

# Cahier des charges

pour le projet de GPI  
Licence d'Informatique troisième année

B'OnTime : Système de gestion de la  
circulation des trains de la ligne L



rédigé par

Hervé-Madelein Attolou, Loïc Bachelot, Amélie  
Rémond, William Rozenberg, Adrien Tual

Janvier 2017

## Table des matières

<b>I</b>	<b>Présentation du projet</b>	<b>2</b>
1)	Définition du sujet . . . . .	2
2)	Présentation de l'équipe . . . . .	2
3)	L'analyse du besoin . . . . .	2
4)	L'avant-projet . . . . .	2
<b>II</b>	<b>Organisation du projet</b>	<b>3</b>
1)	Organisation de l'équipe . . . . .	3
2)	Environnement de travail . . . . .	3
<b>III</b>	<b>Fonctionnalités du programme</b>	<b>4</b>
1)	Fonctionnalités pour la version prototype . . . . .	4
2)	Fonctionnalités pour la version finale . . . . .	4
<b>IV</b>	<b>Gestion du temps</b>	<b>5</b>
1)	Calendriers des délais . . . . .	5
2)	Détails des délais de la version 1 . . . . .	5
3)	Détails des délais de la version 2 . . . . .	6
<b>V</b>	<b>Limites du projet</b>	<b>7</b>

## Liste des tableaux

1	Tableau récapitulatif des délais de la version 1 . . . . .	5
2	Tableau récapitulatif des délais de la version 2 . . . . .	6

## Table des figures

1	Calendrier des délais du mois de Janvier . . . . .	5
2	Calendrier des délais du mois de Février . . . . .	5
3	Calendrier des délais du mois de Mars . . . . .	5

## I Présentation du projet

### 1) Définition du sujet

Dans le cadre du module de GPI, nous devons réaliser une simulation de la ligne L du Transilien en java. Cela prend en compte les horaires, le mécanisme de canton, la gestion de plusieurs branches, la gestion des heures de pointe et des heures creuses et les incidents techniques impliquant des retards et des suppressions.

### 2) Présentation de l'équipe

Pour réaliser ce projet, nous sommes 5 : L'équipe **B'OnTime**.  
Notre équipe est composée de :

- Un chef de projet : Loïc Bachelot
- Un maître d'œuvre : Adrien Tual
- Un maître d'ouvrage : Hervé Attolou
- Un responsable de qualité : William Rozenberg
- Un responsable de documentation : Amélie Rémond

Chacun est responsable d'une partie du projet mais tous les membres du groupe participe à son élaboration.

### 3) L'analyse du besoin

Le programme doit évidemment être fonctionnel, mais il doit également répondre à toutes les attentes du client, tout en étant fiable, intuitif, esthétique et propre. C'est pourquoi ce projet ne consiste pas simplement à coder un programme mais à élaborer une organisation facilitant sa mise en œuvre.

### 4) L'avant-projet

Avant de nous lancer dans le projet en lui-même, nous avons fait une liste des fonctionnalités que le programme doit avoir et nous avons étudié la faisabilité en terme d'organisation et de technique. La suite du document a été approuvée par tous les membres du groupe.

## II Organisation du projet

### 1) Organisation de l'équipe

Pour mener à bien ce projet, il est impératif d'avoir une organisation rigoureuse. Nous nous sommes imposé un rythme de travail minimum de 4h pendant le cours et 4h hors du cours en plus d'une à deux réunions officielles par semaine. Nous avons également choisi un cycle de vie Agile : le cycle Crystal.

Ce cycle de conception nous permet de classer par importance les features et de concevoir chaque groupe de features en parallèle. On crée alors des versions, intermédiaires à chaque rendu, fonctionnelles, documentées, et utilisables. Le rendu final contient l'ensemble des features prévues et la documentation cumulées des versions précédentes.

Nous allons produire un prototype fonctionnel pour le 3 février 2017 et une version finale pour le 31 mars 2017.

### 2) Environnement de travail

Pour faciliter la communication entre les membres de l'équipe, nous utilisons tous :

- L'IDE IntelliJ de JetBrains
- La plateforme de partage de code Git lab
- L'application de communication Slack (en plus des mails, des messages, drives ..etc)

De plus, pour faciliter nos réunions et pour pouvoir travailler dans le calme, nous avons accès au local de la JuniorUCP.

### III Fonctionnalités du programme

Comme nous avons une version prototype et une version finale, nous avons décidé de mettre en œuvre les principales fonctionnalités dans la première version mais appliqués à une seule ligne. Dans la version finale, nous rajouterons des options en plus d'étendre le prototype à l'ensemble de la ligne L.

#### 1) Fonctionnalités pour la version prototype

Voici la liste de ce que le prototype du programme devra avoir :

- Modéliser une ligne de train avec un fonctionnement automatique (une seule branche et toutes les gares)
- Vue complète de la ligne
- Les trains partiront d'un terminus vers l'autre en s'arrêtant à toutes les gares.
- Bouton "pause" pour temporairement arrêter la simulation
- Bouton "création d'accident" pour déclencher une perturbation et observer les conséquences
- Bouton de réparation de l'incident
- Affichage des messages d'erreurs en bas de l'écran en cas de gros incident (ex : trains supprimés)

#### 2) Fonctionnalités pour la version finale

Voici la liste de ce que la version finale du programme devra avoir :

- Modélisation de toutes les branches
- Les trains attendent aux nœuds
- Bouton "visualisation" permettant de voir tous les trains circulant actuellement (ainsi que leur destinations, horaires et arrêts desservis)
- Train qui ne s'arrête pas à toutes les gares
- Possibilité de zoomer selon des niveaux
- Affichage adaptatif suivant le zoom
- L'utilisateur peut cliquer sur une gare pour avoir des informations (ex : horaires des trains de cette gare)
- Réparation automatique du train en montrant le temps restant
- Possibilité d'accélérer le temps

## IV Gestion du temps

Nous avons réparti les fonctionnalités décrites précédemment par rapport au délais imposés :

### 1) Calendriers des délais

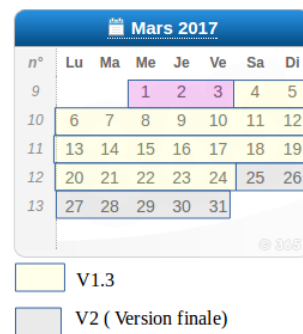
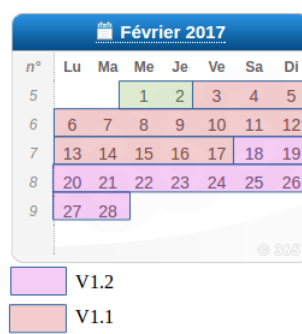
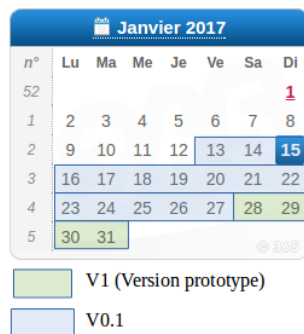


FIGURE 1 – Calendrier des délais du mois de Janvier  
FIGURE 2 – Calendrier des délais du mois de Février  
FIGURE 3 – Calendrier des délais du mois de Mars

### 2) Détails des délais de la version 1

Version 1		03/02/2017
V0.1	<ul style="list-style-type: none"> <li>- Modéliser une ligne de train avec un fonctionnement automatique (une seule branche et toutes les gares)</li> <li>- Les trains partiront d'un terminus vers l'autre en s'arrêtant à toutes les gares.</li> </ul>	27/01/2017
V1	<ul style="list-style-type: none"> <li>- Bouton "pause" pour temporairement arrêter la simulation</li> <li>- Bouton "création d'accident" pour déclencher une perturbation</li> <li>- Bouton de réparation de l'incident</li> <li>- Affichage des messages d'erreurs en bas de l'écran en cas de gros incident (ex : trains supprimés)</li> </ul>	03/02/2017

TABLE 1 – Tableau récapitulatif des délais de la version 1

### 3) Détails des délais de la version 2

Version 2		31/03/2017
V1.1	- Modélisation de toutes les branches - les trains attendent aux noeuds	17/02/2017
V1.2	- Bouton "visualisation" permettant de voir tous les trains circulant actuellement (ainsi que leur destinations, horaires et arrêts desservis) - Train qui ne s'arrête pas à toutes les gares	03/03/2017
V1.3	- possibilité de zoomer selon des niveaux - affichage adaptatif suivant le zoom - L'utilisateur peut cliquer sur une gare pour avoir des informations (ex : horaires des trains de cette gare)	24/03/2017
V2	- réparation automatique du train en montrant le temps restant - affichage adaptatif suivant le zoom - possibilité d'accélérer le temps	31/03/2017

TABLE 2 – Tableau récapitulatif des délais de la version 2

## V Limites du projet

L'organisation joue un rôle clé dans ce projet. Nous avons fixé des délais provisoires en nous basant sur nos connaissances en vue du travail à réaliser. Le temps peut être une limite importante, c'est pourquoi il est essentiel que nous tenions ces délais.

Il faut également respecter le rythme de travail que nous nous sommes fixé bien que nous ayons d'autres projets. En effet, toutes les dates ont été prévues en sachant que nous n'avons pas beaucoup de temps, c'est pour cela que notre projet est faisable.