

GiselleMagicArts.com

Le site de vente de la sorcière Giselle

Rapport de projet



Table des matières

I.	Présentation du projet	4
1.	Contexte	4
2.	Besoins	5
3.	L'équipe projet	5
II.	Planification du projet	6
1.	Découpage du projet : le WBS.....	6
2.	Planning prévisionnel	6
3.	Planning effectif	9
III.	Mise en place de la solution proposée	12
1.	Modélisation : la méthode MERISE.....	12
a.	Modèle conceptuel de données (MCD).....	12
b.	Modèle logique de données (MLD)	16
c.	Choix du SGBDR	17
d.	Modèle physique de données (MPD)	17
2.	Les requêtes SQL	18
a.	Remplissage de la base de données	18
b.	Procédures stockées	18
c.	Soumission de recettes.....	21
d.	Rechercher et consulter.....	22
e.	Sauvegarder	25
3.	Retour sur expérience	27
a.	Synthèse des problèmes rencontrés	27
b.	Bilan de l'équipe.....	27
c.	Bilans individuels	28
	ANNEXE 1 : Créer un serveur MySQL sur un Raspberry Pi	30
	ANNEXE 2 : Script de création de la base de données	31
	ANNEXE 3 : Ressources.....	35

Table des figures

Figure 1 - WBS du projet	7
Figure 2 - Répartition des tâches prévisionnelles	8
Figure 3 - Diagramme de Gantt prévisionnel	9
Figure 4 - Répartition des tâches effective	10
Figure 5 - Diagramme de Gantt effectif	11
Figure 6 - Dictionnaire de données (DD)	13
Figure 7 - Matrice des dépendances fonctionnelles	14
Figure 8 - Modèle Conceptuel de données	15
Figure 9 - Modèle logique de données	16
Figure 10 - Logo de MySQL.....	17
Figure 11 - Procédure d'affichage et de mise à jour du stock des ingrédients	19
Figure 12 - Procédure affichant l'historique des commandes d'un client.....	19
Figure 13 - Procédure de suppression d'un ingrédient arrivé à expiration.....	20
Figure 14 - Procédure de modification du stock après l'envoi d'une commande.....	20
Figure 15 – Procédure d’ajout d’une proposition de recette.....	21
Figure 16 - Résultat de la procédure de suppression d'une proposition qui n'a pas été retenue	22
Figure 17 - Procédure listant les préparations réalisables avec un ingrédient précis.....	23
Figure 18 - Procédure d'affichage des couples (potions,onguents) utilisant la même liste d'ingrédients	23
Figure 19 - Procédure affichant les potions réalisables avec un ingrédient précis.....	24
Figure 20 - Procédure listant les potions classées par température de préparation.....	24
Figure 21 - Procédure affichant la marge que dégage Giselle pour chaque préparation	25
Figure 22 - Procédure donnant le nombre moyen d'ingrédients utilisés pour fabriquer une préparation	25

I. Présentation du projet

1. Contexte

Giselle, jeune sorcière ayant obtenu récemment son permis de magie souhaite se lancer dans la vente de potions et d'ingrédients. Elle souhaite donc réaliser un site e-commerce de magie : gisellemagicarts.com.

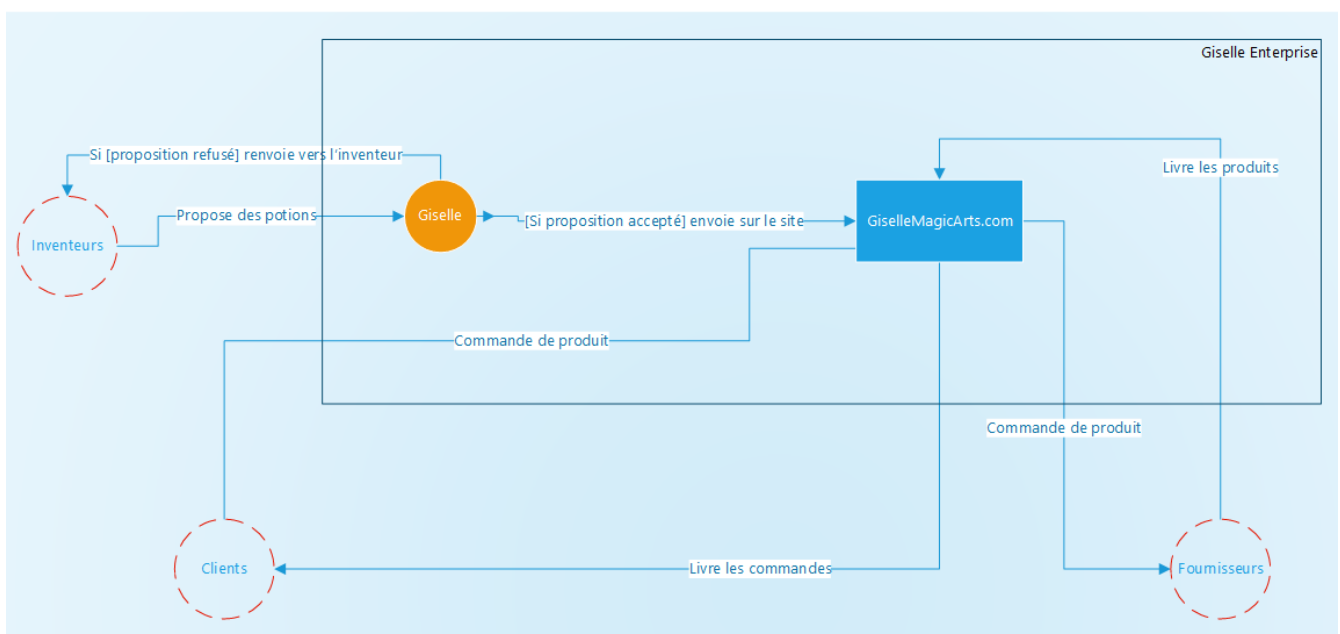
La sorcière a donc besoin d'une interface web ainsi que d'une base de données pour stocker toutes les informations nécessaires au bon fonctionnement de son site. Sur ce dernier sera vendu :

- Des potions
- Des onguents
- Des ingrédients (pour préparer des potions et des onguents)

Pour expliquer le fonctionnement du site GiselleMagicArts nous avons décidé d'établir un diagramme de flux. Le but est de bien comprendre le fonctionnement du futur site pour développer un modèle contextuel de données adaptées.

Tout d'abord listons quelques informations complémentaires concernant le fonctionnement du site :

- Giselle ne se procure pas elle-même les ingrédients mais fait appel à des fournisseurs.
- Les clients commandent puis sont livrés
- De nouveaux inventeurs peuvent proposer des recettes de nouvelles potions. Giselle choisit ensuite si elle souhaite ajouter les propositions de potions au site ou non.



Les acteurs externes sont représentés par des cercles à traits pointillés, les acteurs internes par des cercles à traits pleins et les flux par des flèches de l'émetteur vers le récepteur.

2. Besoins

Un ami de Giselle, Miguel se charge de réaliser le site web. Ce projet consistera donc en la création d'une base de données et des requêtes nécessaires au bon fonctionnement de ce site.

Giselle souhaite surveiller de manière consciencieuse la fraîcheur de chaque ingrédient. En effet, la fraîcheur d'un ingrédient peut avoir un effet sur la qualité d'une potion. Elle souhaite également que les ingrédients qui ne sont plus frais soient automatiquement supprimés de la base de données. De plus les clients peuvent préciser la fraîcheur souhaitée des ingrédients qu'ils commandent (un même ingrédient peut être présent deux fois dans une commande mais avec des fraîcheurs différentes).

Les clients, lorsqu'ils commandent des potions, doivent choisir dans quel récipient celle-ci sera contenu (fioles, pots...). Le prix variera en fonction du type de récipient.

Il faudra préparer un certain nombre de requêtes en langage SQL pour extraire de la base de données des informations clés pour la sorcière.

Giselle veut mettre en place un système de proposition de recettes. Les clients pourront proposer de nouvelles recettes et Giselle choisira si elle décide de les intégrer ou non dans son catalogue.

Le dernier point important pour la base de données de Giselle est que celle-ci doit être sauvegardée. Ainsi, elle pourra être facilement restaurer en cas d'éventuel problèmes ou de migration.

3. L'équipe projet

L'équipe chargée de réaliser le projet de Giselle est composée de quatre membres :

- Florian FRITSCH
- Aurélien KLEIN
- Pierre GEERAERT
- Alexandre LECOMTE (chef de projet)

Dans la suite de ce rapport, « nous » désignera l'équipe du projet présentée ci-dessus.

Nous disposons de 7 jours ouvrés pour une équipe de quatre personnes soit 28 jours homme pour réaliser une base de données fonctionnelle et répondant aux besoins du futur site web de la sorcière Giselle.

II. Planification du projet

1. Découpage du projet : le WBS

Afin de se répartir correctement les tâches et avoir une vision globale du travail pour réaliser le projet nous avons décidé de définir un Work Breakdown Structure (WBS).

Le WBS est défini par un « découpage hiérarchique en livrables spécifiques des travaux à exécuter ». On trouve plusieurs niveaux dans un WBS, chaque niveau détaille un peu plus les tâches à réaliser jusqu'à arriver au niveau final. L'idée est d'obtenir un diagramme facilement compréhensible reprenant tout ce que l'équipe du projet doit réaliser.

Le dernier niveau doit correspondre à une tâche que l'on peut attribuer à une ou plusieurs personnes de l'équipe.

Après avoir clarifié le contexte et défini le périmètre du projet dans la partie précédente, nous avons donc découpé le travail de la manière représenté par la figure 1 de la page suivante.

2. Planning prévisionnel

A partir de ce WBS nous avons établi une répartition des tâches (figure 2) ainsi qu'une estimation de la durée de celles-ci.

A partir du tableau de répartition des tâches nous avons construit un diagramme de Gantt correspondant (figure 3).

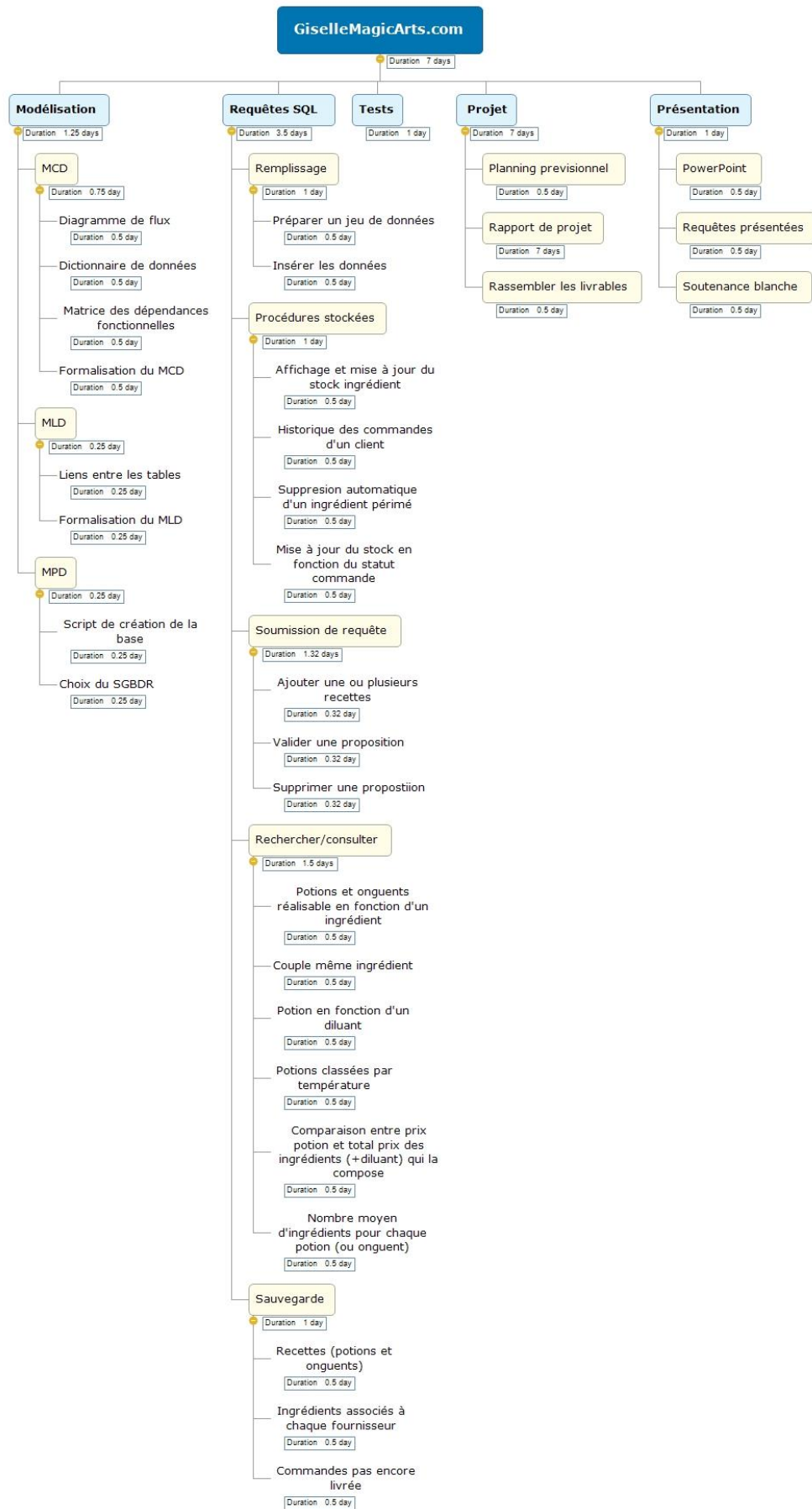


Figure 1 - WBS du projet

	Task Name	Duration	Start	End	Completion	Resources
1	GiselleMagicArts.com	7 days	3/29/2017	4/6/2017	0%	Alexandre;Pierre;Florian;Aurelien
2	<input type="checkbox"/> Modélisation	1.25 days	3/29/2017	3/30/2017	0%	Alexandre;Pierre;Florian;Aurelien
3	<input type="checkbox"/> MCD	0.75 day	3/29/2017	3/29/2017	0%	Alexandre;Pierre;Florian;Aurelien
4	Diagramme de flux	0.5 day	3/29/2017	3/29/2017	0%	Pierre
5	Dictionnaire de donnés...	0.5 day	3/29/2017	3/29/2017	0%	Alexandre;Florian;Aurelien
6	Matrice des dépenda...	0.5 day	3/29/2017	3/29/2017	0%	Pierre;Aurelien
7	Formalisation du MCD	0.5 day	3/29/2017	3/29/2017	0%	Alexandre;Aurelien
8	<input type="checkbox"/> MLD	0.25 day	3/30/2017	3/30/2017	0%	Aurelien;Alexandre
9	Liens entre les tables	0.25 day	3/30/2017	3/30/2017	0%	Alexandre
10	Formalisation du MLD	0.25 day	3/30/2017	3/30/2017	0%	Aurelien
11	<input type="checkbox"/> MPD	0.25 day	3/30/2017	3/30/2017	0%	Florian;Aurelien
12	Script de création de ...	0.25 day	3/30/2017	3/30/2017	0%	Aurelien
13	Choix du SGBDR	0.25 day	3/30/2017	3/30/2017	0%	Florian
14	<input type="checkbox"/> Requêtes SQL	3.5 days	3/30/2017	4/4/2017	0%	Alexandre;Pierre;Florian;Aurelien
15	<input type="checkbox"/> Remplissage	1 day	3/30/2017	3/31/2017	0%	Pierre;Florian
16	Préparer un jeu de do...	0.5 day	3/30/2017	3/30/2017	0%	Pierre;Florian
17	Insérer les données	0.5 day	3/31/2017	3/31/2017	0%	Pierre;Florian
18	<input type="checkbox"/> Procédures stockées	1 day	3/31/2017	4/3/2017	0%	
19	Affichage et mise à j...	0.5 day	3/31/2017	3/31/2017	0%	Pierre
20	Historique des comm...	0.5 day	3/31/2017	3/31/2017	0%	Florian
21	Suppresion automatiq...	0.5 day	3/31/2017	3/31/2017	0%	Florian
22	Mise à jour du stock ...	0.5 day	4/3/2017	4/3/2017	0%	Pierre;Florian
23	<input type="checkbox"/> Soumission de requ...	1.32 days	3/31/2017	4/3/2017	0%	
24	Ajouter une ou plusie...	0.32 day	3/31/2017	3/31/2017	0%	Alexandre;Aurelien
25	Valider une proposition	0.32 day	4/3/2017	4/3/2017	0%	Alexandre
26	Supprimer une propo...	0.32 day	4/3/2017	4/3/2017	0%	Aurelien
27	<input type="checkbox"/> Rechercher/consulter	1.5 days	4/3/2017	4/4/2017	0%	
28	Potions et onguents r...	0.5 day	4/3/2017	4/3/2017	0%	Alexandre
29	Couple même ingrédi...	0.5 day	4/3/2017	4/3/2017	0%	Florian
30	Potion en fonction d'u...	0.5 day	4/3/2017	4/3/2017	0%	Pierre
31	Potions classées par ...	0.5 day	4/3/2017	4/3/2017	0%	Aurelien
32	Comparaison entre pr...	0.5 day	4/4/2017	4/4/2017	0%	Pierre;Alexandre
33	Nombre moyen d'ingr...	0.5 day	4/4/2017	4/4/2017	0%	Florian;Aurelien
34	<input type="checkbox"/> Sauvegarde	1 day	4/4/2017	4/4/2017	0%	
35	Recettes (potions et ...	0.5 day	4/4/2017	4/4/2017	0%	Alexandre;Florian
36	Ingédients associés ...	0.5 day	4/4/2017	4/4/2017	0%	Aurelien
37	Commandes pas enc...	0.5 day	4/4/2017	4/4/2017	0%	Pierre
38	Tests	1 day	4/5/2017	4/5/2017	0%	Alexandre;Pierre;Florian;Aurelien
39	<input type="checkbox"/> Projet	7 days	3/29/2017	4/6/2017	0%	Alexandre;Pierre;Florian;Aurelien
40	Planning previsionnel	0.5 day	3/29/2017	3/29/2017	0%	Alexandre;Pierre;Florian;Aurelien
41	Rapport de projet	7 days	3/29/2017	4/6/2017	0%	Alexandre
42	Rassembler les livrables	0.5 day	4/6/2017	4/6/2017	0%	Alexandre;Pierre;Florian;Aurelien
43	<input type="checkbox"/> Présentation	1 day	4/6/2017	4/6/2017	0%	Alexandre;Pierre;Florian;Aurelien
44	PowerPoint	0.5 day	4/6/2017	4/6/2017	0%	Aurelien
45	Requêtes présentées	0.5 day	4/6/2017	4/6/2017	0%	Pierre
46	Soutenance blanche	0.5 day	4/6/2017	4/6/2017	0%	Florian;Alexandre;Pierre;Aurelien

Figure 2 - Répartition des tâches prévisionnelles

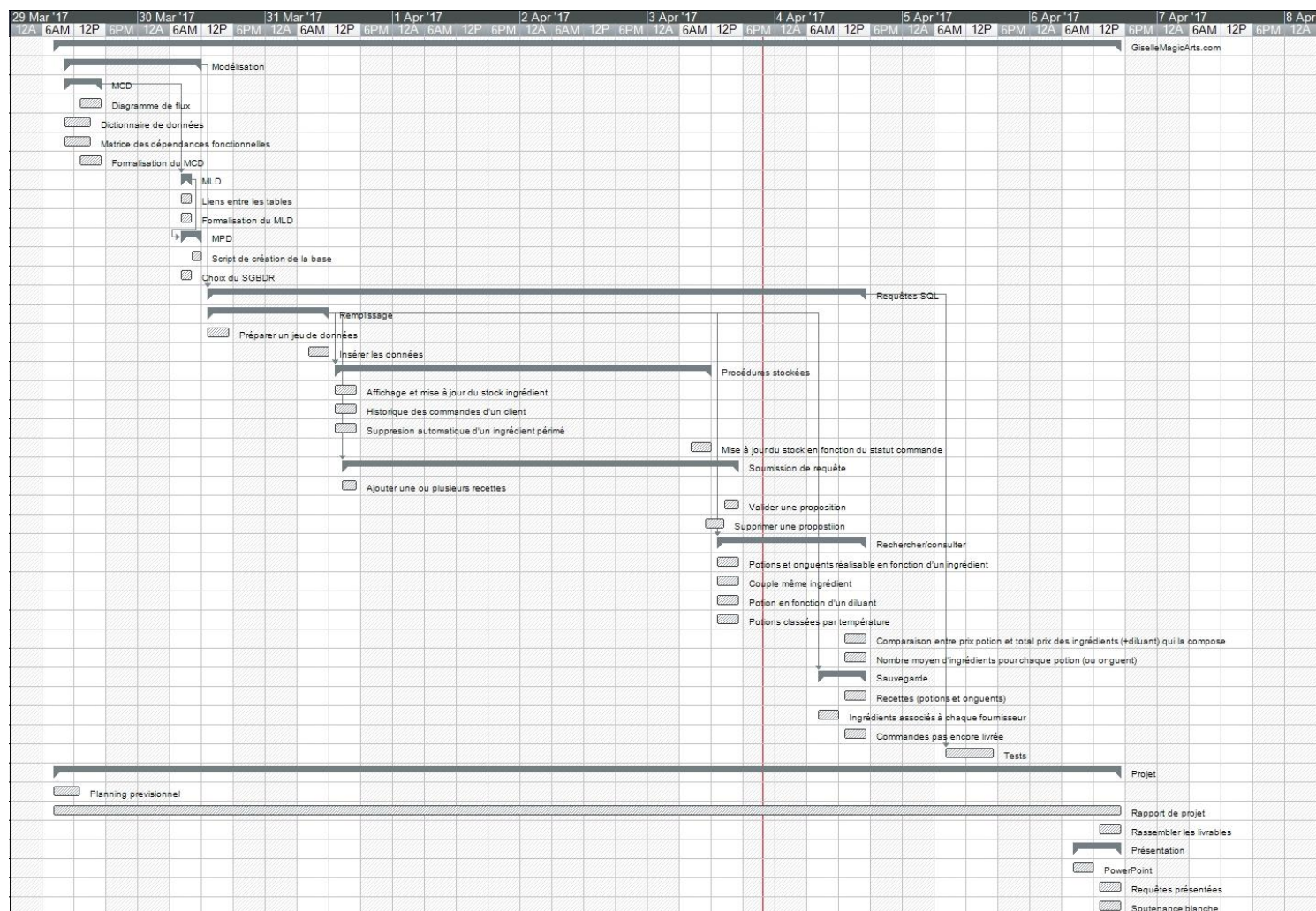


Figure 3 - Diagramme de Gantt prévisionnel

3. Planning effectif

Au bout des 7 jours du projet, nous avons respecté les délais. Tous les livrables sont prêts et ont été soumis à la MOA (Giselle la sorcière).

La figure 4 présente la répartition des tâches effectives et la figure 5 le diagramme de Gantt effectif. Observons les différences par rapport aux prévisions effectuées au début du projet.

La première différence notable concerne l'enchaînement de la méthode MERISE. En effet, nous on peut voir que la conception du MPD à commencer avant la fin de la conception du MCD. Cela provient du fait que la version finale du MCD est arrivée tard dans le déroulement du projet.

De plus, nous avons modifié les ressources humaines attribués à la conception de certaines requêtes car celles-ci demandées plus de temps que prévu ou au contraire moins de temps que prévu.

Dans l'ensemble le planning prévisionnel correspond au planning effectif.

	Task Name	Duration	Start	End	Completion	Resources
1	GiselleMagicArts.com	7 days	3/29/2017	4/6/2017	100%	Alexandre;Pierre;Florian;Aurelien
2	<input type="checkbox"/> Modélisation	6.75 days	3/29/2017	4/6/2017	100%	Alexandre;Pierre;Florian;Aurelien
3	<input type="checkbox"/> MCD	2.24 days	3/29/2017	3/31/2017	100%	Alexandre;Pierre;Florian;Aurelien
4	Diagramme de flux	0.25 day	3/29/2017	3/29/2017	100%	Pierre
5	Dictionnaire de données	0.5 day	3/29/2017	3/29/2017	100%	Alexandre;Florian;Aurelien
6	Matrice des dépendances fonctionnelles	0.25 day	3/29/2017	3/29/2017	100%	Pierre;Aurelien
7	Formalisation du MCD	1.5 days	3/29/2017	3/31/2017	100%	Alexandre;Aurelien
8	<input type="checkbox"/> MLD	0.25 day	3/31/2017	3/31/2017	100%	Aurelien;Alexandre
9	Liens entre les tables	0.25 day	3/31/2017	3/31/2017	100%	Alexandre
10	Formalisation du MLD	0.25 day	3/31/2017	3/31/2017	100%	Aurelien
11	<input type="checkbox"/> MPD	0.25 day	3/31/2017	3/31/2017	100%	Florian;Aurelien
12	Script de création de la base	0.25 day	3/31/2017	3/31/2017	100%	Aurelien
13	Choix du SGBDR	0.25 day	3/31/2017	3/31/2017	100%	Florian
14	<input type="checkbox"/> Requêtes SQL	3 days	3/30/2017	4/4/2017	100%	Alexandre;Pierre;Florian;Aurelien
15	<input type="checkbox"/> Remplissage	1 day	3/31/2017	3/31/2017	100%	Pierre;Florian;Alexandre;Aurelien
16	Préparer un jeu de données	0.5 day	3/31/2017	3/31/2017	100%	Pierre;Florian;Alexandre;Aurelien
17	Insérer les données	0.5 day	3/31/2017	3/31/2017	100%	Pierre;Aurelien
18	<input type="checkbox"/> Procédures stockées	2 days	3/30/2017	4/3/2017	100%	
19	Affichage et mise à jour du stock ingrédient	0.5 day	3/30/2017	3/31/2017	100%	Pierre
20	Historique des commandes d'un client	0.5 day	3/31/2017	3/31/2017	100%	Florian
21	Suppression automatique d'un ingrédient périmé	0.5 day	3/31/2017	4/3/2017	100%	Pierre
22	Mise à jour du stock en fonction du statut commande	0.5 day	4/3/2017	4/3/2017	100%	Florian
23	<input type="checkbox"/> Soumission de requête	0.32 day	4/3/2017	4/3/2017	100%	
24	Ajouter une ou plusieurs recettes	0.32 day	4/3/2017	4/3/2017	100%	Alexandre;Aurelien
25	Valider une proposition	0.32 day	4/3/2017	4/3/2017	100%	Alexandre
26	Supprimer une proposition	0.32 day	4/3/2017	4/3/2017	100%	Aurelien
27	<input type="checkbox"/> Rechercher/consulter	1.5 days	4/3/2017	4/4/2017	100%	
28	Potions et onguents réalisable en fonction d'un ingrédient	0.5 day	4/3/2017	4/3/2017	100%	Alexandre
29	Couple même ingrédient	1.5 days	4/3/2017	4/4/2017	100%	Florian, Alexandre
30	Potion en fonction d'un diluant	0.5 day	4/3/2017	4/4/2017	100%	Pierre
31	Potions classées par température	0.5 day	4/3/2017	4/3/2017	100%	Aurelien
32	Comparaison entre prix potion et total prix des ingrédients...	0.5 day	4/3/2017	4/4/2017	100%	Alexandre
33	Nombre moyen d'ingrédients pour chaque potion (ou on...	0.5 day	4/3/2017	4/4/2017	100%	Aurelien
34	<input type="checkbox"/> Sauvegarde	1 days	4/3/2017	4/4/2017	100%	
35	Recettes (potions et onguents)	0.5 day	4/3/2017	4/4/2017	100%	Pierre
36	Ingrédients associés à chaque fournisseur	0.5 day	4/4/2017	4/4/2017	100%	Alexandre
37	Commandes pas encore livrées	0.5 day	4/4/2017	4/4/2017	100%	Pierre
38	Tests	1 day	4/5/2017	4/5/2017	100%	Alexandre;Pierre
39	Améliorations	1 day	4/5/2017	4/5/2017	100%	Aurelien, Florian
40	<input type="checkbox"/> Projet	7 days	3/29/2017	4/6/2017	100%	Alexandre;Pierre;Florian;Aurelien
41	Planning prévisionnel	0.5 day	3/29/2017	3/29/2017	100%	Alexandre;Pierre;Florian;Aurelien
42	Rapport de projet	7 days	3/29/2017	4/6/2017	100%	Alexandre
43	Rassembler les livrables	0.5 day	4/6/2017	4/6/2017	100%	Alexandre;Pierre;Florian;Aurelien
44	<input type="checkbox"/> Présentation	1 day	4/6/2017	4/6/2017	100%	Alexandre;Pierre;Florian;Aurelien
45	PowerPoint	0.5 day	4/6/2017	4/6/2017	100%	Aurelien
46	Requêtes présentées	0.5 day	4/6/2017	4/6/2017	100%	Pierre
47	Soutenance blanche	0.5 day	4/6/2017	4/6/2017	100%	Florian;Alexandre;Pierre;Aurelien

Figure 4 - Répartition des tâches effective

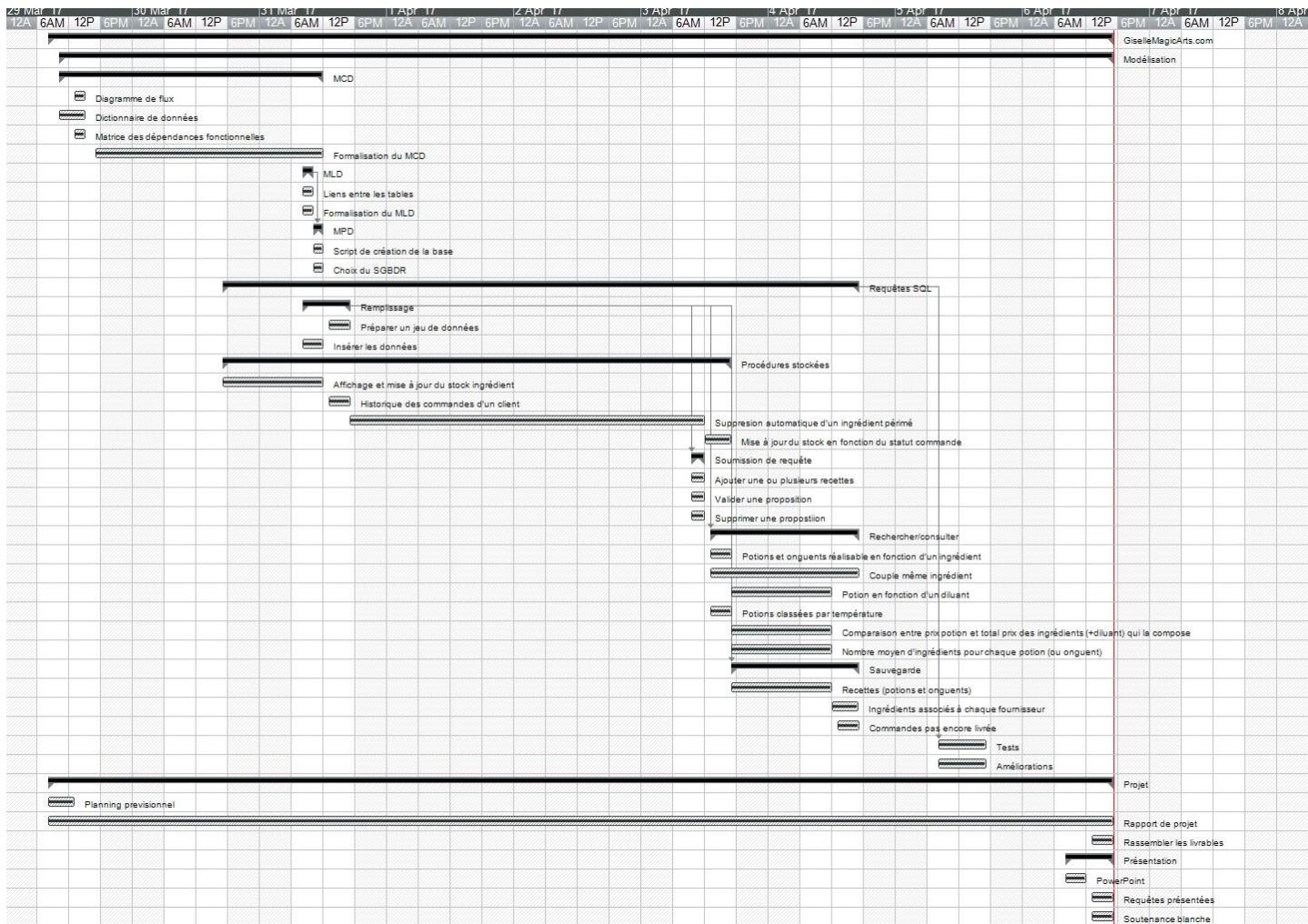


Figure 5 - Diagramme de Gantt effectif

III. Mise en place de la solution proposée

1. Modélisation : la méthode MERISE

Merise a principalement été utilisée en France dans les administrations publiques ou privés et par certaines entreprises de services du numérique. La méthode Merise est une méthode d'analyse, de conception et de réalisation de systèmes d'informations.

La méthode Merise se caractérise par ces différents axes :

- Identifier les acteurs du projet
- Réaliser le schéma directeur (objectifs/stratégie de l'entreprise, cadre du projet)
- Faire l'étude préalable (description des besoins et des attentes des utilisateurs, ébauche des divers modèles)
- Faire l'étude détaillée
- Réalisation de l'étude technique (moyens techniques nécessaires)
- Mise en production
- Gestion de la maintenance (informations sur les acteurs, les documentations et les formations)

La méthode MERISE se déroule en trois grandes étapes :

- Le modèle conceptuel de données (MCD) : Quelles relations lient les données entre elles ?
- Le modèle logique de données (MLD) : Quelles sont les liens qui découlent de ces relations ?
- Le modèle physique de données (MPD) : C'est l'implémentation en langage SQL (Structured Query Language) de la base de données pour un système de gestion choisit.

Nous avons décidé de choisir la méthode Merise pour sa fiabilité, sa rapidité et son efficacité. De plus, c'est sur cette méthode que les membres de l'équipe ont été formés.

Pour formaliser les différentes étapes de la méthode Merise nous avons choisi le logiciel JMERISE, distribué gratuitement au téléchargement. Ce logiciel permet de formaliser un MCD puis de le traduire automatiquement en MLD puis en script SQL (MPD). La conversion des différents modèles est rapide et efficace. En outre, JMERISE gère la version 2 de Merise 2 ; nous permettant notamment d'implémenter la notion d'héritage dans notre MCD.

a. Modèle conceptuel de données (MCD)

Pour pouvoir construire le modèle conceptuel de données¹ on procède en deux étapes :

- La création d'un dictionnaire de données (DD)
- La création d'une matrice des dépendances fonctionnelles (Matrice des DF)

¹ Tous les documents concernant la modélisation de la base de données sont disponibles dans le dossier modélisation sur le GitHub du projet.

Le dictionnaire de données ou recueil d'informations consiste à collecter les données. Nous avons donc répertorié l'ensemble des données que nous jugions nécessaires en analysant le sujet du projet.

Nous aboutissons au dictionnaire de données suivant :

Nom	Description	Type	Contraintes
IDPersonne	Identifiant d'une personne (client et/ou inventeur)	Numérique	Obligatoire
NomPersonne	Nom de la personne	Alphabétique	Obligatoire
PrenomPersonne	Prenom de la personne	Alphabétique	Obligatoire
AdressePersonne	Lieu d'habitation d'une personne	varchar	Obligatoire
IDFournisseur	Identifiant du fournisseur	Numérique	Obligatoire
NomFournisseur	Nom du fournisseur	Alphanumérique	Obligatoire
NumeroFournisseur	Numéro de téléphone du fournisseur	Numérique	Obligatoire
IDDiluant	Identifiant du diluant	Numérique	Obligatoire
NomDiluant	Nom du diluant	Alphabétique	Obligatoire
PrixDiluant	Prix du diluant	Numérique	Obligatoire
IDRecipient	Identifiant du récipient	Numérique	Obligatoire
NomRecipient	Nom du récipient	Alphabétique	Obligatoire
StockRecipients	Quantité disponible de récipients en stock	Numérique	Obligatoire, >= 0
PrixRecipient	Prix du récipient	Décimale	Obligatoire
IDProduit	Identifiant d'un produit (ingrédient, potions et onguents)	Numérique	Obligatoire
NomProduit	Nom de l'ingrédient, de la potion ou de l'onguent	Alphabétique	Obligatoire
PrixProduit	Prix de l'ingrédient, de la potion ou de l'onguent	Décimale	Obligatoire
IDPreparation	Une préparation est une potion ou un onguent	Alphabétique	Obligatoire
TemperaturePreparation	Température d'une préparation si c'est une potion	Numérique	
IDrecettePreparation	Identifiant d'une recette d'un préparation	Numérique	Obligatoire
IDIngredient	Identifiant d'un ingrédient dans la recette	Numérique	Obligatoire
QuantiteIngredient	Quantité d'ingrédient dans la recette	Numérique	
FraicheurMax	Fraicheur max d'un ingrédient dans une recette	Numérique	
FraicheurMin	Fraicheur minimale requise pour la préparation	Numérique	
Temps	Temps de cuisson pour chaque ingrédient de la recette	Numérique	
IDCommande	Numéro de commande	Numérique	Obligatoire
FraicheurCommande	Fraicheur désirée par le client	Numérique	
QuantiteCommande	Quantité de la commande	Numérique	Obligatoire
DateCommande	Date de la commande	date	Obligatoire
StatutCommande	Statut de la commande	Alphabétique	Obligatoire
PrixTotalCommande	Prix total de la commande	Décimale	Obligatoire
IDProposition	Identifiant de la proposition d'une nouvelle recette	Numérique	Obligatoire
NomProposition	Nom de la proposition	Alphabétique	Obligatoire
TemperatureProposition	Température de préparation de la recette proposée	Numérique	Obligatoire
IDDiluantProposition	Diluant dans la recette proposée si c'est une potion	Numérique	
FraicheurMaxProposee		Numérique	
FraicheurMinProposee		Numérique	
QuantiteIngredientProposition		Numérique	Obligatoire
IDstock	Identifiant d'un stock (plusieurs possible pour un même ingrédient)	Numérique	Obligatoire
NbStock	Quantité de stock disponible	Numérique	Obligatoire, >= 0
FraicheurIngredient	Degré de fraicheur (en jour)	Numérique	Obligatoire, >= 0

Figure 6 - Dictionnaire de données (DD)

La deuxième phase de conception du MCD est l'analyse des dépendances. Les dépendances sont les liens qui lient les différentes données. Le document résultant de cette étape est un diagramme ou une matrice des dépendances.

A partir du dictionnaire précédant on construit la matrice des dépendances fonctionnelles suivantes :

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44			
1 IDPersonne																																															
2 NomPersonne	X																																														
3 PrenomPersonne	X																																														
4 AdressePersonne	X																																														
5 IDFournisseur																																															
6 NomFournisseur																																															
7 NumeroFournisseur																																															
8 IDDiluant																																															
9 NomDiluant																																															
10 PrixDiluant																																															
11 IDRecipient																																															
12 NomRecipient																																															
13 StockRecipients																																															
14 PrixRecipient																																															
15 IDProduit																																															
16 NomProduit																																															
17 PrixProduit																																															
18 IDPreparation																																															
19 TemperaturePreparation																																															
20 IDrecettePreparation																																															
21 IDIngredient																																															
22 QuantiteIngredient																																															
23 FraicheurMax																																															
24 FraicheurMin																																															
25 Temps																																															
26 IDCommande																																															

Figure 7 - Matrice des dépendances fonctionnelles

A l'aide de ces deux documents (DD et matrice des DF) on établit le modèle conceptuel de données sur le logiciel JMERISE.

On obtient le MCD suivant :

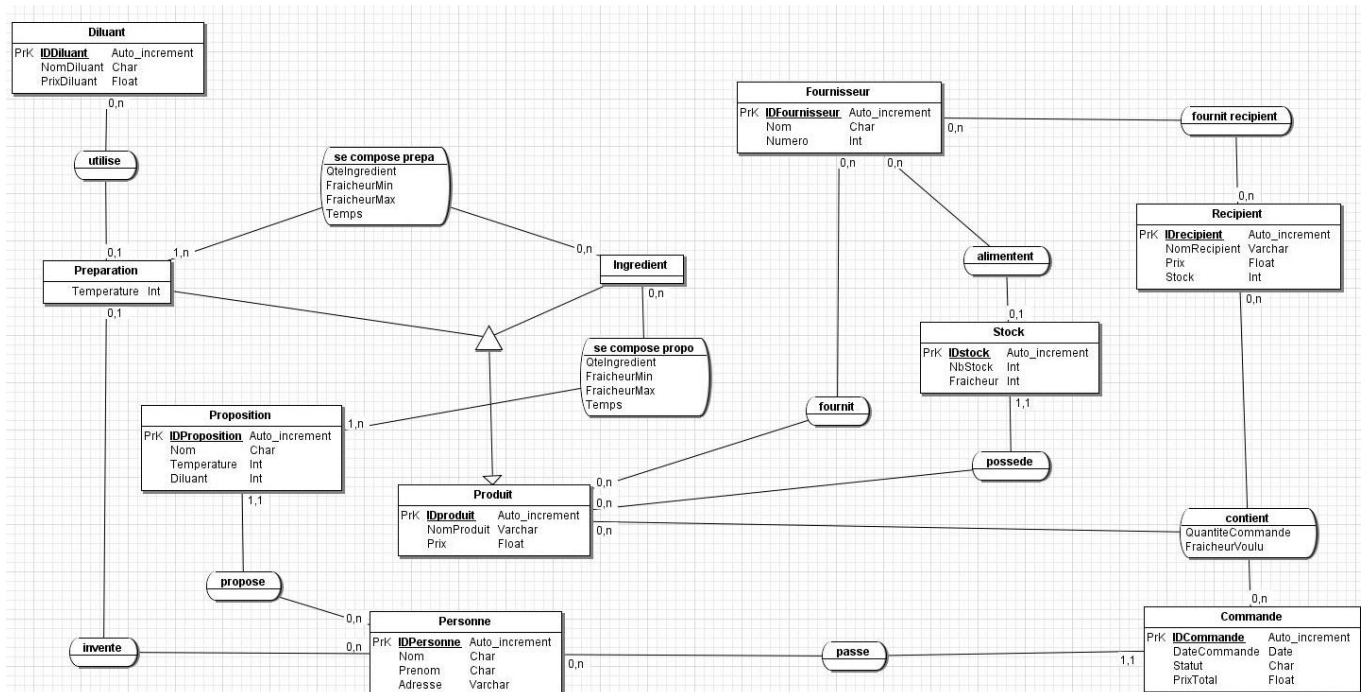


Figure 8 - Modèle Conceptuel de données

La figure 6 présente le modèle conceptuel de données de la future base de données du site GiselleMagicArts.com.

Tout d'abord, l'entité produit rassemble tout ce que la sorcière Giselle vend sur son site ; c'est-à-dire potions, onguents et des ingrédients seuls. Chacun de ces produits sont identifiés de manière unique par un numéro (un peu comme une référence). Cette entité produit est l'entité mère d'une relation d'héritage dans laquelle les deux entités filles sont :

- Ingrédient : entité contenant les références des produits qui sont des ingrédients
- Préparation : entité contenant les références des produits qui sont des préparations (onguents et potions)

Ainsi, l'entité préparation contient aussi bien les onguents que les potions. Pour identifier les onguents des potions et vice versa il suffira de distinguer les préparations qui utilisent un diluant (potions) et celles qui n'en utilisent pas (onguents).

Une préparation utilise un ou aucun diluant (cas des onguents).

L'entité personne rassemble les clients mais aussi les inventeurs. En effet, un inventeur peut aussi être un client et inversement. C'est pour cette raison que la cardinalité du côté de l'entité personne de l'association invente est 0,n. Une personne peut inventer plusieurs ou aucune préparation. L'autre cardinalité de cette association montre que pour une préparation on ne connaît pas toujours l'inventeur.

Une personne passe des commandes qui contiennent des produits ainsi que des réipients. En effet, pour chaque produit commandé il faudra indiquer un réipient pour le transport et sa conservation.

Les personnes peuvent également proposer des idées de préparations qui sont composées de produits (Ingrédients, potions et onguents). De ce fait, il sera possible de créer une potion dans laquelle un des ingrédients est une autre potion ou un onguent.

L'entité fournisseur rassemble l'ensemble des fournisseurs que Giselle sollicitera. Ainsi, les fournisseurs fournissent des produits ; c'est-à-dire des ingrédients mais aussi des potions et des onguents. De cette manière, si Giselle souhaite par la suite acheter des potions ou des onguents directement aux fournisseurs il ne sera pas nécessaire de modifier le MCD. Les fournisseurs peuvent également fournir des récipients. Tout en sachant qu'un même type de récipient ou un même ingrédient peut être fournit par plusieurs fournisseurs différents. Ils alimentent donc un stock de produits.

Un identifiant de stock représente un ingrédient avec une fraîcheur précise. Par conséquent, un même ingrédient peut se retrouver plusieurs fois dans le stock mais avec des fraîcheurs différentes.

b. Modèle logique de données (MLD)

Maintenant nous devons transformer le MCD en modèle logique de données (MLD). La quatrième étape de l'analyse des données est donc la traduction du MCD en MLD. Cette traduction s'effectue en suivant des règles précises.

Le MLD est toujours indépendant du SGBD utilisé. Il est composé uniquement de relations. Une relation est composée d'attributs et possède systématiquement une et une seule clé primaire qui permet d'identifier sans ambiguïté chaque occurrence de cette relation.

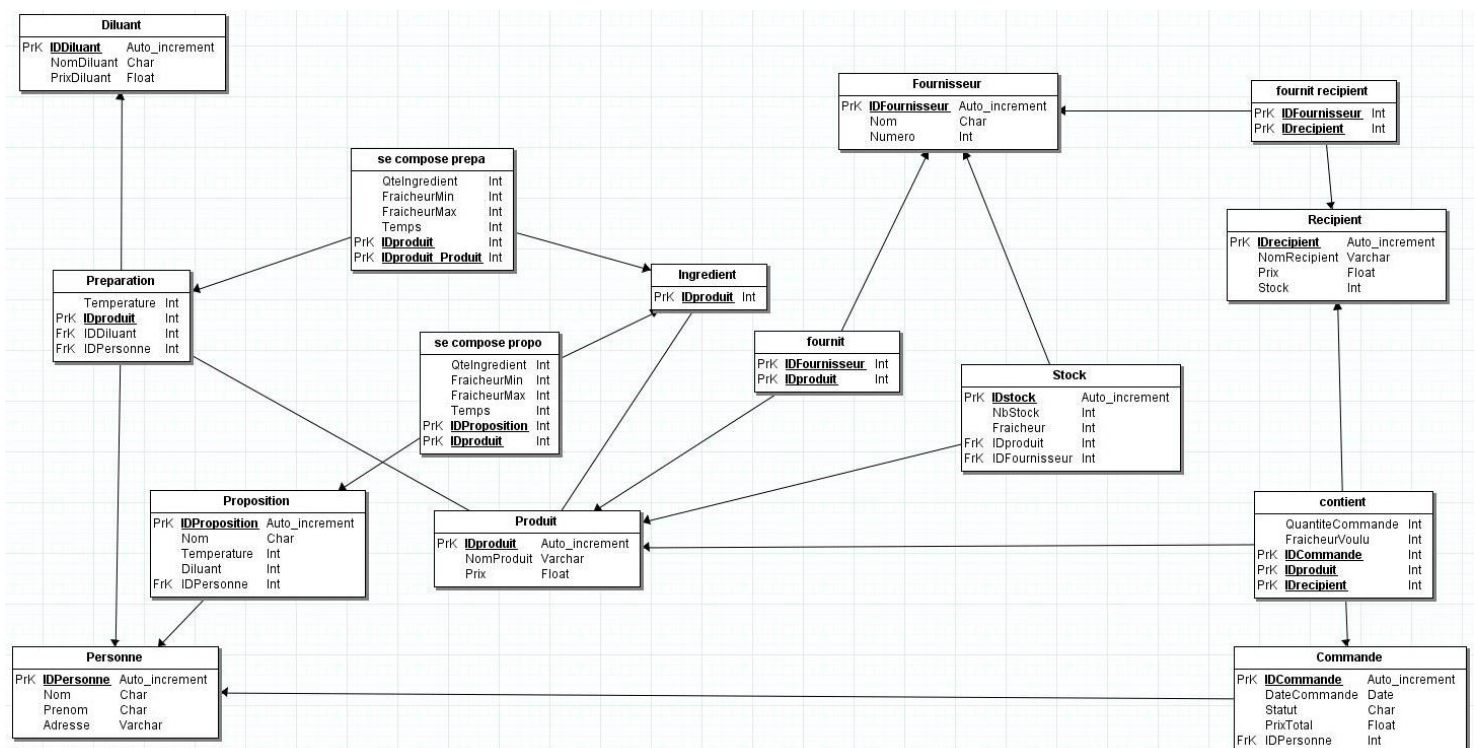


Figure 9 - Modèle logique de données

c. Choix du SGBDR

Nous avons choisi le Système de Gestion de Base de Données Relationnel (SGBDR) : MySQL. Comme la majeure partie des SGBDR, il utilise le langage SQL.

Dans un premier temps, nous avons été formés à l'école sur MySQL. Cela nous a permis d'avoir plus d'aisance dans la création de la base de données sur ce SGBDR.



Figure 10 - Logo de MySQL

De plus MySQL est gratuit et disponible sur plusieurs systèmes d'exploitation. Il est multi-utilisateur, cela nous permet donc de travailler tous ensemble sur la même base de données. Pour cela nous avons donc hébergé notre serveur MySQL sur un Raspberry Pi 3².

MySQL dispose également de fonctionnalité de sécurité. Ce SGBDR contient un système de privilèges qui nous permet de gérer les différents accès sur ce serveur et il dispose d'un système de sauvegardes.

d. Modèle physique de données (MPD)

Le modèle physique de données est l'implémentation du MLD par un logiciel. On va devoir traduire le MLD pour obtenir un MPD. On va notamment préciser le type et la taille (octets, bits) des données pour savoir comment les stocker.

La traduction d'un MLD relationnel en MPD passe par la création d'une base de données hébergée par un SGBDR (ici MySQL).

Le MPD est donc le script de création de l'ensemble des tables et des relations entre elles en langage SQL.

Vous trouverez le fichier script_creation_bdd dans l'annexe 2 ainsi que sur le dépôt GitHub du projet.

² Pour plus d'informations sur la création du serveur MySQL sur un Raspberry Pi 3 se reporter à l'annexe 1

2. Les requêtes SQL

Maintenant que la structure de la base de données est valide et implémentée, nous devons insérer des données et réaliser un certain nombre de traitements que nous aborderons dans cette partie.

a. Remplissage de la base de données

Afin d'insérer un jeu de données cohérent et significatif nous avons décidé de procéder de cette manière :

- 1^{ère} étape : Créer un classeur Excel dans lequel chaque feuille représente une table de la base de données. On ajoute des données dans ces tableaux.
- 2^{ème} étape : Enregistrement de chaque feuille dans un fichier texte au format CSV³.
- 3^{ème} étape : Insertion du contenu des fichiers CSV à l'aide d'un script SQL.

Vous trouverez le fichier Excel du jeu de données, les fichiers CSV et le script de remplissage sur le dépôt GitHub du projet.

Nous avons choisi d'utiliser les fichiers CSV pour l'importation des données car cette méthode nous apparaissait comme la plus simple, la plus efficace et surtout la plus rapide. De plus, cette méthode présente l'avantage de pouvoir afficher les données contenues dans les différents fichiers sous la forme d'un tableau structuré. Il est possible de modifier ces fichiers à l'aide d'un logiciel tableur.

b. Procédures stockées

Les procédures stockées permettent d'automatiser des actions. Une procédure stockée est une sorte de script, une série d'instruction SQL désignée par un nom. Cette procédure est enregistrée dans la BDD qu'on utilise et on peut l'appeler pour exécuter les instructions qu'elle contient. C'est un peu comme une fonction dans un code.

Nous avons choisi de définir des procédures stockées⁴ pour satisfaire les besoins suivants :

- Affichage du stock des ingrédients et mise à jour à la réception de nouveaux ingrédients
- Consultation des commandes d'un client
- Suppression d'un ingrédient arrivé à expiration (pourra être éventuellement automatisé)
- Mise à jour du stock après envoi d'une commande (la commande devra changer de statut)

Nous avons décidé de stocker ces procédures car elles seront surement utilisées fréquemment au quotidien quand le site web fonctionnera.

³ Comma-separated values permet de représenter les données d'un tableau sous formes de valeurs séparées par des virgules.

⁴ Le manuel d'utilisation reprend le rôle et le mode d'emploi de ces procédures

Affichage du stock des ingrédients et mise à jour à la réception de nouveaux ingrédients

Cette procédure prend comme paramètres l'id de l'ingrédient, la quantité à ajouter et sa fraîcheur.

La procédure commence par vérifier si l'ingrédient que l'on veut ajouter existe ou non dans le stock. Ensuite en fonction de cette condition :

- Si l'élément existe on fait une modification en incrémentant la quantité dans le stock à l'aide du clause UPDATE.
- Si l'élément n'existe pas on l'ajoute dans le stock à l'aide de la clause INSERT INTO.

Pour finir, nous affichons l'intégralité des articles après modifications pour permettre un inventaire des stocks.

L'exécution de la procédure se fait à l'aide de la commande :

```
CALL ajout_stock(nb, fraicheur, id_produit, id_fournisseur)
```

Nb correspond à la quantité d'ingrédients à rajouter ; fraicheur correspond à la fraîcheur des ingrédients ; id_produit correspond à l'id d'ingrédient ; id_fournisseur correspond à l'id du fournisseur d'où proviennent les ingrédients.

```
mysql> CALL ajout_stock(10,20,102,4);
```

Figure 11 - Procédure d'affichage et de mise à jour du stock des ingrédients

Historique des commandes d'un client

Cette procédure prend en paramètre le nom du client pour lequel on souhaite obtenir la liste des commandes.

On réalise une jointure entre les tables Commande et Personne puisqu'une commande possède l'identifiant de la personne qui l'a passée. On restreint cette jointure avec une condition sur le nom de la personne dont l'on souhaite connaître l'historique de commandes.

```
mysql> CALL historique_cmd('Naessens');
+-----+-----+-----+-----+-----+-----+-----+-----+
| IDCommande | DateCommande | Statut | PrixTotal | IDPersonne | Nom | prenom | Adresse |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 3 | 2017-01-08 | livraison | 340 | 15 | Naessens | Valentin | 10 Rue Naessens |
| 5 | 2017-05-18 | preparation | 1080 | 15 | Naessens | Valentin | 10 Rue Naessens |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Figure 12 - Procédure affichant l'historique des commandes d'un client

Suppression d'un ingrédient arrivé à expiration

Cette procédure permet de décrémenter d'une unité (journée) la fraîcheur chaque jour. Elle est programmée pour s'exécuter tous les soirs à minuit.

La procédure supprime également les éléments dont la fraîcheur est égale à 0 en effectuant une jointure entre Stock et Produit.

```
MySQL [giselle]> call perime();
```

NomProduit	NbStock	fut supprimé
Néphrite	68	fut supprimé
Élixir de nage	83	fut supprimé

2 rows in set (0.27 sec)

Query OK, 2 rows affected (0.27 sec)

Figure 13 - Procédure de suppression d'un ingrédient arrivé à expiration

Mise à jour du stock après envoi d'une commande

Cette procédure permet de mettre à jour le stock lors de l'envoi d'une commande. Elle prend en argument le numéro de la commande concernée.

La première partie de la procédure stockée permet d'afficher une commande ainsi que son statut, ses produits, quantités et sa quantité dans le stock.

On effectue une jointure entre les tables Commande, Stock et Produit avec un lien entre les ID commande et les IDproduit ainsi que le statut qui doit être en préparation.

Ensuite le stock du/des produits concernés se met à jour en retirant la quantité présente dans la commande pour chaque produit. Le statut de la commande passe à « livraison ».

Pour finir, on affiche la commande avec le statut modifié.

```
MariaDB [giselle2]> call envoi_commande(1);
```

IDCommande	Statut	IDproduit	QuantiteCommande	IDstock	NbStock
1	preparation	16	3	87	6
1	preparation	48	2	19	88
1	preparation	50	5	21	20
1	preparation	50	5	22	20

4 rows in set (0.00 sec)

IDCommande	Statut	IDproduit	QuantiteCommande	IDstock	NbStock
1	livraison	16	3	87	3
1	livraison	48	2	19	86
1	livraison	50	5	21	15
1	livraison	50	5	22	15

4 rows in set (0.11 sec)

Figure 14 - Procédure de modification du stock après l'envoi d'une commande

c. Soumission de recettes

La sorcière Giselle souhaite mettre en place un système de soumission de nouvelles recettes sur son site e-commerce. Nous avons donc développé la base de données en ce sens. Une table proposition permet de stocker les propositions des clients.

Nous devons donc pouvoir ajouter une proposition de recette et ensuite permettre à Giselle de valider ou de refuser cette proposition.

Ajouter une proposition de recette

Cette procédure stockée prend en paramètres tous les éléments de la recette que l'on souhaite proposée. La procédure récupère et insère les informations passées en paramètres dans les tables Proposition et se_compose_propo.

La table se_compose_propo ne contient que des ID, il faut donc récupérer les ID correspondant aux ingrédients passés en paramètres. On utilise pour cela des requêtes de type INSERT INTO ... SELECT ...

Par exemple, pour ajouter le deuxième ingrédient dans la recette :

```
INSERT INTO
se_compose_propo(`QteIngrédient`, `FraicheurMin`, `FraicheurMax`, `Temps`, `ID
Proposition`, `IDproduit`)
SELECT qte2, fraichMin2, fraichMax2, temps2, @IDPropo, IDproduit
FROM Produit
WHERE NomProduit = ingredient2;
```

Une fois la proposition de recette ajoutée on affiche le récapitulatif de celle-ci.

```
mysql> CALL ajouter_proposition('Potion de test',100,'Vin','Sardonyx',5,10,5,10,'Azurite',10,5,1,5,'Zircon',1,5,10,2,'Merlin');
+-----+-----+-----+
| IDProposition | Nom          | Ingrédient |
+-----+-----+-----+
| 6             | Potion de test | Azurite    |
| 6             | Potion de test | Sardonyx   |
| 6             | Potion de test | Zircon     |
+-----+-----+-----+
3 rows in set (0.44 sec)
```

Figure 15 – Procédure d'ajout d'une proposition de recette

Valider une recette qui a été acceptée

La table Proposition stocke les recettes proposées par les clients. Cette table permet de stocker temporairement les recettes en attentes de validation. La sorcière Giselle pourra utiliser la procédure stockée suivante pour valider une proposition de recette et l'ajouter dans les préparations officielles.

La procédure valider_proposition prend en paramètre l'ID de la proposition pour laquelle on veut transférer la recette dans les recettes officielles. Une fois la proposition ajoutée dans les tables Produit, Preparation et se_compose_prepa on la supprime des tables Proposition et se_compose_propo.

Supprimer une recette qui n'a pas été retenue

Cette procédure prend un paramètre l'ID de la proposition que l'on souhaite supprimée car Giselle ne l'a pas retenue. L'exécution de la procédure se fait à l'aide de la commande :

```
CALL refus_proposition(id)
```

Une fois la commande exécutée, l'enregistrement correspondant est supprimé de la table Proposition et la composition de sa recette est retirée de la table se_compose_propo.

```
mysql> select * from Proposition;
```

IDProposition	Nom	Temperature	Validation	Diluant	IDPersonne
1	Potion de prosit	500	2	2	5
2	Onguent de mysql	0	0	0	18
3	Huile de magicarts	10	0	4	12
46	Potion de test	100	NULL	0	22

4 rows in set (0.14 sec)

```
mysql> select * from se_compose_propo;
```

QteIngredient	FraicheurMin	FraicheurMax	Temps	IDProposition	IDproduit
3	5	10	2	1	38
1	5	10	20	1	42
2	5	10	15	1	70
2	2	5	3	2	32
5	1	2	6	2	33
3	10	20	15	2	35
1	2	8	5	3	35
1	3	7	10	3	49
1	4	6	20	3	50
10	5	1	5	46	32
5	10	5	10	46	46
1	5	10	2	46	51

Figure 16 - Résultat de la procédure de suppression d'une proposition qui n'a pas été retenue

d. Rechercher et consulter

La sorcière Giselle avait également besoin de requêtes de recherche/consultation pour pouvoir utiliser le plus facilement et le plus efficacement possible la base de données de son site web. Giselle souhaitait :

- Les potions ou les onguents réalisables avec un ingrédient précis
- Les couples (potions, onguents) utilisant la même liste d'ingrédients
- Les potions réalisables avec un diluant précis
- La liste des potions classée par température de préparation
- La comparaison entre le prix de vente d'une potion et le total du prix des ingrédients (et diluant) nécessaires à sa réalisation
- Le nombre moyen d'ingrédients utilisés pour fabriquer une potion (ou onguent)

Outre, les besoins de Giselle nous avons pris la liberté d'ajouter des requêtes adaptées à notre MCD dans le but d'accéder à toutes les informations facilement. Nous pouvons donc effectuer les traitements suivants :

- Le contenu d'une commande
- La liste des Potions
- La liste des Onguents
- La liste des produits vendus par Giselle ainsi que leur prix
- Les recettes proposées selon les clients
- Le stock de produits disponibles
- La recette d'une préparation (potion ou onguent) précise
- La recette d'une proposition précise

Les potions ou les onguents réalisables avec un ingrédient précis

Cette requête prend la forme d'une procédure stockée prenant en paramètre le nom de l'ingrédient pour lequel on veut connaître la liste des potions/onguents qui le contiennent.

Pour cela on réalise une intersection entre les toutes les préparations et les préparations possédant l'ingrédient passé en paramètre dans leur composition (on utilise une condition).

```
mysql> call preparation_ingredient('Cimeterre de Maître');
+-----+-----+
| Nom Préparation |
+-----+-----+
| Onguent de lévitation |
| Onguent de soins |
| Huile de bénédiction d'arme |
| Potion d'aide-PC |
| Potion de ruse du renard |
+-----+-----+
5 rows in set (0.42 sec)
```

Figure 17 - Procédure listant les préparations réalisables avec un ingrédient précis

Les couples potions/onguents utilisant la même liste d'ingrédients

On réalise cette requête sous la forme d'une procédure stockée. Cette dernière utilise deux vues : la liste des potions et la liste des onguents. On réalise une jointure pour chacune de ces vues avec la table se_compose_prepa pour avoir la composition des préparations. Ensuite on réalise une jointure à partir des deux sous-requêtes précédentes. On fait la jointure sur les identifiants de produit pour afficher côte à côte les potions et les onguents possédant le/les mêmes ingrédients.

Pour finir on ajoute les conditions suivantes :

- L'onguent et la potion doivent avoir le même nombre d'ingrédients
- Ce nombre d'ingrédients doit être égale au maximum du nombre d'ingrédient entre la potion et l'onguent

Si ces conditions sont vérifiées on obtiendra uniquement les potions et les onguents utilisant exactement la même liste d'ingrédients.

```
mysql> call couple_potion_onguent();
+-----+-----+
| Potion | Onguent |
+-----+-----+
| Potion de ruse du renard | Onguent de lévitation |
| Potion d'endurance de l'ours | Onguent d'endurance |
+-----+-----+
2 rows in set (0.30 sec)
```

Figure 18 - Procédure d'affichage des couples (potions,onguents) utilisant la même liste d'ingrédients

Les potions réalisables avec un ingrédient précis

Pour réaliser cette requête on utilise une procédure stockée prenant en paramètre le nom du diluant pour lequel on souhaite obtenir la liste des potions utilisant ce diluant.

Tout d'abord, on sélectionne les noms de produits de la table produit ainsi que le nom des diluant.

Puis on place une condition pour réaliser une jointure entre les tables produit, Preparation et Diluant.

Ensuite nous spécifions le nom du diluant à rechercher grâce à une condition où la paramètre de la procédure permet de repérer quelle(s) potion(s) utilise le diluant choisit et donc celles que nous devons afficher.

```
MySQL [giselle]> call diluant('Vin');
+-----+-----+
| NomProduit | NomDiluant |
+-----+-----+
| Potion de flou | Vin |
| Potion de ruse du renard | Vin |
| Potion de test | Vin |
+-----+-----+
```

Figure 19 - Procédure affichant les potions réalisables avec un ingrédient précis

La liste des potions classées par température de préparation

La procédure affiche la liste des potions triée par température croissante. L'exécution de la procédure se fait à l'aide de la commande :

```
CALL temperature;
```

```
mysql> CALL temperature;
+-----+-----+
| Temperature | NomProduit |
+-----+-----+
| 1 | Potion d'endurance de l'ours |
| 2 | Potion de flou |
| 9 | Potion de protection contre les projectiles |
| 14 | Potion de grâce féline |
| 29 | Potion de ruse du renard |
| 30 | Potion de lévitation |
| 36 | Huile de gourdin magique |
| 41 | Potion d'aide |
| 43 | Potion de soins légers |
| 95 | Huile de bénédiction d'arme |
+-----+-----+
```

Figure 20 - Procédure listant les potions classées par température de préparation

La comparaison entre le prix de vente d'une potion et le total du prix des ingrédients (et diluant) nécessaires à sa réalisation

Cette requête consiste en fait à calculer la marge que dégage Giselle pour chaque préparation qu'elle vend.

On doit donc faire la différence entre le prix des produits (onguents et potions), la somme des prix de chaque ingrédient multiplié par leur quantité et le prix du diluant si c'est une potion.

On récupère le prix total de la préparation et le prix du diluant grâce à une sous requête.

```
mysql> call marge_preparation();
```

IDproduit	NomProduit	Marge
11	Huile de bénédiction d'arme	207
12	Huile de gourdin magique	100
14	Potion d'aide	183
15	Potion de flou	60
16	Potion de grâce féline	38
17	Potion de lévitation	72
18	Potion de protection contre les projectiles	96
19	Potion de ruse du renard	160
20	Potion de soins légers	127
21	Potion d'endurance de l'ours	60

```
10 rows in set (0.15 sec)
```

Figure 21 - Procédure affichant la marge que dégage Giselle pour chaque préparation

Le nombre moyen d'ingrédients utilisés pour fabriquer une potion (ou onguent)

La première partie sert à créer une vue affichant la somme d'ingrédients par potions. Elle est ensuite utilisée dans la deuxième partie pour calculer la moyenne d'ingrédients par potion au sein d'une seconde vue.

L'exécution de la procédure se fait à l'aide de la commande :

```
SELECT * FROM Moyenne_ingredient_potion;
```

```
mysql> SELECT * FROM Moyenne_ingredient_potion;
```

Valeur
6.0000

Figure 22 - Procédure donnant le nombre moyen d'ingrédients utilisés pour fabriquer une préparation

e. Sauvegarder

A juste titre, Giselle souhaitait mettre en place un système de sauvegarde de sa base de données :

- Recettes enregistrées dans la base de données
- Ingrédients associés à chaque fournisseur
- Commandes dont la réception n'a pas encore été confirmée

Pour sauvegarder certaines tables ou toute une base de données MySQL on utilise l'utilitaire mysqldump. La commande de sauvegarde permet de récupérer l'intégralité ou une partie d'une base de données au sein d'un fichier SQL. A partir de ce dernier, on peut restaurer la base sur une autre base de données.

Par conséquent le fichier SQL de la base de données de Giselle généré grâce à mysqldump permettra deux fonctions principales :

- Conserver la structure et les données de la base de données en cas de problème
- Permettre la migration de la base de données (vers une autre machine ou un autre SGBDR)

Pour sauvegarder intégralement le serveur MySQL (toutes les base de données présentes) dans un fichier `databases_magicarts.sql` on peut utiliser les commandes suivantes.

- Lorsqu'on est en dehors du serveur MySQL (sur le système d'exploitation) :

```
mysqldump --user=root --password -A > databases_magicarts.sql
```

- Lorsqu'on est dans l'interface en ligne de commande du serveur MySQL il faut ajouter « system » avant « mysqldump »

Pour sauvegarder toutes les recettes présentes dans la base de données on doit conserver les tables suivantes : `Produit`, `se_compose_prepa`, `Preparation`, `Diluant` et `Ingredient`.

On utilise donc la commande ci-dessous.

```
mysqldump --user=pierre --password giselle Produit se_compose_prepa  
Preparation Diluant Ingredient > tables_produit.sql
```

Pour sauvegarder les ingrédients associés à chaque fournisseur on utilise la commande suivante :

```
mysqldump --user=pierre --password giselle Produit Ingredient Fournisseur  
fournit > tables_ingredient_fournisseur.sql
```

Giselle la sorcière souhaitait sauvegarder les commandes dont la réception n'a pas encore été récupérées. Pour cela on sauvegarde un fichier csv contenant le résultat d'une requête qui permet de récupérer les commandes ayant un statut égal à en livraison. On utilise la procédure stockée : `cmd_livraison()`.

```
CREATE PROCEDURE cmd_livraison()  
BEGIN  
SELECT  
Commande.IDCommande,DateCommande,Statut,PrixTotal,Nom,Prenom,Adresse,Produ  
it.NomProduit,QuantiteCommande,NomRecipient  
INTO OUTFILE '/var/lib/mysql-files/save.csv'  
FROM Recipient,Produit,Personne,Commande,contient  
WHERE Statut='livraison'  
AND Commande.IDpersonne=Personne.IDPersonne  
AND contient.IDCommande=Commande.IDCommande  
AND contient.IDproduit=Produit.IDproduit  
AND contient.IDrecipient=Recipient.IDrecipient;  
END|
```

```
SELECT INTO OUTFILE '/var/lib/mysql-files/cat.txt'
```

Cette partie permet de sauvegarder les résultats d'un select au sein d'un fichier, le chemin est un chemin défini par MySQL.

3. Retour sur expérience

a. Synthèse des problèmes rencontrés

Comme nous l'avons vu précédemment, l'ensemble des besoins formalisés dans le cahier des charges de Giselle sont couverts par la solution que nous avons proposée. Durant le projet GiselleMagicArts nous avons rencontrés plusieurs difficultés.

La toute première a été d'identifier précisément les attentes de la sorcière Giselle. En effet la formulation des requêtes est souvent sujet à diverses interprétations concernant le résultat attendus. Dans la réalité, il aura fallu organiser régulièrement des réunions avec la MOA (ici Giselle) pour être sûr de bien répondre à ses besoins.

La seconde difficulté majeure rencontrée fut la mise au point du Modèle Conceptuel de Données (MCD). Nous l'avons modifié de nombreuses fois. Il aurait fallu s'attarder plus longuement sur sa conception dès la première version pour éviter de le modifier de trop nombreuses fois. Cela a en effet ralenti de manière significative l'avancée du projet.

Le jeu de données devait être cohérent et significatif pour pouvoir réaliser les requêtes. Nous avons eu des difficultés pour que les données soient cohérentes. Au début nous n'avions pas accordés suffisamment de ressources humaines pour cette tâche. Nous avons donc ajoutés deux personnes de plus.

Par la suite, nous avons eu des difficultés à correctement formaliser les requêtes SQL. La plus difficile à développer a été les couples potions/onguents utilisant la même liste d'ingrédients.

Le dernier point que nous pouvons soulever sont les limites de l'utilisation du Raspberry pour notre base de données. Ce nano-ordinateur possède de faibles performances. Il convient parfaitement pour les tests que nous avons effectués pour ce projet. Cependant, pour le site web de Giselle la sorcière il faudra se tourner vers une machine serveur puissante et donc beaucoup plus coûteuse.

b. Bilan de l'équipe

Du point de vue de l'ensemble de l'équipe ce projet a été une réussite. Nous avons répondu à l'ensemble des besoins de la MOA.

Ce fut possible grâce à une bonne organisation. L'utilisation d'outils de productivité et de gestion de projet tel que Trello et les diagrammes de Gantt ont permis une répartition des tâches cohérente. Les outils mis en place par le chef de projet ont été particulièrement appréciés par l'équipe : Trello, GitHub, Raspberry, réunions régulières, partage onedrive pour création du jeu de données.

La bonne cohésion de l'équipe du projet ainsi que la bonne gestion du temps nous ont permis de réfléchir et d'implémenter des perspectives d'améliorations (création de vues, requêtes supplémentaires...)

Pour finir, ce projet a permis à l'ensemble des membres de l'équipe d'approfondir sa maîtrise du langage SQL et d'expérimenter la conception d'une base de données de A à Z.

c. Bilans individuels



Alexandre Lecomte

Ce projet a été une première expérience pour moi en tant que chef de projet. J'ai pu mettre en place des outils étudiés lors des PROSITS de gestion de projet : WBS, Diagramme de Gantt, conception de livrables... J'ai également pu appliquer la méthode MERISE dans le cadre d'un projet et mieux comprendre l'importance de chaque étape et leur succession.



Florian Fritsch

Le projet nous a permis d'exercer nos connaissances dans un objectif plus concret. L'organisation dans notre groupe était parfaite et tout a été respecté. Cela m'a permis de mettre en œuvre tous les points vus pendant les prosits, et donc d'avoir pu améliorer mes compétences. Je maîtrise mieux les requêtes SQL.



Aurélien Klein

J'ai apprécié le projet et le thème qui était dans la continuité des prosits. La manipulation m'a permis de comprendre de nombreux points là où j'avais des difficultés lors des workshops. Le projet s'est déroulé sans problème insurmontable.



Pierre Geeraert

Ce projet m'a tout d'abord permis de me familiariser avec les commandes de MySQL et aussi de comprendre les différentes manières de réflexion de ce langage, j'ai pu apercevoir la portée importante de ce langage et la multitude d'application, cela m'a permis de mieux apprécier ce langage.

J'ai bien réussi à m'intégrer dans mon groupe et à pouvoir comprendre facilement le travail des membres grâce à divers outils très utiles (trello / git).

ANNEXE 1 : Créer un serveur MySQL sur un Raspberry Pi

Nous voulions que l'ensemble des membres de l'équipe du projet puisse se connecter et travailler sur la base de données de manière simultanée. Nous avons donc décidé d'installer un serveur MySQL (version 5.7) sur un Raspberry pi 3. Ce dernier est connecté au domicile d'un membre de l'équipe et accessible depuis l'extérieur.

A l'aide de l'adresse IP publique du routeur domestique et à la création de comptes utilisateurs sur la base de données MySQL, les membres de l'équipe peuvent travailler sur la base de données en même temps.

Détaillons les étapes de déploiement de ce serveur sur un Raspberry pi 3.

Tout d'abord nous conseillons de mettre à jour l'ensemble du système d'exploitation de l'ordinateur.

Ensuite on utilise la commande suivante pour récupérer le paquet d'installation du serveur MySQL version 5.7 (dernière version stable actuellement) :

```
pi@raspberrypi:~ $ sudo apt-get install mysql-server-5.7
```

Après quelques minutes, le serveur MySQL est en place, pour le démarrer on utilise la commande :

```
pi@raspberrypi:~ $ sudo service mysql start
```

On peut maintenant se connecter en local au serveur MySQL en utilisant la commande :

```
mysql -h localhost -u user -ppassword
```

Une fois connecté, on crée un compte utilisateur pour chaque membre de l'équipe du projet à l'aide de la clause GRANT.

```
GRANT ALL PRIVILEGES ON *.* TO 'florian'@'%' IDENTIFIED BY 'password';
```

Ici on attribue tous les privilèges à l'utilisateur florian qui peut se connecter depuis n'importe quelle adresse IP avec le mot de passe password.

Ensuite sur le routeur, on crée une règle de redirection du port 3306 (port MySQL par défaut) arrivant de l'extérieur vers l'adresse IP local du Raspberry comme le montre la capture d'écran suivante.

server-mysql	TCP	Port	3306	192.168.1.48	3306
--------------	-----	------	------	--------------	------

Grâce à cette règle de redirection peut maintenant se connecter à distance au serveur MySQL et donc à la base de données du projet. Pour l'administrer on pourra utiliser un terminal ou alors phpmyadmin.

L'installation de phpmyadmin requiert la mise en place d'un serveur Apache et la redirection du port 80.

ANNEXE 2 : Script de création de la base de données

```
#-----
#      Script MySQL.
#-----

#-----
# Table: Personne
#-----

CREATE TABLE Personne(
    IDPersonne int (11) Auto_increment NOT NULL ,
    Nom        Char (25) NOT NULL ,
    Prenom     Char (25) NOT NULL ,
    Adresse    Varchar (25) NOT NULL ,
    PRIMARY KEY (IDPersonne )
)ENGINE=InnoDB;

#-----
# Table: Ingredient
#-----

CREATE TABLE Ingredient(
    IDproduit Int NOT NULL ,
    PRIMARY KEY (IDproduit )
)ENGINE=InnoDB;

#-----
# Table: Diluant
#-----

CREATE TABLE Diluant(
    IDDiluant  int (11) Auto_increment NOT NULL ,
    NomDiluant Char (25) NOT NULL ,
    PrixDiluant Float NOT NULL ,
    PRIMARY KEY (IDDiluant )
)ENGINE=InnoDB;

#-----
# Table: Recipient
#-----

CREATE TABLE Recipient(
    IDrecipient int (11) Auto_increment NOT NULL ,
    NomRecipient Varchar (255) ,
    Prix         Float NOT NULL ,
    Stock        Int ,
    PRIMARY KEY (IDrecipient )
)ENGINE=InnoDB;

#-----
# Table: Commande
#-----

CREATE TABLE Commande(
    IDCommande int (11) Auto_increment NOT NULL ,
    DateCommande Date NOT NULL ,
    Statut      Char (25) ,
    PrixTotal   Float NOT NULL ,
    IDPersonne  Int NOT NULL ,
    PRIMARY KEY (IDCommande )
)ENGINE=InnoDB;
```

```

#-----
# Table: Fournisseur
#-----

CREATE TABLE Fournisseur(
    IDFournisseur int (11) Auto_increment NOT NULL ,
    Nom           Char (25) NOT NULL ,
    Numero        Int NOT NULL ,
    PRIMARY KEY (IDFournisseur )
)ENGINE=InnoDB;

#-----
# Table: Proposition
#-----

CREATE TABLE Proposition(
    IDProposition int (11) Auto_increment NOT NULL ,
    Nom           Char (25) NOT NULL ,
    Temperature   Int ,
    Diluant        Int ,
    IDPersonne    Int ,
    PRIMARY KEY (IDProposition )
)ENGINE=InnoDB;

#-----
# Table: Stock
#-----

CREATE TABLE Stock(
    IDstock        int (11) Auto_increment NOT NULL ,
    NbStock        Int NOT NULL ,
    Fraicheur      Int ,
    IDproduit      Int ,
    IDFournisseur  Int ,
    PRIMARY KEY (IDstock )
)ENGINE=InnoDB;

#-----
# Table: Produit
#-----

CREATE TABLE Produit(
    IDproduit      int (11) Auto_increment NOT NULL ,
    NomProduit     Varchar (255) ,
    Prix           Float ,
    PRIMARY KEY (IDproduit )
)ENGINE=InnoDB;

#-----
# Table: Preparation
#-----

CREATE TABLE Preparation(
    Temperature    Int ,
    IDproduit      Int NOT NULL ,
    IDDiluant      Int ,
    IDPersonne     Int ,
    PRIMARY KEY (IDproduit )
)ENGINE=InnoDB;

```

```

#-----
# Table: contient
#-----

CREATE TABLE contient(
    QuantiteCommande Int ,
    FraicheurVoulu   Int ,
    IDCommande        Int NOT NULL ,
    IDproduit          Int NOT NULL ,
    IDrecipient        Int NOT NULL ,
    PRIMARY KEY (IDCommande ,IDproduit ,IDrecipient )
)ENGINE=InnoDB;

#-----
# Table: fourniture
#-----

CREATE TABLE fourniture(
    IDFournisseur Int NOT NULL ,
    IDproduit      Int NOT NULL ,
    PRIMARY KEY (IDFournisseur ,IDproduit )
)ENGINE=InnoDB;

#-----
# Table: se compose prepa
#-----

CREATE TABLE se_compose_prepa(
    QteIngredient      Int ,
    FraicheurMin        Int ,
    FraicheurMax        Int ,
    Temps               Int ,
    IDproduit            Int NOT NULL ,
    IDproduitFinal      Int NOT NULL ,
    PRIMARY KEY (IDproduit ,IDproduitFinal )
)ENGINE=InnoDB;

#-----
# Table: se compose propo
#-----

CREATE TABLE se_compose_propo(
    QteIngredient Int ,
    FraicheurMin  Int ,
    FraicheurMax  Int ,
    Temps         Int ,
    IDProposition Int NOT NULL ,
    IDproduit      Int NOT NULL ,
    PRIMARY KEY (IDProposition ,IDproduit )
)ENGINE=InnoDB;

#-----
# Table: fourniture recipient
#-----

CREATE TABLE fourniture_recipient(
    IDFournisseur Int NOT NULL ,
    IDrecipient    Int NOT NULL ,
    PRIMARY KEY (IDFournisseur ,IDrecipient )
)ENGINE=InnoDB;

```

```

ALTER TABLE Ingredient ADD CONSTRAINT FK_Ingredient_IDproduit FOREIGN KEY (IDproduit)
REFERENCES Produit(IDproduit);

ALTER TABLE Commande ADD CONSTRAINT FK_Commande_IDPersonne FOREIGN KEY (IDPersonne)
REFERENCES Personne(IDPersonne);

ALTER TABLE Proposition ADD CONSTRAINT FK_Proposition_IDPersonne FOREIGN KEY (IDPersonne)
REFERENCES Personne(IDPersonne);

ALTER TABLE Stock ADD CONSTRAINT FK_Stock_IDproduit FOREIGN KEY (IDproduit) REFERENCES
Produit(IDproduit);

ALTER TABLE Stock ADD CONSTRAINT FK_Stock_IDFournisseur FOREIGN KEY (IDFournisseur)
REFERENCES Fournisseur(IDFournisseur);

ALTER TABLE Preparation ADD CONSTRAINT FK_Preparation_IDproduit FOREIGN KEY (IDproduit)
REFERENCES Produit(IDproduit) ON DELETE CASCADE ;

ALTER TABLE Preparation ADD CONSTRAINT FK_Preparation_IDDiluant FOREIGN KEY (IDDiluant)
REFERENCES Diluant(IDDiluant);

ALTER TABLE Preparation ADD CONSTRAINT FK_Preparation_IDPersonne FOREIGN KEY (IDPersonne)
REFERENCES Personne(IDPersonne);

ALTER TABLE contient ADD CONSTRAINT FK_contient_IDCommande FOREIGN KEY (IDCommande)
REFERENCES Commande(IDCommande);

ALTER TABLE contient ADD CONSTRAINT FK_contient_IDproduit FOREIGN KEY (IDproduit)
REFERENCES Produit(IDproduit);

ALTER TABLE contient ADD CONSTRAINT FK_contient_IDrecipient FOREIGN KEY (IDrecipient)
REFERENCES Recipient(IDrecipient);

ALTER TABLE fournit ADD CONSTRAINT FK_fournit_IDFournisseur FOREIGN KEY (IDFournisseur)
REFERENCES Fournisseur(IDFournisseur);

ALTER TABLE fournit ADD CONSTRAINT FK_fournit_IDproduit FOREIGN KEY (IDproduit) REFERENCES
Produit(IDproduit);

ALTER TABLE se_compose_prepa ADD CONSTRAINT FK_se_compose_prepa_IDproduit FOREIGN KEY
(IDproduit) REFERENCES Produit(IDproduit);

ALTER TABLE se_compose_prepa ADD CONSTRAINT FK_se_compose_prepa_IDproduitFinal FOREIGN KEY
(IDproduitFinal) REFERENCES Produit(IDproduit) ON DELETE CASCADE ;

ALTER TABLE se_compose_propo ADD CONSTRAINT FK_se_compose_propo_IDProposition FOREIGN KEY
(IDProposition) REFERENCES Proposition(IDProposition) ON DELETE CASCADE ;

ALTER TABLE se_compose_propo ADD CONSTRAINT FK_se_compose_propo_IDproduit FOREIGN KEY
(IDproduit) REFERENCES Produit(IDproduit);

ALTER TABLE fournit_recipient ADD CONSTRAINT FK_fournit_recipient_IDFournisseur FOREIGN KEY
(IDFournisseur) REFERENCES Fournisseur(IDFournisseur);

ALTER TABLE fournit_recipient ADD CONSTRAINT FK_fournit_recipient_IDrecipient FOREIGN KEY
(IDrecipient) REFERENCES Recipient(IDrecipient);

```

ANNEXE 3 : Ressources

Outils de gestion de projet utilisé (WBS, Gantt, Plannings, listes des tâches à réaliser...) :

- <https://www.matchware.com/fr/logiciel-de-mind-mapping>
- <https://trello.com/>

Ressources pour la gestion de projet :

- <http://www.workbreakdownstructure.fr/>
- <http://www.blog-gestion-de-projet.com/conseils-pour-reussir-votre-planification-de-projet/>

La méthode MERISE :

- <http://perso.modulonet.fr/~placurie/Cgo1.htm> (chapitres 5 à 8)

Logiciel de formalisation du MCD, MLD et MPD de la méthode Merise :

- <http://www.jfreesoft.com/JMerise/>

Documentation officielle de MySQL :

- <https://dev.mysql.com/doc/refman/5.7/en/>

Requêtes SQL :

- SQL, *Ronald R. Plew & Ryan K. Stephens, édition CampusPress, traduction de l'ouvrage Teach Yourself SQL in 24 Hours – Second Edition.*
- <https://sql.sh/>