

IoT를 이용한 빌딩형 스마트팜

김영민, 김정명, 민형기, 방현준*
*수원대학교 정보통신공학과

kymin9601@naver.com, lord4255@naver.com, alsugdr11101@naver.com, guswns2811@naver.com

Building Smart Farm using the Internet of Things

Young-Min KIM, Jeong-Myeong Kim, Hyeong-Gi Min, Hyun-Jun Bang*

*Dept. of Information Communication, University of Suwon

요 약

현재 우리나라에서 스마트팜이 존재하고 있지만, 아직까지는 많은 불편함을 겪고 있다. 따라서 이에 본 논문은 사용자가 IoT를 이용하여 더 편하게 이용하고, 불편함을 최소화 할 수 있는 기술을 서술하여, 사용자에게 부담을 줄이고 더 많은 이득을 챙길 수 있도록 기대한다.

1. 서론

최근 정부에서는 농가 단위의 스마트팜 보급 전략을 보완하고, 정책대상을 청년 농업인, 진후방 산업으로 확대하고, 집적화된 확산거점을 조성하기로 했다. 스마트팜은 정보통신기술(ICT)을 활용해 ‘시간과 공간의 제약없이’ 원격으로, 자동으로 작물의 생육환경을 관측하고 최적의 상태로 관리하는 과학 기반의 농업방식이다. 농산물의 생산량 증가는 물론, 노동시간 감소를 통해 농업 환경을 획기적으로 개선한다. 빅데이터 기술과 결합해 최적화된 생산·관리의 의사결정이 가능하다. 최적화된 생육환경을 제공해 수확 시기와 수확량 예측뿐만 아니라 품질과 생산량을 한층 더 높일 수 있다[1].

최근 스마트폰 사용률의 증가로 인해 스마트폰을 이용한 시스템 구축이 용이한 실정이다. 20대부터 50대는 95% 이상이 스마트폰을 사용하고 2012년부터 2018년까지의 증가율로 볼 때 사용량은 지속적으로 증가할 것으로 예측된다[2]. 기존의 존재하는 어플리케이션은 단순히 사용자가 농장의 실시간을 확인하는 식이 많았고, 사용자가 스마트팜의 어플리케이션을 사용함에 있어 UI가 불편하거나, 필요한 기술이 없는 단점이 존재하였다.

이에 본 논문에서는 IoT를 접목한 빌딩형 스마트팜을 구축하여 기존에 존재하였던 어플리케이션의 단점을 보완하고, 사용자의 만족도와 효율을 상승시키도록 하였다. 본 논문에서 설계한 시스템 중 어플리케이션에서는 사용자들에게 로그인 시스템을 도입하여 농장의 보안성을 높이도록 하였고, 농장의 실시간 환경을 제공하고, 사용자의 효율을 높이기 위하여 자동화 시스템을 구현함으로써 추가적인 노동력을 감소하였다. 또한 사용자에게 특이사항 발생 시 알림을 제공함으로써 특이사항의 대처율을 높이도록 하여 사용자가 본 시스템을 사용함에 있어서 신뢰성을 더 높이도록 하였다. 또한, 서버에서는 농장의 실시간 데이터를 저장함으로써 사용자가 농장의 환경을 관측 및 활용을 하여 농식물의 생육을 극대화 할 수 있도록 하였다. 추가

로 사용자의 정보가 서버에 저장되는 것을 활용하여 보안성을 더 높이도록 하였다. 마지막으로 빌딩형으로 농장을 설계함으로써 사용자가 다양한 농작물을 키울 수 있고, 농장의 단점인 부지의 문제를 해결할 수 있도록 하였다. 본 논문에서 설계하고 구현한 프로그램을 이용한다면, 사용자가 초기비용의 필요성이라는 단점만 극복한다면 만족도를 크게 높일 수 있다고 기대한다.

2. 관련연구

2.1 안드로이드 스튜디오(Android Studio)

안드로이드 스튜디오는 안드로이드에서의 어플리케이션 개발을 위하여 Jet Brains 사의 IntelliJ IDEA를 기반으로 만든 통합 개발 환경이다. 무료로 제공되고 있으며 SDK 매니저를 통해서 안드로이드 버전 별 리소스나 C 및 C++을 빌드 할 수 있는 네이티브 어플리케이션 제작 등 다양한 개발도구들을 추가적으로 지원한다[3]. 이처럼 어플리케이션 구현에 활용도가 높기 때문에 본 논문에서는 Android Studio를 사용하여 어플리케이션을 디자인하고 기능을 구현하였다.

2.2 Retrofit2

Retrofit은 Square사에서 제공하는 오픈소스 라이브러리로 REST API를 안드로이드에서 쉽게 이용할 수 있게 해주는 도구 안드로이드 앱 프로젝트의 그래들에 다음과 같이 implementation 'com.google.code.gson:gson:2.8.5' implementation 'com.squareup.retrofit2:retrofit:2.6.0'를 추가하면 쉽게 Retrofit2 통신기술을 사용할 수 있다[4].

2.3 아두이노(Arduino)

아두이노는 IDII(Interaction Design Institutelvera)에서 하드웨어에 익숙지 않은 사람들이 자신들의 디자인 작품을 손쉽게 제어할 수 있게 하려고 고안되어 AVR을 기반으로 만든 통합 개발 환경이다. 마이크로컨트롤러를 쉽게

동작시킬 수 있으며 윈도를 비롯해 맥 OS X, 리눅스와 같은 여러 OS를 모두 지원한다. 아두이노 통합 개발 환경을 제공하며 임베디드 시스템 중의 하나로 쉽게 개발할 수 있는 환경을 이용하여, 장치를 제어할 수 있다. 이처럼 임베디드 시스템 구현에 활용도가 높기 때문에 본 논문에서는 Arduino를 사용하여 실제 생육환경 제어 기능을 구현하였다[5].

2.4 Django FrameWork

Django는 2005년부터 시작된 Python의 오픈 소스 웹 프레임워크이자 풀 스택 프레임워크이다. 당시 PHP 기반으로 작성된 웹사이트에서 규모가 큰 웹개발에 적합한 파이션 도구가 없다는 것을 깨닫고 홀로바티와 윌리슨은 장고를 개발하였다. 장고는 MVT[1] 패턴에서 Model 의 경우 데이터베이스와 연동이 편하게 만들어져 있다.

기본적으로 Model에 해당하는 DBMS는 SQLite를 사용하게 되어있으며, Django Admin 기능을 통해 쉽게 조회/편집이 가능하다 기본적으로 객체와 관계와의 설정을 나타내는 ORM(Object-relational mapping)을 기본적으로 제공하며 자주 사용되는 로직들을 묶어서 기본적으로 제공한다. 따라서 웹 개발에서 번거로운 요소들을 새로 개발할 필요 없이 내장된 기능만을 이용해 빠른 개발을 할 수 있다는 장점이 있다[6].

2.5 OpenWeatherMap

“OpenWeatherMap API 가이드”는 날씨 API 서비스를 원활하게 사용하기 위한 유용한 정보, 링크, 문서를 찾는 데 도움이 된다. OpenWeatherMap 무료임에도 불구하고 회원가입과 개인 API키 값을 받으면 어디서든 날씨에 대한 정보를 XML형태로 받을 수 있다. 당일의 날씨뿐만 아니라 5일 3시간 간격의 날씨정보, 일주일의 날씨 정보, 위도-경도에 기반한 날씨 정보까지 수집이 가능하다.

2.6 설문조사

그림 1은 본 논문의 생육환경 자동화 농장 구현의 타당성을 위해 실시한 설문조사 가운데 유의미한 결과를 나타내는 그래프이다. 설문조사는 경기도 농업기술원이 2019년도 귀농, 귀촌 교육 수료생 174명을 대상으로 실시하였다.

설문조사 결과 응답자의 대부분이 영농기술, 경험의 부족이 가장 어려웠던 점으로 꼽았다. 이에 본 논문에서 이를 바탕으로 농업 종사자들이 좀 더 쉽게 생육환경을 조성할 수 있도록 생육환경의 자동화 시스템을 구현하고 어플리케이션을 통해 문제점을 해결할 수 있는 기능을 적용하였다.

구분	인원	비율
영농기술·경험의 부족	22	69%
경제적 이유	10	27%
지역주민들과의 편견	4	11%
기타	1	3%

(그림 1) 설문 조사

3. 설계 및 구현

3.1 시스템 구조도

그림 2는 ‘IOT를 이용한 빌딩형 스마트팜’에 대한 SWOT 분석이다.

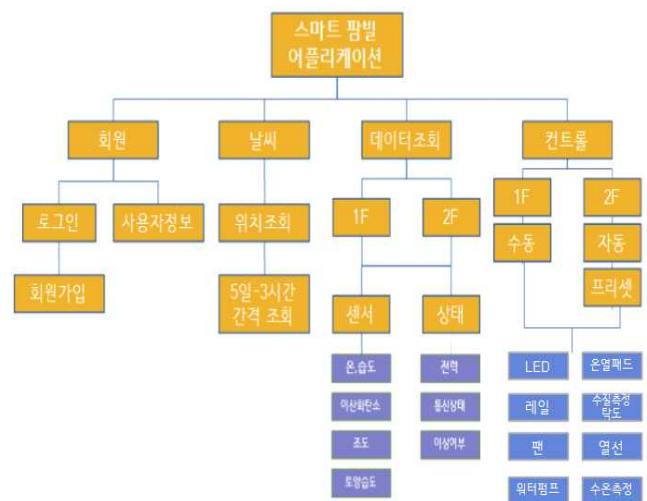


(그림 2) SWOT 분석

SWOT 분석을 통하여 SO 전략으로 증가하는 농업 종사자들에게 안전하고 편리하게 생육환경을 조성할 수 있는 생육환경 자동화 시스템을 제공하고 간편한 조작과 농장의 실시간 데이터를 제공하는 어플리케이션을 제공함으로써 많은 수의 이용자를 유입한다. ST 전략으로 잠재적 경쟁자보다 빌딩형 스마트팜으로서 같은 면적의 토지 대비 다양한 작물 생육과 많은 생산량을 유지할 수 있고 위험요소와 시스템 장애에 대한 즉각적 알람을 통해 안정된 생육환경을 조성할 수 있는 안정된 기능을 지원하여 시장 경쟁력을 유지한다[7].

3.2 시스템 구조도

그림 3은 본 논문의 어플리케이션의 핵심적인 기능을 나타낸 시스템 구조도이다.



(그림 3) 시스템 구조도

처음 로딩화면을 지난 후 로그인 화면이 나오고 등록된 사용자, 혹은 새로 회원가입을 한 후 어플리케이션 서비스 시작. 사용자 정보에 따라서 권한이 부여된 층의 농장의 데이터 확인과 각종 액추에이터 수동 컨트롤 서비스 실행 현재 위치의 위도, 경도에 맞춰 5일-3시간 간격의 날씨 정보를 받아옴. 해당 층(농장)의 4종 센서값 및 상태 값들을 얻어 정상작동 여부를 판별, 긴급상황 시 수동으로 액추에이터들을 컨트롤 하거나 상시에는 각 적용된 프리셋에 따라 자동으로 컨트롤한다.

본 논문에서 설계한 어플리케이션은 사람이 직접 식물들을 키우는 것이 아닌 값과 각종 정보들을 조합하여 액추에이터들이 자동으로 생육환경을 제배하는 편리함을 제공하고 사용자는 어플리케이션이 실시간으로 제공해주는 돌발상황에 대한 알림이나 이상 현상에 대해 수동으로 직접 처리만 해주면 된다.

3.3 DB설계

그림 4는 본 논문의 어플리케이션 구현을 위해 사용한 데이터베이스의 관계를 나타낸 관계도이다.



(그림 4) Database

본 논문에서는 auth_user, smartfarm_data_values_1f, smartfarm_data_values_2f, smartfarm_ctl_values_1f, smartfarm_ctl_values_2f 다섯 개의 데이터 베이스를 가지고 있다. 그림 5와 그림 6 그림 7은 각각 auth_user와 data_values, ctl_values 데이터베이스의 정의를 나타내고 있다.

id	PRI	int(11)
username	Field	varchar(150)
password	Field2	varchar(128)
is_superuser	Field3	tinyint(1)
email	Field4	varchar(240)
last_login	Field5	datetime(6)
is_staff	Field6	tinyint(1)
is_active	Field7	tinyint(1)

그림 5) auth_user

auth_user는 django에서 기본적으로 제공하는 관리자 페이지를 이용하였다. django model에 해당하는 DBMS와 SQLite를 연동하여 사용하였으며 회원가입과 로그인의

인증 암호화 기능이 제공되고 있다.

id	PRI	int(11)
name	Field	varchar(20)
tem_data	Field2	double
hum_data	Field3	double
light_data	Field4	double
sm_data	Field5	double
co2	Field6	double
created_at	Field7	datetime(6)

(그림 6) Data_values

smartfarm_data_values_(층)는 자신이 키우고자 하는 층에 sensor data를 실시간으로 저장하는 테이블이다.

id	Key	int(11)
fan_ctl	Field2	tinyint(1)
water_ctl	Field3	tinyint(1)
light_ctl	Field4	tinyint(1)
slide	Field5	tinyint(1)

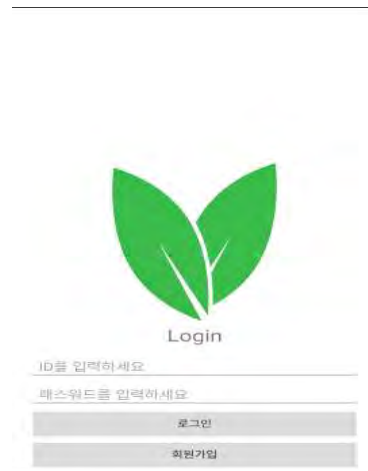
(그림 7) Ctl_values

smartfarm_ctl_values_(층)는 자신이 키우고자 하는 층에서 환경에 맞는 액추에이터를 작동시키기 위해서 on/off 기능을 할 수 있는 boolean 데이터로 구성하였다.

본 논문에서는 Android에서 회원가입을 하고 자신이 키우고자 하는 층으로 로그인을 하여 센서값을 보고 환경에 맞는 액추에이터를 작동시킬 수 있도록 데이터베이스를 구성하였다. User의 password는 암호화 기능으로 층별 보안을 강화하였고 너무 많은 data가 있으면 정보를 찾는 데 방해되기 때문에 data_values의 테이블 row 수를 제어하였다. 또한 data_values를 활용한 matplotlib로 데이터를 시각화 할 수 있도록 하였고 data_ctl를 통한 자동화를 할 수 있도록 기초를 마련하였다.

3.4 구현

그림 7은 메인화면 진입 전 로그인 화면, 등록된 사용자 혹은 회원가입을 통한 추가 된 사용자만이 어플리케이션 서비스를 사용할 수 있도록 설계



(그림 8) 로그인,회원가입 화면

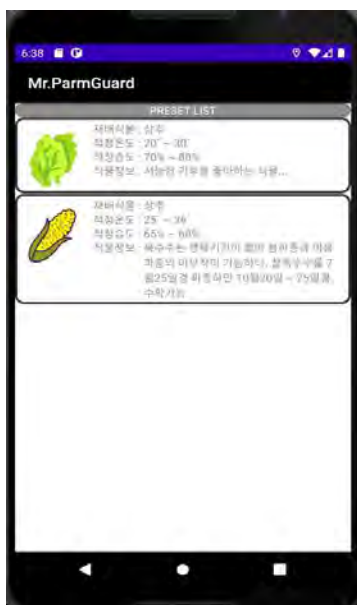
로그인 성공시 서버의 리스폰스 값으로 코드와 메시지를 받은 후 올바른 값일시 메인화면으로 이동

그림 8은 로그인 성공시 진입하게 되는 메인화면이다. 메인화면에는 현재 위치에 따른 각종 날씨 정보 및 센서 값 확인 액츄에이터들을 컨트롤 할 수 있는 패널 및 다른 서비스들을 연결해주는 버튼들이 구성되어 있다.



(그림 9) 메인 화면

그림 9는 자동화 버튼을 눌렀을 때 이동하게 될 자동화 프리셋 화면, (현재 2가지의 식물만 구현되어있지만 추후 다양한 식물을 추가할 예정) 식물에 대한 생육 정보들이 간단히 적혀있고 키우려고 하는 식물의 프리셋 버튼만 누르면 각종 액츄에이터들이 프리셋 정보와 센서값들을 기반해 자동으로 작동하게 됨



(그림 10) 자동화 프리셋 화면

4. 결론

본 논문에서는 농업 종사자를 위한 자동화 시스템을 탑재한 빌딩형 스마트팜 설계와 실시간으로 농장을 관리할 수 있는 빠르고 편리한 어플리케이션을 구현하였다. 본 논문에서 설계하고 구현한 어플리케이션은 각종 센서들을 통해 여러 환경에 대한 실시간 데이터를 활용하여 특이사항에 대하여 대처를 할 수 있도록 하였으며, 자동화 기능으로 농장물을 관리하는 데 있어 편의성 높이고 안정성 증가를 기대하였다. 또한, 수동으로도 사용자가 직접 컨트롤 할 수 있도록 설계하여 긴급 및 장애 상황을 빠르게 대처할 수 있도록 구현하였다. 추가로, 프리셋 기능을 통하여 농장물의 생육환경 정보를 제공한다. 본 논문에서 설계하고 구현한 시스템은 농업에 대한 전문적인 지식 없는 사람도 충분히 농업을 할 수 있도록 하였고, 지리적인 제약에 구애받지 않을 것이라고 예상한다.

참고문헌

- [1] 대한민국 정책브리핑, 정책위키(한눈에 보는 정책), <http://www.korea.kr/special/policyCurationView.do?newsId=148864055>
- [2] 한국갤럽, “2012~2018 스마트폰 사용 추이”, 한국갤럽 데일리 오피니언, 2018
- [3] Wikipedia - Android Studio, “https://en.wikipedia.org/wiki/Android_Studio”
- [4] Square Retrofit Github.io API Declaration - Request Method 참조
<https://square.github.io/retrofit/>
- [5] Wikipedia - Arduino, “<https://ko.wikipedia.org/wiki/%EC%95%84%EB%91%90%EC%9D%B4%EB%85%B8>”
- [6] Wikipedia - Django FrameWork, “[https://ko.wikipedia.org/wiki/%EC%9E%A5%EA%B3%A0_\(%EC%9B%B9_%ED%94%84%EB%A0%88%EC%9E%84%EC%9B%8C%ED%81%AC\)](https://ko.wikipedia.org/wiki/%EC%9E%A5%EA%B3%A0_(%EC%9B%B9_%ED%94%84%EB%A0%88%EC%9E%84%EC%9B%8C%ED%81%AC))”
- [7] 농림축산식품부 ' 정부규제개혁' 2017년 1월

[본 논문은 과학기술정보통신부 정보통신창의인재양성사업의 지원을 통해 수행한 ICT멘토링 프로젝트 결과물입니다]