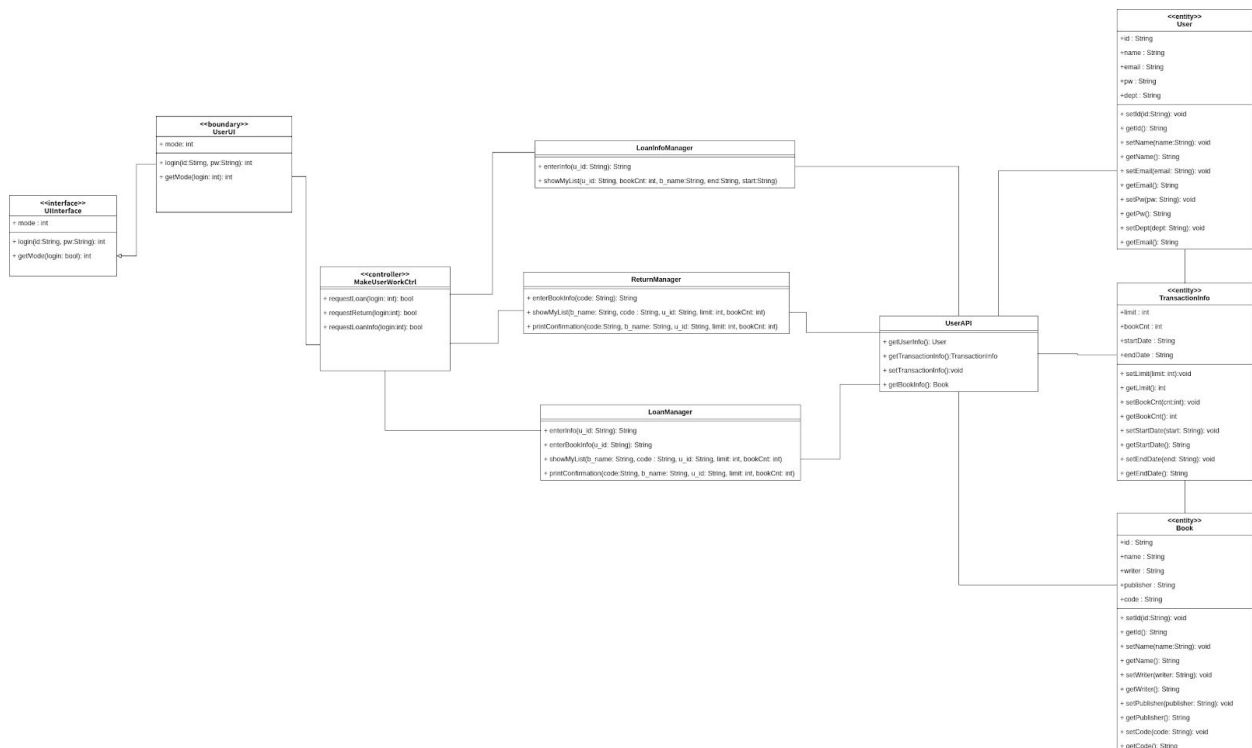


HW5: FINAL Project and FINAL Report

20171677 소프트웨어학부 이정하

소프트웨어 시스템 구성도 및 설명

- Class Diagram



[그림 1] 무인 도서 대출,반납시스템 Class Diagram

클래스 다이어그램에서는 구체적인 flow를 보이기 위해 구체적으로 작성하였다. 크게 일반 사용자(User)와 관리자(Admin)으로 나누었다. 일반사용자는 UI를 이용하여 로그인하고 이후 대출, 반납, 대출현황 확인 서비스를 사용할 수 있다. 그리고 이 작업들의 요청을 기반으로 controller가 작업에 맞게 manager class를 호출한다. manager 클래스는 해당하는 정보를 얻기 위해 API를 호출하여 데이터에 접근한다. HW4에서 계획했던 Scope에서 축소시켜 사용자 기능만 개발하게 되었다.

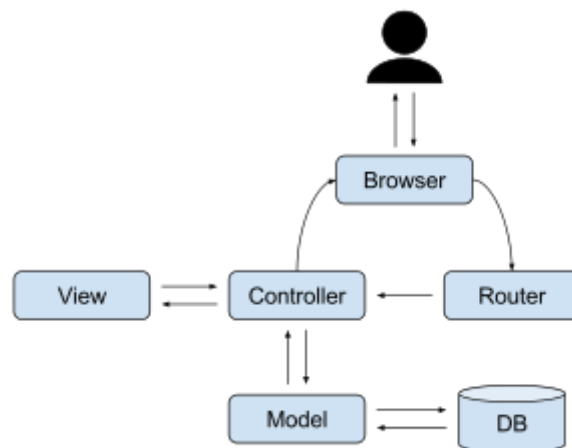
사용하는 Software architecture

- Serverless architecture

Serverless Architecture는 타사 서비스에서 애플리케이션을 호스팅하는 소프트웨어 디자인 패턴으로, 개발자가 서버 소프트웨어 및 하드웨어 관리를 할 필요가 없다. 응용프로그램은 개별적으로 호출하고 확장할 수 있는 개별 기능으로 나뉜다.

인터넷에서 소프트웨어 응용 프로그램을 호스팅하려면 일반적으로 일종의 서버 인프라를 관리해야한다. 일반적으로 이는 관리해야하는 가상 또는 물리적 서버뿐만 아니라 응용 프로그램을 실행하는 데 필요한 운영체제 및 기타 웹 서버 호스팅 프로세스를 의미한다. 클라우드 제공 업체의 가상 서버를 사용한다는 것은 물리적 하드웨어 문제를 제거하는 것을 의미하지만 운영체제 및 웹 서버 소프트웨어 프로세스의 일부 수준의 관리가 여전히 필요하다.

- **MVC Architecture**



[그림 2] 웹 어플리케이션에서 일반적인 MVC구성요소 다이어그램

모델-뷰-컨트롤러는 응용프로그램을 세 가지의 구성요소로 나눈다. 각각의 구성요소들 사이에는 다음과 같은 관계가 있다.

뷰는 사용자가 볼 결과물을 생성하기 위해 모델로부터 정보를 얻어 온다.

컨트롤러는 모델에 명령을 보냄으로써 모델의 상태를 변경할 수 있다. 또, 컨트롤러가 관련된 뷰에 명령을 보냄으로써 모델의 표시 방법을 바꿀 수 있다.

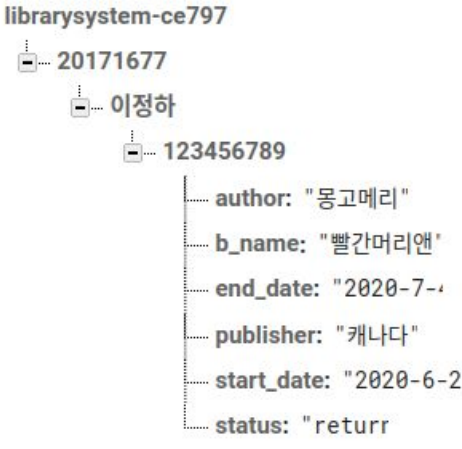
모델은 모델의 상태에 변화가 있을 때 컨트롤러의 뷰에 이를 통보한다. 이와 같은 통보를 통해서 뷰는 최신의 결과를 보여줄 수 있고, 컨트롤러는 모델의 변화에 따른 적용 가능한 명령을 추가, 제거, 수정할 수 있다. 어떤 MVC 구현에서는 통보 대신 뷰나 컨트롤러가 직접 모델의 상태를 읽어오기도 한다.

이 프로젝트에서 컨트롤러와 모델은 Firebase DB이며 뷰는 React.js로 개발된 웹 UX/UI라고 볼 수 있다.

프로젝트하면서 어려웠던 점

첫 번째, DB를 설계하는 과정이 어려웠다. Firebase DB는 json형식으로 되어있어 어떻게 해야 효율적인지 고민하게 되었다. 그래서 unique key인 사용자의 학번을 기준으로 데이터를 저장하기로 결정하였다. 아래에 설계한 Firebase DB구조와 실제 Firebase DB에 저장된 데이터를 비교하였다.

(Firebase DB가 json key를 기준으로 자동 정렬하여 저장한다.)

<pre>{학번: {이름: {도서 일련번호: {author: 저자이름, b_name:책 이름, end_date:반납일, publisher: 출판사, start_date:대출시작일, status: 대출반납상태 } } } }</pre>	
---	---

두 번째, React.js를 이용하여 효율적인 작동을 하는 웹을 구현하는 것 또한 고민이었다. Login페이지에서 로그인을 한 후, 대출확인 페이지에서 해당 사용자의 대출, 반납 기록을 DB에서 불러와야 한다. 고심 끝에 React의 Lifecycle을 이용하여 렌더링 전에 데이터를 가져올 수 있도록 구현하였다.

구현코드

이캠퍼스 과제란에 업로드하였습니다.