

# **Mise à jour d'un calendrier**

## **Spécification d'exigences logicielles**

**Version <1.2>**  
**07/11/2012**

**GOHIER - ROBERGE**

Mise à jour d'un calendrier en ligne	Version : <1.2>
Spécification d'exigences logicielles	Date : 07/11/12

## Historique des modifications du document

Date	Version	Description	Auteur
15/10/12	1.0	Analyse du besoin :Mettre à jour un calendrier Google Agenda à partir d'un fichier ICS local ou en ligne.	GOHIER Brian – ROBERGE Thibaut
22/10/12	1.1	Élaboration des cas d'utilisation	GOHIER Brian – ROBERGE Thibaut
03/11/12	1.2	Rédactions des exigences	GOHIER Brian – ROBERGE Thibaut

Mise à jour d'un calendrier en ligne	Version : <1.2>
Spécification d'exigences logicielles	Date : 07/11/12

## Table des matières

1. Introduction.....	4
1.1 Objectif du document.....	4
1.2 Portée du document.....	4
1.3 Définitions, acronymes et abréviations.....	4
1.4 Références.....	4
1.5 Vue d'ensemble.....	4
2. Description générale.....	5
2.1 Perspectives du produit.....	5
2.1.1 Interfaces système.....	5
2.1.2 Interfaces utilisateurs.....	5
2.1.3 Interfaces matérielles.....	5
2.1.4 Interfaces de communication.....	6
2.1.5 Contraintes de mémoire.....	6
2.2 Fonctions du produit.....	6
2.3 Caractéristiques des utilisateurs.....	6
2.4 Contraintes.....	6
2.5 Hypothèses et dépendances.....	6
2.6 Exigences reportées.....	6
3. Exigences spécifiques.....	6
3.1 Fonctionnalités.....	6
3.2 Spécification des cas d'utilisation.....	6
3.3 Exigences supplémentaires.....	12
3.3.1 Utilisabilité .....	12
3.3.2 Fiabilité.....	12
3.3.3 Performance.....	12
3.3.4 Maintenabilité.....	12
4. Contraintes de conception.....	12
5. Sécurité.....	12
6. Exigences de documentation utilisateur et d'aide en ligne.....	12
7. Normes applicables .....	12

Mise à jour d'un calendrier en ligne	Version : <1.2>
Spécification d'exigences logicielles	Date : 07/11/12

# Spécification d'exigences logicielles

## 1. Introduction

Cette spécification des exigences logicielles (SEL) est un descriptif des exigences d'une interface de synchronisation avec un Google Agenda. Premièrement nous présenterons une vue d'ensemble du logiciel et sa portée, ensuite nous exposerons le programme de manière générale comprenant ses différents cas d'utilisation, et enfin nous le décrirons le programme en détails.

### 1.1 Objectif du document

Cette spécification des exigences logicielles (SEL) est un descriptif des exigences d'une application de gestion d'emploi du temps. Nous y présenterons notre document SEL ainsi que le logiciel, leurs objectifs, leur portée et le vocabulaire spécifique employé.

### 1.2 Portée du document

Le logiciel a pour but de mettre à jour un calendrier en ligne « Google Agenda » (utilisant le protocole CalDav) à partir d'un emploi du temps de cours au format ICS. Le fichier ICS pourra être un fichier enregistré en local ou un URL (ex : <http://www.edt-sciences.univ-nantes.fr/g20291.ics>).

Ici, les utilisateurs seront des étudiants, qui exploiteront leur emploi du temps scolaire à l'aide du logiciel afin que leurs horaires de cours soient mis à jour sur leur calendrier Google. De plus, Ils pourront ajouter d'autres informations personnelles, en plus de leur emploi du temps scolaire.

### 1.3 Définitions, acronymes et abréviations

**Google Agenda** : Calendrier Google gratuit en ligne.

**Utilisateur** : Une personne qui se sert du logiciel pour mettre à jour son calendrier Google Agenda par rapport à son emploi du temps de cours.

**Fichier ICS** : Fichier contenant du texte définissant un emploi du temps.

**CalDav** : Protocole ouvert qui vous laisse éditer un calendrier, ici un Google Agenda, avec un autre dispositif.

**IHM** : Interface Homme-Machine, l'interface graphique du logiciel.

**SWING** : Une librairie dédiée au langage Java permettant de faire une interface fenêtrée.

**Checkbox** : Élément graphique en forme de case à cocher.

**Java** : Langage de programmation orienté objet pouvant s'exécuter sur tout type de machine.

### 1.4 Références

CalDAV4j Overview : <http://code.google.com/p/caldav4j/> Google

### 1.5 Vue d'ensemble

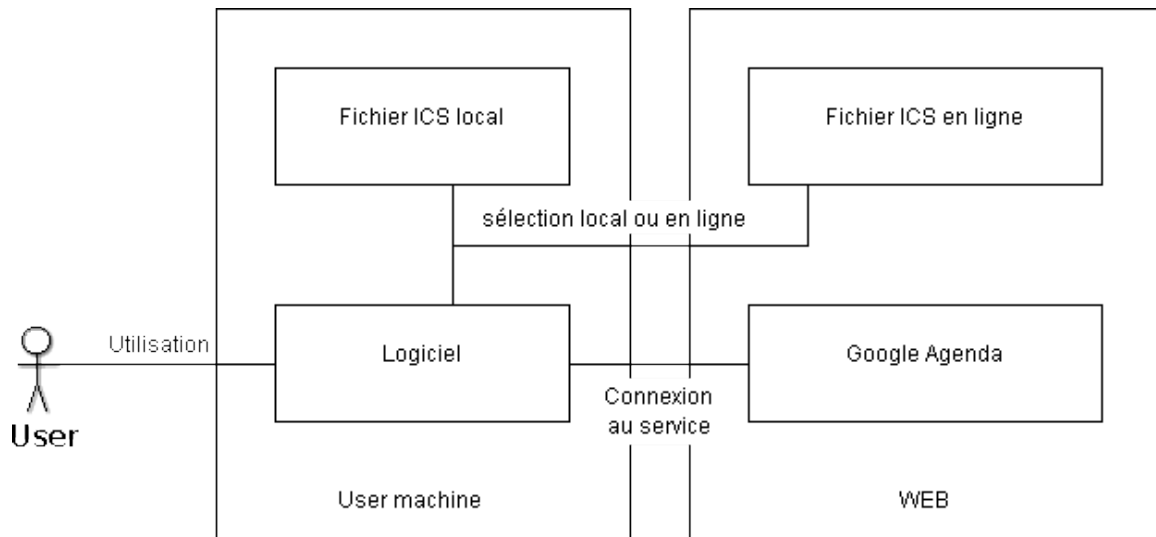
Ce document présente tout d'abord une description générale du logiciel, décrivant chacune des interfaces en prenant en compte chaque facteur qui affecte le produit, les différentes fonctions, les compétences requises de l'utilisateur et les contraintes. Nous décrivons ensuite plus en détails les exigences du projet, en décrivant les cas d'utilisation, la performance recommandée, sa fiabilité, sa maintenabilité. Nous terminerons enfin par les contraintes de conception et de sécurité.

Mise à jour d'un calendrier en ligne	Version : <1.2>
Spécification d'exigences logicielles	Date : 07/11/12

## 2. Description générale

### 2.1 Perspectives du produit

#### 2.1.1 Interfaces système



L'utilisateur interagit avec le logiciel. Il choisit un fichier ICS local ou en ligne. Il entre ses identifiants Google Agenda.

Le logiciel va ensuite parser le fichier ICS pour en extraire les cours. Il va ensuite comparer ces derniers avec ceux présents sur le calendrier Google Agenda. Il met à jour les cours. En cas de conflit avec un événement autre qu'un cours, il va notifier l'utilisateur.

L'utilisateur valide la mise à jour. Si un cours est en conflit avec un autre élément de l'emploi du temps qui n'est pas un cours, (que nous appellerons désormais événement) il devra choisir entre garder le cours ou garder cet événement.

#### 2.1.2 Interfaces utilisateurs

L'interface utilisateur devra être fenêtrée (Java SWING). Lors du lancement l'utilisateur devra renseigner le chemin du fichier ICS ainsi que ses identifiants Google Agenda.

Pour faciliter la lecture des deux versions, le logiciel devra afficher l'emploi du temps par semaine (de la semaine en cours par défaut) du calendrier tiré du fichier ICS ainsi que celui présent sur Google Agenda, et l'utilisateur pourra naviguer de semaine en semaine avec des boutons « Semaine suivante » et « Semaine précédente ».

Une fois les vérifications faites, le logiciel affiche les conflits si il y en a pour offrir le choix à l'utilisateur sur les événements à garder, c'est à dire un système de formulaire de type RadioButton ayant 2 choix par conflit, l'ancienne version ou la nouvelle.

Puis le logiciel demande confirmation avant de mettre à jour le calendrier Google Agenda.

#### 2.1.3 Interfaces matérielles

L'application doit être développée afin de pouvoir être exécutées sur des machines Windows, Linux et Mac.

Mise à jour d'un calendrier en ligne	Version : <1.2>
Spécification d'exigences logicielles	Date : 07/11/12

#### 2.1.4 Interfaces de communication

Caldav est le protocole de communication entre le logiciel et le calendrier Google Agenda.

La librairie Java CalDAV4j sera utilisée afin d'ajouter et de supprimer des éléments dans les calendriers.

#### 2.1.5 Contraintes de mémoire

La mémoire de la machine virtuelle (utilisation de Java) devra être prise en compte pour être capable de supporter la création de tous les événements. Il n'y a aucune contrainte supplémentaire car le logiciel n'est pas amené à être exécuté sur des systèmes embarqués.

### 2.2 Fonctions du produit

La fonction du logiciel est de mettre à jour ses cours sur un calendrier en ligne (Google Agenda), et de gérer les conflits entre les éléments de la nouvelle version à importer et l'ancienne version.

### 2.3 Caractéristiques des utilisateurs

Les utilisateurs devront être des étudiants, ou toute personne ayant des emploi du temps de cours. Ces utilisateurs n'ayant pas forcément nécessairement des compétences avancées en informatique, le logiciel se doit être intuitif.

### 2.4 Contraintes

Ce projet subit une contrainte temporelle, exigeant une version finale du produit pour le 5 décembre 2012.

### 2.5 Hypothèses et dépendances

Le logiciel se connecte à un calendrier Google Agenda via Caldav. L'utilisateur devra impérativement avoir un accès internet pour pouvoir se connecter à son calendrier en ligne.

### 2.6 Exigences reportées

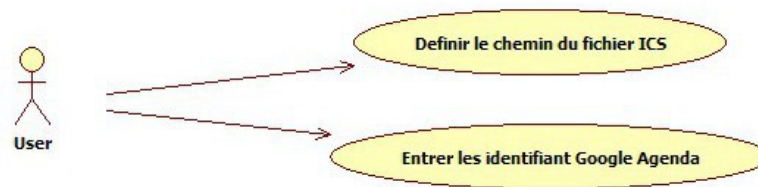
Dans les versions futures on peut supposer que l'utilisateur puisse choisir entre différents services de calendriers en ligne utilisant le protocole caldav, ou encore de stocker sous un autre format l'emploi du temps sur sa propre machine.

## 3. Exigences spécifiques

### 3.1 Spécification des cas d'utilisation

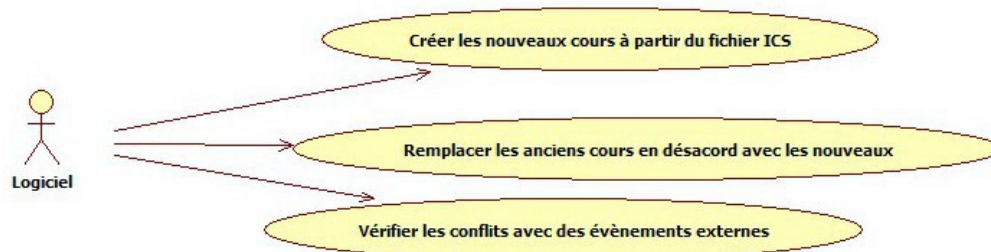
- L'utilisateur lance le logiciel :

Mise à jour d'un calendrier en ligne	Version : <1.2>
Spécification d'exigences logicielles	Date : 07/11/12



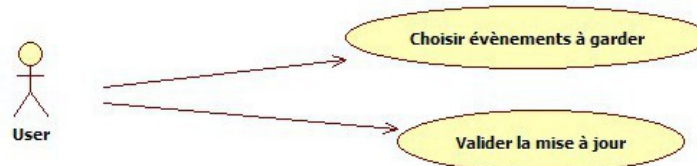
L'utilisateur doit spécifier le chemin du fichier ICS ainsi que ses identifiant Google Agenda.

- Le logiciel analyse les données :



Le logiciel créer les nouveaux événements à partir du fichier ICS donné en entrée. Il va modifier les anciens cours qui ont été remplacés par de nouveaux. Si un nouveau cours est à cheval sur un ou plusieurs événements externes, on avertira l'utilisateur.

- Validation de la mise à jour :



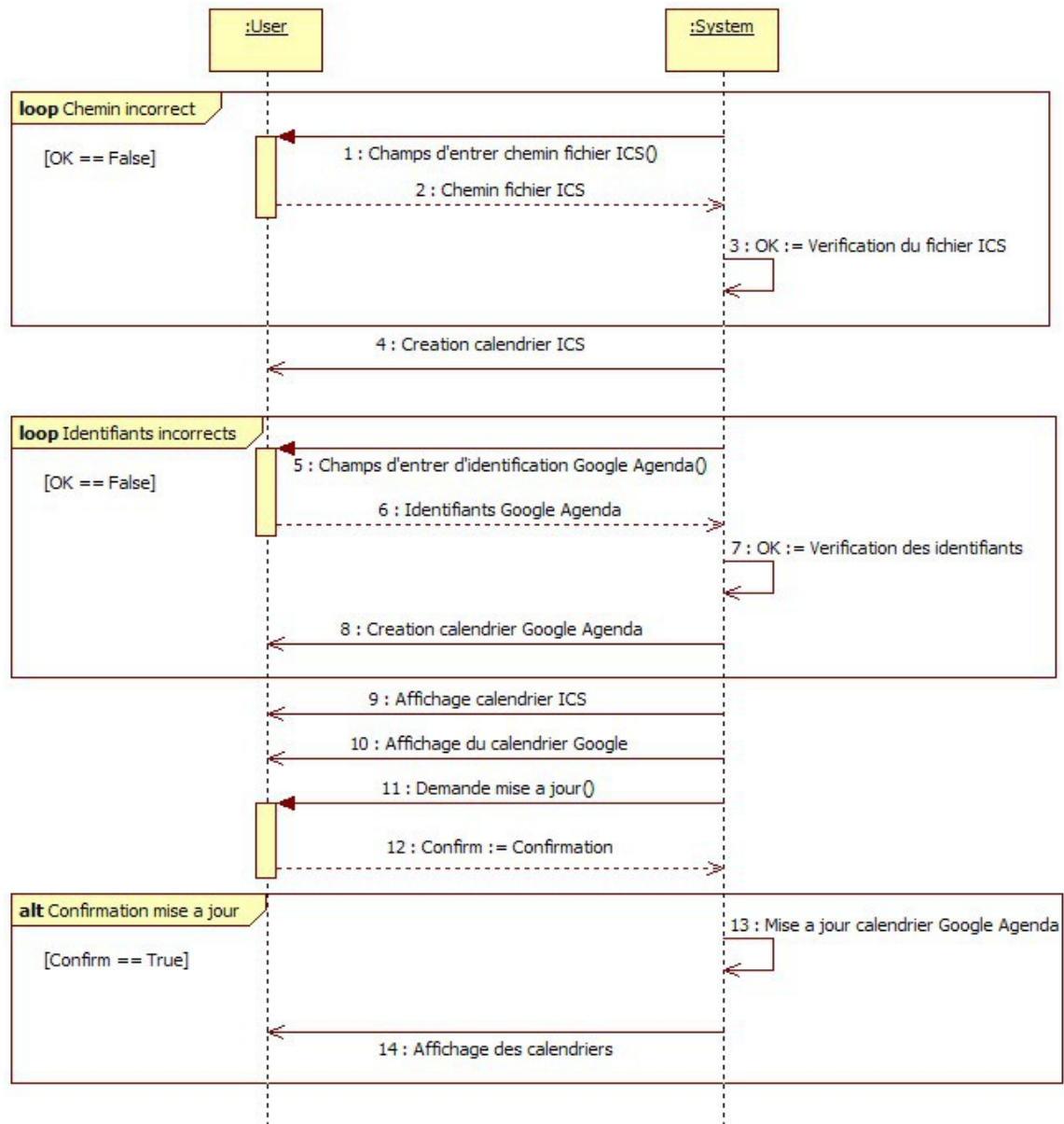
L'utilisateur devra valider la mise à jour en ligne du calendrier Google Agenda. Si il y avait des conflits avec des événements extérieurs, l'utilisateur devra au préalable choisir les événements à garder.

Mise à jour d'un calendrier en ligne	Version : <1.2>
Spécification d'exigences logicielles	Date : 07/11/12

### Use case #1 :

Le calendrier Google Agenda ne contient pas de cours :

- L'utilisateur entre le chemin du fichier ICS et ses identifiant Google Agenda
- Le logiciel copie le fichier ICS et créer les événements
- Le logiciel demande confirmation à l'utilisateur pour enregistrer les données en ligne
- Le logiciel ajoute les événements sur le calendrier en ligne



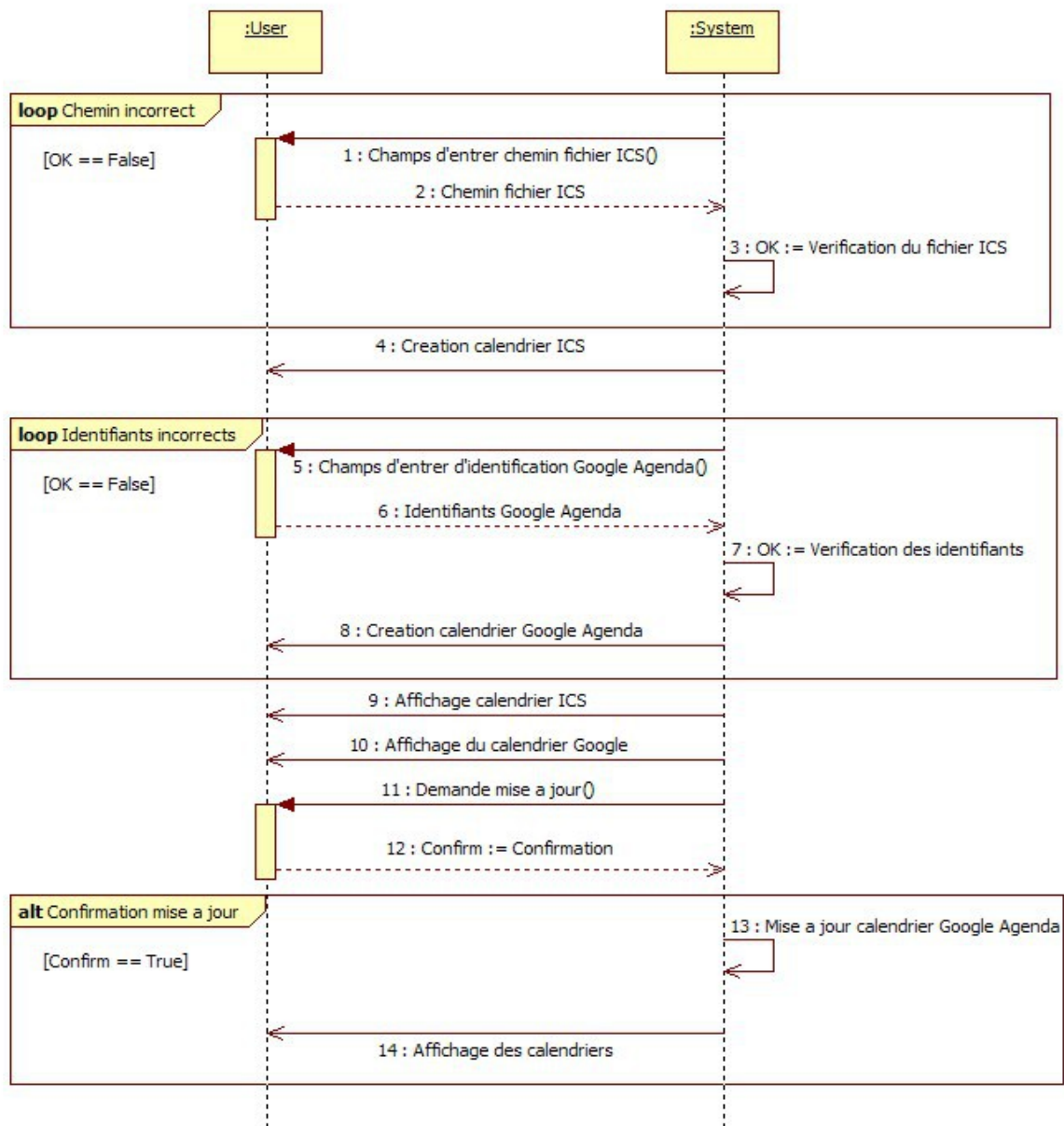


Mise à jour d'un calendrier en ligne	Version : <1.2>
Spécification d'exigences logicielles	Date : 07/11/12

### Use case # 2 :

Le calendrier ne contient des cours, le fichier ICS ne présente aucun conflit avec des événements extérieurs :

- L'utilisateur entre le chemin du fichier ICS et ses identifiant Google Agenda
- Le logiciel copie le fichier ICS et créer les événements
- Le logiciel demande confirmation à l'utilisateur pour enregistrer les données en ligne
- Le logiciel remplace les anciens cours par les nouveaux si ils sont différents

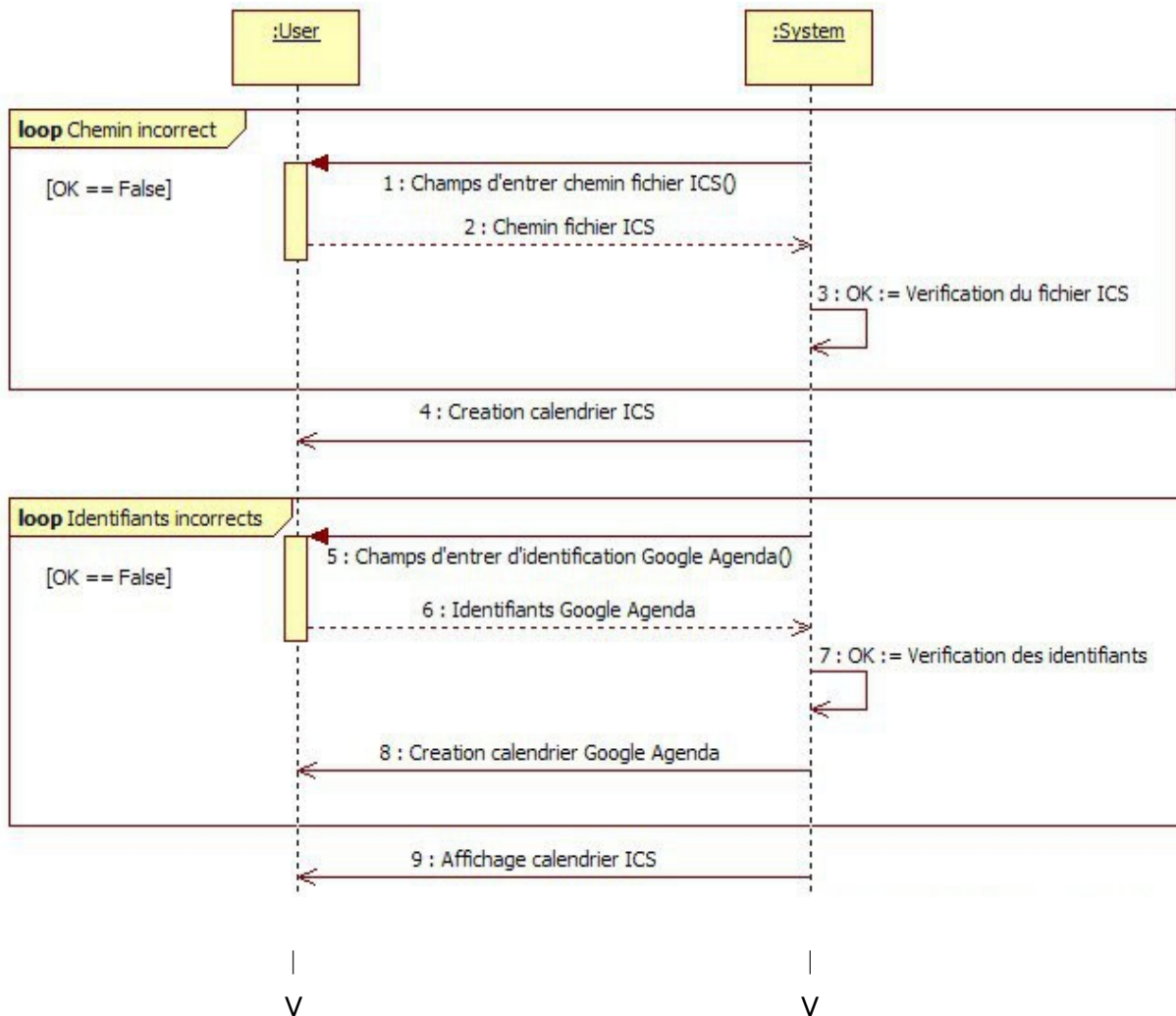


Mise à jour d'un calendrier en ligne	Version : <1.2>
Spécification d'exigences logicielles	Date : 07/11/12

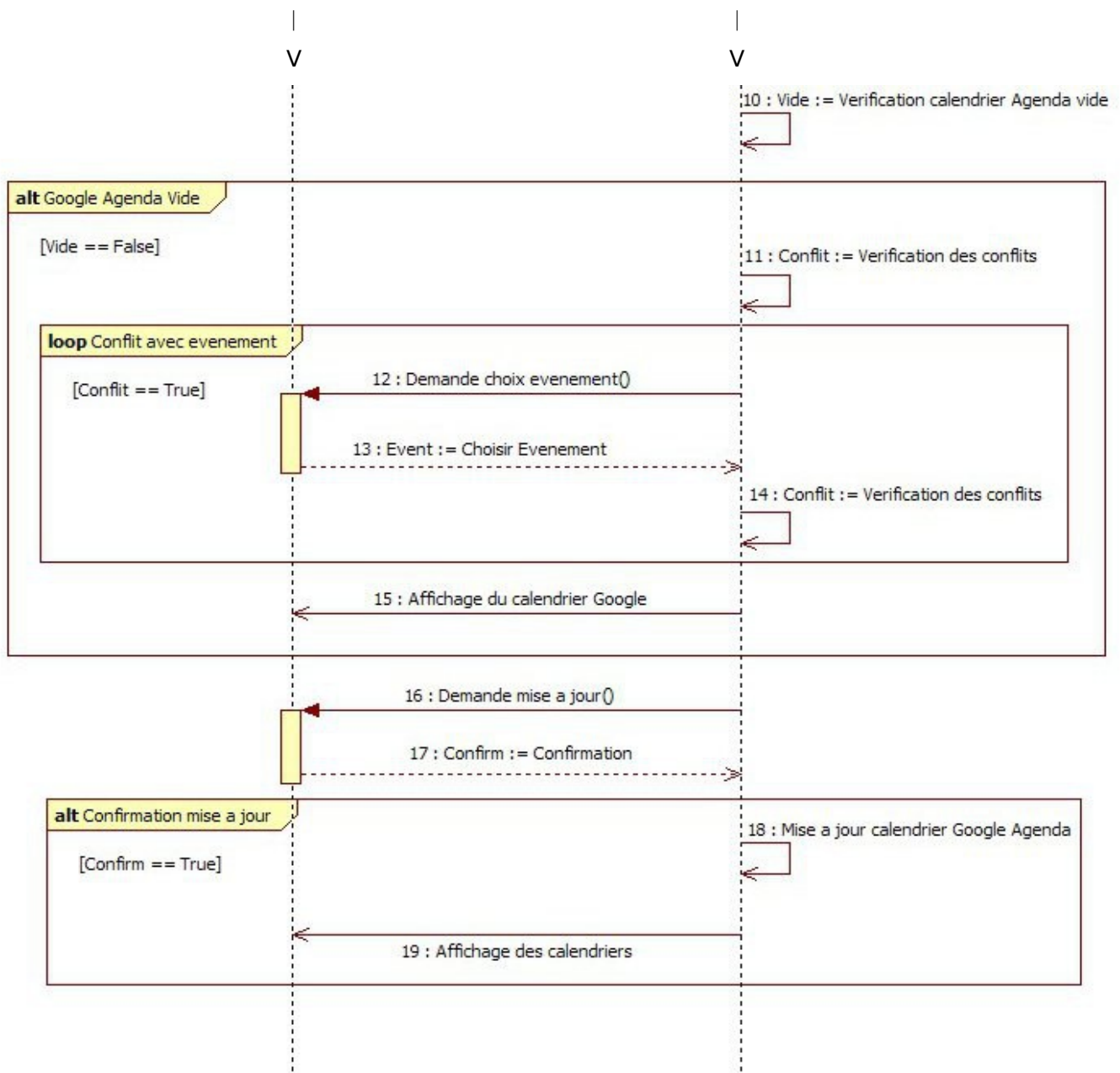
### Use case #3 :

Le calendrier ne contient des cours, le fichier ICS présente des conflits avec des événements extérieurs :

- L'utilisateur entre le chemin du fichier ICS et ses identifiant Google Agenda
- Le logiciel copie le fichier ICS et créer les événements
- Le logiciel présente les possibilités entre les anciens événements et le nouveaux cours
- Le logiciel demande confirmation à l'utilisateur pour enregistrer les données en ligne
- Le logiciel remplace les anciens cours par les nouveaux si ils sont différents
- Le logiciel remplace les anciens événements par les nouveaux cours si l'utilisateur l'y a autorisé



Mise à jour d'un calendrier en ligne	Version : <1.2>
Spécification d'exigences logicielles	Date : 07/11/12



Mise à jour d'un calendrier en ligne	Version : <1.2>
Spécification d'exigences logicielles	Date : 07/11/12

## **3.2 Exigences supplémentaires**

### **3.2.1 Utilisabilité**

L'utilisateur ne doit avoir besoin d'aucune compétence particulière pour pouvoir se servir du logiciel. Le logiciel demandera lors de la connexion si l'utilisateur souhaite enregistrer ses identifiants, si c'est le cas, le logiciel sera capable de se remplir automatiquement les champs d'identification au calendrier Google Agenda.

Il demande avec simplicité les identifiants de l'utilisateur, et d'aller chercher son fichier ICS à importer ou son URL, tout comme il présente distinctement chaque conflit qu'il peut y avoir entre les deux versions d'emploi du temps.

Chacune de ces tâches ne prennent que quelques secondes :

- Saisies des identifiants
- Saisie de l'URL ou recherche du chemin où se trouve le fichier ICS
- Validation de la synchronisation
- Validation des conflits entre les versions de l'emploi du temps

### **3.2.2 Fiabilité**

L'application est effective tant que Google Calendar l'est lui aussi. La probabilité que les serveurs de Google s'éteignent est quasiment nulle; mais dans le cas contraire, la connexion ne pourra avoir lieu et le logiciel ne pourra rien synchroniser. Il faudra alors attendre que les serveurs Google soient de nouveau en marche mais que l'application soit effective de nouveau elle aussi.

### **3.2.3 Performance**

Seule une connexion Internet est requise.

### **3.2.4 Maintenabilité**

Le code devra être sera doté d'une architecture MVC (Modèle Vue Contrôleur) afin de faciliter la compréhension et la maintenabilité du logiciel.

## **4. Contraintes de conception**

Le logiciel devra être codé en Java. L'interface homme machine devra utiliser la librairie standard SWING.

## **5. Sécurité**

Les identifiants Google Agenda devront être protégés. (pas d'affichage directe, pas d'enregistrement brute des données). Si l'utilisateur choisit d'enregistrer ses identifiants, il faudra donc crypter ces données.

## **6. Exigences de documentation utilisateur et d'aide en ligne**

La documentation devra décrire le fonctionnement du logiciel en exprimant les cas d'utilisations décrits dans ce document, ainsi que la définition des objets et protocoles utilisés dans ce logiciel.

## **7. Normes applicables**

Norme [IEEE830-1998] : Recommended Practice for Software Requirements Specifications