

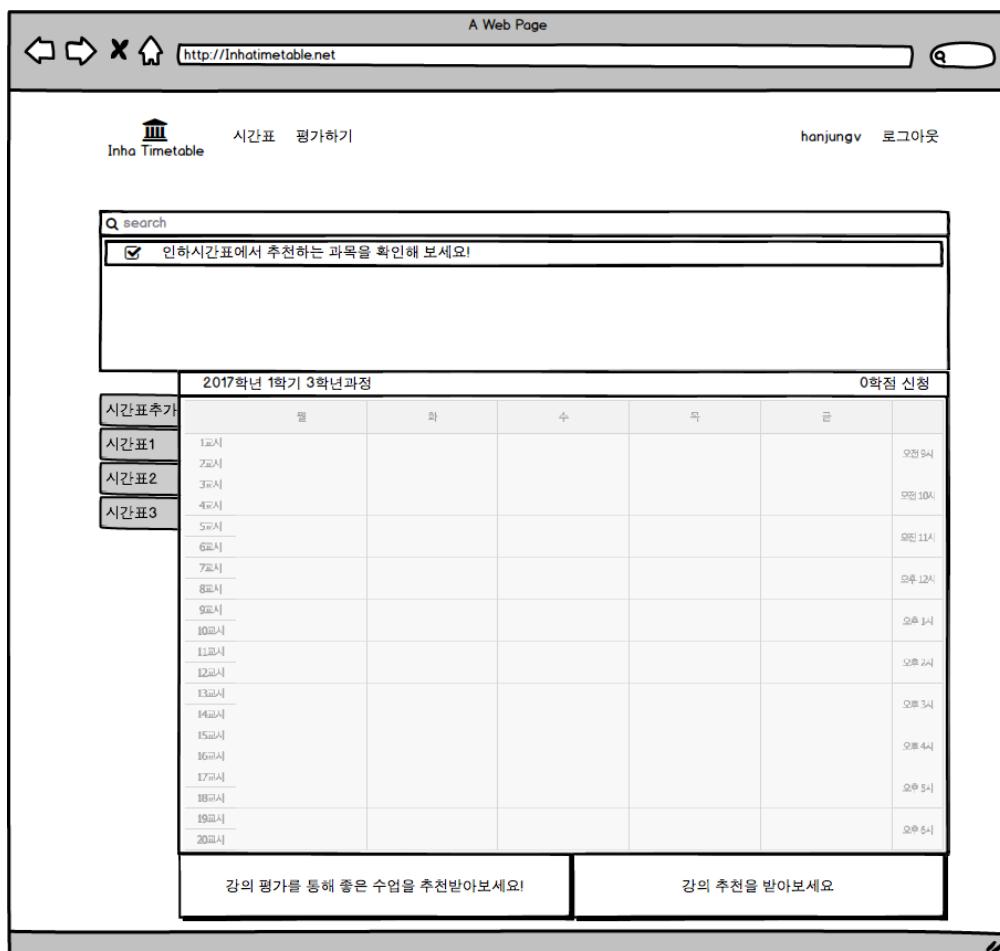
# 종합설계 제안서

캡짱스톤 조 : 한정(컴공, 12114497), 박인규(컴공, 12121484), 이강호(컴공, 12142380)

## 주제

최종목표 : 여러 사용자 환경을 만족시키는 추천 기반 시간표 작성 웹을 완성도 있게 제작한다.

- 시간표를 웹, 모바일 웹 모두에서 쉽게 작성할 수 있게 도와준다.
- 현재의 성향을 파악하여 과목을 추천 받을 수 있게된다.
- 강좌의 선택 폭이 다양해질 수 있다.
- 몇명이 이 과목에 관심이 있는지 알 수 있다.
- UI prototype

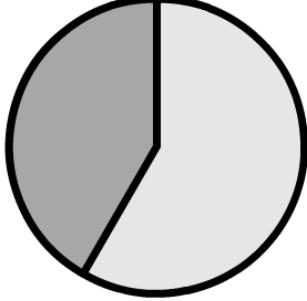


메인 화면입니다.

A Web Page  
http://Inhatimetable.net

Inha Timetable 시간표 평가하기 hanjungv 로그아웃

평가 결과



총 20과목을 수강하셨습니다!

통계 평가한 과목(20) 코멘트(12)

선호 교수님

순위	교수님 이름	평균 점수
1.	OOO 교수님 (컴퓨터공학과)	75점
2.	OOO 교수님 (컴퓨터공학과)	60점
3.	OOO 교수님 (컴퓨터공학과)	40점

개인정보 화면입니다



추천 화면입니다

## 배경

과거 20%의 주 품목들이 전체 매출의 80%를 차지하였다. 그러나 점차 개인 성향이 중시되며, 인터넷의 성장으로 정보의 접근이 쉬워지자 나머지 80%의 비주류 품목의 전체 매출이 무시 못할 규모가 되었다. 최근 IT 기업들은 개인의 성향에 중시하여 추천 알고리즘을 통해 시스템을 구축하기 시작했고, 그 결과 괄목할만한 결과를 얻어내었다. 이에 평소 알고리즘에 관심이 많았던 캡짱스톤 조는 강의 추천 시스템을 구현해 추천 알고리즘을 실제 적용을 해보고자 한다.

- 이 프로젝트 완료로부터 기대되는 효과
  - 간편한 UI / UX 제공으로 쉬운 시간표 제작
  - 강의 몰림현상을 줄임
  - 좋은 강의들의 인원 부족으로 인한 강의의 폐강을 줄임
  - 학생들의 성향 파악을 통한 추천으로 만족도를 높이고 수강 포기 확률 감소
- 관련 분야 survey
  - Stanford Jeffrey D. Ullman *mining massive datasets* 강의 수업자료
    - 기본적인 Recommend System 개념 학습
    - 신뢰도 계산법 및 적절한 선택법 학습
  - 카이스트 제 8회 ROSAEC 워크샵자료 *Recommendation System "협업 필터링을 중심으로"*
    - 기본적인 Recommendation System 개념 학습

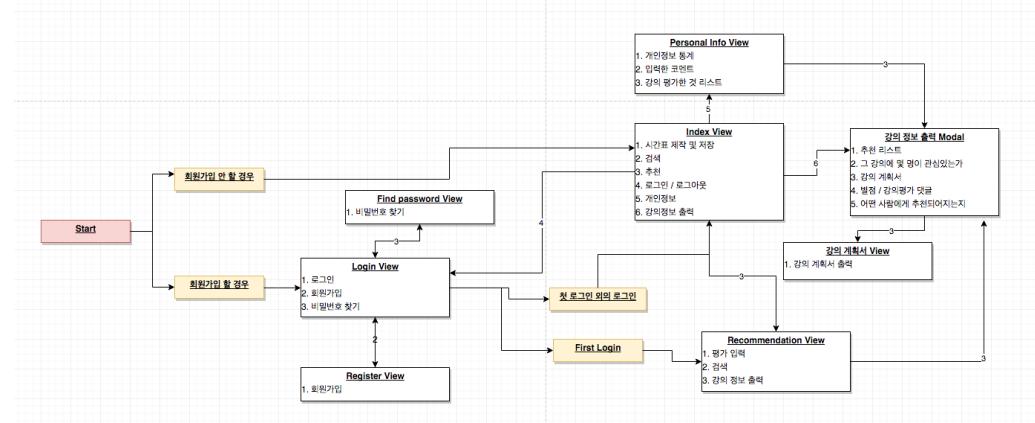
- mahout을 이용한 recommendation system Overview
  - recommendation system library 확인
- python 과 mysql linking, 기존에 가지고 있던 데이터 조작 연습
  - python으로 sql data 조작 후 원래 서버와 linking하여 저장하는 연습이 필요
- 유사 프로젝트 검색 및 해당 프로젝트 결과물들의 문제점
  - [인하대학교 수강신청, http://sugang.inha.ac.kr/sugang/](http://sugang.inha.ac.kr/sugang/) : 학교 수강신청 사이트
    - 강의 선택이 한 화면에서 이뤄지지 않음
    - 모바일에 최적화 되어있지 않음
  - [에브리타임, http://everytime.kr/](http://everytime.kr/) : 시간표 제작 및 익명 커뮤니티
    - 추천 시스템 부재
    - 실제 시간표가 수강신청 사이트에 등재 된 후 약 1주일 후 반영
  - [SNUTT, http://snutt.kr/](http://snutt.kr/) : 서울대학교 시간표 제작
    - UI / UX 디자인 부재
    - 서울대 자대에만 사용되는 시스템
    - 추천 시스템 부재
  - [주관식 강의백서,https://goo.gl/PKoyQc](https://goo.gl/PKoyQc) : 인하대학교 주관식 강의백서
    - 불편한 UI / UX
    - 어려운 검색, 강의에 대한 자료의 낮은 가독성
    - 모바일 최적화 부재
- 기존 프로젝트와 제안 내용과의 차이 및 기존 문제 해결 방안
  - 추천 시스템
  - 사용이 간편한 UI / UX
- 기술적인 어려움
  - 추천 과목(item)이 적절하게 추천이 되었는지에 대한 수량적 사용자 만족도 판단의 어려움
  - 특정 OS 기반이 아닌 브라우저 기반으로 어려운 디바이스 최적화
  - 다양한 브라우저 호환 필요
  - 특정 기간에 급격히 늘어나는 요청에 대한 대응 필요
  - 비용이 큰 잦은 연산에 의한 시스템을 분리 필요

## 개발 내용

---

- 개발 플랫폼 : **Web Application**
  1. 높은 호환성(OS에 국한되지 않음)
  2. 자바스크립트 기반의 높은 확장성
  3. 상대적으로 많은 라이브러리와 자료 존재
  4. 서버 언어로 `Node.js` 를 사용하여 클라이언트와 동시 개발 용이
- 상세 요구사항
  - 기능적 요구사항 (functional requirements)
    - 과목 recommendation systems
    - 시간표 제작

- 모바일 환경을 고려한 반응형 웹
- 기존보다 쉽고 간편한 검색
- 비기능적 요구사항 (non-functional requirements)
  - 해당 과목에 관심이 있는 인원 명시를 통한 경쟁 가능성 제시로 수강 인원 분산 유도
  - 추천 시스템 사용 유도



추천 화면입니다

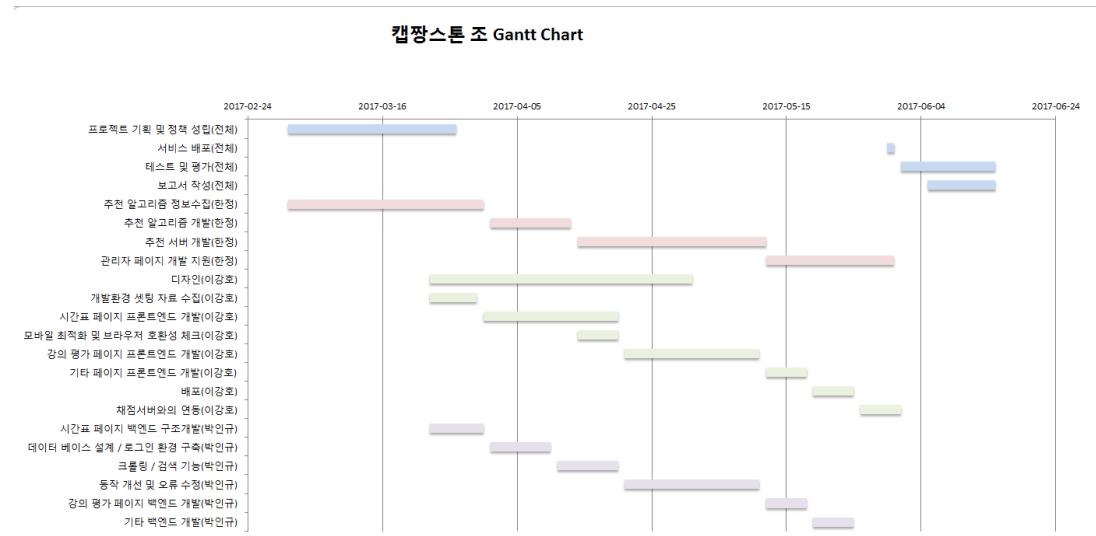
## 개발 방법

- 사용하게 될 open sources
  - React
    - facebook에서 만든 유저인터페이스 라이브러리
    - 페이스북, 인스타그램, 넥플릭스에서 널리 쓰임
    - UI를 component화 해서 손쉬운 재사용이 쉽게 만드는게 특징
    - VIRTUAL DOM이라는 개념을 사용해서 상태의 변함에 따라서 선택적으로 UI를 렌더링
  - Babel
    - ES6와 ES7 모든 웹 브라우저에서 호환 가능하도록 변환해주는 transcompiler
  - Redux
    - React로 이루어진 앱에서 효율적인 데이터 흐름을 위해 사용하는 라이브러리
  - Node.js
    - Javascript 엔진으로 만들어진 언어로써, Non-blocking I/O와 단일 스레드 이벤트 루프를 통한 높은 처리 성능이 특징인 언어
  - express
    - Node.js에서 가장 많이 쓰이는 웹 프레임워크
  - passport
    - Node.js에서 가장 많이 쓰이는 유저 인증을 위한 미들웨어
  - SASS
    - CSS를 확장하는 pre-processor

- webpack
  - 자바스크립트 모듈화를 지원해주는 모듈화 엔진
- mysql
  - 가장 널리 쓰이고 있는 오픈소스 RDBMS
- redis
  - 메모리 기반의 key/value NoSQL DBMS
- 가용 가능한 data sources
  - sugang.inha.ac.kr에서 수강가능한 시간표
  - 예전 총학생회에서 조사한 주관식 교양 백서
  - 인하대학교 정보통신처에서 학생들의 수강신청 기록들
  - '인하대학교 시간표'에서 데이터 요청
- 윤리, 정책 등 비기술적 고려사항
  - 평가가 이뤄지는 시기(6월)가 수강신청이 실제로 이뤄지지 않는 시기
    - 학생들의 지속적인 유입을 어떻게 유도할지
  - 정보 수집 범위
    - 회원가입시 정보 수집에 대한 동의
    - 개인정보 없이 직접적인 강의 평가를 받아 정보 누적
  - 학생들의 악의적으로 입력하는 데이터의 필터링 문제

## 계획 및 일정 (Plan and schedule)

- 프로젝트 규모 산정
  - 예상 소요기간
    - 기획 및 디자인 : 2017.03.02 ~ 2017.03.27
    - 개발 기간 : 2017.03.27 ~ 2017.05.31
    - 성과 측정 기간 : 2017.06.01 ~ 2017.06.15
  - 프로젝트 규모 : 두개의 서버(추천서버, 웹서버)와 8개(예정)의 뷰
- 요소 작업들에 대한 roadmap 구성
  - Gantt chart를 이용한 각 구성 작업 당 소요 기간



간트차트 입니다.

- 평가기준
  - 강의 추천의 정확도
  - 추천 시스템 연산의 속도를 어떻게 빠르게 할 것인지
  - 사용자의 웹 및 모바일 환경에서 접근 용이성
  - 적절한 UI / UX 디자인이 쓰였는가

## 참고문헌

- web part
  - 이런 논문 1
  - 이런 사이트 1
  - 이런 논문 2
  - 등등등
- recommendation part
  - Stanford Jeffrey D. Ullman *mining massive datasets* 강의 수업자료 (<http://infolab.stanford.edu/~ullman/mmds/ch9.pdf>)
  - 카이스트 제 8회 ROSAEC 워크샵자료 *Recommendation System* "협업 필터링을 중심으로" (<http://rosaec.snu.ac.kr/meet/file/20120728b.pdf>)
  - 미네소타 대학 Technical Report, "Application of Dimensionality Reduction in Recommender System -- A Case Study"
  - 웃자 추천시스템 대학교 발표 PPT (<https://www.slideshare.net/itsociety/4-151106-58384179>)
  - mahout을 이용한 recommendation system  
Overview (<https://mahout.apache.org/users/recommender/recommender-documentation.html>)
  - recommendation system 구축 엔진 선택에 관한 Quora text (<https://www.quora.com/How-can-I-start-building-a-recommendation-engine>)
  - mysql-python connector overview (<https://dev.mysql.com/doc/connector-python/en/connector-python-connectargs.html>)