

Cahier des charges :

Sujet B



Le cahier des charges (CDC) est un document contractuel à respecter lors d'un projet et représente une étape clé.

Le cahier des charges va permettre au maître d'ouvrage de faire savoir au maître d'œuvre ce qu'il attend de lui lors de la réalisation du projet.

Il doit décrire précisément les besoins auxquels le prestataire ou le soumissionnaire doit répondre, et organise la relation entre les différents acteurs tout au long du projet.

Préface

Le document suivant s'adresse aux entreprises et entités candidates, et potentiellement soumissionnaires dans la réalisation du produit demandé par la République de Sealand (SRYEM).

Ce cahier des charges sera introduit par une brève présentation du contexte, de la demande et des objectifs de la réalisation logicielle. Les exigences relatives au logiciel seront énoncées puis détaillées, dans le cadre de spécification sémantique et de format de données. Une conclusion permettra la mise en perspective des objectifs.

Ce document est susceptible d'être révisé à la suite de remarques et questions. Les mises à jour et révisions seront suivies dans la table qui suit.

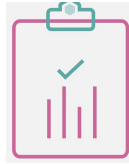
Version	Date	Mise à jour
1	01/11/21	première version officielle

Tables des matières



<i>Préface</i>	2
<i>Tables des matières</i>	3
<i>Introduction</i>	3
Contexte et définition du problème	4
Objectif du projet logiciel	
<i>Glossaire</i>	4
<i>Spécification générale des exigences</i>	4
Introduction :	4
Liste des exigences fonctionnelles	4
Liste des exigences non-fonctionnelles	4
<i>Spécification détaillée des exigences</i>	5
Détail de chacune de exigences :	5
Format de données :	5
Sémantiques des données :	5
<i>Conclusion</i>	6

Introduction



Contexte et définition du problème



Le ministère de l'éducation nationale de la République de Sealand (SRYEM) souhaite accompagner son principal établissement d'éducation secondaire dans une transition numérique de la pédagogie en fournissant un système d'évaluation en ligne. La transition vers l'informatique pour les enseignants peut paraître complexe si les solutions informatiques proposées ne sont pas accessibles ou incompréhensibles. Les moyens d'évaluation informatique doivent nécessairement respecter les mêmes normes que les moyens d'évaluation classiques, c'est-à-dire la fiabilité, la clarté et la pertinence du test.

Objectif du projet logiciel



Le projet consiste ainsi en la réalisation d'un logiciel utilisable en invite de commande qui permettra aux gestionnaires et enseignants de composer des examens et des tests au format **GIFT*** à réaliser en ligne. Ces tests seront réalisés à partir de la large banque de questions certifiée dont disposent les services du ministère. Le projet répondra aux besoins des utilisateurs potentiellement novices en informatique et également aux contraintes formulées dans la suite de ce cahier des charges. Dans un premier temps seront présentées les spécifications générales des exigences fonctionnelles et non fonctionnelles, puis dans un second temps les contraintes de format et sémantique des données.

Glossaire



GIFT: Le format GIFT est le plus complet des formats d'importation de questions disponible dans Moodle. Il a été conçu pour permettre aux enseignants d'écrire simplement des questions dans un fichier texte. Les différents types de questions peuvent être mélangés dans un même fichier texte, et le format permet également l'insertion de commentaires, titres de questions, feedbacks et pondération des points en %.

VCard: Norme de format de fichier pour cartes de visite électroniques. La version 4.0 de 2011 de vCard est définie par les RFC 6350 et 6868. Un fichier vCard porte une extension vcf.

Spécification générale des exigences

Introduction

Ci-dessous sont listées les **exigences fonctionnelles**, qui sont des exigences définissant les fonctions du système à développer; ce que le système doit faire. Et les **exigences non-fonctionnelles** qui caractérisent une propriété (qualité) désirée du système telle que sa performance, sa robustesse, sa convivialité, sa maintenabilité, etc.



Liste des exigences fonctionnelles

SPEC_1 : S'authentifier sur la durée de vie d'une session

SPEC_2 : rechercher et visualiser une question pour la choisir

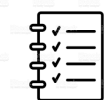
SPEC_3 : regrouper un ensemble de questions pour créer un fichier GIFT d'examen

SPEC_4 : générer un fichier d'identification et de contact de l'enseignant au format [VCard*](#)

SPEC_5 : simuler la passation du test par un étudiant avec bilan des réponses à l'issue

SPEC_6 : vérification de la qualité des données d'examen préparées par un enseignant:

- a. doublons parmi les questions
- b. pas moins de 15 questions et pas plus de 20



Liste des exigences non-fonctionnelles

SPEC_NF_1 : dresser le profil d'un examen GIFT ou d'une banque de question, visualiser un histogramme des différents types de questions incluses (choix multiples, vrai-faux, correspondance, mot manquant, numérique, question ouverte)

SPEC_NF_2 : comparaison d'un profil d'examen (type de question) avec le profil moyen d'un ou plusieurs fichiers de la banque nationale de questions

Spécification détaillée des exigences

Détail de chacune de exigences fonctionnelles

Identifiant	SPEC_1
Titre	S'authentifier
Acteur	Utilisateur qui a lancé l'invite de commande
Contexte d'utilisation	Un professeur souhaite accéder à ses tests, en créer un, etc.
Motivation	L'authentification permet à l'utilisateur à avoir accès à ses préférences, son historique et ses tests
Précondition(s)	Avoir un accès à l'utilitaire.
Postcondition(s)	Le professeur est identifié auprès du système, il a accès à ses données
Déclencheur	Lancement de l'utilitaire sur le terminal, demande d'entrée du mot de passe.
Scénario principal	L'utilisateur démarre l'utilitaire en invite de commande (par exemple : <nomUtilitaire> start)

Identifiant	SPEC_2
Titre	Rechercher et visualiser une question
Acteur	Utilisateur : Professeur authentifié
Contexte d'utilisation	Un professeur souhaite constituer un examen en ligne, il va chercher l'ensemble des questions concernant le sujet de l'examen, puis en visualise certaines intéressantes et fait une sélection.
Motivation	Le professeur peut chercher le type de question adapté au test qu'il souhaite constituer, il peut filtrer.
Précondition(s)	Avoir les droits d'accès à la banque de questions, avoir créé un test.
Postcondition(s)	Liste de questions correspondantes à la recherche
Déclencheur	Commande de recherche avec en paramètre le critère
Scénario principal	L'utilisateur crée un nouveau test Il recherche une question Il ajoute la question souhaitée au test
Scénario alternatif	L'utilisateur recherche une question Il l'ajoute à un test déjà existant

Identifiant	SPEC_3
Titre	Regrouper un ensemble de questions pour créer un fichier GIFT* d'examen
Acteur	Utilisateur : Professeur
Contexte d'utilisation	Un utilisateur souhaite créer un test et y intégrer un ensemble de questions choisies au préalable.
Motivation	L'utilisateur peut ajouter un ensemble de questions à un test, il n'a pas besoin de rechercher distinctement chaque question
Précondition(s)	Avoir les identifiants des questions à ajouter au test
Postcondition(s)	Un test complet avec un identifiant unique
Déclencheur	Commande de création d'un test avec en paramètre la liste des identifiants des questions à ajouter
Scénario principal	Un professeur crée un test avec la commande avec les questions du test, il peut ensuite ajouter des questions au test existant
Scénario alternatif	exemple de commande : <code>createTest <nom> -q {<IDquestion1>, ..., <IDquestionN>}</code>

Identifiant	SPEC_4
Titre	Générer un fichier d'identification et de contact de l'enseignant au format VCard*
Acteur	Utilisateur : Professeur authentifié
Contexte d'utilisation	Un fichier VCard sera généré avec les informations d'identification du professeur
Motivation	Le professeur peut associer son fichier VCard à ses tests sur le serveur d'examen. Ainsi les Autres utilisateurs connaissent l'identité du créateur des tests
Précondition(s)	être authentifié
Postcondition(s)	Un fichier VCard
Déclencheur	Commande de création d'un fichier d'identification (par exemple <code>createVCard</code>)
Scénario principal	L'utilisateur doit répondre à des questions une à une dans l'invite de commande, son fichier est ensuite généré est affiché dans l'invite de commande, l'utilisateur doit confirmer les informations ([Yes/no])

Identifiant	SPEC_5
Titre	Simuler la passation du test par un étudiant avec bilan des réponses à l'issue
Acteur	Utilisateur : Professeur
Contexte d'utilisation	Il est possible de simuler la passation d'un test en invite de commande.
Motivation	L'éditeur d'un test pourra voir ce à quoi ressemble le test d'un point de vue de l'étudiant, et s'assurer de sa cohérence.
Précondition(s)	Être authentifié et être l'éditeur du test
Postcondition(s)	/
Déclencheur	Commande de lancement d'une simulation (par exemple : <i>simulate <IDtest></i>)
Scénario principal	Chaque question est posée avec l'invite de commande à la suite pour que l'utilisateur entre une réponse, une fois le test terminé, un récapitulatif est affiché avec une note.
Scénario alternatif	Si le test en offre la possibilité, au moment du récapitulatif l'utilisateur peut confirmer ses réponses et avoir sa note ([Yes/no]) ou entrer le numéro d'une question pour y revenir et soumettre une nouvelle réponse.

Identifiant	SPEC_6
Titre	Vérification de la qualité des données d'examen préparées par un enseignant
Acteur	Professeur éditeur du test
Contexte d'utilisation	Un algorithme du système parcourt le test et vérifie qu'il n'y ai pas de doublons parmi les questions , et qu'il n'y pas moins de 15 questions et pas plus de 20
Motivation	Les examens seront donc supervisés, pour parer les potentielles erreurs(de niveau, d'égalité, etc.)
Précondition(s)	Avoir un test complet
Postcondition(s)	Test validé ou non avec affichage des potentielles erreurs.
Déclencheur	Commande <i>verify <nomTest></i> par exemple
Scénario principal	L'éditeur d'un test lance une validation sur son test, un compte rendu s'affiche sur le terminal. Il peut relancer une validation après avoir corrigé ses erreurs s'il y en a.

Détail de chacune de exigences non fonctionnelles

Identifiant	SPEC_NF_1
Titre	Dresser le profil d'un examen GIFT ou d'une banque de question, visualiser un histogramme des différents types de questions incluses
Acteur	Éditeur du test
Contexte d'utilisation	Un professeur veut analyser la structure de son test, il a la possibilité de générer un histogramme avec les types de questions (choix multiples, vrai-faux, correspondance, mot manquant, numérique, question ouverte)
Motivation	Un professeur peut avoir une vue d'ensemble sur son test, s'assurer de son équilibre et de sa cohérence.
Précondition(s)	Avoir créé et terminé un test
Postcondition(s)	Affichage d'un histogramme dans le terminal
Déclencheur	Commande <i>profil</i> <nomTest> par exemple
Scénario principal	L'utilisateur crée et édite son test Il génère l'histogramme de son test à l'aide de la commande Il obtient un histogramme affiché dans son terminal

Identifiant	SPEC_NF_2
Titre	Comparaison d'un profil d'examen (type de question) avec le profil moyen d'un ou plusieurs fichiers de la banque nationale de questions
Acteur	Utilisateur / éditeur d'un test
Contexte d'utilisation	Après la réalisation d'un test, l'utilisateur peut vérifier que son test colle bien au profil souhaité
Motivation	Amélioration des données de la banque de questions
Précondition(s)	Connaître le profil moyen des fichiers que l'utilisateur souhaite comparer
Postcondition(s)	/
Déclencheur	commande <i>compare</i> <nomTest> -p <nomProfil> par exemple
Scénario principal	L'utilisateur crée et édite son test Il compare son test au profil de test souhaité (par exemple QCM) Il obtient une comparaison avec pourcentage, et si la comparaison n'est pas pertinente le système peut proposer un autre profil avec lequel comparer
Remarques	Les profils de tests sont générés automatiquement au sein de la banque de question, à partir des tests réalisés par tous les utilisateurs

Format de données

Ci dessous le format des données contenues dans les fichiers GIFT, spécifié sous forme de Backus-Naur augmentée (ABNF).

```
questionGift    = [commentaire] titre [CRLF] question [WSP]
reponses [question]
titre           = "::-" idTest WSP [page] nomTest idQuestion "::-"
idTest          = [*UPALPHA WSP] "U"1*DIGIT
page           = "p" 1*DIGIT ["-" 1*DIGIT] WSP
nomTest         = *(ALPHA / WSP)
idQuestion      = WSP 1*VCHAR
question        = texte
reponses        = "{" [CRLF] (*reponse / *vraiOuFaux) "}"
reponse         = ("~" / "=") [pourcentPoints] 1*VCHAR
[*retroaction]
vraiOuFaux      = "FALSE" / "TRUE" / "F" / "T" [*retroaction]
retroaction     = "#" texte
commentaire     = "//" texte CRLF
texte           = 1*(WSP / VCHAR)
pourcentPoints  = "%" 1*2DIGIT "%"
```

Sémantiques des données

Principal type d'objet :

Titre : QUESTION

Sorte : Question

Description : Spécifie un type de question mais surtout un type de réponse adaptée (vrai faux, choix multiple, à réponse courte, à mot manquant, question ouverte)

Référence(s) : String, Boolean

Pré-conditions :

question = {question \in String}

reponses = {reponses \in Ens<String> }

Signature :

créerQuestion : question x reponses \rightarrow Question

poserQuestion : Question \rightarrow question

donnerRéponses : Question \rightarrow reponses

estIdentique : Question x Question \rightarrow Boolean

typeDeQuestion : questionsChoixMultiple \rightarrow QCM (boolean)

: vraiOuFaux \rightarrow V/F (booléen)

: àMotManquant \rightarrow mw (missing word) (char)

: àRéponseNumérique \rightarrow valNum (float)/(int)

: questionOuvverte \rightarrow texte (char)

: reponseCourte \rightarrow phrase (char)

Axiomes :

poserQuestion(créerQuestion(question, reponses)) = question

donnerRéponses(créerQuestion(question, reponses)) = reponses

estIdentique(créerQuestion(q1, r1), créerQuestion(q2, r2)) =
q1==q2 && r1==r2

Spécification algébrique d'une collection permettant de contenir la sorte Question :

Titre : ENSEMBLE (QUESTION)

Sorte : Ens (Question)

Description : Spécifie un ensemble de différentes questions de type Question. Permet l'ajout et les opérations ensemblistes d'union et intersection. L'appartenance d'un item à un ensemble peut être vérifiée et le nombre de ses éléments compté. Un ensemble ne contient qu'un exemplaire d'un même élément.

Référence(s) : Ens<I>, Question, String

Signature :

Créer : Ens<Question>

Ajouter : Ens<Question> x Question -> Ens<Question>

Union : Ens<Question> x Ens<Question> -> Ens<Question>

Intersection : Ens<Question> x Ens<Question> -> Ens<Question>

Axiomes :

Conclusion



L'utilitaire réalisé dans ce projet devra fournir des fonctionnalités précises et distinctes, avec la plus large couverture d'erreurs possible. Dans l'ensemble, les utilisateurs pourront créer des tests, y intégrer une ou plusieurs questions sélectionnées dans la banque nationale de questions. Ils pourront également simuler la passation du test, en dresser son profil, visualiser son histogramme ou encore le comparer avec des profils de tests types.

La soumission d'une proposition de réalisation nécessitera un respect intégral des exigences fonctionnelles.

L'implémentation d'un tel système peut être multiple. Il peut être pertinent d'en modéliser l'architecture au préalable, afin de confirmer la cohérence des choix d'implémentation.