|  |  |
| --- | --- |
|  | **DOSSIER DE GESTION** |

**SAÉ**

**GRAMA**

|  |  |
| --- | --- |
| Bénéficiaire :  IUT Lyon 1 - Département Informatique  92 boulevard Niels Bohr  69100 VILLEURBANNE  [iutdoua.info@univ-lyon1.fr](mailto:iutdoua.info@univ-lyon1.fr) | Informations :  Auteurs :   * François GRAUX * Elliot CURVAT   Date de conception : 8 avril 2022 |

Table des matières

[Initialisation 3](#_Toc106174636)

[Objectifs 3](#_Toc106174637)

[Technologies 3](#_Toc106174638)

[Java : 3](#_Toc106174639)

[CSV : 4](#_Toc106174640)

[Dates clés 4](#_Toc106174641)

[Gestion de projet 5](#_Toc106174642)

[Méthode SCRUM 5](#_Toc106174643)

[Outils 5](#_Toc106174644)

[Réunions 5](#_Toc106174645)

[Organisation 5](#_Toc106174646)

[Pré-Sprint 7](#_Toc106174647)

[Sprint #0 7](#_Toc106174648)

[Sprint #1 8](#_Toc106174649)

[Sprint #2 8](#_Toc106174650)

[Sprint #3 9](#_Toc106174651)

[Sprint #4 9](#_Toc106174652)

[Sprint #5 10](#_Toc106174653)

# Initialisation

## Objectifs

L’objectif est de créer une application qui sera capable de lire un fichier de type CSV afin de modéliser un graphe. La finalité est de pouvoir manipuler ce graphe en obtenant des caractéristiques sur celui-ci.

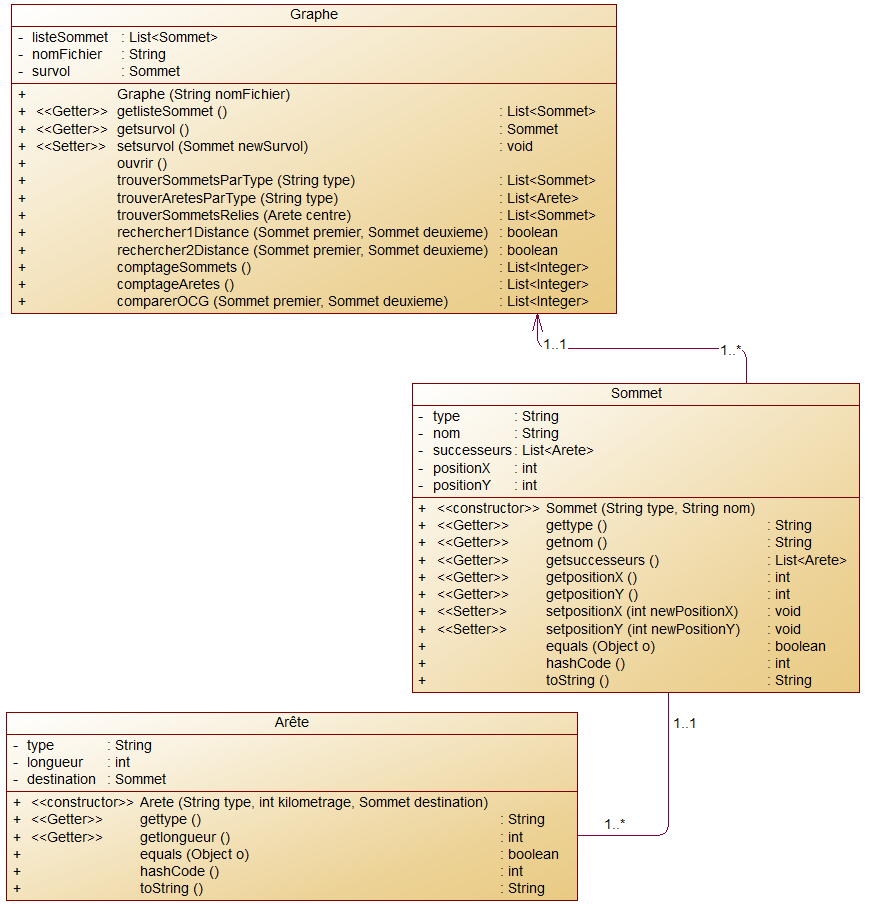
## Technologies

### Java :

Java est un langage de programmation apparu en 1991 et aujourd’hui majoritaire tant dans le monde privé que public. Celui-ci est orienté objet, organisé en paquets et sécurisé.

Dans le cadre de notre projet, Java est la solution la plus pratique pour nous. Le langage est régulièrement mis à jour, et s’adapte parfaitement avec nos applications actuelles.

Diagramme de classe de la structure utilisée :



### CSV :

CSV ou Comma Separated Values est un type de fichier qui, comme son nom l’indique, représente des données sous forme de lignes dont les valeurs sont séparées par des virgules. L’avantage principal de ce type de fichier est qu’il est universellement compréhensible et que beaucoup de logiciels le supportent.

Pour notre application, nous souhaitons pouvoir accéder à n’importe quel graphe à partir d’un fichier CSV correctement constitué.

Format du CSV : une ligne décrira tous les sommets reliés à un premier.

Par exemple : Sommet traité:Arête|Sommet destination;Arête|Sommet destination…

## Dates clés

Fourniture du code source le 14 juin à 23h59 maximum.

Fourniture des livrables le 16 juin à 23h59 maximum.

# Gestion de projet

## Méthode SCRUM

Pour ce projet, nous avons choisi la méthode SCRUM qui est un type de méthode agile. C’est une suite de sprints (périodes de 30 jours maximum), durant lesquels on se focalise sur une fonctionnalité en particulier.

Dans l’ordre et pour chaque sprint : on planifie, on design, on code, on test, on déploie, on review, on finalise, et on planifie le prochain sprint.

Au départ, l’équipe se réunit et décide de l’organisation des sprints.

Plusieurs points positifs nous ont dirigé vers les méthodes agiles et plus particulièrement la méthode Scrum :

* Facilité de retour en arrière : si un problème est détecté, on n’a pas besoin de tout recommencer,
* Tests réguliers : les tests effectués durant chaque sprint valident les fonctionnalités et les finalisent définitivement,
* Livraison partielle possible : si nécessaire, il est possible de livrer un produit partiellement fini. Il lui manquera certaines fonctionnalités, mais celles traitées fonctionneront totalement ;

## Outils

* IDE : ayant tous deux reçu un enseignement sur Apache NetBeans 12.6, c’est l’environnement de développement que nous allons utiliser - <https://netbeans.apache.org>
* Gestionnaire de versions : mis en place dès que possible, c’est vers Git que nous nous sommes tournés. Nous avons été formés sur Git et c’est un outil compatible avec notre choix d’IDE - <https://github.com>
* Calendrier : Apple iCloud Calendar - <https://www.icloud.com/calendar>
* Édition collaborative : Apple iCloud Pages - <https://www.icloud.com/pages>
* WBS, PBS et Gantt : MindView8 - <https://www.matchware.com/fr/>

## Réunions

Réunion d’initialisation : première réunion d’analyse du projet.

Réunion de répartition : réunion dédiée à la confection et à l’organisation des sprints en général.

Réunion de fin de sprint : pour conclure un sprint, revenir sur ce qui a bien été fait, ce qui a pris plus de temps que prévu… Réunion aussi prévue pour préparer le sprint suivant en détail.

Réunion de conclusion : retour sur le projet en général. Que nous a-t-il apporté ? A-t-on pu respecter les délais, les besoins du client... ?

## Organisation

Pour ce projet, nous avons décidé de développer les interfaces IHM au fur et à mesure de l’implantation des fonctionnalités. En effet, si l’on venait à avoir des problèmes de temps, on aurait un produit qui se rapprochera réellement d’une application et pas simplement un code inutilisable. Cependant, l’affichage du graphe sera traité à la toute fin du projet.

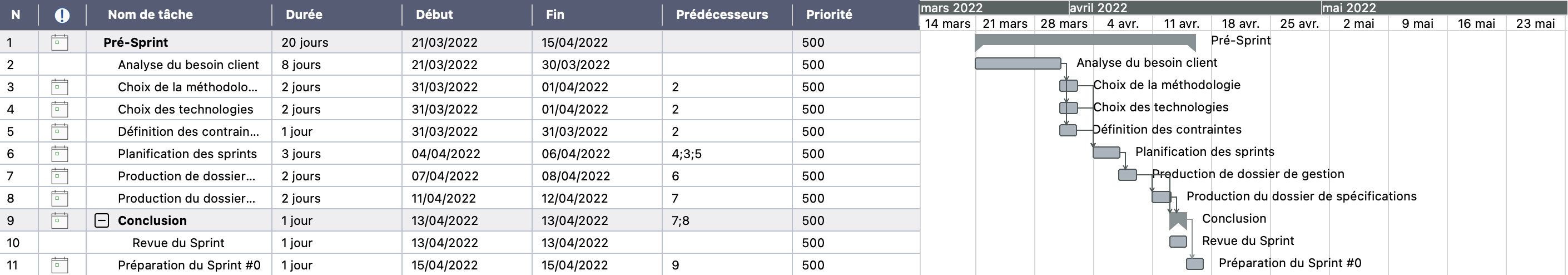
|  |  |
| --- | --- |
| Planification globale du projet : |  |
| Product Breakdown System : |  |
| Work  Breakdown System : | MindMap Preview |
| Matrice RACI : | |  |  |  | | --- | --- | --- | |  | François Graux | Elliot Curvat | | Cahier des charges | R | C | | Dossier de gestion | C | R | | Dossier de spécifications | A | R | | Structure du programme | R | A | | Développement des fonctionnalités | I | R | | Développement des IHM | R | A | | Javadoc | R | A | |

### Pré-Sprint

#### Tâches

* Analyse du besoin client
* Choix de la méthodologie
* Choix des technologies
* Définition des contraintes
* Planification des sprints
* Production du dossier de gestion
* Production du dossier de spécifications
* Conclusion
  + Revue du Sprint
* Préparation du Sprint #0

#### Planification



### Sprint #0

#### Tâches

* Structure du programme
* Importation du CSV
* IHM écran 0
  + Fonction d’affichage
  + Fonction de listage
  + Fonction de comptage
* Tests unitaires
* Conclusion
  + Revue du Sprint
* Préparation du Sprint #1

#### Planification

### Sprint #1

#### Tâches

* IHM écran 1
  + Voisins directs d’un nœud
  + Nœuds reliés par un lien
* Tests unitaires
* Conclusion
  + Tests d’intégration
  + Revue du Sprint
* Préparation du Sprint #2

#### Planification

### Sprint #2

#### Tâches

* IHM écran 2
  + Vérifier si un nœud est exactement entre deux autres
* Tests unitaires
* Conclusion
  + Tests d’intégration
  + Revue du Sprint
* Préparation du Sprint #3

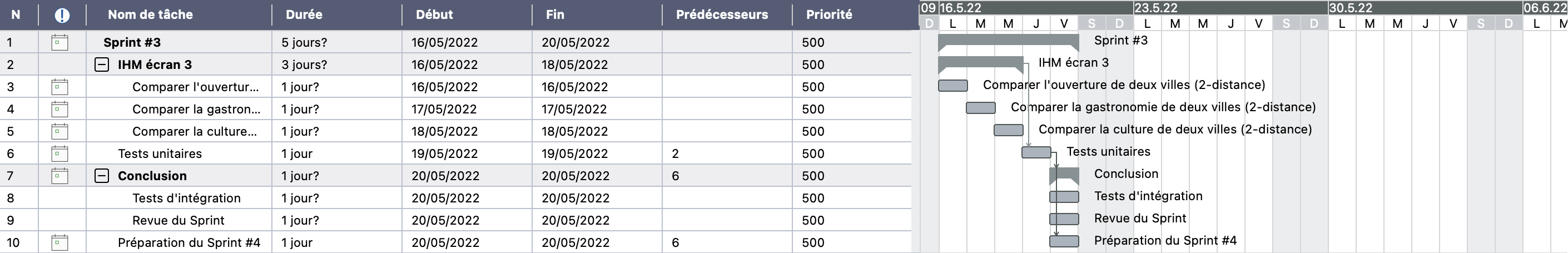
#### Planification

### Sprint #3

#### Tâches

* IHM écran 3
  + Comparer l’ouverture de deux villes
  + Comparer la gastronomie de deux villes
  + Comparer la culture de deux villes
* Tests unitaires
* Conclusion
  + Tests d’intégration
  + Revue du Sprint
* Préparation du Sprint #4

#### Planification



### Sprint #4

#### Tâches

* IHM écran 4
  + Trouver la plus courte distance entre deux noeuds
* Tests unitaires
* Conclusion
  + Tests d’intégration
  + Revue du Sprint
* Préparation du Sprint #5

#### Planification

### Sprint #5

#### Tâches

* IHM écran 4
  + Trouver un chemin passant par un type de nœud choisi
  + Trouver un chemin passant par X nœuds (de types différents ou identiques)
* Tests unitaires
* Conclusion
  + Tests d’intégration
  + Revue du Sprint
* Tests de validation

#### Planification