Development of an Android Mobile App for Real Time Maize Stem Borers Monitoring in Precision Farming

Ezeofor Chukwunazo Joseph
Department of Electrical/Electronic Engineering
University of Port Harcourt
Choba, Rivers State, Nigeria
Email: chukwunazo.ezeofor [AT] uniport.edu.ng

Abstract— **Development of an Android mobile app for real time maize stem borers' monitoring in precision agriculture is presented. In farmland, cultivated maize requires farmers' constant care and monitoring during the developing stage to avoid sudden attack of insect pests such as stem borers in the field. The maize monitoring process taken by farmers to ensure attack free and healthy growth is very strenuous and time consuming. The sudden invasion of the *Spodoptera* species (stem borers) to maize farm early 2016 caused huge loss to farmers and imposed food scarcity in the land. These species are hardly distinguished from one another by farmers in the farm because they look alike in appearance. Rural farmers do not know the right insecticides to apply for the effective control of these species. These issues kept on lingering and now have become serious concern to farmers. Hence, this work is to bridge the gap by providing android mobile app that would enable farmers to effectively monitor these species remotely. The mobile app architecture consists of various sections such as captured insects, categories of *spodoptera* species, insect pest population plots, determination of economic injury level (EIL) and economic threshold (ET), and control measure was successfully designed. The mobile app structure and behavior were also designed using Unified Modeling Language (UML). The maize Stem borers App was developed in android studio using Kotlin programming language. The App is linked to the cloud server where all the captured and recognized species are stored for downloading and farmers' visualization. The Internet of Things (IoT) hardware was setup in the maize farm which captured these targeted insect pests, processed via Nividia Jetson Nano and sent to the cloud server. The mobile App synchronized successfully with the cloud server and could download stored maize insect pests in the farmer's Android phone.**

Keywords- ***Android Studio, Kotlin, Maize Stemborers, Spodoptera Species***

I. INTRODUCTION

Stem borers (Lepidopteran) are considered as groups of insect pests that attack cereal crops such as maize, sorghum, rice and sugarcane in many parts of the world. These stem borers such as *Busseola fusca* (Fuller), *Chilo partellus*, Fall Armyworm (FAW), African Armyworm; Egyptain cotton leafworm etc. in developing countries like Africa feed aggressively on leaves, stems and ears of maize and significantly reducing the crops yield. The Fall Armyworm (*Spodoptera frugiperda),* African armyworm (*Spodoptera exempta)* and Egyptian cotton leafworm (*Spodoptera* littoralis) are more aggressive and invade maize farms like wildfire [1]. These invasive alien insect pests FAW were reported in Africa, Nigeria in early January 2016, and currently in more than 20 African countries. It has destroyed many staple crops like maize thereby becoming a threat to food security across the region. Damage to maize alone by this pest could amount to total of USD$3 billion in a year. Crop losses in African countries due to these insect pests are estimated at 49% of the expected total crop yield each year, according to the Centre for Agriculture and Biosciences International [2], But some crop losses can be even worse, and the effects of the changing climate are expected to increase the damage done by these insect pests [3]. Mobile application for maize Stem borers' management is an application that would aid farmers visualizes the presence of aggressive maize moths in the farm. The app would enable the farmers apply right control measures to reduce the population of the aggressive maize insect pests in the farm for optimal crop yield. The maize stem borers' app is built using Android Studio (Integrated Development Environment IDE) and programmed using Kotlin programming language for **Android** mobile applications.

II. REVIEW OF RELATED WORKS

In [4], Christine *et al* discussed on Mobile communications technology that has quickly become the world's most common way of transmitting voice, data, and services in the developing world. Given this dramatic change, mobile applications (m-apps) in general and mobile applications for agricultural and rural development (m-ARD apps) in particular hold significant potential for advancing development. They could provide the most affordable ways for millions of people to access information, markets, finance, and governance systems previously unavailable to them.

In [5], Saha *et al* developed a satisfactorily tested m-Sahayak Innovative Android based Application for Real-time Assistance in Indian Agriculture and Health Sectors. The application takes care of certain problems in agriculture and health care by concurrently capturing images, audio and video and sending them to a specified server. Agricultural Scientists or Doctors can view or listen to this images/video/audio information and provide proper solutions, accordingly. In [6], Meena *et al* discussed mobile apps in the agricultural domain, considerations and challenges for successful mobile app development and focused on a cross-platform mobile development approach adopted by University of Florida/Institute of Food and Agricultural Sciences (UF/IFAS). In [7], Sotiris *et al* reviewed a smartphone app use and capabilities in agriculture for m-government. In [8], Ankur *et al* developed android application on Orchid farming and hosted on Google play store as well as on the official website of ICAR-National Research Centre for Orchids for focusing orchid growers, entrepreneurs and stakeholders. This application provides vital information regarding the management of different types of commercial orchids with a suitable environment. It supports Android smartphone with minimum version 4.1 and above that covers 96.4% android users. In the current era, 90.67% Android Operating system works in India. 'Orchid Farming' mobile app covers four genera and one hybrid viz, Cymbidium, Dendrobium, Phalaenopsis, Vanda & Mokara.

In [9], Priti *et al* proposed an automated method for real time crop monitoring and disease detection. Images were captured on a daily basis for a field of 8 acres of land. Image features were extracted using the Speeded-Up Robust Features (SURF) after the Maximally Stable External Regions (MSER) method find blobs. The features were used to classify the images using Kmeans Clustering in the training phase. The ground truth diseased crop images were stored in a database with the same features to act as prototypes and were compared to real time images for disease detection using nearest neighbor classification. The experimental dataset currently consists of rice crop and maize crop with 100 diseased images and approximately 1000 normal crop images. Results showed 83.3% accuracy and provided information to farmers about their crop and if required, alert them of the disease presence to take corrective action.

III. MAIZE APP DESIGN METHODOLOGY

The android mobile app adopts agile methodology approach [10] as shown in fig.1. The approach is a practice that promotes continuous iteration of development and testing throughout the software development life cycle. This iterative and flexible approach can be used in complex apps where the customer requirements change frequently. A big software project can be broken down into smaller parts and agile methodology can be applied to each of these small parts [11].

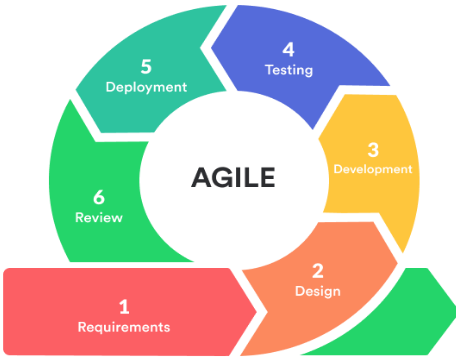


Fig.1: Agile Methodology for Mobile App Development [10]

A. *Requirements*

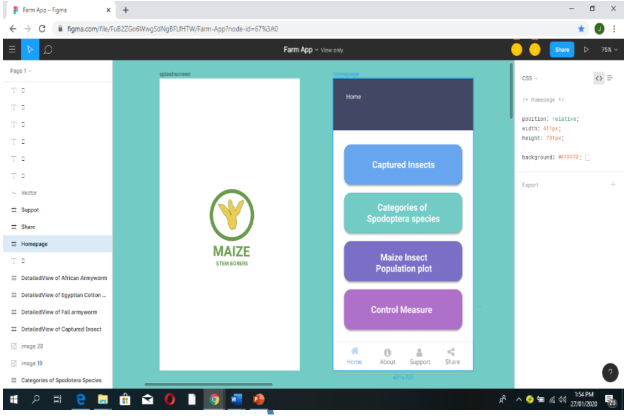
In order to design and develop the maize Stem borer mobile app, high speed internet laptop, Cloud-based design tool (Figma), knowledge of Kotlin programming language (for multiple android mobile Operating systems platform) and App hosting are required which were used.

B. *The Mobile App Design*

The android mobile app mock-up was designed using Figma, one of the cloud-based user interface (UI) design tools. The researcher created an account in figma.com and used it to design each of the modules based on the functions required.

1) *Mobile App Splash & Home Page:* The maize logo was the first to be designed such that when tapped, the app stays for a second before opening to the home platform where user can initiate different requests.

2) *Home page:* this was designed to include captured insects' button, categories of *Spodoptera* species button, maize insects' population plots button, and control measure button while about, support and share options at the lower task bar of the app were added to provide user detailed information about the app, how to share and get help from entomologist. The designed pages are shown in fig. 2. Each of the buttons can be click once to navigate into the next page.

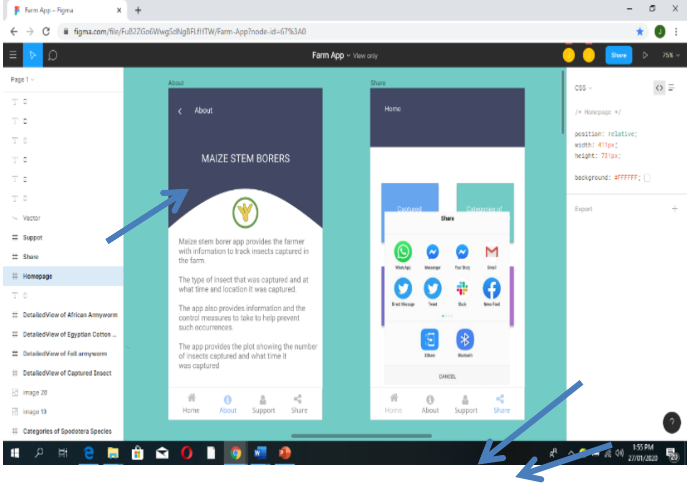

Farm App – Figma
figma.com/file/FuB2ZGo6WwgSdNgBFLfHTW/Farm-App?node-id=67%3A0
Farm App View only
Share
75%
Page 1
Vector
Suppot
Share
Homepage
DetailedView of African Armyworm
DetailedView of Egyptian Cotton ...
DetailedView of Fall armyworm
DetailedView of Captured Insect
image 20
image 19
Categories of Spodotera Species
splashscreen
MAIZE
STEM BORERS
Homepage
Home
Captured Insects
Categories of Spodoptera species
Maize Insect Population plot
Control Measure
Home
About
Support
Share
411×731
CSS
Export
1:54 PM
27/01/2020

```
/* Homepage */

position: relative;
width: 411px;
height: 731px;

background: #FFFFFF;
```

Fig. 2: The App start-up logo and home pages designed in Figma

3) *About option:* This platform helps the farmers to know more about the stem borer app. The information can be gotten by tapping on the about option button (fig. 3).

4) *Share:* This platform helps the farmers to share information gotten from cloud server about the species to friends in social media through WhatsApp, messenger, tweet, Gmail etc. as shown in fig. 3.



Maize stem borer app provides the farmer with information to track insects captured in the farm.

The type of insect that was captured and at what time and location it was captured.

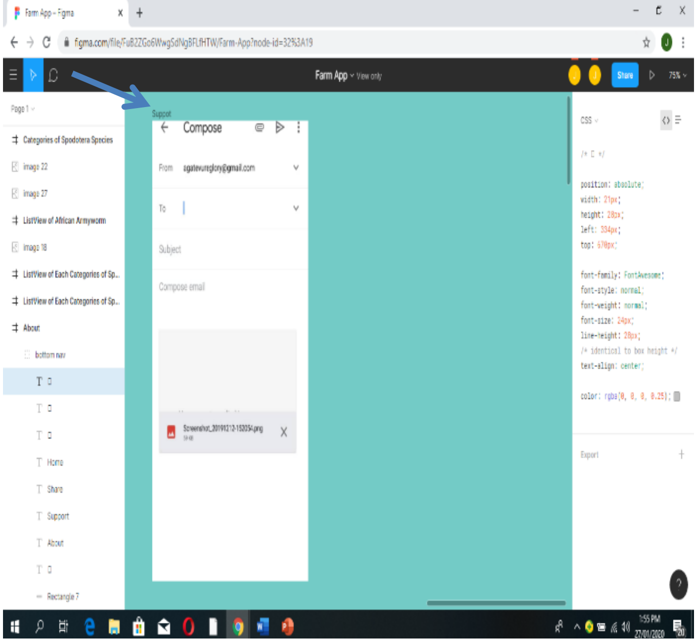
The app also provides information and the control measures to take to help prevent such occurrences.

The app provides the plot showing the number of insects captured and what time it was captured

```
/* Homepage */

position: relative;
width: 411px;
height: 731px;

background: #FFFFFF;
```

Fig. 3: About and Share options designed in Figma web app

5) *Support:* This platform (fig.4) is designed to support farmers to communicate with the experts (entomologist) via emails for advice and latest update.

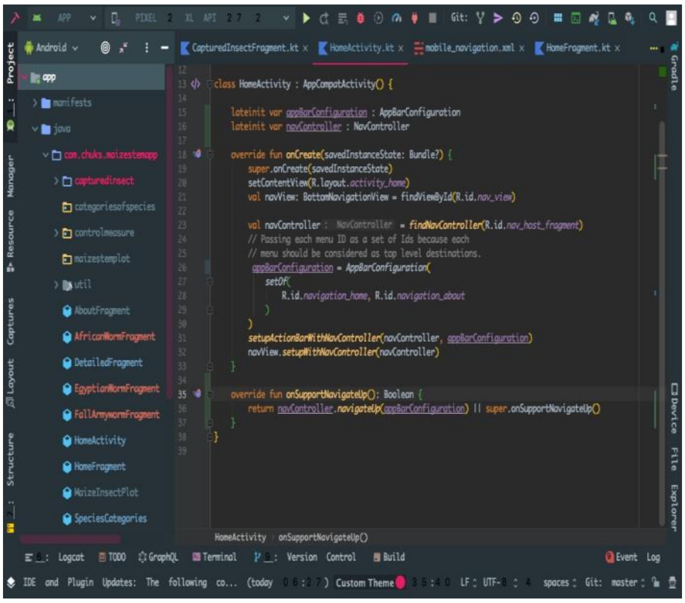

Farm App – Figma
figma.com/file/FuB2ZGo6WwgSdNgBFLfHTW/Farm-App?node-id=32%3A19
Farm App · View only
Share
75%
Page 1
Categories of Spodoptera Species
image 22
image 27
ListView of African Armyworm
image 18
ListView of Each Categories of Sp...
ListView of Each Categories of Sp...
About
bottom nav
Home
Share
Support
About
Rectangle 7
Support
Compose
From
To
Subject
Compose email
Export
1:55 PM
27/01/2020

```
CSS

/* □ */

position: absolute;
width: 21px;
height: 28px;
left: 334px;
top: 670px;

font-family: FontAwesome;
font-style: normal;
font-weight: normal;
font-size: 24px;
line-height: 28px;
/* identical to box height */
text-align: center;

color: rgba(0, 0, 0, 0.25);
```

Fig. 4: Support platform designed in Figma

C. *Integrated Development Environment for Maize App Development*

The Android Studio Integrated Development Environment (IDE) shown in fig.5 was used to develop the mobile app since the app would be installed in an Android phones, Personal Digital Assistant (PDA) etc. The user interfaces (UI) were designed first and finally coded. Kotlin programming language was used to write the code that makes each designed button to function accurately. Kotlin is a statically-typed programming language, created by JetBrains Company. It provides full compatibility with Java and runs on the Java Virtual Machine (JVM) which enables the developer to write less code, has readable syntax, and compile to JavaScript source code [12].



```
class HomeActivity : AppCompatActivity() {

    lateinit var appBarConfiguration : AppBarConfiguration
    lateinit var navController : NavController

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_home)
        val navView: BottomNavigationView = findViewById(R.id.nav_view)

        val navController : NavController = findNavController(R.id.nav_host_fragment)
        // Passing each menu ID as a set of Ids because each
        // menu should be considered as top level destinations.
         appBarConfiguration = AppBarConfiguration(
            setOf(
                R.id.navigation_home, R.id.navigation_about
            )
        )
        setupActionBarWithNavController(navController, appBarConfiguration)
        navView.setupWithNavController(navController)
    }

    override fun onSupportNavigateUp(): Boolean {
        return navController.navigateUp(appBarConfiguration) || super.onSupportNavigateUp()
    }
}
```

Fig. 5: Android Studio IDE with Kotlin Programming language

IV APP MODELLING STRUCTURE AND BEHAVIOUR

The unified modeling language (UML) as in [13] is used to describe the structure and behavior of the Android mobile app (maize Stem borer). It showed how the app is structured by pointing out various elements of the system such as classes, objects, packages, components etc. and relationships that exist between those elements. The UML diagrams were designed using online tool called lucidchart.com. Very few important designed structures and behavior diagrams are shown below.

A. *Function Diagram*

The function diagram shows various functional and sub-functional parts of the app as shown in fig. 6. The forward arrow is used to indicate the flow moving from one module to another.

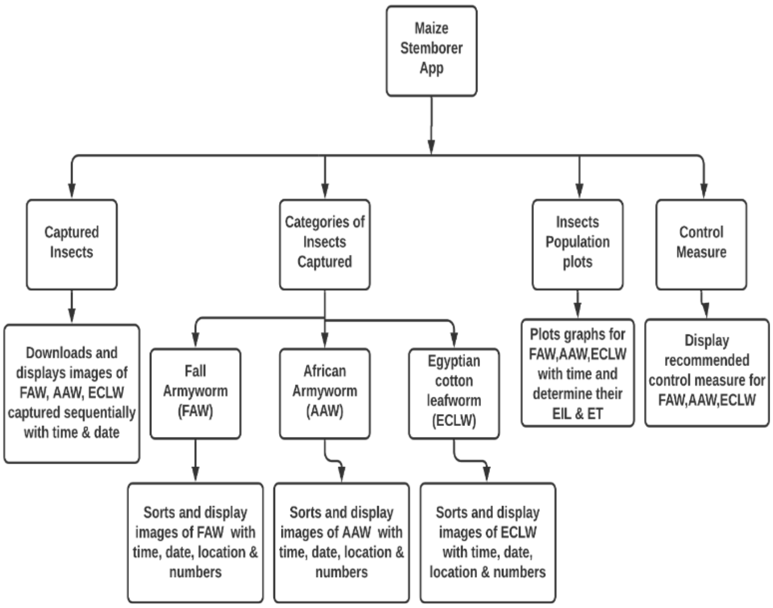

Maize Stemborer App
Captured Insects
Categories of Insects Captured
Insects Population plots
Control Measure
Downloads and displays images of FAW, AAW, ECLW captured sequentially with time & date
Fall Armyworm (FAW)
African Armyworm (AAW)
Egyptian cotton leafworm (ECLW)
Plots graphs for FAW,AAW,ECLW with time and determine their EIL & ET
Display recommended control measure for FAW,AAW,ECLW
Sorts and display images of FAW with time, date, location & numbers
Sorts and display images of AAW with time, date, location & numbers
Sorts and display images of ECLW with time, date, location & numbers

Fig. 6: Function diagram of the Mobile App

B. UML Class Diagram

The class diagram represents the static view of the app system. It is used for describing, visualization and documentation and has class, object, attributes and operation/method as showcased in fig. 7. The UML class diagram consists of four classes namely captured insects, categories of *Spodoptera* species, maize insect density population plots, and control measure with three subclasses and their respective attributes and functions for effective operation of the app. The association, aggregation, and extended relationship are used to initiate flow in the system.

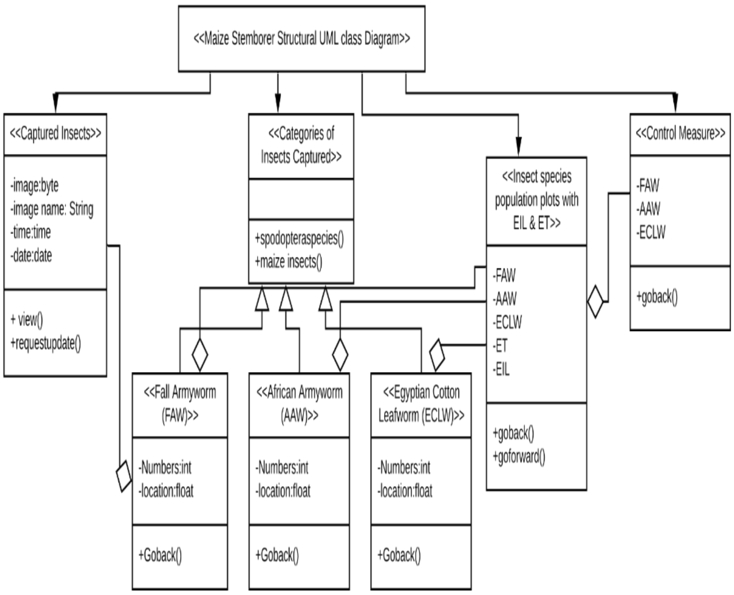


Fig. 7: UML class diagram for the Mobile App

C. UML Use Case Diagram

Use case diagram is used to model the maize stem borer app system/subsystem for Android mobile application. This showcased the system, actors, use cases and relationships in the developed application as shown in fig. 8. The farmer that would use the app to visualize the species from the cloud server is an actor. The cloud server is also an actor because it stores all the detected, captured and recognized insects, then releases the insects on request from the farmer's mobile app. The system consists of all the use cases and relationships such as association, include, and exclude which were used to link the elements.

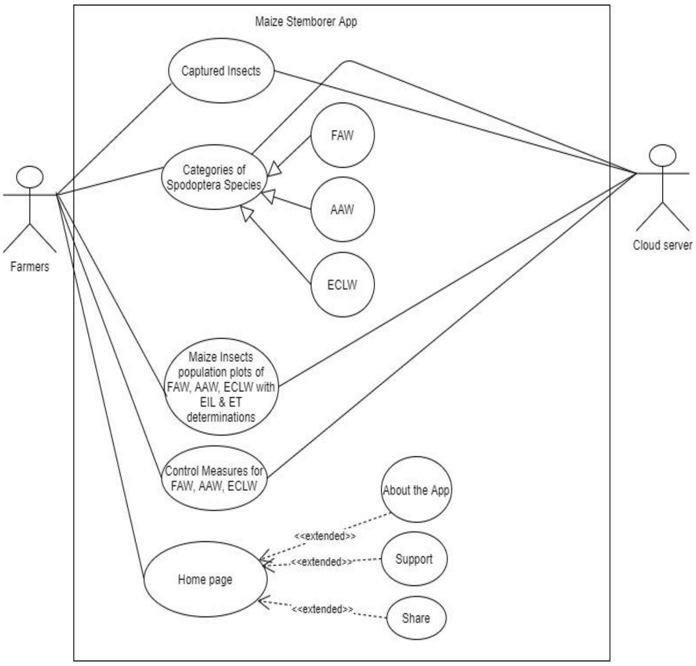

Maize Stemborer App
Captured Insects
FAW
Categories of Spodoptera Species
AAW
ECLW
Maize Insects population plots of FAW, AAW, ECLW with EIL & ET determinations
Control Measures for FAW, AAW, ECLW
About the App
<<extended>>
Support
<<extended>>
Home page
<<extended>>
Share
Farmers
Cloud server

Fig. 8: UML use case diagram for Maize Stem borer App

D. Sequence Diagram

Sequence diagram shows the order in which the objects interact. The objects are the phone and cloud server. The farmer (an actor) initiates the commands from the phone buttons which are sent to the cloud server for processing and response as shown in fig. 9. The cloud server responds to the farmer as soon as possible by querying its database and sends the request.

E. Maize App Flow Chart

The flow chart (fig. 10) states the step-by-step process used by farmer to download the captured insects from the Amazon cloud server, visualize them and get the recommended control measure. The farmer starts by connecting mobile phone to internet and tapping on the app icon to open. When the app opens, the app is synchronized to the cloud server. Then, farmer can tap any option on the home page platform for more detailed view.

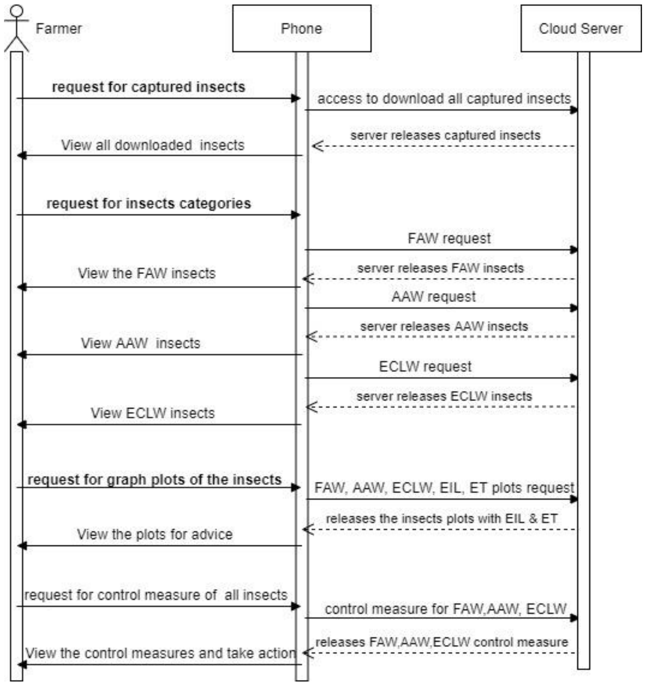

Farmer
Phone
Cloud Server
request for captured insects
access to download all captured insects
server releases captured insects
View all downloaded insects
request for insects categories
FAW request
server releases FAW insects
View the FAW insects
AAW request
server releases AAW insects
View AAW insects
ECLW request
server releases ECLW insects
View ECLW insects
request for graph plots of the insects
FAW, AAW, ECLW, EIL, ET plots request
releases the insects plots with EIL & ET
View the plots for advice
request for control measure of all insects
control measure for FAW,AAW, ECLW
releases FAW,AAW,ECLW control measure
View the control measures and take action

Fig. 9: Sequence diagram of farmer interactions with objects

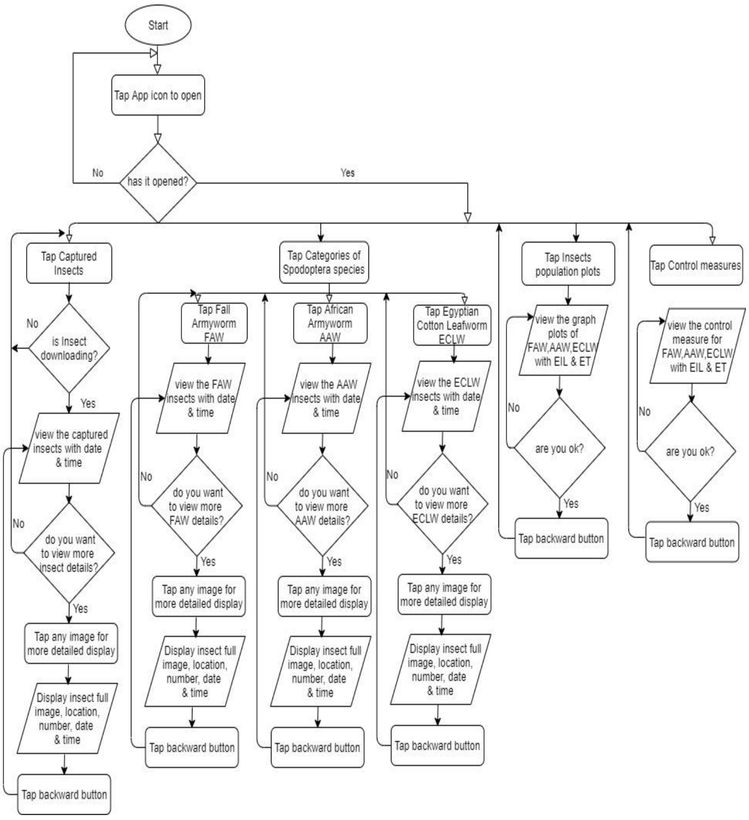

Start
Tap App icon to open
has it opened?
No
Yes
Tap Captured Insects
is Insect downloading?
view the captured insects with date & time
do you want to view more insect details?
Tap any image for more detailed display
Display insect full image, location, number, date & time
Tap backward button
Tap Categories of Spodoptera species
Tap Fall Armyworm FAW
view the FAW insects with date & time
do you want to view more FAW details?
Tap African Armyworm AAW
view the AAW insects with date & time
do you want to view more AAW details?
Tap Egyptian Cotton Leafworm ECLW
view the ECLW insects with date & time
do you want to view more ECLW details?
Tap Insects population plots
view the graph plots of FAW,AAW,ECLW with EIL & ET
are you ok?
Tap Control measures
view the control measure for FAW,AAW,ECLW with EIL & ET

Fig. 10: Maize stem borer app Flow Chart (designed using online tool-draw.io)

F. *Insect Pest Detection and Capturing*

The Internet of Things (IoT) hardware system for sensing, detecting and capturing insect pest image was setup in the maize farm. The IoT system is equipped with sensors (Camera, GPS etc.) that enable it to sense, detect and capture the maize moths. Once the sensor senses the presence of the insects, it sends signals to the control unit which triggers the camera to take the image of the insect pest. The captured image is further processed by raspberry pi through written python scripts to detect the targeted maize moths. Once that is done, the image of the insect pest is sent to the Amazon cloud for image recognition and classification using trained model. The developed IoT architecture is shown in fig. 11a and fig.11b.

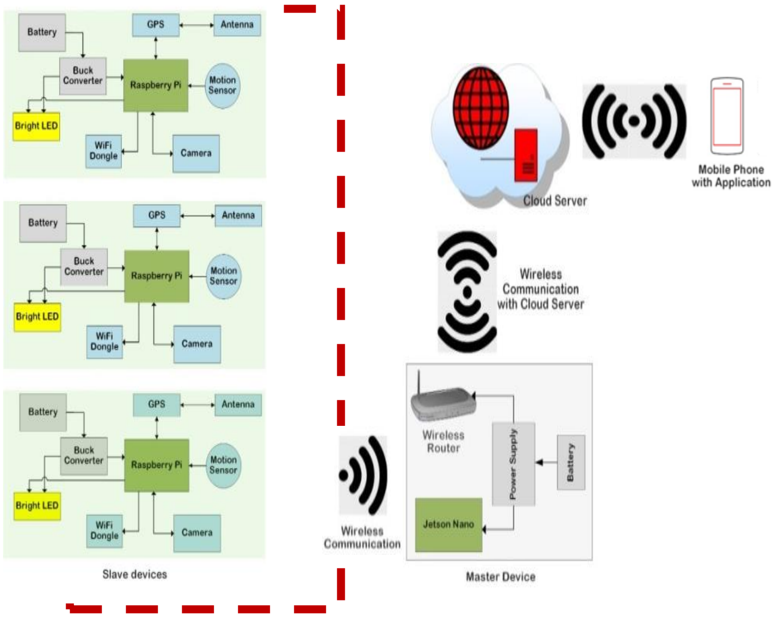


Fig. 11 on and capturing and

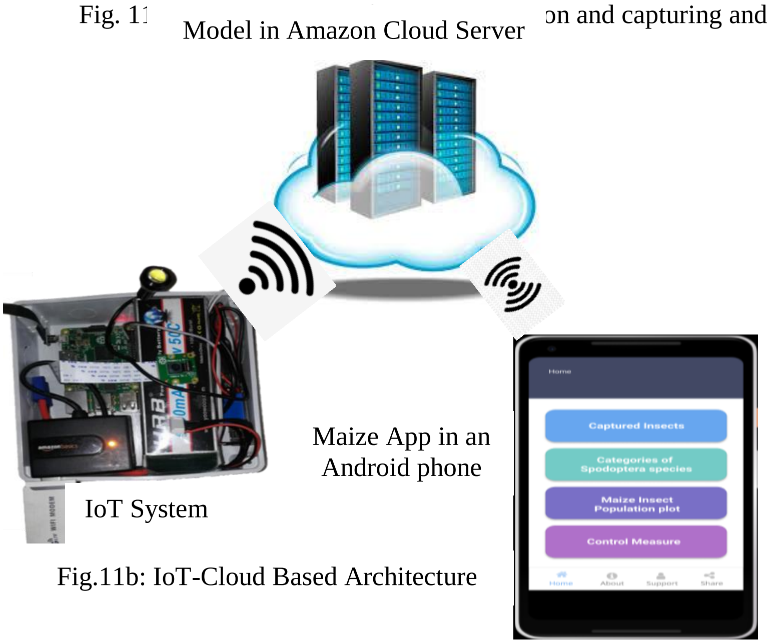

Model in Amazon Cloud Server
Maize App in an Android phone
IoT System
Home
Captured Insects
Categories of Spodoptera species
Maize Insect Population plot
Control Measure
Home
About
Support
Share

Fig.11b: IoT-Cloud Based Architecture

V. MAIZE APP TEST RESULTS OBTAINED

Maize App Installation: Farmer is expected to install this App in his/her Android phone. After installation, the farmer should tap the icon once to boot-up the app in a second. During boot-up, fig.12 screen would appear.

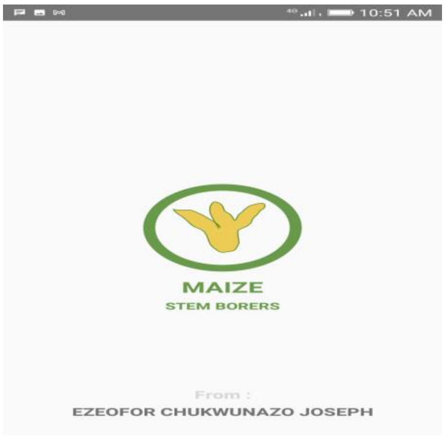


Fig. 12: Maize App login logo

A. The home page:

The home page (fig. 13) would enable the farmer to view the captured insects, check detailed categories of *Spodoptera* species captured, maize insects density population plots (showcasing the determined Economic Injury Level (EIL) and Economic Threshold (ET)), and recommended control measure. Also, in the home page platform, support, share and about options were added to aid farmers seek for support through emails, share the insects information to friends and read about the app and its content.

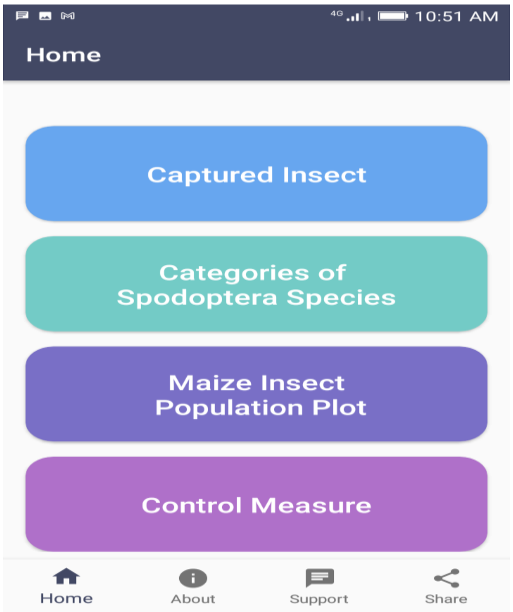


Fig. 13: Designed home page of the Maize App

B. Captured Insects Display

When captured insects' button is tapped, fig. 14a shown would appear first and downloads all the stored insect pests from the Amazon cloud server. To further view the full image of any downloaded insect, tap on each downloaded insect to show details as shown in fig. 14b **(**including information about the insect such as insect name, captured date, location , and number of count).

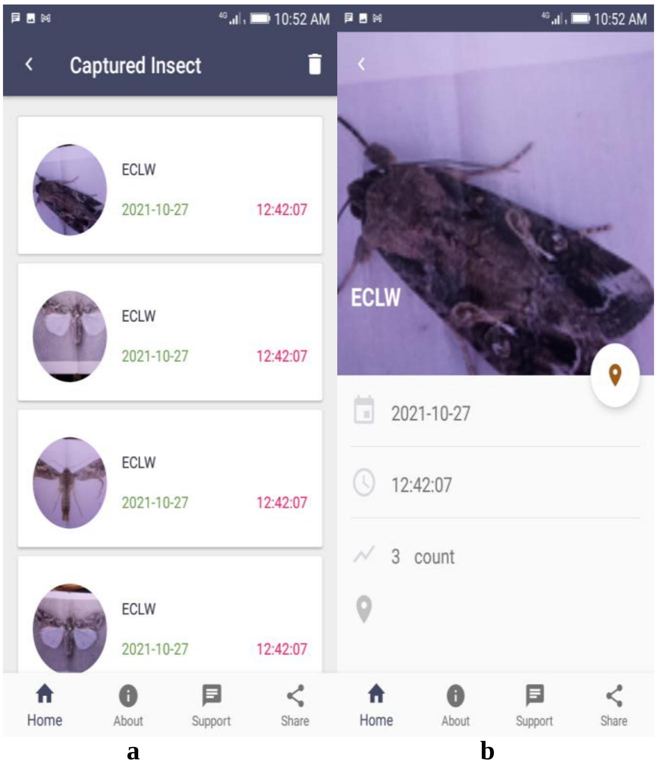


Fig. 14: captured insects display pattern

C. Categories of Spodoptera Species

This option enables the farmer to view the three categories of *Spodoptera* species captured instead of mixing all the insects together. When the button is tapped, the categories which include Fall Armyworm-FAW (*Spodoptera frugiperda*), African armyworm-AAW (*Spodoptera exempta)* and Egyptian cotton leafworm-ECLW (*Spodoptera* littorallis) are shown as shown in fig. 15. Each category can be viewed by tapping on the option for more display. For example, tapping on FAW option would display all the FAW insects recognized and stored in the Amazon cloud server. Tapping on AAW option would display all the recognized AAW insects recognized and stored in the cloud server. Finally, tapping on ECLW option would also display all the ECLW insects recognized and stored in the cloud server including the time captured, date captured, the location, number of insects' recognized, full image and name.

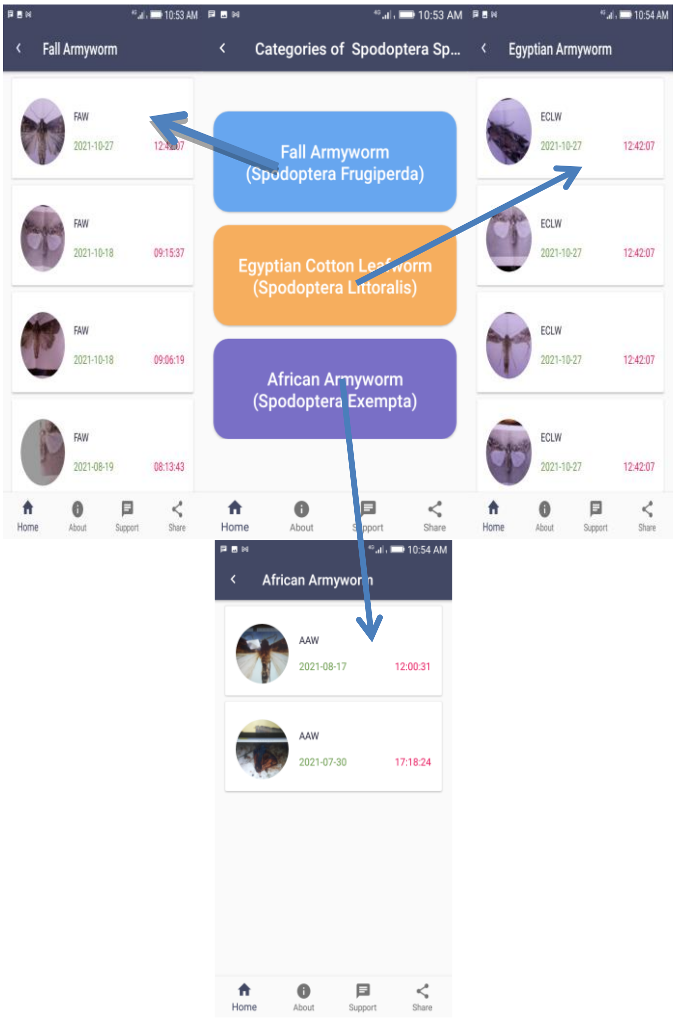


Fig. 15: Three categories of the species

D. *Insect Population plots*

The farmer tap this button to view the total number of insects that infiltrated the maize farm through graph plots of FAW, AAW, and ECLW (fig. 16) together with their respective determined Economic Injury Level (EIL) and Economic Threshold (ET). Colours were used to differentiate FAW, AAW, ECLW, EIL and ET to synchronize with the legends used. Since EIL is the smallest number of insects that would cause yield losses, the mathematical expression to determine the EIL is expressed in [14, 15]. The EIL equation is shown in equation 1.

$$\mathrm{EIL}=\frac{C}{(V\ x\ DI\ xK)} \qquad (1)$$

Where C = Pest management costs,
V = Market value of the commodity,
DI = Yield loss per pest,
K = Pest population controlled.
The ET is actually a time to take action, that is, numbers are simply an index of that time (action threshold). To be determine by an entomologist or crop management expert.

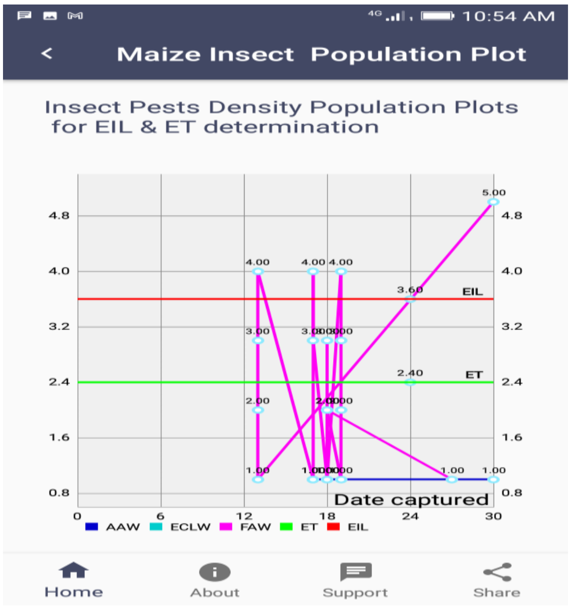
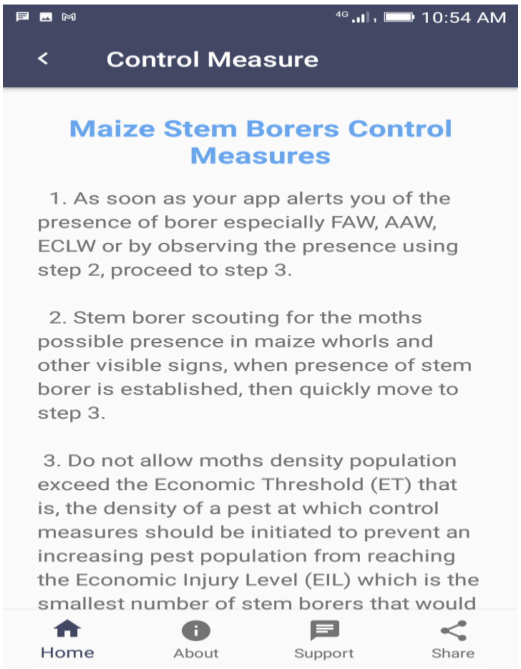


Fig. 16: Maize Stem Borers population plots showcasing EIL & ET

E. *Pest Control Measure*

This button when tapped, gives the farmer total guide on the best recommended control measure (fig. 17) to take having seen the insects' population plots to identify the number of insects' attack. The platform provides various actions to be taken to apply insecticides in order to effectively control the influx of the aforementioned insects in the maize farm.



Maize Stem Borers Control Measures

1. As soon as your app alerts you of the presence of borer especially FAW, AAW, ECLW or by observing the presence using step 2, proceed to step 3.

2. Stem borer scouting for the moths possible presence in maize whorls and other visible signs, when presence of stem borer is established, then quickly move to step 3.

3. Do not allow moths density population exceed the Economic Threshold (ET) that is, the density of a pest at which control measures should be initiated to prevent an increasing pest population from reaching the Economic Injury Level (EIL) which is the smallest number of stem borers that would

Fig. 17: Insect pests control measure

VI CONCLUSION

The development of this maize Stem borer app is timely and highly demanding in this era. Due to the serious attack recorded on daily basis by farmers in their various maize farms in Nigeria concerning the sporadic invasion of these aggressive moths. These insects had caused serious scarcity of maize in the land which was a very big threat to food security. The app platform would aid farmers to visualize the attack of the maize crop growth in the field via their android mobile phones by continuous checking of the presence of these *Spodoptera* species. The control measures to be taken are well detailed to enable farmers reduce the influx of the insect population in the farm to increase crop yield.

This app, maize Stem borer is recommended to farmers both in rural and urban for use in visualizing and monitoring the intrusion of the aggressive maize insects in the farm. For future work, this android app (maizestem) can be modified for flexibility to run in various mobile operating systems such as iphoneOS, windows, blackberry, webOS, Symbia etc.

ACKNOWLEDGMENT

I wish to acknowledge Dr. U. Zakka, an Entomologist from University of Port Harcourt, Department of Crop and Soil Science for all his initiative and support. My special thanks to Mrs Agatevure Glory for her programming ideas that facilitates the success of the developed maize App. A million thanks, to Petroleum Technology Development Fund (PTDF), for funding this research till the end. I say thank you all.

REFERENCES

[1] P. Richard, "Armyworm: The hungry caterpillar threatening a global food crisis", 2017, Available online at: www.theguardian.com/global-development-professionals network/2017/may/16/armyworms-the-hungry-caterpillar-threatening-a-global-food-crisis, accessed 12 November 2019.

[2] B. Anna-Maria, J. K. Karl, M. Joyce, and H. F. Christine, "Defining biotechnological solutions for insect control in sub-saharan African", Food and Energy Security (WILEY), pp. 1-21, 2019. Available at:www.onlinelibrary.wiley.com/doi/epdf/10.1002/fes3.191>, accessed 13 December 2019.

[3] H. C. Simon, N. Mayumbo, P. Jackson , and S. Philemon , "An Application of Machine Learning Algorithms in Automated Identification and Capturing of Fall Armyworm (FAW) Moths in the Field", Proceedings of the Information and Communication Technology Society of Zambia (ICTSZ) International Conference in ICTS (Icict2018) - Lusaka, Zambia, 2018.

[4] Z.C.Qiang, S. C. Kuek, A. Dymond, and S. Esselaar, "Mobile Applications for Agriculture and Rural Development", *ICT Sector Unit World Bank*, 2020.

[5] Biswajit S., A. Kowsar, B. Premankur, and C. Amit, "Development of m-Sahayak- the Innovative Android based Application for Real-time Assistance in Indian Agriculture and Health Sectors", The Sixth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM., pp.133-137, 2012.

[6] R.L. Meena1, B. Jirli, M. Kanwat and N.K. Meena, "Mobile Applications for Agriculture and Allied Sector",International Journal of Current Microbiology and Applied Sciences (IJCMAS), vol.7, no.2, pp. 2317-2326, 2018.

[7] K.Sotiris, C. Constantina, and S. Alexander, "Developing a smartphone app for m-government in agriculture", Journal of Agricultural Informatic, vol.5, no. 1, pp. 1-8, July 2014.

[8] T. Ankur, K. P. Ravi , L. C. De Rampal, K.S.Rakesh,and D.R. Singh "Mobile App - Android application on 'Orchid Farming" based on North Eastern States of India", Short communication, Indian J. Hort., vol.76, no.4, pp.752-756, December 2019.

[9] M. Priti, S. Mukul, M. Timothy, and K. Rakesh, "Mobile detection of crop diseases for agricultural yield management", Real-Time Image Processing and Deep Learning, SPIE Defense + Commercial Sensing, Baltimore, Maryland, United States, vol. 10996, pp.2-14, May 2019.

[10] P. Pawan, 'Insights to Agile Methodologies for Software Development', 2019, <hackernoon.com/a-case-study-type-insight-into-agile-methodologies-for-software-development-cd5932c6>, accessed 10 January 2020.

[11] Queppelin, "Agile Methodologies for Mobile Application Development", 2016, <www.queppelin.com/agile-methodologies-for-mobile-application-development/>, accessed 10 January 2020.

[12] B. Subham, K. Aditi, M. Madhuleena, and B. Madhurima,"A Comparative Study: Java Vs Kotlin Programming in Android Application Development", International Journal of Advanced Research in Computer Science, vol.9, no.3, pp.41-45, 2018.

[13] O. Janis and D.Uldis, "Unified Modeling Language", 2017, <https://www.sciencedirect.com/topics/computer-science/unified-modeling-language>, accessed 5 December 2019.

[14] B.M. Prasanna, J.E. Huesing, R. Eddy, and V.M. Peschkem, "Fall Armyworm in Africa: A guide for Integrated Pest management", 1st edn, feed the future: the U.S. Government's Global Hunger & Food Security Initiative, 2018.

[15] S. Tadele and G. Emana, "Determination of the economic threshold level of tomato leaf miner, Tuta absoluta Meyrick (Lepidoptera: Gelechiidae) on tomato plant under glasshouse conditions", Journal of Horticulture and Forestry, vol.10, no.2, pp. 9-16, 2018.