

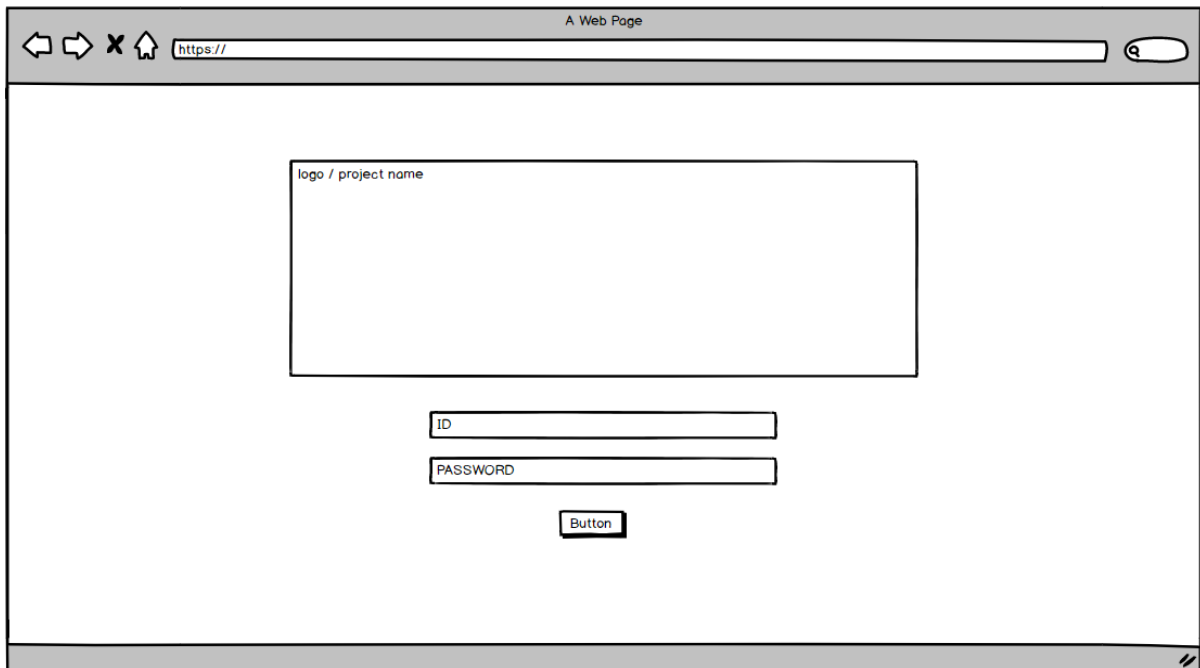
소프트웨어 구조 설계서

1. 화면 정의
2. 프로토타입 소프트웨어 구조 설계
 - A. 데이터 흐름도
 - B. Vue Component diagram
3. 컴포넌트 설명
4. UI 구축 및 사용자 경험 설계

1. 화면 정의

Single Page Application으로 설계하여, 각 화면은 라우터를 이용해 렌더링되거나, Vue가 제공하는 방법을 통해 html 갱신이 아닌, 화면의 컴포넌트를 렌더링하여 사용자에게 표시된다. 사용자가 경험하는 화면은 작은 컴포넌트가 바뀌는 단위를 제외하고 크게 4가지이다.

<화면1: 로그인 페이지>



라우터 가드에 의해, 사용자가 접근하게 될 첫 페이지이다. 프로토타입 내에서는 store에 보관되어있는 users의 ID, PassWord를 입력하면 접속이 가능하게 구현할 것이다.

해당화면은 구글형 디자인으로, 화면이 복잡하지 않고, 해당 페이지의 기능과 역할을 분명히 표시할 수 있고, 깔끔하되 허전하지 않도록, 로고를 제작하여 부착한다.

<화면2:검색 및 열람>

The screenshot shows a web application titled "A Web Page". It features a search bar with the text "search" and a "Button". Below the search bar, there is a section titled "기본 : 전체 검색 : 태그와 인덱스를 통한 검색(DB에 테이블을 따로 두어 검색)". The main content area displays a list of documents, each with a title and a brief description. A "작성제" (Author) field is visible. The interface also includes a "One Two" toggle and a "검색" (Search) button.

두번째 페이지는 문서를 검색하고, 해당하는 문서에 엮인 여러 문서들을 보는 화면이다. 왼쪽에는 검색창이 있으며, 기본적으로는 모든 문서를 표시한다. 현재의 프로토 타입을 지나, 데이터베이스를 구현하여 연결하면, 데이터베이스로부터 받은 부서정보를 이용하여 열람할 수 있는 문서만을 표현하도록 한다.

<화면3 문서 수정 및 작성>

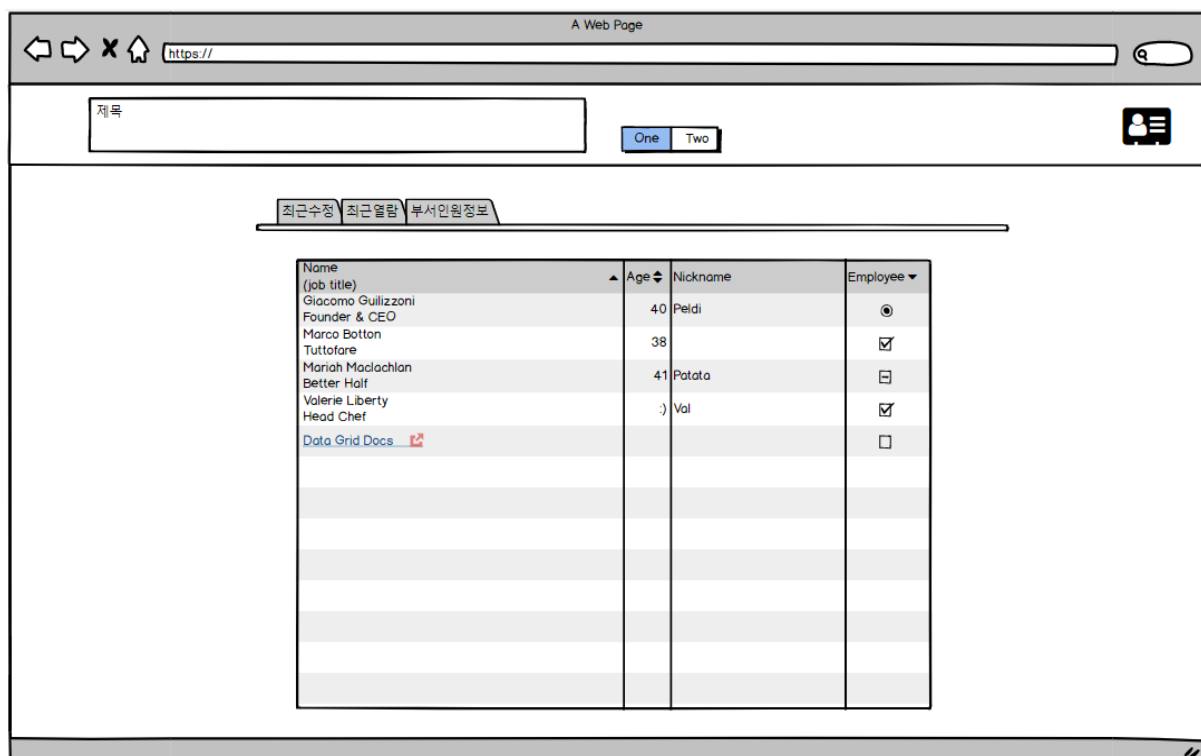
The screenshot shows a web application titled "A Web Page". It features a search bar with the text "search" and a "Button". Below the search bar, there is a section titled "기본 : 전체 검색 : 태그와 인덱스를 통한 검색(DB에 테이블을 따로 두어 검색)". The main content area displays a list of documents, each with a title and a brief description. A "작성제" (Author) field is visible. The interface also includes a "One Two" toggle and a "검색" (Search) button. The document view area shows a detailed view of a selected document, including a "수정" (Edit) button and a "작성제" (Author) field.

화면2에서 수정버튼이나 +버튼을 누르면 오른쪽 컴포넌트를 글 작성 컴포넌트로 바꾸어 준다.

이때, ManualHub의 수정은 일반적인 게시판과는 다르다. 왼쪽에 리스트에 항목을 파일철 하나라고 연상하면, 생각하기 쉽다. 하나의 파일철에 해당 파일철에 들어있는 글들을 수정하여 넣을 수 있는 기능이 이 웹에서의 수정의 의미이다.

해당 기능은, 동일한 양식이나 포맷을 가진 문서들을 재작성하기 쉽고, 통합적으로 모을 수 있게 도와주는 핵심 기능이다.

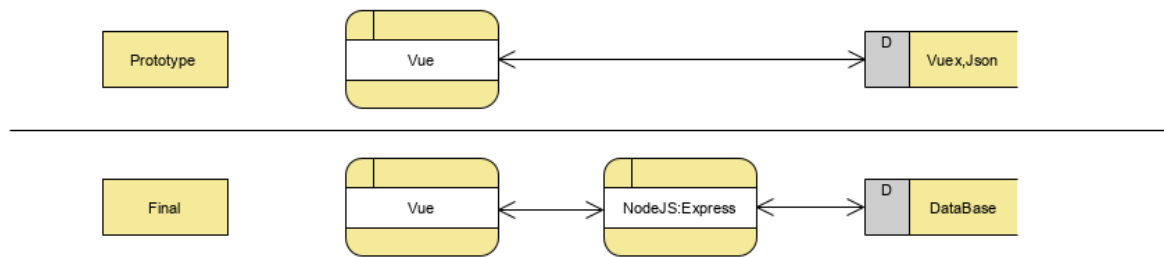
<화면4 : 최근 열람 목록 및 수정목록 사용자 부서원 정보>



사용자의 정보와, 최근 수정된 목록, 열람된 목록 등 히스토리의 일부를 보여주는 기능을 담은 페이지이다. 이 페이지에서 열람되거나 수정된 문서가 있는 표의 줄을 누르면, 그 문서를 열도록 구현한다. 해당 페이지는, 일지나 신청서 등 여러 글에 쓰여진 최근 목록(즉, 최근 일지나 최근에 제출된 신청서)을 관리자가 편하게 확인할 수 있도록 도와주는 기능을 한다.

2. 프로토타입 소프트웨어 구조 설계

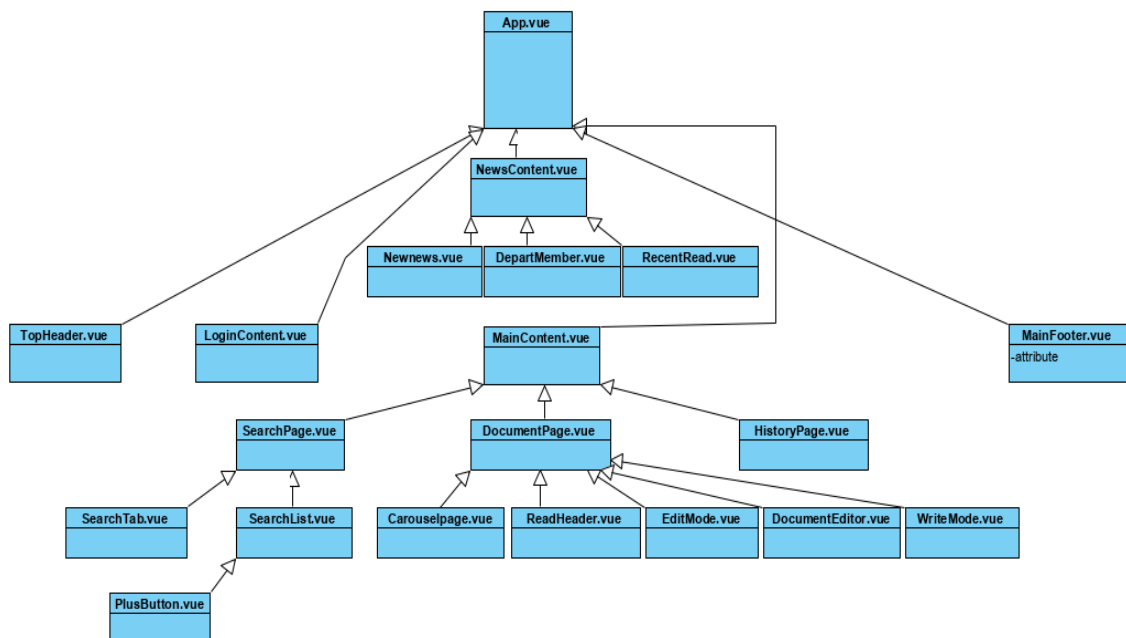
<A> 데이터 흐름도



<통신 및 데이터 흐름도>

프로토타입의 구현은, 해당 프로젝트의 프론트엔드를 개발하여, 해당 프로젝트가 유의미한 서비스를 제공할 수 있는 지를 우선 파악 후, 서버와 데이터베이스를 구축하는 방식을 위함으로, 사용자의 경험을 통한 구체적 평가를 위해, 중요 기능들의 구현을 중점으로 둔다. 해당 개발캠프는 프로토타입 구현에 집중한다.

 Vue component diagram



<Vue Component Diagram>

3. 컴포넌트

위의 컴포넌트 다이어그램을 보면, App.vue를 루트노드로 화면을 header, main, footer로 나눈다.
Main은 라우터에 의해 login창, news창, 열람창으로 렌더링된다

디렉토리	파일	설명
public/image	0~3.jpg	사용자 그룹별 이미지
public/image	그외 .jpg 파일	MOCKUP 데이터에
src	App.vue	Vue 컴포넌트들의 Head
src/components	store.js	vuex 글로벌 변수
src/components	TopHeader.vue	페이지 헤더
src/components	MainFooter.vue	페이지 푸터
src/components	MainContent.vue	메인 read 페이지
src/components	NewsContent.vue	메인 news 페이지
src/components	LoginContent.vue	메인 로그인 페이지
src/components	SearchPage.vue	read 페이지의 search
src/components	SearchTab.vue	search 페이지 검색창
src/components	SearchList.vue	search 페이지
src/components	PlusButton.vue	글쓰기 버튼
src/components	HistoryPage.vue	해당 문서 작업 이력
src/components	DocumentPage.vue	read 페이지의 문서창
src/components	ReadHeader.vue	document 페이지의
src/components	CarouselPage.vue	document 페이지의
src/components	DocumentEditor.vue	문서 Editor 페이지
src/components	EditMode.vue	Editor 페이지의
src/components	WriteMode.vue	글 작성 Header
src/components/newscomponent	DepartMember.vue	news 페이지 부서원 탭
src/components/newscomponent	Newnews.vue	news 페이지 최근 추가된
src/components/newscomponent	RecentRead.vue	news 페이지 읽었던 문서

4. UI 구축 및 사용자 경험 설계

해당 서비스를 사용하는 이들은 특정 일지, 양식에 대한 문서를 통합관리 하기 위하여 사용할 것이다. 따라서, 한 파일철에 들어가 책을 읽듯이 슬라이드로 다음장으로 넘어갈 수 있도록 UI를 설계해야한다.

한 파일철에 대한 슬라이드는 아래 element-ui를 이용하여 구현한다.

Arrows

You can define when arrows are displayed



예를 들면, 신청서 파일철이 하나 있다면, 좌우로 움직여 해당 파일철의 문서들을 볼 수 있고, 수정버튼을 누르면 현재 파일을 수정하여 추가작업 할 수 있도록 한다.

또한, 검색과, 최근기록로그가 찍힌 문서로의 접근이 손쉽도록 해야한다.