe. En plus de cela nous utiliseront un champ pour définir les permissions ainsi qu'un ID. Le login se faisant sur le mail de l'utilisateur, cette information sera reprise comme étant unique et ne pouvant pas avoir de doublons dans la base de données.

USERS_TABLE		
id	integer	PK
name	text	
surname	text	
mail	text	
password	text	
isAdmin	Boolean	

Figure 4: Table des users

4.4. Gestion utilisateurs

Lors de l'accès au menu principal, si l'utilisateur actuel est un administrateur, le bouton de gestion utilisateurs sera disponible.

L'affichage proposera une liste des utilisateurs actuels repris dans la base de données. L'administrateur pourra créer de nouveaux utilisateurs, mais aussi modifier les utilisateurs existants en cliquant sur un utilisateur puis en éditant les champs voulus.

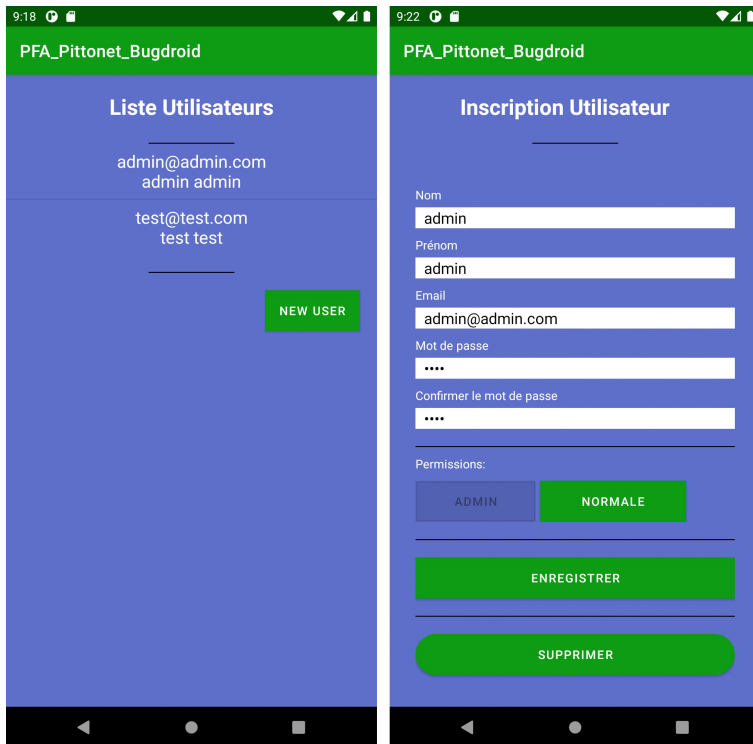


Figure 5: Gestion utilisateurs

4.5. Connexion Automates

Avant de pouvoir se connecter aux automates, tout utilisateur (admin ou basic) devra renseigner l'IP de l'automate qu'il tente de contacter ainsi que le rack et le slot de l'automate.

Adresse IP:
192.168.1.221

Rack
0

Slot
2

SE CONNECTER

Figure 6: Connexion Automates

Si les informations sont valides, une connexion est établie et l'utilisateur pourra alors voir en temps réel les informations envoyées par l'automate.

4.6. Envoie de données

Dans le cas où l'utilisateur est un Administrateur, il lui est également possible de cliquer le bouton de modification. Cela a pour effet d'afficher les boutons radio permettant de choisir le databloc, le champ texte permettant d'écrire la valeur à envoyer, ainsi que le bouton pour valider l'envoi des données.

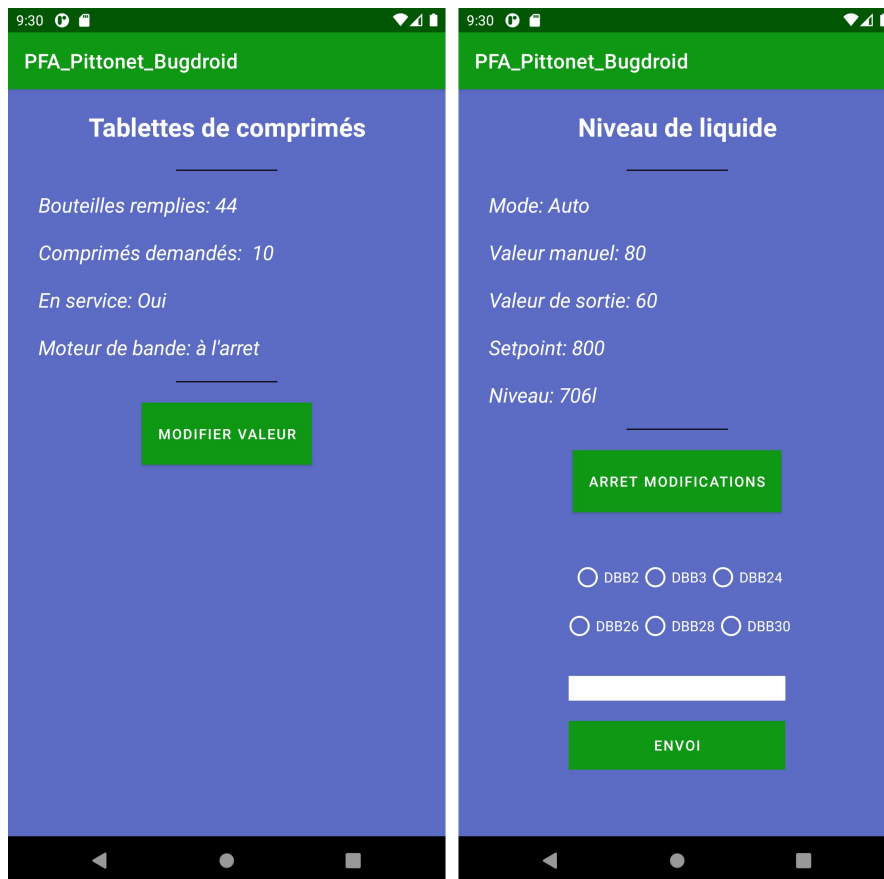


Figure 7: Lecture et écriture Automates

Afin de vérifier que l'envoi des données est correcte, nous devons tout d'abord ouvrir SIMATIC manager et charger le projet voulu. Après quoi nous pouvons dérouler les options jusqu'à atteindre les Databloc pour ensuite choisir le la DB5.

Cette procédure est décrite dans le syllabus du cours et renseigne également les registres de la DB5 utilisés dans le cadre du projet :

Automate de comprimés

Opérations d'écriture de 16 valeurs (BOOL) dans les Variables suivantes :

1. DB5.DBB5
2. DB5.DBB6

Opérations d'écriture de 8 valeurs (BOOL) dans la Variable suivante :

1. DB5.DBB7

Opérations d'écriture d'un BYTE (8 bits) dans la Variable suivante :

1. DB5.DBB8

Opérations d'écriture dans une variable INTEGER (16 bits) :

1. DB5.DBW18

Automate de niveaux de liquide

Opérations d'écriture de 8 valeurs (BOOL) dans la variables suivantes :

1. DB5.DBB2

Opérations d'écriture de 8 valeurs (BOOL) dans la Variable suivante :

1. DB5.DBB3

Opérations d'écriture dans 4 variables INTEGER (16 bits) :

1. DB5.DBW24
2. DB5.DBW26
3. DB5.DBW28
4. DB5.DBW30

S7-1214Cfw4.1...
CPU 1214C

Figure 8: Registres d'écriture des Automates (voir Cours)

Nous pouvons donc désormais vérifier l'envoi de données depuis l'application et la voir apparaître en temps réel dans VM du projet.

9:45

PFA_Pittonet_Bugdroid

Niveau de liquide

Mode: Auto

Valeur manuel: 80

Valeur de sortie: 71

Setpoint: 800

Niveau: 706l

ARRET MODIFICATIONS

☐ DBB2
☒ DBB3
☐ DBB24

☐ DBB26
☐ DBB28
☐ DBB30

5

es	nom	Type	Valeur initiale	Valeur en cours
0.0	IO_B_1	BYTE	B#16#0	B#16#20
1.0	IO_B_2	BYTE	B#16#0	B#16#00
2.0	IO_B_3	BYTE	B#16#0	B#16#00
3.0	IO_B_4	BYTE	B#16#0	B#16#05
4.0	IO_B_5	BYTE	B#16#0	B#16#00
5.0	IO_B_6	BYTE	B#16#0	B#16#00
6.0	IO_B_7	BYTE	B#16#0	B#16#00
7.0	IO_B_8	BYTE	B#16#0	B#16#00
8.0	IO_B_9	BYTE	B#16#0	B#16#00
9.0	IO_B_10	BYTE	B#16#0	B#16#00
10.0	IO_B_11	BYTE	B#16#0	B#16#00
11.0	IO_B_12	BYTE	B#16#0	B#16#00
12.0	IO_B_13	BYTE	B#16#0	B#16#00
13.0	IO_B_14	BYTE	B#16#0	B#16#00
14.0	IO_W_15	BYTE	B#16#0	B#16#00
15.0	IO_W_151	BYTE	B#16#0	B#16#00

Figure 9: Envoi de données aux Automates

5. Améliorations et conclusions

Beaucoup de choses pourraient être améliorées sur ma version du projet.

L'affichage des données pourraient inclure un visuel de l'automate afin que cela soit plus parlant pour les utilisateurs. L'utilisation d'un singleton permettrait aussi de simplifier la gestion des données de l'utilisateur courant. Le design, bien que valable, mériterait pas mal d'améliorations également et il est bien entendu toujours possible d'améliorer la stabilité du code ainsi que la sécurité.

Ce projet fut assez intéressant à plusieurs niveaux. Il m'a permis de travailler les interactions entre une application et un serveur applicatif, ce qui de nos jours s'avère crucial pour la plupart des applications. Il permet également de mieux comprendre les enjeux des applications android dans le cadre d'une utilisation se destinant à des automates industriels.

6. Table des figures

Figure 1: (R)Evolution Industrielle.....	4
Figure 2: Algorithme général.....	6
Figure 3: Design général.....	7
Figure 4: Table des users.....	7
Figure 5: Gestion utilisateurs.....	8
Figure 6: Connexion Automates.....	8
Figure 7: Lecture et écriture Automates.....	9
Figure 8: Registres d'écriture des Automates (voir Cours).....	10
Figure 9: Envoi de données aux Automates.....	10