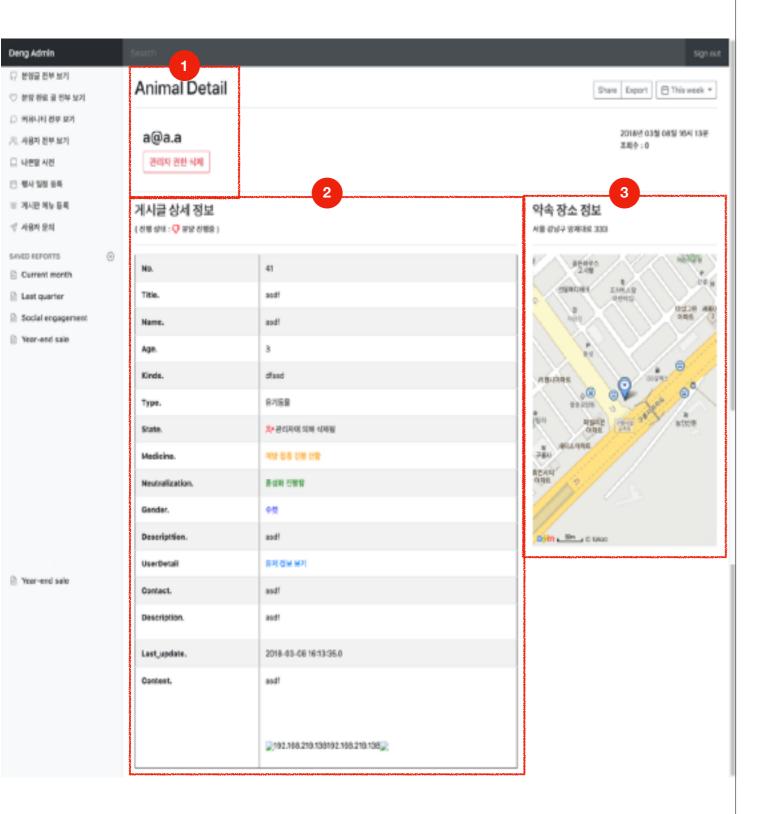
## Deng Admin 개발 기능 (상세)

- 상세 페이지



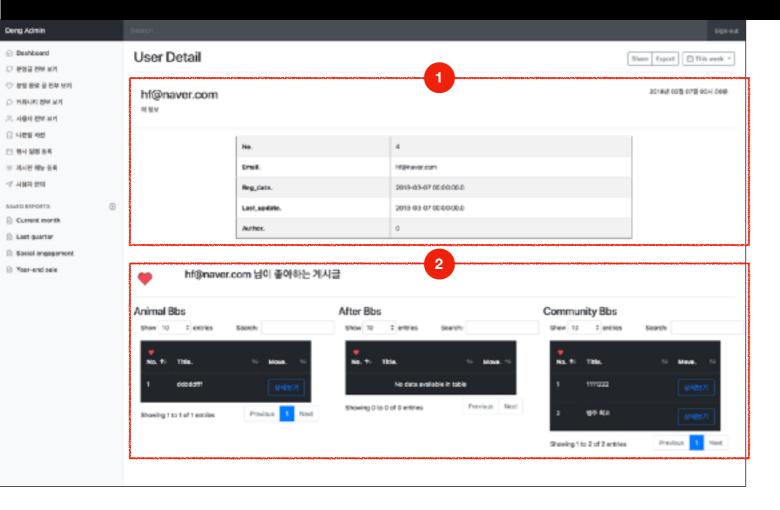
- ajax를 통해 해당 게시글을 상태를 **관리자 권한에 의한 삭제**로 변경한다.
- 2 게시글에 대한 정보를 표시하고, **상태를 표시한다**.

```
function drawMap(addr) {
   var mapContainer = document.getElementById('map');
   mcpOption = {
       center: new daum.maps.LatLng(33.450701, 126.570667), // 지도의 중심라트
       level: 3 // 지도의 확대 레벨
   // 지도를 생성합니다
   var map = new daum.maps.Map(mapContainer, mapOption);
   // 주소-좌표 변환 객체를 생성합니다
    var geocoder = new daum.maps.services.Geocoder();
   // 주소로 작표를 검색합니다
    geocoder.addressSearch(addr, function(result, status) {
       // 정상적으로 검색이 완료됐으면
        if (status === daum.maps.services.Status.OK) {
           var coords = new daum.maps.latlng(result[0].y, result[0].x);
           // 결과값으로 받은 위치를 마커로 표시합니다
           var marker = new daum.maps.Marker({
              position: coords
           E):
           // 인포윈도우로 장소에 대한 설명을 표시합니다
           var infowindow = new daum.maps.InfoWindow({
              content: '<div style="width:150px;text-align:center;padding:6px 0;">hello</div>'
           infowindow.open(map, marker);
           // 지도의 중심을 결과값으로 받은 위치로 이동시킵니다
           map.setCenter(coords);
   3);
```

페이지가 로드할 때, drawMap() 함수를 통해 분양글을 등록할 때 작성했던 주소를 좌표로 변환하고 Daum 지도로 변환하여 반환한다.

## Deng Admin 개발 기능 (상세)

- 유저 상세 페이지



- 1 유저에 대한 정보를 표시한다.
- 2 -1 페이지를 로드할 때, ajax를 통한 함수 동작

```
function drawTable() {
    $.ajax({
        url : 'AdminControl',
        method : 'POST',
        data : { command : 'getLikedTable', seq : ${userDto.seq} },
        success : function (data) {
            var datas = JSON.parse(data);

            var animal = datas[0];
            var after = datas[1];
            var commu = datas[2];

            drawTableWithData(animal, 1);
            drawTableWithData(after, 2);
            drawTableWithData(commu, 3);
        }
    }
}
```

- 2 -2 각 게시글 List를 <u>List<Object> 형태</u>의 하나의 리스트에 담는다. (List<List<Object>> 형태)
- public List<Object> getLikedBbs(int seq) {
   List<Object> list = new ArrayList<>();

   List<AfterBbsDto> afters = bbsDao.getLikeAfterdBbs(seq);
   List<AnimalBbsDto> animals = bbsDao.getLikedAnimalBbs(seq);
   List<CommuBbsDto> commus = bbsDao.getLikedCommuBbs(seq);

   list.add(afters);
   list.add(animals);
   list.add(commus);

   return list;
  }

2 -3 리스트를 파싱하여 javascript, jQuery를 사용하여 뿌려준다.

```
var result = [];
for(var i=0; i<datas.length; i++){
    var temp = [];
    temp[0] = i+1;
    temp[1] = datas[i].title;
    temp[2] = '<a class="btn btn-outline-primary" href="'+url+datas[i].seq+'">상세보기</a>'
    result[i] = temp;
}

$('#'+table).DataTable().destroy();
$('#'+table).DataTable({
        "order": [[ 0, "asc" ]],
        "columns": [{width: '5%'},{width: '80%'},{width: '15%'}],
        "data": result
});
```