



Archive

“ 3차 주간 성과 발표 ”

순 서

소 개

프론트
개발

백앤드
개발

계획

종합
보고

Archive

Archive

소개

예상 그림



예시 이미지.



디테일 구조.

창민호

Arch



사용자 문화생활
패턴 및 감성
분석

문화 또는
관심사 공유

새로운 감성의 3세대 SNS

분석 데이터를
통한
사용자 맞춤형
반응, 서비스

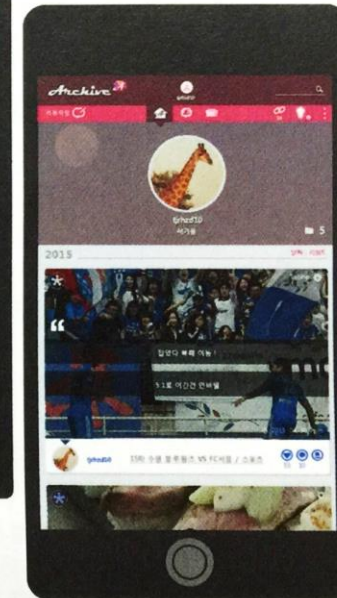
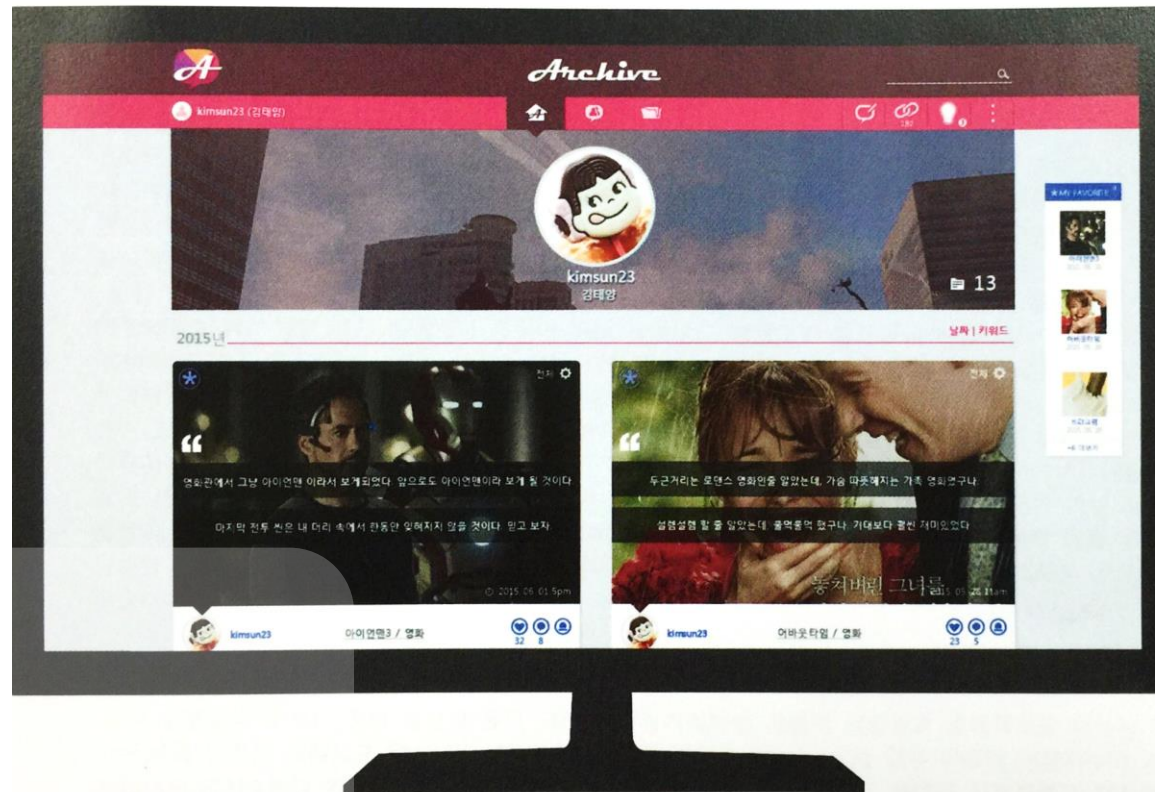
관심사를
통한
인맥 형성

Archive

예상 그림



예시 이미지.

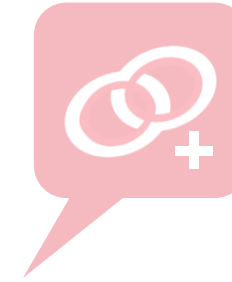


디테일 구조.

정민호

Archive

예상 그림



디테일 구조.

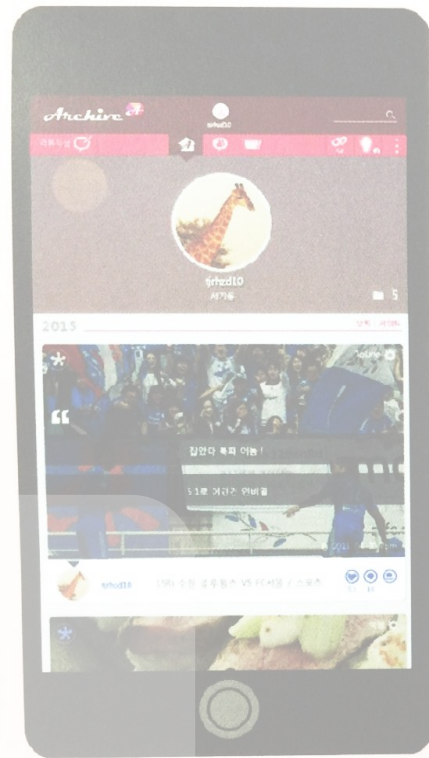


사진 한 장
+
두 줄 평.



Archive

Archive

프론트
개발

구현 페이지

33342/test1/uc/index.html

Sign up

7



프로필보기

친구보기

대보기 >

ADS

ADS

8



New York

Sat. 26 November 2010

Buy Tickets



New York

Sat. 26 November 2015

Buy Tickets

3

5

아카이브

Archiv

로고 디자인



Archive

페이지 상세설계

회원관리	회원가입	<ul style="list-style-type: none"> 각각의 ID값을 갖는 Input Box가 공란 없는 하나의 폼으로 작성되게 함. 등록된 회원정보는 회원 테이블에 저장됨.
	로그인	<ul style="list-style-type: none"> 회원가입 시 입력한 ID와PW와 사용자가 입력한 값과 비교하여 일치할 경우 로그인 성공
	아이디찾기	<ul style="list-style-type: none"> 생년월일과 이름을 입력받아 ID값을 호출
	비밀번호찾기	<ul style="list-style-type: none"> 생년월일과 이름과 ID를 입력받아 PW값을 호출
	비밀번호변경	<ul style="list-style-type: none"> ID와 PW를 DB와 비교하여 일치할 경우 새로운 PW를 입력받아 DB에 저장
Main 네비게이션 바	로그버튼	<ul style="list-style-type: none"> 최상단 중앙에 위치한 로그 버튼으로 홈 화면으로 링크됨
	검색창	<ul style="list-style-type: none"> 카테고리,키워드를 통한 게시물 검색기능
	signup버튼	<ul style="list-style-type: none"> 회원가입을 위한 모달대화상자가 열리는 버튼으로 우측에 위치
	login	<ul style="list-style-type: none"> 서비스를 이용하기 위해 로그인을 하는 대화상자가 열리는 버튼으로 우측에 위치

Archive

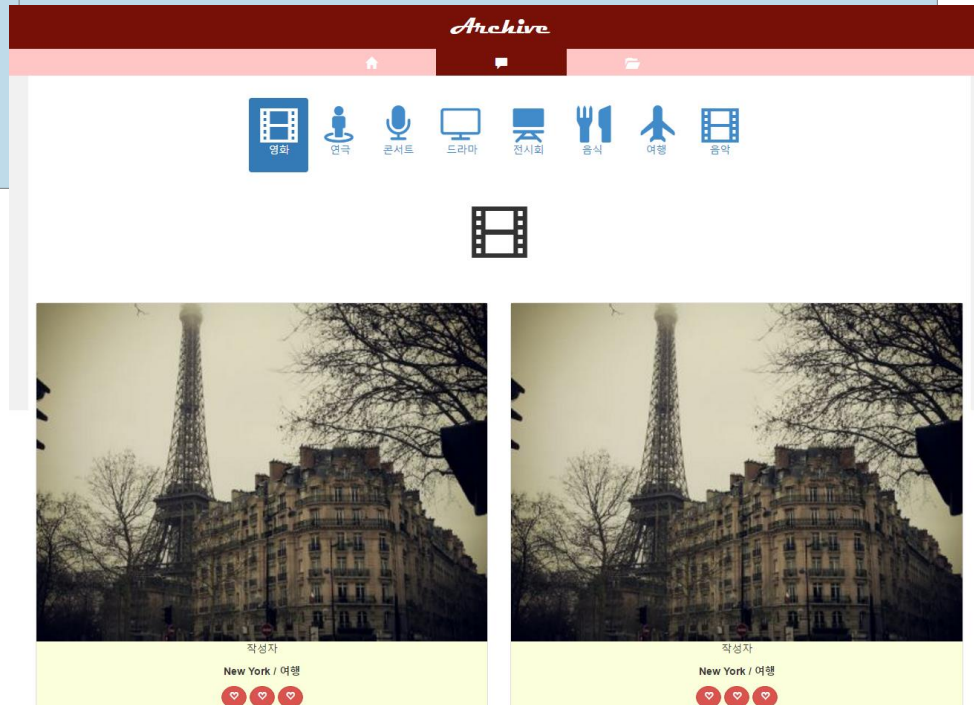
페이지 상세설계

Sub 네비게이션 바	사용자프로필	<ul style="list-style-type: none"> ▪ 사용자가 로그인하게 되면 사용자의 로그인상태를 알리기 위해 아이디와 프로필사진을 작게 표시하게 함
	홈 버튼	<ul style="list-style-type: none"> ▪ 각각의 사용자들이 올린 게시물이 나열되는 페이지로 이동하는 버튼
	카테고리 버튼	<ul style="list-style-type: none"> ▪ 음식,영화,공연,스포츠 등등 카테고리 하나를 선택을 하여 그에 맞는 게시물만 보여주는 페이지로 이동하는 버튼
	저장소 버튼	<ul style="list-style-type: none"> ▪ 내가 올린 게시물과 내가 Favorite한 게시물을 모두 보여주는 페이지로 이동하는 버튼
	게시글작성 버튼	<ul style="list-style-type: none"> ▪ 게시글을 작성하는 폼이 있는 페이지로 이동하는 버튼
	친구목록 버튼	<ul style="list-style-type: none"> ▪ 나의 친구들의 목록을 보여주는 페이지로 이동하는 버튼. 친구목록에 나온 친구들의 프로필을 클릭하면 친구의 저장소로 이동할 수 있다.
	알림	<ul style="list-style-type: none"> ▪ 다른 사람이 나의 게시물을 공감하거나,Favorite한 이벤트, 또는 Toline 신청이 걸려온 즉시 사용자에게 알림
푸터 네비게이션 바	게시물작성 버튼	<ul style="list-style-type: none"> ▪ Sub 네비게이션 바에도 있지만 사용자가 게시물 업로드가 주 기능인 Archive에서는 푸터 에도 위치하게 함
	설정 버튼	<ul style="list-style-type: none"> ▪ Archive 테마 변경 등 각종 설정변경 페이지로 링크된 버튼
	Info 버튼	<ul style="list-style-type: none"> ▪ 앱 개발 정보가 간략히 표시된 공간(개발자, 개발도구 등등...)

Archive



페이지 상세설계

메인페이지	사용자정보	<ul style="list-style-type: none"> 사용자의 커버사진과 프로필 사진, 게시물 숫자, Favorite받은 숫자, 공감 숫자등을 표시해주는 틀로 메인페이지 상단에 위치함
	게시물공간	<ul style="list-style-type: none"> Toline으로 연결되어 있는(친구)들이 올린 게시물이 나열되는 공간으로 댓글을 남기거나 Favorite, 공감등 기능을 수행 할 수 있다.
카테고리 페이지		<ul style="list-style-type: none"> 카테고리 버튼에서 선택한 기능에 맞게 카테고리 맞는 모든 사용자들의 게시물을 찾아볼 수 있는 페이지



Archive

페이지 상세설계

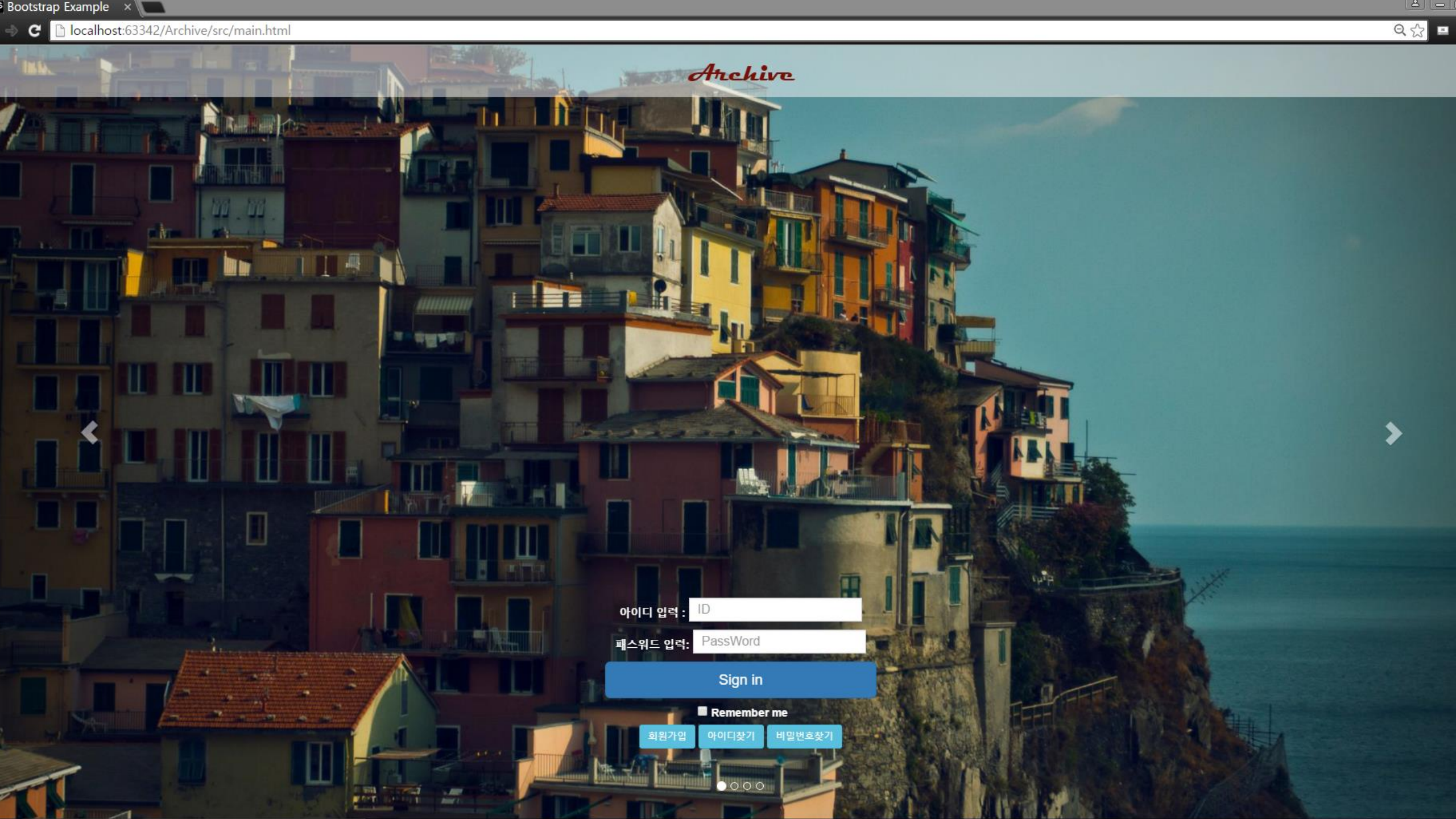
저장소 페이지		<ul style="list-style-type: none"> ▪ 내가 올린 게시물이나 내가 Favorite한 게시물을 <u>한번에</u> 볼 수 있게 하는 페이지
		<ul style="list-style-type: none"> ▪ 저장소 <u>첫페이지</u> : 내 게시물만 볼지, 내가 <u>페이비릿</u>하여 저장소에 <u>담아</u>분 게시물도 같이 볼지 선택할 수 있으며, 그 밑에 저장소 폴더가 나열된다.(사용자는 게시물 등록 또는 <u>페이비릿</u>시에 저장할 폴더를 선택하게 되며, 폴더를 추가할 수 있다.) 폴더를 클릭하게 되면 "폴더 페이지"로 이동한다. 맨 아래에 폴더별로 <u>게시글</u>이 몇 개씩 분포되어 있는지 캔버스로 도표화 되어 표시된다. (추후 감성분포도도 추가) ▪ <u>폴더 페이지</u> : 폴더로 들어가게 되면 폴더의 <u>최근사진</u>이 <u>커버사진</u>에 보여지고, <u>게시글수</u>와 <u>공감수</u>와 <u>분석표보기</u> 버튼이 있다. <u>분석표</u> 보기를 누르면 <u>폴더내 게시물</u>들의 <u>감성분석표</u>가 <u>팝업창</u>으로 보여지게된다.그 밑으로 <u>폴더안에있는 게시물</u>들이 <u>열로</u> 배치된다.(기본화면:2줄/ 모바일화면:1줄) <u>게시글</u> 작성자가 자신일 경우 작성자사진에 "나"라고 표시된다.

Archive

페이지 상세설계

게시글작성 페이지	게시글 작성	<ul style="list-style-type: none"> ▪ 사용자는 게시글의 열람권한(나만보기,Toline,전체공개),카테고리를 필수 입력하고, 사진 1장을 불러오고 두줄 평을 입력하여 게시물을 업로드 한다.
친구목록 페이지	친구목록 관리	<ul style="list-style-type: none"> ▪ 원하는 친구의 저장소로 이동하거나 나의 친구 목록을 볼 수 있는 페이지 ▪ Toline 요청을 수락하거나 Toline을 해제 할 수 있다.
알림페이지	알림	<ul style="list-style-type: none"> ▪ 나에게 Toline들어오거나, 나의 게시물에 대한 공감이나,Favoite등 각종 Event를 사용자에게 알려주는 페이지
우측화면 분할	각종 추천	<ul style="list-style-type: none"> ▪ 사용자가 올린 두줄 평의 단어를 분석하여 얻는 감정(감정수치)를 이용하여 마음이 맞는 사용자 추천,문화생활 패턴분석을 통한 추천, 비슷한 감정을 가졌던 다른 게시물등을 2차생산물로서 추천한다.

Archive



Archive



아이디 입력 :

ID

패스워드 입력:

PassWord

Sign in

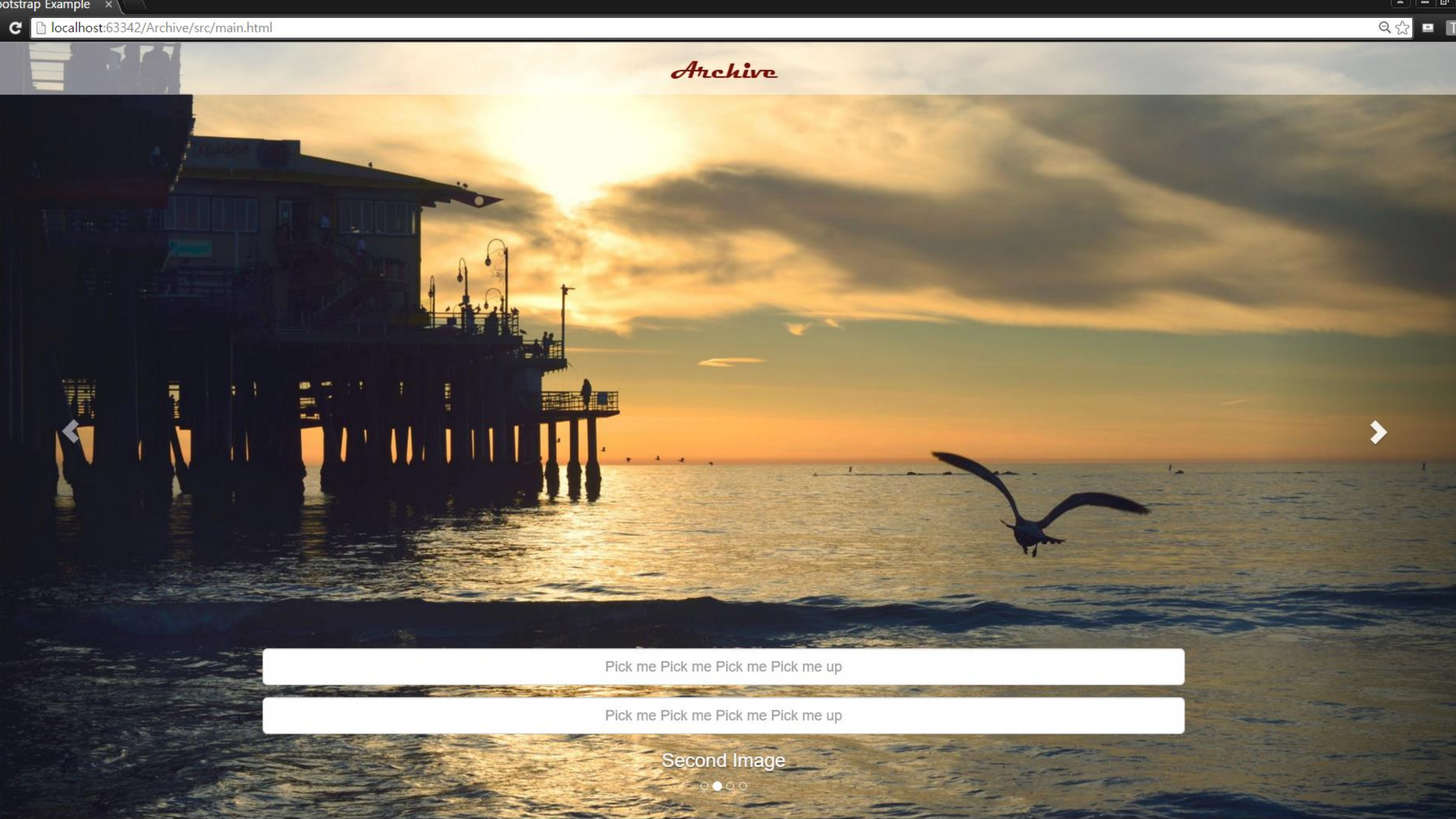
☐ Remember me

회원가입

아이디찾기

비밀번호찾기





Archive

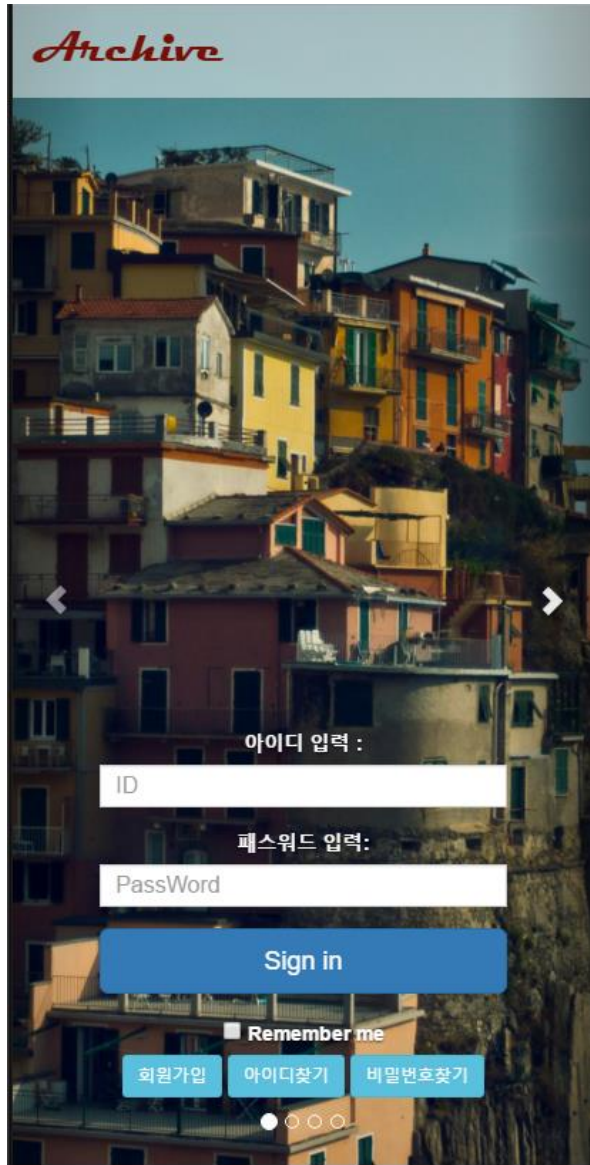
Pick me Pick me Pick me Pick me up

Pick me Pick me Pick me Pick me up

Second Image



모바일 메인화면



로그인 시 메인화면으로 이동하도록 수정

(로그인 관련 처리 모두 수행 / 보안성 향상)

Archive

Archive

백엔드
개발

Maven 프로젝트

```
ngEx
src/main/java
  com.javalec.spring_ex_pjt
    > HomeController.java
    > Member.java
  com.javalec.spring_ex_pjt_board_comman
    > BCommand.java
    > BContentCommand.java
    > BDeleteCommand.java
    > BListCommand.java
    > BModifyCommand.java
    > BReplyCommand.java
    > BReplyViewCommand.java
    > BWriteCommand.java
  com.javalec.spring_ex_pjt_board_controll
    > BController.java
  com.javalec.spring_ex_pjt_board_dao
    > BDao.java
  com.javalec.spring_ex_pjt_board_dto
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.6]
Maven Dependencies
src
target
pom.xml
```

```
<properties>
<java-version>1.8</java-version>
<org.springframework-version>3.1.1.RELEASE</org.springframework-
version>
<org.aspectj-version>1.6.10</org.aspectj-version>
<org.slf4j-version>1.6.6</org.slf4j-version>
</properties>
<repositories>
<repository>
  <id>oracle</id>
  <name>ORACLE JDBC Repository</name>
  <url>http://maven.jahia.org/maven2</url>
</repository>
</repositories>
```

```
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-jdbc</artifactId>
<version>4.1.4.RELEASE</version>
</dependency>
```

```
<dependency>
<groupId>org.mybatis</groupId>
<artifactId>mybatis</artifactId>
<version>3.2.8</version>
</dependency>
<dependency>
<groupId>org.mybatis</groupId>
<artifactId>mybatis-spring</artifactId>
<version>1.2.2</version>
</dependency>
```

```
<dependency>
  <groupId>com.oracle</groupId>
  <artifactId>ojdbc6</artifactId>
  <version>11.1.0.7.0</version>
</dependency>
```

Arch

로그인/ 회원가입

```
@RequestMapping(value="/sample/regist.do")
public ModelAndView regist(CommandMap commandMap,
    HttpServletRequest request) throws Exception{
    ModelAndView mv = new
    ModelAndView("redirect:/sample/login.do");

    sampleService.regist(commandMap.getMap(), request);

    return mv;
}
```

```
@RequestMapping(value="/sample/checkIdpw.do")
public ModelAndView checkIdpw(String id, String pw, HttpSession session)
throws Exception{
    ModelAndView mv = new ModelAndView("/views/layout");

    HashMap<String, String> map = new HashMap<String, String>();
    map.put("id", id);
    map.put("pw", pw);

    int loginIdentify = sampleService.checkIdpw(map);

    System.out.println("loginIdentify = " + loginIdentify);

    if (loginIdentify == 1){
        session.setAttribute("id", id);
        session.setAttribute("pw", pw);
        return mv;
    }

    ModelAndView mvc = new ModelAndView("redirect:/sample/login.do");

    return mvc;
}
```

Archive

Tiles

연동 레이아웃 템플릿

```
<tiles-definitions>
<definition name="mainTemplate" template="/WEB-INF/jsp/views/layout.jsp">
  <put-attribute name="title" value="전자정부프레임워크 따라하기" />
  <put-attribute name="header" value="" />
  <put-attribute name="header2" value="" />
  <put-attribute name="left_wall" value="" />
  <put-attribute name="right_wall" value="" />
  <put-attribute name="content" value="" />
  <put-attribute name="footer" value="" />
</definition>
```

인터페이스 레이아웃

*<!-- 아래 name을 "/sample/egovSampleList"라고 정합니다.
"/sample/egovSampleList"는 컨트롤러가 리턴하는 값입니다.₩
컨트롤러에서 리턴할 때 맨앞에 "/"가 포함되어 있다는 점을 잊지*

마세요. -->

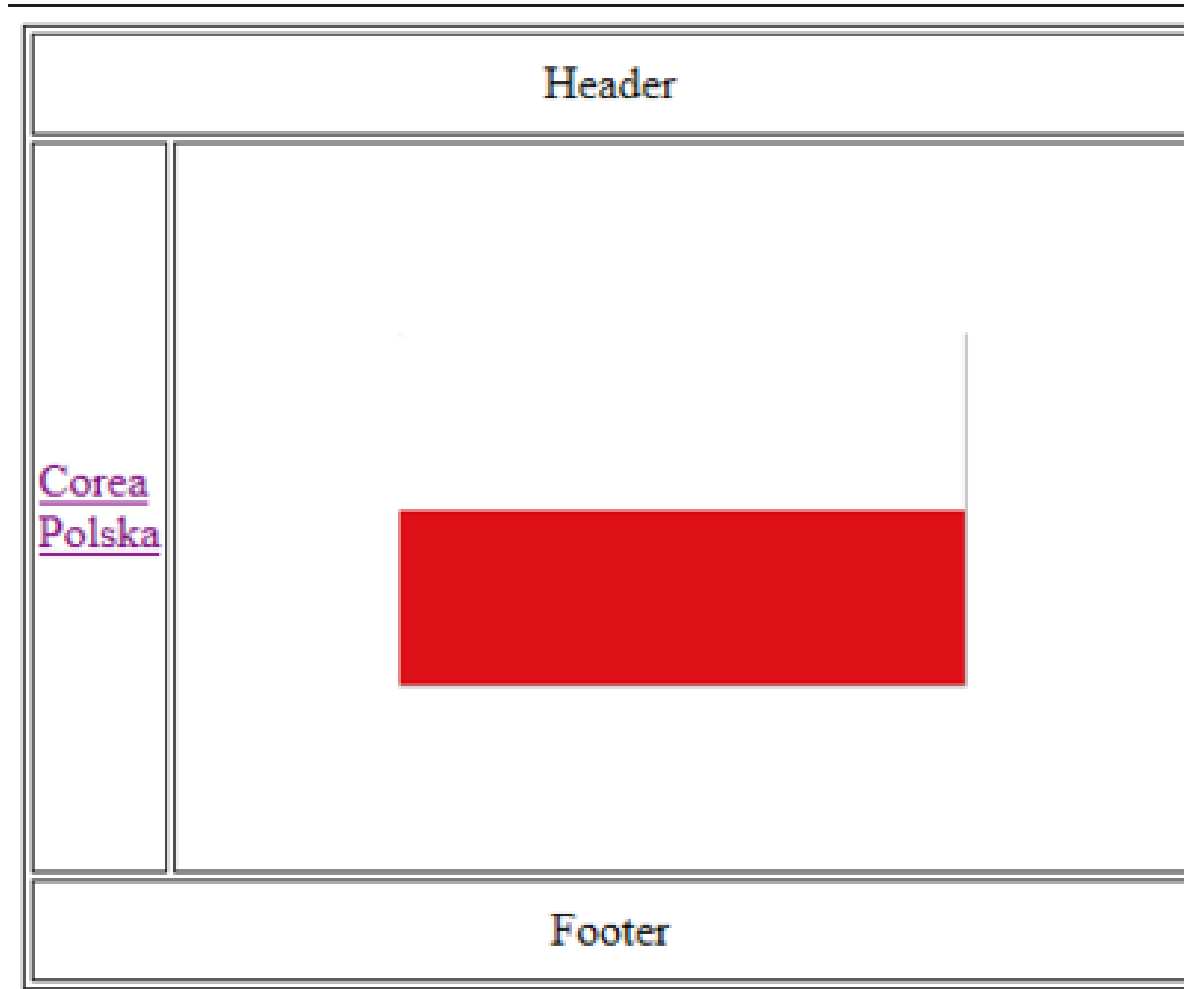
```
<definition name="/views/layout" extends="mainTemplate">
  <put-attribute name="title" value="" />
  <put-attribute name="header" value="/WEB-INF/jsp/views/mainHeader.jsp" />
  <put-attribute name="header2" value="/WEB-INF/jsp/views/subHeader.jsp" />
  <put-attribute name="left_wall" value="" />
  <put-attribute name="right_wall" value="" />
  <put-attribute name="content" value="/WEB-INF/jsp/views/content.jsp" />
  <put-attribute name="footer" value="/WEB-INF/jsp/views/mainFooter.jsp" />
</definition>
```

페이지별 레이아웃 템플릿

Archive

Tiles

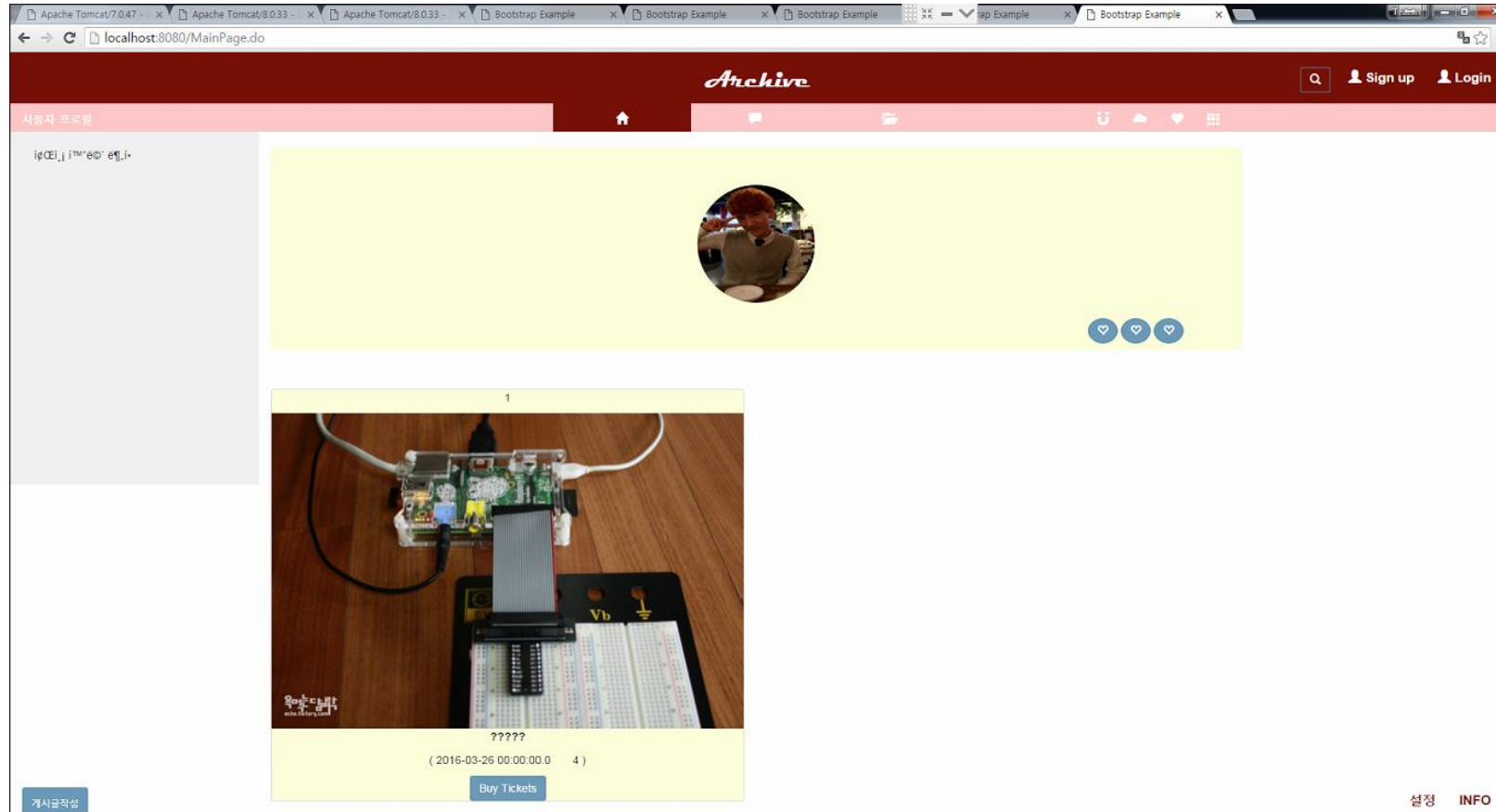
연동 레이아웃 템플릿



Archive

Tiles

연동 레이아웃 템플릿



Archive

Archive

계 획

계 획



veloper	back end developer	Team leader	이름
	(이호세아)	(김태선)	

	3월 1주	3월 2주	4월 1주	4월 2주	5월 1주	5월 2주	6월 1주
기획(디자인, 스토리보드)	●						
연구 스터디	●	●					
사이트 맵		●					
메인페이지 제작		▲					
웹사이트 구축							
DB 설계	■	▲					
서버&DB 구축			■	■	■		
비주얼 디자인				■			
페이지 와이어프레임				■	■		
DB 연동					■	■	
DB 검사						■	■
콘텐츠 분석				■	■		
콘텐츠 제작					■	■	
프로토타입 제작						■	■
에러 수정							■
발표							■

Arch

계 획



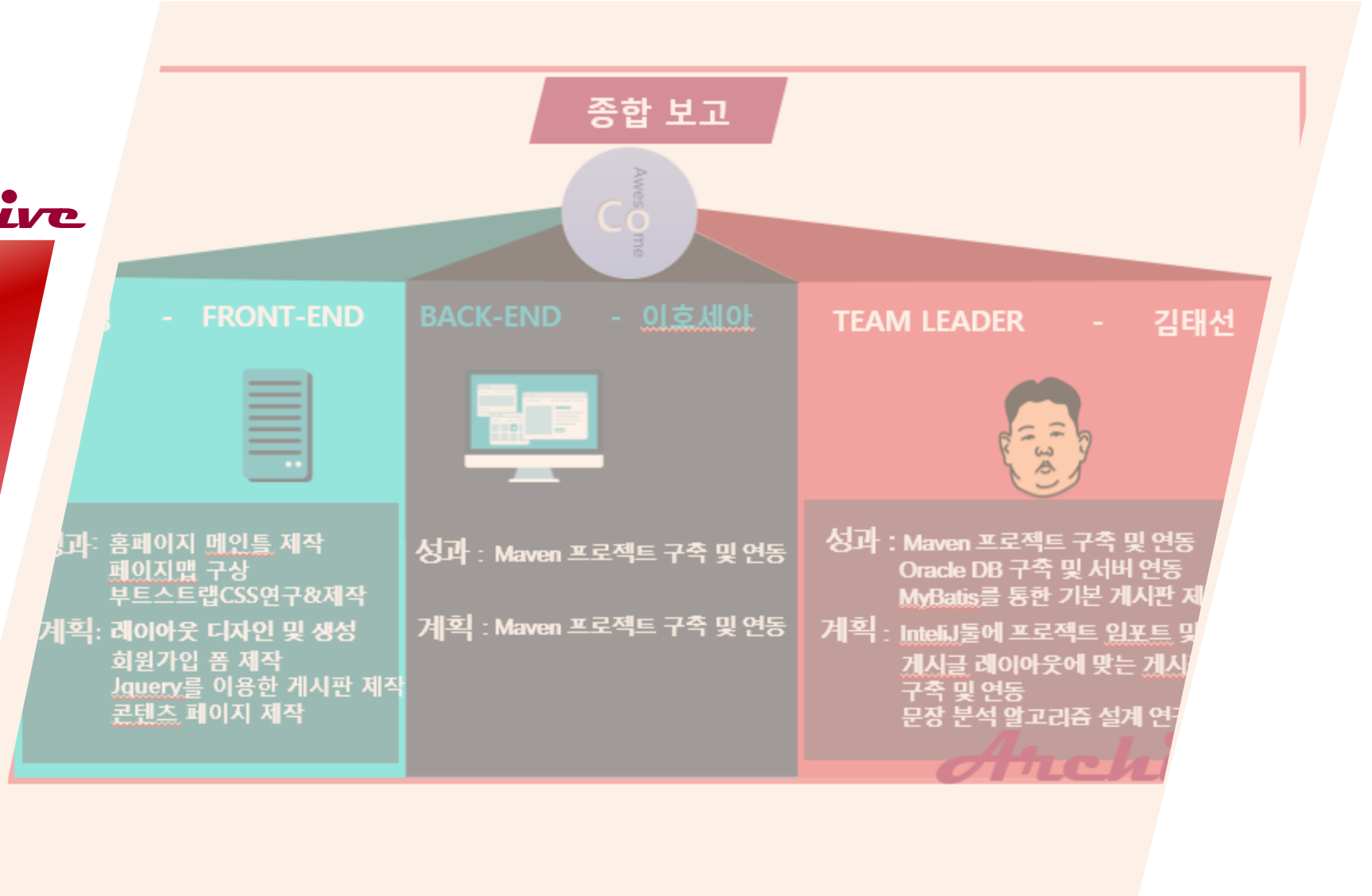
front end developer (이문성)	back end developer (이호세아)	Team leader (김태선)	공통

	3월 초	3월 말	4월 초	4월 말	5월 초	5월 말	6월 초
기획(디자인, 스토리보드)	●						
연구 스터디	●	●	▲				
사이트 맵		●					
메인페이지 제작		●	▲				
웹사이트 구축							
DB 설계		●					
서버&DB 구축							
비주얼 디자인							
페이지 와이어프레임							
DB 연동							
DB 검사							
콘텐츠 분석							
콘텐츠 제작							
프로토 타입 제작							
에러 수정							
발표							

Archive

Archive

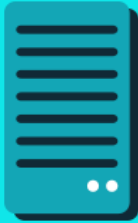
종합
보고



종합 보고



이문성 - FRONT-END



계획 : 저장소 페이지 구축
페이지별 소스 분할
Tiles 레이아웃 연동

BACK-END - 이호세아



계획 : 저장소 DB 설계 및 구축
친구 신청 구축
Tiles 레이아웃 연동

TEAM LEADER - 김태선



계획 : 기능 명세서 최종 설계 및 종합
회원가입/ 저장소 페이지 서버 연동
Tiles 레이아웃 연동

Archive



“ 감사합니다. ”

Archive

허준영 교수님

김 태 선 / 이 문 성 / 이 호 세 아