Étudiant 1 : Etienne JACQUOT

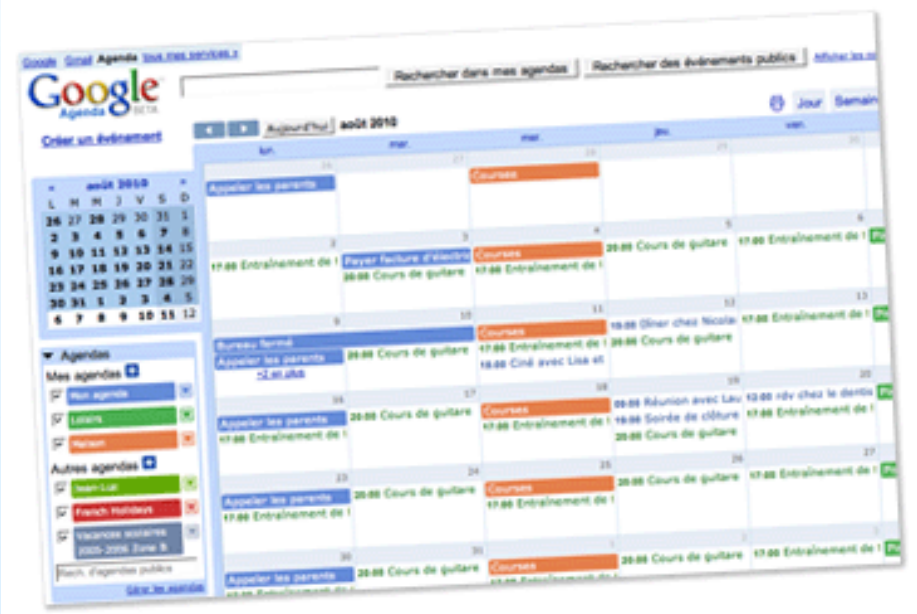
Étudiant 2 : Minh Tim LE

Étudiant 3 : Hamza OBEIDAT
(responsable de groupe)

Étudiant 4 : Jérémie RAMPIN

Semestre : Automne 2014

GL02 : Plan et bilan de tests de recette



Version : 1.0
Date : 11/01/2015

www.utt.fr
Université de technologie de Troyes • 12 rue Marie Curie • BP 2060 • 10010 Troyes cedex

Sommaire

1. Contexte

1.1. Objectifs des tests

Les tests présentés dans ce document doivent permettre de vérifier la conformité de la librairie livrée avec les attentes du cahier des charges. Il s'agira donc, particulier, de :

- Valider la prise en charge des fichiers CSV créés par le logiciel de l'association AMDY
- Valider la compatibilité du format iCalendar généré par la bibliothèque JavaScript
- Vérifier l'affichage d'erreurs en cas d'échecs d'opérations
- Vérifier que les rapports hebdomadaires sont correctement formatés
- Vérifier que les unions, intersections et complémentaires de plannings sont bien pris en charge
- Vérifier que l'ensemble des fonctionnalités demandées dans la cahier des charges a bien été mis en place.

1.2. Environnement de test

Les tests seront réalisés sous le système d'exploitation Microsoft Windows (XP, 7, 8 et 8.1), à l'aide de NodeJS en version 0.10.33 (version en date du 03/12/2014).

1.3. Résultats de tests

Les résultats de tests seront enregistrés sur l'issue tracker de google code pour remonter les anomalies (Type-Defect) ou demande d'intervention (Type-Enhancement) ainsi que dans le présent test de recette, à la section "Bilan de test".

2. Scénarios et cas de tests

2.1. Cas de tests (fonctionnalités à tester)

Identifiant	CDT_1
Titre	Création plannings intervenants
Description	L'utilisateur donne, en entrée de la bibliothèque JavaScript, les trois fichiers CSV de test, ainsi que son choix d'obtenir les plannings pour les intervenants. La bibliothèque JavaScript devra alors retourner les plannings intervenant au format iCalendar.
Jeu de données	Planning_test1.csv, Planning_test2.csv, Planning_test3.csv
Critères de validation	✓ La structure des fichiers iCalendar devra être conforme à la structure donnée dans le cahier des charges (§2.3.1) ✓ Les fichiers iCalendar devront être importables sur un Google Calendar ou Outlook. ✓ Les fichiers iCalendar ne devront pas avoir perdu de données

Identifiant	CDT_2
Titre	Création plannings domicile
Description	L'utilisateur donne, en entrée de la bibliothèque JavaScript, les trois fichiers CSV de test, ainsi que son choix d'obtenir les plannings pour les domiciles. La bibliothèque javascript devra alors retourner les plannings domicile au format iCalendar.
Jeu de données	Planning_test1.csv, Planning_test2.csv, Planning_test3.csv

Critères de validation	✓ La structure des fichiers iCalendar de vra être conforme à la structure donnée dans le cahier des charges (§2.3.1) ✓ Les fichiers iCalendar devront être importables sur un Google Calendar ou Outlook. ✓ Les fichiers iCalendar ne devront pas avoir perdu de données

Identifiant	CDT_3
Titre	Création rapport d'intervention
Description	L'utilisateur donne, en entrée de la bibliothèque JavaScript, les trois fichiers CSV (ou iCal) de test. La bibliothèque javascript devra alors renvoyer le rapport d'interventions au format demandé dans le cahier des charges (§1.2, SPEC_1), à savoir le lom de l'intervenant et pour chacun le nombre d'interventions et le volume horaire.
Jeu de données	Planning_test1.csv, Planning_test2.csv, Planning_test3.csv Planning_test1.ics, Planning_test2.ics, Planning_test3.ics
Critères de validation	✓ Le planning renvoie l'ensemble des données demandées. ✓ Pas de perte de données. ✓ En cas de créneaux consécutifs, le rapport ne devra compter qu'une intervention.

Identifiant	CDT_4
Titre	Création d'une union de plannings
Description	L'utilisateur donne les trois fichiers CSV ou ics en entrée, ainsi que son choix d'obtenir l'union des plannings. La bibliothèque devra renvoyer un fichier iCalendar contenant les informations demandées.

Jeu de données	Planning_test1.ics, Planning_test2.ics, Planning_test3.ics Planning_test1.csv, Planning_test2.csv, Planning_test3.cvs
Critères de validation	✓ Le fichier créé affiche bien l'ensemble des informations, y compris les créneaux où il y a plusieurs intervenants. ✓ La structure des fichiers iCalendar devra être conforme à la structure donnée dans le cahier des charges (§2.3.1) ✓ Les fichiers iCalendar devront être importables sur un Google Calendar ou sur Outlook.

Identifiant	CDT_5
Titre	Création d'une intersection de plannings
Description	L'utilisateur donne les trois fichiers CSV ou ics en entrée, ainsi que son choix d'obtenir l'union des plannings. La bibliothèque devra renvoyer un fichier iCalendar contenant les informations demandées.
Jeu de données	Planning_test1.ics, Planning_test2.ics, Planning_test3.ics Planning_test1.csv, Planning_test2.csv, Planning_test3.cvs
Critères de validation	✓ Le fichier créé affiche bien l'ensemble des informations, y compris les créneaux où il y a plusieurs intervenants. ✓ La structure des fichiers iCalendar devra être conforme à la structure donnée dans le cahier des charges (§2.3.1) ✓ Les fichiers iCalendar devront être importables sur un Google Calendar ou sur Outlook.

Identifiant	CDT_6
Titre	Création d'un complémentaire de plannings

Description	L'utilisateur donne les trois fichiers CSV en entrée, ainsi que son choix d'obtenir l'union des plannings. La bibliothèque devra renvoyer un fichier iCalendar contenant les informations demandées.
Jeu de données	Planning_test1.ics, Planning_test2.ics, Planning_test3.ics Planning_test1.csv, Planning_test2.csv, Planning_test3.cvs
Critères de validation	✓ Le fichier créé affiche bien l'ensemble des informations, y compris les créneaux où il y a plusieurs intervenants. ✓ La structure des fichiers iCalendar devra être conforme à la structure donnée dans le cahier des charges (§2.3.1) ✓ Les fichiers iCalendar devront être importables sur un Google Calendar ou sur Outlook.

Identifiant	CDT_7
Titre	Erreurs dues à un CSV aberrant
Description	Il peut arriver que le fichier CSV donné en entrée ne soit pas valide ou corrompu, à cause par exemple d'un bug du logiciel de l'association ou s'il a été modifié par un tiers. Ici nous vérifions qu'un CSV volontairement modifié de façon aberrante (jours dans le désordre, intervenants non complets, manque de créneaux) est refusé par la bibliothèque JavaScript.
Jeu de données	Planning_error.csv
Critères de validation	✓ Le programme devra comprendre que le fichier donné en entrée n'est pas valide et en informer l'utilisateur ✓ Le programme ne devra pas générer de fichiers "déchets" après la fin de l'erreur

Identifiant	CDT_8

Titre	Création d'un complémentaire de planning plein
Description	Il s'agit de vérifier la gestion d'un planning dont tous les créneaux sont pleins. Ce cas de figure pourrait en effet arriver en cas de gestions des conflits dans le cas d'un grand nombre de plannings. Ici, on simule ce comportement avec un seul planning CSV dont tous les créneaux sont remplis. L'utilisateur demande la création d'un planning de conflits avec l'option "complémentaire".
Jeu de données	planning_full.ics
Critères de validation	✓ Le programme doit avertir l'utilisateur qu'il est impossible de généré un complémentaire à ce planning car tous les créneaux sont utilisés. ✓ La fonction ne devra pas endommager le CSV ni créer des fichiers "déchet"

2.2. Scénarios de tests

Scénario 1 :

Déroulement : Le gestionnaire d'AMDY souhaite éditer un rapport des interventions réalisées sur la semaine passée. Il a en sa possession un ensemble de plannings au format CSV ou iCalendar, représentant les interventions d'une semaine pour un domicile. Il donne donc, à la librairie, l'ensemble de ces fichiers et attend, en sortie, un rapport au format CSV avec le nom de l'intervenant, le nombre d'interventions et le volume horaire.

Cas de test concernés : CDT_3, CDT_7

Scénario 2 :

Déroulement : Le gestionnaire d'AMDY a réalisé les plannings d'intervention pour la semaine sur le logiciel interne à l'association. Il souhaite désormais éditer les plannings au format iCalendar avant leur diffusion à ses clients. Tout d'abord, il donne l'ensemble des CSV et demande l'édition des plannings hebdomadaires pour chaque domicile. Il récupère les fichiers iCalendar de chaque domicile. Il redonne à la bibliothèque les fichiers CSV et demande, cette fois, l'édition des iCalendar de chaque intervenant. Il récupère ces fichiers. Il

a désormais en sa possession, l'ensemble des iCalendar nécessaire à la diffusion des plannings à ses clients et à ses fournisseurs de services.

Cas de test concernés : CDT_1, CDT_2, CDT_7

Scénario 3 :

Déroulement : Le gestionnaire d'AMDY souhaite comparer des plannings et savoir s'il y a des conflits. Il s'agit par exemple, de vérifier que des plannings envoyé par des fournisseurs de services, ne sont pas en conflits avec les interventions planifiées par AMDY. Le gestionnaire demande donc à accéder à la gestion des conflits. Il va vérifier si les trois iCalendar en sa possession n'ont pas d'intersection commune. Puis il vérifiera que l'union de ces trois plannings ne génère pas de surcharge de travail pour ces prestataires. Il récupère, de la bibliothèque, un fichier au format iCalendar avec les données demandées.

Cas de test concernés : CDT_4, CDT_5

Scénario 4 :

Déroulement : Le gestionnaire d'AMDY souhaite avoir l'ensemble des créneaux disponibles d'un de ces prestataires. Celui-ci lui a donné un fichier iCalendar reprenant son planning de la semaine suivante, avant qu'AMDY lui ait donné ses interventions. Le gestionnaire souhaite donc le complémentaire du iCalendar donné par son prestataire. Il va donc demander à la bibliothèque l'outils « Complémentaire de planning » disponible dans la gestion des conflits. Il donne à la bibliothèque, le iCalendar en sa possession et récupère le complémentaire de ce planning au format iCalendar.

Cas de test concernés : CDT_6, CDT_8

3. Bilan de test

3.1. Erreurs globales

Le format de CSV pris en charge n'est pas bon. L'entête des fichiers CSV en possession de l'utilisateur se présente sous la forme : "Lundi;Mardi;Mercredi;Jeudi;Vendredi;Samedi;Dimanche" et les développeurs ont en entête "20141201;20141202;20141203;20141204;20141205;20141206;20141207;"

En effet, à aucun moment dans le cahier des charges, l'élément de format du fichier CSV n'a été mentionné. Hors toutes les fonctions découlent de l'exploitation de ce format. Aussi, pour mener à bien les tests, nous avons utilisé le format de l'équipe de développement avec un entête comportant les dates.

3.2. Erreurs fonctionnelles remontées par les cas de test

CDT_7 : L'envoi d'un fichier aberrant à la bibliothèque ne retourne aucune erreur. Elle crée tout de même le fichier ics en remplissant avec les données obtenues, même si il n'y a rien, ou s'il y a deux fois la même date/jour.

CDT_1, CDT_2 : Pas de possibilité de choisir si on veut un iCal pour un intervenant ou pour un planning. La SPEC_1 du cahier des charges, mentionnant ce détail dans le scénario, n'est pas respectée.

CDT_3 : Création du rapport d'intervention fonctionnel pour 1 fichier ical. Il faudrait donc, pour générer un planning de plusieurs domiciles (c'est à dire une semaine d'exploitation pour l'association), fusionner tous les plannings d'intervention domicile avant de générer un rapport d'intervention. La SPEC_2 mentionnait pourtant une conversion de la totalité des CSV ou iCal d'une semaine pour la génération d'un rapport l'intervention globale. Le cahier des charges n'est donc pas totalement respecté. Il est en revanche possible de reprendre assez facilement la solution développée.

CDT_4 : Union fonctionnelle seulement entre deux fichiers ics mais pas pour des fichiers CSV. De plus, on ne peut envoyer, au plus, que 2 fichiers à la bibliothèque. Le cahier des charges (SPEC_3 et 3.1) n'est pas respecté totalement.

CDT_ 5 : Intersection fonctionnelle seulement entre deux fichiers ics mais pas pour des fichiers CSV. De plus, on ne peut envoyer, que plus, que 2 fichiers à la bibliothèque. Le cahier des charges (SPEC_3 et 3.2) n'est pas respecté totalement.

CDT_6 : Complémentaire fonctionnel seulement entre deux fichiers ics mais pas pour des fichiers CSV. De plus, on ne peut envoyer, que plus, que 2 fichiers à la bibliothèque Le cahier des charges (SPEC_3 et 3.3) n'est pas respecté totalement.

CDT_8 : L'application ne crée pas d'erreur. Elle crée un fichier iCal ne contenant aucun créneau. La SPEC_3.3 n'est pas respectée par rapport au scénario alternatif.

Conclusion

Au terme de ce plan de test, il apparait que le cahier des charges n'a pas été pris en compte pour un certain nombre de points. La plupart du temps, il s'agit d'éléments décrits dans les scénarios des spécifications. Il aurait donc fallu trouver un moyen de mettre en évidence les éléments clés du programme à prendre en compte, et mettre en valeur les détails de ces spécifications (formats pris en compte, nombre de fichiers en entrées, détails sur l'interface homme-machine, …). Il apparait tout de même un certain nombre d'erreurs dans le cahier des charges :

- Le format de CSV à prendre ne compte en entrée n'est pas clairement spécifié.
- Le nombre de fichiers CSV ou iCal à prendre en entrée pour les outils « Intersection », « Union », « Complémentaire », n'est pas défini spécifiquement.
- Les scénarios d'utilisation côté utilisateur ne sont pas définis, décris spécifiquement.

Au terme de ce projet, il apparait que l'utilisation, dans le cahier des charges, de spécifications normées, ne laissant pas de place à l'interprétation, est essentielle. Les spécifications de tous les formats pris en charge doit être donné, tous les traitements doivent être spécifiés, et la structure du programme désiré doit être claire. Pour que le développeur soit capable de prendre en compte la totalité de ces informations, il doit avoir la possibilité d'avoir une vision globale du livrable attendu, comme par exemple par l'intermédiaire de diagrammes UML (cas d'utilisation, diagramme de classes, …).