

PDP cahier besoin

SENRENS–MAUNY max

March 2023

1 Introduction :

Ce document à pour but de décrire un Cahier des besoins à propos d'un logiciel de questionnaire/sondage ayant certaines propriétés manquantes dans la majorité des alternatives existantes. Cela dans le cadre de l'UE de projet de programmation du master informatique de l'université de Bordeaux.

2 Description des besoins généraux et du domaine :

il est question pour ce projet de faire un site de questionnaire. Pour avoir un site de questionnaire fonctionnelle, il faut au moins qu'il soit possible de créer un questionnaire, de l'envoyer à d'autres personnes, que ces personnes puissent y répondre et enfin que le créateur du questionnaire puisse récupérer les réponses. Voici donc la base de ce que nous visons. Cependant dans le cadre de ce projet il faudra aussi porter une attention particulière à la mise en place d'un certain nombre de propriété supplémentaires:

- La possibilité d'établir un ordre de préférence entre différent choix.
- L'envoi automatique d'un mail personnalisé aux utilisateurs qui répondent à un questionnaire.
- la possibilité d'importer et d'exporter un questionnaire dans un format texte comme par exemple json.
- L'existence d'un outil de personnalisation graphique de questionnaire orienté vers une utilisations pour des informaticien(ne)s.
- Avoir une accessibilité des données relative à un questionnaires restraints à l'administrateur du questionnaires seulement. Et en particulier non accessible par l'administrateur du site en lui même.

Ceci étant dis il n'est pas question pour ce projet de mettre en place une solution rivalisant avec l'existant mais de s'intéresser principalement aux propriétés manquantes ou rares dans l'existant tout en prodiguant les fonctionnalités de

bases attendues pour ce genre d'outils. Les deux ensembles de propriétés étant explicitement listés ci dessus.

3 Description de l'existant :

Google Forms, LimeSurvey, Microsoft Forms, Survey Monkey, Framaforms... (compléter)

4 Description des besoins :

4.1 Non fonctionnels

-Confidentialité des données vis à vis de l'administrateur réseau.

Priorités : Peu importante. Spécifié dans le cahier des charges mais non nécessaire au bon fonctionnement du site et non primordial vis à vis de l'objectif premier.

Robustesse : La robustesse se doit d'être importante, si cette propriété est implémenté il est primordial qu'elle soit robuste. Si on promet un certain degré de confidentialité il faut absolument le tenir.

Faisabilité : difficile, à approfondir avec l'analyse du backend.

Difficultés techniques: à approfondir aussi.

-Peu de dépendances.

Priorités : Importante, spécifié dans le cahier des charges, mais non nécessaire au bon fonctionnement du site.

Faisabilité : Difficile sur le papier en ayant choisi de travailler avec un framework. Cependant le framework quasar que nous avons choisi d'utiliser comporte une commande permettant d'éliminer les dépendance non nécessaire à l'application ce qui permet de grandement réduire la quantité de dépendance par rapport à une utilisation naïve d'un framework.

Difficultés techniques:

-Personnalisation par informaticien/ne

Priorités : Importante. Spécifié dans le cahier des charges, mais non nécessaire au bon fonctionnement du site.

Faisabilité : Oui, modifier la mise en page d'un site est loin d'être impossible, d'autant plus qu'ici cela visera une variété restreinte d'objets. Si ces objets sont bien structurés il ne devrait y avoir aucun mal à les composer entre eux et de faire en sorte que leurs aspects n'interagissent pas négativement entre eux.

Difficultés techniques: D'une part l'insertion de code et d'autre part faire en sorte que le questionnaire reste cohérent avec cette injection. Il faut faire attention à bien séparer l'affichage du reste du code et qu'il ne puisse pas y avoir d'effets indésirables. Il faudra aussi s'assurer de la cohérence de l'affichage du

questionnaire peu importe les personailsation apportées par l'utilisateur.

-Versions téléphone.

Priorités : Bonus. Non nécessaire et non spécifié mais interessant tout de même.

Faisabilité : Oui, il semble tout à fait raisonnable de penser qu'un créateur de questionnaire ait une version téléphone, c'est d'ailleurs le cas pour certains existants.

Difficultés techniques: à définir

4.2 Fonctionnels

-Créer un questionnaire.

Priorités : Essentiel. Propriété nécessaire, sans ca il n'y a pas de site.

Robustesse : Peu importante, on peut se permettre qu'il y ait parfois des bugs lors de la créations de questionnaire tant que ces bugs ne sont pas récurrent et n'impact pas les autres potentiels applications en cours de fonctionnement chez l'utilisateur.

Faisabilité : Oui

Difficultés techniques: Injection de code "en temps réel" par l'utilisateur.

-Récupérer les données d'un questionnaire.

Priorités : Essentiel. Propriété nécessaire, sans ca il n'y a pas de site.

Robustesse : Importante, si un questionnaire est créé et remplis il est problématique de ne pas pouvoir accéder aux résultats. Si ce sont des bugs qui impliquent une impossibilité temporaire d'accéder aux données alors cela est raisonnable, si les données demeurent à jamais inaccessible ca ne l'est plus.

Faisabilité : oui

Difficultés techniques: récupérer les informations de plusieurs utilisateur "répondant", et cela sur une plage horraire arbitraire. rassembler ces données en une structure lisible et accessible seulement par le créateur du questionnaire.

-Plusieurs type de réponse:

1. ordre de preference entre plusieurs choix.
2. réponse texte.
3. réponse oui/non..
4. case à cocher.
5. tableaux.

6. autre...

Priorités : Essentiel. Propriété nécessaire, il n'est pas question de faire des questionnaires n'ayant que des réponses oui/non. La première fonctionnalité du cahier des charges est de pouvoir établir un ordre entre plusieurs choix, il faut au moins cela.

Faisabilité : oui, l'existant le fait bien et des prototypes mis en place de notre côté montre aussi qu'il est possible de créer et concaténer différents types de questions dans un questionnaire.

Difficultés techniques: Pour l'ordre par exemple permettre que l'ordonnement soit partiel et qu'il ne soit pas nécessaire d'ordonner toutes les possibilités. Cela est particulièrement utile s'il y en a beaucoup.

-Envois automatique d'un mail à la fin du remplissage.

Priorités : Importante. Spécifié dans le cahier des charges, mais non nécessaire au bon fonctionnement du site.

Vitesse : Rapide. Il ne semble pas nécessaire que l'envoi soit instantané mais il ne faut pas non plus qu'il ait lieu plusieurs heures après l'envoi des réponses au questionnaire. Au plus quelques minutes de latence.

Robustesse : relativement importante. L'email peut servir comme confirmation que le questionnaire a bien été rempli ou avoir d'autres utilités, il faut être quasiment certains qu'un questionnaire rempli implique un email envoyé (à la bonne personne)

Faisabilité : oui, il est possible d'automatiser l'envoi de mails personnalisés.

Difficultés techniques: gérer les cas de mauvaises adresses mail ou de réponses par exemple.

-Import export de fichiers texte (json).

Priorités : Importante. Spécifié dans le cahier des charges, mais non nécessaire au bon fonctionnement du site.

Vitesse : Très rapide. Il ne doit pas se passer plusieurs minutes entre l'import d'un fichier et la création du questionnaire associé, autrement autant le faire "à la main".

Robustesse : relativement importante, on veut que les questionnaires soient bien générés en accord avec les fichiers texte importés. Il peut y avoir des bugs tant qu'ils ne rendent pas l'application inutilisable et qu'ils ne compromettent pas les autres applications potentielles de l'utilisateur. De même il ne faut pas que les bugs soient récurrents et si de tel bug surviennent il faut que l'utilisateur puisse importer son fichier sans trop attendre (refresh la page par exemple).

Faisabilité : oui, implémenté dans des prototypes.

Difficultés techniques: La transcription texte \rightarrow code et code \rightarrow texte qui se doit d'être relativement unique, on aimerait pouvoir produire une bijection.