



Université Mohammed V de Rabat
École Nationale Supérieure d'Informatique et
d'Analyse des Systèmes



RAPPORT DU PROJET JAVA EE

RÉALISATION D'UNE APPLICATION WEB DE GESTION DES FÊTES

Réalisé par :
MOUFID Ibtissam
NASRALLAH MERIEM
MOUJIB Sara

Encadrées par :
Pr. M. ElHamlaoui

Année Universitaire : 2019-2020

REMERCIEMENT

Au terme de ce travail, nous tenons à exprimer nos vifs respects et notre profonde gratitude à tous ceux qui ont contribué à l'élaboration de ce projet dans les meilleures conditions.

De surcroît, nous remercions M. EL HAMLAOUI MAHMOUD, encadrant du projet JEE, pour nous avoir assuré la possibilité d'évoluer en mettant en application les acquis de cette période dans les conditions les plus favorables.

Finalement, nous remercions tout le corps professoral et administratif de l'école nationale d'informatique et d'analyse des systèmes pour la formation qu'ils nous ont prodigué tout au long de ce parcours. Et nous témoignons par le présent travail notre vive reconnaissance à toute personne ayant aidé de près ou de loin à la réalisation de ce projet.

RESUME

Le présent document synthétise le travail réalisé dans le cadre du projet JEE à l'Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes. Ce projet a pour mission de concevoir et de développer une application web se basant sur la technologie Java/JEE.

Partant du processus du demande-offre dans le domaine d'organisation des fêtes, qui nécessite plusieurs efforts des deux côtés client-organisateur soit pour trouver le service convenable ou bien pour faire la publicité au service offert.

En effet , il se voit judicieux que ce système soit implémenté

Pour ce faire, nous avons travaillé sur la conception et la réalisation d'une plateforme permettant de lister les types d'organisations de fêtes existantes et leurs organisateurs.Chaque organisateur possède une page de profil contenant les informations nécessaires,le client voulant effectuer une réservation peut envoyer une demande avec les détails de sa fête.

ABSTRACT

This document summarizes the work carried out within the framework of the JEE project at the National School of Computer Science and Systems Analysis. This project's mission is to design and develop a web application based on Java / JEE technology.

Starting from the request-offer process in the area of party organization, which requires several efforts on both sides client-organizer either to find the suitable service or to advertise the service offered.

Indeed, it makes sense that this system is implemented

To do this, we worked on the design and implementation of a platform to list the types of existing party organizations and their organizers. Each organizer has a profile page containing the necessary information, the customer wanting to make a reservation can send a request with the details of his party.

Liste des abréviations

- **UML** : Unified Modeling Language
- **JSP** : Java Server Pages
- **XML** : Extensible Markup Language
- **HTTP** : Hypertext Transfer Protocol
- **API** : Application Programming Interface
- **JEE** : Java Enterprise Edition
- **HTML** : HyperText Markup Language
- **CSS** : Cascading Style Sheets

Table des figures

2.1	Diagramme des cas d'utilisation de l'organisateur	15
2.2	Diagramme des cas d'utilisation du client	16
2.3	Diagramme de Séquence de l'Organisateur	17
2.4	Diagramme de Séquence du client	18
2.5	Diagramme de classe	19
3.1	Schéma du Pattern MVC	25
3.2	Page de connexion	26
3.3	Nouveau compte	27
3.4	Nouveau menu	28
3.5	Liste réservation	29
3.6	Réponse réservation	29
3.7	Connexion	30
3.8	Accueil.1	31
3.9	Accueil.2	31
3.10	type organisiateur	32
3.11	liste organisiateur	32
3.12	profile organisiateur	33
3.13	demande de réservation	33
3.14	état de réservation	34

Table des matières

Liste des abréviations	4
INTRODUCTION	8
1 Context général du projet	9
1.1 Introduction	10
1.2 Déscription du projet	10
1.2.1 Context general du projet	10
1.2.2 Problematique	10
1.2.3 Périmètre	10
1.2.4 Objectifs	11
1.3 Conclusion	11
2 ANALYSE ET CONCEPTION	12
2.1 Introduction	13
2.2 Spécification des besoins	13
2.2.1 Besoins fonctionnels	13
2.2.2 Besoins non fonctionnels	13
2.3 Analyse des besoins	14
2.3.1 Identification des acteurs	14
2.3.2 Diagramme de cas d'utilisation	14
2.4 Conception de données	16
2.4.1 Diagramme de séquence	16
2.4.2 Diagrammes de classes	18
2.4.3 Description de classes	20
2.5 Conclusion	20
3 REALISATION	21
3.1 Introduction	22
3.2 Environnement de développement	22
3.2.1 Outils utilisés	22
3.2.2 Technologies et langages utilisées	23
3.2.3 La démarches de développement	24
3.3 Implémentation des interfaces	25

3.3.1	Réalisation des interfaces ‘organisateur’	25
3.3.2	Réalisation des interfaces ‘client’	30
CONCLUSION		35
RÉFÉRENCES		36

INTRODUCTION

Les fêtes sont des moments de rassemblements collectifs organisés en l'honneur d'une divinité, d'une personne, d'événements importants ou bien en commémoration de l'anniversaire de ces événements. Dans ce sens vient l'idée de l'application présentée dans ce rapport.

A travers ce projet , nous avons procédé à la simplification du processus d'offre et d'organisation des fêtes en réalisant une application web permettant de gérer ces opérations . Ceci à partir de l'implémentation des interfaces organisateur et client, permettant de proposer des services et de donner le client le choix de chercher, donner son avis et puis réserver un service sans avoir à se déplacer, il peut tous faire à partir l'application PlannerIfy.

Partant de cet objectif , le projet à été réalisé en trois grandes étapes colligées dans le présent rapport :

- En premier lieu , une présentation du contexte général ainsi que l'identification de la problématique et la définition des objectifs du projet.
- Nous abordons par la suite l'analyse et la spécification des besoins, la présentation de l'architecture respectée lors de l'implémentation de notre plateforme, ainsi que la conception du projet ,traduite par la r'ealisation d'un ensemble de diagrammes.
- Finalement, le dernier chapitre concerne les outils utilis'ees lors de la r'ealisation de ce projet, ainsi que la pr'esentation des diff'erentes interfaces impl'ement'ees et leur fonctionnement.

Chapitre 1

Context général du projet

1.1 Introduction

Dans ce chapitre, nous allons situer le projet dans son contexte général, ceci en présentant sa problématique ainsi que les besoins et les objectifs déterminés pour la résoudre.

1.2 Description du projet

1.2.1 Context general du projet

Les visiteurs ont le droit de voir tous les menus disponibles dans PlannerIfy or pour qu'ils puissent réserver ou bien poster des menus ils doivent premièrement avoir un compte.On distingue entre deux types de comptes :

- Organisateur
- Client

L'organisateur ne peut pas réserver ,il est ici pour offrir des services sous forme de menus ,et de répondre aux demandes de réservations des clients

Le Client choisit le type de fête puis le type d'organisateur qu'il veut,après choisir un menu,il a le choix de voir le profil de l'organisateur et de réserver.

1.2.2 Problematique

Un client voulant organiser une fête doit se déplacer entre différents organisateurs pour faire la comparaison entre leurs menus et le prix de chacun d'eux, ce qui prend du temps. L'organisateur doit de sa coté imprimer beaucoup de cartes visites et chercher à les transmettre aux clients possibles cela nécessite du temps et d'argent. Notre application a pour objectif de résoudre ces problèmes ,comment faire pour arriver à ceci donc ?

1.2.3 Périmètre

La cible de l'application est la population marocaine en général.

1.2.4 Objectifs

La complexité de ces processus nous a incité à penser élaborer une solution convenable afin de résoudre les problèmes mentionnés ci-dessus, ceci en réalisant une application permettant de gérer le processus d'offre-demande d'organisation de fêtes les différentes procédures effectuées par chacun des organisateurs et clients. Tout en gérant le flux de mises constituant le déroulement logique de toute demande de réservation.

Afin de répondre à cet objectif , ce projet repose sur la conception et la réalisation d'une plateforme de vente en ligne aux enchères organisant le concept décrit sous deux parties :

- Une partie dédiée au client , où il pourra :
 - Choisir le type d'organisation.
 - Choisir le type d'organisateur.
 - Avoir accès aux menus disponibles.
 - Accéder aux détails du menu désiré.
 - Accéder aux profiles des organisateurs
 - Demander la réservation d'un menu.
- Une partie au profit de l'organisateur , lui offrant la possibilité de :
 - poster des menus
 - Gérer les demandes de réservations de ses menus

1.3 Conclusion

Finalement, une personne intéressée par ces services de PlannerIfy pourrait éviter les procédures classiques et réduire le temps habituel à ces dernières en s'inscrivant sur la plateforme décrite et en choisissant entre les deux profils disponibles .

Chapitre 2

ANALYSE ET CONCEPTION

2.1 Introduction

Dans ce chapitre, nous allons aborder la phase de conception, ceci en clarifiant en premier lieu la vue globale décrite précédemment, en analysant les besoins du système et en décrivant l'architecture générale que nous allons suivre dans la partie réalisation de notre projet. Puis, dans un deuxième lieu nous allons détailler notre choix conceptuel à travers plusieurs diagrammes.

2.2 Spécification des besoins

2.2.1 Besoins fonctionnels

- Consulter les différents menus exposés par les organisateurs.
- Consulter les profils des différents organisateurs ainsi que leurs catalogues.
- Réserver un article ou une offre proposée par un organisateur.
- Donner son avis sur les différentes propositions.

2.2.2 Besoins non fonctionnels

— L'extensibilité :

à tout moment on peut ajouter ou modifier des nouvelles fonctionnalités.

— La sécurité :

les informations ne devront pas être accessibles depuis l'extérieur.

— La performance :

l'application doit réagir dans un délai précis.

— La convivialité :

l'application doit être simple et facile à utiliser.

2.3 Analyse des besoins

Afin de modéliser cette étape, nous avons adopté le langage UML . Ceci permet ainsi : Un gain de précision, un gage de stabilité et l'utilisation d'outils. UML est un support de communication performant : Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes. Son caractère polyvalent et sa souplesse lui font un langage universel.

On a choisi alors de travailler avec UML puisqu'il exprime mieux la vue statique et dynamique des applications web, il est nécessaire alors de faire une analyse très approfondie pour pouvoir dégager les nécessités de développement ainsi que quelques scénarios d'exécution.

2.3.1 Identification des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes qui interagissent directement avec le système étudié.

Deux acteurs sont identifiés dans notre application :

- **L'organisateur** : Personne disposant d'un compte organisateur et désirant exposer des menus.
- **Le client** : Personne désirant réserver un menu.

2.3.2 Diagramme de cas d'utilisation

Un cas d'utilisation est une manière spécifique d'utiliser un système. C'est l'image d'une fonctionnalité du système, d'éclenchée en réponse à la stimulation d'un acteur externe. Les cas d'utilisation apportent une solution aux problèmes de la détermination et de la compréhension des besoins.

Le diagramme ci-dessous décrit l'utilisation que permet l'application aux acteurs décrits précédemment :

Diagramme de cas d'utilisation Organisateur

Un utilisateur disposant d'un compte organisateur sur l'application peut donc :

- Consulter le rating et les commentaires à propos de lui.
- Consulter les demandes de réservations des clients.
- Ajouter ou bien modifier le menu et le prix correspondant.

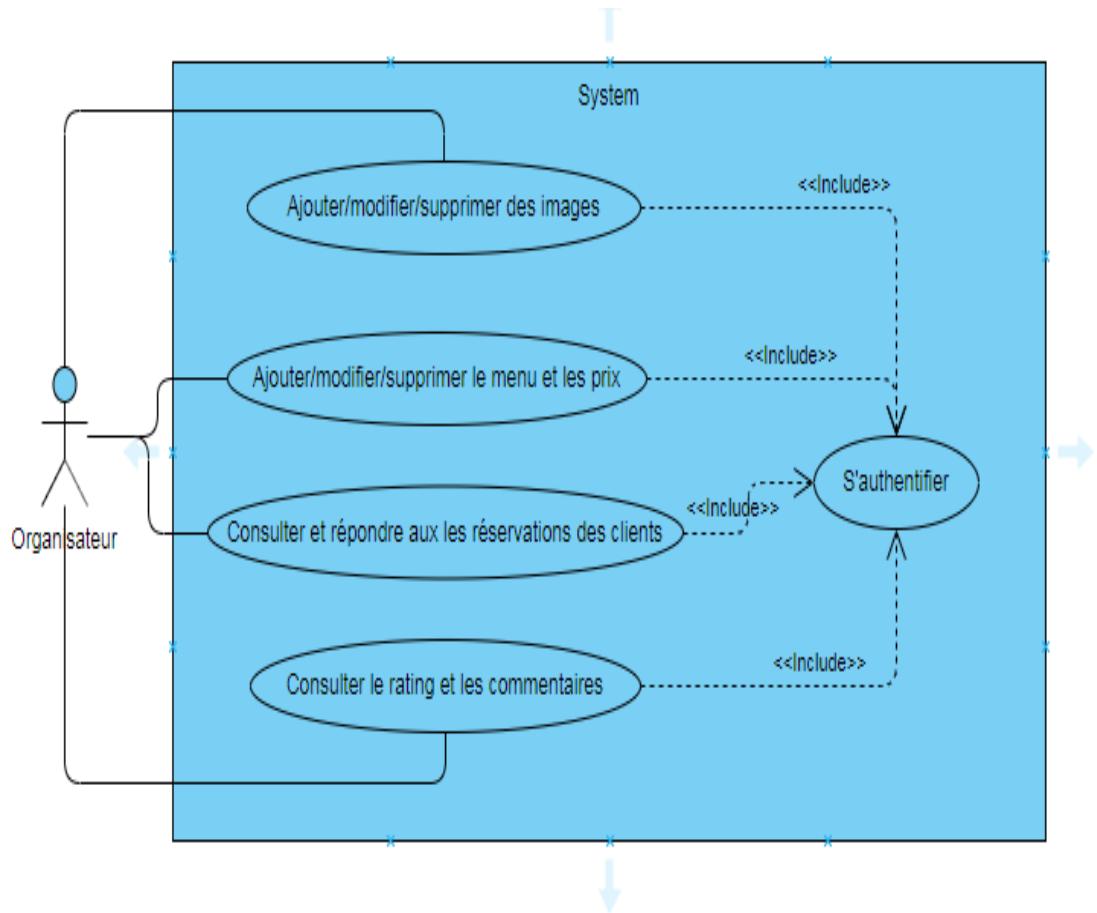


FIGURE 2.1 – Diagramme des cas d'utilisation de l'organisateur

- Ajouter, modifier ou bien supprimer des images.

Diagramme de cas d'utilisation Client

Un utilisateur disposant d'un compte client sur l'application peut donc :

- Choisir une fête à organiser.
- Evaluer et commenter.
- Réserver un organisateur.
- Visiter les profils des organisateurs.

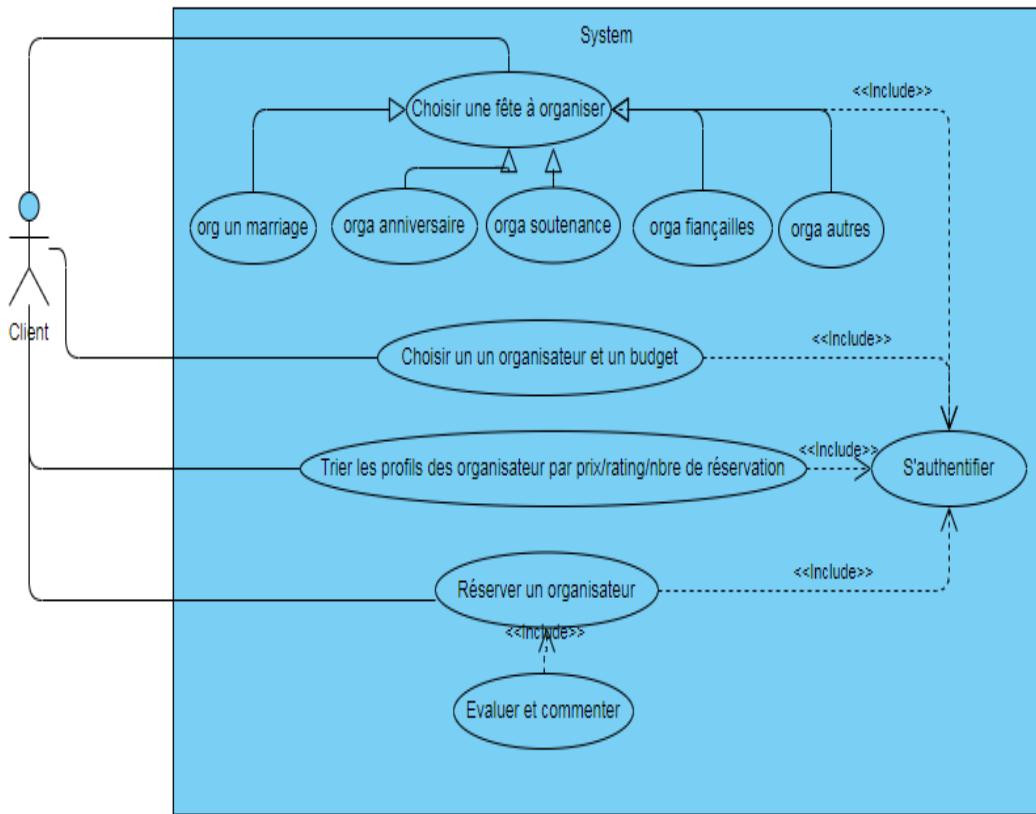


FIGURE 2.2 – Diagramme des cas d'utilisation du client

2.4 Conception de données

2.4.1 Diagramme de séquence

Diagramme de Séquence de l'Organisateur

Après créer un compte l'organisateur doit s'authentifier ,il peut éditer son profil,ajouter un menu ,consulter les réservations et choisir de les accepter ou bien l'ignorer.

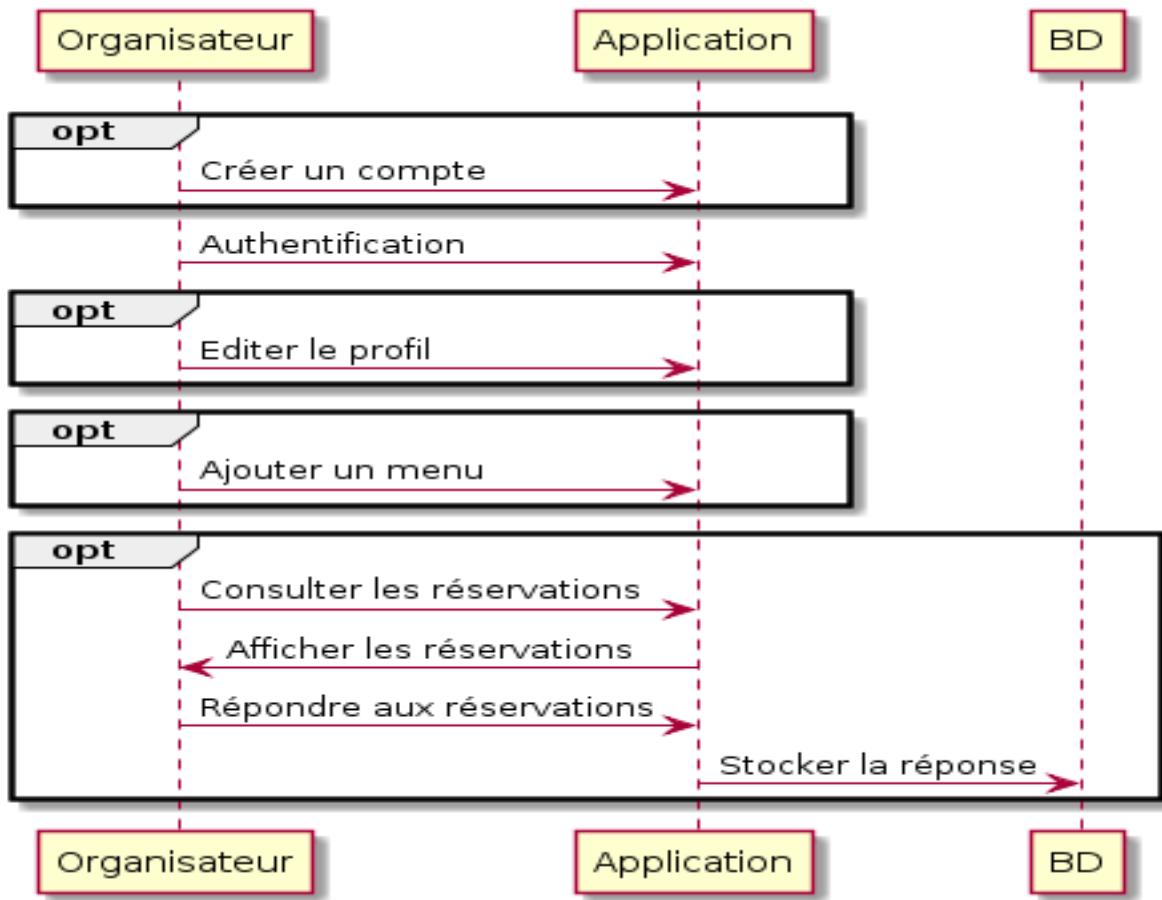


FIGURE 2.3 – Diagramme de Séquence de l'Organisateur

Diagramme de Séquence du client

Le client doit dans un premier temps créer un compte pour effectuer les opérations optionnelle suivante : choisir un événement, visiter un organisateur, réserver un menu.

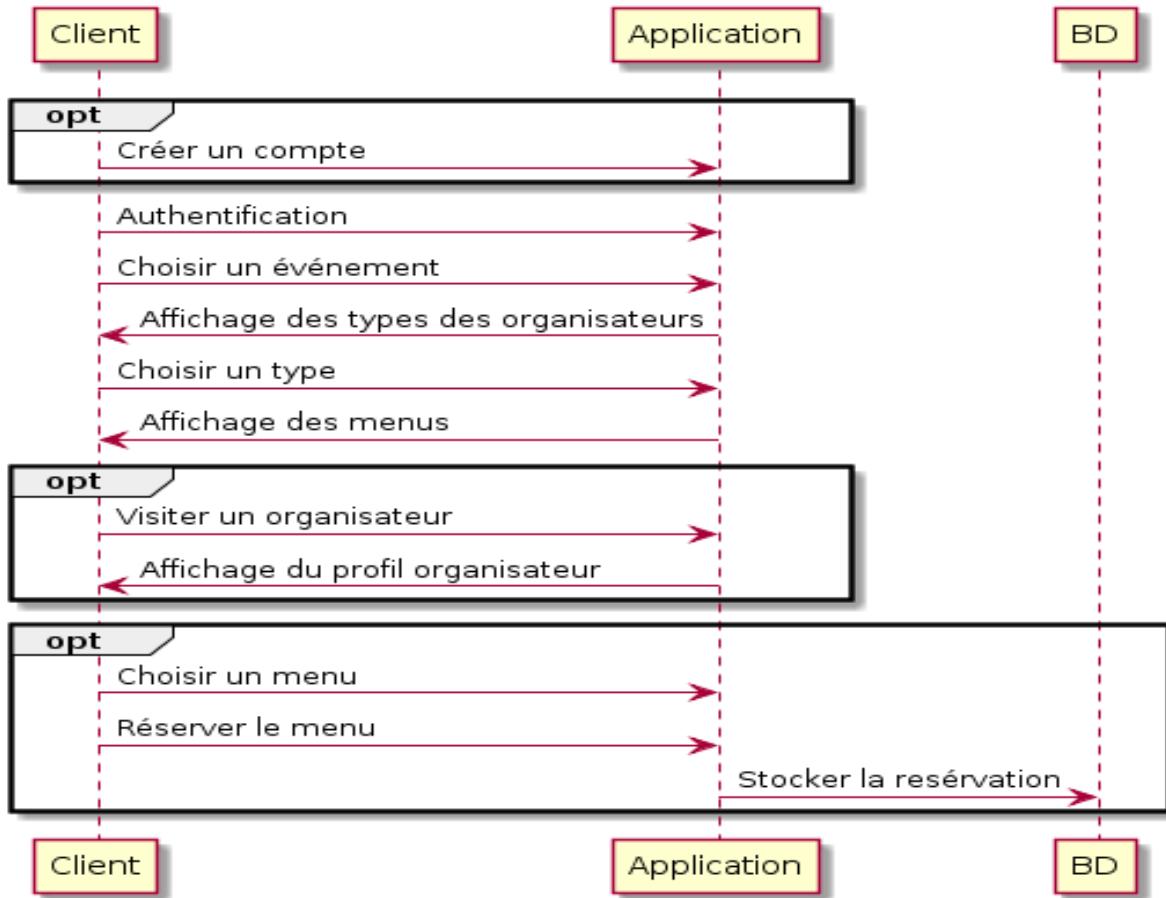


FIGURE 2.4 – Diagramme de Séquence du client

2.4.2 Diagrammes de classes

Le diagramme de classe représente l'un des plus importants diagrammes de la modélisation UML lors de la phase de la conception, il représente les différentes classes métiers, leurs attributs et les relations entre elles.

Classes - Class Diagram

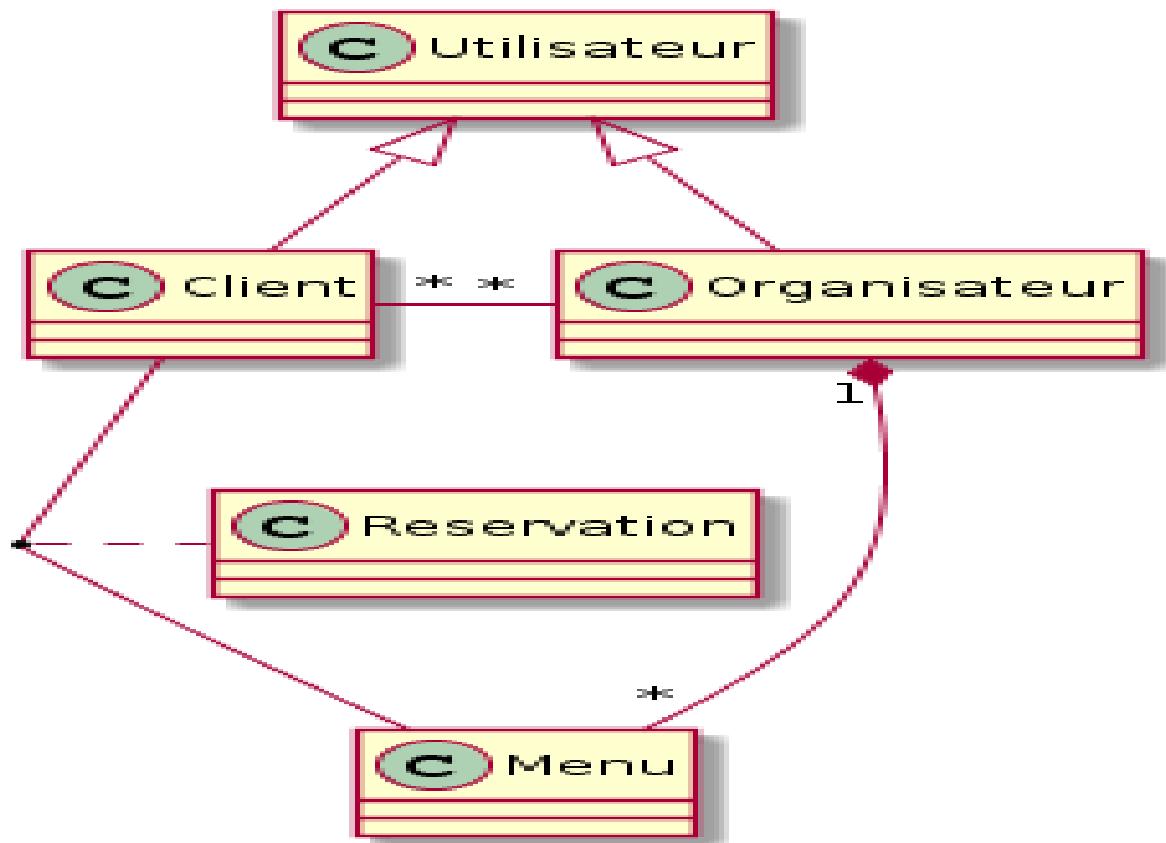


FIGURE 2.5 – Diagramme de classe

2.4.3 Description de classes

Classe	Description
Utilisateur	Cette classe englobe les différents attributs en commun qui caractérisent les utilisateurs de PlannerIfy.
Organisateur	Cette classe hérite de la classe Utilisateur et elle contient d'autres attributs qui sont spécifiés comme type.
Client	Cette classe hérite de la classe Utilisateur.
Menu	Cette classe contient les menus offerts d'organisateurs avec les détails de prix et description.
Réservation	cette classe contient la date de demande de réservation ,menu et détail et la date de réservation.
Catalogue	Cette classe contient les menus d'un organisateur donné.

2.5 Conclusion

Dans ce chapitre, nous avons effectués l'analyse et la conception de notre projet à partir des différents diagrammes de modélisation UML. Quant au chapitre suivant, il est consacré à la réalisation de notre application.

Chapitre 3

REALISATION

3.1 Introduction

Dans ce chapitre, nous allons expliquer les outils de développement utilisés pour élaborer notre application, ensuite montrer les interfaces finales de notre application.

3.2 Environnement de développement

3.2.1 Outils utilisés

Eclipse

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.

Apache Tomcat

Apache Tomcat est un conteneur web libre de servlets et JSP. Issu du projet Jakarta, c'est un des nombreux projets de l'Apache Software Foundation. Il implémente les spécifications des servlets et des JSP du Java Community Process, est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP.

phpMyAdmin

phpMyAdmin (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée principalement en PHP et distribuée sous licence GNU GPL.

Cette interface pratique permet d'exécuter, très facilement et sans grandes connaissances en bases de données, des requêtes comme les créations de table de données, insertions, mises à jour, suppressions et modifications de structure de la base de données, ainsi que l'attribution et la révocation de droits et l'import/export. Ce système permet de sauvegarder commodément une base

de données sous forme de fichier .sql et d'y transférer ses données, même sans connaître SQL.

3.2.2 Technologies et langages utilisées

Java EE

Java EE est une collection de technologies et d'API pour la plate-forme Java conçue pour prendre en charge les applications "d'entreprise" qui peuvent généralement être classées comme des applications à grande échelle, distribuées, transactionnelles et hautement disponibles, conçues pour prendre en charge les exigences commerciales critiques.

La technologie JSP

Le JavaServer Pages ou JSP est une technique basée sur Java qui permet aux développeurs de créer dynamiquement du code HTML, XML ou tout autre type de page web. Cette technique permet au code Java et à certaines actions prédefinies d'être ajoutés dans un contenu statique.

HTML

Le langage HTML est un langage permettant de définir la structure logique d'un document à l'aide de différents éléments de base (titres, paragraphes, liens vers d'autres documents, ...).

HTML n'est pas un langage de traitement de texte en ce sens qu'il ne permet pas de décrire les détails de mise en forme des divers éléments composant un document. La mise en forme dépend du navigateur et des options choisies par le lecteur. Le résultat de la visualisation peut donc varier mais la structuration logique est préservée dans tous les cas.

CSS

Les CSS, Cascading Style Sheets (feuilles de styles en cascade), servent à mettre en forme des documents web, type page HTML ou XML. Par l'intermédiaire de propriétés d'apparence (couleurs, bordures, polices, etc.) et de placement (largeur, hauteur, côté à côté, dessus-dessous, etc.), le rendu d'une page web peut être intégralement modifié sans aucun code supplémentaire dans la

page web. Les feuilles de styles ont d'ailleurs pour objectif principal de dissocier le contenu de la page de son apparence visuelle.

JavaScript

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs² avec l'utilisation (par exemple) de Node.JS.

C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés.

Bootstrap

Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

3.2.3 La démarches de développement

Patron MVC

MVC est un patron de conception (design pattern en anglais) très répandu pour réaliser des sites web. Ce patron de conception est une solution éprouvée et reconnue permettant de séparer l'affichage des informations, les actions de l'utilisateur et l'accès aux données.

Rôles des composants

- La partie Modèle d'une architecture MVC encapsule la logique métier ainsi que l'accès aux données. Il peut s'agir d'un ensemble de fonctions (Modèle procédural) ou de classes (Modèle orienté objet).
- La partie Vue s'occupe des interactions avec l'utilisateur : présentation, saisie et validation des données.

- La partie Contrôleur gère la dynamique de l'application. Elle fait le lien entre l'utilisateur et le reste de l'application.

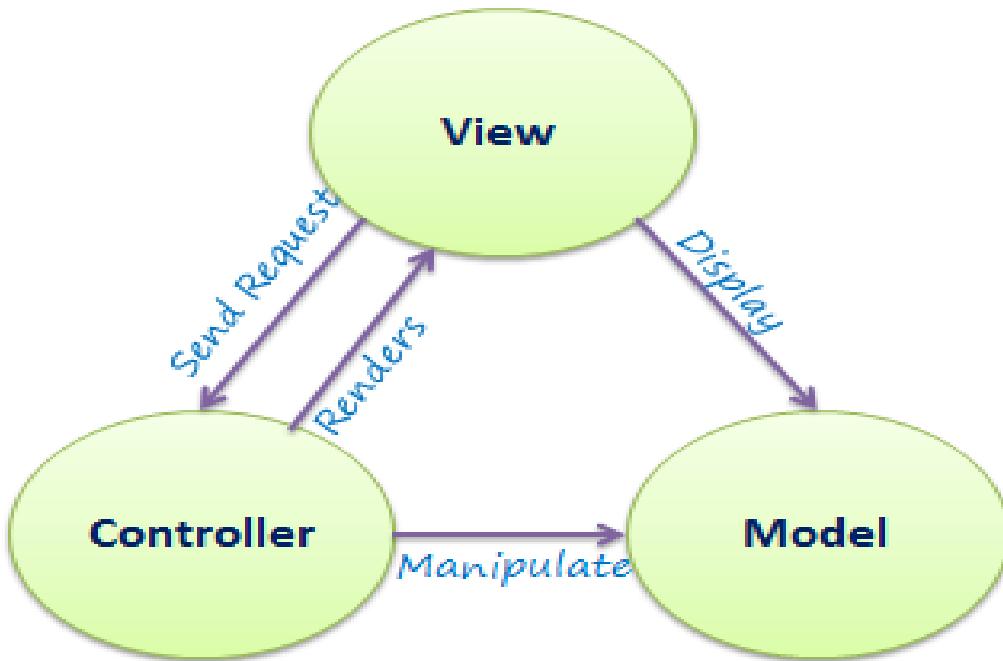


FIGURE 3.1 – Schéma du Pattern MVC

Modèle DAO

L'utilisation de DAO permet de s'abstraire de la façon dont les données sont stockées au niveau des objets métier. Ainsi, le changement du mode de stockage ne remet pas en cause le reste de l'application. En effet, seules ces classes dites "techniques" seront à modifier (et donc à re-tester). Cette souplesse implique cependant un coût additionnel, dû à une plus grande complexité de mise en œuvre.

3.3 Implémentation des interfaces

3.3.1 Réalisation des interfaces ‘organisateur’

Les captures suivantes montrent l'allure de notre application :

1. L'interface de connexion : permet aux différents utilisateurs de s'authentifier s'ils ont déjà un compte sinon ils peuvent créer un.

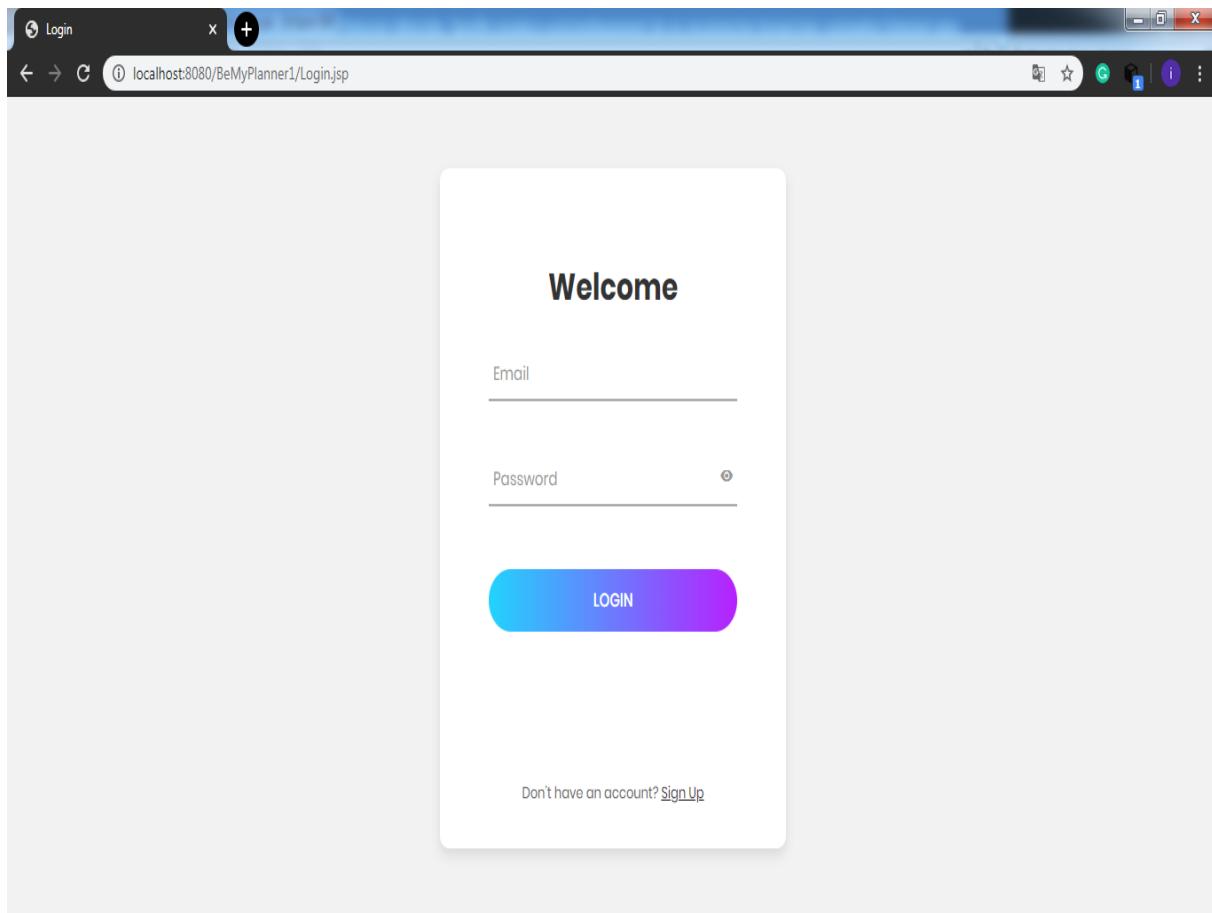


FIGURE 3.2 – Page de connexion

2. Après cliquer sur créer un compte dans la page de connexion, le visiteur est dirigé à cette page où il doit remplir ces champs pour s'inscrire.

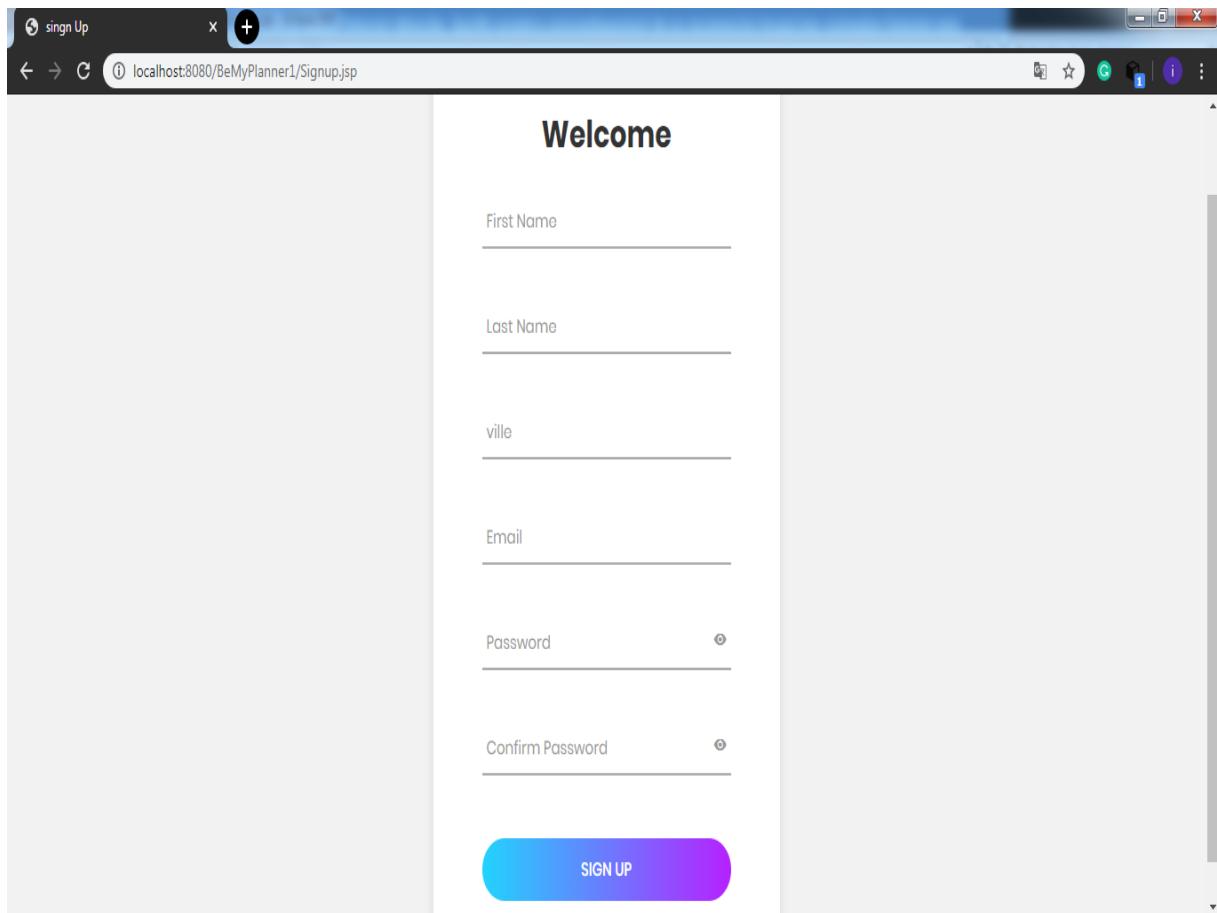


FIGURE 3.3 – Nouveau compte

3. Dans le cas d'organisateur, il peut ajouter des menus

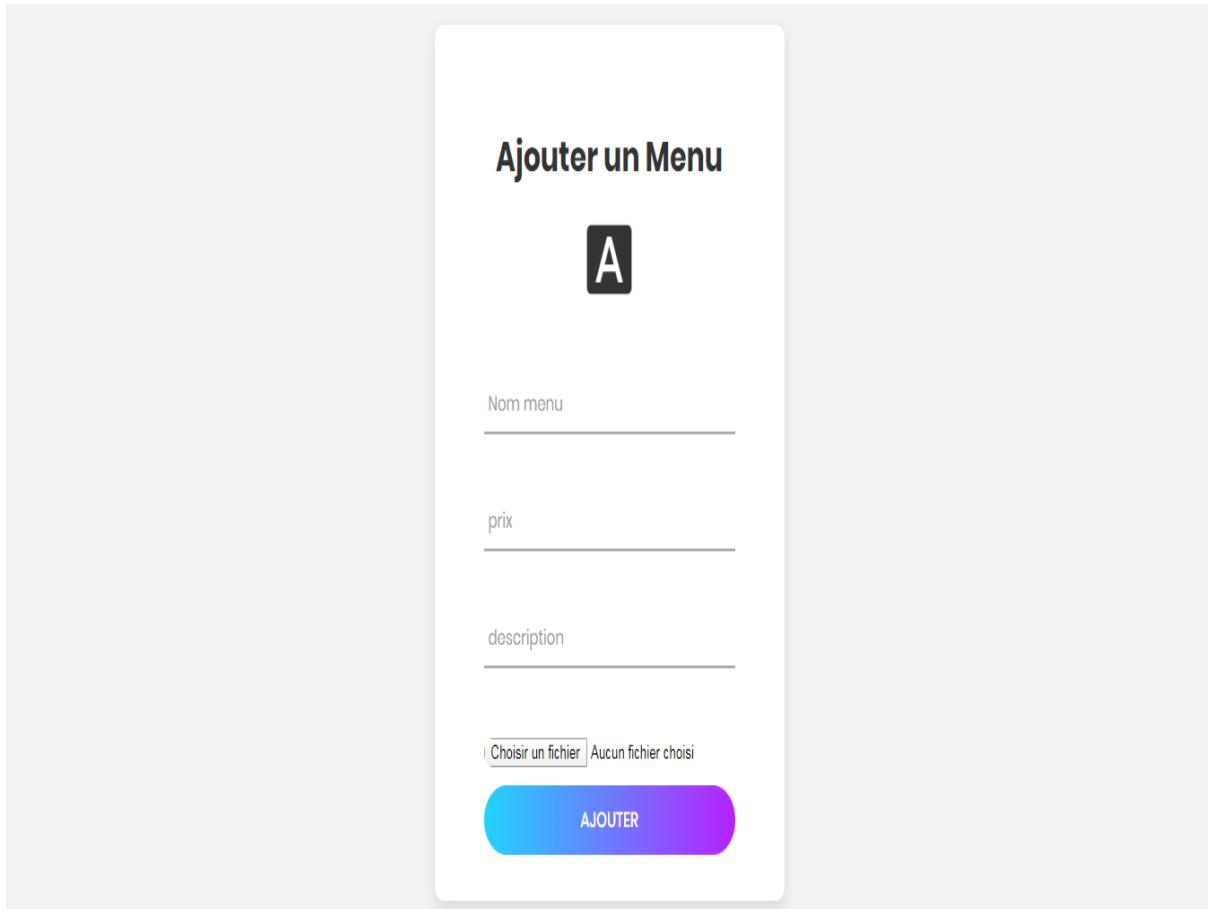


FIGURE 3.4 – Nouveau menu

3. il peut aussi voir la liste des demandes de réservation et choisir de ou de ne pas répondre.

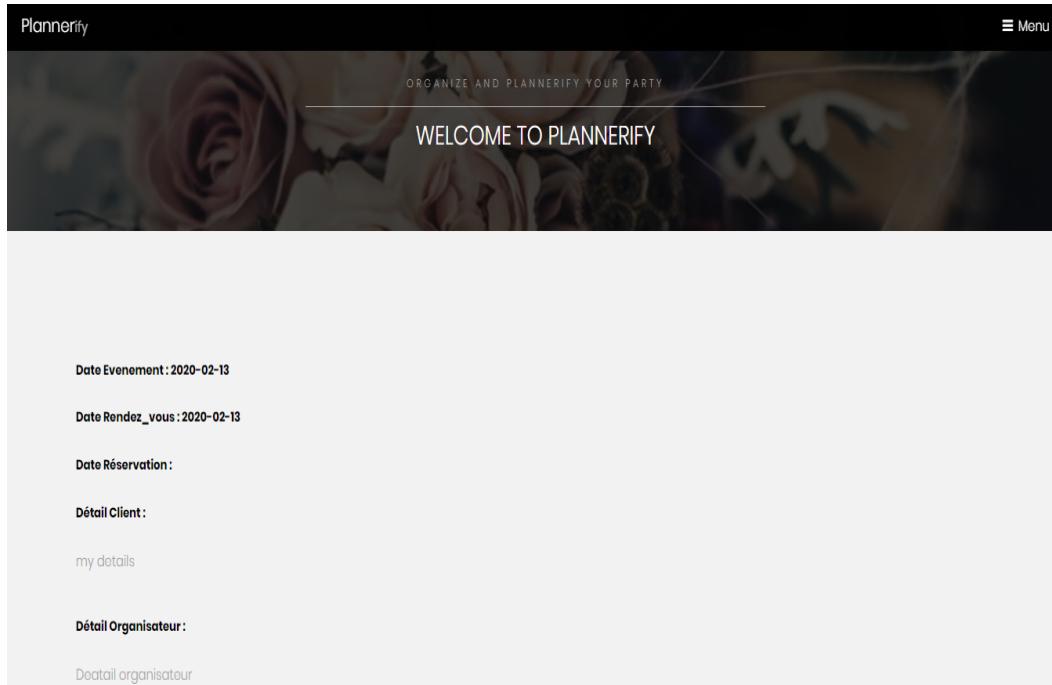


FIGURE 3.5 – Liste réservation

4. s'il a choisi de répondre il doit remplir les champs de détails du demande.

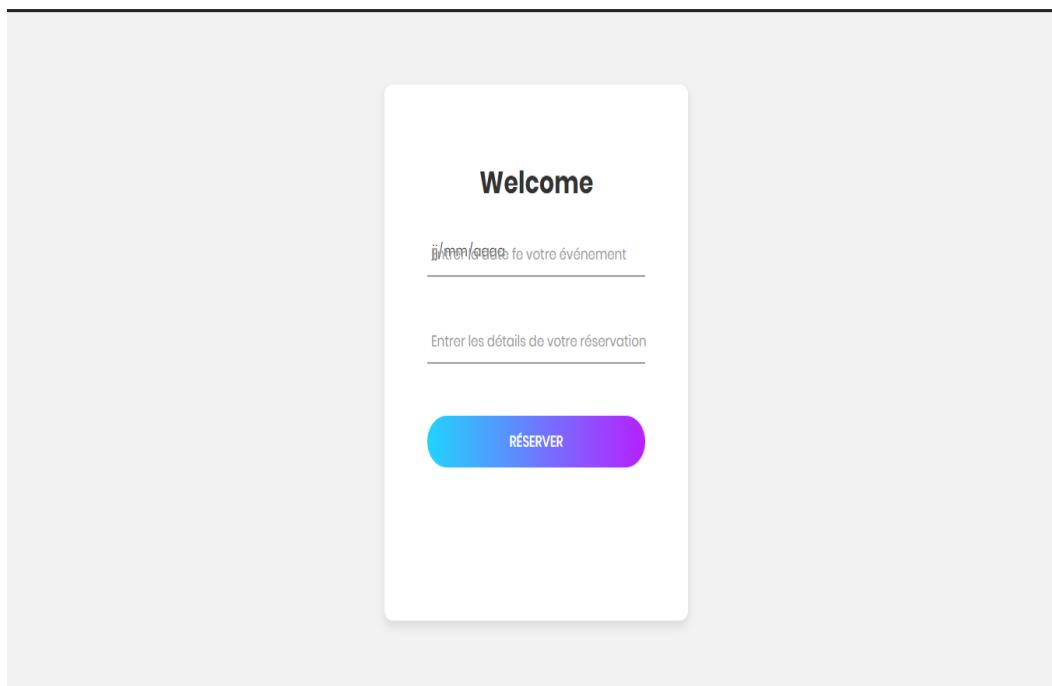


FIGURE 3.6 – Réponse réservation

3.3.2 Réalisation des interfaces ‘client’

1. le client lui aussi doit soit se connecter s'il a déjà un compte ou bien s'inscrire.

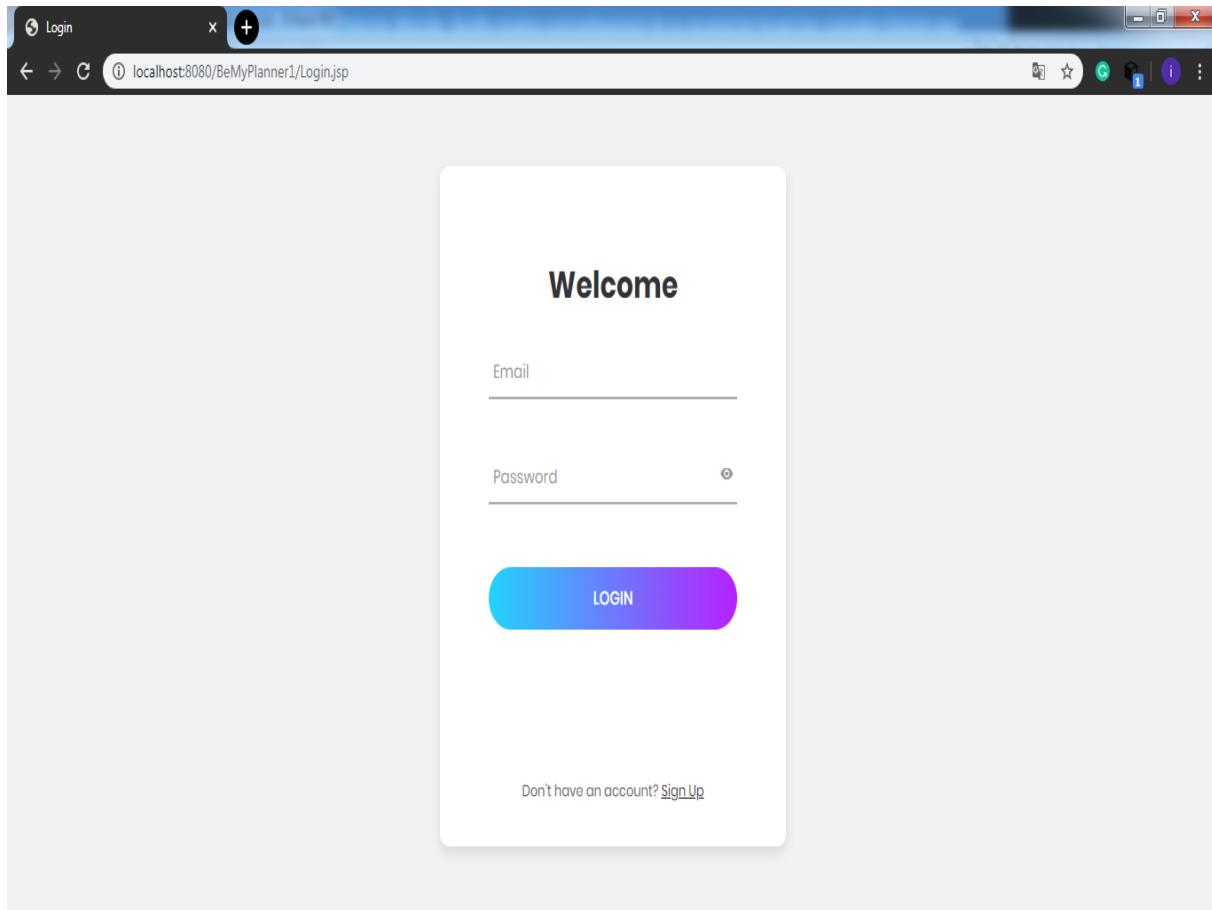


FIGURE 3.7 – Connexion

2. Après avoir se connecter ,le client est dirigé vers la page accueil,où il peut choisir le type d'organisation qu'il désire.

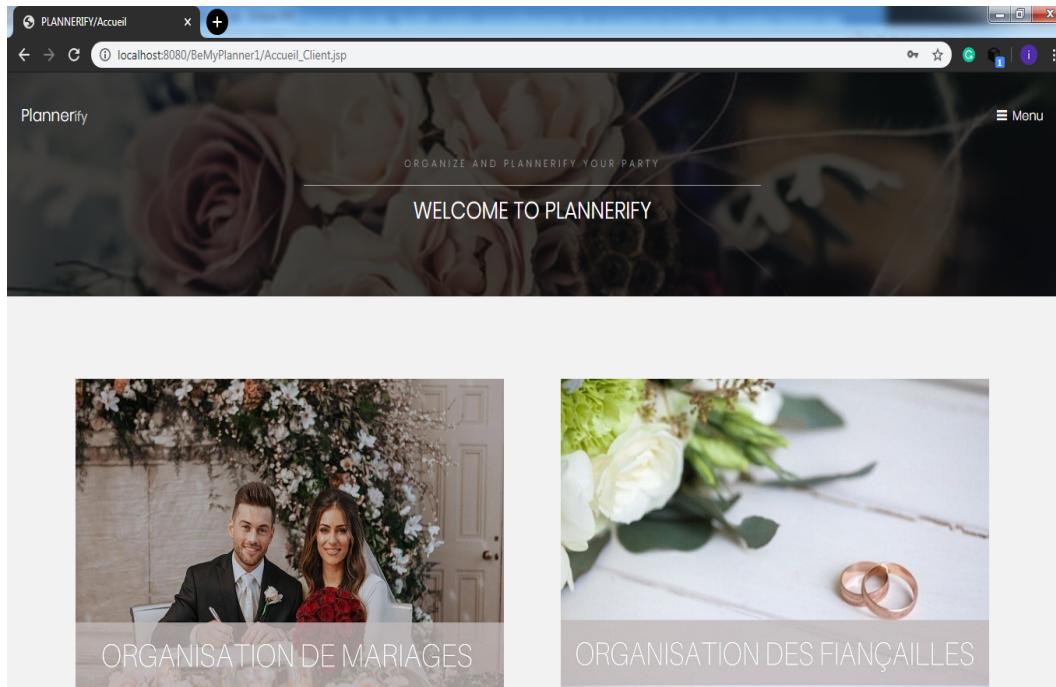


FIGURE 3.8 – Accueil.1

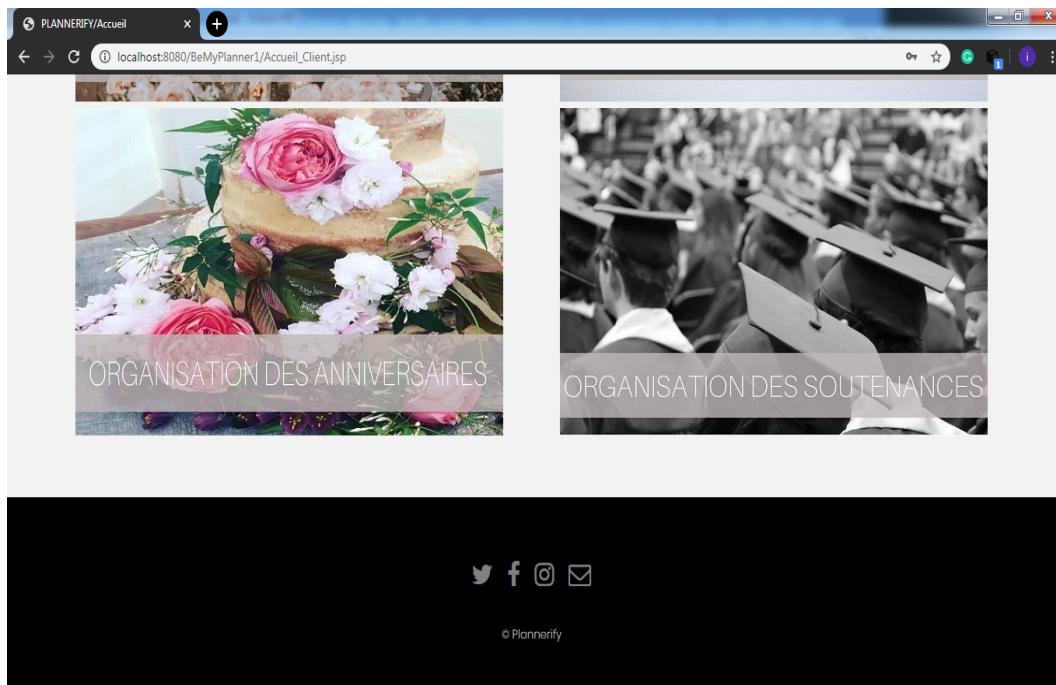


FIGURE 3.9 – Accueil.2

3. Après choisir le type d'organisation le client peut ensuite choisir le type d'organisateur.

4. PlannerIfy liste tous les menus correspondant aux type d'organisation et aux organisateurs choisis dans un ordre de prix décroissant.

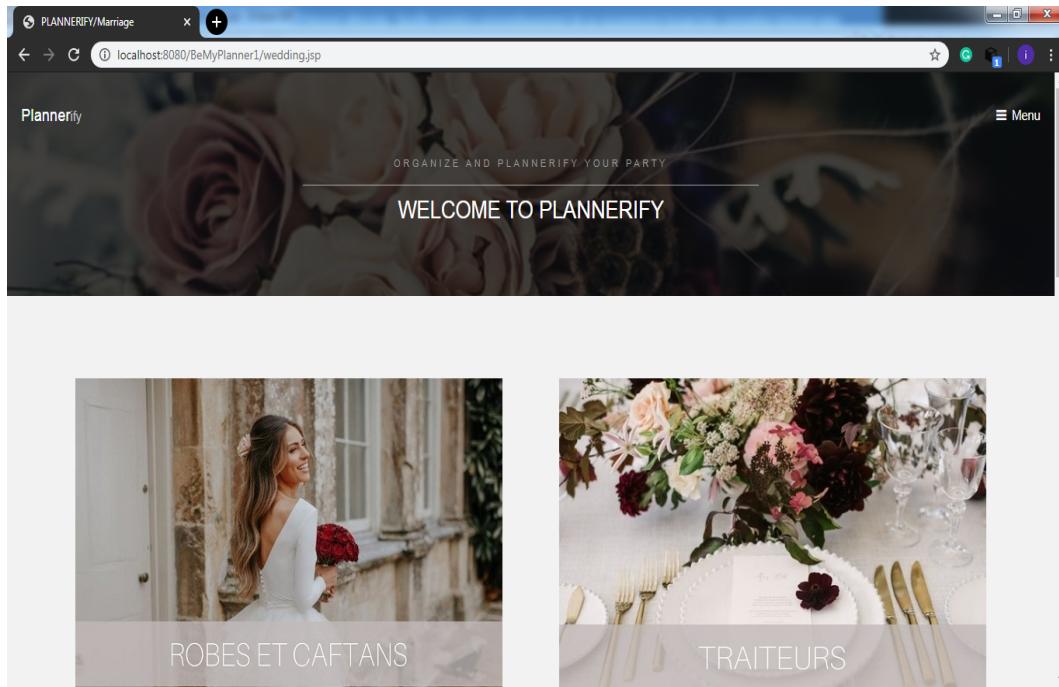


FIGURE 3.10 – type organiasiateur

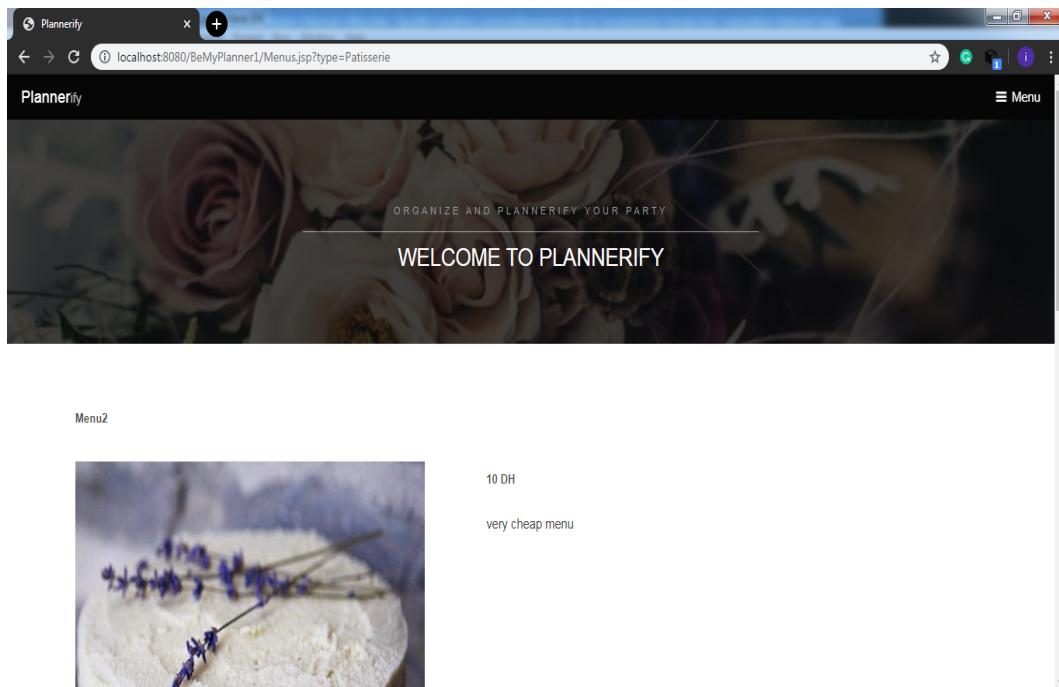
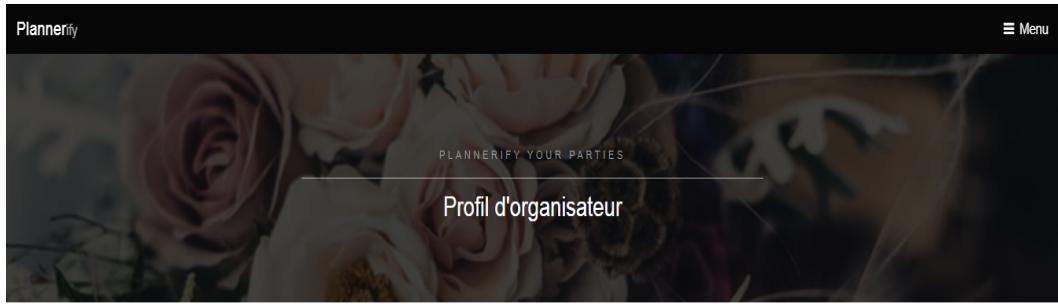


FIGURE 3.11 – liste organiasiateur

5. Pour avoir plus de détails à propos de l'organisateur ,le client a la possibilité de visiter son profil.

6. le client -en cliquant sur réserver- est dirigé vers la page de demande de résrvation



Sweet by Nature

Description

We create dream cakes that your guests will rave about.



★★★★★

FIGURE 3.12 – profile organisateur

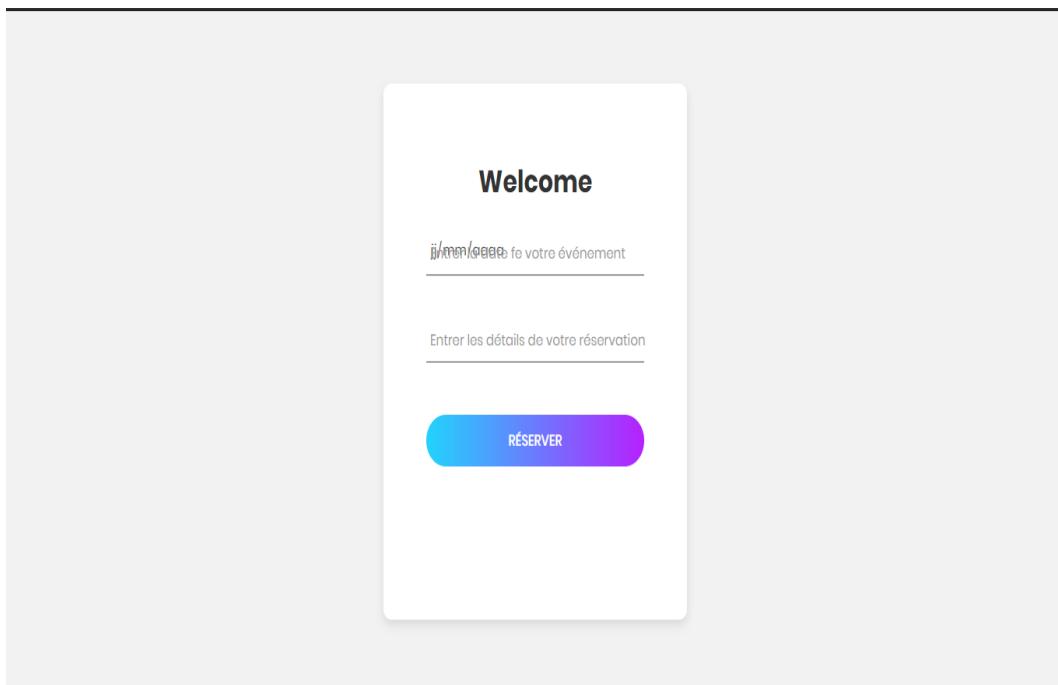


FIGURE 3.13 – demande de réservation

7.il peut aussi suivre l'état de son réservation.

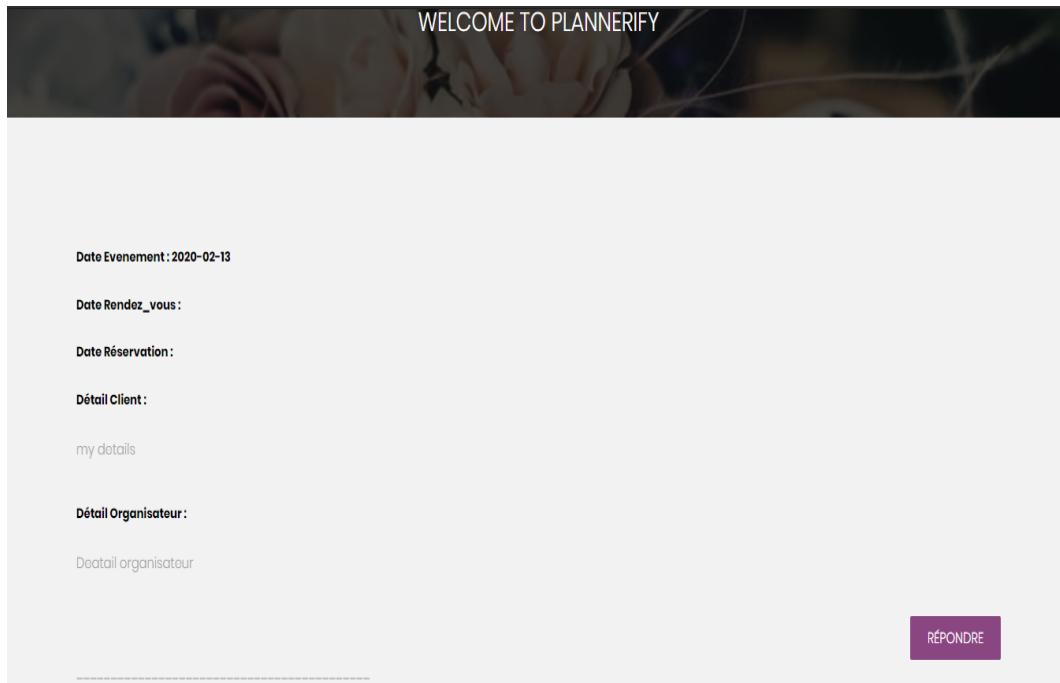


FIGURE 3.14 – état de réservation

CONCLUSION

Ce projet consistait à développer une application web de gestion des organisations de fêtes. Tout au long de ce rapport, nous avons présenté la démarche que nous avons suivie pour mettre au point cette application. Afin de réaliser notre application et atteindre les objectifs que nous avons visés, le choix s'est porté sur la méthode UML pour modéliser l'analyse et la conception de notre application, et tenter de répondre aux besoins exprimés précédemment. Ce qui importait aussi est l'étape de la réalisation du projet, qui s'est faite grâce à la plateforme Java EE en utilisant la démarche MVC.

Le projet a été une expérience très enrichissante et bénéfique pour nous. Il nous a permis d'approfondir nos connaissances, d'augmenter nos compétences et de découvrir des langages divers qui s'avèrent très utiles pour notre cursus en tant qu'ingénieurs en génie logiciel. Ce projet nous a permis aussi de faire face à plusieurs contraintes et de développer les techniques de recherche d'information. De plus, nous avons pu acquérir l'esprit du travail de groupe et ainsi apprécier l'aspect humain, écouter les autres, défendre nos idées, se partager le travail et prendre des initiatives.

RÉFÉRENCES

Documentation de phpMyAdmin :<https://fr.wikipedia.org/wiki/PhpMyAdmin>.

Documentation de CSS :https://en.wikipedia.org/wiki/Cascading_Style_Sheets
<https://docs.oracle.com/javaee/7/tutorial/>

Documentation de HTML :<https://en.wikipedia.org/wiki/HTML>

Cours magistral et tps de M. ElHamlaoui