

Document d'architecture technique



MyMuffin

Application de vente en ligne de muffins !

Table des matière

Table des matière	1
Lexique	1
Contexte	2
Extrait du cahier des charges	2
Besoin principal	2
Fonctionnalités souhaitées	2
Architecture applicative/technique	3
Diagramme activité	3
Scénario classique utilisateur	3
Déroulement	3
Proposition de technologie	4
Architecture physique / infrastructure	4
C4	4
Environnements	5
Développement	5
Production	5
Coûts	5

Lexique

Firestore : Firestore est un ensemble de services d'hébergement pour n'importe quel type d'application. Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale, et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel.

SDK : Un kit de développement logiciel (SDK) est un ensemble d'outils fourni avec une plateforme matérielle (généralement), un système d'exploitation ou un langage de programmation.

E-commerce : Le commerce en ligne, commerce électronique ou e-commerce, est l'échange pécuniaire de biens, de services ou d'informations par l'intermédiaire des réseaux informatiques, notamment Internet.

Contexte

On souhaite créer une application de vente en ligne de muffins et autres pâtisseries (liste des autres pâtisseries à définir plus tard). **Attention** ! Prévoir l'extension de l'application pour prendre en compte de nouveaux types d'articles de vente. Comme des boissons par exemple !

Extrait du cahier des charges

Besoin principal

Pouvoir **commander** des muffins :

- Si possible personnalisables au niveau des ingrédients
- Sur une appli e-commerce

Fonctionnalités souhaitées

Authentification

- Pour utiliser l'application : compte utilisateur
 - + Mot de passe oublié
 - + A envisager : système de fidélité ?

Livraison de muffins à la maison

- voir le service de livraison à utiliser

Personnaliser son muffin / sa commande

- Différents ingrédients, goûts, toppings, accompagnements

Paieement en ligne de la commande

Confirmations par mail

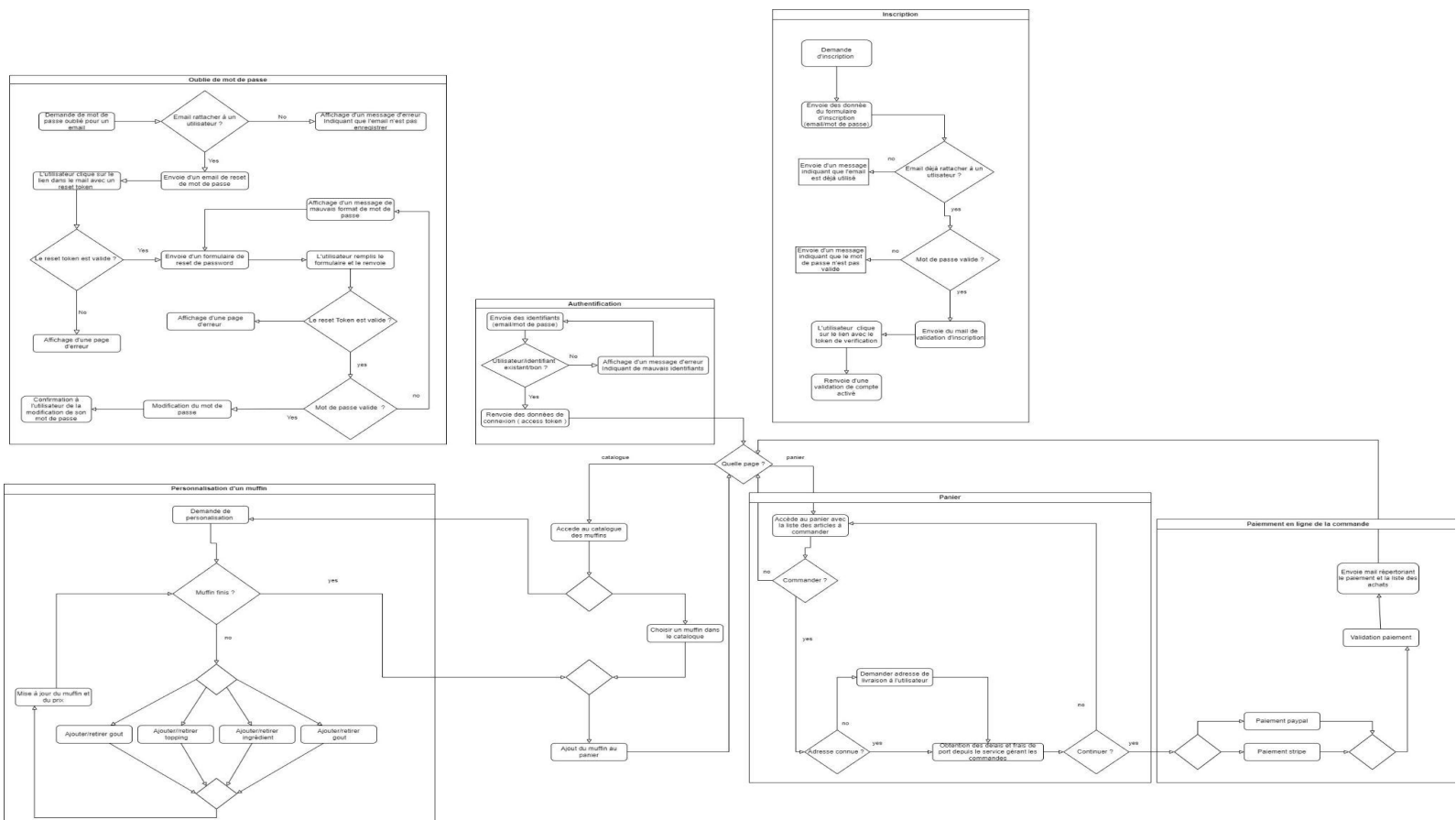
- de l'inscription
- Récapitulatif de la commande

Architecture applicative/technique

Diagramme activité

Lien du [diagramme d'activité](#) + voir version exporter (diagramme d'activité.jpeg)

Adapter le diagramme en fonction de l'utilisation de firebase ou non et du service de livraison à définir avec le client.



Scénario classique utilisateur

Déroulement

Je me connecte ou je m'authentifie

Je suis connecté

Pouvoir renseigner son adresse de livraison

J'accède au catalogue des différents muffins

Je choisis un muffin ou autre

> **Personnalisation** du muffin : liste des ingrédients + calcul du coût

Je rajoute au panier mon article / mes articles

Je vais voir la synthèse de ma commande / mon panier d'achats

Je valide la commande

Je choisis mode de paiement

Je paye via PayPal ou Stripe par exemple

Affichage à l'utilisateur du temps estimé pour sa commande

Capable de calculer le temps estimé, comment ?

Proposition de technologie

On pourrait utiliser flutter qui permet avec une même base de code (selon les package utilisé) d'évoluer vers une version mobile (android/ios) et web rapidement.

Pour la partie backend utiliser les service Firebase :

- [Firebase Hosting](#) : déploiement du serveur web
- [Firebase Trigger Mail](#) + [SendGrid](#) : pour les emails de confirmation de commande et autre envoie lié à des ajouts en base
- [Firebase Realtime Database](#) (voir aussi pour firestore, à étudier) : base de donnée nosql type document json (pour firestore collection et document)
- [Firebase Auth](#) : pour l'authentification des utilisateurs

Pour les paiements (ou voir extension marketplace firebase) :

- flutter_stripe: ^7.0.0
- paypal_sdk: ^1.0.5

Firebase ayant déjà développer les packages client flutter pour leur service on gagnerait du temps sur ces parties :

- Pour l'authentification : firebase_auth: ^4.2.3
- Pour l'accès à la base de donnée : firebase_database: ^10.0.8

Pour évolution : Firebase Messaging (FCM) pour les notifications (exemple : notification des commande ou nouveau stock de muffin préféré etc)

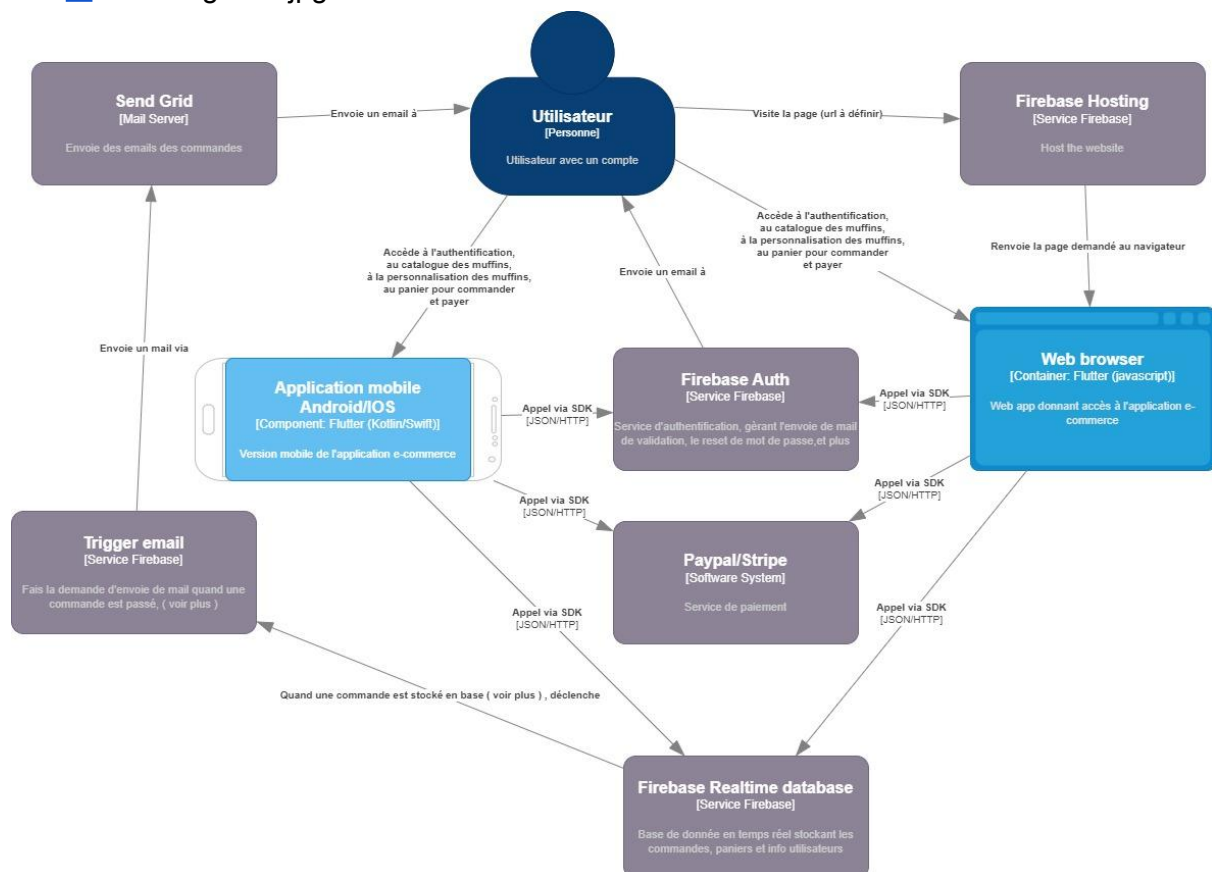
Utilisation de Stream/Provider en fonction de la base de données utilisée.

Architecture physique / infrastructure

Toute l'infrastructure sera déléguée au service Firebase comme décrit dans [Proposition de technologie](#) .

C4

Voir [ici](#) ou l'image C4.jpg



Environnements

Développement

Pour le web et le mobile on utilise le mode debug de flutter (+ simple via android studio). On pourrait utiliser le package dhttpd: ^4.0.1 en local pour les tests avant un déploiement. Pour la partie back avec firebase-emulator pour tout ce qui est lié à Firebase (voir [ici](#)) . Peut-être essayé de dockeriser tout ça pour avoir un environnement de travail facilement répliquable (voir [ici](#)) .

Production

Déploiement du serveur web sur Firebase Hosting et, si appli mobile , sur les store Play Store et App Store. Gestion du back end via les services Firebase (voir [Architecture applicative/technique](#))

Coûts

Voir les prix sur la page Firebase [ici](#).