



PROJET VITAMEAL

Restauration hospitalière



Nicolas SYMPHORIEN
Sonia OTHMANI
Jean-Félix BENITEZ

Feuille de suivi des évolutions

Indice	Éléments concernés	Date	Raison et nature de l'évolution
-	Toutes les pages	09/03/2017	Création du document
-	Partie élaboration	11/03/2017	Ajout de la description de l'usine logicielle

Table des matières

1	Présentation	6
1.1	Objet du document	6
1.2	Domaine d'application	6
1.3	Description du document	6
1.4	Emplacement du document	6
2	Documents	7
2.1	documents applicables	7
2.2	documents de référence	7
3	Terminologie	8
3.1	Abréviations	8
3.2	Glossaire	8
4	Initialisation	9
4.1	Définition du problème	9
4.2	Vision du projet	9
4.2.1	Solution envisagée	9
4.2.2	Périmètre	9
4.3	Analyse des exigences	9
4.3.1	Partie prenantes	9
4.3.2	Les besoins	9
4.3.3	Les contraintes	9
4.3.4	Exigences	9
4.4	TODO Estimation globale	10
5	Élaboration	11
5.1	Planification des activités	11
5.2	Affectation des ressources	12
5.3	Description de l'usine logicielle	12
5.3.1	Outils utilisés	13
5.3.2	Schema de fonctionnement	13
5.4	TODO Analyse	14
5.5	TODO Vision détaillée	14
5.6	TODO Cible	14
5.7	TODO Risques	14
5.8	TODO Besoins précis	14
5.9	TODO Définition itérative de l'architecture	14
5.10	TODO Estimation fine	14
6	TODO Construction	15
6.1	TODO Implémentation itérative	15
6.2	TODO Préparation déploiement	15

7	TODO	Transition	16
7.1	TODO	Tests bêta	16
7.2	TODO	Déploiement	16

Table des figures

5.1	Gantt	11
5.2	PERT	12
5.3	Ressources	12
5.4	Usine logicielle de Vitameal	13

Liste des tableaux

3.1	abréviations	8
3.2	glossaire	8

1 Présentation

1.1 Objet du document

Ce document est le rapport du travail fait sur le projet d'outil informatique destiné à la restauration hospitalière.

1.2 Domaine d'application

Formation du CNAM en ingénieur informatique première année.

1.3 Description du document

Les trois premiers chapitres définissent le contenu de ce document ; les chapitres suivants décrivent le travail fait sur ce projet.

1.4 Emplacement du document

<https://github.com/Seikomi/Vitameal/tree/master/doc>

2 Documents

2.1 documents applicables

Sans objet.

2.2 documents de référence

Sans objet.

3 Terminologie

3.1 Abréviations

TABLE 3.1 – abréviations

3.2 Glossaire

Ingrédients	Aliments de bases
Produits	Composé de plusieurs aliments

TABLE 3.2 – glossaire

4 Initialisation

4.1 Définition du problème

L'élaboration de menus dans un hôpital pour la restauration des patients est une tâche complexe, et doit tenir compte des différentes pathologies rencontrées. Faute de moyens (temps et argent) seules quelques grandes lignes de restauration sont retenues ; alors qu'idéalement, chaque patient devrait pouvoir avoir un repas adapté à sa pathologie.

4.2 Vision du projet

4.2.1 Solution envisagée

Mise en place d'un outil informatique permettant d'élaborer les menus des patients en fonction des profils diététiques, paramétré par le corps médical.

4.2.2 Périmètre

C'est un diététicien qui renseigne le profil diététique des patients, sous les directives des médecins. C'est aussi un diététicien qui élabore les menus des patients. L'outil permettra donc au diététicien d'élaborer les menus par filtrage des produits correspondants aux profils diététiques des patients.

4.3 Analyse des exigences

4.3.1 Partie prenantes

- Participantes : les diététiciens, le service restauration
- Concernés : les médecins, la direction (budget)
- Impactées : les patients

4.3.2 Les besoins

- Les diététiciens renseignent les profils diététiques de chaque patient.
- Les diététiciens élaborent les menus.
- Le service restauration commande les produits et ingrédients mis en œuvre dans les menus
- Le service restauration prépare les menus élaborés.

4.3.3 Les contraintes

- Les médecins doivent pouvoir vérifier / valider les profils diététiques des patients.
- La direction fixe un budget maximum par menu.

4.3.4 Exigences

- Fonctionnelles
 - Chaque patient a un profil diététique, renseigné par le diététicien

- Chaque menu élaboré par le diététicien, correspond à un ou plusieurs profils diététiques patients.
- À l'issue de l'élaboration des menus, la liste des produits et ingrédients (avec leur quantité) est faite afin que le service restauration puisse les commander.
- La liste des différents menus à réaliser est mise à disposition du service restauration.
- Non fonctionnelles
- À évaluer !

4.4 **TODO** Estimation globale

5 Élaboration

5.1 Planification des activités

Nous fixons la date de livraison à 2 semaines avant la présentation. La présentation du projet étant prévue pour le 14/09/2017 ; notre date de livraison est donc le **31/08/2017**. Entre le 4 mars et le 31 août, il y a 181 jours moins 7 jours fériés, nous disposons donc de **174 jours**.

Nous avons identifié huit étapes de développement :

- Analyse des exigences
- Cas d'utilisation
- Modèle de domaine.
- Séquences système
- Classes participantes.
- Diagramme d'interactions.
- Classes de conception.
- Code.

Pour évaluer la part de chaque étapes de développement, nous nous basons sur l'affirmation suivante « Aujourd'hui, un projet c'est 80% de réflexion et 20% de développement » (voir <http://www.logadap.fr/methodologie-creation-logiciel/>). Ainsi, le code va occuper 20% de notre temps, soit 35 jours ; reste 139 jours à répartir entre les 7 étapes précédentes, soit 20 jours chacune. Le diagramme de GANTT est donc le suivant :

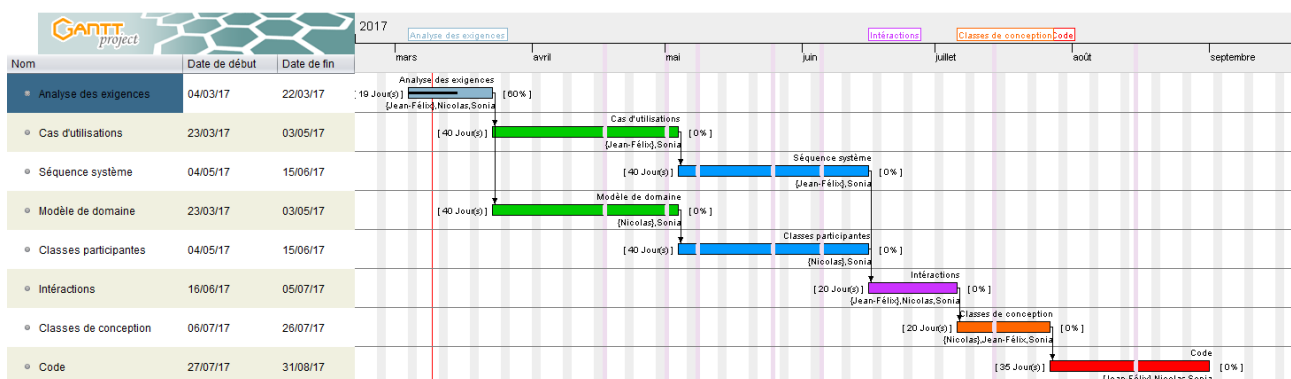


FIGURE 5.1 – Gantt

Le diagramme de PERT donne une autre vues de la répartition et de l'enchaînement des taches :

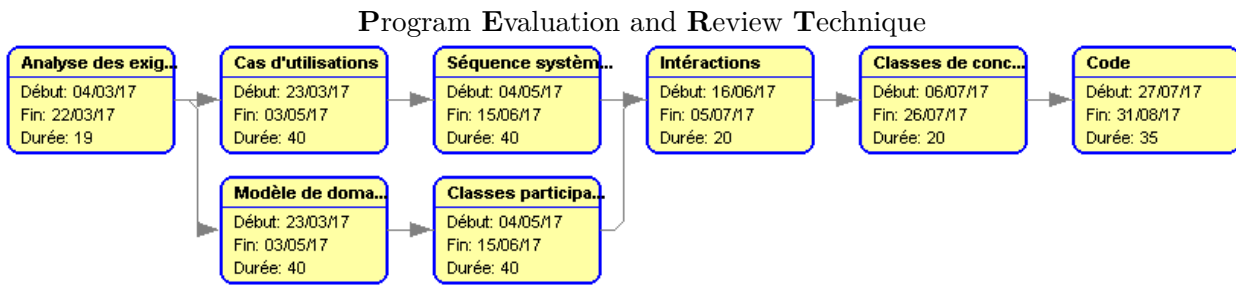


FIGURE 5.2 – PERT

5.2 Affectation des ressources

Les ressources sont affectées comme suit :

Tâches	Ressources
Analyse des exigences	Nicolas, Sonia, Jean-Félix
Cas d'utilisation	Jean-Félix 67%, Sonia 33%
Modèle de domaine	Nicolas 67%, Sonia 33%
Séquences système	Jean-Félix 67%, Sonia 33%
Classes participantes	Nicolas 67%, Sonia 33%
Diagramme d'interactions	Nicolas, Sonia, Jean-Félix
Classes de conception	Nicolas, Sonia, Jean-Félix
Code	Nicolas, Sonia, Jean-Félix

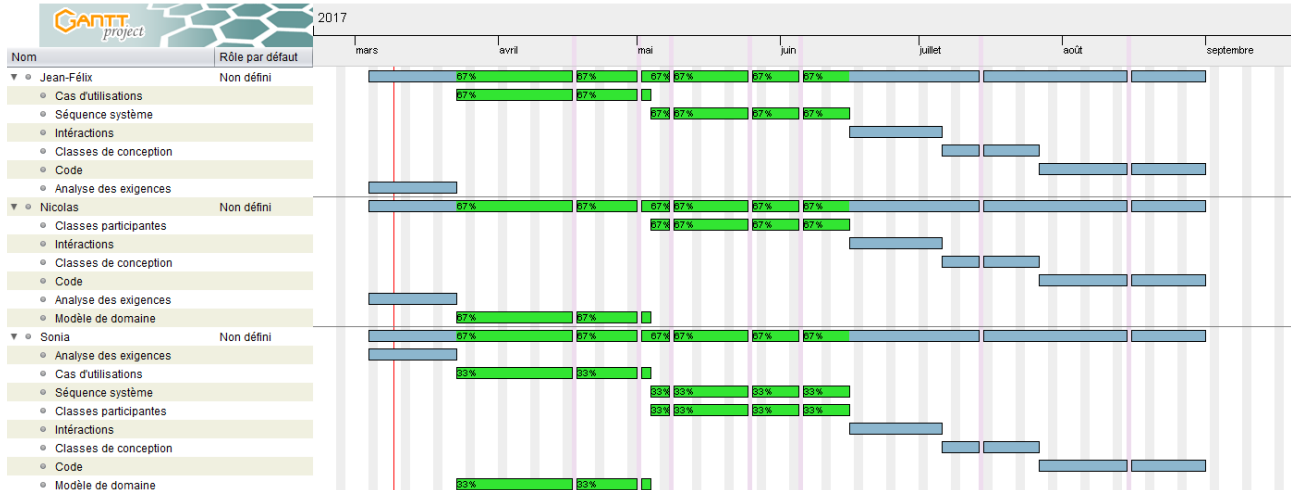


FIGURE 5.3 – Ressources

5.3 Description de l'usine logicielle

L'usine logicielle de Vitameal répond aux exigences suivantes :

- respecter les règles de qualités ;
- avoir une documentation claire et intégrée au projet ;
- gérer les erreurs et assurer leurs suivies ;
- versionnionner le code source et la documentation ;
- avoir un espace commun accessible à distance ;
- gérer un espace de livraison générant des indicateurs de santé sur le projet ;

- avoir un outil de conception UML couvrant la methode minimal UML ;
- gérer la planification du projet.

5.3.1 Outils utilisés

Les outils utilisés par l'usine logicielle de Vitameal se sépare en deux catégories :

- Le côté poste de développement qui correspond aux outils installés par chaque développeur sur sa machine ;
- Le côté espace d'intégration continue qui correspond aux outils composant l'espace communs de collaborations.

La documentation du projet est assurée par l'utilisation de la syntaxe *markdown* intégrée à l'outil *GitHub* et le langage de génération des livrables est *LaTeX*.

Le langage cible de cette usine est Java, mais elle peut facilement être adaptée à d'autre langage.

Côté poste de développement

- **Eclipse** comme IDE pour écrire/éditer le code de l'application ;
- **Maven** comme constructeur du projet (gestion des dépendances, automatisation de la construction ;
- **JUnit** pour écrire les tests unitaires de l'application et **Codertura** pour analyser la couverture du projet par ces tests ;
- **Git** pour versionner les sources du projet ;
- **StarUML** pour modéliser selon le standard UML le projet ;
- **GanttProject** pour planifier le projet avec un diagramme de Gantt ;
- **TEXMaker** pour éditer les fichiers `.tex` avec un comportement proche des *WYSIWYG* (optionnel).

Côté espace d'intégration continue

- **GitHub** comme gestionnaire à distance du repositorie Git principal, comme tracker de bug et comme affichage visuel des taches à faire ;
- **Jenkins** comme serveur d'intégration continue ;
- **SonarQube** comme analyseur de qualité du code.

5.3.2 Schema de fonctionnement

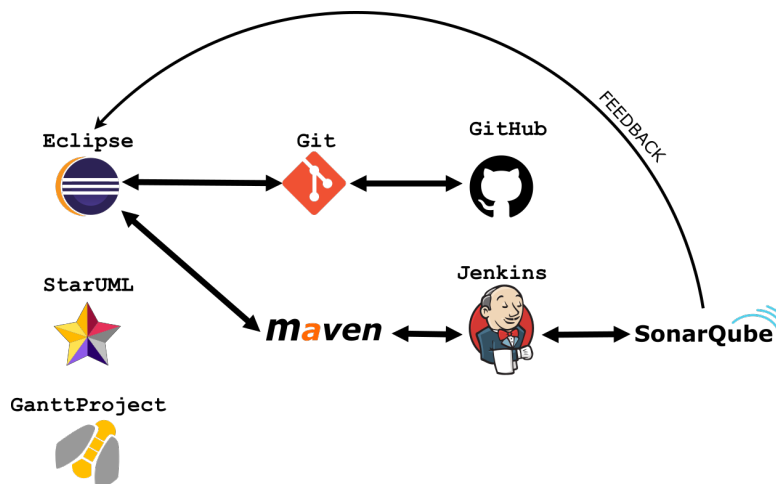


FIGURE 5.4 – Usine logicielle de Vitameal

- 5.4 **TODO** Analyse
- 5.5 **TODO** Vision détaillée
- 5.6 **TODO** Cible
- 5.7 **TODO** Risques
- 5.8 **TODO** Besoins précis
- 5.9 **TODO** Définition itérative de l'architecture
- 5.10 **TODO** Estimation fine

6 **TODO** Construction

6.1 **TODO** Implémentation itérative

6.2 **TODO** Préparation déploiement

7 **TODO** Transition

7.1 **TODO** Tests b ta

7.2 **TODO** D ploiement