

Java: Cartes et Calendrier

Sommaire

Sommaire	2
Introduction	3
Evaluation des besoins	3
Scénario	3
Modélisation UML	4
Représentation des Classes utilisée	5
Vcard	5
Nomenclature d'une Vcard	5
Calendrier	5
Nomenclature d'un fichier iCalendar	5
Autres classes	6
Explication de la solution	7
Avancement du Projet et Amélioration	8
Bug Rencontrés non résolu	8
Point d'avancement du projet	8
Idée d'amélioration future	9
Conclusion	9
Bibliographie	9
Annexes	11
Annexe 1 : Diagramme de Gantt	11
Annexe 2: Diagramme de classe UML	12

I. Introduction

Le projet “Java: Cartes et Calendrier” est un projet où l’objectif est de créer une application, à la fois en console et graphique qui permettra de générer et manipuler des cartes de visite ainsi que des événements.

Pour ce faire nous avons décidé de découper le projet en trois parties.

Tout d’abord nous travaillerons uniquement en mode console et travaillerons sur les cartes de visite pour le mois janvier et début février.

Puis nous étudierons la gestion des événement et du Calendrier dans le mois de février

Enfin nous travaillerons sur l’interface Graphique. dans le mois de mars

Le projet a débuté le 8 janvier et finira le 23 mars. Chaque membre sera a moment donnée chef de projet. Ainsi :

Ulysse Perret sera le premier chef de projet pour le mois de Janvier

Amadou Ly sera chef de projet le mois de Février

Nikola Marinkovic sera chef de projet le mois de Mars

Nous prévoyons de faire une check hebdomadaire tous les jeudis.

Nous avons établis un Diagramme de Gantt que vous retrouverez en Annexe 1

II. Evaluation des besoins

A. Scénario

Pour nous aider à nous représenter le projet et nous fixer un scénario de cas où l’application pourrait nous servir.

Contexte du Scénario 1 :

Un chef d’entreprise veut organiser un entretien d’embauche pour rencontrer un nouveau candidat. L’entretien se déroulera au sein de son entreprise et seront présent le candidat, la RH et lui-même

Voici les grandes étapes que nous avons établis :

Etape 1: Création et Modification de la carte candidat

Etape 2 : Création de l'événement

Etape 3 : Invitation des participants

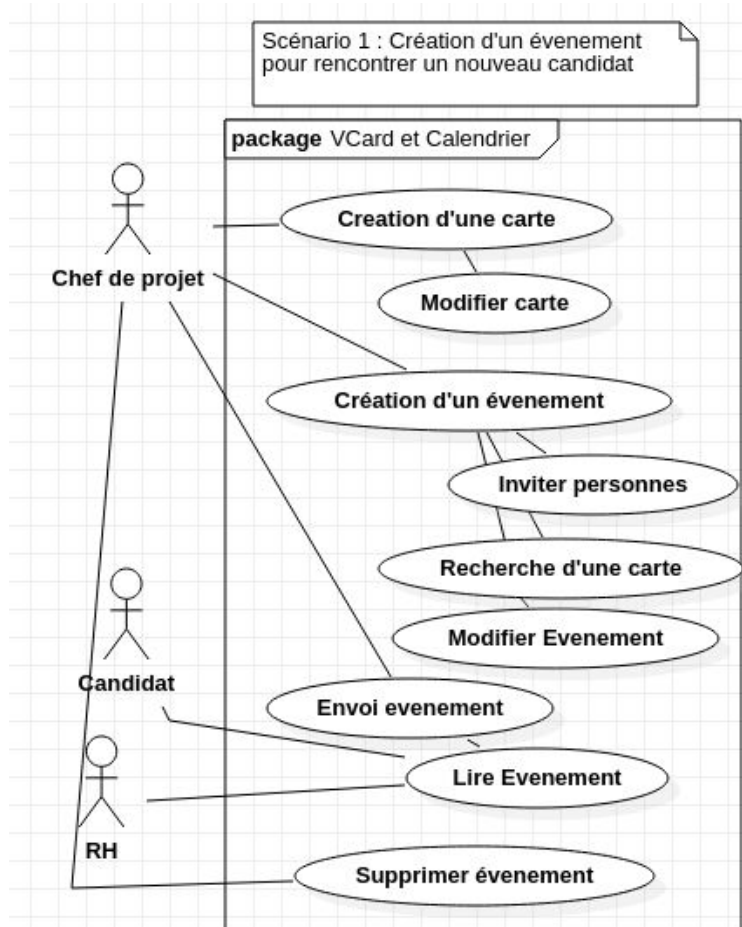
Etape 4 : Modification et finalisation de l'événement

Etape 5 : Envoi de l'événement aux intéressés

Étape 6 : Suppression de l'événement une fois l'évènement passé

B. Modélisation UML

Voici le diagramme d'utilisation pour notre Scénario



Capture d'écran issu de StarUML version 3, document personnel

Nous voyons que les actions important sont :

- Création d'une carte
 - Modifier carte
- Création d'un événement
 - Inviter personnes , pour indiquer et inviter les personne à l'événement
 - Recherche d'une carte, pour recherche avec le nom d'une carte à inclure
 - Modifier événement
- Envoi événement
 - Lire Evenement. il faut en effet que les personne qui reçoivent l'événement puisse consulter celui-ci
- Supprimer événement

Le diagramme des classes complet se trouve en Annexe 2

II. Représentation des Classes utilisée

A. Vcard

1. Nomenclature d'une Vcard

Un fichier vCard est comme une carte de visite. il est sous format de carnet d'adresse. Le fichier vCard porte une extension vcf mais peut être lu par n'importe quel éditeur de texte. Cependant il doit respecter en plus de son extension une certaine nomenclature. Notamment il doit commencer par BEGIN, finir par END, et à chaque ligne a le nom de la propriété, suivi par sa valeur comme suit : PROPRIÉTÉ : valeur propriété

Exemple de carte , Jean_Dupont.vcf

BEGIN:VCARD

VERSION:2.1

FN:Jean Dupont

N:Dupont;Jean

ADR;WORK;PREF;QUOTED-PRINTABLE;;Bruxelles 1200=Belgique;6A Rue Th. Decuyper

LABEL;QUOTED-PRINTABLE;WORK;PREF:Rue Th. Decuyper 6A=Bruxelles

1200=Belgique

TEL;CELL:+1234 56789

EMAIL;INTERNET:jean.dupont@example.com

UID:

END:VCARD

Exemple pris sur : <https://fr.wikipedia.org/wiki/VCard>

2. La classe vCard

La Classe Vcard permet de Lire, Afficher, Modifier une carte à l'heure actuelle

B. Calendrier

1. Nomenclature d'un fichier iCalendar

Un fichier iCalendar permet l'échange et l'ajout d'événement dans le calendrier utilisateur, comme celui de google.

Il est en format .ics.

Voici un exemple de fichier .ics

BEGIN:VCALENDAR

VERSION:2.0

PRODID:-//hacksw/handcal//NONSGML v1.0//EN

BEGIN:VEVENT

DTSTART:19970714T170000Z

DTEND:19970715T035900Z

SUMMARY:Fête à la Bastille

END:VEVENT

END:VCALENDAR

exemple pris sur : <https://fr.wikipedia.org/wiki/ICalendar>

2. La classe Calendrier

La Classe Calendrier n'as pas encore été utilisé.

C. Autres classes

Dans notre diagramme UML Nous avons définis d'autres classes autre que celles de VCards et Calendrier :

Événement :

Classe qui permet de modifier l'événement contenu dans le calendrier

Flipcard :

Classe qui permet de stocker plusieurs VCard

III. Explication de la solution

Si vous avez lu notre Readme, vous êtes arrivé à notre Main.

Celui-ci contient et appelle le constructeur Calendrier() et donc la classe Calendrier.java.

Sur celui-ci, celui ci appelle la une fonction Menu, qui affichera une liste de choix que vous trouvez sur la capture d'écran.

Capture D'écran 1

```
Projet Agenda et Vcard
Bienvenu sur notre programme de gestion de carte de l'agenda
Que voulez vous faire?
1.Travaille sur les Vcards
2.Travaille sur l'Agenda
3.Quitter L'appli
Quel est votre Choix? 1.Vcard 2.Agenda 3.Quitter
```

Nous procédons par menu dès que possible et demandons à l'utilisateur de rentrer un chiffre pour faire son choix.

Nous faisons ainsi aussi pour le menu des Vcards, ou des fonctions annexes.

Il a noté que notre code contient beaucoup de méthodes de vérifications tel que CheckChaineVide, fonction qui permet de voir si on a rentrer une chaîne null ou vide. Celle-ci permet notamment de faire boucle répétitive, demandant d'entrer une valeur à l'utilisateur.

Nous avons aussi dû gérer la problématique des adresses, et notamment le fait qu'une personne pouvait avoir une ou des adresses.

Pour la modification, nous avons stocker les valeurs originel dans la classe en variable, pour ainsi pouvoir les ressortir et les modifier après.

Pour la fonction Gestion de fichier, à l'heure actuel nous n'avons fait que la transformation en html et la suppression du fichier.

Nous avons aussi gérer les problèmes des exceptions, notamment pour les problèmes de fichiers, répertoires.

IV. Avancement du Projet et Amélioration

1. Bug Rencontrés non résolu

Nous avons rencontré un bug problématique, qui fait qu'on n'arrive pas à lire et à accéder aux chemins des fichiers en mode console.

De ce fait, la majorité des fonctionnalités proposés par le programme ne fonctionne pas via ce mode. Nous avons tenté plusieurs pistes de solution, aucune semble correspondre à notre besoin.

Le bug est présent car en IDE nous pouvons travailler en locale et en chemin relatif alors que la console préfère les chemins absolus. Notre bug n'as pas de souci sur nos machine car nous travaillons sur la même arborescence de dossier et avons mis en place les outils pour travailleurs sur nos IDE préféré, Intelj IDE ou Eclipse, que nous pouvons lancer notamment grâce à notre Readme.

Ce bug est gênant car il ne permet pas de livrer le produit en fonctionnement console comme souhaité, et reste bloquant pour la suite du projet, notamment pour les interfaces graphique.

Pour palier à ce bug, nous avons préféré travailler uniquement en IDE et concentrer sur les fonctionnalités de celui-ci.

2. Point d'avancement du projet

Actuellement, nous estimons avoir livré un projet qui fonctionne pour la partie Vcard et récemment des fonctionnalité sur l'Agenda. Cependant, le programme fonctionne uniquement en IDE, via Eclipse ou Intelj IDE mais pas en console.

Pour la partie Cartes de visite nous avons traité que des points suivants :

- Lecture et Affichage
- Creation
- Modification
- Gestion de fichier :
 - Transformation en html
 - Suppression

Pour la partie Agenda nous avons traité que des points suivants:

- Lecture et Affichage
- Création

Nous avons passé tous le mois de Janvier et février pour avancer et finir la partie sur les cartes de visites.

Puis en mars nous somme passé sur la partie Calendrier et finaliser le tout.

Nous n'avons pas traiter les interfaces graphiques par manque de temps.

3. Idée d'amélioration future

A l'heure actuelle si nous devons prendre notre projet et l'améliorer ce serait avec de l'aide extérieur.

En effet, dans un premier temps nous essayerons et prendrons du temps pour résoudre notre bug avec ceux-ci.

Puis nous essayerons de mettre les fonctionnalités que nous n'avons pas pu mettre en place, tel que l'ajout de fichier, d'événement, l'invitation, l'export par mail, etc...

Une piste d'évolution qui a été abordée au commencement a été la lecture avec synthèse vocale. Nous l'avons de côté car nous voulons concentrer sur les fonctionnalités importantes.

V. Conclusion

Les points forts :

- Nous avons pu nous entendre et communiquer et travailler en groupe au même moment et endroit . Le projet a pu renforcer la capacité de communication.
- Nous avons pu mener et avancer sur le projets malgré les bugs, les problématiques rencontrés.
- Nous étions un groupe de travailleur, ce qui nous a permis de fournir du code conséquent
- La mise en commun et la reprise des documents ,notamment par les chef des projet n'as eu aucun souci.

Les points faibles

- Il a eu des tension mais qui ont résolu rapidement.
- La prise en compte du bug nous a ralenti sur le projet.
- La mise en commun a été délicate, nous avons du changer le fonctionnement de prise en compte des modifications apportées par chacun

VI. Bibliographie

Document personnels et cours donnée par l'université

Recherche :

<https://fr.wikipedia.org>

<https://www.alsacreations.com/>

Jeu de données

<https://icalendar.org/>

<https://www.w3.org/2002/12/cal/vcard-notes.html>

<http://microformats.org/wiki/h-event>

<http://microformats.org/wiki/h-card>

<https://support.office.com/en-us/article/Send-and-save-contacts-as-vCards-vcf-files-94A17A6F-105F-46C7-9308-33658C1C2690>

<https://support.mozilla.org/en-US/kb/how-use-virtual-card-vcard>

Validation

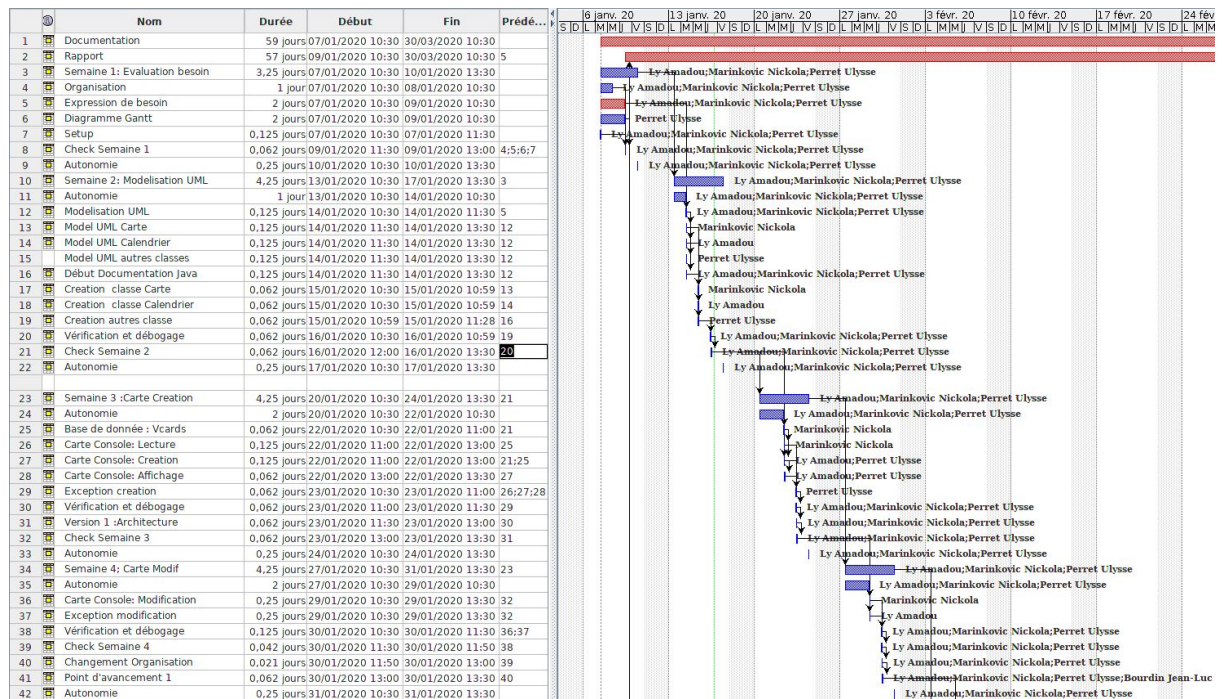
<https://icalendar.org/validator.html>

<https://validator.w3.org/>

VII. Annexes

A. Annexe 1 : Diagramme de Gantt

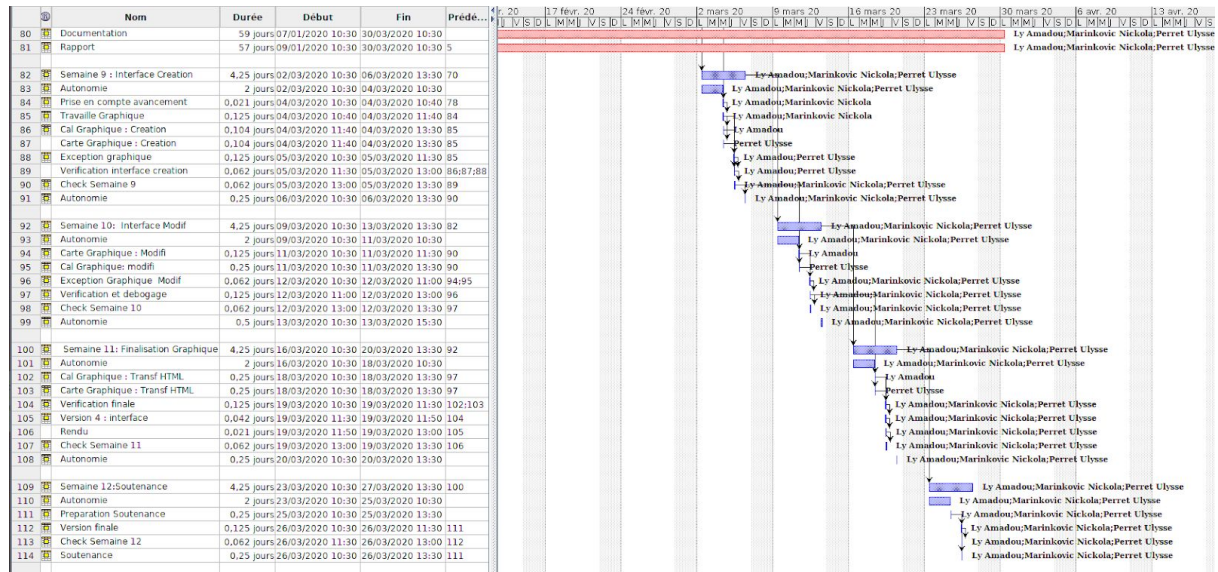
Document 1 : Semaine 1-4 : Du 7 au 31 Janvier



Document 2 : Semaine 5-8: Du 3 au 27 Février

	📅	Nom	Durée	Début	Fin	Préde...
43	📅	Documentation	59 jours	07/01/2020 10:30	30/03/2020 10:30	
44	📅	Rapport	57 jours	09/01/2020 10:30	30/03/2020 10:30	5
45	📅	Semaine 5: Vcard	4,25 jours	03/02/2020 10:30	07/02/2020 13:30	34
46	📅	Autonomie	2 jours	03/02/2020 10:30	05/02/2020 10:30	
47	📅	Prise compte Avancement	0,021 jours	05/02/2020 10:30	05/02/2020 10:40	41
48	📅	Carte console : Transf HTML	0,229 jours	05/02/2020 10:40	05/02/2020 13:30	47
49	📅	Exception Vcard	0,229 jours	05/02/2020 10:40	05/02/2020 13:30	47
50	📅	Vérification et débogage	0,125 jours	06/02/2020 11:30	06/02/2020 11:30	48:49
51	📅	Versionning 2 : Vcard	0,062 jours	06/02/2020 11:30	06/02/2020 13:00	50
52	📅	Check Semaine 5	0,062 jours	06/02/2020 13:00	06/02/2020 13:30	51
53	📅	Autonomie	0,25 jours	07/02/2020 10:30	07/02/2020 13:30	
54	📅	Semaine 6 :Calendar Creation	4,25 jours	10/02/2020 10:30	14/02/2020 13:30	45
55	📅	Autonomie	2 jours	10/02/2020 10:30	12/02/2020 10:30	
56	📅	Calendrier Console: Lecture	0,062 jours	12/02/2020 10:30	12/02/2020 11:00	52
57	📅	Calendrier Console: Affichage	0,062 jours	12/02/2020 11:00	12/02/2020 11:30	56
58	📅	Calendrier Console: Creation	0,125 jours	12/02/2020 10:30	12/02/2020 11:30	52
59	📅	Exception creations	0,125 jours	12/02/2020 11:30	12/02/2020 13:30	56:57:58
60	📅	Vérification et débogage	0,125 jours	13/02/2020 10:30	13/02/2020 11:30	56:57:58
61	📅	Check Semaine 6	0,062 jours	13/02/2020 13:00	13/02/2020 13:29	60
62	📅	Autonomie	0,25 jours	14/02/2020 10:30	14/02/2020 13:30	
63	📅	Semaine 7: Calendar Modif	4,25 jours	17/02/2020 10:30	21/02/2020 13:30	54
64	📅	Autonomie	2 jours	17/02/2020 10:30	19/02/2020 10:30	
65	📅	Carte Console: Modification	0,25 jours	19/02/2020 10:30	19/02/2020 13:30	61
66	📅	Exception modification	0,25 jours	19/02/2020 10:30	19/02/2020 13:30	
67	📅	Vérification et débogage	0,188 jours	20/02/2020 10:30	20/02/2020 13:30	65
68	📅	Check Semaine 7	0,062 jours	20/02/2020 13:00	20/02/2020 13:30	67
69	📅	Autonomie	0,25 jours	21/02/2020 10:30	21/02/2020 13:30	
70	📅	Semaine 8 : Fin Calendrier	4,25 jours	24/02/2020 10:30	28/02/2020 13:30	63
71	📅	Autonomie	2 jours	24/02/2020 10:30	26/02/2020 10:30	
72	📅	Carte console : Transf HTML	0,062 jours	26/02/2020 10:30	26/02/2020 10:59	
73	📅	Exception Transformation	0,062 jours	26/02/2020 10:30	26/02/2020 10:59	
74	📅	Vérification et débogage	0,062 jours	27/02/2020 10:30	27/02/2020 11:30	72:73
75	📅	Versionning 3: Calendrier	0,062 jours	27/02/2020 11:00	27/02/2020 11:30	74
76	📅	Changement Organisation	0,021 jours	27/02/2020 11:30	27/02/2020 11:40	75
77	📅	Check Semaine 8	0,042 jours	27/02/2020 11:40	27/02/2020 13:30	76
78	📅	Point d'avancement 2	0,062 jours	27/02/2020 13:00	27/02/2020 13:30	52
79	📅	Autonomie	0,25 jours	28/02/2020 10:30	28/02/2020 13:30	

Document 3 : Gantt des Semaine 9-12: Du 2 au 26 Mars



B. Annexe 2: Diagramme de classe UML

