

Cahier des charges

Contexte SRYEM

Réalisé par KAMGUE MAGNE Ange, CAPAR Sila, RANVOISY Nicolas

Année : 2024-2025



Sommaire

1. Préface.....	3
2. Introduction.....	4
3. Spécification générale des exigences.....	5
3.1. Exigences fonctionnelles.....	5
3.2. Exigences non fonctionnelles.....	6
3.3. Priorisation des exigences.....	6
4. Spécification détaillée.....	7
4.1. Détail de chacune des exigences en format structuré.....	7
4.2. Formats de données.....	16
4.2.1. Format de données pour les fichiers .GIFT :.....	16
4.2.2. Format de donnée pour VCard :.....	16
4.3. Sémantique des données.....	18
5. Conclusion.....	21

1. Préface

Ce document est un cahier des charges destiné à l'équipe de développement logiciel et aux responsables techniques du ministère de l'Éducation de la République de Sealand (SRYEM). Il vise à définir les spécifications fonctionnelles nécessaires à la conception d'un utilitaire en ligne de commande permettant de faciliter la gestion et la création de tests d'évaluation au format GIFT.

Le document vise à guider les intervenants lors des phases de conception, de recherche de questions, et de génération de fichiers GIFT et VCard. Les exigences sont spécifiées sous forme de tableau afin de présenter de façon claire et précise les fonctionnalités du logiciel.

Les révisions de ce cahier de charge seront basées sur des retours réguliers entre les parties impliquées dans la conception de l'outil et un historique sera mis en place. Ce dernier sera mis à jour lors de chaque révision afin d'assurer la traçabilité des modifications qui ont été apportées au projet.

2.Introduction

Dans le cadre de l'UE GL02, nous sommes amenés à mettre en pratique nos compétences en génie logiciel afin de concevoir une solution technique répondant à un problème spécifique. Le projet consiste ici en la réalisation d'un utilitaire en ligne de commande, permettant la gestion et la préparation de tests ou d'examens en ligne.

Ce logiciel s'appuie sur un jeu de données et permet de générer des examens au format GIFT. Il offre à l'utilisateur la possibilité de contrôler la qualité des questions, de naviguer entre elles, et de s'assurer que les tests respectent les exigences de conformité. Le logiciel sera développé en langage Javascript et utilisera des commandes Bash pour son interaction en ligne de commande.

Ce cahier des charges présente les différentes fonctionnalités du logiciel ainsi que les exigences associées à chacune d'elles. Une attention particulière est portée à la gestion du format GIFT et à la sémantique des données utilisées pour garantir une solution conforme aux attentes du ministère de l'Éducation de la République de Sealand.

3. Spécification générale des exigences

Pour ce projet, nous avons défini un ensemble de neuf exigences fonctionnelles, détaillant les capacités que l'application doit offrir aux utilisateurs, ainsi que quatre exigences non fonctionnelles, précisant les contraintes techniques et d'utilisation du programme.

3.1. Exigences fonctionnelles

SPEC_1 Composition des tests au format GIFT : L'application doit permettre de composer des tests en format GIFT, en utilisant des questions qui auront été sélectionnées par l'utilisateur.

SPEC_2 Visualisation de questions : L'utilisateur doit pouvoir visualiser chaque question pour en prendre connaissance avant de l'ajouter à un test ou de la modifier si nécessaire.

SPEC_3 Recherche et sélection de questions : L'outil doit inclure des fonctions de recherche permettant à l'utilisateur de retrouver facilement des questions spécifiques et de les sélectionner pour les ajouter aux tests.

SPEC_4 Vérification de la conformité des données : L'application doit vérifier la validité des données du test, en s'assurant qu'il contient le bon nombre de questions et que celles-ci ne sont pas redondantes.

SPEC_5 Visualisation du profil d'un examen GIFT : L'outil doit fournir une vue d'ensemble des caractéristiques de l'examen, incluant les types de questions utilisées.

SPEC_6 Histogramme des types de questions : Le programme doit générer un histogramme pour représenter les types de questions inclus dans le test, en calculant les moyennes de chaque type.

SPEC_7 Comparaison de profils d'examen : L'application doit permettre de comparer le profil d'un examen, en termes de types de questions, avec celui d'un ou de plusieurs autres fichiers d'examen.

SPEC_8 Simulation de passage de test et bilan : L'outil doit inclure une fonctionnalité permettant de simuler la passation du test par un étudiant fictif et de fournir un bilan de ses résultats.

SPEC_9 Génération de fichier VCard pour l'enseignant : Il doit être possible de générer un fichier au format VCard contenant les informations d'identification et de contact de l'enseignant, en vue de faciliter la gestion des contacts.

3.2. Exigences non fonctionnelles

1. SPEC_NF_1 Utilisation de JavaScript : Le programme doit être intégralement développé en JavaScript pour garantir sa compatibilité et son interopérabilité avec les outils web.
2. SPEC_NF_2 Accès restreint aux enseignants : Seuls les enseignants doivent pouvoir modifier l'utilitaire, garantissant ainsi la sécurité et l'intégrité des données.
3. SPEC_NF_3 Nombre de questions par examen : Chaque examen doit inclure entre 15 et 20 questions pour assurer un volume suffisant de contenu et maintenir une difficulté raisonnable.
4. SPEC_NF_4 Conformité des fichiers VCard : Les fichiers VCard générés doivent être conformes aux normes RFC 6350 et 6868, garantissant ainsi une compatibilité maximale avec les outils de gestion de contacts.

3.3. Priorisation des exigences

Certaines de ces fonctionnalités sont jugées essentielles au bon fonctionnement de l'application. Les exigences fonctionnelles relatives à la composition de tests (SPEC_1), à la visualisation des questions (SPEC_2), à la recherche et sélection de questions (SPEC_3), à la vérification de conformité des données (SPEC_4), ainsi qu'à la visualisation du profil d'examen (SPEC_5) sont considérées comme prioritaires et devront être implémentées en premier.

4. Spécification détaillée

4.1. Détail de chacune des exigences en format structuré

Identifiant	SPEC_1
Titre	Composer des tests au format GIFT à l'aide des questions sélectionnée
Acteur	Enseignant
Traitements	L'ensemble des données des questions sont extraites pour être formatées en test GIFT
Objectifs	Avoir une commande permettant de déclencher la composition du test GIFT
Précondition(s)	L'ensemble des questions sont sélectionnées
Postconditions(s)	Un test GIFT a été créé
Entrées	Une commande
Sorties	Un test GIFT
Gestion des erreurs	Si moins de 15 ou plus de 20 questions ont été sélectionnées, indiquer un message d'erreur, de même si une question apparaît 2 fois

Identifiant	SPEC_2
Titre	Visualiser une question
Acteur	Enseignant
Traitements	Les données de la question sont exportées dans un format visible et structuré pour l'utilisateur
Objectif	L'utilisateur désire lire une question
Précondition(s)	La question est déjà créée dans la banque de questions.
Postconditions(s)	La question est affichée
Entrées	Sélectionner une question
Sorties	Un affichage à l'écran
Gestion des erreurs	/

Identifiant	SPEC_3
Titre	Rechercher et sélectionner une question
Acteur	Enseignant
Traitements	Tri des questions relatives à la recherche Puis sélection de la question
Objectifs	Pouvoir trouver les questions requises à la création d'un examen
Précondition(s)	L'utilisateur est connecté avec un compte enseignant
Postconditions(s)	La question est sélectionnée
Entrées	Une recherche (mot clef, tag)
Sorties	Une question
Gestion des erreurs	Quand aucune question ne correspond, mettre un message d'erreur

Identifiant	SPEC_4
Titre	Vérifier la conformité des données (Nombre de questions et questions uniques)
Acteur	Enseignant
Traitements	Le logiciel vérifie s'il y a des doublons et s'assure que le nombre de questions est compris entre 15 et 20
Objectif	L'enseignant souhaite vérifier la conformité des données de son examen
Précondition(s)	Un test GIFT a été ou est en train d'être créé
Postconditions(s)	Le nombre de question est compris entre 15 et 20 Les questions qui reviennent plusieurs fois ont été supprimées
Entrées	Un test GIFT est en train d'être créé, ou un test est soumis à vérification
Sorties	Les questions du test conforme sont affichées
Gestion des erreurs	Si le test n'est pas conforme, afficher un message d'erreur pour l'enseignant

Identifiant	SPEC_5
Titre	Visualiser le profil d'un examen GIFT
Acteur	Enseignant
Traitements	Comptabiliser type de question dans l'examen donnée en entrée
Objectifs	Pouvoir visualiser les différents types de questions présents dans un examen GIFT
Précondition(s)	Des examens sont présents dans la banque de données
Postcondition(s)	L'examen est visible à l'écran
Entrées	Un examen
Sorties	le profil de l'examen
Gestion des erreurs	/

Identifiant	SPEC_6
Titre	Réaliser un histogramme des types de question avec la moyenne des questions
Acteur	Administrateur
Traitements	Comptabilisation de tous les types des profils des examens présent dans la banque de données
Objectifs	Pouvoir visualiser les différents types de questions présents dans un examen GIFT
Précondition(s)	Des examens sont créés dans la banque de données
Postcondition(s)	Un histogramme est créé
Entrées	Tous les examens de la banque de données
Sorties	Histogramme des types de question avec la moyenne pour la chaque type de question
Gestion des erreurs	/

Identifiant	SPEC_7
Titre	Comparer le profil en terme de type de question d'un examen à celui d'un ou de plusieurs fichiers
Acteur	Enseignant
Traitement	Les types de questions qui sont utilisés pour un examen sont comparés avec les questions présentes dans les fichiers de la banque nationale des questions
Objectif	L'enseignant souhaite déterminer les types de questions de son examen qui sont en commun avec les types de questions dans les fichiers
Précondition(s)	Des questions sont disponibles dans la banque de questions et dans l'examen
Postconditions(s)	Il y a des types en commun avec certains fichiers
Entrées	L'enseignant appuie sur le bouton "comparer"
Sorties	Les types communs et les fichiers concernés sont affichés
Gestion des erreurs	/

Identifiant	SPEC_8
Titre	Simuler le passage d'un test par un étudiant puis bilan de ce test
Acteur	Etudiant
Traitements	Le logiciel lance le test en demandant les réponses à chaque question du test GIFT
Objectifs	Pouvoir s'assurer que le test GIFT est fonctionnel et que le test peut être effectué dans de bonnes conditions
Précondition(s)	Un questionnaire a été créé
Postcondition(s)	L'étudiant factice a pu passer le test et peut consulter le bilan du test
Entrées	Les réponses de l'étudiant
Sorties	Un bilan a été rédigé
Gestion des erreurs	/

Identifiant	SPEC_9
Titre	Générer un fichier d'identification et de contact de l'enseignant au format VCard
Acteur	Administrateur
Traitements	Les informations de l'enseignant sont formatées dans un fichier VCard
Objectifs	Pouvoir identifier l'enseignant ayant créé l'examen et pouvoir le contacter
Précondition(s)	Un enseignant est enregistré
Postcondition(s)	Un fichier VCard est créé et permet de contacter l'enseignant
Entrées	Les informations de l'enseignant tels que son nom, prénom, email, etc...
Sorties	fichier VCard
Gestion des erreurs	Non respect des normes d'écriture du mail, nom ou prénom vide, etc.. entraîne des avertissement ciblés sur ces entrées

4.2. Formats de données

4.2.1. Format de données pour les fichiers .GIFT :

fichier_gift = *categorie *question-gift

categorie = "\$CATEGORY:" WSP chemin_categorie CRLF

chemin_categorie = texte

question-gift = "::" identifiant-question "::" texte "{" liste_reponses "}" texte CRLF

identifiant_question = *(ALPHA / DIGIT / WSP / "." / "-")

liste-reponses = *(reponse_correcte / reponse_fausse)

reponse_correcte = "=" texte

reponse_fausse = "~" texte

texte = *(VCHAR/WSP)

4.2.2. Format de donnée pour VCard :

VCard = "BEGIN:VCARD" CRLF

*ligne_contenu

"END:VCARD" CRLF

ligne_contenu = nom ":" valeur CRLF

FN = "FN:" nom_complet CRLF

N = "N:" nom_de_famille ";" prenom CRLF

TEL = "TEL:" tel CRLF

EMAIL = "EMAIL:" adresse_email CRLF

ADR = "ADR:" adresse_postale CRLF

ORG = "ORG:" organisation CRLF

nom_complet = texte

nom_de_famille = texte

prenom = texte

tel = "+" 1*DIGIT

adresse_email = texte "@" texte
adresse_postale = texte ";" texte ":" *DIGIT ";" texte
organisation = texte

4.3. Sémantique des données

On a 2 spécifications algébriques différentes: L'une pour les questions et l'autre pour le test (qui est un ensemble de questions)

Titre de la spécification : Spécification algébrique d'une question

Sorte : question(i)

Référence : string, boolean

Description:

"i" représente une chaîne de caractères qui représente le libelé de la question. Plusieurs opérations peuvent être appliquées. CréerQuestion qui nous permet de créer une question, TypeQuestion qui renvoie le type de la question, EgaliteQuestion qui va vérifier si 2 questions sont identiques, VisualiserQuestion qui produit un affichage à l'écran de la question et rechercherQuestion qui utilise un mot clé pour trouver la question dans le test.

Signature:

CréerQuestion : i -> question(i)

TypeQuestion : question -> string

EgaliteQuestion : question(i) * question(i) -> boolean

VisualiserQuestion: question(i) * int -> string

RechercherQuestion: question(i) * string -> question(i)

Axiomes:

EgaliteQuestion (CréerQuestion(i) , CréerQuestion(j)) = vrai Ssi i=j

VisualiserQuestion (CréerQuestion(i), numeroQuestion) = "La question numéro " + numeroQuestion + "est :" + question(i)"

RechercherQuestion (CréerQuestion(i), motCle) = "La question que vous recherchez est :" + question(i)

TypeQuestion(CréerQuestion(i)) = "choix_multiple" si i contient des options de choix multiple
 | "vrai_faux" si i est une question à vrai ou faux | "correspondance" si i est une question de
 type correspondance | "mot_manquant" si i est une question de type mot manquant |
 "numerique" si i est une question numérique | "question_ouverte" si i est une question
 ouverte

Titre : Ensemble de questions

Sorte : $\text{Ens}\langle \text{question}(i) \rangle$

Références : int, boolean, question, String

Signature :

CréerEnsQuestion : $\text{Ens}\langle \text{question} \rangle \rightarrow \text{Ens}\langle \text{question}(i) \rangle$

AjouterQuestion : $\text{Ens}\langle \text{question}(i) \rangle \times \text{question}(i) \rightarrow \text{Ens}\langle \text{question}(i) \rangle$

CardQuestion : $\text{Ens}\langle \text{question}(i) \rangle \rightarrow \text{int}$

Appartient : $\text{Ens}\langle \text{question}(i) \rangle \times \text{question}(i) \rightarrow \text{boolean}$

Union : $\text{Ens}\langle \text{question}(i) \rangle \times \text{Ens}\langle \text{question}(i) \rangle \rightarrow \text{Ens}\langle \text{question}(i) \rangle$

Intersection : $\text{Ens}\langle \text{question}(i) \rangle \times \text{Ens}\langle \text{question}(i) \rangle \rightarrow \text{Ens}\langle \text{question}(i) \rangle$

CompteTypeQuestion : $\text{Ens}\langle \text{question}(i) \rangle \times \text{String} \rightarrow \text{int}$

Axiomes :

$\text{Appartient}(\text{CréerEnsQuestion}(), \text{it}) = \text{Faux}$

$\text{Appartient}(\text{AjouterQuestion}(\text{ens}, \text{it1}), \text{it2}) = \text{Vrai}$ ssi $\text{it1} = \text{it2} \vee \text{Appartient}(\text{ens}, \text{it2})$

$\text{Appartient}(\text{Union}(\text{ens1}, \text{ens2}), \text{it}) = \text{Appartient}(\text{ens1}, \text{it}) \vee \text{Appartient}(\text{ens2}, \text{it})$

$\text{Appartient}(\text{Intersection}(\text{ens1}, \text{ens2}), \text{it}) = \text{Appartient}(\text{ens1}, \text{it}) \wedge \text{Appartient}(\text{ens2}, \text{it})$

$\text{CardQuestion}(\text{CréerEnsQuestion}()) = 0$

$\text{CardQuestion}(\text{AjouterQuestion}(\text{ens}, \text{it})) = \text{CardQuestion}(\text{ens}) + 1$ ssi $\text{Appartient}(\text{ens}, \text{it}) = \text{Faux}$

$\text{CardQuestion}(\text{Union}(\text{ens1}, \text{ens2})) = \text{CardQuestion}(\text{ens1}) + \text{CardQuestion}(\text{ens2}) - \text{CardQuestion}(\text{Intersection}(\text{ens1}, \text{ens2}))$

$\text{Union}(\text{AjouterQuestion}(\text{ens}, \text{it}), \text{CréerEnsQuestion}()) = \text{AjouterQuestion}(\text{ens}, \text{it})$

$\text{Intersection}(\text{AjouterQuestion}(\text{ens}, \text{it}), \text{CréerEnsQuestion}()) = \text{CréerEnsQuestion}()$

$\text{CompteTypeQuestion}(\text{CréerEnsQuestion}(), \text{type}) = 0$

$\text{CompteTypeQuestion}(\text{AjouterQuestion}(\text{ens}, \text{question}(i)), \text{type}) = \text{CompteTypeQuestion}(\text{ens}, \text{type}) + 1$ ssi $\text{TypeQuestion}(\text{question}(i)) = \text{type}$, sinon $\text{CompteTypeQuestion}(\text{ens}, \text{type})$

$\text{CompteTypeQuestion}(\text{Union}(\text{ens1}, \text{ens2}), \text{type}) = \text{CompteTypeQuestion}(\text{ens1}, \text{type}) + \text{CompteTypeQuestion}(\text{ens2}, \text{type}) - \text{CompteTypeQuestion}(\text{Intersection}(\text{ens1}, \text{ens2}), \text{type})$

$\text{CompteTypeQuestion}(\text{Intersection}(\text{ens1}, \text{ens2}), \text{type}) = \text{CompteTypeQuestion}(\text{ens1}, \text{type})$ si $\text{Appartient}(\text{ens1}, \text{question}(i)) \wedge \text{Appartient}(\text{ens2}, \text{question}(i))$, sinon 0

5. Conclusion

Ce cahier des charges décrit les bases du projet que nous avons conçu pour faciliter la création et la gestion d'examens au format GIFT pour les enseignants de la République de Sealand. L'objectif principal est de créer un outil en ligne de commande simple et efficace, qui permet de manipuler des questions et de structurer des tests d'évaluation. Pour cela, nous avons défini des exigences claires, tant fonctionnelles que non fonctionnelles, pour répondre aux besoins des utilisateurs finaux tout en respectant les contraintes techniques.

Au niveau des exigences fonctionnelles, nous avons priorisé des fonctionnalités essentielles, comme la possibilité de composer des tests à partir de questions sélectionnées, d'afficher et de vérifier les questions pour s'assurer qu'elles respectent le format et les critères définis, et de donner un aperçu de la structure des examens. Le logiciel doit aussi permettre d'afficher des profils d'examens pour une analyse plus complète, ainsi que de générer des fichiers VCard pour faciliter la gestion des contacts des enseignants.

Les exigences non fonctionnelles, quant à elles, assurent la sécurité et la compatibilité de l'outil. En limitant son usage aux enseignants, en adoptant JavaScript pour la compatibilité avec les outils modernes, et en respectant les standards pour les fichiers VCard, le projet répond aux attentes de sécurité et de fiabilité nécessaires pour un usage dans un contexte éducatif.

Pour la mise en œuvre, quelques recommandations pratiques :

- Adopter une structure modulaire pour le code, en séparant bien les parties de manipulation de données, d'interface utilisateur, et de génération de fichiers, ce qui rendra le projet plus clair et facile à maintenir.
- Développer des tests unitaires pour valider chaque exigence, en particulier les fonctions liées aux ensembles de questions, afin de s'assurer que tout fonctionne correctement dès le début.
- Automatiser la vérification des formats de données pour GIFT et VCard dès l'entrée des données, ce qui permet de limiter les erreurs et de garantir la conformité.

En conclusion, ce projet vise à offrir une solution pratique et accessible pour les enseignants. En développant un outil utile et simple d'utilisation, nous espérons faciliter la gestion des examens et l'organisation pédagogique, tout en laissant la porte ouverte à d'éventuelles évolutions futures en fonction des retours des utilisateurs finaux.