

# 포트폴리오

## 기술 스택

```
gantt
dateFormat YYYY-MM
section Language
Swift :i3, 2018-03, 2025-01
typeScript :a1, 2024-10, 2024-12
typeScript :a1, 2019-04, 2020-01
Objective-c :l5, 2015-08, 2019-02
MFC(c++) :l3, 2014-03, 2015-04
SQL :l2, 2012-09, 2014-01
Android(Java) :l1, 2011-09, 2016-06

section Architecture
MVVM :a1, 2020-05, 2025-01
TMA: a5, 2024-01, 2025-01
MVI :a2, 2022-07, 2025-01
Clean Architecture :a1, 2022-01, 2025-01
VIPER: a4, 2020-06, 2022-06
Ribs: a4, 2020-06, 2022-09
MVC : a3, 2015-05, 2020-12

section UI
SwiftUI : c1, 2023-07, 2024-12
PinLayout/FlexLayout : u3, 2022-03, 2022-07
SnapKit : c3, 2020-01, 2025-01
AutoLayout : c2, 2016-07, 2025-01
Autoresizing Mask : u2, 2015-09, 2017-03
StoryBoard : u1, 2015-09, 2023-01

section Asynchronous
Async/Await: r3, 2022-07, 2025-01
Combine: r2, 2020-02, 2025-01
RxSwift: r1, 2019-07, 2025-01

section Analytics/Ad
AirBridge : s3, 2023-01, 2024-12
Amplitude : s2, 2022-03, 2024-12
Braze : s2, 2021-07, 2024-12
Google Ads: aa2, 2019-11, 2021-06
IGAWorks : aa1, 2017-10, 2021-06
FireBase(GoogleAnalytics) : s2, 2016-12, 2025-01

section Tools
Xcode-cloud: t6, 2022-06, 2025-01
Github-Action: t6, 2022-01, 2025-01
Slack: t5, 2021-07, 2024-12
```

```
Jira: t4, 2021-02, 2024-12  
Figma: t3, 2020-07, 2025-01  
Zeplin: t2, 2018-07, 2025-01  
Git: t1, 2016-06, 2025-01  
SVN: t0, 2011-09, 2019-02
```

## 경력 사항

```
gantt  
dateFormat YYYY-MM  
section 정규직  
비주얼 :i3, 2021-07, 2024-12  
하나모바일 :i2, 2018-01, 2021-06  
새움테크 :i1, 2015-05, 2018-01  
유니텍 : c1, 2014-07, 2015-04  
section 인턴/계약직  
강원대학교 :a2, 2011-09, 2013-03
```

## 비주얼 (21.07 ~ 24.12)

아몬즈 아키텍처 개선 Tuist 적용 및 TMA 구조 확립 (24.04 ~ 24.12)

역할

- iOS 개발 및 리딩 (참여인원: iOS 3)

주요 기술

- [Tuist](#), [The Modular Architecture \(TMA\)](#), [Swinject](#)

도전 과제 및 해결 방법

- Application / Feature / Domain / Core / Shared 계층 분리
- 모듈별 구조 확립 및 테스트 코드 작성 유도
- 모듈생성시 스크립트를 통해 동일한 구조로 자동 생성

```
flowchart TD  
FeatureExample --> Feature  
FeatureExample --> FeatureTesting  
Feature --> FeatureInterface
```

```
FeatureTests --> Feature
FeatureTests --> FeatureTesting
FeatureTesting --> FeatureInterface
```

## 성과/지표

- CI 빌드테스트 속도 50% 감소(12분 -> 5분)
- 개발속도 10% 증가(모듈별 빌드)
- 테스트 커버리지 13% 증가
- 머지 컨플릭 비율 감소(.xcodeproj/\* ignore)
- [관련 블로그](#)

## 배운점/아쉬운점

- 명확한 계층정의로 협업 멤버 공통된 형태의 모듈 개발
- 모듈별 테스트가 추가되어 안정성 향상
- project 파일이 더이상 추가되지 않아 코드 형상 관리가 간편해짐
- 모듈별 interface로 더 명확한 캡슐화
- 클린아키텍처, cocoapods -> SPM으로 모두 변경, 모듈 레이어 등 선행이 필요한 업무가 늦어지며 너무 늦게 반영한 데에 대한 아쉬움
- 마이그레이션 하며 기존 코드도 지우는 프로세스를 확립하지 못함
- 80% 이상 모듈로 대체하려 했으나 회사 사정으로 모두 완수하지 못한데에 대한 아쉬움

## App팀 배포절차 도입 (24.07 ~ 24.12)

### 역할

- 스쿼드 리더 (참여 인원 : Android 1, iOS 1, PO 1, QA 1)

### 주요 기술

- [Trunk Based Development\(TBD\)](#), Feature Flag, Jira Automation, Slack

### 도전 과제 및 해결 방법

- 배포 관련 일정 논의 및 결정
  - 매주 1회
  - 검수 프로세스 월~목
  - 심사 제출 금
  - 실 배포 화요일
- 배포 자동화
  - FeatureFlag 상태가 개발중 -> 작업완료시 자동 내부 배포
  - PO / QA 슬랙을 통해 배포 앱 생성 기능 제공
  - 금요일 오전 최종 확인용 앱 자동 내부 배포
- 슬랙 리마인더
  - 작업 담당자 가이드 문구 자동 노출
  - PO / QA 내용 공유
  - 포함되는 기능(티켓 + 피쳐플래그) 내용 공유

- 지라 상태 자동화
  - 개발 완료시 상태 변경 자동화
  - 담당자 지정 자동화
  - (실) 배포 완료시 상태 변경 자동화

#### 성과/지표

- 주당 배포 횟수 21% 증가
- 슬랙 배포 관련 논의 15% 감소
- 피쳐플래그를 통해 잘못된 배포 disable 처리 기능 제공
- [관련 블로그](#)

#### 배운점/아쉬운점

- 유연한 CI/CD를 토대로 빠른 적용
- 하나의 팀으로 정확한 내용공유
- Store 심사제출까지 자동화 하지 못한 아쉬움

## Swagger -> Swift 파일 전환 자동화 (23.05 ~ 23.06)

### 역할

- 메인 개발 (참여 인원 : iOS 1)

#### 주요 기술

- Swift, OpenAPISpec 3.0, RestfulAPI, Moya, Alamofire, yaml

#### 도전 과제 및 해결 방법

- OAS 3.0기준 Swagger 문서를 UserAPI모듈로 생성하기
  - SwaggerCodeGen은 단순 DTO만 생성해서 탈락
  - 직접 대체할수 있도록 Swift로 제작
- 네트워크 모듈 형식 결정 필요
  - TMA 외부 노출 인터페이스 생성
  - 내부 형태는 Moya 형식으로 사용하기로 결정
  - Rx 비율을 낮추기 위해 Asyn/Await으로 결정
- 팀원 동일한 형식 사용
  - 별도 앱으로 생성해 작업시 변경되면 담당자가 덮어씌우도록 결정

#### 성과/지표

- API관련 이슈 90% 감소
- 디코딩 에러 파악율 100% 달성
- 작업속도 10% 이상 증가
- API관련 논의 30% 감소
- [관련 블로그](#)

#### 배운점/아쉬운점

- 사람은 실수를 할 수 있어도, 기계는 실수하지 않는다.

- 자동화된 테스트코드 생성이 안정성 증가에 큰 도움을 줌
- User API가 변경되면 Github에서 PR이 생성되서 대체하고 싶었으나 리소스 부족으로 포기
  - 당시 Feature Flag가 확립되지 않아, 서비스에 영향을 끼칠 우려가 있어 미룸
  - 파이썬으로 코드를 대체해 Github Action으로 만들 구상까지 하고 다른 업무로 포기

## Clean Architecture 도입 및 개선(23.02 ~ 23.12)

역할

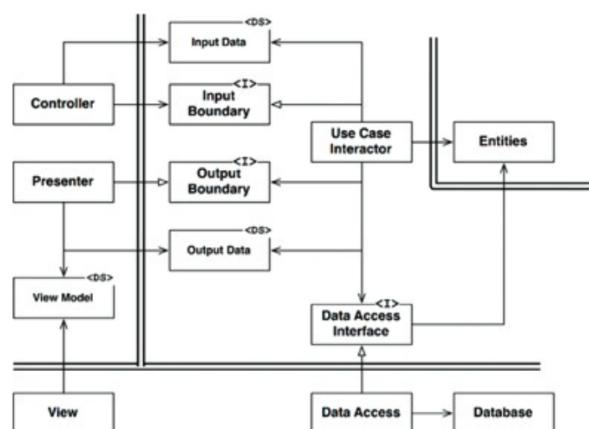
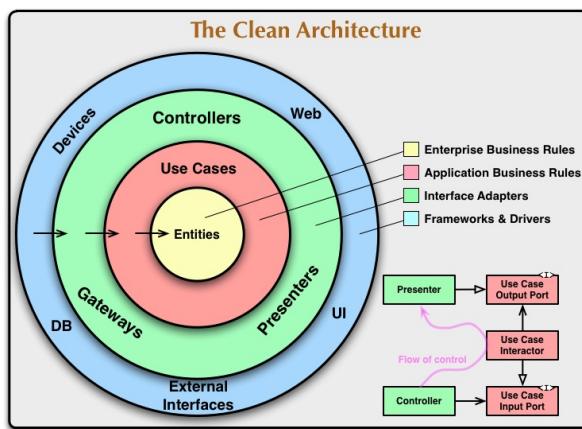
- App 팀 리드 (참여 인원 : Android 2, iOS 3, PO 1, QA 1)

주요 기술

- Clean Architecture, Snapshot Test

도전 과제 및 해결 방법

- 클린아키텍처에 대한 전반적인 지식 부족
  - 스터디 진행 및 지속적 가이드 실시
- 전체 프로젝트에 대한 적용의 어려움
  - ViewModel 마이그레이션이 완료된 부분에 한해 적용 시작
  - ViewModel에서 Domain/Data 부분 분리를 우선적으로 시작
- 명확한 계층 분리에 대한 의견 대립
  - PO가 이야기하는건 Domain, Back-end가 이야기하는건 Data, UX가 이야기하는건 Presenter로 단순화해 내용 공유
  - 비지니스 로직에 대해, 크게 바꿔지 않으면서도 중요한 부분이란 인식을 학습시킴
- 마이그레이션이 부족한 화면의 경우 DTO에 너무 강하게 묶임
  - DTO와 ViewState의 분리에 대해 강조 후 서로 다른 인지시키도록 학습
  - 스냅샷 테스트를 통해, ViewState로 대체해가면서도 안정성있게 작업할 수 있도록 테스트코드 우선



성과/지표

- 버그율 5% 감소
- 작업속도 5% 증가
- Android, iOS 공통 도메인 로직 생성
- [관련 블로그](#)

배운점

- Data에 휘둘리지 않는 작업이 가능해짐

- Backend-Client 동시 작업이 가능해짐
- 개발자간의 대화보다도 중요한 Domain의 중요성에 대해 팀적 공유
  - 추가) Domain Driven Design(DDD) 스터디 추가 진행
  - 추가) Design First 도입

## ReactorKit을 활용한 ViewModel 마이그레이션 (22.03 ~ 22.12)

### 역할

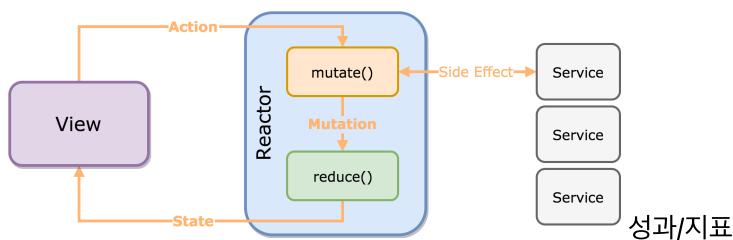
- iOS 파트 리드(참여 인원 : iOS 2)

### 주요 기술

- ReactorKit, MVVM, RxSwift, Action/State

### 도전 과제 및 해결 방법

- Input/Output 패턴보다 더 명확한 Action/State로 변경
- 



- 전체 페이지의 60% ReactorKit으로 대체
- ReactorKit으로 대체한 화면의 Action당 테스트코드 0.7개 생성
- 테스트커버리지 15% 향상
- 버그 발생율 30% 감소

### 배운점/아쉬운점

- Action -> State로 단방향 흐름이 되며, 복잡한 기능은 없음을 다시 한번 경험
- 페이지별 분리가 아닌, 컴퍼넌트별 모듈 분리로 진행했다면 더 빠른 모듈 분리가 진행되었겠으나, iOS만 진행되어 페이지 단위로 한정
  - 추후 모듈별 상세 분리를 진행하며 추가 분리 가능해짐
- Android, iOS, (mobile)Web 모두 비슷한 로직임에도 서로 대화없이 진행됨에 대한 아쉬움
  - Reactive Native, Flutter의 도입에 대해 논의 했으나 미뤄짐
- 

## MVC -> MVVM 개선 (21.10 ~ 22.01)

### 역할

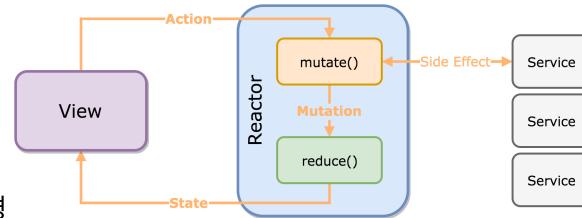
- iOS 파트 리드(참여 인원 : iOS 2)

### 주요 기술

- MVVM, [input/output](#) 패턴

## 도전 과제 및 해결 방법

- MVC로 화면에 있는



- Input/Output 패턴보다 더 명확한 Action/State로 변경

## 성과/지표

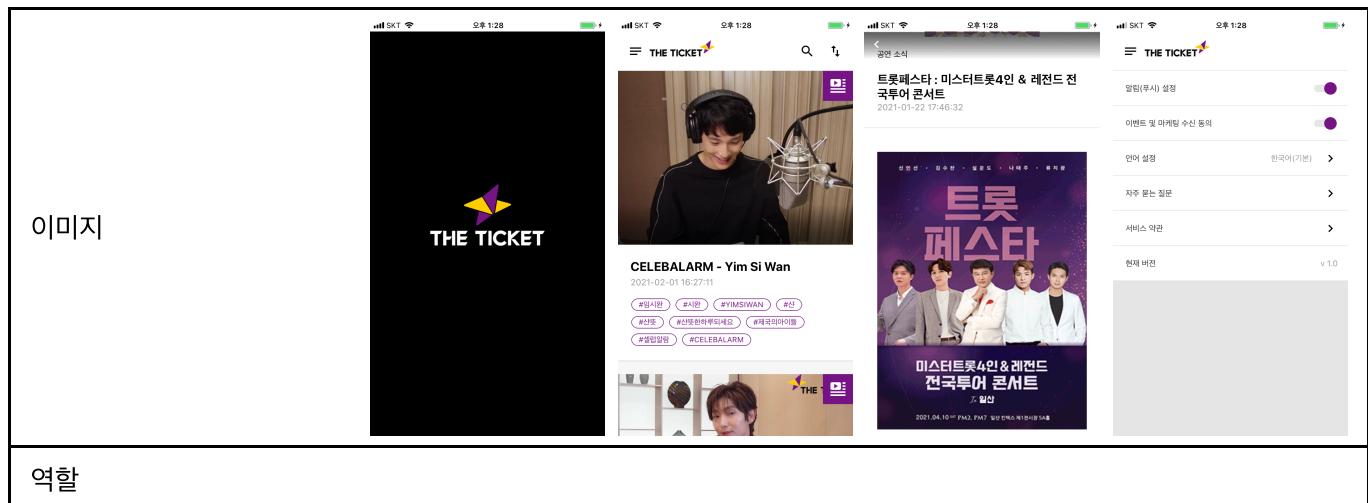
- 3개의 페이지에 적용
  - 추후 ReactorKit으로 대체
- Crash율 0.5% 상승
- 버그율 10% 상승

## 배운점/아쉬운점

- 업무 파악 전 무리한 적용으로 안정성 하락
- 테스트 코드 필요성 확인
- 팀원간 협업 및 내용 공유 부족함을 실감 이를 위한 명확한 아키텍처/패턴 통일 필요성 확인

## 하나모바일 (18.01 ~ 21.06)

### 더티켓



- iOS 메인 개발자 (참여 인원 : iOS 1)

## 주요 기술

- Alamofire, Firebase, RxSwift, RxCocoa, RxGesture

## 도전 과제 및 해결 방법

- 티켓 예약 내역 및 이벤트/관람 정보를 확인 가능
- 별도 상세 접수화면에서 추가 사항 기입 가능
- SNS를 통한 간편 로그인 기능 제공
- Rx를 활용한 MVVM구조 프로젝트 작성

## 흔들어대리운전 (20.10 ~ 20.11)

이미지

흔들어대리운전 이용내역

1588-3535

#1 접수 내역

서울특별시 영등포구 영등포동 618-501 스크린

상당원 접수	어플 접수*	퀵 접수*	카드 관리
10% 적립	현금 15%	현금 15%	카드 10%

이용내역

배자 10,000원

출발 서울특별시 동대문구 원산로43길 82  
도착 서울특별시 동대문구 원산로43길 82

경유 서울특별시 종로구 원선로43길 82  
서울특별시 종로구 원선로43길 82  
서울특별시 종로구 원선로43길 82

분의 10,000원

출발 서울특별시 동대문구 원산로43길 82  
도착 원점

배자 10,000원

출발 서울특별시 영등포구 영등포동 618-501  
도착 서울특별시 영등포구 영등포동 618-501

도착지 입력

영등포역1번출구(서울특별시 영등포구 영등포동 618 ...  
서울특별시 영등포구 영등포동 618-509  
서울특별시 영등포구 영등포동 618-509  
서울특별시 영등포구 영등포동 618-509

경유지 입력

2종으로 2종스티 1종으로 1종스티

현금 15%적립 카드 15%적립 마일리지

예상 요금 없음 한여 마일리지 10,000원

역할

- iOS 메인 개발자 (참여 인원 : iOS 1)

## 주요 기술

- Alamofire, NMapsMap(네이버맵), Firebase
- RxSwift, RxCocoa, RxGesture

## 도전 과제 및 해결 방법

- 위치좌표 및 모션 캡쳐를 활용한 대리운전 앱
- 네이버 맵을 이용한 현재 위치 표현
- 별도 상세 접수화면에서 추가 사항 기입 가능
- Rx를 활용한 MVVM구조의 프로젝트 작성
- 위치좌표의 정확성에 따른 알럿창 구분

## 두줄운세

이미지

오늘운세 2020.07.10

68 | 운전이 잇달아 달려 들어날 예후입니다.

장혜원님의 운세 상세보기

혈액형 연예인 특별운세 행운의번호 공항

내정보 박창규 1988.04.04 대비난시간 : 월 09:30 ~11:30

상담 실시간 운세상담

상당기록

상담사행정

기능 알림 설정

전체서비스

고객센터 공지사항

이벤트

1:1문의

자주찾는 질문

역할

- iOS 메인 개발자 (참여 인원 : iOS 1)

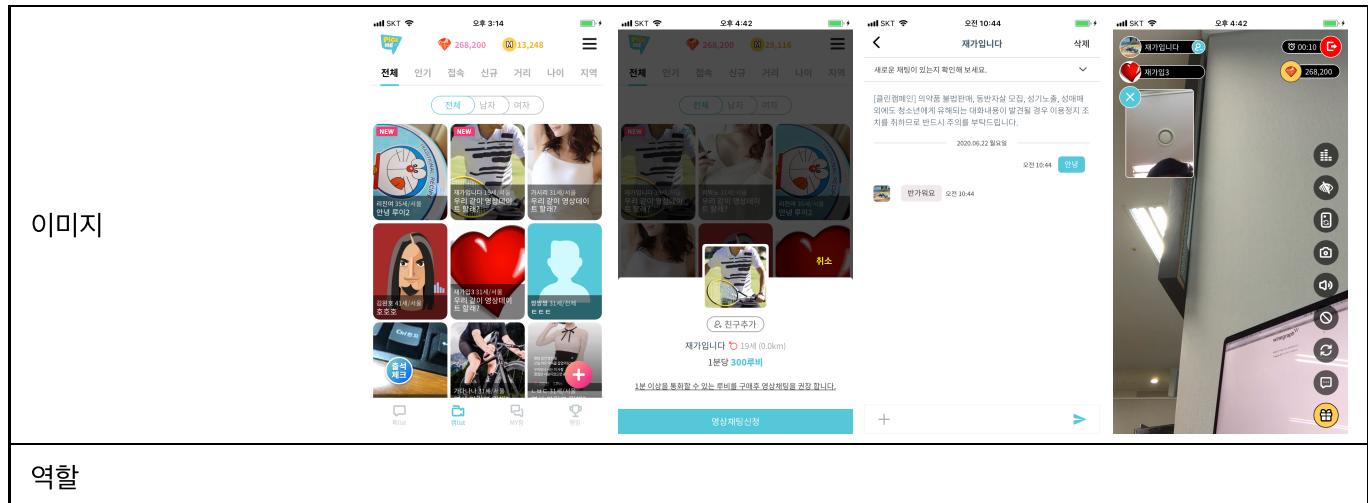
## 주요 기술

- Alamofire, Socket.IO-Client-Swift, Firebase, Kingfiser
- RxSwift, RxCocoa, RxGesture

## 도전 과제 및 해결 방법

- 운세 상담 및 전화상담 서비스 제공 App
- 메인 페이지 및 회원가입을 네이티브로 제작
- 그외 추가 상세페이지는 웹뷰로 이동

## Pickmecam / 위픽캠



- iOS 메인 개발자 (참여 인원 : iOS 1)

## 주요 기술

- Alamofire, Socket.IO-Client-Swift, Firebase, SVProgressHUD, Toask-Swift
- StoreKit IgaworksCore, AdPopcornOfferwall, TwilioVideo, AdBrixRemastered

## 도전 과제 및 해결 방법

- 1:1 실시간 동영상 채팅 지원 App
- 채팅과 동영상채팅 등의 실시간 지원을 위한 소켓 통신 지원
- SNS 로그인 지원 ( 페이스북, 구글, 카카오톡, 네이버 )
- 광고 지원 ( Adbrix, Nas )
- 인앱 구매 지원
- 동영상 채팅은 Twilio를 사용해 지원
- 소켓통신을 통해 실시간 채팅, 음성채팅 요청 및 연결 실시간 지원

## 더스타 온라인(iOS/Web)





역할

- iOS 메인 개발 및 web 개발

주요 기술

- typescript, Angular2

도전 과제 및 해결 방법

- 더스타 매거진
- Angular2로 제작

## Enjoy J Tour(iOS/Web)

이미지



역할

- iOS 메인 개발 및 web 개발

주요 기술

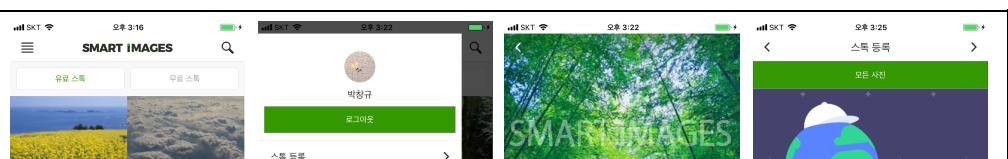
- typescript, Angular2

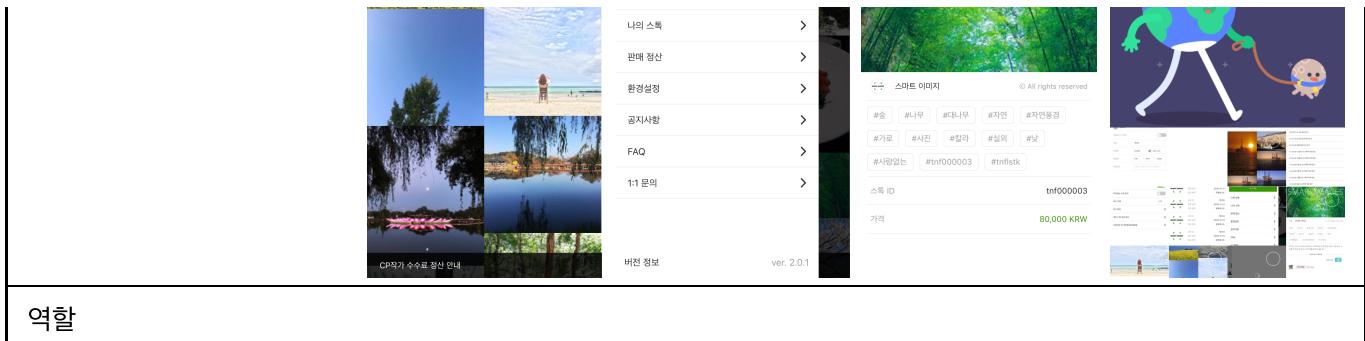
도전 과제 및 해결 방법

- 일본 내 숙소 예약용 페이지
- Angular2로 제작
- 예약의 전반적인 기능 수행

## 스마트이미지

이미지





역할

- iOS 메인 개발자 (참여 인원 : iOS 1)

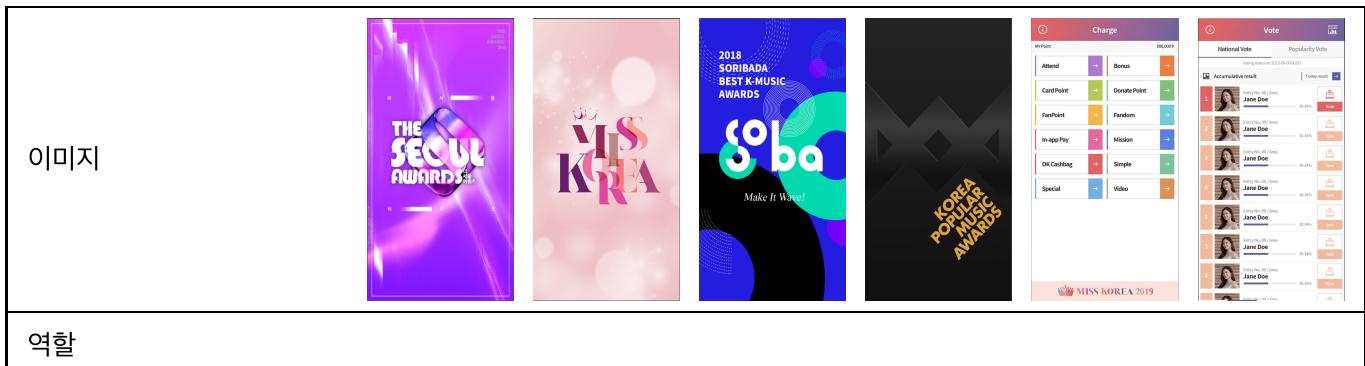
## 주요 기술

- Alamofire, Firebase(Core, Messaging)
- MXParallaxHeader, MXSegmentedPagerAdapter, Kingfisher

## 도전 과제 및 해결 방법

- 유료이미지 등록 및 판매를 위한 앱
- 스마트폰에 있는 이미지 등록 지원
- 이미지 크기 조절을 위한 이미지 프로세싱 기능 지원
- 이미지 설명을 위한 태그 기능 지원
- 서버 부담을 줄이기 위해 최대 이미지 고정 및 자동 사이즈 조정
- 이미지별 사이즈에 맞게 보여주기 위한 이미지 리사이즈 리스트 반영 지원

## 시상식 투표앱



역할

- iOS 메인 개발자 (참여 인원 : iOS 1)

## 주요 기술

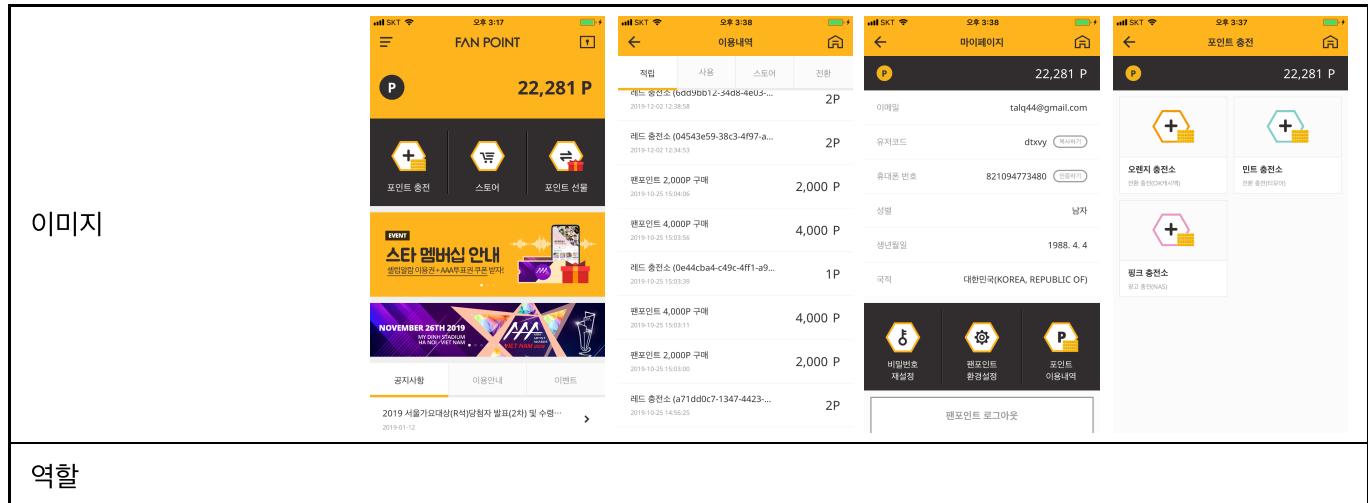
- Alamofire, RealmSwift, Firebase
- GoogleSignIn, TwitterKit, Facebook
- StoreKit, IgaworksCore, AdPopcornOfferwall

## 도전 과제 및 해결 방법

- 더서울어워즈, 미스코리아, 서울가요대전, 소리바다, KPMA
- 각 시상식별 투표를 위한 앱
- 지역화 지원 ( 한국어, 영어, 중국어 간체/번체 )
- 시상식 후반 100만건 이상 접속을 위한 서버 통신 최적화

- 인앱 결제 지원, 광고 지원 ( AdPopcorn, Adbrix, Tapjoy )
- 타 플랫폼과 포인트 교환을 위한 Scheme 지원
- UI의 변경을 쉽게 변경하기 위해 Design System화
- 공통 모듈화 진행
- 파이어베이스 DB를 별도 구축해 실시간 인앱구매 내역을 저장

## FanPoint



- iOS 메인 개발자 (참여 인원 : iOS 1)

### 주요 기술

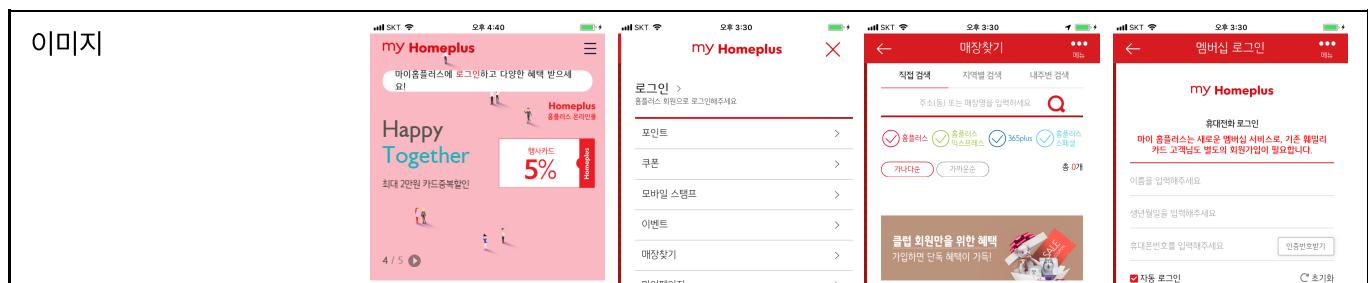
- Alamofire, Kingfisher, Firebase(Core, Messaging, Crash), RealmSwift
- StoreKit, TabjoySDK, IgaworksScore, AdPopcornOfferwall

### 도전 과제 및 해결 방법

- 시상식 앱과 연동해 포인트 교환 및 결제를 지원하는 앱
- 지역화를 통해 4개언어 지원( 한국어, 영어, 중국어 간체/번체 )
- 인앱결제 추가 및 인앱결제 재구매 프로세스 추가
- Swift 1달 학습 후 투입된 첫 프로젝트
- 각종 시상식과 연계 가능한 Scheme 모듈화 진행

## 새움테크 (15.05 ~ 18.01)

### マイホームプラス포인트





역할

- iOS 메인 개발자 (참여 인원 : iOS 1)

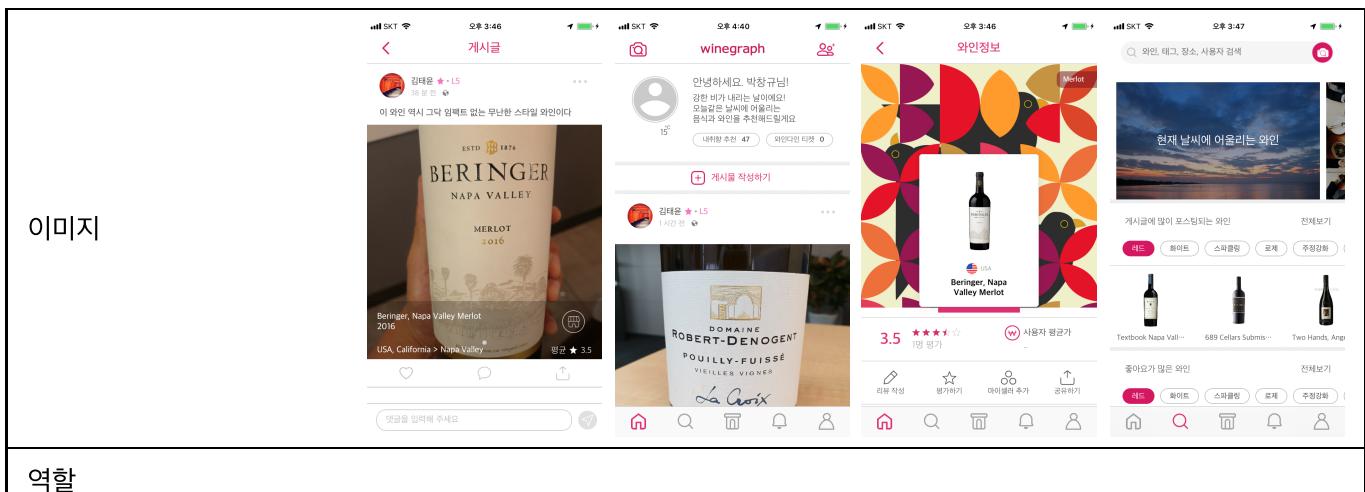
## 주요 기술

- AFNetworking, ZXingObjc, Firebase(Core, Messaging, Crash)
- Facebook(Core, Login, Share), Bolts, IgaworksCore

## 도전 과제 및 해결 방법

- 홈플러스 사용 고객 마일리지 적립을 위한 서비스 App
- 추후 관리 용이성을 위해 Webview로 제작
- 마일리지 적립을 위한 바코드 리더 추가
- 모바일 통신을 처음 경험한 서버개발자와 API 구축
- 기초 20개 지원 후 10개 API 추후 추가
- 업데이트를 최소화 하기 위해 스마트폰에 데이터를 저장/읽기 기능 지원

## 와인그래프



이미지

- iOS 메인 개발자 (참여 인원 : iOS 1)

## 주요 기술

- AFNetworking, Firebase(Core, Messaging, Crash)

## 도전 과제 및 해결 방법

- 와인 정보와 게시물을 확인할 수 있는 와인 SNS 종합 정보 앱
- 페이스북과 인스타그램의 장점만을 종합한 앱
- SNS 로그인 지원
- SNS 게시글 빠른 로드를 위해 API통신 추가 개발
- 와인 라벨 스캔을 통한 검색 기능 지원

## 나눔로또

이미지

역할

The screenshot shows the Naver NaverLotto app interface. It includes sections for lottery details, QR code scanning, and various lottery game categories like 마리풀, 토토볼, 스피드카노, etc. A sidebar on the right lists Seoul and surrounding city options.

- iOS 메인 개발자 (참여 인원 : iOS 1)

### 주요 기술

- AFNetworking, Realm, Firebase(Core, Messaging), MXParallaxHeader

### 도전 과제 및 해결 방법

- 나눔로또 정보 확인 및 판매점 정보 확인을 위한 App
- 서버 부하를 방지하기 위해 앱에 페이지 정보 데이터 저장
- 업데이트를 통해 웹페이지 데이터 다운로드 및 변경 지원
- QR Reader를 통해 복권 당첨여부 실시간 확인 가능
- 복권 판매점 정보를 스마트폰에 저장을 위해 Realm 사용
- 현재 위치 좌표 수신 및 실시간 사용
- Git을 통해 프로젝트 공유 경험

## 현대라이프 상담사용

역할

- iOS 메인 개발자 (참여 인원 : iOS 1)

### 주요 기술

- AFNetworking, Realm, Firebase
- SVProgressHUD, ZBarSDK, SDWebImage, SDVersion
- MDM, Applron, INISAFE, MVaccine, MTransKey

### 도전 과제 및 해결 방법

- 보험설계사가 고객 가입을 위해 사용하는 App
- 보험권 App이기에 보안 라이브러리 다수 사용
- iPhone/iPad 사이즈 모두 지원 ( 별개 스토리보드로 구현 후 지원 )
- Enterprise 계정을 사용한 앱 공유
- 공인인증서 지원

- 웹 + 네이티브 병합된 형태로 script통신 지원
- 프로젝트 이전 개발자가 1달 개발 후 나가 이어서 작업 시작
- Firebase Crashlytics 지원으로 앱 크래시 체크
- 실시간 디버깅을 위해 디버그 모드 및 로그 출력 화면 추가 지원

## CJ One Card

이미지

역할

- iOS 메인 개발자 (참여 인원 : iOS 1)

### 주요 기술

- Objective-c, WebView, Deeplink, APNs
- Server-Driven UI, Firebase, Mobile application Accessibility

### 도전 과제 및 해결 방법

- 앱 모바일 접근성을 위한 UI배치, 가이드 문구 연결, 포커싱 지정
- 서버드리븐한 홈 구성성을 위한 다이나믹 UI 적용
- 디자인 가이드 정의를 통한 값 변경 유동성 제공
- 버튼마다 Firebase Analytics 전송을 위한 컴파넌트 정의
- 그외 페이지는 관리 용이성을 위해 웹뷰를 사용
- 애니메이션, 속도 향상에 중점적 작업
- Voice Over 기능 확인

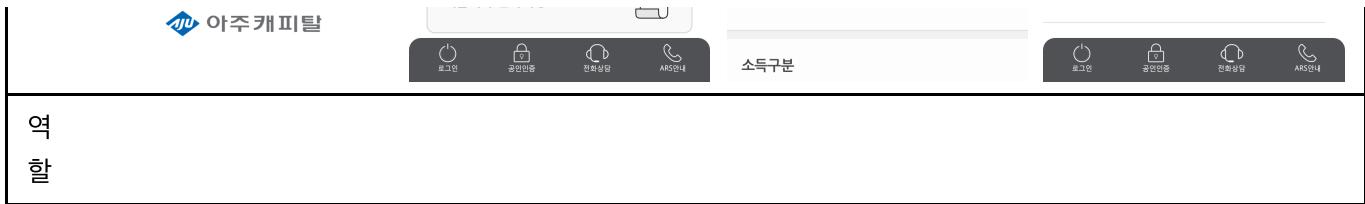
## 아주캐피탈

이  
미  
지  
아주좋은  
사람들이  
함께합니다.

개인 대출  
우수고객  
스마트대출  
신규고객  
다이렉트론  
내차 대출  
캠시스  
전기차 대출  
신차  
다이렉트  
중고차  
다이렉트  
대출계약 전자서명

STEP1. 실명인증 및 약관동의  
대출 신청을 위한 실명인증 및 약관동의 단계입니다.  
고객확인절차는 고객님의 신용등급에 영향을 주지 않습니다.

이름  
휴대폰 번호  
주민번호  
1 ② ③  
다이렉트  
다이렉트  
전화상담  
거래내역  
아주좋은  
멤버스  
간편비밀번호  
설정  
회사소개  
대표전화 1544-8600 개인대출 1688-0060  
상담시간 평일 09:00 ~ 18:00 (주말/공휴일 제외)



역

할

- iOS 메인 개발자 (참여 인원 : iOS 1)

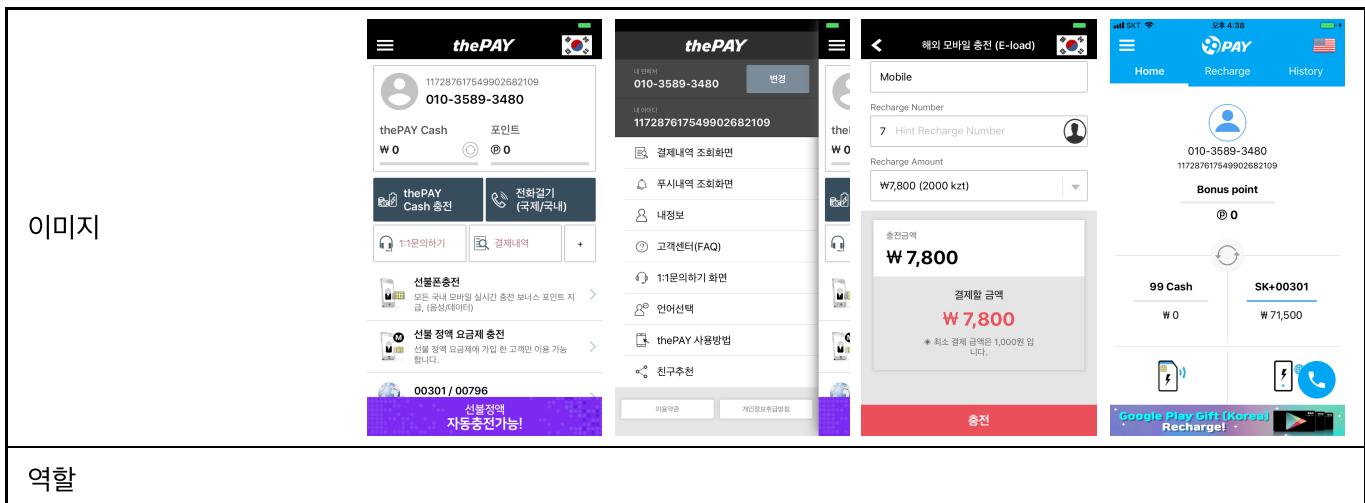
## 주요 기술

- Sanne, NAppProtect, MDM, NFilter, SmartAIBFramework
- KWPki, MagicSE, MagicXSign, MagicMRS

## 도전 과제 및 해결 방법

- 아주캐피탈 리뉴얼과 아주캐피탈 직원용 앱 개발
- 직원용의 경우 **Enterprise** 계정을 통해 내부 배포
- 관리의 용이성을 위해 Webview 사용률을 높임
- 은행권 앱이기에 다양한 보안 모듈 사용 경험
- 와이파이가 지원되지 않아 폐쇄적인 상황에서도 정상적인 개발 실시

## thePay / 99Pay



- iOS 메인 개발자 (참여 인원 : iOS 1)

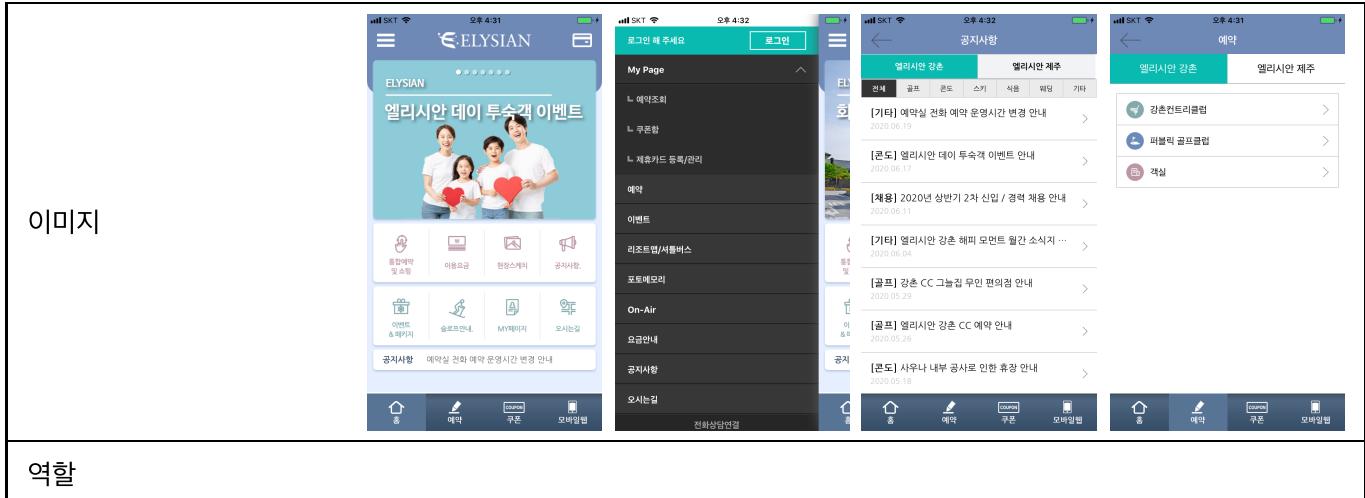
## 주요 기술

- Objective-c, WebView, APNs, AFNetworking, Localizable

## 도전 과제 및 해결 방법

- 13개 언어 지원을 위해, 화면 노출 시점에 문구 설정
- 디바이스 언어 외에도 직접 변경이 가능하도록 shared 한 언어 설정값 사용
- sheet -> Strings.localizable 변경되는 스크립트 생성
- Strings.localizable -> enum 자동 변경되는 스크립트 생성
- (파일을 통한)모듈화로 앱간의 통일성 향상
- 하단 유동 배너 광고 지원

## 엘리시안



이미지

역할

- iOS 메인 개발자 (참여 인원 : iOS 1)

#### 주요 기술

- Objective-c, WebView, APNs, AFNetworking

#### 도전 과제 및 해결 방법

- 엘리시안 강촌/제주 예약 지원 앱
- CCTV를 통해 스키장 현재 상태 확인 가능
- 숙소 예약의 경우 외부 결제 지원
- 네이티브와 웹의 병합된 형태
- APNS 푸시 지원

## 유니텍 (14.07 ~ 15.04)

### MFCC(Multi Function Control console)

역할

- 인터페이스 연결 및 UI 개발 담당(투입인원: 3명)

#### 주요 기술

- MFC, c++

#### 도전 과제 및 해결 방법

- 선유도 어뢰 관리통제를 위한 사용자 프로그램
- LIG 넥스원과 차기중어뢰 개발 협력
- 선에 연결된 어뢰와 데이터를 실시간으로 주고받으며 화면에 표시
- 모의 전투를 위한 시뮬레이션 기능 제공
- 유속, 방향, 레이더, 어뢰 속도 등의 추가 정보 화면에 표시
- 연결 선의 최적 길이 및 속도 확인 가능

- 인터페이스 연결 및 실시간 반영 속도 최적화
- 진해 국방과학연구소에서 2달이상 파견 근무
- 독일 어뢰업체로부터 기술 자문 ( 업체명은 극비 사항 )
- 실제 어뢰발사를 위해 바다에서 테스트
- 발사한 위치와 현재 발사체와의 거리 추가 기능 지원

## 강원대학교 전산원 (11.07 ~ 13.02)

### 강원대학교 App



- Android 메인 개발자 (참여 인원 : Android 1)

### 주요 기술

- Android Java

### 도전 과제 및 해결 방법

- 강원대학교 학생 편의를 위한 App
- 시간표, 식단표, 셔틀버스, 성적 조회 등 편의성 기능 제공
- 최초 Native제작 참여, 추후 Flex Mobile로 변경 시 어시스턴스
- 현재위치좌표, 암호화, 내부 DB사용
- 안드로이드 디바이스 8가지 최소 지원