Scientific Research Publishing

Journal of Computer and Communications, 2017, 5, 45-56
http://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Agricultural Traceable and Marketing System Based on iOS-Platform and Wireless Sensor Network

Haiyuan Sun[1], Jianping Zhang[2]*, Guiling Sun[1], Yangyang Li[1]

[1]College of Electronic Information and Optical Engineering, Nankai University, Tianjin, China
[2]School of Electronic Information and Electrical Engineering, Shanghai Jiaotong University, Shanghai, China
Email: haiyuansun@mail.nankai.edu.cn, *zhangjianping@sjtu.edu.cn, sungl@nankai.edu.cn, liyangyang@mail.nankai.edu.cn

How to cite this paper: Sun, H.Y., Zhang, J.P., Sun, G.L. and Li, Y.Y. (2017) Agricultural Traceable and Marketing System Based on iOS-Platform and Wireless Sensor Network. *Journal of Computer and Communications*, **5**, 45-56.
https://doi.org/10.4236/jcc.2017.56003

Received: March 7, 2017
Accepted: April 21, 2017
Published: April 24, 2017

Abstract

With the rapid development of Things of Internet and its related technologies such as Wireless Sensor Network and remote access system development, its application on agricultural and farming domain has become a new research direction. This paper discusses a practical and efficient system by applying mobile software development technologies and ZigBee communication network. To achieve a compact and flexible iOS-based client, this study takes advantage of Model-View-Viewmodel pattern as well as Client/Server style. Several encryption technologies such as Message Digest algorithm and Rivest Shamir Adleman algorithm are combined to ensure information security. Compared to the common portable marketing system, the proposed system also provides accurate growth information which is collected by wireless sensor network, and proposes a more flexible design pattern for mobile client hence obtaining a better performance on iOS devices.

Keywords

iOS Client, Online Agricultural Marketing, QR Code, MVVM Pattern, Wireless Sensor Network

1. Introduction

With the development of Internet of Things, Wireless Sensor Network (WSN) is widely applied in many agricultural applications. What's more, as the agricultural products safety issue becomes a focus of every country and the public, the application of WSN on agricultural growth trace is becoming a new challenging in this area. Recent years some short distance communication technologies such as

ZigBee and RFID have been discussed for agricultural product inspection, and [1] show a typical wireless sensor network deployed for agricultural and farming applications. However, the typical wireless sensor network applications cannot fulfill the entire products marketing and inspection tasks, and the traditional e-commerce mobile applications are redundant on its framework, hence they are not suitable for this purpose.

This paper proposes a system which combines the application of Internet of Things (IoT) and iOS platform-based mobile software development to collect and display agricultural products growth information, and also to offer products marketing function. The proposed iOS based client adopts Model-View-Viewmodel (MVVM) design pattern, which can directly update the view by binding without extra codes and therefore reduce the amount of code.

1.1. WSN Architecture Overview

This study presents a complete system as shown in **Figure 1**, which includes a wireless sensor network and a Client-Server (CS) structure. The wireless sensor network is deployed on fields for agricultural inspection and information collection. The field consists of sensor nodes powered with specific low-cost microprocessor such as CC2530. End devices in on-field sensor network communicate with routers using ZigBee communication technology. Since it is energy-efficient, low cost and reliable [1], the ZigBee technology is preferred for WSN-based applications in the area of farmland. The ZigBee-enabled devices, including

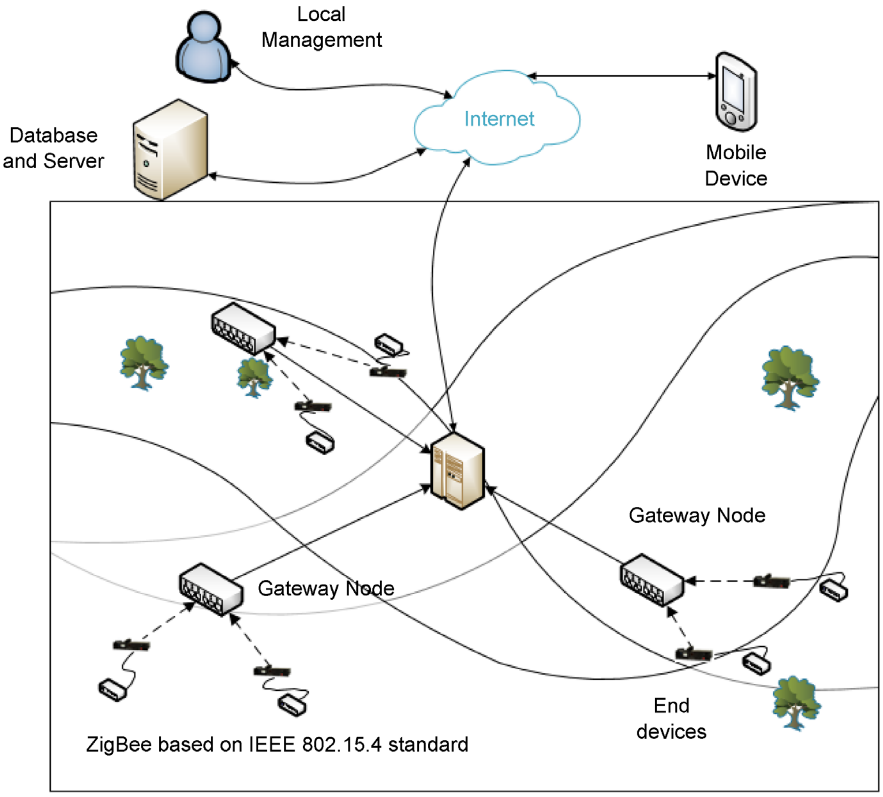


Figure 1. The architecture of the proposed system.

CC2530 SoC, support a low-duty cycle, and therefore are suitable for agricultural applications such as irrigation management and growth information collection, where periodic information update is required [1].

The network is a multi-tier architecture [1] [4] application and uses star topology [2] [3] [4] to communicate. The star topology for ZigBee network is shown as **Figure 2**. The sensors that deployed at a fixed positon and a base coordinator station form the basic unit of the network, and then these gateway nodes again form the upper tier [2]. The information collected by sensor nodes is eventually transmitted to a database deployed in the remote server from a PC connected with Internet. The local administrators access to the database for monitoring periodic information update from sensors and to make merchandise management. Products information and purchase products produced from the fields can be inquired by remote users carrying mobile devices through the proposed mobile client.

1.2. Client Overview

The proposed client is based on MVVM design structure [5], using Automatic Reference Counting (ARC) to manage memory automatically.

Client consists of following functions: agricultural products information display, agricultural products recommendation, agricultural products tracing, consumer/order management and other supplementary functions. Client/Server model [6] is applied in the integrated system, which requires an instant respond from the server when the client requests with a few data. Client and server exchange information through JSON format [7] file that provides a settled interface for data exchange. Besides, in the agricultural information tracing part, the QR Code as shown in **Figure 3** is chosen as the electronic recorder to identify the product.

2. Client Design and Implementation

2.1. Modularization

This client is separated into following modules:

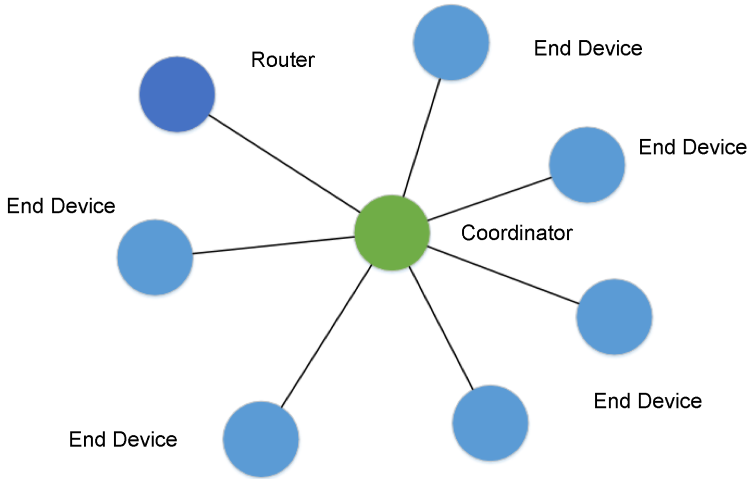


Figure 2. Star network topology for ZigBee.

- Product Display Module: This module is the core module. It displays the selling products from local agricultural base. Users explore and search the displaying items and select them according to their preferences. Agricultural growth information also can be obtained from corresponding view.
- Product Recommend Module: Product recommend module collects user shopping preferences, and display daily recommend items according to the optimized recommend algorithm from the backend server. It also provides push services when users are in a given local area or at a timestamp to promote products.
- Product Tracing Module: There are two important functions this module needs to achieve: the first is to display plant farming information through a web view, the other one is to scan the QR Code on the external pack of product as shown in **Figure 2** through the camera on mobile devices, and then parse and decode response data to users.
- Order/user Management Module: This module is in charge of users' order and personal information management. Users can browse, modify and cancel their order forms. This module will automatically send HTTPS requests to get order and personal information and refresh UI to display these data.
- Supplementary Modules: Other supplementary modules include: shopping trolley, payment as well as preference settings.

2.2. Methodology

MVVM Pattern on iOS

The typical iOS-Platform based software and software on other mobile operation system normally adopts MVC (Model-View-Controller) design pattern [8], which is demonstrated as **Figure 4**. This structure can decouple model and view

Figure 3. Using QRCode for identification.

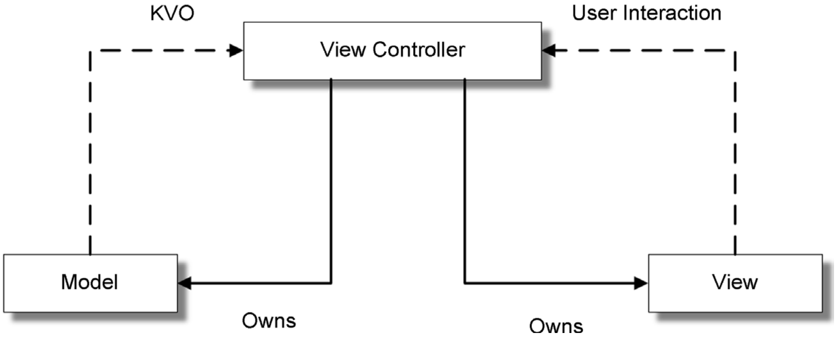


Figure 4. MVC architecture on iOS platform.

for code reuse and parallel development. According to MVC, the controller communicates with both View and Model. For example, when a model object changes, the model object usually communicates that change to a controller object, which then requests one or more view objects to update themselves accordingly. The direct contact between model and view is forbidden in this structure. Numerous characteristics such as simultaneous development, high cohesion, low coupling and multiple views for a model are benefits from this pattern [8].

MVC has been introduced for over 30 years, and its advantages have been appreciated by many software developers. However, in mobile platform such iOS, when the functions and modules of program have been build more and more complicated, a large amount of services, for example, responding to delegation messages or observing notifications, responding to action messages, managing the life cycle of owned objects (for example, releasing them at the proper time), and establishing connection between objects and performing initialization, are necessary to be provided by the controller. Therefore, the controller will become extremely massive and hence pose a threat to the program's efficiency and fluency.

To deal with this problem, the proposed client uses an iOS-based Model-View-ViewModel (MVVM) design pattern. MVVM is initially developed for .NET platform to deal with architectural pattern problem on WPF application system, mainly apply on project development on WPF and Silverlight. MVVM pattern can separate business and logic, increase the reusability of code [9]. At the same time, it makes program easier to develop, test and maintain.

The hierarchical structure of MVVM pattern is shown as **Figure 5**. Model tier is in charge of real data and information, records all the business objects and the relations of objects. View tier communicates with users frequently, and it displays the information to them. The duty of view model tier is coordinating controls in view tier and objects in model tier, and therefore loose coupling between view tier and view model tier. Similar to MVC architecture, the model tier is not supposed to exchange data with view tier.

Li L [5], Gao [9] and Li X [10] presented various applications, which are based on MVVM design pattern, in their papers to solve different problems on software architecture. Inspired by their works, we organize a MVVM pattern based on iOS system architecture and this is also suitable for other mobile operating systems. The following steps have been taken:

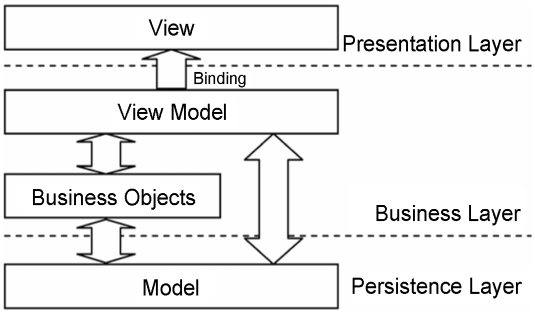


Figure 5. Typical MVVM patter structure.

- Define view model classes.
- Decompose the network services module from the controller, and move it to view model class.
- Move the data processing and presentation logics, including data formatting, data localization operation and information display management, into view model classes.
- Supply data checking and error report in view model tier.

As shown in Figure 6, view model tier mergers complicated logics, leaving the tailored controller become a part of view tier.

Interface Building and Optimization

On iOS, the information display mostly relies on UITableView and UICollectionView SDKs from UIKit, which is provided by Apple and is the crucial infrastructures to manage interface. These components get data from the data source object, which is an object that conforms to the data source and delegate protocol. The data source provides information that UITableView and UICollectionView needs to construct and manages the data model when rows of a table are inserted, deleted, or reordered. The delegate manages table row configuration and selection, row reordering, highlighting, accessory views, and editing operations.

There are many modules integrated in the client, for example, product display, product recommend and user information management, use UICollectionView and UITableView to present data. To build heterogeneous interfaces, we create numbers of custom objects in with content or behavioral characteristics that are different from each other. However, the problem is, when we code these module, a large amount of repeated codes make the controller become inevitably heavy. According to the business logic for specified system and object oriented programming (OOP) principle, beside set up MVVM architecture stated above, some efficient optimization measures are proposed in this paper.

As an example, Figure 7 shows the overall encapsulation workflow for UITableView, which is similar to UICollectionView. The native protocols provided by UIKit are assigned to custom objects, which defines some custom handling code. Custom base objects conform to the above user-defined protocol rather than native APIs. Base objects also define common operations. Therefore, child objects inherited from base objects simply implements their custom functions

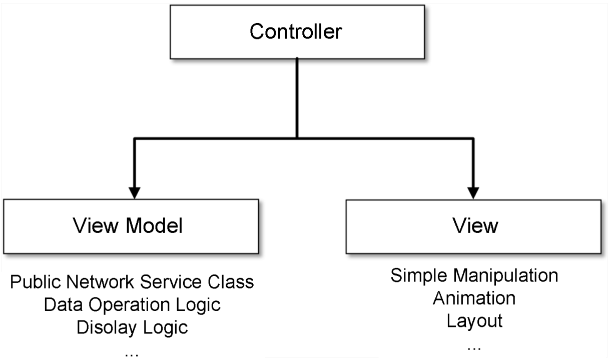


Figure 6. New view model tier and view tier.

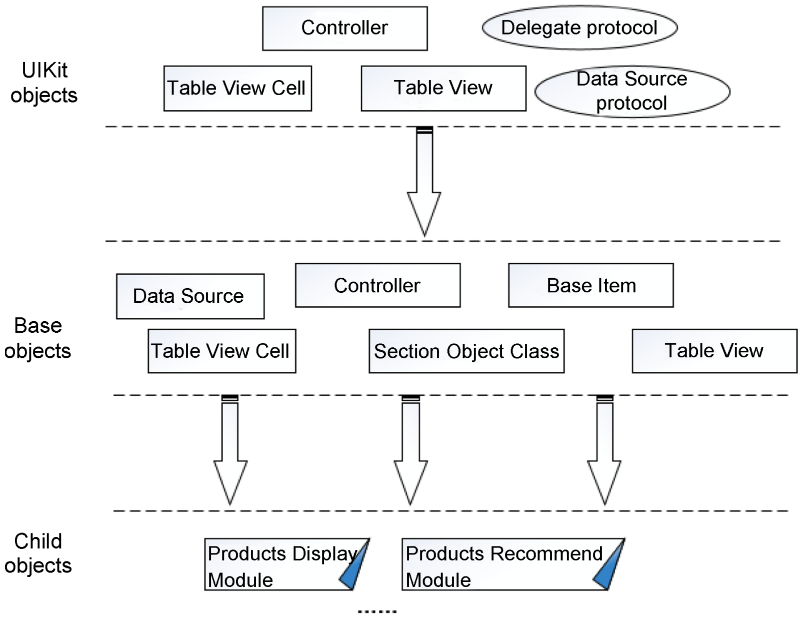


Figure 7. Define base classes to encapsulate UI Table View.

and methods according to different interface scenarios.

Information Security

As a C/S application, HTTPS protocol is contained for network communication. HTTPS, the secure version of HTTP, runs HTTP on top of SSL/TLS, which is the standard protocol for providing authenticity and confidentiality on top of TCP connections [11]. Compared to HTTP, HTTPS provides authentication of the client and server, and additionally, bidirectional encryption of communications. These characteristics help protecting against many hacking actions. In addition, user's personal information such as account and payment information are encrypted through Message Digest Algorithm (MD5) [12] before transmit to the server.

In the products information tracing module, mobile devices users use internal cameras to scan QR Code on the product outer packages to read planting information. To handle the QR Code recognition issue, some effective QR Code scanning algorithms are presented by researchers [13] [14]. However, to reduce the client system complexity, we choose to use native APIs such as CIDetector class to recognize the barcode on images. Messages in QR Code are decrypted with Rivest Shamir Adleman (RSA) algorithm from clear data. RSA involves a public key and a private key. The public key can be known by everyone and is used for encrypting messages. Security. Framework from iOS SDK provides RSA decryption function using the private key. The workflow of RSA decryption is shown as Figure 8.

Data Cache and Persistence

The presentation of product requires to download a large amount of image data from the server. To control the network flow, the client uses a two-stage data cache mechanism: using NSCache object to cache thumbnails that was downloaded, because these images will be browse frequently in short time periods; on the other hand, the client uses NSFileManager to cache those large images onto local disk.

The cache system is shown as Figure 9. Every image downloaded from the server is labeled by a specific key for a NSCache object and also as the key for the image file on the disk. Before the client send network requests for image, it will check if there is an image in the memory cache according to the image key. If there is not exist such image, the client will create a NSInvocationOperation object to locate this image asynchronously on the disk and return the result in the main thread. If this image is stored on the disk, it will be added to the memory. If these above steps cannot obtain the image, the client will create a new child thread to download the image asynchronously.

2.3. User Interface Presentation and Test

The proposed client keeps its user interface simple, clear and convenient for manipulation. This application is implemented in the 6th generation of iPhone with iOS 9.2. The actual runtime test is illustrated as following. Figure 10(a) shows the home module that displays full product item lists for users. The bottom tab bar is provided for user to manage the controllers and their relative views. The shopping cart module is shown as Figure 10(b). Users can look over

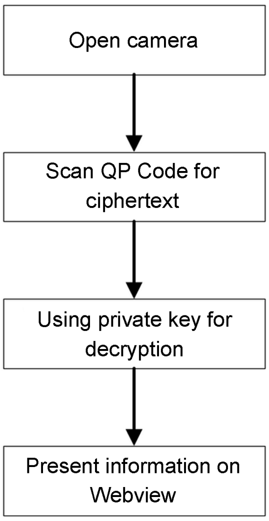

Open camera
Scan QP Code for ciphertext
Using private key for decryption
Present information on Webview

Figure 8. Workflow for RSA decryption.

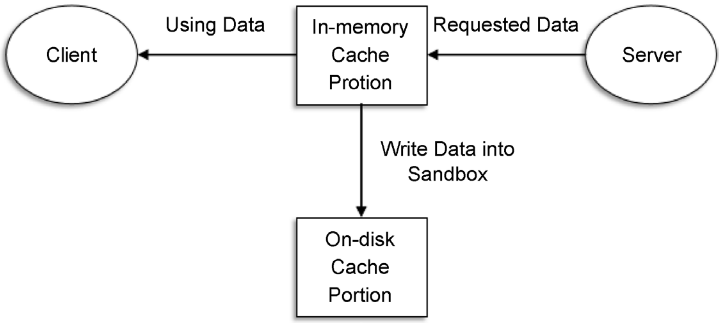
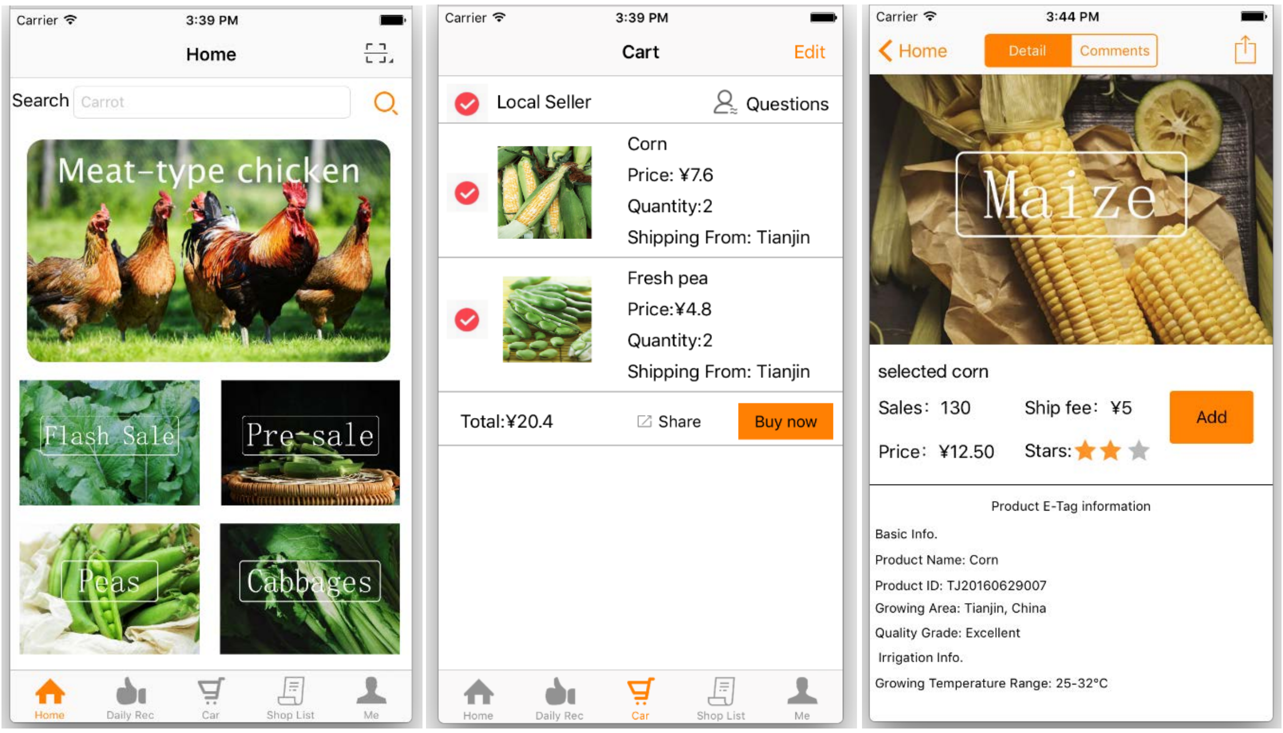


Figure 9. Cache strategy.

selling and planting information through product detail module as shown in Figure 10(c).

3. System Implementation and Test

To test the integrated function of the proposed system, we deployed 16 nodes on field and divided the subject terrain into 16 areas based on different farming purpose, and the subject area is 65,000 m^2. Each node consists of one image capture sensor and other pertinent sensors. Table 1 shows us the actual deployment parameters. These nodes transmit their data to the base station every 10 min. Our test results on on-field ZigBee module indicate that 80 - 100 meters is the most reliable transmission distance range. Besides, other planting messages such as fertilization and weeding information are manually input into the database.



Corn
Price: ¥7.6
Quantity:2
Shipping From: Tianjin

Fresh pea
Price:¥4.8
Quantity:2
Shipping From: Tianjin

Total:¥20.4

selected corn
Sales: 130 Ship fee: ¥5
Price: ¥12.50 Stars:

Product E-Tag information
Basic Info.
Product Name: Corn
Product ID: TJ20160629007
Growing Area: Tianjin, China
Quality Grade: Excellent
Irrigation Info.
Growing Temperature Range: 25-32°C

(a) Home module (b) Cart information (c) Product detail

Figure 10. UI test.

Table 1. Deployment parameters.

Parameter	Value
Temperature and humidity senor	DHT11 (http://www.aosong.com)
Light intensity sensor	SM3590B (http://www.sonbus.com)
ZigBee radio module	CC2430 (http://www.ti.com)
No. of nodes	16 for each area
Transmission range	80 - 100 m
Inter-node communication	ZigBee, 2.4 GHz
Architecture	Multi-tier heterogeneous star network

Table 2 shows the table that records the planting environmental information on the remote server.

With Internet access, the client delivers user's shopping information to the server, which keeps the data as UML class in the database. The UML object is shown as Figure 11. We also evaluated the QR Code recognition function on our hardware architecture. We captured 50 distinct barcode images on product wrappings with normal light conditions, the recognition rate is about 95% and the average time consumption is less than 500 ms, which is acceptable for our system.

With Internet access, the client delivers user's shopping information to the server, which keeps the data as UML class in the database. The UML object is shown as Figure 11. We also evaluated the QR Code recognition function on our hardware architecture. We captured 50 distinct barcode images on product wrappings with normal light conditions, the recognition rate is about 95% and the average time consumption is less than 500 ms, which is acceptable for our system.

4. Conclusion

The wireless sensor network and iOS based agricultural products marketing and traceable system has been proposed in this paper. The on-field sensors collect multiple agricultural growing data, and then transmit these data to base station through ZigBee. C/S structure is suitable for this application. The data on base

Table 2. Growth environmental information.

Field Name	Description	*Type*	*Length*	*Default value*	*Key*
SAF01	Serial number	*Int*	20		*Yes*
SAF02	Temperature	*Varchar*	20		
SAF03	Humidity	*Varchar*	20		
SAF04	Light Intensity	*Varchar*	20		
SAF05	Fertilizer	*Varchar*	20		
SAF06	Pesticide	*Varchar*	20		
SAF07	Record time	*Varchar*	20		

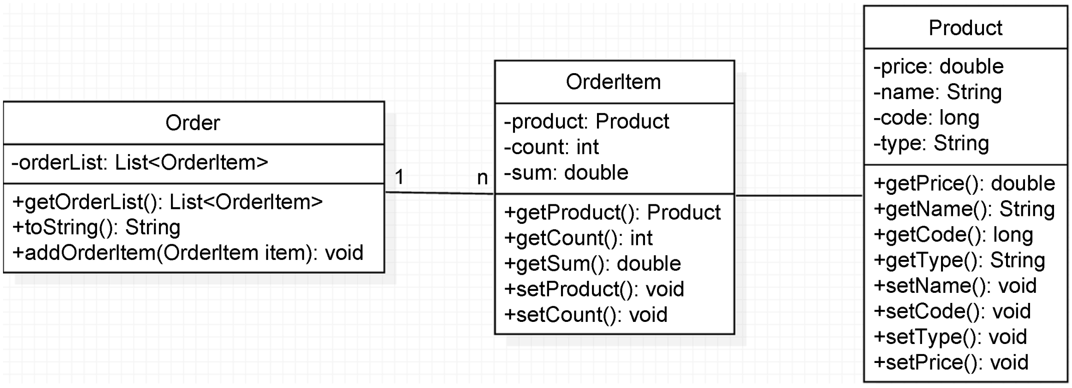


Figure 11. UML object on database for an order.

station are transmitted to remote server through GPRS network. The iOS-based portable client sends HTTPS requests to the server for agricultural planting and transaction data. These data to be transmitted are encrypted through MD5 algorithm. QR Code that encrypted by RSA is applied for product identification and therefore user can read the traceable information. Test results on QR Code scanning function indicate that the native APIs detect as well as read the barcode images on product packages efficiently without any optimization algorithms. Optimization steps on UIKit such as UITableView for iOS markedly simplified the business logic, and these measures make a contribution to a better UI performance. Additionally, compared to traditional MVC software architecture, a mobile-based MVVM pattern not only reduces the complexity of controllers, but also separates the complicated business and network logics from presentation module, avoiding the confusion in multi-view application development and making the client more convenient to develop and maintain.

Acknowledgements

Our work is supported by The Application of Internet of Things on Agriculture and Green Source Products Information Development project, Tianjin Rural Work Committee.

References

[1] Ojha, T., Misra, S. and Raghuwanshi, N.S. (2015) Wireless Sensor Networks for Agriculture: The State-of-the-Art in Practice and Future Challenges. *Computers & Electronics in Agriculture*, **118**, 66-84.

[2] Li, J., Zhu, X., Tang, N., *et al.* (2010) Study on ZigBee Network Architecture and Routing Algorithm. *International Conference on Signal Processing Systems*, Dalian, 5-7 July 2010, 389-393. https://doi.org/10.1109/icsps.2010.5555486

[3] Yu, X., Wu, P., Han, W., *et al.* (2013) A Survey on Wireless Sensor Network Infrastructure for Agriculture. *Computer Standards & Interfaces*, **35**, 59-64. https://doi.org/10.1016/j.csi.2012.05.0019

[4] Farahani, S. (2008) ZigBee Wireless Networks and Transceivers. Newnes, Newnes.

[5] Li, L. (2012) MVVM Pattern. *Microcomputer Applications*, **28**, 57-60.

[6] Jiang, J. (2015) Research on the Distributed Big Data Management of Bank Based on the Three-Tier C/S Model. *Metallurgical & Mining Industry*, **9**, 56-62.

[7] Severance, C. (2012) Discovering JavaScript Object Notation. *Computer*, **45**, 6-8. https://doi.org/10.1109/mc.2012.132

[8] Sanderson, S. (2010) Pro ASP.NET MVC Framework. Apress, New York.

[9] Gao, B., Zhang, S. and Yao, N. (2012) A Multidimensional Pivot Table Model Based on MVVM Pattern for Rich Internet Application. *International Symposium on Computer*, *Consumer and Control*, Taichung, 4-6 June 2012, 24-27. https://doi.org/10.1109/IS3C.2012.16

[10] Li, X. (2015) Application of MVVM Design Pattern in MES. 2015 *IEEE International Conference on Cyber Technology in Automation*, *Control*, *and Intelligent Systems*, Shenyang, 8-12 June 2015, 1374-1378. https://doi.org/10.1109/CYBER.2015.7288144

[11] Naylor, D., Finamore, A., Leontiadis, I., *et al.* (2014) The Cost of the "S" in HTTPS.

Proceedings of the 10*th ACM International on Conference on Emerging Networking Experiments and Technologies*, Sydney, 2-5 December 2014, 133-140. https://doi.org/10.1145/2674005.2674991

[12] Bindiya, M.K. and Kumar, R. (2015) Securing Big Data over Network Using MD5 Algorithm Technique. *International Journal of Computer Applications*, **123**, 14-17.

[13] Liu, Y, Yang, J. and Liu, M. (2008) Recognition of QR Code with Mobile Phones. *Control and Decision Conference*, Yantai, 2-4 July 2008, 203-206.

[14] Belussi, L. and Hirata, N. (2011) Fast QR Code Detection in Arbitrarily Acquired Images. 24*th SIBGRAPI Conference on Graphics*, *Patterns and Images*, Sibgrapi, 28-31 August 2011, 281-288.