**소프트웨어 개발 계획서**

Noty 개발 설계서

2019.09.06

이근혁

1. 개발 배경 및 개요

저는 작업한 내용에 대해 사소한 것이라도 기록 하자는 목표를 가지고 있고 종종 할 일과 진행한 작업 내용에 대해 기록하고 있습니다. 개인적인 학습 계획이나 업무 또는 학업 등 다양한 내용에 대해 기록하고, 진행한 내용에 대해서는 간단한 메모와 같이 기록하는 습관을 들이려고 하고 있습니다. 하지만 이것 저것 바쁘게 하다 보면 종종 잊을 때가 있고 한 번 놓치면 지난 작업 내용에 대해 잊어버리거나 밀리게 되어 기록하기 어려운 부분이 있었습니다. 이로 인해 일상에서 해야할 일을 기록하고 효율적으로 관리하는 서비스에 대해 고민을 하게 되었습니다. 이미 할 일을 관리해주는 비슷한 서비스가 있지만 이번에 구상한 Noty라는 서비스는 할 일 관리 뿐만 아니라, 메모를 기록하고 카테고리별로 분류하여 효율적으로 관리할 수 있도록 새롭게 구상한 서비스 입니다. 이 외에도 다양한 플랫폼에서 언제 어디서든 사용할 수 있도록 웹 기반의 서비스로 계획하게 되었습니다.

웹 기반의 서비스이기 때문에 브라우저가 있는 PC, 모바일 등 어느 기기와 플랫폼에서 사용할 수 있으며, 사용자 별 카테고리와 할 일 목록, 노트 3가지 주요 기능을 제공하여 카테고리별로 일정과 계획을 정리하고 진행한 사항에 대해선 작업 내용을 노트에 따로 정리할 수 있도록 하는 것이 핵심 기능입니다. 또한 할 일 목록과 작성한 노트에 대한 검색 기능을 제공하여 사용자가 추후 어떤 작업을 했는지 궁금할 경우 쉽게 조회할 수 있도록 편의 기능도 제공할 예정입니다.

|  |
| --- |
|  |

[ Noty 기본 기능 3종류 ]

할 일(TODO) 목록에 해야 할 일을 기록하고 작업 완료/진행중 상태를 설정할 수 있습니다. 또한 할 일 목록 마다 태그(카테고리)를 설정하여 태그 별로 관리할 수 있으며 노트 작성 시 할 일을 링크할 수 있는 편의 기능을 제공합니다. 노트 – 할 일 – 태그 기능의 밀접한 연관성을 통해 효율적으로 관리하고 기록할 수 있도록 서비스를 제공합니다.

1. 개발 목표

다양한 최신 웹 기술을 활용하여 기존의 웹 애플리케이션보다 더 나은 사용자 경험을 제공하는 것이 목표입니다. SPA(Single Page Application) 기반의PWA(Progressive Web App)로 구현하여 웹 환경에서 앱을 사용하는 듯한 사용자 경험을 제공할 예정이며 오프라인 환경에서 웹 서비스를 사용하거나 푸시 알림을 제공하는 등의 다양한 기술을 적용할 계획입니다.

더 나아가 서버와 클라이언트간의 데이터 송/수신을 위해 주로 사용되던 RESTful API 방식 대신 새롭게 떠오르고 있는 GraphQL을 활용하여 RESTful API의 단점을 보완하고 더 나은 사용성을 제공할 계획입니다.

최종적인 서비스는, 오프라인 상태에서도 본 서비스를 사용할 수 있으며 일반 앱과 같이 푸시 알림을 받아 사용자가 서비스에 재참여 할 수 있도록 유도하는 기능을 구현하고, 앞서 설명한 주요 기능들을 제공하여 앱을 설치하는 번거로움 없이 다양하 플랫폼에서 Noty를 제공하여 할 일과 메모 관리를 쉽고 간편하게 할 수 있도록 제공하는 것이 목표입니다.

이와 더불어 모던 웹 개발에 사용되고 있는 프레임워크와 최신 웹 기술을 접목하여 서비스를 구현해 봄으로써 새로운 기술과 지식을 습득하고 추후 웹 생태계의 발전에 기여할 수 있는 개발자가 되는 것 또한 하나의 목표입니다.

1. 예상 결과물

* (모든 사진은 모바일 페이지 기준입니다.)

|  |  |
| --- | --- |
|  | 로그인 화면 |
|  | 회원가입 화면 |
|  | 메인 화면 (노트)  노트 작성시 # 문자를 사용하여 TODO를 노트에 링크할 수 있음 |
|  | 메인 화면 (Todo)  완료 체크표시  Todo 항목별 ID  내용  태그  완료된 항목 또는 완료되지 않은 항목만 필터링하여 확인  특정 태그만 필터링하여 확인 |
|  | 메인 화면 (Tag)  태그를 추가하거나 삭제, 수정할 수 있음 |
|  | 메뉴 |
|  | 설정 메뉴 |
|  | 대화상자 |
|  | Noty 색상 팔레트 |

1. 설계 및 계획

|  |
| --- |
|  |

[ Noty 전체 아키텍처 ]

* **서버**
  + Node.js 환경의 Express 프레임워크를 사용하여 웹 애플리케이션 구현
  + graphql-yoga 라이브러리를 사용하여 GraphQL 서버 구축
  + 데이터베이스는 NoSQL DBMS중 MongoDB를 사용하며 Docker에서 작동하도록 구성
  + Prisma를 통해 데이터베이스를 추상화하고, Prisma 클라이언트와 GraphQL 서버를 연동하여 CRUD 기능 구현
  + FCM(Firebase Cloud Messaging)과 연동하여 웹 브라우저에 푸시 알림 전달
* **클라이언트**
  + SPA 프레임워크로 Vue.js를 사용하며 Vuex, Vue-Router와 같은 Vue 에코시스템 사용
  + 서버와 데이터를 주고받기 위해 GraphQL을 사용하며 GraphQL 클라이언트는 Apollo를 사용하여 구현
  + 서비스워커를 통해 PWA의 주요 기능 구현 (푸시 알림을 수신하고, 웹 페이지 리소스를 캐싱하여 오프라인 환경에서 사용 가능하게 하거나, 빠른 웹 페이지 로드 환경을 제공)

|  |
| --- |
|  |

[ Server – Client 인증 방식 ]

* **인증**
  + JWT 기반의 사용자 인증 방식을 사용
  + GraphQL을 통한 데이터 접근은 JWT 인증으로 권한 제어
* **진행 계획**
  + 총 15주차의 기간이 주어졌으며 초기에는 기술적인 검토와 설계를 진행하여 문서로 작성
  + 설계가 마무리되면 개발을 시작할 예정이며 서버 및 클라이언트 개발 일정은 유동적으로 변동될 것으로 예상됩니다.
  + 매주 진행상황에 대해 주차별 학습일지에 기록하여 제출합니다.