

IC4U(I See For You)

- 스마트 냉장고 시스템 -

- 제정일자 : 2015년 01월 9일
- 문서버전 : Ver 3.0
- 팀명 : 하얏게 불태웠어

문 서 승 인 정 보

프로젝트 명	IC4U(I See For You)
TASK 명	스마트 냉장고 시스템
문 서 명	프로젝트 완료보고서
발행 년 월일	2015년 01월 9일

구 분	성 명	서 명	일 자
작 성 자	김주영		2015.01.09
작 성 자	김성민		2015.01.09
작 성 자	김연욱		2015.01.09
작 성 자	변소정		2015.01.09
Technical Leader	김연욱		2015.01.09
Team Leader	김주영		2015.01.09
프로젝트관리자	문상환		2015.01.09

문 서 이 력 정 보

Ver.	Page	작성일자	변경사항	작성자	승인자	승인일자
1.0	18	2014.11.18	제 정	김주영		2014.11.19
1.1	19	2014.11.19	수 정	김주영		2014.11.24
1.2	23	2014.11.21	1. 프로그램 기능 수정 2. 내용 구체화	김주영		2014.11.24
2.0	23	2014.12.12	1차 버전 이후 변경 사항에 의한 최신화	김주영		2014.12.15
3.0	67	2015.01.07	기능수정 및 고도화	김주영		2015.01.09
3.0	88	2015.01.11	내용 추가	김주영		2015.01.11

목차

1. 프로젝트 배경
 - A. 솔루션 개발의 필요성
 - B. 신기술 개발 시 파급효과 및 활용방안
 - C. 프로젝트 개발 목적 및 당위성
2. 프로젝트 진행 보고
 - A. 프로젝트 개발 분야
 - B. 프로젝트 진행 일정
 - C. 수행업무 및 담당자
 - D. 개발 도구
 - E. 프로젝트 목표
 - F. 단계별 아키텍처
 - G. 팀 개발일정
3. 개인별 일정
4. 프로젝트 배경 기술/ 지식/ 알고리즘
 - A. Server/Client/DB
 - B. Raspberry Pi
 - C. Image processing
 - D. Web page
5. 상세 개발 내용
 - A. 시스템 아키텍처
 - B. 시스템 설계 및 구현
 - C. 시스템 특징,장점
 - D. 적용 분야 및 확장 방안
 - E. 개발 기술

6. 사용자 매뉴얼

- A. 필수 S/W, H/W, Program
- B. 개발환경 설치
- C. 프로젝트 컴파일
- D. 실행

7. 프로젝트 마무리

- A. 기대효과
- B. 문제점
- C. 개선방안
- D. 참고문헌 및 논문
- E. 참고사이트
- F. 팀원 별 소감

1. 프로젝트 배경

A. 솔루션 개발의 필요성

현재의 스마트 냉장고의 경우 가격 비싸다는 단점 때문에 보급이 크게 이루어지지 않았으며, 시장에는 여전히 내부 카메라가 존재하지 않는 냉장고가 대부분인 상황이다. 그에 따라서 저렴하고 사용하기 쉬운 외부 디바이스(Raspberry Pi)를 제공하여, 스마트 냉장고가 제공하고 있는 기능을 일반 냉장고에도 사용할 수 있는 기반을 제공한다.

B. 신기술 개발 시 파급효과 및 활용방안

기존 냉장고를 저렴한 가격으로 스마트 냉장고에 기능을 사용할 수 있다.

C. 프로젝트 개발 목적 및 당위성

냉장고 내부에서 촬영된 정보를 처리 및 관리하는 기능을 제공하여 사용자가 냉장고 식자재를 효율적으로 관리하는 데 도움을 줄 것이다.

2. 프로젝트 진행 보고

A. 프로젝트 개발 분야

스마트 냉장고 시스템은 일종의 사물인터넷 시스템이라고 할 수 있다. 사물인터넷이란 생활 속의 사물들이 유무선 네트워크로 연결하여 인터넷으로 정보를 주고받는 기술을 의미한다. 사물인터넷 시스템은 네트워크로 정보를 공유하기 때문에 수 많은 정보가 모이게 된다. 합쳐진 정보들로 하여금 새로운 데이터를 추출하게 되는데 이 개념이 바로 Big데이터이다. Big데이터와 사물인터넷의 접목은 경험의 추가로 데이터가 정교해지며, 정교한 데이터는 사용자에게 더 나은 서비스를 제공한다. 그렇기 때문에 차세대 기술로서 각광받고 있다.




개발된 "스마트 냉장고 시스템"은 Embedded, 웹 서버, 웹 페이지, 영상처리의 접목으로 이루어진다. 네트워크로 연결된 사물기기(Embedded)는 서버로 정보를 전송하여 일련의 과정으로 영상을 해석하고 유용한 정보를 추출한다. 결과적으로 사용자의 인터페이스에는 자동으로 새로운 정보가 업데이트된다. 이 과정에서 스마트 냉장고는 사물인터넷과 Big데이터의 접목으로 정교화 할 수 있다. 식자재의 출입을 구별하기 위한 Template과 식 자재 자체의 품목을 구분하기 위한 Template는 자동으로 서버로 전송되며, 유용한 정보를 추출하여 데이터베이스에 추가

한다. 경험으로 학습된 정보는 더 높은 인식률로서 정보를 제공한다. 데이터베이스에 추가된 정보는 사용자에게 고급정보를 제공한다.











스마트 냉장고 디바이스 단에서 식자재의 출입의 기준점이 되는 Hand Detection에서 최초의 Template과 경험으로 학습된 가족 구성원의 Hand Template가 데이터 베이스에 추가되면서 더욱 정교한 Template를 구성할 수 있다. 또한 식자재의 품목을 구별하는 데에도 이와 같은 기술이 접목 가능하다. 최초 식자재의 Template는 경험으로 추가되는 데이터에 의해 구분에 있어 더욱 민감해진다.

최초로 Embedded 기계(카메라와 Raspberry PI)를 냉장고 내부에 설치한다. 설치된 카메라는 사용자가 식 자재를 집어 넣을 때, 집어 놓을 때 사진을 찍어 서버로 전송한다. 서버에서는 데이터의 흐름을 총괄적으로 관리한다. 첫 번째로, 이미지데이터가 들어오면 순서대로 영상처리 과정을 거쳐 식자재의 품목을 문자열데이터로 얻어낸다. 얻어낸 품목데이터는 웹 페이지와 연동되어있는 데이터베이스에 추가, 삭제되어 사용자가 웹 페이지에서 확인 할 수 있다.

B. 프로젝트 진행 일정

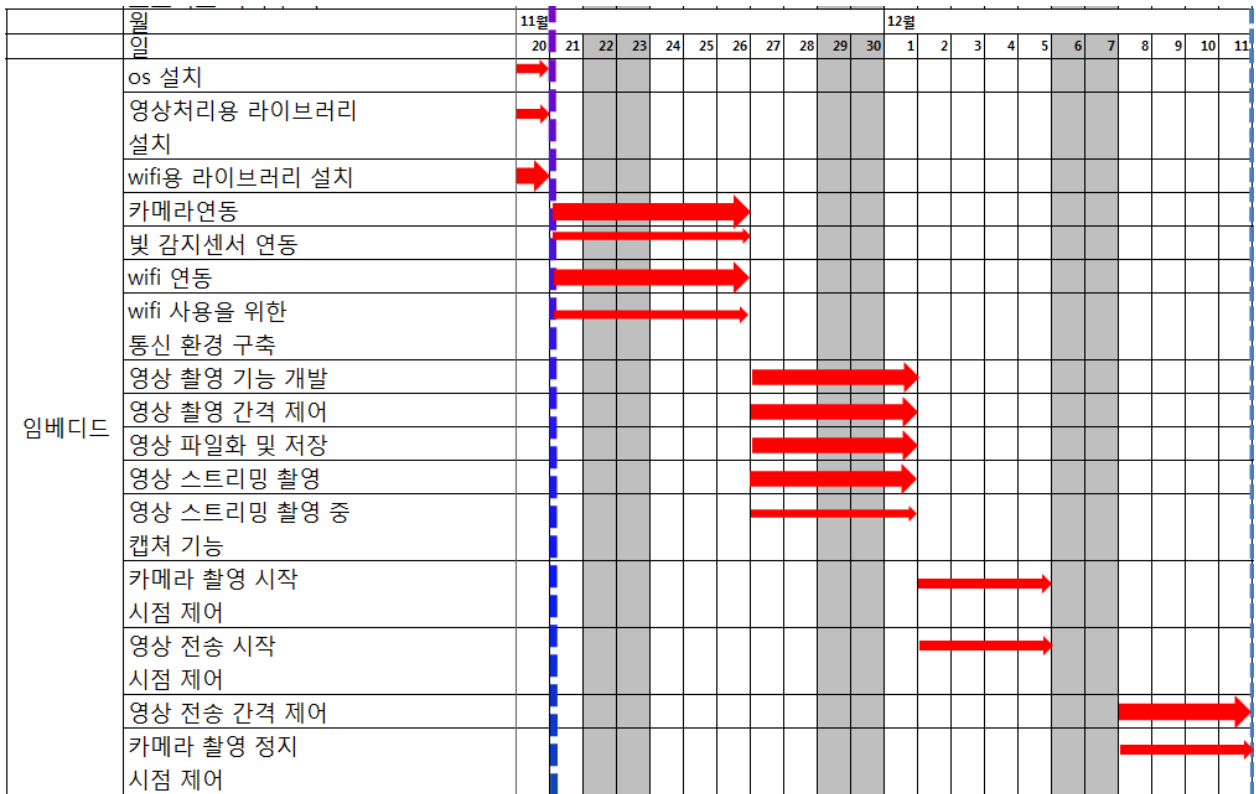
일정 완료	
일정 진행 중	
공휴일	

1) 프로젝트 기획 /분석 /설계

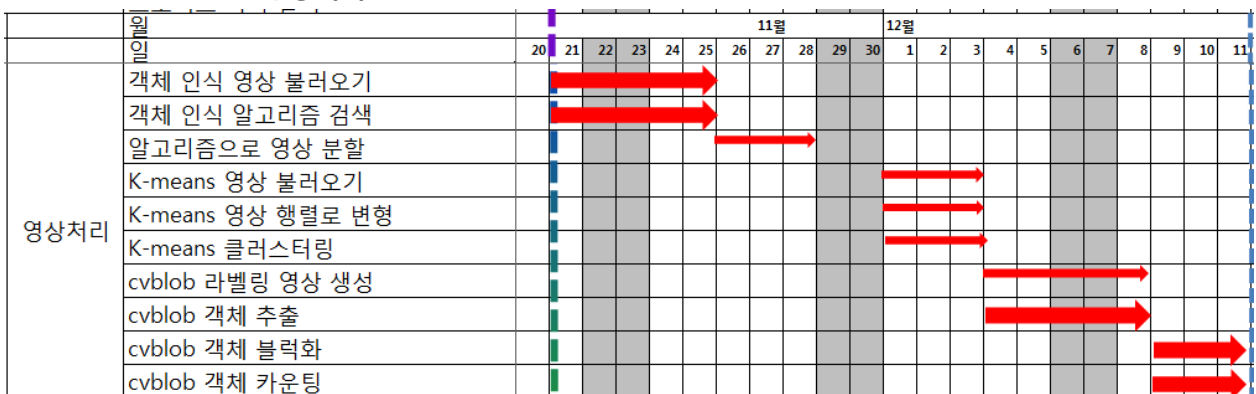
업무 분야	일 일	11월																	
		7	8	9	10	11	12	13	14	15	16	17	18	19	20				
프로젝트 기획	개인별 주제 발표																		
	팀원 구성																		
	브레인스토밍																		
	주제 선정																		
	팀원 역할 선정																		
	일정 예측																		
	기획서 작성																		
	프로젝트 서버 설치																		
프로젝트 분석	기능 정의																		
프로젝트 설계	SW_UI																		
	HW_외부구조설계서 작성																		

2) 1순위

i. 임베디드



ii. 영상처리



iii. 서버/DB

월	일	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11
	11월																						
	12월																						
서버/DB	외부 접속 포트 설정																						
	디바이스 소켓 개발(python)																						
	서버 소켓 개발(c#)																						
	프로토콜 정의																						
	통신 구조 설계																						
	FTP 서버 구축																						
	서버 관리자 ID 생성																						
	외부 접속 ID 생성 및 저장																						
	ic4u DB 설계 및 생성																						
	통신 기능 통합																						

3) 2순위

i. 임베디드

월	일	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	12월																				
임베디드	외부 서버와 접속																				
	로컬 서버 내에서의 이미지 전송																				
	이미지 스트리밍 전송																				
	영상 내 색상 검출																				
	색상 Tracking																				
	손 동작 인식 시점 확인																				
	영상 캡처 간격 조절																				

ii. 영상처리

월	일	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	12월																			
영상처리	영상 분할 및 객체 인식 알고리즘 결정																			
	Canny Edge Detection Smoothing																			
	Canny Edge Detection																			
	Edge 추출																			
	Canny Edge Detection																			
	Edge 검출 정교화																			
	Canny Edge Detection																			
	연관성 판별																			
	Canny Edge Detection																			
	영상 표시																			

iii. 서버/DB

		12월																														
		12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31											
서버/DB	웹 서버 구축																															
	DB 서버 구축																															
	통합 서버 환경 구축																															
	서버 가동 설정																															
	이미지 프로세싱 데이터 입력 쿼리																															
	웹 페이지 정보 입출력 쿼리																															
	웹 페이지 정보 검색 쿼리																															

iv. 웹사이트

월		12월																														
일		12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31											
웹 사이트	목록 페이지 UI 설계																															
	세부 페이지 UI 설계																															
	정보 수정 페이지 UI 설계																															
	웹 서버 구축																															
	클라이언트 요청에 따른 정보 처리																															
	MySQL 설치 및 환경 설정																															
	수정 페이지에서 입력되는 웹 DB 설계																															

4) 3순위

i. 임베디드

월		12월							1월						
일		29	30	31	1	2	3	4	5	6	7				
임베디드	외부 통신 환경 구축														
	외부 서버로 이미지 스트리밍 전송														
	손가락 움직임 파악														
	파일화된 이미지의 위치와 이름 설정														
	불필요한 파일 삭제 기능														
	영상 캡처 시점의 최적화														

ii. 영상처리

월	12월			1월							
일	29	30	31	1	2	3	4	5	6	7	
영상처리	객체 추출 고도화										
	객체 인식 고도화										
	CAM Shift 영상 얻어오기										
	CAM Shift 객체 분리										
	CAM Shift 객체 판별										
	CAM Shift 정보 비교										

iii. 서버/DB

월	12월			1월							
일	29	30	31	1	2	3	4	5	6	7	
서버/DB	멀티 프로세스 서버로 전환										
	웹 서버 로그인 기능										

iv. 웹사이트

월	12월			1월							
일	29	30	31	1	2	3	4	5	6	7	
웹 사이트	로그인 페이지										
	회원 가입 페이지										
	회원 가입 DB										

5) 마무리

월	12월			1월														
일	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13		
머징 작업																		
발표 준비																		
PPT 제작																		

C. 수행업무 및 담당자

책임	직무	사진	이름	역할 설명
Leader	팀장		김주영	일정 관리 기획서 관리 회의 주제 및 회의록 정리
임베디드	정		김연욱	라즈베리파이, 임베디드 프로그래밍
영상처리	정		김성민	영상처리 프로그래밍
서버/DB	정		김주영	서버/DB 프로그래밍
웹 사이트	정		변소정	웹 프로그래밍

D. 개발 도구

#	이름	사용처
1	C / C++	프로젝트 표준 언어 ① OpenCV 라이브러리 응용
2	C SHARP	FTP서버 제작 ① 처리된 영상 정보를 전송 받고 화면에 출력 ② DB 내의 데이터를 전송 주고 받으며 웹 페이지에 데이터 업로드
3	OpenCV	영상 처리 기능 개발 ① 영상 전처리 알고리즘 ② 객체 Capture 타이밍 제어 ③ 객체 식별 ④ 문자인식
4	Tesser_OCR	문자 인식 ① 영상 전처리 알고리즘 ② 객체 Labeling 알고리즘 ③ 문자영역 UTF-8형식 추출 ④ ANSI형식 변환 후 DB전송
5	Python	임베디드 환경에서의 개발 ① 임베디드 환경에서의 개발 언어 ② 임베디드 환경에서 영상처리를 위한 언어 ③ Chapture 타이밍 제어를 위한 개발 언어
6	PHP	웹 사이트 개발 ① MFC와 동일한 기능 모두 포함 ② DB 검색 기능 ③ User Interface 구현
7	MySQL	데이터베이스 개발 및 제어 ① 데이터베이스 테이블 제작 ② DB에 데이터 입출력 ③ 사용자와의 인터페이스에 출력
8	APACHE	웹 서버 표준 프로토콜 ① 사용자와 웹 페이지 간의 소통

		② 사용자가 원하는 페이지를 보여줌
9	JAVA SCRIPT	웹/ 서버/ 클라이언트 개발환경 ① 웹 브라우저에서 동작을 구현 ② DB연동 ③ 웹 페이지 전환

E. 프로젝트 목표

- 1단계 : 전체 프로젝트 달성을 위한 기본적인 기능들을 구현한다.

업무 분야	기능
임베디드	① 디바이스 조립: 냉장고 내부에 설치할 외부 장치를 만드는 것이 목적이다. A. 기본 보드: Raspberry Pi B. Module : CAM, Bluetooth, Light Sensor ② 모듈 제어: 디바이스에 부착된 모듈을 제어하여 데이터를 처리한다. A. CAM i. 냉장고 내부를 촬영한다 ii. 냉장고 내부의 움직임을 감지한다 B. Bluetooth: 촬영된 이미지를 서버로 전송한다 C. Light Sensor: 빛의 유무를 판별하여, CAM의 작동 여부를 결정한다 i. 빛이 있을 때: CAM이 작동한다. ii. 빛이 없을 때 CAM의 작동이 멈춘다.:
영상 처리	① 영상 전처리 : 객체를 뽑기 위한 영상으로 변환시킨다.
서버/데이터베이스	① 메인 프로세스: 이미지 데이터베이스에서 저장된 이미지 데이터를 영상 처리 기능으로 처리할 수 있는 환경을 구축한다. ② 이미지 데이터베이스: 디바이스가 전송하는 이미지 데이터를 시간 순서대로 받아 저장한다.

- 2단계 : 임베디드 및 영상처리 기능에서 보다 정밀하고 구체적으로 동작할 수 있는 기능들이 추가되어 프로그램의 기능이 고도화되고, 서버의 규모를 늘리고 역할을 세분화 한다. 유저 클라이언트를 개발하여 사용자 환경을 구축한다.

업무 분야	기능
1단계까지 개발된 기능은 기본적으로 포함된다.	
임베디드	① 1단계에서 개발된 기능을 고도화 A. 디바이스 동작 타이밍 세분화 i. 빛 센서를 통해 냉장고 문의 개폐 상태를 감지하여 동작한다 B. 영상 촬영 및 전송 타이밍 세분화 i. 핸드 트래킹 기능을 시스템에 내장하여 냉장고 내부에 사용자의 손이 움직이는 것이 포착되면 이미지 전송을 시작한다. C. 디바이스를 냉장고 내부에 설치가 가능한 상태 조립 상태로 만든다.
영상처리	① 식품 구분 알고리즘: 1단계에서 분할 처리한 객체의 색상을 인식하여 해당 객체의 속성을 분류한다. A. 채소/과일류를 20 종류 이상 분류할 수 있음
서버/데이터베이스	① 웹 서버 구성 A. 디바이스로부터 데이터를 전송 받아 메인 서버의 DB에 전달한다. B. 메인 서버에서 처리된 정보를 유저 클라이언트에 전달한다. ② 통합 서버 환경 A. 2단계에서 추가되는 영상처리 기능이 동작할 수 있도록 한다. ③ 데이터베이스 다양화 A. 식품 데이터베이스 i. 영상처리 기능이 작동할 시 사용자의 디바이스에서 전송되는 데이터의 비교군이 될 수 있는 식품 정보들이 저장되어 있다.

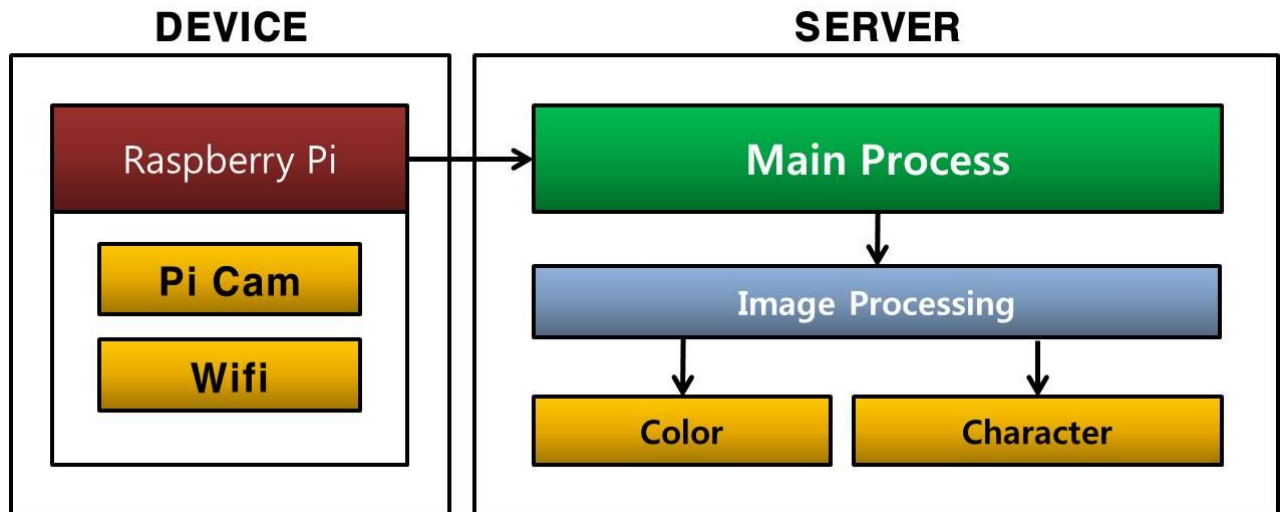
	<ul style="list-style-type: none"> B. 회원 데이터베이스 <ul style="list-style-type: none"> i. 사용자들의 정보가 저장되어 있다. C. 냉장고 데이터베이스 <ul style="list-style-type: none"> i. 회원 데이터베이스의 slave DB이다. ii. 영상처리를 통해 추출된 데이터들이 각 회원들마다 각각 구분되어 저장되어 있다.
웹 사이트	<ul style="list-style-type: none"> ① HTTP 프로토콜 환경에서 작동하는 웹 사이트를 제작한다. A. 개발언어는 PHP를 사용한다. B. GUI: 사용자가 보다 편하게 정보를 습득할 수 있는 GUI 환경을 제공(개발도구는 미정) C. 기능 <ul style="list-style-type: none"> i. 서버에서 영상처리 된 이미지 데이터를 전송 받아 출력한다. ii. 서버에서 냉장고 DB에 저장된 데이터를 전송 받아 출력한다. iii. 전송 받은 이미지를 자체 DB에 임시로 저장한다. iv. 입력 기능을 통해 냉장고 DB의 내용을 교체할 수 있다..

- 3단계 : 유저용 클라이언트가 안드로이드 웹 사이트로 교체되고, DB 검색 기능이 추가된다. 영상 처리 알고리즘의 속도를 높이고 문자인식 기능을 추가한다.

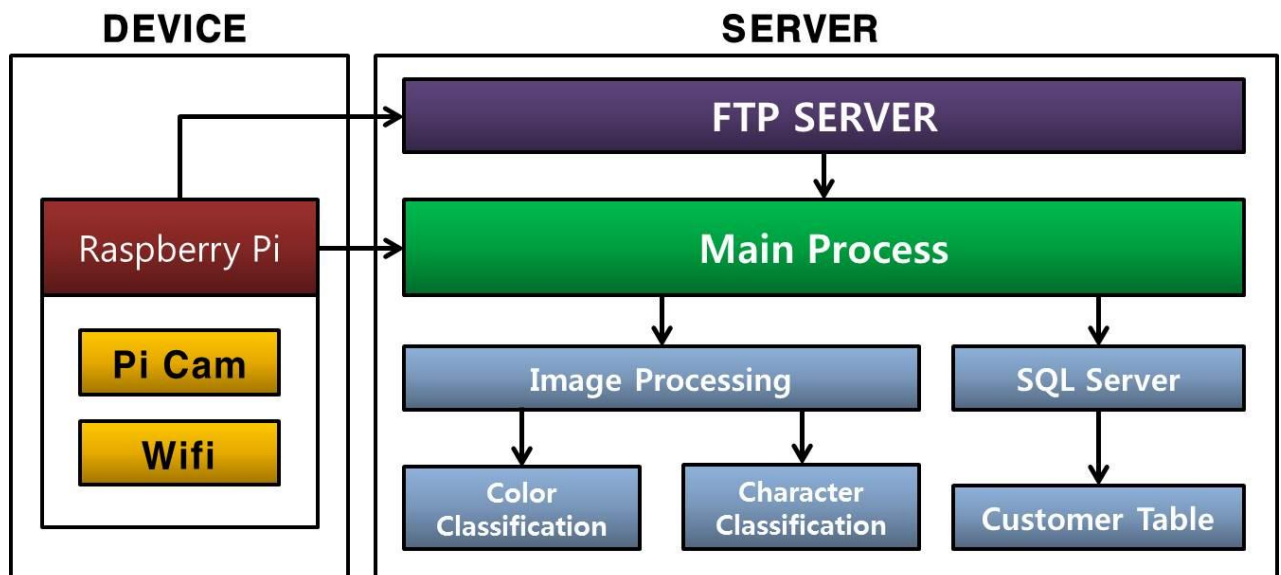
업무 분야	기능
2단계까지 개발된 기능은 기본적으로 포함된다.	
임베디드	추가 개발 사항 없음
영상처리	① 영상 전처리 과정을 거쳐 분할 가능한 환경을 설정한다 ② 문자 영역을 Labeling 해서 영역을 분리한다. ③ 문자 영역에서 문자를 추출한다. ④ 추출된 문자 형식을 변환한다.
서버/데이터베이스	① 로그인 기능: 사용자 클라이언트가 서버에 접속할 경우 로그인 정보를 요구하고, 서버는 이를 확인하여 서버 접속 권한을 부여한다. ② 검색 기능: 유저 클라이언트가 요구하는 정보를 검색하여 전송한다.
유저 클라이언트	① 검색 기능: 사용자가 자신의 냉장고 DB를 검색하여 원하는 정보를 찾을 수 있다.

F. 단계별 아키텍처

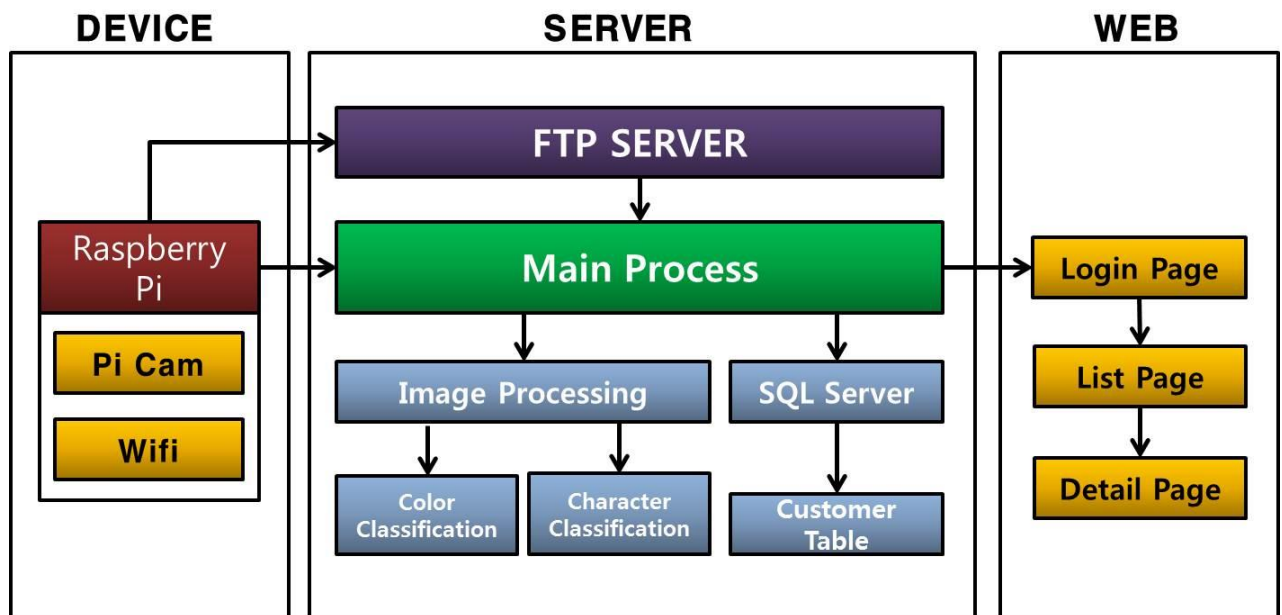
● 1차 아키텍처



● 2차 아키텍처



● 3차 아키텍처



G. 팀 개발일정

11월 셋째 주				
날짜	사용자 인터페이스	영상처리	임베디드	서버/DB
11/17	프로젝트를 위한 Brainstorming			
11/18	조 이름/ 프로젝트 명 결정			
11/19	프로젝트를 위한 기반지식 습득			
11/20	프로젝트를 위한 기반지식과 기능 구현가능성 검토			
11/21	프로젝트를 위한 기반지식과 기능 구현가능성 검토 후 프로젝트 주제 결정			

11월 넷째 주				
날짜	사용자 인터페이스	영상처리	임베디드	서버/DB
11/24	프로젝트 진행 시에 작성하는 보고서에 대한 역할 분담 조원들간의 파일/정보전달을 위한 네트워크 구축			
11/25	프로젝트를 위한 파트 분리 및 각 파트 별 개발담당자 결정			
11/26	프로젝트를 위한 기자재 준비			
11/27	프로젝트 1단계 개발을 위한 파트 별 개발시작			
11/28	프로젝트 목표 재설정 1차 Prototypel발표를 위한 기능간 Merging을 최우선 순위로 함			

12월 첫째 주				
날짜	사용자 인터페이스	영상처리	임베디드	서버/DB
12/1	Android APP개발 - DB연동	Template Matching구현	MMAL driver, GPIO를 python으로 제어	DB서버와 소통을 위한 소켓통신
12/2	Android APP개발 - DB연동 - DB search기능	Template Matching구현	MMAL명령어를 대신하는 MOTION기능 구현	서버의 외부접속 뚫기
12/3	Android APP개발	Template	PID종료를 위한	- 로컬호스트간

	- DB연동 - DB search기능	Matching구현 - 기능 고도화	방법 모색	이미지 전송에 성공 - PYTHON소켓 제작
12/4	Android APP개발 - DB와 image list의 연동	색상인식 - 인식범위 고도화	소켓 접속을 위한 외부 접속문제 해결	Python 클라이언트와 문자 메시지 통신에 성공
12/5	Android와 ISO환경에서 동작하기 위해 웹 페이지 제작으로 기술변경	겹친 물체에 대한 객체 분리 및 색상인식	소켓 접속을 위한 외부 접속문제 해결	Python 소켓을 통한 이미지 파일 전송

12월 둘째 주

날짜	사용자 인터페이스	영상처리	임베디드	서버/DB
12/8	웹 페이지에 대한 설계 구상	색상 인식 세분화 - 색상에 민감하게 반응하도록 재조정	소켓 접속을 위한 외부 접속문제 해결	Python 소켓을 통한 이미지 전송 성공 - 전송된 이미지에 잡음발생
12/9	- 1차 Prototype발표를 위한 기술 Merging - 발표를 위한 PPT제작 - 발표를 위한 시연 준비			
12/10	웹 페이지에 대한 설계 - 인터페이스 구상	색상 인식 세분화 - 색상에 민감하게 반응하도록 재조정	로컬 외부에서 통신이 가능하도록 포트 조정	Python 소켓을 통한 이미지 전송 성공 - 전송된 이미지에 잡음발생
12/11	- 1차 Prototype발표 - 추후 개발 기능 논의			
12/12	- PHP기반 웹 페이지 화면 구성	영상처리를 서버 단에 추가하기 위한 방법모색	이미지 전송을 위한 코드변환 자료조사	이미지 전송을 위한 서버 재 구축 - C#기반의 서버구축

12월 셋째 주

날짜	사용자 인터페이스	영상처리	임베디드	서버/DB
12/15	웹 페이지 제작 - UI제작	C#기반으로 코드변환 방법 모색	이미지 전송을 위한 서버와의 통신 방법 모색	이미지 업로드와 영상처리를 자동화
12/16	웹 페이지 제작 - UI제작	서버에서 자동으로 이미지 processing을 위한 방법 모색	이미지 전송을 위한 서버와의 통신 방법 모색	이미지 업로드와 영상처리를 자동화
12/17	웹 페이지 제작 - UI제작	DLL을 제작하여 서버와 통합 성공	이미지 전송을 위한 서버와의 통신 방법 모색	임베디드와 서버간의 이미지 업로드 방법모색
12/18	웹 페이지 제작 - LIST제작	DLL을 제작하여 서버와 통합 성공	- V4l2 driver 설치 - Mjpg-stream설치 - V4l2에서 제공하는 opencv설치	웹 서버 구축
12/19	웹 페이지 제작 - 웹 페이지의 동작 설정	DLL을 제작하여 서버와 통합 성공	- V4l2 driver 설치 - Mjpg-stream설치 - V4l2에서 제공하는 opencv설치	C#기반의 웹 서버 구현 및 테스트

12월 넷째 주

날짜	사용자 인터페이스	영상처리	임베디드	서버/DB
12/22	웹 페이지의 동작설정	글자인식 구현	캡처 타이밍 최적화	Widow기반 FTP서버로 통신 완성
12/23	웹 페이지와 DB연동 로컬 외부접속 기능	글자인식 구현	캡처 타이밍 최적화	서버와 DB테이블 연동
12/24	웹 페이지와 DB연동 로컬 외부접속 기능	글자인식 구현	-캡처 타이밍 최적화 -디바이스 개별 시리얼넘버를 전송	이미지 LIST를 불러오기
12/25	회원 별 개별DB제작	글자인식 인식률	-캡처 타이밍	Multithreading구현

	회원 ID의 중복확인	고도화 및 인식환경 최적화	최적화 -디바이스 개별 시리얼넘버를 전송	- 디바이스 고유의 시리얼번호를 인식
12/26	서버와 웹 페이지가 DB를 공유 서버와 웹 페이지 연동	글자인식 인식률 고도화 및 인식환경 최적화	- 파이카메라와 openCV연동	서버와 웹 페이지가 DB를 공유 서버와 웹 페이지 연동

12월 넷째 주

날짜	사용자 인터페이스	영상처리	임베디드	서버/DB
12/29	서버와 웹 페이지가 DB를 공유 서버와 웹 페이지 연동	글자인식 인식률 고도화 및 인식환경 최적화	- 파이카메라와 openCV연동	서버와 웹 페이지가 DB를 공유 서버와 웹 페이지 연동
12/30	웹 페이지 UI개선	글자인식 고도화	- 제공되는 영상처리 라이브러리의 한계	디바이스 별로 DB생성 및 수정기능
12/31	웹 페이지 UI개선 및 DB연동	글자인식 고도화	- 파이카메라의 영상처리를 위한 방법 모색	서버 단에서 발생하는 예외처리
1/1	웹 페이지 UI개선 및 DB연동	글자인식 고도화	- 이미지를 서버로 전송	DB연동
1/2	웹 페이지 UI개선 및 DB연동	인식률을 위한 환경조성 및 최적화	캡처 타이밍 최적화 Skin_color_detection	DB연동

12월 넷째 주

날짜	사용자 인터페이스	영상처리	임베디드	서버/DB
1/5	웹 페이지 UI개선 및 DB연동	인식률을 위한 환경조성 및 최적화	타이밍 제어를 위한 시간 최소화	DB연동
1/6	웹 페이지 UI개선 및 DB연동	인식률을 위한 환경조성 및 최적화	-Template matching -motion detection	DB연동

1/7	DB연동와 실시간 업데이트를 위해 서버와 웹 서버를 연동	글자인식과 식자재 식별 함수를 하나의 DLL로 통합	기능합치기	DB연동와 실시간 업데이트를 위해 서버,웹 서버를 연동
1/8	모든 기능 Merging			
1/9	예비 리허설 진행			

3. 개인별 일정

- 개인별 일정은 별도 문서를 첨부합니다

2014 년 11 월 3 주차(11 월 19 일 ~ 11 월 21 일)		
11 월 19 일 수요일		
담당자	업무 이름	업무 내용
김주영	MySQL 제어문 분석	서버 구축을 위한 기술 습득 및 분석
윤지혜	영상분할 알고리즘 분석	영상을 분할하는 다양한 알고리즘을 습득하고 분석
	라이브러리 검색	영상 분할을 할 수 있는 OpenCV 라이브러리 검색 및 분석
	관련 문서/소스 백업	프로젝트 자료 및 소스(영상처리) 백업 작업
김성민	영상 분할 알고리즘 분석	영상을 분할하는 다양한 알고리즘을 습득하고 분석
김연욱 변소정	라즈베리파이 기능 분석	디바이스용 보드의 기능 파악
	라즈베리파이 모듈 분석	디바이스용 보드에 부착될 외부 모듈 제어에 대한 정보 분석
11 월 20 일 목요일		
담당자	업무 이름	업무 내용
김주영	기본 서버 구축	기초적인 서버를 구축하여 데이터 전송 및 DB 베이스 설치에 대한 업무를 파악
윤지혜	영상분할 알고리즘 분석	영상을 분할하는 다양한 알고리즘을 습득하고 분석
	라이브러리 검색	영상 분할을 할 수 있는 OpenCV 라이브러리 검색 및 분석
	관련 문서/소스 백업	프로젝트 자료 및 소스(영상처리) 백업 작업
김성민	영상 분할 알고리즘 분석	영상을 분할하는 다양한 알고리즘을 습득하고 분석
	라이브러리 검색	영상 분할을 할 수 있는 OpenCV 라이브러리 검색 및 분석

김연욱	라즈베리파이 기능 분석	디바이스용 보드의 기능 파악
변소정	라즈베리파이 모듈 분석	디바이스용 보드에 부착될 외부 모듈 제어에 대한 정보 분석
11 월 21 일 금요일		
담당자	업무 이름	업무 내용
김주영	기본 서버 구축	기초적인 서버를 구축하여 데이터 전송 및 DB 베이스 설치에 대한 업무를 파악
윤지혜	템플릿 매칭 소스 분석	템플릿 매칭 관련 사항(논문,자료) 및 소스 분석
	관련 문서/소스 백업	프로젝트 자료 및 소스(영상처리) 백업 작업
김성민	템플릿 매칭 소스 분석	템플릿 매칭 관련 사항(논문,자료) 및 소스 분석
김연욱	라즈베리파이 기능 분석	디바이스용 보드의 기능 파악
변소정	라즈베리파이 모듈 분석	디바이스용 보드에 부착될 외부 모듈 제어에 대한 정보 분석
2014 년 11 월 4 주차(11 월 24 일 ~ 11 월 28 일)		
11 월 24 일 월요일		
담당자	업무 이름	업무 내용
김주영	DB 서버 구축	DB 가 저장될 메인 서버 구축
윤지혜	템플릿 매칭 소스 분석	템플릿 매칭 관련 사항(논문,자료) 및 소스 분석
	관련 문서/소스 백업	프로젝트 자료 및 소스(영상처리) 백업 작업
김성민	템플릿 매칭 소스 분석	템플릿 매칭 관련 사항(논문,자료) 및 소스 분석
김연욱 변소정	Edit Text(검색 입력칸)에서 문자열 입력 받기	검색 입력칸에서 문자열 입력 부분 구현
	Search 버튼 누르면 DB 에서 데이터 검색하기	검색 버튼을 누르면 DB 에서 데이터를 검색하는 부분 구현
	Listview(목록)에서 품목을 클릭하면 품목 정보창 띄우기	목록에서 품목을 클릭하면 품목 정보창 띄우는 기능 구현
	검색 결과가 없음을 알리는 Text 출력	검색 결과가 없음을 알리는 Text 출력 구현

	Button 을 누르면 화면 닫기	버튼을 누르면 화면이 닫혀지는 기능 구현
	이전에 열려 있던 Activity 로 돌아가기	이전에 열려 있던 Activity 로 돌아가는 기능 구현
11 월 25 일 화요일		
담당자	업무 이름	업무 내용
김주영	DB 서버 구축	DB 가 저장될 메인 서버 구축
윤지혜	템플릿 매칭 소스 분석	템플릿 매칭 관련 사항(논문,자료) 및 소스 분석
	매칭 구현	템플릿 매칭 구현
	관련 문서/소스 백업	프로젝트 자료 및 소스(영상처리) 백업 작업
김성민	템플릿 매칭 소스 분석	템플릿 매칭 관련 사항(논문,자료) 및 소스 분석
	매칭 구현	템플릿 매칭 구현
김연욱 변소정	품목을 ListView 로 출력하기	식자재 품목을 리스트로 출력하는 기능 구현
	Listview 에서 항목을 클릭하면 품목 정보창 띄우기	리스트에서 항목을 클릭하면 품목 정보창 띄우기 기능 구현
11 월 26 일 수요일		
담당자	업무 이름	업무 내용
김주영	서버/디바이스 통신 소켓	서버/디바이스간 데이터를 전달하는 소켓을 개발
	냉장고 DB	이미지 프로세싱으로 처리된 데이터를 저장한 데이터베이스
윤지혜	매칭 구현	템플릿 매칭 구현
	관련 문서/소스 백업	프로젝트 자료 및 소스(영상처리) 백업 작업
	DB 이미지 상세화	DB 에 저장될 이미지 상세화
김성민	매칭 구현	템플릿 매칭 구현
	DB 이미지 상세화	DB 에 저장될 이미지 상세화
김연욱 변소정	품목을 ListView 로 출력하기	식자재 품목을 리스트로 출력하는 기능 구현
	Listview 에서 항목을 클릭하면 품목 정보창 띄우기	리스트에서 항목을 클릭하면 품목 정보창 띄우기 기능 구현
	OS 설치	라즈베리파이 OS 설치하기

	영상처리용 라이브러리 설치	라즈베리파이에 영상처리 라이브러리 설치
	블루투스용 라이브러리 설치	라즈베리파이에 블루투스 라이브러리 설치
11 월 27 일 목요일		
담당자	업무 이름	업무 내용
김주영	냉장고 DB	이미지 프로세싱으로 처리된 데이터를 저장한 데이터베이스
	이미지 프로세싱 동작 환경 구축	메인 서버에 영상처리 알고리즘이 동작하는 환경 구축
	냉장고 DB 입출력	냉장고 DB 에 정보 입출력이 될 수 있는 쿼리 개발
	서버 관리자 ID 생성	서버 관리자 ID 생성
윤지혜	매칭 구현	템플릿 매칭 구현
	DB 이미지 상세화	DB 에 저장될 이미지 상세화
	관련 문서/소스 백업	프로젝트 자료 및 소스(영상처리) 백업 작업
김성민	매칭 구현	템플릿 매칭 구현
	DB 이미지 상세화	DB 에 저장될 이미지 상세화
김연욱 변소정	Edit Text 에서 String 정보를 입력 받기	에디트 텍스트에서 문자열 정보 입력받는 기능 구현
	Button1 을 누르면 품목 정보 수정	Button1 을 누르면 품목 정보가 수정되는 기능 구현
	Button2 를 누르면 수정없이 창 닫기	Button2 을 누르면 수정없이 창 닫는 기능 구현
	카메라 연동	라즈베리파이에 카메라 연동하기
	빛 감지센서 연동	라즈베리파이에 빛 감지 센서 연동하기
	블루투스 연결	라즈베리파이에 블루투스 연결하기
11 월 28 일 금요일		
담당자	업무 이름	업무 내용
김주영	이미지 프로세싱 동작 환경 구축	메인 서버에 영상처리 알고리즘이 동작하는 환경 구축

	냉장고 DB 입출력	냉장고 DB 에 정보 입출력이 될 수 있는 쿼리 개발
	서버 관리자 ID 생성	서버 관리자 ID 생성
윤지혜	매칭 구현	템플릿 매칭 구현
	DB 이미지 상세화	DB 에 저장될 이미지 상세화
	관련 문서/소스 백업	프로젝트 자료 및 소스(영상처리) 백업 작업
김성민	매칭 구현	템플릿 매칭 구현
	DB 이미지 상세화	DB 에 저장될 이미지 상세화
김연욱 변소정	Edit Text 에서 String 정보를 입력 받기	에디트 텍스트에서 문자열 정보 입력받는 기능 구현
	Button1 을 누르면 품목 정보 수정	Button1 을 누르면 품목 정보가 수정되는 기능 구현
	Button2 를 누르면 수정없이 창 닫기	Button2 을 누르면 수정없이 창 닫는 기능 구현
	빛 감지센서 연동	라즈베리파이에 빛 감지 센서 연동하기
	블루투스 연결	라즈베리파이에 블루투스 연결하기
2014 년 12 월 1 주차(12 월 1 일 ~ 12 월 5 일)		
12 월 1 일 월요일		
담당자	업무 이름	업무 내용
김주영	어플리케이션 패킷 설계	서버/어플리케이션 통신을 위한 패킷 설계
	어플리케이션 패킷 개발	서버/어플리케이션 통신을 위한 패킷 개발
윤지혜	DB 이미지 상세화	DB 에 저장될 이미지 상세화
	객체 데이터 변환	템플릿 매칭을 통해 얻은 객체 영역을 검출하여 데이터로 변환
	관련 문서/소스 백업	프로젝트 자료 및 소스(영상처리) 백업 작업
김성민	DB 이미지 상세화	DB 에 저장될 이미지 상세화
	객체 데이터 변환	템플릿 매칭을 통해 얻은 객체 영역을 검출하여 데이터로 변환

김연욱 변소정	웹 서버에서 문자 받기 및 확인 Test	웹 서버에서 문자 받기 확인 및 테스트 수행
	웹 서버에서 여러 개의 문자 데이터 받기 및 확인 test	웹 서버에서 여러 개의 문자 데이터 받기 및 확인 Test
	영상 전송 시작 시점 제어	라즈베리파이에서의 영상 전송 시작 시점 제어
	영상 전송 간격 제어	라즈베리파이에서의 영상 전송 간격 제어
	wifi 모듈 제어	라즈베리파이에서의 와이파이 통신 환경 제어
12 월 2 일 화요일		
담당자	업무 이름	업무 내용
김주영	어플리케이션 소켓 설계	서버/어플리케이션 통신을 위한 패킷 설계
	어플리케이션 소켓 개발	서버/어플리케이션 통신을 위한 패킷 개발
	서버/어플리케이션 통신 소켓	서버/어플리케이션 통신을 위한 소켓 개발 및 테스트
윤지혜	객체 데이터 변환	템플릿 매칭을 통해 얻은 객체 영역을 검출하여 데이터로 변환
	관련 문서/소스 백업	프로젝트 자료 및 소스(영상처리) 백업 작업
김성민	객체 데이터 변환	템플릿 매칭을 통해 얻은 객체 영역을 검출하여 데이터로 변환
김연욱 변소정	웹 서버에서 여러 개의 문자 데이터 받기 및 확인 test	웹 서버에서 여러 개의 문자 데이터 받기 및 확인 Test
	영상 전송 시작 시점 제어	라즈베리파이에서의 영상 전송 시작 시점 제어
	영상 전송 간격 제어	라즈베리파이에서의 영상 전송 간격 제어
	wifi 모듈 제어	라즈베리파이에서의 와이파이 통신 환경 제어
12 월 3 일 수요일		
담당자	업무 이름	업무 내용
김주영	서버/어플리케이션 통신 소켓	서버/어플리케이션 통신을 위한 소켓 개발 및 테스트
	웹 서버 구축	메인 서버/ 엽 서버를 분리를 위한 웹 서버 구축

	웹 서버 / 디바이스 통신 연동	웹 서버와 디바이스 소켓 연동
	웹 서버 / 어플리케이션 통신 연동	웹 서버와 어플리케이션 소켓 연동
	어플리케이션 DB / 냉장고 DB 연동	냉장고 DB 의 정보를 어플리케이션 DB 와 연동하여 표시
윤지혜	객체 데이터 변환	템플릿 매칭을 통해 얻은 객체 영역을 검출하여 데이터로 변환
	관련 문서/소스 백업	프로젝트 자료 및 소스(영상처리) 백업 작업
김성민	객체 데이터 변환	템플릿 매칭을 통해 얻은 객체 영역을 검출하여 데이터로 변환
김연욱 변소정	웹 서버에서 이미지 데이터 받기	웹 서버에서 이미지 데이터를 받는 기능 수행(안드로이드)
	영상 전송 시작 시점 제어	라즈베리파이에서의 영상 전송 시작 시점 제어
	영상 전송 간격 제어	라즈베리파이에서의 영상 전송 간격 제어
12 월 4 일 목요일		
담당자	업무 이름	업무 내용
김주영	서버/어플리케이션 통신 소켓	서버/어플리케이션 통신을 위한 소켓 개발 및 테스트
	웹 서버 구축	메인 서버/ 엡 서버를 분리를 위한 웹 서버 구축
	웹 서버 / 디바이스 통신 연동	웹 서버와 디바이스 소켓 연동
	웹 서버 / 어플리케이션 통신 연동	웹 서버와 어플리케이션 소켓 연동
	어플리케이션 DB / 냉장고 DB 연동	냉장고 DB 의 정보를 어플리케이션 DB 와 연동하여 표시
윤지혜	객체 데이터 변환	템플릿 매칭을 통해 얻은 객체 영역을 검출하여 데이터로 변환
	관련 문서/소스 백업	프로젝트 자료 및 소스(영상처리) 백업 작업
김성민	객체 데이터 변환	템플릿 매칭을 통해 얻은 객체 영역을

		검출하여 데이터로 변환
김연욱 변소정	웹 서버에서 이미지 데이터 받기	웹 서버에서 이미지 데이터를 받는 기능 수행(안드로이드)
	wifi 통신 예제/자료 찾기	라즈베리파이에서의 wifi 통신 예제/자료 찾기 및 분석
	wifi 연동 및 확인 test	라즈베리파이에서의 wifi 연동 및 확인 테스트
	오픈된 포트로 웹 서버와 연동 및 Test	라즈베리파이에서 오픈된 포트로 웹 서버와의 연동 및 테스트
12 월 5 일 금요일		
담당자	업무 이름	업무 내용
김주영	웹 서버 구축	메인 서버/ 엽 서버를 분리를 위한 웹 서버 구축
	웹 서버 / 디바이스 통신 연동	웹 서버와 디바이스 소켓 연동
	웹 서버 / 어플리케이션 통신 연동	웹 서버와 어플리케이션 소켓 연동
	어플리케이션 DB / 냉장고 DB 연동	냉장고 DB 의 정보를 어플리케이션 DB 와 연동하여 표시
	ID 생성 및 저장 기능	서버에 접속할 수 있는 ID 를 생성하는 기능
	서버 접근 권한 분할	서버에 접근할 수 있는 권한을 여러 단계로 나눈다
	데이터 외부 입력 기능 개발	외부 입출력에 의해 서버에 저장된 정보를 수정할 수 있는 기능
	데이터 입력 인터페이스 개발	데이터 입출력을 위한 인터페이스
윤지혜	DB 서버와 연동	DB 서버 연동
	관련 문서/소스 백업	프로젝트 자료 및 소스(영상처리) 백업 작업
김성민	DB 서버와 연동	DB 서버 연동
김연욱 변소정	웹 서버에서 문자데이터 받아 DB 에 추가하기	웹 서버에서 문자데이터 받아 DB 에 추가하기(안드로이드)

	웹서버에서 이미지데이터 받아 DB 에 추가하기	웹 서버에서 이미지 데이터 받아 DB 에 추가하기(안드로이드)
	wifi 통신 예제/자료 찾기	라즈베리파이에서의 wifi 통신 예제/자료 찾기 및 분석
	wifi 연동 및 확인 test	라즈베리파이에서의 wifi 연동 및 확인 테스트
	오픈된 포트로 웹 서버와 연동 및 Test	라즈베리파이에서 오픈된 포트로 웹 서버와의 연동 및 테스트
2014 년 12 월 2 주차(12 월 8 일 ~ 12 월 12 일)		
12 월 8 일 월요일		
담당자	업무 이름	업무 내용
김주영	회원 DB 입출력	DB 서버에 있던 이미지 프로세싱 기능을 웹 서버로 이관하여 동작하도록 한다.
	템플릿 DB 생성	템플릿 이미지 저장을 위한 DB 생성
	회원 DB 생성	웹 서버와 DB 서버 연동
	어플리케이션 DB 생성	중간 발표를 위한 머징 작업
윤지혜	DB 서버와 연동	DB 서버 연동
	관련 문서/소스 백업	프로젝트 자료 및 소스 백업 작업
김성민	DB 서버와 연동	DB 서버 연동
김연욱 변소정	DB 목록 수정 가능하게 하기	DB 목록 수정이 가능하게 하기(안드로이드)
	이미지 스트리밍 전송	라즈베리파이에서 웹 서버로 이미지 스트리밍 전송
12 월 9 일 화요일		
담당자	업무 이름	업무 내용
김주영	회원 DB 입출력	DB 서버에 있던 이미지 프로세싱 기능을 웹 서버로 이관하여 동작하도록 한다.
	템플릿 DB 생성	템플릿 이미지 저장을 위한 DB 생성
	회원 DB 생성	웹 서버와 DB 서버 연동
	어플리케이션 DB 생성	중간 발표를 위한 머징 작업
윤지혜	DB 서버와 연동	DB 서버 연동

	관련 문서/소스 백업	프로젝트 자료 및 소스 백업 작업
김성민	DB 서버와 연동	DB 서버 연동
김연욱	DB 목록 수정 가능하게 하기	DB 목록 수정이 가능하게 하기(안드로이드)
변소정	이미지 스트리밍 전송	라즈베리파이에서 웹 서버로 이미지 스트리밍 전송
12 월 10 일 수요일		
담당자	업무 이름	업무 내용
김주영	1 차 버전 머징 및 발표 준비	현재 시점까지 개발된 기능 머징, 중간 발표 준비
윤지혜		
김성민		
김연욱		
변소정		
12 월 11 일 목요일		
담당자	업무 이름	업무 내용
김주영	1 차 버전 머징 및 발표 준비	현재 시점까지 개발된 기능 머징, 중간 발표 준비
윤지혜		
김성민		
김연욱		
변소정		
12 월 12 일 금요일		
담당자	업무 이름	업무 내용
김주영	1 차 버전 머징 및 발표 준비	현재 시점까지 개발된 기능 머징, 중간 발표 준비
윤지혜		
김성민		
김연욱		
변소정		

2014 년 12 월 3 주차(12 월 15 일 ~ 12 월 19 일)		
12 월 15 일 월요일		
담당자	업무 이름	업무 내용
김주영	1 차 버전 기능 최종 확인	1 차 버전으로 완성된 기능들을 정리 및 확인한다
김성민	객체 검출 라이브러리 검색	객체구분을 위한 라이브러리를 검색한다.
김연욱	영상처리용 라이브러리 설치	리눅스 환경에서 영상처리를 활용하기 위한 라이브러리 조사와 설치를 한다.
변소정	웹 페이지 UI 작성 웹 서버 구축 및 DB 연동	1 차 UI: 목록 페이지, 세부 페이지, 수정 페이지 웹 서버를 구축하고 DB 와 연동할 수 있도록 한다.
12 월 16 일 화요일		
담당자	업무 이름	업무 내용
김주영	1 차 버전 기능 최종 확인	1 차 버전으로 완성된 기능들을 정리 및 확인한다
김성민	객체 검출 라이브러리 검색 예제 코드 분석 및 간결화	객체구분을 위한 라이브러리를 검색한다. 예제코드를 분석하여 사용가능성 여부를 판별한다.
김연욱	파이카메라를 위한 환경구축	파이카메라를 C 언어로 코딩하기 위한 환경구축과 자료조사를 한다
변소정	웹 페이지 UI 작성 웹 서버 구축 및 DB 연동	1 차 UI: 목록 페이지, 세부 페이지, 수정 페이지 웹 서버를 구축하고 DB 와 연동할 수 있도록 한다.
12 월 17 일 수요일		
담당자	업무 이름	업무 내용
김주영	DB 서버 구축	Window SQL SERVER 2012 express 를 사용한 서버 환경 구축

김성민	객체 검출 라이브러리 검색 예제 코드 분석 및 간결화	객체 구분을 위한 라이브러리를 검색한다. 예제코드를 분석하여 사용가능성 여부를 판별한다.
김연욱	파이카메라를 활용한 영상촬영과 영상파일화	카메라로 캡처하고 이미지파일을 저장하기 위한 자료조사와 TEST 를 진행한다
변소정	웹 페이지 UI 작성 웹 어플리케이션의 처리 프로그램 코딩	1 차 UI: 목록 페이지, 세부 페이지, 수정 페이지 사용자 요청에 따른 웹 어플리케이션 서버의 처리를 코딩한다.
12 월 18 일 목요일		
담당자	업무 이름	업무 내용
김주영	DB 서버 구축	Window SQL SERVER 2012 epress 를 사용한 서버 환경 구축
김성민	예제 코드 분석 및 간결화 기존 코드 머징	예제코드를 분석하여 사용가능성 여부를 판별한다. 기존에 작성된 코드에 포함시켜 기능을 고도화한다.
김연욱	파이카메라를 사용한 스트리밍 촬영과 스트리밍중 캡처	원하는 순간에 캡처하여 파일화한다.
변소정	웹 페이지 UI 작성 웹 어플리케이션의 처리 프로그램 코딩	1 차 UI: 목록 페이지, 세부 페이지, 수정 페이지 사용자 요청에 따른 웹 어플리케이션 서버의 처리를 코딩한다.
12 월 19 일 금요일		
담당자	업무 이름	업무 내용
김주영	이미지 프로세싱 정보의 데이터 베이스 입력 쿼리개발	DB 서버에 이미지 프로세싱을 통해 추출된 유저 데이터를 입출력 할 수 있는 쿼리문을 제작한다

김성민	예제 코드 분석 및 간결화 기존 코드 머징	예제코드를 분석하여 사용가능성 여부를 판별한다. 기존에 작성된 코드에 포함시켜 기능을 고도화한다.
김연욱	촬영간격을 좁게하기	촬영간격을 좁히고 촬영시점을 제어한다.
변소정	웹 페이지 UI 작성 웹 어플리케이션의 처리 프로그램 코딩	1 차 UI: 목록 페이지, 세부 페이지, 수정 페이지 사용자 요청에 따른 웹 어플리케이션 서버의 처리를 코딩한다.
2014 년 12 월 4 주차(12 월 22 일 ~ 12 월 26 일)		
12 월 22 일 월요일		
담당자	업무 이름	업무 내용
김주영	웹 서버와 DB 서버 연동	웹 서버와 DB 서버를 연동시키니다
김성민	객체와 객체를 구분 방법 모색 라이브러리 검색 및 구현	객체와 객체 관계를 구분할수 있는 방법을 모색한다.
김연욱	영상 내 색상검출	실시간으로 색상을 검출하여 skin color 가 들어오는 시점을 확인한다.
변소정	웹 페이지 UI 작성 회원가입 포맷 DB 작성	2 차 UI: 로그인 페이지, 회원가입 페이지 회원가입 포맷에 맞추어 DB 를 작성한다: 아이디, 패스워드, 디바이스 시리얼 넘버, 이메일 주소
12 월 23 일 화요일		
담당자	업무 이름	업무 내용
김주영	웹 서버 명령을 통한 입출력 기능 개발	웹 서버의 명령을 통해 DB 서버 내의 정보를 입출력하는 쿼리와 코드를 개발

김성민	객체와 객체를 구분 방법 모색 라이브러리 검색 및 구현	객체와 객체 관계를 구분할수 있는 방법을 모색한다.
김연욱	색상 tracking	움직이는 손의 동선을 파악하기 위해 skin color 을 tracking 한다
변소정	웹 페이지 UI 작성 웹 어플리케이션의 처리 프로그램 코딩 및 DB 연동	2 차 UI: 로그인 페이지, 회원가입 페이지 로그인 요청에 따른 처리 코딩, 회원가입 요청에 따른 처리 코딩, 회원가입 페이지에서 입력한 값들에 대한 처리와 DB 연동
12 월 24 일 수요일		
담당자	업무 이름	업무 내용
김주영	웹 서버 명령을 통한 입출력 기능 개발	웹 서버의 명령을 통해 DB 서버 내의 정보를 입출력하는 쿼리와 코드를 개발
김성민	라이브러리 검색 및 구현	객체와 객체를 구분할수 있도록 제공된 라이브러리 검색한다.
김연욱	색상 tracking	움직이는 손의 동선을 파악하기 위해 skin color 을 tracking 한다
변소정	웹 어플리케이션의 처리 프로그램 코딩 및 DB 연동	로그인 요청에 따른 처리 코딩, 회원가입 요청에 따른 처리 코딩, 회원가입 페이지에서 입력한 값들에 대한 처리와 DB 연동
12 월 25 일 목요일		
담당자	업무 이름	업무 내용
김주영	서버 안정화	웹, DB, FTP 서버가 안정적으로 연동되어 있는 지 확인 작업
김성민	라이브러리 검색 및 구현	객체와 객체를 구분할수 있도록 제공된 라이브러리 검색한다.

김연욱	손의 IN/OUT 시점확인	손이 움직이는 동선에서 손이 냉장고로 손이 들어오는 동작과 냉장고에서 손이 나가는 동작을 분류한다.
변소정	웹 어플리케이션의 처리 프로그램 코딩 및 DB 연동	로그인 요청에 따른 처리 코딩, 회원가입 요청에 따른 처리 코딩, 회원가입 페이지에서 입력한 값들에 대한 처리와 DB 연동
12 월 26 일 금요일		
담당자	업무 이름	업무 내용
김주영	서버 가동 설정	서버가 별도의 외부 동작없이 계속하여 작동하도록 설정한다.
김성민	기존 코드 머징	기존코드에 삽입하여 사용가능성 여부를 판별한후 추가한다.
김연욱	손의 IN/OUT 시점확인	손이 움직이는 동선에서 손이 냉장고로 손이 들어오는 동작과 냉장고에서 손이 나가는 동작을 분류한다.
변소정	웹 어플리케이션의 처리 프로그램 코딩 및 DB 연동	로그인 요청에 따른 처리 코딩, 회원가입 요청에 따른 처리 코딩, 회원가입 페이지에서 입력한 값들에 대한 처리와 DB 연동
2014 년 12 월 5 주차(12 월 29 일 ~ 1 월 2 일)		
12 월 29 일 월요일		
담당자	업무 이름	업무 내용
김주영	웹 페이지용 검색 쿼리 개발	웹 페이지에서 DB 서버 내의 정보를 검색할 수 있는 기능을 제작
김성민	속도 향상 방안 검색 및 구현	구현된 기능에 대한 속도향상을 위한 방법을 모색한다.
김연욱	손가락의 움직임파악	새로들어온 식자재와 나간 식자재의 구분을 명확하게 하기

		위해 손가락으로 물건을 놓고 집는 순간을 확인한다.
변소정	웹 서버, 웹 어플리케이션 서버 이동 및 설정 변경	웹 서버나 DB, 웹 어플리케이션 서버를 작업 컴퓨터에서 서버 컴퓨터로 이동시키고 설정을 변경
12 월 30 일 화요일		
담당자	업무 이름	업무 내용
김주영	멀티 프로세스 서버 개발	다수의 디바이스가 접속할 수 있는 멀티 프로세스 서버 개발
김성민	속도 향상 방안 검색 및 구현	구현된 기능에 대한 속도향상을 위한 방법을 모색한다.
김연욱	손가락의 움직임파악	새로 들어온 식자재와 나간 식자재의 구분을 명확하게 하기 위해 손가락으로 물건을 놓고 집는 순간을 확인한다.
변소정	웹 서버, 웹 어플리케이션 서버 이동 및 설정 변경	웹 서버나 DB, 웹 어플리케이션 서버를 작업 컴퓨터에서 서버 컴퓨터로 이동시키고 설정을 변경
12 월 31 일 수요일		
담당자	업무 이름	업무 내용
김주영	멀티 프로세스 서버 개발	다수의 디바이스가 접속할 수 있는 멀티 프로세스 서버 개발
김성민	속도 향상 방안 검색 및 구현	구현된 기능에 대한 속도향상을 위한 방법을 모색한다.
김연욱	영상파일의 위치와 이름변경	captured 된 사진이 들어온 식자재인지 나간 식자재인지 파일명으로 확인하게 한다.
변소정	웹 서버, 웹 어플리케이션 서버 이동 및 설정 변경	웹 서버나 DB, 웹 어플리케이션 서버를 작업 컴퓨터에서 서버 컴퓨터로 이동시키고 설정을 변경

1 월 1 일 목요일		
담당자	업무 이름	업무 내용
김주영	멀티 쓰레드 서버 개발	다수의 디바이스가 접속할 수 있는 멀티 쓰레드 서버 개발
김성민	프로젝트 전체 머징 오류 검출/ 디버깅	임베디드, 영상처리, 네트워크 각각 구현된 기능들을 합쳐보고 그에 따른 오류 및 변경사항을 검사한다.
김연욱	파일삭제 기능 추가	원하는 이미지파일을 남기고 보내진 파일은 삭제하는 기능을 추가한다.
변소정	프로젝트 전체 머징에 따른 웹 사이트 정상 작동 확인	실제 프로젝트에서 웹 사이트가 정상 작동하는지 확인하고 수정하기
1 월 2 일 금요일		
담당자	업무 이름	업무 내용
김주영	멀티 쓰레드 서버 개발	다수의 디바이스가 접속할 수 있는 멀티 쓰레드 서버 개발
김성민	프로젝트 전체 머징 오류 검출/ 디버깅	임베디드, 영상처리, 네트워크 각각 구현된 기능들을 합쳐보고 그에 따른 오류 및 변경사항을 검사한다.
김연욱	영상을 캡처하는 시간을 단축하기 위한 자료조사	캡처 간격을 좁힌다.
변소정	프로젝트 전체 머징에 따른 웹 사이트 정상 작동 확인	실제 프로젝트에서 웹 사이트가 정상 작동하는지 확인하고 수정하기
2015 년 1 월 2 주차(1 월 5 일 ~ 1 월 9 일)		
1 월 5 일 월요일		
담당자	업무 이름	업무 내용
김주영	웹 서버 로그인 기능	웹 서버에서 서버로 접근할 때 로그인 기능알 사용하도록 기능을 개발한다

김성민	프로젝트 전체 머징 오류 검출/ 디버깅 문서작성	임베디드, 영상처리, 네트워크 각각 구현된 기능들을 합쳐보고 그에 따른 오류 및 변경사항을 검사한다. 프로젝트에 대한 발표준비 및 기획서를 작성한다.
김연욱	최적화	촬영된 이미지 파일이 잘보이도록 촬영시점과 촬영 위치등을 재정비 한다.
변소정	프로젝트 전체 머징에 따른 웹 사이트 정상 작동 확인	실제 프로젝트에서 웹 사이트가 정상 작동하는지 확인하고 수정하기
1 월 6 일 화요일		
담당자	업무 이름	업무 내용
김주영	웹 서버 로그인 기능	웹 서버에서 서버로 접근할 때 로그인 기능알 사용하도록 기능을 개발한다
김성민	프로젝트 전체 머징 오류 검출/ 디버깅 문서작성	임베디드, 영상처리, 네트워크 각각 구현된 기능들을 합쳐보고 그에 따른 오류 및 변경사항을 검사한다. 프로젝트에 대한 발표준비 및 기획서를 작성한다.
김연욱	최적화	촬영된 이미지 파일이 잘보이도록 촬영시점과 촬영 위치등을 재정비 한다.
변소정	문서 작성	발표 ppt 작성, 추합하기. 대본 작성, 기타 제출 서류 작성 및 확인
1 월 7 일 수요일		
담당자	업무 이름	업무 내용
김주영	프로젝트 전체 머징 오류 검출/ 디버깅 문서작성	임베디드, 영상처리, 네트워크 각각 구현된 기능들을 합쳐보고 그에 따른 오류 및 변경사항을 검사한다.

		프로젝트에 대한 발표준비 및 기획서를 작성한다.
김성민	프로젝트 전체 머징 오류 검출/ 디버깅 문서작성	임베디드, 영상처리, 네트워크 각각 구현된 기능들을 합쳐보고 그에 따른 오류 및 변경사항을 검사한다. 프로젝트에 대한 발표준비 및 기획서를 작성한다.
김연욱	외부포트로 통신하기	로컬 밖으로 통신하기 위한 포트를 연다
변소정	문서 작성	발표 ppt 작성, 추합하기. 대본 작성, 기타 제출 서류 작성 및 확인
1 월 8 일 목요일		
담당자	업무 이름	업무 내용
김주영	프로젝트 전체 머징 오류 검출/ 디버깅 문서작성	임베디드, 영상처리, 네트워크 각각 구현된 기능들을 합쳐보고 그에 따른 오류 및 변경사항을 검사한다. 프로젝트에 대한 발표준비 및 기획서를 작성한다.
김성민	프로젝트 전체 머징 오류 검출/ 디버깅 문서작성	임베디드, 영상처리, 네트워크 각각 구현된 기능들을 합쳐보고 그에 따른 오류 및 변경사항을 검사한다. 프로젝트에 대한 발표준비 및 기획서를 작성한다.
김연욱	외부포트로 통신하기	로컬 밖으로 통신하기 위한 자료조사를 하고 기능을 추가한다.
변소정	문서 작성	발표 ppt 작성, 추합하기. 대본 작성, 기타 제출 서류 작성 및 확인

1 월 9 일 금요일		
담당자	업무 이름	업무 내용
김주영	기능 합치기	서버와 데이터베이스, 영상처리와 기능을 합치고 예기치 않게 발생하는 문제들을 해결한다.
김성민		
김연욱		
변소정		
2015 년 1 월 3 주차(1 월 12 일 ~ 1 월 16 일)		
1 월 12 일 월요일		
김주영	발표준비	발표준비를 위해 기능을 모으고 PPT 를 준비하며, 시연할 기능들을 최종으로 TEST 한다.
김성민		
김연욱		
변소정		
1 월 13 일 화요일		
김주영	발표준비	발표준비를 위해 기능을 모으고 PPT 를 준비하며, 시연할 기능들을 최종으로 TEST 한다.
김성민		
김연욱		
변소정		
1 월 14 일 화요일		
김주영	발표 및 시연	최종 발표
김성민		
김연욱		
변소정		

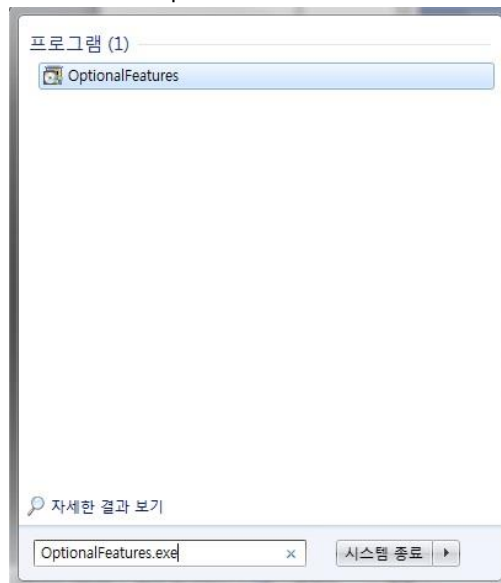
4. 프로젝트 배경 기술/ 지식/ 알고리즘

A. Server/Client/DB

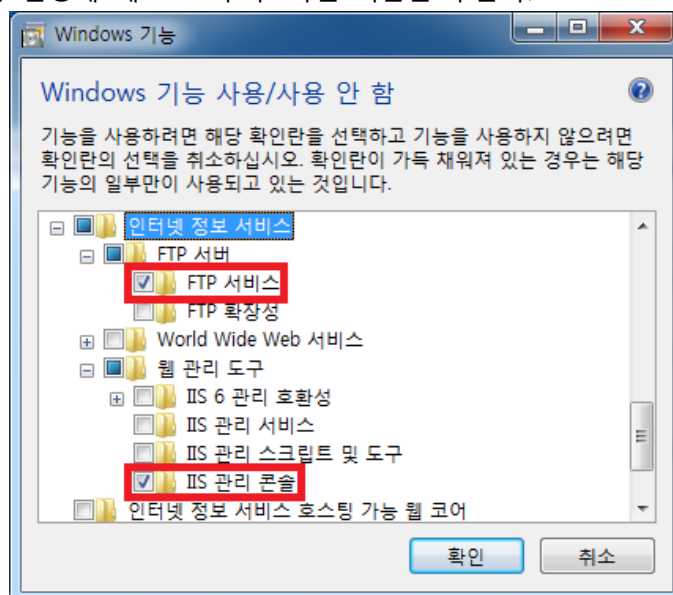
i. 서버 환경 구축

1. Windows7 IIS(Internet Information Service) FTP 서버 설정

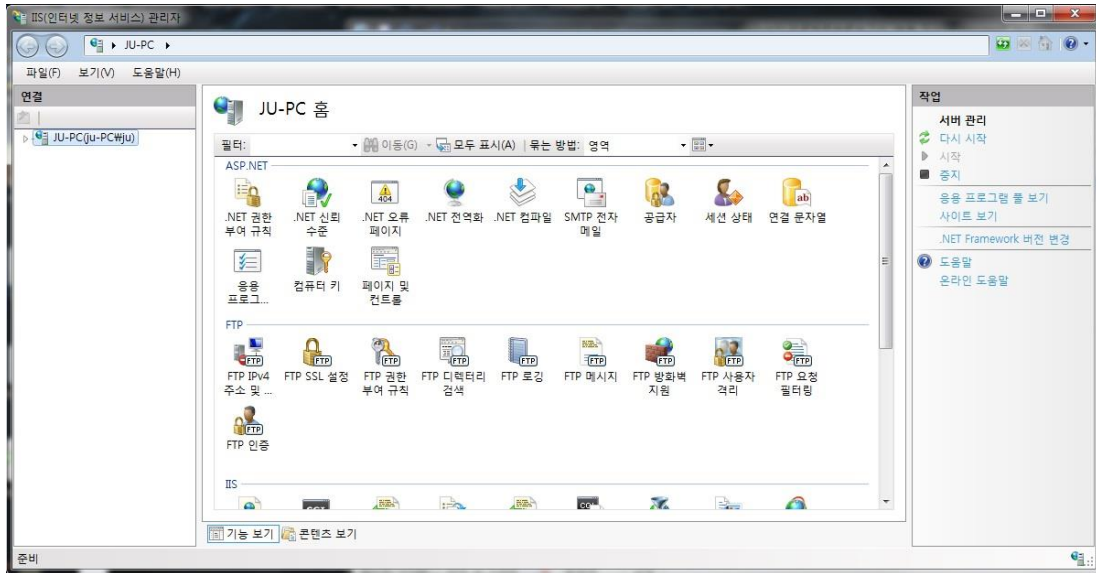
- 처음으로 시작 메뉴 - 검색 창에 'OptionalFeatures.exe'를 입력하고 엔터를 입력한다.



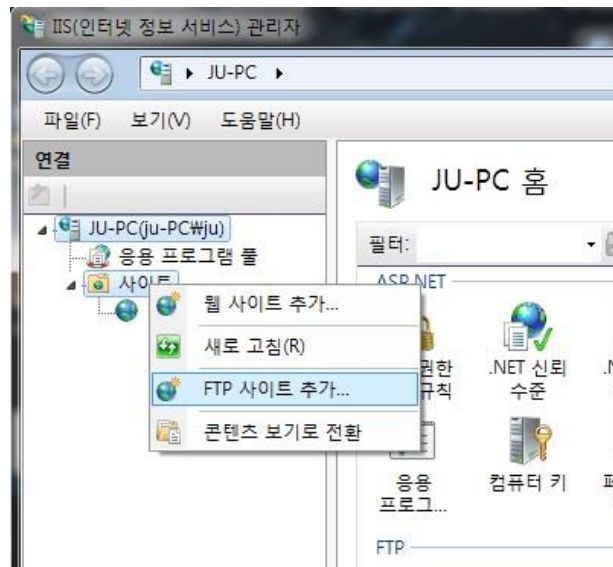
- 아래의 2가지를 설정에 체크 표시 후 확인 버튼을 누른다.



- 다시 시작 메뉴 - 검색창에 'inetmgr.exe'를 입력하고, 엔터를 입력한다

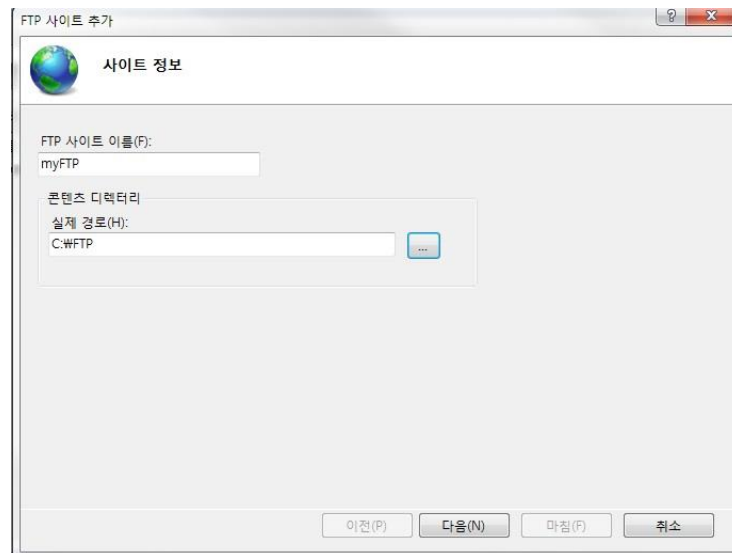


위와 같은 창이 발생하면 우측의 내 PC 아이디를 클릭하며 하단 목록을 활성화 한다.

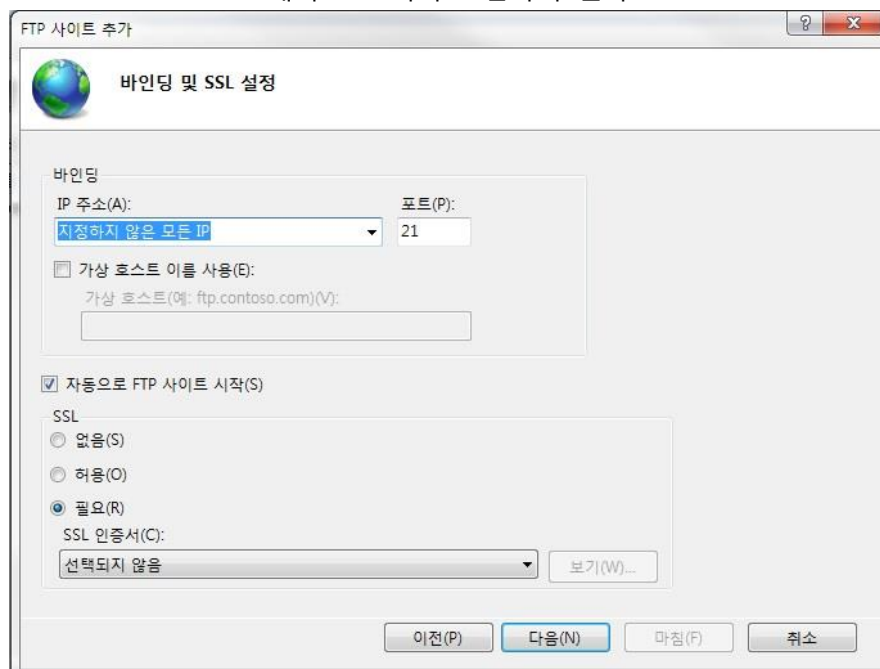


하단 목록의 사이트 탭을 우클릭하면 위의 메뉴가 발생한다.

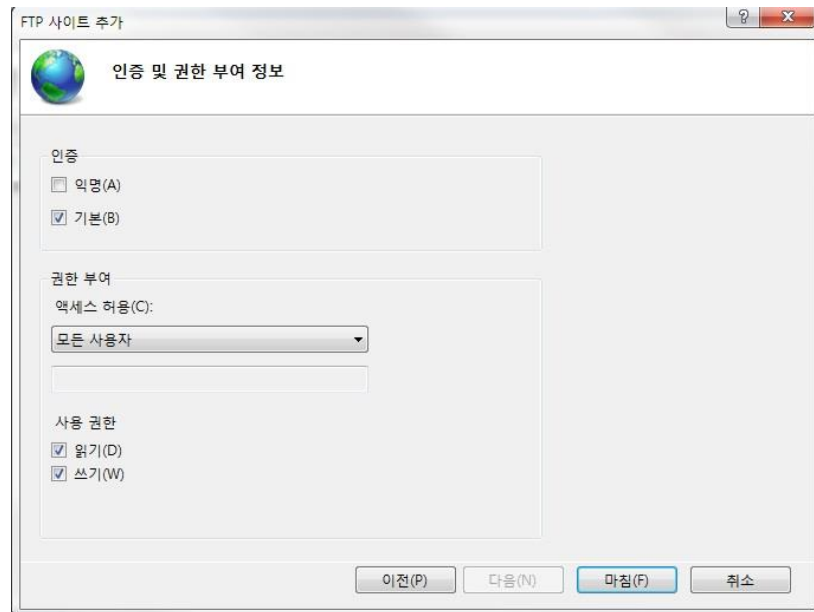
- 원하는 이름과 FTP서버가 사용할 파일 디렉토리를 설정한 후 다음 버튼을 누른다



바인딩 주소를 '지정하지 않은 모든 IP'로 설정해야 외부의 네트워크들이 접근할 수 있다. 또는 원하는 특정 아이피 주소를 입력해줘야 한다. 포트 칸에 원하는 포트 번호를 지정해주면 해당 포트 번호를 통해서 FTP 서버로 접속이 된다.

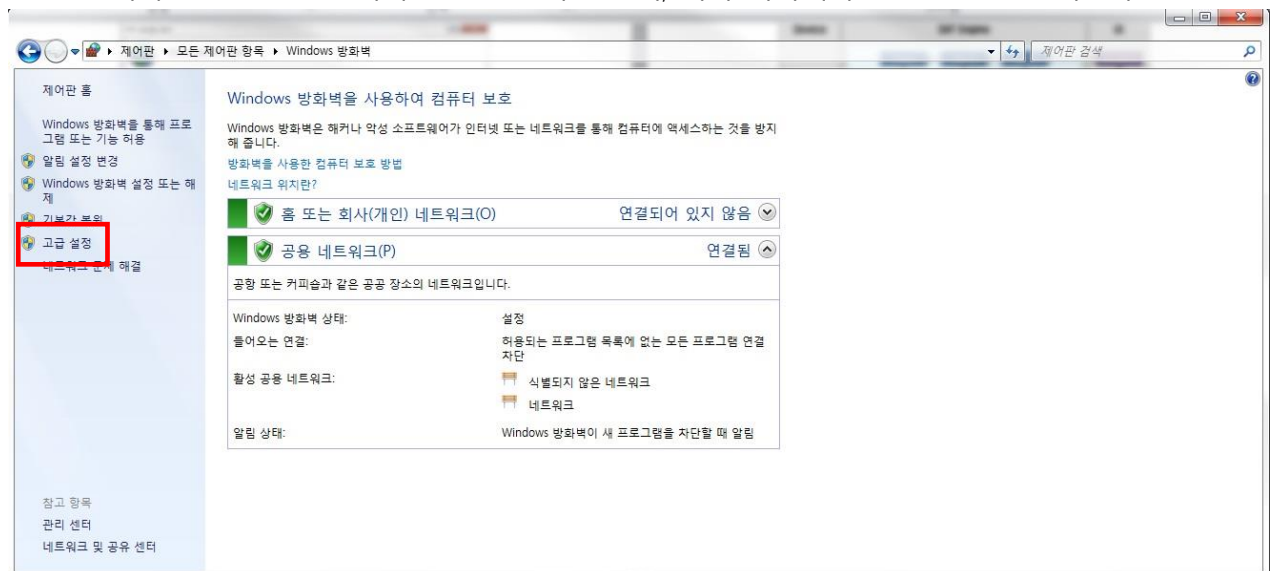


인증과 액세스 및 사용 권한을 설정한다.

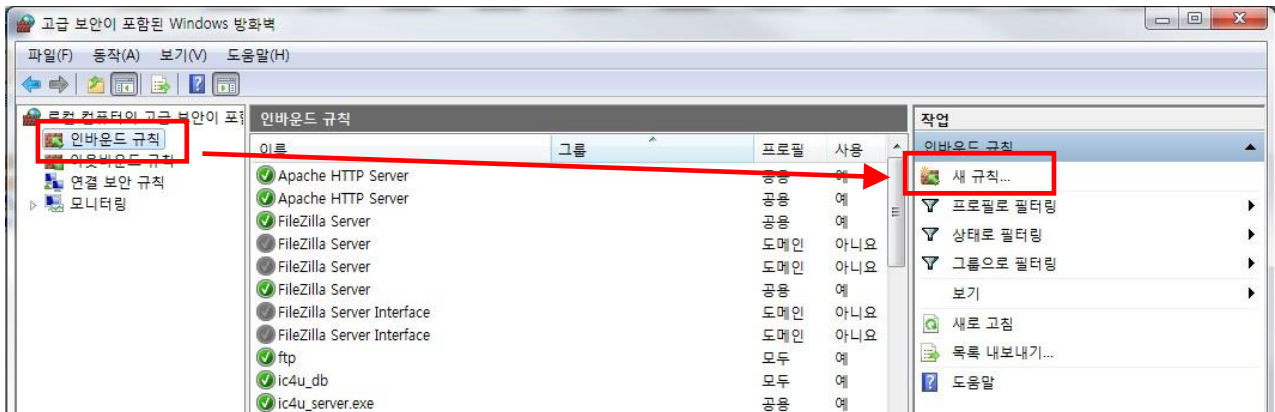


ii. 방화벽 설정 : 서버의 포트 관리를 위한 설정 방법

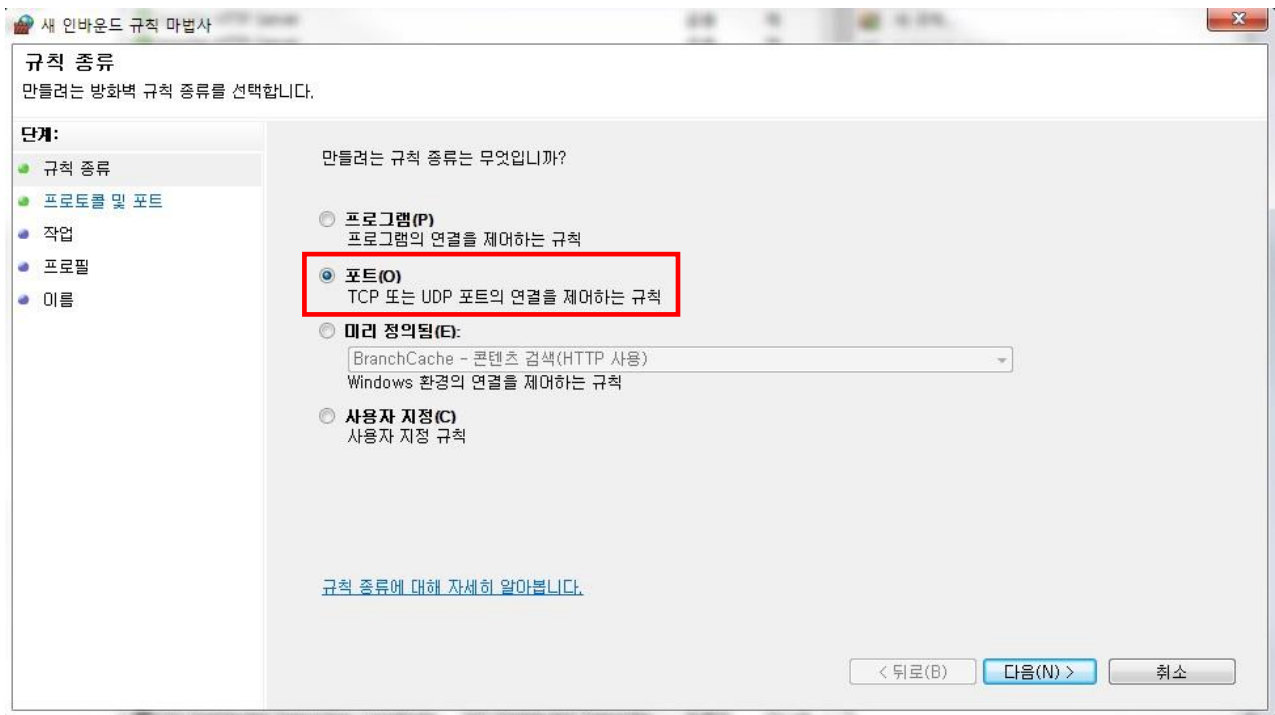
- 제어판 - windows방화벽 설정으로 이동한 뒤, 좌측 목록에서 고급 설정을 클릭한다.



- 고급 설정으로 이동 후, 좌측 목록에서 인바운드 규칙을 클릭한다. 우측에 새로운 메뉴가 나타나면 '새 규칙'을 클릭한다.



- 아래의 메뉴 창이 새로 발생하면, 원하는 포트 번호와 형식을 설정하여, 해당 포트를 설정할 수 있다.



iii. 포트 포워딩을 통한 외부 접속 설정

- 웹 브라우저의 주소 입력 창에 '192.168.0.1' 주소를 입력하면 게이트웨이를 설정할 수 있는 관리도구에 접속할 수 있다.



- 고급 설정 - NAT/라우터 관리 - 포트포워드 설정 순서대로 클릭하면 아래의 메뉴가 화면에 발생한다. 설정할 서버의 주소와 내/외부 포트 번호를 입력하고 프로토콜을 설정한 뒤 추가 버튼을 누르고, 우측 상단의 저장 버튼을 누르면 해당 서버의 포트 포워드 설정이 추가된다. 설정이 완료된 후에는 현재 장소의 외부 아이피 + 포트포워드에 설정한 외부 포트 번호를 함께 입력하면 외부에서도 서버에 접근할 수 있다.



B. 서버 프로그래밍 기술

- i. C#의 FtpWebRequest 클래스를 이용한 FTP 데이터 전송
- 참고 문헌([http://msdn.microsoft.com/ko-kr/library/456dfw4f\(v=vs.110\).aspx](http://msdn.microsoft.com/ko-kr/library/456dfw4f(v=vs.110).aspx))
- System.Net 네임스페이스에 포함되어 있다.

```

public void FtpImageDownload()
{
    string localPath = @"C:\server\";
    string fileName = "abcd123456in01.jpg";

    ServicePointManager.Expect100Continue = false;
    ServicePointManager.MaxServicePointIdleTime = 2000;

    Ftp 연결 요청을 위한 객체 선언
    FtpWebRequest requestFileDownload = (FtpWebRequest)WebRequest.Create("ftp://14.32.7.93/" + fileName);
    requestFileDownload.Credentials = new NetworkCredential(); 자격 증명을 설정을 위한 인스턴스
    requestFileDownload.Method = WebRequestMethods.Ftp.DownloadFile; 서버에 보낼 명령을 설정

    FtpWebResponse responseFileDownload = (FtpWebResponse)requestFileDownload.GetResponse(); 서버의 응답이 반환되어 저장

    Stream responseStream = responseFileDownload.GetResponseStream();
    FileStream writeStream = new FileStream(localPath + fileName, FileMode.Create); 파일읽기 스트림 생성

    int Length = 2048;
    Byte[] buffer = new Byte[Length];
    int bytesRead = responseStream.Read(buffer, 0, Length); 버퍼 2048 바이트 크기만큼 스트림으로부터 계속하여 정보를 읽어와 저장

    while (bytesRead > 0)
    {
        writeStream.Write(buffer, 0, bytesRead);
        bytesRead = responseStream.Read(buffer, 0, Length);
    }

    responseStream.Close();
    writeStream.Close();
}

```

ii. C#의 SqlConnection 네임스페이스를 이용한 MSSQL과의 연동

- 참고문헌([http://msdn.microsoft.com/ko-kr/library/System.Data.SqlClient\(v=vs.110\).aspx](http://msdn.microsoft.com/ko-kr/library/System.Data.SqlClient(v=vs.110).aspx))
- System.Data.SqlClient 네임스페이스에 존재한다.
- .NET Framework에서 제공하는 SQL 서버의 데이터 제공자이다.

```
public void FreezerDataTableCreator(string srlNum)
{
    string serialNum = srlNum;
    string CreateDefTable = @"CREATE TABLE [" + serialNum + "](idx int identity(1,1) not null, List_day int not null, "
    데이터베이스에 입력할 Query 입력
    + "List_term int not null, List_name nvarchar(50) not null, "
    + "List_amount int default 0, List_exp int not null);";

    SqlConnection conn = new SqlConnection(); SQL SERVER와 연결을 위한 SqlConnection 객체 선언
    conn.ConnectionString = @"Server=(local);Database=IC4U;UID=sa;PWD=admin1;"; 연결한 SQL SERVER의 주소, DB 이름, ID, 패스워드
    conn.Open();

    SqlCommand cmd = new SqlCommand(); 연결한 SQL SERVER에 명령을 보내기 위한 SqlCommand 클래스 선언
    cmd.Connection = conn; 명령을 보낼 경로를 SqlConnection으로 오픈한 경로로 설정한다.

    cmd.CommandText = CreateDefTable; 전송할 command를 입력 (string 형식으로 입력한다.
    cmd.ExecuteNonQuery(); Command 실행.

    conn.Close();
}
```

- SqlConnection 클래스 객체를 선언한다. 이 클래스의 주 목적은 현재 작성 중인 프로그램과 SQL SERVER의 연결을 하는 것이다. SqlConnection 클래스 내부에 Connection 메서드에 연결한 SQL SERVER의 주소와 DB이름, ID, Password를 입력한 뒤, Open 메서드를 실행하면 해당 서버와 연결이 된다.
- SqlCommand는 SQL SERVER로 명령을 전송 및 실행시키기 위한 메서드들이 함축되어 있는 클래스이다. 선언 뒤 내부의 Connection 메서드를 위에서 선언된 SqlConnection 클래스를 대입해주면, 명령문을 보내는 SQL SERVER로 지정된다.
- SqlCommand의 CommandText 메서드에 서버로 전송할 쿼리문을 입력한다. String 형식으로 작성된 변수를 대입하거나 또는 직접 문장을 입력한다.
- SqlCommand 내부의 ExecuteNonQuery 메서드를 실행하면, 위에서 입력한 쿼리문이 서버로 전송되어 실행된다.

C. Raspberry Pi

① 라즈베리파이

i. 개발환경

- Raspbian : 리눅스 기반의 라즈베리파이에 제공되는 기본 OS

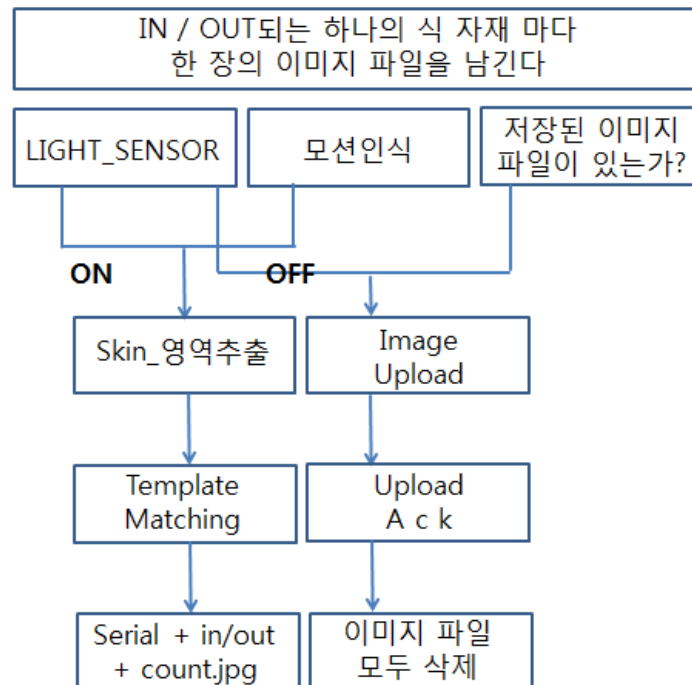
ii. 개발 툴

- V4L2 : Raspberry 환경에서 카메라를 사용하기 위한 통합 드라이버.
- MMAL driver : 카메라의 사용을 위한 driver로 terminal상에서 사용가능.
- FFMPEG : 동영상 코덱이 없는 Raspberry pi에서 동영상 인코딩을 위한 driver.
- OpenCV : 영상처리를 사용하기 위한 라이브러리.
- IDLE : python언어의 개발을 위한 python언어 편집기

iii. 개발 언어

- Python : 리눅스 환경에서 python은 별도의 컴파일러를 필요로 하지 않음.

iv. 알고리즘

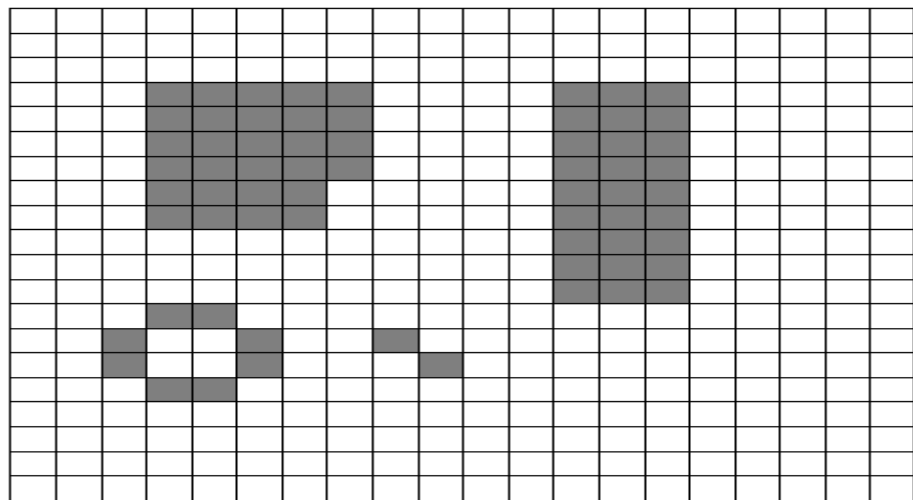


D. Image processing

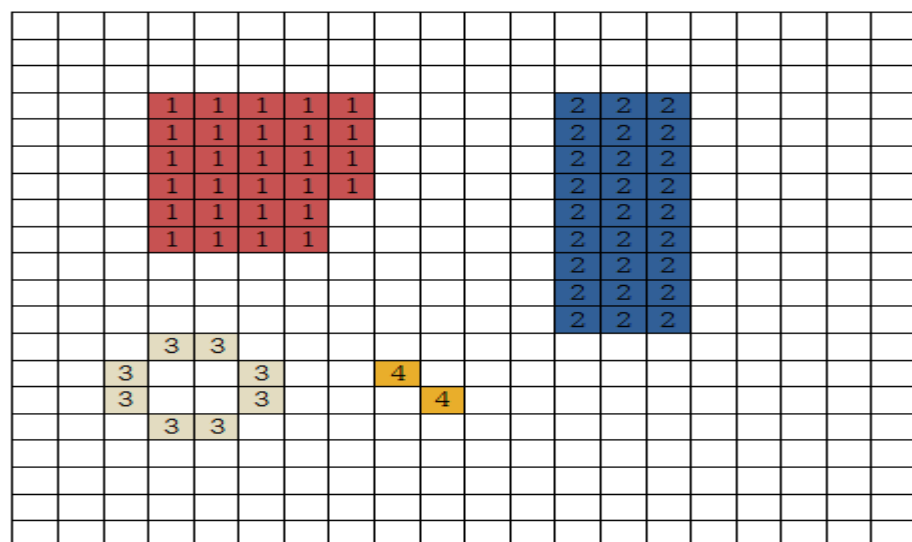
<객체추출을 위한 알고리즘>

1. Labeling

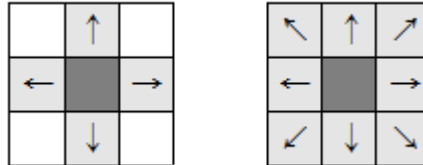
1-channel의 gray scale 영상이 필요하며 이진화된 아래의 이미지처럼 0 or 255(1-channel) 값을 갖는 경우 인접한 영역끼리 그룹을 짓는 것을 Labeling이라고 한다.



아래 그림처럼 인접 영역끼리 Label Number를 매겨 그룹화를 한다.



인접한 pixel을 탐색할 때 8-neighbor 방식을 사용한다. Neighbor는 '이웃'이라는 뜻처럼 8-neighbor는 현재 pixel을 기준으로 몇 개의 이웃 pixel들을 탐색하는지를



뜻한다.

왼쪽 그림은 현재 pixel을 기준으로 위, 아래, 왼쪽, 오른쪽 총 4방향을 탐색하는 4-neighbor 탐색이다. 오른쪽 그림은 4-neighbor에 4개의 대각방향을 더 탐색하는 8-neighbor 탐색이다. 이는 8-neighbor를 사용하였기 때문에 label number가 '4'인 대각 픽셀이 그룹된 것을 확인할 수 있다. 이렇게 인접한 영역들을 Grouping한 것이 Labeling이다.

2. Labeling 구현 순서

- A. 이미지에서 화소 값(255)이 있는 지점까지 탐색
- B. 이미 Labeling된 화소인지 확인
- C. 아니라면 stack에 현재x,y좌표를 저장
- D. 8-neighbor 탐색하면서 픽셀화소가 255이면서 Labeling이 안된 지역을 발견하면 stack에 모두 넣는다.(8-neighbor 모두 탐색한다.)
- E. Stack의 맨 위 좌표를 받아 D번 과정을 반복한다.
- F. 더 이상 grouping할 행렬이 없을 때 (stack이 비었을 때)까지 수행한다.
- G. 이미지의 다음 화소 값이 있는 곳까지 탐색하는 A과정부터 다시 시작한다.
- H. 가장 많은 화소Labelling number를 찾아 Labeling Pixel에 저장한다.
- I. 가장 큰 영역을 남기고 제거한다.

<ColorDetection>

1. IplImage 구조체

영상의 특성과 데이터의 포인터를 갖는 구조체

```
typedef struct_IplImage
{
    int    nSize;                //구조체의 크기
    int    ID;                  //이미지헤더의 버전
    int    nChannels;            //채널의 개수 1~4까지 이용가능
    int    alphaChannel;         //알파채널의 개수
    int    depth;                //픽셀 표현하는 자료형 ex)IPL_DEPTH_8U
    int    origin;               //이미지의 방향
    int    width;                //이미지의 가로 크기
    int    height;               //이미지의 세로 크기
    struct _IplROI *roi;         //ROI(관심영역)의 포인트
    struct _IplImage *maskROI;   // ROI(관심영역)마스크의 포인트
    int    imageSize;            //이미지의 크기를 바이트로 나타낸 것
    char* imageData;             //이미지 픽셀 데이터의 포인터
    int    widthStep;            //이미지의 한 행을 나타내기 위한
    바이트수
    char    *imageDataOrigin     //정렬되지 않은 이미지
    전체에 대한 포인터
}
```

- 자주 사용하는 멤버

Width : 영상 가로 픽셀 크기

Height : 영상 세로 픽셀 크기

nChannels : 색상 평면의 개수, grayscale =1, truecolor =3

depth : 픽셀을 표현하는 자료형에 대한 정보.

IPL_DEPTH_8U : 8비트 unsigned integer (0~255)

IPL_DEPTH_8S : 8비트 signed integer (-128~127)

IPL_DEPTH_16U : 16비트 unsigned integer

IPL_DEPTH_16S : 16비트 signed integer

IPL_DEPTH_32S : 32비트 signed integer

IPL_DEPTH_32F : 32비트 floating-point number

IPL_DEPTH_64F : 64비트 floating-point number

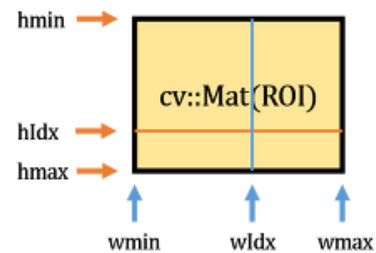
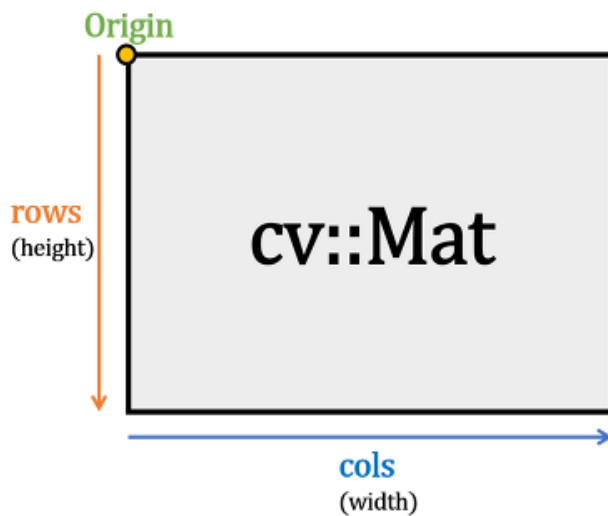
widthStep : 한 행을 표현하기 위해 필요한 바이트 수

origin : 0이면 영상이 정상적인 순서로 저장, 1이면 상하반전

imageData: 픽셀 데이터

2. Mat 클래스

이미지를 행렬로 표현하기 위한 클래스



읽어오는 방법

```
cv::Mat map;
map = cv::imread(~)
for (int hIdx = 0; hIdx < map.rows; ++hIdx)
{
    for (int wIdx = 0; wIdx < map.cols; ++wIdx)
    {
        uchar val = map.at<uchar>(hIdx, wIdx);
        ...
    }
}
```

3. cvCvtColor(CvArr* src, CvArr* dst, int code)

기존 컬러 공간을 특정 컬러 공간 또는 명암도 공간으로 변환

-src : 원본 배열. 8-bit(8u), 16-bit(16u), single-precision floating point(32f) 형태의 이미지.

-dst : 변환된 배열을 저장할 배열.

-code : 색 변환 작업에 관한 파라미터. 그 내용은 다음과 같다.

CV_BGR2GRAY : 흑백영상

CV_BGR2HLS : HLS 채널 영상

CV_BGR2HSV : HSV 채널 영상

4. cvInRange(const CvArr* src, const CvArr* lower, const CvArr* upper, CvArr *dst)

배열의 요소치가 다른 두 개의 배열 요소로 나타내지는 범위내에 위치하는지를 체크한다.

-src : 입력 배열.

-lower : 하한치(그 값을 포함한다)를 나타내는 배열

-upper : 상한치(그 값을 포함하지 않는다)를 나타내는 배열

-dst : 출력 배열(타입은 8U 또는 8S)

함수 cvInRange (은)는 이하와 같이, 입력 배열의 모든 요소에 대해 범위 체크를 실시한다. $dst(l)=lower(l)0 \leq src(l)0 < upper(l)0$

(싱글 채널 배열의 경우),

$dst(l)=lower(l)0 \leq src(l)0 < upper(l)0 \&$

$lower(l)1 \leq src(l)1 < upper(l)1$

(2 채널등의 경우).

src(l) 하지만 범위내이면 dst(l) 에는 0xf(요소 모든 비트가 '1')(을)를 세트 해, 그 이외의 경우는 0 (을)를 세트 한다. 모든 배열(출력 배열을 제외하다)은 같은 타입으로, 배열의 사이즈(또는 ROI 의 사이즈)도 같지 않으면 안 된다.

5. Labeling 실시

Labeling할 이미지와 최소 픽셀수를 설정한다.

```
void CBlobLabeling::SetParam(IplImage* image, int nThreshold)
{
    if (m_recBlobs != NULL)
    {
        delete m_recBlobs;
        m_recBlobs = NULL;
        m_nBlobs = _DEF_MAX_BLOBS;
    }
    if (m_intBlobs != NULL)
    {
        delete m_intBlobs;
        m_intBlobs = NULL;
        m_nBlobs = _DEF_MAX_BLOBS;
    }
    if (m_Image != NULL)    cvReleaseImage(&m_Image);
    m_Image = cvCloneImage(image);
    m_nThreshold = nThreshold;
}
```

6. 추출된 객체에 대한 판별 및 count

Typedef struct CvRect

```
{
    Int    x           //좌측 x좌표
    Int    y           //상각 y좌표
    Int    width       //폭
    Int    height      //높이
}
```

Inline CvRect cvRect(int x, int y, int width, int height)

추출된 객체에 대한 정보를 저장한다.

```
void cvRectangle( CvAr* img, CvPoint pt1, CvPoint pt2, CvScalar color,int thicknes=1,
```

```
int line_type=8, int shift=0 )
```

-img : 그림이 그려질 이미지

-pt1 : 하나의 정점

-pt2 : 반대측의 정점

-color : 선의 색(RGB), 혹은 휘도(그레이 스케일 이미지)

-thickness : 선의 굵기, 부의 값, 예를 들면 CV_FILLED를 지정했을 경우는 전부 칠해진다.

-line_type : 선의 종류

-shift : 좌표의 소수점 이하의 자리수를 나타내는 비트수.

이렇게 그려진 사각형 개수를 파악하여 count를 실시한다.

<character Description Words>

전처리 과정

Gray -> Smooth -> Top-Hat -> Threshold

1. cvCvtColor(CvArr* src, CvArr* dst, int code)

기존 컬러 공간을 특정 컬러 공간 또는 명암도 공간으로 변환

-src : 원본 배열. 8-bit(8u), 16-bit(16u), single-precision floating point(32f) 형태의 이미지.

-dst : 변환된 배열을 저장할 배열.

-code : 색 변환 작업에 관한 파라미터. 그 내용은 다음과 같다.

CV_BGR2GRAY : 흑백영상

CV_BGR2HLS : HLS 채널 영상

CV_BGR2HSV : HSV 채널 영상

2. cvSmooth(const CvArr* src, CvArr* dst, int smoothtype, int param1, intparam2,double param3, double param4)

영상을 스무딩하는데 사용하는 함수

-src : 원본 이미지

-dst : 결과 이미지

-Smoothtype : 스무딩 형태를 지정한다.

CV_BLUR : 가로*세로 근방영역에 대해 합을 구한 다음 가로*세로로 나눈다

CV_GAUSSIAN : 가로*세로 근방영역에 대해 표준편차를 취한다.

CV_MEDIAN : 가로*세로 근방영역에 대해 중앙값을 취한다.

-param1 : 마스크의 가로를 나타낸다.

-param2 : 마스크의 세로를 나타낸다.

-param3 : 표준편차를 지정하는데 만약에 param3=0을 지정한다.

3. cvMorphologyEx(const CvArr* src, CvArr* dst, CvArr* temp, IplConvKernel* element, int operation, int iterations=1)

팽창연산과 침식연산으로부터 확장된 연산 즉, 열림연산, 닫힘연산 그리고 기울기 연산 등을 수행하기 위한 함수이다.

-src : 원본 이미지

-dst : 목적 이미지

-temp : 임시 이미지를 나타내며 열림연산과 닫힘연산인 경우 NULL을 지정한다.

-element : 구조적 요소를 나타낸다. 만약 element=NULL이면 3*3 사각형 구조적 요소가 사용된다

-operation : 다음 연산 중 하나를 지정한다.

CV_MOP_OPEN : 열림연산

CV_MOP_CLOSE : 닫힘연산

CV_MOP_GRADIENT : 기울기 연산

CV_MOP_TOPHAT : top-hat 연산

-iterations : 연산 수행 횟수를 지정한다.

- 침식연산

정해진 영역 내에서 가장 작은 값을 픽셀 중심부의 값으로 바꾸는 최소값필터의 역할을 한다. 이진 영상에서는 객체 영역을 흰색으로 볼 때 이들 영역이 줄고 검은색 영역이 늘어나며, 그레이영상(또는 컬러영상)에서는 밝은 영역이 줄고 어두운 영역이 늘어나는 쪽으로 연산 결과가 변한다.

- 팽창연산

침식연산과 반대로 영역을 넓혀가는 연산이며 최대값 필터의 역할을 한다.

4. cvThreshold(CvArr* src, CvArr* dst, double threshold, double maxValue, CvThreshType type)

영상에서 이진화하기 위한 함수이다.

-src : 원본 이미지

-dst : 결과 이미지

-threshold : 임계값을 나타낸다.

-maxValue : 픽셀의 최대 명암값을 나타낸다.

-type : 이진화 방식을 결정한다.

CV_THRESH_BINARY : 임계값 초과(255), 임계값 이하(0)

CV_THRESH_BINARY_INV : 임계값 초과(0), 임계값 이하(255)

CV_THRESH_BINARY_TRUNC : 임계값 초과(임계값), 임계값 이하(원래 픽셀값)

CV_THRESH_BINARY_TOZERO : 임계값 초과(원래 픽셀값), 임계값 이하(0)

CV_THRESH_BINARY_TOZERO_INV : 임계값 초과(0), 임계값 이하
(원래 픽셀 값)

5. `cvInRange(const CvArr* src, const CvArr* lower, const CvArr* upper, CvArr *dst)`

배열의 요소치가 다른 두 개의 배열 요소로 나타내지는 범위내에 위치하는지를 체크한다.

-src : 입력 배열.

-lower : 하한치(그 값을 포함한다)를 나타내는 배열

-upper : 상한치(그 값을 포함하지 않는다)를 나타내는 배열

-dst : 출력 배열(타입은 8U 또는 8S)

함수 `cvInRange` (은)는 이하와 같이, 입력 배열의 모든 요소에 대해 범위 체크를

실시한다. $dst(I) = lower(I)0 \leq src(I)0 < upper(I)0$

(싱글 채널 배열의 경우),

$dst(I) = lower(I)0 \leq src(I)0 < upper(I)0 \&$

$lower(I)1 \leq src(I)1 < upper(I)1$

(2 채널등의 경우).

`src(I)` 하지만 범위내이면 `dst(I)` 에는 0xf(요소 모든 비트가 '1')(을)를 세트 해, 그 이외의 경우는 0 (을)를 세트 한다.모든 배열(출력 배열을 제외하다)은 같은 타입으로, 배열의 사이즈(또는 ROI 의 사이즈)도 같지 않으면 안 된다.

6. Labeling 실시

1) 레이블링 할 이미지와 최소 픽셀수 등을 설정

`void CBlobLabeling::SetParam(IplImage* image, int nThreshold)`

```
{
    if (m_recBlobs != NULL)
    {
        delete m_recBlobs;
        m_recBlobs = NULL;
        m_nBlobs = _DEF_MAX_BLOBS;
    }
    if (m_intBlobs != NULL)
    {
        delete m_intBlobs;
```

```

        m_intBlobs = NULL;
        m_nBlobs = _DEF_MAX_BLOBS;
    }
    if (m_Image != NULL)    cvReleaseImage(&m_Image);
    m_Image = cvCloneImage(image);
    m_nThreshold = nThreshold;
}

```

2) 레이블링 할 이미지와 최소 픽셀수 등을 설정

```

void CBlobLabeling::_BlobSmallSizeConstraint(int nWidth, int nHeight,
CvRect* rect, int* label, int *nRecNumber)

```

```

{
    if (*nRecNumber == 0)    return;
    int nX;
    int ntempRec = 0;
    CvRect* temp = new CvRect[*nRecNumber];
    int* labeled = new int[*nRecNumber];

    for (nX = 0; nX < *nRecNumber; nX++)
    {
        temp[nX] = rect[nX];
        labeled[nX] = label[nX];
    }
    for (nX = 0; nX < *nRecNumber; nX++)
    {
        if ((rect[nX].width > nWidth) && (rect[nX].height >
nHeight))
        {
            temp[ntempRec] = rect[nX];
            labeled[ntempRec] = label[nX];
            ntempRec++;
        }
    }
}

```

```

    }
    *nRecNumber = ntempRec;
    for (nX = 0; nX < *nRecNumber; nX++)
    {
        rect[nX] = temp[nX];
        label[nX] = labeled[nX];
    }
    delete temp;
    delete labeled;
}

```

3) 가로 전체 크기 90%이상, 세로 전체 크기 90%이상 제거

```

void CBlobLabeling::_BlobBigSizeConstraint(int nWidth, int nHeight, CvRect*
rect, int* label, int* nRecNumber)
{
    if (*nRecNumber == 0)    return;
    int nX;
    int ntempRec = 0;
    CvRect* temp = new CvRect[*nRecNumber];
    int* labeled = new int[*nRecNumber];
    for (nX = 0; nX < *nRecNumber; nX++)
    {
        temp[nX] = rect[nX];
        labeled[nX] = label[nX];
    }
    for (nX = 0; nX < *nRecNumber; nX++)
    {
        if ((rect[nX].width < nWidth) && (rect[nX].height <
nHeight))
        {
            temp[ntempRec] = rect[nX];
            labeled[ntempRec] = label[nX];
            ntempRec++;
        }
    }
}

```

```

    }
}

/*nRecNumber

*nRecNumber = ntempRec;
for (nX = 0; nX < *nRecNumber; nX++)
{
    rect[nX] = temp[nX];
    label[nX] = labeled[nX];
}
delete temp;
delete labeled;
}

```

4) White문자 영역을 추출

5) 추출된 문자 영역을 흑백으로 변경후 이진화

6) 1,2,3 단계를 거친후 TOP-HAT연산을 통해서 테두리 영역을 추출한다.

7) 테두리 영역에서 문자영역을 빼서 문자만 남겨둔다.

7. Tesseract-OCR 문자 검출

- 한글이 학습된 데이터에 대한 경로를 지정

```
TessBaseAPI *tess = new TessBaseAPI();
```

```
tess->Init("C:/tesseract-ocr/tessdata", "kor", OEM_DEFAULT);
```

- 영상배열을 인식기에 포함

```

tess->SetImage(
    (uchar*)mat_ORIROI_SUB.data,           //영상 데이터
    mat_ORIROI_SUB.size().width,           //영상 폭
    mat_ORIROI_SUB.size().height,          //영상 높이
    mat_ORIROI_SUB.channels(),              //영상 채널
    mat_ORIROI_SUB.step1()
);

```

- 검출된 한글을 UTF-8형식 텍스트 파일로 저장

```
const char* search = tess->GetUTF8Text();  
ofstream SaveFile("test.txt");  
SaveFile << search;  
SaveFile.close();
```

8. UTF-8 형식 -> ANSI형식으로 변경

Iconv.lib 사용

```
char tmp_s[255]={0,}, tmp_d[255]={0,};  
strcpy(tmp_s, search);  
const char *ptr_s = tmp_s;  
char *ptr_d = tmp_d;  
size_t n_s = strlen(tmp_s), n_d = sizeof(tmp_d);  
iconv_t cd = iconv_open("EUC-KR", "UTF-8");  
iconv(cd, &ptr_s, &n_s, &ptr_d, &n_d);  
iconv_close(cd);  
fwrite(tmp_d, 1, strlen(tmp_d)-2, fout);  
fclose(fout);
```

9. 검출된 문자열이 오인식될 경우가 있기 때문에 예외로 문자열들을 저장하여 인식률을 올리도록하였다.

E. Web page

(1) HTML:

웹 프로그래밍을 위해서는 우선 HTML에 대해서 알아야 한다. 클라이언트에서 동작하든 서버에서 동작하든, 웹 프로그래밍의 출력 결과가 궁극적으로 HTML로 표현되어야 하기 때문이다. HTML은 웹 문서를 기술하기 위한 마크업 언어로, 다른 HTML 문서와의 연결 기능을 제공한다. HTML이나 CSS, SQL 등에 관한 학습은 w3schools.com을 통해 할 수 있다.

HTML 문서 작성은 편집기로 이루어지는데, 메모장, MS 워드, 한글과 같은 기본적인 텍스트 편집기로 작성할 수도 있고, EditPlus, eclipse와 같은 통합 개발 환경을 통해 작성할 수 있다.

IC4U 프로젝트에서는 EditPlus로 웹 문서를 작성했다.

(2) 웹 프로그래밍 언어:

웹 프로그래밍 기법은 크게 클라이언트 측 프로그래밍과 서버 측 프로그래밍으로 나눌 수 있다. 클라이언트 측 프로그래밍은 프로그램 자체가 클라이언트(웹 브라우저)에서 수행되고, 클라이언트 측 프로그래밍 언어에는 ActiveX, JavaScript 등이 있다. 서버 측 프로그래밍은 프로그램 자체가 서버에서 수행되므로 대용량 파일, DB 연동 등의 복잡한 프로그래밍이 가능하며, 서버 측 프로그래밍 언어에는 ASP, JSP, PHP 등이 있다.

IC4U 프로젝트에서는 웹의 주기능이 DB 연동을 통한 데이터 출력에 있었고, 웹 개발 언어로 PHP가 주로 쓰이고 있는 현황에 맞추어 PHP언어로 개발하였다. 그 외에도 클라이언트 측 프로그래밍 언어인 JavaScript와 jQuery.Ajax가 일부 사용되었다.

(3) 웹 개발 환경 설정:

웹 개발에 필요한 준비물에는 웹 서버, DB 서버, 웹 문서 등이 있다.

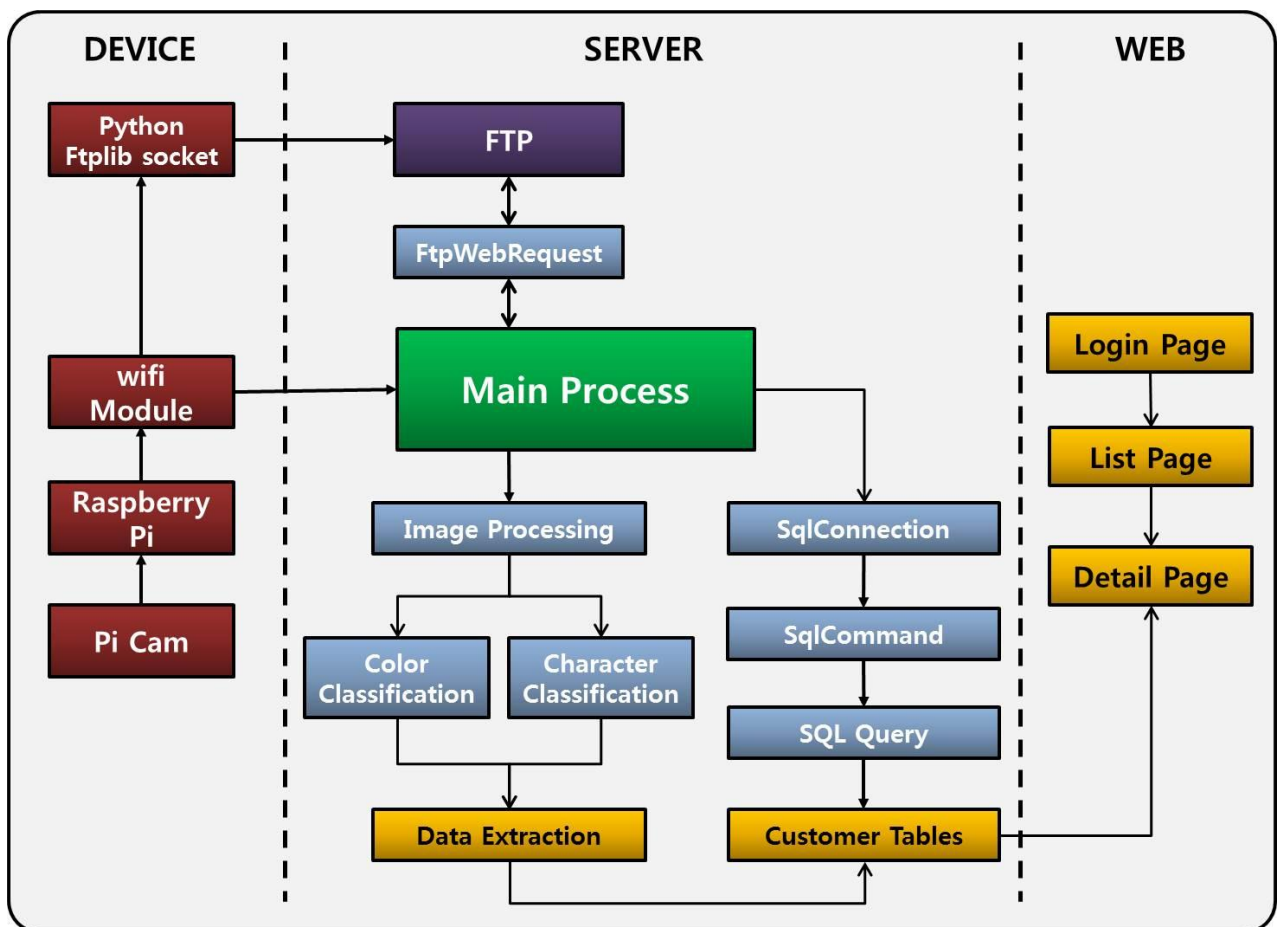
사용자가 브라우저로 웹 서버에 접속하여 해당 페이지를 요청하고, 웹 서버가 요청에 대응하는 처리(처리 결과에 따른 페이지 이동)를 하는 과정을 거침으로써, 사용자는 브라우저를 통해 웹 사이트에 접속하고, 웹 페이지를 이동할 수 있다.

IC4U 프로젝트에서는 DB 서버가 MSSQL을 기반으로 구축되었고, 웹 서버는 Apache를 사용했다. 다만 개발 초기에는 MySQL을 기반으로 웹을 개발했기 때문에 간편하게 APMSETUP을 설치하여 Apache, PHP, MySQL을 이용했다.

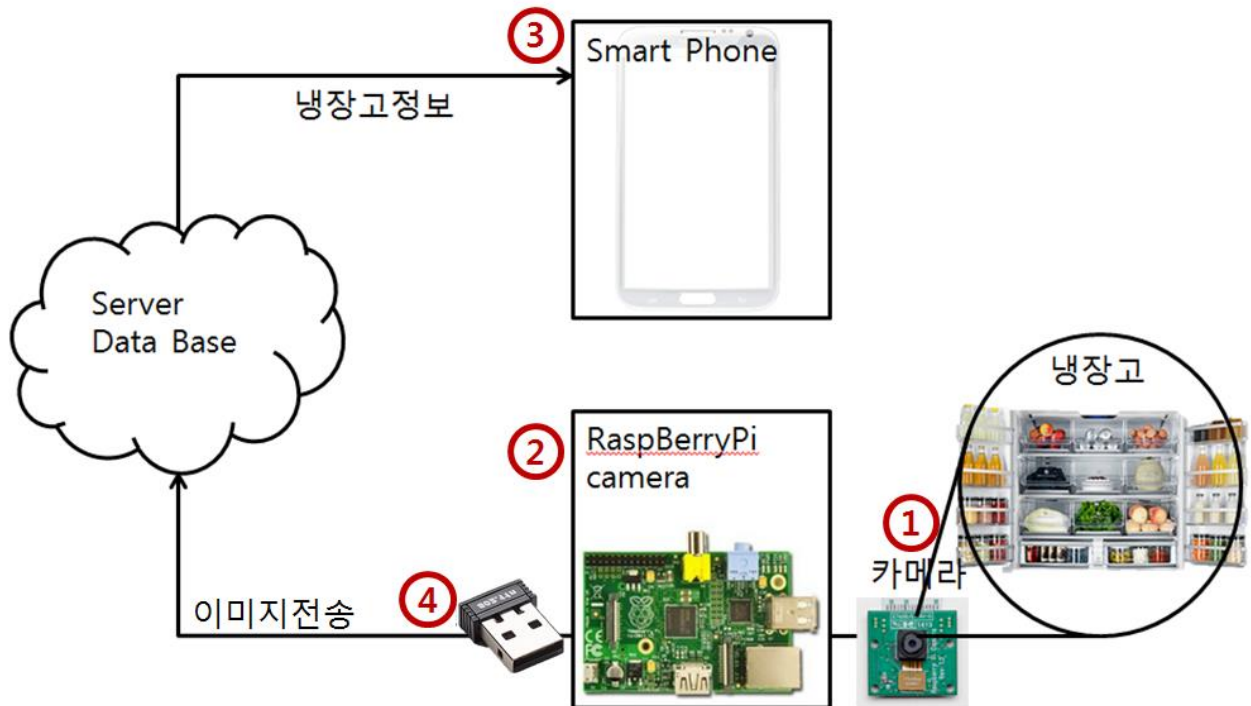
5. 상세 개발 내용

B. 시스템 아키텍처

i. S/W 아키텍처 (최종 아키텍처)



ii. H/W 아키텍처



번호	설명
1	카메라가 냉장고 내부를 촬영한다
2	라즈베리파이 메모리에 촬영된 냉장고 내부 사진이 파일화되어 저장한다
3	스마트폰 또는 PC를 통해 웹사이트 접속하여 DB 정보를 확인한다
4	저장된 이미지 파일이 서버로 전송된다

C. 시스템 설계 및 구현

i. 기능정의리스트

● 임베디드

대분류	중간항목	상세항목	항목번호	설명	담당자	순위
임베디드	설치	OS 설치	Rp-St-01	주로 라즈비안을 설치. 사용될 OS 는 추후 결정	김연욱 변소정	1
		영상처리용 라이브러리 설치	Rp-St-02	라즈베리파이로 모션 인식을 하기 위해 필요한 영상처리용 라이브러리를 설치한다.	김연욱 변소정	1
		wifi 용 라이브러리 설치	Rp- St-03	라즈베리로 통신을 하기 위한 wifi 라이브러리를 설치한다.	김연욱 변소정	1
		카메라 연동	Rp-St-04	라즈베리파이에 카메라 모듈을 설치하고 연동시킨다.	김연욱 변소정	1
		빛 감지센서 연동	Rp-St-05	추후 카메라 촬영 타이밍(시작, 정지) 제어를 하기 위해 라즈베리파이에 빛 감지센서를 설치한다.	김연욱 변소정	1
		wifi 연동	Rp-St-06	라즈베리파이에 wifi 모듈을 설치하고 연동시킨다.	김연욱 변소정	1
	통신	wifi 사용을 위한 고정 IP 얻기	Rp-Co-01	wifi 통신을 하기 위해 라즈베리파이가 잡을 고정 IP 를 설정한다	김연욱 변소정	1
		putty 를 사용한 로컬내 원격 접속	Rp-Co-02	라즈베리파이가 로컬 내 통신이 가능한지 확인하기 위해 putty 를 통한 원격 접속과 삼바서버에 접속한다	김연욱 변소정	1
		오픈 된 포트로 웹서버와 연동 및 TEST	Rp-Co-03	웹서버로 ping 을 보내 연결을 확인한다.	김연욱 변소정	2
		외부통신을 열기 위한 자료조사	Rp-Co-04	로컬영역이 아닌 외부에서도 접속이 가능하도록 외부통신을 열기 위한 자료조사를 한다	김연욱 변소정	3

		로컬 내 이미지 전송	Rp-Co-05	로컬영역에서 서버로 이미지전송을 확인한다.	김연욱 변소정	2
		로컬 밖 이미지 전송	Rp-Co-06	로컬 밖으로 이미지 전송을 확인한다.	김연욱 변소정	3
		이미지 스트리밍 전송	Rp-Co-07	로컬 내에서 이미지 스트리밍의 전송이 가능함을 확인한다.	김연욱 변소정	2
		로컬 밖 이미지 스트리밍 전송	Rp-Co-08	로컬 밖에서 이미지 스트리밍의 전송이 가능함을 확인한다	김연욱 변소정	3
	영상	영상촬영	Rp-lm-01	1 초 간격으로 영상을 찍는다.	김연욱 변소정	1
		영상 촬영 간격 제어	Rp-lm-02	촬영간격의 조정을 쉽게 코딩한다.	김연욱 변소정	1
		영상 파일화 및 저장	Rp-lm-03	jpg 파일로 저장한다.	김연욱 변소정	1
		영상 스트리밍 촬영	Rp-lm-04	영상을 찍기 시작한다.	김연욱 변소정	1
		영상 스트리밍 촬영 중 캡처기능	Rp-lm-05	영상을 찍으면서 원하는 순간에 snap shot 을 찍는다	김연욱 변소정	1
		영상 내 색상 검출	Rp-lm-06	손의 위치를 확인한다	김연욱 변소정	2
		색상 tracking	Rp-lm-07	손의 동선을 확인한다	김연욱 변소정	2
		손의 IN/OUT 의 시점확인	Rp-lm-08	식자재의 이동을 확인한다.	김연욱 변소정	2
		손가락의 움직임 파악	Rp-lm-09	물건을 잡고 놓을 때를 확인하여 물건이 놓여진 위치와 가져갈 위치의 물건을 찾는다	김연욱 변소정	3
		영상 파일화 시에 파일의 위치와 이름 변경	Rp-lm-10	파일이 들어오고 나간 시간의 우선순위 파악 및 들어온 물건과 나간 물건이 있을 때 파일이름에 기록하는 형식	김연욱 변소정	3
		필요한 파일만 남기고 삭제	Rp-lm-11	- 문이 열리는 순간 찍은 최초의 비교 사진 - 손이 들어 왔을 때 마다 찍은 IN 사진	김연욱 변소정	3

				- 손이 나갈 때 마다 찍은 OUT 사진		
	제어	카메라 촬영시작 시점 제어	Rp-Ct-01	빛 감지센서를 통해 냉장고 불이 켜졌을 때를 인식하여 카메라를 동작시킨다.	김연욱 변소정	1
		영상 전송시작 시점 제어	Rp-Ct-02	움직임이 감지되었을 때를 시점으로 영상을 전송한다.	김연욱 변소정	1
		영상 전송 간격 제어	Rp-Ct-03	1 초 간격으로 영상을 전송한다. 코딩 시 시간 간격을 수정할 수 있게 처리한다.	김연욱 변소정	1
		카메라 촬영 정지 시점 제어	Rp-Ct-04	빛 감지센서를 통해 냉장고의 불이 꺼졌을 때 촬영을 정지한다.	김연욱 변소정	1
		영상캡처 시점의 최적화	Rp-Ct-05	식자재가 무엇인지 판별하기 위해 정확하기 들어온 식자재가 잘 보이는 사진이 필요하고 내보낸 식자재가 무엇인지 확인하기 위해 식자재가 나가기 직전의 사진이 필요하기 때문에 영상 캡처의 시점을 최적화 하는 것이 필요하다.	김연욱 변소정	3
		영상 캡처 간격을 좁히기	Rp-Ct-06	빠른 손의 움직임을 쫓기 위해 캡처 간격을 줄일 수 있는 방법을 모색한다.	김연욱 변소정	2

- 영상처리

대분류	중간항목	상세항목	항목번호	설명	담당자	순 위
영상처리	영상분할 및 객체 인식	영상 불러오기	IM-Sg-01	영상을 불러오는 예제 정리를 하고 이를 라즈베리파이 캠과 연결하여 응용한다.	김성민	1
		알고리즘 찾기	IM-Sg-02	영상분할 알고리즘들과 이를 구현한 OpenCV 함수 조사 및 예제 정리한다.	김성민	1
		알고리즘 결정	IM-Sg-03	각 알고리즘들은 분할 정확성과 처리속도에 따라 달라지므로 두 속성을 고려하여 결정한다.	김성민	2
		알고리즘으로 영상 분할	IM-Sg-04	정확성과 처리속도를 고려한 알고리즘으로 입력 이미지의 전경 객체와 배경으로 분할하여 결과 영상을 생성한다.	김성민	1
		객체 추출	IM-Sg-05	분할된 이미지에서 객체를 추출하는 함수를 통해 객체를 추출하여 종류를 분류한다.	김성민	3
		객체 인식	IM-Sg-06	만약 과일/채소를 넣으면 종류가 자동으로 판별되도록 하고 그 외의 식품인 경우 인식을 하지 않도록 한다.	김성민	3
	ColorDetection	영상 받아오기	IM-Cd-01	영상의 특성과 데이터의 포인터를 갖는 구조체(IplImage), 영상을 행렬로 표현하기 위한 클래스(Mat)으로 읽어온다	김성민	2
		color공간 변환	IM-Cd-02	기존 컬러 공간에서 특정 컬러 공간이나 명암으로 변환한다.	김성민	2

		영상 배열변환	IM-Cd-03	IpImage구조체일경우 Mat클래스로 변경해 준다.	김성민	2
		지정해준 배열 범위내에 있는지 체크	IM-Cd-04	변환된 배열의 요소치가 다른 두 개의 배열 요소로 나타내지는 범위내에 위치하는지 체크한다.	김성민	2
		Labeling 실시	IM-Cd-05	Labeling할 이미지와 최소 픽셀수를 사용자가 설정한다.	김성민	2
		추출된 객체에 대한 판별 및 count	IM-Cd-06	추출된 객체에 대한 정보를 받아 카운트를 실시한다.	김성민	2
	character	입력된 영상 받기	IM-Cw-01	영상의 특성과 데이터의 포인터를 갖는 구조체(IpImage), 영상을 행렬로 표현하기 위한 클래스(Mat)으로 읽어온다	김성민	3
		Color공간 변환	IM-Cw-02	기존 컬러 공간에서 특정 컬러 공간이나 명암으로 변환한다.	김성민	3
		영상 smooth	IM-Cw-03	객체 추출을 위한 스무딩작업을 한다. 여기서는 가우시안필터를 사용하였다.	김성민	3
		포폴로지연산을 통한 영상 변환	IM-Cw-04	침식과 팽창을 통해 추출하고자 하는 영역을 구분짓는다.		3
		영상 이진화	IM-Cw-05	이진화를 통해서 문자영역을 확실히 구분지을수 있도록 한다.		3
		지정해준 배열 범위내에 있는지 체크	IM-Cw-06	변환된 배열의 요소치가 다른 두 개의 배열 요소로 나타내지는 범위내에 위치하는지 체크한다.		3
		Labeling 실시	IM-Cw-07	레이블링 할 이미지와 최소 픽셀수 등을 설정한후 가로 전체 크기 90%이상, 세로 전체 크기 90%이상은 제거한다.		3

		추출된 영역에 대한 문자 추출	IM-Cw-08	Tesseract-OCR을 통한 문자를 검출한다. 하지만 한글을 인식하게 위해서는 표본데이터가 필요하여 한글을 학습시킨다.		3
		문자 형식 변환	IM-Cw-09	Tesseract-OCR를 통해 추출된 문자는 UTF-8 형식이므로 ANSI형식으로 변경하여 저장한다.		3

● 서버/데이터베이스

대분류	중간항목	상세항목	항목번호	설명	담당자	순위
서버	통신	외부 접속 포트 설정	COM-PAC-01	FTP서버의 외부 접속이 가능하도록 포트와 호스트를 설정한다	김주영	1
		디바이스 소켓 개발	COM-PAC-02	Python 기반의 데이터 전송 소켓 클라이언트 소켓을 개발한다.	김주영	1
		서버 소켓 개발	COM-PAC-03	C# 기반으로 Ironpython 라이브러리를 이용해 python 데이터 수신 서버 소켓을 개발하여 C# 코드로 동작시킨다.	김주영	1
		프로토콜 정의	COM-PAC-04	서버/클라이언트 간 사용할 프로토콜을 정의하여 코드로 작성 및 적용한다.	김주영	1
		통신 구조 설계	COM-PAC-05	전체 프로그램에서의 통신망 구조를 설계한다	김주영	1
	서버 구축	FTP서버 구축	COM-SVC-01	디바이스의 데이터를 전송 받을 수 있는 FTP 서버를 구축한다	김주영	1
		웹 서버 구축	COM-SVC-02	웹 페이지 정보 제공하는 웹 서버를 구축한다 HTTP 프로토콜을 사용한다.	김주영	2
		DB 서버 구축	COM-SVC-03	MsSQL SERVER 2012 express를 이용하여 데이터베이스 서버를 구축한다.	김주영	2
		통합 서버 환경 구축	COM-SVC-04	FTP, 웹, DB서버를 연동시키는 통합 환경 구축한다	김주영	2
		서버 가동 설정	COM-SVC-05	서버가 24시간 작동할 수 있도록 설정한다	김주영	2
		멀티 프로세스 서버로 전환	COM-SVC-06	복수의 디바이스가 보내는 정보를 분산처리 할 수 있는 멀티 프로세스 서버를 제작하여 적용한다	김주영	3
	서버 관리	서버 관리자 ID 생성	COM-SVIO-01	서버 관리자용 ID와 패스워드를 설정한다	김주영	1

		ID 생성 및 저장	COM-SVIO-02	웹 서버에 접근할 수 있는 ID를 생성 및 접근 권한 분할	김주영	1
		이미지 프로세싱 데이터 입력 쿼리 제작	COM-SVIO-03	이미지 프로세싱 함수를 통해 추출되는 데이터가 DB서버를 통해 테이블이 입력될 수 있는 기능을 개발한다.	김주영	2
		웹 페이지 정보 입출력 쿼리 제작	COM-SVIO-04	웹 페이지 표시되는 정보들을 입출력하는 기능을 개발한다.	김주영	2
		웹 페이지 정보 검색 쿼리 제작	COM-SVIO-05	웹 페이지에서 DB서버 내의 정보를 검색 및 출력할 수 있는 기능을 개발한다.	김주영	2
		웹 서버 로그인 기능	COM-SVIO-06	웹 서버에 로그인 기능을 추가한다.	김주영	3
	데이터베이스 테이블	템플릿 DB 설계	COM-DBT-01	이미지 프로세싱에 필요한 템플릿 DB 테이블을 설계한다	김주영	1
		템플릿 DB 생성	COM-DBT-02	이미지 프로세싱에 필요한 템플릿 DB를 생성한다.	김주영	2
		회원 DB 설계	COM-DBT-03	로그인 및 냉장고 DB 분류를 위해 필요한 회원 DB를 설계한다	김주영	1
		회원 DB 생성	COM-DBT-04	로그인 및 냉장고 DB 분류를 위해 필요한 회원 DB를 생성한다	김주영	2
		어플리케이션 DB 설계	COM-DBT-05	사용자용 정보를 표시하기 위한 어플리케이션 DB를 설계한다	김주영	1
		어플리케이션 DB 생성	COM-DBT-06	사용자용 정보를 표시하기 위한 어플리케이션 DB를 생성한다	김주영	2
		냉장고 DB 설계	COM-DBT-07	이미지 프로세싱에 의해 처리된 사용자의 냉장고 내부 정보를 저장하는 DB를 설계한다.	김주영	1
		냉장고 DB 생성	COM-DBT-08	이미지 프로세싱에 의해 처리된 사용자의 냉장고 내부 정보를 저장하는 DB를 생성한다.	김주영	1

● 웹 페이지

대분류	중간항목	상세항목	항목번호	설명	담당자	순위
웹 사이트	UI/UX	목록 페이지	Web-UI-01	웹사이트 접속시 최초로 뜨는 화면 -최상단: IC4U 문구 -목록: 번호, 촬영 날짜로 구성	변소정	1
		세부 페이지	Web-UI-02	목록 페이지에서 선택된 촬영 날짜의 정보를 보여주는 화면 -최상단: 촬영 날짜 -냉장고 내부 촬영 사진 -냉장고 내부 식자재 목록: 번호, 이름, 개수, 유통기한, 선택으로 구성(선택은 체크박스) -수정하기 버튼: 체크박스에서 체크한 후 수정하기 버튼을 누르면 수정 페이지로 이동할 수 있다. 체크된 것이 없으면 체크된 것이 없음을 알린다.	변소정	1
		수정 페이지	Web-UI-03	세부 페이지에서 체크한 항목의 목록이 보여지며, 유통기한을 수정할 수 있는 화면 -최상단: 촬영 날짜 -체크한 항목의 목록: 이름, 개수, 유통기한으로 구성(유통기한은 사용자가 터치하면 전체 드래그 되고, 사용자로부터 키패드로 날짜를 입력 받음)	변소정	1
		로그인 페이지	Web-UI-04	아이디와 패스워드를 입력하여 로그인하거나 회원가입 페이지로 이동하는 버튼이 있는 화면 -최상단: IC4U 문구 -아이디 입력칸, 패스워드 입력칸, 로그인 버튼, 회원가입 버튼으로 구성: 로그인 성공시 목록 페이지로 이동하며, 아이디 패스워드가 일치하지 않을 경우 해당 아이디나 패스워드가 존재하지 않는다는 알림을 한다.	변소정	2

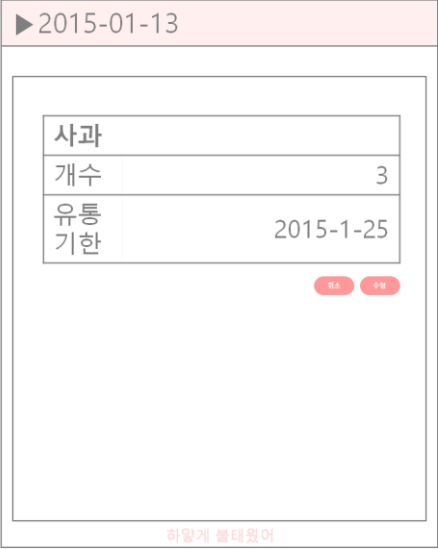
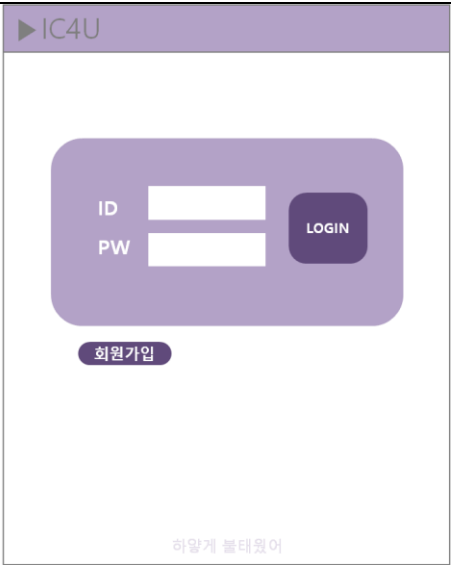
		회원가입 페이지	Web-UI-05	<p>회원가입을 위한 양식을 입력 받는 화면</p> <p>-최상단: IC4U 문구</p> <p>-아이디 입력칸, 패스워드 입력칸, 디바이스 시리얼번호 입력칸, 이메일 입력칸, 이메일 주소 선택칸, 아이디 중복확인 버튼, 디바이스 시리얼번호 중복확인 버튼, 가입 완료 버튼, 취소 버튼으로 구성: 중복확인 버튼을 누르면 DB 와 입력값을 비교하여 중복되면 중복 알림을 하고, 중복되지 않으면 OK 알림을 한다. 가입완료 버튼을 누르면 입력값들을 DB 에 저장하고 로그인 페이지로 돌아간다.</p> <p>취소 버튼을 누르면 입력값을 저장하지 않고 로그인 페이지로 돌아간다.</p> <p>입력된 값이 없을 경우 입력되지 않은 곳에 커서를 위치시키고 입력되지 않았다고 알림을 보인다.</p>	변소정	2
	웹 서버	웹서버 구축	Web-SV-01	Apache 를 설치한다.	변소정	1
		클라이언트 요청에 따른 처리	Web-SV-02	<p>회원가입 시 모든 입력값은 변수에 저장한 후, 가입완료 버튼 처리 후 DB 에 저장한다.</p> <p>회원가입 버튼: 회원가입 페이지 호출(출력)</p> <p>아이디/디바이스시리얼번호 중복확인 버튼: 입력된 아이디와 시리얼 넘버를 변수에 담아 DB와 비교한다. 경우에 상관 없이 알림과 함께 페이지를 유지한다.</p> <p>이메일 주소 선택: 선택한 값을 변수에 저장한다.</p> <p>가입완료 버튼: 회원가입 페이지에서 입력된 데이터를 DB 로 전송한다. 입력이 누락된 것이 있으면 알림과 함께 페이지를 유지한다.</p> <p>로그인 버튼: 목록 페이지 출력 (계속)</p>	변소정	1


				<p>목록 페이지의 촬영 날짜 링크: 세부 페이지 출력</p> <p>세부 페이지의 수정하기 버튼: 수정 페이지 출력</p> <p>수정 페이지의 유통기한 링크: 날짜가 전체 드래그되며, 사용자로부터 입력 받은 값을 변수에 저장한다.</p> <p>수정 완료 버튼: 변수에 저장된 입력값을 DB 값과 바꾸어 저장한다. 입력값이 없을 경우 알림과 함께 페이지 유지</p> <p>취소 버튼: 세부 페이지로 돌아간다.</p>		
	DB(MySQL)	MySQL 설치 및 환경 설정	Web-DB-01	MySQL 을 설치한다.	변소정	1
		회원가입 페이지에서 입력되는 DB 구조설계	Web-DB-02	아이디, 패스워드, 디바이스 시리얼 넘버, 이메일 주소로 구성되어 있다.	변소정	2
		수정 페이지에서 입력되는 DB 구조설계	Web-DB-03	항목 이름과 개수, 유통기한 날짜로 구성되어 있다.	변소정	1

ii. UI 설계

● 전체 UI 레이아웃

구성	UI
<p>목록 페이지</p> <p>로그인 후 유저들에게 보여지는 화면</p> <p>로그인 확인 버튼을 클릭한 후, 유효한 로그인 임이 확인되면 바로 이동함</p>	
<p>세부 페이지</p> <p>목록 페이지에서 촬영 날짜버튼을 선택하여 클릭할 경우 유저에게 보여지는 화면</p> <p>촬영 날짜 버튼을 클릭하면 바로 이동함</p>	

<p>수정 페이지</p> <p>세부 페이지에서 체크한 항목의 목록이 보여지며, 유통기한을 수정할 수 있는 화면</p>	
<p>로그인 페이지</p> <p>아이디와 패스워드를 입력하여 로그인 하거나 회원가입 페이지로 이동하는 버튼이 있는 화면</p>	

<p>회원가입 페이지</p> <p>회원가입을 위해 데이터를 입력 받는 화면</p>	
--	---

- 세부 기능 UI 레이아웃

- 1) 목록 페이지

- i. 정의 및 구성

- 웹 사이트 접속 시 최초로 뜨는 화면(1 차 UI)
- 1 번: 목록의 번호
- 2 번: 버튼, 위에 촬영된 날짜가 써 있다.

▶ IC4U

번호	촬영 날짜
1	2015-01-09
2	2015-01-12
3	2015-01-13

하얏게 볼타웠어

- ii. 기능

- 촬영 날짜 버튼
- 버튼을 클릭하면, 촬영 날짜의 세부 페이지로 이동한다.

번호	이름	개수	유통기한	선택
1	사과	3	2015-1-25	<input type="checkbox"/>
2	바나나	1	2015-1-15	<input type="checkbox"/>

2) 세부 페이지

i. 정의 및 구성

- 목록 페이지에서 선택된 촬영 날짜의 정보를 보여주는 화면
- 번호, 식자재 이름, 개수, 유통기한으로 구성된 표가 있다.

▶ 2015-01-13

20150113.jpg

번호	이름	개수	유통기한	선택
1	사과	3	2015-1-25	<input checked="" type="checkbox"/>
2	바나나	1	2015-1-15	<input type="checkbox"/>

4

하얏게 불태웠어

ii. 기능

- 선택 항목 체크박스와 수정하기 버튼이 있다.
- 선택 항목의 체크 박스를 체크하고 수정하기 버튼을 누르면 수정 페이지로 이동한다.

▶ 2015-01-13

20150113.jpg

번호	이름	개수	유통기한	선택
1	사과	3	2015-1-25	<input checked="" type="checkbox"/>
2	바나나	1	2015-1-15	<input type="checkbox"/>

2

하얏게 불태웠어

➔

▶ 2015-01-13

사과	
개수	3
유통기한	2015-1-25

확인
취소

하얏게 불태웠어

3) 수정 페이지

i. 정의 및 구성

- 세부 페이지에서 체크한 항목이 이름, 유통기한으로 구성된 표로 보여지며, 유통기한을 수정할 수 있는 화면이다.

ii. 기능

- 유통기한 수정: 유통기한 숫자를 터치하면 키패드 또는 키보드 숫자 입력으로 유통기한을 수정할 수 있다
- 수정 완료 버튼: 클릭하면 입력한 유통기한이 저장되고 목록페이지로 이동한다.
- 취소 버튼: 클릭하면 저장하지 않고 목록 페이지로 이동한다.

4) 로그인 페이지

i. 정의 및 구성

- 아이디와 패스워드를 입력하여 로그인하거나 회원가입 페이지로 이동하는 버튼이 있는 화면

IC4U

ID

PW

LOGIN

회원가입

하얏게 불러왔어

ii. 기능

- 로그인 버튼: 아이디와 패스워드를 입력 후 로그인 버튼을 터치하면 목록 페이지로 이동한다.
- 회원가입 버튼: 회원가입 버튼을 누르면 회원가입 페이지로 이동한다.

IC4U

ID

PW

LOGIN

회원가입

하얏게 불러왔어

IC4U

번호	활동 날짜
1	2015-01-09
2	2015-01-12
3	2015-01-13

하얏게 불러왔어

IC4U

ID 등록 확인

PW

다바이스 시리얼 넘버 등록 확인

email @

가입완료 취소

하얏게 불러왔어

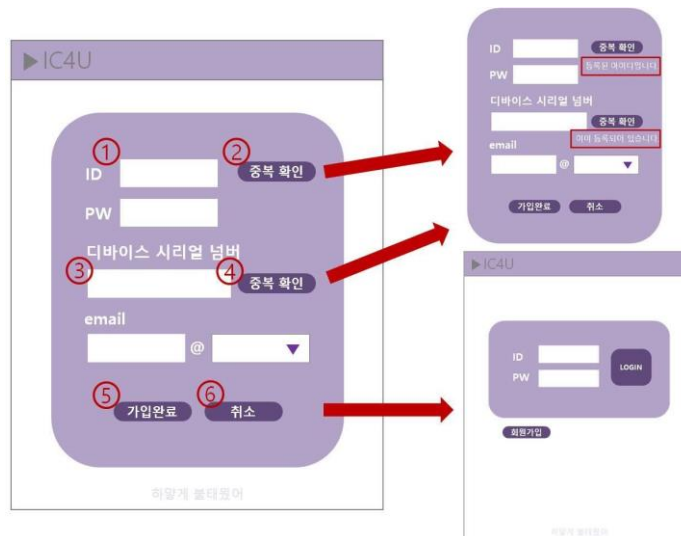
5) 회원가입 페이지

i. 정의 및 구성

- 회원 가입을 위해 데이터를 입력 받는 화면으로 아이디, 패스워드, 시리얼 넘버, 이메일 주소를 입력 받는 칸이 있으면, 아이디 중복 확인 버튼, 디바이스 시리얼 넘버 중복 확인 버튼, 가입 완료 버튼, 취소 버튼으로 구성되어 있다.

ii. 기능

- 아이디 중복 확인 버튼: 아이디 중복되면 중복 알림을, 아니면 OK알림을 한다.
- 시리얼 번호: 시리얼 번호가 중복되면 중복 알림을, 아니면 OK알림을 한다.
- 가입 완료 버튼: 정상적인 경로로 가입이 되었을 경우 로그인 화면으로 이동한다.
- 취소 버튼: 현재 하던 행동과 상관없이 바로 로그인 화면으로 이동한다.

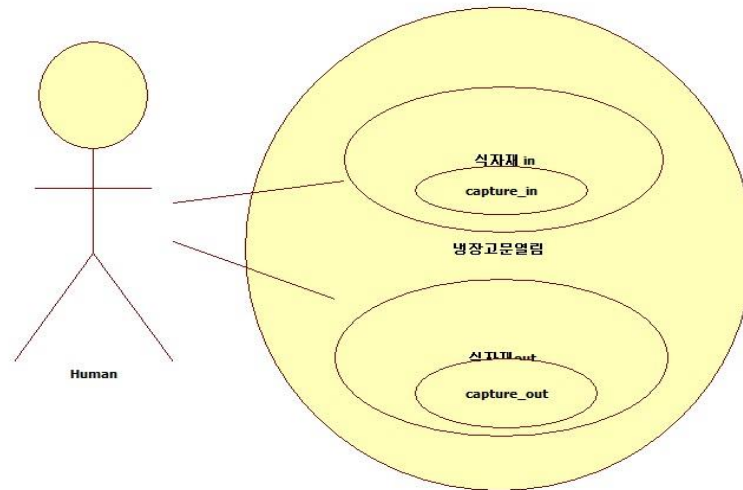


iii. 기능 설계서(Diagram, Table, Packet)

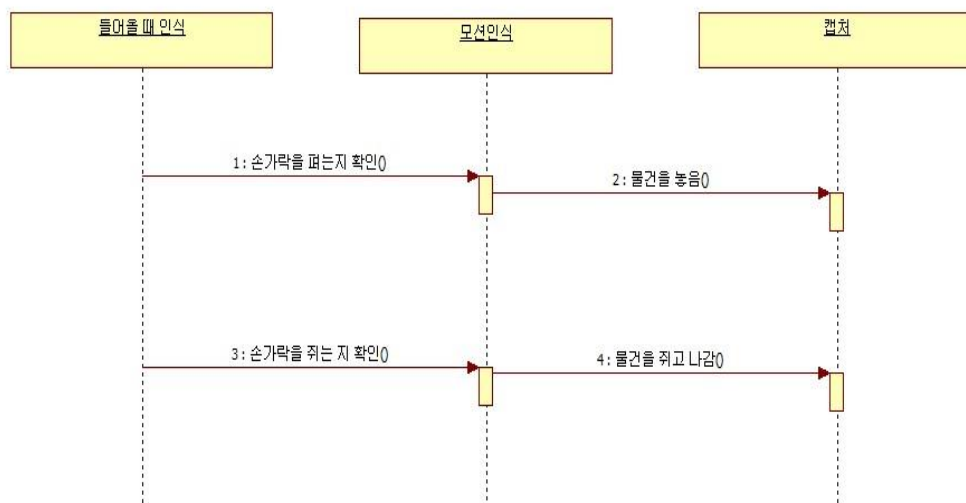
● UML

A. 임베디드

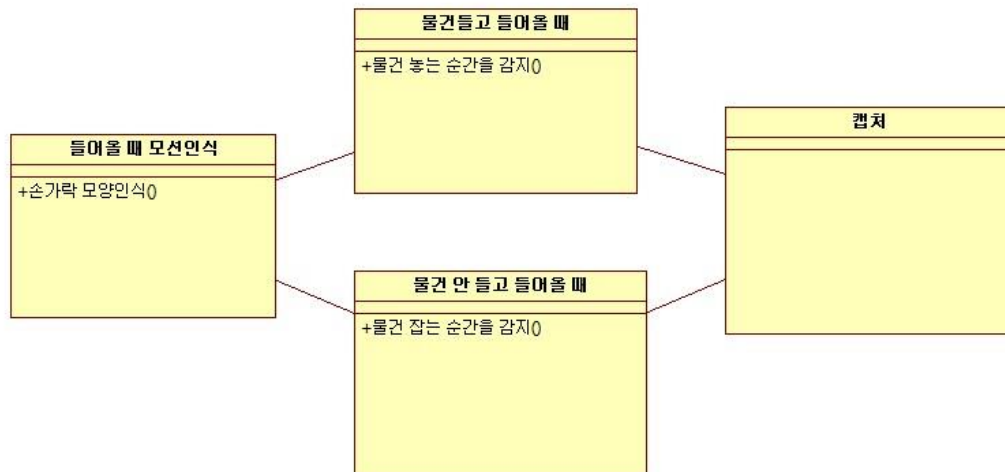
i. Use Case



ii. Sequence

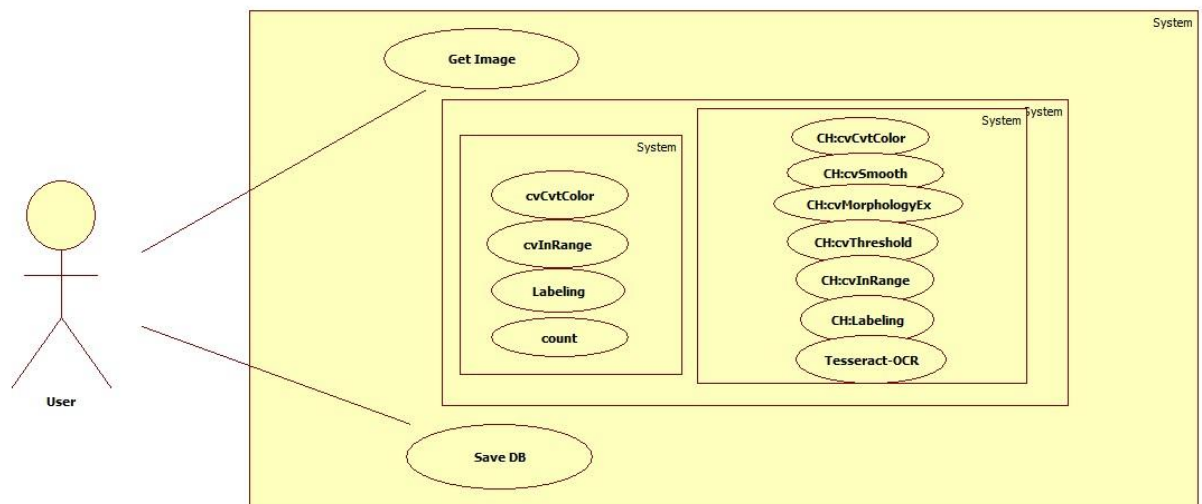


iii. Class

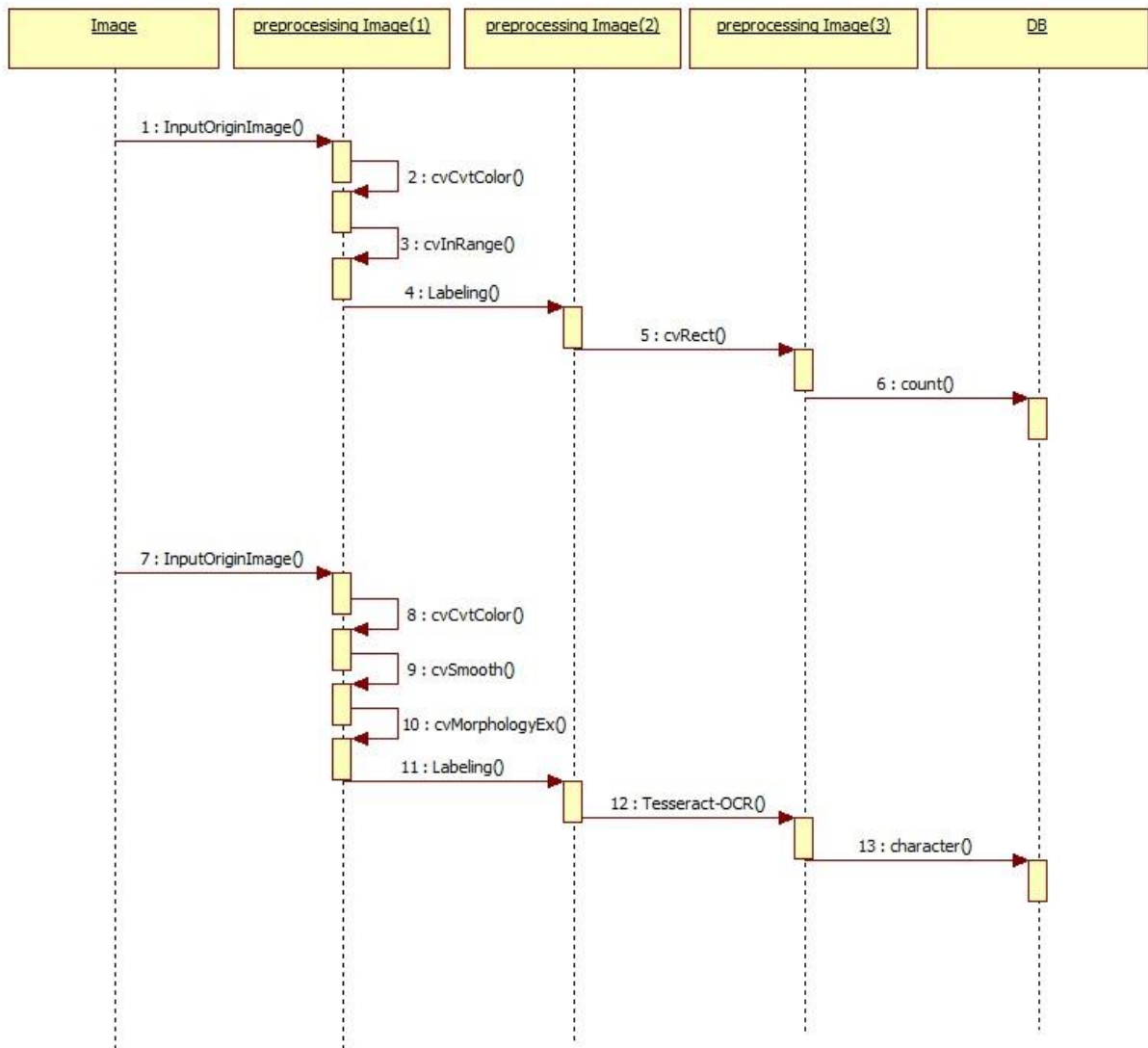


B. 영상처리

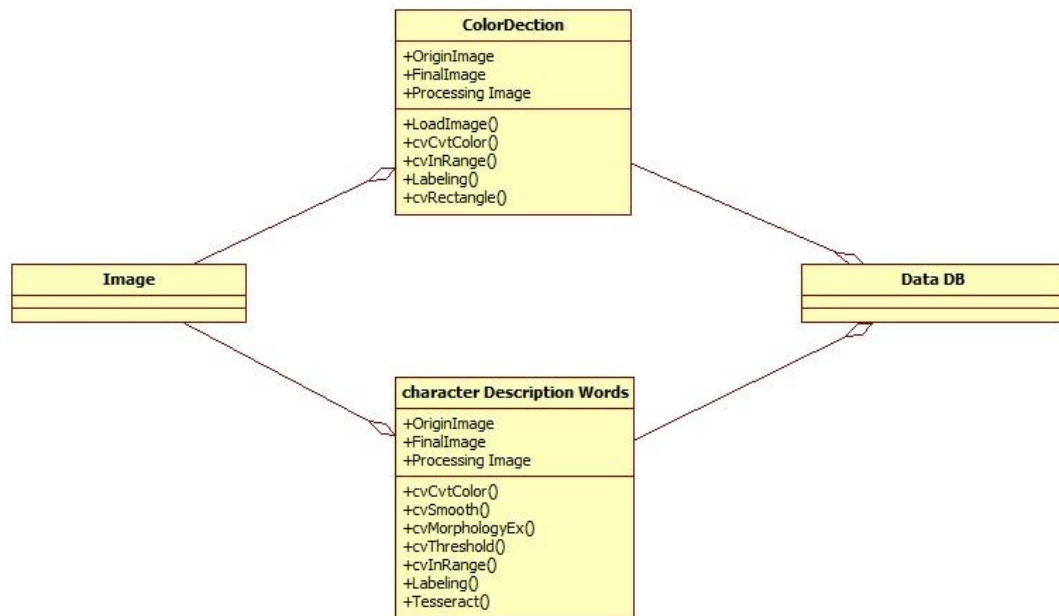
i. Use Case



ii. Sequence

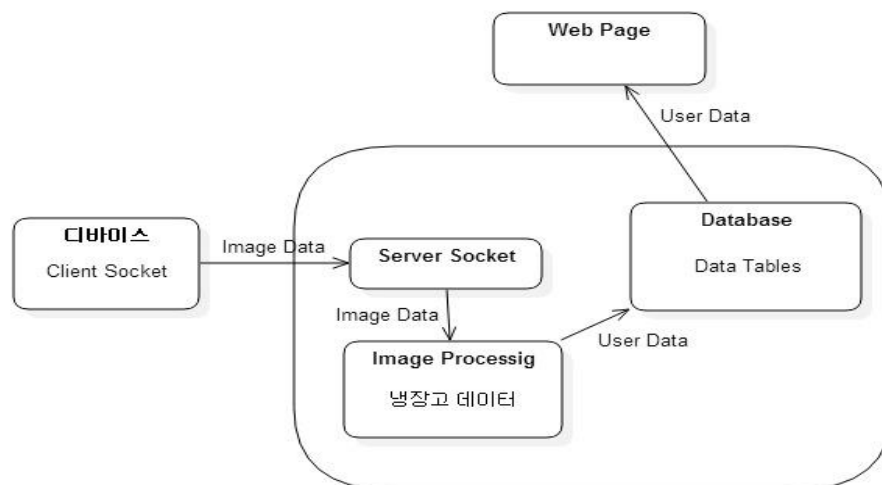


iii. Class

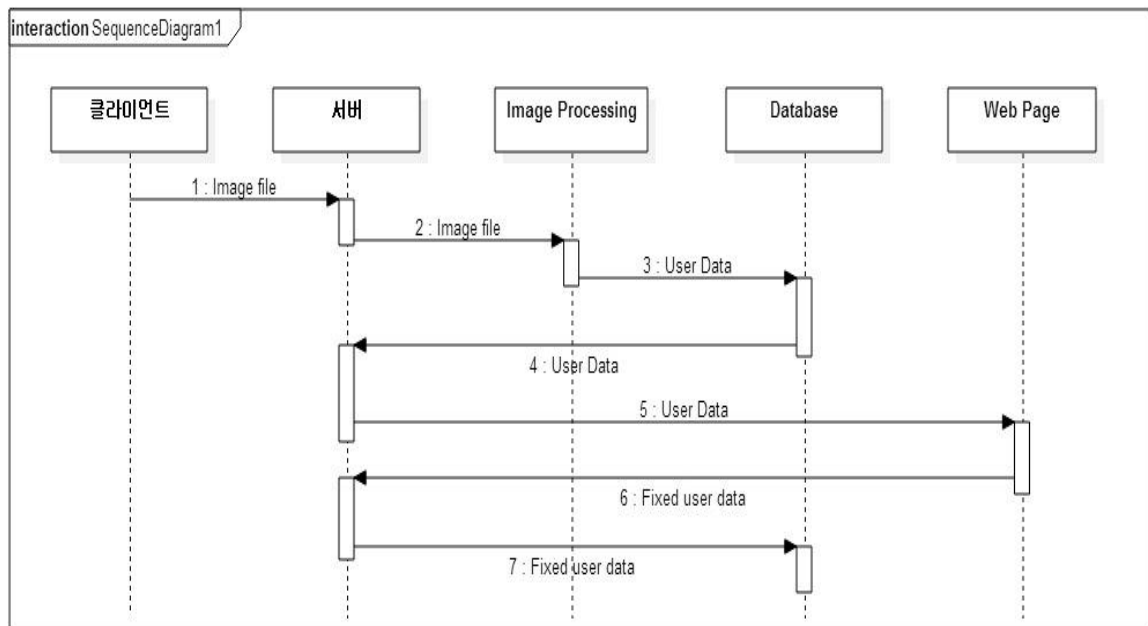


C. 서버 / 데이터베이스

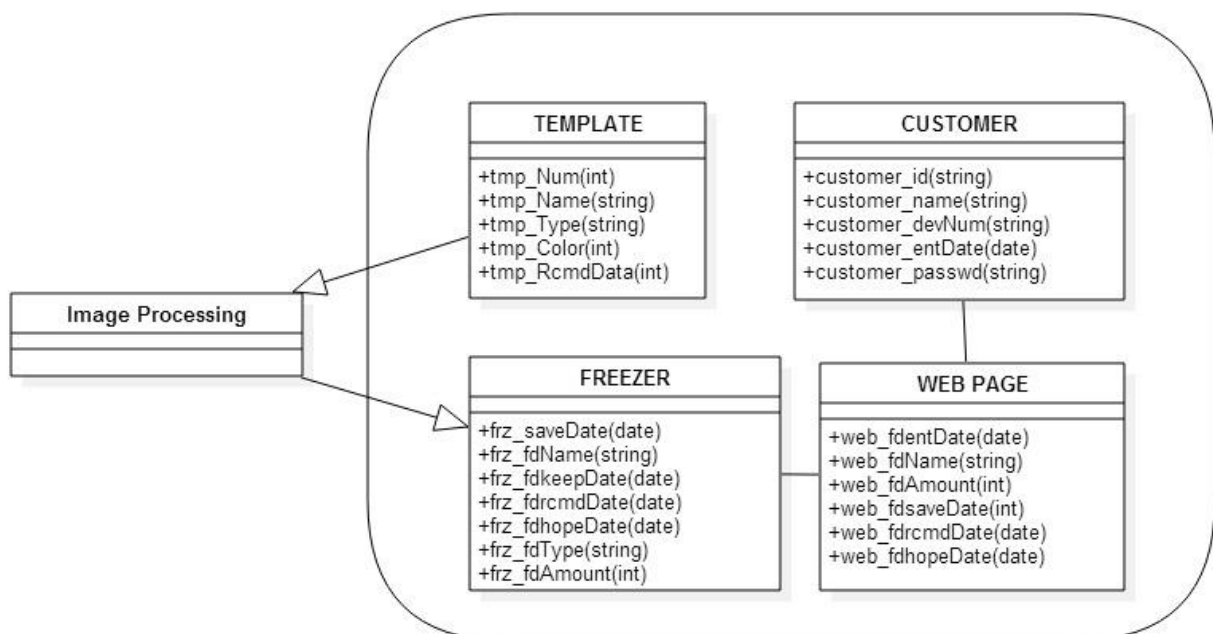
i. Use Case



ii. Sequence

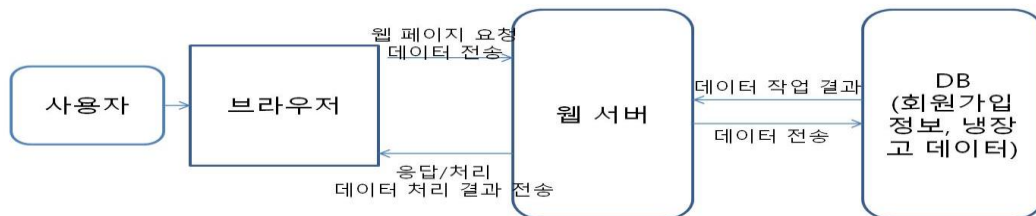


iii. Class

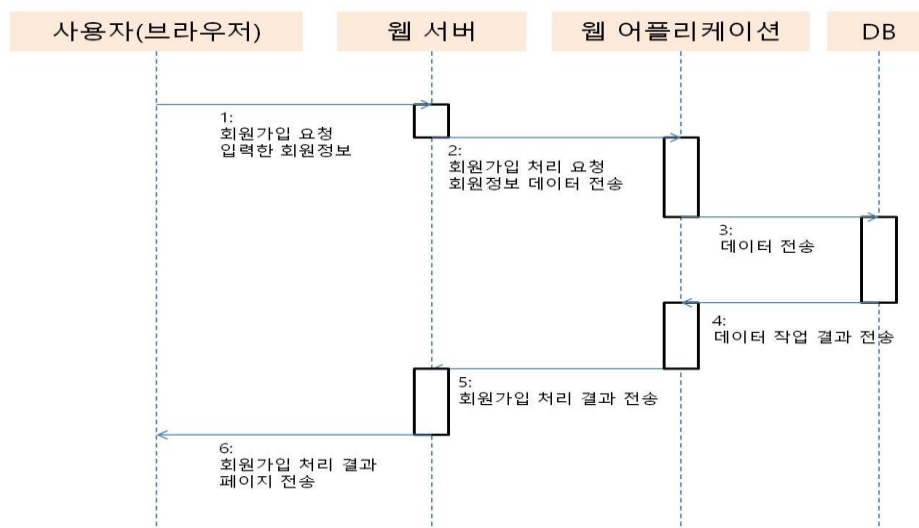


D. 웹페이지

i. Use Case



ii. Sequence



iii. Class

회원가입 +아이디 +패스워드 +디바이스 시리얼 넘버 +이메일 주소 +회원가입처리요청()	항목선택 +날짜 +세부페이지요청()	수정선택 +항목이름 +수정페이지요청()
수정완료선택 +유통기한 +세부페이지요청()	로그인 +아이디 +패스워드 +목록페이지요청()	

D. 시스템 특징, 장점

- i. 라즈베리파이 보드를 사용한 외부 디바이스를 통해 어디서나 간편하게 설치할 수 있다.

E. 확장 방안

- i. 데이터베이스 내부에 존재하는 식자재 정보와 인터넷 검색 알고리즘을 이용하여, 현재 냉장고에 존재하는 식자재를 이용하여 만들 수 있는 요리 레시피를 제공하거나, 해당 식자재의 이름이 tag로 포함되어 있는 블로그 등을 검색하여 식자재 활용에 대한 콘텐츠 정보를 제공할 수 있다.
- ii. 모션 인식 기능, 템플릿 매칭 기능과 같은 영상처리 기술을 추가하여, 식자재의 입출고 과정을 세심하게 관리하고 보다 정확한 식자재 정보를 사용자에게 제공할 수 있다.
- iii. 냉장고 하나 당 1개의 디바이스를 설치하는 것이 아니라, 신선 식품, 냉동실, 김치 보관함, 과일 보관함, 반찬함 등 냉장고 내부의 분리된 구획에 맞춰 여러 개의 디바이스를 설치하고 통합 관리함으로써 통합적인 냉장고 관리 시스템을 구축할 수 있다.

7. 사용자 매뉴얼

A. 필수 S/W, H/W, Program

하드웨어

냉장고에서 영상 촬영과 촬영 영상 전송을 위해 필요한 하드웨어로 (1) 라즈베리파이와 (2) wifi 동글, (3) 웹캠, (4) 충전식 배터리가 있다.

촬영 영상을 수신하고 영상 처리 프로그램으로 영상을 처리한 후, 데이터를 DB 서버에 저장하기 위해 서버로 사용할 (4) 컴퓨터가 필요하다.

사용자에게 영상 처리 결과로 얻은 데이터를 웹으로 보여주고 사용자로부터 데이터를 입력 받기 위해 (5) 컴퓨터나 핸드폰이 필요하다.

소프트웨어 및 프로그램

라즈베리파이가 냉장고에서 움직임을 인식하고 사람의 손을 인식하여 영상을 촬영하기 위해 (1) openCV 라이브러리를 이용하여 (2) IDLE에 파이썬으로 코딩했다.

영상의 송수신을 위한 소켓과 영상 처리 프로그램을 구동하는 데에는 (3) Visual studio 2010에 C#언어를 사용하고, 데이터를 저장하고 관리하기 위해 (4)MSSQL, (5) SQL Server Management Studio를 사용했다.

영상에서 글자나 식자재를 인식하고 데이터를 추출하기 위한 영상 처리 프로그램은 Visual studio 2010에서 C++로 코딩했고, openCV 라이브러리를 이용했다.

웹 페이지를 만들기 위해서는 (6) EditPlus와 (7) SQL Server Management Studio, (8) Apache, (9) 웹 브라우저가 이용되었다.

B. 개발환경 설치

i. 임베디드

라즈베리파이에 OS 로 라즈비안을 설치하고, openCV 를 설치한다.

컴퓨터에 파이썬 인터프리터인 IDLE 를 설치한다.

ii. 서버/DB

영상 처리 프로그램을 구동하고 FTP 서버와 시리얼 통신을 하기 위해서 Visual studio 2010 을 설치한다.

데이터를 저장하고 관리하기 위해 DBMS 로 MSSQL 을 설치하고, SQL Server Management Studio 를 설치한다.

- iii. 영상 처리
컴퓨터에 영상 처리 프로그래밍을 할 Visual studio 2010 을 설치하고, openCV 를 설치한다.
- iv. 웹
웹 개발은 메모장에도 해도 되지만, 좀 더 편한 코딩을 위해 컴퓨터에 EditPlus와 같은 편집기를 설치한다. php와 apache를 설치하고 이용하기 쉽게 APMSETUP을 설치한다. 웹 서버로 사용할 포트를 방화벽에서 외부 접속을 허용한다. 공유기로 ip가 할당된 경우, 포트 포워딩을 해준다.

C. 실행

냉장고에 프로그래밍을 마친 라즈베리파이를 설치하고 전원을 연결한다.
식자재나 이름을 적은 반찬통을 냉장고에 넣으면 라즈베리파이가 움직임을 감지하고 손을 인식하여 촬영을 하고 영상을 서버로 보낸다.
서버에서는 영상을 받아 영상 처리를 한 후, 데이터를 DB 에 저장한다.
사용자는 컴퓨터나 모바일 웹에서 냉장고 식자재와 관련한 데이터를 확인할 수 있다.

8. 프로젝트 마무리

A. 기대효과

- i. 독거 인구가 증가함으로 인해 소형 냉장고를 가진 가구가 증가하고 있다. IC4U의 기능을 이용하여, 이러한 소형 냉장 가전에 스마트 기능을 부가하여 보다 나은 서비스와 세부적인 정보를 사용자들에게 전달할 수 있다.
- ii. 디바이스의 크기가 작고 가볍기 때문에 휴대할 수 있는 형태로 제작을 할 수 있다. 이동식으로 제작하여 간단한 네트워크 설정만으로 어디서든 스마트 냉장고 기능을 사용할 수 있도록 해줄 것이다. 또한, 기기 외부에 버튼이나 터치 스크린 패드를 장착하거나, 스마트폰 어플리케이션 연동을 통한 기기 제어 기능을 추가하여, 냉장고 외부에 보관하는 식품들을 데이터베이스화 하여 관리할 수 있도록 하는 기능을 사용할 수도 있다.

B. 문제점

- i. 디바이스의 작동 환경 제어
 - 디바이스가 냉장고 내부의 저온 다습한 환경 내부에서 장시간 동작할 수 있는 지에

관해 명확한 테스트 결과가 없다.

- 또한, 냉장고 내부의 밀폐된 환경에서 외부 네트워크와 연결 상태를 유지할 수 있는지 알 수 없다.
- 마지막으로 냉장고 내부에서 디바이스의 지속적인 전원을 유지할 수 있는 환경을 마련할 방법을 찾지 못할 경우, 배터리에 의존해야 한다.
- 식자재 모양 변경 시(깨물어 먹거나, 비닐봉지에 넣는 경우) 인식률이 떨어진다
- 사용자가 장갑을 착용하면 인식률이 떨어진다
- 냉장고 환경의 변화(빛의 세기 및 빛의 위치 등등)가 생기면 인식률이 떨어진다
- 네트워크와 통신해야 하기 때문에 wifi 환경이 없는 곳에선 사용이 불가능하다

ii. 영상처리

- 현재 서버가 특정 색상 범위 값을 저장한 상태로 냉장고 내부에 진입하는 물체의 종류를 판별하는 방식으로 판별할 수 있는 물체의 개수가 제한적이며, 추후 판별할 물체의 종류를 늘릴 때 서버에 저장해야 하는 데이터의 용량도 커지기 때문에 확장성에 문제가 될 수 있다.

-

iii. 웹 페이지

- 웹사이트가 사용자들에게 시각적으로 흥미를 유발하는 디자인으로 이루어지지 않았다.

C. 개선방안

i. 디바이스의 보완

- 동작 인식을 통한 사진 촬영 기능의 경우, 타이밍 제어 기능을 보완하여 촬영되어 추출되는 이미지 파일의 질을 향상시켜야 한다.

ii. 영상처리 기능의 보완

- 색상 인식에 의한 물체 판별 방법을 보완한다. 색상 인식의 범위를 세분화 및 다양화하여 인식할 수 있는 물체의 종류와 개수를 증가시켜야 한다.
- 현재 영상 처리 프로세싱의 속도가 느려, 데이터베이스와의 연동이 지연되고 있기 때문에 처리 기능의 속도를 개선해야 한다.

iii. 서버 기능의 보완

- 다수의 디바이스가 서버에 접근할 경우, 스레드가 과도하게 생성되어 서버가 다운되는 현상이 발생하고 있기 때문에 스레드에 대한 세밀한 제어가 가능하도록 기능을 보완해야 한다.
 - 현재 FTP 서버, SQL 서버가 각기 동작하고 있기 때문에 Windows SERVER를 이용하여 서버 기능을 통합하여 안정적인 운영을 할 수 있도록 한다.
- iv. 웹 페이지
- 최근 웹 디자인은 무료로 템플릿을 제공하는 경우도 있어 이를 잘 활용하거나, CSS로 디자인을 화려하게 구사한다.

D. 참고문헌 및 논문

- i. 디바이스
- 차량의 윤곽선과 Chamfer Matching을 이용한 차량의 형태 분류 국회도서관 2010년
 - 개선된 챔퍼매칭 우도기반 2차원 평면 객체 추적 KISTI 2010년
 - Chamfer Matching을 이용한 실시간 템플릿 기반 개체 검출 및 추적 KISTI 2008년
 - Chamfer 거리 보간과 서브픽셀 탐색을 이용 한 개선된 Chamfer Matching KISTI 2007년
- ii. 영상처리
- OpenCV를 이용한 컴퓨터 비전 입문과 활용 -로버트 라가니에-
 - OpenCV Programming -김동근-
 - Learning Opencv: Computer Vision with the Opencv Library -Bradski-
- iii. 서버
- 실무자를 위한 C# 네트워크 프로그래밍 – Richard Blum, 사이텍미디어
 - 뇌를 자극하는 C# 프로그래밍 – 박상현, 한빛미디어
 - 뇌를 자극하는 Red hat Fedora 리눅스 서버 & 네트워크 – 우재남, 한빛미디어
 - 뇌를 자극하는 SQL SERVER 2012 – 우재남, 한빛미디어
 - 윤성우의 열혈강의 TCP/IP – 윤성우, 오렌지미디어

- 윤성우의 열혈 C 프로그래밍 - 윤성우, 오렌지미디어
- 닷넷 프로그래밍 정복 - 김상형, 한빛미디어
- 유닉스 리눅스 프로그래밍 필수 유틸리티 - 백창우, 한빛미디어
- Head First SQL - 린 베일리, 한빛미디어

iv. 웹 페이지

- PHP 웹 프로그래밍, 2014, 생능출판, 김진석/문양세 저
- HTML5와 CSS3를 이용한 모바일 웹 프로그래밍, 2012, T아카데미

E. 참고사이트

i. 디바이스

- www.raspberrypi.org
- cafe.naver.com/openrt
- www.rasplay.org
- raspberrypi.mynetgear.com/dokuwiki
- http://docs.opencv.org/trunk/doc/py_tutorials/py_tutorials.html
- <http://picamera.readthedocs.org/en/release-1.8/>

ii. 영상처리

- 영상처리 구현
 - <http://opencv-code.com/page/2/>
 - http://opencv.jp/sample/basic_structures.html#access_pixels
- 영상처리 Filter
 - <http://blog.secmem.org/538>
- OpenCV 객체 인식 기술
 - <http://codens.info/592>
- OpenCV 영상 산술
 - <http://hongkwan.blogspot.kr/2013/01/opencv-2-6-example.html>

- 문자 인식 기능
 - <http://sehyo88.tistory.com/entry/>
 - <https://code.google.com/p/tesseract-ocr/>
 - <http://blog.secmem.org/489>
- Labeling
 - <http://martinblog.tistory.com/>
- 네이버 영상처리 카페
 - <http://cafe.naver.com/opencv>

iii. 서버

- www.msdn.com
- www.python.org
- www.stackoverflow.com
- www.codeproject.com

iv. 웹 페이지

- www.w3school.com
- www.php.net
- msdn.microsoft.com/ko-kr
- apmsetup.com
- blog.haides25.com/archives/26
- laigo.kr/566

v. www.google.com(검색)

F. 팀원 별 소감

팀원	소감
김주영 	<p>프로그래밍 자체를 처음으로 해 본 데다가 난데없이 조장까지 맡았다. 프로그래밍부터 프로젝트 진행까지 정말 전부 다 몸으로 부딪히면서 하나하나씩 알아가야 했다. 서버 프로그래밍을 지망해서 해당 파트를 맡았는데 모르는 것뿐이어서 참 많이도 해맸던 것 같다. 덕분에 팀원들이 마음 고생을 많이 했을 것 같다. 묵묵히 자기 역할을 해주면서 나를 격려까지 해주었던 팀원들이 있어서 끝까지 올 수 있었다. 이 자리를 빌어 팀원들에게 감사의 말을 전한다. 모두들 앞으로 이어지는 취업 전선에서도 승전하길 바란다.</p>
김연옥 	<p>사소한 기능도 순순히 돌아가 주지 않는 "스마트 냉장고"는 힘든 만큼 기억에 많이 남고, 도움이 많이 된 프로젝트였다. 이런 방대한 프로젝트를 적은 인원에도 불구하고 2달동안 함께 고민하고 힘쓰며 달려왔기 때문에 그만큼 자신감을 얻어갈 수 있었다.</p>
김성민 	<p>처음으로 접해본 영상처리 너무 막연하기만 했었다. 하지만 한 단계씩 넘어가면서 어렵지만 가능하다는 걸 느꼈다. 부족한 점이 많은 걸 이끌어주신 저희 팀원 여러분 사랑합니다.</p>
변소정 	<p>개발 작업에서 한 분야를 맡은 게 처음이라 어떻게 일에 착수해야 할지도 몰랐고, 시행착오도 많았다. 해낸 걸 보면 별 거 아닌 것 같고 한 것도 없어 보이는데, 어쨌든 어렵기만 했다.</p> <p>시간이 더 있었더라면 실력도 더 높아지고 더 좋은 결과를 낼 수 있었을까 하면, 다른 건 몰라도 더 나은 결과를 낼 수 있었을 것 같다.</p> <p>그렇지만 이 프로젝트를 경험함으로써 프로그래밍에 대한 열정이 조금 붙은 것 같다.</p> <p>부족하지만 약 2개월간 함께 해준 우리 팀이 정말 고맙다. 우리 모두 취업 성공하고 항상 즐거웠으면 좋겠다.</p>

