

# 졸업작품 결과 보고서

Grang

201186003 문화콘텐츠학과 김대성

201415028 생명과학과 정지혁

2018.12.07

## 프로젝트 요약

보고 날짜	프로젝트 이름	팀원
2018 년 11 월 30 일	Grang(그랑)	정지혁, 김대성

## 개발 동기

소셜네트워크서비스(이하 SNS)에 모든 일상을 공유하는 시대에 때때로 SNS 에 남아있는 우리의 흔적들은 예상하지 못했던 문제를 발생시키기도 한다. 뿐만 아니라, 주변 지인들 위주로 구성되는 SNS 의 인맥은 개인적 공간인 SNS 에서조차 자신의 모든 것을 표현할 수 없는 또 하나의 굴레로서 작용하게 된다. 이러한 SNS 의 문제점 들에서 착안하여, 자신의 관심사가 비슷한 사람들과 모든 것을 거리낌없이 표현하며 소통할 수 있는 진정한 의미의 SNS 를 만들기로 하였다.

## 프로젝트 소개

Grang(그랑)은 '사람들과 동그랑게 어우러져 세상을 살자' 라는 순 우리말에서 착안하였다. 그랑은 기본적으로 사진 위주의 게시물을 공유할 수 있는 SNS 에 그 바탕이 있다. 따라서 게시물을 작성하고, 피드에 게시물을 나타내고, 다른 사람들을 구독하며 소통할 수 있는 SNS 의 기본적인 요소를 갖추고 있다. 그러나 다른 SNS 와는 크게 3 가지 점에서 차별화된 성격을 가지고 있다.

우선, 그랑은 다른 SNS 와 달리 자신의 연락처나 학연, 지연 등의 일체의 정보도 제공받지 않는다. 따라서 ID 를 직접 알거나 관심사가 같지 않다면 다른 유저의 게시물을 받아 볼 수가 없다. 대신, 모든 사용자는 아이디 마다 고유의 관심사(Category)를 설정할 수 있으며, 게시물을 올릴 때 또한 해당 게시물의 관심사를 설정하도록 되어있다. 유저는 자신의 아이디와 같은 카테고리로 작성된 게시물들을 피드 제공받게 된다. 따라서 지인 위주의 구성이 아닌 관심사가 같은 사람들끼리 모여 소통하는 진정한 의미의 SNS 공간이라 할 수 있다.

둘째로, 그랑은 게시된지 24 시간이 지난 게시물들은 더이상 피드에 나타내지 않는다. 따라서 기록에 대한 부담에서 완전히 자유로워진 사용자들은 자신의 개인적인 공간에서 훨씬 더 편하고 부담없이 다른 사람들과 소통할 수 있을 것으로 기대된다. 그러나, 그랑의 게시물들은 피드에서 사라져 일반 사용자가 접근할 수 없을 뿐, 모든 기록이 서버에서까지 삭제되는 것은 아니다. 따라서, 향후 범죄가 발생할 경우 수사자료로 활용될 수 있으므로 그랑의 이러한 특징을 악용하여 각종 성범죄나 사기에 악용되는 경우에 대해서도 충분히 대비를 할 수 있을 것으로 기대된다.

셋째로, 그랑은 대화 위주의 SNS 이다. 모든 게시물들은 24 시간 이내에 삭제가 되므로 실시간으로 대화할 수 있는 환경의 SNS 를 조성하는 것이 중요하였다. 따라서 실시간 대화 형식에 바탕을 두었다.

따라서 카카오톡과 같이 나의 대화는 오른쪽에, 상대방의 대화는 왼쪽에 시간순으로 나타나게 된다. 그러나 현재 개발 단계에선, 대화 작성 후 새로고침을 해야 한다는 단점이 존재한다. 향후 서비스를 서버에 실시간으로 연결하면 완전한 의미의 대화용 SNS 가 될 수 있을 것으로 기대된다.

## 향후 보완점

앞서 서술하였 듯, 현재는 새로고침을 할 때마다 새로운 대화가 업로드 되기 때문에, 진정한 대화라고 보기 어렵다. 서버에 연결하여 완벽한 의미의 대화를 구현할 필요가 있다.

아직 사용자 편의성 부분에서 많이 미흡한 부분이 있다. 모든 기록이 서버에서는 삭제되지 않는다는 점을 적극적으로 고지함으로서 사용자들이 범죄에 노출되지 않도록 방지하려는 노력이 필요하다. 또한, SNS 의 특성상 음란물에 취약하다. 따라서 음란물에 대해 빠르고 실시간으로 또는 최대한 신속하게 대응할 수 있는 알고리즘의 개발이 필요하다.

UI 와 디자인적인 부분에서 많이 미흡하다는 지적이 있었다. 향후 업데이트를 통해 사용자 편의 기능들을 더욱 직관적으로 표현함과 동시에 디자인적인 부분을 더욱 발전시키려는 노력이 필요하다.

## 개발 환경

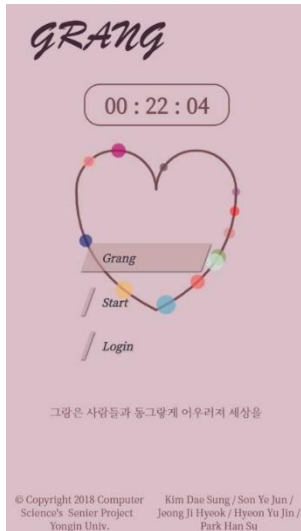
### 김대성 개발환경



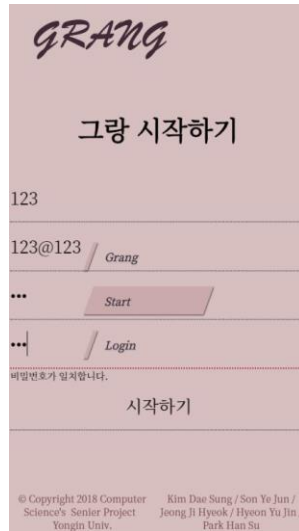
### 정지혁 개발환경



## 정지혁 작품



[메인 화면]



[회원가입]



[관심사 선택]



[로그인후 피드 & 가입후 피드]



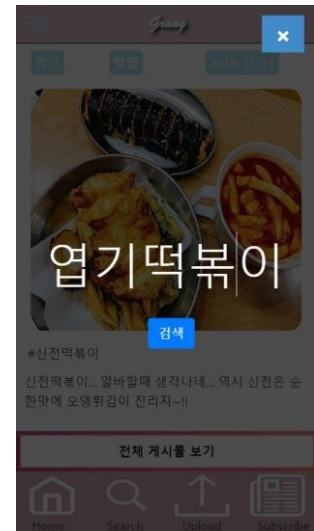
[메뉴 바]



[관심 카테고리 수정]



[관심사 피드]



[검색]

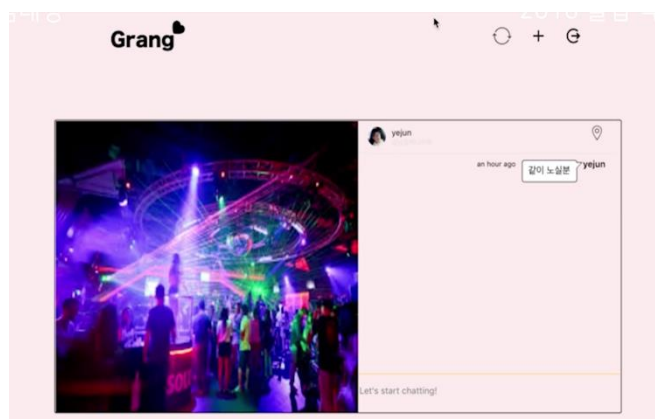


[게시물 업로드]

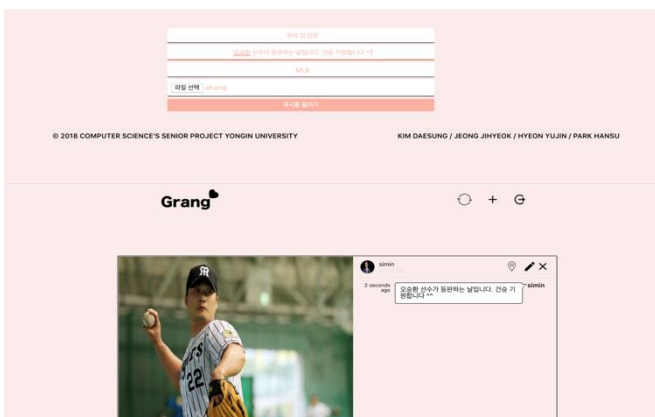
## 김대성 작품



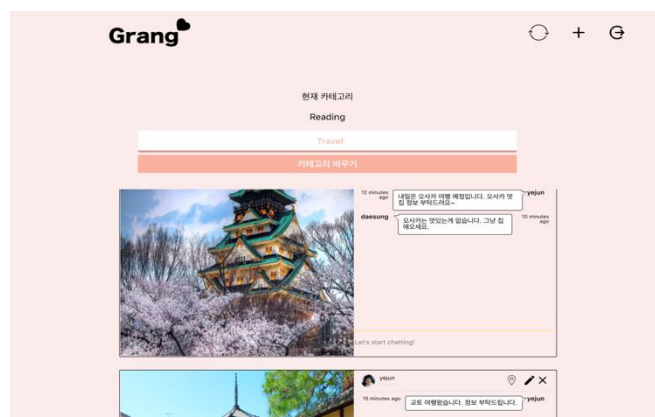
[회원가입 및 로그인화면]



[기본 피드 화면]



[게시물 작성 및 게시물이 나타난 화면]



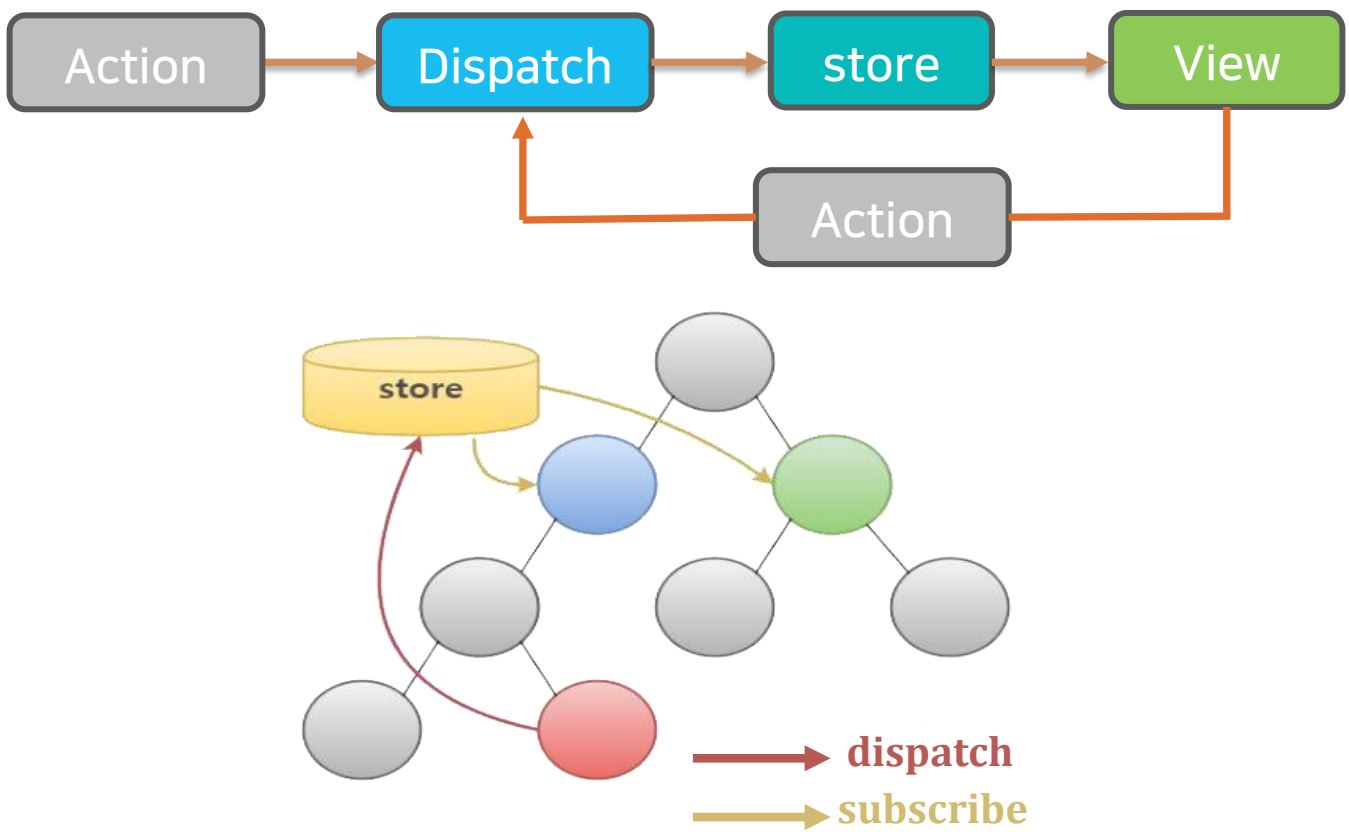
[카테고리 설정 및 카테고리에 따른 피드의 모습]



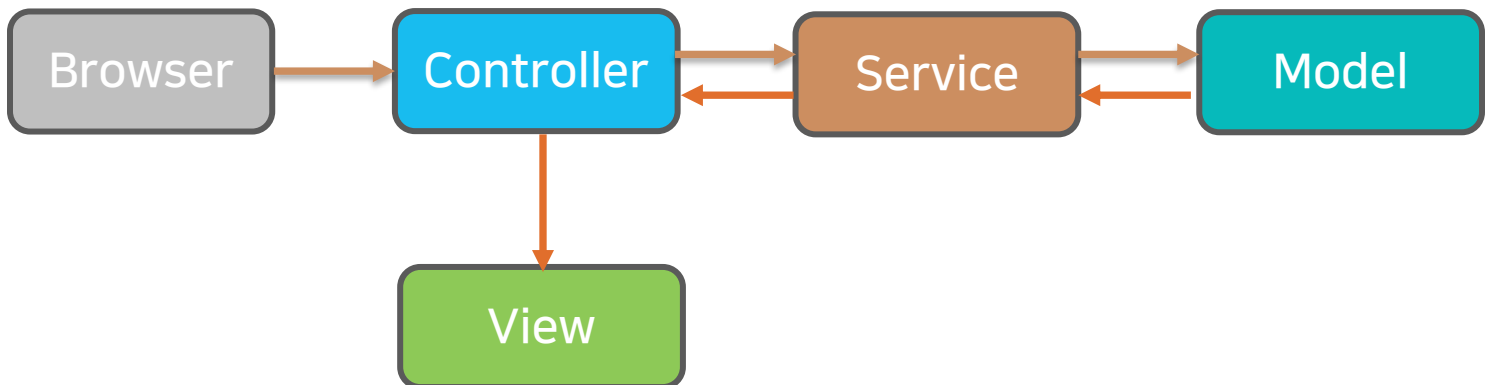
[회원들이 대화하는 모습과 24 시간 후 피드가 삭제된 모습]

## 프로젝트 구조 및 장단점 비교

### React & Django (Redux)



## Servlets & JSP (MVC model2)



	React	Servlets & JSP
유지 보수	자바스크립트 기반의 React 는 크롬의 디버깅 툴을 이용하여 디버깅하기가 쉽고 이벤트가 발생하는 웹 애플리케이션에 최적화되어 있다. 따라서 유지보수가 쉬운 편이다.	뷰가 늘어날수록 복잡성이 높아져 개발자의 높은 수준이 요구된다. 하지만 뷰와 모델의 분리로 큰 프로젝트도 가능하다.
확장성	React 출시 이후 가파른 상승세를 보이는 JS 사용자들을 바탕으로 한 라이브러리를 통해 기능을 확장하는 데 어려움이 없다.	초기 설계를 잘해 두면 기능 확장하는데 어려움이 없다.
가독성	ESLint 를 활용하여, 주로 사용되는 Airbnb 스타일을 적용하여 코드 가독성을 상당히 높일 수 있다.	데이터의 흐름이 직관적이고 코드의 가독성이 높다.
효율성	기능이 다양할수록 기하급수적으로 복잡해지는 MVC 구조와 달리 예측 가능한 코드의 구조를 지향하는 단방향의 Redux 의 구조는 데이터를 불러오고 업데이트 하는 작업에 효율적으로 설계되어 있다.	효율적으로 설계하는 것은 개발자의 개발 능력에 달려있다.

구현 난이도	<p>쉽게 배울 수 있으며 코드의 재사용성을 지향하는 React 구조의 특성상 초기 구조만 잘 파악하면 어렵지 않게 구현해 나갈 수 있으나, 데이터를 전달받는 Redux 구조를 활용하는데 약간의 어려움이 있을 수 있다.</p>	<p>Java 언어의 기반을 두고 있기 때문에 다소 어려움이 있고 초기 설계 과정이 오래 걸리지만 초기 설계 이후에는 빠르게 작업이 가능하다.</p>
구조 설계	<p>React 의 여러가지 Pattern 구조를 활용하여 효율적이고 한눈에 들어오는 구조로 설계할 수 있다.</p>	<p>구조가 직관적이고 구조 설계 시 개발자에 의해 초기 설계가 이루어지기 때문에 개발자의 능력에 따라 효율적인 구조를 설계할 수 있다.</p>
속도	<p>Redux 는 Facebook 에 의해 만들어진 라이브러리이기 때문에 SNS 처럼 이벤트 발생이 많은 웹 앱에서 최적화 되어 있게 만들어졌기 때문에 이벤트 처리 속도가 빠르고 오류가 적은 편이다.</p>	<p>이벤트 처리속도가 빠른 편은 아니지만 알고리즘 설계에 따라 복잡한 기능도 빠르게 처리할 수 있다.</p>
협업	<p>JavaScript 와 React 의 놀라운 성장세와 최신의 동향을 볼 때, 당장은 JAVA 기반의 언어에 비해 협업하기 어려울 수 있으나, 향후엔 더 많은 협업 환경을 기대할 수 있다.</p>	<p>많은 기업에서 Jsp &amp; Servlet 및 Spring 프레임워크를 사용하기 때문에 협업 개발자를 찾기 쉽고 디자이너와 개발자의 분업이 쉬운 편이다.</p>