

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abdelhamid Mehri Constantine 2



Faculté des Nouvelles Technologies de l'Information et de la Communication
Département de l'Informatique Fondamentale et ses Applications

Projet de fin d'études pour l'obtention du diplôme de
Licence en Informatique

Option :
Science d'informatique

Thème

Gestion de réservation au restaurant

Dirigé par :

Dr. Zakaria Benmounah

Réalisé par :

Mosbah Mohammed Haida

Ghebouli Youcef

Remercîments

Nos premiers remerciements s'adressent à Dieu le tout puissant qui par sa bonté et sa miséricorde nous a permis d'avoir le courage, la foi et la volonté de mener à bien ce travail.

Ce travail a pu voir le jour avec énormément d'aide et encouragement des personnes autour de nous. Ce court remerciement ne sera pas suffisant pour récompenser leurs efforts mais tout de même...

Nous tenons aussi à remercier Monsieur Zakaria BENMOUNAH notre encadreur qui ne nous a lésé d'aucune information, qui a été présent à tout moment de la réalisation de ce projet malgré la situation de notre pays et le court temps d'étude, et n'oublions pas le corps professoral et administratif de la Faculté NTIC pour la richesse et la qualité de leur enseignement.

Finalement, nous souhaitons adresser nos remerciements à nos familles qui a toujours soutenu et encouragent pendant la réalisation de ce mémoire.

Merci...

Résumé

La restauration, ce secteur porteur est redéfini en permanence et l'on observe depuis plusieurs années sa digitalisation, et ce à travers différents outils.

Actuellement, la majorité des restaurateurs ont informatisé une partie de leur travail, la prise de commande, l'encaissement ou encore leur carnet de réservation par exemple. On peut dire que les nouvelles technologies ont pris place en restauration commerciale, cela se constate côté consommateur avec la digitalisation des supports commerciaux ou des tables interactives notamment.

Le travail présent consiste à la mise en place d'une application web de réservation en ligne pour un restaurant. Le but étant de permettre au client de parcourir le menu, faire ses choix, réserver une table pour la date et l'heure de son visite. Ce ne que doit pas seulement faciliter la tâche de réservation mais ainsi la planification des travaux à la cuisine, et en conséquence, augmenté la réactivité aux demandes, et réduire le temps nécessaire. Pour arriver à notre objectif cité dans le cahier de charge, Nous avons utilisé, d'une part, le langage Java Script à la réalisation de la page. D'autre part, l'outil MySQL a été utilisé à la création de la base des données. Les Framework lumen et Angular ont été utilisé pour création le back-end et front-end de site. Dans ce travail, nous avons ainsi regroupé les travaux entre les développeurs en utilisant la méthode Agile Scrum.

Sommaire

Remercîments	2
Résumé	3
Liste des figures	7
Liste des tableaux	8
Chapitre 1.....	9
1.Motivation	10
2.Présentation du sujet	10
2.1.Problématique :	10
2.2.Solution proposée :	11
3.Aperçu du document.....	12
Chapitre 2.....	13
1.Introduction.....	14
2.UML.....	14
3.L'approche Pascale Roques	16
4.Cycle de vie linière	17
5.Méthodes agiles	20
6.Choix et motivation.....	22
7.Conclusion	22
Chapitre 3.....	23
1.Introduction.....	24
2.L'architecture web REST (API)	24
3.MySQL Workbench.....	26
4.Laravel.....	27
5.Lumen	28
6.Comparaison entre Laravel et Lumen :	28
7.Angular 7	29
8.Laragon.....	30
9.Postman	31
10.Conclusion	31

Chapitre 4.....	32
1.Introduction.....	33
2.Sprint 1 (Réservation d'une table).....	33
2.1.Introduction.....	33
2.2.Backlog du sprint 1 :	33
2.3.Analyse et réalisation du sprint 1 :	34
2.4.Schéma de la base de données	34
2.5.Les modèle PHP	35
2.6.Les contrôleurs PHP	36
2.7.Les routes.....	37
2.8.Fiche descriptive	37
2.9.Test et validation.....	38
2.10.Les diagrammes	42
3.Sprint 2 (développer la première partie (front-end))	44
3.1.Introduction.....	44
3.2.Backlog du sprint 2	44
3.3.Analyse et réalisation du sprint 2	45
3.4.Lancer une premier version front-end	47
3.5.La connexion entre le front-end et le back-end	47
3.6.La fonctionnalité authentification	47
3.7.Fiche descriptive	48
3.8.Test et validation.....	48
4.Sprint 3 (l'interface pour l'authentification)	50
4.1.Introduction.....	50
4.2.Backlog du sprint 3	50
4.3.Analyse et réalisation du sprint 3	50
4.4.Gestion des comptes	53
4.5.Gestion des avis.....	53
4.6.Fiche descriptive	54
4.7.Test et validation.....	54
5.Sprint 4 (les horaires du travail)	57
5.1.Introduction.....	57
5.2.Backlog du sprint 4	57

5.3.Analyse et réalisation du sprint 4	57
5.4.Spécification des horaires de travail.....	59
5.5.Ajout des exceptions.....	59
5.6.Fiche descriptive	59
5.7.Test et validation.....	60
6.Sprint 5 (les limites)	63
6.1.Introduction.....	63
6.2.Backlog du sprint 5	63
6.3.Analyse et réalisation du sprint 5	63
6.4.La limite entre l'heure de réservation et l'heure d'arrive.....	64
6.5.Limite de réservation d'une table.....	64
6.6.Test et validation.....	65
7.Conclusion	67
Conclusion générale	68
Référence bibliographique	69

Liste des figures

Figure 1: Historique d'UML	14
Figure 2: La hiérarchie des diagrammes UML 2.0.....	15
Figure 3: le modèle en cascade	18
Figure 4: le modèle en V.....	19
Figure 5 : Explication des rôles.....	21
Figure 6: architecture REST	26
Figure 7: le modèle MVC	27
Figure 8: la base de données de la première version	34
Figure 9: exemple d'un modèle PHP	35
Figure 10: exemple d'un contrôleur PHP.....	36
Figure 11: exemple d'un route	37
Figure 12 : test d'ajout d'un table	38
Figure 13: test de verification la disponibilité des tables.....	39
Figure 14: test d'annuler une reservation.....	40
Figure 15 : test de mis à jour d'une table.....	41
Figure 16: diagramme de cas d'utilisation	42
Figure 17: diagramme de séquence	43
Figure 18 : l'index de site	45
Figure 19 : le formulaire de la reservation	45
Figure 20 : liste des tables disponibles	46
Figure 21 : le menu disponible.....	46
Figure 22: Le fichier CorsMiddleware	47
Figure 23 : le test d'authentification	48
Figure 24 : le front d'authentification.....	50
Figure 25: user dashboard.....	51
Figure 26 : admin dashboard	51
Figure 27 : create table "admin"	52
Figure 28 : annulation d'une reservation	52
Figure 29: l'interface de "send message" dans le user.....	53
Figure 30 : l'interface des messages dans l'admin	53
Figure 31: test de réservation d'une table	54
Figure 32: sléction de menu	55
Figure 33: admin privileges	57
Figure 34: user privileges.....	58
Figure 35 : holiday test "front".....	58
Figure 36 : les horaires du travail.....	60
Figure 37 : test de congé	61
Figure 38: la limite de 2 heures	63
Figure 39 : la limite de 3 heures.....	64
Figure 40 : test d'annule la reservation dans le meme jour	64
Figure 41: limite de 2 heues "back"	65
Figure 42 : annule la reservation dans le meme jour "back"	66

Liste des tableaux

Tableau 1: Lumen VS Larevel	28
Tableau 2: Backlog du sprint 1	33
Tableau 3 : fiche descriptive de réserver table	37
Tableau 4: Backlog du sprint 2	44
Tableau 5 : fiche descriptive d'authentification	48
Tableau 6 : Tableau des tests du sprint 2.....	49
Tableau 7: Backlog du sprint 3	50
Tableau 8: fiche descriptive de gestion des comptes	54
Tableau 9 : Tableau des tests du sprint 3.....	56
Tableau 10: Backlog du sprint 4	57
Tableau 11 : fiche descriptive de la gestion des horaires du travail.....	59
Tableau 12 : Tableau des tests du sprint 4.....	62
Tableau 13: Backlog du sprint 5	63
Tableau 14 : Tableau des tests du sprint 5.....	66

Chapitre 1

Introduction

- Motivation
- Présentation du sujet
- Aperçu du document

1.Motivation

Toute organisation est tenue de suivre le développement rapide du monde de la technologie, et en profiter le plus possible afin d'assurer sa continuité et accroître sa compétitivité dans une époque où la concurrence ne cesse de s'accroître.

Le choix de notre travail est posé sur un sujet de projet de fin d'étude à la faculté de NTIC « La conception et la réalisation d'un site web pour la réservation et la livraison d'un restaurant en ligne » qui permettra de faire des actions réciproques entre le client et le site web ou soit entre le restaurateur et son ordinateur. Et ce dernier pourra être avantageux aux clients du restaurant et aux restaurateurs.

2.Présentation du sujet

Ce projet rentre dans le cadre du projet de fin d'études qui vient de conclure notre formation d'ingénieur à l'université NTIC. L'application demandée permet de réserver et commander en ligne dans un restaurant qui s'intitule **Ennassim** dont l'objectif est d'accélérer ses tâches.

La mise en place de Ennassim nécessite le développement de trois axes à savoir :

- **Axe Gestion du projet** : utilisation de la méthode agile <<Scrum>>.
- **Axe modélisation conceptuelle** : utilisation du langage de modélisation UML (Unified Modeling Language) pour développer les axes fonctionnels, statiques et dynamiques.
- **Axe de développement** : utilisation du langage Java et l'utilisation de la plateforme de développement Lumen-Laravel et AngularJS.

2.1.Problématique :

A l'intérieur de ce thème de la réservation en ligne pour un restaurant, nous avons approfondi nos recherches sur l'impact des nouvelles technologies. Les difficultés que connaissent la plupart des restaurants tel que la gestion de commandes de ses clients et la gestion de menu, ne laisse pas la tâche facile aux clients de retrouver un plat. Vu que ce menu se change souvent, cette forme de travail cause souvent des préjudices, pour cela, le choix d'un système informatique multiplateforme avec ses influences devant permettre de fluidifier le processus de traitement des commandes et d'éviter ainsi la falsification constatée en forte période reste indispensable.

De tout ce qui précède, nous nous trouvons face à des problèmes qui vont nous préoccuper dans notre recherche :

- Comment stocker et traiter les différentes informations d'un restaurant moderne en rapport avec les achats des clients ?
- Comment permettre une communication entre les clients et les servants dans un restaurant ?
- Comment faciliter une réservation de choix place de clients tout en étant loin du restaurant ?
- Quel sont les outils utilisés pour réaliser le site web ?
- Restaurant Ennassim pourrait-il mettre à la portée du public toutes ces informations sans contrainte de la distance et du temps ?

2.2.Solution proposée :

Le web est venu pour répondre aux interrogations humaines, lesquelles : Comment partager les informations entre des personnes sans tenir compte de la distance, donc la conception d'un site web pour le restaurant serait une solution pour une réservation à distance, et, toutes les informations seront à jours si seulement notre site est hébergé, qui a permis la diminution de perturbation du service, et pour organiser, amener les clients à tout moment.

Pour ce faire, nous devons utiliser une base de données pour sauvegarder les informations nécessaires des Framework permettant de structurer, d'organiser et de sécuriser cette application.

L'objectif principal est la gestion des réservations.

L'application devra notamment :

- Permettre de réserver une table dans restaurant.
- Permettre de faire la gestion des tables de restaurant.
- Permettre de gérer les horaires de travail du restaurant.
- Permet à propriétaire de restaurant de gérer les réservations.
- Permettre de faire une réservation quand on est distant.

3.Aperçu du document

Chapitre 1 : introduction

Chapitre 2 : Processus de développement logiciel

Chapitre 3 : Architecture et environnement de développement

Chapitre 4 : Implémentation

Chapitre 2

Processus de développement logiciel

- Introduction
- UML
- Pascal Roques
- Cycles de vie linéaire
- Méthodes agiles (Extreme Programming)
- Choix et motivation
- Conclusion

1.Introduction

Dans cette partie, nous montrerons le parcours que nous avons entrepris pour pouvoir créer notre projet tutoré ainsi que l'intérêt que nous y avons porté. En expliquant le processus de développement et on va expliquer les méthodes disponibles pour le déroulement de ce projet et pourquoi on a utilisé cette méthode.

2.UML

2.1.Définition

Unified Modeling Language (UML) est un langage unifié de modélisation objets. Ce n'est pas une méthode, il ne donne pas de solution pour la mise en œuvre d'un projet. C'est avant tout un formalisme graphique issu de notations employées dans différentes méthodes objets.

UML unifie également les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage [1]

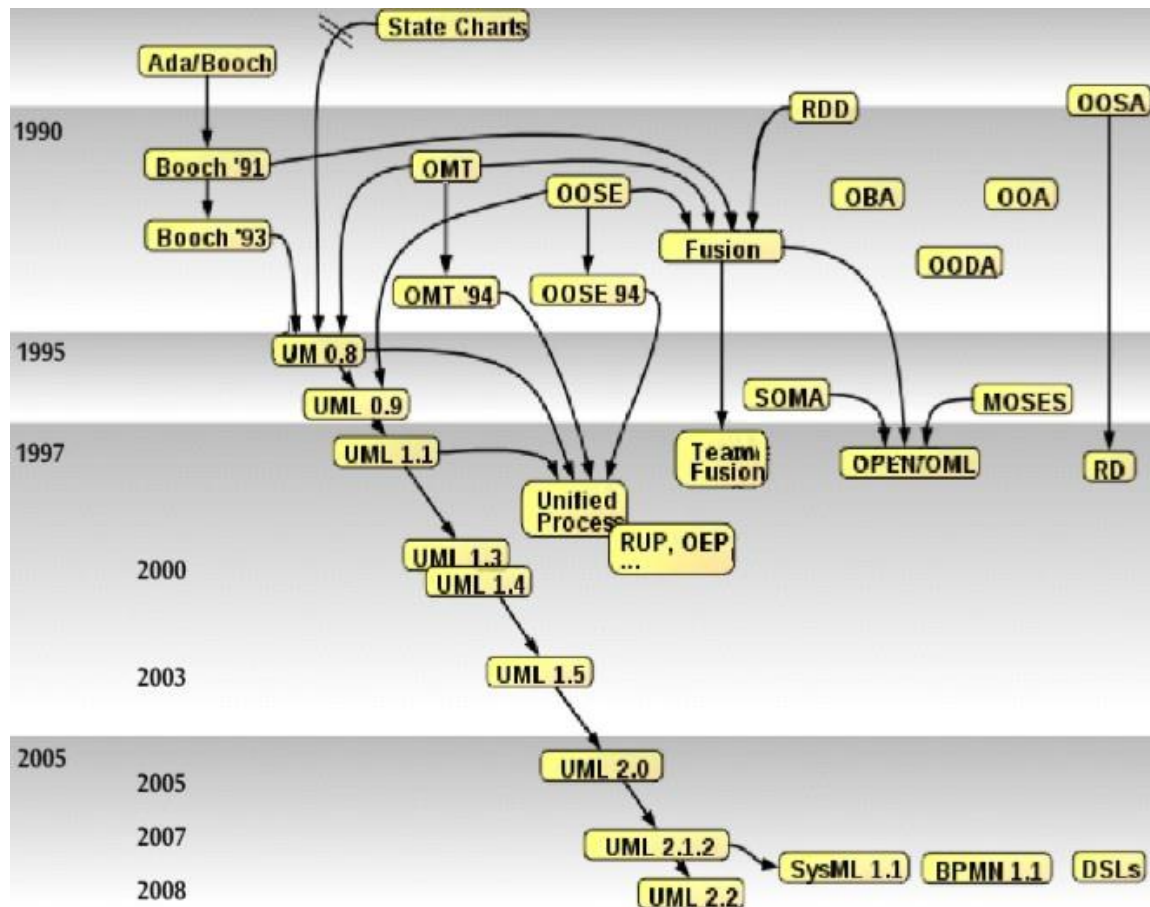


Figure 1: Historique d'UML

2.2. Le formulaire d'UML

UML se décompose en plusieurs sous-ensembles :

2.2.1. Les vues :

Les vues sont les observables du système. Elles décrivent le système d'un point de vue donné, qui peut être organisationnel, dynamique, temporel, architectural, géographique, logique, etc. En combinant toutes ces vues, il est possible de définir (ou retrouver) le système complet.

- Les différentes vues sont : (Vue logique, Vue d'implémentation, Vue des processus, Vue de déploiement)

2.2.2. Les diagrammes :

Les diagrammes sont des éléments graphiques. Ceux-ci décrivent le contenu des vues, qui sont des notions abstraites. Les diagrammes peuvent faire partie de plusieurs vues. Les 14 diagrammes UML sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie.

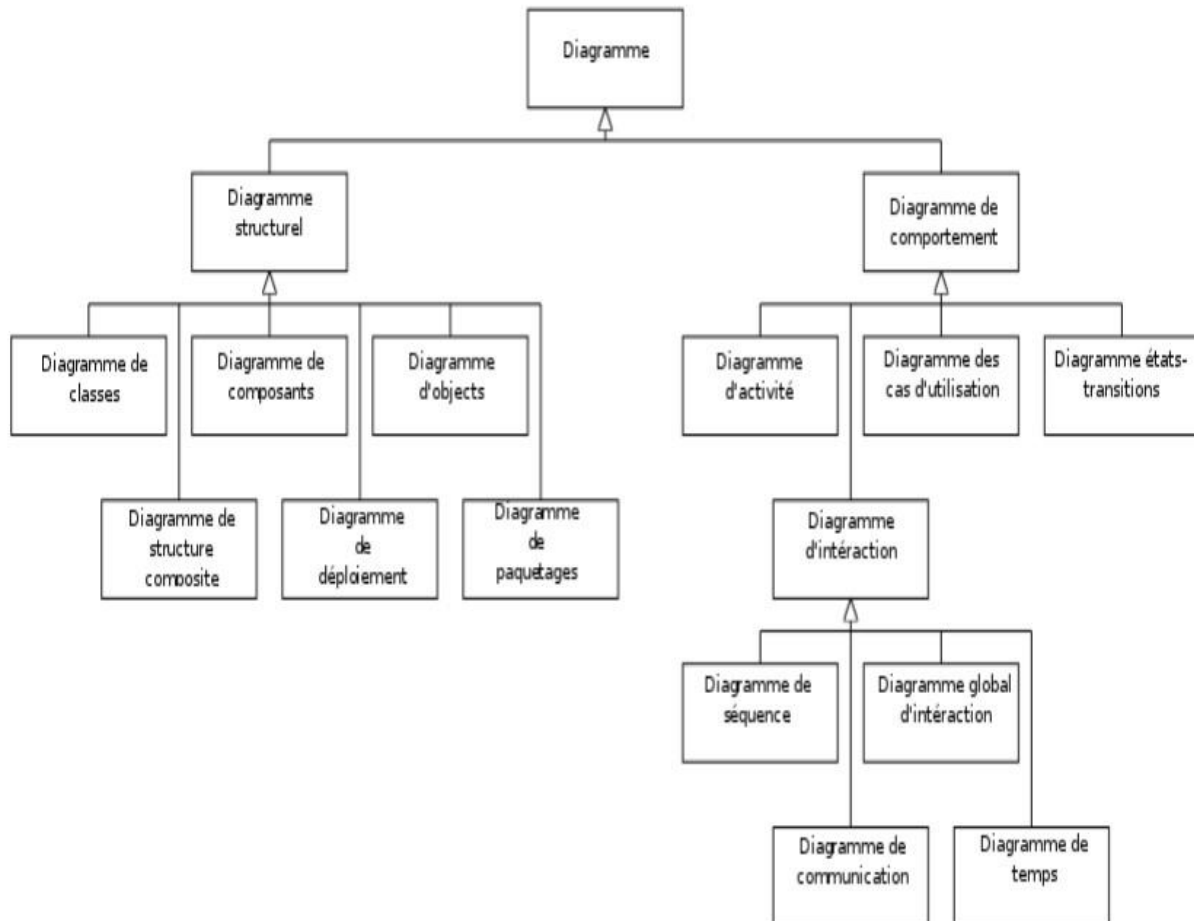


Figure 2: La hiérarchie des diagrammes UML 2.0

2.2.3.Les modèles d'élément

Les modèles d'élément sont les briques des diagrammes UML, ces modèles sont utilisés dans plusieurs types de diagrammes. Exemple d'élément : cas d'utilisation, classe, association, etc.

2.3.Les avantages et les inconvénient d'UML**2.3.1.Les points forts d'UML**

UML est un langage formel et normalisé : Il permet le gain de précision, encourage l'utilisation d'outils et constitue à cet effet un gage de stabilité.

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 en font un langage universel.

2.3.2.Les points faibles d'UML

La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation. Même si l'Espéranto est une utopie, la nécessité de s'accorder sur des modes d'expression communs est vitale en informatique. UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.

Le processus (non couvert par UML) est une autre clé de la réussite d'un projet. Or, l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue. Les auteurs d'UML sont tout à fait conscients de l'importance du processus, mais l'acceptabilité industrielle de la modélisation objet passe d'abord par la disponibilité d'un langage d'analyse.

3.L'approche Pascale Roques**3.1.Introduction**

La modélisation UML avec ses différentes étapes doivent être ordonnées et organisées, pour cela on a besoin d'une approche de développement.

3.2.Définition

Pascal Roque est une méthode simple et générique de développement de logiciel, elle se positionne entre les deux méthodes UP (Unified Process), et XP (eXtreme Programming) qui est une approche minimaliste centrée sur le code. Cette approche est le résultat du livre présenté par Roques (2002) << UML –Modéliser un site e-commerce >>. [1]

3.3.Caractéristiques

- Relativement léger et restreinte, sans négliger les activités de modélisation en analyse et conception.
- Implémente les bonnes qualités des deux approches UP et XP.
- Conduite par les cas d'utilisation, comme UP, mais beaucoup plus simple.
- Fondée sur l'utilisation d'un sous-ensemble nécessaire et suffisant du langage UML.

L'approche Pascale Roques se constitue de quatre étapes principales aussi connues sous le nom de Cycle de vie linéaire, et c'est ce qu'on va aborder dans le point suivant.

4.Cycle de vie linière

4.1.Introduction

Le cycle de vie du logiciel ou (software life cycle) sert à modéliser le déroulement dans le temps d'un processus complexe. C'est l'ensemble de phases qui se déroulent depuis le moment où le client commande l'application chez un développeur jusqu'au moment où cette dernière le lui est livrée et jusqu'à sa disparition. [2]

IL existe plusieurs modèles de cycle de vie d'un logiciel :

Modèles linéaires : (Cascade, Modèle en V ...ect)

Modèles non-linéaires (itératifs) : (Prototypage, Modèles incrémentaux, Modèles en spirale, ...ect)

4.2.Modèle en cascade

Dans ce modèle le principe est très simple : chaque phase se termine à une date précise par la production de certains documents ou logiciels. Les résultats sont définis sur la base des interactions entre étapes, ils sont soumis à une revue approfondie et on ne passe à la phase suivante que s'ils sont jugés satisfaisants [3].



Figure 3: le modèle en cascade

Caractéristique

- Modèle de famille Linéaire
- Retour limité à une phase en amont
- Validation des phases par des revues
- Enchaînement depuis le cahier des charges jusqu'à la réalisation(tests)

Avantages de Modèle en cascade

- Facile à mettre en place
- Facile à utiliser et à comprendre

Inconvénients de Modèle en cascade

- Validation très tardive
- Absence de flexibilité
- Une phase ne peut démarrer que si l'étape précédent est finie
- Erreur d'analyse ou de conception très coûteuse
- Les risques se déclarent vers la fin

4.3.Modèle en V

Est une méthode d'organisation très connue dont l'origine remonte à l'industrie et qui a été adaptée à l'informatique dans les années 80. C'est l'une des premières méthodes qu'on apprend à l'école après le cycle en cascade, et elle reste toujours d'actualité [4].

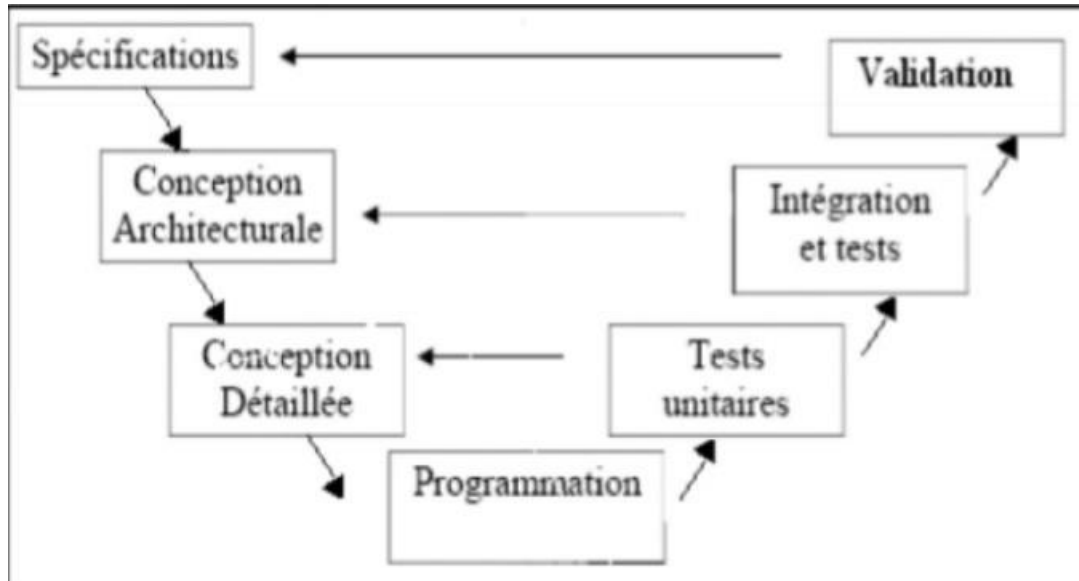


Figure 4: le modèle en V

Caractéristique

- Le modèle en V reste actuellement le cycle de vie le plus connu et certainement le plus utilisé
- Validation finale montante et confirmation de la validation descendante

Avantages de Modèle en V

- Validations intermédiaires
- Modèle encore assez populaire en industrie
- Limitations des risques en cascade par validation de chaque étape
- Modèle éprouvé très utilisé pour de grands projets

Inconvénients de Modèle en V

- Plus complexe que le modèle en cascade
- On ne voit pas toujours de retour sur les phases précédentes
- Plus difficile à mettre en œuvre
- Difficile de séparer les phases de conception et de réalisation

5.Méthodes agiles

5.1.Définition

La méthodologie Agile est un ensemble de pratiques et de principes utilisés pour le développement logiciel, dont la principale caractéristique est la grande adaptabilité aux changements autour du projet. Ces pratiques se basent sur le manifeste Agile, qui énumère les différents principes pour pouvoir considérer un projet comme étant « Agile » [5].

5.2.Les différents types de méthodologies Agile

Il existe différentes manières d'utiliser la méthodologie Agile au sein d'un projet. Elles permettent surtout de s'adapter aux méthodes de travail déjà présentes au sein d'une équipe puisqu'un changement aussi important ne peut se faire du jour au lendemain. On va donc parler de deux de ces méthodologies Agile : « Scrum » et « Extreme Programming », et on va insister sur « Extreme programming ».

5.2.1.Scrum

Scrum est donc une méthodologie Agile, où il est considéré que rien ne peut être déterminé et planifié à l'avance sur de longues périodes de travail, et ainsi le travail en cycles courts est prôné. Une analyse est faite également à chaque fin de sprint pour permettre une amélioration continue. Scrum se base sur trois piliers : (La transparence, L'inspection, L'adaptation), Pour permettre de suivre les principes de ces 3 piliers, il existe différentes méthodes et rôles à appliquer rigoureusement tout au long du projet pour en assurer la réussite.

Les rôles

- **Le product owner** : Il établit la relation, le pont en quelque sorte, entre le client et l'équipe de développement, en tant que représentant des financeurs d'un projet.
- **L'équipe de développement** : Ce sont les personnes qui développent le produit qui sera livré au client.
- **Le scrum master** : C'est un membre de l'équipe de développement, qui est présent pour intégrer les bonnes pratiques Scrum au sein d'un projet et s'assurer que les membres de l'équipe les utilisent à bon escient.

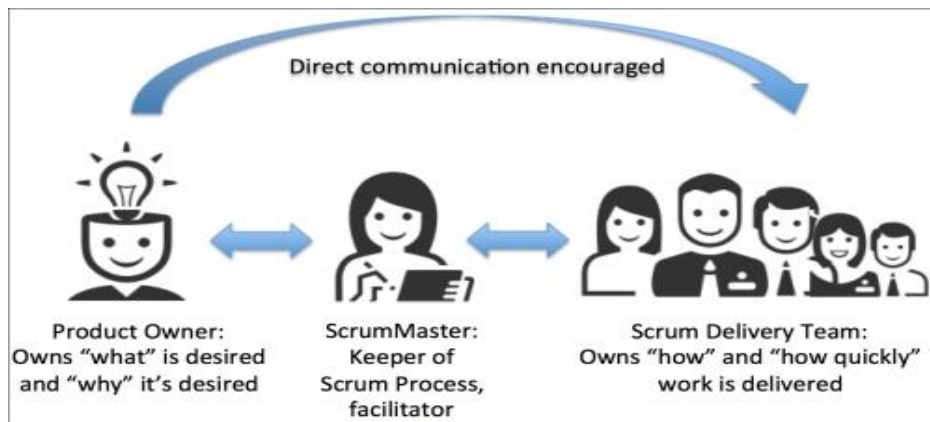


Figure 5 : Explication des rôles

Le workflow (sprints)

- **Le planning des sprints :** C'est la planification du sprint. Il permet de définir l'ampleur du travail à effectuer pendant le déroulement d'un sprint.
- **Les daily scrum :** Ce sont des réunions quotidiennes qui se déroulent tous les matins, une fois que toute l'équipe est réunie.
- **La sprint review :** Permet à l'équipe de vérifier le travail effectué durant le sprint, mais également les tâches qui n'ont pu être finies.
- **La rétrospective :** Est essentiellement utilisée au sein de l'équipe, cela démarre en se posant deux questions : « Qu'est-ce qui s'est bien passé durant ce sprint ? » et « Qu'est-ce qui peut être amélioré pour le prochain sprint ? ».

5.2.2. Extreme programming

Elle se base sur des cycles courts et réguliers de livraison au client, pour s'adapter aux besoins et aux demandes. Elle se base sur des principes qui sont plutôt simples et poussés à l'extrême [6]. Ces valeurs sont :

La communication : Elle doit être établie dès le début et pratiquée en permanence, quel que soit l'étape du projet (planification, programmation, tests...). Elle permet également d'échanger avec les membres du projet en cas de changement.

La simplicité : Pour éviter de complexifier les tâches et réaliser les objectifs au plus vite, la simplicité sera appliquée pour la rapidité de développement et pour permettre à l'application d'avoir de nombreuses perspectives d'évolutions par la suite.

Le feedback : Il sera mis en place via les tests (unitaires et fonctionnels) et les livraisons fréquentes, pour vérifier le bon fonctionnement du code et de l'application.

Le courage : Lors d'un projet utilisant la méthode « Extreme programming », des changements majeurs pourront arriver (changer l'infrastructure complète, jeter une grande partie du code, etc...). Cela demande beaucoup de courage, mais ce sera facilité par la communication, la simplicité et le feedback.

Le respect : Cela passe par le respect des autres, de leur travail, en effectuant des changements qui n'impactent pas leur travail sans leur consentement. Il concerne également le respect de soi, en réalisant la meilleure application possible en termes de qualité et de conception.

L'application de l'Extreme programming

Pour appliquer l'Extreme programming, il faut également utiliser 13 pratiques spécifiques. Parmi celles-ci, certaines se démarquent comme le planning poker qui permet d'évaluer le temps nécessaire à la réalisation des fonctionnalités demandées par un client, via différents scénarios qui seront appliqués par la suite. Il y a également l'intégration continue qui consiste à intégrer dans le produit complet toute modification qui apparaisse suite à la réalisation du projet [6]. Une des autres particularités de cette méthode la programmation en binôme : il s'agit de développer des fonctionnalités de l'application par deux, avec un qui utilise le clavier, pour écrire le code d'application, et le second qui sera là pour vérifier le travail du premier, suggérer de nouvelles fonctionnalités ou indiquer des erreurs s'il y en a dans le code.

L'application de cette méthode ne peut se faire dans tous les cadres par contre. Il concerne plutôt les petites équipes (la communication sera difficile si le nombre de 15 personnes est dépassé). Il faut également que la totalité de l'équipe adopte les valeurs et les pratiques de la méthode pour être certain de leur efficacité. Cela s'applique également au client. L'équipe devra idéalement être placée sur le même site au sein d'un open-space pour pouvoir appliquer la méthode de la programmation en binôme. D'autres contextes peuvent être défavorables à l'application de la méthode, il faut ainsi les éviter pour aboutir à un résultat efficace qui satisfait les demandes du client.

6.Choix et motivation

Parmi les différentes approches que nous avons présentées dans ce chapitre nous optons pour l'utilisation de la méthode Scrum car elle est une méthode complète de management projet qui ne présente que des qualités : elle se recentre sur la qualité, les objectifs, l'efficacité, la réduction de bugs tout en permettant une cohésion et une communication excellente entre les différents tiers du projet à travers le daily Scrum Meeting. Les tâches, le processus de "reviewing", la découpe du projet est parfaitement définie ce qui améliore l'avancement d'une manière totalement transparente.

7.Conclusion

Dans ce chapitre, nous avons présenté les méthodes de conception d'un projet et expliquer pourquoi on a choisi notre choix parmi les autres méthodes, Dans le chapitre qui suit nous allons entamer la partie logicielle et environnements de développement nécessaires utilisés dans notre projet.

Chapitre 3

Architecture et environnement de développement

- Introduction
- L'architecture web REST (API)
- MySQL Workbench
- Laravel
- Lumen
- Comparaison entre Laravel et Lumen
- Angular 7
- Laragon
- Postman
- Conclusion

1.Introduction

On va présenter dans ce chapitre l'architecture web et l'environnement de développement utilisés dans ce projet.

2.L'architecture web REST (API)

2.1.Introduction

Les Web Services sont des services informatiques de la famille des technologies web permettant la communication entre des applications hétérogènes dans des environnements distribués. Basés sur XML (description et échange) et utilisant en général les protocoles du web comme canal de communication. Il existe deux principaux types :

- SOAP
- REST

2.2. API

Une API permet de rendre disponibles les données ou les fonctionnalités d'une application existante afin que d'autres applications les utilisent. Voici qui devrait rendre la notion d'**interface de programmation applicative** plus claire [7].

Utiliser une API permet donc d'utiliser un programme existant plutôt que de le re-développer. C'est donc un grand gain de temps à la clé.

Catégories d'API

- API de détection : ces API vous aident à identifier les opportunités qui impliquent des clients, des employés, des partenaires et des équipements. Elles intègrent des fonctionnalités, telles que la géolocalisation mobile, la surveillance par capteurs, l'analyse prédictive et l'observation humaine.
- API d'enrichissement : ces API améliorent la compréhension de la situation grâce à des données historiques émanant de systèmes de gestion de la relation client (CRM), de fichiers de comptabilité, d'analyses démographiques, de dossiers médicaux, etc.
- API de perception : ces API fournissent un contexte dynamique de la situation actuelle et vous permettent de savoir ce que pensent les personnes que vous ciblez. Il peut s'agir d'API sociales (des personnes partageant des projets futurs ou des intérêts actuels) ou de solutions d'analyse de capteurs (pour l'état global du système, comme la consommation des ressources ou la congestion du trafic).
- API d'action : ces API vous permettent d'agir en quasi-temps réel. Elles peuvent inclure des notifications push, des équipements instrumentés ou des systèmes de gestion de tâches humaines.

2.3.REST

C'est un acronyme de REpresentational State Transfert défini dans la thèse de Roy Fielding en 2000. REST n'est pas un protocole ou un format, contrairement à SOAP, HTTP ou RCP, mais un style d'architecture inspiré de l'architecture du web fortement basé sur le protocole http et il n'est pas dépendant uniquement du web et peut utiliser d'autre protocoles que HTTP [8]. En bref c'est un système architecture et une approche pour construire une application donc les applications respectant l'architecture REST sont dites RESTful.

Utilisation

Utiliser dans le développement des applications orientées ressources (ROA) ou orientées données (DOA).

Caractéristiques

- Les services REST sont sans états (Stateless)
 - Chaque requête envoyée au serveur doit contenir toutes les informations relatives à son état et est traitée indépendamment de toutes autres requêtes
 - Minimisation des ressources systèmes (pas de gestion de session, ni d'état)
- Interface uniforme basée sur les méthodes HTTP (GET, POST, PUT, DELETE)
- Les architectures RESTful sont construites à partir de ressources uniquement identifiées par des URI(s)

Méthodes

- Une ressource peut subir quatre opérations de bases CRUD correspondant aux quatre principaux types de requêtes HTTP (GET, PUT, POST, DELETE)
 - REST s'appuie sur le protocole HTTP pour effectuer ces opérations sur les objets
 - CREATE > POST
 - RETRIEVE > GET
 - UPDATE > PUT
 - DELETE > DELETE

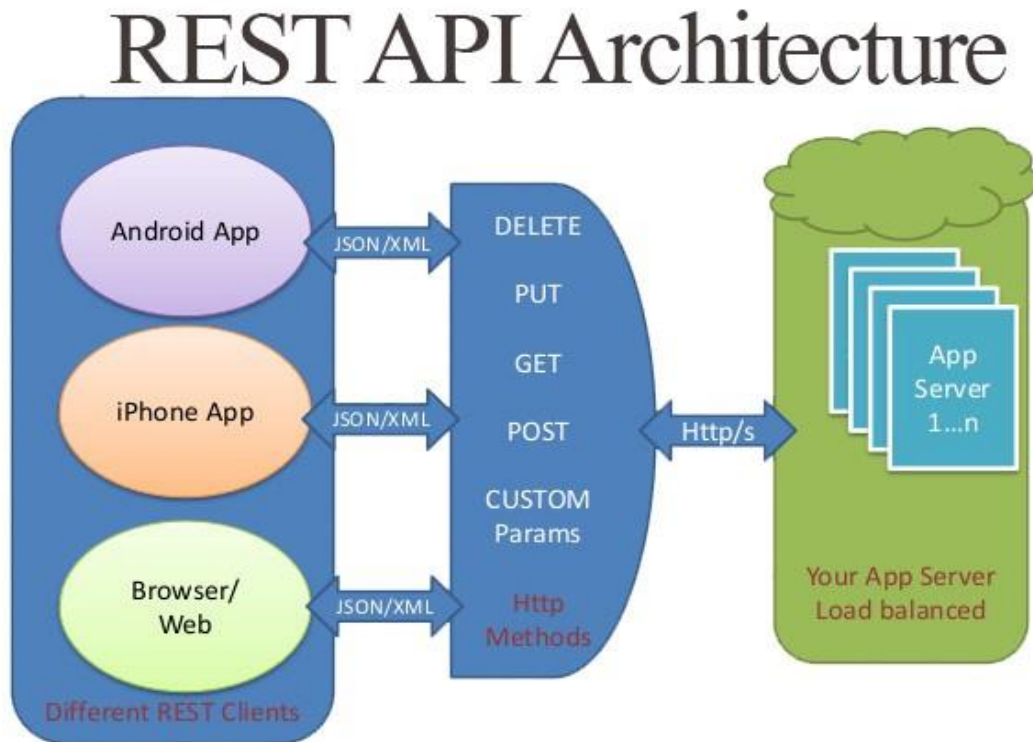


Figure 6: architecture REST

3.MySQL Workbench

3.1.Définition

MySQL Workbench (anciennement MySQL administrator) est un logiciel de gestion et d'administration de bases de données MySQL créé en 2004. Via une interface graphique intuitive, il permet, entre autres, de créer, modifier ou supprimer des tables, des comptes utilisateurs, et d'effectuer toutes les opérations inhérentes à la gestion d'une base de données. Pour ce faire, il doit être connecté à un serveur MySQL [9].

3.2.Caractéristiques

- Modélisation visuelle des données.
- Reverse Engineering (Création du diagramme à partir d'une BDD MySQL).
- Génération du code SQL de création des tables.
- Synchronisation du schéma MySQL : BDD <-> Diagramme.
- Export en image, PDF.
- Compatibilité avec DBDesigner4.
- Diagrammes et designs.

4.Laravel

4.1.Introduction

Un framework informatique est un « ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel » ; autrement dit, c'est une base cohérente avec des briques toutes prêtes à disposition. Il existe des frameworks pour tous les langages de programmation et en particulier pour PHP.

4.2.Définition

Laravel est un framework PHP qui propose des outils pour construire une application web. Le créateur de Laravel, Taylor Otwell, a simplement regroupé les meilleures bibliothèques pour chaque fonctionnalité nécessaire pour la création d'un site web. [10]Il a aussi rajouté des fonctionnalités exclusives, comme le moteur de template Blade.

Laravel est un Framework open-source de PHP, qui respecte le principe de modèle-vue-contrôleur (MVC).

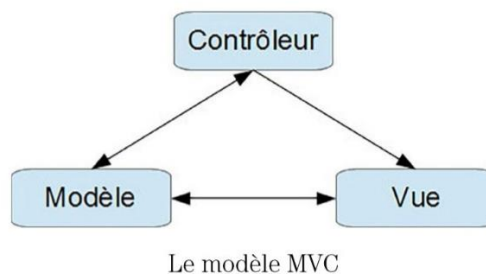


Figure 7: le modèle MVC

4.3.Caractéristiques

- Un système de routage perfectionné (RESTful et ressources)
- Un créateur de requêtes SQL et un ORM performants
- Un moteur de templates efficace
- Un système d'authentification pour les connexions
- Un système de validation
- Un système de pagination
- Un système de migration pour les bases de données
- Un système d'envoi de courriels
- Un système de cache
- Un système d'événements
- Un système d'autorisations
- Une gestion des sessions...ect

5.Lumen

Lumen est un micro-framework qui permet de lancer rapidement une petite application ou mettre en place un web service sans pour autant se surcharger avec toutes les fonctionnalités d'un Framework complet. Comme **Laravel**, ce framework a été créé par **Taylor Otwell** et selon des benchmarks sur le site de Lumen il serait l'un des plus rapides [11].

6.Comparaison entre Laravel et Lumen :

Laravel	Lumen
Il s'agit d'un framework Web full-stack basé sur MVC.	C'est un micro-framework Web qui crée des micro-services.
C'est un framework web full stack basé sur PHP.	C'est une version allégée du framework Laravel.
Il supporte l'intégration et l'utilisation avec le langage de programmation PHP.	Il prend en charge l'intégration en ajoutant des packages tiers ou des outils pour activer les nouvelles fonctionnalités.
Il est compatible avec tous les systèmes d'exploitation ou plates-formes.	Il est compatible aussi car il a été dérivé de Laravel
Il traite moins de requêtes par seconde par rapport à Lumen.	Il traite plus de requêtes par seconde.
Il a une petite communauté.	Il a une communauté plus large et de nombreux contributeurs.
Le temps de réponse est élevé pour Laravel par rapport à Lumen.	Le temps de réponse est inférieur à Laravel.

Tableau 1: Lumen VS Larevel

7. Angular 7

7.1. Definition

Angular est un framework Javascript côté client qui permet de réaliser des applications de type "Single Page Application". Il est basé sur le concept de l'architecture MVC (Model View Controller) qui permet de séparer les données, les vues et les différentes actions que l'on peut effectuer [12].

7.2. Architecture

Pour plusieurs années AngularJS était proche du MVC, mais au fil du temps et grâce à l'améliorations de l'API, il supporte MVW. Ce pattern est un dérivé du célèbre modèle de conception MVC dont Whatever désigne « whatever works for you ».

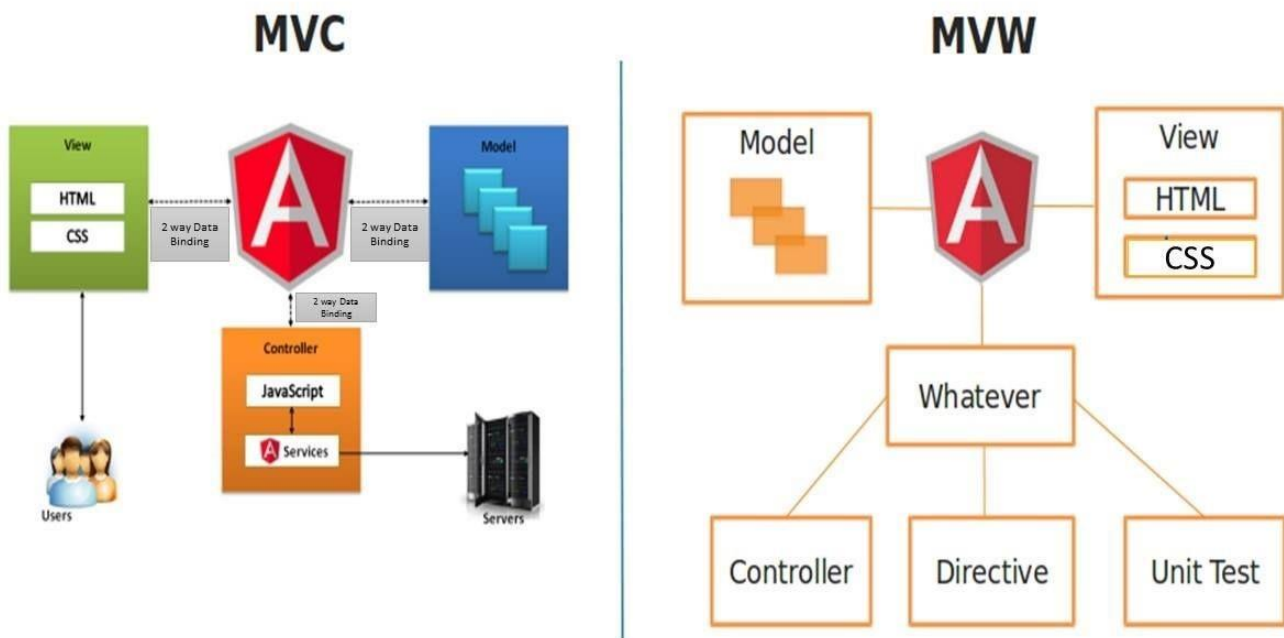


Figure 8 : Angular avec MVC et MVW

7.3 Pourquoi Angular ?

- Angular est un framework
- Point de vue de Vue.js concernant Angular
- TypeScript
- Single-Page App & Progressive Web App
- Abstraction
- Mobile
- Separation of concerns
- Ecosystème très riche et large communauté
- Testabilité comme priorité
- Architecture et maintenabilité
- Performance
- Angular Release Schedule & Long-Term Support

8. Laragon

8.1. Définition

L’Aragon est serveur local Développé par l’équipe du Framework PHP Laravel, dans le but d’en faciliter l’usage, le serveur Laragon est également utilisable comme serveur de développement local, Pour chaque dossier placé dans le répertoire www a généré automatiquement un Virtual host Apache [13].

8.2. Pourquoi Laragon ?

- Des petit URLs
 - Utilisez app.test au lieu de localhost / app.
- Portable
 - Vous pouvez déplacer le dossier Laragon (vers un autre disque, un autre ordinateur portable, la synchronisation avec le Cloud, ...) sans aucun souci.
- Isolé
 - Laragon a un environnement isolé avec votre système d’exploitation - il gardera votre système propre.
- Opération facile
 - Contrairement aux autres qui pré-config pour vous, Laragon auto-config toutes les choses compliquées. C'est pourquoi vous pouvez ajouter d'autres versions de PHP, Python, Ruby, Java, Go, Apache, Nginx, MySQL, PostgreSQL, MongoDB, ... sans effort.
- Moderne et puissant
 - Laragon vient avec l'architecte moderne qui convient pour construire des applications Web modernes. Vous pouvez travailler avec Apache et Nginx comme ils sont entièrement gérés.

9.Postman

9.1.Definition

Postman est un client REST proposé par Google. Il est disponible sous la forme d'une extension Chrome ou bien d'une application stand-alone. Postman permet de créer et envoyer des requêtes HTTP. C'est un outil très pratique pour tester rapidement une API et générer le code dans votre langage favori [14].

9.2.Caractéristiques

Les principales caractéristiques de Postman sont :

- Envoi de requêtes (avec prise en charge de différents schémas d'authentification, cookies, certificats, en-têtes, paramètres de requête, corps de requête et SOAP avec / sans WSDL) et réponses au débogage et à la sauvegarde.
- Organiser vos API en groupes appelés Collections
- Tests d'écriture Les scripts de test peuvent s'exécuter avant la réception d'une réponse et peuvent avoir des concepts de bouclage et de branchement.
- Un moyen de générer automatiquement et de personnaliser la documentation de votre API directement à partir de vos collections. Il peut être privé, partagé avec votre équipe, public et peut également être configuré sur votre domaine personnalisé.
- Génération automatique d'extraits de code dans différentes langues à partir de vos API
- Interception des requêtes
- Historique interactif de toutes vos demandes

10.Conclusion

Dans ce chapitre on a présenté l'architecture et environnement de développement de notre application web et on a expliqué pourquoi on doit utiliser ces architectures et Framework et comment on va les tester. Le chapitre suivant est consacré à la phase d'implémentation.

Chapitre 4

Implémentation

- Sprint 1
 - Introduction (1)
 - Back-log (1)
 - Analyse et réalisation (1)
 - Test et validation (1)
 - Les diagrammes (1)
- Sprint 2
 - Introduction (2)
 - Back-log (2)
 - Analyse et réalisation (2)
 - Test et validation (2)
- Sprint 3
 - Introduction (3)
 - Back-log (3)
 - Analyse et réalisation (3)
 - Test et validation (3)
- Sprint 4
 - Introduction (4)
 - Back-log (4)
 - Analyse et réalisation (4)
 - Test et validation (4)
- Sprint 5
 - Introduction (5)
 - Back-log (5)
 - Analyse et réalisation (5)
 - Test et validation (5)

1.Introduction

Dans ce chapitre nous présentons l'implémentation correspondant à chaque version de notre application web (les sprints) et on va tester ces fonctions principales.

2.Sprint 1 (Réservation d'une table)

2.1.Introduction

Ce sprint a pour but de développer le module de gestion d'une table qui permet au client à réserver une table avec sa position et le nombre des chaises selon la disponibilité et réserver le menu, modifier et ensuite annuler cette réservation en spécifiant la date et l'heure d'arrivée.

2.2.Backlog du sprint 1 :

Le Backlog du sprint contient une liste des tâches, identifiées par l'équipe Scrum qui devront être réalisées avant la fin de sprint.

ID	Histoire utilisateur	ID Tache	Tache
1	Développer une version fonctionnelle (serveur) qui contient l'essentiel du cahier de charge pour la réservation	1.1	Créer la base de données
		1.2	Etudier Laravel_lumen
		1.3	Intaller laragon, composer
		1.4	Création d'une version Fonctionnelle qui contient le cahier de charge
		1.5	Ajouter les modèle PHP
		1.6	Ajouter les contrôleurs PHP
		1.7	Ajouter les routes de l'application
		1.8	Le test de la partie serveur (envoyer les requêtes dans postman).

Tableau 2: Backlog du sprint 1

2.3. Analyse et réalisation du sprint 1 :

Après avoir étudié les différentes tâches du sprint 1, nous avons créé les fonctions pour pouvoir les tester après. Les résultats sont ci-dessous (on n'a pas une version front-end de ce sprint on a juste « test et validation »).

2.4. Schéma de la base de données

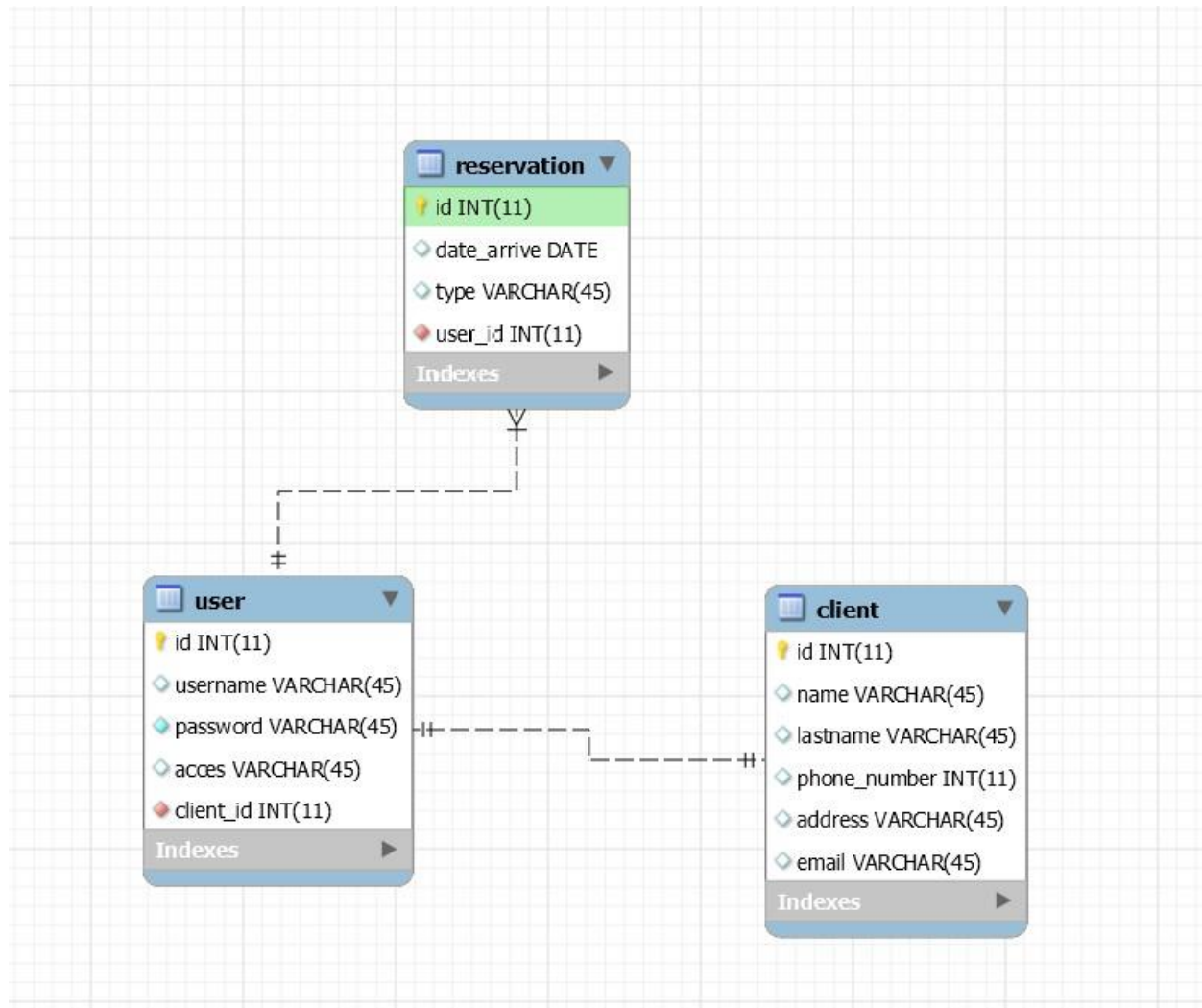


Figure 8: la base de données de la première version

2.5. Les modèle PHP

On va gérer nos données par l'intermédiaire d'un outil appelé Eloquent qui est un ORM (Object Relational Mapping) qui permet de simplifier les interactions avec la base de données en créant des extensions à la classe Model qui portent le nom de la table sur laquelle nous souhaitons interagir on a un exemple de modèle PHP :

```
Reservation_Table.php ×
1  <?php
2
3  namespace app\Models;
4  use Illuminate\Database\Eloquent\Model;
5  class Reservation_Table extends Model{
6      protected $fillable = ['reservations_id','tables_id','HeureRentrer'];
7      protected $guarded = [];
8      protected $primaryKey = 'reservations_id';
9      public static $rules = [
10
11      ];
12      public $timestamps = false;
13
14  }
```

Figure 9: exemple d'un modèle PHP

2.6.Les contrôleurs PHP

La tâche d'un contrôleur est de réceptionner une requête (qui a déjà été triée par une route) et de définir la réponse appropriée, rien de moins et rien de plus. Exemple d'un contrôleur :

```
Controller.php x
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Laravel\Lumen\Routing\Controller as BaseController;
6
7  class Controller extends BaseController
8  {
9      //
10     public function verifier(Request $request,$gool)
11     {
12         $this->validate($this->request, [
13             'emplacemnet' => 'required',
14             'capacite' => 'required'
15         ]);
16         $table = Table::where('emplacemnet', $this->request->input('emlacemnet'));
17         if($table)
18         {
19             $table = Table::where('capacite', $this->request->input('capacite'));
20             if($table)
21             {
22                 $table = Table::where('disponible', $this->request->input('staut'))->firstset();
23                 if($table)
24                 {
25                     $gool=$table;
26                     return true;
27                 }
28             }
29             else{
30                 return false;
31             }
32         }
33     }
34 }
35 }
36 }
37 }
```

Figure 10: exemple d'un contrôleur PHP

2.7.Les routes

La classe route nous permet d'enregistrer des routes qui répondent à n'importe quelle requête HTTP. Exemple de route :

```
$router->group(['middleware' => ['role:restaurant owner|super admin']], function ($router) {
    $router->group(['prefix' => '/Owner/Reservation'],function ($router)
    {
        $router->delete('/DeleteForDay/{date}','ReservationController@DeleteForDay');
        $router->get('/ShowForDay/{date}','ReservationController@ShowForDay');
        $router->get('/ShowAR','ReservationController@ShowAR');
    });
});
```

Figure 11: exemple d'un route

2.8.Fiche descriptive

Cas d'utilisation	Réserver table
Objectif	Permet le client de réserver une table.
Acteur principal	Client
Acteur secondaire	/
Précondition	Authentication
Scénario Nominal	<ol style="list-style-type: none"> 1. Le client accède au système et choisit « Réserve ». 2. Le system affiche un formulaire (full name, party size, date, time, type). 3. Le client choisi nbrpressone,date,time et click sur <<check>>. 4. Le system affiche une liste des tables qui sont libre dans la date et time entrée. 5. Le client choisi une des tables et click <<Next>>. 6. Le système affiche le menu disponible. 7. Le client choisit (drinks/desserts/main) par le clique "Checkbox" . 8. Le client click sur <<confirmé>>. 9. Le système sauvegarde la réservation et affiche un message de succès
Alternative Scenario	<ol style="list-style-type: none"> 4.a si : Le client enter une date invalide. 4. a.1 Le system affiche un message d'erreur. 4. a.2 retour à l'étape 2 de Scénario Nominal.
Exceptions	<ol style="list-style-type: none"> 1. Le client annule l'operation.

Tableau 3 : fiche descriptive de réserver table

2.9. Test et validation

Back-end

L'ajoute d'une table :

L'admin doit remplir un formulaire qui contient la capacité min, capacité max et le placement de cette dernière.

The screenshot displays a REST client interface with a POST request to `http://localhost:8000/Table/Add?Emplacement=bien a coté de fenetre&CapaciteMax=4&api_token=kl,fd&CapaciteMini=3`. The request body is a JSON object: `{ "Emplacement": "bien a coté de fenetre", "CapaciteMax": "4", "CapaciteMini": "3", "id": 13 }`. The response status is 200 OK, with a time of 1836 ms and a size of 581 B. The response body is also shown as a JSON object: `{ "Emplacement": "bien a coté de fenetre", "CapaciteMax": 4, "CapaciteMini": 3, "id": 13 }`.

Key	Value	Description
<input checked="" type="checkbox"/> Emplacement	bien a coté de fenetre	
<input checked="" type="checkbox"/> CapaciteMax	4	
<input checked="" type="checkbox"/> api_token	kl,fd	
<input checked="" type="checkbox"/> CapaciteMini	3	

Body Cookies Headers (12) Test Results Status: 200 OK Time: 1836 ms Size: 581 B Download

Pretty Raw Preview JSON ↺

```
1 {  
2   "Emplacement": "bien a coté de fenetre",  
3   "CapaciteMax": "4",  
4   "CapaciteMini": "3",  
5   "id": 13  
6 }
```

```
{  
  "Emplacement": "bien a coté de fenetre",  
  "CapaciteMax": 4,  
  "CapaciteMini": 3,  
  "id": 13  
}
```

Figure 12 : test d'ajout d'un table

La disponibilité :

Le système affiche tous les tables disponibles et l'utilisateur doit choisir l'emplacement préféré

The screenshot shows a REST client interface with a POST request to `http://localhost:8000/AviTab?DateArrive=2019-01-07&HeureArrive=8:00&PartySize=4&api_token=ka2c9A3lcVPNWTnJqRqP...`. The request parameters are listed in a table below.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> DateArrive	2019-01-07	
<input checked="" type="checkbox"/> HeureArrive	8:00	
<input checked="" type="checkbox"/> PartySize	4	
<input checked="" type="checkbox"/> api_token	ka2c9A3lcVPNWTnJqRqPnNcBwEUxrSOq6DguglXP	
Key	Value	Description

The response status is 200 OK, with a time of 13503 ms and a size of 1.2 KB. The response body is shown in JSON format:

```
{
  "id": 5,
  "CapaciteMini": 4,
  "CapaciteMax": 4,
  "Emplacement": "bien a coté de fenetre",
  "restaurants_id": 1
},
{
  "id": 6,
  "CapaciteMini": 2,
  "CapaciteMax": 4,
  "Emplacement": "bien a coté de fenetre",
  "restaurants_id": 1
}
```

Figure 13: test de verification la disponibilité des tables

Annuler une réservation :

Dans notre première version pour annuler une réservation on doit juste la supprimer dans la base de données.

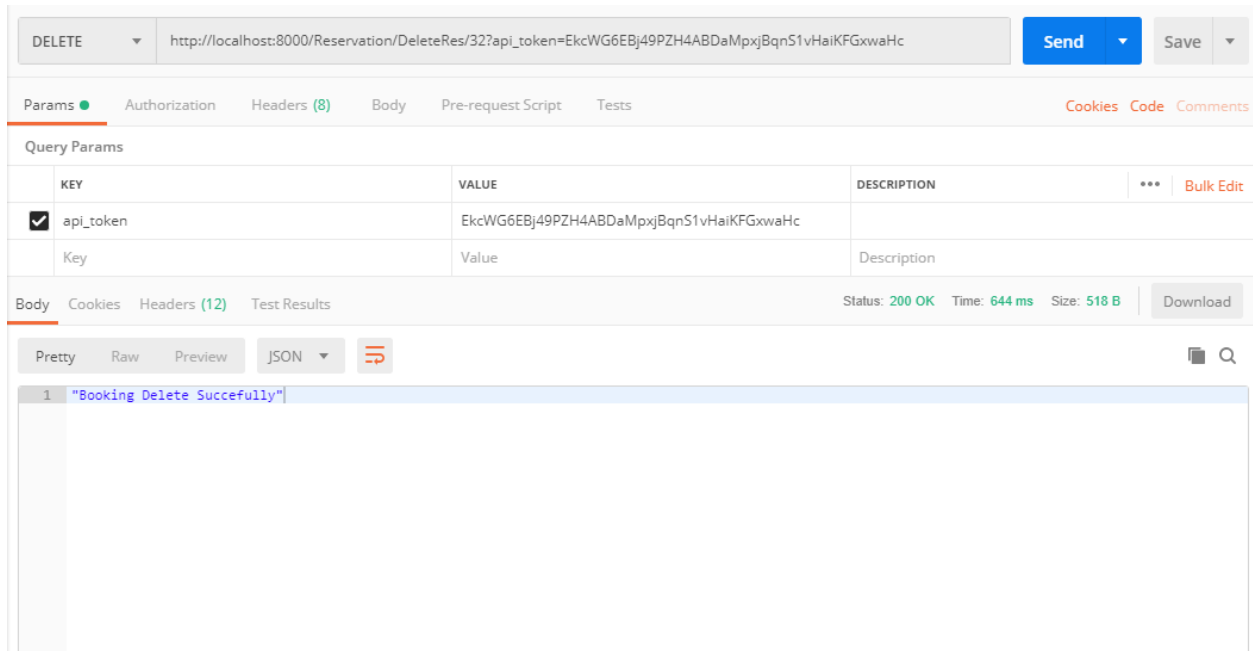


Figure 14: test d'annuler une reservation

Mise à jour d'une table :

L'admin doit changer la capacité de la table ou bien l'emplacement de cette table.

PUT ▼ http://localhost:8000/Table/Update?tables_id=5&CapaciteMax=4&CapaciteMini=3&Emplacement=bient Send Save ▼

Params ● Authorization ● Headers (9) Body Pre-request Script Tests Cookies Code Comments (0)

Query Params

	KEY	VALUE	DESCRIPTION	***	Bulk Edit
<input checked="" type="checkbox"/>	tables_id	5			
<input checked="" type="checkbox"/>	CapaciteMax	4			
<input checked="" type="checkbox"/>	CapaciteMini	3			
<input checked="" type="checkbox"/>	Emplacement	bient			
<input type="checkbox"/>					
	Key	Value	Description		

Body Cookies Headers (12) Test Results Status: 200 OK Time: 301 ms Size: 577 B Save Download

Pretty Raw Preview JSON ≡ 🔍

```
1 {  
2   "id": 5,  
3   "CapaciteMini": "3",  
4   "CapaciteMax": "4",  
5   "Emplacement": "bient",  
6   "restaurants_id": 1  
7 }
```

Figure 15 : test de mis à jour d'une table

2.10. Les diagrammes

Diagramme de cas d'utilisation

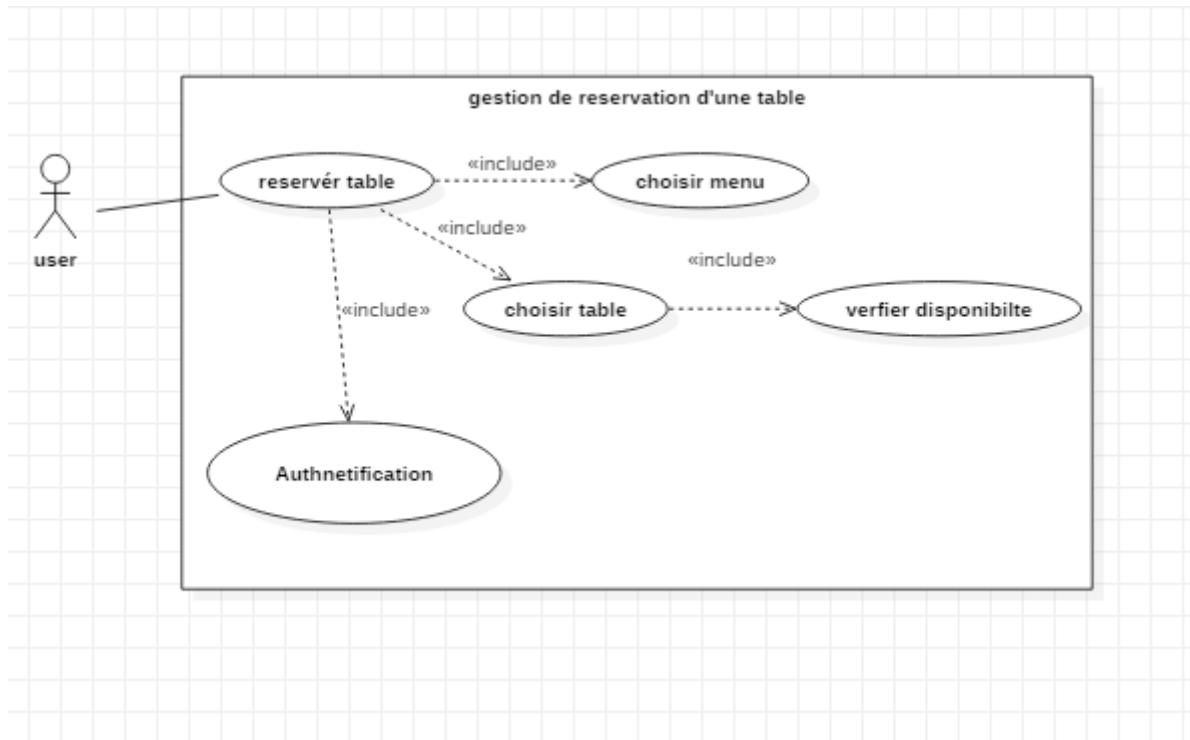


Figure 16: diagramme de cas d'utilisation

Diagramme de séquence

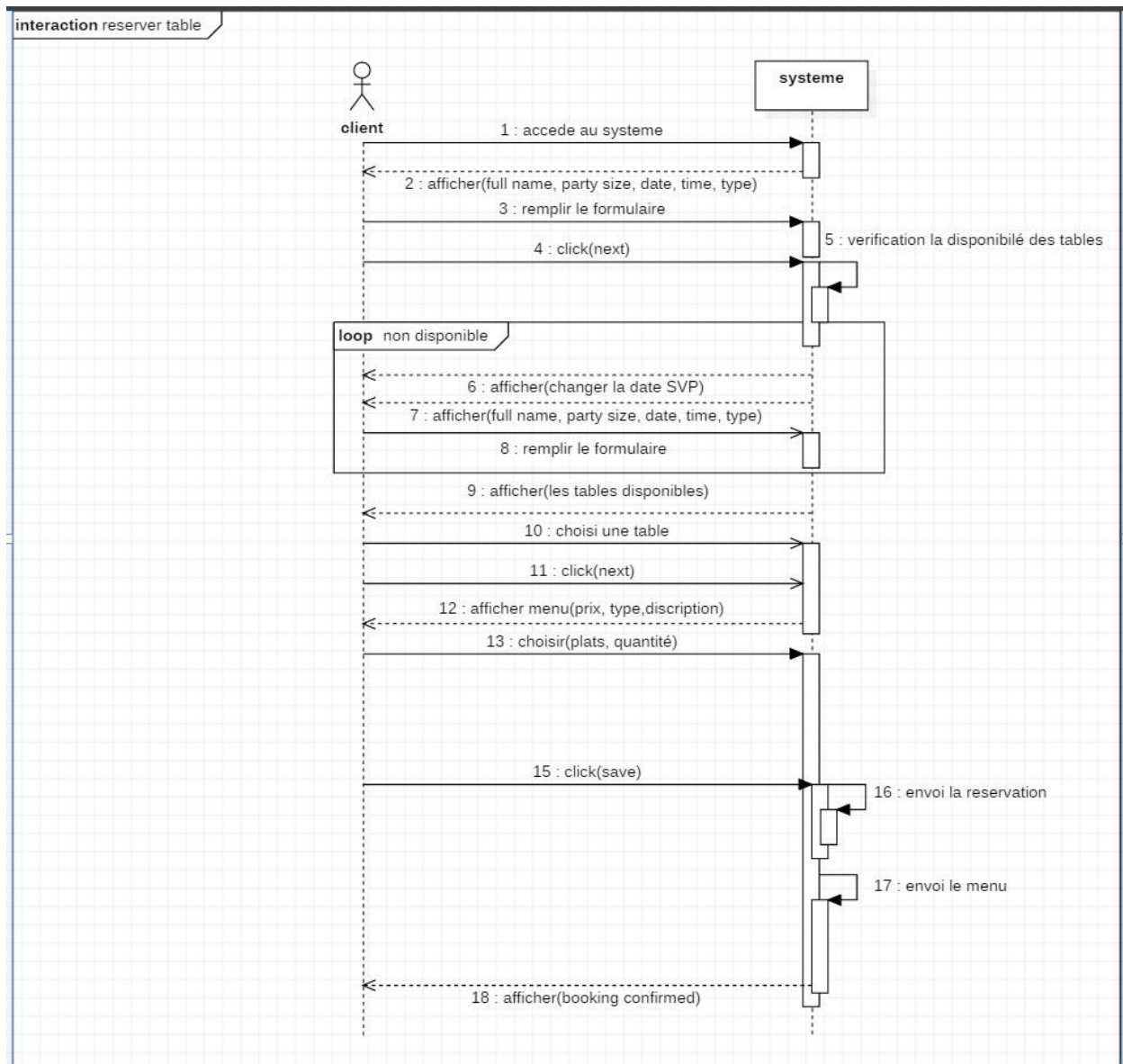


Figure 17: diagramme de sequence

3.Sprint 2 (développer la première partie (front-end))

3.1.Introduction

Ce sprint a pour but de lancer une première version en front-end et ajouter l'authentification en back-end qui permet au client de se connecter.

3.2.Backlog du sprint 2

Le Backlog du sprint contient une liste des tâches, identifiées par l'équipe Scrum qui devront être réalisées avant la fin de sprint.

ID	Histoire utilisateur	ID Tache	Tache
2	Développer la première partie (front-end)	2.1	Ecrire les trois premier chapitres
		2.2	Etudier la base de Angular 7
		2.3	La premier version front-end
		2.4	L'authentification

Tableau 4: Backlog du sprint 2

3.3. Analyse et réalisation du sprint 2

Après avoir étudié les différentes tâches du sprint 2, nous avons créé les fonctions pour pouvoir les tester après. Les résultats sont ci-dessous :

L'index de notre application

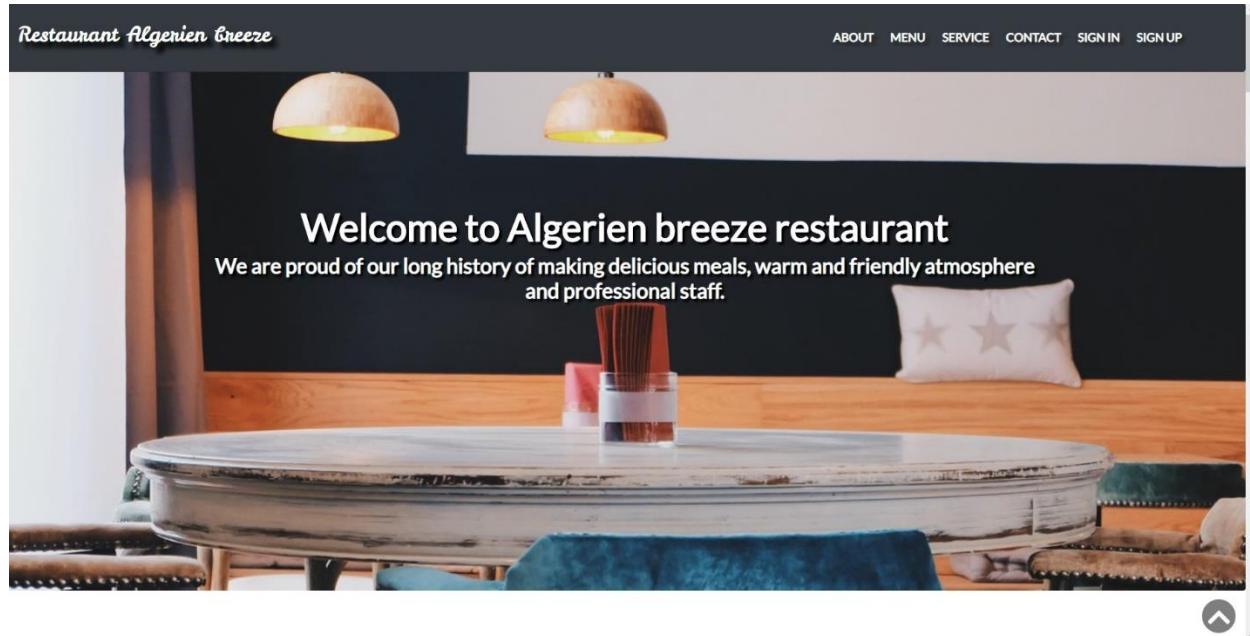


Figure 18 : l'index de site

Formulaire

The image displays a 'Make A Reservation' form overlaid on a background image of various Algerian dishes. The form includes the title 'Make A Reservation' and a subtitle: 'Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life'. The form fields are: 'Fullname' with a placeholder 'name', 'Date:' with a placeholder 'jj/mm/aaaa', 'Party Size' with an empty input field, 'Type' with an empty input field, and 'Time' with an empty input field. An orange 'NEXT' button is located at the bottom center of the form.

Figure 19 : le formulaire de la reservation

La liste des tables disponibles

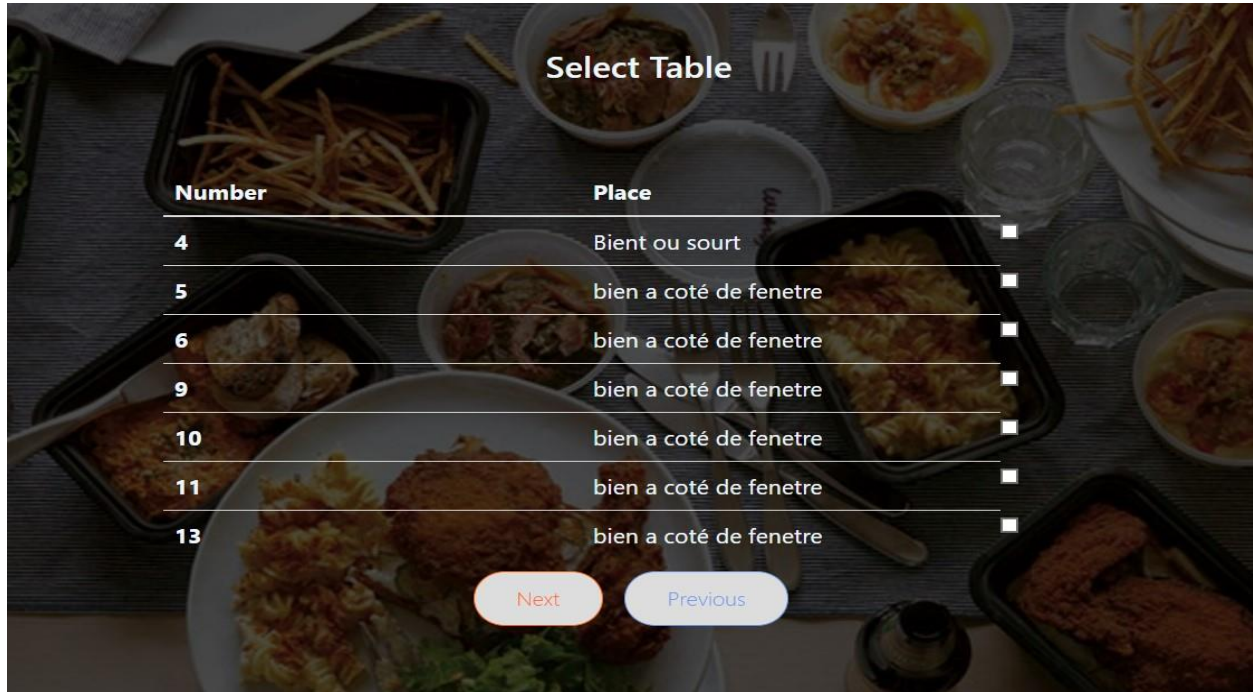


Figure 20 : liste des tables disponibles

Le menu

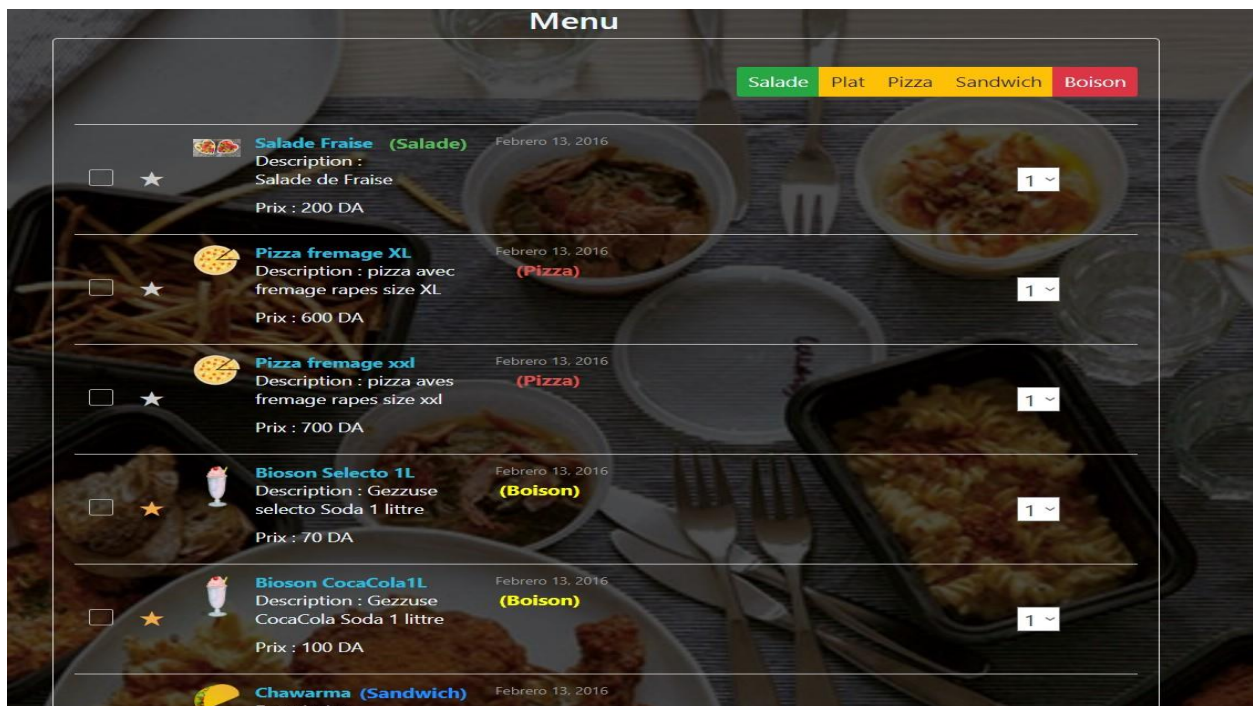


Figure 21 : le menu disponible

3.4.Lancer une premier version front-end

Après le développement d'une version fonctionnelle (serveur) qui contient l'essentiel du cahier de charge pour la réservation on va lancer une premier version Front-end avec Angular 7 qui utilise le back-end précédent.

3.5.La connexion entre le front-end et le back-end

Le fichier CorsMiddleware permet de la connexion entre le front-end et le back-end le fichier est comme suit :

```
CorsMiddleware.php x
1  <?php
2  namespace App\Http\Middleware;
3
4  use Closure;
5
6  class CorsMiddleware
7  {
8      /**
9       * Handle an incoming request.
10      *
11      * @param \Illuminate\Http\Request $request
12      * @param \Closure $next
13      * @return mixed
14      */
15     public function handle($request, Closure $next)
16     {
17         $headers = [
18             'Access-Control-Allow-Origin' => 'http://localhost:4200',
19             'Access-Control-Allow-Methods' => 'POST, GET, OPTIONS, PUT, DELETE',
20             'Access-Control-Allow-Credentials' => 'true',
21             'Access-Control-Max-Age' => '86400',
22             'Access-Control-Allow-Headers' => 'Content-Type, Authorization, X-Requested-With'
23         ];
24
25         if ($request->isMethod('OPTIONS'))
26         {
27             return response()->json(['method':"OPTIONS"], 200, $headers);
28         }
29
30         $response = $next($request);
31         foreach($headers as $key => $value)
32         {
33             $response->header($key, $value);
34         }
35
36         return $response;
37     }
38 }
```

Figure 22: Le fichier CorsMiddleware

3.6.La fonctionnalité authentification

On va créer une fonctionnalité en back-end(Serveur) permet de faire le login d'un utilisateur et l'administrateur.

3.7.Fiche descriptive

Cas d'utilisation	Authentification
Acteurs Principaux	L'administrateur, client
Objectif	Ce cas d'utilisation permet à l'administrateur ou client doit taper son propre nom et son mot de passe pour accéder à l'interface qui le concerne.
Pré-condition	Administrateur authentifié.
Scenarios Nominal	<ol style="list-style-type: none"> 1. L'utilisateur tape son propre nom d'utilisateur et son mot de passe. 2. Le système vérifie la validité des coordonnées de l'utilisateur. 3. L'utilisateur va accéder à la session qui le concerne.
Scenarios Alternatif Post-condition	<ol style="list-style-type: none"> 1. Si le mot de passe est incorrect alors L'enchaînement reprend à l'étape 1 du scénario nominal : 1.1 Le système afficher un message d'erreur puis renvoie à l'étape 2.

Tableau 5 : fiche descriptive d'authentification

3.8.Test et validation

3.8.1.Back-end

Authentification

On doit remplir le formulaire qui contient l'email et le mot-passe. cette fonction va retourner l'état succès ou bien Failure avec une api_token

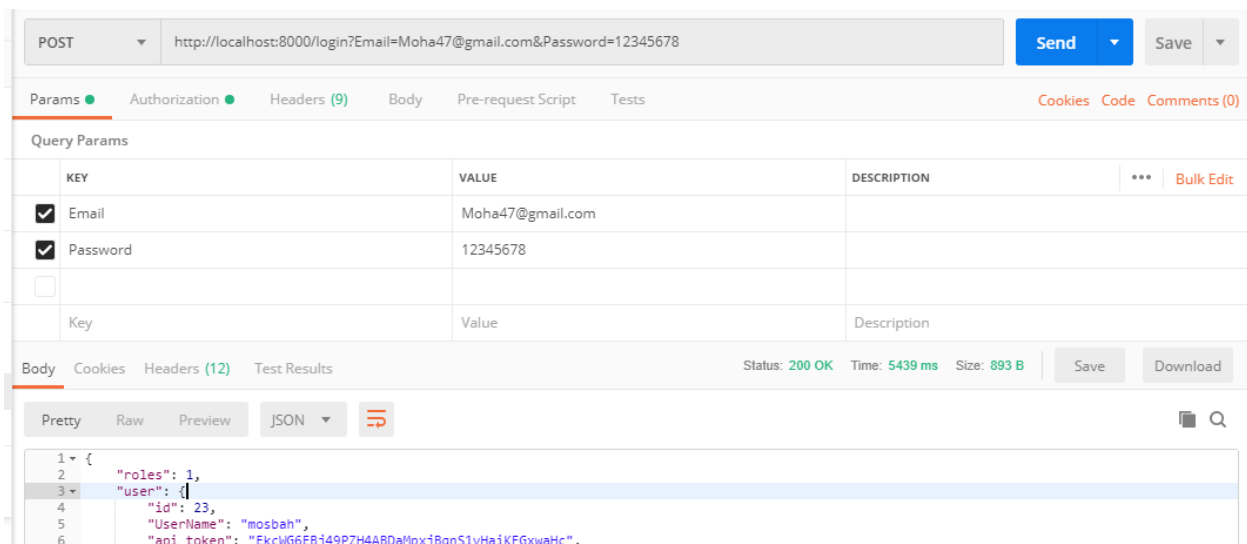


Figure 23 : le test d'authentification

3.8.2.Front-end**Scénario**

Cas de test	Démarche	Comportement attendu	Résultat
Test d'afficher les tables existes.	Importer les tables avec ces informations depuis la base de données et afficher sous forme d'une liste.	L'affichage d'une liste des tables.	Conforme
Affiche le menu disponible	Sélectionner depuis la page d'accueil	L'affichage d'un message d'informer le choix.	Conforme
Test de mis à jour le menu	Sélectionner depuis le dashboard de l'administrateur le choix « edit menu »	L'affichage d'un message d'informer le choix.	Conforme

Tableau 6 : Tableau des tests du sprint 2

4.Sprint 3 (l'interface pour l'authentification)

4.1.Introduction

Ce sprint a pour but de développer en front-end le bouton « login » (l'authentification) et l'ajoute d'une table, créer et annuler une réservation.

4.2.Backlog du sprint 3

Le Backlog du sprint contient une liste des tâches, identifiées par l'équipe Scrum qui devront être réalisées avant la fin de sprint.

ID	Histoire utilisateur	ID Tache	Tache
3	L'interface pour l'authentification	3.1	En utilise l'interface pour l'authentification, ajouter d'une table, et faire la réservation et annuler une réservation

Tableau 7: Backlog du sprint 3

4.3.Analyse et réalisation du sprint 3

Après avoir étudié les différentes tâches du sprint 3, nous avons créé les fonctions pour pouvoir les tester après. Les résultats sont ci-dessous :

L'authentification

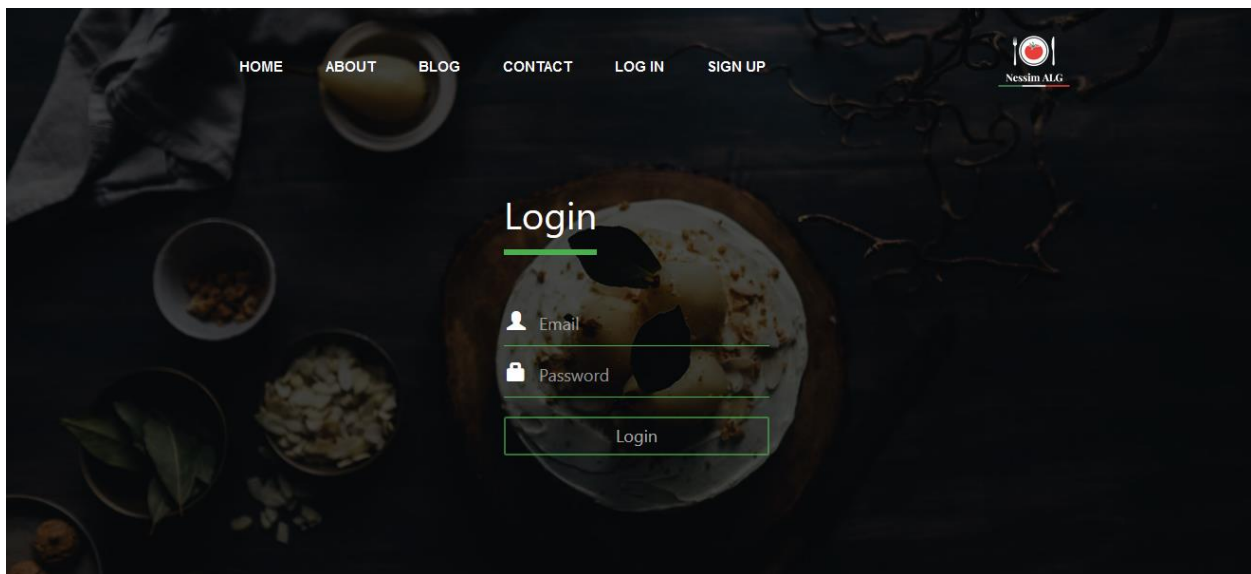


Figure 24 : le front d'authentification

Dashbord de l'utilisateur :

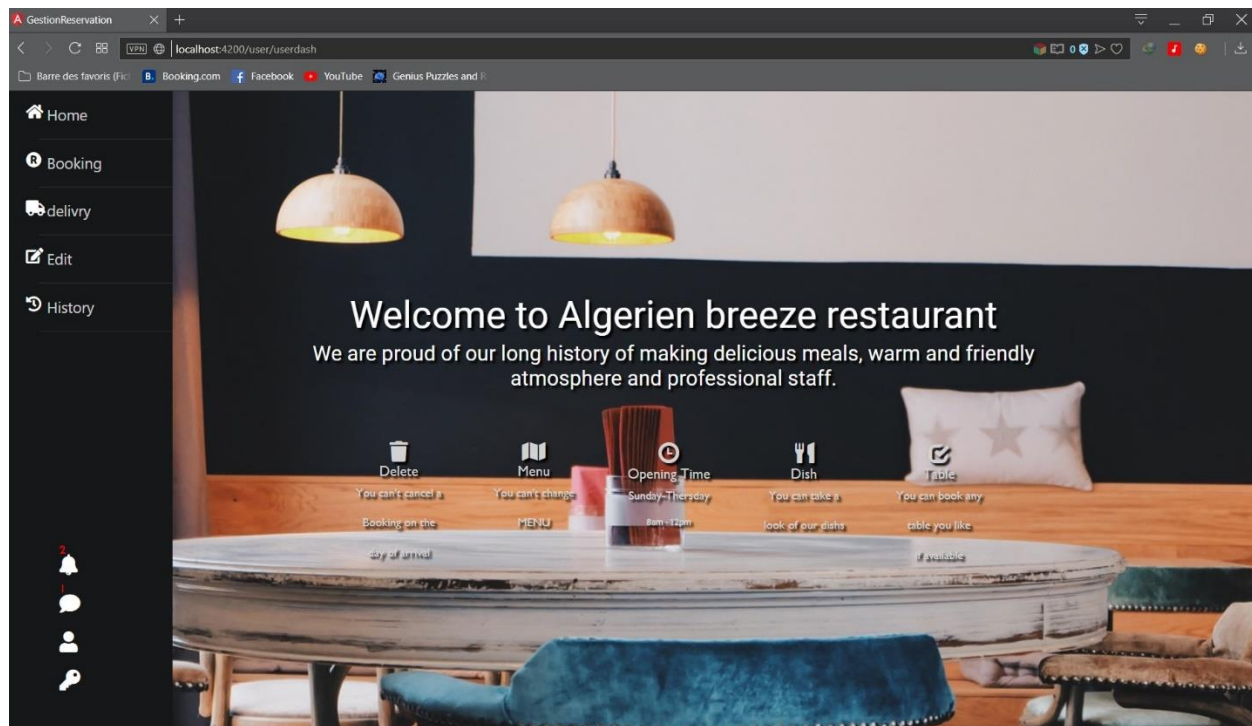


Figure 25: user dashbord

Dashbord de l'administrateur :

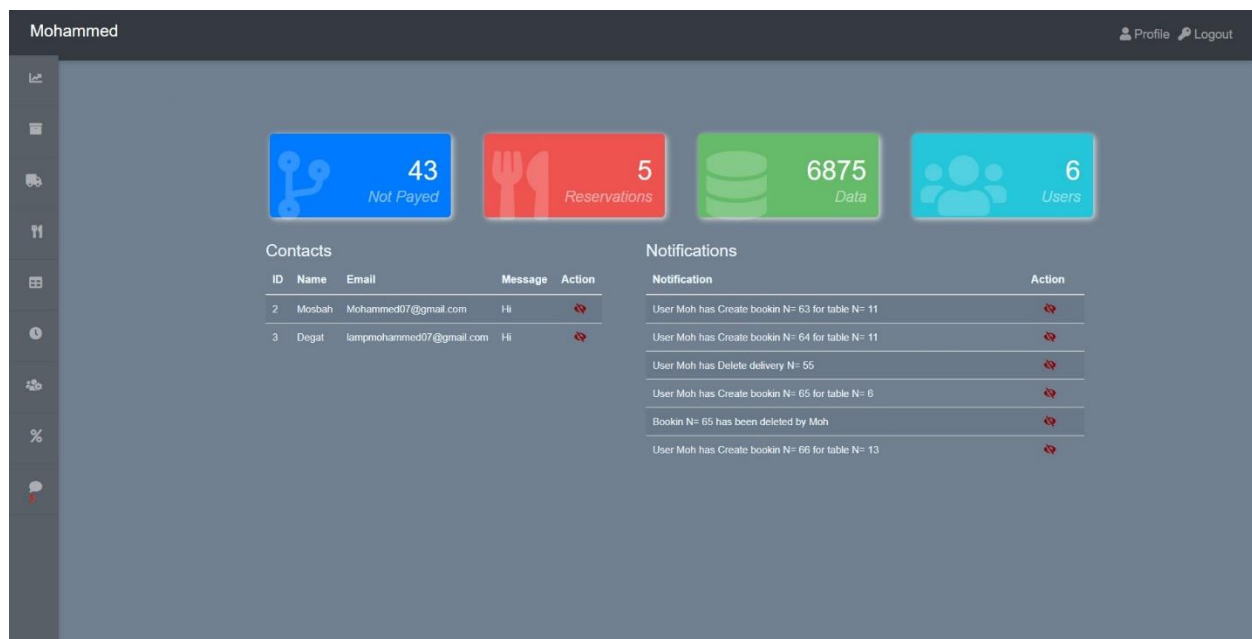


Figure 26 : admin dashboard

L'ajoute d'une table

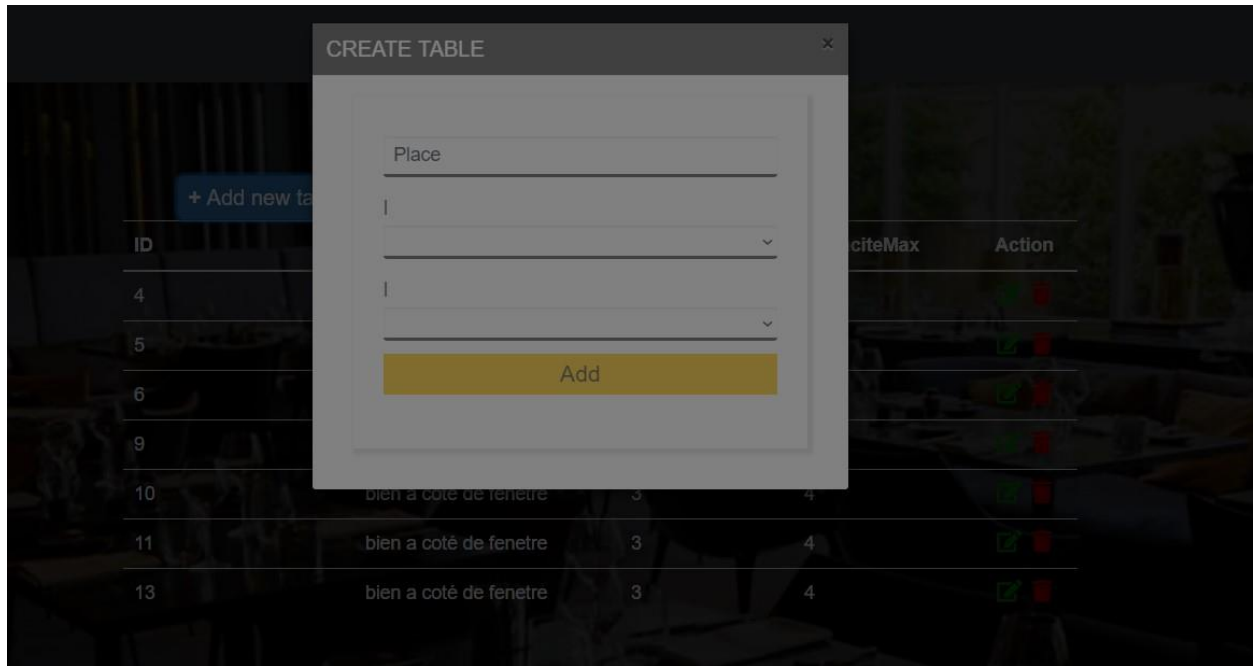


Figure 27 : create table "admin"

Annuler une réservation

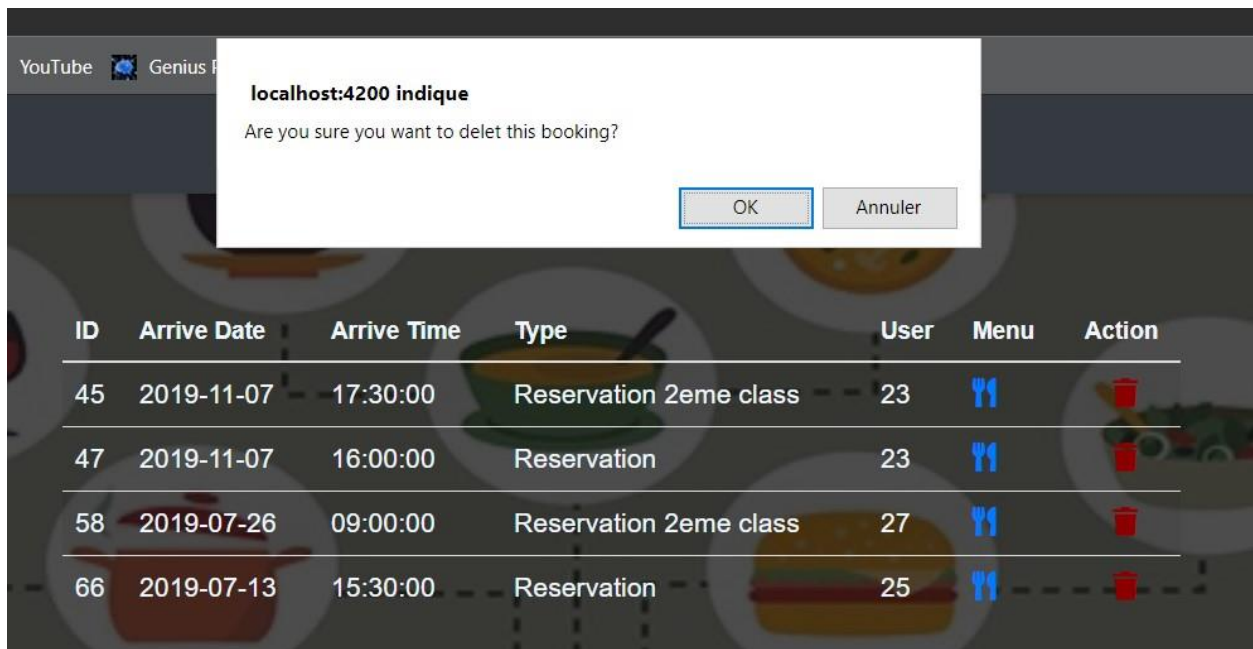


Figure 28 : annulation d'une reservation

4.4.Gestion des comptes

L'administrateur peut gérer les comptes des utilisateurs comme par exemple annuler la réservation à tout moment, supprimer ce compte et bien sûr afficher la liste de comptes qui existe dans la base de données.

4.5.Gestion des avis

Le client peut envoyer un message à l'administrateur pour consulter la qualité de la restaurant et vice versa c'est-à-dire l'administrateur peut envoyer un message au client s'il y a une issue.




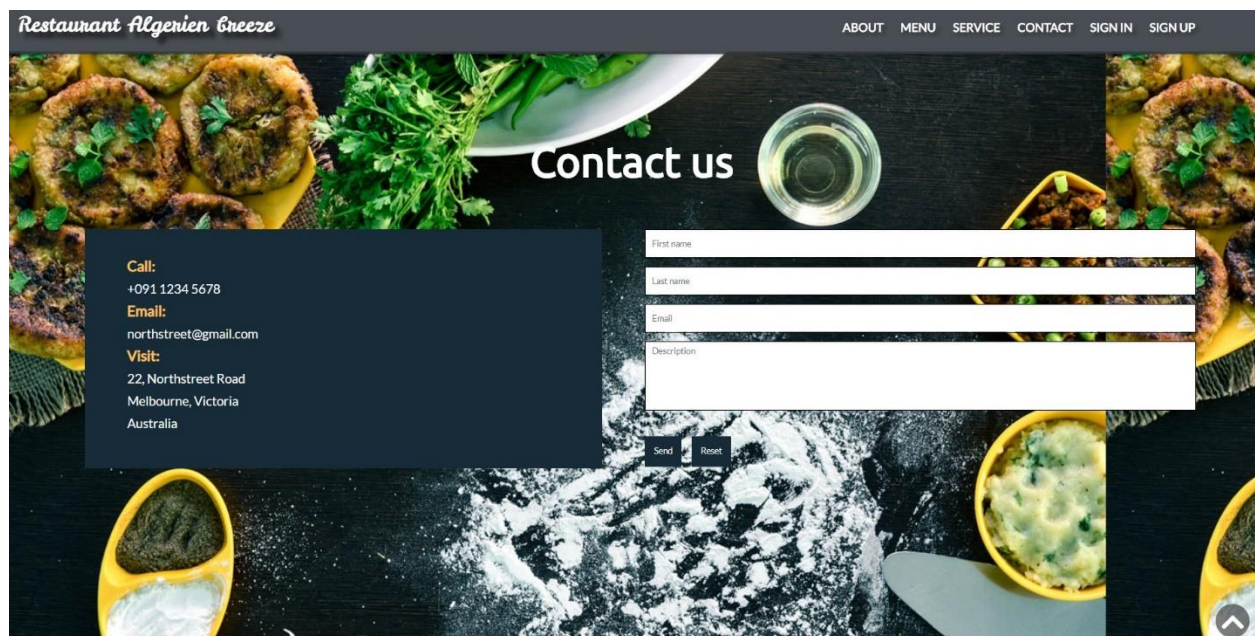
MESSAGES		
UserName	Message	Action
Moh	redme new	
Moh	fddeee	
Moh	faible	

Figure 29: l'interface de "send message" dans le user et l'admin



Restaurant Algerien Breeze

ABOUT MENU SERVICE CONTACT SIGN IN SIGN UP

Contact us

Call:
+091 1234 5678

Email:
northstreet@gmail.com

Visit:
22, Northstreet Road
Melbourne, Victoria
Australia

First name:

Last name:

Email:

Description:

Figure 30 : l'interface des messages pour le visiteur

4.6. Fiche descriptive

Cas d'utilisation	Gestion des comptes
Acteurs Principaux	L'administrateur
Résumé	Ce cas d'utilisation permet Ajoutée, modifiée ou supprimée avec succès.
Pré-condition	Authentifié.
Scenarios Nominal	<ol style="list-style-type: none"> 1. Le système affiche l'interface principale. 2. L'administrateur ou l'utilisateur clique sur l'onglet gérer compte. 3. Le système affiche l'interface de gérer le compte. 4. Le client ou l'administrateur modifier le compte (formulaire) 5. Valider le formulaire. 5. Le système vérifie le formulaire. 6. Le système informer que la modification est vérifiée.
Scenarios Alternatif	<ol style="list-style-type: none"> 5.1 Si le système d'détecte qu'un ou plusieurs champs sont vides : <ol style="list-style-type: none"> 5.1.1 Le système afficher un message d'erreur puis renvoie à l'étape 3.

Tableau 8: fiche descriptive de gestion des comptes

4.7. Test et validation

4.7.1 Back-end

Réservation d'une table :

Le client doit sélectionner type, dateArrive, heureArrive, le nom et la capacité

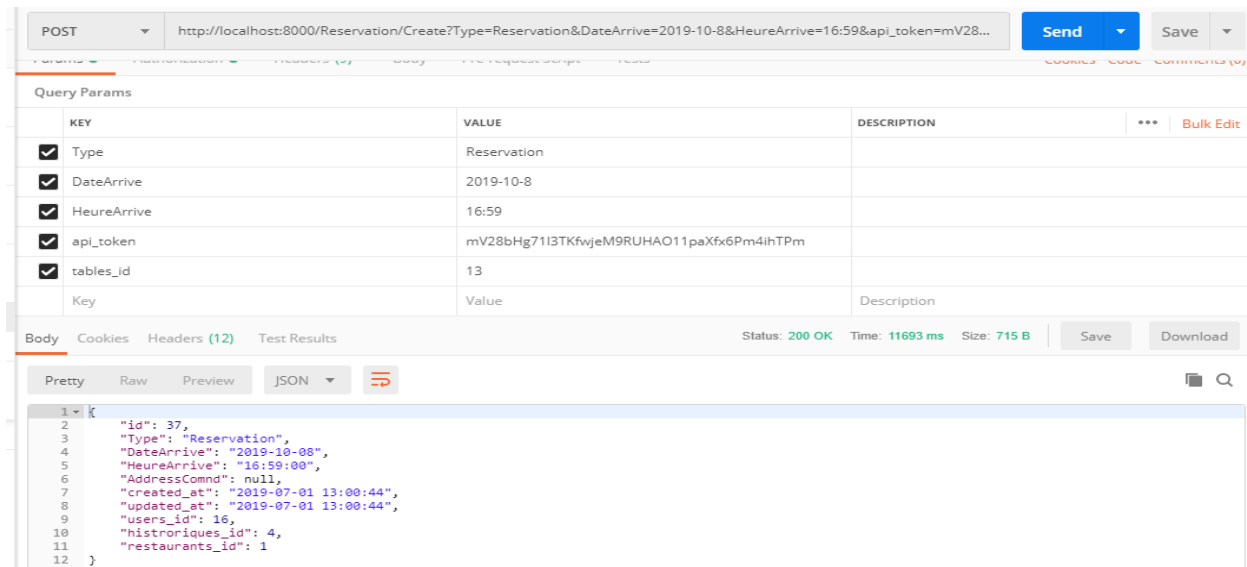


Figure 31: test de réservation d'une table

Sélection d'un menu :

Le client choisi parmi les différent plats, sandwiches, boissons, pizza, salades.

POST http://localhost:8000/Menu/Create?reservations_id=37&api_token=tar1CHcoKiZBUUpSei8adHtTIPvuqVP2G0CfpyD34

Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Cookies Code Comments (0)

none form-data x-www-form-urlencoded raw binary GraphQL BETA

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> p[0][id]	5	
<input checked="" type="checkbox"/> p[0][Quantite]	2	
<input checked="" type="checkbox"/> p[1][id] Text	6	
<input checked="" type="checkbox"/> p[1][Quantite]	2	
Key	Value	Description

Body Cookies Headers (12) Test Results Status: 200 OK Time: 652 ms Size: 497 B Save Download

Pretty Raw Preview JSON

```
1 {"date":
```

Figure 32: sléction de menu

4.7.2.Front-end

Scénario

Cas de test	Démarche	Comportement attendu	Résultat
Test d'authentification	Remplir le formulaire de login	Si le user et le password sont correct il peut aller a le dashboard sinon un message de vérification de votre user est password sera afficher.	Conform
Test d'ajout d'une table	Sélectionner depuis le dashboard de l'administrateur le choix « add table »	L'affichage d'un message de la confirmation de votre choix.	Conforme
Test de remplir le formulaire de réservation	Remplir le formulaire et garder la structure d'entrée des données Et cliquer sur « next »	L'affichage d'un message d'informer le choix.	conforme
Choisis une table.	Sélectionner depuis le checkbox les tables disponibles	L'affichage d'un message d'informer le choix.	Conforme
Choisis le menu	Sélectionner depuis le checkbox	L'affichage d'un message de la confirmation de votre choix.	Conforme
Annuler une réservation	Sélectionner depuis le dashboard le choix « delete booking »	L'affichage d'un message de la confirmation de votre choix.	Conforme

Tableau 9 : Tableau des tests du sprint 3

5.Sprint 4 (les horaires du travail)

5.1.Introduction

Ce sprint a pour but de spécifier les horaires du travail et d'ajouter des exceptions comme prendre des jours de congé ou prolonger les horaires pour certains jours.

5.2.Backlog du sprint 4

Le Backlog du sprint contient une liste des tâches, identifiées par l'équipe Scrum qui devront être réaliser avant la fin de sprint.

ID	Histoire utilisateur	ID Tache	Tache
4	Ajouter les horaires du travail	4.1	Spécifier les horaires de travail
		4.2	Ajouter des exceptions comme par exemple : prendre des jours de congé ou prolonger les horaires pour certains jours.

Tableau 10: Backlog du sprint 4

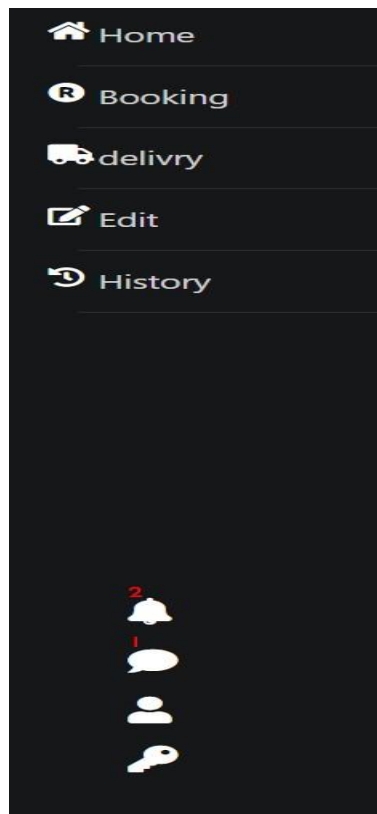
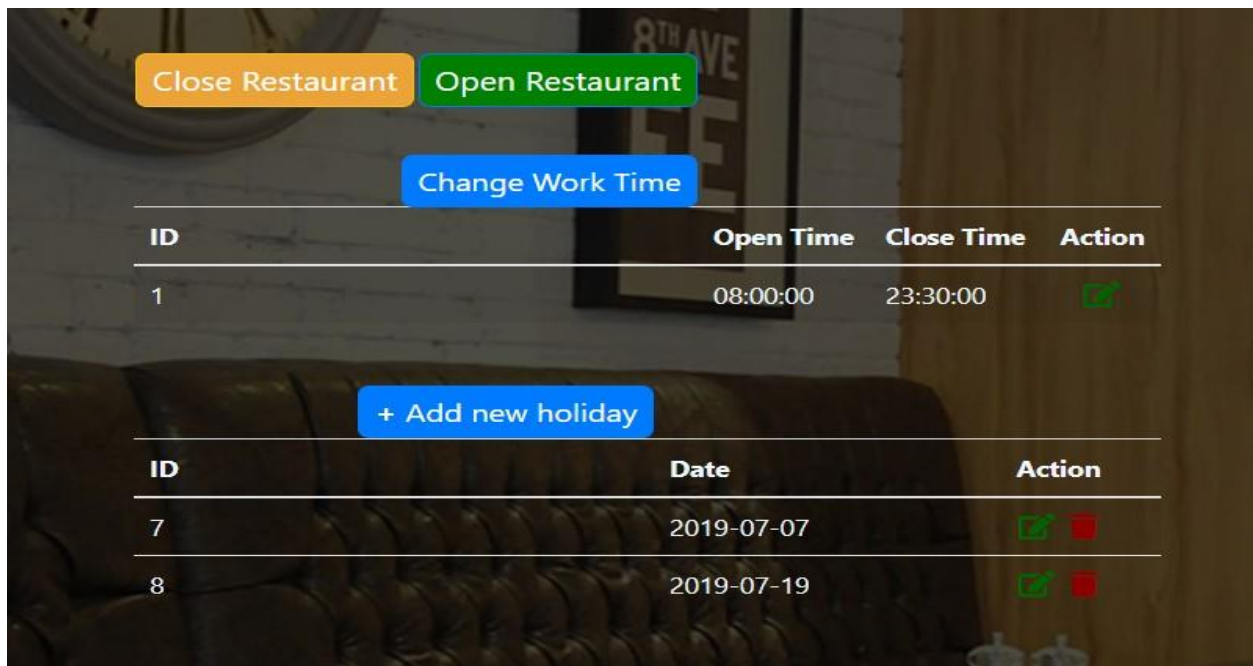
5.3.Analyse et réalisation du sprint 4

Après avoir étudié les différentes tâches du sprint 4, nous avons créé les fonctions pour pouvoir les tester après. Les résultats sont ci-dessous :

Les privilèges de l'administrateur :



Figure 33: admin privileges

Les privilèges de l'utilisateur :*Figure 34: user privileges***Les jours des congés et les horaires de travail :***Figure 35 : holiday test "front"*

5.4. Spécification des horaires de travail

Les horaires de travail dépendant fortement d'activité du restaurant, donc on a spécifié les horaires à partir de 08 :00 jusqu'à minuit alors le client doit réserver dans cette intervalle.

5.5. Ajout des exceptions

Le propriétaire peut ajouter des exceptions comme par exemple : prendre des jours de congé ou prolonger les horaires pour certains jours.

5.6. Fiche descriptive

Cas d'utilisation	Gestion des horaires de travail
Acteurs Principaux	Administrateur
Résumé	Ce cas d'utilisation permet gestion des horaires
Pré-condition	Authentifié.
Scenarios Nominal	<ol style="list-style-type: none"> 1. Le système affiche l'interface principale. 2. L'utilisateur clique sur l'onglet gérer horaire. 3. Le système affiche l'interface pour L'administrateur. 4. L'administrateur valider le formulaire. 5. Le système vérifie formulaire. 6. Le system informer que la modification est vérifier.
Scenarios Alternatif	<ol style="list-style-type: none"> 5.1. Si le system d'detecte qu'un ou plusieurs champs sont vides : <ol style="list-style-type: none"> 5.1.1. Le système afficher un message d'erreur puis renvoie à l'étape 3.

Tableau 11 : fiche descriptive de la gestion des horaires du travail

5.7. Test et validation

5.7.1. Back-end

Spécifier les horaires de travail :

L'administrateur peut entrer une date d'ouverture et de fermeture de la restaurant de chaque jour de la semaine.

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** `http://localhost:8000/Restaurant/ShowAll?api_token=mV28bHg71I3TKfwjeM9RUHAO11paXfx6Pm4ihTPm`
- Params:** `api_token=mV28bHg71I3TKfwjeM9RUHAO11paXfx6Pm4ihTPm`
- Status:** 200 OK
- Time:** 731 ms
- Size:** 745 B
- Response Body (JSON):**

```
{  "id": 1,  "NomRestaurant": "Al Waha",  "Note": 4,  "Chef": "Ahmed Eldjaberi",  "HeureOuvert": "08:00:00",  "HeureFerme": "23:30:00",  "Localisation": "Rue ALN Nouvelle Ville Constantine ",  "Etat": "Active",  "Redect": 50,  "DateRedect": "2019-05-01",  "DateRedect": "2019-11-12"}
```

Figure 36 : les horaires du travail

Prendre des jours de congé :

L'administrateur peut sélectionner n'importe quel jour pour prendre un congé.

POST http://localhost:8000/AviTab?DateArrive=2019-07-06&HeureArrive=5:00&PartySize=4&api_token=ka2c9A3lcVPNWTnjqR... Send Save

Params Authorization Headers (10) Body Pre-request Script Tests Cookies Code Comments (0)

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> DateArrive	2019-07-06			
<input checked="" type="checkbox"/> HeureArrive	5:00			
<input checked="" type="checkbox"/> PartySize	4			
<input checked="" type="checkbox"/> api_token	ka2c9A3lcVPNWTnjqRqPnNcBwEUxr5Oq6DguglXP			
Key	Value	Description		

Body Cookies Headers (12) Test Results Status: 403 Forbidden Time: 274 ms Size: 529 B Save Download

Pretty Raw Preview JSON

```
1 "We Are On Holiday Change Date"
```

Figure 37 : test de congé

5.7.2.Front-end**Scénario**

Cas de test	Démarche	Comportement attendu	Résultat
Test de réserver hors de l'intervalle de travail	Remplir le formulaire de la réservation hors de l'intervalle	Affichage d'un message que vous ne pouvez pas réserver dans ce temp vouez changer l'heure.	Conforme
Test de modification les horaires de travailles	Affiche le tableau des horaires puit modifier l'intervalle des heures.	L'affichage d'un message de la confirmation de votre choix.	Conforme

Tableau 12 : Tableau des tests du sprint 4

6.Sprint 5 (les limites)

6.1.Introduction

Ce sprint a pour but d'ajouter les limites de réservation par exemple : La durée du temps entre l'heure de réservation et l'heure d'arrivée au restaurant ne doit pas être moins de deux heures, la durée de réservation d'une table ne peut pas dépasser trois heures, le client ne peut pas annuler des réservations dans le même jour d'arrivée au restaurant.

6.2.Backlog du sprint 5

Le Backlog du sprint contient une liste des tâches, identifiées par l'équipe Scrum qui devront être réalisées avant la fin de sprint.

ID	Histoire utilisateur	ID Tache	Tache
5	Ajouter les limites	5.1	La durée du temps entre l'heure de réservation et l'heure d'arrivée au restaurant ne doit pas être moins de deux heures.
		5.2	La durée de réservation d'une table ne peut pas dépasser trois heures.
		5.3	Le client ne peut pas annuler des réservations dans le même jour d'arrivée au restaurant.

Tableau 13: Backlog du sprint 5

6.3.Analyse et réalisation du sprint 5

Après avoir étudié les différentes tâches du sprint 5, nous avons créé les fonctions pour pouvoir les tester après. Les résultats sont ci-dessous :

La durée du deux heures :

The screenshot shows a web form titled "Make A Reservation". It includes a hamburger menu icon on the left. The form fields are: Fullname (filled with "khalil"), Type (filled with "Reservation 2eme class"), Date (filled with "2019-07-03"), Time (filled with "5:30pm"), and Party Size (filled with "3"). Below the fields is an orange "NEXT" button. A red error message at the bottom states: "Change time you can't booking less than 2hour before your coming". Above the form, there is a quote: "Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life".

Figure 38: la limite de 2 heures

La limite de 3 heures :

POST http://localhost:8000/ApiTab?DateArrive=2019-07-03&HeureArrive=5:00&PartySize=1&api_token=ka2c9A3lcVPNWTnJqR... Send Save

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> DateArrive	2019-07-03	
<input checked="" type="checkbox"/> HeureArrive	5:00	
<input checked="" type="checkbox"/> PartySize	1	
<input checked="" type="checkbox"/> api_token	ka2c9A3lcVPNWTnJqRqPnNcBwEUxr5Oq6DguglxP	
Key	Value	Description

Body Cookies Headers (12) Test Results Status: 403 Forbidden Time: 218 ms Size: 529 B Save Download

Pretty Raw Preview JSON

```
1 {"Change time you can't booking"}
```

Figure 39 : la limite de 3 heures

L'annulation de la reservation :

ID	Arrive Date	Arrive time	Type	Action
24	2019-12-03	13:00:00	Reservation	edit Del
26	2019-06-30	13:30:00	Reservation	edit Del
28			Commande	edit Del
30	2019-01-03	08:30:00	Reservation	edit Del
31	2019-01-04	08:30:00	Reservation	edit Del
35	2019-09-12	10:30:00	Reservation	edit Del
36	2019-09-12	10:30:00	Reservation	edit Del
40	2019-07-03	18:30:00	Reservation 2eme class	edit Del

You cant delete booking in coming day

Figure 40 : test d'annule la reservation dans le meme jour

6.4.La limite entre l'heure de réservation et l'heure d'arrive

On va limiter l'intervalle entre l'heure de réservation et l'heure d'arrivée au restaurant à partir de plus de deux heures.

6.5.Limite de réservation d'une table

On va limiter également la durée de réservation d'une table pour ne peut pas dépasser trois heures et le client ne peut pas annuler des réservations dans le même jour d'arrivée au restaurant.

6.6. Test et validation

6.6.1. Back-end

L'heure de réservation et l'heure d'arrivée au restaurant ne doit pas être moins de deux heures :

Le system vérifie si la réservation est compatible ou non.

POST http://localhost:8000/AviTab?DateArrive=2019-07-03&HeureArrive=18:00&PartySize=1&api_token=ka2c9A3IcVPNWTnjRqPnNcBwEUxr5Oq6DguglxP

Params Authorization Headers (10) Body Pre-request Script Tests Cookies Code Comments (0)

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> DateArrive	2019-07-03	
<input checked="" type="checkbox"/> HeureArrive	18:00	
<input checked="" type="checkbox"/> PartySize	1	
<input checked="" type="checkbox"/> api_token	ka2c9A3IcVPNWTnjRqPnNcBwEUxr5Oq6DguglxP	
Key	Value	Description

Body Cookies Headers (12) Test Results Status: 400 Bad Request Time: 333 ms Size: 566 B Save Download

Pretty Raw Preview JSON

```
1 {"Change time you can't booking less than 2hour before your coming"}
```

Figure 41: limite de 2 heures "back"

La limite de la réservation dans le même jour :

Le client ne peut pas annuler des réservations dans le même jour d'arrivée au restaurant.

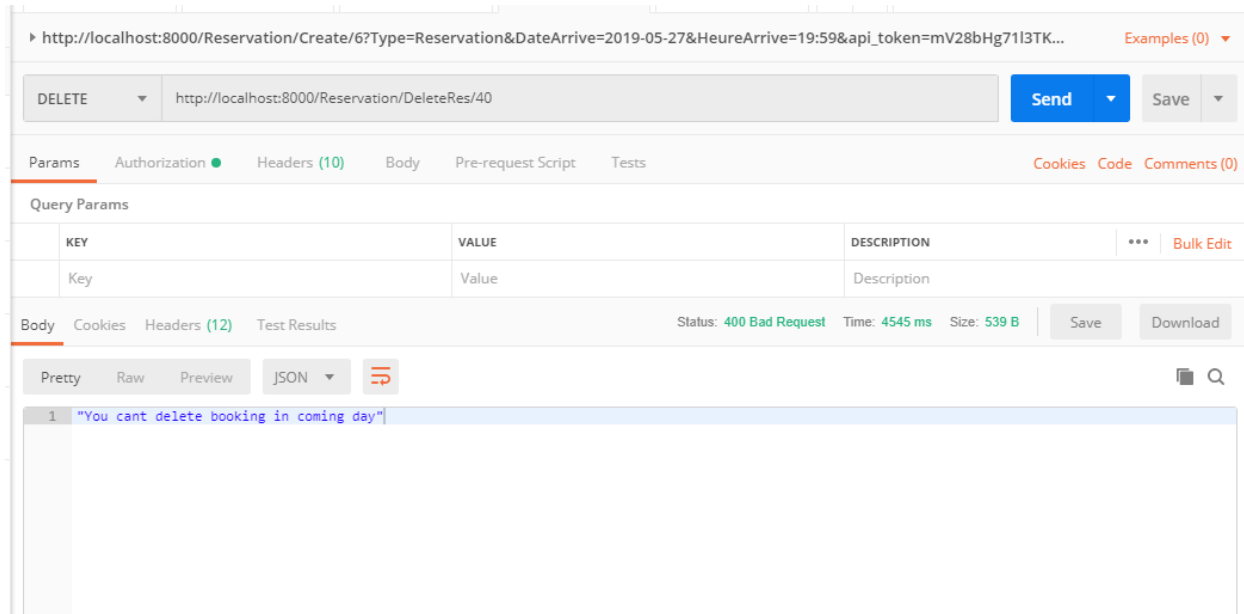


Figure 42 : annule la reservation dans le meme jour "back"

6.6.2.Front-end**Scénario**

Cas de test	Démarche	Comportement attendu	Résultat
Tester une réservation si elle est 2 h après la réservation	Test le temp à ce moment avec la date entrée par l'utilisateur	L'affichage message de validation ou non	Conforme
Suppression d'une réservation dans le même jour d'arriver	Par cliquer l'option « delete booking »	L'affichage d'un message d'informer le qui il ne peut pas faire cette action.	Conforme

Tableau 14 : Tableau des tests du sprint 5

7.Conclusion

Au cours de ce chapitre, nous avons effectué la conception et la réalisation du module de gestion d'espace administrateur qui gère les détails de la réservation « Table, menu, les horaires de travailles, les utilisateurs...ect » et l'espace d'utilisateur qui peut gérer son compte.

Conclusion générale

Ce mémoire représente le développement d'une application web pour la réservation en ligne d'un restaurant, nous avons bien compris la mise en œuvre et le déroulement d'un cycle de vie d'un logiciel, aussi il était une raison pour renforcer et enrichir nos expériences en utilisant des méthodes technologiques modernes comme la méthode agile grâce à sa efficacité au niveau mondial.

Pour accomplir notre projet on a utilisé les méthodes Agiles(Scrum) et pour la modélisation on a utilisé le langage UML. Notre application a une architecture RESTFUL, la partie back-end est développée en utilisant Lumen et pour la partie Front-end on a travaillé avec Angular 7 ainsi que les langages de programmation connus, y compris : html, css, javascript, php.

Nous avons réalisé notre projet pour le but de répondre aux besoins nécessaires des propriétaires des restaurants, en développant service web sert à les aider dans la gestion du restaurant. Notre application fait la réservation de restaurant et de réservation de table en ligne qui permette aux Restaurateurs d'attirer les clients consulter leurs menus et aider à maximiser leur profit en offrant des solutions innovantes grâce à notre expertise technique. Alors Notre travail contient presque toutes les fonctionnalités de bases citées dans le cahier de charge. Malgré tout cela, la technologie de l'information évolue constamment au jour le jour, de sorte que le jeune programmeur doit suivre ce développement pour continuer à donner.

Référence bibliographique

- [1] H. Bersini, La programmation orientée objet Cours et exercices en UML 2, avec Java 5, C# 2, C++, Python, PHP 5 et LINQ.
- [2] S. Mohamed, «cycle de vie liniere,» 2013-2014.
- [3] Cth027, «Modèle en cascade,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Modèle_en_cascade.
- [4] Christophe.moustier, «Cycle en V,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Cycle_en_V.
- [5] L. E. d. Faré, «Manuel d'introduction à Scrum et aux méthodes agiles,» January 25, 2017.
- [6] C. A. Kent Beck, «Extreme Programming Explained: Embrace Change, 2nd Edition,» 2004.
- [7] «C'est quoi une API,» [En ligne]. Available: <https://www.agencedebord.com/api-definition-utilisation/>.
- [8] S. Allamaraju, «RESTful Web Services Cookbook,» 2010.
- [9] Oracle, «MySQL Workbench Guide de modélisation des données pour les développeurs et les DBA,» 2010.
- [10] D. HERRENSCHNEIDER, «Présentation du framework Laravel,» [En ligne]. Available: <https://www.supinfo.com/articles/single/5637-presentation-framework-laravel>.
- [11] bestmomo, «LUMEN,» 2015. [En ligne]. Available: <https://laravel.sillo.org/lumen/>.
- [12] N. Squad, «Deviens un ninja avec Angular».
- [13] «Documentation Laragon,» [En ligne]. Available: <https://laragon.org/docs/index.html>.
- [14] S. O. individuals, «apprenez postman ebook gratuit».