

# Compte-rendu 4 : réunion du 24/10/2024

## SIForms : Application de saisie de formulaires

Auteurs :  
**Adèle DESMAZIERES**  
**Léa EL ABOUD**  
**Kevin XU**

Destinataires :  
**Paul GUYOT**  
**Julien KERLIDOU**  
**Cédric BESSE**  
**Lamia LARAQUI**

Responsable : **Adèle DESMAZIERES**

Type : **Compte-rendu 4 : réunion du  
24/10/2024**

Date : **24/10/2024**

Nombre de pages : **4**

Statut : **Final**

Adresse : [typst.app/PISTL](https://typst.app/PISTL)

Version	Date	Modification	Rédacteur
v.0.1	24/10/2024	Rédaction du CR	Adèle DESMAZIERES

## Ordre du jour

<b>1. Introduction : offre de stage</b> .....	<b>1</b>
<b>2. Révision du STBE</b> .....	<b>1</b>
2.1. Présentation du projet .....	1
2.2. Comparaison des technologies .....	1
2.3. Use Case .....	1
2.3.1. Diagramme front-end .....	1
2.3.2. Scénario et tests .....	1
2.3.3. Diagramme du back-end .....	2
<b>3. Précisions sur les formulaires</b> .....	<b>3</b>
3.1. Risques .....	3
<b>4. Suite</b> .....	<b>4</b>

## 1. Introduction : offre de stage

SI a reçu le message de campagne d'appelle d'offre de stages par l'université. Ils vont ouvrir une place en stage, après les congés de Paul, soit dans une à deux semaines.

## 2. Révision du STBE

### 2.1. Présentation du projet

Très bien rédigé, très clair, félicitations de Paul.

### 2.2. Comparaison des technologies

Très important pour les professeurs, et plutôt bien. Il est intéressant de laisser la phrase sur la préférence de Paul pour le Django, car ça montre une analyse des préférences client en plus des caractéristiques techniques.

#### Parenthèse sur le cycle en V

Le Master STL est l'héritier d'un ancien DESS de Génie Logiciel (animé par Giles Bien ?), qui enseignait déjà le cycle en V. Et à l'époque c'était déjà considéré comme une méthode archaïque, même si elle a des atouts, notamment quant à l'analyse des besoins.

Pour le serveur en Python, il serait pertinent de questionner ce choix, et mentionner si Go aurait été plus adapté par exemple. Après la comparaison, on peut expliquer que SI utilise Python et Django, ce qui est plus pratique pour intégrer notre outil dans leur système au final. Pour la comparaison des données, faire attention au fait que YAML est un sous-ensemble de JSON, expliquer pourquoi on a opposé les deux.

### 2.3. Use Case

#### 2.3.1. Diagramme front-end

Pas de commentaire pour le diagramme du front-end.

#### 2.3.2. Scénario et tests

##### 2.3.2.1. Front-end

- Bcp de détails, pas vrmt nécessaire du côté de la compagnie, mais c'est ce qui est demandé par les enseignants.

- Dans le monde de l'entreprise, il y a une notion de recette : il faut que la recette soit bonne, sinon les dev doivent refaire des corrections sans être payé.

#### 2.3.2.1.1. Créer un compte

- Test nominal : il manque le n°4 dans la liste.
- Changer le mail d'exemple pour mettre un email vraiment incorrect (par exemple pas de @)
- Il n'est pas possible de vérifier la validité d'un email, donc retirer cela. Par contre il faut bien vérifier que l'email n'existe pas déjà dans la base.
- Enlever l'alternative reçoit email avant d'avoir créé un compte car à ce moment là le serveur ne connaît pas l'email de la personne.
- Exception à ajouter : si la personne ne reçoit pas son mail (fire and forget ?). Choisir ce qu'il faut faire dans ce cas.

#### 2.3.2.2. Se connecter à un compte

- Connexion : prévoir le cas sans réseau
- Corriger la typo : « préchargé » (au lieu de « précharger »)

#### Fonctionnement des QRcodes

Les QRcodes ont une URL, qui pointe soit vers un site internet, soit vers un « scheme » qui correspond à qqc dans notre application. Il faut trancher pour le produit entre ces deux options. A priori, il faudrait plutôt une URL vers une page internet, pour récupérer en ligne les données du formulaire.

- > Se documenter sur comment fonctionnent les QRcodes
- > Se documenter sur l'ouverture automatique d'une application mobile

#### 2.3.2.3. Envoi d'un formulaire

- Retirer le cas « envoi immédiat », car les forms sont trop grands. Afficher forcément « formulaire en cours d'envoi », comme pour un mail quand on l'envoie.
- Prévoir dans les tests un **envoi partiel** du formulaire, avec une connexion internet instable qui coupe pendant l'envoi. Cela arrivera souvent (par exemple avec un opérateur mobile avec une connexion de basse qualité, et quand les techniciens seront dans des sous-sols).

#### 2.3.2.4. Langue des formulaires

Il faut trancher entre configurer la langue par formulaire ou globalement pour le compte. Rendre cohérent ce choix entre le test du use case et la maquette (actuellement : le test configure la langue pour un formulaire, alors que la maquette configure la langue pour le compte).

#### 2.3.3. Diagramme du back-end

**Priorité de l'application mobile sur le serveur** : Ils veulent un serveur mock, donc pas besoin d'interface utilisateur du côté de SI. En effet, l'objectif est de faire une preuve de concept sur la partie application mobile du sujet.

Il suffit de spécifier le protocole de l'interface avec le serveur. Les employés de SI peuvent utiliser des fichiers statiques dans le serveur (pas besoin de fichiers dynamiques, pas le cœur du sujet). Mettre un fichier à la main dans un dossier est ok pour créer un formulaire pré-rempli. Le chef de Florant a des idées pour exploiter l'application, donc

on doit éviter de perdre du temps sur back-end, qui sera de toute façon intégré à leur propre système. Les attentes sont sur l'application mobile.

- Renommer « front-end » en application mobile.
- Remplacer appels de fonction avec requête HTTP (POST /route, REPONSE /...)
- Décrire avec les principes Rest d'API
- Back-end serveur web. Front-end application web. Communiquent avec fonctions HTTP.
- Connexion : voir meilleures méthodes protocoles. POST pour créer un objet session. Cet objet sera utilisé dans les requêtes suivantes.
- Le serveur ne valide pas la réponse au form. Mais il dit à l'appli qu'il a bien reçu ce qu'il cherche. Ne pas faire les flèches retour.

### 3. Précisions sur les formulaires

Il y a 3 classes de formulaires, et un formulaire peut être pré-rempli.

**Formulaire pré-rempli** : a déjà des réponses à certaines questions, qui réduit le nombre de questions, ou alors dont certaines réponses sont déjà remplies ? Il faut expliciter entre les deux. Est-ce que les réponses pré-remplies peuvent être modifiées par le technicien ? Oui.

**Classes de formulaire** :

- visite technique
- certificat d'installation
- information site

**Le schéma de formulaire** : JSON schema qui décrit tous les champs du form. Il y en a 3, ce sont les 3 classes de formulaires. **Template de formulaire**. Ils répondent à une grammaire.

**Réponse au form** : valeurs remplies pour un objet donné. Peut être pré-rempli avec des valeurs initiales, qui peuvent être complétées ou modifiées par le technicien. Les réponses sont branchées à 2 objets dans le back-end actuel à SI : **un site** et **une installation** (une armoire électrique). Chaque formulaire a une partie dédiée au site, et une partie dédiée à une répétition de n installations. N'importe quelle partie peut être pré-remplie.

Un site est donc représenté par un objet template et un objet valeurs réponses, et idem pour les installations.

La syntaxe générale d'un JSON-schema :

- template (utilisé pour la classe du formulaire)
- valeurs (utilisé pour la réponse au formulaire)
- ordre interface (pas utilisé)

Un technicien répond à un site et à n installations.

**Accéder à un formulaire** : accéder au schéma et aux valeurs, pour le site et les installations.

**Le QRcode** permettre d'accéder au site d'identifiant 3 comprenant les installations 5, 6 et 7. Le technicien peut ajouter une installation, sans que cela ne modifie le template (cela modifiera seulement les valeurs réponses).

L'application ne renvoie au serveur que la réponse avec les valeurs, pas le template.

#### 3.1. Risques

**Risque** : changement du template dans le back entre la réception par le technicien et l'envoi de sa réponse.

**Solution** : versionner les formulaires. Chaque template a un numéro de version. Si le template a changé, on ne fait pas re-répondre le technicien et on conserve ses réponses pour la version qu'il a eu.

Eventuellement ajouter un état « validé par SI » dans la liste des formulaires ?

**PDF** : où le pdf est-il créé ? Serveur. Permettre l'affichage du pdf dans l'appli.

A mentionner dans le STBE : comment est représenté le dynamisme des formulaires. Dans le template : écrire ce qui est conditionné à autre chose, le représenter visuellement dans le formulaire en grisant ou autre, pour les formulaires dynamiques. Utiliser **ReactJSONschemaForm** pour ça. Pour les blocks répétés, pour les questions qui s'affichent en fonction d'une question précédente.

Figma : donner une classe aux formulaires.

## 4. Suite

Prochaine réu le 5.

Reste à préciser : grammaire.