



심용기 정유진

FloBot

*Flabot*

# 목차

1. 기획의도 및 배경
2. 목표
3. 기대효과
4. 제안의 특징점
5. 주요기능
6. 개발방법론
7. 시스템 아키텍처
8. 딥러닝 모델 흐름도
9. 벤치마킹 사이트
10. 마인드맵
11. 요구사항분석

12. DB설계
  - 엔티티타입 정의서
  - 속성정의서
  - ERD
  - 테이블정의서
13. 다이어그램
  - 활동다이어그램
  - 시퀀스다이어그램
14. 애플리케이션 아키텍처
15. 기능명세서(FBS)
16. 스토리보드
17. 소스코드
18. 일정 (WBS)
19. 조직도(WBS)
20. 사용된 프로그램



# 1. 기획의도

논스톱  
서비스

궁금했던 꽃 이름부터 구매까지 한번에!

Easy  
구매  
Busy  
판매

소비자와 판매자가 한자리에서 쉽게!

# 1. 기획의도 및 배경

추천상품  
서비스

어떤 꽃이 예쁘지? FloBot이 알려주지!

추천상품  
서비스

어떤 꽃이 잘팔리지? FloBot이 알려주지!

## 2. 목표

### 이지 구매시스템

이름을 몰라 구매하기 어려웠던  
꽃부터 뭘 사야할지 몰라 했던  
고민들까지 없애주는 시스템

### 비지 판매시스템

멀리 있는 소비자를 가깝게,  
잘 팔리는 화훼 추천까지 판매자를  
바쁘게 만들어준 시스템

**이지&비지  
원원 시스템**

### 3. 기대효과



#### 구매자와 판매자의 쉬운 연결

- 판매자와 구매자의 물리적거리를 좁혀주는 인터넷 쇼핑몰 시스템으로, 시간적, 정신적 여유가 없던 구매자들에게 손쉬운 화훼구매가 가능.



#### 업무 효율성 증진

- 유통기한 및 보관기간이 짧은 화훼의 특성상 재고부족이나 재고처리가 발생하기 마련인데, 딥러닝을 통한 제품수량 측정 및 추천으로 재고 효율성 증진 및 매출데이터 분석을 통한 업무효율성 증진



#### 구매자의 편의성 확대

- 사진을 통한 모르는 꽃 이름 검색기능부터 회원정보를 바탕으로한 상품추천 기능까지 딥러닝을 통해 구매자가 구매까지 걸리는 시간을 단축시켜 판매량 증가





## 4. 제안의 특징점



업무프로세스 자동화



딥러닝



업무의 효율성 증대

RPA(Robotic Process Automation)  
를 이용하여 입점 신청업체의 정보체크,  
꽃 데이터 가져오기와 같은  
단순 반복 노동의 자동화를  
통한 근무시간 확보



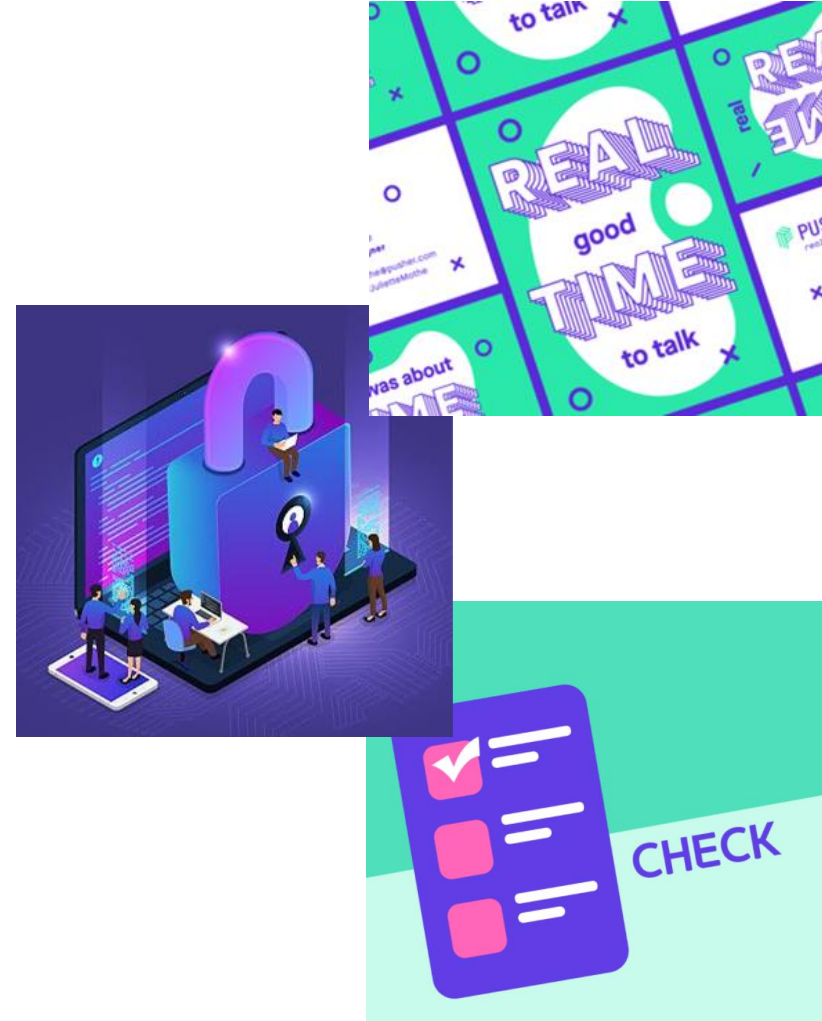
딥러닝을 통한 꽃 정보 검색,  
회원 니즈에 따른 상품 추천,  
고부가가치 재배상품 예상 및 추천으로  
회원과 판매자 모두 만족하는 맞춤형  
서비스 제공



단순 노동 시간 감소 및  
딥러닝을 이용한 데이터  
분석으로 불필요한 노동으로  
인한 시간낭비를 줄여  
근무의 효율성 확보

## 5. 주요기능

1. BcryptPasswordEncoder를 사용한 암호화
2. HandlerInterceptorAdapter를 활용한 보안
3. Validation을 이용한 유효성 검사
4. RPA를 활용한 사무업무 자동화
5. Ajax를 사용한 비동기식 페이지 전환
6. Email과 SMS를 활용한 인증서비스
7. 딥러닝을 통한 데이터분석 및 추천





# 6. 개발방법론

## 1.현행시스템 분석

- 현행시스템  
지원기능분석
- 시스템에 대한  
요구사항 분석
- 개발범위확정
- 개발계획

## 2.목표 업무 분석

- 목표기능  
요구사항 분석 및  
정의
- 개발환경 구성
- 기능의 상세  
요구사항 분석
- 개발범위의  
상세함의 도출

## 3.목표 업무 설계

- 목표기능 모델링
- 목표기능 기능설계
- 개발목표업무에  
대한 기능모델링 및  
기술설계
- ERD, DB Schema

## 4.목표 업무 개발

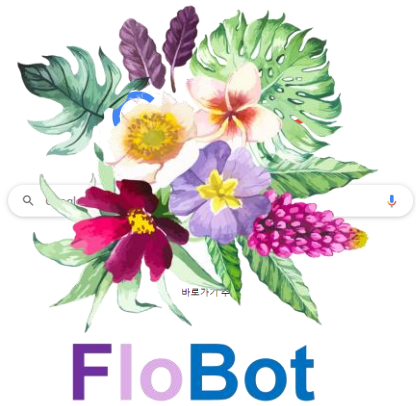
- 코딩
- 단위 통합  
인수테스트
- 데이터 전환단위  
테스트
- 통합테스트
- 성능테스트
- 테스트 결과 보고

## 5.목표 업무 이행

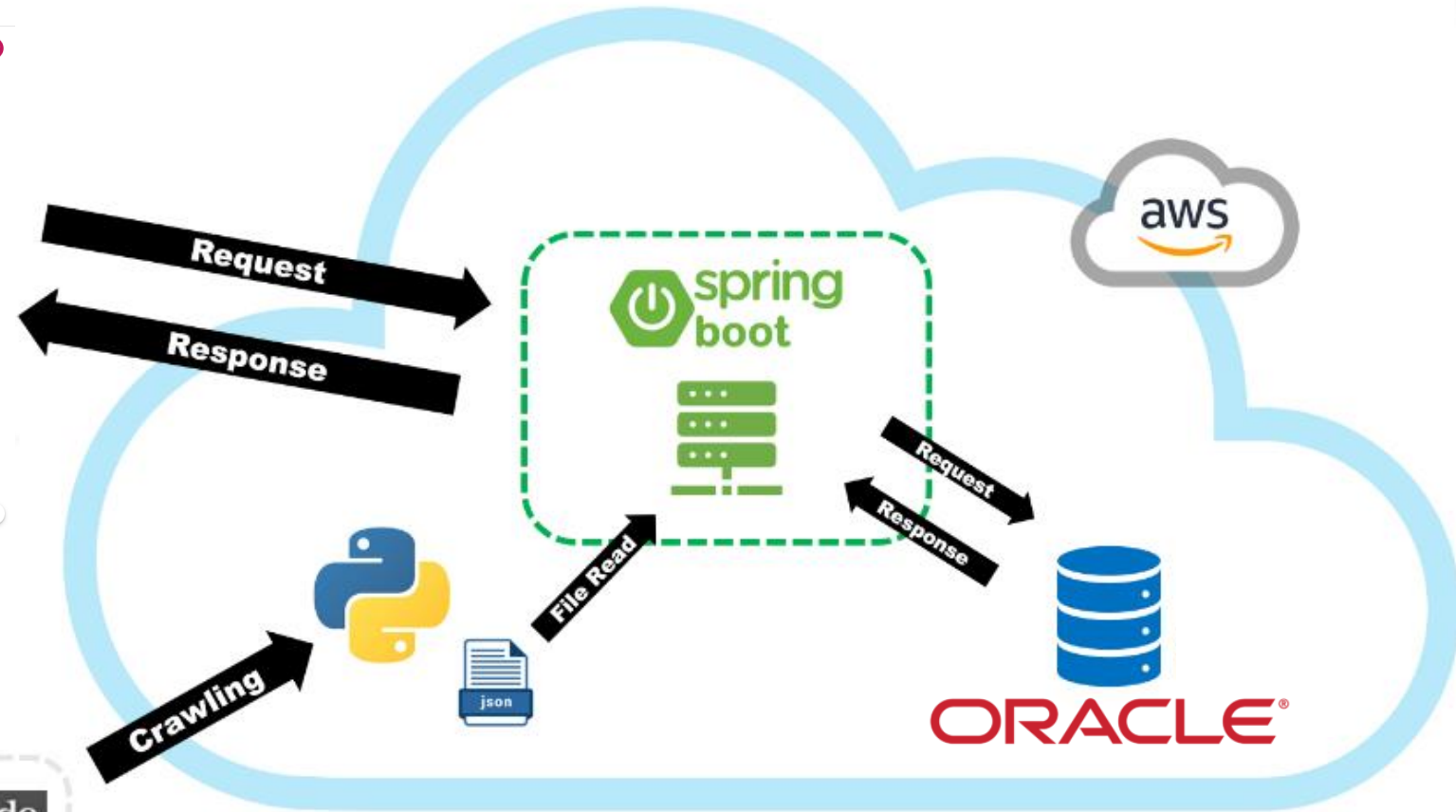
- 시범운영
- 시범운영 결과보고
- 교육실시 및  
시범운영 실시
- 완료 보고서



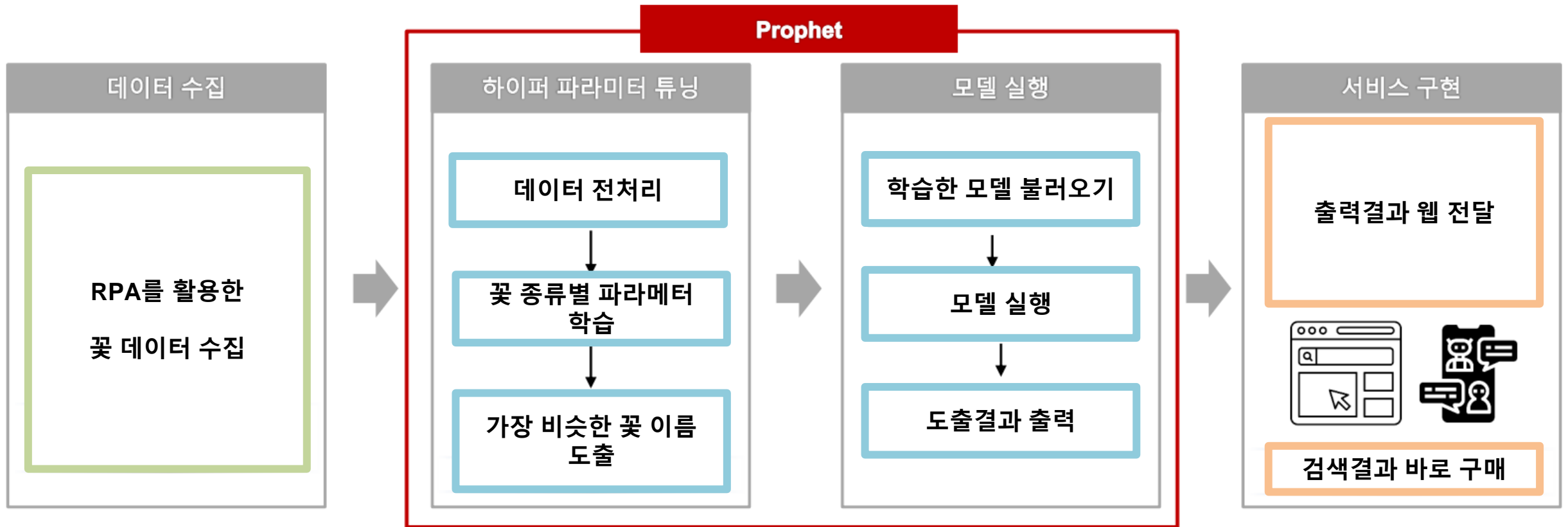
# 7. 시스템 아키텍처



Chrome 맞춤설정



## 8. 딥러닝 모델 흐름도



## 9. 벤치마킹 사이트

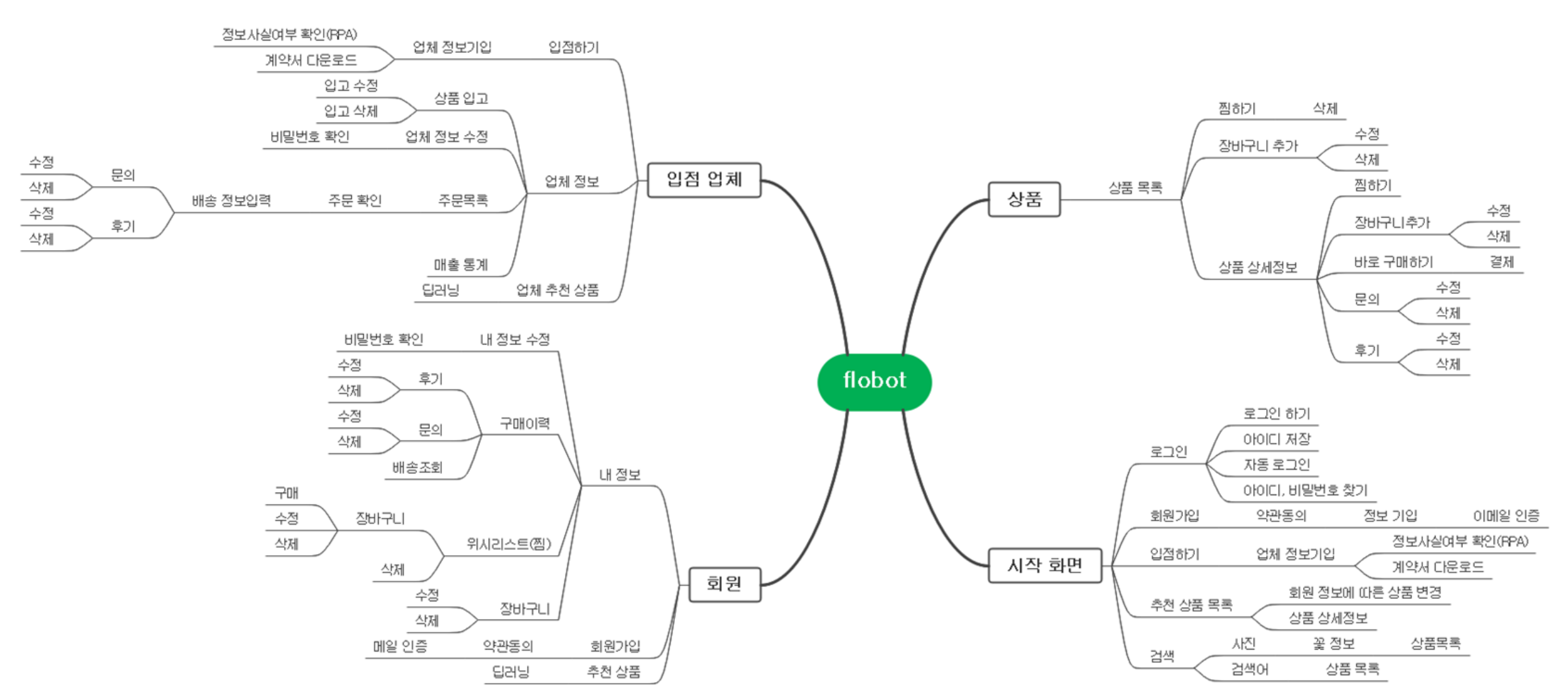
옥션 : <http://www.auction.co.kr/>



꽃핀들플라워 : <https://www.xn--js0bu0i68b9wzdzq4ia.com/>



## 10. 마인드 맵



# 11. 요구사항 분석

구분		내용
메인페이지		<ul style="list-style-type: none"><li>• 메뉴(꽃 종류: 꽃, 허브, 나무, 다육, 씨앗, 모종, 기타)</li><li>• 기타 : 화분, 모종삽</li><li>• 광고 상품 리스트 출력 3 x 2</li><li>• 문의할 쇼핑몰 전화번호, 이메일</li><li>• 입점하기 버튼</li></ul>
메인페이지	로그인	<ul style="list-style-type: none"><li>• 아이디, 비밀번호 입력, 확인버튼</li><li>• 아이디, 비밀번호 찾기</li><li>• 자동로그인, 아이디 저장(30일 저장)</li><li>• 회원가입 버튼</li><li>• 성공적인 로그인 : 로그인 후로 이동</li><li>• 로그인 오류 : '아이디 또는 비밀번호가 일치하지 않습니다' 문구 출력</li><li>• 이메일 미인증시 : 아이디 밑에 '이메일을 확인하세요' 문구 출력</li></ul>



# 11. 요구사항 분석

구분		내용
메인페이지	로그인 후	<ul style="list-style-type: none"><li>• 상품추천기능</li><li>• 메인 페이지에 광고 상품대신 회원 별(회원 정보에 따른) 상품추천 3 x 2</li><li>• 회원 정보에는 구매이력, 나이, 성별, 기혼여부 등</li></ul>
검색	텍스트	<ul style="list-style-type: none"><li>• {상품명}{사진}{돋보기버튼}</li><li>• 상품명</li><li>• 상품명 검색 -&gt; 상품 목록으로 이동</li></ul>
검색	사진	<ul style="list-style-type: none"><li>• 사진</li><li>• 이미지, 이름, 설명, 상품 목록으로 이동 버튼</li></ul>

# 11. 요구사항 분석

구분		내용
메인 페이지	상품 목록	<ul style="list-style-type: none"><li>• 입점 업체 점수에 따라 내림차순으로 상품 목록 출력</li><li>• 1페이지에 상품 5개씩</li><li>• 하단에 페이지 버튼 10개씩 출력</li><li>• 상품이미지, 상품명, 가격, 후기 개수, 구매 개수, 후기 평균 별점, 배송비, 업체명, 찜하기, 장바구니 추가</li><li>• 이미지, 상품명 클릭시 상세 페이지로 이동</li><li>• 입고된 상품이 10개 이하일 경우 이미지 대신 매진임박 글씨 출력</li></ul>
메인 페이지	상품 상세 페이지	<ul style="list-style-type: none"><li>• 이미지, 상품명, 후기 개수, 구매 개수, 후기 평균 별점, 배송비, 업체명, 찜하기, 장바구니 추가, 구매하기</li><li>• 상세설명, 구매후기(개수), 상품문의(개수)</li></ul>

# 11. 요구사항 분석

구분		내용
입점 업체 관리		<ul style="list-style-type: none"> <li>• 입점 업체와는 별도의 계약을 맺고 있으며 계약을 맺지 않은 입점 업체와는 거래 불가</li> <li>• 한 번 계약을 맺었다가 해지할 수 있으며, 다시 계약을 맺을 수도 있다. 이때 이전 계약상태에 대한 정보도 필요하다. 계약의 갱신은 여러 번 가능하다.</li> </ul>
입점 업체	회원가입 (입점하기)	<ul style="list-style-type: none"> <li>• 사업자 등록번호, 업체명, 업체구분, 업체주소, 업체 연락처, 업체 대표명, 업체 메일주소, 사업자 등록한 날짜를 기입 -&gt; 계약서 양식 전송(이메일) 수동 -&gt; 계약서 작성해서 보내주면 flobot이 검토(RPA)</li> <li>• 검토 내용 사업자 등록번호, 업체명, 업체구분 사실여부-&gt; 검토후 승인</li> <li>• 승인 후 DB에 업로드(대기는 나중에)</li> </ul>
입점 업체	내정보 (업체 전용)	<ul style="list-style-type: none"> <li>• 등록한 상품 목록, 통계, 업체정보 수정, 주문목록, 입고</li> </ul>
입점 업체	주문 목록 (업체 전용)	<ul style="list-style-type: none"> <li>• 상품명, 이미지, 가격, 개수, 총 가격, 아이디, 결제일</li> <li>• 확인버튼 -&gt; (구매이력)상품준비중으로 바뀜</li> <li>• 배송업체, 송장번호 입력 -&gt; (구매이력)배송중</li> </ul>

# 11. 요구사항 분석

구분		내용
업체 정보	입고(업체 전용)	<ul style="list-style-type: none"> <li>• 상품명, 이미지, 상품가격, 배송비, 개수, 상품설명(필수)</li> <li>• 필수사항 미기입시 : {해당사항을}입력하세요 문구 출력</li> </ul>
업체 정보	등록한 상품 목록	<ul style="list-style-type: none"> <li>• 상품명, 이미지, 재고수량, 판매 개수, 판매액 , 매출액</li> <li>• 매출액 그래프(일, 분기 선택)</li> <li>• 수정, 삭제</li> </ul>
업체 정보	통계	<ul style="list-style-type: none"> <li>• 업체에 상품 추천 출력</li> <li>• 기준 : 판매액</li> </ul>
업체 정보	업체 정보수정	<ul style="list-style-type: none"> <li>• 비밀번호 확인후 수정페이지 이동가능</li> <li>• 업체명, 업체주소, 이메일 주소, 비밀번호, 업체 연락처, 대표명, 변경 가능</li> </ul>

# 11. 요구사항 분석

구분		내용
회원가입		<ul style="list-style-type: none"> <li>flobot에서 상품을 구입하려면 반드시 이 회사의 회원으로 등록해야 구매가 가능하다.</li> <li>flobot의 회원 등록 정보에는 회원번호, 주소, 이메일 주소, 아이디, 비밀번호, 휴대폰번호, 생년월일, 성별(필수),</li> <li>기혼여부(선택)이 있다.</li> <li>필수란 미가입시 : '다시 한번 확인해 주십시오'가 나타난다.</li> </ul>
회원가입	약관동의	<ul style="list-style-type: none"> <li>회원가입 버튼 -&gt; 약관 동의 페이지</li> <li>개인정보 활용 동의, 쇼핑몰 약관(라디오)</li> <li>동의, 동의하지 않음</li> <li>동의 하지 않음을 선택할 시 경고창: 미동의 시 회원가입이 불가능 합니다. 경고창 확인버튼 - 약관동의 페이지</li> <li>동의 선택 시 : 회원가입 폼으로 넘어감</li> </ul>
회원가입	완료	<ul style="list-style-type: none"> <li>성공적인 회원가입</li> <li>회원가입 화면의 하단에 있는 등록 버튼</li> <li>회원가입 완료 페이지 : '{아이디}님 {이메일}에서 가입확인 버튼을 누르고 회원가입을 완료하세요 '</li> </ul>

# 11. 요구사항 분석

구분		내용
내정보		<ul style="list-style-type: none"><li>구매이력, 위시리스트(찜), 장바구니 이동가능, 내 정보수정</li></ul>
내정보	구매이력	<ul style="list-style-type: none"><li>상품명, 상품이미지, 총 가격, 개수, 결제일, 문의 하기</li><li>결제상태(결제전, 결제완료, 상품준비중, 배송중(송장등록완료), 배송완료, 구매결정-&gt;후기등록가능)</li></ul>
내정보	내 정보수정	<ul style="list-style-type: none"><li>비밀번호 확인후 수정페이지 이동가능</li><li>주소, 이메일 주소, 비밀번호, 휴대폰번호, 생년월일, 성별(필수),</li><li>기혼여부(선택) 변경 가능</li></ul>



# 11. 요구사항 분석

구분	내용
위시리스트(찜하기)	<ul style="list-style-type: none"> <li>• 상품명 옆에 위치한 빈 하트를 누르면 빨간 하트로 변하고 로그인 한 회원의 위시리스트에 품목이 추가</li> <li>• 하트를 한번 더 누르면 빈 하트로 변하고 상품 삭제 가능</li> <li>• 추가된 상품은 위시리스트페이지에서 확인 가능</li> <li>• 위시리스트 페이지에서 리스트에 있는 상품을 선택하여 바로 구매 혹은 장바구니로 이동</li> <li>• 위시리스트 페이지에서 상품 삭제가능(개별, 전체 삭제가능)</li> </ul>
장바구니	<ul style="list-style-type: none"> <li>• 상품 수량 수정</li> <li>• 수량 1개 일 때 : '-' 클릭시 '장바구니에서 삭제하려면 삭제버튼을 누르세요' 문구 출력</li> <li>• 상품 삭제</li> <li>• 상품 선택(전체 선택, 개별선택)</li> <li>• 선택된 상품 총 가격</li> <li>• 구매 페이지 이동</li> </ul>
구매	<ul style="list-style-type: none"> <li>• 배송지, 수령인, 연락처, 배송메모, 상품명, 수량, 상품이미지, 상품금액, 배송비, 합계 금액, 결제하기</li> </ul>
결제하기	<ul style="list-style-type: none"> <li>• 이니시스</li> </ul>

# 11. 요구사항 분석

구분		내용
후기		<ul style="list-style-type: none"> <li>1페이지에 상품 5개씩</li> <li>하단에 페이지 버튼 10개씩 출력</li> <li>별점, 내용(필수, 최소 글자수제한 : 10자), 사진(선택)</li> <li>아이디, 등록일(자동 입력)</li> <li>평균 별점</li> <li>입력</li> <li>수정, 삭제</li> </ul>
문의		<ul style="list-style-type: none"> <li>1페이지에 상품 5개씩</li> <li>하단에 페이지 버튼 10개씩 출력</li> <li>카테고리(배송, 반품/환불, 상품설명, 기타)</li> <li>제목, 내용 (필수)</li> <li>아이디, 등록일(자동 입력)</li> <li>답변상태</li> </ul>
문의	답변하기	<ul style="list-style-type: none"> <li>문의 내용 밑에 답변 내용 추가</li> </ul>

# 12. DB 설계

엔티티타입 정의서

엔티티타입명	엔티티타입 설명	관련속성
회원	flobot에서 상품을 구매하는 회원에 대한 정보, 환불, 후기작성 등 flobot 이용의 전반적인 프로세스에서 활용된다. 또 상품 구매시 배송을 위해서 사용된다.	회원번호, 회원이름, 회원 아이디...
업체	flobot에 입점한 업체에 대한 정보를 확인하고 매출통계 서비스제공을 위해 사용한다.	사업자 등록번호, 업체명, 업체 아이디...
꽃	flobot의 데이터베이스에 입력되어있는 꽃의 이름, 종류, 설명, 이미지에 대한 정보를 가지고 있다.	상품번호 ...
상품 정보	flobot에서 판매하는 상품에 대한 정보와 제조사에 대한 정보를 포함하고 있다.	상품번호, 상품명, 제조사...
상품 문의	회원과 상품 정보를 포함 하고 있으며 판매 상품에 관한 문의 사항을 포함하고 답변하는 업체의 정보를 포함하고 있다.	회원번호, 상품번호, 문의 제목...
관심상품	회원이 마음에 드는 상품을 모아놓은 정보	회원번호, 상품번호...
장바구니	회원이 구매를 원하는 상품을 모아놓은 정보	회원번호, 상품번호...

Flobot

# 12. DB 설계

엔티티타입 정의서

엔티티타입명	엔티티타입 설명	관련속성
구매	회원이 구매한 상품내역을 확인하기 위한 정보, 상품을 결제하기 위한 정보, 구매시 등록한 회원정보, 판매된 상품의 정보를 포함하고 있다	구매번호, 구매일, 구매가격...
결제	회원이 결제를 하기위해서 필요한 정보, 상품별 가격과 배송비 정산을 위한 필요한 정보를 포함하고 있다.	구매번호, 결제방법...
대금	업체에 결제대금을 지급해주기 위한 총 대금과 대금 정산 및 입금을 위한 업체정보, 주문정보를 포함하고 있다.	사업자등록번호, 구매번호 ...
배송	회원이 구매한 상품을 전달하기 위해 필요한 정보	구매번호, 송장번호...
후기	상품을 구매한 회원이 상품에 대한 후기를 남기기위해 필요한 정보와 상품에 따라 회원이 점수를 부여한 값에 대한 정보를 포함하고 있다.	회원번호, 상품번호, 문의 제목...
주문 상품 리스트	구매 한 상품에 대한 정보	구매번호, 상품번호...
상품입고	flobot에 판매하는 상품 정보, 입고 현황을 파악하기 위한 정보를 포함하고 있다.	상품번호, 입고번호...
통계	Flobot에 입점한 업체의 매출과 영업이익 등을 기록하고 분석하기 위한 정보를 포함하고 있다.	사업자등록번호, 통계번호

# 12. DB 설계

속성 정의서

엔티티타입명	속성명	내용
회원	회원번호	회원관리를 위한 고유번호
	회원아이디	회원을 구분하는 고유 아이디
	회원비밀번호	로그인시 필요한 암호화된 번호
	회원이름	회원의 이름
	회원주소	회원의 주소
	회원상세주소	회원의 상세주소
	우편번호	회원주소의 우편번호
	등록일	회원 가입일
	성별	회원의 성별
	연락처	회원의 연락처
	이메일	회원의 이메일
	생년월일	회원의 생년월일
	결혼여부	회원의 결혼여부
	이메일확인	회원의 이메일인증 여부
	관리자	관리자 아이디 관리를 위한 속성

Flabot

# 12. DB 설계

속성 정의서

엔티티타입명	속성명	내용
업체	사업자등록번호	업체의 사업자등록번호
	업체아이디	업체를 구분하는 고유 아이디
	업체비밀번호	로그인시 필요한 암호화된 번호
	업체명	업체의 이름
	업체주소	업체의 주소
	업체상세주소	업체의 상세주소
	우편번호	업체주소의 우편번호
	입점일	업체 가입일
	업체연락처	업체의 연락처
	업체이메일	업체의 이메일
	사업자등록일	업체의 사업자등록일
	계좌번호	업체의 대금을 결제 받을 계좌번호
	거래은행	업체의 대금을 결제 받을 거래은행
	업종	업체의 업종

Flabot



# 12. DB 설계

속성 정의서

엔티티타입명	속성명	내용
상품정보	상품번호	상품을 구분하고 관리하기 위한 고유 번호
	상품명	상품의 이름
	상품내용	상품의 상세설명
	대문이미지	상품의 대표 이미지의 변환파일 이름
	대문원본명	상품의 대표 이미지의 원본파일 이름
	이미지	상품의 상세설명의 변환파일 이름
	이미지원본명	상품의 상세설명의 원본파일 이름
	배송비	상품의 배송비
	제조사	상품의 제조사
	조회수	상품에 대한 회원들의 조회 수
	상품종류	상품의 카테고리
배송	송장번호	입점 업체가 등록한 배송 송장번호
	배송업체	입점 업체가 등록한 배송업체명

# 12. DB 설계

속성 정의서

엔티티타입명	속성명	내용
상품 입고	입고번호	상품의 입고를 구분하고 관리하기 위한 고유 번호
	수량	입고된 상품의 수량
	입고일자	상품이 입고된 일자
	제조일	상품이 제조된 일자
	가격	상품의 가격
상품 문의	문의번호	문의를 구분하고 관리하기 위한 고유 번호
	문의제목	사용자가 입력한 문의의 제목
	문의종류	문의의 구분
	문의내용	사용자가 입력한 문의의 내용
	문의답글	입점 업체가 입력한 문의의 답변
	문의등록일	사용자가 입력한 문의의 등록일
	답변등록일	입점 업체가 입력한 문의의 답변 등록일

## 12. DB 설계

속성 정의서

엔티티타입명	속성명	내용
구매	구매번호	구매정보를 구분하고 관리하기 위한 고유 번호
	구매일	상품을 구매한 일자
	구매자	상품 구매자가 입력한 수취인
	연락처	상품 구매자가 입력한 수취인의 연락처
	주소	상품 구매자가 입력한 수취인의 주소
	상세주소	상품 구매자가 입력한 수취인의 상세주소
	우편번호	상품 구매자가 입력한 수취인의 우편번호
	요청사항	상품 구매자가 입력한 요청사항
	구매가격	총 상품가격과 총 배송비를 더한 총 구매 가격
주문상품리스트	수량	구매하는 상품의 수량
	공급가액	구매하는 상품의 개별 가격
	주문상태	상품의 주문상태 : 결제 중, 결제완료, 배송준비 중, 배송 중, 배송완료, 구매완료

# 12. DB 설계

속성 정의서

엔티티타입명	속성명	내용
결제	승인번호	이니시스에서 발급한 결제 승인 번호
	카드번호	회원이 등록한 카드번호
	카드코드	회원이 등록한 카드의 코드
	TID	이니시스에서 제공하는 결제 아이디
	총가격	총 결제 가격
	결과문자	결제의 결과 메시지
	결제방식	결제의 방식
	결제날짜	결제의 일자
	결제시간	결제의 시간
후기	후기번호	후기를 구분하고 관리하기 위한 고유 번호
	후기등록일	회원이 후기를 등록한 일자
	후기내용	회원이 작성한 후기의 내용
	별점	회원이 지정한 상품의 별점

Flabot

# 12. DB 설계

속성 정의서

엔티티타입명	속성명	내용
대금	납입번호	대금을 구분하고 관리하기 위한 고유 번호
	납입액	입점 업체에 납입 해야하는 금액
	납입일	입점 업체에 납입한 일자
꽃	꽃번호	꽃을 구분하고 관리하기 위한 고유 번호
	꽃이름	꽃의 이름
	꽃종류	꽃의 종류
	꽃설명	꽃에 대한 설명
	꽃이미지	꽃의 이미지 파일 이름
장바구니	수량	회원이 장바구니에 추가한 상품의 수량
관심상품	등록일	회원이 관심상품에 상품을 등록한 일자

# 12. DB 설계

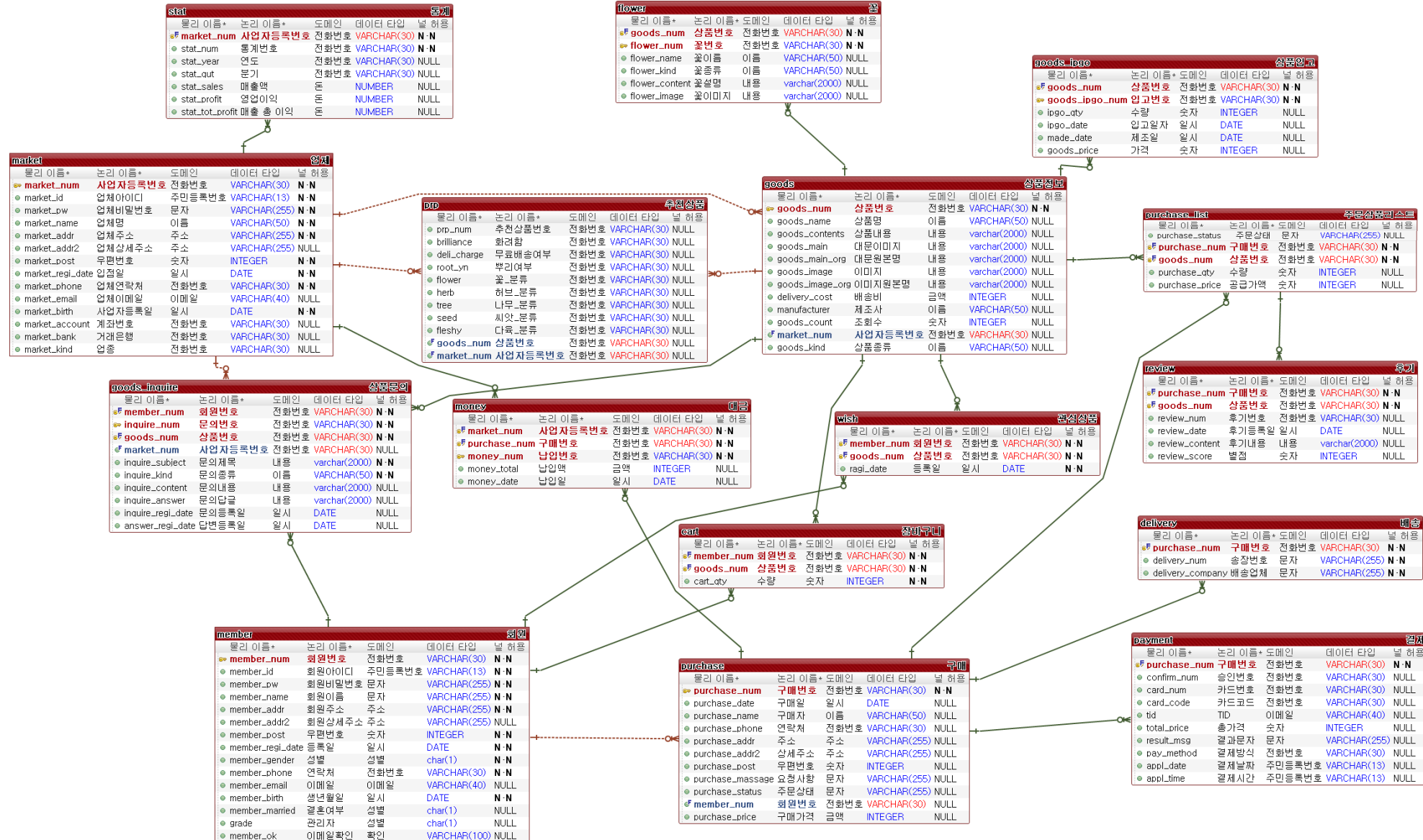
속성 정의서

엔티티타입명	속성명	내용
통계	마켓번호	업체관리를 위한 고유번호
	연도	연도
	분기	분기
	매출액	업체의 매출액
	영업이익	업체의 영업이익
	매출 총 이익	업체의 매출 총 이익
추천상품	추천상품 번호	추천상품 가중치를 관리하기위한 번호
	화려함	화려함에 대한 선호도
	무료배송여부	무료배송 여부 또는 선호도
	뿌리여부	뿌리의 여부 또는 선호도



# 12. DB 설계

ERD



Flabot

# 12. DB 설계

테이블정의서

엔티티타입명	회원						
테이블명	member						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
1	member_num	회원번호	전화번호	VARCHAR	30	NOT NULL	PK
2	member_id	회원아이디	주민등록번호	VARCHAR	13	NOT NULL	
3	member_pw	회원비밀번호	문자	VARCHAR	255	NOT NULL	
4	member_name	회원이름	문자	VARCHAR	255	NOT NULL	
5	member_addr	회원주소	주소	VARCHAR	255	NOT NULL	
6	member_addr2	회원상세주소	주소	VARCHAR	255	NULL	
7	member_post	우편번호	숫자	INTEGER		NOT NULL	
8	member_regi_date	등록일	일시	DATE		NOT NULL	
9	member_gender	성별	성별	char	1	NOT NULL	
10	member_phone	연락처	전화번호	VARCHAR	30	NOT NULL	
11	member_email	이메일	이메일	VARCHAR	40	NULL	
12	member_ok	이메일확인	확인	VARCHAR	100	NULL	

Flabot

# 12. DB 설계

테이블정의서

엔티티타입명	회원						
테이블명	member						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
13	member_birth	생년월일	일시	DATE		NOT NULL	
14	member_married	결혼여부	성별	char	1	NULL	
15	grade	관리자	성별	char	1	NULL	

엔티티타입명	대금						
테이블명	money						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
1	money_num	납입번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
2	purchase_num	구매번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
3	market_num	사업자등록번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
4	money_total	납입액	금액	INTEGER		NULL	
5	money_date	납입일	일시	DATE		NULL	

Flabot

# 12. DB 설계

테이블정의서

엔티티타입명	업체						
테이블명	market						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
1	market_num	사업자등록번호	전화번호	VARCHAR	30	NOT NULL	PK
2	market_id	업체아이디	주민등록번호	VARCHAR	13	NOT NULL	
3	market_pw	업체비밀번호	문자	VARCHAR	255	NOT NULL	
4	market_name	업체명	이름	VARCHAR	50	NOT NULL	
5	market_addr	업체주소	주소	VARCHAR	255	NOT NULL	
6	market_addr2	업체상세주소	주소	VARCHAR	255	NULL	
7	market_post	우편번호	숫자	INTEGER		NOT NULL	
8	market_regi_date	입점일	일시	DATE		NOT NULL	
9	market_phone	업체연락처	전화번호	VARCHAR	30	NOT NULL	
10	market_email	업체이메일	이메일	VARCHAR	40	NULL	
11	market_birth	사업자등록일	일시	DATE		NOT NULL	

Flabet

# 12. DB 설계

테이블정의서

엔티티타입명	업체						
테이블명	market						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
12	market_account	계좌번호	전화번호	VARCHAR	30	NULL	
13	market_bank	거래은행	전화번호	VARCHAR	30	NULL	
14	market_kind	업종	전화번호	VARCHAR	30	NULL	

엔티티타입명	꽃						
테이블명	flower						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
1	goods_num	상품번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
2	flower_num	꽃번호	전화번호	VARCHAR	30	NOT NULL	PK
3	flower_name	꽃이름	이름	VARCHAR	50	NULL	
4	flower_kind	꽃종류	이름	VARCHAR	50	NULL	
5	flower_content	꽃설명	내용	VARCHAR	2000	NULL	
6	flower_image	꽃이미지	내용	VARCHAR	2000	NULL	

Flabot

# 12. DB 설계

테이블정의서

엔티티타입명	상품정보						
테이블명	goods						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
1	goods_num	상품번호	전화번호	VARCHAR	30	NOT NULL	PK
2	goods_name	상품명	이름	VARCHAR	50	NULL	
3	goods_contents	상품내용	내용	VARCHAR	2000	NULL	
4	goods_main	대문이미지	내용	VARCHAR	2000	NULL	
5	goods_main_org	대문원본명	내용	VARCHAR	2000	NULL	
8	goods_image	이미지	내용	VARCHAR	2000	NULL	
6	goods_image_org	이미지원본명	내용	VARCHAR	2000	NULL	
8	delivery_cost	배송비	금액	INTEGER		NULL	
9	manufacturer	제조사	이름	VARCHAR	50	NULL	
10	goods_count	조회수	숫자	INTEGER		NULL	
11	market_num	사업자등록번호	전화번호	VARCHAR	30	NULL	FK
12	goods_kind	상품종류	이름	VARCHAR	50	NULL	

Flabot

# 12. DB 설계

테이블정의서

엔티티타입명	상품입고						
테이블명	goods_ipgo						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
1	goods_num	상품번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
2	goods_ipgo_num	입고번호	전화번호	VARCHAR	30	NOT NULL	PK
3	ipgo_qty	수량	숫자	INTEGER		NULL	
4	ipgo_date	입고일자	일시	DATE		NULL	
5	made_date	제조일	일시	DATE		NULL	
6	goods_price	가격	숫자	INTEGER		NULL	

# 12. DB 설계

테이블정의서

엔티티타입명	상품문의						
테이블명	goods_inquire						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
1	member_num	회원번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
2	inquire_num	문의번호	전화번호	VARCHAR	30	NOT NULL	PK
3	goods_num	상품번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
4	market_num	사업자등록번호	전화번호	VARCHAR	30	NULL	FK
5	inquire_subject	문의제목	내용	VARCHAR	2000	NOT NULL	
6	inquire_kind	문의종류	이름	VARCHAR	50	NOT NULL	
7	inquire_content	문의내용	내용	VARCHAR	2000	NULL	
8	inquire_answer	문의답글	내용	VARCHAR	2000	NULL	
9	inquire_regi_date	문의등록일	일시	DATE		NULL	
10	answer_regi_date	답변등록일	일시	DATE		NULL	



# 12. DB 설계

테이블정의서

엔티티타입명	구매						
테이블명	purchase						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
1	purchase_num	구매번호	전화번호	VARCHAR	30	NOT NULL	PK
2	purchase_date	구매일	일시	DATE		NULL	
3	purchase_name	구매자	이름	VARCHAR	50	NULL	
4	purchase_phone	연락처	전화번호	VARCHAR	30	NULL	
5	purchase_addr	주소	주소	VARCHAR	255	NULL	
6	purchase_addr2	상세주소	주소	VARCHAR	255	NULL	
7	purchase_post	우편번호	숫자	INTEGER		NULL	
8	purchase_message	요청사항	문자	VARCHAR	255	NULL	
9	member_num	회원번호	전화번호	VARCHAR	30	NULL	FK
10	purchase_price	구매가격	금액	INTEGER		NULL	

Flabot

# 12. DB 설계

테이블정의서

엔티티타입명	주문상품리스트						
테이블명	purchase_list						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
1	purchase_num	구매번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
2	goods_num	상품번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
3	purchase_qty	수량	숫자	INTEGER		NULL	
4	purchase_price	공급가액	숫자	INTEGER		NULL	
9	purchase_status	주문상태	문자	VARCHAR	255	NULL	

엔티티타입명	배송						
테이블명	delivery						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
1	purchase_num	구매번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
2	delivery_num	송장번호	문자	VARCHAR	255	NOT NULL	
3	delivery_company	배송업체	문자	VARCHAR	255	NOT NULL	

Flabot

# 12. DB 설계

테이블정의서

엔티티타입명	장바구니						
테이블명	cart						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
1	member_num	회원번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
2	goods_num	상품번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
3	cart_qty	수량	숫자	INTEGER		NOT NULL	

엔티티타입명	관심상품						
테이블명	wish						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
1	member_num	회원번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
2	goods_num	상품번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
3	regi_date	등록일	일시	DATE		NOT NULL	

# 12. DB 설계

테이블정의서

엔티티타입명	결제						
테이블명	payment						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
1	purchase_num	구매번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
2	confirm_num	승인번호	전화번호	VARCHAR	30	NULL	
3	card_num	카드번호	전화번호	VARCHAR	30	NULL	
4	card_code	카드코드	전화번호	VARCHAR	30	NULL	
5	tid	TID	이메일	VARCHAR	40	NULL	
6	total_price	총가격	숫자	INTEGER		NULL	
7	result_msg	결과문자	문자	VARCHAR	255	NULL	
8	pay_method	결제방식	전화번호	VARCHAR	30	NULL	
9	appl_date	결제날짜	주민등록번호	VARCHAR	13	NULL	
10	appl_time	결제시간	주민등록번호	VARCHAR	13	NULL	

Flabot

# 12. DB 설계

테이블정의서

엔티티타입명	후기						
테이블명	review						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
1	purchase_num	구매번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
2	goods_num	상품번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
3	review_num	후기번호	전화번호	VARCHAR	30	NULL	
4	review_date	후기등록일	일시	DATE		NULL	
5	review_content	후기내용	내용	VARCHAR	2000	NULL	
6	review_score	별점	숫자	INTEGER		NULL	

# 12. DB 설계

테이블정의서

엔티티타입명	통계						
테이블명	stat						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
1	market_num	마켓번호	전화번호	VARCHAR	30	NOT NULL	PK, FK
2	stst_num	통계번호	전화번호	VARCHAR	30	NOT NULL	PK
3	stat_year	연도	전화번호	VARCHAR	30	NULL	
4	stat_qut	분기	전화번호	VARCHAR	30	NULL	
5	stat_sales	매출액	금액	NUMBER		NULL	
6	stat_profit	영업이익	금액	NUMBER		NULL	
7	stat_tot_profit	매출 총 이익	금액	NUMBER		NULL	

# 12. DB 설계

테이블정의서

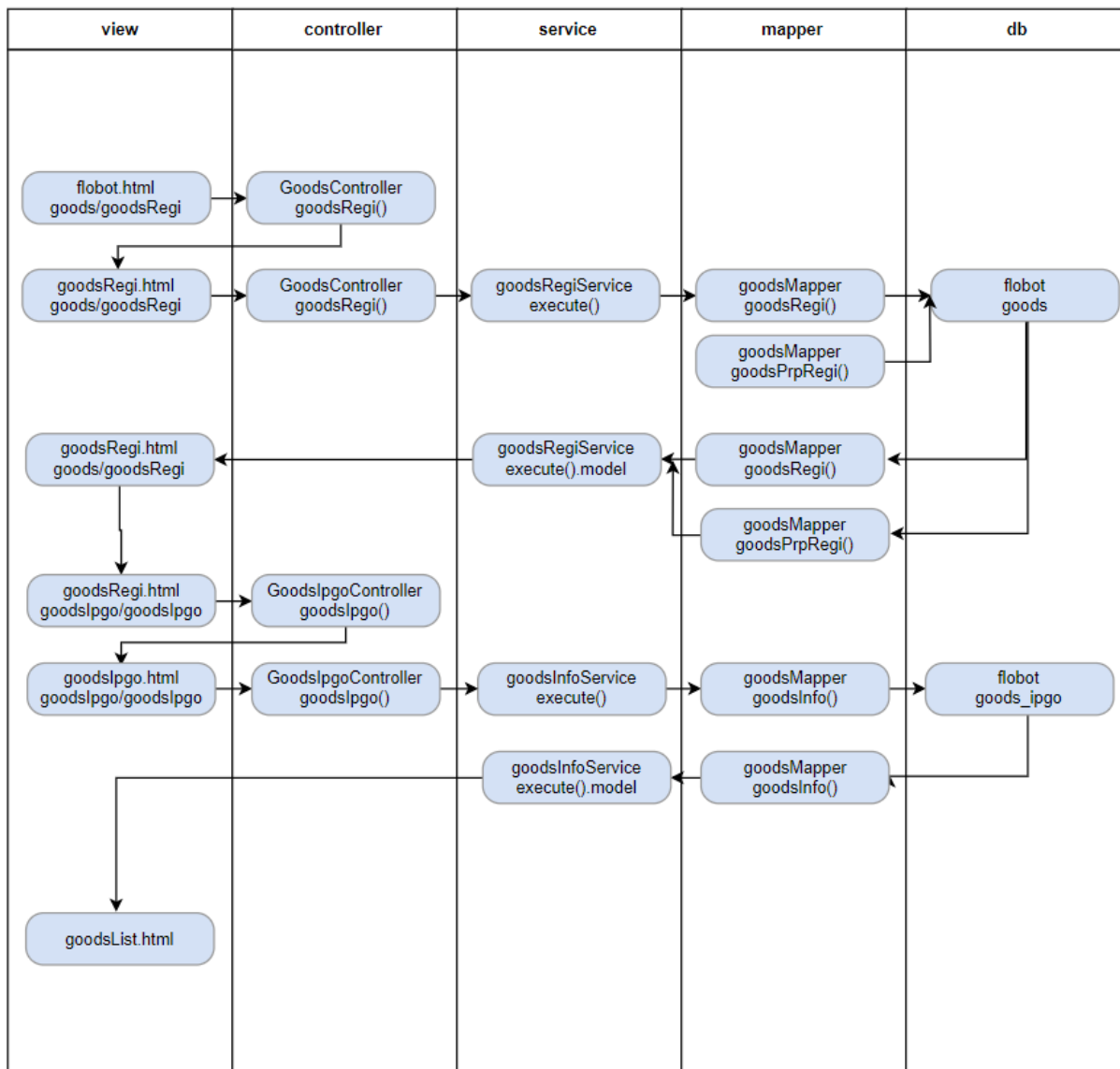
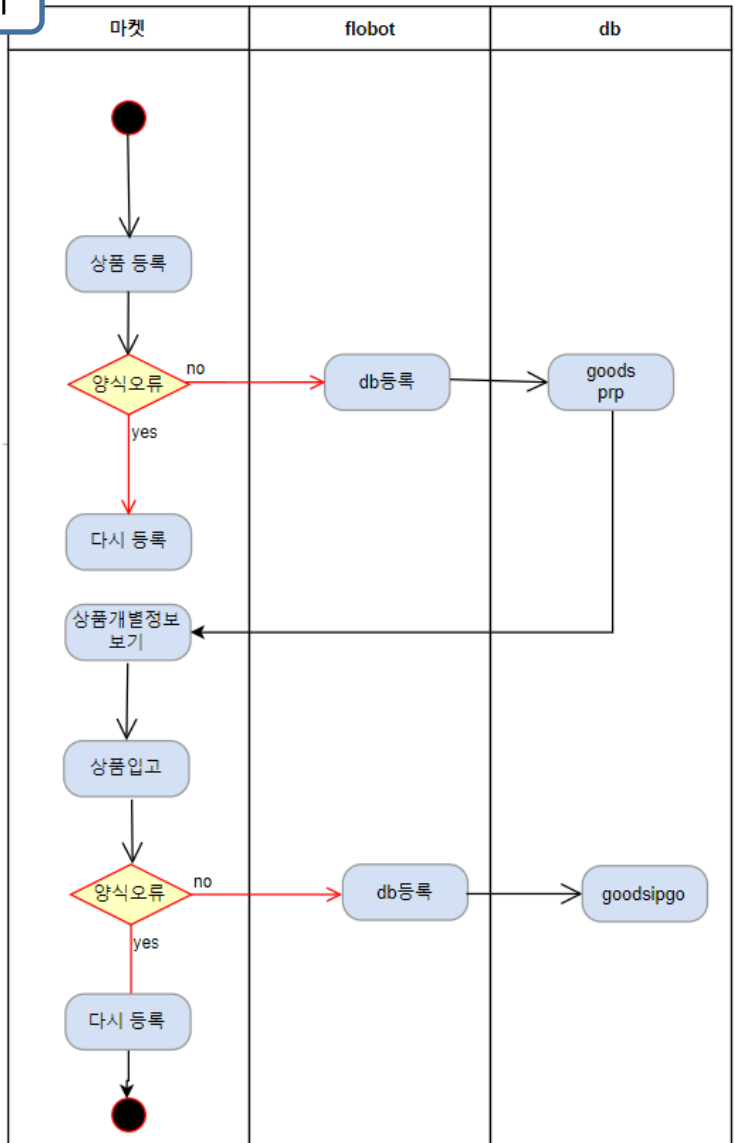
엔티티타입명	통계						
테이블명	stat						
번호	컬럼명	속성명	도메인	데이터타입	길이	NULL 여부	KEY
1	prp_num	마켓번호	전화번호	VARCHAR	30	NOT NULL	PK
2	goods_num	상품번호	전화번호	VARCHAR	30	NULL	FK
3	Market_num	마켓번호	전화번호	VARCHAR	30	NULL	FK
4	brilliance	화려함	전화번호	VARCHAR	30	NULL	
5	deli_charge	무료배송여부	전화번호	VARCHAR	30	NULL	
6	root_yn	뿌리여부	전화번호	VARCHAR	30	NULL	
7	flower	꽃_분류	전화번호	VARCHAR	30	NULL	
8	seed	씨앗_분류	전화번호	VARCHAR	30	NULL	
9	herb	허브_분류	전화번호	VARCHAR	30	NULL	
10	tree	나무_분류	전화번호	VARCHAR	30	NULL	
11	fleshy	다육_분류	전화번호	VARCHAR	30	NULL	

Flabot

# 13. 다이어그램

활동다이어그램 / 시퀀스다이어그램

## 상품관리

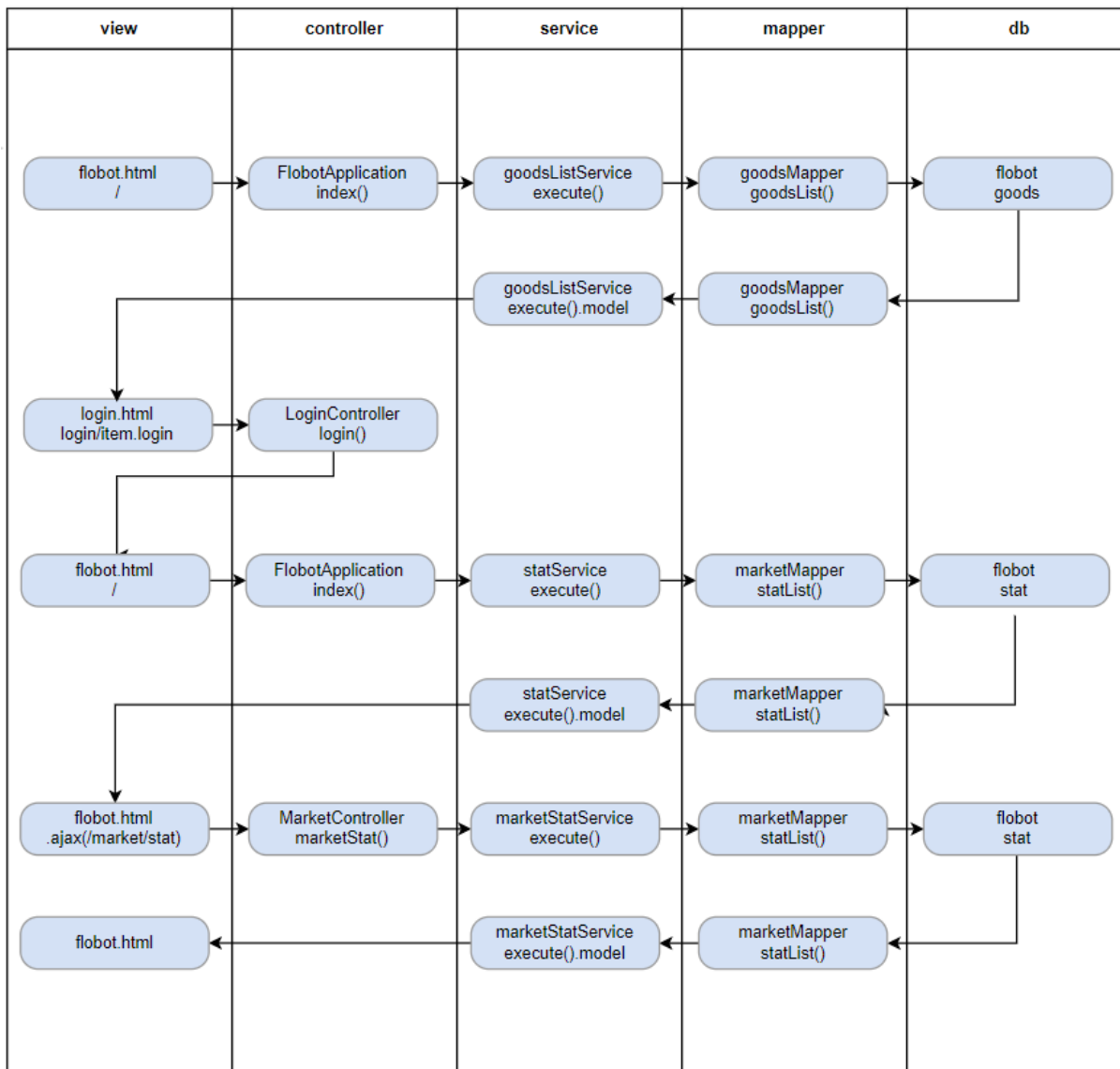
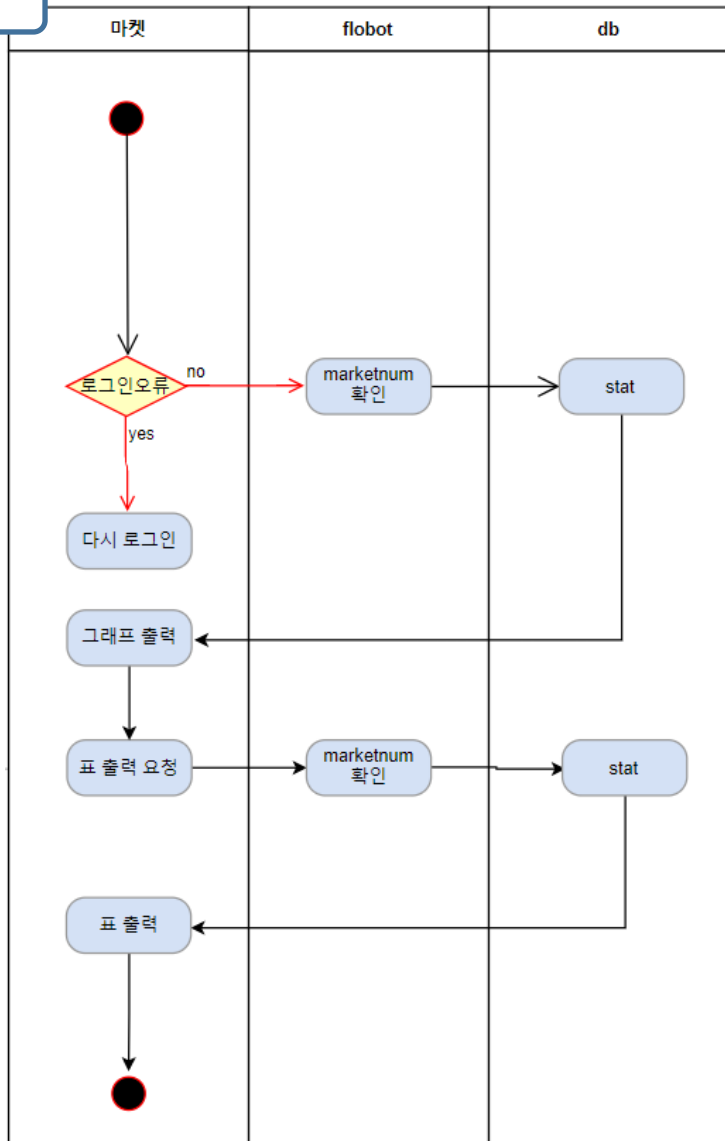




# 13. 다이어그램

활동다이어그램 / 시퀀스다이어그램

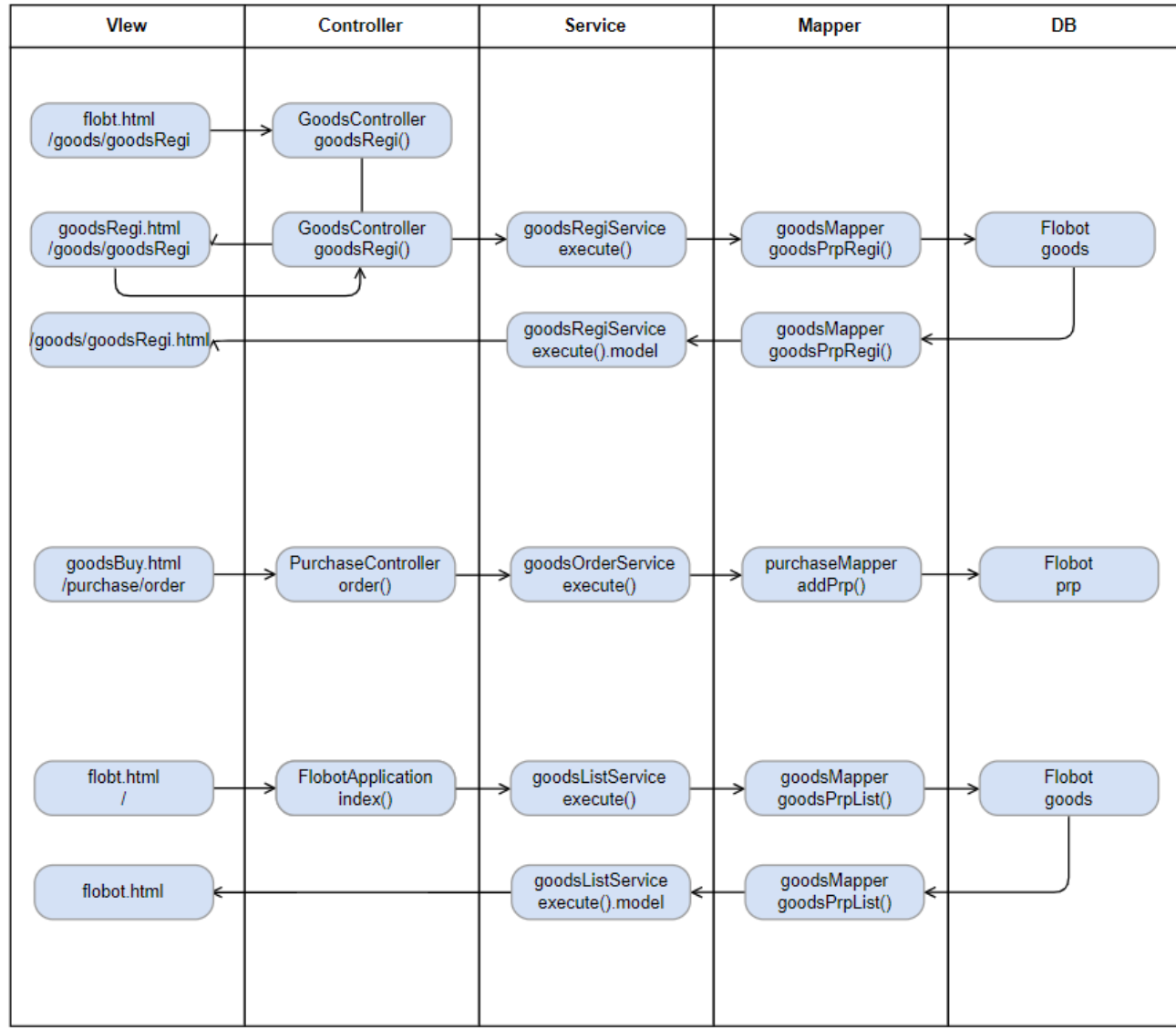
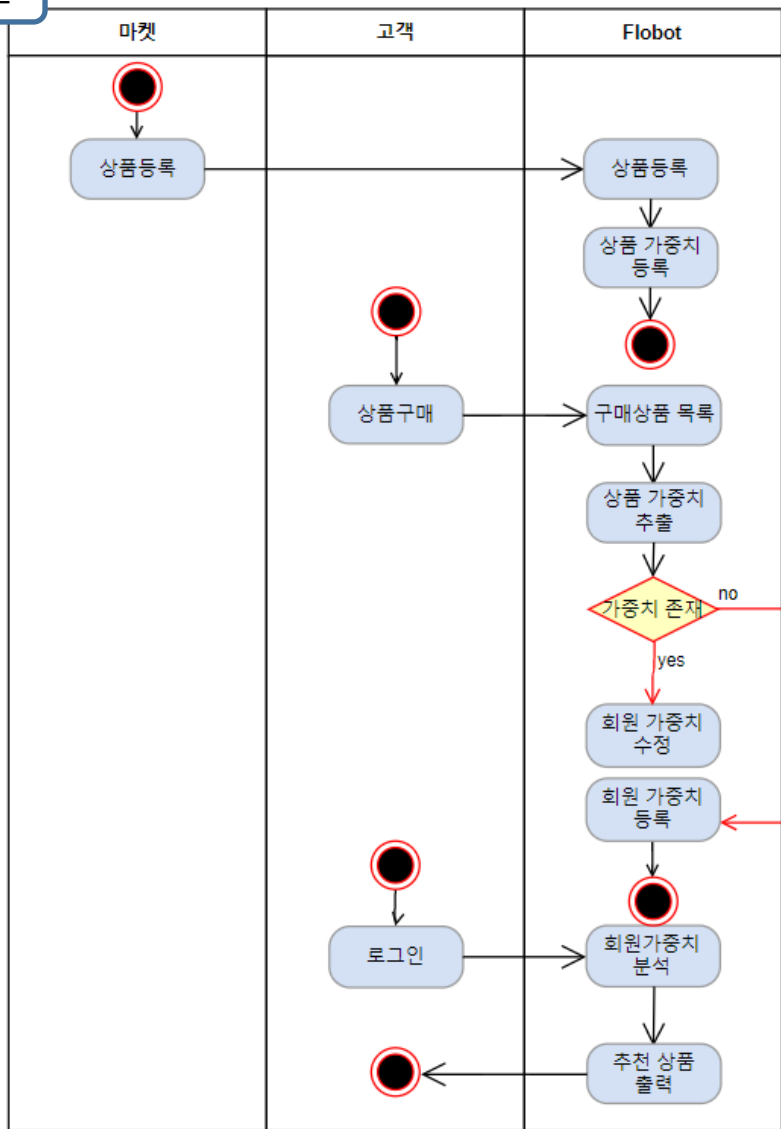
그래프



# 13. 다이어그램

활동다이어그램 / 시퀀스다이어그램

## 상품추천

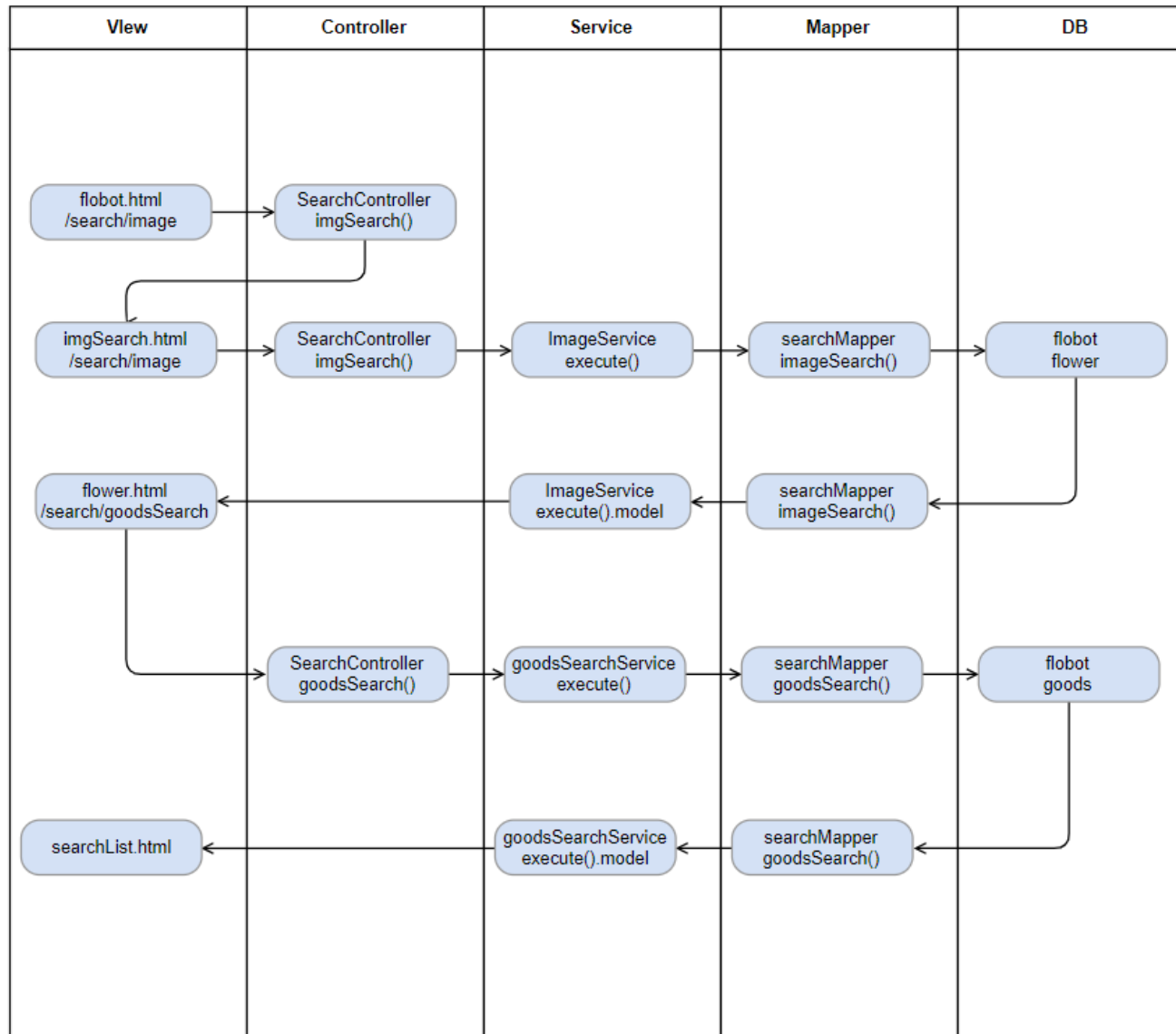
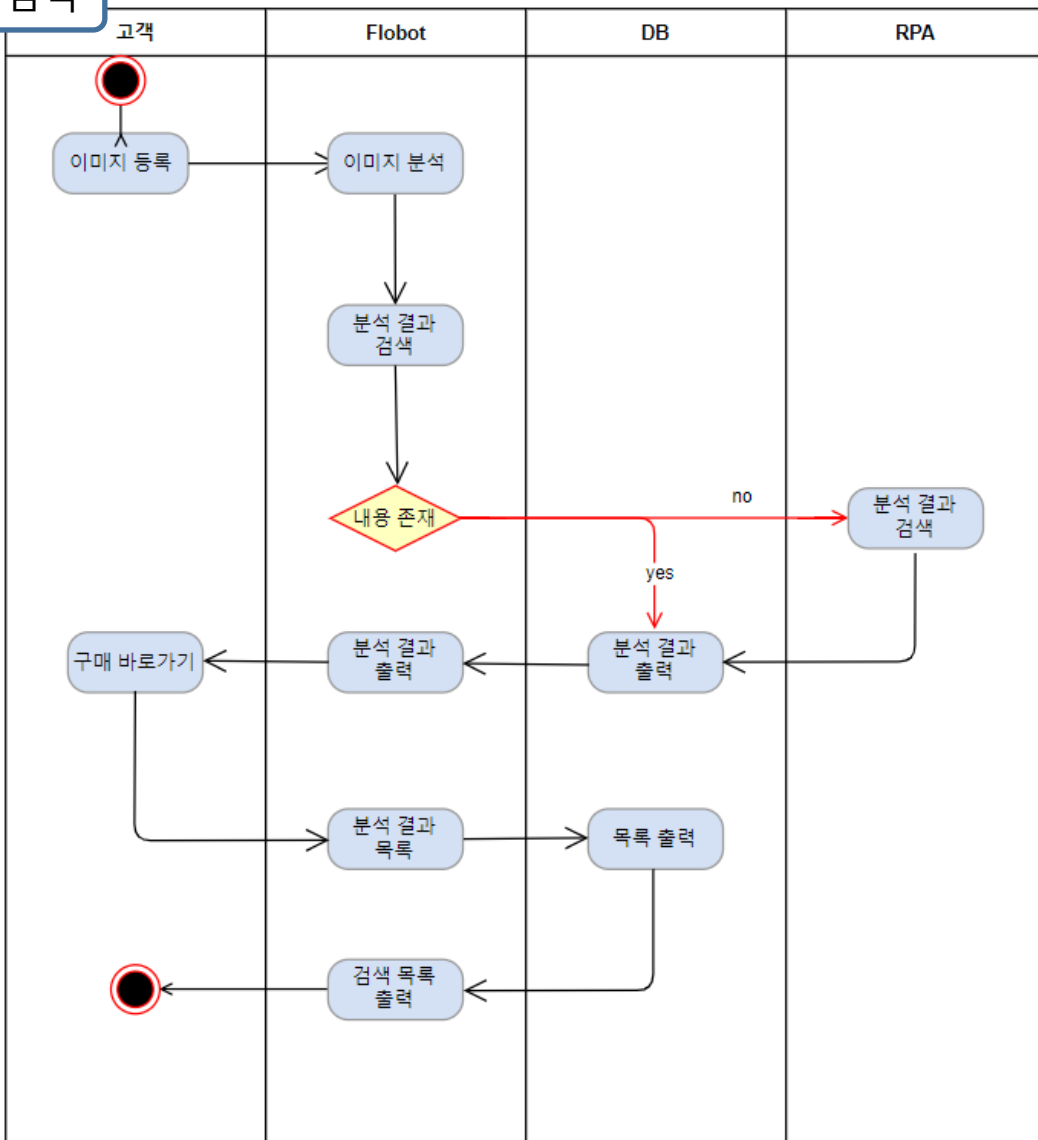


rebot

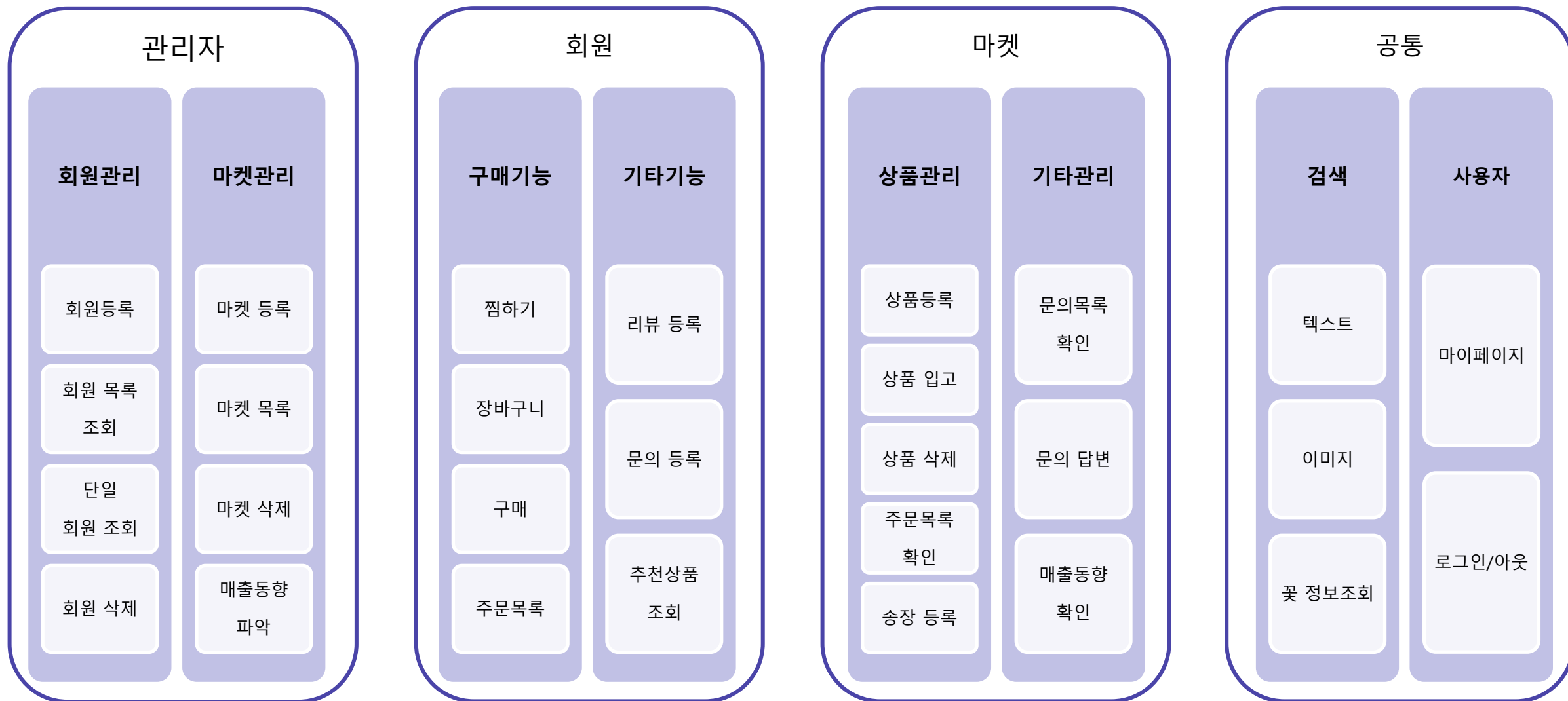
# 13. 다이어그램

활동다이어그램 / 시퀀스다이어그램

## 이미지검색



# 14. 애플리케이션 아키텍처



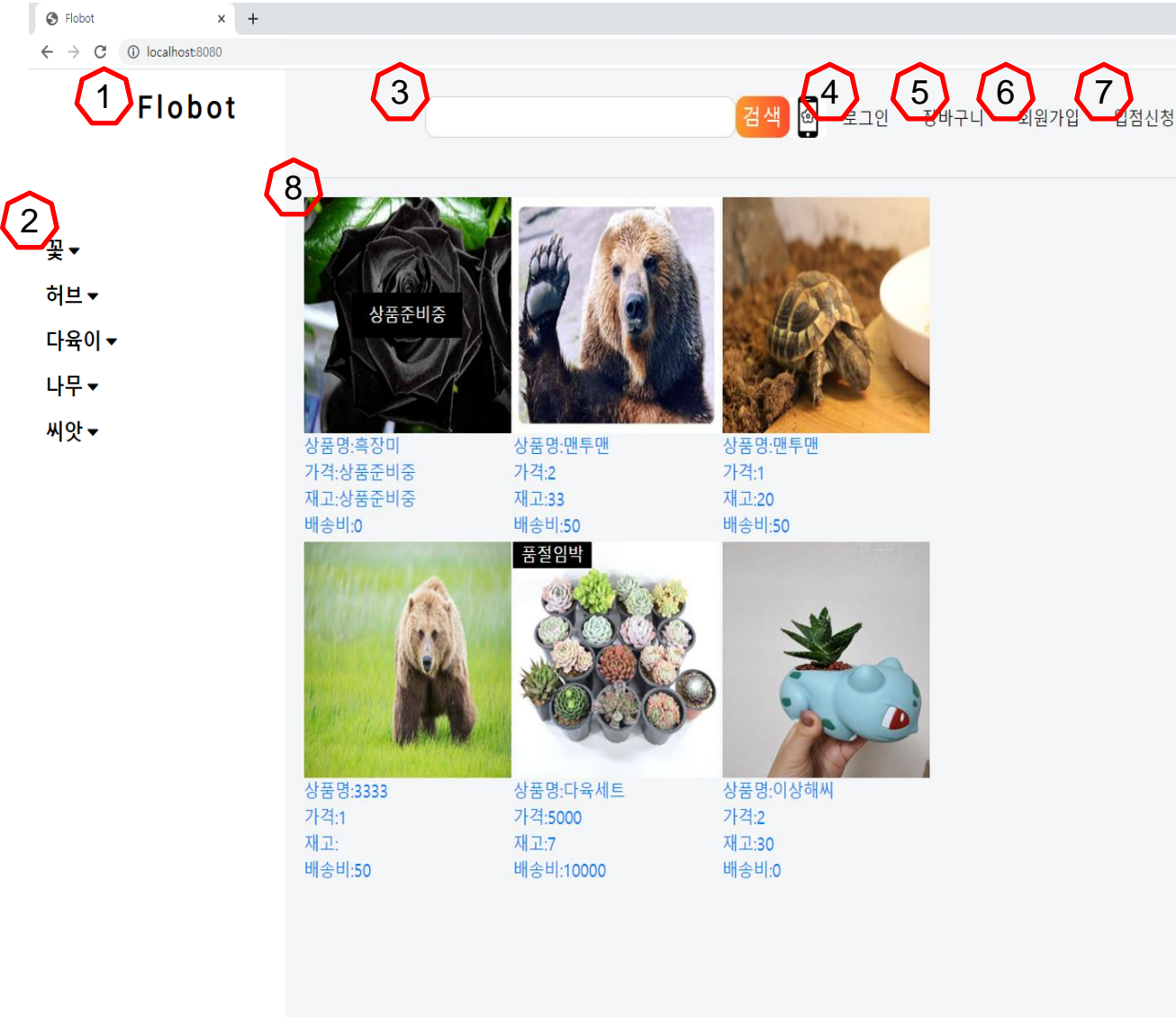
# 15. 기능 명세서

어플리케이션	페이지명	파일명	요소	상세내용	개발	테스트
이미지 검색	메인페이지	flobot.html	form	검색어 입력시 페이지 이동		
			input-text	검색어 입력		
			input-submit	'검색' 버튼 클릭시 form 실행		
			a href	사진 등록시 페이지 이동		
			img	사진 클릭시 a href 실행		
매출 차트	메인페이지	flobot.html	button(id:table)	'표' 버튼 클릭시 함수 실행, ajax로 페이지 로드, '표', '그래프' css 변화		
			button(id:notice2)	'그래프' 버튼클릭시 함수 실행, ajax로 페이지 로드, '표', '그래프' css 변화		

# 15. 기능 명세서

어플리케이션	페이지명	파일명	요소	상세내용	개발	테스트
상품추천	메인페이지	flobot.html	table	페이지 오픈 시 회원의 선호 상품 정보를 불러와 표시한다. 표시내용: 상품이미지, 상품명, 가격, 재고, 배송비, 상품이 입고가 안돼있으면 '상품준비중'이 이미지에 표시됨 재고가 10개 이하일 경우 '품절임박'이 이미지에 표시됨		
구매	구매페이지	goodsBuy.html	table	페이지 오픈시 구매할 상품 정보를 불러와 표시한다. 표시내용: 번호, 상품명, 상품이미지, 수량, 상품가격, 이미지, 합계금액 수령인, 연락처,우편번호, 주소, 상세주소, 배송메세지		
			form	배송정보 입력시 페이지 이동		
			input-text	배송메세지 입력		
			input-button	'검색' 버튼 클릭시 함수 실행		
			input-submit	'결제하기' 버튼 클릭시 form실행		

# 16. 스토리보드



회원 메인페이지	
번호	설명
1	<ul style="list-style-type: none"> <li>- 화면 로고</li> <li>- 클릭시 메인 페이지로 화면 이동</li> </ul>
2	<ul style="list-style-type: none"> <li>- nav</li> <li>- 카테고리가 나열되어있음</li> <li>- 클릭시 카테고리로 검색이 되며 검색 목록 페이지로 화면 이동</li> </ul>
3	<ul style="list-style-type: none"> <li>- 검색어 입력</li> <li>- '검색'버튼 클릭시 해당 검색 목록 페이지로 화면 이동</li> <li>- 검색어 입력 안하고 '검색' 버튼 클릭시 검색 목록 페이지 화면으로 이동하고 상품 나열 기준 : goodsNum</li> </ul>
4	<ul style="list-style-type: none"> <li>- '로그인' 클릭시 로그인창이 팝업 화면으로 열림</li> </ul>
5	<ul style="list-style-type: none"> <li>- '장바구니' 클릭시 로그인이 되어있으면 장바구니 페이지로 화면 이동</li> <li>- 로그인이 되어있지 않을 경우 로그인 창이 팝업 화면으로 열림</li> </ul>
6	<ul style="list-style-type: none"> <li>- '회원가입' 클릭 시 회원가입 페이지로 화면 이동</li> </ul>
7	<ul style="list-style-type: none"> <li>- '입점신청' 클릭시 입점신청 페이지로 화면 이동</li> </ul>
8	<ul style="list-style-type: none"> <li>- 로그인이 되어있지 않을 경우 상품등록 순서대로 상품 나열</li> </ul>

# 16. 스토리보드



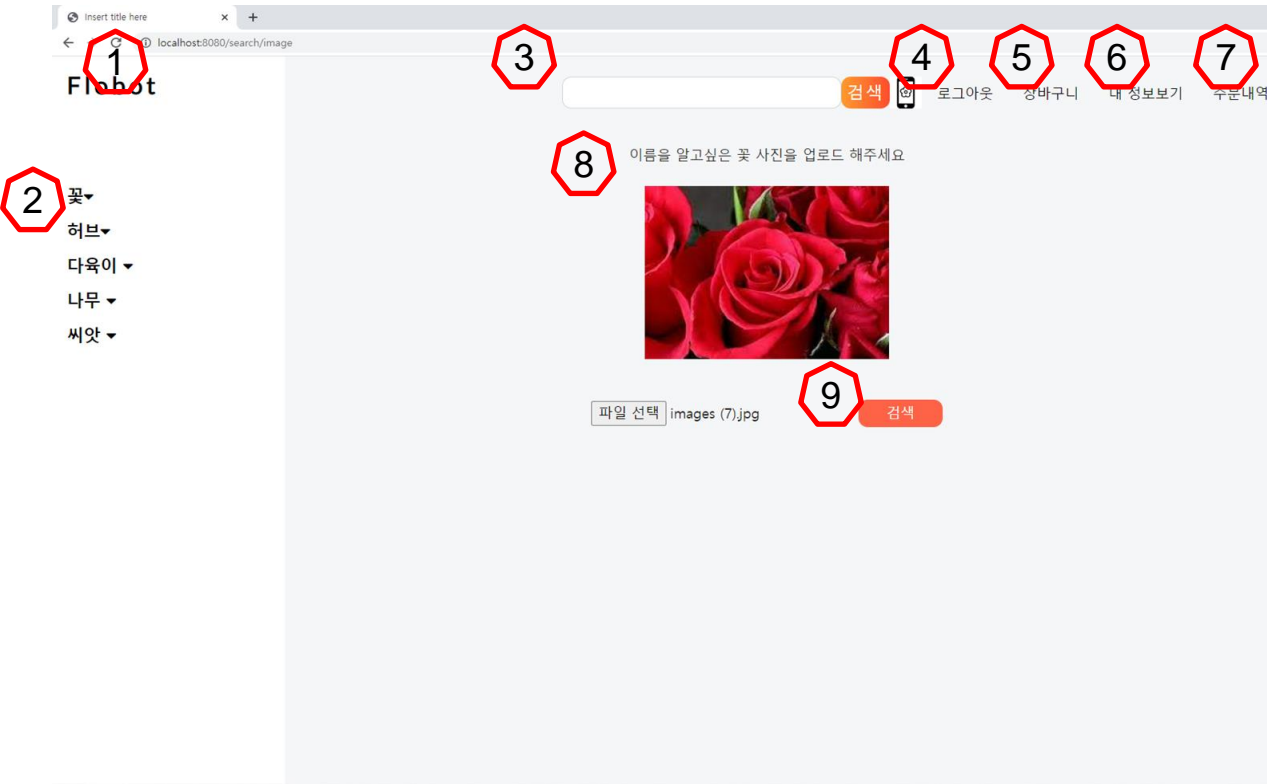
Storyboard image 2 shows the detail page for '내 마켓보기'. A red circle with number 1 points to the '그래프' button. Below the button is a table with the following data:

분기	매출액	영업이익	매출 중 이익
2022.1	583.9	56.1	261.1
2022.2	592.1	55.9	271.0
2022.3	582.8	22.6	270.6
2022.4	615.0	62.0	

입점업체 메인페이지	
번호	설명
1	<ul style="list-style-type: none"> <li>'그래프' 버튼이 클릭 되었을 때를 기준으로 화면 구성</li> <li>'표' 버튼이 클릭 되었을 경우 화면에 표 출력</li> <li>각 버튼 클릭 시 버튼 색상 변경</li> </ul>
2	<ul style="list-style-type: none"> <li>'내 마켓보기' 클릭 시 내 마켓보기 페이지로 화면 이동</li> </ul>
3	<ul style="list-style-type: none"> <li>'내 상품목록' 클릭 시 내 마켓 상품목록 페이지로 화면 이동</li> </ul>
4	<ul style="list-style-type: none"> <li>'내 상품등록' 클릭시 내 상품등록 페이지로 화면 이동</li> </ul>
5	<ul style="list-style-type: none"> <li>'주문목록' 클릭시 주문목록 페이지로 화면 이동</li> </ul>
6	<ul style="list-style-type: none"> <li>'문의목록' 클릭시 문의 목록 페이지로 화면 이동</li> </ul>
7	<ul style="list-style-type: none"> <li>'로그아웃' 클릭시 로그아웃 되며 메인 페이지로 화면 이동</li> </ul>



# 16. 스토리보드



검색 페이지	
번호	설명
1	<ul style="list-style-type: none"> <li>- 화면 로고</li> <li>- 클릭시 메인 페이지로 화면 이동</li> </ul>
2	<ul style="list-style-type: none"> <li>- nav</li> <li>- 카테고리가 나열되어있음</li> <li>- 클릭시 카테고리로 검색이 되며 검색 목록 페이지로 화면 이동</li> </ul>
3	<ul style="list-style-type: none"> <li>- 검색어 입력</li> <li>- '검색'버튼 클릭시 해당 검색 목록 페이지로 화면 이동</li> <li>- 검색어 입력 안하고 '검색' 버튼 클릭시 검색 목록 페이지 화면으로 이동하고 상품 나열 기준 : goodsNum</li> </ul>
4	<ul style="list-style-type: none"> <li>- '로그아웃' 클릭시 로그아웃 되며 메인 페이지로 화면 이동</li> </ul>
5	<ul style="list-style-type: none"> <li>- '장바구니' 클릭시 장바구니 페이지로 화면 이동</li> </ul>
6	<ul style="list-style-type: none"> <li>- '내 정보보기' 클릭시 내 정보보기 페이지로 화면 이동</li> </ul>
7	<ul style="list-style-type: none"> <li>- '주문내역' 클릭시 주문보기 페이지로 화면 이동</li> </ul>
8	<ul style="list-style-type: none"> <li>- '파일선택' 버튼 클릭시 팝업 창이 열림</li> <li>- 팝업창에서 파일 선택시 팝업창이 닫아지고 해당 파일명과, 이미지가 화면에 출력</li> </ul>
9	<ul style="list-style-type: none"> <li>- 파일을 선택하고 '검색' 버튼을 클릭 시 검색 결과 페이지로 화면 이동</li> <li>- 파일을 선택하지 않고 '검색' 버튼을 클릭시 '파일을 선택하세요' 경고 문구 출력</li> </ul>

# 16. 스토리보드



## 검색 결과 페이지

번호	설명
1	- 검색결과 해당 꽃의 대표 이미지 출력
2	- 검색한 꽃의 이름과 설명 출력
3	- '상품 구매' 버튼 클릭시 상품 목록 페이지로 화면이동

# 17. 소스코드

상품관리

## controller

```
@GetMapping("goodsRegi")
public String goodsRegi(GoodsCommand goodscommand) {
    return "thymeleaf/goods/goodsRegi";
}

@PostMapping("goodsRegi")
public String goodsRegi(@Validated GoodsCommand goodsCommand, BindingResult result, Model model, HttpSession session) {
    if(result.hasErrors()) {
        return "thymeleaf/goods/goodsRegi";
    }
    goodsRegiService.execute(goodsCommand, model, session, result);
    return "thymeleaf/goods/goodsInfo";
}
```

## service

```
public void execute(GoodsCommand goodsCommand, Model model, HttpSession session, BindingResult result) {
    AuthInfoVO authInfo = (AuthInfoVO) session.getAttribute("authInfo");
    GoodsVO vo = new GoodsVO();
    String goodsNum = goodsMapper.goodsCreateNum();
    vo.setGoodsNum(goodsNum);
    vo.setDeliveryCost(goodsCommand.getDeliveryCost());
    vo.setGoodsContent(goodsCommand.getGoodsContent());
    vo.setGoodsKind(goodsCommand.getGoodsKind());
    vo.setGoodsName(goodsCommand.getGoodsName());
    vo.setManufacturer(goodsCommand.getManufacturer());
    vo.setMarketNum(authInfo.getUserNum());

    // 파일 경로 설정
    String filedir = "/view/goods/image";
    String filepath = session.getServletContext().getRealPath(filedir);

    MultipartFile mf = goodsCommand.getGoodsMain(); // MultipartFile로 가져온 커맨드 가져오기
    String mainOrg = mf.getOriginalFilename(); // MultipartFile String 타입으로 파일 이름 가져오기
    String extension = mainOrg.substring(mainOrg.lastIndexOf(".")); // 파일 확장자 가져오기
    String randomName = UUID.randomUUID().toString().replace("-", ""); // 랜덤으로 이름 짓는데 파일이름에 -는 들어가면 안되서 없애기
    String htmlFile = randomName + extension;

    File file = new File(filepath + "/" + htmlFile); // 생성한 랜덤이름과 확장자로 파일 생성

    try {
        mf.transferTo(file); // 생성한 파일에 원본이미지 저장
    } catch (Exception e) {
        e.printStackTrace();
    }
    vo.setGoodsMain(htmlFile); // 생성한 파일이름 VO에 저장
    vo.setGoodsMainOrg(mainOrg); // 원본 파일이름 VO에 저장

    if(!goodsCommand.getGoodsImage()[0].getOriginalFilename().isEmpty()) {
        String orgTotal = "";
        String htmlTotal = "";
        for( MultipartFile mf1 : goodsCommand.getGoodsImage()) { // 파일이 여러개있으면 반복으로 가져오기
            mainOrg = mf1.getOriginalFilename(); // MultipartFile String 타입으로 파일 이름 가져오기
            extension = mainOrg.substring(mainOrg.lastIndexOf(".")); // 파일 확장자 가져오기
        }
    }
}
```

## repository

```
@Mapper
public interface GoodsMapper {

    public void goodsRegi(GoodsVO vo);

    public void goodsPrpRegi(PrpVO prp);
}
```

## mapper

```
<insert id="goodsRegi" parameterType="goodsVO">
    insert into goods(GOODS_NUM, GOODS_NAME, GOODS_CONTENT, GOODS_MAIN, GOODS_MAIN_ORG, GOODS_IMAGE, GOODS_IMAGE_ORG
        , DELIVERY_COST, MANUFACTURER, VISIT_COUNT, MARKET_NUM, GOODS_KIND)
    values(#{goodsNum}, #{goodsName}, #{goodsContent}, #{goodsMain}, #{goodsMainOrg}, #{goodsImage}, #{goodsImageOrg}
        , #{deliveryCost}, #{manufacturer}, 0, #{marketNum}, #{goodsKind})
</insert>

<insert id="goodsPrpRegi" parameterType="prpVO">
    insert into prp(prp_num, goods_Num, brilliance, deli_charge, root_yn, flower, seed, herb, tree, fleshy)
    values((select concat('prp00', nvl(max(substr(prp_num,6)),0)+ 1) from prp), #{goodsNum},#{brilliance} ,
        #{deliCharge},#{rootYn},#{flower},#{seed},#{herb},#{tree},#{fleshy})
</insert>
```

Flabot

# 17. 소스코드

그래프

view

```
google.charts.load('current', {'packages':['line']});
google.charts.setOnLoadCallback(drawChart1);
google.charts.setOnLoadCallback(drawChart2);
google.charts.setOnLoadCallback(drawChart3);

function drawChart1() {
    var data = new google.visualization.DataTable();
    data.addColumn('number', '분기');
    data.addColumn('number', '매출액');
    data.addRows([[{statSales}]]);
    var options = {
        chart: {
            title: '매출액',
            subtitle: '1만원'
        }
    };
    var chart = new google.charts.Line(document.getElementById('linechart_material1'));
    chart.draw(data, google.charts.Line.convertOptions(options));
}

function drawChart2() {
    var data = new google.visualization.DataTable();
    data.addColumn('number', '분기');
    data.addColumn('number', '영업이익');
    data.addRows([[{statProfit}]]);
    var options = {
        series: {
            0: { color: '#FE9A2E' }
        },
        chart: {
            title: '영업이익',
            subtitle: '1만원'
        }
    };
    var chart2 = new google.charts.Line(document.getElementById('linechart_material2'));
    chart2.draw(data, google.charts.Line.convertOptions(options));
}

function drawChart3() {
    // ...
}
```

controller

```
@RequestMapping("/")
public String index(Model model) {
    goodsListService.execute(model);
    cookiesService.execute(model);
    statService.execute(model);
    return "thymeleaf/flobot";
}
```

service

```
@Service
public class StatService {
    @Autowired
    MarketMapper marketMapper;

    public void execute(Model model, HttpSession session) {
        AuthInfoVO authInfo = (AuthInfoVO) session.getAttribute("authInfo");
        List<StatVO> list = new ArrayList<StatVO>();
        String i = "[";
        String j = "[";
        String k = "[";
        if(authInfo != null) {
            list = marketMapper.statList(authInfo.getUserNum());
            if(authInfo.getGrade().equals("mar")) {
                for(StatVO vo : list) {
                    i += "["+vo.getStatYear().substring(2)+"."+vo.getStatOut()+"."+vo.getStatSales()+"],";
                    j += "["+vo.getStatYear().substring(2)+"."+vo.getStatOut()+"."+vo.getStatProfit()+"],";
                    k += "["+vo.getStatYear().substring(2)+"."+vo.getStatOut()+"."+vo.getStatTotprofit()+"],";
                }
                i=i.substring(0,i.lastIndexOf(","));
                j=j.substring(0,j.lastIndexOf(","));
                k=k.substring(0,k.lastIndexOf(","));
            }
            i += "]";
            j += "]";
            k += "]";
            model.addAttribute("statSales", i);
            model.addAttribute("statProfit", j);
            model.addAttribute("statTotPorfit", k);
        }
    }
}
```

repository

```
@Mapper
public interface MarketMapper {

    public List<StatVO> statList(String userNum);
}
```

mapper

```
<resultMap type="statVO" id="statResultMap">
<id column="stat_Num" jdbcType="VARCHAR" property="statNum" />

<result column="stat_Year" jdbcType="VARCHAR" property="statYear"/>
<result column="stat_Qu" jdbcType="VARCHAR" property="statQu"/>
<result column="stat_Sales" jdbcType="VARCHAR" property="statSales"/>
<result column="stat_Profit" jdbcType="VARCHAR" property="statProfit"/>
<result column="stat_tot_profit" jdbcType="VARCHAR" property="statTotprofit"/>
<result column="market_Num" jdbcType="VARCHAR" property="marketNum"/>

</resultMap>

<select id="statlist" parameterType="string" resultMap="statResultMap">
select stat_year , stat_qu ,
       stat_sales , stat_profit , stat_tot_profit
from stat
<if test="userNum != null">
where market_Num = #{userNum}
</if>
</select>
```

Flobot

# 17. 소스코드

상품추천

## view

```
<table>
<tr>
<th:block th:each="vo,idx : ${list}">
<th:block th:if="${idx.count < 7}">
<td>
<div class="w3-display-container">
<a th:href="/corner/prodInfo/${vo.goodsVO.goodsNum}"/>

<span th:if="${vo.ipgoVO.totalIpgoQty<10 && vo.ipgoVO.totalIpgoQty>0}" class="w3-tag w3-display-topleft">품절임박</span>
<div class="w3-display-middle">
<span th:if="${vo.ipgoVO.totalIpgoQty == null}" class="w3-button w3-black">상품준비중</span>
</div>
</div>
상품명:[${vo.goodsVO.goodsName}]<br/>
가격:<span th:if="${vo.ipgoVO.goodsPrice == null}">상품준비중</span>
<span th:if="${vo.ipgoVO.goodsPrice!=null}">[[${vo.ipgoVO.goodsPrice}]]</span> <br/>
재고:<span th:if="${vo.ipgoVO.totalIpgoQty == null}">상품준비중</span>
<span th:if="${vo.ipgoVO.totalIpgoQty<10 && vo.ipgoVO.totalIpgoQty>0}">[[${vo.ipgoVO.totalIpgoQty}]]</span>
<span th:if="${vo.ipgoVO.totalIpgoQty>10}">[[${vo.ipgoVO.totalIpgoQty}]]</span><br/>
배송비:[[${vo.goodsVO.deliveryCost}]]
</td>
<th:block th:if="${idx.count % 3 == 0}" th:utext="'</tr><tr>'"/>
</th:block>
</th:block>
</table>
```

## controller

```
@SpringBootApplication
@Controller
public class FlobotApplication {
    @Autowired
    GoodsListService goodsListService;
    @Autowired
    CookiesService cookiesService;
    @Autowired
    StatService statService;

    public LoginCommand loginCommand() {}
    public static void main(String[] args) {}
    @RequestMapping("/")
    public String index(Model model, HttpSession session,
        goodsListService.execute(model, session);
        cookiesService.execute(request, model);
        statService.execute(model, session);
        return "thymeleaf/flobot";
    }
}
```

## service

```
@Service
public class GoodsListService {
    @Autowired
    GoodsMapper goodsMapper;

    public void execute(Model model, HttpSession session) {
        AuthInfoVO authInfo = (AuthInfoVO) session.getAttribute("authInfo");
        if(authInfo!=null) {
            if(authInfo.getGrade().equals("mar")) {
                List<GoodsGoodsIpgoPrpVO> list = goodsMapper.goodsList(authInfo);
                model.addAttribute("list", list);
            }else if(authInfo.getGrade().equals("mem")){
                List<GoodsGoodsIpgoPrpVO> list = goodsMapper.goodsPrpList(authInfo);
                model.addAttribute("list", list);
            }else{
                List<GoodsGoodsIpgoPrpVO> list = goodsMapper.goodsList(null);
                model.addAttribute("list", list);
            }
        }else {
            List<GoodsGoodsIpgoPrpVO> list = goodsMapper.goodsList(null);
            model.addAttribute("list", list);
        }
    }
}
```

@Mapper

public interface GoodsMapper {

public GoodsVO goodsListselect(String goodsNum);

public List<GoodsGoodsIpgoPrpVO> goodsPrpList(String userNum);

public void goodsPrpRegi(PrpVO prp);

```
<resultMap type="goodsVO" id="goodsResultMap">
<resultMap type="ipgoVO" id="goodsIpgoResultMap">
<resultMap type="prpVO" id="prpResultMap">
<resultMap type="GIPVO" id="GIPResultMap">
<select id="goodsPrpList" parameterType="string" resultMap="GIPResultMap">
select *
from ( select g.GOODS_NUM, GOODS_NAME, GOODS_CONTENT, GOODS_MAIN, DELIVERY_COST, MANUFACTURER, VISIT_COUNT , MARKET_NUM , GOODS_KIND
, i.TOTAL_IPGO_QTY, i.goods_price
from goods g left outer join (SELECT GOODS_NUM, SUM(IPGO_QTY) AS TOTAL_IPGO_QTY, MAX(GOODS_PRICE) AS GOODS_PRICE
FROM GOODS_IPGO
GROUP BY GOODS_NUM) i
on g.goods_num = i.goods_num) g join prp p
on g.goods_num = p.goods_num
order by ((abs((select brilliance
from prp p join member m
on m.member_num = p.member_num
where m.member_num = #{userNum})-p.brilliance)*0.4)+(abs((select deli_charge
from prp p join member m
on m.member_num = p.member_num
where m.member_num = #{userNum})-p.deli_charge)*3)+(abs((select root_yn
from prp p join member m
on m.member_num = p.member_num
where m.member_num = #{userNum})-p.root_yn )*3)+(select count(*) as cnt
from goods g join purchase_list l
on g.goods_num = l.goods_num join (select * from purchase where member_num = #{userNum}) p
on l.purchase_num = p.purchase_num
where goods_kind='flower'
group by goods_kind)*p.flower+NVL((select count(*) as cnt
from goods g join purchase_list l
on g.goods_num = l.goods_num join (select * from purchase where member_num = #{userNum}) p
on l.purchase_num = p.purchase_num
where goods_kind='fleshy'
group by goods_kind),0)*p.fleshy+NVL((select count(*) as cnt
from goods g join purchase_list l
on g.goods_num = l.goods_num join (select * from purchase where member_num = #{userNum}) p
on l.purchase_num = p.purchase_num
where goods_kind='tree'
group by goods_kind),0)*p.tree+NVL((select count(*) as cnt
from goods g join purchase_list l
on g.goods_num = l.goods_num join (select * from purchase where member_num = #{userNum}) p
on l.purchase_num = p.purchase_num
where goods_kind='herb'
group by goods_kind),0)*p.herb+NVL((select count(*) as cnt
from goods g join purchase_list l
on g.goods_num = l.goods_num join (select * from purchase where member_num = #{userNum}) p
on l.purchase_num = p.purchase_num
where goods_kind='seed'
group by goods_kind),0)*p.seed) desc
</select>
```

## repository

## mapper

bat

# 17. 소스코드

이미지검색

view1

view2

```
</a> href="/search/image"></a>
<div align="center">
  <form action="image" method="post" enctype="multipart/form-data">
    이름을 알고싶은 꽃 사진을 업로드 해주세요
    <br/><br/>
    <div id="image_container"></div>
    <br/><br/>
    <div class="search_submit">
      <input type="file" name="searchImg" id="image" accept="image/*" onchange="setThumbnail(event);" multiple/>
      <input type="submit" value="검색" />
    </div>
  </form>
</div>
```

```
@Autowired
ImageService imageService;
@PostMapping("/image")
public String imgSearch(LoginCommand loginCommand, @RequestParam(value="searchImg") MultipartFile mf) {
    String filedir = "C:/Flobot/image";
    MultipartFile mf = searchImg; // MultipartFile로 가져온 커맨드 가져
    String mainOrg = mf.getOriginalFilename(); // MultipartFile String
    File file = new File(filedir + "/" + mainOrg); // 생성한 랜덤이름과
    try {
        mf.transferTo(file); // 생성한 파일에 원본이미지 저장
    } catch (Exception e) {
        e.printStackTrace();
    }
    String result="";
    try {
        Process process = Runtime.getRuntime().exec("powershell -Command");
        BufferedReader reader = new BufferedReader(
            new InputStreamReader(process.getInputStream()));
        String line = null;
        StringBuffer sb = new StringBuffer();
        while ((line = reader.readLine()) != null) {
            sb.append(line);
        }
        result = sb.toString().split("step")[1].replace("\n", "");
    } catch (Exception e) {
        e.printStackTrace();
    }
    file.delete();
    model.addAttribute("flower", result);
    String command = "schtasks /run /tn \"BrityRPA_P_TestFlolbot\"; //
    Process process;
    try {
        process = Runtime.getRuntime().exec(command);
        BufferedReader reader = new BufferedReader(new InputStreamReader(
            process.getInputStream()));
        reader.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
    imageService.execute(model, result);
    return "thymeleaf/search/flower";
}
```

controller

```
@Service
public class ImageService {
    @Autowired
    SearchMapper searchMapper;

    public void execute(Model model, String result) {
        System.out.println("[+result+]");
        if(result.equals("roses")) {
            result = "장미";
        }
        else if(result.equals("daisy")) {
            result = "국화";
        }
        else if(result.equals("dandelion")) {
            result = "민들레";
        }
        else if(result.equals("sunflowers")) {
            result = "해바라기";
        }
        else if(result.equals("tulips")) {
            result = "tulips";
        }

        //System.out.println(result + "=====");
        List<FlowerVO> list = searchMapper.imageSearch(result);

        model.addAttribute("list", list);
    }
}
```

service

repository

```
@Mapper
public interface SearchMapper {

    public List<GoodsVO> goodsSearch(PageVO vo);

    public Integer goodsCount(String keyword);

    public List<FlowerVO> imageSearch(String result);
}
```

mapper

```
<select id = "imageSearch" parameterType="string" resultType="flowerVO">
  select flower_Num flowerNum, flower_Name flowerName, flower_Kind flowerKind,
    flower_Content flowerContent, flower_Image flowerImage
  from flower
  where flower_name = #{result}
</select>
</mapper>

1 import numpy as np
2 import os
3 import PIL
4 import tensorflow as tf
5 from tensorflow import keras
6 from tensorflow.keras import layers
7 from tensorflow.python.keras.layers import Dense, Flatten
8 from tensorflow.keras.models import Sequential
9 from tensorflow.keras.optimizers import Adam
10 from PIL import Image
11 import cv2
12 import pathlib
13 from pathlib import Path
14
15 loaded_model = keras.models.load_model("C:/Flobot/ma.h5")
16
17 data_dir = Path('C:/Flobot/image')
18 data_dir = pathlib.Path(data_dir)
19 flower = list(data_dir.glob('.*'))
20 PIL.Image.open(str(flower[0]))
21 import cv2
22 image=cv2.imread(str(flower[0]))
23 image_resized= cv2.resize(image, (180,180))
24 image=np.expand_dims(image_resized,axis=0)
25 pred=loaded_model.predict(image)
26 output_class=['daisy', 'dandelion', 'roses', 'sunflowers', 'tulips'][np.argmax(pred)]
27 print(output_class)
```

python

Flobot



# 17. 소스코드

## 이미지검색 모델 트레이닝

python

```
In [1]: import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
import pathlib

dataset_url = "https://storage.googleapis.com/download.tensorflow.org/example_images/flower_photos.tgz"
data_dir = tf.keras.utils.get_file('flower_photos', origin=dataset_url, untar=True)
data_dir = pathlib.Path(data_dir)
img_height, img_width = 180, 180
batch_size = 32

train_ds = tf.keras.preprocessing.image_dataset_from_directory(
    data_dir,
    validation_split=0.2,
    subset="training",
    seed=123,
    label_mode="categorical",
    image_size=(img_height, img_width),
    batch_size=batch_size
)

val_ds = tf.keras.preprocessing.image_dataset_from_directory(
    data_dir,
    validation_split=0.2,
    subset="validation",
    seed=123,
    label_mode="categorical",
    image_size=(img_height, img_width),
    batch_size=batch_size
)

resnet_model = Sequential()
pretrained_model = tf.keras.applications.ResNet50(
    include_top=False,
    input_shape=(180, 180, 3),
    pooling='avg',
    classes=5,
    weights='imagenet'
)

for layer in pretrained_model.layers:
    layer.trainable = False

resnet_model.add(pretrained_model)
resnet_model.add(Flatten())
resnet_model.add(Dense(512, activation='relu'))
resnet_model.add(Dense(5, activation='softmax'))

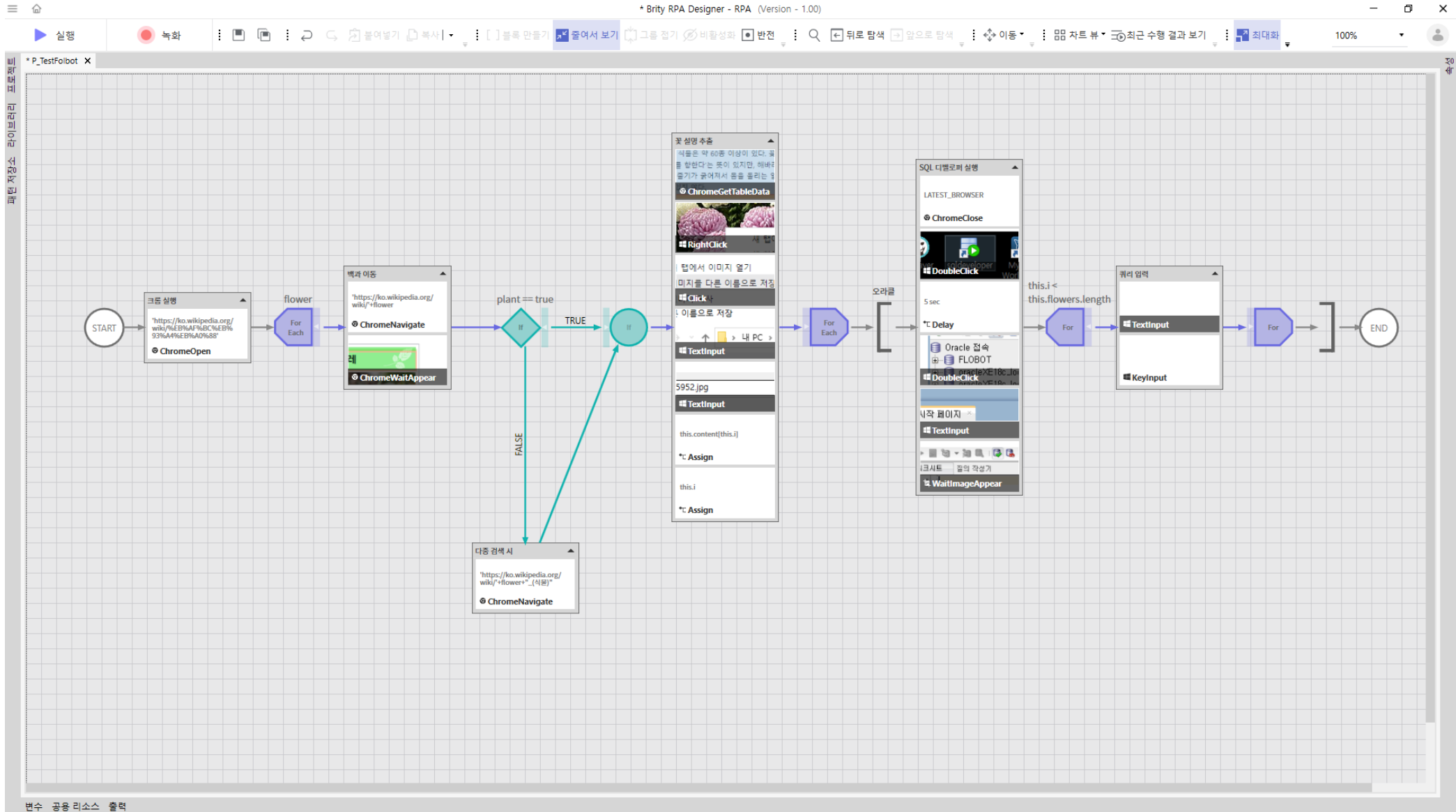
epochs = 3
resnet_model.compile(
    optimizer='adam',
    loss=tf.keras.losses.CategoricalCrossentropy(),
    metrics=['accuracy']
)

history = resnet_model.fit(train_ds, validation_data=val_ds, epochs=epochs)

# 모델 저장
resnet_model.save("ma.h5")
```

# 17. 소스코드

이미지검색



Robot



# 17. 소스코드

이미지검색

The screenshot displays the RPA software interface for a project named 'P\_TestFolbot'. The interface is divided into several sections:

- Project Tree (Left):** Shows the project structure, including 'RPA', '프로세스' (Processes), and 'P\_TestFolbot'.
- Workspace (Center):** Contains a flowchart with various steps and connectors. Visible steps include 'Click', 'Get Data', 'For Each', 'ChromeClose', 'MessageBox', 'Textinput', and 'KeyInput'.
- Right Panel:** Displays settings for the selected step, including a 'Delay' setting and a 'Textinput' field.
- Bandicam Recording Window:** Overlaid on the workspace, showing a recording interface with a 'REC' button and a 'BANDICUT' button.
- Status Bar (Bottom):** Displays logs and a file path: 'C:\Users\hik-10\Documents\Brity RPA Projects\RPA\P\_TestFolbot.proc is saved.'

Flabot

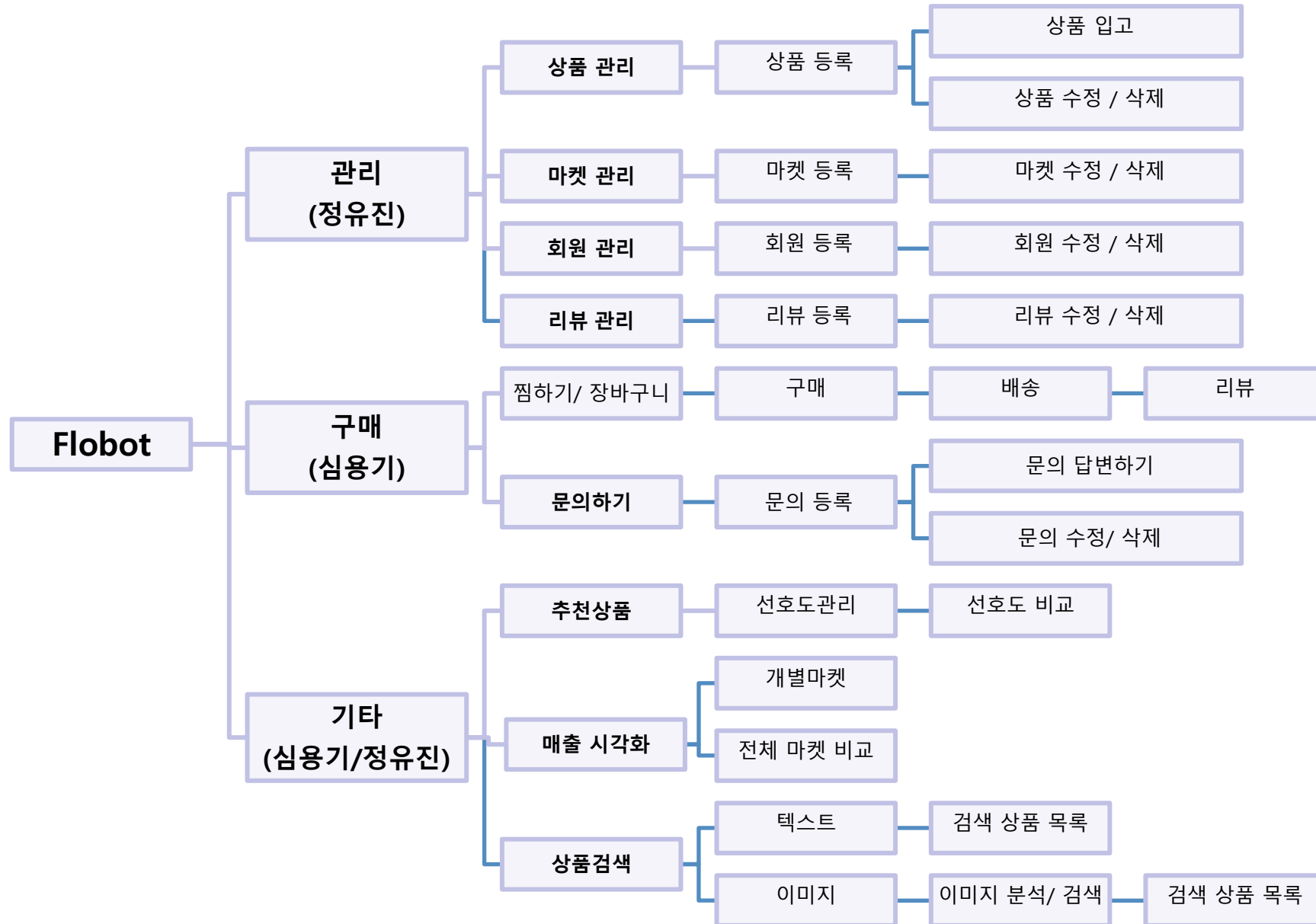
# 18. 일정(WBS)

타이틀	진행제목	진행자	4월				5월					6월	
			1주	2주	3주	4주	1주	2주	3주	4주	5주	1주	2주
개념설계	기획 아이디어 선정	팀 전체											
	시나리오 구성												
	시나리오 업무 흐름												
	업데이트 타임 정의												
	관계정의 (ERD)												
	속성정의												
	식별자정의												
	트레이닝 정의												
논리설계	장규칙	팀 전체											
	데이터베이스 정의												
프로젝트 준비	프로젝트 기본세팅 구성												
	발표자료												
	Html 세팅												
회원 프로세스	회원가입	정유진											
	회원정보보기												
	회원정보수정												
	회원탈퇴												
	회원 비밀번호 수정												
마켓관리 프로세스	마켓가입	심용기											
	마켓정보보기												
	마켓정보수정												
	마켓탈퇴												
	마켓 비밀번호 수정												
로그인 프로세스	아이디 / 비밀번호 찾기	팀 전체											
	로그인 / 로그아웃 / 아이디 기억하기 / 로그인유지												

# 18. 일정(WBS)

타이틀	진행계획	진행자	4월						5월						6월	
			1주	2주	3주	4주	5주	6주	1주	2주	3주	4주	5주	6주	1주	2주
관리자 페이지	회원 목록보기	심용기														
	마련 목록보기															
	회원 관리	정유진														
	마련 관리															
상품	상품 등록	심용기														
	상품 목록 출력															
	상품 상세 설정															
	장바구니 / 위시리스트	정유진														
	상품 입고															
	리뷰쓰기															
	문의하기															
결제 프로세스	상품 결제 / 구매	팀 전체														
	회원 주문 목록															
	상품 문의 목록															
	마련 주문 목록															
	마련 문의 목록 / 답글															
	마련 상품 발송															
배송 정보	개별 배송 정보	팀 전체														
	통합 배송 정보															
이미지 검색	리스트 상품 검색	팀 전체														
	이미지 상품 검색															
	검색 상품 목록															
후원 상품	후원 상품 목록	팀 전체														
테스트	통합 테스트	팀 전체														

# 19. 주요 기능(WBS)



## 20. 사용된 프로그램



ORACLE®

PARTNER  
PROGRAM



MyBatis



flabot