

# Black Company

## 프로젝트를 소개해 주세요!

EEOS는 Econovation Event Operation System의 준말로, 에코노베이션의 모든 행사들을 관리할 수 있는 체계입니다.

## 프로젝트 하면서 어떤 문제를 겪었나요?

EEOS는 현재 에코노베이션에서 실제 사용 중인 프로젝트입니다. 그러다보니 실제 사용과 관련하여서 겪었던 문제들이 있었습니다.

대표적으로는 로그인과 관련된 문제가 있었는데요.

EEOS는 에코노베이션의 회원만을 대상으로 한 프로젝트이기에, 회원임을 검증할 수 있는 도구가 필요했습니다. 그래서 저희는 에코노베이션의 소통 채널인 Slack을 활용한 로그인 이용하기로 했습니다. 실제 배포 전까지는 아무런 문제 없이 개발이 진행되었습니다.

하지만 배포를 얼마 앞두고 QA과정에서 큰 오류를 발견합니다. 특정 사용자에게 한해서 저희 서비스에 이름이 표기되지 않았는데요. 사실 그전까지는 문제가 없었기 때문에, 어디서부터 어떻게 문제를 해결해 나가야 할지 막막했습니다. 사실 개발 과정에서는 문제가 없었습니다만, 사용자들을 면밀하게 검토하지 못한 문제가 있었습니다. Slack에는 성명과 표시 이름이 있는데요. 저희는 표시 이름을 기준으로 데이터를 받아왔습니다. 하지만 실제 회원들 중에서는 표시 이름을 작성하지 않은 경우가 있었죠. 이런 경우에 저희 서비스에 이름이 표기되지 않았었습니다.

실제 서비스 사용까지 3일 남은 과정에서, 데이터를 다시 받아오게 수정하는 것은 무리였습니다. 그래서 저희는 표시 이름이 설정되어 있지 않을 때는 로그인을 할 수 없게 임시적으로 막는 방법을 사용했습니다.

실제 사용자가 있는 프로젝트에서는 기획, 디자인, 개발 과정에서는 가늠할 수 없는 여러 문제 상황들과 맞닥뜨리는 경우가 존재합니다. 이러한 상황에서 어떻게 사용자에게 최대의 편의를 줄 수 있을지 고민해본 경험을 EEOS를 통해 한 것 같아 매우 뿌듯합니다.

## 프로젝트 하기 전후 달라진 점이 있다면?

- 매주 고정적인 회의와 코어타임을 통해 프로젝트의 프로세스를 경험할 수 있었습니다. 또한 이를 토대로 다른 곳에서도 적용할 수 있었구요.
- 팀원들의 분야가 모두 달랐기에, 다른 분야에 대한 이해도와 관심도가 이전에 비해 많이 향상 되었어요!
- 사용자와 피드백하는 프로젝트였기에 실제 사용자와 만날 일이 많았습니다. 사용자는 우리의 생각보다 냉정하다는 것을 몸소 체험할 수 있었습니다.

## 프로젝트를 시작하는 팀에게 전해줄 꿀팁을 말해주세요!

- 주기적인 회의, 팀의 커뮤니케이션 톨을 규칙적으로 보기 등, 소통에 대한 강제성을 그라운드 룰로 설정해 보세요! 팀 간의 많은 대화가 있을수록 프로젝트의 퀄리티가 올라갑니다.
- 자신과 다른 분야와 협업을 하여도 관심을 가져보세요. 더욱 완성도 있는 프로젝트를 만들 수 있습니다.

- 일정을 보수적으로 잡아야 할 때도 있습니다. 우리의 생각보다 오래 걸릴 수도 있습니다.
- 프로젝트를 몰아서 하지 마세요! 조금이라도 꾸준히가 중요합니다.
- Kick Off를 가져보세요. 각자의 목표를 조정하고 이를 토대로 기획한다면 모두가 만족하는 프로젝트가 될 수 있습니다!

---

## 프로젝트의 주요 목표와 성과는 무엇이었고, 이를 달성하기 위해 어떤 전략을 사용했나요?

**PM :**

실제 사용자를 토대로 한 서비스를 운영하고 싶은 개인적인 니즈가 있었습니다.

프로젝트를 위한 프로젝트에서 벗어나서, 지속가능한 서비스를 만들고 싶었습니다. 그렇기에 고정적인 사용자의 확보가 필요했고 에코노베이션 회원들을 대상으로 한 서비스를 기획하였습니다.

저희는 한 학기의 프로젝트를 총 3개의 기간으로 설정하였습니다. 첫번째 기간에는, 빠르게 MVP를 찾아내 서비스를 배포하였고, 두번째와 세번째 기간에는 사용자의 피드백을 바탕으로 하여 서비스를 발전시켰습니다.

이렇게 기간을 설정하고, 매 기간마다 스프린트를 운영하여 팀으로서의, 또한 각자의 분야에 맞는 목표를 설정하였습니다. 매 스프린트가 끝난 이후 회고를 진행하여서 진행도를 점검하였고 이후 반복되는 스프린트마다 회고를 토대로한 발전된 방향으로 프로젝트를 진행할 수 있었습니다.

현재는 에코노베이션의 모든 활동 회원들이 EEOS에 가입되어 있으며, 유료 회원과 졸업 회원분들도 EEOS에 가입하였습니다. 에코노베이션 내의 다양한 수요조사와 출석체크가 EEOS를 통해 이루어지고 있으며, 더욱 더 고도화된 서비스를 구상하고 있습니다.

---

## 프로젝트 개발 중 어려움을 겪은 경험이 있나요? 어떻게 해결했으며, 그 과정에서 얻은 교훈은 무엇인가요?

**AI:**

아무래도 저번 개발과는 방향성이 다르다보니, 오류 해결에서 조금 어려움을 겪었던 것 같아요. 다른 AI 팀원과 함께 했던 저번 프로젝트와 달리 AI 팀원이 저 혼자 뿐이다보니 갈팡질팡했던 것도 같고요. 이를 해결하기 위해서 오류 자체를 더욱 꼼꼼히 보는 버릇을 들이려고 노력했고, 팀 내부의 회고뿐 아니라 개인 회고를 꾸준히 진행하면서 해결했던 것 같습니다.

그 과정에서 코드를 확실히 이해하고 어느 부분에서 문제가 발생했는지 파악하는 게 정말! 중요하다는 걸 새삼 깨닫게 되었습니다. 같은 오류라도 발생하는 상황이 다르고, 다른 사람이 사용한 방법을 통해 해결되지 않을 수도 있기 때문에 스스로의 오류를 잘 파악하는 게 중요한 것 같아요.

---

## Winter Dev 프로젝트에서 개발자로 참여한 경험을 설명해 주세요. 어떤 역할을 맡았고, 주요 기술 스택은 어떻게 구성되었나요?

**FE:**

안녕하세요. Black Company 프론트엔드 개발자입니다! 저는 이번 EEOS 프로젝트에서 프론트엔드 개발과 웹 UI 디자인을 담당했습니다. 디자이너가 없는 팀이었기 때문에 PM, AOS 개발자 그리고 FE 개발자 협력하여 디자인을 완성시켰습니다! 저 혼자 UI 디자인을 담당했었을 때보다 PM과 함께 디자인 작업을 진행하니, 사용자의 니즈를 더 고려하면서 디자인을 할 수 있던 것 같습니다. 아쉬웠던 점은 반응형 UI를 디자인하지 않고 개발에 들어갔기 때문에, 개발 과정에서 디자인 수정과 함께 반응형 구현을 진행해서 시간이 오래 걸렸던 것 같아요.

EEOS 프로젝트 기술 스택은 크게 Typescript, Next.js, React query 등을 사용했어요. Typescript와 Next.js에 대한 지식이 많지 않아 프로젝트를 진행하며 습득하기 위해 선택한 기술 스택입니다. Next.js를 사용하며 app router과 같은 Next.js에서 제공하는 기능을 사용하려고 노력했습니다. 다음 학기 프로젝트에서는 Next.js가 제공하는 기능을 더 많이 사용해보고 싶습니다!

---

## 개발자로서의 역량 향상을 위해 어떤 노력을 기울였으며, 이 프로젝트를 통해 어떤 기술적 성장을 이루었나요?

### AOS:

저는 이번 프로젝트에서 역량 향상을 위해 노력했던 부분을 크게 두 가지 측면으로 나누어 이야기하고 싶습니다. 첫째로 **기술적 측면**입니다. 이번 EEOS Android 프로젝트는 안드로이드 권장 앱 아키텍처를 지키기 위해 노력했습니다. 이를 위해 Domain Layer와 Data Layer를 도입하고 Dependency Injection 라이브러리인 Hilt를 사용해 보았습니다. 이 과정에서 각 계층이 실제로 어떻게 사용되는지 확인하기 위해 깃허브 public 레퍼지토리를 주로 참고하였습니다. 이를 통해 서로 다른 계층 사이의 커뮤니케이션이 어떻게 이루어지는지 알게 되었고, 타인이 작성한 코드를 읽는 능력을 향상할 수 있었습니다.

둘째로 **프로젝트의 유지 및 보수 측면**입니다. 먼저 기존 기술 스택인 Jetpack Compose를 더 잘 다룰 수 있도록 노력하였습니다. 차후 유지보수가 수월하도록 컴포넌트 단위의 분리에 신경을 썼고, 디렉터리 구조를 정리하고, 네이밍의 통일성을 가져가기 위해 노력했습니다. 또한 문서화 작업에도 주의를 기울였습니다. 초기 EEOS 팀은 특정 인원이 프로젝트를 그만두게 되더라도, 프로젝트 자체는 유지 및 운영이 되기를 바란다는 공통의 목표를 공유하였습니다. 이를 위해 저희 팀에서 생각했던 방안 중 하나가 문서화를 잘해두자는 것이었습니다. 문서화 과정에서 새로운 팀원이 들어왔을 때, 어떻게 프로젝트를 시작하면 되는지와 어려움이 있을 수 있는 부분에 대한 개념을 정리해 두었습니다. 이러한 작업을 통해 새로운 팀원이 들어왔을 때 해당 팀원의 온보딩이 더 수월하게 진행되기를 바랐습니다. 이렇게 유지보수 측면에 대한 고민이 있었기 때문에, 지속적으로 사용되는 서비스를 개발하는 역량을 향상할 수 있었습니다.