Proceedings Paper:

Fabiyi, SD orcid.org/0000-0001-9571-2964, Ren, J, Han, Y et al. (2 more authors) (2022) Mobile Platform for Livestock Monitoring and Inspection. In: 2022 3rd International Informatics and Software Engineering Conference (IISEC). 3rd International Informatics and Software Engineering Conference, 15-16 Dec 2022, Ankara, Turkey. IEEE . ISBN 978-1-6654-5996-9

https://doi.org/10.1109/IISEC56263.2022.9998297

Mobile Platform for Livestock Monitoring and Inspection

Samson Damilola Fabiyi
School of Computing
University of Leeds
Leeds, United Kingdom
s.d.fabiyi@leeds.ac.uk

Jinchang Ren
National Subsea Centre
Robert Gordon University
Aberdeen, United Kingdom
j.ren@rgu.ac.uk

Yukang Han
Innovent Technology
Roslin, Midlothian, United Kingdom
yh@itlscotland.co.uk

Qiming Zhu
Innovent Technology
Roslin, Midlothian, United Kingdom
qz@itlscotland.co.uk

David Barclay
Innovent Technology
Roslin, Midlothian, United Kingdom
db@itlscotland.co.uk

***Abstract*—Livestock keepers acquire and manage information (e.g. identification numbers, images, etc.) about livestock to identify and keep track of livestock using systems with capabilities to extract such information. Examples of such systems are Radio Frequency Identification (RFID) systems which are used to collect and transmit livestock's information to host devices. Sophisticated RFID readers are very expensive, and more functional than the cheap ones whose use are mostly limited to reading and transmission of tag IDs. Cross-platform mobile applications will allow monitoring of livestock irrespective of the platform on which mobile devices are being operated. Farmers' secured access to records via web services is not limited to a device as they can login on any mobile device with the installed application. In this work, a mobile platform which consists of a cross-platform mobile application, web-service and database is developed to cost-effectively manage and exploit records of livestock acquired using a cheap RFID reader. The mobile application was developed using a Xamarin form framework. The programming language and development environment used are C# and Visual studio respectively. Records of livestock were acquired, posted, updated, deleted and retrieved from the database via a web service. Additional advantages offer by the solution implemented include, exporting of animals' records via email and SMS, viewing of animal's record by scanning their tags or QR code of animals' passports, and login system to sign users in and out of the application. Development of RFID readers with sensors to acquire health-related parameters for health monitoring is recommended.**

***Keywords*—livestock, mobile application, RFID systems, web service, xamarin, database**

I. Introduction

Commercial livestock keepers work towards satisfying the interest of their customers in products which are of high quality [1]. They are not only concerned with profit maximization (one of the key objectives of any farm business) but are also mindful of the need to meet the specifications and requirements of their clients [2]. Livestock keepers monitor and inspect livestock under their watch from time to time in other to ensure their healthy living. Decreased productivity arising from poor growth and sudden death of livestock can be attributed to failure on the part of commercial livestock keepers to take pre-emptive steps [3]. In precision Agriculture, certain information about livestock like weights, temperature, identification numbers, images, etc. are usually collected by livestock keepers and stored to facilitate inspection and monitoring of livestock's health and safety [1, 4, 5]. Keeping track, identification and performance monitoring of livestock can be realized by the use of systems and devices with capabilities to extract information about livestock. Examples of devices and systems with such capabilities are those that use barcodes and Radio Frequency Identification (RFID) technologies.

Quick response (QR) codes are two-dimensional barcodes which are used to encode and store information [6, 7]. Their major advantages over one-dimensional barcodes are the high rate at which information stored in them can be read and their capacity to store large amount of information [7, 8]. Though QR codes were originally invented for tracking operations in the automotive industry, they have found great applications in device authentication, attendance management, login systems, online payment, multimedia management, and several other smartphone applications [7, 8, 9]. Barcode technology used to be a prominent technology in electronic identification and keeping track of livestock. However, RFID technology is becoming more prominent with wide acceptance [3].

Radio Frequency Identification (RFID) systems are used to automatically and wirelessly identify and keep track of tagged objects [10, 11]. They operate by transmitting radio waves wirelessly from an RFID reader via a transmitter (antennae), modulating and reflecting the transmitted signal (once it is detected by RFID tags which are within the transmission range) back to the RFID reader. The reflected signal usually contains information about animals which are either pre-stored in the RFID tags or measured by sensors which are attached to the RFID tags. The RFID reader receives this information (also via the antennae) and transmit it to a host device for processing and records management [11, 12]. Hence, the three key constituents of RFID systems are RFID reader, RFID tag and a host device [11, 13]. Fig. 1(a) shows the diagram of an RFID reader. The key constituents of a tag are chips, antennae (for signal transmission and reception) and memory (to store information about livestock). Information about livestock that can be stored in the tag's memory for identification and monitoring are tag IDs, body temperature, weights and others which are environmentally related [13, 14]. Fig. 1(b) shows the diagram of a tagged animal [15].

Processing of information extracted by RFID readers and management of livestock records are done on computers and mobile devices. Software applications are installed on these host devices to facilitate information processing and records management. Livestock records are stored in databases. The database can be a local database installed on a host device alongside the application or web-based database which can be accessed remotely by the application. Webservice, an application programming interface (API) that allows systems to interact with one another using web technology, can facilitate remote access to a web-based database [16, 17, 18].

978-1-6654-5995-2/22/$31.00 ©2022 IEEE

(a) (b)

Fig. 1 (a) An RFID reader (b) Tagged animals

Farmers' secured access to records via webservice will not be limited to a device as they can login on any mobile devices with the installed mobile application.

Users of mobile applications possess mobile devices with different platforms. Different platforms provide different APIs (Application Programming Interfaces) and required different programming languages and development environments, hence the cost, time and effort required to bring different people with different expertise together to use separate development environments, different programming languages and software development kits (SDKs) far exceed those required when the application targets only one platform. A cost and time effective approach would be to use frameworks that allow applications which are developed using a single code base, a single programming language, a unified API, and the same development environment to be deployed across different platforms. Mobile applications which are developed using this approach are commonly referred to as cross-platform mobile applications [19, 20]. Use of cross-platform mobile applications will allow monitoring of livestock irrespective of the platform on which mobile devices are being operated.

Sophisticated RFID readers are generally very expensive, and relatively more functional with advanced features ranging from large screen for high readability, keypads with enough keys to manipulate records stored on readers, software to connect to host devices for downloading of livestock's records, large memory capacity and relatively long reach. In contrast, cheap RFID readers lack the aforementioned features and their use are mostly limited to reading and transmission of data to host devices. Since records acquired are usually downloaded to host devices when connected to the readers, access may become impossible due to impairment or loss of host devices. This work is therefore aimed at implementing a cost-effective solution to identify and keep track of livestock (outgoing, incoming and on-farm) and matching farm animals with their IDs. Specifically, in this work, we use a less functional RFID reader, which can only scan tags and transmit their IDs to host devices, and make the following contributions:

1. A mobile platform which consist of a mobile application, web service and a database (which can be remotely accessed via the web service) is developed to manage and exploit the records of livestock acquired by the RFID reader.

2. Tag IDs and QR codes of animals' passports are acquired using the developed mobile platform. By reading the tag IDs on livestock using the RFID reader or simply scanning the QR codes of animals' passport using the mobile device with the installed application, livestock keepers can identify, view the records and keep track of livestock on their farms.

3. Using a xamarin form framework and c# programming language, a cross-platform mobile application that targets Android, iOS and Windows platforms is developed to ensure that livestock keepers can use the mobile platform to monitor their livestock irrespective of the platforms on which their mobile devices are being operated.

4. Features are incorporated in the mobile platform to allow exporting of animals' records via email and SMS. Also, a web service is implemented to ensure that livestock keepers can log in on any device with the installed mobile application. This ensures on-farm and off-farm access to the records of livestock.

5. The development of RFID readers which incorporate sensors that can acquire not just IDs but also other information like weights and temperature for health monitoring is recommended.

Additionally, portions of this paper were submitted by the first author in project report form in fulfilment of the requirements for the award of MSc degree at the University of Strathclyde, Glasgow, UK [21].

II. Materials And Methods

A. System Architecture

Architecture of the system implemented in this work is shown in Fig. 2. The major components of the system adopted are mobile application, mobile phone, database, web service endpoint, RFID reader and tags. The mobile application was developed, installed on a mobile phone and used to capture the image and content of QR code of an animal's passport. The RFID reader was used to scan the RFID tag and extract its ID. The RFID reader transmits extracted data to the mobile application. The mobile application is responsible for the transmission of the read tag IDs and information about the QR code to a database on a server via a web service endpoint.

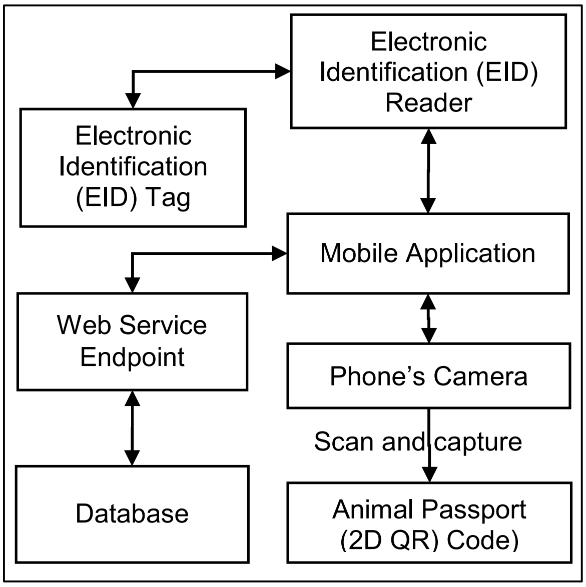

Fig. 2 System architecture

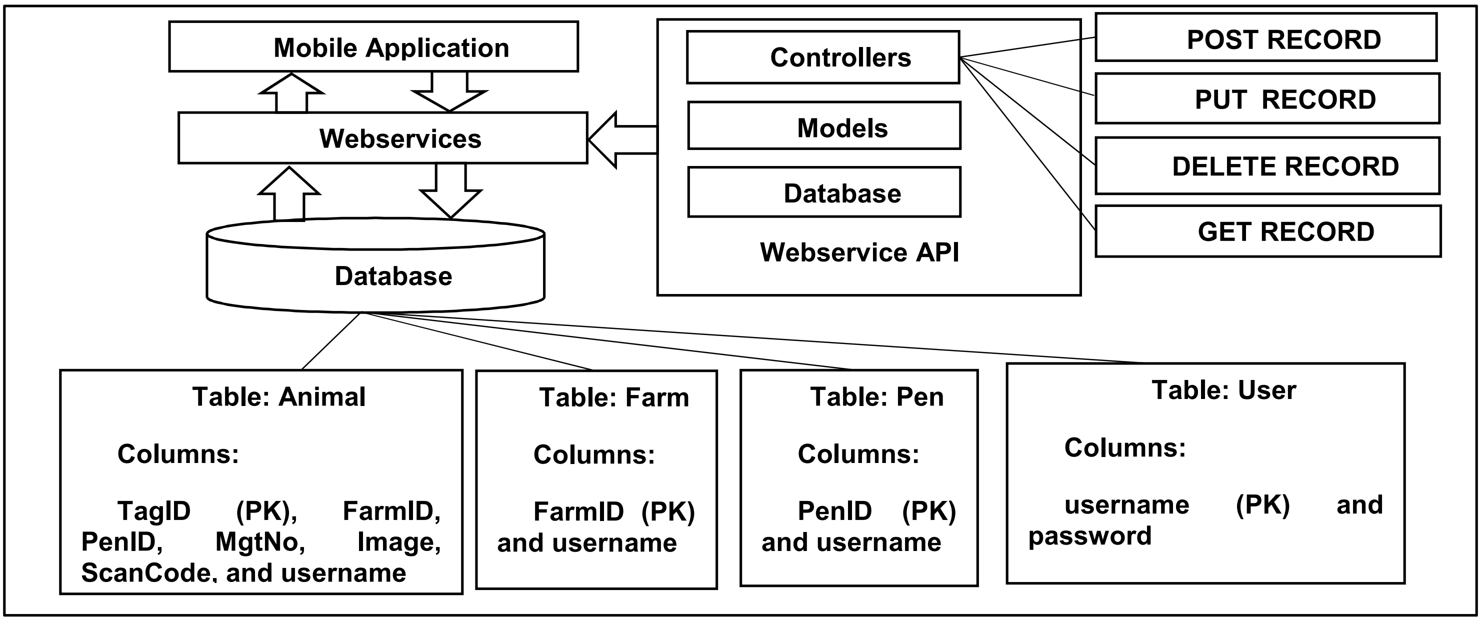

Fig. 3 Web service and database representations

B. System Requirement and Implementation

1. Programming Languages

The mobile application was developed by writing a common code (single codebase) which was later shared across the targeted platforms. The coding was done in Microsoft visual studio and using Xamarin forms as a native framework. The Xamarin forms is built on two files namely xaml and code-behind files. Coding in the xaml file was done using xaml (xml) and was responsible for setting how and where elements of the user interface appear as well as their positions. Coding in the code-behind file was done using C# programming language and was responsible for enabling and manipulating the behaviour of elements which are defined on the user interfaces of mobile applications. SQL (Structured Query Language) server is the database used in this work and the language of coding was SQL.

2. Application Packages

Microsoft visual studio and Xamarin studio were considered as development environments. Microsoft visual studio was selected because the programming was to be performed on a Windows machine. The version of the Microsoft visual studio used is Enterprise 2017. Microsoft SQL Server 2017, an express version, was installed separately on the Windows machine. The sever contains the database and tables where records of livestock are stored and manipulated. The database can be accessed via a webservice API developed. Postman (API development environment) was also installed on the Windows computer and used to test functionalities of the mobile application like records posting, updating, deleting and retrieval before the testing of the mobile application on real devices. XCODE is a tool used by mobile application developers to create mobile applications for iOS devices. The version of XCODE installed on the MAC mini is 9.4.1. XCODE provided the platform used to deploy application, which run on the Windows machine, to the real iOS devices that was connected to a MAC mini.

3. Database

SQL Server is the database adopted for this work. The database was developed such that it can be accessed via a web service. The database consists of four tables. The first table stores the tag ID, management number, location information (pen IDs and farm IDs), and contents and image of QR code of animal passports. These records of livestock can be stored, edited and deleted in the first table. Users of the mobile application are expected to select pen IDs and farm IDs, for each animal whose records are to be stored or edited. Pen IDs and farm IDs are to be pre-stored separately in the second and third tables for selection on the user-interface designed for storing records of farm animals. The fourth table stores users' login details. Representation of the database and webservice is presented in Fig. 3. The web service is an Application Programming Interface (API) set up in Microsoft visual studio. An ASP.NET Core webservice application project was created and named WebApp to serve as a database on the webserver. Tables were created on the server by setting up models (classes) for each table and their controllers. Four classes (models) were created in the web service API project. In each of the models, primary keys (PK) were defined by placing [Key] above the statement declaring the selected fields as primary key (PK).

Controllers were created for each of the models and located in the Web service API project. The controllers are what provide the codes, requirements and links to access records in the tables. The controllers, when added to the folder, were automatically generated by Microsoft visual studio. Each controller has methods, POST (for records posting), PUT (for updating records by field set as primary key in the model class), DELETE (for deleting records by field set as primary key in the model class) and GET (for getting record) data from the database via webservice as illustrated in Fig. 3. There are two get methods. One gets all the record in the table while the other get records of specific animal by the field set as primary key in the model class. Another method set up in each of the controllers was called Dispose (). Its main function was to dispose all previously opened connection to the database. A constructor was set up in each controller to establish connection with the database and tables via database context. Database context are also automatically generated whenever a controller is added. The database context provides connection to the database via models and controllers of the web service application project.

4. Animal RFID Reader and Tags

The reader and tags used for this work are those that use Radio Frequency Identification (RFID) technology for their operations. The RFID readers can read the RFID tags and connect to mobile phones via Bluetooth. The RFID Reader used in this work is presented in Fig. 1(a). It operates at low frequency (125 – 134 kHz) and has three keys and a medium screen.

5. Mobile Phones and Laptop/Personal Computer

An Android phone and iPhone are used to test functionalities of the mobile application on iOS and Android platforms respectively. A laptop (local machine) running on Windows 10 was used to test the mobile application on universal Windows platform. A laptop running on Windows was used as a machine for development. A MAC mini was used to build applications for iOS, test and debug on real iOS devices.

III. Results And Discussion

In this section, we present the results obtained when we tested the mobile platform using data provided by two users. The users signed up using the mobile application, acquired and stored records of two animals. The data utilized therefore contain the users' account details (usernames and passwords) and records of animals provided by the users which are presented in Table I to II. We used various features and functionalities of the developed mobile platform to manage and exploit records of animals stored by the users and present the results obtained in Fig. 4 to Fig. 11.

TABLE I. List of Farm and Pen IDs Stored by the Users

Users	Pen IDs	Farm IDs
First	333	222
Second	130	159

TABLE II. Set of Records Stored by One of the Users

S/N	Records	Input
1	Tag ID	969000000505907
2	Pen ID	333
3	Farm ID	222
4	Management number	200
5	Content of QR code	20001
6	Image of QR code	

Firstly, the developed cross-platform mobile application are used to capture image and acquire content of QR code on iOS, Android and Windows platforms for storage in the database. An Interface displaying the captured image and extracted content on iOS platform is shown in Fig. 4. Records of livestock are then acquired, posted, and updated on the database from the installed mobile applications on all targeted platforms via the web service. Screenshots of interfaces created to acquire records of new livestock before and after record acquisition on a Windows platform are also taken and presented in Fig. 5 and Fig. 6 respectively. Records of livestock acquired and stored in the database are viewed by supplying their IDs or by simply scanning the QR code of animals' passports. Screenshots of interfaces created for viewing records of all animals and profile of specific animals taken before and after viewing of records on Android and iOS platforms are obtained and presented in Fig. 7, Fig. 8 and Fig. 9.

Records of livestock acquired and stored in the database are exported via email and SMS. A screenshot of an interface that allows users to export records of livestock from the installed mobile application on an Android platform using the default email application is taken and presented in Fig. 10. By clicking the send buttons on the interfaces, the default email or messaging application with records of livestock, recipients and subject (in case of an email) loaded can be opened as illustrated in Fig. 11. In addition to the aforementioned features, a drop down menu bar is created at the top right corner of each interface to allow users navigate between menus. A login system to sign users in and out of the system on all the targeted platforms are implemented. Interfaces to pre-store and delete farm and pen IDs are also created.

A less functional Radio Frequency Identification (RFID) reader has been used to scan RFID tags for transmission to the mobile phone with the installed application. The developed mobile application has the capabilities to open the Bluetooth settings for connection with the RFID readers. IDs and other information (captured image and content of QR code, management number, pen and farm IDs) about livestock are successfully acquired, managed and exploited using the mobile platform.

It has been shown that with the mobile application installed, commercial farmers can use less functional RFID readers to effectively acquire, manage and exploit records of livestock and need not worry about the high cost of sophisticated RFID readers in the market. Also, the database has been accessed via the web service created to ensure that livestock keepers can log in on any mobile device with the installed application which also ensures access to the records whether livestock keepers are on-farm or off-farm. Since the developed mobile platform incorporates a mobile application that targets Android, Windows and iOS platforms, livestock keepers can monitor their livestock irrespective of the platforms on which their mobile devices are operating. While the programming was performed in Visual studio on a Windows machine, it should be noted that, as required by Apple, the Windows machine was connected to a MAC with XCODE installed to allow deployment and testing of mobile application on iOS platforms. This could limit the capacity of Xamarin forms as frameworks that targets not just Android and Windows platform but also iOS platform when MAC is out of reach.

Fig. 4 An interface displaying the QR code's image and content acquired on the iOS platform

Fig. 5 An interface for records acquisition on the Windows platform

Fig. 6 An interface for record acquisition after records have been acquired

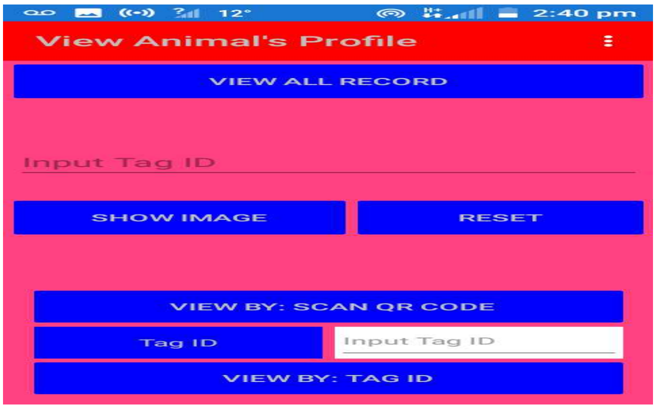

Fig. 7 An interface for viewing livestock's records on the Android platform

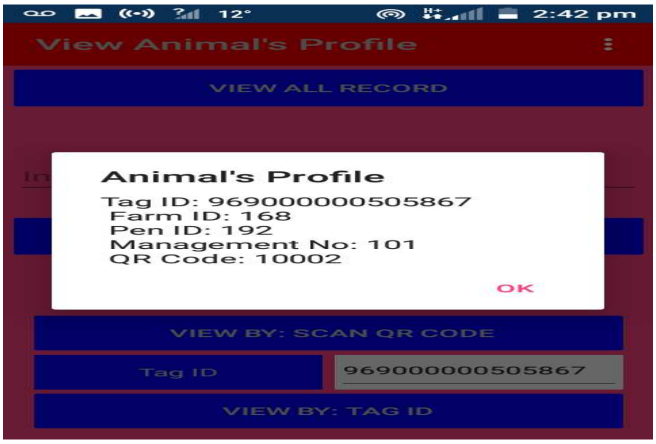

Fig. 8 A profile of an animal as viewed on the Android platform

Fig. 9 Records of all animals as viewed on the iOS platform

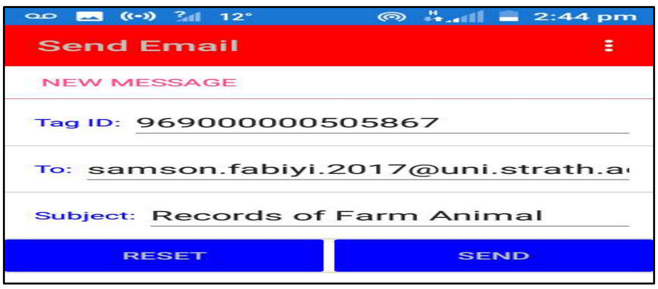

Fig. 10 An interface for exporting records of livestock via email on the Android platform

Fig. 11 Records of livestock loaded via a default email application on an iOS platform

RFID readers can be developed using microcontrollers, and RFID sensors and modules. Sensors that can measure heartbeat, blood pressure and weights can be integrated into such RFID-based systems for monitoring of livestock's health. Future solutions should consider enhancing the reading reach and memory capacity of the RFID reader cost-effectively. Also, strong measures to enable worldwide access of mobile applications, essential security mechanisms are needed to put strong anti-hacking measures in place to further secure records in the database on the server. Image processing techniques can be used to extract and match features of livestock to their tag IDs. This will, in addition to identifying and keeping track of livestock, facilitate the tracking of the livestock's movement and whereabouts.

IV. Conclusion

A mobile platform has been developed to identify and keep track of livestock and match farm animals with their IDs cost-effectively. The mobile platform consists of a mobile application, database and web service. Developed mobile application is cross-platform and targets iOS, Android and Windows platforms. Records of livestock received from a radio frequency identification reader were successfully managed and exploited by the mobile application via a web service. Benefits offer by the implemented solution include, exporting of animals' records via email and SMS, viewing of animal's record by their IDs or by simply scanning the QR code of animals' passports, and a login system to sign users in and out of the mobile application. Development of radio frequency identification readers which will incorporate sensors to measure heartbeat, blood pressure and weights for monitoring of livestock's health is recommended.

Acknowledgment

The authors deeply appreciate the financial support from Harbro and Innovent Technology and for motivating this work. Great thanks to extra colleagues Dr He Sun and Dr Julius Tschannerl for their advice.

References

[1] S. Laokok and U. Ketprom, "Web services with poultry traceability system," in 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2008, 2008, vol. 1, pp. 197–200.

[2] A. R. Frost, C. P. Schofield, S. A. Beaulah, T. T. Mottram, J. A. Lines, and C. M. Wathes, "A review of livestock monitoring and the need for integrated systems," Comput. Electron. Agric., vol. 17, no. 2, pp. 139–159, 1997.

[3] M. H. Ariff, I. Ismarani, and N. Shamsuddin, "RFID based systematic livestock health management system," in Proceedings - 2014 IEEE Conference on System, Process and Control, ICSPC 2014, 2014, pp. 111–116.

[4] Q. Zhu, J. Ren, D. Barclay, S. McCormack, and W. Thomson, "2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing.," 2015.

[5] H. Sun, Q. Zhu, J. Ren, D. Barclay, and W. Thomson, "Combining Image Analysis and Smart Data Mining for Precision Agriculture in Livestock Farming," in Proceedings - 2017 IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, IEEE Cyber, Physical and Social Computing, IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData 2017, 2018, vol. 2018-January, pp. 1065–1069.

[6] M. Xu et al., "Stylized Aesthetic QR Code," IEEE Trans. Multimed., vol. 21, no. 8, pp. 1960–1970, Aug. 2019.

[7] L. Li, M. Fan, and G. Wang, "LWSQR: Lightweight secure QR code," Commun. Comput. Inf. Sci., vol. 879, pp. 241–255, 2018.

[8] S. Tiwari, "An introduction to QR code technology," in Proceedings - 2016 15th International Conference on Information Technology, ICIT 2016, 2017, pp. 39–44.

[9] I. Tkachenko, W. Puech, O. Strauss, C. Destruel, J. M. Gaudin, and C. Guichard, "Rich QR code for multimedia management applications," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9280, pp. 383–393, 2015.

[10] K Y Chetanraj; K S Shushrutha; A R Phani; C S Naveen, "The design of active RFID for food quality and safety sensors," in 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon), 2017.

[11] M. Konieczny, B. Pawłowicz, J. Potencki, and M. Skoczylas, "Application of RFID Technology in Navigation of Mobile Robot," in 2017 21st European Microelectronics and Packaging Conference (EMPC) & Exhibition, 2017.

[12] M. G. Aboelmaged, "Enablers and Impediments of RFID Implementation in Health Service Operations," in 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2017, pp. 213–219.

[13] Tarushi Wasson, Tanupriya Choudhury, Shilpi Sharma, and Praveen Kumar, "Integration of Rfid And Sensor In Agriculture Using Iot," in 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon), 2017.

[14] S. S. Park, "An IoT application service using mobile RFID technology," in International Conference on Electronics, Information and Communication, ICEIC 2018, 2018, vol. 2018-January, pp. 1–4.

[15] R. E. Floyd, "RFID in Animal-Tracking Applications," IEEE Potentials, vol. 34, no. 5, pp. 32–33, Sep. 2015.

[16] C. Zhang and X. Yin, "Design and implementation of single-service multi-function Webservice," in 2011 International Conference on Computer Science and Service System, CSSS 2011 - Proceedings, 2011, pp. 3912–3915.

[17] Y. Tan, X. Li, Z. Yang, and M. Huang, "The design and preliminary realized of DotTopModel based on WebService under B/S architecture," in 2011 International Conference on Information Science and Technology, ICIST 2011, 2011, pp. 352–356.

[18] J. Bian, Y. Cheng, and J. Kuang, "Research on webservice technology based on oms platform," in 2010 International Conference on Advanced Intelligence and Awarenss Internet (AIAI 2010), 2010.

[19] N. Boushehrinejadmoradi, V. Ganapathy, S. Nagarakatte, and L. Iftode, "Testing cross-platform mobile app development frameworks," in Proceedings - 2015 30th IEEE/ACM International Conference on Automated Software Engineering, ASE 2015, 2016, pp. 441–451.

[20] W. S. El-Kassas, B. A. Abdullah, A. H. Yousef, and A. M. Wahba, "Enhanced Code Conversion Approach for the Integrated Cross-Platform Mobile Development (ICPMD)," IEEE Trans. Softw. Eng., vol. 42, no. 11, pp. 1036–1053, Nov. 2016.

[21] S.D. Fabiyi, "Mobile Platform for Intelligent Livestock Monitoring and Inspection," MSc Thesis, University of Strathclyde, Glasgow, UK, 2018.