

# 세상을 더 편리하게 만들고 싶은 iOS 개발자

## 최인호 입니다



### Info.

#### Introduction

- 안녕하세요! 저는 세상을 더 편리하게 만들고 싶은 iOS 개발자입니다.
- 존중하고 존중 받을 수 있는 개발자로 목표로 하고 있습니다.
- 흥미로 시작된 끈질긴 생각으로 만든 몰입이 많은 것을 바꿀 수 있다고 생각합니다. 그렇기에 일상 속 문제들을 해결하기 위해 흥미를 붙이고 끈질기게 생각합니다.

#### Contact & Channel

- |         |                    |          |                                                                              |
|---------|--------------------|----------|------------------------------------------------------------------------------|
| • Email | eknow25@icloud.com | • Blog   | <a href="https://toby.hashnode.dev">https://toby.hashnode.dev</a>            |
|         |                    |          | <a href="https://slowsure.tistory.com">https://slowsure.tistory.com</a> *구버전 |
| • Phone | 010 - 9303 - 7422  | • GitHub | <a href="https://github.com/E-know">https://github.com/E-know</a>            |

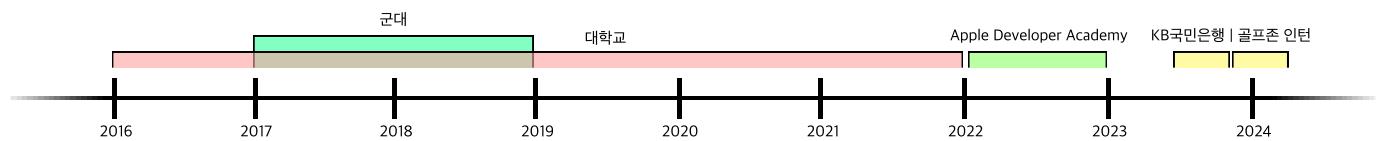
#### Graduation

세종대학교 컴퓨터공학과 [ 학점: 4.02 / 4.5 ] [\\*성적증명서](#)

#### Tools

- |       |  |  |  |         |  |  |  |
|-------|--|--|--|---------|--|--|--|
| • Dev |  |  |  | • Other |  |  |  |
|-------|--|--|--|---------|--|--|--|

#### Timeline



## Project.

### AzitQRCode (개인프로젝트)

#### 소비자 입장에서 입장 프로세스의 불편함을 걷어낼 수는 없을까?

사용자의 입장에서 앱 사용성을 바라봤습니다.

- QR코드를 만들기 위해 앱을 다운로드하는 것은 효율적이지 않다고 판단하여, App Clip을 이용하여 빠르고 쉽게 QR코드를 생성할 수 있도록 했습니다.
- 자료가 많지 않은 상황에서 기술을 구현해야 했습니다. 자료가 많지 않았기 때문에, 애플 공식 문서를 참고하여 개발했습니다. 이 과정에서 공식 문서의 중요성과 어떻게 읽어야 하는지에 대해 배울 수 있었습니다.
- 새로운 기술을 통해서 새로운 경험을 제공할 수 있다는 것을 깨달았습니다. 새로운 기술에 대해서 끊임 없이 흥미를 붙여 공부해야겠다고 생각했습니다.

사용자 뿐만 아니라 운용자의 입장에서도 입장 프로세스의 불편함을 걷었습니다.

- QR 코드를 인식한 후, 사용자 및 스태프의 추가 작업 없이 출석을 자동화 할 수 있도록 구현했습니다.



<QR코드를 이용하여 출석하고 있는 모습>



<AppClip을 통해서 설치 없이 앱을 실행하는 모습>

# Project.

## Workade

(팀프로젝트 7명)

[앱 스토어](#)

[GitHub](#)

[Notion](#)

- UIKit ( Programmatically )
- NaverMaps
- Google Sign In
- Sign In with Apple
- Combine

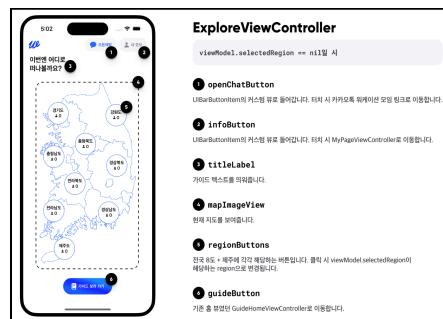
### 디지털 노마드의 [다양한 직군과 커뮤니케이션 욕망]을 해소해 줄 수는 없을까?

#### 앱 로그인에 대한 UX 개선했습니다.

- 앱 실행시 바로 로그인 창을 띄우기보다는 워케이션을 시작할 때 로그인 로직 순서 변경 제안 및 수정하였습니다. 잘 알려지지 않은 앱의 실행시 로그인보다는 사용자가 앱에 대해서 알아가는 것이 더 좋은 UX라고 판단했기 때문입니다.

#### 팀으로서 협업에 대해서 고민하고, 더 잘 협업할 수 있도록 다양한 방법을 도입했습니다.

- 새로운 View가 추가되면 리뷰하는 사람이 쉽게 이해할 수 있도록 PR내용에 이미지 설명을 도입했습니다.
- PR에 대한 사전 배경 없이도 코드 리뷰를 할 수 있게 하고 전체적인 View 구조를 이미지화하여 Pull Request의 가독성을 개선하였습니다.



< View 구조 이미지 >

#### Code Review Convention

• 아래의 라벨링을 코멘트 앞에 붙여주세요.

Label	Review	Mean	Description
[P1]	Request changes	꼭 반영해주세요.	반영할 수 없다면 합리적인 의견을 통해 리뷰어를 설득할 수 있어야 합니다.
[P2]	Request changes	적극적으로 고려해주세요.	수용할 수 없다면 적합한 의견을 들어 토론을 권장합니다.
[P3]	Comment	웬만하면 반영해 주세요.	수용할 수 없다면 반영할 수 없는 이유를 설명하거나 Issue를 만들어서 다음에 반영할 것을 표현합니다.
[P4]	Approve	반영해도 좋고 넘어가도 좋습니다.	의견을 달지 않고 이모지로 반응을 남겨도 됩니다. 의견을 반영하는게 좋을지 고민했다면 충분합니다.
[P5]	Approve	그냥 사소한 의견입니다.	작성 및 이모지를 남기지 않아도 충분합니다.
[의견]	Approve	궁금한 혹은 정말 개인적 인 질문들	

< 코드 리뷰 라벨링 >

- 코드리뷰 라벨링을 도입하여 리뷰의 중요도를 구분하여 리뷰어가 쉽게 리뷰할 수 있도록 환경을 구성했습니다. ([예시](#))

- 온라인의 협업을 증가시킴으로써 다음을 얻을 수 있었습니다.

1. 코드 리뷰를 통해 다른 팀원의 코드를 살펴보면서, 더 나은 코드를 작성하는 방법을 배우고 개인적인 성장과 팀의 성장을 학칠 수 있었습니다.
2. 모든 팀원이 적극적인 코드 리뷰를 위한 환경을 조성하게 되었습니다. 이로 인해 코드보지 않고 Approve하는 팀원도 코드 리뷰를 하게 되었고 팀원 모두 전반적인 코드 이해도가 증가했습니다.
3. 이전 프로젝트에 비해 View 간 연결 시 생기는 버그가 줄었습니다. 적극적인 코드 리뷰 덕분에 다른 팀원의 코드 구조 및 구현 방법을 View 간 연결 전에 미리 파악할 수 있었고, 이러한 배경이 버그를 줄일 수 있었습니다.

UIKit을 사용하여 로그인 화면 및 로그인 로직을 구현하였습니다.

NaverMaps를 이용하여 오피스 주변 지도 및 추천 여행지를 마커로 표시하였습니다.

Firebase 및 Combine을 이용하여 지역별 이용자 수 및 직업 인원 수를 동기화하였습니다.

## Award.

### JUNCTION ASIA 2022 - 🏅 AWS Track FINALIST 2nd

2022.08.21

산책을 걷으면서 땅따먹기 놀이 하며 유저들과 대결하여 더 많은 땅을 먹는 경쟁 게임입니다.

[GitHub](#)

- SwiftUI & UIKit
- MapKit
- CLLocation



### 제 11회 창의설계경진대회 - 🥇 우수상 수상

2021.06.18

미세먼지 데이터, 국민건강보험 데이터를 이용해서 미세먼지 노출량에 따른 질병예측 프로그램입니다.

[GitHub](#)

- Python
- Pandas

# Expereince.

---

## 골프존

Intern

2023.11. ~ 2024.02

### 존중하는, 존중 받는 개발자에 대하여

구현 과제 중에 custom UICollectionViewCell의 위치가 원하는 위치에 나오지 않아서 기본 layout을 사용해야하나 고민하고 있었습니다. 비록 인턴과 제이지만 구현 방식을 수정하는 것에 대해서 여러 팀원들에게 의견을 물어보았을 때, 존중에 대해서 더 깊이 생각할 수 있었습니다. A 팀원은 코드를 보지 않고, 문제 상황만을 보고 자신의 짠 코드 중 비슷하게 구현한 코드를 소개해줬습니다. B 팀원은 구조와 함께 코드를 보면서 질문을 통해 해당 코드를 작성한 이유에 대해서 이해하고 놓쳤던 부분에 대해서 설명해줬습니다. 이 경험을 통해서 개발자의 존중에 대해 깊이 생각하게 되었습니다. 비록 인턴이지만, 인턴의 생각과 이유에 대해서 듣고 같이 고민하여 같이 해결해 주었던 B와 인턴이니깐 준 코드를 보고 배우라는 A의 방식의 차이에서 B 팀원의 방식이 존중받고, 같이 일하고 싶은 개발자였다는 걸 깨달았습니다. 해당 경험을 통해서 배움의 깊이와 상관없이 존중하고, 존중받는 개발자가 되어야겠다는 개발자 커리어를 계획했습니다.

## KB국민은행

Intern

2023.07. ~ 2023.10

### 뜻하지 않은 상황에서도 배울 것을 찾는 법

인턴 기간 중 iOS 개발 경험 하길 희망했지만, 그렇지 못했습니다. 하지만 좌절하지 않고 할 수 있는 영역 내에서 무언갈 얻어가고자 했습니다. 은행앱은 오랜 기간 유지보수를 하면서 갖게 된 레거시 코드도 있었지만, 초기 설계적인 실수로 생기는 레거시 코드와 실수가 많았습니다. 예를 들어 콜백함수가 1개밖에 없을 것이라 생각하고 함수명을 callBack\_success라고 지었지만 이후 확장하며 여러 콜백 함수를 다루게 되면서 callBack\_success2, callBack\_success3로 코드가 작성된 것을 보았습니다. 이러한 코드를 보면서 확장성을 염두에 두는 설계가 중요하다는 것을 깨닫게 되었습니다.

## KWDC | AsyncSwift

Organizer

2022.07. ~ Now

- SyncSwift 2022 개최
- KWDC23 개최

### 커뮤니케이션 능력을 기를 수 있었습니다.

컨퍼런스를 계획하면서 수 많은 회의를 통해서 팀원들과 업무를 조율하고 토론해야 했습니다. 단순히 회의를 많이 했다고 커뮤니케이션 능력을 기를 수는 없었습니다. 컨퍼런스가 끝나면 Try-Problem-Solution을 기반으로 한 회고를 진행하며 더 나은 커뮤니케이션 방법에 대해 고민하고 개선시켜 나갈 수 있었습니다.

[AsyncSwift 공식 사이트](#)

[KWDC 공식 사이트](#)

### 책임감을 가지면 더 빛이 난다.

KWDC 연사자팀에서 일했습니다. 연사자의 발표는 컨퍼런스의 성공 여부를 좌우할 수 있기 때문에 섭외는 매우 중요한 일입니다. 그렇기에 연사자 섭외에 있어 책임감을 가지고 임했습니다. 연사자 선정 기준을 정하고, 연사자와의 커뮤니케이션을 원활하게 진행했습니다. 또한, 연사자가 발표를 준비하면서 생기는 문제들에 대해서 케어하면서 연사자들의 발표가 빛 날 수 있도록 노력했습니다. 이러한 노력의 결과로, KWDC는 성공적으로 개최되었고, 많은 개발자, PM/PO 그리고 디자이너들이 유익한 정보를 얻어 갈 수 있었습니다. 저는 이번 경험을 통해 책임감의 중요성을 배웠습니다. 책임감은 어떤 일을 할 때 가장 중요한 덕목이라고 생각합니다. 돈을 받지 않고 컨퍼런스를 위해서 책임감을 가지고 일했기 때문에 다른 일보다 더 책임감의 중요성에 대해서 배울 수 있었던 계기가 되었습니다.

## Apple Developer Academy @ POSTECH

Jr. Learner

2022.03 ~ 2022.12

### 꼭 개발만이 답이 아니라는 것을 깨달았습니다.

개발 실력의 한계로 인해 더 나은 사용자 경험을 제공하지 못한 경우도 있었습니다. 개발 실력 한계와 정해진 마감기한을 지키려 하다보면 의도한 대로 앱이 나오지 않는 경우가 있었습니다. 그러나 프로젝트 회고에서 이러한 점을 이야기하니 디자이너와 기획자 모두 일찍 대화할 수 있었으면 다른 대안을 같이 고민할 수 있었다는 피드백을 받았습니다. 제품의 좋은 경험을 제공하기 위해서는 **개발이 아니더라도 기획이나 디자인을 수정하여 제공할 수 있다는 것을 깨달았습니다.** 이 경험을 통해 프로젝트에서 막히면 그 답이 개발이 아닐 수 있다는 것을 깨닫고, 같은 팀원과 함께 더 나은 방향을 모색할 것입니다.

### 팀의 목표를 위해서 저의 역할에 대해서 고민하고 배울 수 있었습니다.

아카데미 내에서 모든 구성원들은 기획부터 앱 출시까지 함께했습니다. 4번의 프로젝트를 하면서 팀의 목표를 이루기 위해서 제가 해야하고 가장 잘할 수 있는 것에 대해서 고민했습니다. 기획과 디자인 단계에서 **현실 가능성과 미래 가능성을 제시해줘야 한다**고 생각했습니다. 지금의 개발에서는 현실화할 수 없는 기획과 디자인을 팀원들에게 알려 다른 대안을 같이 찾고, 기획과 디자인이 전하고 싶은 사용자 경험이 있을 때에는 기술을 제시하며 더 다양한 방법을 권할 수 있어야 한다고 생각했습니다.