Rapport PS6

LOSCIALE Vivian

AYOUB Hamza

MARTIN Thomas

KHAYATI Hiba

Nom d’équipe : TooTiredToWork

Table des matières

Les Personas et leurs scénarios 2

Dorothéa 2

Scénario 2

Léontine 2

Scénario 4

Baudoin 4

Scénario 4

Présentation de l’architecture Client Serveur 6

Front 6

Back 6

Présentation de l’évaluation croisée et analyse des résultats obtenus 9

Conclusion et perspective 10

Annexes 11

Répartition des tâches dans l’équipe 12

Vidéo de démonstration finale 13

# Les Personas et leurs scénarios

Le but de ce projet est de créer une application de quiz pour les personnes âgées étant atteint de troubles spécifiques. Dans notre cas, nous nous sommes penchés sur la thématique des personnes âgées ayant des problèmes visuels.

Pour mener ce projet à bien, nous avons utilisé des personas ayant chacun un scénario différent. Nous avons à notre actif 3 personas différents. Une aide-soignante, et deux personnes âgées ayant des problèmes de vue différents.

## Dorothéa

Dorothéa a 38 ans, elle est aide-soignante et s’occupe des personnes âgées dans le centre. Elle est en charge de créer les quiz et/ou de les maintenir. Comme elle est très occupée, elle aimerait une création simple et rapide du quizz. Elle aimerait aussi pouvoir modifier ou supprimer un quizz si celui-ci est erroné ou n’est plus utile.

### Scénario

Je m’occupe de la création des quizz.

Pour cela,

* Je vais dans le mode création afin de gérer les quizz.
* Je choisis d’éditer le thème qui contient le quiz en sélectionnant les petits rouages qui apparaissent dans la case du thème.

Je veux créer un nouveau questionnaire.

Pour cela,

* Je défile dans le carrousel afin d’arriver au bouton + pour ajouter un quiz.
* Je clique dessus.
* Je donne un nom au futur quizz.
* Je clique sur créer.
* Dans la page d’édition du quizz que j’ai créé, je crée une nouvelle question en appuyant sur le bouton +.
* Je remplis la box dédiée à la question puis je remplis les box destinées aux réponses.
* Je confirme ma saisie et je crée d’autres questions.
* Je pense que le niveau du quizz est difficile, je mets donc le niveau de difficulté à 3.

## Léontine

Léontine a 82 ans, elle a un glaucome[[1]](#footnote-1). Elle a l’habitude d’utiliser l’outil informatique pour faire ses recherches personnelles. Elle veut mettre à rude épreuve ses connaissances et veut connaître ses résultats pour chaque quizz. Elle voudrait pouvoir lire les questions sans à constamment bouger la tête. Elle aime créer des quiz pour ses amies grâce aux connaissances qu’elle a.

### Scénario

* Je choisis le thème « Disney » en appuyant sur la case appropriée.
* Avant de choisir un quizz, j’appuie sur le rouage en haut à droite pour gérer les paramètres.
* J’augmente la taille des marges aux extrémités pour pouvoir voir l’intégralité de la question dans mon champ de vision.
* J’appuie sur confirmer et je choisis le quizz « La petite Sirène ».
* À la fin du quizz, je regarde mon taux de réussite à ce quizz.
* Je choisis de faire une relecture et de voir où étaient mes erreurs en cliquant sur relecture.
* Quand j’ai fini ma relecture, j’appuie sur le bouton quitter.

J’ai envie de rajouter un quiz sur « La reine des neiges ».

Pour cela,

* J’appuie sur le bouton mode création.
* J’édite le thème qui correspond à mon quiz, en l’occurrence, Disney.
* À l’aide du carrousel, je fais défiler les pages jusqu’à trouver une case avec un « + » et j’appuie dessus.
* Je rentre le nom du nouveau quiz et je clique sur créer.
* Je crée une nouvelle question en appuyant sur la case « + ».
* Je remplis les différents champs qui me sont proposés et je confirme la création de question.
* Après avoir créé différentes questions, j’appuie sur la flèche en haut à gauche jusqu’à revenir au menu principal.

## Baudoin

Baudoin a 88 ans, ses petits-enfants lui ont appris à utiliser les nouvelles technologies. Il aime faire des quizz pour se détendre, et apprendre de nouvelles choses sur des sujets qu’il ne connait pas forcément. Il a du mal à lire de près. Il possède aussi une cataracte[[2]](#footnote-2). Il voudrait donc pouvoir changer la police d’écriture ainsi que la couleur de fond en fonction de son état de fatigue.

### Scénario

* Au début de l’application, je configure la taille de police afin de pouvoir lire les questions.
* Je veux faire un quizz sur le « sport ». Je choisis donc ce thème.
* Je remarque que j’ai encore un peu de mal à lire, j’appuie sur l’icône rouage en haut à droite pour changer les paramètres.
* J’augmente encore la taille de police et je change la palette de couleur du site.
* Quand les paramètres me vont, j’appuie sur confirmer.
* Je sélectionne le quizz sur le « handball » et je réponds aux différentes questions.
* À la fin du quizz, je sélectionne quitter.

# Présentation de l’architecture Client Serveur

Ce site a été réalisé dans un cadre d’apprentissage. Aussi, nous avons eu des cours sous formes de travaux pratiques pour prendre en main et comprendre le fonctionnement de la technologie utilisé : Angular. Lors de ceux-ci, nous avons vu qu’un site web était séparé en deux partie, le front et le back.

Le front est la partie de projet, du code, qui va être envoyée au navigateur web pour afficher la page internet. Le back, quant à lui, est la partie « serveur » du site. C’est elle qui contient les données et qui reçoit les requêtes des utilisateurs.

Tout au long de la réalisation du ce projet, nous avons tenu à garder cette séparation, si bien que nous avons deux dépôts git différents. Nous avons donc travaillé en parallèle sur ces deux parties de notre site, et nous allons présenter l’architecture en suivant la même logique.

## Front

## Back

Le back récupère les requêtes provenant du front afin de renvoyer les données demandées par celui-ci. Nous allons vous expliquer plus en détails le fonctionnement du back.

Nous possédons plusieurs routes distinctes, effectuant des instructions spécifiques selon celle utilisée.

Pour récupérer les données, il faut spécifier dans la requête GET, les données de l’élément que nous voulons. Par exemple, si nous voulons récupérer tous les quiz d’un thème, il nous suffira d’utiliser la route GET suivante :

/theme/:id

Avec « :id » qui correspond au nom du thème en question. De la même manière, si nous voulons avoir les données d’un quiz se trouvant dans un thème, on utilisera la route GET suivante :

/:theme/:id

Avec « :theme » et « :id » les noms du thème et du quiz respectif.

Il n’y a pas de route spécifique pour récupérer de question, en effet, le front a besoin uniquement du quiz voulu et récupèrera les questions via le quiz.

Maintenant, dans le cas d’ajout de thème, de question ou bien de quiz, il faudra utiliser une requête POST.

Il suffira d’utiliser le suffixe « /add » à la fin de la route. Par exemple, si nous voulons ajouter un nouveau thème, la route utilisée sera :

/add

Nous pouvons remarquer ici la non-présence d’une variable spécifiant un thème. Ici, nous ne sommes pas dans une requête demandant de renvoyer des données. Par conséquent, nous avons juste besoin de rajouter dans les données du back le thème crée se trouvant dans le corps de la requête.

Pour rajouter un quiz, il nous faut l’information du thème dans lequel nous voulons l’ajouter. La route ressemblera donc à :

/:theme/add

Avec « :theme » qui correspond au nom du thème.

Pour finir, nous passerons aux routes pour supprimer des questions, celle-ci utiliserons la requête DELETE. Ces routes ressemblent beaucoup à celles utilisées dans les requêtes GET. Comme les explications de leur fonctionnement ont été données précédemment, nous allons seulement vous montrer les routes que nous utilisons.

Pour supprimer un thème, nous utilisons la route suivante :

Pour supprimer un quiz :

/:theme/:id

Il y a une exception, c’est pour supprimer une question. Contrairement aux requêtes POST et GET, la requête DELETE est la seule qui a une route pour changer les questions. À l’instar de l’ajout, il nous faut savoir la position de la question dans le quiz. Dans le cas de l’ajout de question, il suffit juste de le rajouter à la fin de la liste des questions.

Mais dans le cas d’une suppression, nous avons besoin de connaître son emplacement afin de ne pas supprimer la mauvaise question.

/:theme/:id/:label

Le « label » correspond à la position de la question dans la liste du quiz de nom « :id » se trouvant dans le thème de nom « :theme»

Pour la sauvegarde des données, nous utilisons un fichier JSON. Pour cela, nous avons des modèles d’objets, un pour le quiz, un pour le thème, et un autre pour les questions.

Ainsi notre fichier JSON se compose d’une liste de thème, et l’objet thème est défini de la façon suivante :

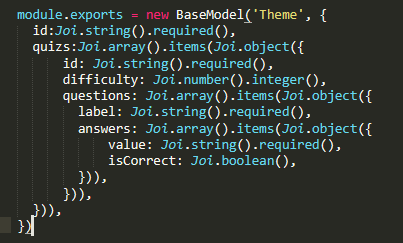


Figure 1- Modèle de l'objet Thème

Comme on peut le voir, l’objet Thème possède un « id », qui correspond au nom du thème et d’une liste d’objet Quiz. Cet objet Quiz est défini avec un « id », le nom du quiz, le niveau de difficulté de celui-ci et une liste de d’objet Question.

Une question est composée d’un « label », qui correspond à l’intitulé de la question, et de réponse. Dans notre projet, la liste des réponses est toujours de taille 4.

Concernant les ressources, nous utilisons le modèle Quiz pour faire toutes les opérations nécessaires sur la gestion des données de l’application.

Nous récupérons dans le corps de la requête un objet quiz, et c’est avec le modèle du quiz que l’on peut voir s’il y a eu un changement sur le quiz. Que se soit le niveau de difficulté, un ajout de question, ou autre.

C’est pour cela que dans les routes de notre API, nous spécifions toujours le thème dans lequel le quiz se trouve. Comme l’objet quiz ne nous donne pas d’information sur le thème, il fallait récupérer l’information d’une autre manière.

# Présentation de l’évaluation croisée et analyse des résultats obtenus

L’évaluation croisée est un outil qui nous a permet d’avoir des retours sur notre site et des remarques extérieurs qui peuvent améliorer notre site.

Les utilisateurs ont remarqués que la taille des caractères dans les pop up qui permettent de confirmer la suppression d’un élément sur le site ne sont pas conformes à la taille que l’utilisateur a choisi dans la page de configuration mais leurs taille reste fix à une taille standard, alors on peut adapter la taille des caractères dans ce cas aussi car cela peut poser un problème pour une personne malvoyante qui n’arrivera pas à lire ce qui est écrit sur les pop up. Donc il faut que les pop up aient les mêmes configurations choisissent par l’utilisateur.

On a eu aussi des retours à propos de la différenciation entre les éléments cliquables et ceux qui le sont pas, et cela peut créer des confusions pour les personnes âgées alors il sera mieux de bien les guider et préciser quels sont les endroits ou ils doivent cliquer et les éléments non cliquables afin d’éviter la confusion et la perte de temps.

La partie relecture peut être développées vu qu’à l’instant ça nous permet juste de relire toutes les questions avec les réponses correctes, mais on peut améliorer ça en ajoutons les réponses fausses de l’utilisateur avec les bonnes réponses pour ces questions et aussi ajouter un système de score qui permet de savoir les réponses fausses qu’il a eu et son évolution.

# Conclusion et perspective

# Annexes

# Répartition des tâches dans l’équipe

# Vidéo de démonstration finale

1. Le glaucome est une maladie dégénérative du nerf optique qui entraîne une perte progressive de la vision commençant tout d'abord en périphérie et progressant graduellement vers le centre. (<https://fr.wikipedia.org/wiki/Glaucome>) [↑](#footnote-ref-1)
2. La cataracte est l'opacification partielle ou totale du cristallin, lentille convergente située à l'intérieur de l'œil. Cette opacification est responsable d'une baisse progressive de la vue, au début accompagnée de gêne à la lumière (photophobie). (<https://fr.wikipedia.org/wiki/Cataracte_(maladie)>) [↑](#footnote-ref-2)