

# 이태수

---

이메일: tslee.dev@gmail.com | 전화: 010-4635-6536 | **GitHub**: <https://github.com/tsleedev>

본 이력서는 아래 주소에서 최신 업데이트본을 확인할 수 있습니다:

<https://github.com/tsleedev/resume>

안녕하세요. iOS 개발자 이태수입니다.

프로젝트 규모와 팀 상황에 맞는 **실용적 아키텍처 설계**와 **팀 생산성 향상**을 주도해 왔습니다.

- **사람인HR**

- 30여 개 앱 유지보수를 총괄하며, 기술 구조 개선을 리딩
- **Objective-C → Swift 전환, RxSwift + MVVM 도입**으로 개발 효율성과 코드 일관성 향상
- 웹 브릿지 구조 개선 및 가이드화로 협업 효율 강화
- 알림 통계 시스템 제안·구현으로 사용자 참여도 향상

- **핀크(Finnq)**

- **Tuist 기반 Microfeatures 아키텍처**를 설계·도입하여 기능 독립성, 테스트 용이성, 개발 속도, 유지보수 효율을 향상
- 모듈 간 의존도 정리 및 빌드 최적화 작업을 주도, 재빌드 범위 축소와 빌드 시간 안정화를 통해 대규모 프로젝트 유지보수성 확보
- **SwiftLint, SwiftGen, Fastlane**을 팀 내 표준으로 정착시켜 코드 품질 관리와 배포 자동화 수준 강화

- **개발 철학**

- 새로운 기술을 빠르게 학습·적용하되, 팀 상황에 맞는 **점진적·지속 가능한 도입**을 지향
- **안정적이고 확장 가능한 앱**을 통해 사용자에게 실질적 가치를 제공하는 것이 제 개발의 중심입니다.

---

## 주요 성과 및 문제 해결 경험

Tuist 기반 모듈 아키텍처 도입 (핀크, 2023~)

### 왜 이 작업을 시작했나

핀크 앱이 성장하면서 개발 경험이 점점 나빠지고 있었습니다. 화면 하나를 수정하려면:

1. 전체 앱을 빌드 (오래 걸림)
2. 여러 화면을 거쳐 수정할 화면까지 이동
3. 수정 사항 확인 후 다시 1번부터 반복

서버 작업 중이거나 API가 불안정하면 개발이 멈췄고, 시뮬레이터와 Preview도 사용할 수 없는 구조였습니다.

### 어떻게 해결했나

- 구글링과 다른 회사 사례를 찾아보며 Tuist + Microfeatures Architecture를 알게 됨
- Tuist 공식 문서를 공부하며 모듈 구조 설계 방법 학습
- 각 Feature를 5개의 구성 요소로 분리하는 구조 적용:

- **Interface:** 다른 모듈이 사용할 수 있는 공개 인터페이스 (순환 참조 방지)
- **Source:** 실제 구현 코드
- **Testing:** 테스트를 위한 Mock 데이터
- **Tests:** 유닛 테스트
- **Example:** 독립 실행 가능한 예제 앱 (서버 없이 개발 가능)

### 기술적으로 어려웠던 점

- 기존 모놀리식 코드베이스를 모듈로 분리하는 작업이 방대했음
- 모듈 간 순환 참조를 하나씩 찾아서 제거해야 했음
- Tuist 자체가 팀에 처음 도입하는 도구라 학습 곡선이 높았음
- 팀원들도 처음엔 "왜 이렇게 복잡하게 하나"며 회의적이었음

내부 세미나를 여러 차례 열어 Tuist의 필요성과 사용법을 공유했고, 상세한 가이드 문서를 작성해 팀원들이 점진적으로 적응할 수 있도록 도왔습니다.

### 결과와 임팩트

- 각 Feature의 Example 앱으로 독립적으로 개발하고 테스트 가능
- Preview 활용으로 UI 수정 시 즉각 확인 가능 (빌드-실행-네비게이션 과정 불필요)
- Mock 데이터로 서버 작업과 무관하게 개발 진행 가능
- 빌드 시간 대폭 단축 및 증분 빌드 최적화
- 팀원들도 "예전으로 돌아갈 수 없다"고 말할 정도로 만족

이 작업을 통해 단순히 기술을 도입하는 것을 넘어, 팀 전체의 개발 경험을 개선하고 생산성을 높이는 것이 얼마나 중요한지 배웠습니다.

## 예약송금 기능 제안 및 UX 개선 (핑크, 2020~2021)

### 왜 이 작업을 시작했나

야간에 송금하려다 은행 점검시간에 걸려 송금하지 못한 경험이 있었습니다. "지금 송금할 수 없으니 나중에 다시 시도하세요"라는 안내만 있었고, 사용자는 점검시간이 끝날 때까지 기다리거나 나중에 다시 기억해서 송금해야 했습니다. 경쟁사에는 이미 예약송금 기능이 있었고, 이 불편함을 해결하면 사용자 경험이 크게 개선될 것 같아 기획팀에 제안했습니다.

### 어떻게 해결했나

- 송금 중 점검시간이 되면 예약송금을 제안하는 플로우 설계
- 사용자가 입력한 송금 정보(금액, 받는 사람 등)를 그대로 유지하여 다시 입력하지 않도록 처리
- 사용자가 끊임 없이 자연스럽게 예약송금으로 전환될 수 있도록 UX에 집중

### 결과와 임팩트

- 점검시간에 송금을 시도한 사용자가 바로 예약송금으로 전환하여 편리하게 이용
- 예약송금 기능이 활발하게 사용되어 사용자 만족도 향상
- 직접 경험한 불편함을 해결하여 실질적인 사용자 가치 제공

본인이 직접 불편함을 경험하고 이를 개선한 경험으로, 사용자 입장에서 생각하고 자연스러운 UX를 만드는 것의 중요성을 배웠습니다.

## JavaScript Bridge 모듈화 및 협업 체계 구축 (사람인, 2019)

## 왜 이 작업을 시작했나

사람인 앱은 웹과 네이티브가 혼재된 하이브리드 구조였습니다. 웹팀에서 네이티브 기능을 호출해야 할 때마다 앱 개발자에게 요청이 들어왔고, 월평균 10건 이상의 Bridge 관련 작업이 발생했습니다. 코드 수정이 필요한 경우도 있었지만, 주로 "어떻게 사용하는지" 설명하고 가이드하는 데 시간을 소비했습니다. Android와 iOS의 호출 방식이 달라 웹팀이 플랫폼별로 다르게 구현해야 하는 불편함도 있었습니다. 이런 반복적인 작업이 앱 개발자의 병목이 되고 있었고, 근본적인 해결이 필요하다고 판단했습니다.

## 어떻게 해결했나

- WKWebView의 ScriptMessageHandler를 활용해 통일된 Bridge 시스템 구축
- Android와 동일한 방식으로 호출할 수 있도록 인터페이스 설계
- 필요해 보이는 네이티브 기능들을 미리 구현하여 제공
- 샘플 웹페이지를 만들어 웹팀이 직접 테스트하고 학습할 수 있도록 지원
- 상세한 가이드 문서 작성 (호출 방법, 예제 코드, 에러 처리 등)
- 웹팀 대상 세미나를 진행하여 사용법 전파

## 기술적으로 어려웠던 점

- 웹과 네이티브 간 비동기 통신 구조 설계 및 에러 핸들링
- Android와 다른 WKWebView의 특성을 고려하면서도 동일한 인터페이스로 통일
- 기존 레거시 코드와의 호환성 유지하며 점진적으로 전환

## 결과와 임팩트

- 웹팀이 샘플 웹페이지와 가이드 문서를 보고 스스로 Bridge를 활용할 수 있게 됨
- 앱 개발자에게 오는 Bridge 관련 요청이 대폭 감소
- 웹-앱 간 커뮤니케이션 비용 절감 및 협업 효율 극대화

단순히 코드를 작성하는 것을 넘어, 팀 전체의 워크플로우를 개선하고 문서화와 교육을 통해 지속 가능한 협업 문화를 만드는 경험이었습니다.

## 알림 수신 통계 시스템 제안 및 구축 (사람인, 2017)

### 왜 이 작업을 시작했나

사람인 앱에서 채용공고 마감 알림 등을 매일 수천 건 발송하고 있었지만, 사용자가 실제로 알림을 받았는지, 클릭했는지 추적할 방법이 없었습니다. 마케팅팀도 알림 효과를 측정하고 싶어했지만 데이터가 전혀 없는 상황이었습니다. iOS 10에서 Notification Service Extension이 출시되면서, 이를 활용하면 Rich Push와 함께 알림 수신 시점을 추적할 수 있겠다고 판단했습니다. 사용자 경험을 개선할 수 있을 것 같아 주도적으로 제안했습니다.

### 어떻게 해결했나

- Notification Service Extension을 활용한 알림 수신 추적 구현
- 알림 수신 → 클릭 → 앱 실행 → 상세 페이지 진입까지 전체 퍼널 추적 설계
- 백엔드팀과 협업하여 통계 수집 API 및 분석 시스템 구축
- 시간대별/요일별 알림 수신률 및 클릭률 데이터 추적

### 기술적으로 어려웠던 점

- Notification Service Extension의 제한된 실행 시간과 메모리 내에서 안정적으로 로그 전송
- 백엔드팀과 협업하여 대용량 로그 수집 및 집계 시스템 설계

## 결과와 임팩트

- 알림 발송 시간대 최적화 및 개인화된 발송 전략 수립 가능
- 데이터 기반으로 알림 문구 A/B 테스트 진행
- 알림을 통한 사용자 재방문율 향상
- 회사 전체의 알림 전략 수립에 기여

사용자 경험 개선과 비즈니스 가치 향상을 위해 스스로 문제를 발견하고 해결책을 제안한 경험이었습니다.

---

## 학력

- 동국대학교 - 컴퓨터공학과 전공 (2000년 3월 ~ 2007년 8월)

## 자격

- 정보처리기사 (2009년 6월 1일)

## 해외경험

- 어학연수 (싱가포르, 2007.08 ~ 2008.11)
- 해외출장 (캐나다 토론토, 2010.04 ~ 2010.05)
- 해외출장 (미국 시애틀, 2011.02 ~ 2011.03)
- 해외출장 (미국 댈러스, 2012.03 ~ 2012.04)

## 경력

### 핑크 (2020년 4월 ~ 현재)

- 기술 전환 및 코드 품질 관리:
  - Objective-C > Swift 전환
  - AFNetworking > Alamofire 전환
  - SwiftLint 및 SwiftGen 적용
  - 테스트 코드를 작성하여 앱의 안정성과 신뢰성 강화
  - 정기적인 코드 리뷰를 통해 팀 전체의 코드 품질 향상
  - 팀 세미나를 주최하여 최신 기술 동향 및 모범 사례 공유
  - GitLab CI/CD에 Ollama(32B) 기반 자동 코드 리뷰 시스템 도입
- 아키텍처 혁신 및 표준화:
  - RxSwift 및 MVVM 패턴 도입
  - Composable Architecture 도입
  - FlexLayout, PinLayout 도입
  - Tuist와 Microfeatures Architecture 도입 → Preview 및 ExampleApp 활용 최적화
  - 피그마를 활용한 Typography 및 Color Style 표준화
- 다양한 기능 개발 및 관리:
  - 출금, 송금, 예약송금, 이미지 푸시, 제로페이 모바일상품권, 스크래핑, 전자문서지갑 등

### 사람인HR (2013년 11월 ~ 2020년 3월)

- 기술 전환 및 코드 품질 관리:
  - Objective-C > Swift 전환

- AFNetworking > Alamofire 전환
- UIWebView > WKWebView 전환
- 아키텍처 혁신 및 표준화:
  - RxSwift 및 MVVM 패턴 도입
  - JavaScript Bridge 모듈화 및 가이드

## 플레타뮤토 (2009년 10월 ~ 2013년 7월)

- Objective-C로 앱 개발 및 유지 보수
- CoreData 사용

## 프로젝트 (핑크)

### 핑크 (2020.04 ~ 현재)

- 소개: 대출, 투자, 마이데이터 서비스
- 관련 기술: RxSwift, Tuist, Storyboard, Auto Layout, SnapKit, FlexLayout, PinLayout, WKWebView, Alamofire, Keychain, UUID, APNS, REST API, MVVM, SwiftLint, SwiftGen, Fastlane, GitLab CI/CD, Figma 등
- 역할 & 성과:
  - 담당 업무:
    - iOS 설계, 개발, 유지보수 (기여도 35%)
    - 출금, 송금, 예약송금, 이미지 푸시, 제로페이 모바일상품권, 스크래핑, 전자문서지갑 등
  - 기술 개선:
    - Objective-C > Swift로 전환 > RxSwift MVVM 도입 → 코드 가독성 및 유지보수성 향상 (기여도 80%)
    - AFNetworking > Alamofire 전환 → 네트워크 레이어 현대화 (기여도 80%)
    - SwiftLint, SwiftGen 적용 → 코드 품질 일관성 확보 (기여도 100%)
    - 피그마의 Typography 및 Color Style 표준화 → 디자이너 협업 효율 강화 (기여도 100%)
    - Composable Architecture 도입 → 단방향 데이터 플로우 및 테스트 가능한 구조 확립 (기여도 100%)
    - Tuist 및 Microfeatures Architecture 도입 → 빌드 시간 단축, 개발 속도 향상, 테스트 용이성 향상, 병렬 개발 가능 (기여도 100%)
    - 테스트 코드 작성 → 앱 안정성 및 신뢰성 강화
    - GitLab CI/CD에 Ollama(32B) 기반 자동 코드 리뷰 시스템 도입 → 리뷰 품질 일관성 및 팀 생산성 향상
    - 정기적인 코드 리뷰 → 팀 전체 코드 품질 향상
    - 팀 세미나 주최 → 최신 기술 동향 및 모범 사례 공유
- 경험 & 성장한 점:
  - 레거시 전환 프로젝트를 주도하며 점진적 마이그레이션 전략 수립 능력 습득
  - 팀 세미나와 코드 리뷰 문화를 정착시키며 기술 리더십 역량 강화
  - 모듈화 아키텍처 설계 및 의존성 최적화를 통해 대규모 프로젝트 빌드 시스템 개선 경험
  - 새로운 아키텍처 및 개발 패러다임 도입을 통해 팀 전체의 개발 워크플로우 개선 경험
- 확인경로: <https://apps.apple.com/app/id1260871482>

## 프로젝트 (사람인)

## 아이엠그라운드 (2019.06 ~ 2020.03)

- 소개: 모의면접 AI 분석 서비스 (인적성 검사, 면접 질문에 대한 답변을 녹화 후 AWS에서 분석)
- 관련 기술: RxSwift, Storyboard, Auto Layout, WKWebView, Keychain, UUID, APNS, REST API, 동영상 촬영, AWS S3, 인앱 결제, Dynamic Link 등
- 역할:
  - iOS 설계, 개발, 유지보수 (기여도 100%)
  - RxSwift MVVM 패턴으로 앱 개발 및 유지보수 (기여도 100%)
- 확인경로: <https://apps.apple.com/app/id1472955601>

## 사람인 (2013.11 ~ 2020.03)

- 소개: 구인구직 포털 대표 모바일 앱 (채용공고 검색, 입사지원)
- 관련 기술: RxSwift, TodayExtension, Storyboard, Auto Layout, WKWebView, Keychain, UUID, APNS, REST API, AES256 암호화, 3D Touch, 소셜로그인(Naver, Facebook, Google, Kakao), Dynamic Link, AdMob 등
- 역할:
  - iOS 설계, 개발, 유지보수 (기여도 100%)
  - Objective-C > Swift로 전환 > RxSwift MVVM 도입 (기여도 100%)
  - UIWebView > WKWebView 전환 > JavaScript Bridge 모듈화 및 가이드 (기여도 100%)
- 경험 & 성장한 점:
  - 일정 조율 및 프로젝트 관리 등을 맡아서 진행한 경험
  - 작은 웹과 앱의 연동 요청으로 JavaScript Bridge 모듈화 및 가이드 작성 후 세미나를 통해 전파한 경험
  - 3D Touch, 이미지 푸시(Notification Service Extension), TodayExtension 등 새로운 기술을 적용 건의해 추가한 경험
  - Notification Service Extension을 활용해서 알림 수신 확인 통계를 건의해 추가한 경험
  - Google Analytics, Firebase, Fabric 등을 활용한 통계, Crash 수집 등을 통해 서비스 안정화에 기여함
  - 입사 초기 앱의 안정적인 서비스를 위해 서버점검, 업데이트 팝업을 적용할 수 있게 건의 적용한 경험
- 확인경로: <https://apps.apple.com/app/id739013038>

## 연봉계산기 (2013.11 ~ 2020.03)

- 소개: 연봉으로 월 실수령액을 계산해주는 앱
- 관련 기술: Storyboard, Auto Layout, REST API, SHA512 암호화 등
- 역할:
  - iOS 개발, 유지보수 (기여도 100%)
  - Objective-C > Swift로 전환 (기여도 100%)
- 확인경로: <https://apps.apple.com/app/id536455221>

## 맞춤취업사람인 (2013.11 ~ 2018.09)

- 소개: 사용자의 설정에 맞춰 채용공고를 보여주는 앱
- 관련 기술: Storyboard, Auto Layout, WKWebView, Keychain, UUID, APNS, REST API, AES256 암호화 등
- 역할:
  - iOS 개발, 유지보수 (기여도 100%)
  - Objective-C > Swift로 전환 (기여도 100%)
- 확인경로: 서비스 종료

## 공채의명가 (2013.11 ~ 2018.02)

- 소개: 공채를 준비하는 타겟을 위한 공채 정보 전달 앱
- 관련 기술: Storyboard, Auto Layout, WKWebView, Keychain, UUID, APNS, REST API, AES256 암호화 등
- 역할:
  - iOS 개발, 유지보수 (기여도 100%)
  - Objective-C > Swift로 전환 (기여도 100%)
- 확인경로: 서비스 종료

## 권역/버티컬 (2013.11 ~ 2016.12)

- 소개: 지역별(4종) 업종별(13종)로 채용공고를 보여주는 앱
- 관련 기술: Storyboard, Auto Layout, REST API, Google Map 등
- 역할:
  - Objective-C > Swift로 전환 (기여도 100%)
- 경험 & 성장한 점:
  - 같은 컨셉의 앱을 추가 출시했다가 스팸으로 판단되어 리젝당한 경험
  - 같은 소스코드로 타켓만 분리하여 관리한 경험
- 확인경로: 서비스 종료

## 거기어때 (2013.11 ~ 2015.06)

- 소개: 페이스북 친구들이 다니는 회사를 보여주며 회사에 대해 문의하는 앱
- 관련 기술: Storyboard, Auto Layout, REST API, Facebook 연동 등
- 역할: iOS 개발 및 기능 업데이트
- 확인경로: 서비스 종료

## 오투잡 (2016.04 ~ 2020.03)

- 소개: 투잡 포털 오투잡의 대표 모바일 앱
- 관련 기술: 이니시스 결제 모듈 연동, Swift, Storyboard, Auto Layout, WKWebView, Keychain, UUID, APNS, REST API, 소셜로그인(Naver, Google, Facebook, Kakao), 파일 업/다운로드 등
- 역할:
  - iOS 설계, 개발, 유지보수 (기여도 100%)
  - Objective-C > Swift로 전환 (기여도 100%)
  - UIWebView > WKWebView 전환 (기여도 100%)
- 확인경로: <https://apps.apple.com/app/id1093727086>

## 맞춤알바 (2013.11 ~ 2016.04)

- 소개: 사용자의 설정에 맞춰 알바공고를 보여주는 앱
- 관련 기술: Storyboard, Auto Layout, APNS, REST API 등
- 역할:
  - iOS 설계, 개발, 유지보수 (기여도 50%)
  - Objective-C > Swift로 전환 (기여도 50%)
- 확인경로: 서비스 종료

## 쉽고빠른알바인 (2013.11 ~ 2016.04)

- 소개: 위치 기반으로 알바공고를 보여주는 앱 (구인공고를 앱에서 바로 등록 가능)
- 관련 기술: Storyboard, Auto Layout, APNS, REST API, Google Map 등
- 역할:
  - iOS 설계, 개발, 유지보수 (기여도 50%)
  - Objective-C > Swift로 전환 (기여도 50%)
- 확인경로: 서비스 종료

### 알바네비 (2013.11 ~ 2016.04)

- 소개: 내 주변 일자리 정보 제공 지역 기반 커뮤니티 앱
- 관련 기술: Storyboard, Auto Layout, REST API, Facebook 연동 등
- 역할:
  - iOS 설계, 개발, 유지보수 (기여도 50%)
  - Objective-C > Swift로 전환 (기여도 50%)
- 확인경로: 서비스 종료

## 프로젝트 (플레타뮤토)

### iOS ChatON (챗온)

- 소개: 삼성 채팅 앱
- 관련 기술: Xib, 연락처 동기화(AddressBook.framework), 멀티 이미지 전송(AssetsLibrary.framework), CoreData 등
- 역할: iOS 개발 및 기능 업데이트 (친구 목록 관리)
- 경험 & 성장한 점:
  - 실무에서 첫 iOS 프로젝트로 여러 사람과 협업하는 방법을 배움
  - iOS의 배포 프로세스 및 인증서 관리 등을 경험
- 확인경로: 서비스 종료

### 3G 모델 미주 향

- 소개: 3G 단말기를 AT&T, TMO, Tracfone 등 사업자별로 맞춤화
- 역할: 사업자 특성에 맞춰 메뉴 및 UI 변경, 필드 이슈 대응

## 오픈 소스 개발

### 토스트 관리 (2024.06)

- 소개: 토스트 메시지를 편리하게 사용하기 위해 제작
- 확인경로: <https://github.com/tsleedev/TSToast>

### 광고 관리 (2023.07)

- 소개: AdMob과 AdManager를 편리하게 사용하기 위해 제작
- 확인경로: <https://github.com/tsleedev/TSAdView>

### SwiftUI 커스텀 달력 (2025.01)



- 소개: Widget Extension에서 순수 SwiftUI만 사용 가능해 SwiftUI로 달력 제작
- 확인경로: <https://github.com/tsleedev/TSCalendar>

## 커스텀 달력 (2018.04)

- 소개: 달력을 커스터마이징하여 사용하기 위해 제작
- 확인경로: <https://github.com/tsleedev/TSCalendar2>

## 로컬 알림 관리 (2021.06)

- 소개: 로컬 알림 개수 제한(64개)을 최대한으로 활용하기 위해 가장 빠른 일정의 알림을 효율적으로 등록하도록 도와주기 위해 제작
- 확인경로: <https://github.com/tsleedev/TSUserNotifications>

## WKWebView 쿠키 관리 (2021.07)

- 소개: WKWebView는 쿠키를 직접 관리해야 해서 쿠키 관리 및 JavaScript Bridge 활용을 Android와 같은 방법으로 웹에서 호출 가능하도록 하기 위해 제작
- 확인경로: <https://github.com/tsleedev/TSWebView>

## 화면 전환 애니메이션 (2021.10)

- 소개: 화면 전환 애니메이션을 사용하기 쉽게 하기 위해 제작
- 확인경로: <https://github.com/tsleedev/TSPresentation>

## 그 외 샘플과 데모 프로젝트를 꾸준히 작성 중

- github: <https://github.com/tsleedev?tab=repositories>

## 그 외 활동

### 사람인 사내 iOS 세미나 강사 (2019.04)

- 오프라인으로 매회 1~2시간 강의 14회 이상. 커리큘럼:
  1. Hello, World! (2H)
  2. iOS 앱의 구조 (2H)
  3. 스토리보드 (2H)
  4. 화면 전환 (2H)
  5. 다른 뷰 컨트롤러와 데이터 주고 받기 (4H)
  6. Delegate, Notification (2H)
  7. 테이블 뷰 (4H)
  8. CoreData 사용해보기 (2H)
  9. 네트워크 통신 (4H)
  10. 웹 뷰 (6H)
  11. 오토 레이아웃 (2H)
  12. Firebase 활용하기 (2H)
  13. 배포 해보기 (2H)
  14. 초간단 웹앱 만들기 (10H)
- [강의자료](#)

## 오프라인 수강

- 프로젝트로 정복하는 RxJava 2기 (2019.09 ~ 2019.10)
- JavaScript 부트 캠프 12기 (2019.06 ~ 2019.08)
- Flutter로 구현하는 크로스플랫폼 앱 개발 1기 (2019.05 ~ 2019.06)
- RxSwift로 완성하는 iOS 앱 개발 3기 (2018.07 ~ 2018.08)
- Oracle DB 설계 & Java Modeling (SIST 쌍용교육센터, 2009.02 ~ 2009.08)
- [교육과정 및 수료증](#)

## 온라인 수강

- 앨런 iOS Concurrency(동시성) - 디스패치큐와 오퍼레이션큐의 이해 (2024.09)
- The RED : 슈퍼앱 운영을 위한 확장성 높은 앱 아키텍처 구축 by 노수진 (2022.03)
- [교육과정 및 수료증](#)

## 컨퍼런스

- FESTA : GDG Campus Korea, "Try! Flutter - 우리 함께 해봐요" (2019.07.27)
- NAVER DEVIEW 2017, 2018
- AWS Summit Seoul 2017

## 개인 프로젝트

### 위젯 캘린더 : 깔끔하고 심플한 달력 (2015.12 ~ 2019.11)

- 소개:
  - TodayExtension을 활용한 위젯 캘린더
  - 등록된 캘린더의 일정을 가져와서 보여줌
- 관련 기술: TodayExtension, EventKit, CloudKit, App Groups, RxSwift, Realm, 다국어 지원, Firebase
- 역할: 기획, 개발, 운영 모든 부분을 100% 기여
- 성과: 스승의 날 기념 컨셉으로 피처링 선정 (2017.04)
- 확인경로: <https://apps.apple.com/kr/app/id1078875308>

### 위젯 미리알림 (2017.01 ~ 2019.11)

- 소개:
  - TodayExtension을 활용한 위젯 미리알림
  - 기본 미리 알림의 데이터를 가져와서 위젯에서 사용하기 편하게 만들
- 관련 기술: TodayExtension, EventKit, App Groups, 다국어 지원, Firebase
- 역할: 기획, 개발, 운영 모든 부분을 100% 기여
- 확인경로: 서비스 종료

## 스터디 프로젝트

### Today Class (2013.02)

- 소개:

- 첫 화면에서 간단하게 자신만의 배경 선택 및 오늘 수업 시간표 제공
  - 두 번째 화면에서 전체 수업 시간표 제공
- 관련 기술: CoreData, 다국어 지원, GoogleAnalytics
- 역할:
  - 기획 및 운영 (기여도 30%)
  - iOS 개발 (기여도 100%)
- 성과: AppStore에서 4만명 이상 다운로드 (고등학생, 대학생 버전 등록)
- 확인경로: 서비스 종료

## Life Graphy (2013.12)

- 소개: 습관 관리 앱
- 역할:
  - 기획 및 운영 (기여도 30%)
  - iOS 개발 (기여도 100%)
- 관련 기술: CoreData, 다국어 지원, GoogleAnalytics
- 성과: 미국 스토어에 피처링 선정 (2013.12)
- 확인경로: 서비스 종료

## MindMap (2014.09)

- 소개: MIND TREE를 이용하여 마인드 맵핑, 생각정리, 업무정리
- 관련 기술: CoreData, 다국어 지원, GoogleAnalytics
- 역할: 기획, 개발, 운영 모든 부분을 20% 기여
- 확인경로: 서비스 종료

감사합니다.