

SENIOR



20121079
20131114
20141025
20141057

김송명
장지수
권예진
박다솜

Senior Contents

1

시니어 소개

Always bear in mind that your own resolution to succeed is more important than any one thing.

2

시스템 개요 및 기능

Always bear in mind that your own resolution to succeed is more important than any one thing.

3

구조 설계 및 구현 방법

Always bear in mind that your own resolution to succeed is more important than any one thing.

4

역할 분담

Always bear in mind that your own resolution to succeed is more important than any one thing.

5

시연

Always bear in mind that your own resolution to succeed is more important than any one thing.

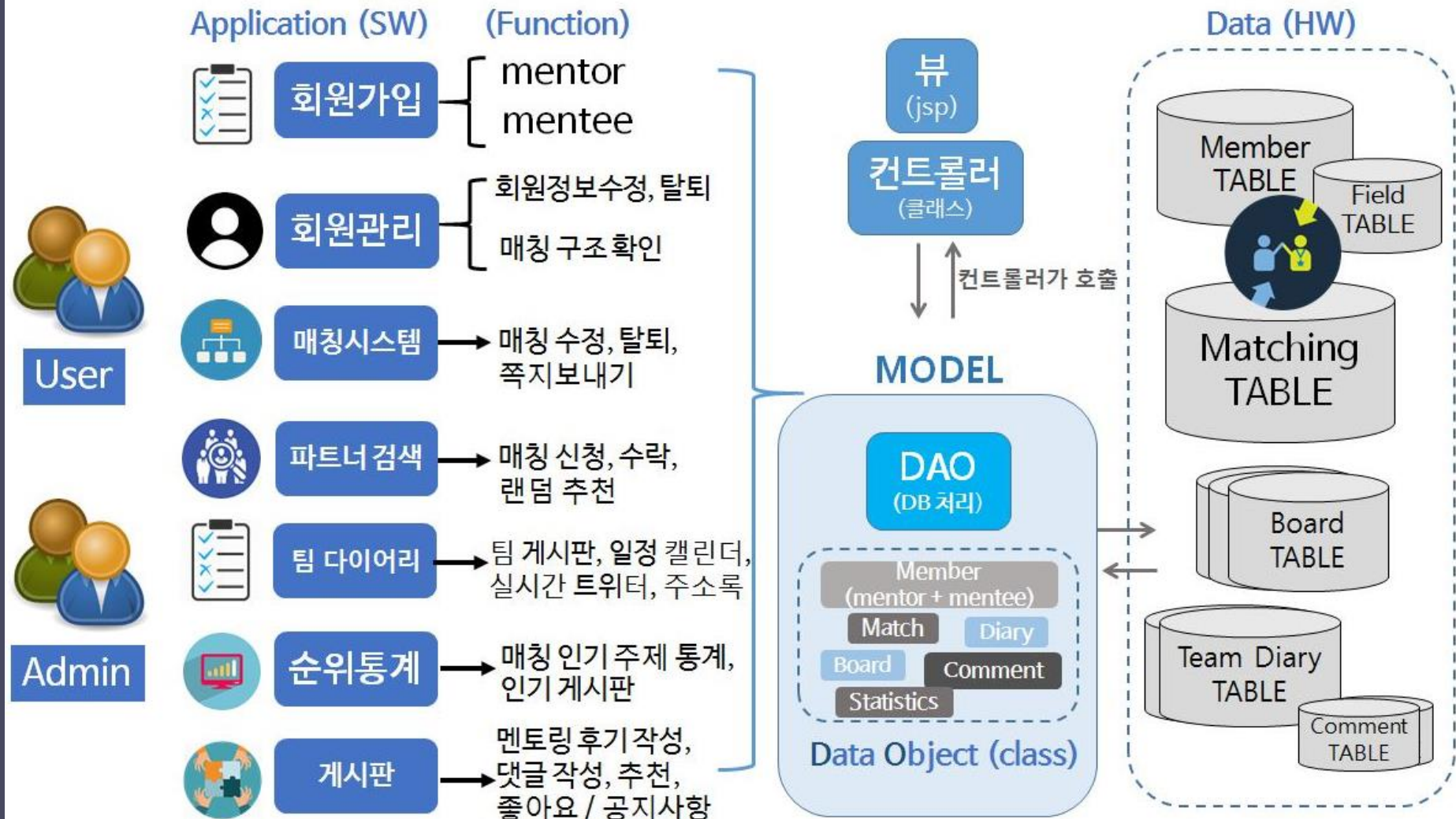
시니어 소개

“시니어”란 은퇴 후에 여가생활이나 사회 활동을 활발히 즐기는 50-60대인 ‘액티브 시니어’분들을 위해 스마트폰이나 컴퓨터를 이용하고 어떤 것이든 배우고 또 자신의 조언이나 노하우 등을 전수해 주고 싶으신 시니어 분들과, 사회나 특정 분야의 유경험자인 시니어의 조언을 필요로 하는 젊은 층의 사람들과 서로 매칭하여 사회 활동을 장려하는 웹 서비스이다.

일반적인 멘티.멘토 서비스 (기존 '프립' 과 같은 취미생활 공유 앱 혹은 멘토 멘티 웹 혹은 앱 서비스) 의 타겟이 주로 2~30대의 젊은 층에 머물러 있다. 자녀도 다 키우고, 노후 마련 준비되어 있는 시니어 세대(통크족)를 타겟으로 한 웹 서비스이다. 또한 시니어들만의 소통공간이 아닌, 젊은 층과 시니어 세대의 활발한 교류의 장을

제공하며, 더 나아가 개개인의 삶의 질을 높일 수 있으며 사회활동을 장려할 수 있다는 장점이 있다.

시스템 개요 및 기능



시스템 개요 및 기능

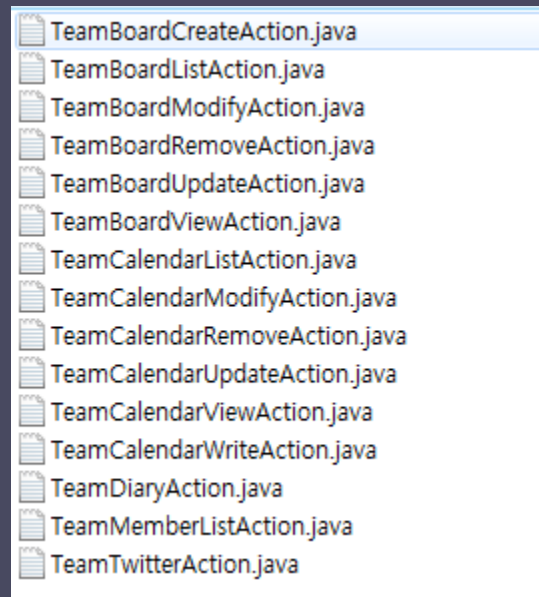
	기능 내용	구현 여부
팀 게시판	게시판 수정, 삭제, 목록 기능	구현완료
팀 주소록	해당 팀에 속한 멤버 정보 출력 기능	구현완료
팀 스케줄러	월/일 별 일정 수정, 삭제, 목록 기능	구현완료
팀 트위터	팀 멤버끼리 실시간 트윗 기능	구현완료
팀 목록이동	팀 메뉴 선택 전의 목록 페이지로 이동	구현완료
공지사항 게시판	관리자 권한으로 공지사항 게시판 등록, 수정, 삭제 가능	구현완료
매칭주제 통계	현재 매칭된 팀들의 주제 인기도를 통계치로 나타내는 기능	구현완료
로그인/회원가입	관리자 혹은 사용자로 로그인 및 회원 가입할 수 있는 기능	구현완료
팀 개설	자신이 멘토로써 팀을 개설하는 기능.	구현완료
멘토 검색	자신의 조건과 비슷한 멘토들을 랜덤으로 추천해주며, 자신이 입력한 조건에 맞는 멘토들이 개설한 팀을 검색해줌.	구현완료
댓글 및 좋아요	멘토링 경험을 한 유저가 남긴 후기 게시판 내부에서 발생하는 기능/ 후기게시판을 읽은 유저는 한 아이디 당, 하나의 게시물에 좋아요가 가능	구현완료
후기게시판 및 인기 게시판	멘토링 경험을 다른 유저들과 공유할 수 있게 만든 게시판 기능	구현완료
팀 매칭	멘토와 멘티는 일 대 다의 구성으로 이뤄지며 팀을 구성함	구현완료
쪽지 보내기	본인이 받은 쪽지목록은 마이페이지의 쪽지함에서 확인이 가능	구현완료

시스템 구조 설계 및 구현 방법

■ 팀 다이어리 기능

- DTO : Team, TeamBoard, TeamMember, (Member)
- DAO : TeamBoardDAO – 모든 Action객체에서 이 객체의 메소드 사용
- Action : 하단 이미지 참조

각 기능별 Action



뷰(jsp) → ^{***.m2}

TeamBoardDAO ↔ DB접근

메소드 사용 위임

TeamManager
(Action이 사용)

시스템 구조 설계 및 구현 방법

■ 팀 다이어리 기능

팀 다이어리 메뉴 클릭 -> team_diary.m2?command=teamDiary (TeamDiaryAction)

(Action에서 로그인이 안되어있으면 로그인 페이지로 이동)

TeamManager.getTeamList(member_id)

team_diary.jsp

(세션에 저장된 회원 아이디로
가입된 팀을 검색하여 보여줌)

특정 팀 클릭

-----> team_menu.jsp?team_id=<%team.getTeam_id()%>

(가입된 팀이 없으면

멘토 검색페이지

search_frame.jsp로 include)

team_menu.jsp (세션에 선택한 팀의 team_id를 저장)

- 1) 팀 게시판
- 2) 팀 주소록
- 3) 팀 스케줄러
- 4) 팀 트위터
- 5) 팀 목록이동

team_menu2.jsp

팀 메뉴

team_main.m2?command=team_main

팀 정보

(TeamMainAction)



Team_diary.jsp에서 특정 팀을 선택한 시점부터, (-> team_menu.jsp)
해당 팀의 team_id와 team_name을 세션에 저장 후, 활용
대부분의 기능이 이 session에 저장된 team_id를 비교하여,
해당 팀에 대한 정보, 자료를 출력하고 요구사항을 제어

시스템 구조 설계 및 구현 방법

■ 팀 다이어리 기능

팀 메뉴
클릭

team_menu2.jsp

①

②

①에서 메뉴 선택시, ② 프레임에 보여줌

- 1) 팀 게시판 `team_board_list.m2?command=team_board_list` 게시판 수정, 삭제, 목록 기능
(TeamBoardListAction)
- 2) 팀 주소록 `team_addr.m2?command=team_member_list` 해당 팀에 속한 멤버 정보 출력 기능
(TeamMemberListAction) `manager.getTeamMemberList(team_id);`
- 3) 팀 스케줄러 `team_calendar_list.m2?command=team_calendar_list`
(TeamCalendarListAction) 월/일 별 일정 수정, 삭제, 목록 기능
- 4) 팀 트위터 `team_tweet.m2?command=team_twitter&team_id=<%=team_id%>`
(TeamTwitterAction) 팀 멤버끼리 실시간 트윗 기능
- 5) 팀 목록이동 `team_diary.m2?command=teamDiary&team_id=<%=team_id%>`
(TeamDiaryAction) 팀 메뉴 선택 전의 목록 페이지로 이동
프레임 제거

시스템 구조 설계 및 구현 방법

■ 팀 다이어리 기능

팀 게시판 team_board_list.jsp (Action에서 사용자가 작성자가 아닐시, 권한 제어
로그인 한 세션의 id와 DTO의 writer_id 비교

1) 글쓰기 클릭 : team_board_write.jsp -> 확인 버튼 클릭 -> team_board_list.m2?
command=team_board_write

TeamBoardCreateAction -> TeamManager.write(Board board) -> team_board_list.m2?command=team_board_list로 포워드

2) 글제목 클릭 : team_board_view.m2?command=team_board_view&
board_num=\${board.board_num}

TeamBoardViewAction -> TeamManager.getBoard(board_num, team_id)-> team_board_view.jsp로 포워드

3) team_board_view.jsp에서 삭제 클릭 -> team_board_list.m2?command=
team_board_remove &writer_id=\${board.writer_id }&board_num=\${board.board_num}

TeamBoardRemoveAction -> TeamManager.getBoard(board_num, team_id)-> team_board_view.jsp로 포워드

시스템 구조 설계 및 구현 방법

■ 팀 다이어리 기능

팀 게시판 team_board_list.jsp (Action에서 사용자가 작성자가 아닐시, 권한 제어)
로그인 한 세션의 id와 DTO의 writer_id 비교

4) team_board_view.jsp에서 수정 클릭 -> *team_board_modify.m2?command=*
(편집 페이지) *team_board_update&writer_id=\${board.writer_id}*
&board_num=\${board.board_num}

TeamBoardUpdateAction -> *TeamManager.getBoard(request.getParameter("board_num")) team_id); ->*

team_board_modify.jsp로 포워드

5) team_board_modify.jsp에서 수정 클릭 -> *team_board_view.m2?command=*
team_board_modify&writer_id=\${board.writer_id} &board_num=\${board.board_num}

TeamBoardModifyAction -> *TeamManager.update(board); ->* *team_board_view.jsp로 포워드*

6) team_board_modify.jsp에서 목록 클릭 -> *team_board_list.m2?command=team_board_list*

TeamBoardListAction -> *TeamManager.getBoardList(team_id); ->* *team_board_list.jsp로 포워드*

시스템 구조 설계 및 구현 방법

■ 팀 다이어리 기능

팀 스케줄러

team_calendar_list.jsp

(Action에서 사용자가 작성자가 아닐시, 권한 제어)
로그인 한 세션의 id와 DTO의 writer_id 비교

1) 달력에서 특정 일 클릭(일정 추가) : team_calendar_write.jsp?month="+ (m+1) +"&day="+ i (반복문내부)

2) team_calendar_write.jsp(일정 작성 폼) : *team_calendar_list.m2?command=team_calendar_write*

TeamCalendarWriteAction -> TeamManager.calendar_write(calendar); team_calendar_list.m2?command=team_calendar_list 로
(TeamCalendarListAction) 포워드

3) team_calendar_list.jsp에서 일정 클릭 -> team_calendar_view.m2?command=team_calendar_view&
calendar_id="+calendar.getCalendar_id()

TeamCalendarViewAction -> TeamManager.getCalendar(calendar_id, team_id); team_calendar_view.jsp로 포워드

4) team_calendar_view.jsp에서 수정 클릭 -> team_board_update.m2?command=team_calendar_update
&writer_id=\${calendar.writer_id}&calendar_id=\${calendar.calendar_id}

TeamCalendarUpdateAction -> manager.getCalendar(calendar_id, team_id); -> team_calendar_update.jsp로 포워드

5) team_calendar_view.jsp에서 삭제 클릭 -> team_board_list.m2?command=team_calendar_remove&
writer_id=\${calendar.writer_id}&calendar_id=\${calendar.calendar_id}

TeamCalendarRemoveAction -> TeamManager.calendar_remove(calendar_id, team_id); -> team_calendar_list.m2?
command=team_calendar_list 로 포워드

시스템 구조 설계 및 구현 방법

■ 팀 다이어리 기능

팀 트위터 team_twitter_list.jsp

`team_tweet.m2?command=team_twitter&team_id=<%=team_id%>` team_tweet.jsp 로 포워드
(TeamTwitterAction)

```
team_tweet.jsp -> sendRedirect-> team_twitter_list.jsp");
```

구현 : 세션 객체에 실시간 메시지를 저장하고,
실시간으로 전달된 메시지 리스트를 Application 객체에 저장.

처음 메시지를 등록 할 때,
사용자가 초기에 선택한 해당 team_id값과
사용자 아이디인 member_id값을 message List 객체에 넣은 후, team_id를 비교

➡ 현재 접속한 팀에 대한 메시지만 (team_id가 같은 메시지만)
새로운 list객체에 저장하여 그 객체를 team_twitter_list에 전달

시스템 구조 설계 및 구현 방법

■ 팀 다이어리 기능

팀 주소록 team_addr.jsp

`team_addr.m2?command=team_member_list` 해당 팀에 속한 멤버 정보 출력 기능

```
(TeamMemberListAction)    manager.getTeamMemberList(team_id);
```

구현 : DAO객체에서 team_id를 매개변수로 사용하여,
SQL문에서 해당 team_id를 가진 (where team_id=?)
멤버리스트를 가지고 온 후, team_addr.jsp에서 각 회원 정보를 출력

MemberDTO와 TeamMemberDTO를 join하여 사용
TeamMember의 특정 team_id에 대한 member_id 값 리스트를 가져와
Member에서 해당 member_id값들에 대한 행들을 출력

시스템 구조 설계 및 구현 방법

■ 공지사항 게시판

공지사항 게시판

사용자와 관리자 모두 공지사항 리스트 열람 가능하나, 공지사항 게시물 등록, 수정, 삭제는 관리자만이 가능

- 1) 글쓰기 클릭 : qna_board_write.jsp-> 등록 버튼 클릭 -> NotiBoardAddAction.m2?
command=NotiBoardAddAction
- 2) 글제목 클릭 : *NotiBoardDetailAction.m2?*
num=<%=bl.getNOTI_NUM()%>&command=NotiBoardDetailAction
- 3) qna_board_view.jsp에서 삭제 클릭 -> *NotiBoardDelete.m2?command=*
NotiBoardDeleteAction&num=<%=board.getNOTI_NUM() %>
- 4) qna_board_view.jsp에서 수정 클릭 -> *NotiBoardModify.m2?command=*
(편집페이지) *NotiBoardModify&num=<%=board.getNOTI_NUM() %>*
- 5) qna_board_modify.jsp에서 수정 클릭 -> modifyform.command.value=
"NotiBoardModifyAction";-> NotiBoardModifyAction.m2?
command=NotiBoardModifyAction
- 6) qna_board_modify.jsp에서 목록 클릭 -> NotiBoardListAction.m2?command=NotiBoardListAction

시스템 구조 설계 및 구현 방법

- 매칭 주제 인기순위 통계 기능 현재 매칭된 팀들의 주제들 중 인기가 높은 주제들을 나열하여 통계치를 제공

통계 pServlet.jsp

“ Google Chart Library jsp내부에 import하여 사용 ”

1) Jsp파일에 라이브러리 로딩 : `src=https://www.gstatic.com/charts/loader.js`

2) Chart그리는 함수 구현: `function drawChart()`

3) Chart Data 셋팅 :

```
var data = google.visualization.arrayToDataTable([  
    ['Country', 'Area(square km)'],  
    <c:forEach items="${pieDataList}" var="entry">  
        [ '${entry.key}', ${entry.value} ],  
    </c:forEach>  
]);
```

4) Chart Options 설정

```
var options = {  
    'title' : 'Senior Mentoring Matching Team', //title which will be shown right above the Google Pie Chart  
    is3D : true, //render Google Pie Chart as 3D  
    pieSliceText: 'label', //on mouse hover show label or name of the Country  
    tooltip : {showColorCode: true}, // whether to display color code for a Country on mouse hover  
    'width' : 900, //width of the Google Pie Chart  
    'height' : 500 //height of the Google Pie Chart  
};
```

시스템 구조 설계 및 구현 방법

- 매칭 주제 인기순위 통계 기능 현재 매칭된 팀들의 주제들 중 인기가 높은 주제들을 나열하여 통계치를 제공

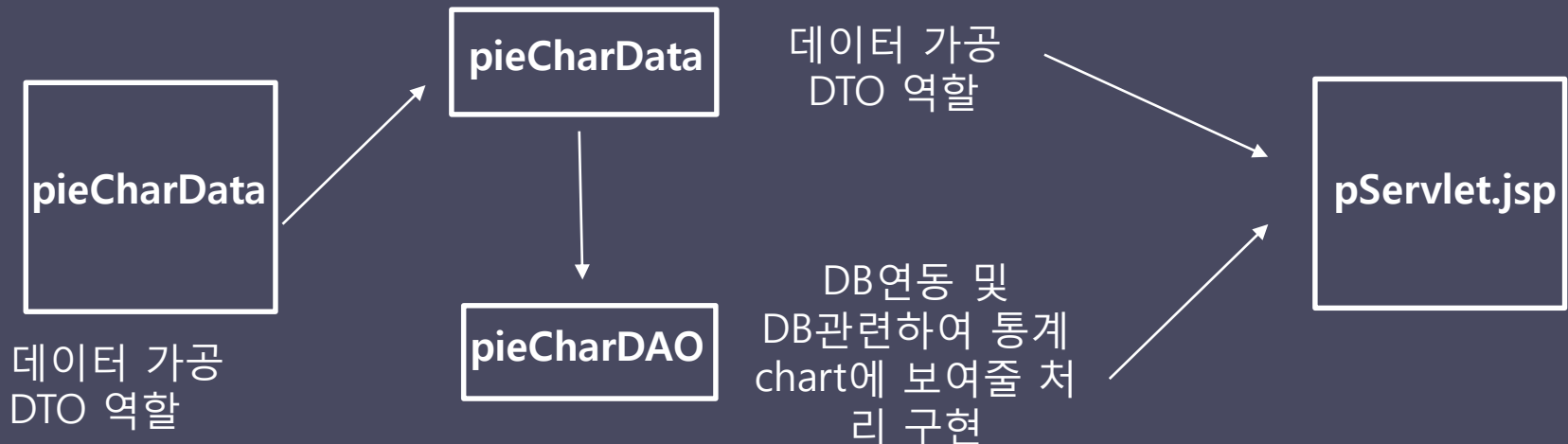
통계 pServlet.jsp

“ Google Chart Library jsp내부에 import하여 사용 ”

5) Chart 그리기 :

```
// Instantiate and draw our chart, passing in some options.  
var chart = new google.visualization.PieChart(document.getElementById('chart_div'));  
chart.draw(data, options);
```

6) Google Chart Library ->오라클 데이터베이스와 연동하여 응용
: 흐름도



시스템 구조 설계 및 구현 방법

■ 매칭 기능

1) 매칭 기능-①


- 기능 설명: 로그인한 완료된 유저는 팀만들기 기능을 통해 멘토가 될 수도 있고 동시에 팀 신청을 통해 멘티가 되는 것이 가능하다. 멘토와 멘티는 일 대 다의 구성으로 이뤄지며 멘티가 되고 싶은 유저는 팀에 지원서를 보내고 팀을 개설한 멘토는 이 내용을 확인한 후 일반 유저를 팀멤버로 승격시킨다.(유저는 멘티가 된다.)

2) 매칭 기능-②구조

- DTO: Apply, Member, TeamMember
- DAO: ApplyDAO
- Action:

▶  applyCreateAction.java

▶  applyRemoveAction.java

▶  teamMemberCreateAction.java

▶  RetrieveApplyListAction.java

▶  retrieveApplyListByMemeber_idAction.java

시스템 구조 설계 및 구현 방법

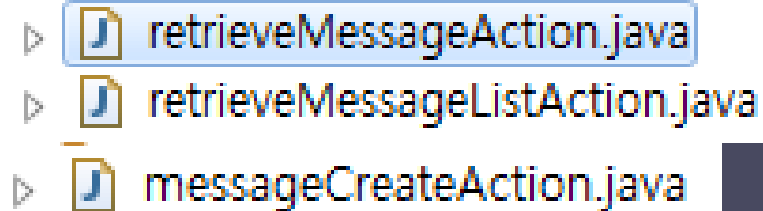
■ 쪽지 기능

1) 쪽지 기능-①

- 기능 설명: 로그인한 완료된 유저는 자신이 원하는 상대방에게 직접적인 통신을 위해 추가한 기능이다.본인이 받은 쪽지목록은 마이페이지의 쪽지함에서 확인이 가능하다.

2) 쪽지 기능-②구조

- DTO: Message, Member
- DAO: MessageDAO
- Action:



```
▶ retrieveMessageAction.java
▶ retrieveMessageListAction.java
▶ messageCreateAction.java
```

■ 댓글 기능

1) 댓글 기능-①

- 기능 설명: 멘토링 경험을 한 유저가 남긴 후기 게시판 내부에서 발생하는 기능이다. 게시판 내부의 글의 피드백을 위한 목적으로 이용되며, 실시간으로 댓글 수를 카운트한다.

시스템 구조 설계 및 구현 방법

■ 댓글 기능(계속)

2) 댓글 기능-②구조

- DTO: Board
- DAO: BoardDAO
- Action:

■ 후기 게시판 기능

1) 후기 게시판 기능-①

- 기능 설명: 멘토링 경험을 다른 유저들과 공유할 수 있게 만든 게시판 기능이다.
게시물을 등록한 사용자만 수정/삭제가 가능하며 로그인한 유저만 이용이 가능하다.
사용자의 요구에 따라 게시판 검색이 가능하며, 조회수, 추천수를 이용하여 인기게시물이
출력 가능하다.

시스템 구조 설계 및 구현 방법

■ 후기 게시판 기능(계속)

2) 후기 게시판 기능-②구조

- DTO: Board
- DAO: BoardDAO
- Action:

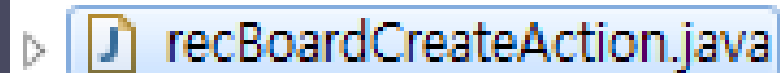
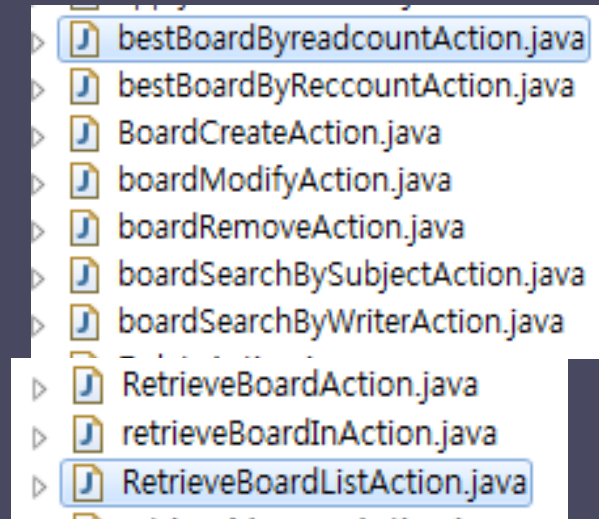
■ 좋아요 기능

1. 좋아요 기능-①

-기능 설명: 후기게시판을 읽은 유저는 한 아이디 당, 하나의 게시물에 좋아요가 가능하다. 추후 좋아요의 숫자를 바탕으로 인기게시물을 선별하여 다른 유저에게 통계자료를 보여줄 수 있다.

2. 좋아요 기능-②구조

- DTO: RecBoard, Board, Member
- DAO: RecDAO
- Action:



시스템 구조 설계 및 구현 방법

■ 로그인/회원가입 기능

1) 기능설명

회원제를 통한 멘토/멘티 매칭을 구현하기 위한 기능과 동시에 로그인시 사용한 아이디로 관리자와 일반 회원들을 분리하여 관리자는 공지사항에 중요공지를 업로드해주는 역할을 하며, 일반 회원들은 활발한 멘토링 활동을 하게 된다.

- 회원가입, 회원정보수정, 회원탈퇴, 회원정보조회 기능

2) 구조 : DTO : MemberDTO

DAO : MemberDAO

Action : DeleteAction, InsertAction, LoginAction, ListAction,
LogoutAction, MemberInfoAction, InsertFormAction,
LoginFormAction, UpdateFormAction, UpdateAction,
ViewAction

3) 흐름도 : index.jsp에서 로그인 폼의 회원가입 버튼 클릭 -> join.m2?command=insertForm -> join.jsp -> 회원가입 양식 작성
-> join1.m2?command=insert -> index.jsp(가입완료) ->

로그인 폼을 입력한 후 로그인 버튼 클릭 -> login.m2?command=login
-> index.jsp(로그인 완료)

시스템 구조 설계 및 구현 방법

■ 멘토 검색 기능

1) 기능설명

자신의 조건과 비슷한 멘토들을 랜덤으로 추천해주며, 자신이 입력한 조건에 맞는 멘토들이 개설한 팀을 검색해줌.

2) 흐름도 :

Index.jsp에서 메뉴의 멘토찾기 버튼 클릭 -> *search_frame.jsp* ->

1. *random_partner.m2?command=random* (상단에 랜덤 추천)
2. *search_partner.jsp* -> 검색버튼 클릭 -> *SearchList.jsp* -> 검색완료된 리스트

■ 팀 개설 기능

1) 기능설명

자신이 멘토로써 팀을 개설하는 기능.

2) 흐름도 :

Index.jsp에서 멘토팀 버튼 클릭 -> *team_list.m2?command=retrieveTeamList* -> 팀개설 버튼 클릭 -> *create_team.jsp* -> 등록버튼 클릭 -> *team.m2?command=team_insert* -> 팀 개설 후 *team_list.m2?command=retrieveTeamList* 로 목록 이동

역할 분담



김송명

후기 게시판,
인기 게시판,
팀 매칭,
썸해보내기



장지수

팀 다이어리 기능
(내부 4가지 기능),
랜덤 추천



권예진

공지사항 게시판
Google Chart 인기순위
통계 기능
PPT 제작



박다솜

멘토 검색 기능
로그인/회원가입,
팀 개설

시연



Thank You