HAROL HIGUERA
iOS DEVELOPMENT

TASK&SCHEDULE

This is an IOS mobile calendar application with tons of custom features for internal usage in organizations. This system does not contain a dedicated server but it interacts with Cloud Firestore and Firebase Authentication services for authentication, remote processing and storage.
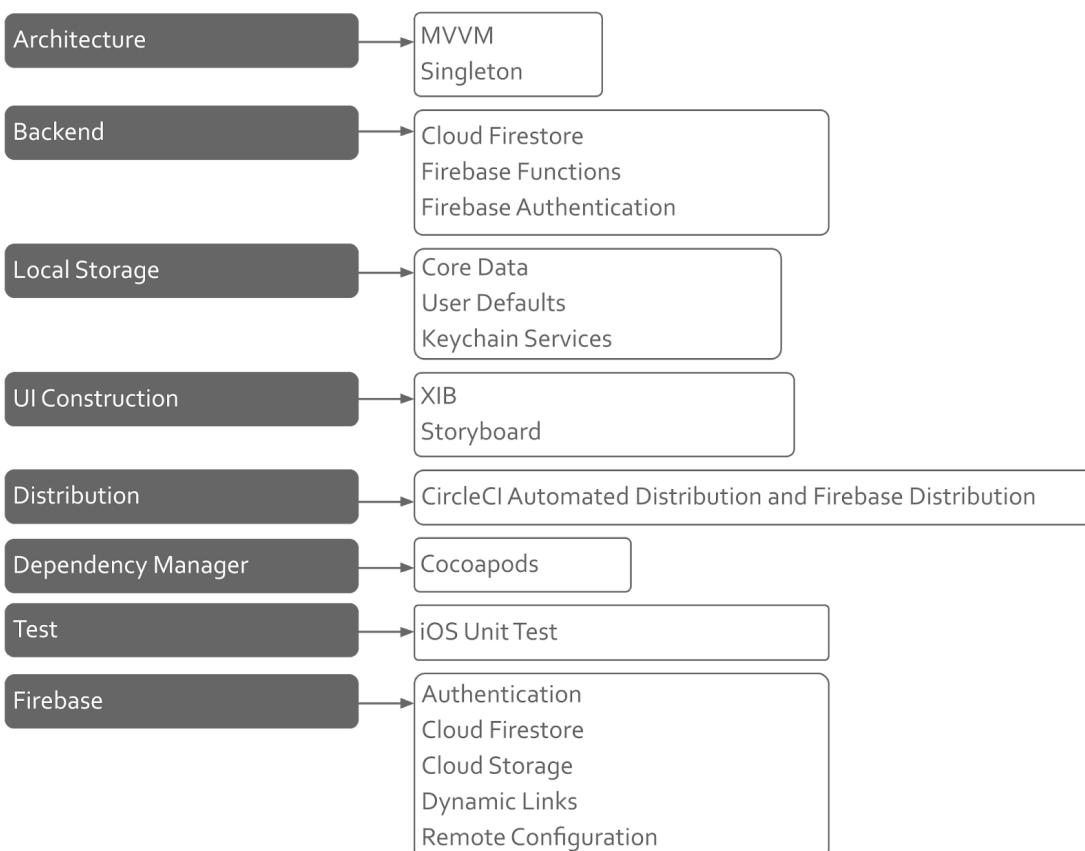
The app uses Core Data as an internal storage mechanism and iCloud to back-up certain data.

ROLE: iOS engineer
LANGUAGE: Swift
COUNTRY: Japan
DEVELOPMENT: 2019/09 – 2020/10

APP STORE:
https://apps.apple.com/jp/app/task-schedule/id1188240518

TECH STACK

Architecture	MVVM Singleton

Backend	Cloud Firestore Firebase Functions Firebase Authentication

Local Storage	Core Data User Defaults Keychain Services

UI Construction	XIB Storyboard

Distribution	CircleCI Automated Distribution and Firebase Distribution

Dependency Manager	Cocoapods

Test	iOS Unit Test

Firebase	Authentication Cloud Firestore Cloud Storage Dynamic Links Remote Configuration

MISUMI ASIA

This app was developed for MISUMI GROUP INC, an international platform that engages in the manufacturing and distribution sectors providing parts procurement services.
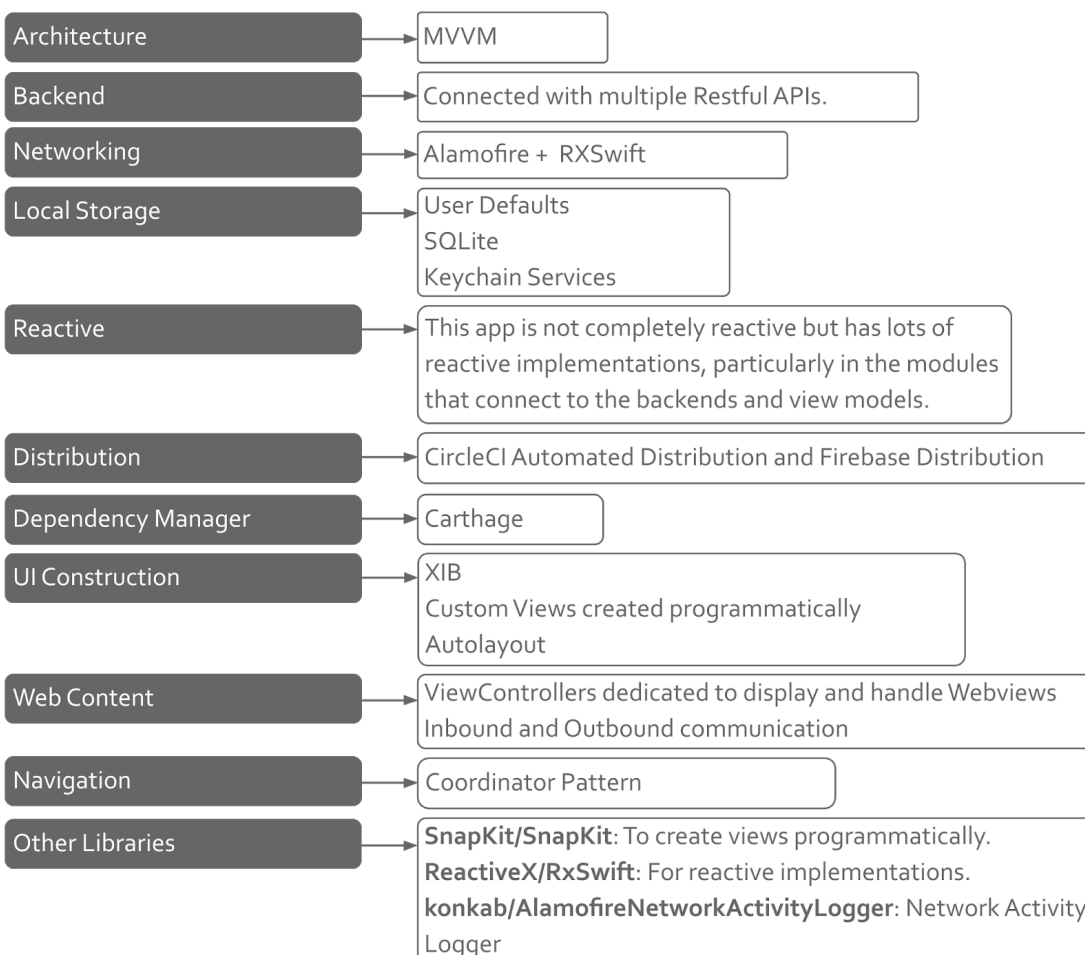
The app allows distributors to find parts, get quotations, or buy parts for their business. This app was developed for three countries in the same source code.

ROLE: Lead iOS Engineer
LANGUAGE: Swift
COUNTRY: Vietnam, Thailand, South Korea

DEVELOPMENT: 2018/10 – 2020/02
URL: Thai site. Vietnamese site.
APP STORE: KR, VN, TH

TECH STACK

Architecture	MVVM
Backend	Connected with multiple Restful APIs.
Networking	Alamofire + RXSwift
Local Storage	User Defaults SQLite Keychain Services
Reactive	This app is not completely reactive but has lots of reactive implementations, particularly in the modules that connect to the backends and view models.
Distribution	CircleCI Automated Distribution and Firebase Distribution
Dependency Manager	Carthage
UI Construction	XIB Custom Views created programmatically Autolayout
Web Content	ViewControllers dedicated to display and handle Webviews Inbound and Outbound communication
Navigation	Coordinator Pattern
Other Libraries	**SnapKit/SnapKit**: To create views programmatically. **ReactiveX/RxSwift**: For reactive implementations. **konkab/AlamofireNetworkActivityLogger**: Network Activity Logger

HAROL HIGUERA
iOS DEVELOPMENT

This application was developed for HITACHI and aimed to keep a close eye on the elders by displaying data taken from activity sensors installed in their homes and Hitachi appliances.

The UI of this app contains multiple graphs with chronological and real-time data retrieved from the backend side. These 2D-graphs were built with highly customized views and Android Canvas drawings. During development and testing, it was deployed with CircleCI automated deployment configuration and Firebase Distribution service.

ROLE: Lead iOS Engineer
LANGUAGE: Swift
COUNTRY: Japan

DEVELOPMENT: 2019/01 – 2019/07
URL: https://kadenfan.hitachi.co.jp/doshiteru/
APP STORE: apps.apple.com/jp/app/ ドシテル /id1452488575

TECH STACK

Architecture	MVVM
Backend	Connected to Restful API
Networking	Moya + RxSwift
Local Storage	Keychain Services
UI Construction	Storyboard Autolayout
Distribution	Testflight
Dependency Manager	Cocoapods
In App Purchases	Non-Consumable Receipts locally validated with OpenSSL library
Firebase	Cloud Messaging
Other Libraries	**danielgindi/Charts**: To create Charts.

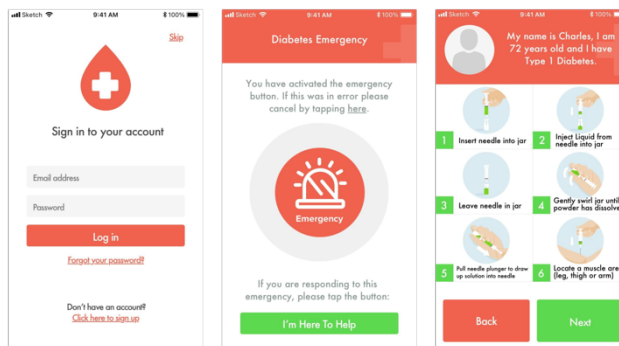**DIABETES EMERGENCY**

This is an app that could help a total stranger save the life of someone suffering severe symptoms of Diabetes on the spot. It has a mechanism so the patient could trigger an alarm sound so people around could notice and provide assistance.

It also contains an interface with guidance on how to proceed when a person is suffering from severe symptoms.
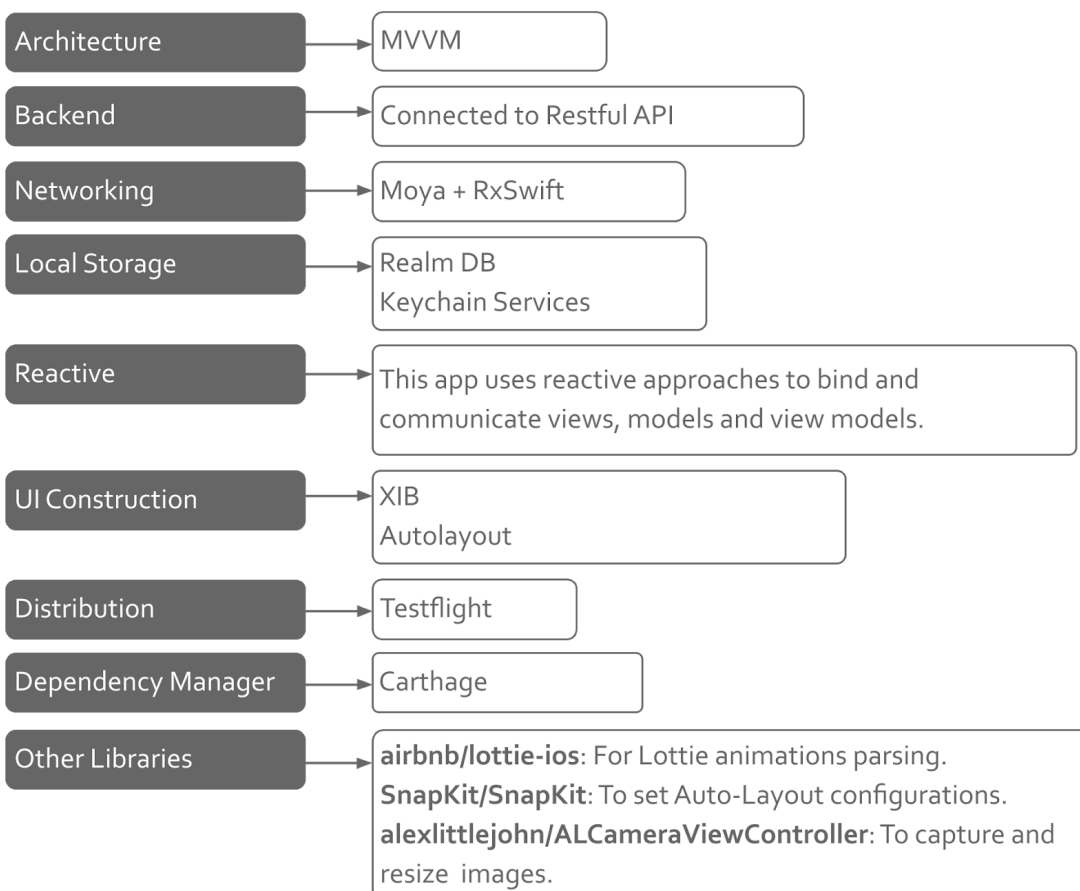
ROLE: Single iOS engineer
LANGUAGE: Swift
COUNTRY: Australia
DEVELOPMENT: 2019/05 – 2019/11

Due to the Coronavirus pandemic this app hasn't been released.

Facebook Updates:
https://www.facebook.com/DiabetesEmergencyApp/

TECH STACK

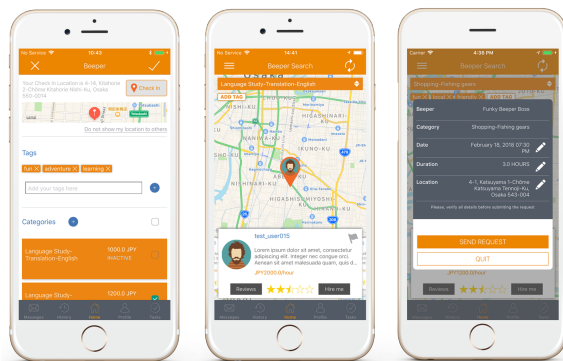|	
Architecture	MVVM
Backend	Connected to Restful API
Networking	Moya + RxSwift
Local Storage	Realm DB Keychain Services
Reactive	This app uses reactive approaches to bind and communicate views, models and view models.
UI Construction	XIB Autolayout
Distribution	Testflight
Dependency Manager	Carthage
Other Libraries	**airbnb/lottie-ios**: For Lottie animations parsing. **SnapKit/SnapKit**: To set Auto-Layout configurations. **alexlittlejohn/ALCameraViewController**: To capture and resize images.

4

HAROL HIGUERA
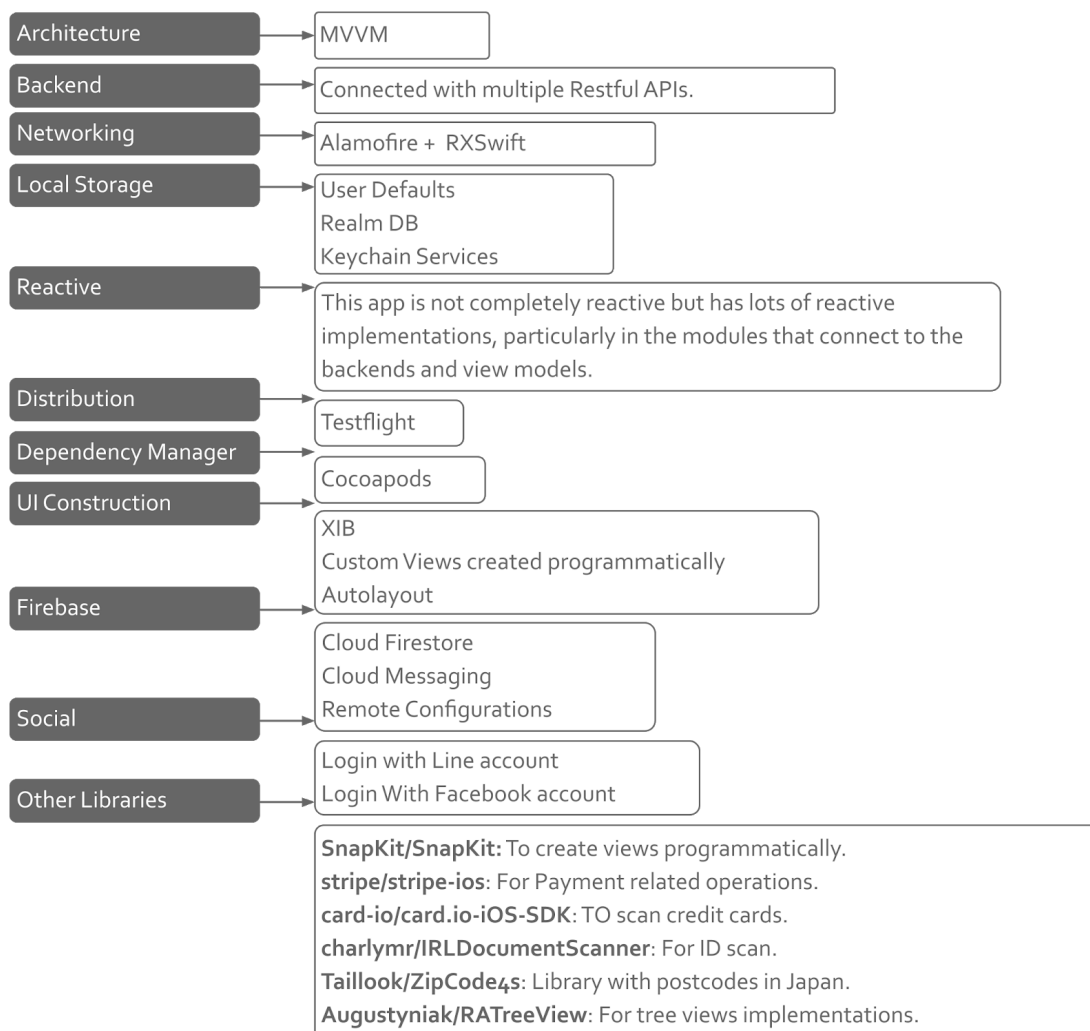## iOS DEVELOPMENT

BEEPNOW APP

A mobile application that connects "service seekers" and "service providers" together, allowing people to earn income on a freelance basis. The application behaves differently whether the User is a seeker or a provider. Users who become providers guide to follow a more exhaustive registration process including a "Know your Customer" flow and entry of Bank Account details to receive payments. It all is handled by backend services and Stripe Services. Although the application was initially released to allow payments with credit cards, it was intended to be functional for payments in cryptocurrency as well. The crypto-wallet system was developed in Malaysia and further integrated into the Beepnow system.

ROLE: Single iOS engineer
LANGUAGE: Swift
COUNTRY: Japan

DEVELOPMENT: 2017/04 – 2018/09
This app was removed from the App Store due to known Owners's reasons.

TECH STACK

Architecture	→	MVVM
Backend	→	Connected with multiple Restful APIs.
Networking	→	Alamofire + RXSwift
Local Storage	→	User Defaults / Realm DB / Keychain Services
Reactive	→	This app is not completely reactive but has lots of reactive implementations, particularly in the modules that connect to the backends and view models.
Distribution	→	Testflight
Dependency Manager	→	Cocoapods
UI Construction	→	XIB / Custom Views created programmatically / Autolayout
Firebase	→	Cloud Firestore / Cloud Messaging / Remote Configurations
Social	→	Login with Line account / Login With Facebook account
Other Libraries	→	

SnapKit/SnapKit: To create views programmatically.
stripe/stripe-ios: For Payment related operations.
card-io/card.io-iOS-SDK: TO scan credit cards.
charlymr/IRLDocumentScanner: For ID scan.
Taillook/ZipCode4s: Library with postcodes in Japan.
Augustyniak/RATreeView: For tree views implementations.
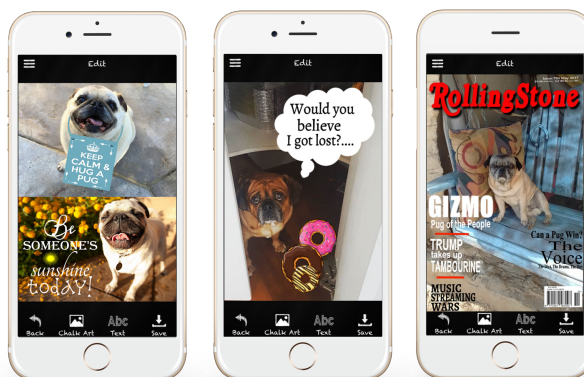
HAROL HIGUERA
iOS DEVELOPMENT

PUG CHALK

An almost standalone application made for pug's lovers who want to share customized pictures of their pugs by adding and customizing stickers from collections that can be free or paid.

This was the first application I developed for iOS.

Although this app does not have a dedicated server its backend mechanism consists of images and JSON configuration files stored in a AWS S3 bucket. In this way stickers can be updated by the application's Owner. Unfortunately

ROLE: Single iOS engineer
LANGUAGE: Swift
COUNTRY: Australia
DEVELOPMENT: 2016/10 – 2017/05

INSTAGRAM:https://www.instagram.com/pug_chalk/
APP STORE:
https://itunes.apple.com/us/app/pug-chalk-sticker-photo-editor/id1234105875?mt=8

TECH STACK

Architecture	MVC
Backend	AWS S3
Local Storage	User Defaults
UI Construction	Custom Views High usage of UIGestureRecognizer
Dependency Manager	Cocoapods
Social	This app was made so User can share their pics on social platforms such as Facebook, Twitter, Instagram and Pinterest. Also by email, by SMS or other available methods.
Other Libraries	**onevcat/Kingfisher**: For downloading and caching images. FacebookSDK: For social sharing.