

116123-02

보안 과제(), 일반 과제(○) / 공개(○), 비공개()발간등록번호()
첨단생산기술개발사업 제2차 연도 최종 보고서

11-1543000-002568-01

**IoT 기술을 활용한 도-농 연계 영농 플랫폼 개발
최종보고서**

2018.12.28.

주관연구기관 / 강남대학교 산학협력단
협동연구기관 / 주식회사 아람인
사업신청기관 / 융합기술진흥원

농 립 축 산 식 품 부

(전문기관) 농림식품기술기획평가원

IoT 기술을 활용한 도-농 연계 영농 플랫폼 개발 최종보고서

2018

농림식품기술기획평가원
농림축산식품부

제 출 문

농림축산식품부 장관 귀하

본 보고서를 “IoT 기술을 활용한 도-농 연계 영농 플랫폼 개발”(개발기간 : 2016. 12. 05 ~ 2018. 12. 04.)과제의 최종보고서로 제출합니다.

2018. 12. 28.

주관연구기관명 : 강남대학교 산학협력단 (대표자) 이상철 (인)
협동연구기관명 : 주식회사 아람인 (대표자) 김상일 (인)
과제신청기관명 : 융합기술진흥원 (대표자) 이계천 (인)

주관연구책임자 : 김규태 교수
협동연구책임자 : 김상일 대표
참여기관책임자 : 이계천 원장

국가연구개발사업의 관리 등에 관한 규정 제18조에 따라 보고서 열람에 동의합니다.

보고서 요약서

과제고유번호	1545013562	해 당 단 계 연 구 기 간	2017.12.05. ~ 2018.12.04	단 계 구 분	2/2
연구사업명	단 위 사 업	농식품기술개발사업			
	사 업 명	침단생산기술개발사업			
연구과제명	대 과 제 명	(해당 없음)			
	세부 과제명	IoT 기술을 활용한 도-농 연계 영농 플랫폼 개발			
연구책임자	김규태	해당단계 참여연구원 수	총: 6명 내부: 6명 외부: 0명	해당단계 연구개발비	정부:50,000천원 민간:17,000천원 계:67,000천원
		총 연구기간 참여연구원 수	총: 8명 내부: 8명 외부: 0명	총 연구개발비	정부:100,000천원 민간: 34,000천원 계:134,000천원
연구기관명 및 소속부서명	강남대학교 산학협력단			참여기업명: (주) 아람인	
국제공동연구	상대국명:			상대국 연구기관명:	
위탁연구	연구기관명:			연구책임자:	

※ 국내외의 기술개발 현황은 연구개발계획서에 기재한 내용으로 같음

연구개발성과의 보안등급 및	해당사항 없음.
-------------------	----------

사유	
----	--

9대 성과 등록·기탁번호

구분	논문	특허	보고서 원문	연구시설 ·장비	기술요약 정보	소프트 웨어	화합물	생명자원		신품종	
								생명 정보	생물 자원	정보	실물
등록·기탁 번호	1	2	·	·	·	1	·	·	·	·	·

국가과학기술종합정보시스템에 등록된 연구시설·장비 현황

구입기관	연구시설· 장비명	규격 (모델명)	수량	구입연월일	구입가격 (천원)	구입처 (전화)	비고 (설치장소)	NTIS 등록번호
해당사항 없음								

본 “ IoT 기술을 활용한 도-농 연계 영농 플랫폼”은 농가와 도시가구를 연결하여 도시가구가 영농활동에 직접 참여하게 함으로써 안전한 먹거리의 제공 뿐 만 아니라 농가의 유휴 농지와 노동력을 활용하여 수익의 창출을 가능하게 한다. 농가와 도시가구의 가입 및 인증, 텃밭관리 및 영농활동의 직접 참여를 위해 모바일 어플리케이션 형태로 사용자 인터페이스를 제공하며, IP CCTV 설치 및 스트리밍 서버의 구축을 통해 도시가구의 실시간 모니터링을 가능하게 한다. 이를 통해 도시가구는 영농활동의 참여와 더불어 농가의 지원을 받아 먹거리의 직접 재배와 수확이 가능함으로써 먹거리에 대한 신뢰성을 높이고 농가는 유휴 농지의 활용과 노동력을 통한 수익을 창출하게 하는 Win-Win 플랫폼으로써 역할을 수행할 수 있다.

보고서 면수

<요약문>

<p>연구의 목적 및 내용</p>	<ul style="list-style-type: none"> · 생산자와 소비자 간의 텃밭 계약 매칭 및 자동화 솔루션 개발 · IoT 기술을 활용하여 텃밭 관리 및 모니터링 시스템 설계 및 개발 · 사용자를 위한 모바일 어플리케이션 설계 및 개발 				
<p>연구개발성과</p>	<ul style="list-style-type: none"> · 텃밭 모니터링을 위한 IP CAMERA 인프라 구축 및 NVR(Network Video Recorder) 시스템 구성 · 실시간 스트리밍을 위한 스트리밍 서버 구축 · 사용자 편의를 위한 모바일 어플리케이션 구축 · 농가 및 사용자 관리를 위한 웹 기반 관리자 시스템 구축 · 지난 영상의 조회를 위한 스토리지 시스템 구축 · 위치기반 밭의 조회 및 임대 신청 기능 구축 				
<p>연구개발성과의 활용계획 (기대효과)</p>	<ul style="list-style-type: none"> · 실시간 모니터링 기능의 제공으로 먹거리 불신 개선 · 채소 및 식용작물의 안전한 먹거리 제공 · 직접 재배활동에 참여하여 경제적인 채소 및 식용작물 공급이 가능 · 농가의 유휴 농지 및 노동력의 활용에 따른 수익창출 				
<p>국문핵심어 (5개 이내)</p>	정보통신기술	사물인터넷	모바일 어플리케이션	영상처리	자동화
<p>영문핵심어 (5개 이내)</p>	Internet Communication Technology	Internet of Things	Mobile Application	Video Streaming Processing	Automation

※ 국문으로 작성(영문 핵심어 제외)

< 목 차 >

제 1장 연구개발과제의 개요	1
제 1절 연구개발의 목적	1
제 2절 연구개발의 필요성 및 배경	2
제 3절 연구개발 내용 및 범위	7
제 2장 연구수행 내용 및 결과	9
제 1절 연구개발 추진 전략 및 방법	9
제 2절 연구개발 추진 체계	10
제 3절 연구개발 추진 일정	11
제 4절 연구개발 성과	12
제 5절 연구개발 결과	13
1. 기술적 성과	13
2. 경제적 성과	29
제 3 장 목표 달성도 및 관련 분야 기여도	30
제 1절 목표달성도	30
제 2절 연구개발 추진 체계	38
제 4 장 연구결과의 활용 계획 등	39
<붙임1>. 참고 문헌	42
<붙임2>. TTA V&V 시험 결과서	43

제 1 장 연구개발과제의 개요

제 1 절 연구개발 목적

본 “IoT 기술을 활용한 도-농 연계 영농 플랫폼”은 농가와 도시가구를 연결하여 도시가구가 영농활동에 직접 참여하게 함으로써 안전한 먹거리의 제공 뿐 만 아니라 농가의 유휴 농지와 노동력을 활용하여 수익의 창출을 가능하게 한다. 농가와 도시가구의 가입 및 인증, 텃밭관리 및 영농활동의 직접 참여를 위해 모바일 어플리케이션 형태로 사용자 인터페이스를 제공하며, IP CCTV 설치 및 스트리밍 서버의 구축을 통해 도시가구의 실시간 모니터링을 가능하게 한다. 이를 통해 도시가구는 영농활동의 참여와 더불어 농가의 지원을 받아 먹거리의 직접 재배와 수확이 가능함으로써 먹거리에 대한 신뢰성을 높이고 농가는 유휴 농지의 활용과 노동력을 통한 수익을 창출하게 하는 Win-Win 플랫폼으로써 역할을 수행할 수 있다.

본 플랫폼은 농가와 도시가구에게 제공하는 사용자 인터페이스로써 모바일 어플리케이션을 제공하며, 농가와 사용자의 효과적인 관리를 위한 관리자 시스템이 제공된다. 또한 도시가구의 신뢰성 확보를 위해 실시간 모니터링 시스템이 제공되며, 이는 IP CCTV 설치와 스트리밍 서버를 통해 실시간 영상 뿐 만 아니라 24시간 녹화되는 모든 영상의 확인이 가능하다. 본 플랫폼을 통해 소비자의 식품에 대한 안전성, 경제성, 그리고 편의성에 대한 대표적인 세 가지 니즈의 효과적인 해결책을 제시한다.

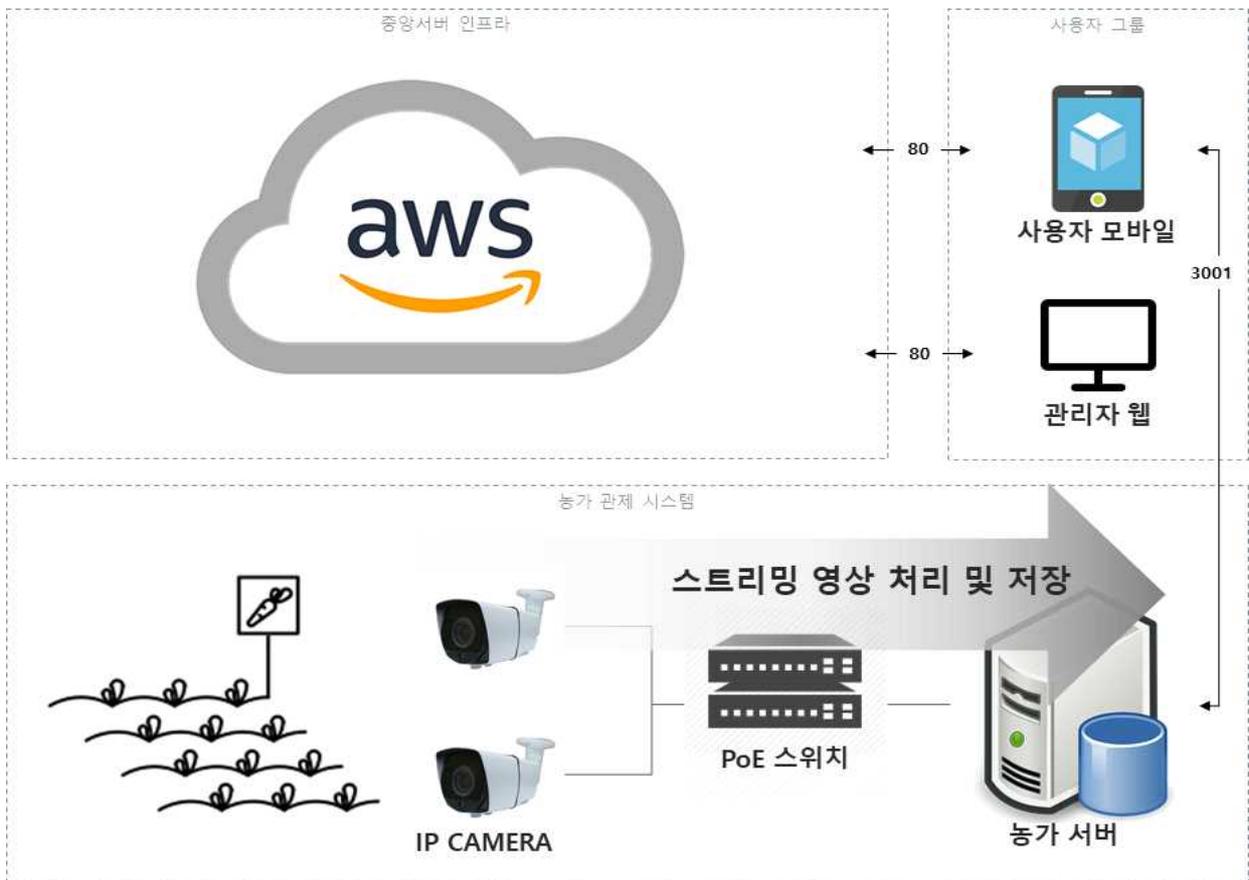


그림 1 시스템 구성도

제 2 절 연구개발의 배경 및 필요성

1. 연구개발 대상의 국내·외 현황

가. 기술현황

텃밭에 대한 IT 측면의 기술 접목은 주로 앱을 이용한 작물의 파종시기와 재배 방법 등을 기술해 놓은 것이 대부분입니다. 따라서 본 제안에서 제시한 IOT 기술을 활용한 농가와 도시가구간 상생을 위한 텃밭관리 솔루션은 특별한 경쟁 솔루션이 없습니다.

출시되어 있는 텃밭 또는 농사에 관련된 앱은 채소 및 식용작물을 파종하는 시기와 시기별 취해야 하는 조치 등 직접 농사를 짓는 것에 초점을 맞추고 있습니다. 따라서, 대부분의 농사에 관련된 앱은 농업에 관련된 기술전달 자체에 그치고 있는 실정이어서 농민이나 유관산업에 미치는 영향은 미미하다고 할 수 있습니다.

제안한 텃밭관리 솔루션은 기본적으로 IOT 기술을 활용하여 농가와 도시 가구 간 인터넷을 통한 연결이 가능하게 할 뿐만 아니라 실시간으로 언제, 어디서나 자신이 운영(농가의 관리 지원을 받으며)하는 텃밭의 상태확인, 주문 및 배송 등을 본인의 스마트폰을 통해 가능하게 하는 솔루션입니다. 부가적으로 농민들의 소득증대와 유통의 중간단계를 제거함에 따라 절감된 비용을 농민과 도시가구의 이익으로 되돌려 줄 수 있고, 본 솔루션 활성화 시 비 경작지의 활용, 농민의 소득증대에 따른 농촌의 공동화 현상 완화 및 정부의 정책방향인 IT와 농업의 결합에 의한 ICT 융합 산업의 농업 분야 중요한 사례로 활용 가능할 것으로 사료됩니다.

나. 세계시장 현황

FiBL과 IFOAM이 공동 출간한 보고서에 따르면, 세계 유기농식품의 시장규모는 1990년 150억 달러에서 2013년 720억 달러로 빠르게 성장하고 있다. 2008년 글로벌 경제위기로 산업투자와 소비가 크게 줄어들어 소폭 위축되기도 하였지만, 이후 다시 높은 성장률을 보이고 있다. 가장 큰 유기농식품 시장은 미국으로, 270억 달러의 규모로 파악된다. EU는 245억 달러의 규모로 두 번째 큰 시장을 형성하고 있다.

1인당 유기농식품 구입이 많은 국가는 스위스로, 1인당 230달러를 소비하였으며 다음으로 덴마크가 180달러를 소비하였다. 8위인 미국은 85달러를 소비하였다. 유기농 생산과 소비는 유럽과 북미 대륙을 중심으로 지속적으로 성장해왔으나, 이들 지역을 제외한 지역에서 유기농시장이 차지하는 비중은 아직 크지 않다. 농경지에서의 유기농산물 생산은 북미와 유럽 등 선진국에서 많이 이루어지며, 야생채집 등에 의한 생산은 인도와 아프리카 등의 개도국에서 많이 이루어지고 있다.

다. 국내시장 현황

친환경 채소에 대한 전체 소비자 중에서 본 서비스의 주이용층은 기존에 전자상거래를 통해 채소를 구입하고 있는 소비자라고 판단하였다. 점차 온라인을 통한 신선식품 구입이 증가하는 추세지만, 아직은 혁신수용자 혹은 초기수용자를 중심으로 형성된 미성숙한 시장이기 때문에 전자상거래 이용경험이 있는 기존 소비자를 타겟층으로 설정하였다. 통계

청에 따르면 2015년 기준 음식료품 전체시장에서 전자상거래가 차지하는 비중이 약 8%라고 한다. 이에 친환경 채소 시장 규모를 적용하면 465억 원 규모의 친환경 채소 구입이 전자상거래를 통해 이뤄지고 있다고 추정할 수 있다. 더불어 주말농장을 운영하고 있는 소비자 또한 본 서비스로의 유입이 가능하다. 이들은 기본적으로 채소의 안정성과 경제성에 대한 니즈를 가진 사용층이며, 주말농장에서 직접 채소를 재배하는 과정이 부담스러운 소비자가 일부 포함되어 있을 것으로 추정된다. 따라서 본 서비스가 가진 특징점이 주말농장 이용층의 이탈을 유도할 가능성이 높다. 2011년 농림수산부가 발표한 자료에 따르면, 서울시 15만 명을 포함하여 전국 70만 명 정도가 주말농장을 운영하거나 취미로 농사를 짓고 있는 것으로 파악된다. 평당 시가는 1만원에서 1.5만원으로 책정되어 있다. 평당 임대비용을 1.25만 원, 1인당 운영 평수를 10평으로 가정했을 때, 주말농장의 시장규모는 1,050억 원의 규모로 추정된다. 따라서 본 기술개발의 경쟁시장은 전자상거래를 통한 친환경 채소 구입 시장과 주말농장 시장을 합한 1,465억 원 규모로 추산된다.

구 분	현재의 시장규모(2014년)	예상 시장규모(2020년)
세계 시장규모	65,407	363,405
국내 시장규모	1,460	2,477
산출 근거	<ul style="list-style-type: none"> · FIBL and IFOAM, The World Of Organic Agriculture(2016) · 통계청, 가계동향조사(2015) · 농림수산식품부, 도시농업활성화 추진 보도자료(2011) · 한국농촌경제연구원, 국내외 친환경농산물 생산실태 및 시장전망(2015) 	

표 1 국내·외 주요시장 규모

라. 경쟁기관현황

· 만나박스

아쿠아포닉스라는 자연친화적인 농법으로 재배한 채소 꾸러미를 당일수확하여 당일배송하는 서비스를 제공하고 있다. 소비자들은 주로 월 4회 혹은 월 2회 정도의 정기구매를 통해 서비스를 이용한다. 특유의 패키지를 통해 채소의 뿌리까지 살아있는 상태로 배송되어 상당히 신선한 이미지를 소비자에게 심어주고 있다. 2014년 창업 이후 2015년 30억원의 매출에 이어 2016년 150억원의 매출을 예상하는 등 급격한 성장율을 보이고 있고 소비자 반응 또한 상당히 좋은 편이다. 안전성만큼은 채소 배송 서비스 중에서 가장 뛰어난 것으로 판단되며, 모바일 구매를 지원하지는 않지만 편의성 또한 좋은편이다. 하지만 가격이 만만치 않아 경제성이 떨어지는 약점을 보인다. 또한 꾸러미에 포함되는 채소의 종류를 선택할 수 없어 어떻게 요리를 해야할 지 고민하게 만드는 점이 불편하다는 피드백을 종종 확인할 수 있었다.

· 언니네 텃밭

전국여성농민회총연합에서 운영하는 서비스로, 꾸러미를 주문하면 일정 기간 동안 골고루 먹을 수 있는 식재료들을 배송해준다. 가입비를 받아 회원제로 운영하고 있으며 정기 배송도 가능해서 한 번 주문으로 정시적으로 식재료를 받아볼 수 있다. 현지에서 가장 먹기 좋게 자란 제철채소를 모아서 보내주기 때문에 신선함 식재료를 이용할 수 있다. 제철에 어떤 식재료가 몸에 좋은지, 어떻게 조리해야 맛있는지를 고려하여 구성하므로 소비자의 신뢰도가 상당히 높다. 농업에 대한 이해와 생산자/소비자간의 상생 관계에 대한 이해가 높은 서비스로서 여러 참고점이 될 만하다. 다만, 4인 기준으로 꾸러미 배송시에 연 100만원이 넘는 비용이 지출되어, 만나박스과 마찬가지로 경제성의 측면에서 약점을 보인다.

경쟁사명	제품명	판매가격 (천원)	연 판매액 (천원)
① 언니네 텃밭	제철꾸러미 4인 기준 12개월분	127	1,300,000
② 만나박스	패밀리박스 12개월분	108	3,000,000
③ 인터넷 오픈마켓	무농약 모듬 쌈채소 1kg *24회분	67	-
④ 주말농장	주말농장 10평 임대료	125	105,000,000

표 2 국내 주요시장 경쟁사

2. 연구개발의 필요성



그림 2 기존 텃밭 시스템의 한계와 새로운 서비스에 대한 니즈

소비자가 채소 및 식용작물을 구입하는 구매 패턴을 조사한 결과, 공통적으로 크게 세 가지의 니즈를 가지고 있음을 확인할 수 있었다. 바로 ‘안전성’, ‘경제성’ 그리고 ‘편의성’에 대한 니즈다. 안전한 채소를 보다 저렴하고 편리한 방법으로 구매하고자 한다는 것이다. 본 과제는 소비자에게 안전하고 경제적인 농산품의 제공을 위해 “계약재배를 통한 농작물 주문형 영농 시스템”을 개발하고 제공함을 목표로 한다.

안전성에 대한 관심은 유기농이나 친환경 농산물 소비로 이어지고 있다. 주말을 이용해 1년 단위로 임대한 도시 근교의 농지에서 소규모로 채소를 기르는 주말농장의 이용이 지속적으로 증가하고 있는데, 이 또한 채소를 직접 기르면서 재배과정을 꼭 지켜본 채소이기 때문에 믿고 먹을 수 있다는 ‘안전성’을 바탕으로 한다. 임대비용 외에는 크게 추가비용이 들지 않는다는 ‘경제성’의 장점도 있다. 서울연구원에 따르면 서울시의 주말농장과 도시텃밭이 5년간 5배의 증가율을 보일 만큼 높은 성장률을 보이고 있다. 그러나 업무나 학업에서 자유로울 수 없는 탓에 주말에 따로 시간을 내서 텃밭을 가꾸는 것이 부담이 되는 경우가 많아 ‘편의성’이 상당히 떨어진다.

이러한 이유로 최근에는 텃밭에서 재배되는 채소를 소비자에게 직접 배달해주는 서비스 형태가 새로운 트렌드로 각광받고 있다. 언니네 텃밭, 만나박스 등의 서비스가 있다. 하지만 유기농이나 텃밭 채소 배달 서비스의 경우 가구소비의 지출을 증가시키는 한계를 가지고 있다. 통계청에 따르면 가구당 연 43만 원 정도를 채소류 구입에 지출한다고 하는데, 위 서비스를 이용할 경우 4인 기준 연간 100~120만 원 가량이 지출되는 것으로 파악되었다. ‘안전성’과 ‘편의성’은 보장되지만 ‘경제성’을 만족시키지 못하는 것이다. 세 가지의 소비자 니즈를 모두 만족시키는 서비스는 여전히 부재하고 있는 실정이다.

· 안전성

소비자가 소비할 채소의 재배과정을 확인할 수 있도록 하는 방법이 가장 효과적으로 파악되었다. 이 방법은 유기농 채소가 거치는 까다로운 검증절차보다도 훨씬 직관적으로 소비자에게 수용되었다. 또한 재배과정을 지켜보는 것은 채소에 대한 애정도를 증가시켜 채소를 소비하는 것 자체에 즐거움을 더해주는 것으로 나타났다.

· 경제성

현재의 지출보다 적은 비용으로 동질의 상품을 구입할 수 있거나, 같은 비용으로 양질의 상품을 구입할 수 있을 때 경제성을 만족할 수 있다. 평균적으로 가구당 연간 43만원을 채소 구입에 지출하는 것으로 조사되었다. 채소의 재배과정에서 비용을 절감하거나, 유통과정을 단순화함으로써 경비를 절감하는 방법이 있다.

· 편의성

채소의 재배와 소비 과정 동안 소비자의 시간과 노력이 얼마나 최소한으로 투입되는지를 의미한다. 소비자를 대신하여 전문적인 농업종사자가 채소와 식용작물을 재배하여 최단 시간 내 집으로 배송해주는 경우 편의성이 극대화될 수 있다.

구분	주말농장	친환경 농산물	텃밭채소 배송	본 과제
안전성	○	○	○	○
경제성	○	×	×	○
편의성	×	△	○	○

표 3 채소 구매방법에 따른 소비자 만족

이상 세 가지의 소비자 니즈를 해결하기 위해 “스마트폰과 관리 시스템, 텃밭카메라 등의 사물인터넷 기술을 활용한 텃밭관리 솔루션”을 제안하고자 한다. 솔루션은 크게 세 단계로 진행된다.

· 텃밭 매칭 단계

텃밭과 소비자, 그리고 생산자와 소비자를 연결시켜주는 단계다. 소비자에게 경작하고자 하는 작물이나 텃밭 희망 위치 등의 정보를 바탕으로 생산자를 추천한다. 소비자가 생산자와 텃밭을 선택하면 연 단위로 텃밭을 관리한다는 계약이 진행된다.

· 텃밭 관리 단계

소비자는 생산자 즉 농부가 채소 및 식용작물을 재배하는 과정을 모바일 어플리케이션을 통해 확인할 수 있다. 모바일 어플리케이션과 텃밭 및 농장에 설치된 IP CAMERA와 관리 시스템이 실시간으로 정보를 송·수신하기 때문이다. 소비자는 원격으로 실시간 영상을 확인할 수 있으며, 24시간 녹화된 영상의 확인을 통해 위탁 재배한 채소 및 식용작물에 대한 모니터링이 가능하다. 이를 통해 텃밭을 확인하여 재배 작물에 대한 신뢰도를 높일 수 있다.

· 수확/배송 단계

재배하는 채소가 수확기에 접어들면 배송을 요청하거나 수확 알림을 받을 수 있다. 농가의 영농지원을 통해 수확한 채소 및 식용작물은 사용자의 요청에 직접 수확 및 배송되어 진다. 모바일 어플리케이션을 통해 해당 내역이 알림으로 전달되어 지며, 택배를 통해 도시가구로 배송되어 진다.

제 3 절 연구개발 내용 및 범위

구분	내용																								
최종목표	<p>CCTV, 인터넷, 관리서버시스템 및 스마트폰 등 IoT 기술을 활용하여 농가와 도시가구 간 인터넷을 통한 연결이 가능하게 할 뿐만 아니라 실시간으로 언제, 어디서나 자신이 운영(농가의 관리 지원을 받으며)하는 텃밭의 상태를 확인하고, 본인 텃밭의 생산물을 주문 및 배송 등을 스마트폰을 통해 가능하게 하는 것이 최종 목표이다.</p>																								
세부목표	<p>- 모바일 단말 개발 내용 (안드로이드)</p> <table border="1" data-bbox="335 817 1431 2033"> <thead> <tr> <th data-bbox="335 817 639 891">개발 항목</th> <th data-bbox="639 817 1193 891">설 명</th> <th data-bbox="1193 817 1431 891">비 고</th> </tr> </thead> <tbody> <tr> <td data-bbox="335 891 639 1081">기능 설계</td> <td data-bbox="639 891 1193 1081"> <ul style="list-style-type: none"> • 단말과의 연동 기능 설계 • UI 설계 • 기능 설계 </td> <td data-bbox="1193 891 1431 1081">인증절차 설계 포함</td> </tr> <tr> <td data-bbox="335 1081 639 1238">UI 디자인 및 개발</td> <td data-bbox="639 1081 1193 1238"> <ul style="list-style-type: none"> • UI 디자인 • UI 개발 </td> <td data-bbox="1193 1081 1431 1238"></td> </tr> <tr> <td data-bbox="335 1238 639 1395">홈 페이지 서버 연동 기능 개발</td> <td data-bbox="639 1238 1193 1395"> <ul style="list-style-type: none"> • 가입/관리/해제 절차 및 인증과 주문/배송 관리 절차 </td> <td data-bbox="1193 1238 1431 1395"></td> </tr> <tr> <td data-bbox="335 1395 639 1608">농장 PC 및 NVR과의 연동 기능 개발</td> <td data-bbox="639 1395 1193 1608"> <ul style="list-style-type: none"> • 저장된 영상 정보 검색 및 수신 영상 출력 기능 </td> <td data-bbox="1193 1395 1431 1608">카메라 업체 의 제공 기능 활용</td> </tr> <tr> <td data-bbox="335 1608 639 1720">영상 정보 출력 기능</td> <td data-bbox="639 1608 1193 1720"> <ul style="list-style-type: none"> • 카메라에서 수신한 영상 정보 출력 기능 </td> <td data-bbox="1193 1608 1431 1720"></td> </tr> <tr> <td data-bbox="335 1720 639 1843">작물/채소의 주문</td> <td data-bbox="639 1720 1193 1843"> <ul style="list-style-type: none"> • 소비자에 의한 작물/채소의 주문 기능 </td> <td data-bbox="1193 1720 1431 1843"></td> </tr> <tr> <td data-bbox="335 1843 639 2033">작물/채소의 배송 및 텃밭 관리 기능</td> <td data-bbox="639 1843 1193 2033"> <ul style="list-style-type: none"> • 생산자의 텃밭 정보 입력 및 사진/영상 등록 • 소비자의 주문에 의한 배송 기능 </td> <td data-bbox="1193 1843 1431 2033"></td> </tr> </tbody> </table>	개발 항목	설 명	비 고	기능 설계	<ul style="list-style-type: none"> • 단말과의 연동 기능 설계 • UI 설계 • 기능 설계 	인증절차 설계 포함	UI 디자인 및 개발	<ul style="list-style-type: none"> • UI 디자인 • UI 개발 		홈 페이지 서버 연동 기능 개발	<ul style="list-style-type: none"> • 가입/관리/해제 절차 및 인증과 주문/배송 관리 절차 		농장 PC 및 NVR과의 연동 기능 개발	<ul style="list-style-type: none"> • 저장된 영상 정보 검색 및 수신 영상 출력 기능 	카메라 업체 의 제공 기능 활용	영상 정보 출력 기능	<ul style="list-style-type: none"> • 카메라에서 수신한 영상 정보 출력 기능 		작물/채소의 주문	<ul style="list-style-type: none"> • 소비자에 의한 작물/채소의 주문 기능 		작물/채소의 배송 및 텃밭 관리 기능	<ul style="list-style-type: none"> • 생산자의 텃밭 정보 입력 및 사진/영상 등록 • 소비자의 주문에 의한 배송 기능 	
개발 항목	설 명	비 고																							
기능 설계	<ul style="list-style-type: none"> • 단말과의 연동 기능 설계 • UI 설계 • 기능 설계 	인증절차 설계 포함																							
UI 디자인 및 개발	<ul style="list-style-type: none"> • UI 디자인 • UI 개발 																								
홈 페이지 서버 연동 기능 개발	<ul style="list-style-type: none"> • 가입/관리/해제 절차 및 인증과 주문/배송 관리 절차 																								
농장 PC 및 NVR과의 연동 기능 개발	<ul style="list-style-type: none"> • 저장된 영상 정보 검색 및 수신 영상 출력 기능 	카메라 업체 의 제공 기능 활용																							
영상 정보 출력 기능	<ul style="list-style-type: none"> • 카메라에서 수신한 영상 정보 출력 기능 																								
작물/채소의 주문	<ul style="list-style-type: none"> • 소비자에 의한 작물/채소의 주문 기능 																								
작물/채소의 배송 및 텃밭 관리 기능	<ul style="list-style-type: none"> • 생산자의 텃밭 정보 입력 및 사진/영상 등록 • 소비자의 주문에 의한 배송 기능 																								

구분	내용		
	- 서버 개발 내용		
개발 항목	설 명	비 고	
기능 설계	<ul style="list-style-type: none"> • 단말과의 연동 기능 설계 • 인증 절차 설계 • UI 설계 • 기능 설계 		
UI 디자인 및 개발	<ul style="list-style-type: none"> • UI 디자인 • UI 개발 		
단말과의 연동 기능 개발	<ul style="list-style-type: none"> • 생산자 및 소비자 사용 인증 기능 개발 		
생산자 및 소비자 가입/등록/해제 기능 개발	<ul style="list-style-type: none"> • 생산자 및 소비자 등록/해제 및 로그인, 로그아웃 기능 개발 		
텃밭 관리 기능 개발	<ul style="list-style-type: none"> • 생산자 텃밭 등록 • 소비자의 텃밭 검색 및 원하는 텃밭 선택 기능 개발 		
인증 관리 기능	<ul style="list-style-type: none"> • 텃밭 관리 기능에 등록된 소비자의 스마트폰 인증 기능 개발 		
주문 및 배송 관리 기능	<ul style="list-style-type: none"> • 소비자의 작물/채소 주문 통합 관리 기능 • 생산자의 배송 처리 통합 관리 기능 개발 		
결제 기능	1차 기능은 생산자와 소비자간 온라인 입금으로 해결		

제 2 장 연구수행 내용 및 결과

제 1 절 연구개발 추진전략·방법

1. 연구개발추진 전략

가. 텃밭 매칭 단계

텃밭과 소비자, 그리고 생산자와 소비자를 연결시켜주는 단계다. 소비자에게 경작하고자 하는 작물이나 텃밭 희망 위치 등의 정보를 바탕으로 생산자를 추천한다. 소비자가 생산자와 텃밭을 선택하면 연 단위로 텃밭을 관리한다는 계약이 진행된다.

나. 텃밭 관리 단계

소비자는 생산자 즉 농부가 채소 및 식용작물을 재배하는 과정을 스마트폰을 통해 확인할 수 있다. 스마트폰 어플리케이션과 텃밭 및 농장에 설치된 카메라와 관리 시스템이 실시간으로 정보를 송·수신하기 때문이다. 소비자는 원격으로 카메라 줌인/줌아웃과 방향전환을 하면서 텃밭을 관찰할 수 있고 채소의 상태 및 현황을 확인할 수 있다.

다. 수확 및 포장·배송 단계

재배하는 채소가 수확기에 접어들면 배송을 요청하거나 수확 알림을 받을 수 있다. 배송을 요청하면 생산자는 최대한 빠른 시간 내에 수확 및 포장·배송을 진행한다. 이러한 일련의 과정은 모바일 앱 상에서 진행되며, 소비자는 몇 번의 스마트폰 조작을 통해 텃밭의 채소를 집으로 받아볼 수 있다.

2. 연구개발 추진 시 발생할 문제점 및 해결을 위한 접근

가. 밭에 위치하는 원격 카메라 시스템 및 카메라

- 협동기관 및 위탁기관은 야외 환경 하에서 동작가능 여부와 비/바람 등에 대해 충분히 견딜 수 있는 카메라 모델 사전 조사.
- 협동기관 및 위탁기관은 HD급 화질 전송이 가능하며, 가옥 내 서버로 랜선을 통해 스트리밍 데이터 전송이 가능한 대당 20 ~ 30만원 선의 경제적인 가격의 제품 선정
- 위탁기관은 실외에 설치되는 카메라의 내구성과 유지보수를 위한 전략적 방법 연구
- 협동기관 및 위탁기관은 화질 및 실제 농작물의 모습이 소비자에게 적절히 보여지도록 테스트 중심의 연구를 진행

나. 협동기관은 동영상 인코딩 및 전송에 관련된 부하 테스트를 반영한 연구진행

다. 협동기관은 서버 인프라 설계 및 시스템 구성 문제점 극복 연구

- 확장이 용이하며 안전성이 보장됨
- 스트림 데이터 관리를 위한 데이터 보관 정책 필요
- 사용자에게 따른 인프라 비용 예측 필요

라. 신청,참여,협동기관은 지식재산권 확보·회피 방안을 모색

본 과제와 직접적으로 관련된 지식재산권은 현재 존재하지 않으며, 본 과제에서 제안하는 플랫폼에 대한 특허는 현재 출원 상태이다. 이후 과제 진행에 따라 발전된 특허권 출원을 지속적으로 수행할 계획이다.

제 2 절 연구개발 추진체계



제 3 절 연구개발 추진일정

1차년도																
일련 번호	연구내용	월별 추진 일정												연구 개발비 (단위:천원)	책임자 (소속 기관)	
		1	2	3	4	5	6	7	8	9	10	11	12			
1	계획수립 및 자료조사	■	■												5,000	공통
2	시스템 설계			■	■	■	■								8,000	공통
3	카메라 성능 테스트					■	■	■							5,000	공통
4	야외 인프라 구축 및 테스트					■	■	■	■	■					6,000	공통
5	야외 카메라 설치										■	■	■		10,000	공통
6	야외 서버 인프라 구축											■	■		3,000	협동
7	중앙 서버 개발							■	■	■	■	■	■		15,000	협동
8	모바일 개발										■	■	■		10,000	협동
9	관리자 시스템 개발											■	■		5,000	협동
2차년도																
1	야외 서버 증설	■	■												10,000	공통
2	농가 서버 고도화 (대용량 스트리밍 처리)	■	■	■	■	■	■	■							10,000	협동
3	관리자 시스템 고도화					■	■	■	■						5,000	협동
4	모바일 어플리케이션 고도화				■	■	■	■							8,000	협동
5	결제 모듈							■	■						2,000	협동
6	배송시스템							■	■	■					4,000	협동
7	중앙서버 보안강화								■	■	■	■			8,000	협동
8	데이터베이스 증설 및 백업 전략 구축											■	■		6,000	공통

제 4 절 연구개발 성과

1. 국내외 논문 게재

No	논문 명	학술지 명	저자명	호	국명	발행기관	SCI 여부	게재연도	등록번호
1	IoT 기술을 활용한 도-농 연계 영농 플랫폼 개발	정보과학회논문지	이계천	.	대한민국	한국정보과학회	비SCI	2018	24660825

2. 국내 및 국제학술회의 발표

No	회의명칭	발표자	발표일시	장소	국명
1	2018 한국컴퓨터종합학술대회	이계천 김상일	2018.6.21	제주	한국

3. 지식재산권

No	지식재산권 등 명칭	국명	출원			등록			기여율
			출원인	출원일	출원번호	등록인	등록일	등록번호	
1	IoT 기술을 활용한 도농 연계 영농시스템과 그 방법	대한민국	이계천	2017.12.24	특하-2017-0172000				100
2	텃밭 임대 통합관리 시스템 및 그 방법	대한민국	이계천	2018.07.19	특하-2018-0084144				100

4. 저작권(소프트웨어)

No	저작권명	창작일	저작자명	등록일	등록번호	저작권자명	기여율
1	우리집 텃밭	2018.12.01	이계천	2018.12.28	C-2018-040409	이계천	100

제 5 절 연구개발 결과

1. 기술적 성과

가. 농가 모니터링을 위한 IP CAMERA 분석 및 설치

본 플랫폼의 핵심 기능인 실시간 모니터링을 위해 농가에 IP CAMERA를 설치하였다. 야외에 설치되는 IP CAMERA는 4계절의 온도와 습도에 내구성이 있어야 하며, 다수의 IP CAMERA가 설치되는 만큼 경제성을 함께 고려해야 한다. 3개사의 IP CAMERA를 고려하여 실 테스트를 수행하였으며 내구성과 화질 그리고 Onvif 표준의 지원여부를 고려하였다. 아래는 실사용 테스트를 수행한 각각의 카메라에 대한 성능 및 경제성 분석에 대한 내용이다.

· EasyCAM ES10D POE

국내 대표적인 IP CAMERA 업체인 EasyCAM 사의 ES10D 모델로 Vstar사의 제품에 비해 저렴한 가격이 큰 장점이다. 기본적으로 방수가 가능하며 화소는 100만 화소로 동일하다. 하지만 카메라의 조작이 불가능한 점이 가장 큰 단점으로 여겨진다. PoE를 지원하며 무선 연결이 불가능하다. 또한 카메라 설치 시 방향의 설정이 다소 어려운 구조로 야외에 설치하는 지지대의 별도 제작이 필요한 점은 비용이 수반되어 설치비용이 증가한다. EasyCAM 역시 자체 제공하는 모바일 어플리케이션과 프로그램이 존재하지만 본 과제에 맞춘 커스터마이징은 불가하며, 대다수의 IP CAMERA 업체에서 프로그램의 임의 수정이 어렵다는 회신을 받았다.

사진	상세스펙
	화소: 100만 화소 방수: 가능 확대: 디지털 줌 야간: 야간 감시 가능 조작: 카메라 조작 불가 무선: WIFI 미지원 가격: 10 만원대 전원: PoE 지원 기타: 모션 디렉션 기능

· Vstarcam-100H

Vstar사의 IP CAMERA로 높은 화질과 내구성을 제공한다. 카메라의 조작이 가능하며 여러 텃밭의 모니터링이 가능한 장점을 갖고 있다. 야간 감시가 가능하며, 무선 통신이 가능한 점도 큰 장점으로 여겨진다. Vstar사에서 제공되는 모바일 앱과 통신 시스템이 있지만 본 플랫폼에 맞춘 커스터마이징은 불가능하다.

사진	상세스펙
	화소: 100만 화소 방수: 가능 확대: 디지털 줌 (17배) 야간: 야간 감시 가능 조작: 카메라 조작이 가능 (상하좌우) 무선: WIFI 지원 가격: 20 ~ 30만원대 전원: PoE 지원

· BY-756TS-PX

최초 고려한 2개의 IP CAMERA는 국내 업체로 IP CAMERA와 더불어 제공되어지는 모바일 소프트웨어의 SDK의 별도 제공, 혹은 프로그램의 커스터마이징을 염두하였지만 모든 부분에서 불가능하던 회신을 받았다. 또한 처음 고려한 Vstar사의 IP CAMERA는 카메라의 조작이 가능하지만 가격이 그만큼 증가하여 사업 수지가 맞지 않았다. 그래서 동일한 화소에 방수를 지원하며, 저렴한 가격으로 설치가 가능한 카메라를 목표로 검색한 결과 BY-756TS-PX를 선택하게 되었다. 가장 큰 장점은 추후 MOU를 통해 가격 협상이 가능하던 점이다. 발 분양의 단가가 정해져 있으며, 수익률이 높지 않은 점을 감안할 때 가장 적합한 카메라로 여겨졌으며, 설치 시 편의가 가장 적합하였다.

사진	상세스펙
	화소: 100만 화소 방수: 가능 확대: 디지털 줌 야간: 야간 감시 가능 조작: 카메라 조작 불가 무선: WIFI 미지원 가격: 5 만원대 (MOU 체결 시) 전원: PoE 지원 * MOU 체결 가능

위 3개사의 IP CAMERA를 실제 구입하여 테스트해 보았다. 각 IP CAMERA의 경제성, 지원 기능 그리고 사업 운영상의 여러 요소를 비교 평가하여 최종 카메라를 선정하게 되었다. 이후 사업화 시 카메라의 구매에 따른 확장성과 경제성이 가장 큰 요소로 작용하였으며, 최종 BY-756TS-PX 모델을 선택하여 9대를 추가 구매 하였다.

구분	화소	조작 여부	PoE 지원	경제성	확장성
Vstarcam-100H	100만	O	O	하	하
EasyCAM ES10D POE	100만	X	O	중	하
BY-756TS-PX	100만	X	O	상	상

테스트를 위해 구매한 2대의 카메라와 최종 선정한 카메라 10대를 포함 총 12대의 카메라를 구매하였으며, 실제 텃밭에 11대의 카메라를 설치하였다. “경기도 화성시 반정동”에 위치한 200평 가량의 텃밭에 카메라를 설치하였다. 일부 카메라는 농가 시설에 부착하여 설치하였지만 실외에 카메라를 설치해야 하기에 별도의 카메라 지지대를 설치한 후에 카메라를 설치하였다. 아래와 같은 지지대를 총 3군데 설치하였으며, 각 지지대에 2~3대의 카메라를 설치하였다.

전체 밭의 배치와 카메라 설치 위치, 그리고 카메라 화각은 아래와 같다. 카메라의 조작성이 불가능한 만큼 전체 밭의 모니터링이 가능하도록 고르게 배치하였으며, 비닐하우스 안과 해피하우스 앞까지 텃밭의 전체 시설을 빠짐없이 모니터링 가능하도록 카메라를 배치하였다.



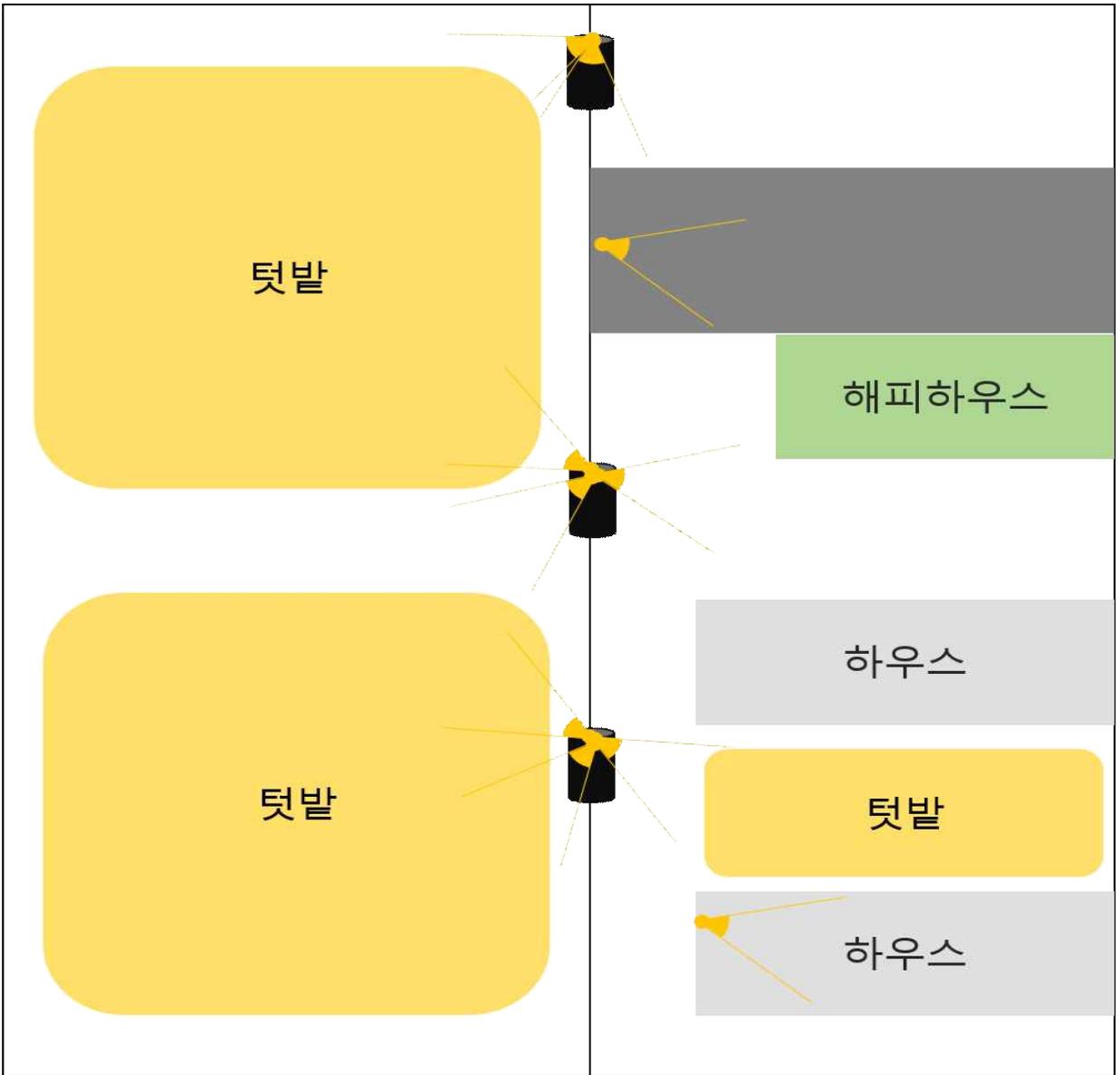


그림 12 텃밭 카메라 배치도

나. Onvif 표준 기반의 NVR 개발 및 실시간 스트리밍 서버 구현

본 플랫폼의 핵심 기능인 실시간 모니터링 및 저장 영상의 조회를 위해 카메라 영상 처리가 필요하다. 해당 카메라 제조사의 자체 모바일 어플리케이션이 존재하지만 실시간 영상의 조회만이 가능하며, 이 역시 외부에서 접속하기 위해서는 별도의 고정 IP가 필요하다. 또한 저장 영상의 제공을 위해서는 상용 NVR(Network Video Recorder) 이 필요한데 이는 별도의 추가 비용이 발생한다. 실시간 스트리밍 영상의 효과적인 저장을 위해 Onvif 표준 기반 NVR을 직접 개발하였다.

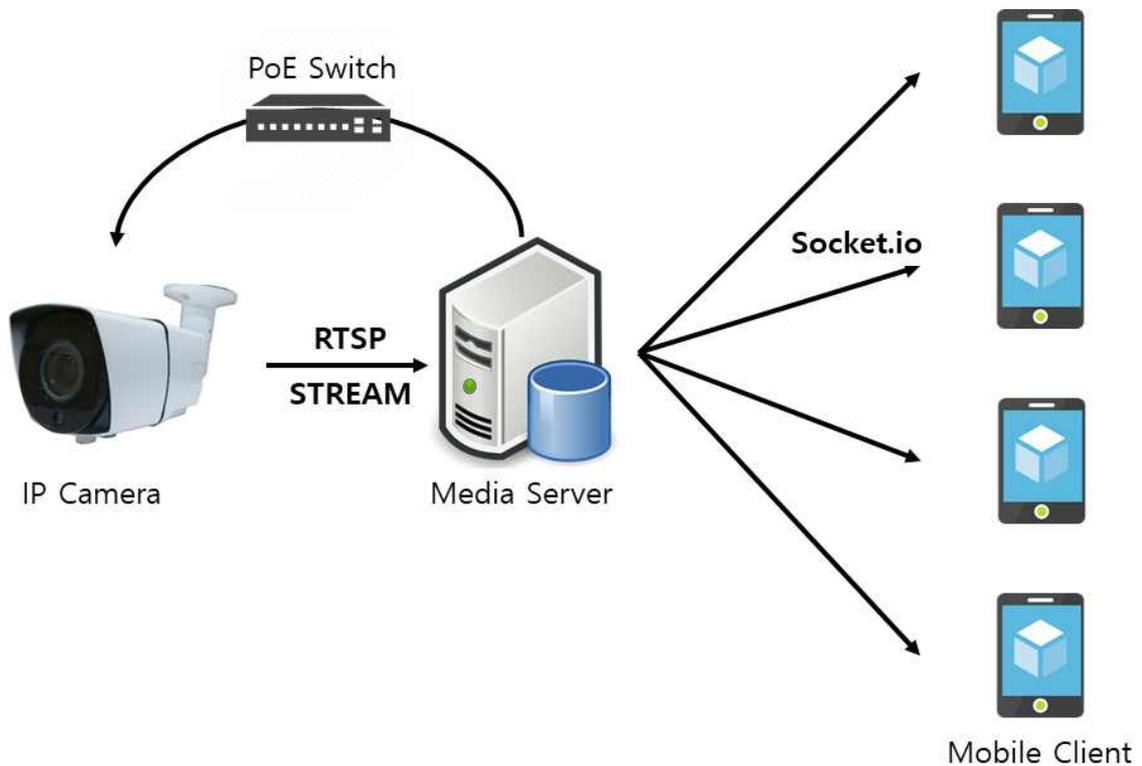


그림 13. Onvif NVR 서버 개요도

비교분석을 통해 선택한 카메라는 Onvif 표준(ONVIF Core Specification, v2.1, www.onvif.org)을 기반으로 하여 해당 스펙에 따라 카메라 제어 서버를 개발하였으며, 미디어서버는 Onvif 표준을 통해 카메라와 통신하며 카메라로부터의 스트리밍 데이터를 저장하는 역할을 동시에 수행한다. 미디어서버의 스펙은 아래와 같다.

- OS : Windows Server 2012 R2
- CPU : Intel i7
- RAM : 64GB
- HDD : 4TB

미디어서버는 카메라로부터의 RTSP 스트림을 저장하는 파일서버의 역할과 모바일 사용자의 요청에 따라 실시간 영상 및 저장된 영상을 제공하는 미디어 서버의 역할을 수행한다. 실시간 스트림을 처리하는 이유로 메모리 사용량이 다소 높을 것으로 예상하였다. 이에 따라 위와 같은 서버 성능을 확정하였으며, 카메라 10대의 처리에 큰 문제가 없었다.

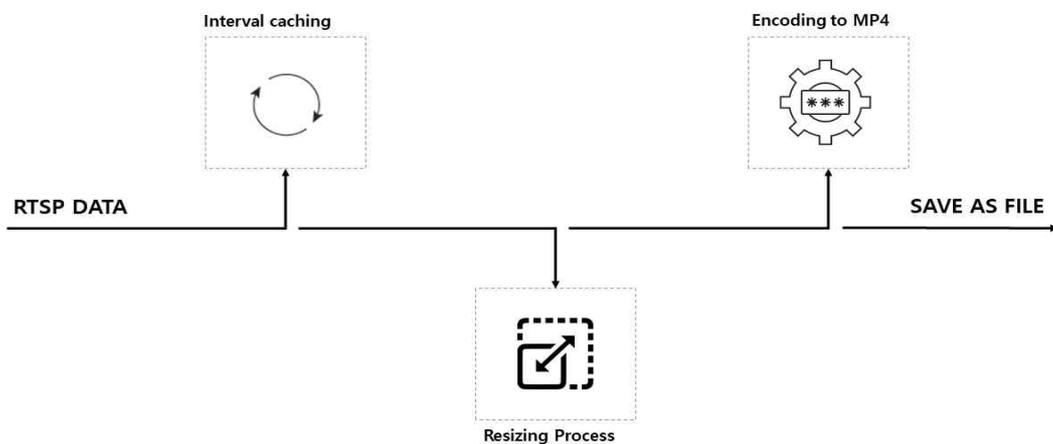
하이브리드 앱 형태로 만들어진 모바일 어플리케이션은 Socket.io를 통해 서버로부터의 실시간 스트림을 수신하여 사용자에게 보여준다. HTTP/HTTPS 프로토콜은 연결성을 유지하지 않아 스트리밍 처리에 다소 적합하지 않지만, Socket.io를 통해 서버로부터 처리된 실시간 영상 및 저장된 영상을 사용자의 플레이 시점에 맞추어 모바일로 전달해 줄 수 있게 하였다.

마지막으로 카메라의 전원 공급에 대해서는 PoE를 선택하였다. PoE(Power over Ethernet)은 랜선을 통해 데이터를 송/수신 할 뿐만 아니라 전원을 함께 공급한다. 실외 설비로 전원을 연장하여 설비하는 것은 안전성과 비용등 문제가 많았다. PoE를 도입함으로써 랜선의 연장을 통해 전원 공급과 데이터 송/수신 등 모든 문제를 해결하였다. 하지만 PoE는 랜선의 거리에 따른 제약이 있어 최대 100미터 이내로 길이를 유지해야 한다. 설비한 PoE 스위치는 아래와 같은 스펙을 갖추었다.

미디어 서버는 크게 카메라로부터의 스트리밍 영상을 처리하고 저장하는 미디어 프로세싱 과정과 처리된 영상 데이터를 모바일 사용자의 요청에 따라 스트리밍하는 스트리밍 서버의 역할로 나눌 수 있다. 각 기능에 대한 상세한 내용은 아래와 같다.

· **Media Processing Process**

Onvif 표준에 따라 실시간 스트리밍 프로토콜 (RTSP: Real Time Streaming Protocol) 기반의 데이터를 카메라로부터 수신받을 수 있다. RTSP는 IETF가 1998년에 개발한 통신 규약이다. RFC 2326(<https://www.ietf.org/rfc/rfc2326.txt>)에 정의되어 있다. RTSP는 스트리밍 시스템에 사용되며, 미디어 서버를 원격으로 제어할 때 쓰인다. 명령어는 "PLAY", "PAUSE" 같이 VCR 동작하고 비슷하며 시간 정보를 바탕으로 서버에 접근을 한다. 농가의 카메라는 24시간 무정지 상태로 영상을 농가서버로 전송함으로써 항상 "PLAY" 상태이며 농가서버는 항상 레코딩 기능을 수행하고 있다.



RTSP 데이터 처리 프로세스

실시간 스트리밍 프로토콜을 통해 전송된 영상 데이터는 변환을 거쳐 1시간 단위로 서버에 저장되어 진다. 스트리밍 영상 데이터의 변환을 위해 FFmpeg 라이브러리를 사용하였으며, 실제 영상이 모바일 환경에 수월하게 노출될 수 있도록 전체 크기와 화질을 조정한다. FFmpeg의 디지털 음성 스트림과 영상 스트림에 대해서 다양한 종류의 형태로 기록하고 변환하는 컴퓨터 프로그램이며, 명령어를 직접 입력하는 방식으로 동작하고, 그 외 여러가지 자유 소프트웨어와 오픈 소스 라이브러리로 구성되어 있다. 라이브러리 중에는 libavcodec도 들어있는데, 이 라이브러리는 음성/영상 코덱 라이브러리로 여러 프로젝트에서 쓰이고 있다. 또, libavformat 라는 음성/영상 다중화, 역다중화 라이브러리도 있다.

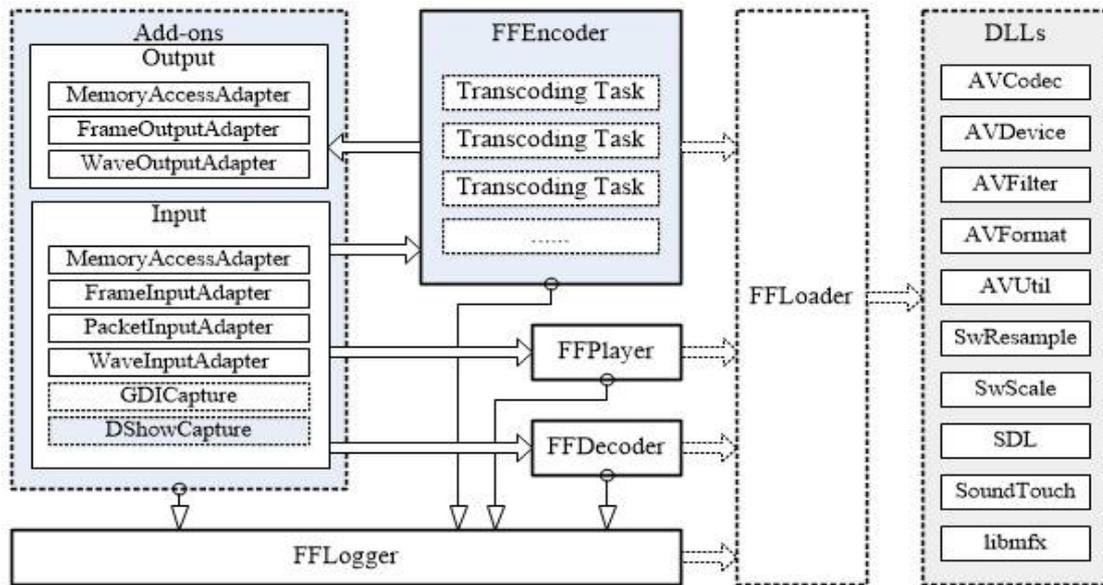


그림 FFmpeg Architecture

· Streaming Server

저장된 영상 혹은 실시간 영상을 사용자의 요청에 따라 모바일 장치로 전송하는 스트리밍 서버 역할 또한 수행한다. 최초 계획에 따라 AWS S3 스토리지를 사용하여 중앙 관리하는 방식을 설계/개발 하였지만 경제적인 이유로 미디어 서버로 해당 기능을 국한하였다. 최초 스토리지 용량과 트래픽의 제한으로 중앙 집중 방식을 생각하였지만 하나의 텃밭에 해당하는 분양 사용자의 수는 제한적이라 농가서버에서 직접 스트리밍을 수행하여도 큰 무리가 없다고 판단하였으며, 농가서버의 성능에 따른 비용 향상도 크지 않았다. 현재의 구현 방식은 농가서버 자체에 3개월 간의 녹화 데이터를 1시간 단위로 저장하며, MP4 형식의 압축으로 인코딩하였을 때 용량이 약 25Mbyte이다. 본 과제를 위해 설치한 카메라의 수가 10대인 점을 감안할 때 1일 기준 총 5.9Gbyte 정도의 스토리지 용량을 필요로 하게 된다. 4Tbyte 하드디스크 용량을 감안할 때 3개월 그 이상의 기간동안 데이터 저장이 가능하단 결론을 내렸다. 리사이징과 인코딩을 진행한 영상을 실제 녹화, 재생해본 결과 모바일 화면에서 식별력이 충분하며 하드디스크의 금액을 고려

하였을 때 경제적으로 경쟁력이 있었다.

농가서버와 사용자 모바일 어플리케이션이 직접 통신을 위해 농가서버 고유의 식별자가 필요하다. 일반적으로 인터넷 회선의 경우 유동아이피를 사용하기 때문에 항상 같은 식별자로 구별된다는 보장을 할 수 없다. 이를 위해 농가서버는 본 서버의 정보를 중앙서버로 보고하며, 중앙서버는 유지된 농가서버의 정보를 사용자 모바일에 전달하며 직접 통신이 가능하도록 한다. 이를 위해 Schedule Job 및 사용자 요청에 응답하기 위한 웹서버가 농가서버에 구동된다.

타입	요청주소	파라미터	설명
GET	live/:device_id	-	지정된 디바이스 아이디에 해당하는 스트리밍 영상 전송
GET	video/:device_id	*dateStr: 일시정보	해당하는 디바이스 아이디의 지정된 영상을 전송

다. 중앙서버 인프라 구축

모바일 사용자의 회원가입 및 텃밭 정보의 등록과 분양 등의 비즈니스 로직을 위해 웹 어플리케이션 서버 개발이 필요하다. 모바일 어플리케이션의 모든 기능을 위해 사용되며, 아마존 웹서비스를 기반으로 구축하였다. 대표적인 클라우드 서비스인 아마존 웹 서비스를 사용한 이유는 초기 인프라 구축비용이 저렴하며, 사용자 트래픽에 따라 서버 인프라의 성능을 조절하여 유지비용을 효과적으로 관리할 수 있기 때문이다.

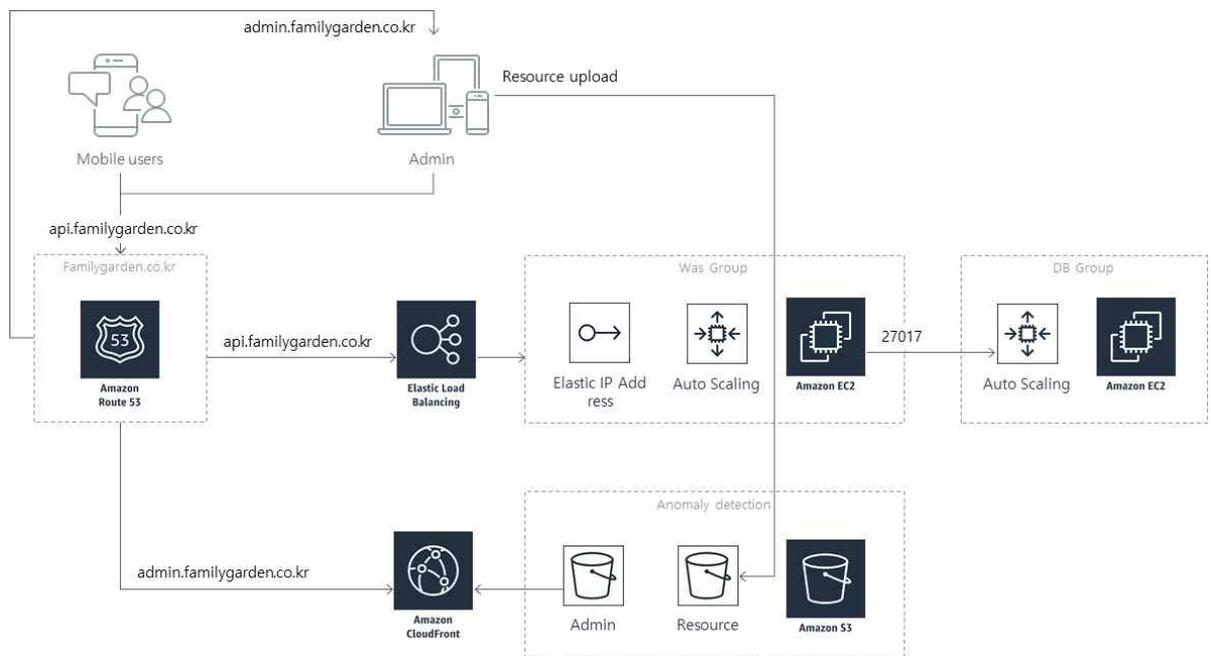


그림 16 중앙서버 인프라 모식도

아마존 웹 서비스를 기반으로 구축한 시스템 인프라는 위 그림 13의 모식도와 같으며, 클라우드 서비스의 각 요소에 대한 설명을 아래에 상세히 한다.

· Route 53

Route 53은 DNS (Domain Name Server) 역할을 수행한다. 본 과제의 도메인은 “familygarden.co.kr”이며, 주요 서브 도메인은 서비스 소개 웹, 관리자용 웹, Rest Api 요청 End-Point 등으로 구성되어 있다.

www.familygarden.co.kr - “우리집 텃밭” 모바일 어플리케이션에 대한 랜딩페이지로 간략한 소개와 다운로드 할 수 있는 링크를 제공하는 홍보용 웹 페이지이다. 본 페이지는 회원가입 및 로그인 등 서버와 통신하는 기능은 존재하지 않으며, 단순 사용자에게 서비스의 간략한 소개와 다운로드 유입을 위해 사용되어 진다. 서버 기능을 사용하지 않음으로 AWS S3의 Static Web Hosting을 사용하여 간단하게 서비스된다.

admin.familygarden.co.kr - “우리집 텃밭”의 사용자 관리, 분양관리, 텃밭 관리, 문의 관리 등 서비스의 전반적인 관리기능을 위한 관리자 웹 인터페이스이다. 관리자의 아이디로 로그인이 가능하며 서비스의 운영을 위한 다양한 관리기능을 제공한다.

api.familygarden.co.kr - 모바일 및 관리자 웹에서 비즈니스 로직에 맞춘 기능을 수행하기 위해 웹 어플리케이션 서버(WAS: Web Application Server)로 요청을 보내게 된다. 해당 WAS를 가리키고 있어 80번 포트를 통해 HTTP 통신을 하게 된다.

· Elastic Load Balancer

과도한 트래픽에 대비하여 다수의 WAS 서버를 배치할 수 있으며, 이를 위해 요청을 일정 조건에 따라 WAS 서버로 분배하는 역할을 한다. 물리적인 장비로 설비하는 것이 일반적이지만 비용이 과다하고 회수할 수 없는 비용임으로 논리적 구성인 Elastic Load Balancer를 사용한다. 해당 AWS의 로드밸런서는 Was Group의 각 노드에 대해 Health Check를 수행하며 이를 통해 노드를 관리한다. 관리된 노드를 대상으로 지정한 알고리즘에 기반하여 사용자로부터의 요청을 노드 그룹의 각 노드에게 배분한다. 이를 통해 다수개의 Was를 운영할 수 있으며 과도한 트래픽으로부터 안정적인 서비스가 가능해 진다.

· Amazon EC2 - Was

컴퓨팅 리소스를 확보하여 임의의 서비스를 세팅 제공가능하게 하는 서비스로 컴퓨팅 성능을 다양하게 책정할 수 있어 비용 상 경제적이다. 개발 당시 t2.micro (cpu 1, ram 2gb) 로 설정하며 유지비용을 낮출 수 있었으며, 서비스 제공 시 간단한 조작으로 컴퓨팅 파워를 높일 수 있다. Ubuntu 16.04 의 리눅스 기반의 운영체제를 사용하였으며, 로드밸런서를 거쳐 3000번 포트를 통해 요청이 수신된다. Was를 데이터베이스를 직접 조작하며 다양한 비즈니스 로직을 수행하게 된다. 모바일 어플리케이션 및 관리자 웹으로

부터의 다양한 요청을 처리하는데, 요청의 형태는 Http 요청으로 설계는 RestFul 형태로 하였다. Was 서버의 상세 설정 정보는 아래와 같다.

- 운영체제: Ubuntu 1.04 LTS
- 실행환경: Node.js v10.x
- 서버 프레임워크: Express.js v4.x
- 요청 포트: 3000

Was 서버는 노드 기반으로 개발되었으며, 웹서버 프레임워크로는 익스프레스를 활용하였으며, 설계한 상세 프로토콜을 기준으로 사용자 요청에 응답한다. 프로토콜은 Rest Api 형태로 설계되었으며, 각 상세 요청은 아래와 같다.

타입	요청주소	파라미터	설명
POST	sign/up	*id: 아이디 *pass: 패스워드 *name: 사용자명 *phone: 사용자핸드폰	회원가입
POST	sign/in	*id: 사용자 아이디 *pass: 사용자 패스워드	로그인
DELETE	sign	-	로그아웃
POST	users	*_id: 아이디 *pass: 패스워드 *name: 사용자명 *phone: 사용자핸드폰 *level: 사용자 레벨	회원 추가 (관리자기능)
GET	users	*offset: 조회 시작위치 *count: 조회 개수	회원 리스트 조회 (관리자 기능)
GET	users/me	-	사용자 정보조회
PUT	users/pass	*oldPass: 기존 비밀번호 *newPass: 신규 비밀번호	사용자 비밀번호 수정
PUT	users/phone	*phone: 핸드폰번호	사용자 핸드폰번호 수정
GET	users/:user_id	-	사용자 정보 조회 (관리자 기능)
PUT	users/:user_id	*name: 사용자 이름 *phone: 사용자 핸드폰번호 *level: 사용자 레벨	사용자 정보수정 (관리자 기능)

DELETE	users/:user_id	-	사용자 삭제 (관리자 기능)
GET	ping	-	Health Check (로드밸런서 사용)
POST	farms	<ul style="list-style-type: none"> *owner_id: 농장주 아이디 *owner_name: 농장주 명 *title: 텃밭 명 *cover: 커버이미지 *images: 텃밭 이미지들 *address: 텃밭 주소 *lat: 위도 *lng: 경도 *area: 분양면적 *crops: 재배작물 *price: 단위면적 분양금액 *bank_name: 예금주 *bank_account: 계좌번호 *note: 유의사항 	농장추가 (관리자 기능)
GET	farms	<ul style="list-style-type: none"> *offset: 조회위치 *count: 조회개수 *lat: 위도 *lng: 경도 *radius: 반경 (미터) *keyword: 검색어 	농장 검색
GET	farms/:farm_id	-	농장 상세 정보 조회
PUT	farms/:farm_id	<ul style="list-style-type: none"> *owner_id: 농장주 아이디 *owner_name: 농장주 명 *title: 텃밭 명 *cover: 커버이미지 *images: 텃밭 이미지들 *address: 텃밭 주소 *lat: 위도 *lng: 경도 *area: 분양면적 *crops: 재배작물 *price: 단위면적 분양금액 *bank_name: 예금주 *bank_account: 계좌번호 *note: 유의사항 	농장 정보 수정 (관리자 기능)

PUT	farms/:farm_id/report	*cameras: 카메라 정보	카메라 정보 업데이트
POST	rents	*farm_id: 농장 고유키 *area: 분양 면적	농장 분양
GET	rents	*offset: 분양정보 조회 위치 *count: 분양정보 조회 개수 +farm_id: 텃밭 고유키 +user_id: 사용자 아이디	농장 분양 조회 (관리자 기능)
GET	rents/me	*type: 분양 타입	분양정보 조회
GET	rents/:rent_id	-	분양정보 상세조회
PUT	rents/:rent_id	*area: 분양면적 *start: 분양 시작일 *end: 분양 종료일 *state: 분양 상태	분양정보 수정 (관리자 기능)
DELETE	rents/:rent_id	-	분양정보 삭제
GET	inquiries	*offset: 문의 조회 시작 위치 *count: 문의 조회 개수 +field: 검색 필드 +keyword: 검색어	1:1 문의 조회
POST	inquiries	*title: 문의 제목 *contents: 문의 내용 +files: 첨부파일	1:1 문의 등록
GET	inquiries/me	-	나의 1:1 문의 내역 조회
GET	inquiries/:id	-	문의 상세 내역 조회
PUT	inquiries/:id	*title: 문의 제목 *contents: 문의 내용 +files: 첨부파일	문의내역 수정
DELETE	inquiries/:id	-	문의내역 삭제
POST	inquiries/:id/answer	*contents	문의 답변 등록

- **AWS S3 - 스토리지 및 Web Hosting**

관리자 및 사용자가 업로드하는 다양한 리소스의 저장을 위해 AWS S3를 사용하였다. 스토리지를 독립적으로 사용하여 다수개의 Was 서버를 사용함에 따라 발생하는 공용사용 문제를 해결하였으며, 안정적인 리소스와 트래픽을 분리하여 보다 효과적인 서버 운영 관리가 가능하다. 또한 관리자 웹 페이지의 경우 SPA (Single Page Application) 형태로 개발되어 별도의 서버 렌더링이 필요하지 않다. 이러한 환경에서 S3에서 제공하는 Static Web Hosting을 활용하여 관리자 웹에 대한 요청 트래픽 역시 분산하였다.

라. 사용자 및 텃밭 관리를 위한 관리자 웹 시스템

모바일 사용자는 회원가입을 통해 서비스를 사용할 수 있으며, 텃밭의 정보검색 및 분양의 신청이 가능하다. 또한 관리자에게 1:1 문의 등 소통을 위한 기능의 사용도 가능하다. 이러한 정보의 효과적인 관리와 비즈니스 로직의 처리를 위해 관리자는 웹 시스템을 통해 아래와 같은 기능을 조작하여 서비스를 운영한다.

- **공지사항 관리**

관리자는 시스템 점검 및 사용자에게 공지할 내용을 등록/수정/삭제 할 수 있다.

- **사용자 관리**

사용자의 추가/수정/삭제 가 가능하며, 사용자의 정보를 조회할 수 있다.

- **문의 관리**

매니저 및 최고관리자는 사용자가 등록한 문의에 대해 답변할 수 있다.

- **텃밭 관리**

텃밭의 등록/수정/삭제/조회가 가능하다.

- **주문 관리**

텃밭의 분양 요청에 대해 상태를 확인하고 입금 여부에 따라 분양 면적, 시작일, 종료일 등을 지정하여 분양 상태를 조정할 수 있다.

- **로그**

시스템의 전반적인 로그를 조회할 수 있다.

관리자 시스템의 구현결과물은 아래와 같이 웹 인터페이스로 이루어져 있다. 계정의 종류는 매니저와 최고관리자로 구분되어 지며, 농장주는 매니저로써 자신의 텃밭을 분양한 사용자의 정보조회 및 문의에 대응할 수 있다. 웹 시스템에 접근하기 위해서는 로그인 필요하다.



LOGIN TO YOUR ACCOUNT

Email

Password

Remember Me [Forgot Password?](#)

LOGIN

OR

SEED-ADMIN

농장
레시피 수정

기본정보 상세주소 확인 사진 업로드

농장 주소*
경기도 수원시 팔달구 영등포 92 남문타워

농장주 아이디*
aram@test.com

농장주 이름*
테스터

농장주 거래은행*
국민

농장주 계좌번호*
123123

농장명(이름)*
아람 텃밭

SEED-ADMIN

Rents
Rent 수정

요청정보

상태*
요청

신청일
2018-12-07

종인시 계약 시작일(당일기준)
2018-12-28

종인시 계약 종료일(당일기준)
2019-12-28

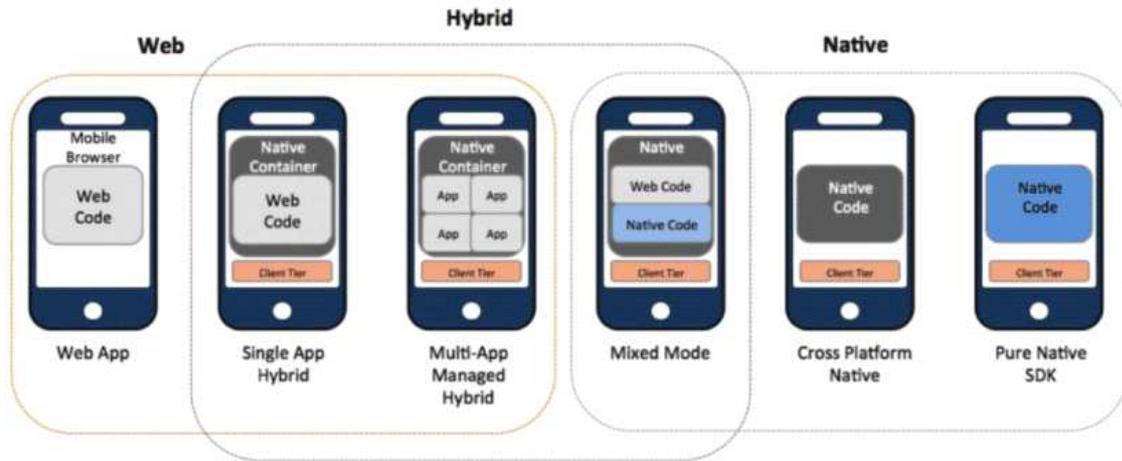
요청면적
111

금액
3330000

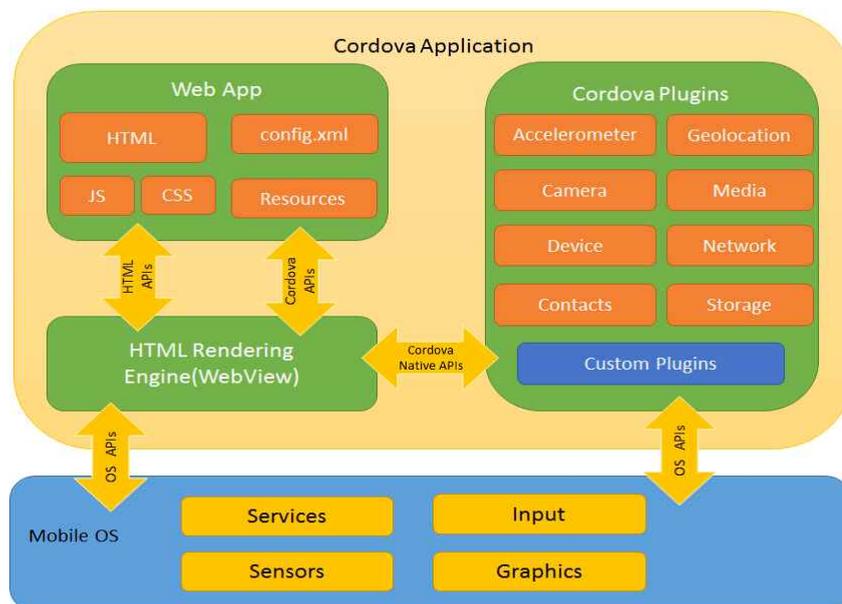
농장명
아람 텃밭

마. 텃밭 분양 및 검색을 위한 사용자 모바일 어플리케이션

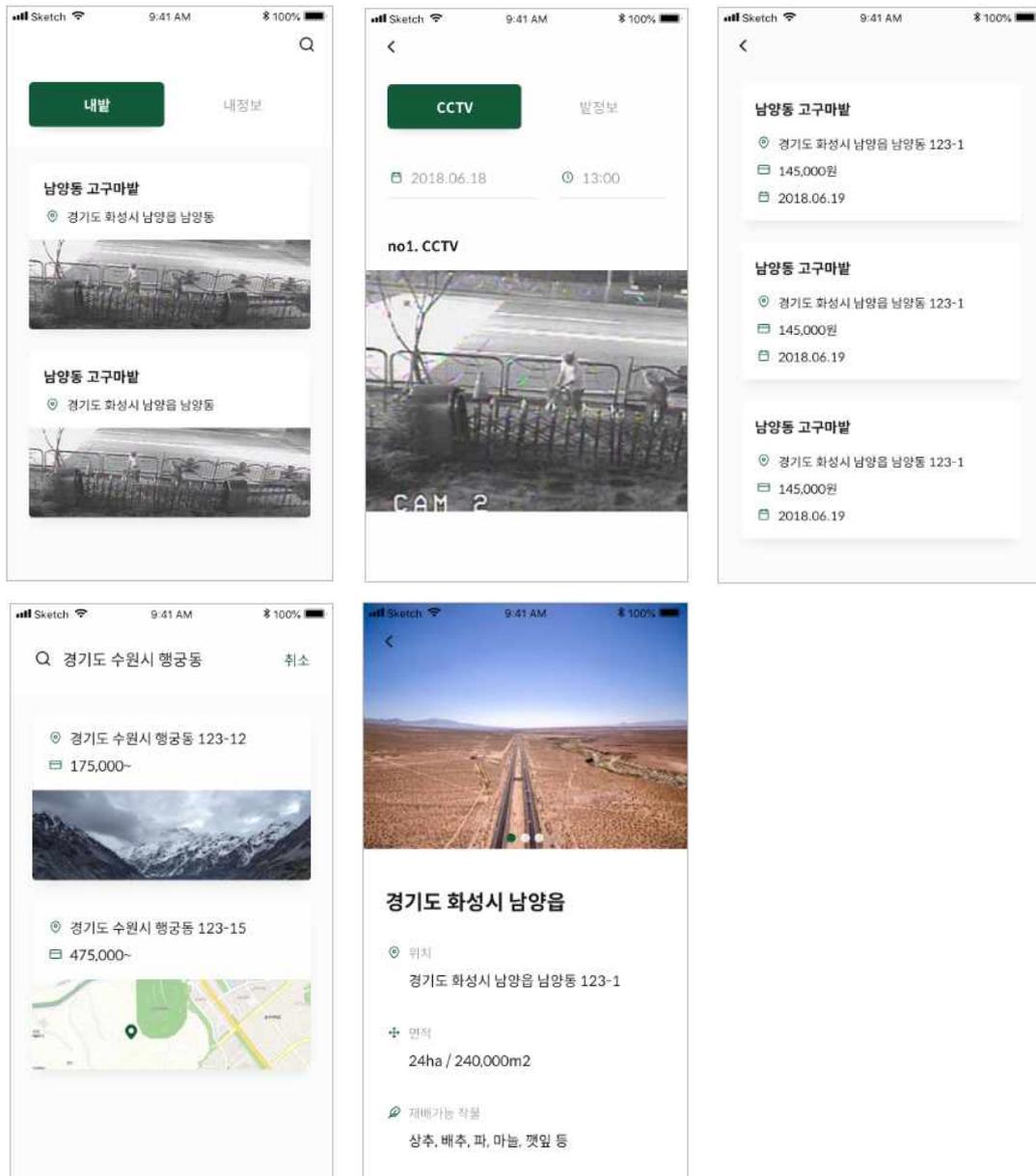
본 모바일 어플리케이션은 하이브리드 앱 형태로 개발되었다. 일반적인 Native App은 Android와 IOS에 대응하기 위해 각 개발 라이브러리에 맞추어 독립적으로 개발해야 하지만, 하이브리드앱의 경우 사용자 UI/UX의 경우 공통으로 사용하며 기기의 특성을 사용하는 네이티브 기능에 대해서만 각 개발 라이브러리에 맞추어 별도 개발한다. 이러한 개발 방식은 여러 모바일 OS에 대응한 서비스를 만들기에 적합하며 시간적인 이득을 취할 수 있다.



각 모바일 OS에 맞춘 네이티브 기능을 구현하기 위해 Cordova 프레임워크를 사용하였다. 하이브리드 앱의 특성상 WebView 위에서 구동하기 때문에 네이티브 기능을 사용하기에 다소 어려움이 있다. 이를 효과적으로 극복하기 위해 네이티브 기능을 사용한 로직 부분만을 네이티브 코드로 작성하는 방식의 Cordova 프레임워크를 차용하였으며, 제공하는 다양한 Plugin을 활용하여 효과적으로 개발하였다.



Cordova를 활용한 하이브리드 앱 형태로 개발된 본 사용자 모바일 어플리케이션은 텃밭 정보를 검색하고 이를 분양할 수 있다. 사용자는 자신의 위치를 기반으로 Map 인터페이스 기반에서 텃밭 정보를 검색하며 재배작물과 텃밭의 사진을 조회한 후 분양 주문정보를 요청할 수 있다. 등록되어진 밭의 분양평수와 분양기간을 조회하여 분양이 가능한 경우 주문정보를 요청한다. 아래의 그림은 실제 개발된 모바일 어플리케이션의 화면이다.



2. 경제적 성과

본 과제를 통해 개발한 플랫폼은 아직 시험단계로 일부사용자를 대상으로 클로즈베타를 진행하고 있다. 200평 가량의 텃밭을 대상으로 인프라를 구축하고 커뮤니티 사용자를 대상으로 유료 분양을 수행하였다. 개발 시기적으로 가을 및 겨울에 진입하여 많은 베타 사용자를 유치할 순 없었지만 실제 비용을 지급한 유의미한 매출 실적이 있다. 본 과제가 종료된 후 2019년 봄에 본격적인 마케팅을 계획하고 있으며, 200평에 해당하는 모든 텃밭을 분양할 계획이다. 또한 개발을 진행하는 동안 협업기관인 (주)아람인에서 1명의 정규직 고용이 있었으며, 시스템을 지속 유지보수하며 완성도를 높일 수 있는 전문 개발인력을 배치하였다.

본 시스템을 통해 그 동안 주말텃밭의 단점을 보완하며 단순 활동에 그치는 것이 아닌 경제적이고 건강한 먹거리의 제공이란 측면에서 타 서비스와 차별성을 갖는다. 그만큼 베타서비스를 끝낸 후 주말농장 시장과 먹거리 시장에 큰 부분을 점유할 수 있을 것으로 생각된다.

제 3 장 목표 달성도 및 관련 분야 기여도

제 1절 연구개발 목표

주요 성능지표 ¹⁾	단 위	최종 개발목표 ²⁾	세계최고수준 (보유국/보유기업)	가중치 ³⁾ (%)	객관적 측정방법	
					시료 수 (n≥5개)	시험규격 ⁴⁾
1. 카메라 영상 출력		안드로이드 4.0 이상 폰에서 영상 출력	안드로이드 4.0 이상에서 가능 (한국/브이스타캡)	20	5	안드로이드 4.0, 아상의 폰에서 카메라 영상
2. 영상 저장 정보 출력		안드로이드 4.0 이상의 폰에서 PC 또는 NVR 저장 영상 정보 출력	안드로이드 4.0, 아상의 폰에서 PC와 NVR의 저장된 영상 정보를 생산자/소비자 스마트폰에서 출력 (한국/브이스타캡)	20	5	안드로이드 4.0, 아상의 폰에서 PC와 NVR에 저장된 영상 정보를 출력 확인
3. 텃밭 정보 등록	개	10만개 이상	10만개 이상 (한국/)	20	5	홈 페이지 서버에서 10만개 이상 일일 등록 가능 여부 확인
4. 생산자/소비자 등록/해제	개	10만개 이상	10만개 이상 (한국/)	20	5	생산자/소비자 스마트폰을 통해 등록된 홈 페이지 서버에서 10만개 이상 일일 등록 가능 여부 확인
5. 주문 처리	개	10만개 이상	10만개 이상/일 (한국/)	20	5	사용자의 일일 주문 요청 10만개 이상 가능여부 확인
<input type="checkbox"/> 시료수 5개 미만 (n<5개)시 사유						
<input type="checkbox"/> 해당사항 없음						
<input type="checkbox"/> 측정결과의 증빙방법 제시						
<input type="checkbox"/> 한국 정보통신 기술협회 (TTA) 산하 소프트웨어 시험인증 연구소(바우처)의 공인 시험성적서로 증빙						

제 2 절 연구개발 목표 달성여부

1. 목표달성 확인 방법

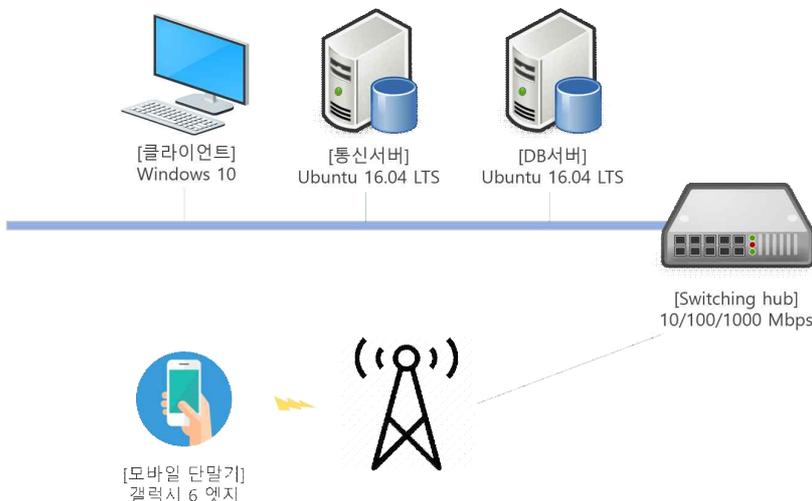
가. 시험 신뢰성 확보

시험 신뢰성을 확보하기 위해 공인된 시험인증기관에 시험 의뢰한다. 한국정보통신기술협회에서 공인된 V&V 시험을 통해 신뢰성을 확보한다. 시험기간은 총 5일로 과제 기간 중 시험의뢰를 신청하였지만 연말의 일정이 밀려 1월 초에 시험이 예정되어 있다. 시험에 관련된 대략적인 내용은 아래와 같다.

· 시험대상 운영환경

소프트웨어 정보		
운영체제	서버	Ubuntu 16.04 LTS * 2대
	클라이언트	Android 7.0 * 1대
특이사항 (제품구동요구사항)	아마존 웹 서비스 기반	
하드웨어 정보		
하드웨어 사양	서버	AWS EC2 (t2.medium, t3.large)
	클라이언트	CPU: Samsung Exynos 5 Octa, 메모리: 4GB, HDD: 32G
네트워크 환경	WiFi 무선 네트워크 환경	
기타 환경	<ul style="list-style-type: none"> - 클라우드 서비스 기반, 통신서버, 데이터베이스 서버 별도 존재 - 텃밭에 IP Camera 설치 - IP Camera의 영상을 서버로 전송 - 클라이언트에서 일자별로 해당 영상 조회 가능 	
기타사항		
<ul style="list-style-type: none"> - 한국정보통신기술협회에서는 제품 시험을 위한 시험 장소를 제공함 - 강남대학교 산학협력단에서는 시험에 필요한 HW 및 SW를 제공함 · HW : 스마트폰 1대 · SW : 우리집 텃밭 v1.0 - 강남대학교 산학협력단에서는 시험환경구성, 제품설치 및 기술교육을 지원함 		

· 시험환경 구성도



2. 목표달성 확인 세부 절차

가. 카메라 실시간 영상 출력 확인

· 시험목표

시험대상제품에서 ‘내 발’ 메뉴 클릭 시, 사용자의 텃밭이 실시간으로 영상을 통해 출력되는지 확인

* 실시간 영상은 5분 재생함.

· 목표치

정상동작 여부의 확인

· 시험 세부 절차

ID	구분	내용
TC1	시험 항목명	실시간 영상 출력 확인
	시험 목표	- 시험대상제품에서 ‘내 발’메뉴 클릭 시 사용자의 텃밭을 실시간으로 조회 가능한지 확인
	사전 조건	- IP Camera 설치 및 구동 - 농가 Streaming 서버 동작 - 클라이언트에 본 시험 프로그램(모바일 앱) 설치 - 사용자 가입 및 텃밭 분양 데이터 사전 생성
	시료	클라이언트 1대, IP Camera 4대
	반복시험 횟수	20회 반복 측정
	시험 절차	- 클라이언트에서 로그인 수행 (아이디/패스워드) · 아이디: test@test.com · 패스워드: 1q2w3e - 클라이언트에서 분양받은 텃밭 조회 · 하단 탭 메뉴에서 “내 발” 메뉴를 클릭 · “반정동 텃밭”을 클릭하여 상세정보 조회 - 현재 텃밭의 실시간 영상이 조회 되는지 확인 · 상세 페이지에서 CCTV 탭 클릭 · 각 CCTV에서 현재 영상이 정상 출력되는지 확인
	예상결과	5분 이상 실시간 영상 조회 성공
	예외사항	클라이언트 및 서버의 네트워크 연결 상태가 불안정할 경우 다시 수행

나. 저장영상 출력 확인

· 시험목표

CCTV로 녹화된 영상을 시험대상제품에서 일자별로 재생하여 볼 수 있는지 확인

* 저장영상은 CCTV로 24시간 녹화되는 영상이며 파일형태로 저장

· 목표치

정상동작 여부의 확인

· 시험 세부 절차

ID	구분	내용
TC2	시험 항목명	저장 영상 출력 확인
	시험 목표	- 클라이언트에서의 카메라 저장 영상이 조회되는지 확인
	사전 조건	- IP Camera 설치 및 구동 - 농가 Streaming 서버 동작 - 클라이언트에 본 시험 프로그램(모바일 앱) 설치 - 사용자 가입 및 텃밭 분양 데이터 사전 생성
	시료	클라이언트 1대, IP Camera 4대, CCTV 영상(시험의뢰기업 제공)
	반복시험 횟수	20회 반복 수행
	시험 절차	- 클라이언트에서 로그인 수행 (아이디/패스워드) · 아이디: test@test.com · 패스워드: 1q2w3e - 클라이언트에서 분양받은 텃밭 조회 · 하단 탭 메뉴에서 “내 밭” 메뉴를 클릭 · “반정동 텃밭”을 클릭하여 상세정보 조회 - 텃밭의 저장된 영상이 조회 되는지 확인 · 상세 페이지에서 CCTV 탭 클릭 · 각 CCTV 클릭 후 조회는 원하는 임의 날짜 (시험 검사일 전 2일) 선택 · 해당 CCTV에서 저장된 영상이 출력되는지 확인
	예상결과	5분이상 녹화 영상 재생 성공
	예외사항	클라이언트 및 서버의 네트워크 연결 상태가 불안정할 경우 다시 수행

다. 텃밭 정보 등록 처리량

· 시험목표

하루에 등록 가능한 텃밭 데이터 정보가 10만개 이상인지 확인

※ 성능측정 도구인 j-meter를 이용하여 시료정보를 등록 하며, 시료데이터 파일은 시험의뢰기업에서 제공

※ DB를 통해 등록된 텃밭 개수 확인

· 목표치

일일 10만개 이상 등록 처리 가능

· 시험 세부 절차

ID	구분	내용
TC3	시험 항목명	텃밭 정보 등록 처리량
	시험 목표	- 일일 기준 10만개 이상의 텃밭 정보 등록 처리 확인
	사전 조건	- 통신서버 및 데이터베이스 서버 구동
	시료	임의 생성된 텃밭 정보 10만개
	반복시험 횟수	2회
	시험 절차	<ul style="list-style-type: none"> - 데이터베이스 발 정보 수 확인 <ul style="list-style-type: none"> · 명령어: <code>db.getCollection('Farms').count({});</code> - 시료데이터 등록 <ul style="list-style-type: none"> · JMeter를 활용하여 시료 데이터 입력 (JMeter 설정 파일 전달) · 요청: [POST] http://api.familygarden.co.kr/farms · 요청 파라미터 <pre>{ owner: {_id: 텃밭주 아이디, name: 텃밭주 이름}, addr: 텃밭 주소, location: {lat: 위도, lng: 경도}, crops: 재배가능 작물, price: 분양비용, note: 기타 공지내용 }</pre> - 10만개 시료데이터 1일 이내 등록 확인 <ul style="list-style-type: none"> · 명령어: <code>db.getCollection('Farms').count({});</code>
	예상결과	1일 이내 10만개 텃밭정보 입력
	예외사항	JMeter 오류 시 재수행

라. 사용자 등록 처리량

· 시험목표

하루에 등록 가능한 사용자 데이터 정보가 10만개 이상인지 확인

※ 성능측정 도구인 j-meter를 이용하여 사용자 정보를 등록하며, 사용자데이터 파일은 시험의뢰기업에서 제공

※ DB를 통해 등록된 사용자 수 확인

· 목표치

일일 10만개 이상 등록 처리 가능

· 시험 세부 절차

ID	구분	내용
TC4	시험 항목명	사용자 등록 처리량
	시험 목표	- 하루에 등록 가능한 사용자 데이터 정보가 10만개 이상인지 확인
	사전 조건	- 통신서버 및 데이터베이스 서버 구동
	시료	임의 생성된 사용자 정보 10만개
	반복시험 횟수	2회
	시험 절차	<ul style="list-style-type: none"> - 데이터베이스 받 정보 수 확인 <ul style="list-style-type: none"> · 명령어: db.getCollection('Users').count({}); - 사용자 데이터 등록 <ul style="list-style-type: none"> · JMeter를 활용하여 시료 데이터 입력 (JMeter 설정 파일 전달) · 요청: [POST] http://api.familygarden.co.kr/sign/up · 요청 파라미터 <pre>{ _id: 사용자 아이디, pass: 사용자 비밀번호, phone: 사용자 핸드폰 번호, name: 사용자 이름 }</pre> - 10만개 시료데이터 1일 이내 등록 확인 <ul style="list-style-type: none"> · 명령어: db.getCollection('Users').count({});
	예상결과	1일 이내 10만개 사용자 정보 입력
예외사항	JMeter 오류 시 재수행	

마. 주문 처리량

· 시험목표

하루에 등록 가능한 사용자 주문 데이터가 10만개 이상인지 확인

※ 성능측정 도구인 j-meter를 이용하여 사용자 정보를 등록하며, 사용자데이터 파일은 시험의뢰기업에서 제공

※ DB를 통해 등록된 주문 데이터 개수 확인

· 목표치

일일 10만개 이상 등록 처리 가능

· 시험 세부 절차

ID	구분	내용
TC5	시험 항목명	주문 처리량
	시험 목표	- 일일 기준 10만개 이상의 주문 처리 확인
	사전 조건	- 통신서버 및 데이터베이스 서버 구동
	시료	임의 생성된 사용자 정보 10만개
	반복시험 횟수	2회
	시험 절차	<ul style="list-style-type: none"> - 데이터베이스 발 정보 수 확인 <ul style="list-style-type: none"> · 명령어: <code>db.getCollection('Rents').count({});</code> - 시료데이터 등록 <ul style="list-style-type: none"> · JMeter를 활용하여 시료 데이터 입력 (JMeter 설정 파일 전달) · 요청: [POST] <code>http://api.familygarden.co.kr/rents</code> · 요청 파라미터 <pre style="margin-left: 20px;">{ _f: 텃밭 고유키, area: 분양 면적, price: 분양 가격, expire: 분양 종료일, type: 결제 종류, note: 요청사항, date: 주문 일시 }</pre> - 10만개 시료데이터 1일 이내 등록 확인 <ul style="list-style-type: none"> · 명령어: <code>db.getCollection('Rents').count({});</code>
	예상결과	1일 이내 10만개 사용자 정보 입력
	예외사항	JMeter 오류 시 재수행

3. 목표달성 확인 결과

가. TTA V&V 시험 결과

ID	시험항목	시험목표	결과
TC1	실시간 영상 출력 확인	시험대상제품에서 '내 발' 메뉴 클릭 시, 사용자의 텃발이 실시간으로 영상을 통해 출력되는지 확인 * 실시간 영상은 5분 재생함	정상동작
TC2	저장 영상 출력 확인	CCTV로 녹화된 영상을 시험대상제품에서 일자별로 재생하여 볼 수 있는지 확인 * 저장된 영상은 CCTV로 24시간 녹화되는 영상이며 파일형태로 저장	정상동작
TC3	텃발 정보 등록 처리량	하루에 등록 가능한 텃발 데이터 정보가 10만개 이상인지 확인 * 성능측정 도구인 j-meter를 이용하여 텃발 정보를 등록하며, 텃발 데이터 파일은 시험의뢰기업에서 제공	평균 100,011건 등록 (19분 19초 소요)
TC4	사용자 등록 처리량	하루에 등록 가능한 사용자 데이터 정보가 10만개 이상인지 확인 * 성능측정 도구인 j-meter를 이용하여 사용자 정보를 등록하며, 사용자 데이터파일은 시험의뢰기업에서 제공 * DB를 통해 등록된 사용자수 확인	평균 100,001건 등록 (1시간 10분 59초 소요)
TC5	주문 처리량	하루에 등록 가능한 주문 데이터 정보가 10만개 이상인지 확인 * 성능측정 도구인 j-meter를 이용하여 주문정보를 등록하며, 주문 데이터 파일은 시험의뢰기업에서 제공 * DB를 통해 등록된 주문 데이터 개수 확인	평균 100,009건 등록 (18분 27초 소요)

* 상세 시험결과는 TTA V&V 시험결과서 첨부.

제 3 절 목표 미달성 시 원인(사유) 및 차후대책

해당사항 없음.

제 4 장 연구결과의 활용 계획 등

제 1 절 연구성과의 활용 분야 및 방안

1. 자체 사업화

본 “IoT 기술을 활용한 도-농 연계 영농 플랫폼”은 신청기업인 융합기술진흥원의 자체 사업화를 전제로 하고 있다. 핵심 기술인 “대용량 NVR 시스템”을 활용한 실시간 농가 관제 시스템을 개발하고 개발된 시스템을 기반으로 모바일 어플리케이션의 형태로 서비스할 예정이다. 이를 통해 주말농장과 같은 주말 텃밭과 더불어 안전하고 경제적인 먹거리를 제공받길 원하는 도시민과 유희 농지와 노동력을 활용하여 농가에겐 수익을 가져다 줄 수 있다. 현재 클로즈베타를 진행 중으로 2017년 이미 유료분양 등 유의미한 사업적 성과를 거두었다. 테스트 사용자의 의견을 적극적으로 수용하여 시스템의 안정화와 개선 후 2019년 정식 서비스할 예정이다.

2. 대용량 NVR 시스템의 응용

개발된 “대용량 NVR 시스템”은 농가의 관제 이외에 축사나 다량의 IP CAMERA를 필요로 하는 분야에 기존의 NVR 대신 비용대비 효과적인 적용이 가능하다. 기존의 NVR이나 DVR은 카메라를 확장할 때마다 장비를 구입해야하며, 그에 따라 비용이 지속적으로 수반된다. 본 과제를 통해 개발된 NVR은 일반 PC와 PoE 스위치를 활용하여 기존 사용화 제품보다 경제적으로 시스템의 구축이 가능하다. PoE로 전원을 공급하기 때문에 서버와 IP CAMERA의 거리가 먼 경우보다 밀집된 곳에 다수의 카메라가 설치되는 환경에 적합하다.

3. 연구 분야의 범위 확장

본 과제는 “영농 플랫폼”이란 목표를 갖고 진행하였다. 핵심기술인 “대용량 NVR 시스템”을 활용하여 관제 방식을 개발하고 이를 통해 농가와 도시민의 Win-Win 플랫폼을 설계하였다. 이와 유사한 방식으로 축산 및 수산의 영역으로 해당 플랫폼의 사업 영역을 확장할 수 있을 것으로 생각된다.

제 2 절 추가 연구의 필요성

1. 관제시스템의 전원 공급 문제 개선

현재 개발된 관제시스템은 IP CAMERA의 전원공급을 위해 PoE를 사용하며, 이를 통해 설치비용이 저렴하다. 하지만 거리적 제약이 있기 때문에 넓은 범위의 설치가 불가능하다. 현재 시범텃밭의 경우 큰 문제없이 설치가 진행되었지만 그 보다 넓은 영역일 경우 전원공급

의 문제가 대두된다. 이를 극복하기 위한 증폭기의 설치를 검토하였지만 본 과제에서는 진행되지 않았다. 추후 넓은 범위의 IP CAMERA 설치 진행을 위한 문제 개선 연구를 진행하면 보다 확장성이 좋아질 것으로 생각된다.

2. 사용 편의성 개선

개발된 모바일 어플리케이션은 아직 많은 사용자의 사용편의성 검사를 거치지 못하였다. 일부 사용자에게 제공되었으며 그들의 피드백을 받아 일부 사용 편의성을 개선하였지만 아직 개선해야할 부분이 많이 남아 있다. 추후 지속적인 피드백을 받아 모바일 어플리케이션의 사용편의성을 개선해야할 것이다.

3. 비용 개선을 위한 시스템 최적화

본 과제를 통해 만들어지는 서비스의 비즈니스 모델은 텃밭의 분양에 따른 분양 수익이 가장 큰 부분이다. 고정된 부동산에 따른 수익모델로 비용이 수익에 민감한 역할을 한다. 농가의 유희농지를 활용한 만큼 대부분의 투자 및 유지비용은 시스템의 도입 비용과 유지비용이다. 최초 과제로 다소 높게 비용이 책정되었지만 텃밭 면적에 따라 설치되는 카메라의 개수가 상의할 것이며, 이에 따라 농가 서버의 성능과 비용을 적절히 조절해야 할 것이다.

녹화된 영상의 저장을 중앙 서버에 진행하려 하였지만 전체 시스템의 유지비용이 크게 늘어갈 것으로 생각되어 농가서버를 통해 각 사용자의 요청에 따라 저장된 영상을 제공하기로 하였다. 과제 중에도 비용을 줄이기 위한 노력을 하였지만 추후 실제 서비스를 운영하면서 비효율적인 부분을 지속 개선하여 수익률을 높일 수 있도록 해야 하겠다.

제 4 절 기업화 추진 방안

1. 클로즈베타 진행 및 정식 오픈

2017년 커뮤니티 사용자를 대상으로 클로즈베타 서비스를 진행하였다. 1년 25만원의 비용으로 텃밭을 분양하였으며, 유료 결제를 하는 유의미한 성과를 내었다. 클로즈베타가 완료된 후 시스템 보완 및 과제의 지속수행 후 2019년 정식 서비스를 오픈할 예정이다. 현재 신청기업인 융합기술진흥원이 사업의 주체로 이후 지속적으로 텃밭의 개수와 사용자 증가를 위한 마케팅을 진행할 예정이다.

2. 수직적 시장 확장

소비자에게 안전하고 신뢰성있는 먹거리를 제공하는 것이 본 플랫폼의 가장 큰 목적이다. 사용자는 먹거리를 배송받을 뿐만 아니라 주말농장으로 직접 재배활동에 참여할 수 있다. 사업적 수익창출을 위해 바비큐 및 숙박, 체험학습 등 사용자가 다양하게 이용할 수 있는 방식

을 개발하여 채소 및 재배작물과 더불어 소비자가 느끼는 서비스 만족도를 높일 수 있는 수직적 시장을 확장해 나갈 것이다.

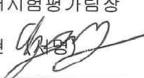
3. 수평적 시장 확장

채소나 텃밭작물에 대해서만 본 플랫폼을 적용하기 보다는 보다 넓은 영역에 적용 가능할 것으로 생각된다. 우리가 접하는 단순 소분 식품이나 가공품, 또는 축/수산물까지 그 범위를 확대할 수 있을 것으로 생각된다. 최근 축산물 혹은 수산물 등을 개인이 직접 키우거나 수확하여 인터넷을 통해 판매하는 행위를 종종 볼 수 있다. 소비자에게 안전하며 신뢰성있는 먹거리를 제공한다는 측면에서 본 플랫폼이 기여할 수 있는 부분이 많을 것으로 생각된다. 1차적으로 채소 및 재배작물에 대해 본 플랫폼을 적용하여 안정화를 이룬 후에 다른 분야에 플랫폼을 적용해 보는 방향으로 사업의 수평적 확장을 도모할 계획이다.

붙임1. 참고문헌

- Network Interface Specification, Onvif, <https://www.onvif.org/profiles/specifications>
- Web Application Hosting, AWS, <https://aws.amazon.com/ko/architecture>
- 안전한 귀갓길을 위한 모바일 어플리케이션, 이계천, 한국정보과학회, 한국정보과학회 학술발표논문집 , 2017, 357-359
- 정보통신 및 위치기반 기술을 활용한 O2O 판매 플랫폼, 이계천, 한국정보과학회, 한국정보과학회 학술발표논문집 , 2015, 401-403

붙임2. TTA V&V 시험 결과서

시험 결과서		
한국정보통신기술협회 소프트웨어시험인증연구소 주소: 경기도 성남시 분당구 분당로 47 전화: 031-780-9250, Fax: 031-724-0189	결과서번호: BT-A-18-0705	
<p>1. 의뢰자</p> <ul style="list-style-type: none"> · 회사(기관)명 : 강남대학교 산학협력단 · 주 소 : 경기도 용인시 기흥구 강남서로 20, 2층 202호(구갈동, 강남대학교 창조산학관) · 계약일자 : 2018년 11월 9일 <p>2. 제품명 : 우리집 텃밭 v1.0</p> <p>3. 시험기간 : 2019. 1. 2. ~ 2019. 1. 3. (1차) 2019. 1. 23. ~ 2019. 1. 25. (2차)</p>		
확인	작성자 성명 : 양희은 	승인자 직위 : 소프트웨어시험평가팀장 성명 : 유재현 
2019. 1. 25. 한국정보통신기술협회 회장 		

TTA-SE1-19-A(R)

결과서번호 : BT-A-18-0705

우리집 텃밭 v1.0



2019년 1월 25일

한국정보통신기술협회
Telecommunications Technology Association

본 문서는 한국정보통신기술협회(TTA) 소프트웨어시험인증연구소의 시험결과서로 한국정보통신기술협회(TTA) 승인 없이 문서의 전부 또는 일부를 탈취하여 배포, 복제 및 이용할 수 없습니다.

TTA

목 차

1. 개 요	4
1.1 회사개요	4
1.2 수행과제개요	4
2. 시험목적 및 절차	5
2.1 시험목적	5
2.2 시험절차	5
2.3 수행내용 및 시험기간	5
3. 시험항목 및 방법	6
3.1 시험항목	6
3.2 시험방법	7
4. 시험환경	9
5. 시험결과	10
<첨부1> 상세 시험결과	11

1. 개요

본 시험결과서는 한국정보통신기술협회(이하 “TTA”)가 강남대학교 산학협력단(이하 “시험의뢰기업”)의 ‘우리집 텃밭 v1.0’(이하 “시험대상제품”)을 대상으로 시험한 결과를 기술한 문서이다. 본 문서에 명시된 시험결과는 ‘3. 시험항목 및 방법’과 ‘4. 시험환경’에서 확인한 결과에 국한된다.

1.1 회사개요

회사(기관)명	강남대학교 산학협력단	대표자	서경환
설립일	2004년 4월 7일	사업자등록번호	135-82-10867
전화번호	031-283-6170	FAX번호	031-243-1646
대표자 E-mail	kicf@kangnam.ac.kr	홈페이지	www.kicf.ac.kr
주소	경기도 용인시 기흥구 강남서로 20, 2층 202호(구갈동, 강남대학교 창조산학관)		

1.2 수행과제개요

과제전담기관	농림식품기술기획평가원
수행과제명	IoT 기술을 활용한 도-농 연계 영농 플랫폼 개발

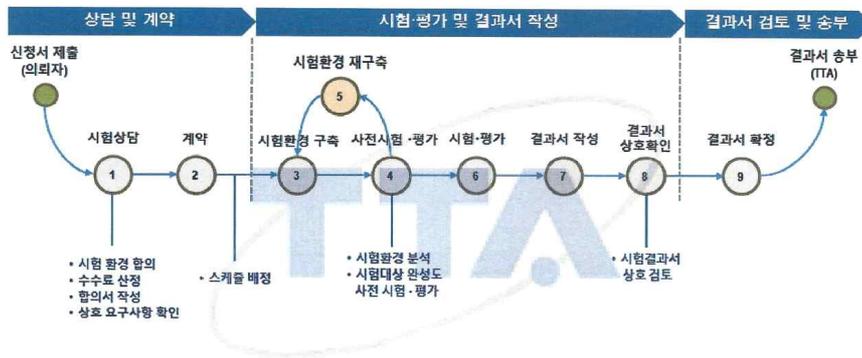
2. 시험목적 및 절차

2.1 시험목적

본 시험의 목적은 시험의뢰기업이 요청한 시험항목에 대해 제3자 전문시험기관인 TTA가 제품의 기능 및 성능을 객관적으로 검증하는데 있다.

2.2 시험절차

시험절차는 다음과 같다.



2.3 수행내용 및 시험기간

수행내용 및 시험기간은 다음과 같다.

시험절차	수행내용	시험기간
시험·평가 및 결과서 작성	· 시험환경 구축 · 제품 분석 및 사전 시험·평가	5일
	· 시험항목에 대한 시험 및 평가 수행	
	· 시험결과서 작성 · 시험결과서 상호 확인	

3. 시험항목 및 방법

3.1 시험항목

시험의뢰기업에서 요청한 시험 항목은 다음과 같다.

ID	시험항목	시험목표
TC1	실시간 영상 출력 확인	시험대상제품에서 '내 발' 메뉴 클릭 시, 사용자의 텃발이 실시간으로 영상을 통해 출력되는지 확인 ※ 실시간 영상은 5분 재생함
TC2	저장 영상 출력 확인	CCTV로 녹화된 영상을 시험대상제품에서 일자별로 재생하여 볼 수 있는지 확인 ※ 저장된 영상은 CCTV로 24시간 녹화되는 영상이며 파일형태로 저장
TC3	텃발 정보 등록 처리량	하루에 등록 가능한 텃발 데이터 정보가 10만개 이상인지 확인 ※ 성능측정 도구인 j-meter를 이용하여 텃발 정보를 등록하며, 텃발 데이터 파일은 시험의뢰기업에서 제공 ※ DB를 통해 등록된 텃발 개수 확인
TC4	사용자 등록 처리량	하루에 등록 가능한 사용자 데이터 정보가 10만개 이상인지 확인 ※ 성능측정 도구인 j-meter를 이용하여 사용자 정보를 등록하며, 사용자 데이터 파일은 시험의뢰기업에서 제공 ※ DB를 통해 등록된 사용자수 확인
TC5	주문 처리량	하루에 등록 가능한 주문 데이터 정보가 10만개 이상인지 확인 ※ 성능측정 도구인 j-meter를 이용하여 주문 정보를 등록하며, 주문 데이터 파일은 시험의뢰기업에서 제공 ※ DB를 통해 등록된 주문 데이터 개수 확인

3.2 시험방법

시험항목별 시험방법은 다음과 같다.

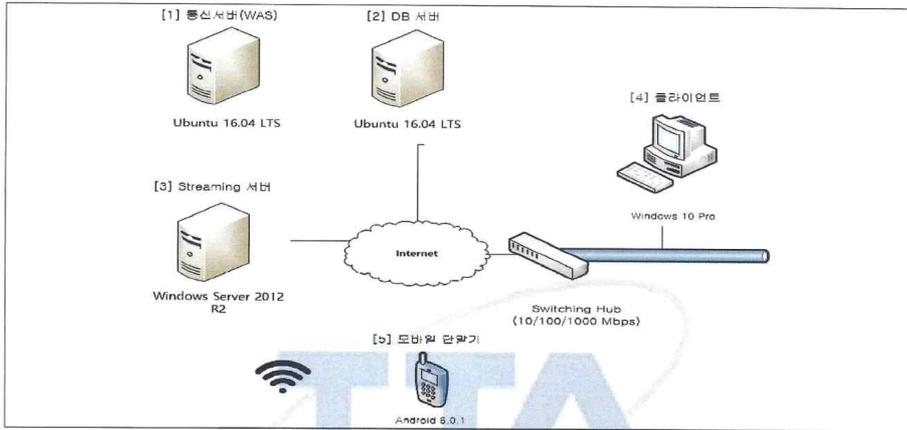
ID	시험방법
TC1	<p><사전 조건></p> <ul style="list-style-type: none"> - 텃밭에 IP Camera가 설치됨 (설치된 장소는 시험의뢰기업에서 제공) - 텃밭 사용자 정보 및 텃밭 정보가 사전에 등록됨 <p><시험 방법></p> <ol style="list-style-type: none"> 1. 시험대상제품(모바일앱)을 실행 2. 하단의 [내밭] 메뉴를 선택하고 로그인 (ID: test@test.com PW: 1q2w3e) 3. 등록된 텃밭을 선택하고 기 등록된 카메라 4대 중 1대를 선택 후, [OK]를 클릭 4. 현재 텃밭의 실시간 영상이 정상적으로 조회되는지 확인 ※ 영상 속 시각은 중국 기준 시간이어서 현재 시간보다 1시간 늦게 조회됨 5. 모든 카메라에 대해 반복 수행하고 모두 정상적으로 조회되는지 확인
TC2	<p><사전 조건></p> <ul style="list-style-type: none"> - TC1의 사전조건과 동일함 <p><시험 방법></p> <ol style="list-style-type: none"> 1. 시험대상제품(모바일앱)을 실행 2. 하단의 [내밭] 메뉴를 선택하고 로그인 (ID: test@test.com PW: 1q2w3e) 3. 등록된 텃밭을 선택하고 카메라를 고른 다음 [OK]를 클릭 4. 임의의 일자를 선택하고 해당 일자에 저장된 영상이 정상적으로 출력되는지 확인 5. 모든 카메라에 대해 반복 수행하고 모두 정상적으로 출력되는지 확인
TC3	<p><사전 조건></p> <ul style="list-style-type: none"> - 통신 서버 및 데이터베이스 서버가 정상 구동됨 - 시료데이터 10만 건이 준비됨 <p><시험 방법></p> <ol style="list-style-type: none"> 1. DB 관리 툴을 통해 등록 전 데이터 건수를 확인 (db.getCollection("Farms").count({});) 2. JMeter를 이용하여 시료데이터 10만건을 등록 요청: [POST] http://api.familygarden.co.kr/farms 요청 파라미터 : <pre> { owner: { _id: 텃밭주 아이디, name: 텃밭주 이름}, addr: 텃밭 주소, location: {lat: 위도, lng: 경도}, crops: 재배가능 작물, price: 분양비용, note: 기타 공지내용 } </pre> 3. 등록이 완료되면 데이터베이스에 등록 된 데이터 건수를 확인 4. 2회 반복 수행하고 10만 건의 데이터가 정상적으로 등록되는지 확인

ID	시험방법
TC4	<p><사전 조건> - TC3의 사전조건과 동일함</p> <p><시험 방법></p> <ol style="list-style-type: none"> 1. DB 관리 툴을 통해 등록 전 데이터 건수를 확인 (db.getCollection('Users').count({})); 2. JMeter를 이용하여 시료데이터 10만건을 등록 요청: [POST] http://api.familygarden.co.kr/sign/up 요청 파라미터 : { _id: 사용자 아이디, pass: 사용자 비밀번호, phone: 사용자 핸드폰 번호, name: 사용자 이름 } 3. 등록이 완료되면 데이터베이스에 등록 된 데이터 건수를 확인 4. 2회 반복 수행하고 10만 건의 데이터가 정상적으로 등록되는지 확인
TC5	<p><사전 조건> - TC3의 사전조건과 동일함</p> <p><시험 방법></p> <ol style="list-style-type: none"> 1. DB 관리 툴을 통해 등록 전 데이터 건수를 확인 (db.getCollection('Rents').count({})); 2. JMeter를 이용하여 시료데이터 10만건을 등록 요청: [POST] http://api.familygarden.co.kr/rents 요청 파라미터 : { _f: 텃밭 고유키, area: 분양 면적, price: 분양 가격, expire: 분양 종료일, type: 결제 종류, note: 요청사항, date: 주문 일시 } 3. 등록이 완료되면 데이터베이스에 등록 된 데이터 건수를 확인 4. 2회 반복 수행하고 10만 건의 데이터가 정상적으로 등록되는지 확인

4. 시험환경

시험환경은 TTA 소프트웨어시험인증연구소 시험실에 구축하였고, 시험환경구성 및 세부사항은 아래와 같다.

<시험환경구성>



<세부사항>

No	Role	OS	CPU	Memory	HDD	Pre-Requisite
1	통신서버 (WAS)	Ubuntu 16.04 LTS	Intel(R) Xeon(R) CPU E5-2686 v4 2.30GHz	4GB	8GB	- Node.js v8.11.3
2	DB서버	Ubuntu 16.04 LTS	Intel(R) Xeon(R) CPU E5-2686 v4 2.30GHz	8GB	8GB	- MongoDB v4.0
3	Streaming 서버	Windows Server 2012 R2 Datacenter	Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz	64GB	2TB	- Node.js v10.14.2 - ffmpeg N-92857
4	클라이언트	Windows 10 Pro	Intel(R) Core(TM) i5-3230M CPU @2.60GHz	8GB	113GB	- JMeter 5.0 - Robo 3T 1.2.1 - java 1.8.0_191
5	모바일 단말기	- 모델명: Galaxy S7 edge(SM-G935A) - OS: Android 6.0.1				- 시험대상제품(모바일 앱)

5. 시험결과

시험 항목별 시험 결과는 아래와 같다. (상세 시험 결과는 <첨부 1> 참고)

ID	시험항목	시험목표	결과
TC1	실시간 영상 출력 확인	시험대상제품에서 ‘내 발’ 메뉴 클릭 시, 사용자의 뒷발이 실시간으로 영상을 통해 출력되는지 확인 ※ 실시간 영상은 5분 재생함	정상동작
TC2	저장 영상 출력 확인	CCTV로 녹화된 영상을 시험대상제품에서 일자별로 재생하여 볼 수 있는지 확인 ※ 저장된 영상은 CCTV로 24시간 녹화되는 영상이며 파일형태로 저장	정상동작
TC3	뒷발 정보 등록 처리량	하루에 등록 가능한 뒷발 데이터 정보가 10만개 이상인지 확인 ※ 성능측정 도구인 j-meter를 이용하여 뒷발 정보를 등록하며, 뒷발 데이터 파일은 시험의뢰기업에서 제공	평균 100,011건 등록 (19분 19초 소요)
TC4	사용자 등록 처리량	하루에 등록 가능한 사용자 데이터 정보가 10만개 이상인지 확인 ※ 성능측정 도구인 j-meter를 이용하여 사용자 정보를 등록하며, 사용자 데이터 파일은 시험의뢰기업에서 제공 ※ DB를 통해 등록된 사용자수 확인	평균 100,001건 등록 (1시간 10분 59초 소요)
TC5	주문 처리량	하루에 등록 가능한 주문 데이터 정보가 10만개 이상인지 확인 ※ 성능측정 도구인 j-meter를 이용하여 주문 정보를 등록하며, 주문 데이터 파일은 시험의뢰기업에서 제공 ※ DB를 통해 등록된 주문 데이터 개수 확인	평균 100,009건 등록 (18분 27초 소요)

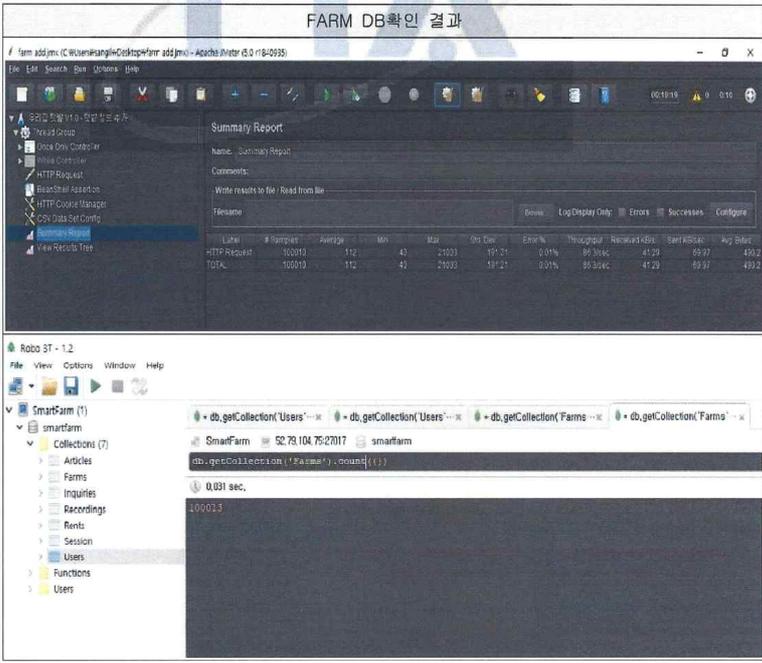
<첨부1> 상세 시험결과

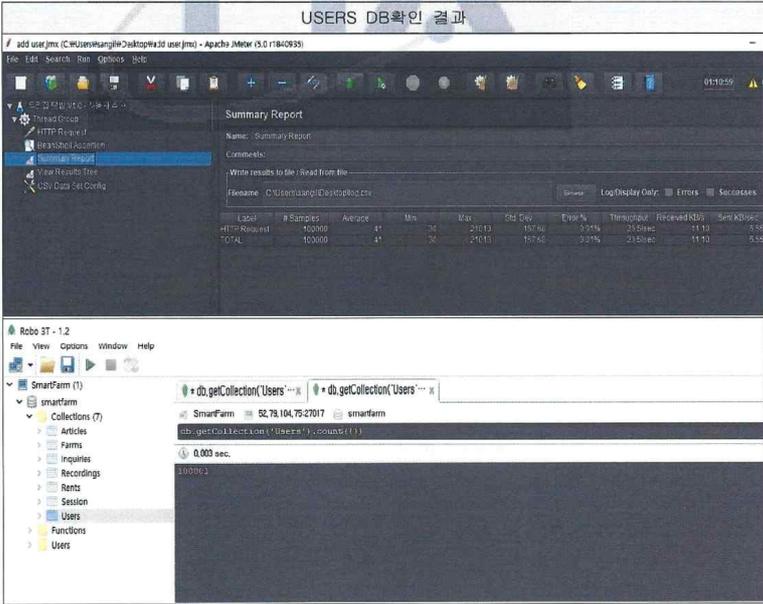
ID	TC1																																								
시험목표	시험대상제품에서 '내 밭' 메뉴 클릭 시, 사용자의 텃밭이 실시간으로 영상을 통해 출력되는지 확인																																								
상세 시험결과	<p>- 시험대상제품을 통해 사용자의 텃밭 영상이 실시간으로 출력되는지 확인한 결과, <u>정상적으로 동작하는 것을 확인함</u></p> <p>※ 영상 속 시각은 중국 시간(UTC+08:00 베이징)기준으로 현재 시각(UTC+09:00 서울)보다 1시간 느림</p> <p>※ "5. 시험결과"의 측정치에는 <u>20회 반복측정의 평균값</u>을 기재함</p> <p style="text-align: center;"><실시간 영상 출력 확인></p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 10%;"></td> <td style="width: 15%;">2019/1/23 14:20</td> <td style="width: 15%;">2019/1/23 15:18</td> <td style="width: 15%;">2019/1/23 16:41</td> <td style="width: 15%;">2019/1/24 12:46</td> <td style="width: 15%;">2019/1/24 09:03</td> </tr> <tr> <td>카메라1</td> <td>정상동작</td> <td>정상동작</td> <td>정상동작</td> <td>정상동작</td> <td>정상동작</td> </tr> <tr> <td>카메라2</td> <td>정상동작</td> <td>정상동작</td> <td>정상동작</td> <td>정상동작</td> <td>정상동작</td> </tr> <tr> <td>카메라3</td> <td>정상동작</td> <td>정상동작</td> <td>정상동작</td> <td>정상동작</td> <td>정상동작</td> </tr> <tr> <td>카메라4</td> <td>정상동작</td> <td>정상동작</td> <td>정상동작</td> <td>정상동작</td> <td>정상동작</td> </tr> </table> <p style="text-align: center;"><실시간 영상 결과 확인 예시></p> <p style="text-align: right;">※ 확인 일시는 한국 시간을 기준으로 기재함</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 10%;"></td> <td style="width: 40%;">2019/1/23 14:20</td> <td style="width: 50%;">2019/1/23 15:18</td> </tr> <tr> <td>카메라1</td> <td></td> <td></td> </tr> </table>						2019/1/23 14:20	2019/1/23 15:18	2019/1/23 16:41	2019/1/24 12:46	2019/1/24 09:03	카메라1	정상동작	정상동작	정상동작	정상동작	정상동작	카메라2	정상동작	정상동작	정상동작	정상동작	정상동작	카메라3	정상동작	정상동작	정상동작	정상동작	정상동작	카메라4	정상동작	정상동작	정상동작	정상동작	정상동작		2019/1/23 14:20	2019/1/23 15:18	카메라1		
		2019/1/23 14:20	2019/1/23 15:18	2019/1/23 16:41	2019/1/24 12:46	2019/1/24 09:03																																			
	카메라1	정상동작	정상동작	정상동작	정상동작	정상동작																																			
	카메라2	정상동작	정상동작	정상동작	정상동작	정상동작																																			
	카메라3	정상동작	정상동작	정상동작	정상동작	정상동작																																			
	카메라4	정상동작	정상동작	정상동작	정상동작	정상동작																																			
	2019/1/23 14:20	2019/1/23 15:18																																							
카메라1																																									

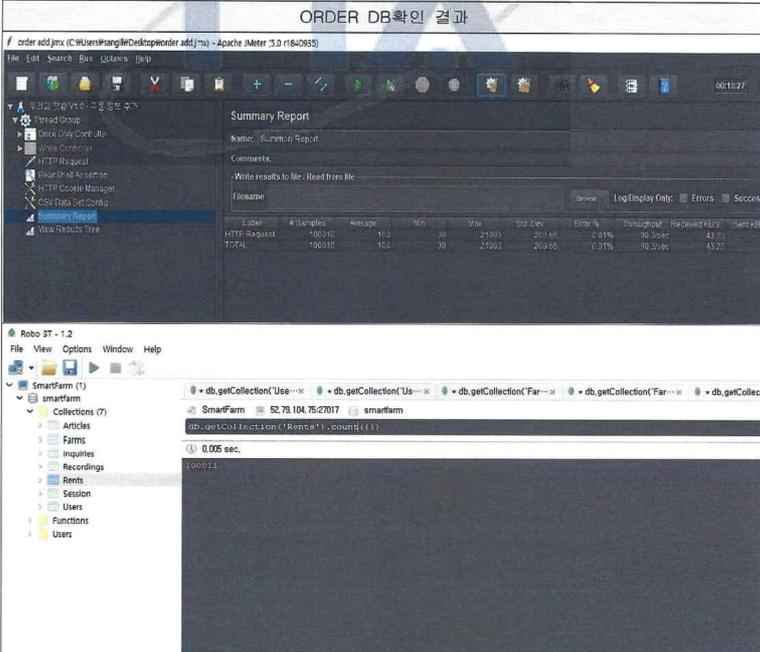
ID	TC1		
상세 시험결과	확인 일시	2019/1/23 14:20	
	항목	2019/1/23 15:18	
	카메라2		
	카메라3		
	카메라4		

ID	TC2					
시험목표	CCTV로 녹화된 영상을 시험대상제품에서 일자별로 재생하여 볼 수 있는지 확인 ※ 저장된 영상은 CCTV로 24시간 녹화되는 영상이며 파일형태로 저장					
상세 시험결과	- 시험대상제품에서 날짜별로 CCTV영상을 볼 수 있는지 확인한 결과, <u>정상적으로 동작하는 것을 확인함</u>					
	※ “5. 시험결과”의 측정치에는 <u>20회 반복측정의 평균값</u> 을 기재함					
	<저장 영상 출력 확인>					
	확인 일시	2019/1/22 09:00	2019/1/22 12:00	2019/1/22 20:00	2019/1/23 01:00	2019/1/23 08:00
	항목					
	카메라1	정상동작	정상동작	정상동작	정상동작	정상동작
카메라2	정상동작	정상동작	정상동작	정상동작	정상동작	
카메라3	정상동작	정상동작	정상동작	정상동작	정상동작	
카메라4	정상동작	정상동작	정상동작	정상동작	정상동작	
<날짜별 영상 결과 확인 예시>						
※ 확인 일시는 한국 시간을 기준으로 기재함						
확인 일시	2019/1/22 09:00		2019/1/22 12:00			
항목						
카메라1						
카메라2						

ID	TC2				
상세 시험결과					
	<table border="1"> <tr> <td data-bbox="379 696 496 775">확인 일시</td> <td data-bbox="496 696 831 775">2019/1/22 09:00</td> <td data-bbox="831 696 1169 775">2019/1/22 12:00</td> </tr> </table>	확인 일시	2019/1/22 09:00	2019/1/22 12:00	
	확인 일시	2019/1/22 09:00	2019/1/22 12:00		
	카메라3	 <p>08:00:01</p>			
카메라4	 <p>08:00:27</p>				

ID	TC3																				
시험목표	<p>하루에 등록 가능한 텃밭 데이터 정보가 10만개 이상인지 확인</p> <p>※ 성능측정 도구인 j-meter를 이용하여 텃밭 정보를 등록하며, 텃밭 데이터 파일은 시험의뢰기업에서 제공</p> <p>※ DB를 통해 등록된 텃밭 개수 확인</p>																				
상세 시험결과	<p>- JMeter를 이용하여 Farm테이블에 데이터를 넣었을 때, 등록된 데이터 건수를 확인한 결과 <u>평균 100,011건이 입력되는 것으로 확인됨</u></p> <p>※ Threads= 10, Ramp-up Period= 1, Loop Count= 10,000으로 설정함</p> <p>※ “5. 시험결과”의 측정치에는 2회 반복측정의 평균값을 기재함</p> <p style="text-align: center;"><텃밭 정보 등록 처리량></p> <table border="1" data-bbox="399 801 1157 958"> <thead> <tr> <th>회차</th> <th>샘플 데이터 수</th> <th>Error(%)</th> <th>Throughput</th> <th>등록 후 데이터 건수</th> </tr> </thead> <tbody> <tr> <td>1회차</td> <td>100,010</td> <td>0.01</td> <td>86.3/sec</td> <td>100,013</td> </tr> <tr> <td>2회차</td> <td>100,010</td> <td>0.01</td> <td>86.3/sec</td> <td>100,009</td> </tr> <tr> <td colspan="4" style="text-align: center;">평균(개)</td> <td>100,011</td> </tr> </tbody> </table> <p style="text-align: center;">FARM DB확인 결과</p>  <p>The screenshot shows two windows. The top window is the JMeter Summary Report for 'Farm add.jmx'. It displays a table with columns: Label, # Samples, Average, Min, Max, Std Dev, Error %, Throughput, Response Time, Sent Bytes, and Avg. Bites. The 'TOTAL' row shows 100,010 samples, an average of 112, and a throughput of 86.3/sec. The bottom window is Robo 3T, showing a query result for 'db.getCollection('Farms')' with a response time of 0.031 sec and a count of 100013.</p>	회차	샘플 데이터 수	Error(%)	Throughput	등록 후 데이터 건수	1회차	100,010	0.01	86.3/sec	100,013	2회차	100,010	0.01	86.3/sec	100,009	평균(개)				100,011
회차	샘플 데이터 수	Error(%)	Throughput	등록 후 데이터 건수																	
1회차	100,010	0.01	86.3/sec	100,013																	
2회차	100,010	0.01	86.3/sec	100,009																	
평균(개)				100,011																	

ID	TC4																				
시험목표	<p>하루에 등록 가능한 사용자 데이터 정보가 10만개 이상인지 확인</p> <p>※ 성능측정 도구인 j-meter를 이용하여 사용자 정보를 등록하며, 사용자 데이터 파일은 시험의뢰기업에서 제공</p> <p>※ DB를 통해 등록된 사용자수 확인</p>																				
상세 시험결과	<p>- jMeter를 이용하여 User테이블에 데이터를 넣었을 때, 등록된 데이터 건수를 확인한 결과 평균 100,001건이 입력되는 것으로 확인됨</p> <p>※ Threads= 1, Ramp-up Period= 1, Loop Count= 100,000으로 설정함</p> <p>※ “5. 시험결과”의 측정치에는 2회 반복측정의 평균값을 기재함</p> <p style="text-align: center;"><사용자 정보 등록 처리량></p> <p style="text-align: right;">※ 테이블에 데이터 1건이 기본적으로 있음</p> <table border="1" data-bbox="395 817 1158 972"> <thead> <tr> <th>회차</th> <th>샘플 데이터 수</th> <th>Error(%)</th> <th>Throughput</th> <th>등록 후 데이터 건수</th> </tr> </thead> <tbody> <tr> <td>1회차</td> <td>100,000</td> <td>0.01</td> <td>23.5/sec</td> <td>100,001</td> </tr> <tr> <td>2회차</td> <td>100,000</td> <td>0.01</td> <td>23.5/sec</td> <td>100,001</td> </tr> <tr> <td colspan="4" style="text-align: center;">평균(개)</td> <td>100,001</td> </tr> </tbody> </table> <p style="text-align: center;">USERS DB확인 결과</p>  <p>The screenshot shows two windows. The top window is the JMeter Summary Report for 'add user.jmx'. It displays a table with columns: Label, # Samples, Average, Min, Max, Std. Dev, Err% (Avg), Throughput, Received Kbytes, and Sent Kbytes. The 'TOTAL' row shows 100,000 samples, an average of 41, and a throughput of 23.5/sec. The bottom window is Rebot 1.2, showing a tree view of collections. The 'Users' collection is highlighted, showing a count of 100,001 and a duration of 0.003 sec.</p>	회차	샘플 데이터 수	Error(%)	Throughput	등록 후 데이터 건수	1회차	100,000	0.01	23.5/sec	100,001	2회차	100,000	0.01	23.5/sec	100,001	평균(개)				100,001
회차	샘플 데이터 수	Error(%)	Throughput	등록 후 데이터 건수																	
1회차	100,000	0.01	23.5/sec	100,001																	
2회차	100,000	0.01	23.5/sec	100,001																	
평균(개)				100,001																	

ID	TC5																																																		
시험목표	<p>하루에 등록 가능한 주문 데이터 정보가 10만개 이상인지 확인</p> <p>※ 성능측정 도구인 j-meter를 이용하여 주문 정보를 등록하며, 주문 데이터 파일은 시험의뢰기업에서 제공</p> <p>※ DB를 통해 등록된 주문 데이터 개수 확인</p>																																																		
상세 시험결과	<p>- jMeter를 이용하여 Order테이블에 데이터를 넣었을 때, 등록된 데이터 건수를 확인한 결과 평균 100,009건이 입력되는 것으로 확인됨</p> <p>※ Threads= 10, Ramp-up Period= 1, Loop Count= 10,000으로 설정함</p> <p>※ “5. 시험결과”의 측정치에는 2회 반복측정의 평균값을 기재함</p> <p style="text-align: center;"><주문 정보 등록 처리량></p> <table border="1" data-bbox="395 779 1155 936"> <thead> <tr> <th>회차</th> <th>샘플 데이터 수</th> <th>Error(%)</th> <th>Throughput</th> <th>등록 후 데이터 건수</th> </tr> </thead> <tbody> <tr> <td>1회차</td> <td>100,010</td> <td>0.01</td> <td>86.3/sec</td> <td>100,007</td> </tr> <tr> <td>2회차</td> <td>100,010</td> <td>0.01</td> <td>86.3/sec</td> <td>100,011</td> </tr> <tr> <td colspan="4" style="text-align: center;">평균(개)</td> <td>100,009</td> </tr> </tbody> </table> <p style="text-align: center;">ORDER DB확인 결과</p>  <p>The screenshot shows the JMeter Summary Report for 'ORDER DB확인 결과'. The report table is as follows:</p> <table border="1"> <thead> <tr> <th>Label</th> <th># Samples</th> <th>Average</th> <th>Min</th> <th>Max</th> <th>Std. Dev.</th> <th>Error %</th> <th>Throughput</th> <th>Received KB/s</th> <th>Sent KB/s</th> </tr> </thead> <tbody> <tr> <td>HTTP Request</td> <td>100010</td> <td>1.53</td> <td>89</td> <td>21503</td> <td>213.65</td> <td>0.01%</td> <td>86.3/sec</td> <td>43.21</td> <td>43.21</td> </tr> <tr> <td>TOTAL</td> <td>100010</td> <td>1.53</td> <td>89</td> <td>21503</td> <td>213.65</td> <td>0.01%</td> <td>86.3/sec</td> <td>43.21</td> <td>43.21</td> </tr> </tbody> </table> <p>Below the report, the Robo 3T interface shows a query execution result for 'db.getCollection(*Rent*').count() with a response time of 0.005 sec and a value of 100014.</p>	회차	샘플 데이터 수	Error(%)	Throughput	등록 후 데이터 건수	1회차	100,010	0.01	86.3/sec	100,007	2회차	100,010	0.01	86.3/sec	100,011	평균(개)				100,009	Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/s	Sent KB/s	HTTP Request	100010	1.53	89	21503	213.65	0.01%	86.3/sec	43.21	43.21	TOTAL	100010	1.53	89	21503	213.65	0.01%	86.3/sec	43.21	43.21
회차	샘플 데이터 수	Error(%)	Throughput	등록 후 데이터 건수																																															
1회차	100,010	0.01	86.3/sec	100,007																																															
2회차	100,010	0.01	86.3/sec	100,011																																															
평균(개)				100,009																																															
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/s	Sent KB/s																																										
HTTP Request	100010	1.53	89	21503	213.65	0.01%	86.3/sec	43.21	43.21																																										
TOTAL	100010	1.53	89	21503	213.65	0.01%	86.3/sec	43.21	43.21																																										