

LEDAIM--OLIVIER Naig
LE CACHEUX Maïwenn
LIZEN Antoine
MASSY Léo
MORVAN Enora
ROLLAND Kevin



RAPPORT

KawaRésa



Sommaire

1. Introduction	2
1.1. Cadre du projet	2
1.2. Descriptif du projet	3
1.2.1. Gestion des produits	3
1.2.2. Besoins générés	4
1.2.3. Contraintes et limites	6
2. Gestion du projet	7
2.1. La répartition des rôles	7
2.2. Le diagramme des tâches	8
2.3. Diagrammes de Gantt	8
3. Conception	10
3.1 Analyse des besoins	10
3.1.1. Concertation	10
3.1.2. Entretiens avec le personnel du Kawa	11
3.1.3. Questionnements sur le RGPD	12
3.1.4. Solutions techniques	12
3.2. Maquettes de la web application	13
3.2.1. Maquettes Utilisateur	13
3.2.2. Maquettes gestionnaire	16
3.2.3. Maquettes administrateur	20
3.3. Schéma Navigationnel d'Interface (SNI)	21
3.4. Diagramme de classe	25
4. Réalisation	26
4.1. Le Front-End	26
4.2. Le Back-End	27
4.3. La base de données	29
4.4. Maintenance	30
4.5. Évolutions futures	30
5. Conclusion	31
6. Table des illustrations	32

1. Introduction

Ce rapport concerne le projet tutoré de la web-application KawaRésa. Il présente les différentes étapes de conception et de réalisation ainsi que la gestion de projet relatives à cette web-application.

1.1. Cadre du projet

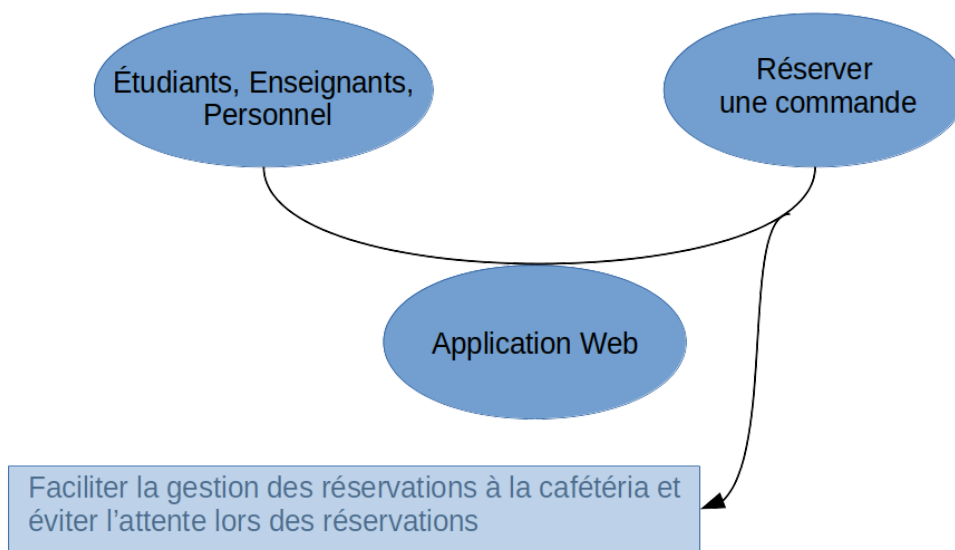


Figure 1 : Bête à cornes

Ce projet consiste à créer une web-application qui permettra aux étudiants, enseignants et au personnel de réserver leur repas à la cafétéria de l'IUT, le Kawa, pour le jour même. L'objectif de ce projet est de faciliter et surtout de rendre plus rapide la

réservation des commandes pour les utilisateurs mais également de ne pas compliquer la tâche au personnel du Kawa et donc de créer une application rapide et intuitive.

Ce projet possède trois aspects :

- **l'aspect utilisateur** : il correspond aux étudiants, enseignants et au personnel de l'IUT qui souhaitent effectuer une réservation.
- **l'aspect gestionnaire** : il correspond au personnel du Kawa qui gère les réservations et les stocks.
- **l'aspect administrateur** : c'est lui qui va gérer les comptes utilisateurs.

1.2. Descriptif du projet

1.2.1. Gestion des produits

Tous les produits actuellement disponibles au Kawa sont réservables sur la web-application. Il est possible d'ajouter un nouveau produit mais également de modifier ou supprimer un produit existant.

La liste des produits disponibles à la réservation sera actualisée tous les matins par le personnel du Kawa.

1.2.2. Besoins générés

Voici les principaux besoins générés :

Pour les utilisateurs :

- Consulter la liste des produits disponibles
- Consulter un produit en particulier
- Réserver un produit
- Gérer son compte

Pour les gestionnaires :

- Gérer les produits
- Gérer les stocks
- Gérer les réservations
- Gérer leur compte

Pour l'administrateur :

- Gérer les comptes utilisateurs
- Remettre la web-application à zéro
- Gérer son compte

Ci-dessous, le diagramme de l'ensemble des besoins (use case) générés par ce projet.

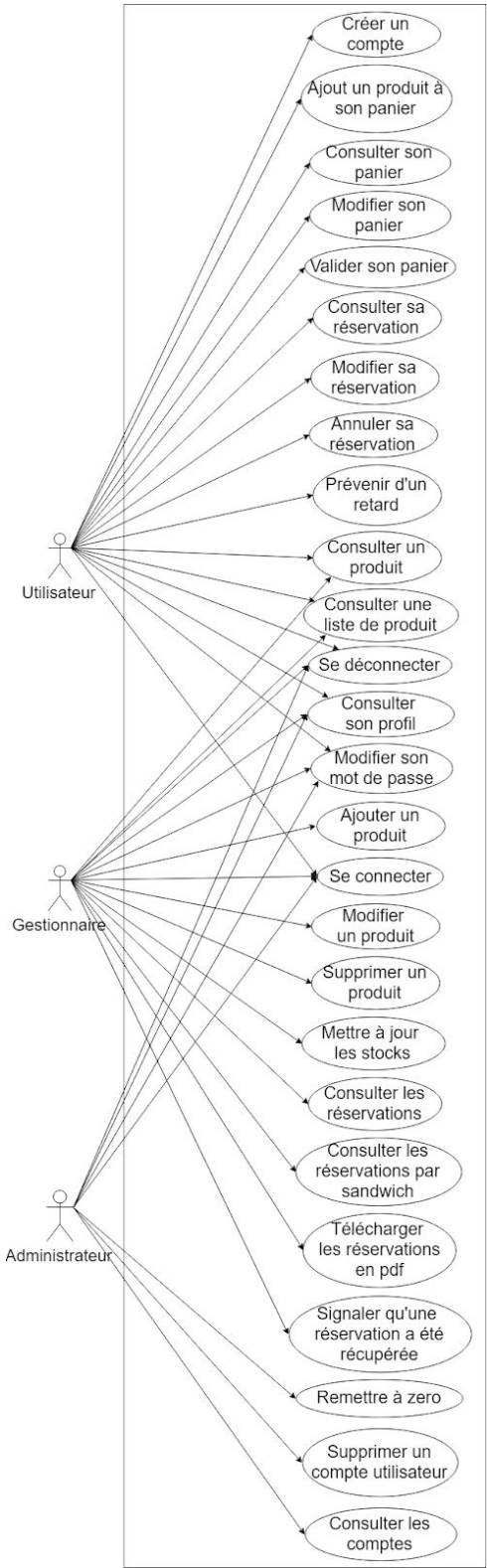


Figure 2 : Diagramme des Use Case

1.2.3. Contraintes et limites

Contraintes

- Responsivité de la web-application sur tous supports (téléphone, tablette et ordinateur) ;
- RGPD : obligation d'anonymiser les utilisateurs et de ne collecter aucune donnée personnelle afin de ne pas avoir besoin de nommer un Responsable des Traitements de Données ;
- Obligation d'utiliser son adresse mail universitaire Rennes 1 et de la faire vérifier pour créer un compte utilisateur ;
- Blocage de deux semaines puis bannissement définitif de la web-application pour toute personne qui réserverait une commande mais ne viendrait pas la récupérer plusieurs fois.

Limite

Ce projet ne concerne pas le paiement de la réservation, cela se passera auprès du personnel du Kawa à la récupération de la commande.

2. Gestion du projet

Dans ce chapitre, nous allons vous présenter les différents points qui forment la gestion de notre projet. C'est-à-dire la répartition des rôles, le diagramme des tâches ainsi que les diagrammes de Gantt prévisionnel et réel.

2.1. La répartition des rôles

Voici le tableau de répartition des rôles de chaque membre :

Nom	Prénom	Rôle
Le Cacheux	Maiwenn	Chef de projet , développeuse back-end
Ledaim--Olivier	Naig	Développeuse front-end
Morvan	Enora	Responsable Base De Données
Massy	Léo	Développeur back-end
Lizen	Antoine	Développeur back-end
Rolland	Kévin	Développeur back-end

Figure 3 : Rôles des membres

2.2. Le diagramme des tâches

Voici le diagramme des tâches décrivant la totalité des tâches du projet :

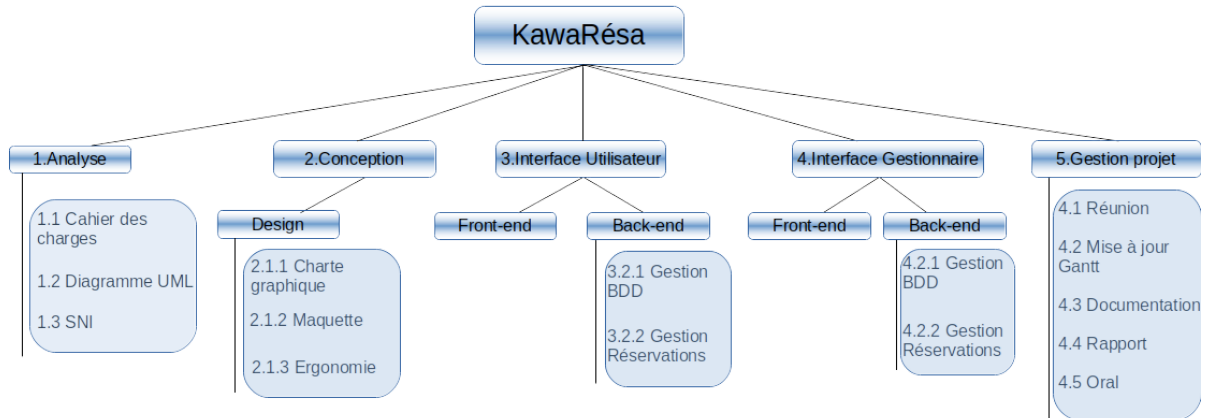


Figure 4 : Diagramme des tâches (WBS)

2.3. Diagrammes de Gantt

Voici le diagramme de Gantt prévisionnel que nous avons réalisé au moment de la création du cahier des charges :

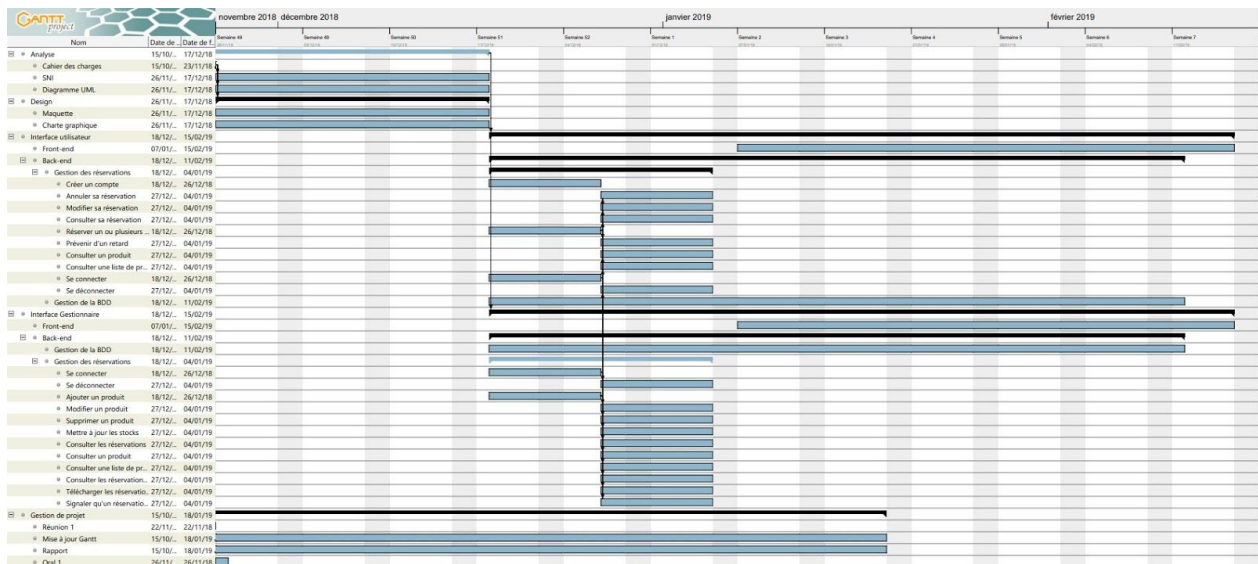


Figure 5 : Diagramme de Gantt prévisionnel

Et voici le diagramme de Gantt réel. Il y a plusieurs petites différences liées à l'imprévisibilité du côté pratique. Le plus gros décalage concerne la programmation back-end et front-end qui ont été fortement impactées par le changement de framework, comme expliqué plus tard dans la partie **Réalisation (partie 4)**.

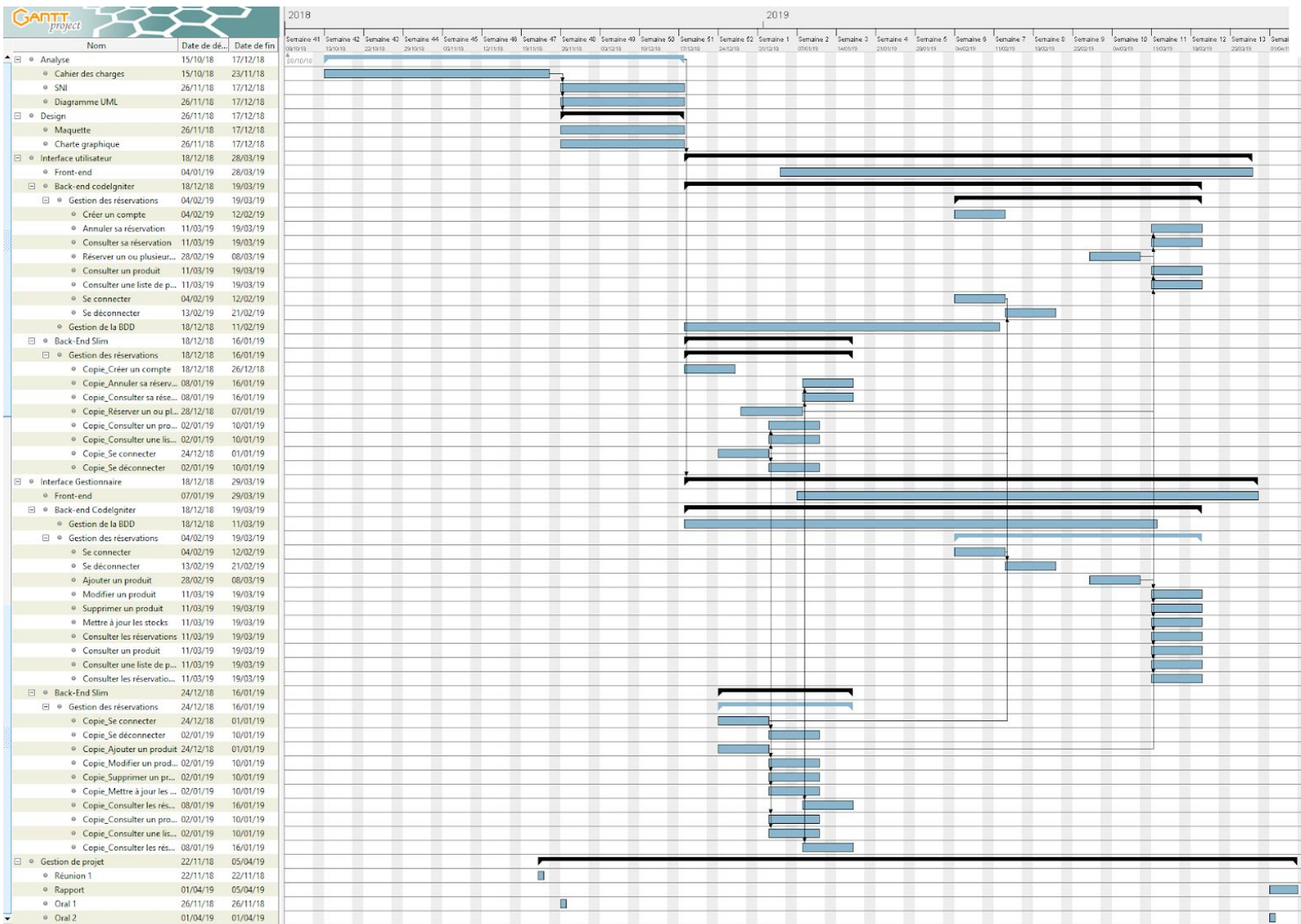


Figure 6 : Diagramme de Gantt réel

3. Conception

Avant de se lancer dans la programmation d'un logiciel, il est nécessaire de passer au préalable par une phase de conception. Celle-ci permet de définir concrètement ce à quoi le logiciel va devoir correspondre et permet de même de vérifier l'avancement du projet.

3.1 Analyse des besoins

La conception de ce projet se base sur les différents besoins que nous avons soulevés au point 1.2.2. Pour arriver à la liste de ces besoins, nous avons dû passer par plusieurs étapes.

3.1.1. Concertation

Tout d'abord, comme le projet venait de notre propre initiative, nous nous sommes concertés pour repérer les différentes fonctionnalités qui nous semblaient importantes. En plus d'apporter notre point de vue de développeurs et notre connaissance en gestion de projet, nous sommes également la cible de ce projet. Effectivement, nous nous sommes vus comme de futurs utilisateurs et avons essayé d'imaginer la façon dont nous pourrions l'utiliser. Ainsi, nous avons pu combiner nos compétences et nos besoins en tant qu'étudiants à l'IUT de Lannion mangeant régulièrement au Kawa.

3.1.2. Entretien avec le personnel du Kawa

Notre application ne concernant pas uniquement les clients du Kawa, mais également le personnel du Kawa, nous nous sommes arrangés pour pouvoir obtenir un rendez-vous avec ces derniers. Nous avons donc eu une entrevue officielle avec deux des membres du personnel ainsi que la directrice du CROUS de Lannion. Nous avons pu poser des questions à chacune des parties sur leurs préférences d'utilisation :

- Quels étaient leurs besoins vis-à-vis de leur service ;
- De quelle manière fallait-il mettre en place l'interface pour leur rendre la tâche facile ;
- Quelles sanctions devraient être appliquées pour les personnes ne respectant pas leurs engagements.

Cette réunion nous a non seulement permis de définir les besoins des gestionnaires (personnel du Kawa) pour cette web-application, mais également de pouvoir nous entretenir avec la directrice du CROUS de Lannion pour nous assurer qu'elle était d'accord pour le lancement de ce projet. Enfin, en nous entretenant ainsi, nous avons pu répondre aux questions du personnel et de la directrice et surtout identifier leurs craintes pour pouvoir les rassurer et nous assurer nous-mêmes des points qui étaient importants, à savoir :

- La possibilité de sanctionner les utilisateurs qui ne viendraient pas chercher leur repas ;
- La possibilité de signaler qu'un repas sera récupéré tard ou non ;
- Les horaires à respecter pour pouvoir passer une commande, la modifier ou l'annuler ;
- La simplicité d'utilisation ;
- La rapidité d'utilisation.

Ce projet ne devait pas être une charge de travail supplémentaire pour le personnel du Kawa. Suite à cette réunion, nous nous sommes régulièrement entretenus avec le personnel (sur de courts laps de temps de 5 à 10 min) pour parler de l'avancement du projet et s'assurer qu'il répondait à leurs besoins. Ces entrevues nous ont permis de détecter un besoin qui n'avait pas été souligné avant : la possibilité pour les gestionnaires de rajouter une commande pour quelqu'un d'autre.

3.1.3. Questionnements sur le RGPD

Comme souligné dans les contraintes, nous nous devons de respecter le RGPD. Cependant, nous avons tout de même besoin de pouvoir authentifier nos utilisateurs, et surtout les limiter aux membres de l'IUT. Cette réglementation étant passée tout récemment, nous n'étions pas certains de nos droits et devoirs à ce sujet. C'est dans l'optique de mieux les déterminer et de pouvoir définir nos options que nous nous sommes entretenus avec Mr. Bertrand De Villeneuve, professeur de droit à l'IUT de Lannion. Nous savions donc qu'il nous était impossible de conserver des données à caractère personnel, c'est-à-dire toute donnée permettant d'identifier une personne. Avec l'aide de notre tuteur, Mr. Gildas Quiniou, nous avons donc pu déterminer plusieurs manières de contourner ce problème : conserver l'adresse mail des utilisateurs, mais hashée, ce qui la rend totalement impossible à identifier. En consultant une autre fois Mr. De Villeneuve, il a pu nous confirmer que cette solution respecte le RGPD.

3.1.4. Solutions techniques

Après avoir fait ces recherches sur les différents besoins auxquels doit répondre notre projet, nous avons pu nous entretenir avec notre tuteur, Mr. Gildas Quiniou, qui a pu nous suggérer différentes solutions techniques adaptées.

3.2. Maquettes de la web application

3.2.1. Maquettes Utilisateur

La page d'accueil de la web-application correspond à la page de connexion. On peut grâce à cela déterminer si la personne est un simple utilisateur, un gestionnaire ou encore un administrateur. La personne doit entrer son login et son mot de passe pour s'identifier. Dans le cas où celle-ci ne possède pas encore de compte, elle peut se diriger vers la page d'inscription en cliquant sur "Pas encore inscrit ?" pour créer un compte utilisateur.

La maquette présente une page de connexion sur un fond rouge. En haut, un bandeau blanc contient le titre "Bienvenue sur KawaRésa, le site de réservation du Kawa !" en noir, flanqué de deux cercles bleus. À gauche, un rectangle blanc encadré contient le mot "Connexion". À droite, il y a deux champs de saisie blancs : "Login :" suivi d'un champ, et "Mot de passe :" suivi d'un champ. En bas à droite, le lien "Pas encore inscrit ?" est visible.

Figure 7 : Connexion

Si la personne est identifiée comme étant un utilisateur, elle arrive sur la liste de l'ensemble des produits commercialisés au Kawa. Les produits sont répartis dans différentes catégories selon leur origine (sandwichs, paninis, salades ou plats chauds). Pour chaque produit on peut voir son nom, sa photo et ses allergènes. Les plats disponibles seulement pendant une durée limitée sont mis en avant en étant plus imposants. A partir de là, l'utilisateur aura aussi la possibilité d'aller dans l'espace "Mon compte" pour voir ses informations et changer son mot de passe.

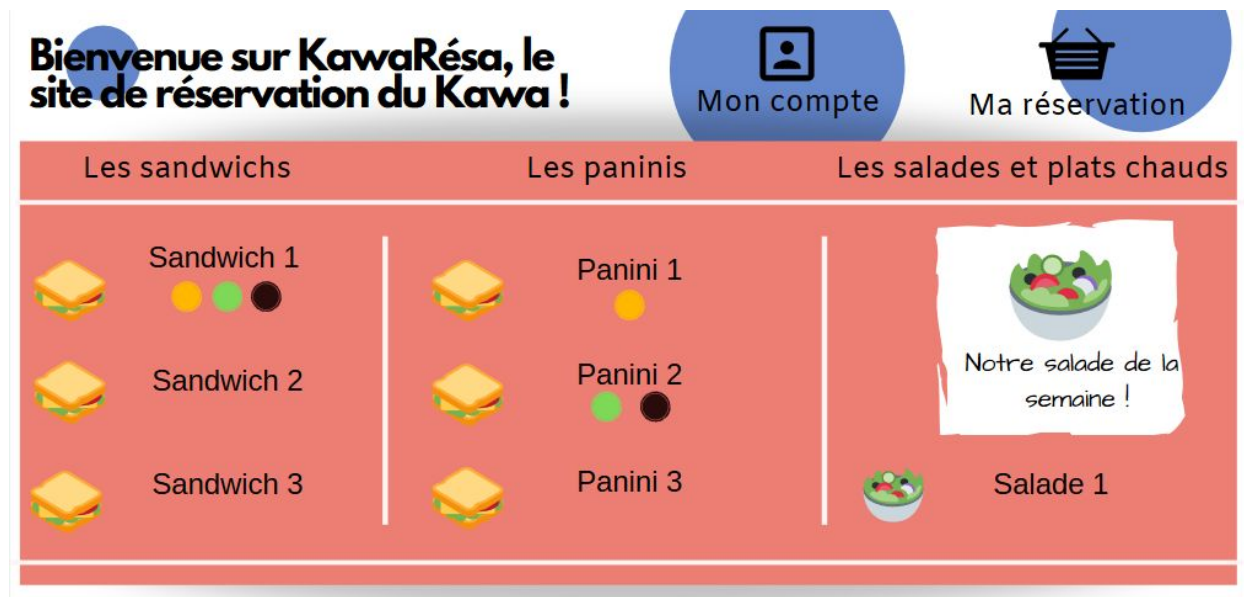


Figure 8 : Accueil Utilisateur (Liste produit)

Si l'utilisateur a l'intention de réserver un produit ou alors tout simplement de vérifier les ingrédients le composant, il devra sélectionner celui-ci à partir de la **figure 8**. Il arrivera ensuite dans le descriptif détaillé du produit. Comme dit précédemment, il pourra ajouter le produit à partir d'ici uniquement, dans le cas où il est toujours disponible. Ceci sera indiqué dans le coin supérieur droit de l'écran.

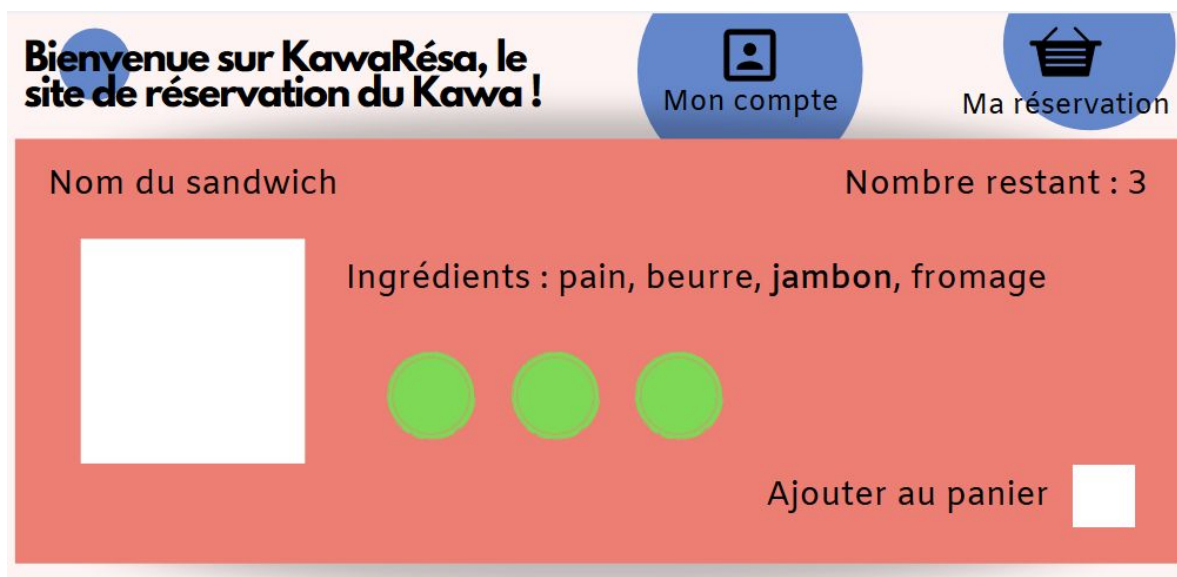


Figure 9: Détails du produit

A partir de sa page d'accueil, l'utilisateur a accès directement à son panier. Dans le cas où celui-ci n'a ajouté aucun produit à son panier, il ne pourra valider sa réservation.



Figure 10 : Panier vide

Si l'utilisateur a au préalable ajouté des produits à son panier, la liste des produits qu'il a sélectionné sera visible. Les attributs visibles sont le nom des produits, les ingrédients, le prix, la formule ainsi que la quantité. La quantité maximale de produit pour une réservation est limitée à 3. Il est important de noter qu'il n'est pas possible de modifier sa réservation mais seulement de la supprimer.

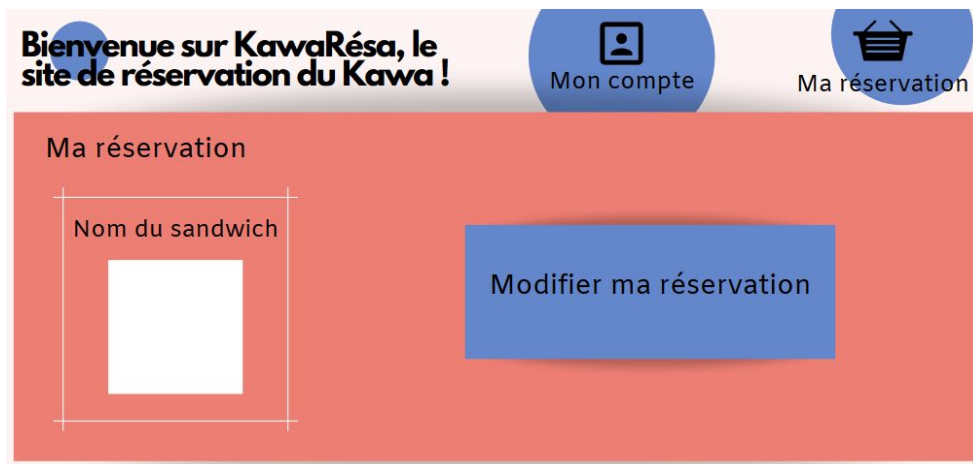


Figure 11 : Réservation

3.2.2. Maquettes gestionnaire

Si la personne est identifiée comme étant un gestionnaire, elle arrive sur une page avec plusieurs options. Elle peut aller gérer les stocks, les réservations, les produits ou encore aller sur l'espace " Mon compte".

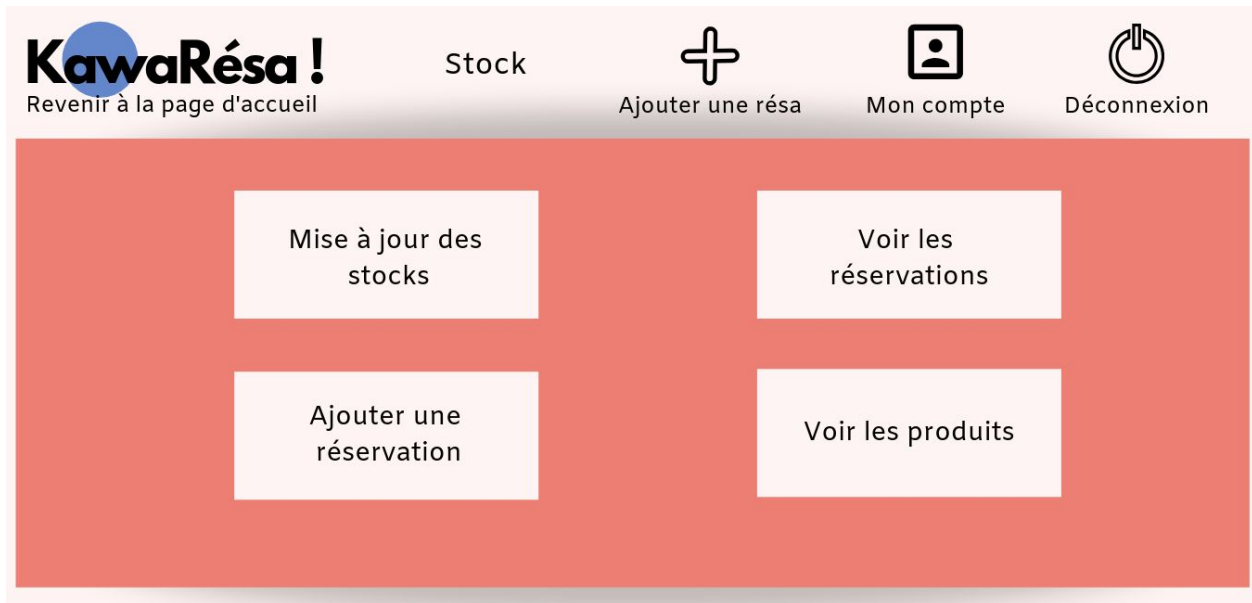


Figure 12 : Accueil gestionnaire

Dans le cas où le gestionnaire décide d'aller mettre à jour les stocks, il arrive sur la **figure 13**, c'est-à-dire la liste de tous les produits avec un bouton pour pouvoir mettre le nombre d'unités disponibles pour chaque produit.

KawaRésa!
Revenir à la page d'accueil

Stock

Ajouter une résa

Mon compte

Déconnexion

Mise à jour des stocks

Sandwichs

Jambon beurre	3	Jambon beurre	
Jambon fromage		Jambon beurre	
Volaille		Jambon beurre	

Figure 13 : Mise à jour des stocks

Il peut aussi avoir accès à la liste des produits tout comme l'utilisateur. Il aura la possibilité de modifier le produit qu'il est en train de consulter.

KawaRésa!
Revenir à la page d'accueil

Stock

Ajouter une résa

Mon compte

Déconnexion

Ajouter un produit

Les sandwichs

Sandwich 1

Sandwich 2

Les paninis

Panini 1

Panini 2

Les salades et plats chauds

Salade de la semaine

Salade 1

Figure 14 : Liste pour ajouter un produit

Si un nouveau produit est commercialisé (telle que la salade de la semaine par exemple), le gestionnaire pourra sélectionner “ajouter un produit” pour l’ajouter à la liste des produits disponibles. Pour cela il devra entrer son nom, ses ingrédients, ses allergènes, son prix, sa formule et si possible une photo du produit.

Figure 15 : Ajouter un produit

Il se peut qu’un utilisateur ne passe pas directement par l’application pour réserver, de ce fait nous avons pensé à permettre au gestionnaire de générer une réservation pour autrui. L’utilisateur qui optera pour cette démarche devra s’être inscrit sur la web-application avant cette opération.

Figure 16 : Ajouter une réservation

Etant donné que le rôle du gestionnaire est entre-autre de préparer les réservations, celui-ci a à disposition la liste des réservations passées pour le jour-même. Toutes les réservations sont de base à l'état "en cours", si l'utilisateur vient récupérer sa commande elle passe à l'état "récupérée". Ce procédé se fera directement en cours de service par le gestionnaire qui appuiera sur le bouton "commande récupérée". A la fin du service, le gestionnaire aura la possibilité d'appuyer sur le bouton "bannir" des commandes qui n'ont pas été récupérées. Les utilisateurs ayant passé ces commandes n'auront alors plus accès à la web-application pendant deux semaines la première fois, puis trois semaines la deuxième fois. Au bout de la troisième fois, l'utilisateur n'aura plus du tout accès à la web-application.

Les réservations

Télécharger en PDF

Aujourd'hui

Sandwichs

Jambon beurre	3	✗	Jambon beurre	3	
Jambon fromage	3		Jambon beurre		
Volaille	3	✗	Jambon beurre		

Figure 17 : Liste des réservations

Dans le cas où le Kawa aurait un problème de connexion internet, nous avons prévu la possibilité de télécharger sous forme de PDF la liste des réservations du jour.

3.2.3. Maquettes administrateur

Tout comme les autres types de compte, l'administrateur pourra consulter ses informations et modifier son mot de passe en cliquant sur "Mon compte". Une fois connecté, l'administrateur aura un menu telle que dans la figure 17.



Figure 18 : Menu administrateur

La particularité du compte administrateur est qu'il a accès à la liste des comptes utilisateurs de la web-application. Il aura la possibilité de supprimer le compte d'un utilisateur.

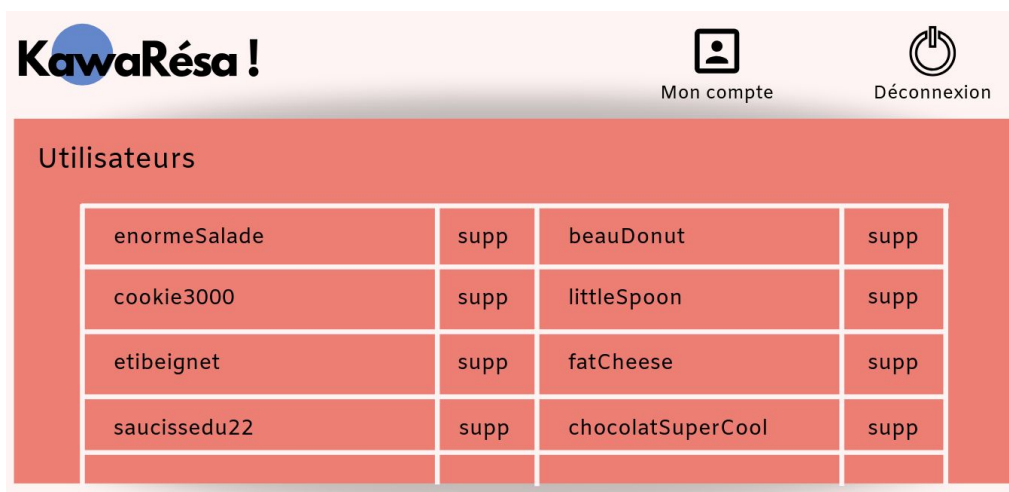


Figure 19 : Liste des utilisateurs

Nous avons pris en compte le fait que chaque année, les comptes seront réinitialisés afin qu'il n'y ait pas une surcharge inutile de comptes dans la base de données. L'administrateur aura pour rôle d'entrer un code de sécurité afin de pouvoir réaliser la réinitialisation.



Figure 20 : Remise à zéro

3.3. Schéma Navigationnel d'Interface (SNI)

Le Schéma Navigationnel d'Interface plus couramment appelé SNI est un modèle conceptuel permettant de représenter les exigences fonctionnelles. Il se lit toujours de haut en bas et permet de concevoir et de représenter l'enchaînement du dialogue entre le logiciel et l'utilisateur, tout en tenant compte du comportement supposé de ce dernier. L'utilisateur pourra alors naviguer entre les différentes informations fournies par ledit SNI.

Il ne faut pas oublier qu'il est indépendant de la plate-forme physique que ce soit sur mobile ou sur un ordinateur.

Lors de notre projet nous avons donc réfléchi à un SNI qui serait adapté à l'utilisateur tout en répondant aux besoins du logiciel et nous avons donc convenu ceci.

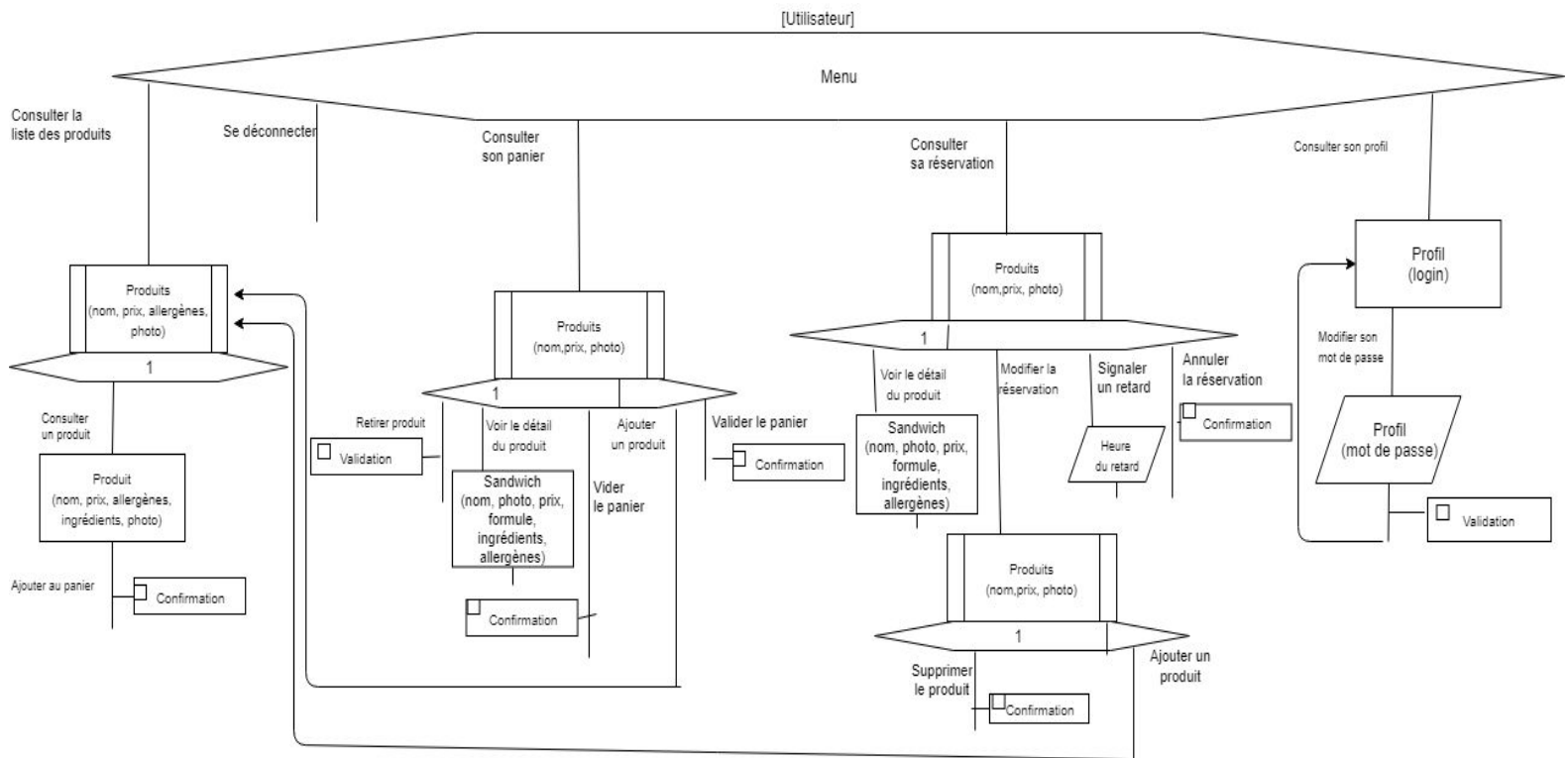


Figure 21 : Partie Utilisateur

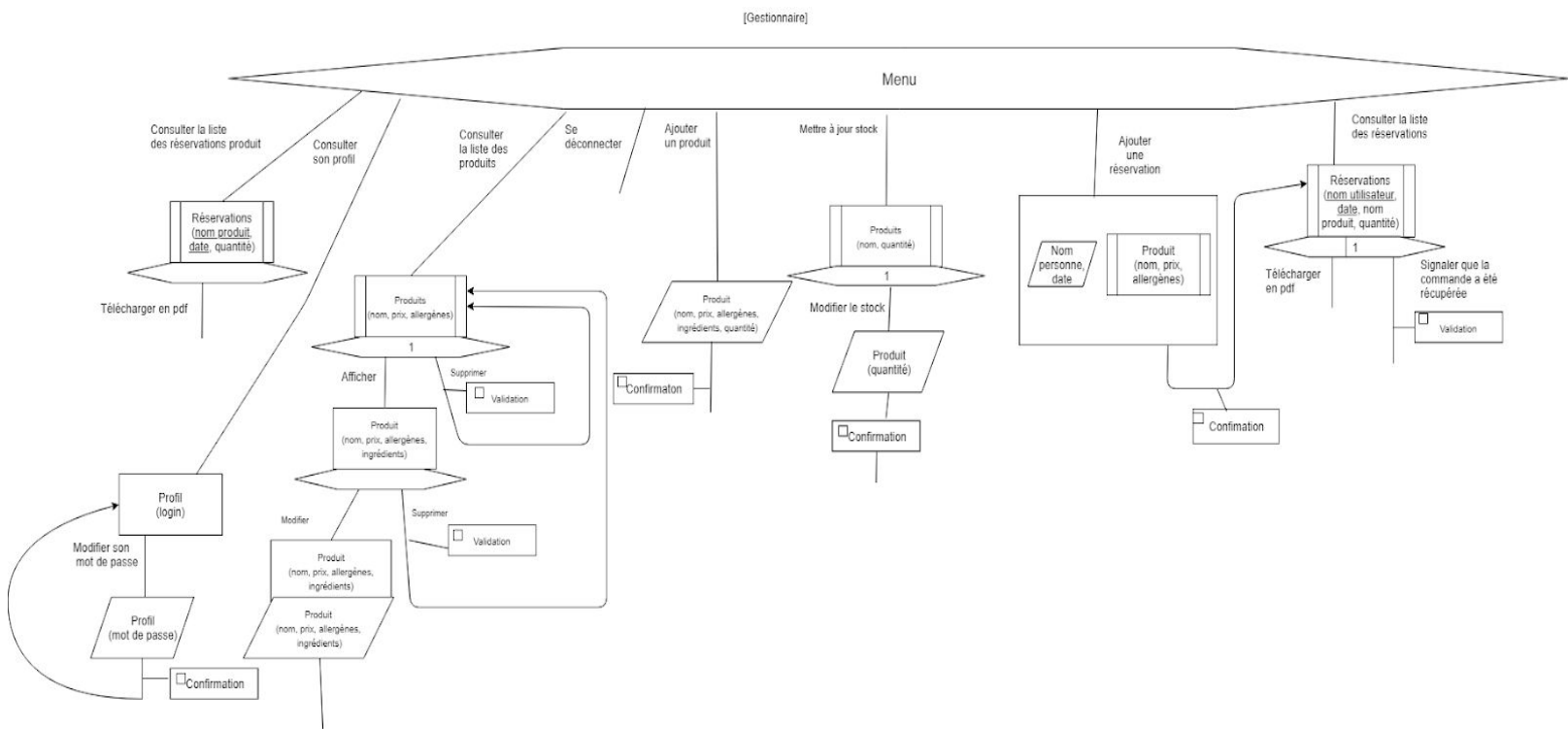


Figure 22 : Partie Gestionnaire

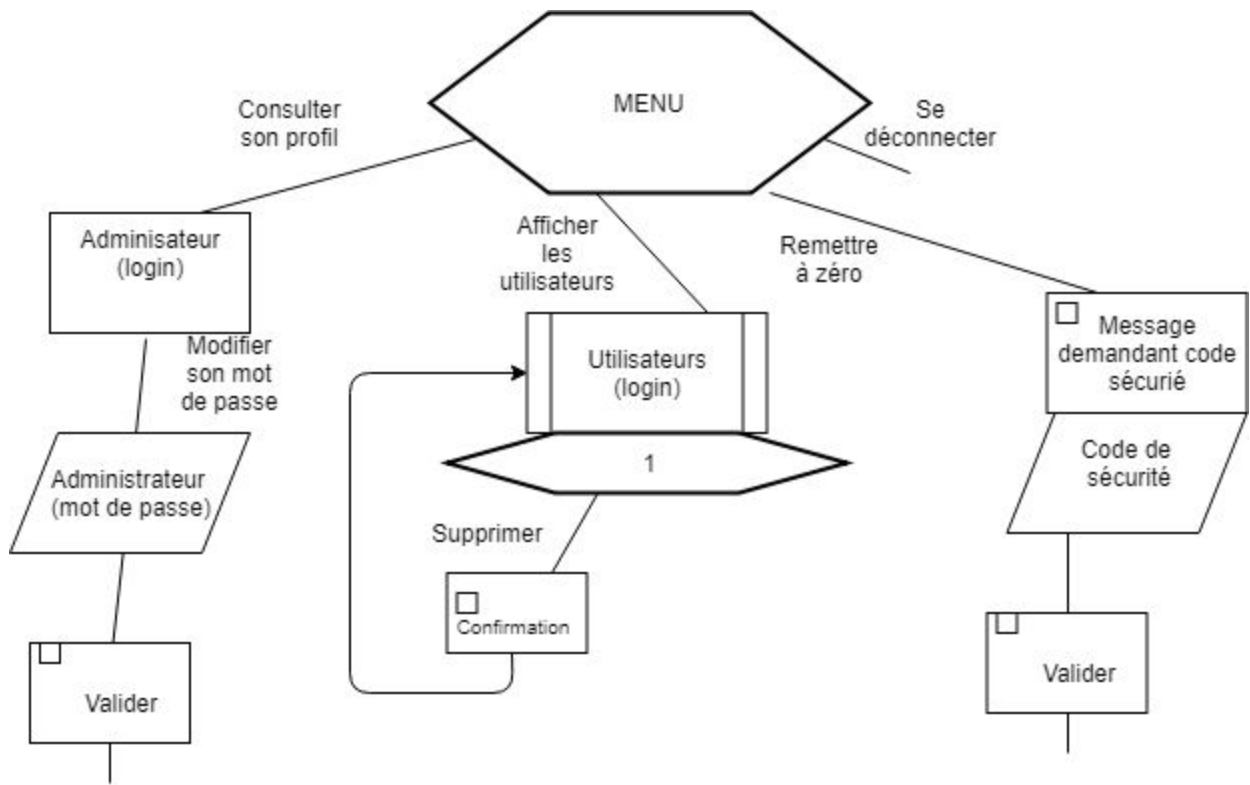


Figure 23 : Partie Administrateur

3.4. Diagramme de classe

Lors de la conception de ce projet, nous avons dû réfléchir aux différents objets qui allaient devoir être pris en compte et la façon dont ils allaient interagir entre eux. Après plusieurs réflexions, nous sommes finalement arrivés au diagramme de classe suivant :

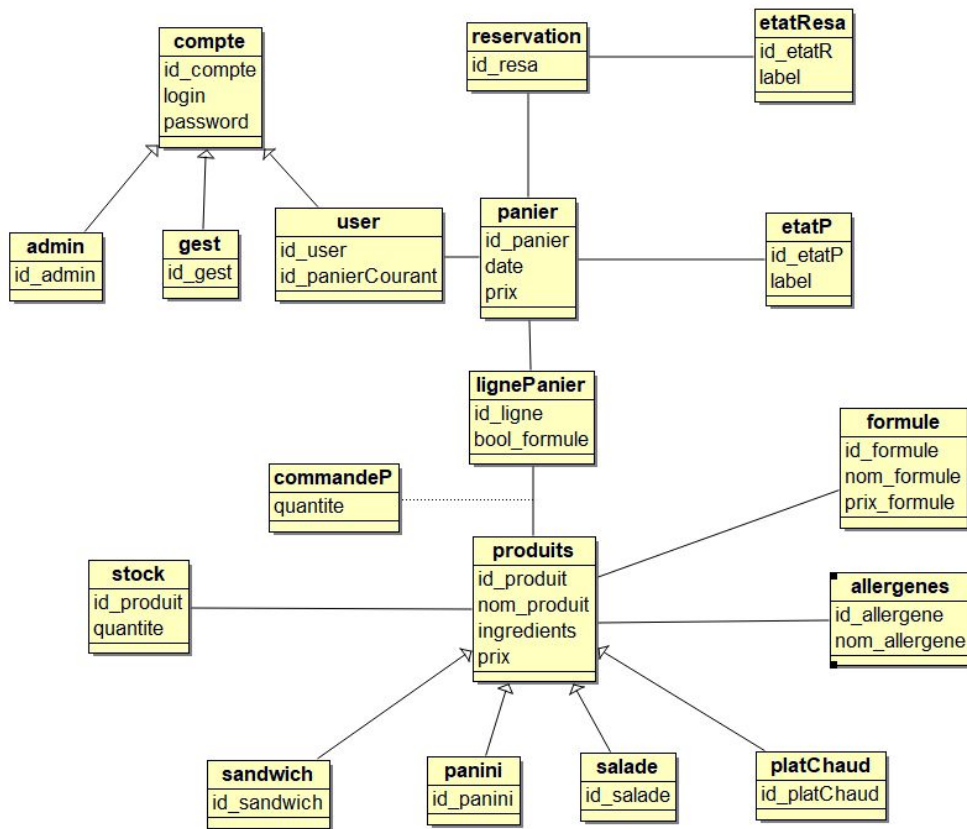


Figure 24 : Diagramme de classe final

Nous sommes passés par plusieurs étapes pour arriver à ce diagramme, afin de pouvoir conserver une même méthode pour l'ajout ou la modification d'un produit et donc ne pas avoir à fonctionner de deux manières différentes. Ce diagramme est la version la plus optimisée que nous ayons pu trouver pour répondre à notre problème.

4. Réalisation

4.1. Le Front-End

La partie Front-End de notre projet a été réalisée en HTML5 et CSS3, ainsi que Bootstrap. Comme pour le Back-End, l'idée principale était de travailler avec Slim. Nous avons donc fait des recherches et il s'est avéré que Twig est un template du langage PHP souvent utilisé avec le framework Slim. De plus, l'apprentissage de Twig aurait pu apporter de nouvelles compétences aux développeurs. Cependant, comme pour Slim, l'apprentissage de ce template était trop fastidieux dû au peu de ressources disponibles. Lorsque nous avons décidé d'arrêter de travailler avec Slim, nous avons donc décidé que la partie Front-End se ferait classiquement avec HTML5, CSS3 et Bootstrap.

Pour ce qui est de l'organisation de ce code, il a été programmé avec une classe "hidden". C'est-à-dire qu'il existe une page "index" pour chaque utilisateur: "administrateur", "gestionnaire" et "utilisateur". Toutes les pages de chaque personne utilisant la web-application sont programmées sur la même page HTML nommée "index", "index_gestionnaire" et "index_admin". Cette classe va permettre de cacher les pages dont on n'a pas besoin. Les différentes pages sont donc codées dans des blocs auxquels on applique la classe hidden. Normalement, en fonction de la page dont on a besoin, on appliquera ou pas la classe hidden.

Le Front-End devait suivre le plus fidèlement possible les maquettes et être fait pour tous types de supports. Finalement toutes les pages ont été réalisées mais uniquement pour le support ordinateur pour l'instant. Nous pensons qu'avec deux semaines de plus, la partie Front-End aurait pu être totalement terminée sur tout support.

4.2. Le Back-End

La partie Back-End de la web-application a été réalisée en utilisant le framework CodeIgniter du langage PHP. Nous avons réparti les rôles de manière à être quatre sur cette phase du développement ce qui nous permettait d'être sûrs d'avoir le maximum de fonctionnalités disponibles. Notre web-application a été réalisée avec plusieurs pages et non en single-page. Nous avons initialement commencé le développement en utilisant le framework Slim mais un problème de Driver PostgreSQL ainsi qu'un manque de connaissances sur cette technologie nous a empêché de continuer plus loin avec celui-ci. Afin d'éviter de perdre encore plus de temps nous avons décidé de reprendre sur des bases que nous maîtrisons et donc de recommencer toute la phase de développement en utilisant le framework CodeIgniter.

Il fallait tout d'abord que n'importe quelle personne utilisant l'application puisse se connecter à un compte ; distingué entre utilisateur, gestionnaire ou administrateur. Nous avons donc créé des identifiants spécifiques pour le compte des gestionnaires et celui de l'administrateur.

Il faut néanmoins gérer la création de comptes utilisateurs. Pour cela nous avons pensé à utiliser la fonction mail() du langage PHP qui nous permettait de bien vérifier que l'adresse mail de l'utilisateur existait, cependant l'envoi de mail au sein de l'IUT est sécurisé et cette fonction n'est donc pas disponible. Par la suite nous avons donc réalisé une inscription basique avec simplement un pseudonyme ainsi qu'un mot de passe tout en restant anonyme et ainsi respecter le RGPD.

Une fois la création de compte terminée, ainsi que la connexion, il nous fallait gérer toute la partie liée aux produits et à leur gestion. Nous avons donc décidé de réaliser un tableau regroupant tous les produits et de les afficher avec leurs différents attributs : leurs ingrédients, leur prix, leur disponibilité et une image pour illustrer le produit. Nous n'avons cependant jamais reçu les photos des produits, nous avons donc

mis la même image pour tous les produits afin de montrer à quoi la page devrait ressembler.

A partir de cette liste, l'utilisateur a la possibilité d'accéder à son panier pour ajouter au maximum trois produits. L'utilisateur pourra par la suite retirer un produit de son panier s'il le souhaite, en cas d'erreur ou bien s'il change d'avis, grâce à un bouton spécifique. Il pourra ensuite transformer son panier en une réservation avec ce qu'il se trouve dans son panier. Après la réservation il ne sera plus possible de revenir en arrière. En effet, les fonctionnalités permettant de vérifier les conditions sur les horaires de réservation et d'annulation ne sont pas intégrées sur la web-application. Il n'y a pas non plus de support papier disponible à partir de la web-application. Ces différentes fonctionnalités n'ont pas été réalisées par manque de temps et de connaissances.

Le gestionnaire a la possibilité d'ajouter un produit avec tous les attributs nécessaires puis de le modifier ou de le retirer (comme le sandwich du mois par exemple). Il a également accès à la liste de tous les produits et de leur stock disponible et peut mettre à jour la quantité disponible d'un produit afin d'informer les utilisateurs des produits qui sont à leur disposition.

Enfin, il aura accès à la liste des réservations par produit (avec le nom du produit et la quantité réservée pour ce produit) ou par utilisateur (avec le login de l'utilisateur et le produit réservé). Lors du service, il pourra signaler qu'un utilisateur vient bien récupérer sa réservation. Si l'utilisateur ne vient pas récupérer sa réservation, alors un système de bannissement temporaire de deux, puis trois semaines a été mis en place afin d'éviter ce genre d'abus. Au bout de la troisième fois, l'utilisateur est banni définitivement. Cette fonctionnalité n'a cependant pas encore été faite par manque de temps et de connaissances.

L'administrateur quant à lui a accès à une liste des comptes utilisateurs et pourra en supprimer. Il pourra également remettre à zéro la totalité des comptes utilisateurs en

début d'année afin de ne pas stocker des comptes qui ne seront plus actifs après le départ des étudiants de l'IUT.

Enfin, les utilisateurs, les gestionnaires et l'administrateur auront la possibilité d'accéder à leurs informations personnelles et changer leur mot de passe depuis un espace "Mon compte".

4.3. La base de données

Pour la mise en place de la base de données, nous avons tout d'abord pensé à l'utilisation du langage MySQL. C'est un langage que nous ne connaissons pas et nous voulions pouvoir l'apprendre. Cependant, à force d'avancer dans le projet et de voir toutes les nouvelles technologies que nous allions devoir apprendre à manipuler, nous avons préféré revenir vers le langage PostGreSQL. De plus, ce langage était compatible avec les serveurs présents à l'IUT.

Puisqu'un diagramme de classe a été mis au point pendant la période de conception, la construction de la base de données s'est appuyée sur ce dernier. Cependant, comme le diagramme de classe a été modifié au cours du projet pour permettre son optimisation, la structure de la base de données a elle aussi été modifiée plusieurs fois. De plus, comme nous avons créé des fichiers CSV pour peupler plus rapidement la base (ces fichiers ayant été créés à la main à partir des informations que nous avons pu récupérer auprès du personnel du Kawa), il a également fallu modifier chaque fichier un à un, tant au niveau de la donnée que de la structure.

Enfin, après avoir obtenu une structure qui répondait pleinement aux besoins du projet ainsi que des fichiers de peuplement conformes, nous avons ajouté des triggers afin de sécuriser un peu plus le projet.

4.4. Maintenance

Etant donné que nous sommes étudiants en deuxième année, nous ne serons plus présents à l'IUT de Lannion dans les années qui viennent, nous ne serons donc plus aptes à effectuer la maintenance de la web-application. Cependant, si la web-application est réellement mise en place au Kawa, cela veut dire que les droits auront été cédés au CROUS et donc qu'ils pourront, s'ils le souhaitent, charger des personnes de leur organisme d'effectuer une maintenance.

4.5. Évolutions futures

Nous avons songé à plusieurs pistes d'amélioration pour ce projet :

- Développer la web-application en single-page, afin de fluidifier la navigation ;
- Intégrer du code dans les langages NodeJS et JQuery dans le même but, ainsi que de permettre un affichage sans rafraichissement de page ;
- Pouvoir gérer les plages d'horaires, c'est-à-dire empêcher les utilisateurs de réserver un produit après 10h15 et les empêcher de modifier la réservation après 12h ;
- Créer un générateur automatique de login amusants en rapport le Kawa (comme "cookie3000" par exemple), lors de la création d'un compte ;
- Permettre un affichage de l'historique de réservation par utilisateur et permettre des suggestions de produits en suivant ce dernier ;
- Faire des statistiques des ventes grâce à l'historique des réservations.

5. Conclusion

Nous avons passé presque la totalité de notre seconde année à l'IUT sur ce projet, et malgré le fait d'avoir eu beaucoup de soucis tout le long de sa réalisation, nous estimons avoir atteint notre objectif de départ qui était de rendre un projet qui répond aux principaux besoins. Cependant, il reste de nombreux points à améliorer.

Ce projet nous a permis de nous rendre compte de l'ampleur que peut avoir un vrai projet, que ce soit sur la durée, la difficulté de développement ou les problèmes rencontrés. Nous avons réussi à contourner les problèmes sur lesquels nous sommes tombés avec plus ou moins de difficulté. Nous avons cependant réussi à avoir une bonne gestion de projet. Nous sommes tout de même déçus de ne pas avoir totalement mené à bien ce projet et espérons qu'il sera repris par d'autres étudiants après notre départ. Nous espérons également que ce projet pourra se concrétiser dans le futur, tout en prenant en compte les pistes de fonctionnalités laissées dans la partie précédente.

6. Table des illustrations

<i>Figure 1 : Bête à cornes</i>	2
<i>Figure 2 : Diagramme des Use Case</i>	5
<i>Figure 3 : Rôles des membres</i>	7
<i>Figure 4 : Diagramme des tâches (WBS)</i>	8
<i>Figure 5 : Diagramme de Gantt prévisionnel</i>	8
<i>Figure 6 : Diagramme de Gantt réel</i>	9
<i>Figure 7 : Connexion</i>	13
<i>Figure 8 : Accueil Utilisateur (Liste produit)</i>	13
<i>Figure 9: Détails du produit</i>	14
<i>Figure 10 : Panier vide</i>	15
<i>Figure 11 : Réservation</i>	16
<i>Figure 12 : Accueil gestionnaire</i>	17
<i>Figure 13 : Mise à jour des stocks</i>	17
<i>Figure 14 : Liste pour ajouter un produit</i>	18
<i>Figure 15 : Ajouter un produit</i>	18
<i>Figure 16 : Ajouter une réservation</i>	19
<i>Figure 17 : Liste des réservations</i>	20
<i>Figure 18 : Menu administrateur</i>	21
<i>Figure 19 : Liste des utilisateurs</i>	21
<i>Figure 20 : Remise à zéro</i>	22
<i>Figure 21 : Partie Utilisateur</i>	23
<i>Figure 22 : Partie Gestionnaire</i>	24
<i>Figure 23 : Partie Administrateur</i>	25
<i>Figure 24 : Diagramme de classe final</i>	26