

# PROJET : Meal4Sorec

## CAHIER DES CHARGES

| VERSION | AUTEUR  | DATE<br>D'EDITION | VALIDATEUR | DATE<br>VALIDATION |
|---------|---|-------------------|------------|--------------------|
| 1.0     | <ul style="list-style-type: none"><li>• El ALAMI Achraf</li><li>• BOURAS Adam</li></ul> | 13-03-2024        |            |                    |

# Sommaire

|        |   |           |
|--------|---|-----------|
| 1.     | PERIMETRE.....  | 5         |
| 1.1.   | BUT .....   | 5         |
| 1.2.   | MISSIONS.....   | 5         |
| 1.3.   | CONTRAINTES.....  | 5         |
| 1.4.   | LIVRABLES D’ENTREE.....                                     | 5         |
| 1.5.   | LIVRABLES DE SORTIE.....                                    | 6         |
| 2.     | ETUDE DE L’EXISTANT .....                                   | 6         |
| 2.1.   | PRESENTATION DES APPLICATION EXISTANTES .....               | 6         |
| 2.2.   | INCONVENIENT DES APPLICATIONS.....                          | 7         |
| 3.     | ETUDE FONCTIONNELLE .....                                   | 8         |
| 3.1.   | OBJECTIFS FONCTIONNELS.....                                 | 8         |
| 3.2.   | BESOINS FONCTIONNELS : FONCTIONNALITES (USER STORIES) ..... | 8         |
| 3.2.1. | <i>Bloc fonctionnel : Gestion des commandes .....</i>       | <i>8</i>  |
| 3.2.2. | <i>Bloc fonctionnel : Gestion des Repas. ....</i>           | <i>9</i>  |
| 3.2.3. | <i>Bloc fonctionnel : Gestion des Comptes. ....</i>         | <i>9</i>  |
| 3.2.4. | <i>Bloc fonctionnel : Gestion des Fonctionnalité .....</i>  | <i>10</i> |
| 3.3.   | ACTEURS .....   | 11        |
| 4.     | BESOINS NON FONCTIONNELS.....                               | 11        |

# Table des figures

FIGURE 1: INTERFACES DE L'APPLICATION ZOMATO..... 7

FIGURE 2: INTERFACES DE L'APPLICATION FLIPDISH..... 7

# Table des tableaux

|   |    |
|---|----|
| TABLE 1: LIVRABLE D'ENTREE.....                     | 5  |
| TABLE 2: LIVRABLES DU PROJET.....                   | 6  |
| TABLE 3: REGLE DE GESTION DE COMMANDE .....         | 9  |
| TABLE 4: REGLE DE GESTION DE REPAS.....             | 9  |
| TABLE 5: REGLE DE GESTION DES COMPTES.....          | 10 |
| TABLE 6: REGLE DE GESTION DES FONCTIONNALITES ..... | 10 |
| TABLE 7: ACTEURS DU SYSTEME.....                    | 11 |

# 1. Périmètre

## 1.1. But

Le projet vise à mettre en place une application mobile permettant aux employés de l'entreprise Sorec de commander des repas auprès des services de restauration partenaires. L'application devra couvrir l'ensemble du processus de commande, depuis la sélection des plats jusqu'à la livraison des repas, avec les technologies Flutter, Spring Boot.

Le système cible, couvre les modules fonctionnels suivants :

- Gestion des commandes
- Gestion des repas
- Gestion de Fonctionnalités
- Gestion des comptes

## 1.2. Missions

- Etude fonctionnelle
- Etude techniques
- La conception
- La réalisation

## 1.3. Contraintes

- L'application doit être développée avec Spring Boot du côté serveur (backend) et du côté client (frontend), elle doit être développée avec Flutter.
- **Délais** : 5 mois
- **Langues du projet** : l'intégralité du projet sera réalisée à terme en français et en anglais.
- **Compatibilité Multiplateforme** : L'application devra être compatible avec les principaux systèmes d'exploitation mobiles tels qu'Android et iOS.
- **Intégration Harmonieuse** : Assurer une intégration fluide avec les systèmes existants de Sorec, y compris la gestion du système interne du service de restauration.1.4

## 1.4. Livrables d'entrée

| Livrable           |  | Date livraison | Date livraison réelle | Etat |
|--------------------|--|----------------|-----------------------|------|
| Cahier des Charges |  |                |                       |      |

Table 1: Livrable d'entrée

## 1.5. Livrables de sortie

| Livrable                               | Phase           | Date livraison | Auteur | Date validation |
|--|-----------------|----------------|--------|-----------------|
| <b>Plan d'assurance qualité (PAQ)</b>  | Etude préalable |                |        |                 |
| <b>Dossier d'analyse (DSF)</b>         | Analyse         |                |        |                 |
| <b>Dossier de conception</b>           | Conception      |                |        |                 |
| <b>Dossier Etude technique</b>         | Etude préalable |                |        |                 |
| <b>Module de gestion des commandes</b> | Réalisation     |                |        |                 |
| <b>Module de gestion des repas</b>     | Réalisation     |                |        |                 |
| <b>Module de gestion des comptes</b>   | Réalisation     |                |        |                 |
| <b>Manuel d'utilisation</b>            | Déploiement     |                |        |                 |
| <b>Manuel technique</b>                | Déploiement     |                |        |                 |
| <b>Support</b>                         | Maintenance     |                |        |                 |

*Table 2: Livrables du projet*

## 2. Etude de l'existant

### 2.1. Présentation des application existantes

Parmi les meilleures applications mobiles sur ' Play Store et App Store ', nous avons trouvé l'application « Zomato » comme indique la figure ci-dessous, cette application permet la gestion de restaurant qui permet aux restaurants de gérer leurs profils en ligne, de prendre des commandes en ligne, de gérer les réservations de table, de collecter des avis clients, et de suivre les performances à l'aide d'analyse détaillées.

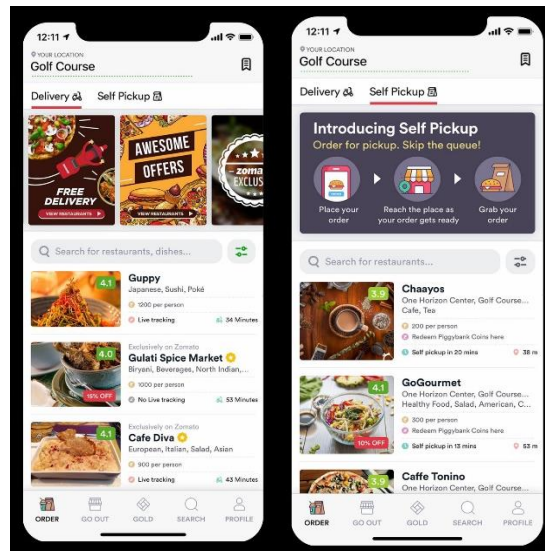


Figure 1: Interfaces de l'application Zomato

Nous avons considéré aussi l'application « Flipdish », dont les interfaces présentées dans la figure ci-dessus. Cette application offre les mêmes fonctionnalités que l'application précédentes.

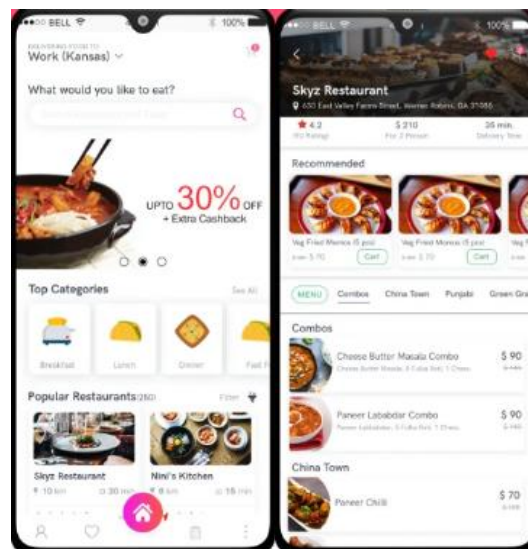


Figure 2: Interfaces de l'application Flipdish

## 2.2. Inconvénient des applications

- Certaines applications peuvent être complexes à utiliser.
- L'intégration des applications de gestion de la cantine avec les systèmes de gestion existants de l'entreprise peut être complexe.
- Les applications nécessitent une maintenance régulière pour garantir leur bon fonctionnement.

### 3. Etude fonctionnelle

#### 3.1. Objectifs fonctionnels

Le système doit permettre :

- La saisie et gestion des commandes Employées.
- La saisie et gestion et consultation des Repas.
- La gestion des Utilisateurs et le suivi des Commandes.
- La gestion des Fonctionnalités et paramétrage du Système.

#### 3.2. Besoins fonctionnels : Fonctionnalités (user stories)

##### 3.2.1. Bloc fonctionnel : Gestion des commandes

Le système doit permettre dans ce bloc :

- 1- Enregistrement des détails de la commande tels que les Repas (Entrées, plats, desserts, boissons), l'horaire de récupération.
- 2- Possibilité d'ajouter, modifier ou supprimer des Repas dans une commande existante.
- 3- Statut en temps réel des commandes (en attente, traitées, prêt, etc.).
- 4- Historique des commandes pour un suivi et une référence ultérieure.
- 5- Notifications automatiques en cas de modification de l'état de la commande.

| Règle de gestion | Description  | Type   |
|------------------|--|--------|
| <b>RG01</b>      | Structure d'une commande : <ul style="list-style-type: none"><li>• Référence de la commande (automatique)</li><li>• Date de la commande</li><li>• Matricule de l'employée</li><li>• Montant</li><li>• Horaire de Récupérations</li><li>• Etat de la commande</li><li>• Liste des Repas (référence, libelle)</li><li>• Type de command (surplace ou a emporté)</li><li>• Nombre maximale des commandes<br/>surplace=NMCS</li><li>• Nombre actuel de commandes surplace=NACS</li></ul> | Métier |
| <b>RG02</b>      | Horaire de Récupérations doit être disponible  | Métier |
| <b>RG03</b>      | Horaire de Récupérations disponible $\Leftrightarrow$ NMCS > NACS  | Métier |
| <b>RG04</b>      | Une commande ne peut pas être supprimer par l'employée si l'état de cette dernier est «en cours»   | Métier |



|             |   |        |
|-------------|---|--------|
| <b>RG05</b> | Un client « bloqué », n'a pas le droit d'effectuer une commande,    | Métier |
| <b>RG06</b> | Une fois la commande ajoutée, l'interface est réinitialisée (reset) | WF     |

### 3.2.2. Bloc fonctionnel : *Table 3: Règle de Gestion de Commande* **Gestion des Repas.**

Le système doit permettre dans ce bloc :

- 1- Enregistrement des détails du Repas tel que le type du repas (entrée, plat, dessert, boisson), ingrédients, moyen du repas.
- 2- Possibilité d'ajouter, modifier ou supprimer des Repas.
- 3- Ajoute un repas aux favoris, et notification lors de présence de ce dernier.
- 4- List des commandes contenant ce repas.

| Règle de gestion | Description  | Type   |
|------------------|--|--------|
| <b>RG07</b>      | Structure d'un repas : <ul style="list-style-type: none"> <li>• Référence du repas (automatique)</li> <li>• Nom du repas</li> <li>• Catégorie du repas (Entrée, plat, boisson, dessert) ?</li> <li>• Avis sur le Repas</li> <li>• Descriptions des ingrédients</li> <li>• HasToBeUnique</li> <li>• Image du repas</li> </ul> | Métier |
| <b>RG08</b>      | Un repas peut avoir à une seul catégorie   | Métier |
| <b>RG09</b>      | Un staff peut pas ajouter deux repas avec le même nom  | Métier |
| <b>RG10</b>      | Un repas ne peut pas être ajoute sans (Image, catégorie, Description des ingrédients)  | Métier |

*Table 4: Règle de Gestion de Repas*

### 3.2.3. Bloc fonctionnel : **Gestion des Comptes.**

Le système doit permettre dans ce bloc :

- 1- Authentications sécurise via le matricule d'un employée Active.
- 2- Possibilité d'ajouter, modifier ou supprimer Un Compte
- 3- Possibilité d'affecter un Rôle a un Compte.

| Règle de gestion | Description  | Type   |
|------------------|--|--------|
| <b>RG11</b>      | Structure d'un Compte : <ul style="list-style-type: none"> <li>• Matricule</li> <li>• Nom d'utilisateur</li> <li>• Mot De Passe</li> <li>• Email</li> <li>• Numéro de téléphone</li> <li>• Rôle</li> </ul> | Métier |
| <b>RG12</b>      | Le Compte doit absolument avoir un matricule d'un employé qui existe et qui est actif  | Métier |
| <b>RG13</b>      | Un staff ne peut pas avoir des commandes   | Métier |
| <b>RG14</b>      | Un Rôle peut être soit « Employé », « admin », « Staff »   | Métier |
| <b>RG15</b>      | Un compte « Admin » ne peut pas effectuer une commande pour le faire, il devra se connecter à son compte employée  | Métier |

Table 5: Règle de Gestion des Comptes

### 3.2.4. Bloc fonctionnel : Gestion des Fonctionnalités

Le système doit permettre dans ce bloc :

- 1- Consulter la liste des fonctionnalités du système.
- 2- Activer/Désactiver une fonctionnalité du système.

| Règle de gestion | Description  | Type   |
|------------------|--|--------|
| <b>RG16</b>      | Seul l'admin Peut accéder à tous les fonctionnalités                                   | Métier |
| <b>RG17</b>      | Le Compte doit absolument avoir un matricule d'un employée qui existe et qui est actif | Métier |

Table 6: Règle de Gestion des Fonctionnalités

### 3.3. Acteurs

| Acteur                           | Rôle  |
|----------------------------------|---|
| Staff Du Service De Restauration | <ul style="list-style-type: none"><li>- Editer et gérer les commandes</li><li>- Gérer les Repas</li></ul>   |
| Administrateur                   | <ul style="list-style-type: none"><li>- Gestion des comptes des utilisateurs</li><li>- Gestion de fonctionnalité</li></ul>                                    |
| Employé                          | <ul style="list-style-type: none"><li>- Effectuer une Commande</li><li>- Consulter l'historique des commandes</li><li>- Donner un avis sur un repas</li></ul> |

Table 7: Acteurs du Systeme

## 4. Besoins non fonctionnels

- **Sécurité des Données** : Garantir la sécurité des informations personnelles des utilisateurs en mettant en place des protocoles de sécurité robustes.
- **Compatibilité Multiplateforme** : L'application devra être compatible avec les principaux systèmes d'exploitation mobiles tels qu'Android et iOS.
- **Intégration Harmonieuse** : Assurer une intégration fluide avec les systèmes existants de Sorec, y compris la gestion des employés et la comptabilité ainsi que le système interne du service de restauration.
- **Disponibilité** : le système doit assurer un fonctionnement normal lors de son exploitation, et ne pas dépasser 8h (<8h/an) par an de dysfonctionnement.