|  |  |
| --- | --- |
| RÃ©sultats de recherche d'images pour Â«Â logo cegeptrÂ Â» | Département d’informatiqueAnalyse et conception **420-418-RI** **Groupe 00001** |

**Informatique 420.00**

**Projet Diététique**

**Remis à**

**M. François Jacob**

**Présenté par**

**Guillaume Ricard,**

**Sébastien Poulain et**

**Albert Ndjiki**

**10 mai 2019**

Tables des matières

[Département d’informatique 1](#_Toc8298375)

[Analyse et conception 1](#_Toc8298376)

[Définition du Projet 4](#_Toc8298377)

[Normes 6](#_Toc8298378)

[Normes d’écriture du code 6](#_Toc8298379)

[Normes « UX Design » 6](#_Toc8298380)

[Boutons 6](#_Toc8298381)

[Libellés et entêtes 8](#_Toc8298382)

[Datagridview 10](#_Toc8298383)

[Backlog 11](#_Toc8298384)

[Bac à glace 13](#_Toc8298385)

[Contenu des itérations 14](#_Toc8298386)

[Sprint 0: 14](#_Toc8298387)

[Contenu planifié 14](#_Toc8298388)

[Contenu réalisé 14](#_Toc8298389)

[Sprint 1: 14](#_Toc8298390)

[Contenu planifié 14](#_Toc8298391)

[Contenu réalisé 15](#_Toc8298392)

[Sprint 2: 16](#_Toc8298393)

[Contenu Planifié 16](#_Toc8298394)

[Contenu réalisé 18](#_Toc8298395)

[Sprint 3: 19](#_Toc8298396)

[Contenu planifié 19](#_Toc8298397)

[Contenu réalisé 21](#_Toc8298398)

[Sprint 4 22](#_Toc8298399)

[Contenu Planifier 22](#_Toc8298400)

[Contenu réalisé 22](#_Toc8298401)

[Burnup chart 23](#_Toc8298402)

[23](#_Toc8298403)

[La répartition des tâches 24](#_Toc8298404)

[Outil utilisé pour la gestion des stories 26](#_Toc8298405)

[Problèmes principaux rencontrés 26](#_Toc8298406)

[Apprendre le code déjà existant 26](#_Toc8298407)

[WampServer 26](#_Toc8298408)

[Schéma de la base données 27](#_Toc8298409)

# Définition du Projet

**Membre de l’équipe**

Guillaume Ricard-[guillaume9617@gmail.com](mailto:guillaume9617@gmail.com)

Sébastien Poulain- [sebsoccer30@gmail.com](file:///C:\Users\sebas\Downloads\sebsoccer30@gmail.com)

Albert Ndjiki - [neymarbreezy33@gmail.com](file:///C:\Users\sebas\Downloads\neymarbreezy33@gmail.com)

**Nom** : Projet diététique

**But :** Poursuivre l’application de diététique commencée l’année dernière

**Durée des sprints**: Sprint = 2 semaines et le dernier de 1 semaine

**Outils de collaboration**: Trello et Google décochent

**Matériel informatique**: (un ordinateur, un moniteur, une souris, un clavier) par programmeur

**Ressource**: (un bureau, une chaise, rallonge multiprise) par programmeur, une cafetière, une machine à eau, tableau blanc, divan, console de jeu

**IDE**: VisualStudio

**Langages de prog**: vb.net et mySQL

Définition de fini d’une release

Les définitions de fini d’un sprint.

Tests fonctionnels d’intégration faits et réussis.

Toutes les stories commencées sont terminées

L’équipe a validé le résultat de la release.

**Définition de fini pour Feature:**Test de diversité, Test de l’environnement, Test d’intégrité,Test unitaire, commentées, déboguer, toutes les stomies fonctionnent entre elles sans bogue,

**Définition de fini pour Stories:**

Le code des fonctionnalités prévues a été complété.

Le code a été écrit selon les normes d’écriture du code définies.

L’interface est en français standard et sans fautes.

L’interface a été réalisée selon les normes « UX Design » définies.

Tests fonctionnels réalisés et réussis.

Des commentaires documentent le code.

**Définition de fini d’un sprint:**

Les définitions de fini d’une story.

Le backlog a été affiné.

Tests fonctionnels d’intégration faits et réussis.

**Définition des tests**

**Test de diversité :** réaliser des tests avec 7 individus ayant des profils différents qui donnent leur opinion sur l’application.

**Test de l’environnement :** S’assurer que le backend fonctionne sans erreurs dans l’écosystème IOS, Android et Windows.

**Test d’intégrité** : Tester le volume maximal de données gérable, tester des valeurs impossibles et effectuer des vérifications aux limites puis vérifier que l’intégrité de celles-ci est respectée.

**Test unitaire**: Tester l’application avec un enregistrement complet

# Normes

## Normes d’écriture du code

* Les noms des variables et des fonctions sont selon le format « nom\_variable ».
* Première lettre en majuscule dans les noms de classes, procédures et fonctions.
* Commentaires au-dessus du code à commenter.

## Normes « UX Design »

### Boutons

#### **Bouton de fermeture**

* 50x28 pixels.
* Flat.
* Aucune bordure.
* La couleur devient « Firebrick » quand l’on passe le curseur.

#### **Bouton minimiser**

* 50x28 pixels.
* Flat.
* Aucune bordure.

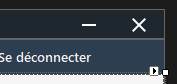


Figure 1 : Bouton de fermeture (à droite) et minimiser (à gauche).

#### **Boutons du menu**

* 239x35 pixels.
* Flat.
* Aucune bordure.
* La couleur devient blanche et le texte noir quand l’on passe le curseur.



Figure 2 : Bouton du menu

#### **Bouton standard**

* 110x32 pixels.
* Flat.
* Aucune bordure.
* Couleur : « RGB (52, 73, 94) » ou « DimGray »
* Couleur du libellé : « Control »
* La couleur devient blanche et le texte noir quand l’on passe le curseur.



Figure 3 : Bouton standard.

#### **Bouton standard large**

* 185x32 pixels.
* Flat.
* Aucune bordure.
* Couleur : « RGB (52, 73, 94) » ou « DimGray »
* Couleur du libellé : « Control »
* La couleur devient blanche et le texte noir quand l’on passe le curseur.



Figure 4 : Bouton standard large.

#### **Bouton standard petit**

* 95x32 pixels.
* Flat.
* Aucune bordure.
* Couleur : « RGB (52, 73, 94) » ou « DimGray » ou RGB (72, 84, 96)
* Couleur du libellé : « Control »
* La couleur devient blanche et le texte noir quand l’on passe le curseur.



Figure 5 : Bouton standard petit.

#### **Bouton standard mini**

* 97, 25 pixels.
* Flat.
* Aucune bordure.
* Couleur : RGB (72, 84, 96)
* Couleur du libellé : « Control »
* La couleur devient blanche et le texte noir quand l’on passe le curseur.



Figure 6 : Bouton standard mini.

#### **Bouton Excel**

* 115 32 pixels.
* Flat.
* Aucune bordure.
* Couleur : RGB (63, 124, 63)
* Couleur du libellé : « Control »
* La couleur devient blanche et le texte noir quand l’on passe le curseur.



Figure 7 : Bouton Excel.

### Libellés et entêtes

#### **Libellé standard**

* Segoe UI
* 10 pt
* Couleur : « Black »



Figure 8 : Libellé standard.

#### **Libellé mini**

* Segoe UI
* 9 pt
* Couleur : « Black »



Figure 9 : Libellé mini.

#### **Libellé format**

* Segoe UI
* 8.25 pt
* Couleur : « Gray »



Figure 10 : libellé format.

#### **Libellé erreur**

* Segoe UI
* 8.25 pt
* Couleur : « Firebrick »



Figure 11 : Libelléerreur.

#### **Entête standard**

* Couleur : RGB (72, 84, 96)ou RGB (30, 39, 46)
* Couleur du libellé : « Control »
* Segoe UI
* 14.25 pt
* Largeur : 35 pixels
* Longueur : Selon le contrôle le suivant ou la fenêtre.



Figure 12 : Entête standard.

#### **Entête petit**

* Couleur : RGB (72, 84, 96)
* Couleur du libellé : « Control »
* Segoe UI
* 14.25 pt
* Largeur : 21 pixels
* Longueur du contrôle en dessous ou de la fenêtre.



Figure 13 : Entête petit.

# Datagridview

* Segoe UI
* La taille de la police est celle où tous les champs sont visibles.
* Police de l’entête en gras et 1 pt plus gros.
* Entête de ligne désactivée.
* La ligne est complètement sélectionnée à la sélection.
* Couleur à la sélection : RGB (189, 195, 199)
* Couleur lorsque le curseur est sur une ligne : RGB (236, 240, 241)

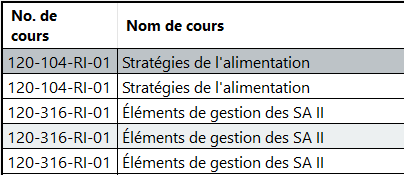


Figure 14 : Datagridview.

# Backlog

1. Créer un onglet recette (désigné) à partir des onglets qui existe déjà
2. Dois permettre d’importer une recette sous forme PDF
3. Pouvoir écrire une recette à la main
4. Pour les profs et admin
5. Ajouter une image à la recette comme pièce jointe
6. Imprimer la recette
7. Sauvegarder les recettes
8. Visualiser les recettes dans une liste
9. Entrer une date de remise lorsque le prof crée un exercice
10. Prends la date du système
11. Différencier les travaux par couleur (du côté élève)  (retard rouge, avant vert, remis jaune)
12. Date de remise des travaux des étudiants
13. Accepte ou non les travaux en retard
14. Notifié de l'élève lorsque le travail est remis par le professeur
15. Déplacer la création de produits pour le mettre dans l’onglet produit
16. Changer les taux des taxes
17. Debugger
18. Case à cocher allergène
19. Enlever les codes-barres et plu
20. Ajouter des produits
21. Historique
22. Envoyer la commande
23. Test unitaire
24. Test d’intégration
25. Commenter le nouveau code
26. Rédaction des tests unitaires
27. Rédaction des tests d’intégration
28. Ajouter les nouveautés au système (FINI)
29. Lier la recette à un exercice que le prof donne
30. La recette transmise par le professeur doit être visible par les étudiants
31. Sauvegarder lien avec eut
32. Création d’une table pour les recettes (nom de l'aliment, quantité, type de mesure)
33. Gérer les recettes pour les profs, étudiant et admin
34. Pareille comme le prof
35. Réception de la commande et validation
36. Ajouter automatiquement à l’inventaire
37. Ajouter espace catégorie produit
38. Produits retirer description et ajouter espace pour nom produit
39. Catégorie + caractères
40. Gestion exercice tous défaut
41. Gestion exercice ajouter recherche par tous
42. Update la table inventaire lors de la validation (réception du produit)
43. 2 profs assignés aux mêmes cours
44. Prof accès à l’exercice
45. Modifier les exercices les 2
46. Voir les travaux des étudiants
47. Brève même opération de prof, mais pour 2
48. Message de confirmation lors de la déconnexion ou de quitter
49. Choix cours double-clic
50. Agrandirez la fenêtre
51. Pouvoir choir un cours en double cliquant plutôt que sélectionner cours et cliquer sur « Poursuivre  (double-clic évent) enlever bouton poursuivre
52. Enseign : mes exercices. Pouvoir visualiser « Tous » plutôt qu’« Actif » comme statut par défaut. (combobox index 0 au lieu de 1)
53. Enseign : Exercices. Pouvoir déposer des pièces jointes (ex. : recettes, menu, instructions). (insertion pièce jointe lors de l’ajout d’une activité dans le form exercice)
54. Admin : Nom des catégories. Prévoir plus de caractère (50 ou plus). (bd catégorie champs nom catégorie varchar2(60))
55. Dans Admin, Gestion des exercices. Visualiser par défaut les statuts « Tous » plutôt qu’« Actif » (comboboxformload 0).
56. Admin, Gestion des exercices, Rechercher. Par…tous (à ajouter, par défaut) (ajout combobox)
57. Admin, Paramètres, cours et groupe : Possibilité d’assigner plus d’un prof à un cours. (enlever la vérification si prof assigné)
58. Montant d’argent : Utiliser des virgules et mettre un espace entre le montant et le signe de dollar (ex. : 3,14 $). (format,regex)
59. Enseign, Exercice, Commande, comparer ma commande. Est-il possible de marque les étudiants qui ont remis le travail (ex. : couleur différente)? (backcolor)
60. Indiquer que l’étudiant a remis le travail
61. L’étudiant peut voir qu’il a remis le travail
62. Régler des bogues sur l'onglet recette
63. Mettre les messages du prof de l’exercice dans l'onglet recette
64. Faire une copie de l’inventaire de l’admin et le prof peut le modifier pour l’exercice
65. Réaliser un panel pour voir les boutons dans la commande
66. Régler le bogue lier à l'ajout de produit lier a l'exercice dans le datagriview

# Bac à glace

1. Inventaire des locaux
2. Réaliser un panel pour voir les boutons dans la commande
3. Inventaire plats/différents formats
4. Définir un onglet lot dans l’onglet commande
5. Faire une copie de l’inventaire de l’admin et le prof peut le modifier pour l’exercice
6. Produits inventaire exercice
7. Commande administrateur

# Contenu des itérations

# Sprint 0:

## Contenu planifié

* Déterminer les définitions de fini.
* Identifier les normes (UX Design et de code).
* Remplir le backlog en prévision de l’ensemble des sprints
* Identification des améliorations techniques
* Compréhension du fonctionnement de l’application
* Assigner les story points au story

## Contenu réalisé

* Déterminer les définitions de fini.
* Identifier les normes (UX Design et de code).
* Remplir le backlog en prévision du Sprint 1.
* Identification des améliorations techniques
* Compréhension du fonctionnement de l’application
* Assigner les story points au story

# Sprint 1:

## Contenu planifié

**GESTION DE RECETTE**

Création de l’onglet recette

* Créer un onglet recette (désigné) à partir des onglets qui existe déjà
* Dois permettre d’importer une recette sous forme PDF
* Pouvoir écrire une recette à la main
* Pour les profs et admin
* Ajouter une image à la recette comme pièce jointe
* Imprimer la recette
* Sauvegarder les recettes
* Visualiser les recettes dans une liste

**GESTION DE TRAVAUX**

Gestion d’exercice LÉA

* Entrer une date de remise lorsque le prof crée un exercice
* Prends la date du système
* Différencier les travaux par couleur (du côté élève)  (retard rouge, avant vert, remis jaune)
* Date de remise des travaux des étudiants
* Accepte ou non les travaux en retard
* Notifié de l'élève lorsque le travail est remis par le professeur

GESTION DE PRODUITS

* Gérer la création de produits
* Déplacer la création de produits pour le mettre dans l’onglet produit
* Changer les taux des taxes
* Debugger
* Case à cocher allergène
* Enlever les codes-barres et plu

Commande administrateur

* Ajouter des produits
* Historique
* Envoyer la commande

**INTÉGRATION**

* Test unitaire
* Test d’intégration
* Commenter le nouveau code
* Rédaction des tests unitaires
* Rédaction des tests d’intégration
* Ajouter les nouveautés au système (FINI)

## Contenu réalisé

**GESTION DE RECETTE**

Création de l’onglet recette

1. Créer un onglet recette (désigné) à partir des onglets qui existe déjà
2. Dois permettre d’importer une recette sous forme PDF, Word, Excel
3. ~~Pouvoir écrire une recette à la main~~
4. ~~Pour les profs et admin~~
5. Ajouter une image à la recette comme pièce jointe
6. ~~Imprimer la recette~~
7. Sauvegarder les recettes
8. Visualiser les recettes dans une liste

**GESTION DE TRAVAUX**

Gestion d’exercice LÉA

* Entrer une date de remise lorsque le prof crée un exercice
* Prends la date du système
* Différencier les travaux par couleur (du côté élève)  (retard rouge, avant vert, remis jaune)
* Date de remise des travaux des étudiants
* Accepte ou non les travaux en retard
* Notifié de l'élève lorsque le travail est remis par le professeur

**GESTION DE PRODUITS**

* Gérer la création de produits
* Déplacer la création de produits pour le mettre dans l’onglet produit
* Changer les taux des taxes
* Debugger
* Case à cocher allergène
* Enlever les codes-barres et plu

Commande administrateur

* ~~Ajouterdes produit~~
* ~~Historique~~
* ~~Envoyer la commande~~

**INTÉGRATION**

* Test unitaire
* Test d’intégration
* Commenter le nouveau code
* Rédaction des tests unitaires
* Rédaction des tests d’intégration
* Ajouter les nouveautés au système (FINI)

# Sprint 2:

## Contenu planifié

**GESTION DE RECETTE**

Prof/ eut/ recette

* Lier la recette à un exercice que le prof donne
* La recette transmise par le professeur doit être visible par les étudiants
* Sauvegarder lien avec eut
* Création d’une table pour les recettes (nom de l'aliment, quantité, type de mesure)
* Gérer les recettes pour les profs, étudiant et admin

**GESTION DE PRODUITS**

Inventaire réel de l’administrateur

* Pareille comme le prof
* Réception de la commande et validation
* Ajouter automatiquement à l’inventaire
* Ajouter espace catégorie produit
* Produits retirer description et ajouter espace pour nom produit
* Catégorie + caractères
* Gestion exercice tous défaut
* Gestion exercice ajouter recherche par tous

Commande administrateur

* Update la table inventaire lors de la validation (réception du produit)

**Gestion de professeur**

Pouvoir ajouter 2 profs assignés aux mêmes cours

* 2 profs assignés aux mêmes cours
* Prof accès à l’exercice
* Modifier les exercices les 2
* Voir les travaux des étudiants
* Brève même opération de prof, mais pour 2

**Menu**

* Message de confirmation lors de la déconnexion ou de quitter
* Choix cours double-clic
* Agrandirez la fenêtre

**INTÉGRATION**

* Test unitaire
* Test d’intégration
* Commenter le nouveau code
* Rédaction des tests unitaires
* Rédaction des tests d’intégration
* Ajouter les nouveautés au système (FINI)

## Contenu réalisé

**GESTION DE RECETTE**

Prof/ eut/ recette

* Lier la recette à un exercice que le prof donne
* La recette transmise par le professeur doit être visible par les étudiants
* Sauvegarder lien avec eut
* Création d’une table pour les recettes (nom de l'aliment, quantité, type de mesure)
* Gérer les recettes pour les profs, étudiant et admin

**GESTION DE PRODUITS**

Inventaire Prof / eut

* Ajouter espace catégorie produit
* Produits retirer description et ajouter espace pour nom produit
* Catégorie + caractères
* Gestion exercice tous défaut
* Gestion exercice ajouter recherche par tous

Inventaire réel de l’administrateur

* ~~Pareille comme le prof~~
* ~~Réception de la commande et validation~~
* ~~Ajouter automatiquement à l’inventaire~~

Commande administrateur

* ~~Update la table inventaire lors de la validation (réception du produit)~~

**Gestion de professeur**

Pouvoir ajouter 2 profs assignés aux mêmes cours

* 2 profs assignés aux mêmes cours
* Prof accès à l’exercice
* Modifier les exercices les 2
* Voir les travaux des étudiants
* Brève même opération de prof, mais pour 2

**Menu**

Menu

* Message de confirmation lors de la déconnexion ou de quitter
* Choix cours double-clic
* Agrandirez la fenêtre

**INTÉGRATION**

* Test unitaire
* Test d’intégration
* Commenter le nouveau code
* Rédaction des tests unitaires
* Rédaction des tests d’intégration
* Ajouter les nouveautés au système (FINI)

# Sprint 3:

## Contenu planifié

**DEMANDE PARTICULIÈRE PROF DIÉTÉTIQUE**

Gestion des travaux

* Pouvoir choir un cours en double cliquant plutôt que sélectionner cours et cliquer sur « Poursuivre  (double-clic event) enlever bouton poursuivre
* Enseign : mes exercices. Pouvoir visualiser « Tous » plutôt qu’« Actif » comme statut par défaut. (combobox index 0 au lieu de 1)

Enseign : Exercices. Pouvoir déposer des pièces jointes (ex. : recettes, menu, instructions).(insertion pièce jointe lors de l’ajout d’une activité dans la forme exercice).

Modification d’élément dans admin

* Admin : Nom des catégories. Prévoir plus de caractère (50 ou plus). (bd catégorie champs nom catégorie varchar2(60))
* Dans Admin, Gestion des exercices. Visualiser par défaut les statuts « Tous » plutôt qu’« Actif » (comboboxformload 0).
* Admin, Gestion des exercices, Rechercher. Par…tous (à ajouter, par défaut) (ajout combobox)
* Admin, Paramètres, cours et groupe : Possibilité d’assigner plus d’un prof à un cours. (enlever la vérification si prof assigné)
* Montant d’argent : Utiliser des virgules et mettre un espace entre le montant et le signe de dollar (ex. : 3,14 $). (format,regex)

Gestion des Exercise/Commande

* Enseign, Exercice, Commande, comparer ma commande. Est-il possible de marque les étudiants qui ont remis le travail (ex. : couleur différente)? (backcolor)
* Indiquer que l’étudiant a remis le travail
* L’étudiant peut voir qu’il a remis le travail

**GESTION DE RECETTE**

Message et bogue

* Régler des bogues sur l'onglet recette
* Mettre les messages du prof de l’exercice dans l'onglet recette

**GESTION DE PRODUITS**

* Faire une copie de l’inventaire de l’admin et le prof peut le modifier pour l’exercice
* Réaliser un panel pour voir les boutons dans la commande
* Régler le bogue lier à l'ajout de produit lier a l'exercice dans le datagriview

**INTÉGRATION**

* Test unitaire
* Test d’intégration
* Commenter le nouveau code
* Rédaction des tests unitaires
* Rédaction des tests d’intégration
* Ajouter les nouveautés au système (FINI)

## Contenu réalisé

**DEMANDE PARTICULIÈRE PROF DIÉTÉTIQUE**

Gestion des travaux

* Pouvoir choir un cours en double cliquant plutôt que sélectionner cours et cliquer sur « Poursuivre  (double-clic évent) enlever bouton poursuivre
* Enseign : Mes exercices. Pouvoir visualiser « Tous » plutôt qu’« Actif » comme statut par défaut. (combobox index 0 au lieu de 1)

Enseign : Exercices. Pouvoir déposer des pièces jointes (ex. : recettes, menu, instructions).(insertion pièce jointe lors de l’ajout d’une activité dans le form exercice).

Modification d’élément dans admin

* Admin : Nom des catégories. Prévoir plus de caractère (50 ou plus). (bd catégorie champs nom catégorie varchar2(60))
* Dans Admin, Gestion des exercices. Visualiser par défaut les statuts « Tous » plutôt qu’« Actif » (comboboxformload 0).
* Admin, Gestion des exercices, Rechercher. Par…tous (à ajouter, par défaut) (ajout combobox)
* Admin, Paramètres, cours et groupe : Possibilité d’assigner plus d’un prof à un cours. (enlever la vérification si prof assigné)
* Montant d’argent : Utiliser des virgules et mettre un espace entre le montant et le signe de dollar (ex. : 3,14 $). (format,regex)

Gestion des Exercise/Commande

* Enseign, Exercice, Commande, comparer ma commande. Est-il possible de marquer les étudiants qui ont remis le travail (ex. : couleur différente)? (backcolor)
* Indiquer que l’étudiant a remis le travail
* L’étudiant peut voir qu’il a Remi le travail

**GESTION DE RECETTE**

Message et bogue

* Régler des bogues sur l'onglet recette
* Mettre les messages du prof de l’exercice dans l'onglet recette

**GESTION DE PRODUITS**

* ~~Faire une copie de l’inventaire de l’admin et le prof peut le modifier pour l’exercice~~
* ~~Réaliser un panel pour voir les boutons dans la commande~~
* Régler le bogue lier à l'ajout de produit lier a l'exercice dans le datagriview

**INTÉGRATION**

* Test unitaire
* Test d’intégration
* Commenter le nouveau code
* Rédaction des tests unitaires
* Rédaction des tests d’intégration
* Ajouter les nouveautés au système (FINI)

# Sprint 4

## Contenu Planifier

* Message et bogue pour l’onglet recette
* Retirer les éléments inutiles de l’application/ ne fonctionnais pas ou pas fini
* Dernière vérification des bogues sur l’ensemble sur système
* Ajout des couleurs dans l’onglet remise des travaux
* Documentation Guide
* Ajouter les nouvelles taxes au calcul de la commande
* Test
* Documentation
* Diagramme UML
* Test unitaire
* Test d’intégration
* Commenter le nouveau code
* Rédaction des tests unitaires
* Rédaction des tests d’intégration
* Ajouter les nouveautés au système (FINI)

## Contenu réalisé

* Message et bogue pour l’onglet recette
* Retirer les éléments inutiles de l’application/ ne fonctionnais pas ou pas fini
* Dernière vérification des bogues sur l’ensemble sur système
* Ajout des couleurs dans l’onglet remise des travaux
* Documentation Guide
* Ajouter les nouvelles taxes au calcul de la commande
* Test
* Documentation
* Diagramme UML
* Test unitaire
* Test d’intégration
* Commenter le nouveau code
* Rédaction des tests unitaires
* Rédaction des tests d’intégration
* Ajouter les nouveautés au système (FINI)

# Burnup chart

# 

# La répartition des tâches

Sprint 1

Points D’efforts : 19

|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Description | Temps | Tâche |
| Guillaume | Création de l’onglet recette | 2 semaines | Scrum master, programmeur |
| Sébastien | Case à cocher allergène | 1 semaine | Programmeur |
| Sébastien | Gérer la création de produits réels | 1 semaine | Programmeur |
| Sébastien | Enlever les codes-barres et plu | 1 heure | Programmeur |
| Guillaume, Sébastien, Albert | Test | 4 heures | Testeur |
| Albert | Remise exercice + date | 2 semaines | Programmeur |
|  |  |  |  |

Sprint 2

Points D’efforts : 30

|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Description | Temps | Tâche |
| Sébastien | Faire les demandes spéciales des clientes | 1 semaine et 1/2 | Programmeur |
| Guillaume | Prof/ étu / recette, soit lier les recettes à l’étudiant et professeur | 1 semaine | Programmeur |
| Guillaume | Lier les exercices et les recettes | 1 semaine | Programmeur |
| Sébastien | Message de confirmation lors de la déconnexion ou de quitter. | 1 heure | Scrum Master, Programmeur |
| Sébastien | Ajouter espace catégorie produit | 4 heures | Scrum Master, Programmeur |
| Sébastien | Produits retirer description et ajouter espace pour nom produit | 1 heure | Scrum Master, Programmeur |
| Sébastien | Gestion exercice tous défaut | 4 heures | Scrum Master, Programmeur |
| Sébastien | Choix cours | 2 heures | Scrum Master, Programmeur |
| Guillaume, Sébastien, Albert | Test | 4 heures | Testeur |
| Albert | Profs Léa | 1 semaine | Programmeur |
| Albert | Ajouter plus d’un prof a un cours | 1 semaine | Programmeur |

Sprint 3

Points D’efforts : 32

|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Description | Temps | Tâche |
| Guillaume | Lier les recettes et exercices | 1 semaine | Programmeur |
| Guillaume | Régler des bogues sur l’interface recette | 6 heures | Testeur |
| Guillaume | Lier le message du professeur avec la recette | 8 heures | Programmeur |
| Sébastien | Changer les premières lettres en majuscule | 1 heure | Programmeur |
| Sébastien | Changer le format $ | 4 heures | Programmeur |
| Sébastien | Modification d’élément dans admin | 8 heures | Programmeur |
| Sébastien | Gestion des Exercise/Commande | 4 heures | Programmeur |
| Sébastien | INTÉGRATION | 2 heures | Programmeur |
| Guillaume, Sébastien, Albert | Test | 4 heures | Testeur |
| Albert | Gestion des Exercices/Commande | 4 heures | Scrum Master, Programmeur |

Sprint 4

Points D’efforts : 10

|  |  |  |  |
| --- | --- | --- | --- |
| Nom | Description | Temps | Tâche |
| Guillaume | Message et bogue pour l’onglet recette | 4h | Programmeur |
| Guillaume | Retirer les éléments inutiles de l’application/ ne fonctionnais pas ou pas fini | 1 jour | Programmeur |
| Sébastien  Guillaume | Dernière vérification des bogues sur l’ensemble sur système | 2 jours | Test |
| Sébastien  Guillaume | Ajout des couleurs dans l’onglet remise des travaux | 2h | Programmeur |
| Albert | Documentation Guide | 5h | Scrum Master, Programmeur |
| Sébastien | Ajouter les nouvelles taxes au calcul de la commande | 5h | Programmeur |
| Guillaume, Sébastien, Albert | Test | 4 heures | Testeur |
| Guillaume | Documentation | 5h | Rédacteur de documentations |
| Albert | Diagramme UML | 1h | Scrum Master, Programmeur |
| Sébastien | Régler des bogues et retravailler le code | 12h | programmeur |

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint | Albert | Guillaume | Seb |
| 0 | 12 | 12 | 12 |
| 1 | 17 | 23 | 32 |
| 2 | 25 | 25 | 27 |
| 3 | 27 | 26h30 | 33 |
| 4 | 29 | 27 | 40h |
| Total | 110h | 114h30 | 146h |

Temps total: 370,5h = 114h30+ 110h + 146h

# Outil utilisé pour la gestion des stories

Trello

# Problèmes principaux rencontrés

Apprentissage difficile, mais courte durée

Apprendre le code déjà existant

Courbe d’apprentissage élevé

WampServer

Gestion des différentes versions du projet

Implantation des parties de chacun dans un code commun.

Solution : utiliser une alternative à google docs pour la gestion des différentes versions du programme et l’implantation des différentes parties dans un programme commun. On aurait dû utiliser Github.

**Rétrospection des sprints**

Sprint 1 : Le sprint a commencé tranquillement, car nous avons pris du temps pour prendre en main le code, et apprendre à travailler en équipe. De plus, nous choisie de partager nos fichiers avec Google drive et faire l'intégration manuellement. Ce premier sprint nous a aussi permis de voir que notre capacité estimée était beaucoup trop grande pour notre vélocité, nous avons aussi modifié les sprints en conséquence.

Sprint 2 : le deuxième, sprint a été très productif, car nous n'avons pas rencontré de gros problème cela nous a permis de nous rapprocher de notre nombre de points visé sans. Les ajustements faits sur la répartition des tâches nous ont permis de mieux arriver dans nos temps et de finaliser nos fonctionnalités dans les temps.

Sprint 3 : Nous avons dû sacrifier plusieurs stories prévues dans ce sprint, car nous avons vu la release approcher et nous devions mettre au propre la documentation en plus de régler plusieurs bogues et retirer les fonctionnalités non finies du sprint 4. De nombreuses tâches non pas pu être terminées.

# Schéma de la base donnée

