



MÉMOIRE FIN DE FORMATION POUR L'OBTENTION DU DIPLÔME DE LICENCE PROFESSIONNELLE

OPTION : TELECOMMUNICATIONS ET INFORMATIQUE

SPÉCIALITÉ : DEVELOPPEMENT D'APPLICATIONS REPARTIES

THEME

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

Sous la direction de

- **M. GUEYE Cheikh**, Développeur à Gainde 2000
- **M. SECK Assane**, Ingénieur informatique

Présenté et soutenu par

Mlle CISSE Fatou Aïssatou

Promotion 2020-2021

DEDICACE

Du fond du cœur, je dédie ce travail à tous ceux qui me sont chers,

A MES CHERS PARENTS

Que ce travail soit l'expression de ma reconnaissance pour les sacrifices consentis, le soutien moral et matériel que vous n'avez cessé de nous accorder.

Vous avez tout fait pour notre bonheur et notre réussite.

Que dieu vous préserve en bonne santé et vous accorde une longue vie.

A MON GRAND-PERE

Qui m'a toujours aimé et chéri.

A MES SCEURS, ONCLES ET TANTES

Vous étiez toujours présents pour m'aider et m'encourager.

Sachez que vous serez toujours dans mon cœur.

A tous mes chers amis qui n'ont cessé de nous encourager et de nous soutenir
et à tous ceux qui nous ont enseigné tout au long de notre cursus scolaire.

A tous mes camarades de la promotion de LPTI 2018-21.

A toute personne m'ayant aidé de près ou de loin, trouve ici l'expression de notre reconnaissance.

REMERCIEMENT

Nous tenons à remercier ALLAH le tout puissant de m'avoir donné la santé, la volonté, le courage et la patience pour mener à terme ma formation et pourvoir réaliser ce travail de recherche.

En guise de reconnaissance, nos remerciements s'adressent très sincèrement au Professeur Jean Marie Preira, responsable de la formation LPTI-3 DAR, nous avons eu l'honneur et la chance de bénéficier de ses connaissances et compétences, de ses précieux conseils et de son suivi tout au long de notre parcours académique. De même que le Professeur Ghislain Akinocho pour sa rigueur scientifique, l'estime et le respect. Merci infiniment.

A ces remerciements nous associons particulièrement le Professeur Assane Seck, pour son encadrement de qualité, sa motivation professionnelle, ses conseils et critiques constructives, ses corrections, sa gentillesse et sa patience ainsi pour le temps qu'il a consacré à la réalisation de ce travail.

A mon maître de stage Cheikh Gueye et à toute l'équipe Orbus Factory de Gainde 2000, qui nous ont ouvert les portes de leur structure dont ils avaient la responsabilité et qui ont tout mis à notre disposition pour nous permettre de travailler dans les meilleures conditions, nous formulons notre reconnaissance. Un grand merci à l'ensemble de la famille, parents et sœurs pour leur amour, leur confiance, leurs conseils ainsi que leur soutien inconditionnel qui nous ont été nécessaires pour la réalisation de ce travail.

Nous souhaitons particulièrement remercier les amies de toujours, Marthe Louise Aby Kalamon et Diala Aicha Niang pour leur accompagnement et leur soutien durant toutes ces années. Nous exprimons, par ailleurs, à exprimer notre reconnaissance à l'endroit de nos camarades : Mame Diarra Ndiaye, Carmen Raïssa Hélène Diouf, Ramatoulaye Ndiaye, Mohamed Bachir Faye, Mamadou Aliou Diallo, Abdou Djafar Amadou, Abdourahmane Traoré et toute la promotion LPTI 2019-2021 qui nous ont également apporté soutien moral et intellectuel tout au long de ce travail.

Nous remercions, aussi, les membres du jury pour leur présence, pour leur lecture attentive de ce mémoire, ainsi que pour les remarques qu'ils m'adresseront lors de cette soutenance afin d'améliorer mon travail. Ainsi que le personnel enseignant, espérant qu'il verra dans ce manuscrit le fruit du dévouement avec lequel il a fait preuve dans la tâche.

À tous, nous présentons nos sincères remerciements et notre profonde gratitude.

AVANT-PROPOS

Dans le cadre de sa mission de développement et de vulgarisation des Télécommunications/TIC en Afrique, l'ESMT développe des sessions de formation continue. Sous forme de séminaires et d'ateliers, elles participent au perfectionnement des techniciens, ingénieurs et cadres des entreprises et institutions africaines.

Grâce à un catalogue évolutif, la formation continue couvre l'ensemble des métiers traditionnels ou modernes des télécommunications/TIC. Elle se traduit par des offres de formation personnalisées, partant d'une analyse des besoins spécifiques du client, à la réalisation et au suivi, en passant par une évaluation sur le terrain.

Les dispositions statutaires de l'établissement prévoient un stage obligatoire de deux mois pour l'obtention d'un diplôme en licence professionnelle en développement d'applications réparties.

Le stage permet en effet aux étudiants :

- d'avoir un contact réel avec l'entreprise,
- d'effectuer une synthèse des connaissances acquises au cours des trois années de formation,
- de prendre conscience de l'environnement socioprofessionnel et de préciser ses aptitudes personnelles,
- De renforcer leurs capacités à concevoir et réaliser des applications ou des systèmes d'information correspondant aux besoins des entreprises.

Et le mémoire, qui constitue un support d'information, permet à l'étudiant de présenter le travail qu'il a effectué durant ce stage, et à l'école de pouvoir évaluer ce travail.

C'est dans cette optique que nous avons effectué un stage de 2 mois à Gainde 2000. Au cours de cette période, nous avons travaillé sur l'étude et la mise en place d'un système de gestion des commandes et de suivi livraison d'un restaurant.

Le déroulement du stage, l'étude et la réalisation du projet, les notions acquises ainsi que les difficultés rencontrées durant cette période font l'objet de ce mémoire.

GLOSSAIRE

BPMN	une représentation graphique permettant de spécifier les processus métier dans un modèle de processus métier .
Chromium	Chromium est un navigateur web libre développé par l'organisation Chromium Project créée par Google en 2008. Chromium sert de base à plusieurs autres navigateurs, dont certains open-source ou certains propriétaires. Le moteur de rendu de Chromium s'appelle Blink.
C#	Un langage de programmation orienté objet. C# fournit des constructions de langage pour prendre en charge directement ces concepts.
ElectronJS	Un environnement permettant de développer des applications multi-plateformes de bureau avec des technologies web. L'infrastructure est codée en node.js, et l'interface est bâtie sur les outils Chromium, la partie open source de Google Chrome.
Java	un langage de programmation orienté objet.
PragmaDev Studio	PragmaDev Studio est un outil de modélisation des systèmes communicants complexes.
StarUML	StarUML est un logiciel de modélisation UML (Unified Modeling Language) open source.
Application hybride	Une application hybride est une application utilisant le navigateur web intégré du support (Smartphone ou tablette) et les technologies Web (HTML, CSS et Javascript) pour fonctionner sur différents OS (iOS, Android, Windows Phone, etc.).

SIGLE

API	Application Programming Interface
BPMN	Business Process Model and Notation
BSD	Berkeley Software Distribution
CEO	Chief Executif Officer
DAC	Diagramme d'activité
DBA	administrateur de base de données
DCL	Diagramme de classes
DCO	Diagramme de communications
DCP	Diagramme de composants
DCU	Diagramme des cas d'utilisation
DET	Diagramme d'état-transitions
DGI	Diagramme global interactions
DOB	Diagramme d'objets
DPA	Diagramme de paquetages
DPL	Diagramme de déploiements
DO	Data Officer
DPO	data Protection Officer
DSC	Diagramme de structures composites
DSE	Diagramme de séquences
DTP	Diagramme de temps
MCD	Modèle Conceptuel des Données
MD	Managing Director
MERISE	Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise
MVC	Modele Vue Controller
OMG	Object Management Group
OMT	Object Modeling Technique
PMO	projet management office
PKI	Public Key Infrastructure
PPP	partenariat public-privé
RESI	responsable du système d'information interne
RTDS	Real Time Developer
SDK	Sotfware Development Kit
SI	Système d'Information
SII	Services en Ingénierie Informatique
TIC	Technologies de l'Information et de la Communication
UI	User Interface
UML	Unified Modeling Language
UWP	Universal Windows Platform
VM	Machine Virtuelle

TABLE DES FIGURES

Figure 1	Organigramme Gainde 2000
Figure 2	Logo d'UML
Figure 3	Les différents diagrammes en UML
Figure 4	Logo de Modelio
Figure 5	Logo de StarUML
Figure 6	Logo de PragmaDev Studio
Figure 7	Le diagramme de cas d'utilisation du client
Figure 8	Le diagramme de cas d'utilisation du restaurateur
Figure 9	Le diagramme de cas d'utilisation du livreur
Figure 10	Le diagramme de classe du système
Figure 11	Le diagramme d'activité
Figure 12	Le diagramme de séquence
Figure 13	Architecture MVC
Figure 14	Logo de Ionic
Figure 15	Logo de Angular
Figure 16	Logo de React Native
Figure 17	Logo de Flutter
Figure 18	Logo de Symfony
Figure 19	Google Trends : flutter & react native
Figure 20	Statisque entre flutter & react native
Figure 21	Logo de Dart
Figure 22	Architecture de Flutter
Figure 23	Interface de l'application STADIA
Figure 24	Interface de l'application New York Times
Figure 25	Interface de l'application Alibaba
Figure 26	Logo de Firebase
Figure 27	Interface de l'application Pic Collage
Figure 28	Interface de l'application Fabulous
Figure 29	Interface de l'application Shazam
Figure 30	Logo de Cloud Firestore
Figure 31	Logo de Cloud Firebase Realtime Database
Figure 32	Logo de Cloud Firebase Authentification
Figure 33	Logo de Cloud Firebase Cloud Storage
Figure 34	Logo de Android Studio

TABLE DES CAPTURES

Capture 1	Profil de la personne
Capture 2	Age de la personne
Capture 3	Avis de la personne pour la confection de l'application mobile
Capture 4	Fonctionnalités restaurateur
Capture 5	Avis du restaurateur
Capture 6	Fonctionnalités client
Capture 7	Avis du client
Capture 8	Fonctionnalités livreur
Capture 9	Avis du livreur
Capture 10	Avis du restaurateur pour le prix de l'abonnement
Capture 11	Page de Bienvenue
Capture 12	Page d'authentification
Capture 13	Page d'inscription
Capture 13	Inscription validée
Capture 14	Inscription validée
Capture 15	Page d'authentification
Capture 16	Accueil client
Capture 17	Menu du restaurant 'Cafétéria ESMT'
Capture 18	Menu du restaurant 'Cafétéria ESMT'
Capture 19	Vérification de la commande
Capture 20	Profil client
Capture 21	Localisation de ESMT
Capture 22	Drawer client
Capture 23	Page d'authentification 'Restaurateur' avec mot de passe incorrecte
Capture 24	Page d'authentification 'Restaurateur' avec mot de passe correcte
Capture 25	Liste des plats
Capture 26	Liste des plats avec 'jus lait menthe archivé'
Capture 27	Désarchivage de 'jus lait menthe'
Capture 28	Suppression de 'jus lait menthe archivé'
Capture 29	Ajout de repas
Capture 30	Rajout de 'jus lait menthe'
Capture 31	Nouvelle liste après l'ajout
Capture 32	Liste commande du restaurant
Capture 33	Commande du client 'Fatou A. C.'
Capture 34	Livraison de la commande de 'Fatou A. C.'
Capture 35	Livraison validée

LISTE DES TABLEAUX

Tableau 1	Quels avantages et inconvénients du systèmes
Tableau 2	Comparaison des différents outils de modélisation
Tableau 3	Liste des classes
Tableau 4	Représentation des cardinalités
Tableau 5	Représentation des associations simples
Tableau 6	Caractéristique de mon pc
Tableau 7	Comparaison entre flutter et react native

SOMMAIRE

INTRODUCTION GENERALE

PREMIERE PARTIE : PRESENTATION GENERALE

**CHAPITRE I : PRESENTATION DE LA STRUCTURE D'ACCUEIL :
GAINDE 2000**

- I. HISTORIQUE
- II. COMPOSITION DE L'ARCHITECTURE ORGANISATIONNELLE
- III. ORGANIGRAMME ET MISSIONS D'ORBUS FACTORY

CHAPITRE II : ETUDE PREALABLE

- I. CONTEXTE
- II. PROBLEMATIQUE
- III. ETUDE DE MARCHE
- IV. PROPOSITION DE SOLUTIONS

DEUXIEME PARTIE : MODELISATION CONCEPTUELLE ET ORGANISATIONNELLE

CHAPITRE I : MODELISATION CONCEPTUELLE

- I. CHOIX DE LA METHODOLOGIE DE CONCEPTION
- II. DIAGRAMME DE CAS D'UTILISATION
- III. MODELE CONCEPTUEL DE DONNEES

CHAPITRE II : MODELISATION ORGANISATIONNELLE

- I. DIAGRAMME D'ACTIVITE
- II. DIAGRAMME D'ETAT TRANSITION

TROISIEME PARTIE : INPLEMENTATION DE L'APPLICATION

- I. ARCHITECTURE DE L'APPLICATION
- II. ENVIRONNEMENT DE TRAVAIL
- III. INTERFACES DE L'APPLICATION

CONCLUSION ET PERSPECTIVES

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant



Réalisé par Fatou Aïssatou CISSE – Promotion LPTI-DAR 2020-2021

INTRODUCTION

L'informatique fait actuellement partie intégrante de notre vie quotidienne. Ses diverses applications, en plus du développement d'internet, font qu'elle est devenue un outil indispensable dans notre vie professionnelle et privée. Bien qu'il soit un peu exagéré de parler d'invasion, l'informatique et ses diverses applications figurent partout dans la vie de tous les jours.

Le monde connaît actuellement une évolution technologique qui ne cesse de croître de façon exponentielle dans tous les secteurs (formel ou informel) et domaines d'activités. Ceci implique donc une révolution commerciale, qui bouleverse les rapports entre producteurs, distributeurs et clients, au profit de tous. Le facteur sociétal est désormais dominant à cause de la société qui devient une vraie partie prenante de l'entreprise, Internet et les réseaux sociaux obligent.

L'informatique qui est en constante mutation, propose un vaste étendu de technologies qui peuvent être utilisées dans presque tous les domaines possibles. Les méthodes de gestions employées jusqu'ici pour la plupart des entreprises ont prouvé leur efficacité, mais elles présentent néanmoins certaines insuffisances tels que le manque d'utilisation des solutions digitales qui ne cessent de montrer leur importance dans chaque domaine d'activité, le manque de services en ligne à réelle valeur ajoutée pour permettre aux consommateurs d'en bénéficier sans le moindre effort à travers des plateformes web, des applications mobiles, etc., le manque de suivi, d'assistance et de gérance des activités des clients, de la qualité des services fournis par ces tiers sont des contraintes qui se soulèvent fréquemment pour la plupart.

C'est au vu de toutes ces contraintes citées plus haut que s'inscrit notre sujet qui vise à développer une application de gestion de commande et de suivi d'un restaurant.

Tout au long de notre rédaction, nous tentons d'abord de présenter une étude préalable, spécification et analyse des besoins sur le sujet qui justifiera le choix des technologies utilisées pour la réalisation de la solution ; ensuite nous abordons l'analyse fonctionnelle et technique en ce qui concerne la mise en œuvre de la solution qui est la conception d'une application de gestion de commande et de livraison ; et enfin nous terminons par la présentation des différents outils utilisés et la réalisation de l'application.

PREMIERE PARTIE : PRESENTATION GENERALE

CHAPITRE I : PRESENTATION DE LA STRUCTURE D'ACCUEIL : GAINDE 2000

- I. HISTORIQUE**
- II. COMPOSITION DE L'ARCHITECTURE ORGANISATIONNELLE**
- III. ORGANIGRAMME ET MISSIONS D'ORBUS FACTORY**

CHAPITRE II : ETUDE PREAMABLE

- I. CONTEXTE**
- II. PROBLEMATIQUE**
- III. ETUDE DE MARCHE**
- IV. PROPOSITION DE SOLUTIONS**

PREMIERE PARTIE : PRESENTATION GENERALE

CHAPITRE I : PRESENTATION DE LA STRUCTURE D'ACCUEIL : GAINDE 2000

I. HISTORIQUE

GAINDE 2000 est une entreprise sénégalaise leader dans le domaine des Technologies de l'Information et de la Communication (TIC), spécialisée dans la facilitation du commerce, la modernisation des activités douanières et la dématérialisation des formalités publiques. Issue d'un partenariat public-privé (PPP), GAINDE 2000 a été créée en 2002 avec pour mission de développer et d'exploiter le guichet unique ORBUS afin de faciliter les formalités du commerce extérieur.

Son cœur de métier est de concevoir, développer et exploiter des solutions informatiques pour les gouvernements, les communautés portuaires et les entreprises. Ses solutions sont adaptées aux besoins des entreprises tout en apportant des services à valeur ajoutée pour une plus grande efficacité et de la performance. GAINDE 2000 accompagne également les pays qui souhaitent améliorer leur environnement des affaires et relever les défis d'une administration de service moderne grâce à la réduction des formalités, de l'usage du papier, des délais et des coûts dans les processus publics.

GAINDE 2000 s'est engagé depuis plus de dix ans dans de grands projets de dématérialisation pour faciliter aux entreprises les formalités administratives, douanières, portuaires et commerciales par la mise en place de plusieurs plateformes électroniques.

En effet depuis 2010, pour renforcer la sécurité de ses plateformes, GAINDE 2000 a mis en place une autorité d'enregistrement déléguée qui permet de produire des certificats électroniques, gage de la sécurisation des plateformes dématérialisées, conformément à la loi n°2008-08 du 25 janvier 2008 sur les transactions électroniques.

Cette loi de 2008 établit l'équivalence juridique entre le document numérique et le document papier à condition de respecter les règles de signature et d'archivage et facilite ainsi la dématérialisation des formalités.

Certifiée ISO 27001 : 2013 et ISO 9001 : 2015 dans le Management de la Qualité et Sécurité de l'Information, l'entreprise est également un leader africain dans la dématérialisation des formalités administratives, la sécurité numérique et les paiements électroniques avec des solutions déployées dans cinq (5) pays qui sont (Kenya, Burkina Faso, Côte d'Ivoire, Guinée Conakry, Guinée Bissau) et des services de consultance réalisés à l'internationale.

II. COMPOSITION DE L'ARCHITECTURE ORGANISATIONNELLE

L'architecture organisationnelle est composée :

D'une part, de trois (03) pôles d'activités fonctionnant comme des entreprises virtuelles autonomes avec une gouvernance appropriée pour favoriser la performance financière, la croissance et la valorisation maximale des actifs du GIE GAINDE 2000. Ces entreprises virtuelles sont :

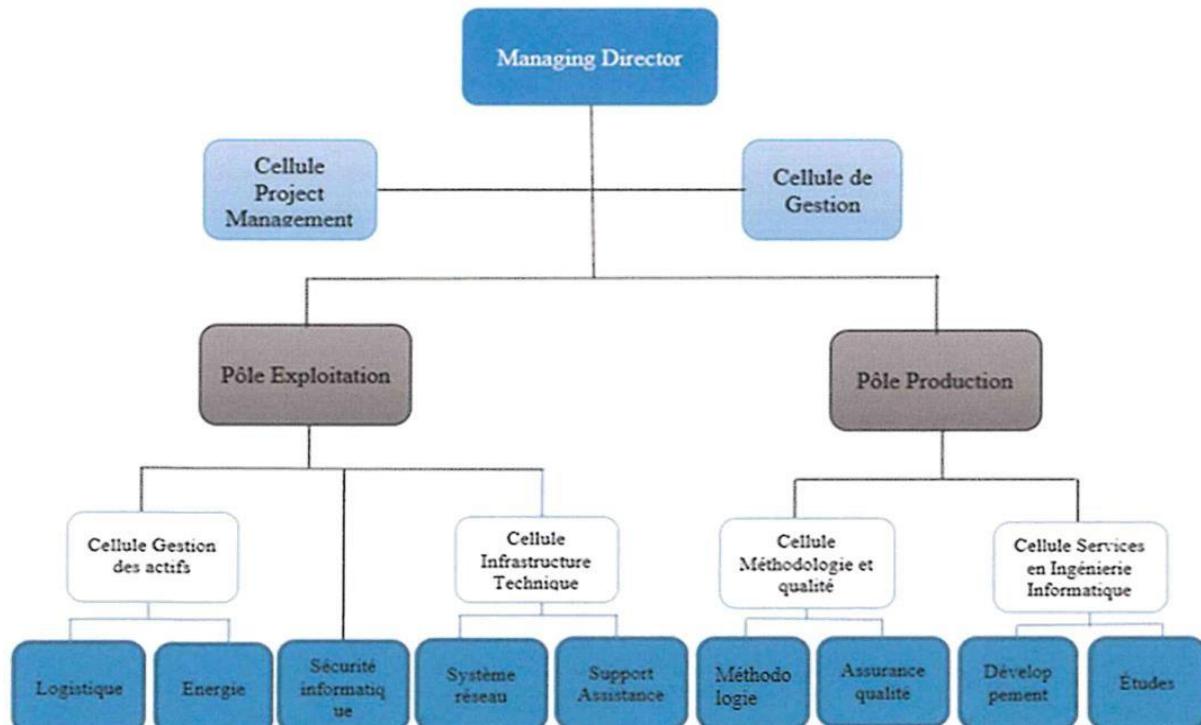
- ORBUS GUICHET UNIQUE DU COMMERCE EXTERIEUR ;
- ORBUS FACTORY;
- ORBUS DIGITAL SERVICES.

Chaque entreprise ayant à sa tête un Managing Director (MD).

D'autre part, d'un CORPORATE BOARD avec à sa tête l'Administrateur Général du GIE GAINDE 2000 qui décentralisera les centres de décision tout en renforçant les instances de supervision pour s'assurer de la bonne marche de l'entreprise. La traduction de la fonction de l'Administrateur Général dans la nouvelle terminologie managériale est « Chief Executif Officer (CEO) ».

III. ORGANIGRAMME ET MISSIONS D'ORBUS FACTORY

III.1 ORGANIGRAMME



III.2 MISSIONS

ORBUS FACTORY est articulée autour de deux pôles : un pôle d'exploitation et un pôle de production. Deux cellules accompagnent le Managing Director dans la gestion.

III.2.1 LA CELLULE PROJECT MANAGEMENT

La cellule renferme en son sein :

- Le projet management office (**PMO**) : qui est responsable du pilotage de l'ensemble du portefeuille de projets et qui réalise des missions de structuration et d'assistance au pilotage de projets. Il intervient, soit directement dans un projet auprès de chefs de projet, soit pour le compte de l'organisation pour un suivi transversal du portefeuille de projets ;
- Les chefs de projet : qui mettent en œuvre et conduisent les projets de système d'information (**SI**) depuis sa conception jusqu'à la réception dans le but d'obtenir un résultat optimal et conforme aux exigences formulées par le client métier en ce qui concerne la qualité, les performances, le coût, les détails et la sécurité.

III.2.2 LE POLE EXPLOITATION

Il s'occupe de la gestion des actifs non informatiques, de la sécurité informatique et de l'infrastructure technique.

La cellule de gestion des actifs non informatiques :

Elle comprend en son sein :

Un volet logistique en charge de la gestion du patrimoine, de l'aménagement et de la maintenance des locaux.

Un volet en charge de la gestion du parc énergétique, de la maintenance préventive et curative et des systèmes de contrôle d'accès et de vidéo surveillance.

La sécurité informatique :

La sécurité informatique accompagne les clients en matière de sécurité informatique. Elle assure la veille sécuritaire et l'exécution des marchés de service de sécurité informatique à l'extérieur. Elle s'occupe, en outre, de la certification sécurité des solutions développées et de certaines tâches de la Public Key Infrastructure (**PKI**).

La cellule de l'infrastructure technique :

Elle a en charge la gestion de l'infrastructure technique et de l'assistance aux utilisateurs. Elle s'occupe de l'infrastructure interne, du data center, de l'administration de la PKI, de l'enregistrement et du suivi des identités numériques, du contrôle et référentiel documentaire à la PKI et du support assistance interne et externe hors guichet unique ORBUS.

La cellule intervient dans l'exploitation des plateformes. Elle coordonne l'exécution des activités inhérentes à la gestion des services pour les utilisateurs et les clients.

On y retrouve les fonctions suivantes :

- administrateur réseau ;
- administrateur système ;
- administrateur de base de données (**DBA**) ;
- data Protection Officer (**DPO**);

- data Officer (**DO**);
- gestionnaire (Administrateur) d'applications;
- technicien assistant.

III.2.3 LE POLE PRODUCTION

Il s'occupe de la production logicielle, des méthodologies de développement et de l'assurance qualité des logiciels. Le pôle est composé d'une Cellule Services en Ingénierie Informatique (SII) et d'une Cellule Méthodologie et Qualité.

La Cellule des Services en Ingénierie Informatique (SII) :

Elle comporte en son sein les activités d'acquisition et de surveillance du système d'information interne, les activités d'études des besoins clients et celles de développement des logiciels.

On y retrouve les fonctions suivantes :

développeur ;
business analyst ;
Ux/UI Designer
chargé d'accompagnement à la conduite du changement ;
responsable du système d'information interne (**RESI**)

La cellule méthodologie et qualité :

Elle comporte en son sein les activités de promotion des méthodologies de développement et/ou des procédés standards et de la mise en œuvre et de la surveillance de la qualité des logiciels fournis.

On y retrouve les fonctions :

- de Scrum-master et d'assistance Devops ;
- de testeurs QA.

III.2.4 LA CELLULE DE GESTION

La cellule de gestion joue le relais avec le CORPORATE et garantit l'alignement entre ORBUS FACTORY et la stratégie du groupe. Elle assure les principales fonctions suivantes :

- **financière** : budget, trésorerie, banque, caisse, facturation, recouvrement ;
- **administrative** : gestion des demandes RH, formation, courriers, juridiques, traitement des missions ;
- **conformité** : performance, qualité, suivi manuel des procédures.

CHAPITRE II : ETUDE PREALABLE

I. CONTEXTE

Dans un monde où les consommateurs se tournent de plus en plus vers un mode de vie où la demande se voit supérieure à l'offre (société de consommation), le secteur de la restauration se voit submergé par l'augmentation de la demande suivie de la livraison de nourriture.

Les estimations de croissance prévues pour ce secteur (la restauration) sont aux alentours de 3%, mais quand on parle de livraison à domicile, on monte à plus de 13%.¹

La demande de livraison devrait aussi entraîner une augmentation des ventes dans l'ensemble de l'industrie de la restauration.

En 2020, la banque Morgan Stanley a prédit que le secteur de la livraison de nourriture pourrait représenter 11% de l'intégralité des ventes des restaurants, soit un total de 32 milliards de dollars.²

Cependant, la mise en place de la livraison n'est pas toujours pratique pour les restaurants : il leur faut une maîtrise de nouvelles technologies, de la formation et peut-être opérer un changement organisationnel.

Cela pourrait engendrer quelques difficultés dans le fonctionnement des processus internes aux restaurants mais, si ces changements sont effectués correctement, les livraisons peuvent être profitables pour booster les ventes et aussi la fidélisation de la clientèle.

L'intégration de la livraison dans un restaurant ne se fait pas du jour au lendemain.

Afin de tirer parti de cette nouvelle source de revenus, il convient d'avoir une stratégie bien pensée et ainsi éviter toute confusion pour les employés et les consommateurs.

Dans ce travail, nous étudions ces possibilités nouvelles, offertes aujourd'hui par internet et les outils digitaux.

II. PROBLEMATIQUE

Aujourd'hui, nous vivons dans une société de consommation où chaque consommateur a de nouvelles attentes et autant d'exigences. Ces attentes plus élevées ont été précipitées certainement par l'arrivée d'internet et la démocratisation de l'information.

Grâce à un smartphone il est possible d'identifier rapidement des restaurants à l'heure de pause, passer une commande et aller récupérer tranquillement ou encore se faire livrer le produit. Cela n'était pas possible il y a quelques années dans nos pays africains car l'accès à internet y était encore un luxe, et le secteur de la technologie très peu développé.

¹ <https://www.restaurantdrive.fr/mettre-en-place-système-livraison-restaurant/>

² Même lien

Avec la transformation digitale, connue de nos jours, nous nous sommes posé les questions à savoir comment gérait-on les systèmes de commandes traditionnelles dans nos restaurants ? Qu'en pensez les consommateurs ? Et bien évidemment quelles solutions peut-on apporter ?

Le secteur de la restauration souffre de plusieurs problématiques en Afrique et au Sénégal :

- ✓ Les heures de pause des travailleurs sont minimes et généralement c'est à des heures de pique où tout le monde essaie de sortir pour aller manger en même temps. Cette ruée vers les restaurants a deux conséquences : les restaurants sont très rapidement débordés car devant offrir un service maximal en un temps record, mais aussi les travailleurs sont en retard à cause de la durée d'attente qui peut prendre tout le temps imparti à la pause.
- ✓ Ce qui crée à coup sûr des frustrations au niveau des clients lorsqu'ils patientent dans les files d'attente mais aussi une perte de temps considérable car cette situation pouvait être évitée.
- ✓ Indirectement les systèmes de commande traditionnels au restaurant (se déplacer et aller au restaurant) peuvent causer des accidents car, en général, les gens conduisent rapidement et dans certains cas inconsciemment pour aller manger rapidement et vaquer à leurs occupations. Mais aussi, il y a le souci d'économie du carburant car à la place d'un déplacement aux heures de pause on pourrait simplement commander et se faire livrer.
- ✓ Enfin la covid-19 nous a montré l'importance des Tics dans la société de consommation particulièrement le secteur de la restauration car grâce à la commande à distance on peut réduire le stress et les contacts et limiter les possibilités de transmissions du virus.

Et donc, au vu de toutes ces problématiques aussi importantes les unes que les autres, deux questions générales s'imposent à notre thématique : Quelles solutions proposer aux restaurants pour la gestion des commandes à distance ? Et comment amener les consommateurs à y adhérer ?

III. ETUDE DE MARCHE

Cette étude a pour objectif d'étudier le système actuel de commande et de suivi de livraison. Elle permet de dégager ses points forts et ses points faibles. Dans ce qui suit, nous présentons une analyse du système actuel, puis détaillons les critiques de ce dernier.

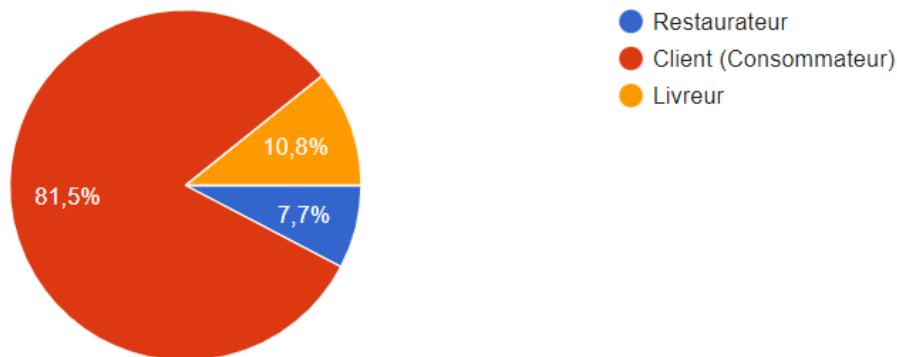
III.1 ENQUETE SUR LE SYSTEME ACTUEL

Pour avoir un aperçu sur le fonctionnement du système actuel de gestion des commandes et livraisons, nous avons procédé à un sondage en ligne à travers **Google form** auquel a répondu un certain nombre d'individus composé de restaurateurs, consommateurs et livreurs. Les résultats ainsi obtenus de cette enquête sont restitués par les figures présentées ci-dessous.

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

Qui êtes vous ?

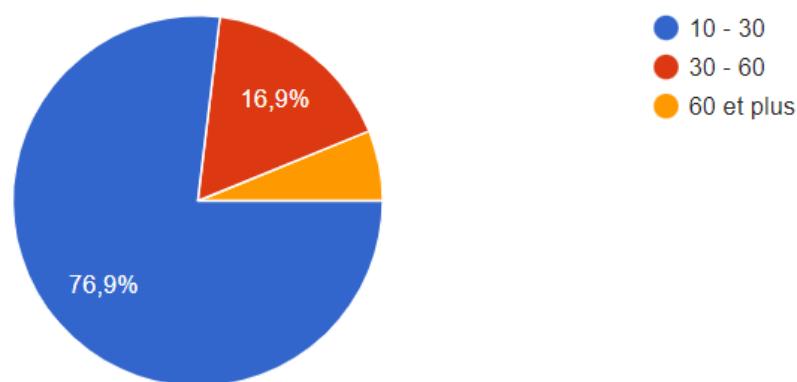
65 réponses



Capture 1 : Profil de la personne

Dans quelle tranche d'âge vous situez-vous ?

65 réponses

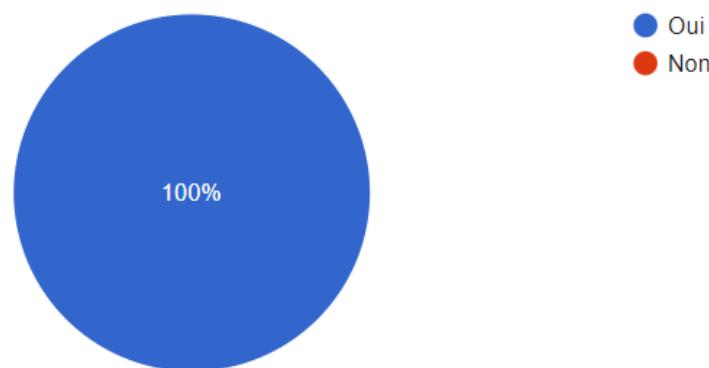


Capture 2 : Age de la personne

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

Pensez-vous que la confection d'une application mobile de gestion de commande et de livraison pour les restaurants disponible sur App Store comme Play Store serait bénéfique pour la société?

65 réponses

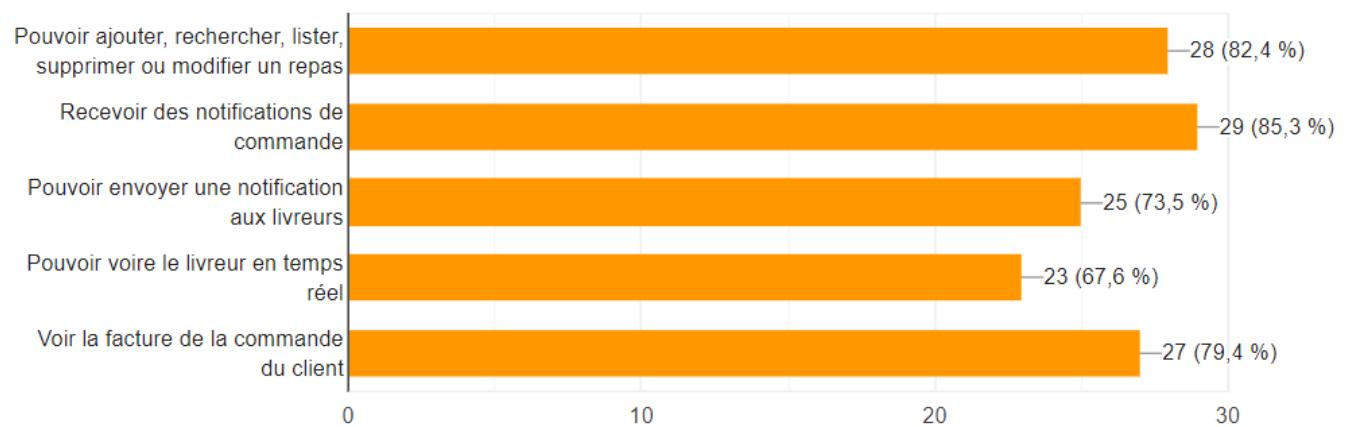


Capture 3 : Avis de la personne pour la confection de l'application mobile



En tant que Restaurateur, parmi les fonctionnalités ci-dessous lesquelles optez-vous pour la bonne gestion de votre espace personnel dans cette application?

34 réponses



Capture 4 : Fonctionnalités restaurateur

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

En tant que Restaurateur, pouvez-vous ajouter d'autres fonctionnalités ne figurant pas dans la liste.

10 réponses

Pouvoir avoir l'avis des clients après la commande

Localisation

Localisation du livreur

Fonction de traçabilité

Avoir une vue d'ensemble sur les étapes de la préparation de commande

Pouvoir savoir qui est en charge de la commande du client qui la traite en interne et les étapes de livraison

Je trouve la liste très bien mais peut-être avoir un diagramme de popularité de mon restaurant par rapport à la satisfaction de la clientèle

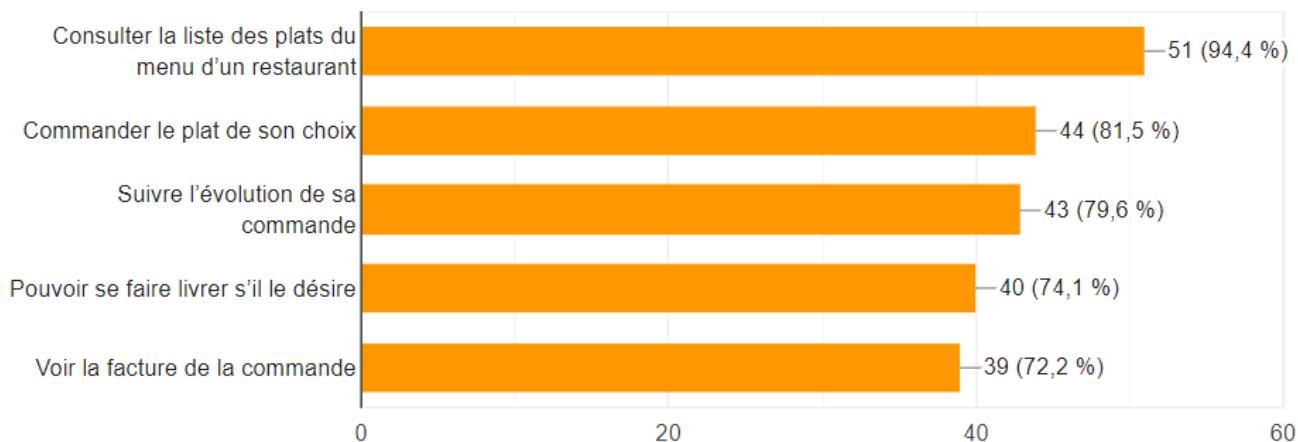
Vérifier toutes les entrées de la caisse

Pouvoir suivre l'état de la commande

Capture 5 : Avis du restaurateur

En tant que Client, parmi les fonctionnalités ci-dessous lesquelles choisissez-vous pour la bonne gestion de votre espace personnel dans cette application?

54 réponses



Capture 6 : Fonctionnalités client

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

En tant que Client, pouvez-vous ajouter d'autres fonctionnalités ne figurant pas dans la liste.

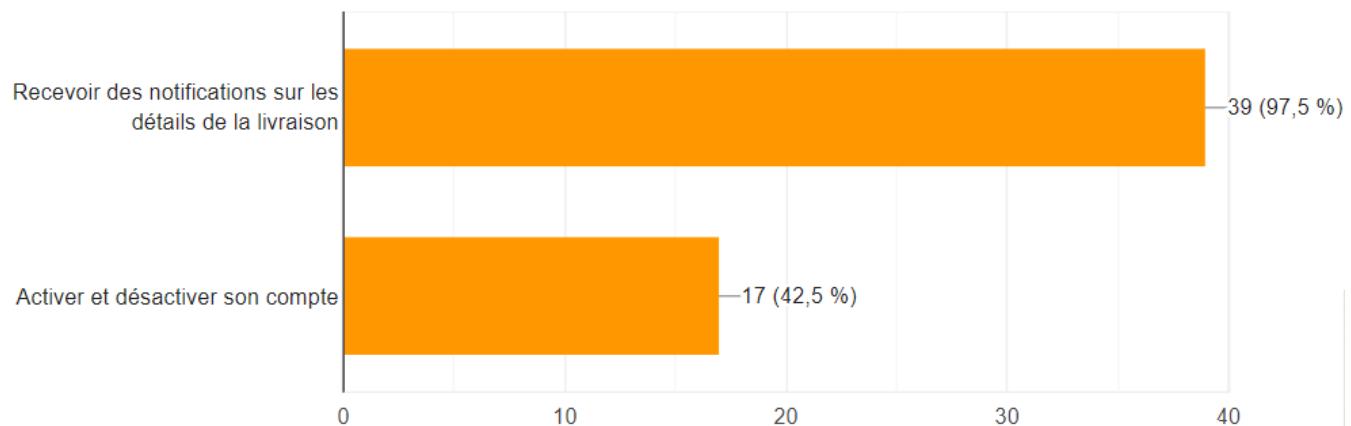
19 réponses

- Pouvoir apprécier la commande
- Localisation
- Localisation du livreur
- Le nom du livreur ; le temps estimé pour ma commande mais aussi ma livraison ; les moyens de paiements
- Pouvoir noter un plat, le livreur
- Rien à ajouter
- Le temps de livraison
- Voir les nouveautés
- Permettre au client de noter les différents livreurs selon leur efficacité

Capture 7 : Avis du client

En tant que Livreur, parmi les fonctionnalités ci-dessous lesquelles choisissez-vous pour la bonne gestion de votre espace personnel dans cette application?

40 réponses



Capture 8 : Fonctionnalités livreur

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

En tant que Livreur, pouvez-vous ajouter d'autres fonctionnalités ne figurant pas dans la liste.

10 réponses

Avoir la géolocalisation du client

Cartographie et localisation rapide

Cartographie et GPS

Par quel moyen le client va payer

Rien à ajouter

Au niveau des détails de la livraison avoir l'adresse ou la position exacte du client

Avoir la liste des livraisons à faire et leur état

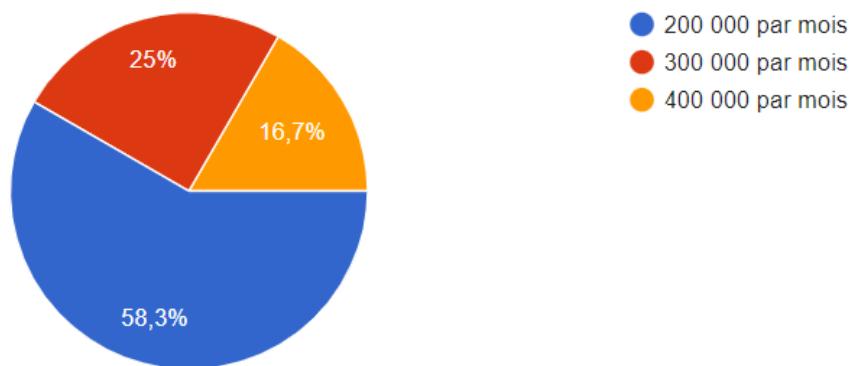
Pouvoir être noté d'après ma performance de livraison!!!!

Avoir la localisation du client

Capture 9 : Avis du livreur

En tant que Restaurateur, combien êtes-vous prêt à dépenser pour avoir un abonnement et figurer dans la liste des restaurants?

12 réponses



Capture 10 : Avis du restaurateur pour le prix de l'abonnement

Réalisé par Fatou Aïssatou CISSE – Promotion IPTI DAR 2020-2021

III.2 CRITIQUE

Le tableau suivant illustre quelques inconvénients découlant des résultats de notre enquête sur le système de commande et livraison actuel :

Inconvénients
Frustration des personnes lorsqu'elles patientent dans des files d'attente
Perte de temps dans les files d'attente
Non-respect de sa place dans la file pouvant entraîner à des embrouilles
Manque de temps dans la préparation de la commande pouvant détériorer la qualité de service
Énormément de stress pour le restaurateur comme pour le client
Énormément de stress pour le client en cas de retard au bureau
Plus de contact et plus de possibilités de transmission de virus (ex covid 19)
Risque d'accident dans la précipitation en allant chercher le repas
Erreur sur la commande du consommateur
Problème de monnaie (la caisse du restaurant)

Tableau 1 : Quels avantages et inconvénients du système

IV. PROPOSITION DE SOLUTIONS

L'étude de l'existant nous a permis de découvrir tous les inconvénients que présente le système de restauration actuel. Pour faire face à ces inconvénients, nous proposons de concevoir et mettre en place une application mobile de gestion de commande et livraison pour les restaurants du nom de 'Senfood'. Donc, notre solution englobe, à la fois, la gestion de commande et de livraison. Ce qui facilite le travail du restaurateur et du livreur.

Voici quelques avantages qui découlent de cette nouvelle gestion de commande et livraison.

- Les commandes en ligne génèrent plus de revenus
- Précision des commandes
- Simple, rapide et intuitif
- Amélioration de la productivité
- Les commandes en ligne roulent 24h/7
- Gardez une longueur d'avance sur la concurrence ; Sécuritaire pour vos clients et votre personnel
- Améliorez la satisfaction de vos clients
- Possibilités de suivre la livraison en temps réel
- Évitez les erreurs
- Gérer facilement le mode de paiement

Dans la solution proposée, nous envisageons de mettre plusieurs fonctionnalités que nous énumérons dans le deuxième chapitre de notre travail.

DEUXIEME PARTIE : MODELISATION CONCEPTUELLE ET ORGANISATIONNELLE

CHAPITRE I : MODELISATION CONCEPTUELLE

- I. CHOIX DE LA METHODOLOGIE DE CONCEPTION**
- II. DIAGRAMME DE CAS D'UTILISATION**
- III. MODELE CONCEPTUEL DE DONNEES**

CHAPITRE II : MODELISATION ORGANISATIONNELLE

- I. DIAGRAMME D'ACTIVITE**
- II. DIAGRAMME D'ETAT TRANSITION**

DEUXIEME PARTIE : MODELISATION CONCEPTUELLE ET ORGANISATIONNELLE

CHAPITRE I : MODELISATION CONCEPTUELLE

Le Modèle conceptuel de données (**MCD**) est une représentation des données, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités, autrement dit c'est une représentation statique du système d'information. Il a pour but d'écrire de façon formelle les données qui seront utilisées dans le système d'information .

Cette section, sera présentée comme suit : nous commençons par le choix de la méthodologie de conception et justification. Ensuite nous identifions les acteurs et les diagrammes des cas d'utilisation et enfin nous présentons le diagramme de classes.

I. CHOIX DE LA METHODOLOGIE DE CONCEPTION

Modéliser, c'est décrire de manière visuelle et graphique les besoins et les solutions fonctionnelles et techniques de notre projet logiciel. L'exemple que l'on peut prendre est celui d'une construction de maison. Un document de texte décrivant de façon précise ce qui doit être réalisé contiendrait plusieurs dizaines de pages. En général, peu de personnes ont envie de lire ce genre de document. De plus, un long texte de plusieurs pages est source d'interprétation et d'incompréhension. Raison pour laquelle les ingénieurs ont pensé le faire de manière graphique. C'est exactement ce à quoi **UML** (Unified Modeling Language) sert dans des projets de réalisation de logiciels !

UML nous aide à faire cette description de façon graphique et devient alors un excellent moyen pour visualiser le futur logiciel. Un logiciel qui a été réalisé sans analyse et sans conception (étapes où l'on modélise le futur logiciel) risque lui aussi de ne pas répondre aux besoins, de comporter des anomalies et d'être très difficile à maintenir. Pour la réalisation de ces graphes ou diagramme, nous utiliserons le logiciel Modelio.

Modelio est un outil UML open source développé par Modeliosoft, basé à Paris en France. Il prend en charge les normes UML2 et **BPMN** (Business Process Model and Notation).

PRESENTATION D'UML



Figure 2 : Logo d'UML

Pour faire face à la complexité des systèmes d'information, de nouvelles méthodes et outils ont été créées. La principale avancée des quinze dernières années réside dans la programmation orientée objet. Face à ce nouveau mode de programmation, les méthodes de modélisation classique (telle que **MERISE**) ont rapidement montré certaines limites et ont dû s'adapter. De très nombreuses méthodes de modélisation ont également vu le jour comme **Booch**³, **OMT**⁴ ...

Dans ce contexte et devant le foisonnement de nouvelles méthodes de conception orientée objet, l'**OMG** (Object Management Group) a eu comme objectif de définir une notation standard utilisable dans les développements informatiques basés sur l'objet. C'est ainsi qu'est apparu UML qui est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui permettent de représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc. Réaliser ces diagrammes revient donc à modéliser les besoins du logiciel à développer.

UML sert à obtenir :

- Une modélisation de très haut niveau indépendante des langages et des environnements.
- De faire collaborer des participants de tout horizon autour d'un même document de synthèse
- De documenter un projet
- De générer automatiquement la partie logiciel d'un système

UML est un langage formel et normalisé :

- Gain de précision
- Gage de stabilité
- Encourage l'utilisation d'outils

³ Voir tableau sigle

⁴ Voir tableau

UML est un support de communication performant

- Il cadre l'analyse.
- Il facilite la compréhension de représentations abstraites complexes.
- Son caractère polyvalent et sa souplesse en font un langage universel.

UML propose, en outre, un méta modèle de tous les concepts et notations associées utilisés dans les treize diagrammes du langage de modélisation regroupés en deux ensembles (diagrammes structurels et diagrammes de comportement).

 **Les diagrammes structurels** : représentent l'aspect *statique* d'un système.

- Diagramme de classes (**DCL**) : représente la description statique du système en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements.
- Diagramme d'objets (**DOB**) : représente les instances des classes (objets) et des liens entre instances.
- Diagramme de composants (**DCP**) : représente les différents constituants (fichiers sources, librairies exécutables, etc.) du logiciel au niveau de l'implémentation du système.
- Diagramme de déploiements (**DPL**) : montrent la disposition physique du matériel qui compose le système et la répartition des composants sur ce matériel.
- Diagramme de paquetages (**DPA**) : donne un vue d'ensemble du système structuré en paquetage. Chaque paquetage représente un ensemble homogène d'éléments du système (classes, composants, ...).
- Diagramme de structures composites (**DSC**) : décrit la structure interne d'un ensemble complexe composé par un exemple de classes ou d'objets et de composants techniques.

 **Les diagrammes de comportement** : représentent la partie *dynamique* d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs.

- Diagramme des cas d'utilisation (**DCU**) : identifient les utilisateurs du système (acteurs) et leurs interactions avec le système.
- Diagramme d'état-transitions (**DET**) : représente les différents états des objets en réaction aux évènements.
- Diagramme d'activités (**DAC**) : représente le comportement d'une méthode ou le déroulement d'un cas d'utilisation.
- Diagramme de séquences (**DSE**) : décrit les scénarios de chaque cas d'utilisation en mettant l'accent sur la **chronologie** des opérations en interaction avec les objets.
- Diagramme de communications (**DCO**) : ce diagramme est une autre représentation des scénarios des cas d'utilisations qui met plus l'accent sur les **objets** et les **messages** échangés.
- Diagramme global d'interactions (**DGI**) : fournit une vue générale des interactions décrites dans le diagramme de séquence et des flots de contrôle décrits dans le diagramme d'activités.
- Diagramme de temps (**DTP**) : ce diagramme permet de représenter les états et les interactions d'objets dans un contexte où le temps a une forte influence sur le comportement du système à gérer.

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

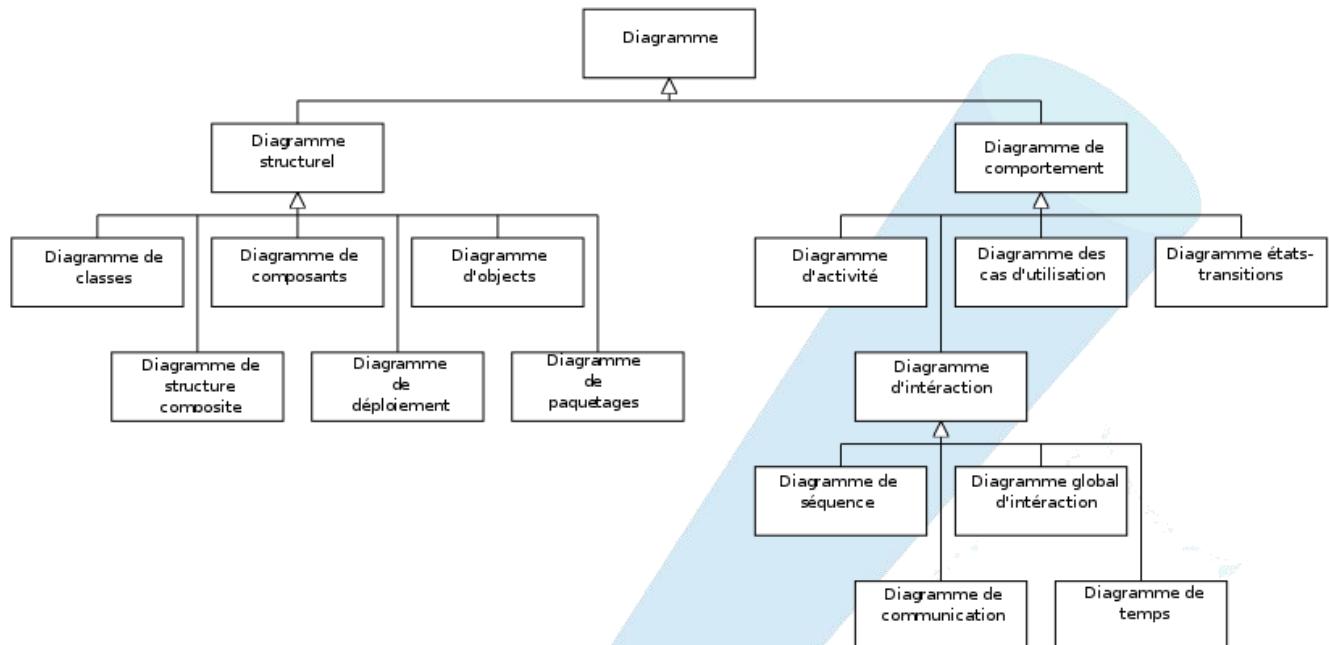


Figure 3 : Les différents diagrammes en UML

CHOIX DE L'OUTIL DE MODELISATION

Voici un tableau qui montre la différence entre les outils de modélisation et qui prouve notre choix :

Nom	Plateforme / OS	Dernière diffusion stable	Open source	Licence logiciel	Langage de programmation utilisé
Modelio	Windows, linux, macOS	24 février 2020	OUI	Commercial avec une édition communautaire gratuite (GPL V3, Apache 2.0)	Java
StarUML	Windows, linux, macOS	19 novembre 2020	OUI	Licence de démo permanente sans limitation, Commercial	ElectronJS
PragmaDev Studio	Windows, linux, mac	28 novembre 2016	OUI	Freeware	C#

Tableau 2 : Comparaison des différents outils de modélisation

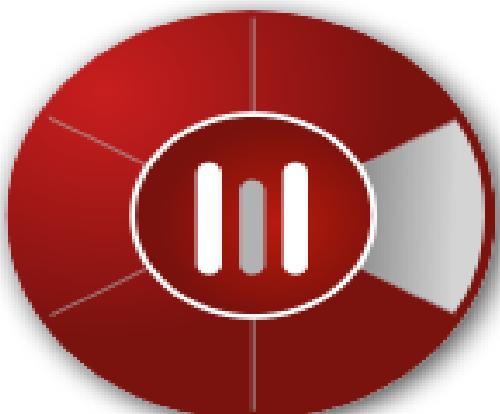


Figure 4 : Logo de Modelio



Figure 5 : Logo de StarUML



Figure 6 : Logo de PragmaDev Studio

Après analyse de ce tableau, nous pensons que Modelio est le logiciel de modélisation le plus optimal déjà à travers son Language de programmation utilisé qui est le "Java", orienté objet avec une grande tendance dans le monde du développement. Ce dernier est de type Open Source (accessible au public) : n'importe qui peut voir, modifier et distribuer le code à sa convenance. Ce type de logiciel est développé de manière collaborative et décentralisée.

II. DIAGRAMME DE CAS D'UTILISATION

II.1 DESCRIPTION DES PRINCIPAUX ACTEURS

Notre application présente trois (3) principaux acteurs, qui sont :

- Le client
- Le restaurateur
- Le livreur

Chaque acteur effectue des tâches bien précis :

- ❖ Le client : devra être en mesure de :
 - s'inscrire ou se connecter avant la validation de sa commande pour la première fois ;
 - consulter la liste des plats du menu d'un restaurant choisi et le ou les plats du jour de ce dernier ;
 - commander le plat de son choix ;
 - suivre l'évolution de sa commande ;
 - pouvoir se faire livrer s'il le désire ;
 - voir le livreur en temps réel ;
 - ajouter une description au menu ;
 - payer en ligne (*mobil money*) ou par cash ;
 - voir la facture de la commande.
- ❖ Le restaurateur : sera capable d'effectuer les actions suivantes :
 - s'inscrire ;
 - mettre un code d'identification personnel du restaurateur ayant gérer la commande du client X en cas de réclamation ;
 - pouvoir ajouter, rechercher, lister, supprimer ou modifier un repas ;
 - recevoir des notifications de commande ;
 - pouvoir envoyer une notification aux livreurs ;
 - Pour plus de sécurité et de fiabilité, le restaurateur recevra une notification après le paiement par cash du client validé par le livreur à son niveau ;
 - pouvoir suivre le livreur en temps réel.
- ❖ Le livreur :
 - s'inscrire ;
 - activer et désactiver son compte ;
 - recevoir des notifications sur les détails de la livraison ;
 - pouvoir scanner le QR-code de la facture pour voir si le client a payé ou non.

NB :

- ❖ Après la session expirée, tout accès aux fonctionnalités de l'application par n'importe quels acteurs du système (Restaurateur, livreur) nécessite une authentification en renseignant un code.

II.2 IDENTIFICATION DES DIFFERENTS DIAGRAMMES

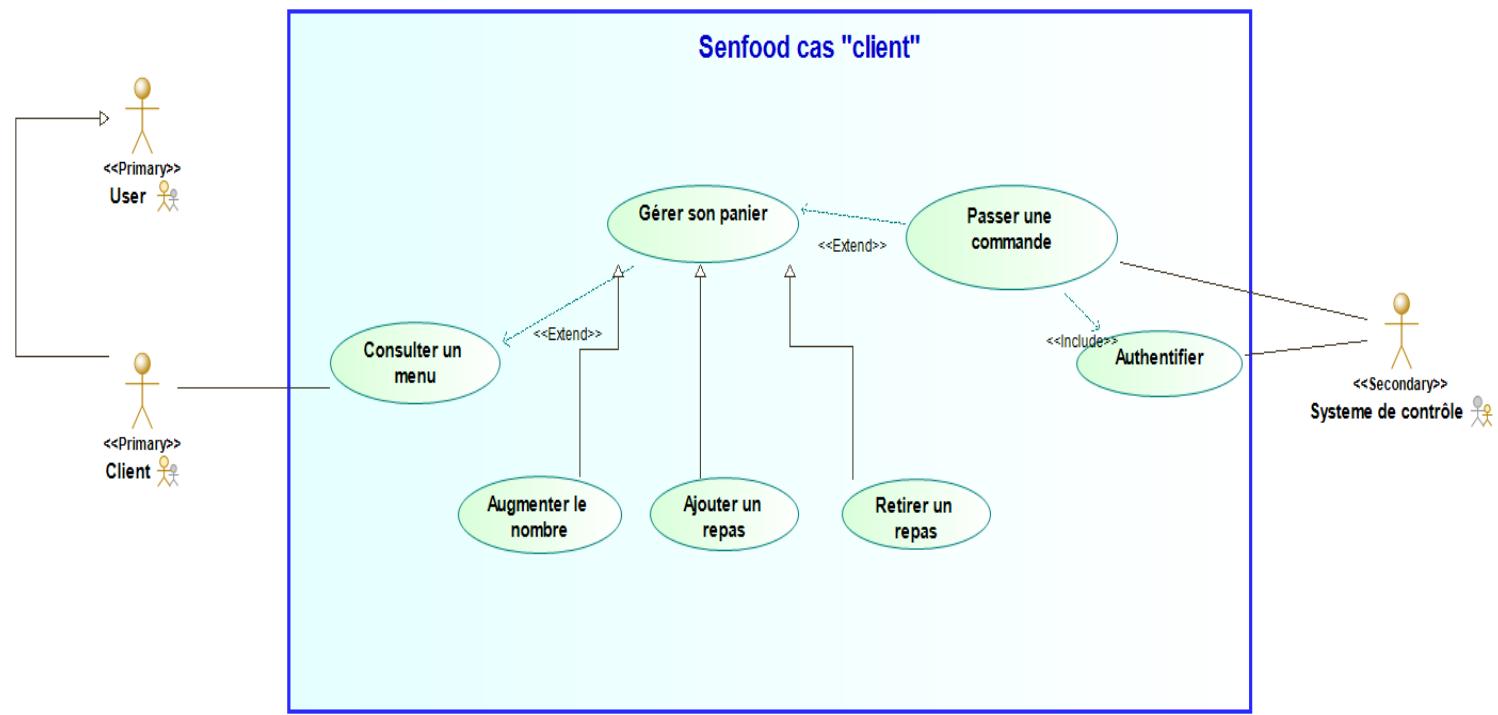


Figure 7 : Le diagramme de cas d'utilisation du client

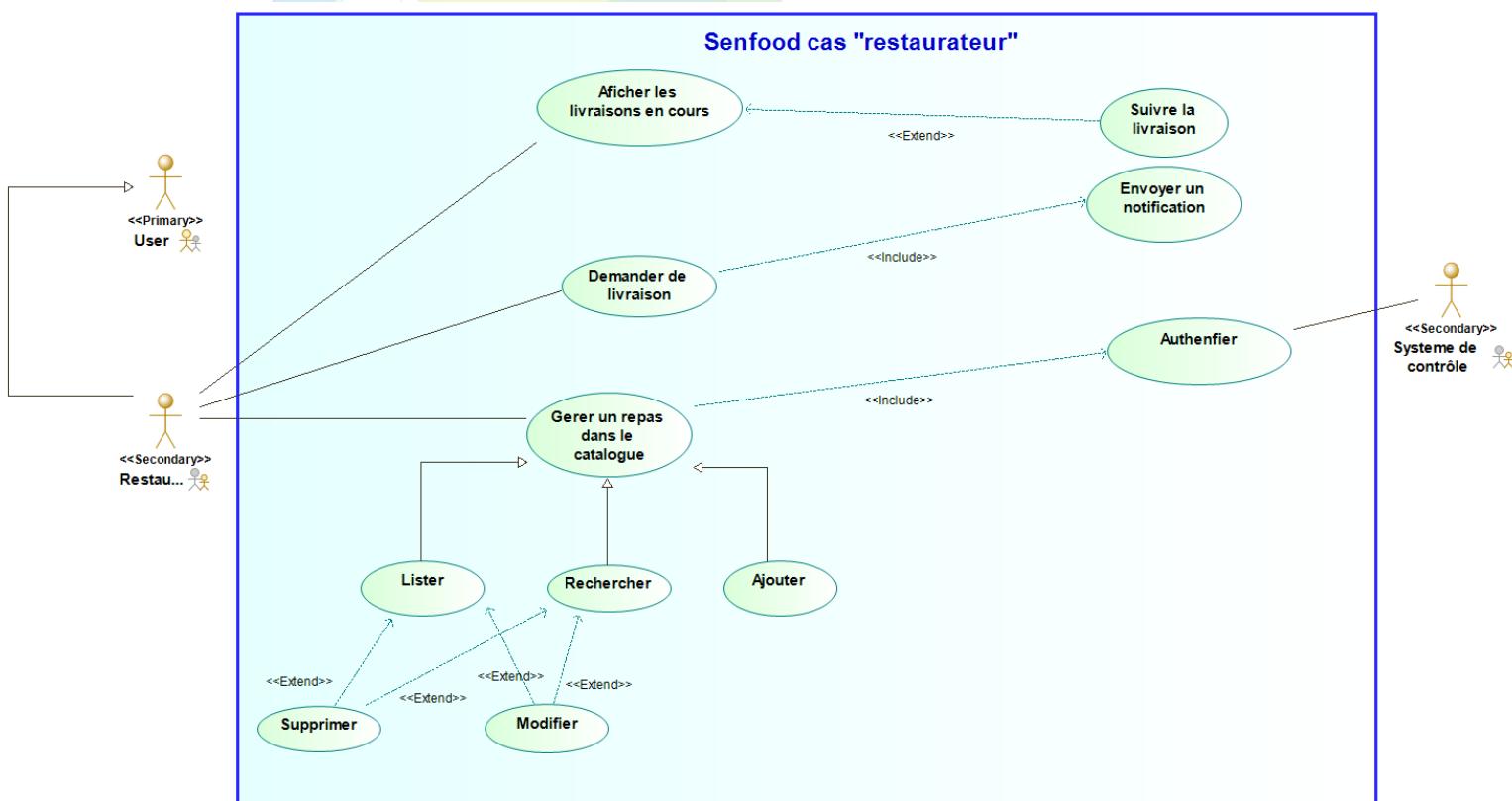


Figure 8 : Le diagramme de cas d'utilisation du restaurateur

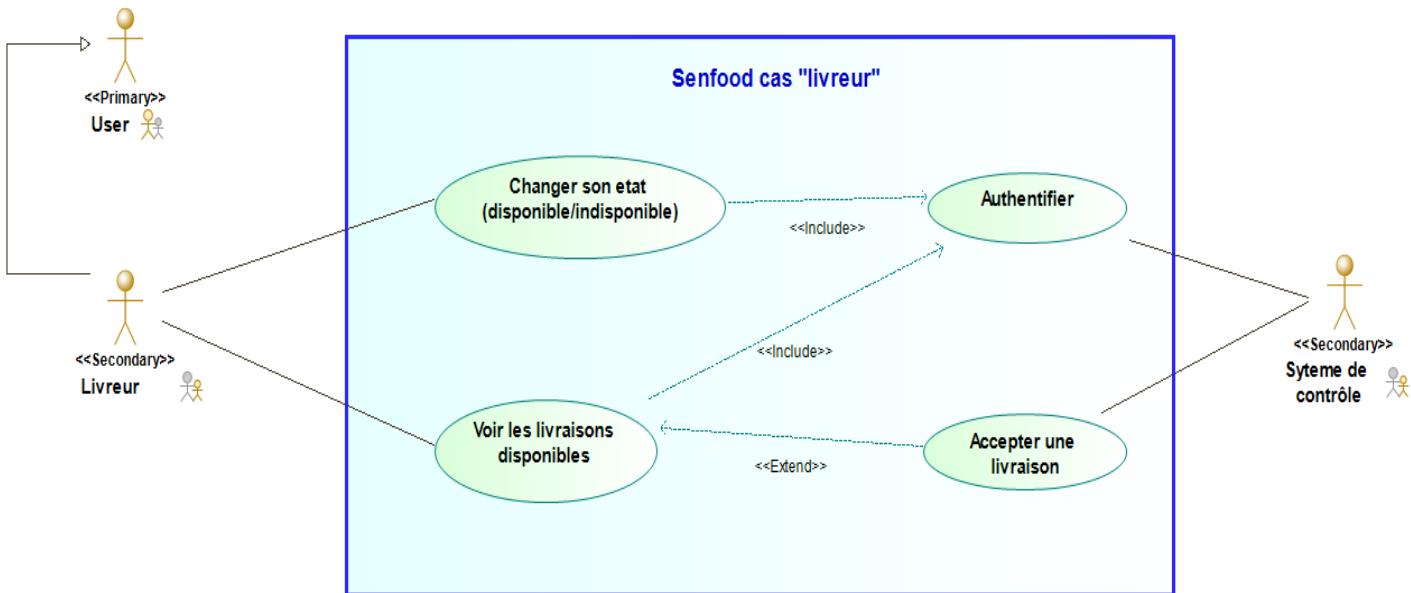


Figure 9 : Le diagramme de cas d'utilisation du livreur

III. MODELE CONCEPTUEL DE DONNEES

Le modèle conceptuel des données (**MCD**) a pour but de constituer l'ensemble des données qui seront utilisées par le système d'information en vue d'élaborer le diagramme de classes. Il s'agit donc d'établir une représentation des données et définir les dépendances fonctionnelles entre celles-ci, facilement compréhensible, permettant de décrire le système d'information à l'aide des concepts proposés par le modèle UML.

III.1 REPRESENTATION DES CLASSES

La modélisation objet consiste à créer une représentation abstraite, sous forme d'objets, d'entités ayant une existence matérielle ou bien virtuelle. Elle est utilisée en UML pour définir les objets-métiers et l'architecture d'une l'application.

La modélisation objet définit le comportement requis par les différentes classes pour assurer la bonne mise en place des cas d'utilisation et des règles de gestion.

Les objets constituent la base de l'architecture des applications, ils peuvent être réutilisés à travers des domaines d'application ou encore être identifiés et dérivés directement de cas d'utilisation ou des domaines d'application. Une classe est composée de :

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

- ✓ **Attributs** : représentant des données dont les valeurs représentent l'état de l'objet.
- ✓ **Méthodes** : il s'agit des opérations applicables aux objets.

Après avoir dégagé le dictionnaire de données épuré, nous pouvons dégager les classes ainsi que leurs méthodes et leurs attributs qui sont présentés dans le tableau suivant :

N°	Nom de la classe	Liste des attributs	Méthodes
1	User	<i>id_utilisateur Prenom nom adresse numero email mot_de_passe sexe</i>	
2	Livreur		
3	Restaurateur		
4	Client		
5	Livraison	<i>id_livraison date_livraison adresse</i>	
6	Restaurant	<i>id_restaurant nom lieu</i>	
7	Repas	<i>id_repas nom_repas description</i>	
8	Categorie	<i>Id_categorie nom_categorie</i>	
9	Commande	<i>Id_commande numero date_commande prix quantite</i>	
10	Facture	<i>id</i>	

Tableau 3 : Liste des classes

III.2 REPRESENTATION DES ASSOCIATIONS ENTRE CLASSES

Les associations sont des relations entre classes. Elles représentent un lieu durable ou ponctuel entre deux objets, une appartenance, ou une collaboration. Le modèle de données d'UML comprend trois associations génériques principales : généralisation, association et composition.

A partir de ces trois associations de base, nous représentons ainsi les différents types d'association qui décrivent les dépendances entre les classes déjà citées.

Association simple : les associations simples sont des liaisons logiques entre entités.

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

Les cardinalités : précisent combien d'objets de classe considérée peuvent être liés à un objet de l'autre classe.

Le tableau suivant illustre une représentation des différents types de cardinalités :

Cardinalité	Désignation
1	<i>Un et un seul</i>
01	<i>Zéro ou un</i>
N	<i>Plusieurs</i>
m..n	<i>De m à n (m et n deux entiers naturels différents)</i>
0..*	<i>Zéro ou plusieurs</i>
1..*	<i>Un ou plusieurs</i>

Tableau 4 : Représentation des cardinalités

Le tableau suivant illustre les associations simples en indiquant leurs désignations, les classes participantes et leurs cardinalités.

N°	Désignation	Classes participantes	Cardinalités
1	<i>Livrer</i>	<i>Livreur</i> <i>Livraison</i>	<i>1</i> <i>0..*</i>
2	<i>Livrer</i>	<i>Livraison</i> <i>Repas</i>	<i>1</i> <i>1..*</i>
3	<i>Appartenir</i>	<i>Restaurateur</i> <i>Restaurant</i>	<i>1</i> <i>1..*</i>
4	<i>Appartenir</i>	<i>Restaurant</i> <i>Repas</i>	<i>1</i> <i>1..*</i>
5	<i>Appartenir</i>	<i>Repas</i> <i>Categorie</i>	<i>1..*</i> <i>1</i>
6	<i>Faire</i>	<i>Client</i> <i>Commande</i>	<i>1</i> <i>*</i>
7	<i>Avoir</i>	<i>Meal</i> <i>Commande</i>	<i>1..*</i> <i>*</i>
8	<i>Appartenir</i>	<i>Commande</i> <i>Facture</i>	<i>1</i> <i>1</i>

Tableau 5 : Représentation des associations simples

Généralisation/spécialisation : C'est une relation d'héritage, dans laquelle les objets de l'élément spécialisé peuvent remplacer les objets de l'élément général.

III.3 REPRESENTATION DU DIAGRAMME DE CLASSE

Le diagramme de classe représente la description statique du système à développer en intégrant dans chaque classe la partie dédiée aux données et celle consacrée au traitement.

C'est un diagramme pivot de l'ensemble de la modélisation d'un système, cette représentation est concentrée sur le concept de classe et d'associations, les traitements sont matérialisés par des opérations.

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

Une classe est une description abstraite d'un ensemble d'objet ayant des propriétés similaires, un comportement commun et des relations communes avec d'autres objets.

La figure ci-dessous récapitule les tableaux précédents dans un diagramme de classes qui contient toutes les informations telles que les classes, les méthodes, les associations et les propriétés.

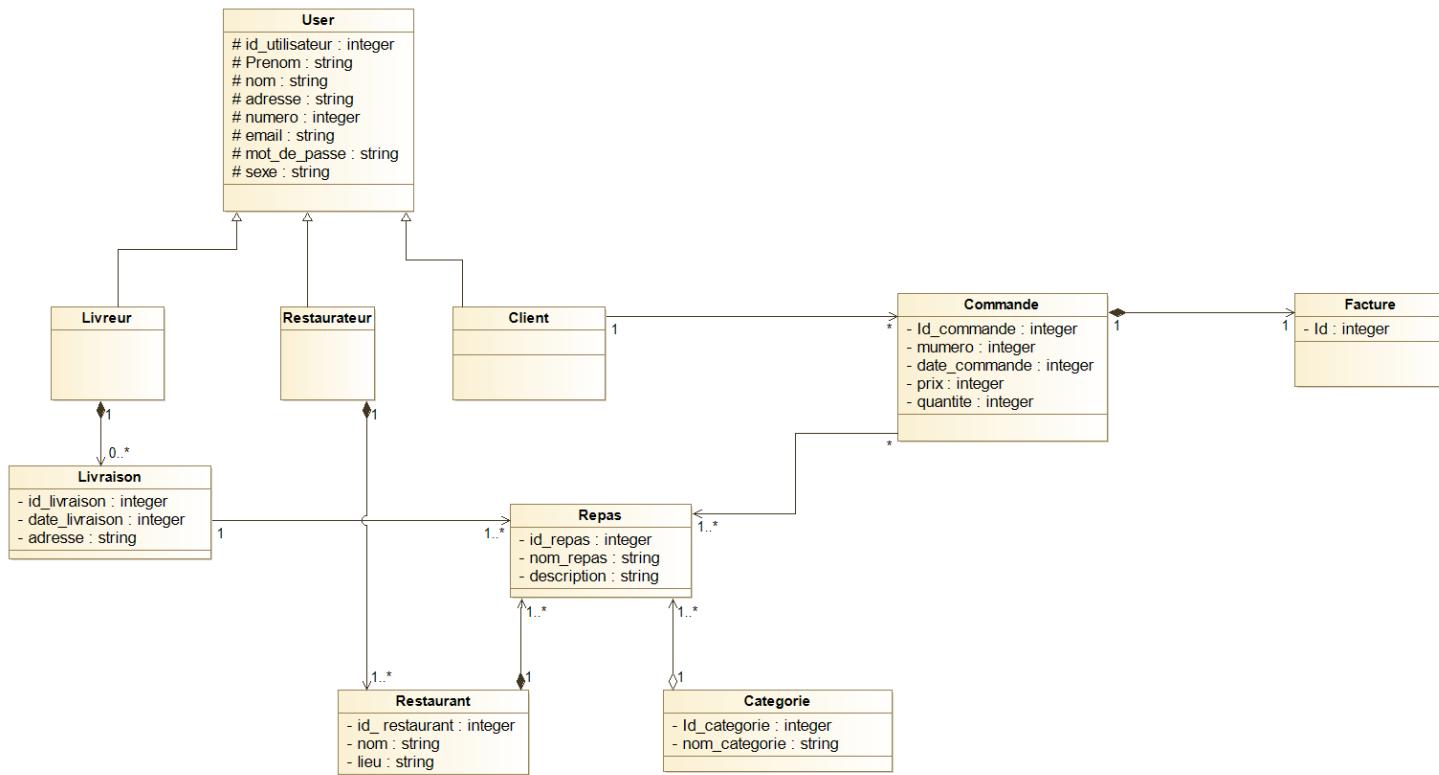


Figure 10 : Le diagramme de classe du système

CHAPITRE II : MODELISATION ORGANISATIONNELLE

Dans la section précédente nous avons proposé une modélisation conceptuelle des données et des traitements en se basant sur l'approche objet UML qui représente l'état de l'art des langages de modélisation objet, il permet de modéliser la structure et le comportement d'un système indépendamment de toute méthode ou langage de programmation.

La fiabilité de cette conception est notre porte vers un modèle organisationnelle de données efficace et confiant. Ainsi, dans cette section, nous présentons le modèle organisationnel de notre application.

I. DIAGRAMME D'ACTIVITE

Le diagramme d'activité offre la possibilité de visionner graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation. Il est donc particulièrement adapté à la modélisation du cheminement de flots de contrôle (enchaînement des actions ou des traitements qui modifient l'état du système) et de flots de données (toutes les définitions de variables, les utilisations).

Une activité représente le comportement d'une partie du système en termes d'actions et de transitions. Le passage d'une activité vers une autre est matérialisé par une transition. Les transitions sont déclenchées par la fin d'une activité et provoquent le début immédiat d'une autre.

Dans la suite, nous présentons notre diagramme d'activité du processus métier ‘Passer une commande’.

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

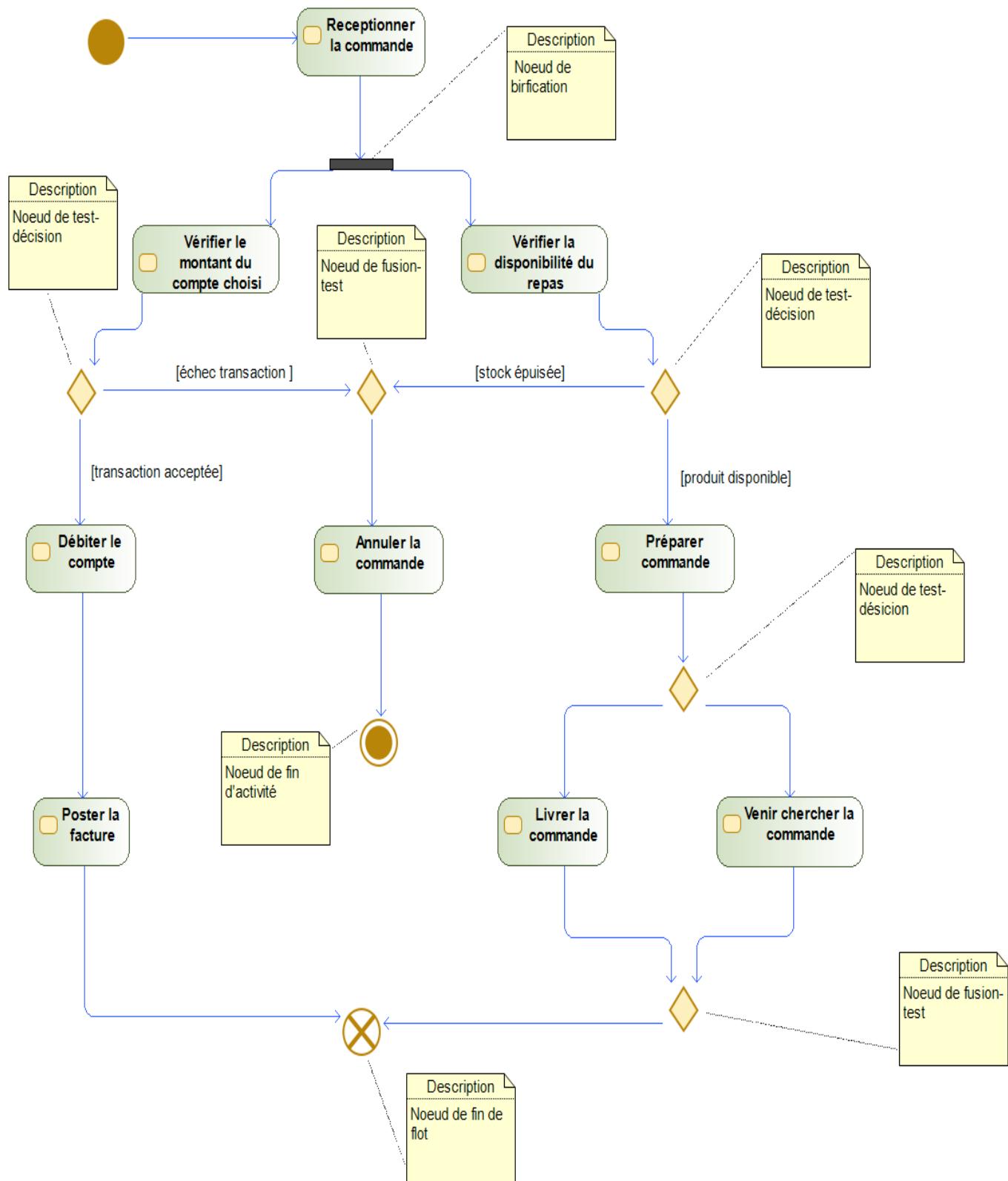


Figure 11 : Le diagramme d'activit茅

II. DIAGRAMME DE SEQUENCE

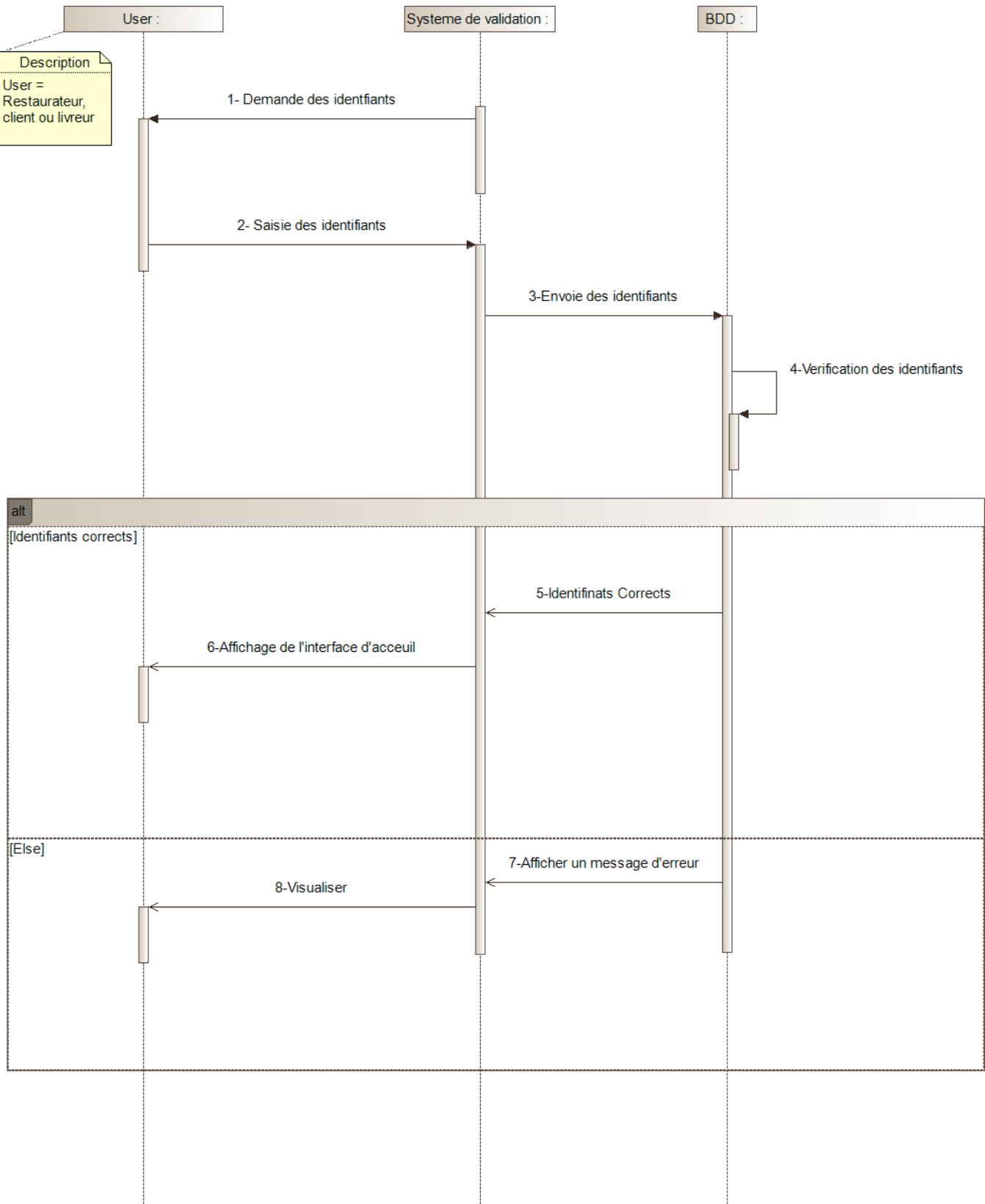


Figure 12 : Le diagramme de séquence

TROISIEME PARTIE : IMPLEMENTATION DE L'APPLICATION

- I. ARCHITECTURE DE L'APPLICATION**
- II. ENVIRONNEMENT DE TRAVAIL**
- III. INTERFACES DE L'APPLICATION**

TROISIEME PARTIE : INPLEMENTATION DE L'APPLICATION

I. ARCHITECTURE DE L'APPLICATION

Ce patron de conception est utilisé pour séparer les différentes couches de l'application.

- ✓ Modèle : Le modèle représente un objet portant des données.
- ✓ Vue : La vue représente la visualisation des données contenues dans le modèle.
- ✓ Contrôleur : Le contrôleur agit à la fois sur le modèle et sur la vue.

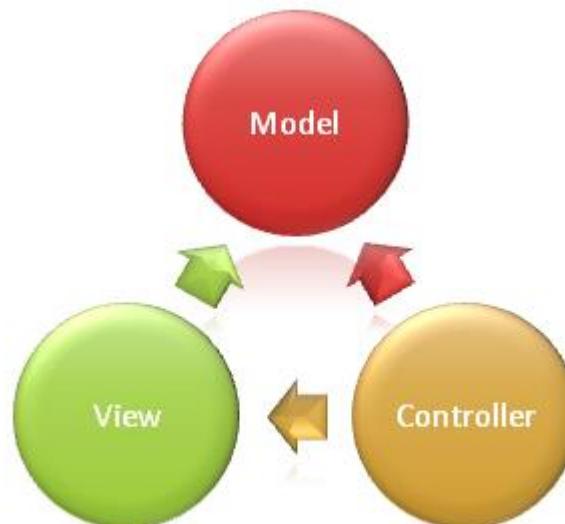


Figure 13 : Architecture MVC

II. ENVIRONNEMENT DE TRAVAIL

Pour la réalisation de ce travail, nous avons eu recours aux environnements suivants :

II.1 ENVIRONNEMENT MATERIEL

Pour développer cette application, nous avons utilisé un ordinateur HP et deux téléphones (Samsung et iPhone) pour les simulations et tests.

Processeur	Intel(R) Core (TM) i3-7020U CPU
RAM	8 Go
Disque dur	214 Go
Système d'exploitation	Microsoft Windows 10

Tableau 6 : Caractéristique de mon pc

II.2 ENVIRONNEMENT LOGICIEL

II.2.1 CHOIX DES TECHNOLOGIES DE DEVELOPPEMENT

II.2.1.1 LES FRAMEWORKS

A. GENERALITE SUR LES FRAMEWORKS

Un framework est une infrastructure de développement, c'est-à-dire un ensemble de composants logiciels pour prendre en charge le développement logiciel. Il s'agit d'une bibliothèque de logiciels qui fournit une structure fondamentale pour prendre en charge le développement d'applications pour un environnement spécifique. Son apparition remonte à juillet 1984 avec le logiciel framework de Ashton-Tate. Il n'était tout autre qu'un ensemble de programmes informatiques dédié spécialement pour les opérations basiques du travail de bureau à savoir la rédaction des lettres, des rapports, comptabiliser des factures, etc.

En développement, l'utilisation d'un framework permet donc de gagner du temps et aujourd'hui, il s'agit d'un standard dans la construction d'un projet web ou mobile. C'est pour cette raison qu'on a pour habitude de les comparer à une boîte à outils.

L'aspect innovateur d'un Framework se trouve dans la structure de ses fichiers. Un composant framework peut contenir une structure arborescente comme une table de matières où prend place les éléments provenant de tous les modules de la suite bureautique. Ainsi, il peut réunir divers éléments relatifs à un même sujet mais de types différents dans un même fichier.

De nos jours, la bureautique n'est pas le seul domaine où l'on note la présence des frameworks. L'univers du développement web et mobile, en est composé d'une multitude spécifique pour chaque langage de programmation. Ceci pour simplifier et booster le domaine du développement d'application. On note à cet effet des frameworks Front-end comme Vue JS, Bootstrap, etc., et Back-end tels que Flask ou encore Spring Boot. La contenance d'un framework est caractérisée par le type de programme et l'architecture cible pour lequel il est conçu.

B. DIFFERENTS TYPES DE FRAMEWORKS

En fonction des besoins et des projets, il existe différents types de frameworks dédiés aux développeurs mobiles, qu'ils soient débutants ou professionnels :

- ❖ Framework d'infrastructure système : framework qui permet le développement des systèmes d'exploitation (OS) et des interfaces graphiques ;

- ❖ Framework d'intégration intergicielle : framework pour fédérer des applications hétérogènes. Pour mettre à disposition différentes technologies sous la forme d'une interface unique ;
- ❖ Framework d'entreprise : framework pour développer des applications spécifiques au secteur d'activité de l'entreprise ;
- ❖ Framework orienté système de gestion de contenu : framework qui a pour mission de créer, collecter, classer, stocker et publier des informations numérisées.

Ces frameworks sont devenus incontournables dans le monde du développement. La maîtrise de l'un d'eux représente donc un véritable atout pour un développeur.

C. QUELQUES EXEMPLES DE FRAMEWORKS

➤ Ionic

C'est un framework créé en 2013 par Max Lynch, Ben Sperry, et Adam Bradley. C'est un open-source complet pour le développement de l'application hybride. Construit sur AngularJS et Cordova, Ionic fournit des outils et des services pour le développement d'applications mobiles hybrides utilisant les technologies web comme le HTML5, CSS. Les applications peuvent être construites avec ces technologies web et ensuite distribuées par le biais des boutiques d'applications natives pour être installées sur les appareils mobiles.



Figure 14 : Logo de Ionic

➤ Angular

Développé en 2016 par Google, Angular est un framework côté client open source basé sur TypeScript. Le TypeScript est une couche supérieure au Javascript développée par Microsoft qui se compile en JavaScript simple. Étant un langage typé, il permet de créer des classes, des variables, des signatures de fonction et l'utilisation de modules. Il est important de noter que l'utilisation du TypeScript est facultative, on peut tout à fait utiliser du JavaScript dans un fichier TypeScript. Il utilise l'architecture MVM (Modèle Vue Modèle), proche du modèle MVC. Cela va permettre de structurer son code et bien séparer la vue (l'interface) des modèles (fonctionnement) avec ce que l'on appelle les composants.

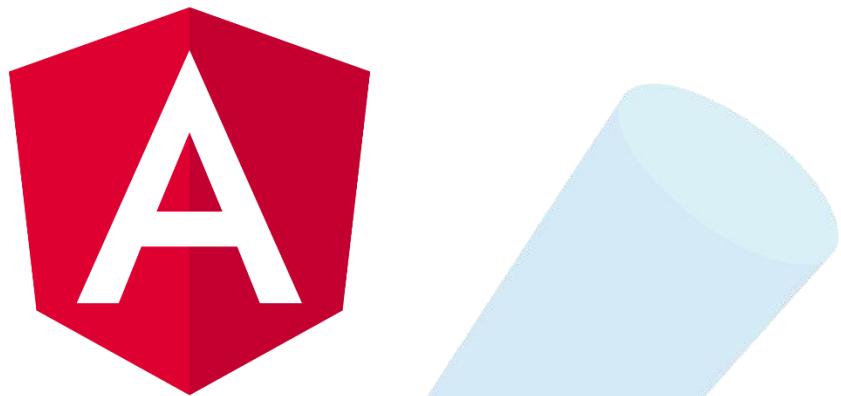


Figure 15 : Logo de Angular

➤ **React Native**

React Native est un framework d'applications mobiles open source créée par Facebook. Il est utilisé pour développer des applications pour Android, iOS et Universal Windows Platform (UWP) en permettant aux développeurs d'utiliser React avec les fonctionnalités natives de ces plateformes.

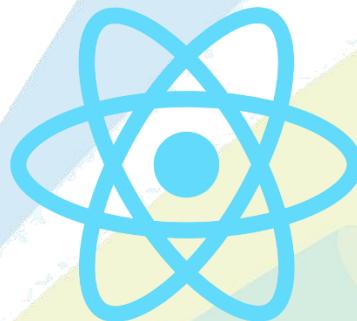


Figure 16 : Logo de React Native

❖ **Futter**

Flutter est un framework gratuit et open source de développement natif multiplateforme créé par Google et prend en charge le mobile (Android & iOS), le desktop (Windows, Mac OS, Linux etc.) ainsi que le web. Il est encore jeune mais, fait déjà ses preuves en séduisant certains grands noms de la tech comme Alibaba, le géant du e-commerce chinois qui l'a utilisé pour son application Xianyu. Il fournit un support de rendu de bas niveau en utilisant la bibliothèque graphique Skia de Google. De plus, il s'interface avec des Kit de développement (**SDK**) spécifiques à la plate-forme, tels que ceux fournis par Android et iOS. Flutter utilise le langage de programmation Dart, également inventé par Google.



Figure 17 : Logo de Flutter

➤ **Symfony**

C'est un framework qui représente un ensemble de composants (aussi appelés librairies) PHP autonomes qui peuvent être utilisés dans des projets web privé ou open source. Mais, c'est également un puissant Framework PHP développé par une société française : SensioLabs. Il permet de réaliser des sites internet dynamiques de manière rapide, structurée, et avec un développement clair. Les développeurs peuvent travailler sur ce framework très facilement, seul ou en équipe, grâce à la facilité de prise en main.



Figure 18 : Logo de Symfony

D. AVANTAGES ET INCOVENIENTS DE L'UTILISATOI DES FRAMEWORKS

– Avantages

- Gain de temps considérable : le framework apporte de nombreux outils qui permettent de ne pas réinventer la roue pour toutes les fonctionnalités courantes
- Meilleure structuration du code : le cadre et les directives fournis par les frameworks nous permettent de donner une structure et, par conséquent, d'améliorer la qualité du code. Ils permettent de construire une unité cohérente.
- Grande efficacité : grâce aux bibliothèques de fonctions réutilisables auxquelles nous pouvons avoir accès librement, ce qui nous évite d'avoir à programmer nous-même certains mécanismes.
- Communauté très présente : pour mettre à jour les frameworks et nous aider en cas de problème.
- Robustesse des applications : Prise en charge d'une grosse partie du code et des éléments techniques.

– Inconvénients

Bien que les frameworks possèdent de nombreux avantages, il est aussi important de présenter quelques-uns de leurs points faibles :

- Nécessité de modifier ou ajouter un code. Le framework n'a pas la réponse à tous les cas de figures. Et, il est quasiment certains que le développeur devra modifier ou créer des segments de code supplémentaires pour se rapprocher du design souhaité ;
- L'incertitude sur la capacité à maintenir un projet dans le temps puisque le code sera grandement dépendant de la solvabilité de ce framework ;
- Contrainte de poids puisque ses composants prennent une place importante dès le début du projet ;
- Gestion des nouvelles mises à jour.

II.2.1.2 ETUDE COMPARATIVE EN FLUTTER ET REACT NATIVE

Les frameworks mobiles multiplateformes aident les développeurs d'applications à obtenir de meilleurs résultats de développement. Un nombre croissant de développeurs optent désormais pour des technologies multiplateformes et les préfèrent aux technologies natives. React Native et Flutter sont deux des cadres multiplateformes les plus populaires aujourd'hui. Ces deux plates-formes offrent diverses fonctionnalités et de très bons avantages. Pour choisir l'option la plus appropriée, il faut tenir compte des différences et similitudes de Flutter et de React Native.

A. QU'EST-CE QU'UN FRAMEWORK MOBILE MULTIPLATFORME ?

Les frameworks de développement d'applications multiplateformes permettent de développer des applications mobiles qui fonctionnent sur plusieurs systèmes d'exploitation comme Android et iOS. Ce type de cadre permet aux utilisateurs d'écrire du code une fois et de l'exécuter sur plusieurs plates-formes. Une plateforme de développement d'applications multi-plateformes permet aux développeurs d'obtenir des versions de logiciels plus rapides et de meilleure qualité.

B. POURQUOI UTILISER UN FRAMEWORK MULTIPLATEFORME ?

- **Une mise sur le marché plus rapide :** Les applications hybrides ont un délai de mise sur le marché plus rapide. Les développeurs peuvent déployer des applications plus rapidement avec ce type de cadre.
 - **Réduction des coûts :** Les développeurs peuvent réaliser des économies plus importantes en utilisant un framework multiplateforme. La diminution des efforts et du temps nécessaires contribue également à réduire les coûts de développement.
 - **Une meilleure réutilisation des codes :** Le développement d'applications mobiles multiplateformes permet aux développeurs d'utiliser une base de code unique pour différentes applications et différents systèmes d'exploitation. L'utilisation de la même base de code rend plus pratique la maintenance et la révision d'une base de code.
- C. Une **expérience** proche de native : **Les caractéristiques du backend et les fonctionnalités des applications hybrides sont assez similaires.** Ainsi, vous pouvez accélérer le développement et garder le même UX pour les utilisateurs finaux.

D. FLUTTER CONTRE REACT NATIVE : SIMILITUDES

Nous allons comparer Flutter et React Native. Commençons par les similitudes des deux frameworks. Il s'agit dans les deux cas de frameworks multiplateformes :

- Open-Source
- Rechargement
- Expérience Native

E. FLUTTER CONTRE REACT NATIVE : COMPARAISON

Le tableau ci-dessous vous donne un aperçu des deux frameworks.

	<i>Flutter</i>	<i>React-Native</i>	<i>Remarques</i>
Soutenu par	Google	Facebook	–
Date de sortie	May 2017	March 2015	React-Native est un framework plus mature.
Langage de programmation	Dart	Javascript	Le JavaScript est un choix plus populaire.
Courbe d'apprentissage	Faible	Faible	React-Native est peut-être plus facile à apprendre si vous connaissez déjà Javascript.
Multiplateforme	Oui	Oui	–
Hot Reload	Oui	Oui	–
Open-Source	Oui	Oui	–
Documentation	Oui	Oui	–
Architecture	Skia	Flux	–

CI/CD	Documentation officielle	Non disponible sur les documents officiels	–
Référentiel Github	Flutter	React Native	–
Github Étoiles/Forks	102k / 14.2k	90k / 20k	–

Tableau 7 : Comparaison entre flutter et react native

F. QUEL EST LE CADRE MULTIPLATEFORME LE PLUS UTILISE ?

Pour répondre à cette question, nous allons comparer les données de Google Trends, Statista et Github.

- **Google Trends**

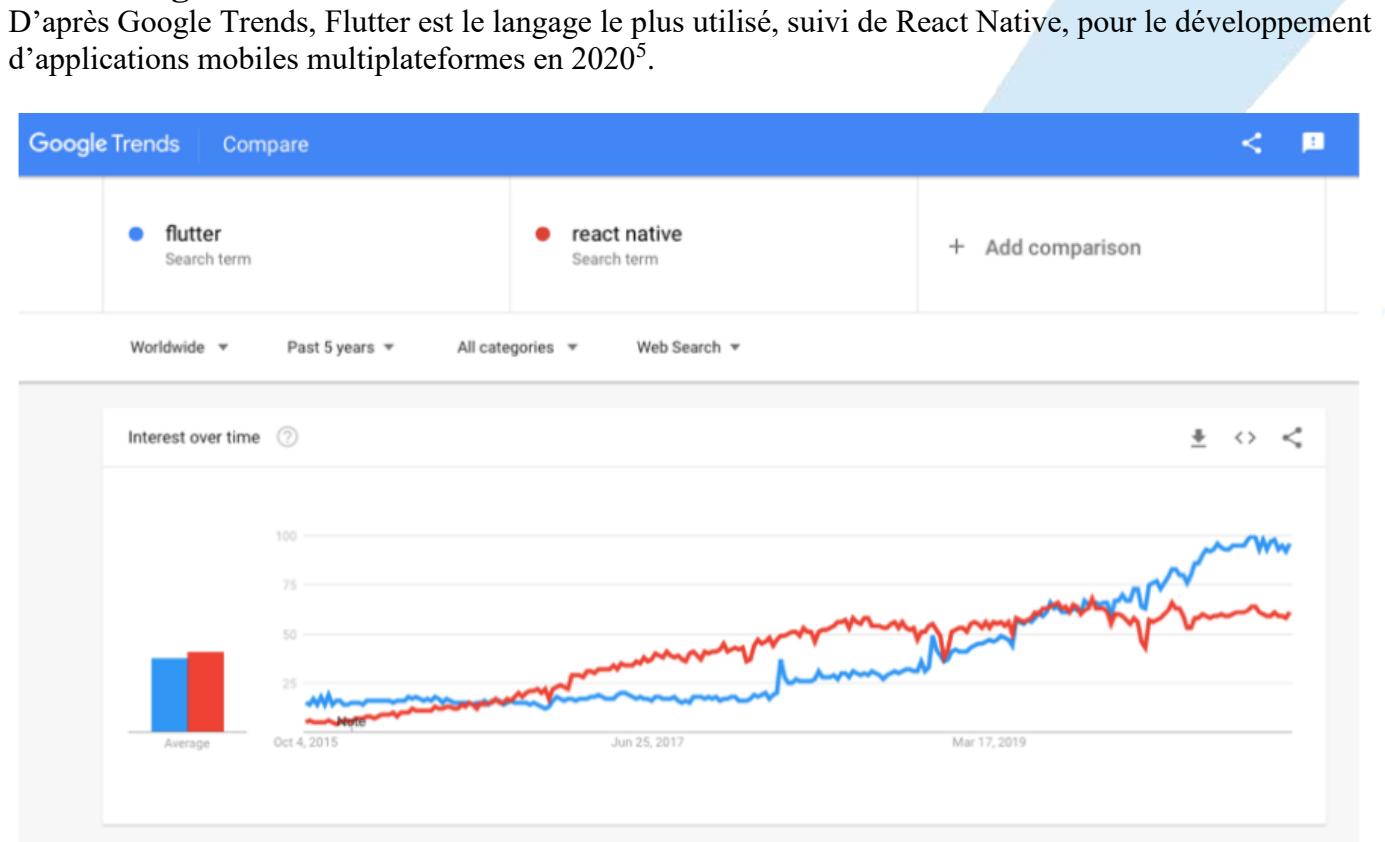


Figure 19: Google Trends : flutter & react native

- **Statista**

Selon Statista, React Native est toujours le framework le plus utilisé avec 42%. Flutter est assez proche avec 39% et a connu une augmentation plus forte entre 2019 et 2020⁶.

⁵ Voir webographie

⁶ Voir webographie

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

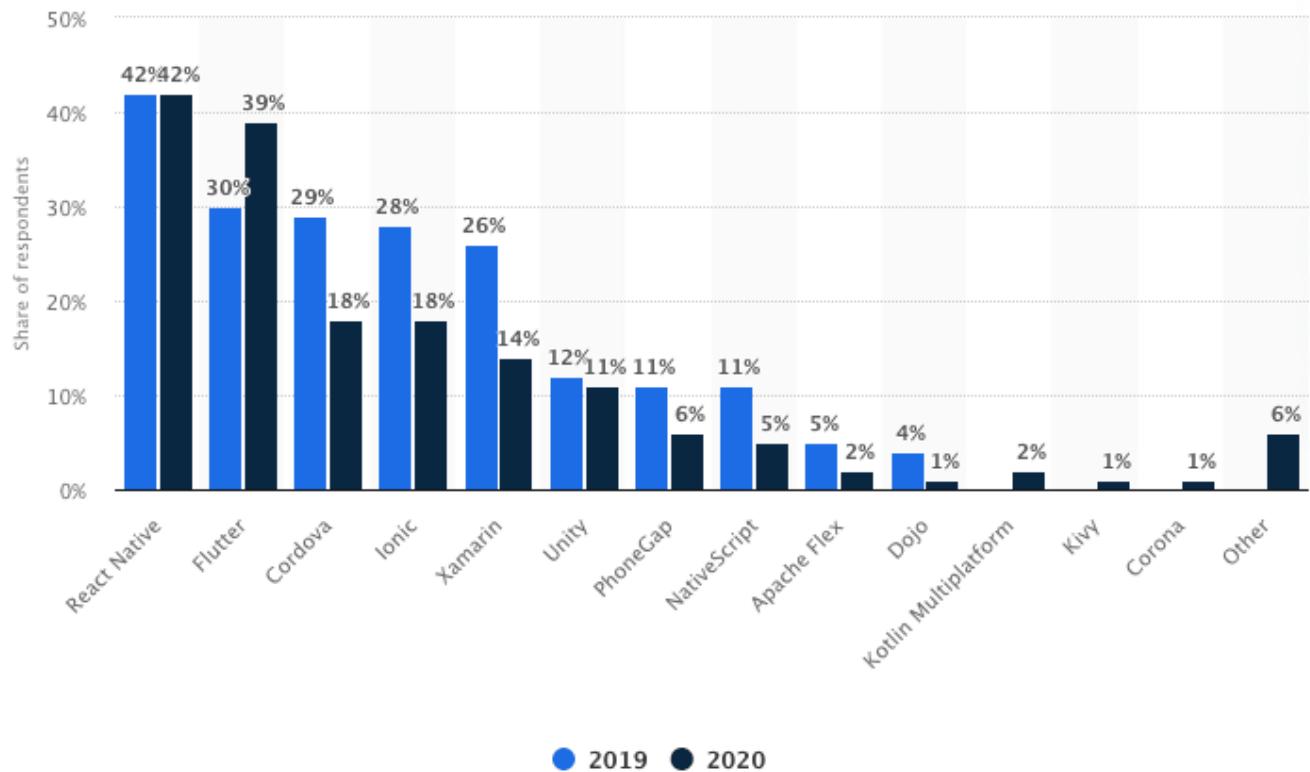


Figure 20 : Statistique entre flutter & react native

• Github

Une autre source essentielle pour comparer la popularité entre Flutter et React Native est GitHub. Flutter mène la course ici avec 102 000 étoiles, mais React Native est assez proche avec 90 000 étoiles. Si l'on considère que la sortie initiale de Flutter a eu lieu en 2017 (contre 2015 pour React Native), il semble que Flutter gagne en popularité plus rapidement selon les données de GitHub.⁷

Ainsi, en termes de popularité, les deux frameworks ont une percée similaire auprès des développeurs, et vous pourrez trouver un soutien communautaire adéquat sur n'importe laquelle de ces deux plateformes.

II.2.1.3 CHOIX DU FRAMEWORK

De nos jours, les utilisateurs de mobiles s'attendent à ce que leurs applications aient un design magnifique, des animations fluides et des performances exceptionnelles. Les développeurs doivent créer une nouvelle fonctionnalité plus rapidement que jamais, sans compromettre la qualité ni les performances.

C'est pourquoi on a choisi Flutter comme framework pour notre application mobile.

⁷ Voir le lien sur la webographie

A. PRESENTATION DU LANGAGE DART

Dart est un langage orienté Web open source, sous licence Berkeley Software Distribution (**BSD**), proposé par Google, très impliqué dans le développement Web. Il s'agit d'un langage orienté objet simple, facile à prendre en main. Il s'exécute dans une Machine Virtuelle (**VM**) aussi bien du côté client (sur le navigateur) que du côté serveur.

Si Google a opté pour Dart, c'est parce qu'il offre deux modes de fonctionnement :

- AOT (Ahead of Time) : fonction qui permet de générer une application native pour chaque plateforme. L'avantage de Flutter, par rapport à ses concurrents, est donc fort, puisque le code sera optimisé directement pour l'architecture sur laquelle il fonctionnera.
- JIT (Just-In-Time) : offre la fonctionnalité de Hot Reload qui permet de rapidement tester des interfaces utilisateurs, des fonctionnalités, corriger des bugs lors du débogage.



Figure 21 : Logo de Dart

B. PARTICULARITES DE FLUTTER

Les frameworks de développement multiplateforme comme Cordova ou Ionic utilisent le langage JavaScript et les WebViews. Les applications conçues avec ces outils sont basées sur les technologies web bien connues (HTML, CSS et JavaScript), mais nécessitent un lien pour se relier avec le code natif. A l'instar de React Native, Flutter fournit des vues réactives. Mais, grâce à l'utilisation d'un langage compilé comme Dart, plus besoin d'intermédiaire pour faire le lien avec le code natif c'est pourquoi Flutter est aujourd'hui le seul (**SDK**) Sotfware Development Kit mobile qui fournit des vues réactives sans code interprété.

Grâce à la fonctionnalité Hot Reload de Dart aussi, le bail des applications est très rapide, ce qui rend quasiment invisible le temps de compilation.

Ensuite, l'un des principes fondamentaux de Flutter est que tout est un widget. Vraiment tout. La classe de l'application est un widget, la structure de l'interface est un widget, le texte affiché est un widget, les images sont des widgets, les boutons, le padding, etc. Chaque widget a une tâche bien particulière et affecte une petite partie de l'**UI** (user interface). Créer une interface utilisateur en Flutter revient alors à assembler des widgets. Et c'est aussi simple que ça en a l'air. D'autant plus que le SDK contient une large variété de widgets intégrés, et déjà stylisés Android ou iOS.

Flutter est conçu comme un système extensible en couches. Il existe sous la forme d'une série de bibliothèques indépendantes qui dépendent chacune de la couche sous-jacente. Aucune couche n'a un accès privilégié à la couche inférieure, et chaque partie du niveau de structure est conçue pour être facultative et remplaçable.

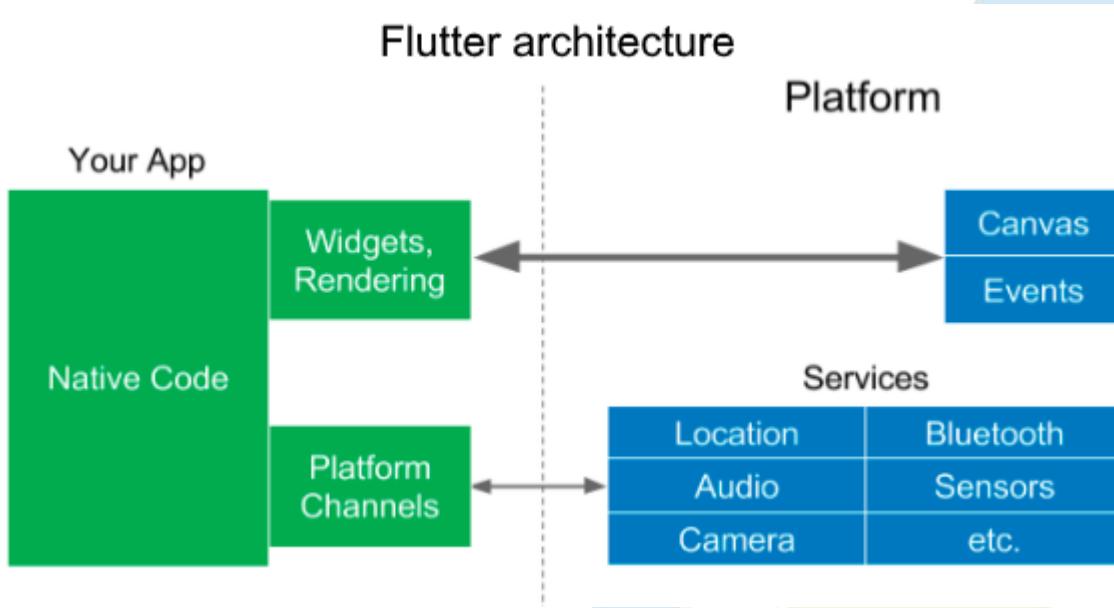


Figure 22 : Architecture de Flutter

C. EXEMPLES D'APPLICATIONS DEVELOPPEES AVEC FLUTTER

- Google Stadia :

C'est une plate-forme de jeux vidéo à la demande conçue par Google permettant de jouer à des jeux vidéo, sur toutes sortes d'appareils, comme un téléphone, un ordinateur.

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant



Figure 23 : Interface de l'application STADIA

➤ New York Times

C'est est un quotidien new-yorkais distribué internationalement et l'un des plus prestigieux journaux américains.



Figure 24 : Interface de l'application New York Times

➤ Alibaba

L'application officielle de la célèbre boutique en ligne Alibaba de Chine.

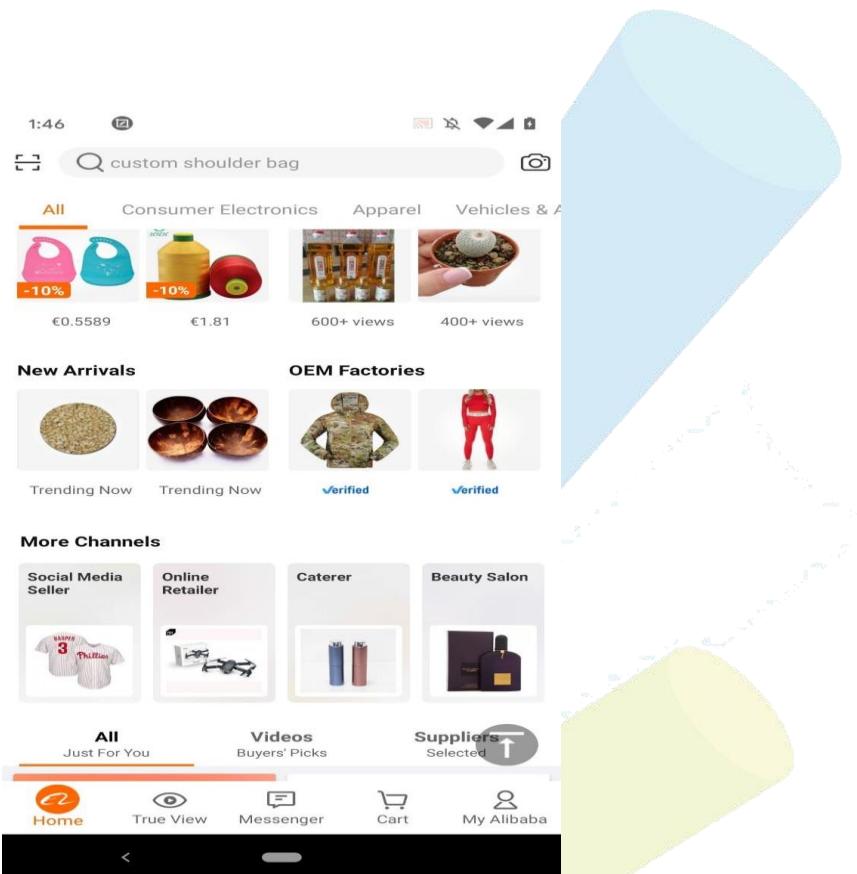


Figure 25 : Interface de l'application Alibaba

II.2.1.4 SGBD DE L'APPLICATION

Flutter propose des options de gestions de données à l'aide desquelles il est possible de traiter, stocker vos données localement ou dynamiquement. Pour mener à bien la bonne gestion des données de notre application, notre choix est porté sur Firebase qui est un ensemble de services d'hébergement.

Pour la gestion des données de notre application, nous utilisons les services Firestore et realtime database qui sont des solutions de base de données en temps réel, firebase authentification et firebase cloud storage.

A. PRESENTATION DE FIREBASE

Firebase est une plateforme mobile créée par Google qui permet de développer rapidement des applications. Elle propose un ensemble de services comme l'hébergement en NoSQL et en temps réel des bases de données, des services d'authentification sociale ou encore des services d'un serveur de communication en temps réel.



Figure 26 : Logo de Firebase

Certaines applications parmi lesquelles on retrouve les services fournis par Firebase sont :

- Pic Collage : un logiciel d'éditeur photo



Figure 27 : Interface de l'application Pic Collage

- Fabulous : un logiciel de coaching, méditation et relaxation



Figure 28: Interface de l'application Fabulous

- Shazam : un logiciel de reconnaissances musicales de chansons



Figure 29: Interface de l'application Shazam

B. PRESENTATION DES SERVICES

- CLOUD FIRESTORE

Cloud Firestore est un service de base de données en temps réel tout comme Realtime Database. Il permet de stocker et synchroniser les données. Cloud Firestore est une base de données flexible et évolutive pour le développement mobile, Web et serveur de Firebase et Google Cloud.

En suivant le modèle de données NoSQL de Cloud Firestore, il stocke les données dans des documents contenant des champs mappés sur des valeurs. Ces documents sont stockés dans des collections, qui sont des conteneurs pour nos documents que nous pouvons utiliser pour organiser nos données et créer des requêtes.

Nous avons quelques capacités du cloud Firestore :

- **La flexibilité** : Le modèle de données Cloud Firestore prend en charge des structures de données hiérarchiques flexibles. Stockez vos données dans des documents, organisés en collections. Les documents peuvent contenir des objets imbriqués complexes en plus des sous-collections.
- **Interrogation expressive** : Dans Cloud Firestore, vous pouvez utiliser des requêtes pour récupérer des documents individuels et spécifiques ou pour récupérer tous les documents d'une collection qui correspondent à vos paramètres de requête. Vos requêtes peuvent inclure plusieurs filtres chaînés et **combiner** filtrage et tri. Ils sont également indexés par défaut, de sorte que les performances des requêtes sont proportionnelles à la taille de votre ensemble de résultats, et non à votre ensemble de données.
- **Mise à jour en temps réel** : Comme Realtime Database, Cloud Firestore utilise la synchronisation des données pour mettre à jour les données sur n'importe quel appareil connecté. Cependant, il est également conçu pour effectuer efficacement des requêtes de récupération simples et ponctuelles.
- **Assistance hors ligne** : Cloud Firestore met en cache les données que votre application utilise activement, afin que l'application puisse écrire, lire, écouter et interroger des données même si l'appareil est hors ligne. Lorsque l'appareil revient en ligne, Cloud Firestore synchronise toutes les modifications locales vers Cloud Firestore.
- **Conçu à l'échelle** : Cloud Firestore vous offre le meilleur de la puissante infrastructure de Google Cloud : réplication automatique des données multirégionales, garanties de cohérence solides, opérations par lots atomiques et prise en charge des transactions réelles. Nous avons conçu Cloud Firestore pour gérer les charges de travail de base de données les plus difficiles des plus grandes applications du monde.

Nous avons porté notre choix sur ce service pour la particularité de sa capacité à gérer les données en temps réels.



Figure 30 : Logo de Cloud Firestore

– FIREBASE REALTIME DATABASE

Realtime Database est un service de base de données en temps réel tout comme son nom l'indique. Il permet de stocker et synchroniser les données. Il fournit un moyen sûr, fiable et rapide pour la synchronisation des données ; il garantit également la persistance des données en stockant les données localement.

Il utilise une base de données NoSQL pour le stockage des données ; elles sont stockées sous la forme d'un objet JSON qui est généralement composé d'une paire de clé/valeur. La clé identifie de façon unique l'objet dans la base de données et la valeur représente les données stockées.



Figure 31 : Logo de Cloud Firebase Realtime Database

– FIREBASE AUTHENTIFICATION

Voici quelques avantages que nous propose l'authentification avec firebase :

- Connexion facile avec n'importe quelle plate-forme**

Firebase Authentication vise à faciliter la création de systèmes d'authentification sécurisés, tout en améliorant l'expérience de connexion et d'intégration pour les utilisateurs finaux. Il fournit une solution d'identité de bout en bout, prenant en charge les comptes de messagerie et de mot de passe, l'authentification téléphonique, les connexions Google, Twitter, Facebook et GitHub, etc.

- Interface utilisateur flexible et prête à l'emploi**

FirebaseUI fournit une solution d'authentification personnalisable, open source, qui gère les flux d'interface utilisateur pour la connexion des utilisateurs. Le composant FirebaseUI Auth implémente les meilleures pratiques pour l'authentification sur les appareils mobiles et les sites Web, ce qui peut maximiser la conversion de connexion et d'inscription pour votre application.

- Sécurité complète**

Construit par la même équipe qui a développé Google Sign-in, Smart Lock et Chrome Password Manager, Firebase Security applique l'expertise interne de Google en matière de gestion de l'une des plus grandes bases de données de comptes au monde.

- Mise en œuvre rapide**

La configuration de votre propre système d'authentification peut prendre des mois et une équipe d'ingénieurs est nécessaire pour maintenir ce système à l'avenir. Configurez l'ensemble du système

d'authentification de votre application en moins de 10 lignes de code est possible même en gérant des cas complexes comme la fusion de comptes.



Figure 32: Logo de Cloud Firebase Authentication

– FIREBASE CLOUD STORAGE

- Stockez les photos et vidéos de vos utilisateurs**

Cloud Storage est conçu pour vous aider à stocker et à diffuser rapidement et facilement du contenu généré par les utilisateurs, comme des photos et des vidéos.

- Construire à l'échelle de Google**

Notre infrastructure est conçue pour passer sans effort du prototype à la production, lorsque notre application deviendra virale, en utilisant la même technologie qui alimente des applications comme Spotify et Google Photos.

- Téléchargements et téléchargements robustes**

Les utilisateurs ne sont pas toujours en ligne, c'est pourquoi nous avons conçu le SDK Firebase pour Cloud Storage en pensant à la connectivité mobile. Il mettra automatiquement en pause et reprendra les transferts au fur et à mesure que l'application perd et regagne la connectivité mobile, économisant ainsi du temps et de la bande passante pour les utilisateurs.

- Forte sécurité basée sur l'utilisateur**

Le SDK Firebase pour Cloud Storage s'intègre à Firebase Authentication pour fournir un contrôle d'accès simple et intuitif. Il est possible d'utiliser le modèle de sécurité déclaratif pour autoriser l'accès en fonction de l'identité de l'utilisateur ou des propriétés d'un fichier, tels que le nom, la taille, le type de contenu et d'autres métadonnées.



Figure 33 : Logo de Cloud Firebase Cloud Storage

II.2.2 OUTILS DE DEVELOPPEMENT

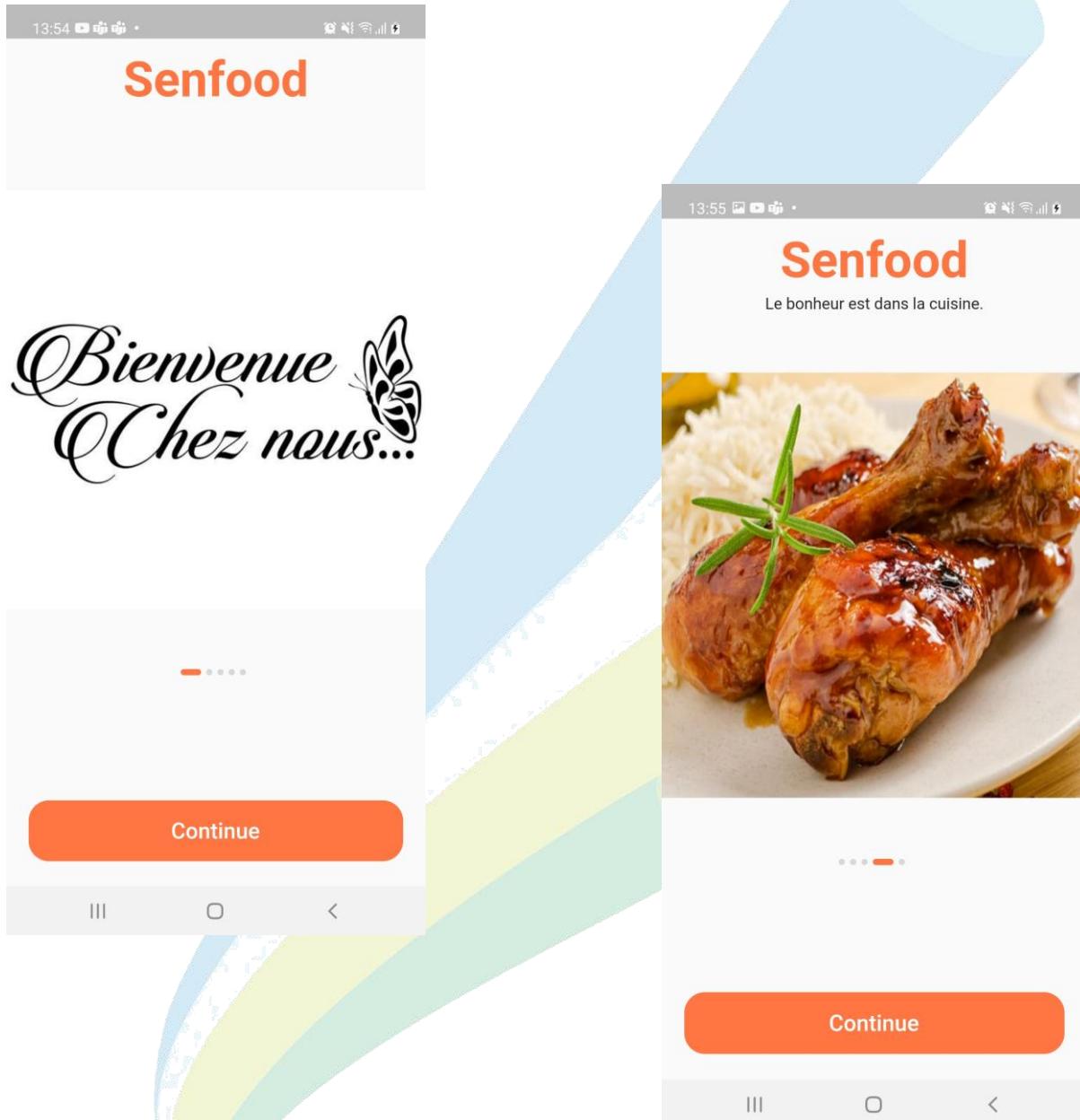
Comme outil de développement, nous avons opté pour Android Studio. Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA (IntelliJ IDEA également appelé « IntelliJ », « IDEA » ou « IDJ ») est un environnement de développement intégré destiné au développement de logiciels informatiques reposant sur la technologie Java.) et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux.

Avant Android Studio, de 2009 à 2014, Google propose comme environnement de développement officiel une distribution spécifique de l'environnement Eclipse, contenant notamment le SDK d'Android. Android Studio est annoncé le 15 mai 2013 lors du Google I/O et une version Early Access Preview est disponible le jour même. Le 8 décembre 2014, Android Studio passe de version bêta à version stable 1.0. L'environnement devient alors conseillé par Google, et Eclipse est délaissé.



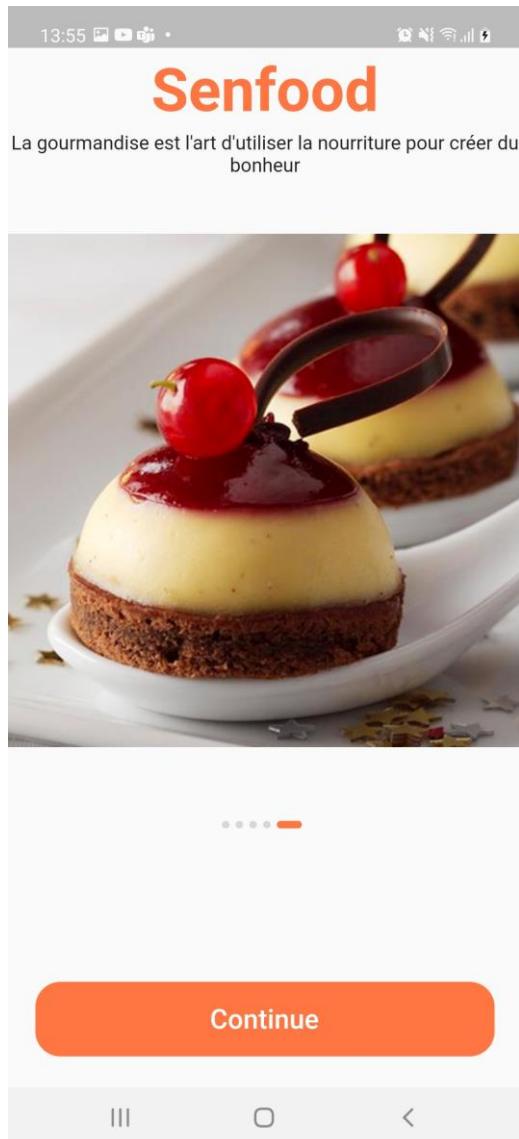
Figure 34 : Logo de Android Studio

III. INTERFACES DE L'APPLICATION



Capture 11 : Page de Bienvenue

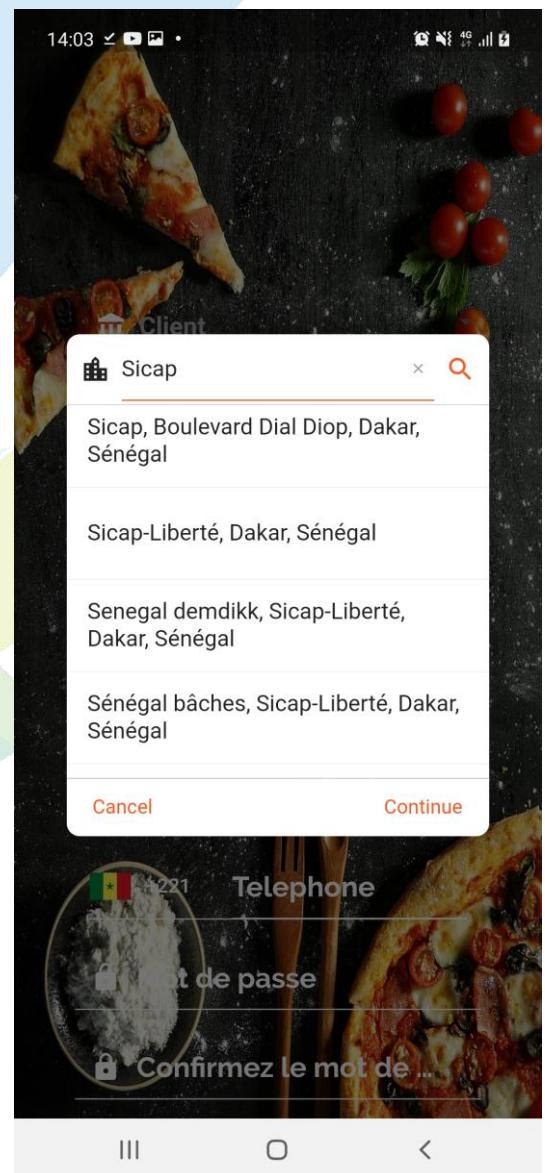
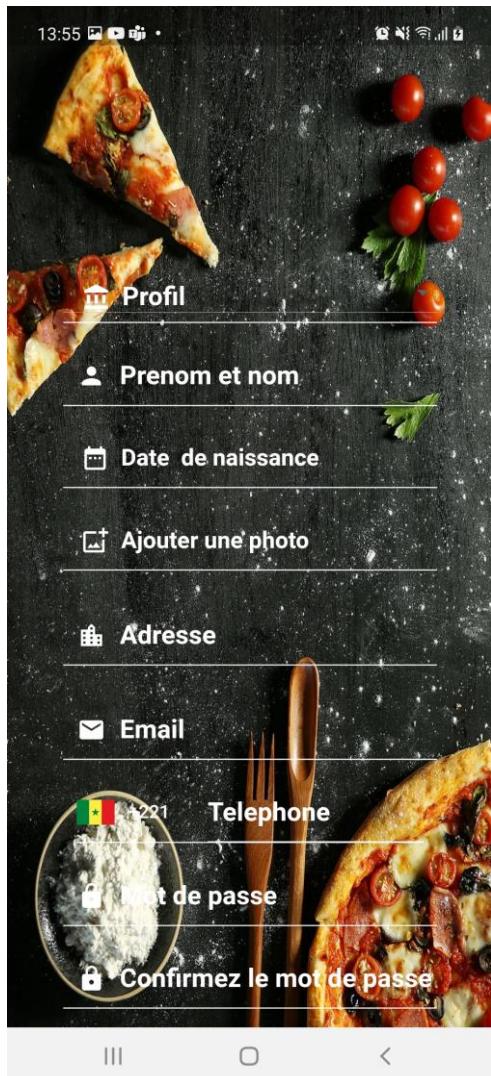
Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant



Capture 12 : Page d'authentification

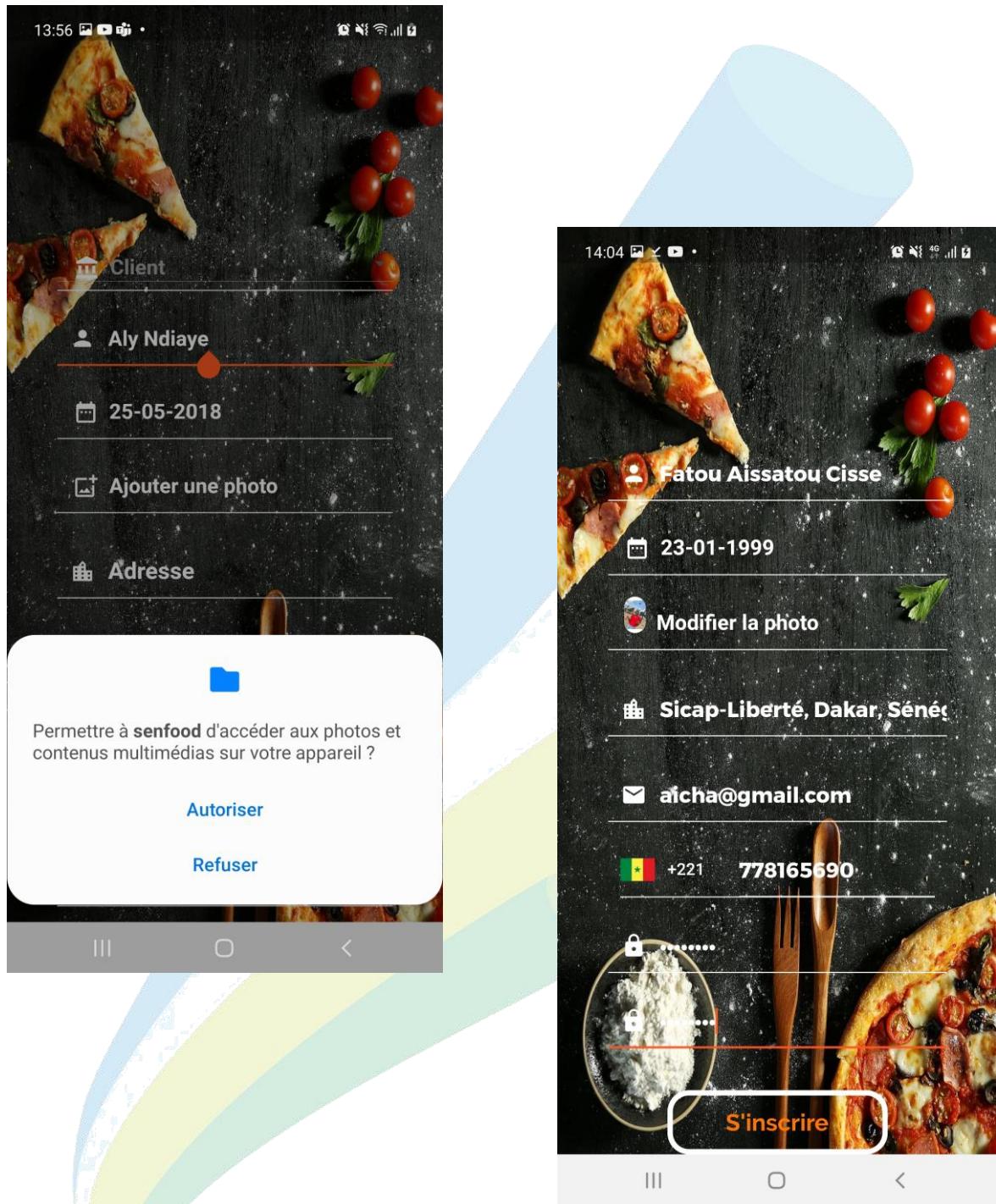


Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant



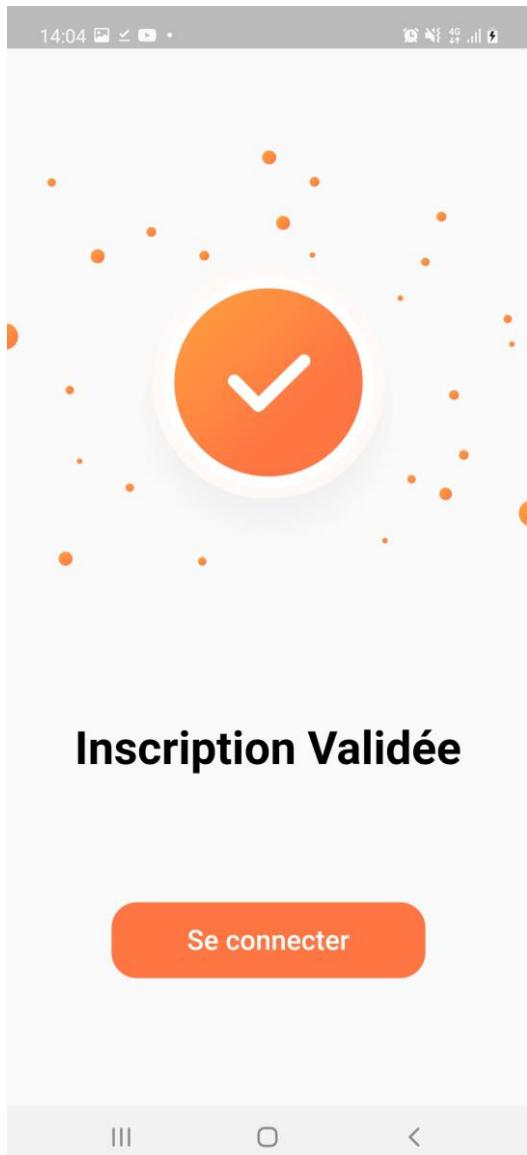
Capture 13 : Page d'inscription

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

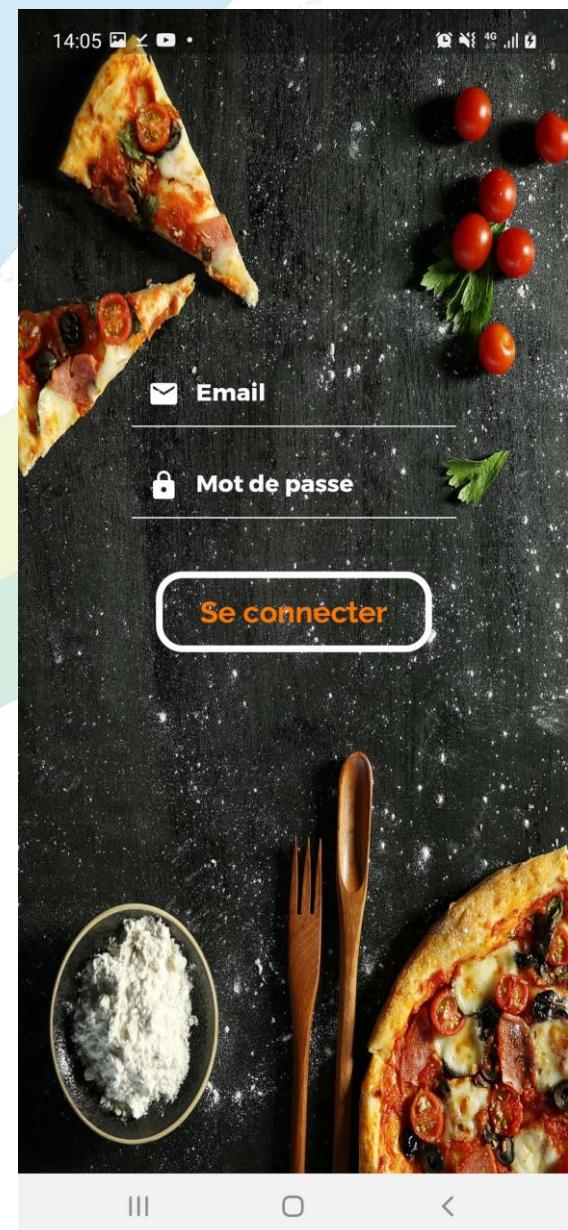


Capture 13 : Page d'inscription

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

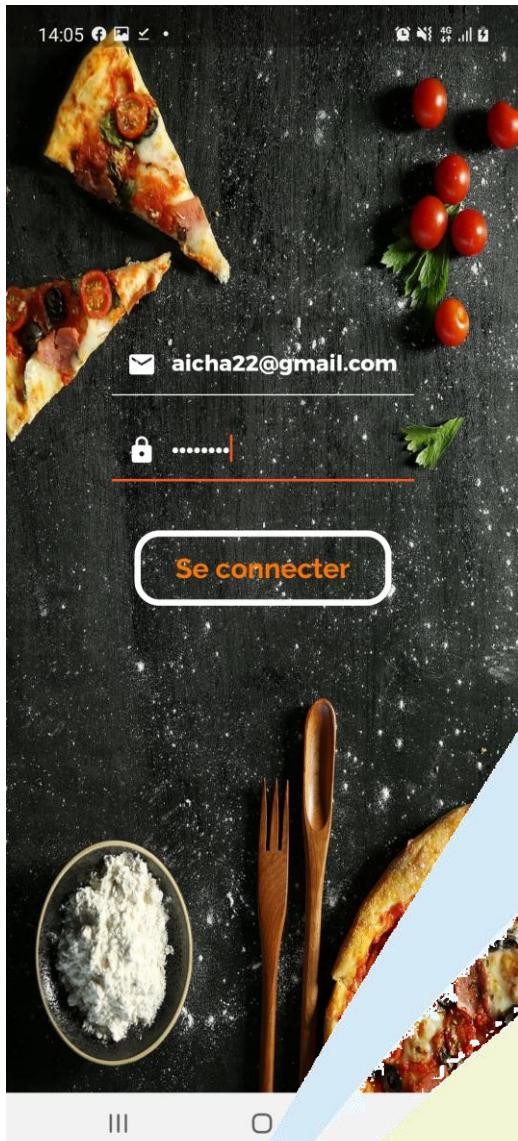


Capture 14 : Inscription validée

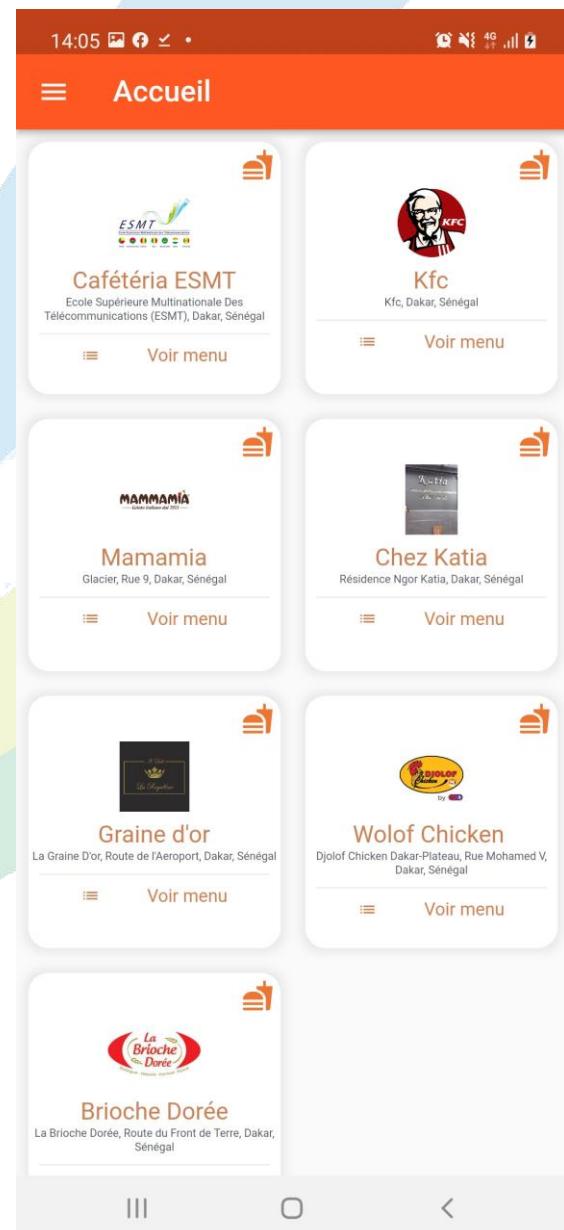


Capture 15 : Page d'authentification

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

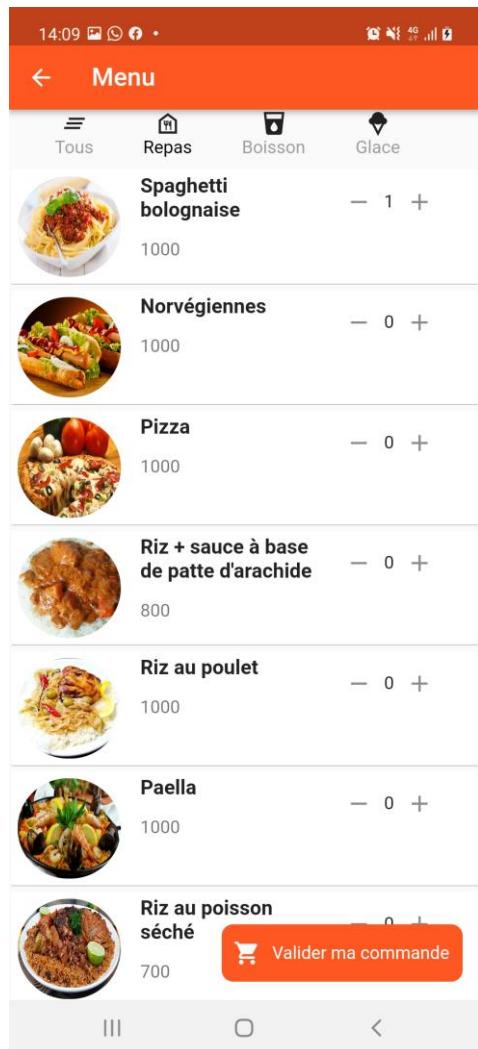


Capture 15 : Page d'authentification

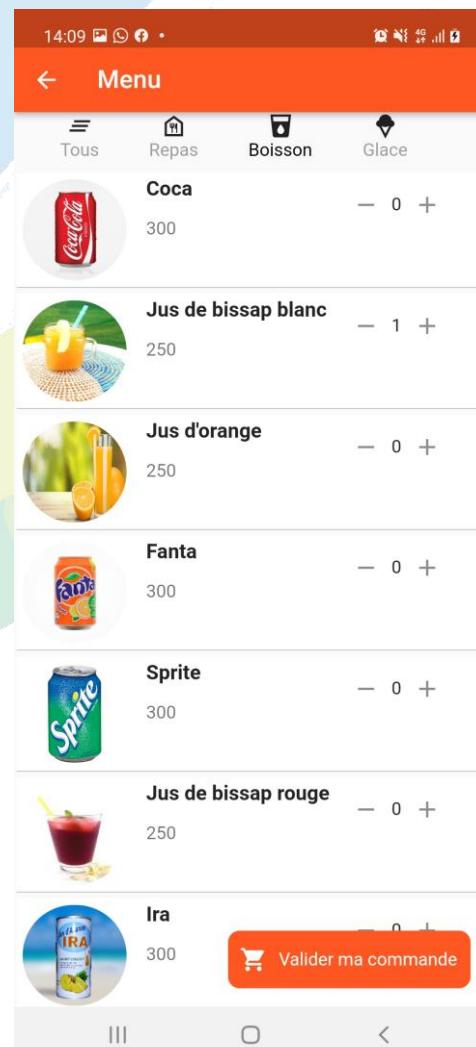


Capture 16 : Accueil client

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

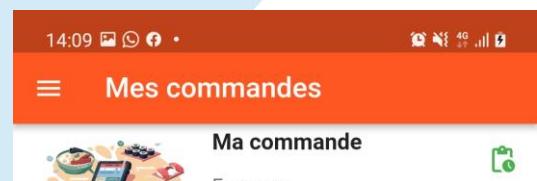
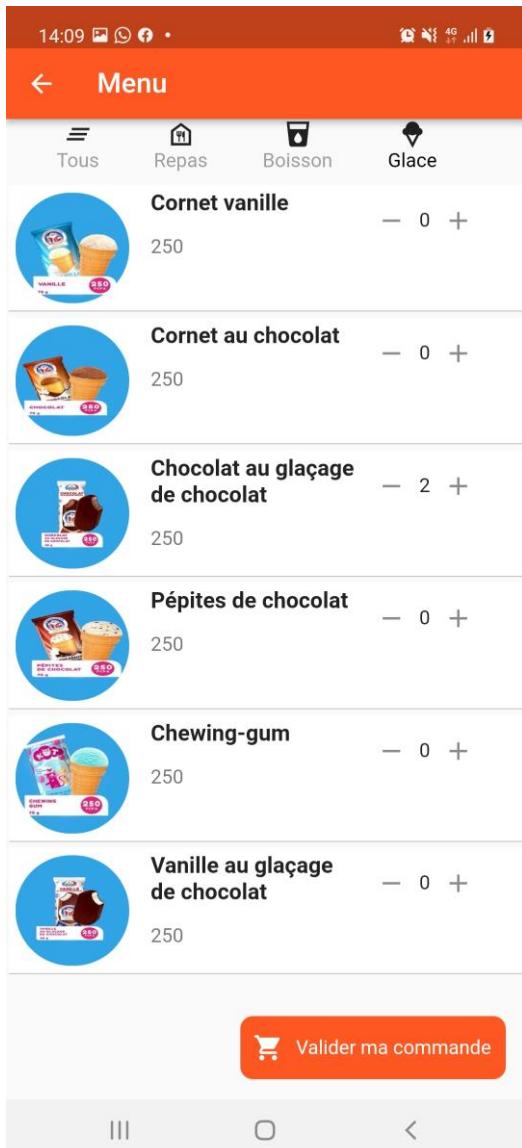


Capture 17 : Menu du restaurant ‘Cafétéria ESMT’



Capture 17 : Menu du restaurant ‘Cafétéria ESMT’

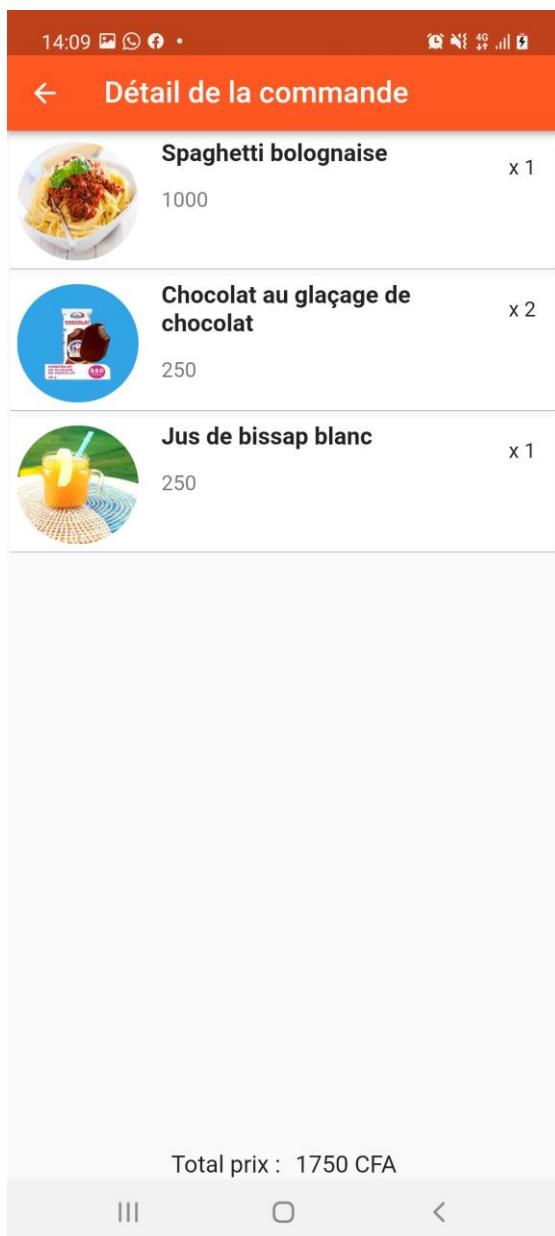
Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant



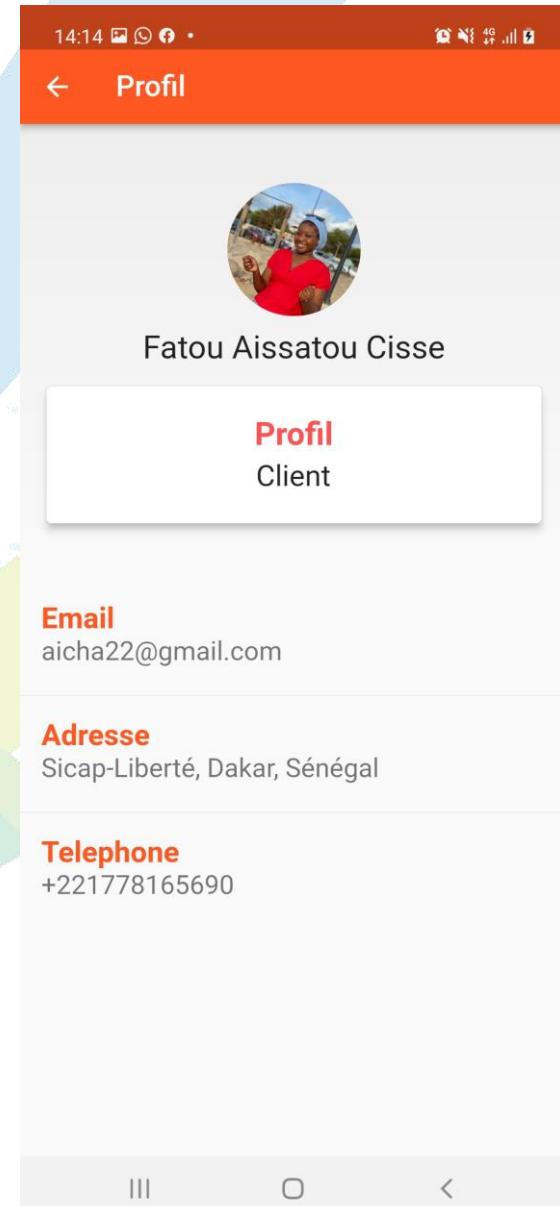
Capture 17 : Menu du restaurant ‘Cafétéria ESMT’

Capture 18 : Menu du restaurant ‘Cafétéria ESMT’

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

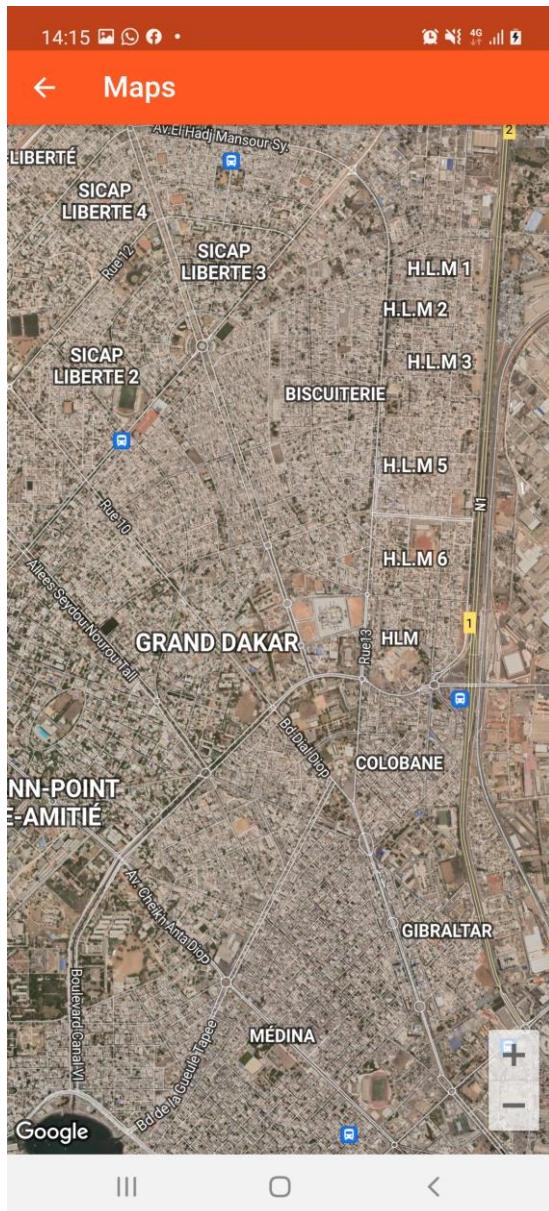


Capture 19 : Vérification de la commande

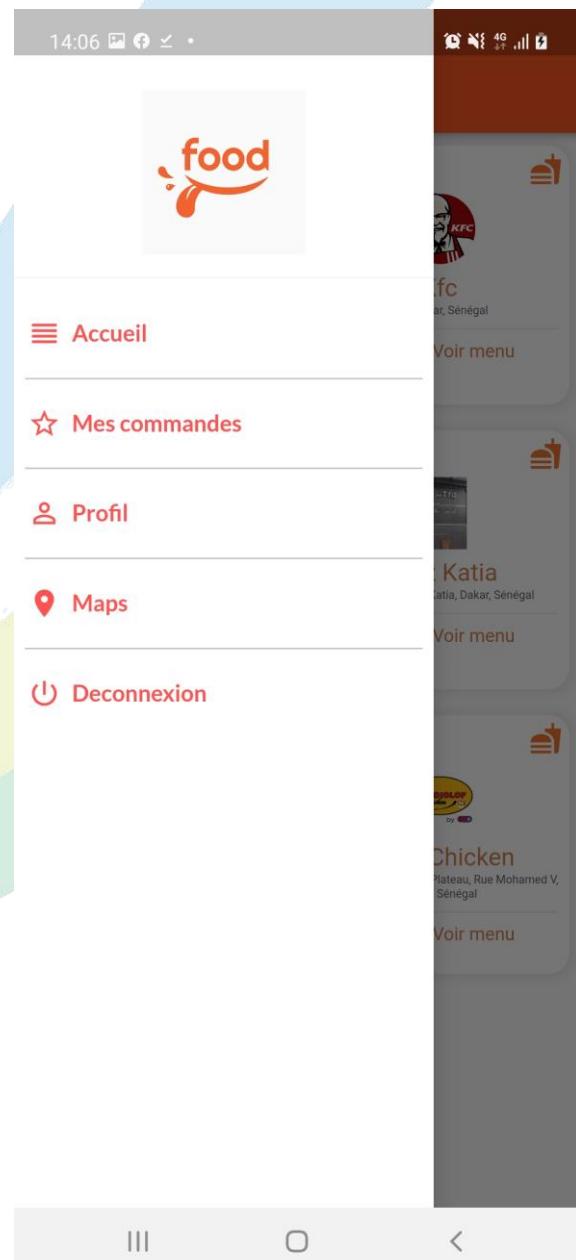


Capture 20 : Profil client

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

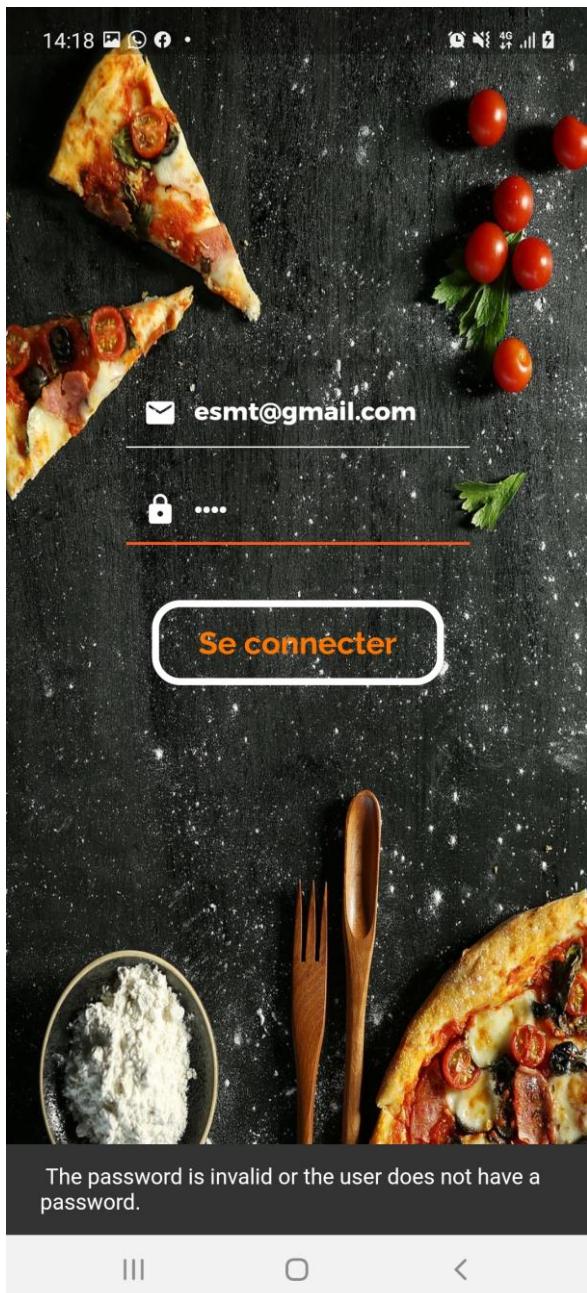


Capture 21 : Localisation de ESMT

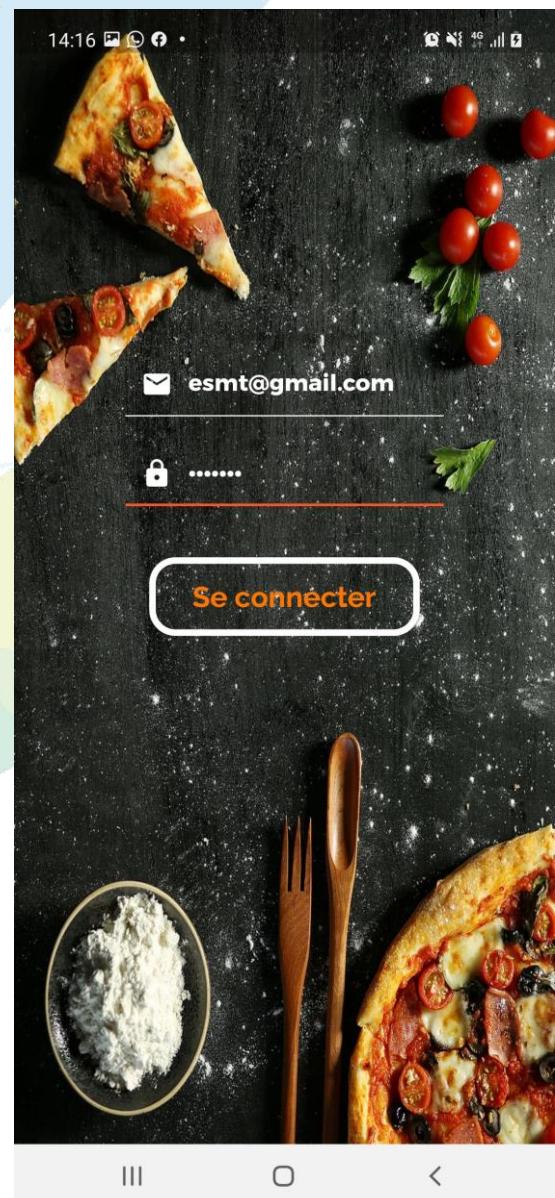


Capture 22 : Drawer client

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

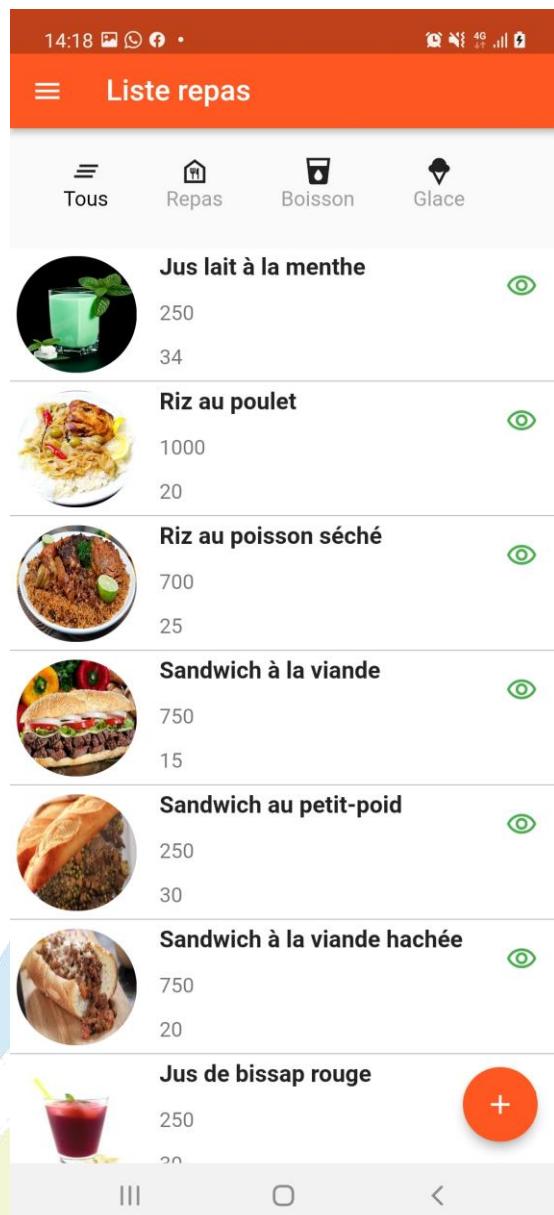


Capture 23 : Page d'authentification 'Restaurateur' avec mot de passe incorrecte



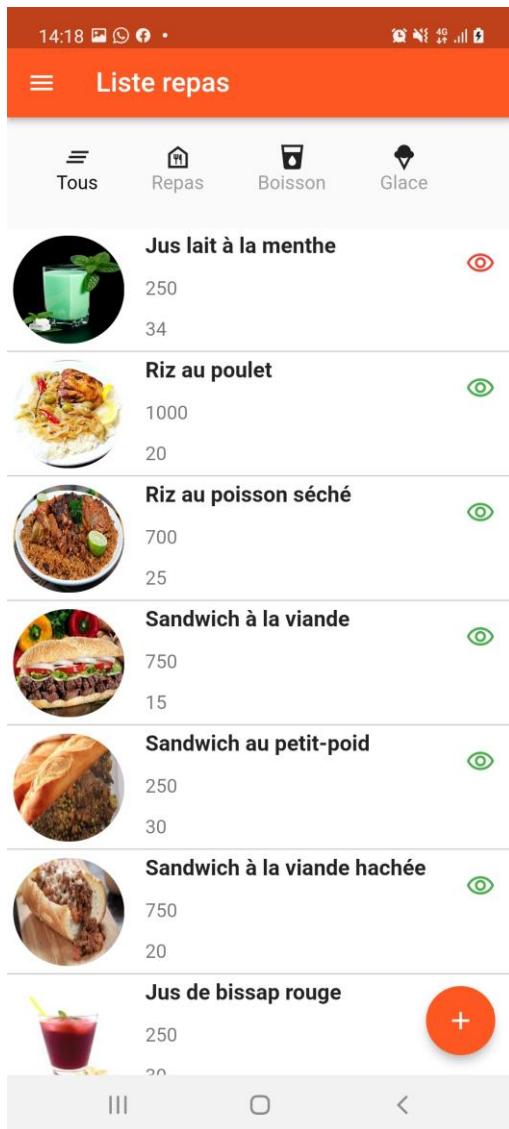
Capture 24 : Page d'authentification 'Restaurateur' avec mot de passe correcte

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

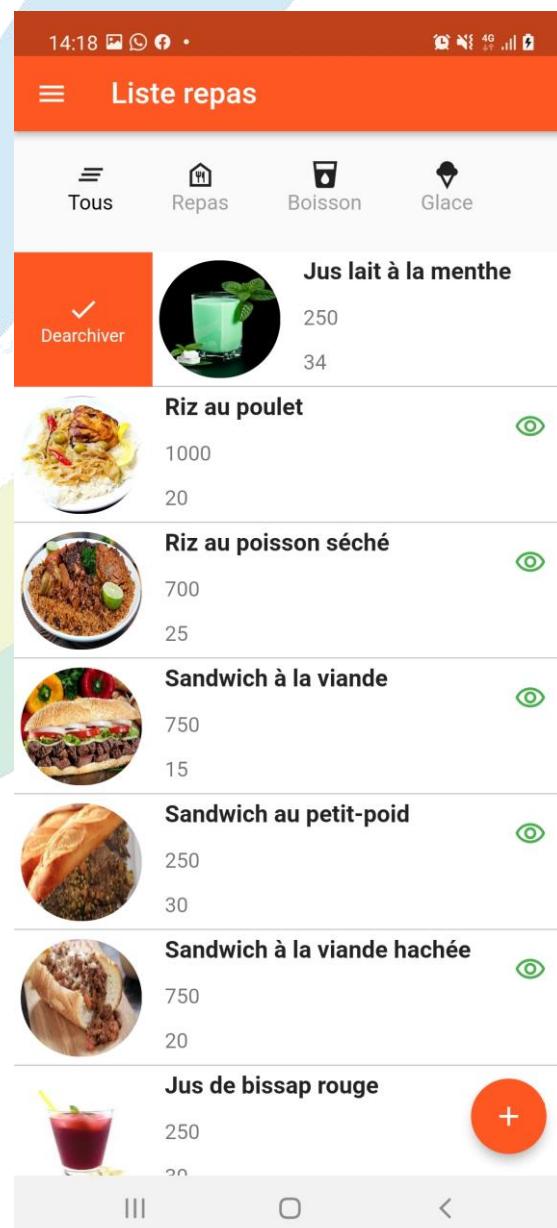


Capture 25 : Liste des plats

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

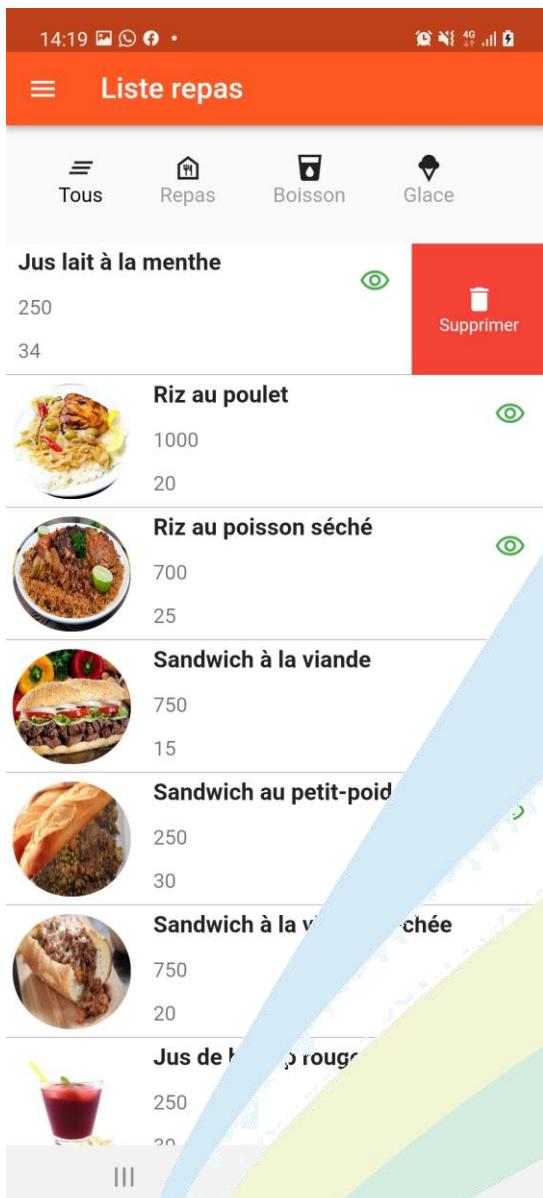


Capture 26 : Liste des plats avec 'jus lait menthe archivé'

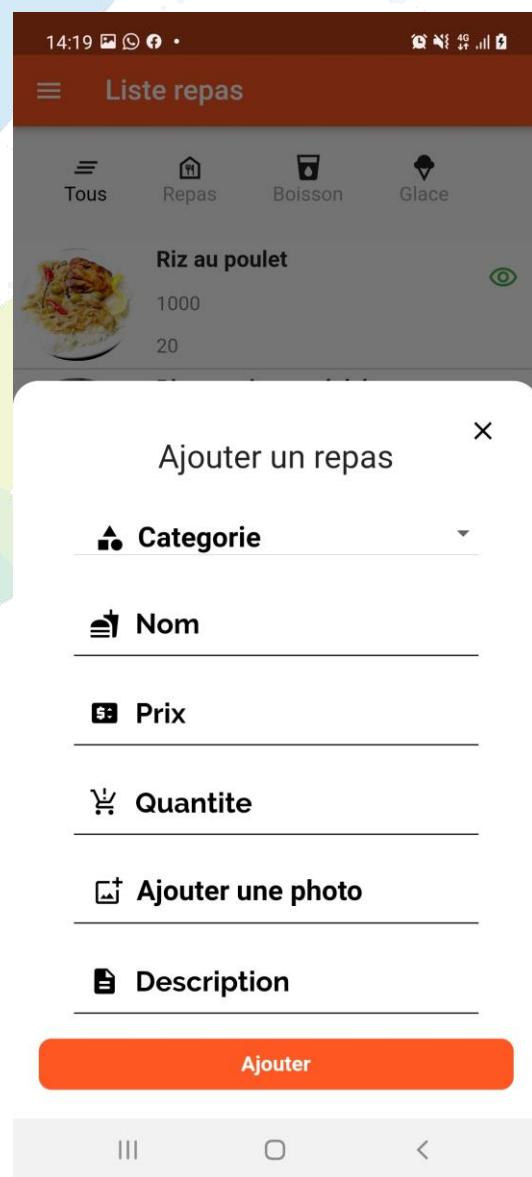


Capture 27 : Désarchivage de 'jus lait menthe'

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant



Capture 28 : Suppression de 'jus lait menthe archivé'



Capture 29 : Ajout de repas

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

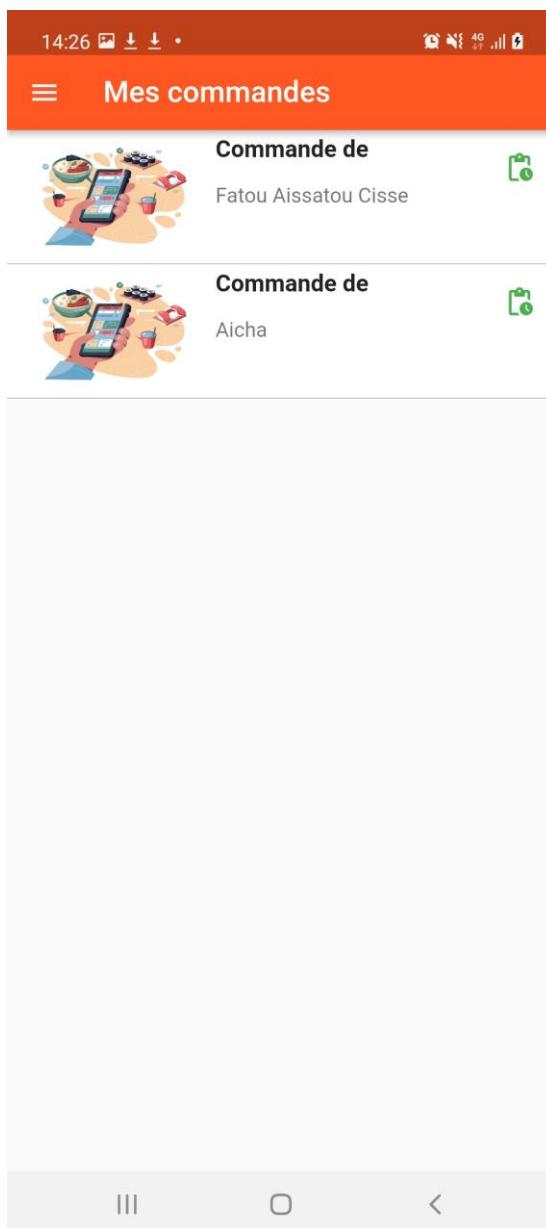


Capture 30 : Rajout de 'jus lait menthe'

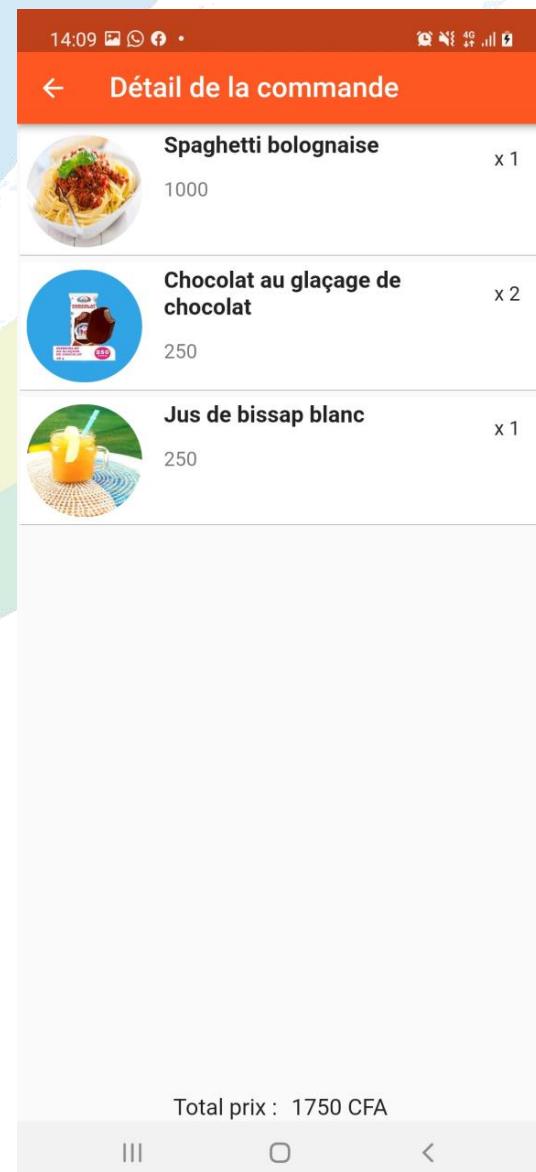
Liste repas	
Tous	Repas
Jus lait menthe	250 25
Riz au poulet	1000 20
Riz au poisson séché	700 25
Sandwich à la viande	750 15
Sandwich au petit-poid	250 30
Sandwich à la viande hachée	750 20
Jus de bissap rouge	250 20

Capture 31 : Nouvelle liste après l'ajout

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

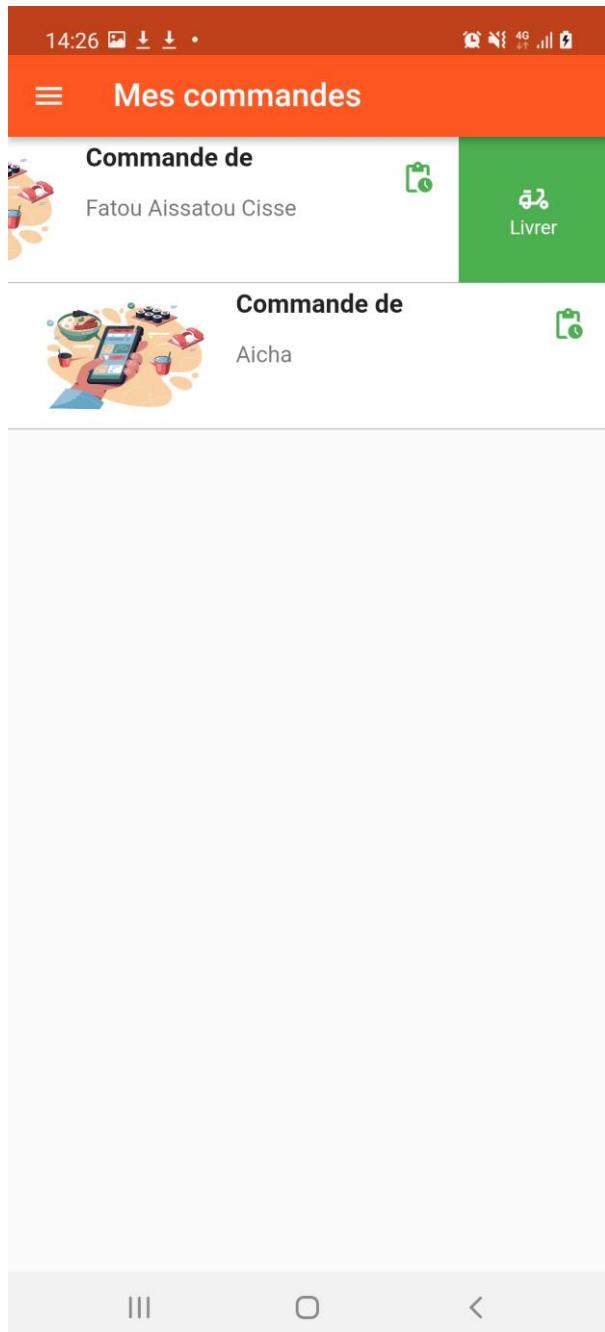


Capture 32 : Liste commande du restaurant

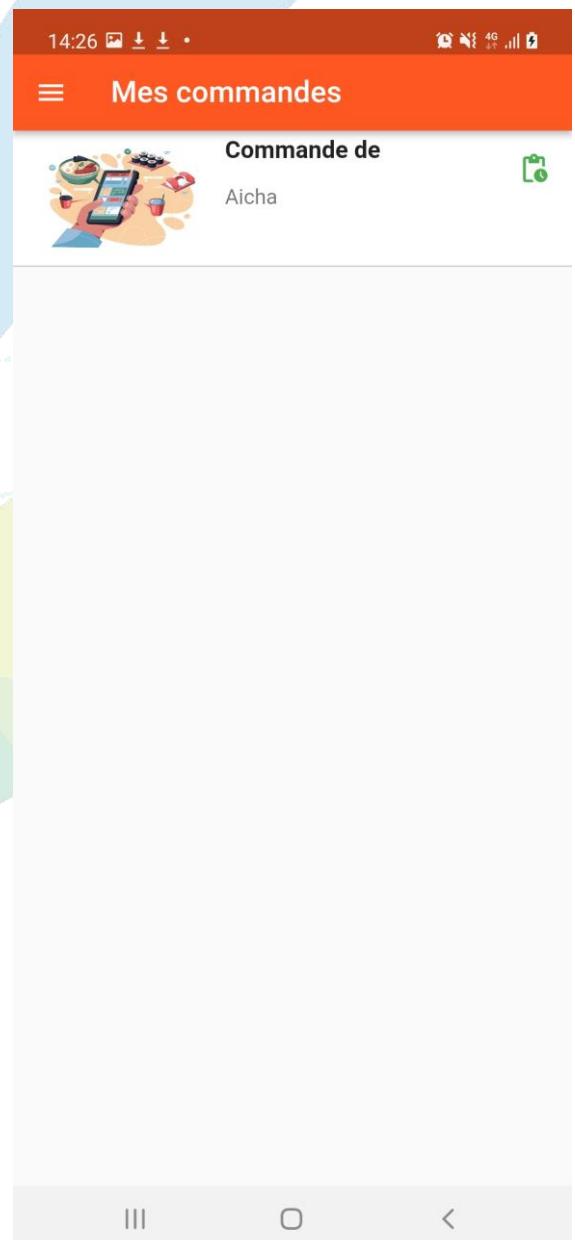


Capture 33 : Commande du client 'Fatou A. C.'

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant



Capture 34 : Livraison de la commande de 'Fatou A. C.'



Capture 35 : Livraison validée

CONCLUSION ET PERSPECTIVES

L'objectif de mémoire était d'intégrer un nouveau moyen de commande et de livraison à travers une plateforme mobile. Cette application n'est rien d'autre que 'Senfood'. En effet, la plateforme permet aux clients de réaliser leurs commandes et livraisons en ligne.

Comme nous l'avons constaté pour chaque aspect de notre application, l'outil adéquat permet au restaurateur de mieux gérer ses données, libère son personnel de certaines tâches ingrates et améliore l'efficacité globale de ses opérations. En intégrant les bons outils, non seulement ils facilitent le travail de son personnel et améliorent l'expérience de ses clients, mais ils collectent des données précieuses concernant sa clientèle. Avec les outils adéquats, ils peuvent exploiter pleinement leurs données et les utiliser pour fidéliser sa clientèle, proposer des offres et des expériences personnalisées, analyser les tendances, et améliorer chaque aspect de l'expérience de ses clients.

Le projet a été réalisé en 3 livrables : l'étude du marché, la conception et l'application elle-même. Cependant, nous avons passé plus de temps sur la conception, car étant essentielle pour la réalisation de ce projet. Par ailleurs, la réalisation de ce dernier ne s'est pas déroulée sans obstacle. Nous avons, en effet, rencontré certains problèmes liés à l'intégration de Flutter. Toutefois, grâce à l'expertise de notre maître de stage, et de sa collaboration, on a pu surmonter cette difficulté.

Après deux mois de stage à Gainde 2000 pour l'obtention du diplôme de License professionnelle en télécoms et informatiques, nous pouvons estimer que l'expérience a été très fructueuse. En effet, elle nous a enrichi en connaissances et surtout en immersion professionnelle.

Le stage a permis d'approfondir nos connaissances techniques grâce à l'utilisation de tous ces outils, et la possibilité offerte de travailler avec beaucoup de rigueur dans une cadre propice au développement personnel.

Cependant, il est bien de préciser que la solution 'Senfood' est à sa première version et pourrait être améliorée notamment intégrer d'autres fonctionnalités par exemple gérer l'intégration de l'API (Application Programming Interface) de Orange pour le paiement.

Il est toujours important de penser à améliorer ses applications car « Toute technologie suffisamment avancée est indiscernable de la magie » (Arthur Clarke).

WEBOGRAPHIE

Référence	Lien	Consulté le
Info restauration	https://www.restaurantdrive.fr/mettre-en-place-systeme-livraison-restaurant/	27/07/2021
Modelio	https://en.wikipedia.org/wiki/Modelio	17/08/2021
BPMN	https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation	17/08/2021
StarUML	https://inf1410.teluq.ca/teluqDownload.php?file=2014/01/INF1410-PresentationStarUML.pdf	17/08/2021
PragmaDev Studio	https://www.programmez.com/mots-cl-s-actualit-s/mod-lisation	17/08/2021
Dart	https://fr.wikipedia.org/wiki/Dart_(langage)	28/08/2021
Flutter / React Native	https://blog.back4app.com/fr/flutter-vs-react-native/	29/08/2021
Google trends flutter vs react native en 2020	https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.techahedcorp.com%2Fblog%2Freact-native-vs-xamarin%2F&psig=AOvVaw1PoyJXHFuSWd_tdH2v2Bce&ust=1632492964237000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCJjnvvWwlfMCFQAAAAAdAAAAABAD	29/08/2021
Statistique flutter vs react native en 2020	https://www.google.com/url?sa=i&url=https%3A%2F%2Fagiletech.medium.com%2Fflutter-vs-react-native-which-is-the-best-development-trend-in-2020-8fee7c059fd6&psig=AOvVaw2qWpKhjeZ47UwFoCOMiK6w&ust=1632493086462000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCPD7j-eklfMCFQAAAAAdAAAABAD	29/08/2021
Firebase authentification	https://firebase.google.com/products/auth?gclid=Cj0KCQjwvaeJBhCvARIsABgTDM6V0aRfewtu7a9ZGKjF22-urzWLhVORl6n5xqkTj9nPzCs6zeTOGP0aAo_EEALw_wcB&gclsrc=aw.ds	30/08/2021
Firebase cloud storage	https://firebase.google.com/products/storage?gclid=Cj0KCQjwvaeJBhCvARIsABgTDM6KDQeRqYT-fnMrRROivzwraUunrG_t6YZuGT8cseo2Yz3SsSN_VjkaApITEALw_wcB&gclsrc=aw.ds	30/08/2021
Android Studio	https://fr.wikipedia.org/wiki/Android_Studio	10/09/2021
IntelliJ IDEA	https://fr.wikipedia.org/wiki/IntelliJ_IDEA	10/09/2021

BIBLIOGRAPHIE

Auteur	Titre du livre
<i>Julien Trillard</i>	Flutter - Développez vos applications mobiles multiplateformes avec Dart
<i>Grady Booch, James Rumbaugh, and Ivar Jacobson</i>	Le guide de l'utilisateur UML

TABLE DES MATIERES

DEDICACE	2
REMERCIEMENT	3
AVANT-PROPOS	4
GLOSSAIRE	5
SIGLE	6
TABLE DES FIGURES	7
TABLE DES CAPTURES.....	8
LISTE DES TABLEAUX.....	9
SOMMAIRE	10
INTRODUCTION	12
PREMIERE PARTIE : PRESENTATION GENERALE.....	14
CHAPITRE I : PRESENTATION DE LA STRUCTURE D'ACCUEIL : GAINDE 2000	14
I. HISTORIQUE.....	14
II. COMPOSITION DE L'ARCHITECTURE ORGANISATIONNELLE	15
III. ORGANIGRAMME ET MISSIONS D'ORBUS FACTORY	15
III.1 ORGANIGRAMME	15
III.2 MISSIONS	16
CHAPITRE II : ETUDE PREALABLE	18
I. CONTEXTE	18
II. PROBLEMATIQUE.....	18
III. ETUDE DE MARCHE	19
III.1 ENQUETE SUR LE SYSTEME ACTUEL	19
III.2 CRITIQUE	25
IV. PROPOSITION DE SOLUTIONS	25
DEUXIEME PARTIE : MODELISATION CONCEPTUELLE ET ORGANISATIONNELLE	27
CHAPITRE I : MODELISATION CONCEPTUELLE.....	27
I. CHOIX DE LA METHODOLOGIE DE CONCEPTION	27
II. DIAGRAMME DE CAS D'UTILISATION	31
II.1 DESCRIPTION DES PRINCIPAUX ACTEURS	31
II.2 IDENTIFICATION DES DIFFERENTS DIAGRAMMES	33
III. MODELE CONCEPTUEL DE DONNEES	34
III.1 REPRESENTATION DES CLASSES	34
III.2 REPRESENTATION DES ASSOCIATIONS ENTRE CLASSES	35
III.3 REPRESENTATION DU DIAGRAMME DE CLASSE.....	36
CHAPITRE II : MODELISATION ORGANISATIONNELLE	38
I. DIAGRAMME D'ACTIVITE.....	38
II. DIAGRAMME DE SEQUENCE	40
TROISIEME PARTIE : INPLEMENTATION DE L'APPLICATION.....	42

Etude et mise en place d'un système de gestion des commandes et de suivi livraison : cas d'un restaurant

I.	ARCHITECTURE DE L'APPLICATION	42
II.	ENVIRONNEMENT DE TRAVAIL	42
II.1	ENVIRONNEMENT MATERIEL.....	42
II.2	ENVIRONNEMENT LOGICIEL	43
III.	INTERFACES DE L'APPLICATION	61
	CONCLUSION ET PERSPECTIVES	78
	WEBOGRAPHIE	79
	BIBLIOGRAPHIE	80
	TABLE DES MATIERES	81
	RESUME DU MEMOIRE.....	83

RESUME DU MEMOIRE

**Ecole Supérieure Multinationale des Télécommunications
Mémoire de fin de formation**

Pour l'obtention du diplôme de licence professionnelle en TIC

Nom et Prénom : CISSE Fatou Aïssatou

**Thème : Etude et mise en place d'un système de gestion des commandes et de
suivi livraison : cas d'un restaurant**

La Covid-19 a bouleversé le fonctionnement des restaurants. En obligeant les professionnels de la restauration, et certain livreur en général, à fermer leurs restaurants et les personnes à rester chez elles, le confinement a boosté l'utilisation des technologies de gestion de commande et de livraison en ligne.

De plus, alors que les restaurants se préparent à relancer leurs activités avec de nouvelles mesures d'hygiène et d'éloignement physique, les technologies de prise de commande et de paiement sans contact sont devenues plus que pertinentes.

Pour s'adapter à l'évolution rapide du secteur, les restaurants se tournent vers des solutions numériques et digitales. Désormais, le principal défi pour les propriétaires consiste à regrouper les bonnes technologies et à les faire communiquer entre elles pour une efficacité maximale.

Comment faciliter la gestion des commandes et livraisons des restaurants ?

Sans objectif préalable pour tout travail scientifique digne de ce nom, il est peu de chance que l'on aboutisse à un résultat conséquent et logique. Le choix de notre travail et de faire une application interactive c'est-à-dire qui permettra de faire des actions réciproques entre le client et l'application ou soit entre le restaurateur et son ordinateur ou soit encore entre le livreur et l'application. Et ce dernier pourra être avantageux et aux clients du restaurant et aux restaurateurs et même aux livreurs inscrits.

Se faisant, l'objectif fixé dans cet exercice est de mettre en place un système de commande et de livraison pour un restaurant, une application mobile disponible sur App Store et Play Store qui facilitera l'interaction client/restaurateur.