

Compte-rendu 2 : réunion du 08/10/2024

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 2 : réunion du
08/10/2024**

Date : **15/10/2024**

Nombre de pages : **3**

Statut : **Final**

Adresse : typst.app/PISTL

Version	Date	Modification	Rédacteur
v.0.1	08/10/2024	Notes de réunion sur LibreOffice	Adèle DESMAZIERES
v.0.2	15/10/2024	Migration sur Typst	Adèle DESMAZIERES

Table des matières

1. Ordre du jour	1
2. Compte-rendu de réunion	1
2.1. Comment partager nos documents entre nous et avec Smart Impulse ?	1
2.2. Retours sur la maquette Figma	1
2.3. Retours sur la comparaison des technologies	2
2.3.1. Tech user-end	2
2.3.2. Tech app mobile	2
2.3.3. Tech serveur	2
2.3.4. Tech données	2
2.4. Diagramme des Use Case	2
3. Synthèse et planification	2
4. Lexique	3

1. Ordre du jour

1. Quel serveur pour héberger nos documents rédigés et pour le dépôt de code ? Forge interne à Smart Impulse, Google Drive, GitHub... ?
2. Présentation de la maquette Figma.
3. Présentation des comparaisons de technologies.
4. Présentation du schéma des Use Case.

2. Compte-rendu de réunion

2.1. Comment partager nos documents entre nous et avec Smart Impulse ?

Notre travail est open source, à mettre sur un système de gestion de version publique comme GitHub. Les formulaires et les éléments métiers sont privés. Ceux-ci sont partagés sur un GoogleDrive avec les professionnels.

2.2. Retours sur la maquette Figma

- trancher si on peut remplir un formulaire sans s'authentifier ou non
- ajouter une possibilité de scanner QRcode depuis un écran où on est déjà authentifié
- trancher s'il est possible de scanner le QRcode dans l'app ou en dehors
- trancher si on envoyer un lien ou un QRcode ?
- Il existe un QRcode est sur le compteur, qui pourrait servir, mais pas forcément
- Il y a 3 types de forms en fonction de la visite que fait le technicien. Quand il est sur site, il répond qu'à un seul form. Les 3 forms sont souvent remplis par des personnes différentes. Voici les 3 forms :
 - pré-installation (form qu'on a testé sur le JotForm) : prend en photo le site pour préparer l'installation, informations techniques sur le site, formulaire atteint à partir d'un lien
 - post-installation : formulaire de confirmation d'installation
 - information site et équipement : informations sur le site et ses équipements
- le lien/QRcode amène à un type de formulaire pré-rempli, pas à un choix
- feature bonus si on peut : possibilité de remplir un forms sans avoir de lien, en choisissant un formulaire vierge depuis l'app
- statut des forms : « attente connexion » = annulable (remise du doc en mode « brouillon »). « Envoie en cours » = pas d'annulation possible car déjà commencé à envoyer les info au serv

2.3. Retours sur la comparaison des technologies

2.3.1. Tech user-end

Coté user : proposer la progressive web app, une solution intermédiaire à comparer. Mais partie offline difficile d'après Paul. Le mentionner.

Application web comme Discord ?

2.3.2. Tech app mobile

ReactNative : argument supplémentaire → maintenance plus facile pour SI car ils font déjà du React. C'est plus simple de trouver des dev de ReactNative + on aimerait se former à cette tech. ReactNative : à double tranchant, à cause de la gestion de dépendances. Mais nouveau truc qui aide : « expo », gestionnaire de paquets. Règle gestions de dépendances.

Flutter : plus apprécié par certains.

Corriger typo « ReactNativ » dernière slide.

2.3.3. Tech serveur

À noter dans slides : serveur en Python imposé par SI

Flask : Paul aimerait faire moins de Django, mais pas recommandé si on a jamais fait de Django.

Django : utiliser la dernière version. ORL intégré, relation avec PostgreSQL facile.

2.3.4. Tech données

Faire attention à différencier ces deux choses :

- Les données relationnelles de l'application (utilisateurs, mdp, compte, identifiants des formulaires remplis, identifiants de leurs réponses...). Conseillé PostgreSQL car SI en fait déjà.
- Les données de réponse aux forms et les fichiers transférés par les techniciens. Le stockage des fichiers lourds est hors scope, mais intéressant de s'y pencher. SI utilise Google Drive, Drop Box, techno S3. Possible également de simplement stocker les fichiers à côté du serveur en local ?

À mentionner dans les comparaisons : Pour formater les formulaires et les réponses, SI utilise le format JSON, mais il faut évaluer les alternatives. Étudier le format de l'objet qui représente le form (champs, radiobox...) et le format de l'objet qui stocke les réponses de l'utilisateur. Spécifier les données échangées avec le serveur, comment est gérée la soumission partielle, si les réponses textuelles sont envoyées avant les photos attachées, etc.

2.4. Diagramme des Use Case

Menu des langues (pas de géolocalisation, la « localisation » signifie adapter les traductions des textes et chiffres.

Éventuellement ajouter une traduction des textes de l'app, mais surtout permettre de traduire les forms. Mettre à jour le use case en fonction de la maquette de Léa. PDF créé soit dans l'app soit côté serv, à choisir. Choix à répercuter sur la maquette.

3. Synthèse et planification

- Mettre à jour la maquette.
- Faire converger le schéma des use case avec la maquette, il faut plus de cohérence entre nos docs.

- Mettre à jour la comparaison des techno avec plus de détails.
- Décrire les données échangées entre serveur et application. Séparer les données relationnelles des données brutes.
- Mettre tous nos documents sur le Drive partagé en les gardant organisés.
- Prochaine réunion : jeudi 17 octobre à 15h, sans Kevin qui ne pourra pas y assister.

4. Lexique

- form : formulaire
- app : application
- serv : serveur
- SI : Smart Impulse