



# Cahier des charges (V1)

GL02

GUTHIER Guillaume, BOUSMAHA--JOUVE Théo, KUNTZ Antoine

25/10/2024

## Préface

Ce cahier des charges est destiné à l'équipe de programmeurs chargés de la mise en place d'un logiciel de suivi d'occupations de salles de cours de l'université centrale de la république de Sealand. Il contient les consignes du client et les informations requises pour établir ce projet.

Dans ce cahier des charges, vous trouverez en premier lieu, une liste des exigences fonctionnelles et non fonctionnelles de l'outil à développer. Cette liste contient une description succincte de chacune de ces exigences.

Une spécification plus précise et détaillée de ces exigences est ensuite présentée. Toute information nécessaire à l'implémentation de ces fonctions est écrite. Le contexte de leur utilisation, les utilisateurs cible de la fonctionnalité, leur motivation à l'utiliser ainsi que les scénarios principaux et alternatifs d'utilisation de la fonctionnalité sont présentés afin d'assurer leur implémentation correcte.

Votre logiciel devra traiter différentes données, des informations vous sont transmises dans ce cahier des charges pour les manipuler au mieux. D'abord, le format des données externes à votre logiciel est spécifié, les emplois du temps au format CRU que votre projet devra pouvoir analyser pour transmettre les informations voulues aux utilisateurs ainsi que les emplois du temps au format iCalendar que votre application doit pouvoir fournir aux utilisateurs.

Ensuite, la sémantique des données utilisées au sein de votre logiciel est spécifiée, c'est-à-dire les créneaux d'enseignement au sein de l'université ainsi que les collections de ces dits créneaux.

## Introduction

Faisant suite à la demande de l'université centrale de la république de Sealand (SRU) d'amélioration de son système de gestion des locaux, ce cahier des charges a pour objectif d'encadrer et d'accompagner le développement d'un logiciel répondant à cette demande. Pour cela, ce cahier des charges tient compte de divers buts afin que le logiciel puisse satisfaire au mieux ces exigences.

Le logiciel devra ainsi servir d'outil de suivi d'occupation des salles de cours, tant pour les étudiants que les enseignants ou le personnel. Il doit donc permettre de connaître les horaires et les salles liées à un cours ainsi que leurs capacités, doit également permettre de savoir quelles salles sont libres pour un créneau donné, et enfin, d'un point de vue plus pratique, permettre de générer des fichiers iCalendar à partir de ces données pour faciliter le quotidien des utilisateurs en leur permettant d'intégrer ces informations à un agenda.

Finalement, il devra aussi avoir des fonctionnalités supplémentaires pour le gestionnaire des locaux, qui doit notamment pouvoir suivre l'occupation de ces salles, avec des statistiques comme le taux d'occupation moyen de la salle, afin de pouvoir enquêter sur la sous-exploitation ou surexploitation des salles et décider d'éventuels aménagements. Le logiciel doit également s'assurer qu'une même salle ne soit pas occupée par deux cours simultanément.

L'objectif des spécifications qui vont suivre est donc de répondre à ces critères afin que le logiciel permette à l'université de gérer ses locaux de la façon la plus fluide et efficace possible à l'avenir de manière durable.

# Spécification générale des exigences

## I. Exigences fonctionnelles

### **SPEC1 : Recherche des salles associées à un cours**

- Description : L'utilisateur (étudiant ou enseignant) doit pouvoir rechercher les salles associées à un cours donné.
- Entrée : Nom du cours (ex: "MC01").
- Sortie : Liste des salles où les créneaux de ce cours sont programmés.

### **SPEC2 : Capacité maximale des salles**

- Description : L'utilisateur doit pouvoir consulter la capacité maximale en termes de places pour une salle donnée.
- Entrée : Code de la salle (ex: "P202").
- Sortie : Capacité maximale (nombre de places).

### **SPEC3 Disponibilité des salles**

- Description : Les utilisateurs doivent pouvoir voir quand une salle est libre pendant la semaine.
- Entrée : Code de la salle.
- Sortie : Créneaux horaires où la salle est libre durant la semaine.

### **SPEC4 : Salles libres pour un créneau donné**

- Description : Les utilisateurs doivent pouvoir connaître quelles salles sont libres pour un créneau horaire spécifique.
- Entrée : Jour de la semaine, plage horaire (ex: "mardi, 10:00-12:00").
- Sortie : Liste des salles libres pendant ce créneau.

### **SPEC5 Exportation des cours au format iCalendar**

- Description : Les utilisateurs doivent pouvoir exporter les cours auxquels ils participent (CM, TD, TP) sous forme de fichier iCalendar (RFC 5545) pour une période donnée.
- Entrée : Période (deux dates de début et de fin), cours ou groupe.
- Sortie : Fichier iCalendar contenant les événements liés aux cours pour cette période.

### **SPEC6 : Vérification des conflits d'emploi du temps**

- Description : Le système doit vérifier la validité des données d'emploi du temps et détecter les conflits dans l'occupation des salles (deux créneaux sur la même plage horaire dans la même salle).
- Entrée : Emploi du temps global.
- Sortie : Rapport des conflits potentiels (recouvrements de créneaux dans une même salle).

### **SPEC7 : Analyse de l'occupation des locaux**

- Description : Le système doit permettre une visualisation du taux d'occupation des salles. Il doit aussi proposer un classement des salles par capacité et taux d'occupation (nombre de places, taux d'utilisation).
- Sortie : Visualisation graphique du taux d'occupation (ex: histogramme, diagramme circulaire) et classement des salles selon leur capacité et leur taux d'occupation.

## **II. Exigences non fonctionnelles**

### **SPEC\_NF1 : Accessibilité**

- Interface utilisateur : Le système doit être simple à utiliser via une interface en ligne de commande, et éventuellement une interface graphique optionnelle pour certains types d'utilisateurs.

### **SPEC\_NF2 : Sécurité**

- Authentification : Le système doit inclure un mécanisme d'authentification pour s'assurer que seuls les étudiants et enseignants autorisés peuvent accéder à certaines fonctionnalités, comme l'exportation des données au format iCalendar.

### **SPEC\_NF3 : Extensibilité et maintenabilité**

- Extensibilité : Le système doit pouvoir s'adapter à des ajouts futurs (par exemple, intégration avec d'autres systèmes d'information ou ajout de nouvelles fonctionnalités).
- Maintenabilité : Le code doit être documenté et structuré pour faciliter la maintenance et les futures modifications.

### **SPEC\_NF4 : Utilisabilité**

- Facilité d'utilisation : L'outil doit être intuitif pour des utilisateurs non techniques (étudiants, enseignants) et permettre une navigation fluide entre les différentes fonctionnalités (recherche de salle, disponibilité, etc.).

## **Spécifications détaillées**

### **I. Détail des exigences**

#### **Identifiant : SPEC1**

Titre : Recherche des salles associées à un cours

Acteur : Visiteur

Contexte d'utilisation : Un utilisateur peut savoir quelles salles sont utilisées par un certain cours. Il doit saisir le code du cours (ex : MC01) pour lequel il recherche cette information.

Motivation : L'utilisateur souhaite savoir dans quelle salle se rendre pour participer à un cours.

Précondition(s) : Néant

Postcondition(s) : Néant

Déclencheur : L'utilisateur sélectionne l'option « recherche de salle » sur le menu de l'application.

Scénario principal :

1- Il est demandé à l'utilisateur de renseigner le code du cours pour laquelle il cherche les salles associées.

2-L'utilisateur rentre le code du cours.

3-Une liste des salles associées s'affiche à l'écran. Les noms de chaque salle dans la liste sont séparés par des retours à la ligne.

Scénario(s) alternatif(s) :

2a-Le code du cours entré par l'utilisateur ne correspond à aucun code existant dans les données. Un message expliquant que le cours est introuvable et qu'une erreur dans la rédaction du code a pu avoir lieu s'affiche.

3a-Il n'y a pas de salle associée au cours demandé par l'utilisateur. Un message indiquant qu'aucune salle n'a pu être trouvée s'affiche alors

### **Identifiant : SPEC2**

Titre : Capacité maximale des salles

Acteur : Visiteur

Contexte d'utilisation : Un visiteur peut savoir combien de personnes peut accueillir une salle. Il doit saisir le numéro de la salle pour laquelle il recherche cette information.

Motivation : L'utilisateur souhaite vérifier qu'une salle est assez grande pour réaliser une activité.

Précondition(s) : Néant

Postcondition(s) : Néant

Déclencheur : L'utilisateur sélectionne l'option « capacité maximale des salles » sur le menu de l'application.

Scénario principal :

1- Il est demandé à l'utilisateur de renseigner le numéro de la salle dont il cherche la capacité.

2-L'utilisateur rentre le numéro de la salle.

3-Une phrase indiquant le nombre de personnes que peut accueillir la salle s'affiche.

Scénario(s) alternatif(s) :

2a-Le numéro de salle entré par l'utilisateur ne correspond à aucun numéro existant dans les données. Un message expliquant que la salle est introuvable et qu'une erreur dans la rédaction du numéro a pu avoir lieu s'affiche.

### **Identifiant : SPEC3**

Titre : Disponibilités des salles

Acteur : Visiteur

Contexte d'utilisation : Un visiteur peut savoir quand une salle est occupée ou non. Il doit saisir le numéro de la salle pour laquelle il recherche cette information.

Motivation : L'utilisateur souhaite utiliser une salle de l'établissement en particulier.

Précondition(s) : Néant

Postcondition(s) : Néant

Déclencheur : L'utilisateur sélectionne l'option « disponibilité des salles » sur le menu de l'application.

Scénario principal :

1- Il est demandé à l'utilisateur de renseigner le numéro de la salle dont il cherche les disponibilités

2-L'utilisateur rentre le numéro de la salle.

3-Une liste des horaires où elle est disponible s'affiche à l'écran. Les horaires dans la liste précisent l'heure et le jour de la semaine et sont séparés par des retours à la ligne.

Scénario(s) alternatif(s) :

2a-Le numéro de salle entré par l'utilisateur ne correspond à aucun numéro existant dans les données. Un message expliquant que la salle est introuvable et qu'une erreur dans la rédaction du numéro a pu avoir lieu s'affiche.

3a-La salle demandée par l'utilisateur n'est jamais libre. Un message indiquant qu'aucun horaire n'a pu être trouvé s'affiche alors.

#### **Identifiant : SPEC4**

Titre : Salles libres pour un créneau donné

Acteur : Visiteur

Contexte d'utilisation : Un visiteur peut savoir quelles salles sont libres sur une période donnée. Il doit saisir le jour et la plage horaire pour lesquels il recherche cette information.

Motivation : L'utilisateur souhaite travailler dans une salle lors d'une période où il est libre.

Précondition(s) : Néant

Postcondition(s) : Néant

Déclencheur : L'utilisateur sélectionne l'option « salles libres pour un créneau donnée » sur le menu de l'application.

Scénario principal :

1- Il est demandé à l'utilisateur de renseigner le jour où il cherche une salle libre

2-L'utilisateur rentre le jour de la semaine qui l'intéresse.

3-Il est demandé à l'utilisateur de renseigner la plage horaire où il cherche une salle libre au format hh:mm-hh:mm.

4-L'utilisateur rentre la plage horaire qui l'intéresse.

5-Une liste de salles disponibles s'affiche à l'écran. Les salles dans la liste sont séparées par des retours à la ligne.

Scénario(s) alternatif(s) :

2a-La chaîne de caractères entrée par l'utilisateur ne correspond pas à un jour de la semaine. Un message expliquant qu'une erreur dans l'orthographe du mot a pu avoir lieu s'affiche.

4a-La plage horaire entrée par l'utilisateur ne respecte pas le format demandé. Un message expliquant que la plage horaire doit être donnée sous la forme hh:mm-hh:mm ainsi qu'un exemple de plage horaire sous cette forme s'affichent.

5a-Aucune salle n'est libre lors de la période demandée par l'utilisateur. Un message indiquant qu'aucune salle n'a pu être trouvée s'affiche alors.



### **Identifiant : SPEC5**

Titre : Exportation des cours au format iCalendar

Acteur : Visiteur (Étudiant ou Enseignant)

Contexte d'utilisation : Un utilisateur souhaite intégrer ses horaires de cours dans son calendrier personnel pour une meilleure organisation. Il doit pouvoir sélectionner la période, les cours ou le groupe pour générer un fichier iCalendar.

Motivation : L'utilisateur veut ajouter les créneaux de ses cours dans son application de calendrier préférée.

Précondition(s) : Les cours doivent être bien renseignés avec leurs créneaux horaires.

Postcondition(s) : Néant

Déclencheur : L'utilisateur sélectionne l'option « Exporter au format iCalendar » dans le menu de l'application.

Scénario principal :

1- L'utilisateur sélectionne les dates de début et de fin de la période concernée.

2- L'utilisateur choisit les cours ou le groupe qu'il souhaite exporter.

3- Un fichier iCalendar est généré et proposé en téléchargement.

Scénario(s) alternatif(s) :

2a- Aucun cours n'est associé à la période ou au groupe choisi. Un message informant qu'aucun cours n'a été trouvé pour cette sélection est affiché.

3a- Une erreur de génération survient, le fichier iCalendar n'est pas créé. Un message d'erreur s'affiche et invite à réessayer plus tard.

### **Identifiant : SPEC6**

Titre : Vérification des conflits d'emploi du temps

Acteur : Administrateur

Contexte d'utilisation : L'administrateur souhaite vérifier la cohérence des données d'emploi du temps afin de détecter les éventuels conflits d'occupation.

Motivation : Éviter les conflits de créneaux dans une même salle, garantissant ainsi une utilisation fiable et non conflictuelle des locaux.

Précondition(s) : Les données d'emploi du temps doivent être importées dans le système.

Postcondition(s) : Néant

Déclencheur : L'administrateur sélectionne l'option « Vérification des conflits » dans le menu du logiciel.

Scénario principal :

1- Le système analyse les données d'emploi du temps pour repérer les recouvrements de créneaux dans une même salle.

2- Si un conflit est trouvé, il est ajouté dans le rapport avec les détails du créneau concerné.

3- Un rapport listant tous les conflits identifiés est généré et affiché

Scénario(s) alternatif(s) :

2a- Aucun conflit n'est trouvé dans les données d'emploi du temps. Un message précisant qu'aucun conflit n'a été détecté est affiché.

#### **Identifiant : SPEC7**

Titre : Analyse de l'occupation des locaux

Acteur : Administrateur

Contexte d'utilisation : L'administrateur souhaite obtenir une vision synthétique de l'utilisation des salles pour analyser et optimiser l'occupation des locaux.

Motivation : Aider à identifier les salles sous-utilisées ou sur-utilisées afin de mieux organiser les ressources et potentiellement optimiser les aménagements futurs.

Précondition(s) : Les données d'emploi du temps et les capacités des salles doivent être disponibles.

Postcondition(s) : Néant

Déclencheur : L'utilisateur sélectionne l'option « Analyse de l'occupation des locaux » dans le menu de l'application.

Scénario principal :

1- Le système génère une visualisation du taux d'occupation des salles.

2- Un classement des salles est produit en fonction de leur capacité et de leur taux d'occupation.

3- Les résultats sont affichés pour consultation.

Scénario(s) alternatif(s) :

2a- Si aucune donnée de taux d'occupation n'est disponible, un message indiquant l'absence de données s'affiche.

#### **Identifiant : SPEC\_NF1**

Titre : Accessibilité

Acteur : Visiteur

Contexte d'utilisation : Un utilisateur doit pouvoir accéder facilement à la fonctionnalité de recherche de salles, que ce soit par le biais d'une interface en ligne de commande ou d'une interface graphique, selon ses préférences.

Motivation : L'utilisateur souhaite que la recherche de salles soit accessible, intuitive et efficace, afin de minimiser le temps et l'effort nécessaires pour trouver les informations sur les salles.

Précondition(s) : Néant

Postcondition(s) : Néant

Déclencheur : L'utilisateur sélectionne l'option « recherche de salle » sur le menu de l'application.

Scénario principal :

1- L'utilisateur est accueilli par une interface claire, avec des instructions pour saisir le code du cours. Un bouton s'affiche dans un coin, lui proposant de changer pour une interface graphique.

2- L'utilisateur choisit de continuer en saisissant le code du cours.

3- Les résultats s'affichent comme indiqués en SPEC1.

Scénario(s) alternatif(s) :

2a- Le code du cours entré par l'utilisateur ne correspond à aucun code existant dans les données. Un message explicatif s'affiche, signalant que le cours est introuvable et suggérant une vérification du code.

3a- Il n'y a pas de salle associée au cours demandé par l'utilisateur. Un message clair indique qu'aucune salle n'a pu être trouvée.

4a- L'utilisateur clique sur le bouton "Interface graphique".

4b- Des cases indiquant chaque code de cours, dans l'ordre alphabétique, apparaissent.

4c- L'utilisateur clique sur un code de cours.

4d- Les résultats des salles et horaires correspondant au cours s'affichent.

## **Identifiant : SPEC\_NF2**

Titre : Sécurité

Acteur : Visiteur

Contexte d'utilisation : Un utilisateur (étudiant ou enseignant) doit pouvoir accéder à des fonctionnalités sensibles de l'application, telles que l'exportation de données au format iCalendar.

Motivation : Garantir que seules les personnes autorisées puissent accéder à des informations privées et effectuer des actions sensibles dans le système.

Précondition(s) : L'utilisateur doit avoir un compte valide et être enregistré dans le système.

Postcondition(s) : L'utilisateur est authentifié et peut accéder aux fonctionnalités autorisées, ou un message d'erreur est affiché en cas d'échec d'authentification.

Déclencheur : L'utilisateur tente d'accéder à une fonctionnalité nécessitant une authentification.

Scénario principal :

1- L'utilisateur est redirigé vers une page de connexion.

2- Il entre ses identifiants (nom d'utilisateur et mot de passe).

3- Le système vérifie les informations d'identification.

4- Si les identifiants sont corrects, l'utilisateur est redirigé vers la fonctionnalité demandée.

Scénario(s) alternatif(s) :

2a- Si l'utilisateur entre un identifiant ou un mot de passe incorrect, un message d'erreur s'affiche, indiquant que l'authentification a échoué.

3a- Si l'utilisateur ne dispose pas des droits nécessaires pour accéder à la fonctionnalité, un message l'informe de l'absence d'autorisation.

4a- Un bouton “Mot de passe oublié” s’affiche pour proposer à l’utilisateur de réinitialiser son mot de passe

4b- Il entre son identifiant ou son adresse mail.

4c- Un email lui est envoyé avec un code temporaire.

4d- L’utilisateur entre le code temporaire.

4e- L’utilisateur peut alors choisir un nouveau mot de passe, qu’il entre également.

4f- Pour éviter les erreurs, l’utilisateur doit réécrire son mot de passe.

4g- Le mot de passe de l’utilisateur est modifié.

5a- Si la confirmation du nouveau mot de passe échoue, l’utilisateur recommence à l’étape 4e.

### **Identifiant : SPEC\_NF3**

Titre : Extensibilité et maniabilité

Acteur : Administrateur

Contexte d’utilisation : Les développeurs et administrateurs doivent pouvoir effectuer des modifications ou des ajouts au système sans complications majeures.

Motivation : Assurer la durabilité et la flexibilité du système pour répondre à l’évolution des besoins et intégrer de nouvelles fonctionnalités ou systèmes.

Précondition(s) : La structure du code et la documentation doivent être en place avant toute modification. Le système doit fonctionner.

Postcondition(s) : Le système doit fonctionner correctement après des modifications ou des ajouts, sans introduire de nouveaux bugs ou rendre inutilisables d’autres fonctionnalités de manière imprévue.

Déclencheur : Les besoins d’évolution du système ou l’ajout de nouvelles fonctionnalités sont identifiés et ajoutés.

Scénario principal :

1- Le développeur identifie une nouvelle fonctionnalité à ajouter ou un système à intégrer.

2- Le code est examiné pour s’assurer qu’il est bien documenté et structuré.

3- Les modifications sont effectuées en suivant les bonnes pratiques de développement.

4- Des tests sont réalisés pour vérifier que le système fonctionne toujours comme prévu.

5- Les modifications sont implémentées.

Scénario(s) alternatif(s) :

2a- Si le code n’est pas bien structuré ou documenté, cela complique les modifications. Le développeur doit d’abord améliorer la documentation avant d’ajouter de nouvelles fonctionnalités.

4a- Si les tests révèlent des problèmes après les modifications, le développeur doit corriger les erreurs identifiées avant de finaliser l'intégration.

#### **Identifiant : SPEC\_NF4**

Titre : Utilisabilité

Acteur : Visiteur

Contexte d'utilisation : Les utilisateurs non techniques, tels que les étudiants et les enseignants, doivent pouvoir interagir facilement avec l'application.

Motivation : Assurer que l'outil soit accessible à tous, facilitant ainsi l'adoption et l'utilisation des fonctionnalités disponibles.

Précondition(s) : Le reste du système doit fonctionner correctement.

Postcondition(s) : Néant

Déclencheur : L'utilisateur lance l'application et navigue vers une fonctionnalité spécifique.

Scénario principal :

1- L'utilisateur ouvre l'application et est accueilli par un écran d'accueil clair et organisé.

2- Il navigue facilement entre les différentes options (recherche de salle, disponibilité, etc.) grâce à une interface intuitive.

3- L'utilisateur choisit une fonctionnalité et suit les instructions clairement affichées pour accomplir sa tâche sans difficulté.

Scénario(s) alternatif(s) :

2a- Si l'utilisateur se sent perdu dans l'interface, il peut accéder à une aide contextuelle ou un guide utilisateur qui l'assiste, via un bouton dans un coin "Aide et assistance"

3a- L'utilisateur peut à tout moment retourner au menu principal via un bouton dans un coin.

## **II. Format de données**

### **SPÉCIFICATIONS EN ABNF DES EMPLOIS DU TEMPS EN .CRU**

Emploi\_du\_temps = Cours Emploi\_du\_temps / Cours

Cours = '+' 2ALPHA 2DIGIT CRLF Créneau

Créneau = DIGIT ',' ALPHA DIGIT 'P=' 2DIGIT 'H=' Jour WSP Heure '-' Heure ','  
ALPHA DIGIT 'S=' ALPHA 3DIGIT Fin

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

Heure = 2DIGIT ':' 2DIGIT

Fin = '/' CRLF Créneau / '/'

## SPÉCIFICATIONS EN ABNF DU FORMAT ICALENDAR

contentline = name \*(";" param) ":" value CRLF

name = iana-token / x-name

iana-token = 1\*(ALPHA / DIGIT / "-")  
; iCalendar identifier registered with IANA

x-name = "X-" [vendorid "-"] 1\*(ALPHA / DIGIT / "-")

vendorid = 3\*(ALPHA / DIGIT)

param = param-name "=" param-value \*(";" param-value)

param-name = iana-token / x-name

param-value = paramtext / quoted-string

paramtext = \*SAFE-CHAR

value = \*VALUE-CHAR

quoted-string = DQUOTE \*QSAFE-CHAR DQUOTE

QSAFE-CHAR = WSP / %x21 / %x23-7E / NON-US-ASCII  
; Any character except CONTROL and DQUOTE

SAFE-CHAR = WSP / %x21 / %x23-2B / %x2D-39 / %x3C-7E  
/ NON-US-ASCII

VALUE-CHAR = WSP / %x21-7E / NON-US-ASCII

NON-US-ASCII = UTF8-2 / UTF8-3 / UTF8-4

CONTROL = %x00-08 / %x0A-1F / %x7F

### III. Sémantique des données

**Titre :** Créneau

**Sorte :** Creneau

**Description :** Spécifie un créneau d'enseignement identifié par son type de cours, sa capacité, son horaire, son index de sous-groupe et la salle. Possibilité de vérifier l'équivalence entre deux créneaux.

**Références :**

String, Entier, Booléen

**Signature :**

CréerCreneau : TypeCours x Capacitaire x Horaire x SousGroupe x Salle  $\rightarrow$  Creneau

TypeCours : Creneau  $\rightarrow$  String

Capacitaire : Creneau  $\rightarrow$  Entier

Horaire : Creneau  $\rightarrow$  String

SousGroupe : Creneau  $\rightarrow$  String

Salle : Creneau  $\rightarrow$  String

Egale : Creneau x Creneau  $\rightarrow$  Booléen

**Axiomes :**

TypeCours(CréerCreneau(tc, cap, horaire, sg, salle)) = tc

Capacitaire(CréerCreneau(tc, cap, horaire, sg, salle)) = cap

Horaire(CréerCreneau(tc, cap, horaire, sg, salle)) = horaire

SousGroupe(CréerCreneau(tc, cap, horaire, sg, salle)) = sg

Salle(CréerCreneau(tc, cap, horaire, sg, salle)) = salle

Egale(CréerCreneau(tc1, cap1, horaire1, sg1, salle1), CréerCreneau(tc2, cap2, horaire2, sg2, salle2))  
 $\Rightarrow$  Vrai ssi tc1 = tc2 et cap1 = cap2 et horaire1 = horaire2 et sg1 = sg2 et salle1 = salle2

**Titre :** Collection

**Sorte :** Collection<Creneau>

**Références :** Creneau, Booléen, Entier

**Description :** Spécifie un ensemble vide ou contenant des éléments de type Creneau. Permet l'ajout et les opérations ensemblistes d'union et intersection.

**Signature :**

CréerCollection :  $\rightarrow$  Collection<Creneau>

Ajouter : Collection<Creneau> x Creneau  $\rightarrow$  Collection<Creneau>

Union : Collection<Creneau> x Collection<Creneau>  $\rightarrow$  Collection<Creneau>

Intersection : Collection<Creneau> x Collection<Creneau>  $\rightarrow$  Collection<Creneau>

**Axiomes :**

Union(Ajouter(collec, creneau), CréerCollection) = Ajouter(collec, creneau)

Intersection(Ajouter(collec, creneau), CréerCollection) = CréerCollection



## Conclusion

L'objectif de ce projet est d'offrir aux utilisateurs une manière simple et claire de consulter diverses informations en lien avec les salles de l'université. Les étudiants comme les enseignants doivent pouvoir savoir dans quelles salles ont lieu quels cours, combien de places contient une salle ainsi que quelles salles sont libres et quand et plus encore.

Ces informations devront être transmises à l'utilisateur sans ambiguïté et si une erreur a lieu, il est nécessaire de le faire savoir à l'utilisateur.

Les quatre premières spécifications mentionnées dans ce cahier des charges sont indispensables. Il s'agit des tâches les plus basiques que votre application doit exécuter et représentent son intérêt premier.

Nous vous rappelons que votre logiciel doit se présenter sous la forme d'un utilitaire en ligne de commande. Pour faciliter l'expérience utilisateur, il vous est conseillé de lui présenter toutes les fonctionnalités de votre application et de les associer à un nombre, l'utilisateur n'aurait alors qu'à entrer le chiffre correspondant à la fonction de son choix. Cette présentation est une des plus simples possibles mais vous n'êtes pas restreint dans la manière dont l'utilisateur accède aux fonctionnalités de votre logiciel. Gardez cependant toujours à l'esprit que votre produit se doit d'être utilisable aisément par tous ses utilisateurs.

## Bibliographie

Source de la spécification en ABNF du format iCalendar:

DESRUISSEAUX Bernard. *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*, Section 3.1. Septembre 2009. Disponible sur : <https://datatracker.ietf.org/doc/html/rfc5545>