		<h2 style="text-align: center;">Capstone Design 과제 결과보고서</h2>		
과 목 명	종합설계프로젝트			
과 제 명	킹고와트			
팀 명	정말 간절히 원하면 우주가 나서서 도와주조(4조)			
지도교수 (과제책임자)	학과(부)	컴퓨터공학과	성명	이진규
	E-mail	jinkyu.lee@skku.edu	휴대전화	
팀장 (대표학생)	학과(부)/학년	컴퓨터 공학과/4학년	성명	홍기원
	E-mail	gch02518@skku.edu	휴대전화	010-2502-5054
팀원	학과(부)/학년	컴퓨터 공학과/4학년	성명	황인규
	학과(부)/학년	컴퓨터 공학과/4학년	성명	이현빈
	학과(부)/학년	컴퓨터 공학과/4학년	성명	이수영
	학과(부)/학년	컴퓨터 공학과/4학년	성명	김동민
	학과(부)/학년	컴퓨터 공학과/4학년	성명	하인수
	학과(부)/학년	컴퓨터 공학과/4학년	성명	김준영
프로젝트 참여기업	기업명		주생산품	
	주소			
	담당자		일반전화	
	휴대전화		E-mail	
과 제 수 행 비	구 분	현 금	현 물	
	국 비(LINC+)	700,000원		
	기 업 체			
	합 계	700,000원		
과제기간	2017 년 3 월 일 - 2017 년 6 월 8 일 (4 개월)			
<p>위와 같이, 성균관대학교 산학협력선도대학(LINC+)육성사업단의 Capstone Design 지원사업 과제결과보고서를 제출합니다.</p> <p style="text-align: right;">2017 . 6 . 8 .</p> <p style="text-align: right;">팀 장: 홍기원 (인)</p> <p style="text-align: right;">지도교수: 이진규 (인)</p> <p style="text-align: right;">LINC+사업단장 귀하</p>				

Capstone Design 과제 결과보고서 요약

과 제 명	킹고와트
팀 구성원 및 역할	<p>홍기원 : 팀장, 클라이언트 개발(로그인, 쿼패널)</p> <p>김동민 : 클라이언트 개발(총괄, 공지 게시판, 회원가입, 회원 정보 수정, 검색)</p> <p>김준영 : 클라이언트 개발(지도 서비스-카테고리, 구역 생성 등)</p> <p>이현빈 : 클라이언트 개발(회원가입)</p> <p>하인수 : 서버 개발(서버, 위키)</p> <p>황인규 : 서버 개발(서버, 데이터 크롤링, 검색)</p> <p>이수영 : 클라이언트 개발(구역 다이얼로그, 위키 페이지), PPT 템플릿 제작</p>
개발동기 목적 및 필요성	<p>성균관대학교에 대한 정보를 얻고자 할 때 원하는 정보들이 흩어져 있는 경우가 많다. 특히 공지사항의 경우 특정 커뮤니티를 통해서만 제공되는 경우 정보를 얻는 데 어려움을 겪는다. 뿐만 아니라 건물이나 학교 내외의 시설들에 대해 찾아볼 때도 지나치게 적은 정보만을 담고 있거나 홍보성의 글들로 원하는 내용을 찾기 힘들다. 따라서 성균관대학교에 새로 입학해 이런 정보가 필요한 신입생들이나 학교에 대한 소식을 한 곳에서 확인하고자 하는 재학생 및 모든 성균인에게 보다 효율적으로 정보를 제공하고자 지도를 중심으로 위키 서비스와 통합 공지 사항, 그리고 위치와 관련된 추가 기능을 포함하는 거점 기반 정보 서비스인 킹고와트를 개발하고자 한다.</p>
과제 해결 방안 및 과정	<p>킹고와트의 과제는 성균관대학교 구성원들이 적극적으로 서비스와 양방향으로 통신함으로써 정보의 분산과 질적 부실함을 개선하고 지리적 정보와 결합된 거점 특화 서비스를 제공하는 것이다. 이를 위한 방안 및 과정은 다음과 같다.</p> <ol style="list-style-type: none"> 1. <u>위키 서비스를 이용한 정보의 질과 양 개선</u> : 성균관대학교에 대한 공식적인 정보가 아닌 성균인들의 경험에 기반한 비공식적 정보에 쉽게 접근하고 공유하기 위해 위키 서비스를 이용한다. 다른 위키에 비해 거점 특화라는 특징을 가지고 있으며 다른 서비스를 통한 높은 접근성을 통해 활성화 된다면 양질의 정보를 제공할 수 있을 것이다. 2. <u>분산되어 있는 공지 사항의 통합</u> : 현재 학교 및 학과의 공식 홈페이지, 총학생회의 페이스북 페이지, 소속 학부 대학의 페이스북 페이지 등 지나치게 다양한 경로를 통해 공지사항이 제공되고 있다. 이를 모두 통합하여 한 곳에서 보여준다면 필요한 정보를 쉽게 파악할 수 있을 것이다. 3. <u>거점에 특화된 지도 서비스 제공</u> : 기존의 일반적인 지도 서비스들은 교내 식당 등, '학교'에 특화된 상세한 건물이나 지리 정보를 제공하지 않는다. 반대로 학교 홈페이지에서 제공하는 캠퍼스 맵은 학교 자체의 정보만을 제공해 해당 거점(지역)에 대한 정보를 확인할 수 없다는 문제점이 있다. 따라서 킹고와트는 거점 기반의 지도를 통해 학교에 특화된 정보 뿐만 아니라 지역에 대한 상세한 정보도 제공한다. 4. <u>여러 단체들의 이벤트 생성</u> : 학교에 특화된 적절한 지도가 없기 때문에 학교의 여러 단체(동아리, 학생회 등)들은 이벤트를 홍보하거나 알리기 위해 여러 커뮤니티에 글을 게시하고 포스터나 줄을 직접 땅과 벽에 붙이는 비효율적 방법을 사용한다. 킹고와트의 지도 서비스를 활용한다면 이런 단

	<p>체들의 이벤트를 구역을 설정해 그곳에 추가하거나 수정함으로써 보다 정확한 위치와 정보를 알릴 수 있고, 효율적인 홍보가 가능할 것이다.</p>			
<p>★대표적 결과물 (도면, 사진 등) → 수정</p>				
<p>기업협력형 프로젝트</p>	<input type="checkbox"/> 주제 제공	<input type="checkbox"/> 멘토링	<input type="checkbox"/> 부품 제공	<input type="checkbox"/> 현장 견학
	<p>기업의 역할 체크(중복 가능) 및 관련 내용 작성</p>			

※ 요약은 3장이내 작성, 결과보고서 본문 내용은 15페이지 이상 다음페이지부터 작성

※ 과제 결과보고서 내용은 추후 우수작품집 제작시 내용이 편집되어 사용될 수 있습니다.

목 차

1. 과제 개요	5
2. 개발동기 및 목적, 필요성	6
2.1 개발 동기 및 필요성	6
2.2 목적	9
3. 과제 해결 방안 및 과정	10
3.1 위키 서비스를 이용한 정보의 질과 양 개선	10
3.2 분산되어 있는 공지사항의 통합	10
3.3 거점 기반의 특화된 지도 서비스 제공	11
3.2 여러 단체들의 이벤트 생성	12
4. 출품과제의 기술	13
4.1 Xwiki	13
4.2 Tomcat	13
4.3 Spring	14
4.4 Elastic search	15
4.5 Daum Map API	16
4.6 Gulp	17
4.7 AngularJS	17
4.8 Crawler	18
5. 개념설계 및 상세설계	20
5.1 전체 시스템 구조	20
5.2 DB 구조	20
5.3 서버 구조	21
5.4 Front-end 구조	23
6. 기타	25
6.1 수행 일정	25
6.2 소요 비용	25
7. 참고문헌	26

1. 과제 개요

(과제명, 참여자, 지도교수, 역할 담당, 참여기업 등)

1.1 과제명

- 거점 기반 종합 정보 서비스 : 킹고와트

1.2 참여자 및 역할 담당

홍기원 : 팀장, 클라이언트 개발(로그인, 쿼패널)

김동민 : 클라이언트 개발(총괄, 공지 게시판, 회원가입, 회원 정보 수정, 검색)

김준영 : 클라이언트 개발(지도 전반-카테고리, 구역 생성 등)

이현빈 : 클라이언트 개발(회원가입)

하인수 : 서버 개발(서버, 위키)

황인규 : 서버 개발(서버, 데이터 크롤링, 지도, 검색)

이수영 : 클라이언트 개발(지도 일부-구역 다이얼로그, 위키 페이지), PPT 템플릿 제작

1.3 지도교수

이진규 교수님

2. 개발동기 및 목적, 필요성

2.1 개발 동기 및 필요성

A. 디지털 큐레이션 개념의 등장

정보의 가치가 산업사회에서의 물질 및 에너지만큼 중요해지고 엄청난 양의 정보가 실시간으로 생산되어 빠르게 이동하고 소비되는 사회를 정보사회라 한다.([그림 1] 참고) 이에 따라 정보를 누가 가지고 있는가 보다 누구나 접할 수 있는 정보들 사이에서 유의미한 정보를 찾아내고 활용하는 능력이 요구되고 있다.



[그림 1 : Big data image]

그러나 정보 능력이 부족하거나 정보를 얻을 수 있는 매체(사람, 커뮤니티)에 접근하지 못하는 사람들은 자신에게 필요한 정보를 얻어내기 어려우며 정보 능력이 뛰어나더라도 오늘날과 같이 방대한 정보들 속에서 정보 수집을 위한 시간이 부족한 경우가 발생한다.

위와 같은 문제를 어느 정도 해결하기 위한 대안으로 큐레이션 서비스가 많이 생기기 시작했고 이를 '디지털 큐레이션'이라 한다. 디지털 큐레이션은 인터넷에 널린 정보 등을 주제별로, 혹은 관련된 연계성, 연관성을 지닌 무엇인가를 모아서 정돈하고 정리해서 스스로나 다른 사람에게 보여주고 공유하는 작업이다. 최근에는 핀터레스트라는 소셜 큐레이션 서비스가 미국에서 각광을 받고 전 세계로 확산되고 있는데, 이를 성균관대학교라는 거점에 적용하였을 때의 이점을 다음의 내용처럼 찾아볼 수 있었다.([그림 2] 참고)



[그림 2 : Flipboard 서비스화면]

B. '성균관대학교'에 대한 정보의 분산과 질적 부실함 개선

'성균관대학교'에 대한 간략한 정보는 대학교의 공식 홈페이지(www.skku.edu)를 통하여 얻을 수 있다. 그러나 '성균관대학교'와 연관되어 있는 깊이 있는 다른 정보들은 흩어져 있다. 예를 들어 연구실 위치나 정보는 각 과별 웹 페이지를 통해 교수님 소개 페이지에 들어가서 각 교수님의 연구실 정보를 클릭해야 얻을 수 있으며, 대학교 주변의 음식점 정보를 얻고자 하는 경우 맛집 어플리케이션을 이용하거나 타 사이트의 커뮤니티를 가입하여 추천, 리뷰 글들을 찾아 읽어야 한다. 간식 배부 정보는 과별 페이스북을 통하여 알림 받아야 하며 학과별 소식, 성대 전체 공지 등등 기본적인 정보를 얻기 위해 다양한 정보수집 경로를 이용하여 내가 필요한 정보가 있는지 없는지 주시해야 하는 불편함이 존재한다. 몇몇의 경로의 경우 커뮤니티 가입을 요구하는데 이는 커뮤니티에 속하지 못한 사람들에게 정보수집의 어려움을 겪게 한다.

또한 기존의 지도 서비스들([그림 3] 참고)이 제공하는 방대한 지도를 통해서도 킹고와 트가 전하고자 하는 상세한 건물-지리-관련 정보들을 제공하기 어렵다.



[그림 3 : 네이버 지도 '성균관대학교' 결과 화면]

반대로 대학교 홈페이지에서 제공하는 캠퍼스 맵은 학교 부지를 제외한 모든 외부 건물, 시설들은 일절 제외되며, 캠퍼스 내의 건물 상세정보 또한 극히 적은 편이다.([그림 4] 참고)



*자료 : 성균관대학교 홈페이지

[그림 4 : 성균관대학교 캠퍼스맵]

이러한 문제점을 해결하기 위해 디지털 큐레이션 서비스와 같이 ‘성균관대학교’와 연관성을 가진 정보들을 수집하고 통합하여 지도를 통한 지리정보를 제공해주는 서비스가 필요하다고 판단했다. 제공하고자 하는 정보와 연관 있는 지리정보를 더욱 효과적으로 제공하기 위해 지도와 문서가 결합된 UI를 통해 서비스를 제공하고자 한다.

C. 지리적 정보와 생활의 밀접한 관련성

‘킹고와트’는 지도를 중심으로 여러 가지 서비스가 밀접하게 연관되어 제공된다. 이는 실제 성균관대학교라는 거점에서 생활하는 동안 필요한 정보가 대부분 장소와 관련이 있기 때문이다. 예를 들어 시험 기간 간식 배부나 채용 설명회 등 학교의 여러 이벤트의 경우 학교 내의 공간에서 이뤄지고 있음을 확인할 수 있다. 뿐만 아니라 식사나 수업 등 지속적이고 반복적인 일들의 경우도 반드시 관련된 장소가 존재하므로 해당하는 위치에 대한 정보를 공유한다면 보다 효율적인 성균관대학교 생활을 하는 데 있어 도움이 될 것이다.

D. 실제 ‘성균관대학교’ 구성원이 직접 작성

거점에 관련된 다양한 공식 정보만을 모아 제공하는 경우에는 정보 자체의 한계점이 존재한다. 오로지 공식적이며 일방적이고 단방향의 특성을 가진 정보만 얻을 수 있고 사람들의 의견이나 공감의 형성될 수 있는 정보들은 얻을 수 없다는 점이다. 또한 인터넷 정보의 특성상 정보의 양은 계속 늘어나고 수시로 내용이 바뀌기 때문에 관련 정보를 서비스 관리자가 수정

해야 한다는 부담 역시 존재한다. 이를 위해 '킹고와트'는 지금까지 서술한 서비스를 제공하는 것 외에도 해당 정보가 담긴 문서에 사용자가 직접 참여하여 정보를 추가 생성하고 수정할 수 있도록 허용한다.

이와 비슷한 서비스로는 위키피디아가 존재한다.([그림 5] 참고) 위키피디아는 리처드 스톨먼이 오픈 소스인 웹 기반으로 제시한 온라인 백과사전으로 이를 통해 누구나 백과사전에 글을 수정하고 생성하여 기여할 수 있다. 글을 작성하는 경우 사전의 일관성을 위해 그 내용의 근거가 검증 가능해야 하며 문서의 이름을 변경하거나 신문기사, 블로그를 인용하는 방법 등의 규칙을 지켜야 한다는 조건이 필요하다.



[그림 5 : 위키피디아 메인 화면]

'킹고와트'의 경우 위키피디아보다 자유로운 작성을 추구하며 성균인들의 포장되지 않는 정보를 공유하는 데에 중점을 둔 서비스이다. 성균관대학교의 구성원임을 인증 받은 사용자들의 참여가 늘어날수록 정보의 질과 양 모두 빠른 속도로 성장할 것이다.

2.2 목적

'성균관대학교'를 기준으로 그와 관련 있는 다양한 정보들을 지도 서비스와 함께 제공하며 사용자들이 정보수집 시간을 단축할 수 있도록 한다. 동시에 사용자들이 직접 정보 생성/수정에 참여가 가능하도록 하여 도움이 될 만한 지식들을 자유롭게 공유할 수 있게 도와준다. 학생들에게는 추가적으로 위치정보를 활용한 서비스도 제공한다. 따라서 최종적으로 공식적 정보인 성균관대학교의 종합 공지사항 제공, 성균인들의 주관적 정보인 위키, 그리고 위치 기반의 정보 제공 서비스를 제공하는 것을 이번 프로젝트의 목표로 한다.

3. 과제 해결 방안 및 과정

본 과제는 위에서 설명된 필요성을 충족하고 그에 관련된 여러 문제점들을 해결하여 사용자들에게 편의성을 제공하는 데에 중점을 두고 있다. 따라서 본 과제의 결과물인 킹고와트는 분산된 정보를 모아서 제공하고, 질적, 양적으로 충분한 정보를 제공하며, 성균관대학교 구성원이 서비스와 양방향으로 소통할 수 있도록 하고, 지리적 요건을 고려한 정보와 학교에 특화된 정보를 제공하여 과제를 해결해야 한다.

따라서 킹고와트는 과제를 해결하기 위한 여러 기능 (방안)들을 제공한다. 우선 질적, 양적으로 충분한 정보를 제공하고 성균관대학교 구성원이 서비스와 양방향으로 소통할 수 있도록 하기 위해 위키 기능을 제공한다. 또한 정보를 모아서 제공하기 위해 분산된 공지사항을 통합하여 보여주는 기능을 제공한다. 그리고 지리적 요건과 학교라는 특징을 고려한 정보를 제공하기 위해 거점 기반의 지도 서비스 기능과 이벤트 생성 기능을 제공한다. 기능들에 대한 자세한 설명과 과정은 아래와 같다.

3.1 위키 서비스를 이용한 정보의 질과 양 개선

성균관대학교에 대한 공식적인 정보는 학과 홈페이지나 학생회 페이스북 페이지 등을 통해 얻을 수 있다. 그러나 성균관대학교에 대한 비공식적인 정보를 찾기란 쉽지 않다. 여기서 비공식적인 정보란 학교 음식점 정보나 학교 생활정보와 같은 학교나 학내 단체에서 직접적으로 다루지 않는 정보들을 의미한다. 이러한 비공식적인 정보를 얻기 위해서는 성대사랑, 에브리타임의 대학 게시판, 디시인사이드 성균관대학교 갤러리 등의 커뮤니티 사이트에서 찾아야 한다.

하지만 커뮤니티 사이트들은 기본적으로 정보가 매우 분산되어있을 뿐만 아니라 해당 정보들은 검증이 안되기 때문에 신뢰도도 떨어지는 편이다. 따라서 정보의 집약도를 높이면서 동시에 신뢰성을 높이기 위해서는 위키의 형태로 정보를 관리하는 것이 좋다. 각 문서에는 그 제목과 관련된 내용이 모여 있고 그것의 편집권한이 기본적으로 모든 사람에게 열려있기 때문에 집약도를 높이면서 학교 구성원들의 피드백을 통해 신뢰성 또한 높일 수 있다.

지금 성균관대학교와 관련된 정보가 있는 위키로는 위키피디아, 나무위키, 성대위키가 있다. 위키피디아와 나무위키는 다른 위키에 비해 활성화가 잘 되어있지만 성균관대학교만을 대상으로 한 위키가 아니기 때문에 성균관대학교의 세부적인 정보를 문서로 작성하기엔 공공성이 떨어져 해당 위키의 편집 지침상 등재하기가 어렵다는 단점이 있다. 성대위키의 경우 성균관대학교만을 대상으로 하는 위키이지만 정보가 아직 빈약하고 활성화가 안 되어 있어 접근성도 떨어진다.

따라서 우리는 지도와 통합된 학교 공지사항 정보를 제공함으로써 사용자를 끌어들이고 이를 통해 위키를 활성화 시켜서 양질의 정보를 제공하는 방안을 생각하였다.

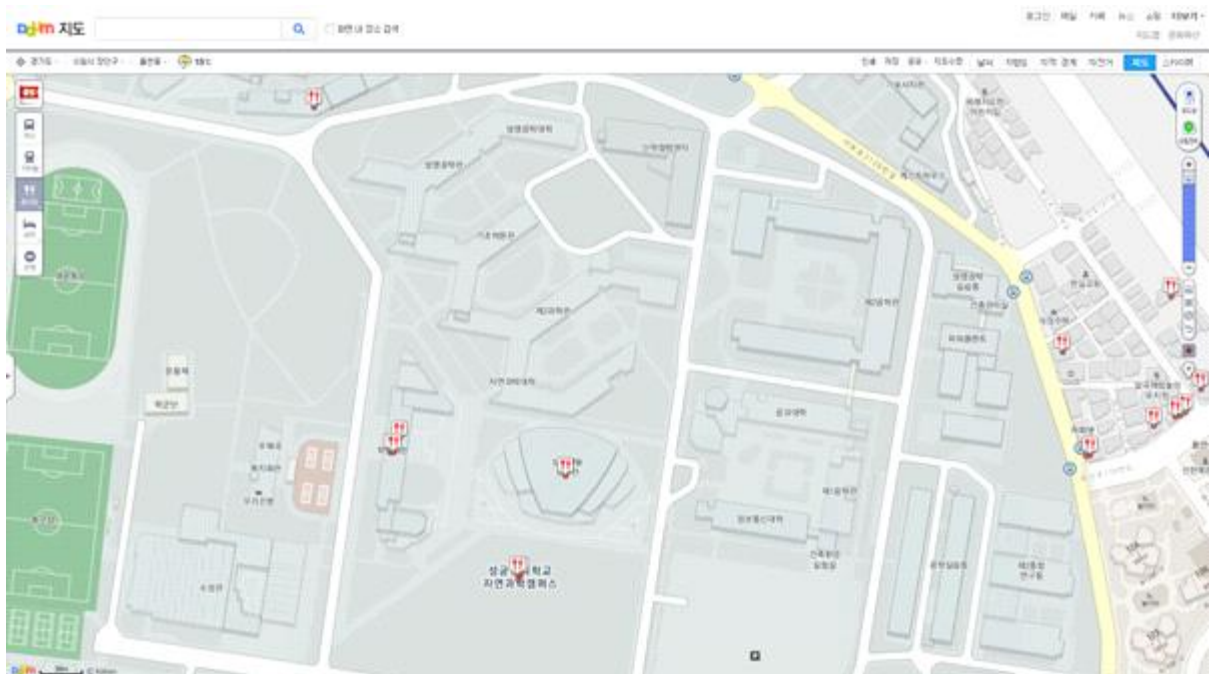
3.2 분산되어있는 공지사항의 통합

현재 성균관대학교의 학생들이 공지사항을 보기 위해서는 정말 다양한 정보수집 경로를 확인해야 한다. 일단 학교 및 학과의 공식 홈페이지 이 두 페이지에서 학교 행사나 취업, 세미나 관련 공지들을 얻을 수 있다. 또 학생회의 공지나 행사가 올라오는 페이스북 페이지를 확인해야 한다. 여기서도 대학 총학생회의 페이지와 소속 학부 대학의 학생회 페이지, 2개의 페이지를 확인해야 한다. 자신에게 필요한 공지사항이 언제 올라올지도 모르는 상황에서 매일 4개의 페이지를 확인해야 하는 것은 학업이나 여러가지 일들로 바쁜 학생들에게는 불가능한 일이다. 심지어 페이스북을 사용하지 않는 학생의 경우, 학생회의 공지사항을 전혀 받을 수 없다.

따라서 '킹고와트'는 학생들을 위해 분산되어있는 공지사항을 통합하여 보여주기로 하였다. 각각의 페이지에서 필요한 공지사항을 가져오고 데이터를 가공하여 사용자가 여러 페이지의 공지사항을 한눈에 파악할 수 있는 방안을 생각하였다.

3.3 거점 기반의 특화된 지도 서비스 제공

앞서 언급하였듯이, 기존의 일반적인 지도 서비스들은 '학교'에 특화된 상세한 건물-지리 관련 정보를 제공하기 어렵다. 그 예로, 이러한 지도 서비스들에 음식점을 검색해 보아도 제 2공대의 '공대 식당'이나 기숙사의 식당들은 검색되지 않는다. ([그림 6] 참고)



[그림 6 : 다음지도 음식점 검색 결과]

이에 반해, 현재 성균관대학교에서 제공하고 있는 지도는 '학교'에 특화된 정보만을 보여줄 뿐, ATM과 같은 편의시설이나, 캠퍼스 근처 교외의 정보는 보여주지 않는다.

본 프로젝트에서는 성균관대학교라는 거점에 특화된 지도를 가진 새로운 서비스를 제공하여 이 문제점을 해결하고자 한다. 이 새로운 지도는 학교에 대한 정보뿐만 아니라, 그 거점에 대한 상세한 정보를 포함하므로, 거점 기반의 지도를 통해 학과나 동아리의 위치처럼 학교에 특화된 정보나, 편의시설이나 교내외의 음식점 등 지역에 대한 자세한 정보도 제공할 수 있다. 따라서 본 서비스를 이용하는 학생들은 성균관대학교와 그 근처 지역에 특화된 유용한 정보를 얻을 수 있다.

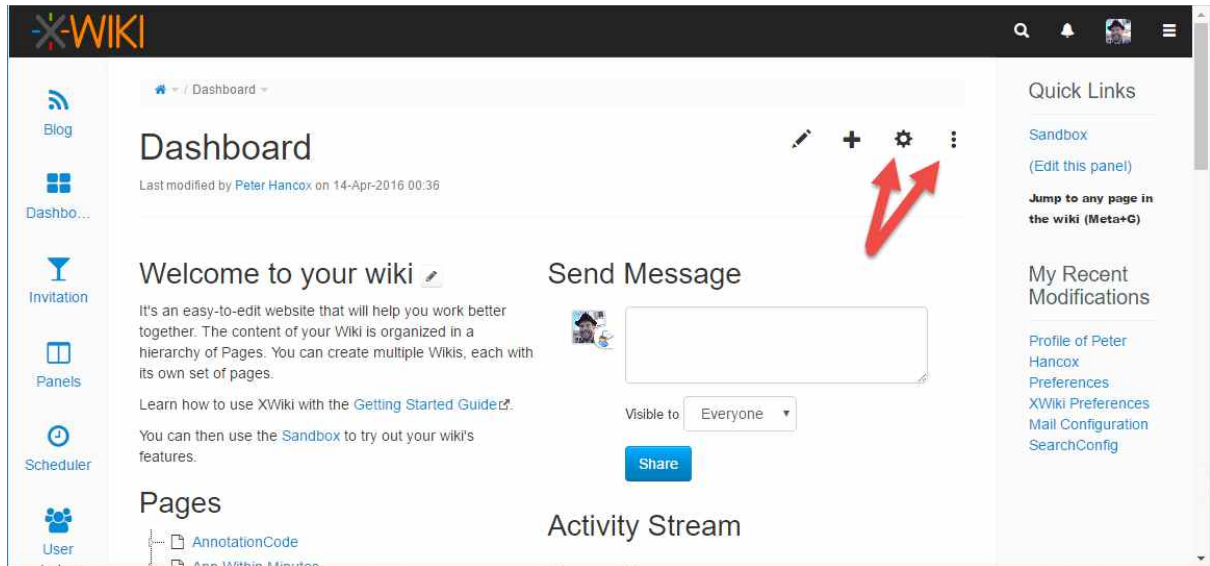
3.4 여러 단체들의 이벤트 생성

성균관대학교에 대한 정보가 분산되어 있고 학교에 특화된 적절한 지도가 없다는 것은 또 다른 문제점을 야기한다. 바로 여러 단체(동아리, 학생회)들이 이벤트를 홍보하거나 알리기가 어렵다는 것이다. 성균관대학교를 대표하는 커뮤니티가 부재하고 정보가 분산되어 있으므로, 여러 단체들은 이벤트를 각각의 커뮤니티들마다 홍보하고, 포스트를 붙이는 등 비효율적인 방법을 사용한다. 또한 이벤트가 학교의 어느 장소에 관련되어 있을 경우 그 위치를 쉽게 알리기 어려운 경우도 존재한다. 예를 들어, 여러 밴드 동아리가 주로 공연하는 학생회관 지하 1층의 '뜨락'의 경우 일반적인 지도에 위치가 표시되어 있지 않아 모르는 학생이 제대로 찾아가기 힘들다. 이 때문에 여러 동아리들이 공연, 행사 등을 할 때 바닥에 줄을 붙여 위치를 안내하는데, 이는 비효율적이다.

이에 본 서비스에서는, 3.3에서 언급하였던 거점 기반의 지도 서비스를 활용하여 여러 단체들이 이벤트를 알릴 수 있는 기능을 지원함으로써 이 문제를 해결하고자 한다. 킹고와트 서비스는 사용자가 지도에 마커와 영역을 포함하는 이벤트 구역을 생성할 수 있게 해주고, 이를 다른 사용자들이 볼 수 있게 한다. 따라서 동아리나 학생회와 같은 여러 단체들이 이 기능을 사용해 이벤트 구역을 생성할 수 있게 되고, 이는 지도 서비스를 사용하는 학생들에게 이벤트를 홍보함과 동시에 찾아오는 위치까지 안내할 수 있게 한다.

4. 출품과제의 기술

4.1 Xwiki



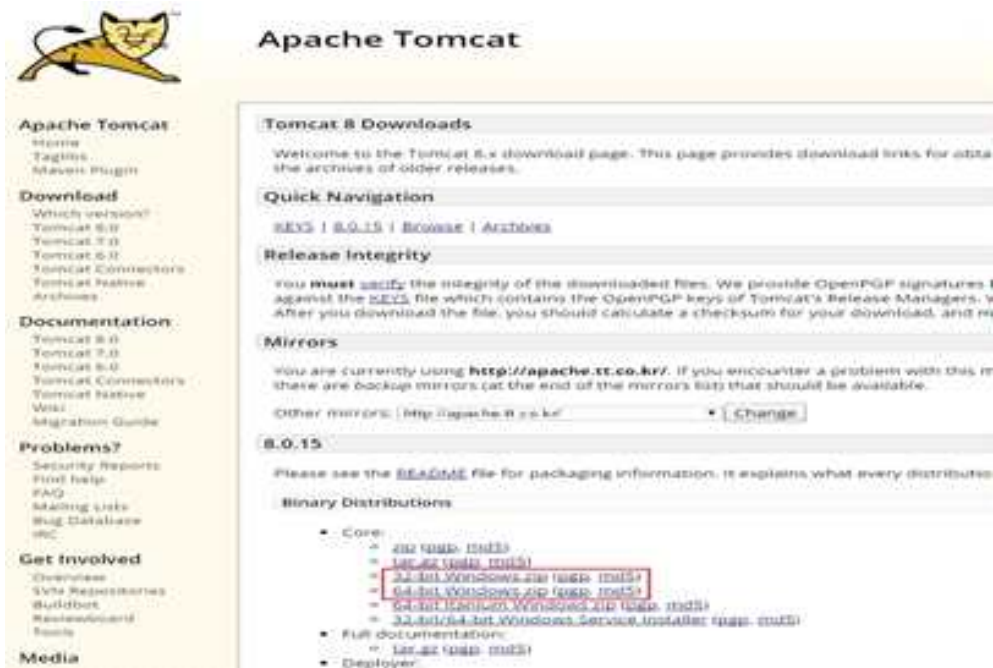
*자료 출처: www.xwiki.org

[그림 7 : Xwiki]

위키는 불특정 다수가 협업을 통해 직접 내용과 구조를 수정할 수 있는 웹사이트를 말한다. 일반적인 위키에서 텍스트는 단순화된 마크업 언어(위키 마크업)를 이용하여 작성되며, 리치 텍스트 에디터의 도움을 받아 편집하기도 한다. 위키는 지식경영이나 기록 등 다양한 용도로 이용되거나 공동체 용 웹사이트나 조직 내 인트라넷에 쓰이기도 한다. 또한 개인적인 용도로 이용되는 위키도 있는데, 이를 개인 위키라고 한다. 최초의 위키 소프트웨어인 위키 웹을 만든 워드 커닝엄은 위키를 "동작하는 가장 단순한 온라인 데이터베이스"라고 설명한다.

1세대 위키는 콘텐츠 공동 작업에 사용된다. 2세대 위키(Structured and Applications Wiki)는 위키 패러다임 및 편집 위키 페이지를 통해 공동 웹 응용 프로그램을 만들 수 있다. 웹용 툴킷인 XWiki는 1세대 Wiki 또는 2세대 Wiki로 사용할 수 있는데, 킹고와트에서는 성균관대학교 구성원 다수가 공동으로 편집하는 위키를 구현하기 위해 사용되었다.([그림 7] 참고)

4.2 Tomcat



*자료 출처: <https://www.tecmint.com/install-apache-tomcat-in-centos>

[그림 8 : Apache Tomcat의 메인 화면]

아파치 톰캣은 아파치 소프트웨어 재단에서 개발된 서블릿 컨테이너(또는 웹 컨테이너)만 있는 웹 애플리케이션 서버이다.([그림 8] 참고) 톰캣은 웹 서버와 연동하여 실행할 수 있는 자바 환경을 제공하여 자바 서버 페이지와 자바 서블릿이 실행할 수 있는 환경을 제공하고 있다. 톰캣은 관리 툴을 통해 설정을 변경할 수 있지만, XML 파일을 편집하여 설정할 수도 있다. 그리고 톰캣은 HTTP 서버도 자체 내장하기도 한다.

아파치 톰캣은 Apache Licence, Version 2를 채용한 오픈소스 소프트웨어로서, 서블릿이나 JSP를 실행하기 위한 서블릿 컨테이너를 제공하며, 상용 웹 애플리케이션 서버에서도 서블릿 컨테이너로 사용하는 경우가 많다. 버전 5.5 이후는 기본적으로 Java SE 5.0 이후를 대응한다.

아파치 톰캣에 내장된 웹 서버로만 웹 시스템을 구성할 수 있지만, 대규모의 사용자가 사용하는 시스템을 구축하려면 웹 서버와 연동하는 안정적인 시스템을 구축해야 한다. 이때, 웹 서버인 아파치 웹 서버와는 연동모듈을 사용하여 연동하고, 연동모듈로는 버전 1.3, 2.0은 mod_jk를 이용하고, 버전 2.2 이후는 mod_proxy_ajp 모듈을 사용한다.

4.3 Spring

Spring

Java Application Framework

*자료 출처: <http://javabeat.net/spring-framework-interview-questions/>

[그림 9 : Spring]

스프링 프레임워크(Spring Framework)는 자바 플랫폼을 위한 오픈소스 애플리케이션 프레임워크로서 간단히 스프링(Spring)이라고도 불린다. 동적인 웹 사이트를 개발하기 위한 여러 가지 서비스를 제공하고 있다. 대한민국 공공기관의 웹 서비스 개발 시 사용을 권장하고 있는 전자정부 표준 프레임워크의 기반 기술로서 쓰이고 있다. 스프링은 다른 프레임워크에 비해 다음과 같은 특징을 가진다.

우선 경량 컨테이너로서 자바 객체를 직접 관리한다. 각각의 객체 생성, 소멸과 같은 라이프 사이클을 관리하며 스프링으로부터 필요한 객체를 얻어올 수 있다. 일반적인 J2EE 프레임워크에 비해 구현을 위해 특정한 인터페이스를 구현하거나 상속을 받을 필요가 없어 기존에 존재하는 라이브러리 등을 지원하기에 용이하고 객체가 가볍다. 또한 스프링은 제어 반전을 지원한다. 컨트롤의 제어권이 사용자가 아니라 프레임워크에 있어서 필요에 따라 스프링에서 사용자의 코드를 호출한다. 스프링은 의존성 주입을 지원한다. 각각의 계층이나 서비스들 간에 의존성이 존재할 경우 프레임워크가 서로 연결시켜준다. 스프링은 관점 지향 프로그래밍을 지원한다. 따라서 트랜잭션이나 로깅, 보안과 같이 여러 모듈에서 공통적으로 사용하는 기능의 경우 해당 기능을 분리하여 관리할 수 있다. 스프링은 연속성과 관련된 다양한 서비스를 지원한다. iBATIS나 Hibernate 등 이미 완성도가 높은 데이터베이스 처리 라이브러리와 연결할 수 있는 인터페이스를 제공한다. 스프링은 확장성이 높다.

4.4 Elastic search



elasticsearch.

*자료 출처: <https://www.sitepoint.com/introduction-to-elasticsearch-in-php/>

[그림 10 : Elastic search]

Elasticsearch는 Apache Lucene 기반으로 개발된 오픈소스 분산 검색 엔진 서버이다. Elasticsearch, Logstash, Kibana 일명 'ELK스택'을 이루는 핵심 기술이다. JSON 기반의 비정형 데이터 분산 검색과 분석을 지원한다. 추가적인 다양한 기능들은 플러그인 형태로 설치해서 사용할 수 있도록 지원한다. 설치와 확장이 편리하다는 장점과 실시간 검색 서비스 지원, 분산 및 병렬 처리 그리고 멀티테넌시 기능을 제공한다. 또한 분산 시스템이기 때문에 검색 대상의 용량이 증가했을 때 대응하기가 수월하다는 장점이 있다.

Elasticsearch는 모든 데이터를 놀랍게 빠른 속도로 액세스하여 활용할 수 있다. 프로토타입에서 운영 배포까지의 과정이 수월하다. 원활한 작업을 위해 Cluster에서 인덱스와 쿼리 배포 방법을 자동으로 관리하면서 초당 엄청난 양의 이벤트를 처리할 수 있도록 수평적 확장이 가능하다. 표준 RESTful API를 지원하고 JSON을 사용하기 때문에 클라이언트 입장에서도 작업하기 쉽고, 사용하기 자연스럽다.

Elasticsearch는 기본적으로 영어 지원을 목적으로 만들어졌기 때문에 '킹고와트'에서는 '은전한닢 프로젝트'의 elasticserch-analysis-seunjeon 플러그인을 설치하여 사용하였다. '은전한닢 프로젝트'란 검색에서 사용 가능한 오픈소스 한국어 형태소 분석기를 개발하는 프로젝트로 '킹고와트'의 크롤러(4.8에서 설명)에서 사용한 KoNLPy의 mecab, elasticsearch-analysis-seunjeon을 개발하였다. 이 플러그인을 이용해 한국어에 맞는 색인을 생성하고 검색에 이용하였다.

Elasticsearch는 admin 계정이나 패스워드 설정 등 보안 관련해서 기본적으로 제공하고 있지 않기 때문에 따로 플러그인을 설치해야 한다. Cluster에 사용자 이름과 암호 설정을 원하면 Security를 설치하면 된다. 또 자신의 검색 서버의 성능을 알고 싶으면 Monitoring을 추가하면 된다. 이 모두가 Elastic에서 만들어놓은 플러그인인 X-Pack을 통해 제공된다. X-Pack은 보안, 모니터링, 알림, 보고 및 그래프 탐색 기능으로 완벽한 Elasticsearch 환경을 제공한다. 추가적으로 대규모 데이터를 Hadoop에서 관리 하려면 Elasticsearch-Hadoop (ES-Hadoop) 커넥터를 사용하여, Elasticsearch의 실시간 검색 및 분석 기능을 사용할 수 있다.

4.5 Daum Map API



[그림 11 : 다음 지도]

킹고와트의 기반이 되는 지도 서비스는 다음의 지도 API를 사용하였다. 다음 지도 API는 무료로 제공되는 지도 Javascript API로서 마커 추가, 컨트롤 설정, 커스텀 오버레이 생성 등의 이벤트를 추가하기 간편하고 Drawing 라이브러리 등 다양한 라이브러리를 제공해줌으로써 확장성이 좋다는 장점이 있다. 킹고와트에서 이용한 다음 지도 API의 기능으로는 마커 클러스터링, 마커 이벤트, Drawing 라이브러리를 이용한 구역 추가, 커스텀 타일셋, 다각형 생성 및 이벤트 등록 등이 있다.

4.6 Gulp



*자료 출처 : <https://velopert.com/1344>

[그림 12 : gulp.js]

gulp.js는 프론트엔드 웹 개발의 빌드 시스템을 자동화 해주는 오픈 소스 자바스크립트 툴과 그 툴의 github 커뮤니티를 뜻한다. 즉, 웹 개발에서 가장 시간이 많이 소요되고 반복적인 작업인 minify, 모듈 삽입, 유닛 테스트, 최적화 등을 task로 나누어 실행하는 Node.js 기반의 task runner이다. gulp는 javascript 모듈인 node stream을 기반으로한 빌드 툴이며 각 stream은 파이프라이닝을 통해 각종 파일 operation을 연결한다. 파일 operation은 gulp.js를 기반으로 하고 있는 플러그인 생태계를 import하여 작동시킨다. 이렇게 gulp는 웹 개발 모듈이 올라와 있는 파일 시스템을 읽고 .pipe() operator를 통해 그 데이터를 플러그인을 통해 가공하고, 다시 다음 플러그인으로 넘긴다. 각각의 플러그인은 task로 정의된다. 최종적으로 가공된 데이터는 destination을 특정하여 원래의 모듈에 삽입할 수도 있고 별개의 모듈에 통합시킬 수도 있다.

gulp는 이러한 유연성을 통해 사용자가 원하는 대로 configuration할 수 있다는 장점이 있지만, 그만큼 javascript와 각 모듈, 플러그인에 대한 지식이 필요하다. 또한 task 단위로 operation을 분리함으로써 각 operation에 대한 엄밀한 객체화가 가능하고 세분화하여 시스템 빌드 작업을 관리할 수 있다.

4.7 AngularJS



*자료 출처: <https://angularjs.org/>

[그림 13 : AngularJS]

AngularJS(AngularJS 1.x)는 싱글 페이지 어플리케이션을 보다 쉽게 개발하기 위해 구글과 개인 및 기업체의 커뮤니티에 의해 개발된 자바스크립트 기반 프론트 엔드 오픈 소스 웹 어플리케이션 프레임워크이다. AngularJS의 주된 목표는 MVC (model-view-controller), MVMM (model-view-viewmodel) 개발 구조 패턴과 웹 어플리케이션에서 주로 사용되는 모듈들을 클라이언트 사이드에 지원하여, 어플리케이션의 개발과 테스트 과정을 단순하게 하는 것이다.

AngularJS 프레임워크는 사용자가 스스로 만든 커스텀 태그 속성이 내장된 HTML 페이지를 읽는 방식으로 동작한다. AngularJS는 이 태그 속성들을 directive로 해석하여 페이지의 입출력값을 자바스크립트 변수의 일종인 model과 binding한다. 이 자바스크립트 변수 값들은 코드 내에서 미리 설정할 수도 있고, 정적, 혹은 동적으로 JSON 리소스를 받아 입력할 수도 있다. 이처럼 AngularJS 프레임워크는 전통적인 HTML을 인정하고 확장하여 two-way data binding을 통한 동적인 콘텐츠를 구현하는데, 이러한 방식은 자동으로 모델과 뷰의 동기화를 가능하게 한다. 결과적으로 AngularJS는 명시적으로 DOM 제어하는 것을 줄였으며, 테스트 용이성과 성능을 향상시켰다.

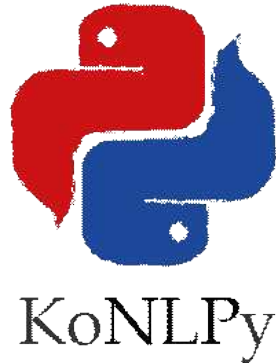
AngularJS는 선언형 프로그래밍으로 유저 인터페이스를 생성하고 소프트웨어 모듈을 연결해야 한다는 신념에 기반하며, 디자인 목적은 다음과 같다. 첫째로, 어플리케이션의 로직으로부터 DOM 제어를 분리하여 코드의 구성을 훨씬 쉽게 한다. 둘째, 서버 사이드와 클라이언트 사이드를 분리하여 동시 개발과 코드 재사용을 용이하게 한다. 마지막으로, UI 디자인에서부터 로직을 구현하고 테스트하기까지의 구조를 제공한다.

4.8 Crawler

곳곳에 흩어져 있는 학교의 공지사항들을 가져와 한눈에 보여주기 위해 '킹고와트'는 python을 써서 크롤러를 만들었다. 크롤링이란 웹 페이지를 가져와서 거기서 필요한 데이터를 뽑아내는 것을 말한다. '킹고와트'는 성균관대학교 홈페이지(<http://www.skku.edu>)와 소프트웨어대학 홈페이지(<http://cs.skku.edu/>)에서 공지사항들을 가져왔고 페이스북 성균관대 총학생회 페이지(<https://www.facebook.com/skkunssc/>)와 소프트웨어대학 학생회 페이지(<https://www.facebook.com/skkusoft/>)에서 게시물들을 가져왔다. 크롤러에 활용한 기술들은 다음과 같다.



[그림 14 : Celery]



[그림 15 : KoNLPy]

A. Celery

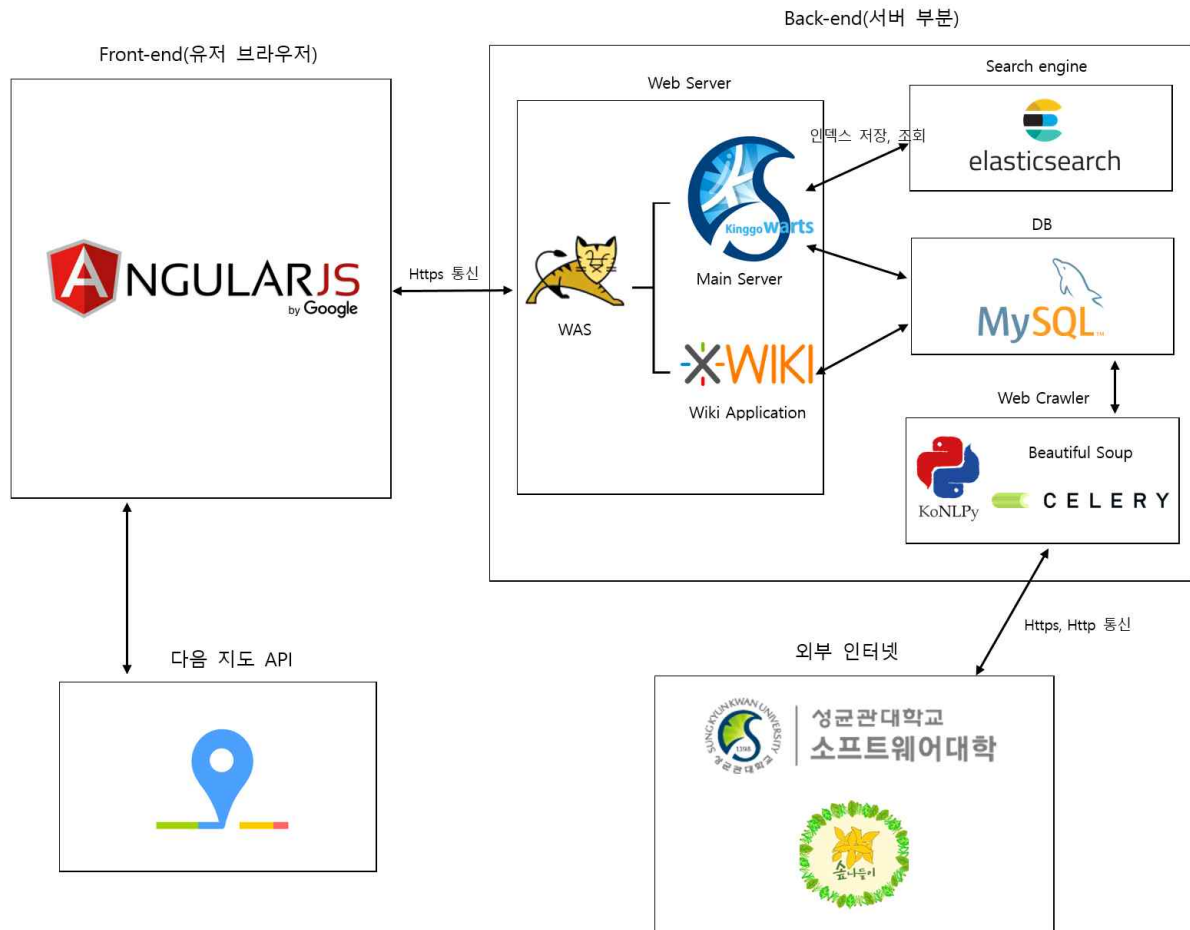
python으로 작성된 비동기 작업을 처리해주는 큐이다. 작업(task), 브로커(broker), 워커(worker)가 비동기 작업을 처리하는 방식이다. 사용자가 지정한 작업이 실행될 때, 브로커에 쌓이게 되고 하나 이상의 워커가 브로커에서 작업을 꺼내 수행하게 된다. '킹고와트'는 브로커로써 redis라는 NoSQL을 사용했다. redis란 NoSQL 중 하나로 Key-Value 저장 공간을 가진다. Latency와 throughput과 관련해 상당한 성능 이익을 내는 in memory 저장소이다. 또한 '킹고와트'의 크롤링 작업은 celery에 포함되어있는 project로 celery beat를 사용했는데 celery beat란 브로커에 주기적인 실행이 필요한 작업들을 넣어주는 역할을 한다. '킹고와트'에서는 크롤링 작업을 4시간마다 반복하도록 celery beat에 task로 등록해서 실행했다.

B. KoNLPy

한국어 정보 처리를 위한 python package이다. 한국어의 형태소를 분석해 품사에 맞게 Tagging을 해서 return해주는 package이다. 여러 태그 클래스를 지원하는데 그중 mecab을 사용했다. mecab이란 일본어 형태소 분석 엔진으로 은전한닢 프로젝트에서 한국어와 일본어의 유사점을 이용해 한국어에 맞게 만든 형태소 분석기이다. 사용자 정의 사전 추가 및 복합어 설정, 분해 차단 설정 등 여러가지 기능을 제공한다. '킹고와트'는 '성균관대학교'라는 거점에 맞는 사용자 사전을 추가해서 사용했다. 각각의 페이지에서 가져온 공지 사항들을 konlpy를 이용해 parsing을 하여 장소와 관련이 있는 단어들을 추출해 공지사항과 지도를 연결해주었다.

5. 개념설계 및 상세설계

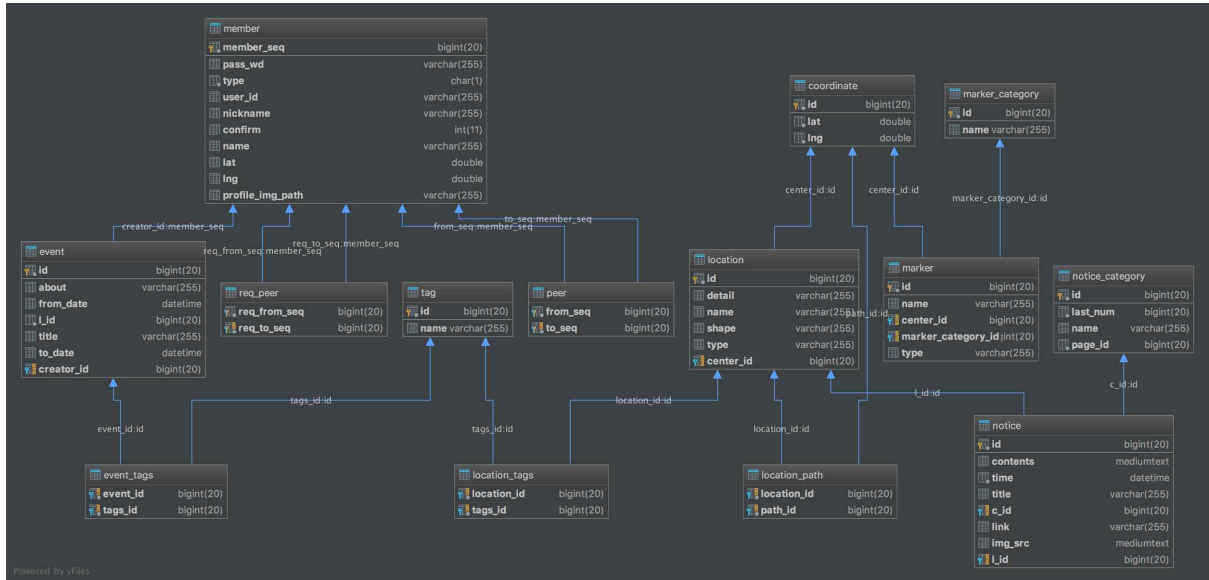
5.1 전체 시스템 구조



[그림 16 : 전체 시스템 구성도]

Kinggowarts의 시스템은 크게 Front-end와 Back-end로 구성되어 있다. Front-end는 기본적으로 AngularJS를 기반으로 이루어져 있고 지도는 다음 지도 API를 이용해서 불러온다. Back-end는 웹서버, DB, Web crawler, Search engine으로 이루어져있다. 웹서버는 Tomcat 서버를 이용하였고 서버에는 Kinggowarts 메인 서버와 Wiki application인 XWiki가 올라가 있다. 또한 DB로는 Mysql server를 사용하고 Python으로 작성한 Web crawler는 소프트웨어 대학의 공지사항과 페이스북 소프트웨어대학 학생회의 글을 크롤링해서 DB에 넣는다. 그리고 Web crawler가 생성한 데이터들은 elasticsearch에서 인덱싱이 되고 이를 통해 메인 서버에 검색기능을 제공한다.

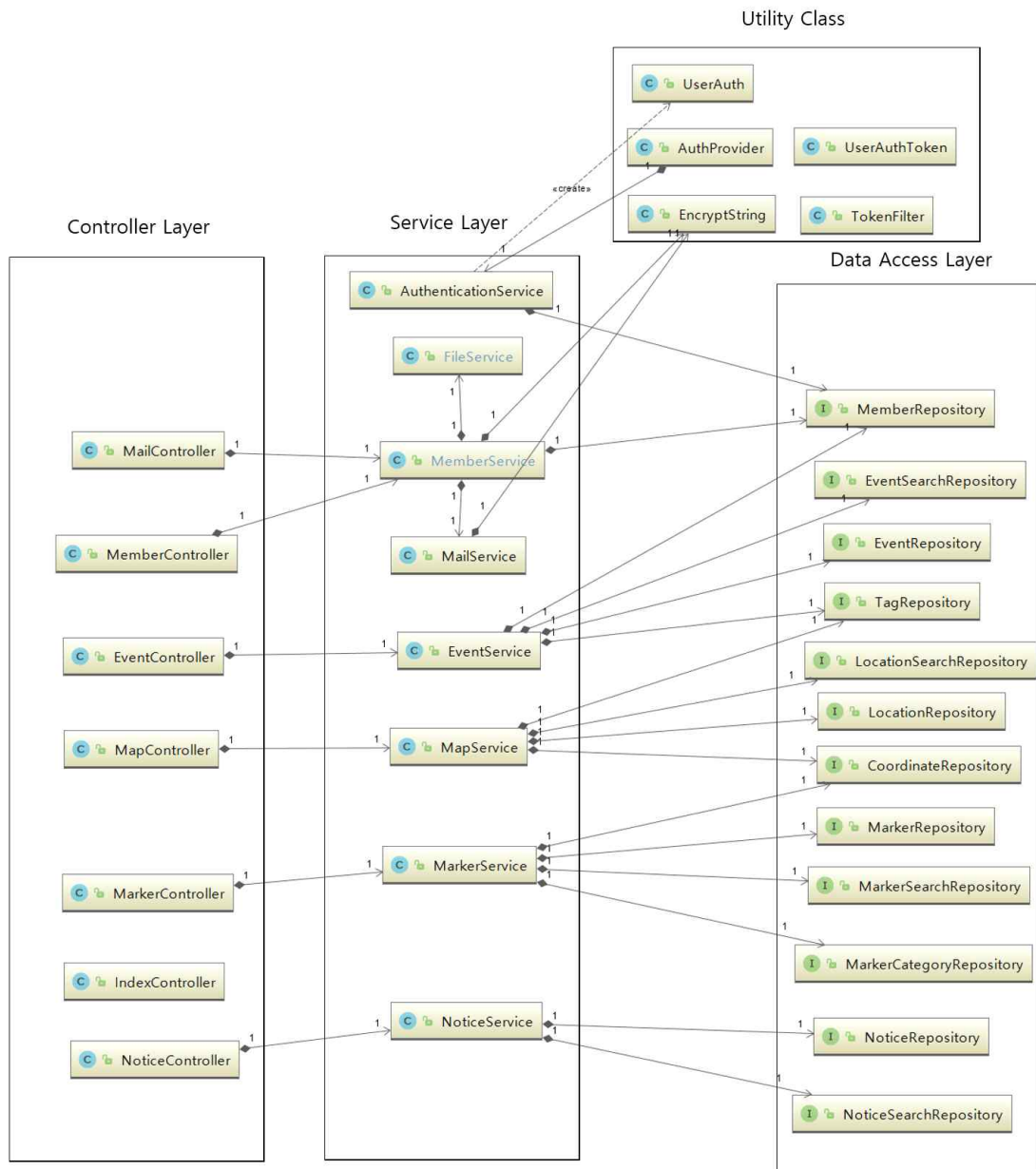
5.2 DB 구조



[그림 17 : DB Relational Schema]

'킹고와트'의 전체 DB Relational Schema는 위 그림과 같다. '킹고와트'는 spring-data-jpa를 활용해 ORM(Object Relational Mapping)을 썼다. ORM이란 RDB 테이블을 객체지향적으로 사용하기 위한 기술이다. RDB 테이블은 객체지향적 특징(상속, 다형성, 레퍼런스, 오브젝트 등)이 없고 자바와 같은 언어로 접근하기 쉽지 않다. 때문에 ORM을 사용해 오브젝트와 RDB 사이에 존재하는 개념과 접근을 객체지향적으로 다루기 위한 기술이다. 이러한 ORM을 활용해 class를 선언하여 table을 만들고 각각의 튜플을 객체로 접근하여 관리하였다. 주요 테이블로는 지도의 구역을 저장한 location, 사용자 정보를 저장한 member, 크롤링한 공지사항을 저장한 notice, 사용자가 만든 행사를 저장한 event, 지도의 자잘한 정보를 저장한 marker 등이 있다.

5.3 서버 구조



[그림 18 : Controller-Service-DAO 패턴]

웹 어플리케이션에서는 UI와 비즈니스 로직을 분리한 MVC 패턴이 자주 쓰인다. 우리는 Spring MVC 를 이용하여 이를 구현하였다. Controller는 사용자가 보낸 요청을 받고 적절한 Service를 호출하고 Service는 모델에 대한 비즈니스 로직을 담당한다. Data access layer에서는 모델의 DB CRUD를 담당한다.



[그림 19 : Model의 Class diagram]

모델은 데이터를 의미하고 ORM을 이용하였기 때문에 모델은 동시에 DB의 Entity를 뜻하게 된다. 이러한 객체와 Entity의 매핑은 Spring ORM, Hibernate를 이용해서 구현된다.

5.4 Front-end 구조

6. 기타

6.1 수행 일정

No .	수행 내용	추진 일정																비 고
		1 주	2 주	3 주	4 주	5 주	6 주	7 주	8 주	9 주	10 주	11 주	12 주	13 주	14 주	15 주	16 주	
1	주제 선정																	
2	계획수립 및 자료조사																	
3	제안서 작성 및 계획발표 준비																	
4	개발 언어 학습																	
5	웹 서버 구성																	
6	웹 어플리케이션 제작																	
7	중간발표 준비																	
8	데이터 수집 및 분석																	
9	제품 평가																	
10	결과보고서 작성 및 최종발표 준비																	

6.2 소요 비용

항목	날짜	품명	수량	단가(원)	금액 (VAT포함)
제작비	17/6/5	지도 디자인 외주	1	80,000	80,000
제작비	17/6/4	로고 외주	1	50,000	50,000
제작비	17/6/7	지도 디자인 외주 범위 확장	1	30,000	30,000
합계		160,000 원			

7. 참고문헌

- [1] 행정안전부, "스마트시대 정보문화 실태조사 개편방안 연구", 2012.9
- [2] 한국 산업인력공단, "정보 능력", 2013.1
- [3] 다음 지도 API, 'Web API 가이드', <http://apis.map.daum.net/web/guide/>
- [4] AngularJS Material, 'AngularJS Material(Documentation Version 1.1.4)',
<https://material.angularjs.org/latest/>
- [5] Fuse, 'Fuse – Admin Theme',
<http://fuse-angular-material.withinpixels.com/dashboard-project>