

Javaholics

Thomas CHARPENTIER

Ilias CHOUHIB

Gabriel EWENCZYK

Yasmine FATHALLAH

Mohamad FAWAZ

Cahier des charges

GL02 - A25 Sujet A. (SRU)

Préface.....	3
Introduction.....	5
Glossaire.....	6
Les Exigences.....	7
Exigences fonctionnelles :.....	7
Exigences non-fonctionnelles :.....	8
Spécification détaillée :.....	9
Formats des données (ABNF).....	16
Sémantique des données (Type abstrait - Créneau d'enseignement).....	17
Sémantique des données (Type abstrait - Collection des créneaux d'enseignement).....	18
Conclusion.....	20

Préface

Historique des révisions :

Version	Date	Auteurs	Modifications
1.0	02/11/2025	Thomas CHARPENTIER Ilias CHOUHIB Gabriel EWENCZYK Yasmine FATHALLAH Mohamad FAWAZ	Version initiale

Audience concernée :

Le présent cahier des charges est destiné à l'équipe de développement du projet de **GL02 (A25)** ainsi qu'à l'équipe pédagogique qui est le client. Il sert enfin de document contractuel à la réalisation du logiciel décrit.

L'équipe de développement est chargée de produire le logiciel tel que décrit dans le respect des règles et des exigences énoncées.

Ce document contient :

1. **Introduction** : Rappel du contexte, objectifs généraux du logiciel
2. **Glossaire** : Définit les termes techniques ou spécifiques au projet..
3. **La spécification générale des exigences** : Liste des exigences fonctionnelles et non-fonctionnelles de haut niveau.
4. **La spécification détaillée** : Description des exigences, format des données sous forme ABNF et types d'objets du logiciel.
5. **Conclusion** : Mise en perspective du projet.
6. **Annexes** : Informations supplémentaires.

Règles de révision :

1. Les demandes de modification suivant la version 1.0 devront faire l'objet d'une demande écrite de la part de l'équipe de développement ou du client.
2. La demande sera étudiée par l'ensemble de l'équipe liée à la réalisation du projet avant d'être acceptée.
3. Toute modification du cahier des charges entraînera l'ajout d'une entrée supplémentaire à l'historique des révisions ci-dessus ainsi qu'une incrémentation du numéro de version de ce dernier.

Introduction

L'Université centrale de la république de Sealand (SRU) fait face à des enjeux d'optimisation de ses salles de cours. Le système existant (système de planification générale des cours 'PGC') permet l'export des données au format CRU.

Afin de faciliter l'interaction avec ce dernier et augmenter l'accès aux informations, l'école souhaite mettre en place un utilitaire en invite de commande qui utilise le jeu de données du système au format CRU afin de permettre à l'utilisateur de les consulter de façon plus accessible.

Le logiciel (en invite de commande) devra permettre aux utilisateurs de :

- Suivre les salles liées à un cours
- Récupérer le nombre de places dans une salle spécifique
- Récupérer la disponibilité d'une salle spécifique à travers la semaine
- Récupérer les salles disponibles selon un créneau spécifique
- Exporter les créneaux de cours suivis entre 2 dates au format iCalendar (RFC 5545)
- Visualiser la capacité de toutes les salles
- Générer un rapport du taux d'occupation des salles

Glossaire

SRU : L'université centrale de la république de Sealand

PGC : Système de planification générale des cours

CRU : Format de fichier exporté par le PGC contenant les créneaux de cours, les salles, les horaires et les capacités.

iCalendar : Format de fichier structuré contenant des informations sur des évènements.

RegEx : Expression régulière (Permet de reconnaître des données spécifiques au sein d'une suite de caractères)

Les Exigences

Exigences fonctionnelles :

- **SPEC_FONC_1** : Le système doit être capable d'importer un fichier au format CRU.
- **SPEC_FONC_2** : Le système doit permettre aux utilisateurs de consulter la capacité maximale d'une salle.
- **SPEC_FONC_3** : Le système doit permettre aux utilisateurs de consulter les salles associées à un cours donné.
- **SPEC_FONC_4** : Le système doit permettre aux utilisateurs de voir quand une salle est disponible.
- **SPEC_FONC_5** : Le système doit permettre à un utilisateur de générer un fichier de type iCalendar entre deux dates données des cours auxquels il est assigné.
- **SPEC_FONC_6** : Le système doit pouvoir vérifier la qualité des données.
- **SPEC_FONC_7** : Le système doit être capable de générer un visuel du taux d'occupation de chaque salle.
- **SPEC_FONC_8** : Le système doit pouvoir générer un classement des salles en fonction de leur capacité d'accueil.

- **SPEC_FONC_9** : Le système doit permettre aux utilisateurs de visualiser toutes les salles disponibles pour un créneau spécifique.

Exigences non-fonctionnelles :

- **SPEC_NFONC_1** : L'application doit être sous forme d'invite de commandes
- **SPEC_NFONC_2** : Le logiciel doit garantir la cohérence des informations importées depuis le fichier CRU en vérifiant qu'aucune salle n'est utilisée simultanément par plusieurs cours et qu'aucun recouvrement de créneaux n'existe.

Spécification détaillée :

Identifiant : SPEC_FONC_1

Titre : Importation d'un fichier au format CRU

Objectifs : Permettre au système de charger les données d'emploi du temps générée par le PGC.

Préconditions :

- Le fichier CRU existe et peut être lu par le système.
- La structure du fichier CRU est correcte.

Postconditions :

- Les données sont importées et utilisables.

Entrées : Fichier CRU à importer

Traitements:

- Lecture du fichier
- Identification de la structure du fichier
- Extraction des informations
- Vérification de la conformité des données

Sorties : Données en mémoire

Gestion des erreurs :

- Si le fichier est introuvable : Message d'erreur
- Si le fichier CRU n'est pas bien structuré : Message d'erreur

Identifiant : SPEC_FONC_2

Titre : Consultation de la capacité maximale d'une salle

Objectifs : Permettre à un utilisateur de connaître le nombre maximum de places disponibles dans une salle donnée.

Préconditions :

- L'importation du fichier CRU a fonctionné.

Postconditions :

- Affichage de la capacité maximale de la salle

Entrées : L'identifiant de la salle

Traitements : Recherche de la salle indiquée et renvoie le nombre maximal de places.

Sorties : Nombre de places maximum dans la salle

Gestion des erreurs :

- Si la salle n'existe pas : Afficher un message (ex : "Salle inconnue")
-

Identifiant : SPEC_FONC_3

Titre : Consultation des salles d'un cours donné

Objectifs : Permettre à un utilisateur de connaître les différentes salles où un cours a lieu.

Préconditions :

- L'importation du fichier CRU a fonctionné.

Postconditions :

- Affichage de la liste des salles.

Entrées : L'identifiant du cours

Traitements : Recherche le cours indiqué et affiche la liste des salles correspondantes.

Sorties : Liste des salles où le cours a lieu

Gestion des erreurs :

- Si le cours n'existe pas : Message d'erreur

Identifiant : SPEC_FONC_4

Titre : Consultation de la disponibilité d'une salle

Objectifs : Permettre à un utilisateur de connaître la disponibilité d'une salle.

Préconditions :

- L'importation du fichier CRU a fonctionné.

Postconditions :

- Affichage des disponibilités de la salle

Entrées : L'identifiant de la salle

Traitements :

- Recherche de la salle indiquée
- Vérification des créneaux
- Renvoyer les disponibilités de la salle

Sorties : disponibilités de la salle

Gestion des erreurs :

- Si la salle n'existe pas : Message d'erreur
-

Identifiant : SPEC_FONC_5

Titre : Génération d'un fichier iCalendar

Objectifs : Permettre à un utilisateur de générer un fichier iCalendar de ses cours entre 2 dates.

Préconditions :

- L'utilisateur est identifié.
- L'importation du fichier CRU a fonctionné.

Postconditions :

- Un fichier iCalendar est créé.

Entrées :

- Date de début
- Date de fin

Traitements :

- Recherche des cours de l'utilisateur
- Tri en fonction des dates
- Génération du fichier iCalendar

Sorties : Téléchargement du fichier iCalendar

Gestion des erreurs :

- Si il y a une erreur dans la création du fichier : Message d'erreur
-

Identifiant : SPEC_FONC_6

Titre : Vérification de la qualité des données

Objectifs : Permettre au système de vérifier si les données du fichier CRU sont valides pour l'utilisation.

Préconditions :

- Le fichier au format CRU existe.

Postconditions :

- Indication de si le fichier est valide ou non

Entrées : Le fichier au format CRU

Traitements :

- Lecture du fichier
- Vérification des données : un créneau dans une salle ne peut être utilisé que par un seul enseignement et sans recouvrement possible.

Sorties : Indique si le fichier est valide

Gestion des erreurs :

- Si un conflit est détecté : Message d'erreur
 - Si il y a une erreur de lecture : Message d'erreur
-

Identifiant : SPEC_FONC_7

Titre : Génération d'un visuel du taux d'occupation

Objectifs : Permettre au gestionnaire de voir quelles salles sont sous-exploitées ou surexploitées.

Préconditions :

- L'importation du fichier CRU a fonctionné.

Postconditions :

- Visualisation générée.

Entrées : Ensemble des données des salles

Traitements :

- Calcul du taux d'occupations de chaque salle
- Génération d'un visuel graphique

Sorties : Affichage du visuel graphique

Gestion des erreurs :

- Si les données ne sont pas disponibles : Message d'erreur
-

Identifiant : SPEC_FONC_8

Titre : Génération d'un classement des salles par capacité d'accueil

Objectifs : Fournir un aperçu de la capacité des salles.

Préconditions :

- L'importation du fichier CRU a fonctionné.

Postconditions :

- Affichage du classement

Entrées : Ensemble des données des salles

Traitements :

- Récupération de la capacité de chaque salle
- Génération d'un classement

Sorties : Affichage du classement

Gestion des erreurs :

- Si les données ne sont pas disponibles : Message d'erreur

Identifiant : SPEC_FONC_9

Titre : Visualiser les salles disponible pour un créneau spécifique

Objectifs : Fournir une liste des salles disponibles à un certain horaire dans la semaine

Préconditions :

- L'importation du fichier CRU a fonctionné.

Postconditions :

- Affichage de la liste des salles disponibles

Entrées : Ensemble des données des salles, créneau voulu

Traitements :

- Récupération de la disponibilité de chaque salle
- Vérification de la correspondance avec le créneau saisi
- Génération de la liste de salles

Sorties : Affichage du classement

Gestion des erreurs :

Si les données ne sont pas disponibles : Message d'erreur

Identifiant : SPEC_NFONC_1

Titre : Interface en invite de commande

Objectifs : Permettre aux utilisateurs d'interagir avec l'application depuis une interface en ligne de commande.

Préconditions :

- Disposer d'une invite de commande

Postconditions :

- L'utilisateur peut exécuter toutes les commandes dans l'invite de commande.

Entrées : Commandes dans l'invite de commande

Traitements :

- Interprétation des commandes et exécution des fonctions correspondantes.

Sorties : Affichage des résultats dans la console

Gestion des erreurs :

- Si il y a une erreur avec le terminal : Message d'erreur
-

Identifiant : SPEC_NFONC_2

Titre : Fiabilité et intégrité des données

Objectifs : Garantir la cohérence des données du fichier CRU et éviter les recouvrements de créneaux.

Préconditions :

- Fichier CRU disponible et lisible.
- Format conforme (+Cours, créneaux).

Postconditions :

- Données cohérentes et sans conflit.

Entrées : Fichier CRU.

Traitements :

- Vérification de la validité des données.
Détection des conflits de salles ou d'horaires.

Sorties : Message de confirmation ou liste d'erreurs.

Gestion des erreurs :

- Si le fichier est invalide : Message d'erreur

Formats des données (ABNF)

Format de données CRU :

EmploiDuTemps = 1*Cours

Cours = '+' UE CRLF 1*(Créneau '//' CRLF)

UE = 2*4UPPERALPHA 2DIGIT

UPPERALPHA = %x41-5A ;Lettre majuscule

Créneau = '1' Type " Capacitaire " Horaire " Index " Salle

Type = 1UPPERALPHA 1DIGIT

Capacitaire = 'P=' 1*DIGIT

Horaire = 'H=' Jour WSD Heure '-' Heure

Jour = 'L' / 'MA' / 'ME' / 'J' / 'V' / 'S' / 'D'

Heure = 2DIGIT ':' 2DIGIT

Index = 1UPPERALPHA 1DIGIT

Salle = 'S=' 4(UPPERALPHA / DIGIT)

Format de données iCalendar (RFC 5545) :

EmploiDuTemps = Titre CRLF Version CRLF Prodid CRLF 1*(Event CRLF) Fin

Titre = 'BEGIN:VCALENDAR'

Version = 'VERSION:' 1*DIGIT '.' 1*DIGIT

Prodid = 'PRODID:' 1*VCHAR ;Texte permettant d'identifier le fichier

Event = DebutEvent CRLF Uid CRLF DTStamp CRLF DTStart CRLF DTEnd CRLF

Description CRLF FinEvent

DebutEvent = 'BEGIN:VEVENT'

Uid = 'UID:' 1*VCHAR

DTStamp = 'DTSTAMP:' Date ;Date de création du fichier

DTStart = 'DTSTART:' Date ;Date de début de l'événement

DTEnd = 'DTEND:' Date ;Date de fin de l'événement

Date = An Mois Jour 'T' Heure Minute Seconde 'Z' ;Date en UTC

An = 4*DIGIT

Mois = 2*DIGIT

Jour = 2*DIGIT
Heure = 2*DIGIT
Minute = 2*DIGIT
Seconde = 2*DIGIT
Description = 'SUMMARY:' 1*VCHAR
FinEvent = 'END:VEVENT'
Fin = 'END:VCALENDAR'

Sémantique des données (Type abstrait - Créneau d'enseignement)

Titre : CRENEAU

Sorte : Créneau

Description : Spécifie un ensemble de données associées au créneau d'enseignement, avec des opérations de création et de comparaison entre créneaux.

Références : Type, Capacitaire, Horaire, Jour, Heure, Index, Salle, Booléen

Signature :

CréerCréneau : Type x Capacitaire x Horaire x Index x Salle
-> Créneau

CréerHoraire : Jour x Heure x Heure -> Horaire

Compatibilité : Créneau x Créneau -> Booléen

RecouvrementHoraire : Horaire x Horaire -> Booléen

Axiomes :

```
Compatibilité (CréerCréneau (t1, c1, h1, i1, s1),  
CréerCréneau (t2, c2, h2, i2, s2)) = !RecouvrementHoraire  
(h1, h2)
```

```
RecouvrementHoraire (CréerHoraire (j1, heureDébut1,  
heureFin1), CréerHoraire (j2, heureDébut2, heureFin2)) = Vrai  
ssi j1 = j2  $\wedge$  ((heureDébut1 > heureFin2  $\wedge$  heureDébut1 <  
heureFin2)  $\vee$  (heureDébut2 > heureDébut1  $\wedge$  heureDébut2 <  
heureFin2) )
```

Sémantique des données (Type abstrait - Collection des créneaux d'enseignement)

Titre : COLLECTION

Sorte : Col

Description : Spécifie une collection de créneaux vide ou contenant un nombre spécifique de créneaux appartenant à une même matière. La collection permet, en plus d'opérations basiques, de vérifier si un créneau proposé est compatible avec les autres créneaux déjà inscrits.

Références : Créneau, Entier, Booléen

Précondition : On définit l'axiome Compatibilité entre deux créneaux (voir ci-dessus le type abstrait Créneau) comme renvoyant Vrai s'il n'y a pas de recouvrement entre les horaires des deux créneaux et Faux sinon.

Signature :

Créer : \rightarrow Col

Ajouter : Col \times Créneau \rightarrow Col

Card : Col \rightarrow Entier

Appartient : Col \times Créneau \rightarrow Booléen

Recouvrement : Col \times Créneau \rightarrow Booléen

Axiomes :

Appartient (Créer(), cr1) = Faux

Appartient (Ajouter (col, cr1), cr2) = Vrai ssi cr1 = cr2 V

Appartient (col, cr2)

Card (Créer()) = 0

Card (Ajouter (col, cr1)) = Card (col) + 1

Recouvrement (Créer(), h) = Faux

Recouvrement (Ajouter (col, cr1), cr2) = Vrai ssi

!Compatibilité (cr1, cr2) V Recouvrement (col, cr2)

Conclusion

Le projet devra prendre la forme d'un exécutable avec lequel il sera possible d'interagir en passant par un invite de commande. Il devra pouvoir importer des fichiers au format CRU du système en place ainsi que générer des fichiers au format .ics (iCalendar)

Le développement du projet pourra utiliser des RegEx afin de valider le format de données du fichier importé ainsi que des tests unitaires afin de s'assurer de la non-régression du système d'import et de lecture des données.