

SURREL
Hugo

BTS SIO



Message

Sommaire

- **Situation**
- **Étude des solutions**
- **Contraintes**
- **Réalisation du projet**
- **Conclusion**

Situation

Aujourd'hui tous le monde possède un smartphone et s'en sert pour communiquer avec des proches ou bien sa famille à travers le monde entier. Pour cela il faut donc utiliser des applications de messagerie.

C'est pour cela que j'ai décidé de coder une application de messagerie instantanée. Elle permettra donc a tous les utilisateurs inscrits sur la plateforme de pouvoir discuter entre eux, s'envoyer des photos et des vidéos ou bien encore leur position géographique.

Ce projet n'est pas à but lucratif seulement fictif.

Dans ce projet je suis tous seul :

- Hugo SURREL Développeur Full-Stack.

Je me suis donc occupé du design ainsi que des fonctionnalités de l'application et de la connexion à la base de données.

J'ai codé cette application en Swift et je me suis aidé de plusieurs librairie tel que Firebase qui me permet de me connecter a ma base de données ou bien de MessageKit pour utiliser les messages et avoir un rendu proche de l'application Message d'Apple.

Étude des solutions

Pour le bon fonctionnement du projet il faut un smartphone Apple équipé de la dernière version d'iOS.

Le projet respectera le contexte de l'environnement technologique du BTS SIO, aucune dépense ne sera donc autorisée.

Le délai pour créer le projet ne devra dépasser la date butoir du projet fixé.

Les contraintes du projet sont la création et la modification d'une base de données compatible avec l'application.

L'application doit correspondre aux attentes fixée au préalable.

Contraintes

Je me suis fixé un budget de 0 euros pour ce projet car il est entièrement fictif.

Pour le développement j'ai utilisé le logiciel de code « Xcode ». J'ai pris ce logiciel dans un soucis de coût de par sa gratuité.

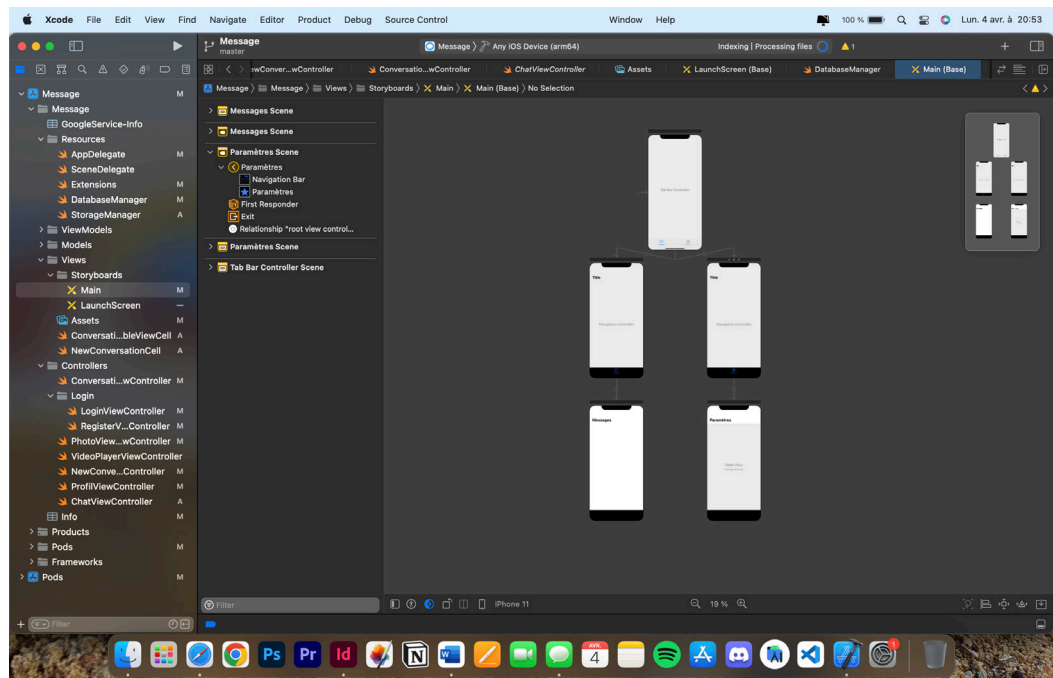
Pour visualiser mon projet je peux donc lancer un émulateur directement sur mon MacBook ou bien connecter mon iPhone et m'en servir comme émulateur.

C'est ce dernier choix que j'ai fais car je peux emporter l'application avec moi et je peux la tester de partout a n'importe quel moment.

Réalisation du projet

J'ai donc commencé par mettre en place un StoryBoard sur Xcode pour donner le design global de mon application avec plusieurs pages :

- La page d'accueil «Message», c'est ici que les messages s'afficheront lorsque l'utilisateur recevra un message ou débutera une conversation.
- Une page secondaire «Paramètres», ici l'utilisateur pourra voir les informations de son profil et se déconnecter.



Après avoir fait le StoryBoard global j'ai fait le StoryBoard de l'écran de lancement pendant lequel le logo de l'application va s'afficher jusqu'à ce que l'application se lance. Pour faire le StoryBoard j'ai utilisé le même principe que pour le StoryBoard global.

Ensuite j'ai effectué la connexion a la base de données pour pouvoir stocker mes comptes d'utilisateurs. Pour la base de données j'ai utilisé Firebase qui me permet de gérer facilement et simplement les utilisateurs ainsi que l'envoi des messages en temps réel plus tard.

```
private let database = Database.database(url: "https://messageapp-a416f-default-rtdb.europe-west1.firebaseio.com").reference()
```

Du côté de la base on peut donc voir la liste des utilisateurs qui ont été créé.

Identifiant	Fournisseurs	Date de création	↓	Dernière connexion	UID utilisateur
gogos200242@gmail.com	✉	30 mars 2022		30 mars 2022	gcWkwVICuseQ1yQtqCNcTNw9K...
a.tantan@gmail.com	✉	28 mars 2022		30 mars 2022	Bmsowl28bhYRSt6ayJthTltapx2

Pour ce qui est du visuel sur l'application lors du premier lancement un formulaire s'affiche nous proposant de nous connecter mais si nous ne sommes pas encore inscrits un autre formulaire nous est proposé pour nous inscrire à l'application et créer son compte. Il nous faut alors renseigner notre nom, prénom, adresse mail, mot de passe et une photo de profil.

21:33 4G 69%

< Se connecter S'inscrire

Prénom

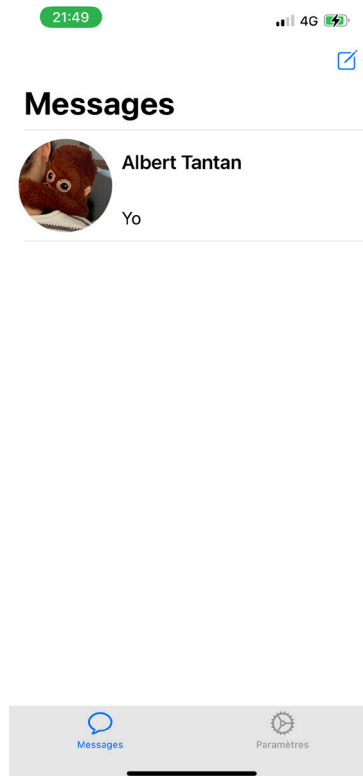
Nom

Adresse mail

Mot de passe

S'inscrire

Une fois que l'utilisateur à créer son compte il est redirigé vers la page d'accueil où ces messages seront affichés et où il peut créer de nouvelles conversations avec des utilisateurs de l'application.



Depuis cette page l'utilisateur peut donc créer un nouvelle conversation en appuyant sur le petit logo bleu en haut à droite de l'écran. Au niveau du code pour créer la nouvelle conversations je vérifie qu'il n'a pas déjà une conversation existante avec l'utilisateur sélectionné au quel cas je le redirige sur cette conversation

```
private func createNewConversation(result: SearchResult) {
    let name = result.name
    let email = DatabaseManager.user(email: result.email)

    DatabaseManager.shared.conversationExists(with: email, completion: { [weak self] result in
        guard let strongSelf = self else {
            return
        }
        switch result{
            case .success(let conversationId):
                let vc = ChatViewController(with: email, id: conversationId)
                vc.isNewConversation = false
                vc.title = name
                vc.navigationItem.largeTitleDisplayMode = .never
                strongSelf.navigationController?.pushViewController(vc, animated: true)
            case .failure(_):
                let vc = ChatViewController(with: email, id: nil)
                vc.isNewConversation = true
                vc.title = name
                vc.navigationItem.largeTitleDisplayMode = .never
                strongSelf.navigationController?.pushViewController(vc, animated: true)
        }
    })
}
```


L'utilisateur peut donc envoyer un message et il sera automatique enregistré sur la base de données et l'utilisateur qui doit recevoir le message pourra le voir en temps réel.

```
guard let conversationId = conversationId, let name = self.title else{
    return
}
DatabaseManager.shared.sendMessage(to: conversationId, otherUserEmail: otherUserEmail, name: name, newMessage: message, completion: { success in
    if success{
        print("message envoyé")
    }else{
        print("Erreur dans l'envoi")
    }
})
```

L'utilisateur peut aussi envoyer différent type de messages comme des photos ou des vidéos. Ces fonctions sont gérées à travers un switch qui me permet donc de connaître le type de message et d'y affecter les valeurs qui lui correspondent.

```
switch newMessage.kind{

case .text(let messageText):
    message = messageText
case .attributedText(_):
    break
case .photo(let mediaItem):
    if let urlString = mediaItem.url?.absoluteString{
        message = urlString
    }
    break
case .video(let mediaItem):
    if let urlString = mediaItem.url?.absoluteString{
        message = urlString
    }
    break
case .location(_):
    break
case .emoji(_):
    break
case .audio(_):
    break
case .contact(_):
    break
case .linkPreview(_):
    break
case .custom(_):
    break
}
```


Conclusion

Pour conclure cette application m'a permis de m'améliorer dans un domaine que je ne connaissais pas vraiment et donc d'apprendre à l'utiliser.

Ce projet m'a permis de m'améliorer en Swift et en UI pour que quelque soit la taille de l'application les informations s'affiche correctement.

J'ai beaucoup apprécié ce projet qui m'a apporté quelques problèmes auxquels j'ai su répondre pour améliorer mes compétences dans ce domaine.