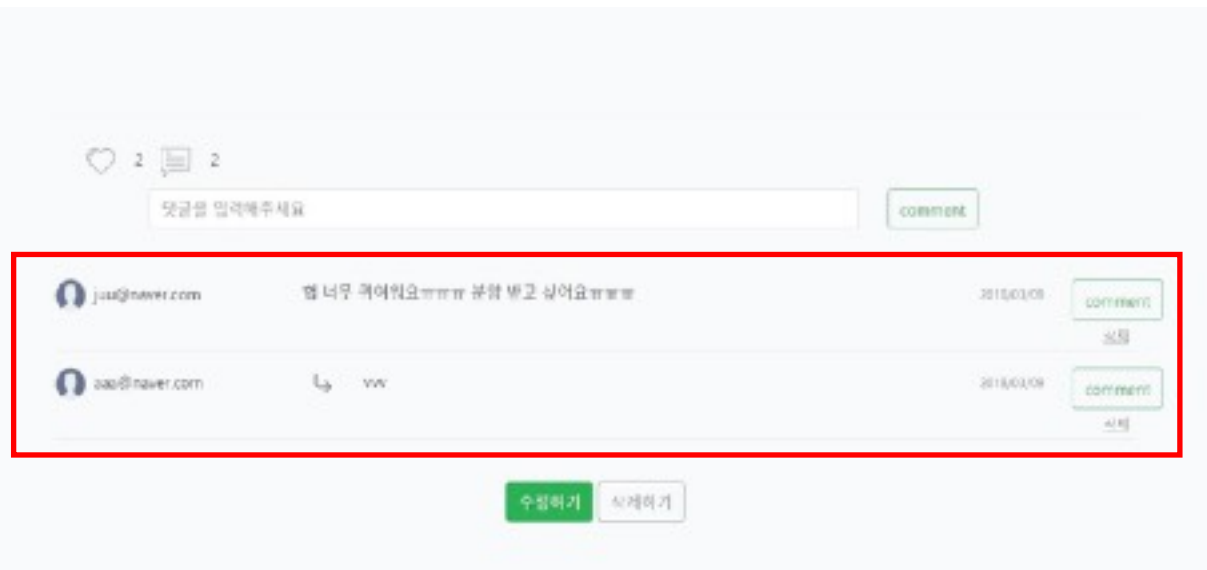


DengDengE 개발 기능(상세)

3. 디테일 페이지(댓글 모든페이지 동일)



1. Ajax를 통해 실시간으로 댓글이 달린다.

```
function addComment(ref, step, depth, index) {  
  $.ajax({  
    url: 'AnimalCommentController',  
    method: 'POST',  
    data: { command: 'addComment', ref: ref, step: step, depth: depth, content: $('#content'+index).val() },  
    success: function (data) {  
      $('#comment-area').children().remove();  
  
      var comments = JSON.parse(data);  
  
      $('#commentCount').html(comments.length);  
      for(var i= 0; i < comments.length; i++){  
        printCommentHtml(comments[i], (i+1));  
      }  
      $('#content0').val('');  
    }  
  });  
}
```

Controller

```
if(command.equals("addComment")) {  
  AnimalCommentService service = AnimalCommentService.getInstance();  
  
  if(Delegator.checkSession(req, resp)) {  
    HttpSession session = req.getSession();  
    User current_user = (User)session.getAttribute("current_user");  
  
    int ref = Integer.parseInt(req.getParameter("ref"));  
    int step = Integer.parseInt(req.getParameter("step"));  
    int depth = Integer.parseInt(req.getParameter("depth"));  
    String content = (String)req.getParameter("content");  
  
    AnimalCommentDto comment = new AnimalCommentDto();  
    comment.setContent(content);  
    comment.setDepth(depth);  
    comment.setRef(ref);  
    comment.setStep(step);  
    comment.setTarget_user_seq(current_user.getSeq());  
    comment.setUser_email(current_user.getEmail());  
  
    List<AnimalCommentDto> list = service.addComment(comment);  
  
    String json = new Gson().toJson(list);  
  
    resp.getWriter().write(json);  
  }  
}
```

Service

```
public List<AnimalCommentDto> addComment(AnimalCommentDto comment) {  
  // update step  
  aniComManagaer.updateStep(comment);  
  
  // add comment  
  boolean result = aniComManagaer.addComment(comment);  
  
  List<AnimalCommentDto> list = new ArrayList<>();  
  
  // get comment list  
  if (result) {  
    list = aniComManagaer.getAllComments(comment.getRef());  
  }  
  return list;  
}
```

DB

DAO

4. 글 작성 페이지(분양동물)

이름	이희	나이	5	모/간주	선	성별	유기독물	*
주소	서울 강남구 강남대로 288			주소찾기	접종여부	<input type="radio"/> YES	* NO	
음성률	<input type="radio"/> YES	* NO	성명	*	예비	<input type="radio"/> 남자		
제목	남녀의 책임분담							
첨가사실	없음							
<div> B I T S E X * A * C O N T R I B </div> <p>대의를 달았자를 제외한 모든 문물 있습니다</p>								
내용								

```

1 $('#summernote2').summernote({
  width: 1000,
  height: 300, // set editor height
  minHeight: null, // set minimum height of editor
  maxHeight: null, // set maximum height of editor
  focus: true, // set focus to editable area after initializing
  lang: 'ko-KR',
  callbacks: {
    onImageUpload: function(files, editor, welEditable) {
      sendFile(files[0], this);
    },
  },
  toolbar: [
    ['style', ['bold', 'italic', 'underline', 'clear']],
    ['font', ['strikethrough', 'superscript', 'subscript']],
    ['fontsize', ['fontsize']],
    ['color', ['color']], ['para', ['ul', 'ol', 'paragraph']],
    ['height', ['height']],
    ['insert', ['picture']]
  ]
});

```

```
function sendFile(file, editor) {
    formdata = new FormData();
    formdata.append("userImage", file);

    $.ajax({
        data: formdata,
        type: "POST",
        url: '${initParam.IMG_SERVER_PATH}/upload',
        cache: false,
        contentType: false,
        processData: false,
        success: function(data) {
            console.log(data);
            var url = '${initParam.IMG_SERVER_PATH }/image/'+data.filename;

            if($('#pic1').val() == ''){
                $('#pic1').val(url);
            }

            alert(url);
            $('#hello').html(url);
            $(editor).summernote('editor.insertImage', url);
            $('#imageDiv > ul').append('<li></li>');
        }
    });
}
```

web.xml 에 설정된 이미지 서버의 도메인과 경로, 파일이름을 합쳐 이미지 경로를 만들어 summer note에 추가한다.

summer note 편집기에 사진을 올리면, ajax함수가 동작하여 미리 제작한 Node.js 이미지 서버에 사진을 올리고 사진 파일명을 반환한다.