

“CallIt”

Team : MINIONZ



목 차



| | |
|------------|-------|
| 1. 프로젝트 개요 | 7 - 3 |
|------------|-------|

1.1 소 개

1.2 배 경

1.3 시나리오

| | |
|----------|-------|
| 2. 기술 스택 | 7 - 6 |
|----------|-------|

프로젝트 개요

1.1 소 개

IT 분야에서 협업과 소통의 중요성이 점차 강조됨에 따라, 개발자들을 위한 효과적인 일정 관리 서비스에 대한 수요가 꾸준히 증가하고 있습니다. 하지만, 현재 사용되고 있는 다양한 협업 도구들이 각기 분산되어 있어, 오히려 업무 효율성을 저하시키는 문제점이 존재합니다.

이러한 문제를 해결하기 위해 고안된 “Calit” 서비스는 통합적인 IT 일정 관리 플랫폼으로, 개발자들과 팀원들이 보다 효율적으로 협업할 수 있도록 설계되었습니다. 이 플랫폼은 개발자들의 일정 관리 기능을 비롯해, 각 팀의 스크럼 미팅 일정을 체계적으로 기록하고 관리할 수 있는 기능을 제공합니다. 또한, 메신저를 통한 실시간 피드백이 가능하여, 팀원 간의 원활한 소통을 지원합니다.

특히, “Calit”는 개발 프로젝트 관리에 특화된 기능을 제공하는데, 코드 블록 및 파일 전송 기능을 통해 개발자들이 더욱 효율적으로 협업할 수 있도록 도와줍니다. 팀원들은 할일 목록과 진행 상황을 손쉽게 공유할 수 있어, 프로젝트의 전반적인 진행 상태를 명확하게 파악하고 조정할 수 있습니다.

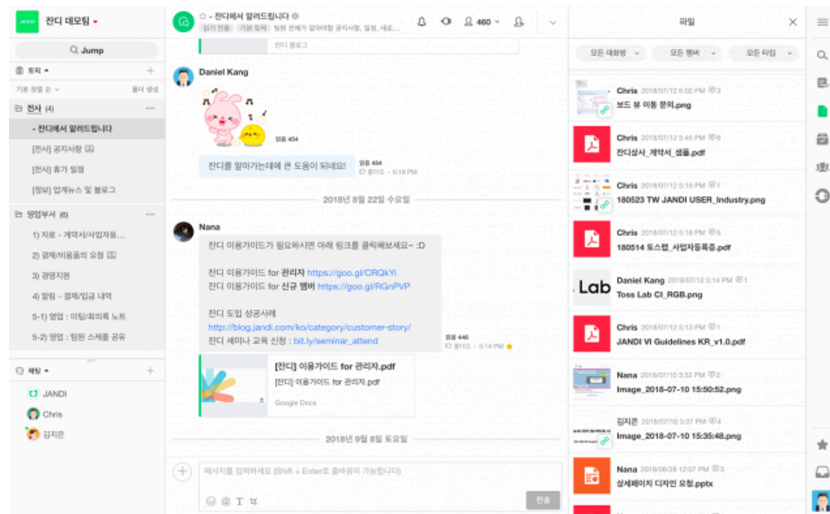
“Calit”는 기존의 분산된 협업 도구들의 단점을 보완하며, 모든 필요한 기능을 하나의 플랫폼에서 제공함으로써, 개발자들의 업무 효율성과 협업 능력을 극대화하는 데 기여할 것입니다.

1.2 배 경

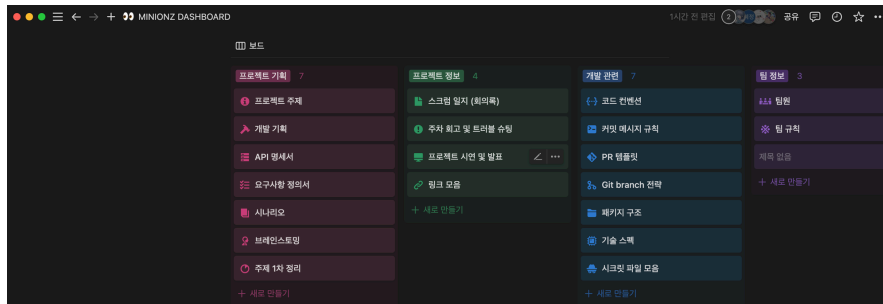
IT 업계는 급속한 성장과 함께 점점 더 높은 아웃풋을 요구받고 있습니다. 이러한 성장 속에서 다양한 분야의 협업 중요성이 크게 부각되고 있으며, 이를 뒷받침하는 협업 툴의 필요성도 함께 증가하고 있습니다. 그러나 현재 시장에서 널리 사용되고 있는 주요 협업 툴 애플리케이션들, 예를 들어 노션(Notion), 지라(JIRA), 네이버 워크스(Naver Works), 슬랙(Slack) 등은 각기 특정 기능에 특화되어 있습니다. 이러한 전문화된 기능들은 특정 작업에 있어서는 강력한 성능을 발휘하지만, 프로젝트 관리와 협업 전반에 걸쳐 다양한 기능을 활용해야 하는 사용자들에게는 오히려

비효율적인 측면이 있습니다. 즉, 여러 툴을 병행하여 사용해야 하는 불편함이 존재한다는 것입니다.

특히, 오늘날 IT 프로젝트에서 중요한 스크럼 회의의 필요성을 고려할 때, 이러한 협업 툴들은 부족한 점이 많습니다. 스크럼 회의는 팀이 짧은 주기로 진행 상황을 점검하고, 문제를 해결하며 다음 작업을 계획하는 중요한 과정입니다. 하지만 기존의 협업 툴들은 스크럼 회의의 효과적인 진행을 돕기에는 통합성과 사용 편의성 측면에서 한계가 있습니다. 각 툴이 개별적으로 강력한 기능을 제공하는 것은 사실이지만, 스크럼의 핵심인 실시간 정보 공유와 일관된 협업을 지원하기에는 부족한 면이 있습니다.



채팅에 특화되어 있는 잔디 애플리케이션 실시간 피드백 및 채팅이 불가능한 지라 애플리케이션



특정 팀간의 일정 공유가 불가능한 노션 애플리케이션

이러한 문제들은 특히 개발자들과 그들의 팀에 큰 부담으로 작용할 수 있습니다. 프로젝트를 원활하게 진행하기 위해서는 일정 관리, 실시간 피드백, 회의록 작성, QA 관리 등 다양한 작업을 신속하고 효과적으로 수행해야 합니다. 하지만, 각각의 작업에 따라 다른 툴을 사용하게 되면, 정보의 일관성이 떨어지고, 협업 과정에서 혼선이 발생할 수 있으며, 나아가 프로젝트의 속도와 품질에도 부정적인 영향을 미칠 가능성이 큼니다.

이러한 상황에서, 우리는 실제 개발 현장에서의 다양한 프로젝트 경험을 바탕으로, 개발자들에게 필수적인 기능들을 하나의 애플리케이션에서 통합적으로 제공할 수 있는 프로그램을 기획하게 되었습니다. 이 프로그램은 개발팀의 원활한 소통과 협업을 지원하기 위해 다음과 같은 주요 기능들을 포함하도록 설계되었습니다.

1. **개발 일정 및 팀 일정 관리:** 프로젝트 전반의 일정을 한눈에 파악하고, 팀원들과의 일정 조율을 용이하게 할 수 있는 캘린더 기능을 제공합니다. 이를 통해 중요한 기한을 놓치지 않고, 체계적으로 업무를 진행할 수 있습니다.
2. **회의록 및 클라이언트 QA 관리:** 프로젝트 관리에서 중요한 회의록 작성과 클라이언트 피드백 관리를 위한 게시판 기능을 통해, 모든 관련 정보를 체계적으로 정리하고 공유할 수 있습니다. 이를 통해 프로젝트 진행 상황을 투명하게 공유하고, 필요한 작업을 명확히 할 수 있습니다.
3. **실시간 피드백 제공:** 팀원들 간의 즉각적인 소통을 지원하는 채팅 기능을 통해, 신속한 피드백과 문제 해결이 가능해집니다. 이는 협업의 효율성을 극대화하고, 문제 발생 시 빠른 대응을 가능하게 합니다.

이러한 기능들을 하나의 애플리케이션에서 관리할 수 있게 함으로써, 우리는 사용자들이 여러 툴을 번갈아가며 사용하는 번거로움을 줄이고, 업무의 일관성과 효율성을 크게 향상시킬 수 있을 것으로 기대하고 있습니다. 통합적인 솔루션 제공을 통해, 프로젝트 관리와 협업에 있어 새로운 표준을 제시하며, IT 업계에서의 생산성과 협업 효율을 극대화하는 데 기여하고자 합니다.

1.3 시나리오

- "calit" 서비스 사용 [사용자] 시나리오

1. 개발 일정 및 팀 일정 관리

※ 누구든 팀 워크스페이스를 생성할 수 있다. -> [관리자] 부여

관리자 :

- 팀원들을 초대할 수 있다.
- 워크 스페이스를 삭제하고 수정할 수 있다.
- 스크럼을 생성, 조회, 수정, 삭제 할 수 있다.
- 스프린트, 이슈를 생성, 조회, 수정, 삭제 할 수 있다.

사용자 :

- 팀 워크 스페이스 초대를 수락하거나 거절할 수 있다.
- 스크럼을 생성, 조회, 수정, 삭제 할 수 있다.
- 스프린트, 이슈를 생성, 조회, 수정, 삭제 할 수 있다.

2. 회의록 및 클라이언트 QA 관리:

※ 모든 사용자가 이용할 수 있는 기능이다.

사용자 :

- 회의록은 모두가 생성, 조회, 수정, 삭제할 수 있다.
(사용자1이 생성하지 않아도 사용자2가 조회 수정 삭제 가능하다.)
- 에러 게시판은 모두가 생성, 조회, 수정, 삭제할 수 있다.
(사용자1이 생성했으면 수정,삭제는 사용자1만 가능하다.)
- QA 게시판은 모두가 생성, 조회, 수정 삭제할 수 있다.
(사용자1이 생성했으면 수정,삭제는 사용자1만 가능하다.)

3. 실시간 피드백 제공:

※ 누구든 단체 채팅방을 생성할 수 있다. -> [관리자] 부여

관리자 :

- 팀원들을 초대할 수 있다.
- 채팅방을 삭제하고 수정할 수 있다.
- 관리자가 나가거나, 삭제하면 채팅방을 더 이상 할 수 없다.

사용자 :

- 채팅방 초대를 수락하거나 거절할 수 있다.

모든 사용자 :

- 채팅방을 조회하거나, 나갈 수 있다.
- 채팅을 보내거나, 수정, 수정, 삭제를 할 수 있다.
- 채팅 대상을 검색할 수 있다.
- 채팅에 코드 블록, 이미지, 파일 등을 전송할 수 있다.

기술 스택

2.1 데이터베이스

- 1) 운영체제 : Linux
- 2) 데이터베이스 : Maria DB 10.3.28
- 3) 컨테이너화 플랫폼 : Docker
- 4) 오케스트레이션 툴 : Kubernetes (k8s)

2.2 프론트엔드

- 1) 프로그래밍 언어 : HTML, CSS, TypeScript
- 2) 웹 프레임워크 : Vue 3.0
- 3) 웹 서버 : Nginx 1.22.1
- 4) 상태 관리 라이브러리 : Pinia
- 5) 클라우드 플랫폼 : AWS S3
- 6) 컨테이너화 플랫폼 : Docker
- 7) 오케스트레이션 툴 : Kubernetes (k8s)
- 8) 형상 관리 : Git, GitHub

2.3 백엔드

- 1) 프로그래밍 언어 : Java openJdk 17
- 2) 웹 프레임워크 : Spring Boot 3.3.3, Spring Security
- 3) 클라우드 플랫폼 : AWS S3
- 4) 컨테이너화 플랫폼 : Docker
- 5) 오케스트레이션 툴 : Kubernetes (k8s)
- 6) 형상 관리 : Git, GitHub

2.4 데브옵스

- 1) CI/CD 도구 : Jenkins, GitHub Actions
- 2) 테스트 프레임워크
 - 프론트엔드 : Selenium, Jest
 - 백엔드 : Junit
- 3) Jenkins Pipeline 알람 : Slack
- 4) 컨테이너화 플랫폼 : Docker
- 5) 오케스트레이션 툴 : Kubernetes (k8s)
- 6) 형상 관리 : Git, GitHub