**Analyse US-14 :**

**Page communauté des clients**

1) Description et objectif de la US :

Cette User Story sera définie par une page web uniquement accessible par le patron du restaurant après s’être connecté avec ses identifiants personnels. Un onglet “Communauté” lui sera alors accessible dans le menu de navigation du site web.  
Une fois sur cette page, le patron pourra voir l’ensemble de ses clients inscrits sur son site. Il pourra en voir 9 par page et disposera de flèches sur les côtés afin d’afficher d’autre clients. Pour faciliter sa recherche, il y aura la possibilité de filtrer plus précisément pour trouver un client spécifique (ou un groupe de client) selon sa localité, son nom/prénom et selon sa popularité auprès du restaurant (blacklisté ou non).

Le patron sera également capable d’ajouter un commentaire sur le client de son choix (par exemple pour signaler une remarque ou autre) et/ou s'il le souhaite, marqué un client comme “blacklisté”.

L’objectif de cette page est donc d’avoir une vue d’ensemble des clients de son restaurant et de noter au fur à mesure si des clients commettent des infractions (fausse commande, spam,) envers son établissement (ou au contraire, placer une note positive peut-être ?).

Il est également à noter que lorsqu’un client sera blacklisté, il pourra toujours accéder à son compte mais sera incapable de passer commande et verra un message sur son profil personnel de la raison pour laquelle il n’est plus capable de passer commande.

2) Critères d’acceptation :

* Seulement lorsque le patron se connecte avec ses identifiants privés, un onglet “Communauté” lui est accessible dans la barre de navigation
* Lors d’un clic sur la page “Communauté” le patron est correctement redirigé vers la page et les 9 premiers clients s’affichent correctement sur la page selon la taille de l’écran utilisée
* Les informations de chaque client correspondent bien aux informations qu’ils ont utilisé lors de leur inscription et se mettent bien à jour si un client à modifié certaines d’entre-elles
* Lorsque le patron écrit un commentaire dans la case d’un client spécifique et qu’il clic sur le bouton “sauvegarder” pour mettre à jour les informations du client, le message est bien envoyé dans la base de données et s’affichera désormais bien dans la page communauté du client spécifique
* Si le patron à cocher la case “blacklister” d’un client et qu’il clic sur le bouton “enregistrer”, les informations se mettent bien à jour dans la base de données et le client blacklisté ne doit plus être capable de passer commande (message à afficher sur sa page personnelle). Un client qui vient d’être blacklisté doit également recevoir un mail lui indiquant son statut.
* Lorsque le patron clic sur une des flèches sur le côté, 9 nouveaux clients défilent bien selon l’ordre alphabétique pour la flèche de droite et anti-alphabétique pour la flèche de gauche (style menu déroulant horizontal)
* Lorsque le parton effectue une recherche d’un client ou d’un groupe de client, les filtres appliqués lui renvoient bien le/les client(s) affectés par cette recherche. Toujours par groupe de 9 au maximum avec la possibilité de naviguer sur d’autre clients avec les flèches.

3) Prérequis :

Afin que cette US puisse être mené à bien, il faut au préalable avoir réalisé :

* Le système d’inscription/connexion (US7)
* Un accès spécifique lorsque le parton entre des informations de connexion (TT + US9)
* Un serveur de contact afin qu’un mail puisse s’envoyer aux clients blacklistés (TT)
* La base de données avec les tables spécifiques et les informations à jour des clients (TT)
* La barre de navigation et le pied de page en tant que “components” React (TT)

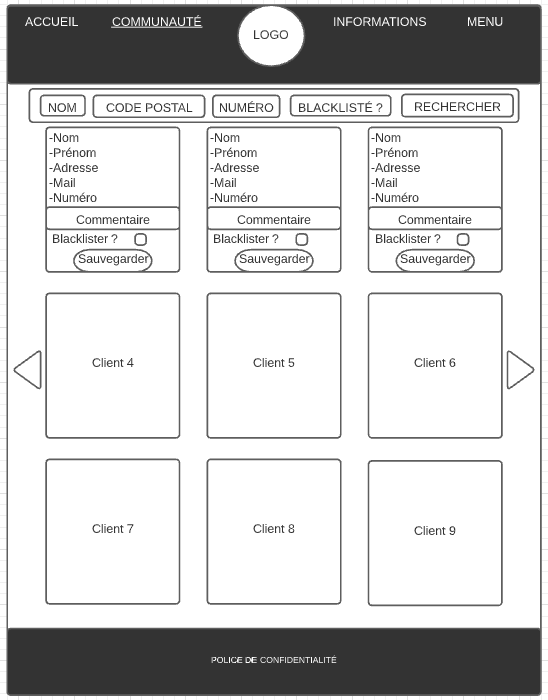
4) Emplacement de la fonctionnalité :

On pourra accéder à cette fonctionnalité de 2 manières. La seule condition est qu’il faut être connecté avec des identifiants correspondant à ceux du patron.

Un bouton “Communauté” sera implémenté dans la barre de navigation et qui sera donc accessible quelle que soit la page où se trouve la patron (car la barre de navigation reste la même, seul l’onglet ou l’on se trouve sera souligné dans la barre de navigation pour plus de clarté). Ce sera donc dans les “components” de React, celui de la barre de navigation plus précisément, que ce bouton sera implémenté.

Le patron peut également y accéder directement avec l’url de la page (qui devrait ressembler à [www.chickandfish.be/communauté](http://www.chickandfish.be/communauté)) à condition qu’il soit connecté.

5) Maquette de la page :



6) Les champs :

* La barre de navigation est un composant à réutiliser (avec React)
* Les informations des clients sont des données à importer de la base de données
* La section “commentaire” disponible pour chaque client, est un champ de type “string” que le patron pourra choisir de remplir comme bon lui semble (une limite de 100 caractères peut être imposée)
* Chaque client sera composé d’un formulaire (<form></form>) de 2 valeurs. Une pour le commentaire et une pour le “Blacklist”
* Le bouton à coté de “Blacklister ?”sera de type radio et sera cocher pour un “TRUE” et non cocher pour un “FALSE”
* “Sauvegarder” est un bouton qui enverra tous les champs du formulaire pour les mettre à jour dans la base de données (donc uniquement commentaire et blacklister)
* La barre de recherche est composée de :
  + Nom : string qui recherche les personnes au nom correspondant
  + Prénom : string qui recherche les personnes au nom correspondant
  + Localité : integer qui recherche les personnes de cette localité
  + Numéro : string qui recherche les personnes avec ce numéro de gsm/téléphone
  + Blacklisté : input de type radio avec deux options (OUI ou NON) qui filtre les personnes blacklistées ou non
  + Recherche : input de type bouton qui va appliquer les filtres ci-dessus et retourner la recherche correspondante

Ce système de filtre permet la recherche d’une ou plusieurs personne(s), les filtres peuvent se cumuler pour affiner la recherche

* Le pied de page (footer) est également un composant React à implémenter.

7) API :

8) Particularités techniques :

Aucune librairie externe ne doit être utilisé pour réaliser cette fonctionnalité. Il faut juste utiliser du CSS/HMTL pour la forme, du JS/AJAX pour les connectivités avec la DB, utiliser node.js (+ son framework express.js) pour le backend et React pour le frontend.

9) Gestions des erreurs :

Si un client ou tout autre personne non-autorisée tente d’accéder à cette page via l’url ([www.chickandfish.be/communauté](http://www.chickandfish.be/communauté) Par exemple), il ne doit pas pouvoir voir le contenu de cette page. Il doit soit être redirigé vers la page de connexion/inscription soit obtenir un message d’erreur sur la page communauté tel que “Personne non-autorisée, vous devez vous connectez en tant qu’administrateur pour avoir accès à cette page.”

Le champs “commentaire” de chaque client, doit contenir l’attribut nécessaire (maxlength=”value” en HTML) pour limiter le nombre de caractères que le patron peut entrer afin que ce nombre correspondent à celui de la case prévue à cet effet dans la base de données afin d’éviter toute erreur d’encodage. De plus, tout le commentaire sera converti en string afin que le type du commentaire soit toujours en string, pour que la base de données reçoive également le bon type (si le patron encode un 18, le résultat est : “18”)

Si la recherche de personnes spécifiques avec la barre de recherche n’aboutit à aucune personne, un message doit s’afficher précisant que, par exemple, “cette recherche ne correspond à aucune personne”.