Iraqi Journal of Science, 2023, Vol. 64, No. 8, pp: 4302-4311
DOI: 10.24996/ijs.2023.64.8.46

ISSN: 0067-2904

An Application for Smartphones and Computers to Diagnose and Control Potatoes Insects

Ahmed Asaad Zaeen[1] , Ruaa Muhammed Dhedan[2], Lakesh K. Sharma[3]
[1]Remote Sensing Unit, College of Science, University of Baghdad, Iraq
[2]Chemistry Dept., College of Science for Women, University of Baghdad, Iraq
[3]Soil and Water Sciences, University of Florida, USA

Received: 23/8/2022 Accepted: 28/10/2022 Published: 30/8/2023

Abstract

Nowadays, a strong relationship between the agriculture sectors and digital technologies is really interesting. The article describes how recent intelligent technologies can improve agricultural fields. Mobile applications are software programs created on smartphones, tablets, and computers. Agricultural fields mainly represent the pillar of the economy and the business sector that fulfills the world's food requirements. The United States has a valuable rank in potato production, which depends on this production economically.

Nevertheless, so many insects affect potato yield production quantitatively and qualitatively. So, a smartphone App was created to help potato growers diagnose insects that directly attack potato crops and treat them. The created App focuses on a list of the common insects that attack potato crops in Maine State. App Inventor Platform, run by the Massachusetts Institute of Technology (MIT), was used to develop the application. Insect images and insecticides information were collected from the Cooperative Extension Department at Presque Isle City, Aroostook County, Maine, USA. The App provides essential details regarding insect types, life cycles, where they are coming from, and the time of attacking the plants. The App includes a list of effective insecticides that control insects. The App also provides helpful instructions concerning trade names, dose per acre of insecticides, and whether it should be applied to soil or plant leaves. Money and time are saved by applying this App since farmers do not need to spend time collecting samples and bringing them to the lab.

Keywords: Smart Agriculture, Potato, Insect, MIT, App Inventor.

تطبيق على الهاتف الذكي والحاسوب لتشخيص ومعالجة حشرات البطاطا

احمد اسعد زعين[1], رؤى محمد ضيدان[2] , لاكيش ك شارما[3]
[1]وحدة الاستشعار عن بعد، كلية العلوم، جامعة بغداد، بغداد، العراق
[2]قسم الكيمياء، كلية لعلوم للبنات، جامعة بغداد، العراق
[3]قسم علوم التربة والمياه، جامعة فلوريدا، الولايات المتحدة الامريكية

*Email: ahmed.asaad@sc.uobaghdad.edu.iq

الخلاصه

في الوقت الحالي، تعد العلاقة بين قطاعات الزراعة والتقنيات الرقمية امرا مثيرا للاهتمام، حيث يصف هذا البحث قابلية التقنيات الذكية لتطوير الجانب الزراعي، والتي هي برامجيات تم اعدادها للعمل على الهواتف والحاسوب والاجهزة اللوحية. ان المنتجات الزراعية بشكل أساسي هي دعامة الاقتصاد وقطاع العمل الذي يلبي الاحتياجات الغذائية وان الولايات المتحدة الامريكية تمتلك ترتيبا ذو قيمة جيدة في انتاج البطاطا على مستوى العالم. رغم ذلك، هنالك العديد من الحشرات التي تؤثر على انتاجية المحصول كما ونوعا. لذلك، تم انشاء تطبيق على الهاتف الذكي لمساعده منتجي البطاطا على تشخيص الحشرات مباشرتا في الحقل ومكافحتها. يحتوي التطبيق على قائمة الحشرات الشائعة التي تصيب محصول البطاطا في ولاية مين سنويا. تم استخدام المنصة الكترونية (App Inventor platform) لانشاء التطبيق والتي تدار من قبل المعهد التكنولوجي في ولاية ماسوجستس الامريكية(MIT) . استحصلت بيانات الحشرات من صور ومبيدات من الدائرة الارشادية التعاونية في مدينة بريسكايل، مقاطعة اروستوك التابعه لولاية مين الامريكية. هذا التطبيق يزود بمعلومات مهمة بخصوص نوع الحشرة ودورة حياتها ومن اين مصدرها والموسم الذي تهاجم في النباتات. كما يحتوي التطبيق على قائمة بالمبيدات الفعالة التي تسيطر على نشاط تلك الحشرات. اضافة الى نشرة تعليمية ارشادية حول تلك المبيدات، مثل الاسم التجاري ونسبة المبيد الى الماء، وكذلك ايضاح عملية الرش على الاوراق او على التربة مباشرتا. هذا التطبيق يوفر الجهد والمال والوقت الذي يبذله المزراعين اثناء جمع العينات والذهاب الى الدوائر الارشادية للتشخيص ومن ثم الذهاب للاسواق، حيث سيقوم الفلاح بتشخيص نوع الحشرة واختيار نوع المبيد بنفسه في الحقل دون الحاجة للعمليات التقليدية الانفة الذكر.

1. Introduction

The term smart agriculture points to technologies like sensors, the Internet of things, robots, location systems, and artificial intelligence on the farm. The most significant purpose is to enhance the quantity and quality of the crops while optimizing the human labor employed [1].

Mobile Apps are software programs designed to work on smartphones, tablets, and other devices [2]. Mobile Apps were established to conduct primary missions of computer programs, such as Email, web browsing, calendar, contacts, weather forecast, etc. However, the increasing demand for new mobile products and services has encouraged App designers to develop mobile Apps for commerce, banking, health, and tourism to provide for the specific needs of various business sectors, though this rarely includes agricultural markets [3]. The agriculture sector is an important sector of the economy, providing food worldwide. However, mobile Apps are more limited in agriculture than in other fields [4]. The possible applications of smartphone technology may lead to sensing nutrient requirements, fertilizer recommendations, and pest management [3].

Smartphones have become a helpful tool for easy mobility, affordable cost, and practical applications in all agricultural fields. Smartphones can be connected to different physical sensors to assist in several farming tasks [5]. Growers may feel it is more beneficial to have a low-cost smartphone that can act interchangeably with many different sensors rather than costly computing technology with limited capabilities [5].

Potato growers can use smartphone technology to reduce fertilizer costs by applying only the needed amount, increasing profit. Dealing with precise nutrition information can improve soil quality when growers stop excessively adding nutrients based on guidelines [6], where better nutrient application guidance decreases leaching to soil layers, ensuring maximum plant use. Agricultural smartphone applications may reduce soil and local aquatic ecosystem stress from an oversaturation of chemical nutrients [7].

The most crucial possibility of smartphone technology is insect and pest management. Approximately $120 billion of crops are lost due to pest attacks, with an additional $10 million in environmental damage [8]. Using developed pest control smartphone applications, growers may diagnose pest attacks in their field directly before exacerbating the problem. Thus, the App may decrease pesticide use, lower overall environmental impacts, and improve crop yield.

Scientists developed a nondestructive imaging approach depending on Otsu segmentation and Bezier curves to measure the foliar damage in leaves with or without border damage [9]. BioLeaf is a methodology of a mobile application that is freely distributed to smartphone users, which has successfully achieved foliar damage quantification with precision comparable to that of human specialists. Researchers think that BioLeaf might support soybean producers, reducing the time to measure foliar damage, reducing analytical costs, and defining a tool that applies not only to soybean but also to different crops like cotton, beans, potatoes, coffee, and vegetables [9].

Gaddis, 2015 [10] mentioned that learning to build a programming language could be challenging. The wasted time is often spent learning programming fundamentals and tracking down the missed semicolons or unbalanced braces. Syntax mistakes in the App Inventor platform are never an obstacle because they never or rarely occur. The programmers can build an app by dragging and dropping blocks, like a puzzle game, to create a fully functional programming database. Since programmers do not need to locate and fix syntax mistakes, they can plan the wanted actions in Apps to carry out and arrange them into a precise sequence. However, runtime and logic mistakes may occur because beginners use the wrong block code. The syntax is not a significant problem; App builders dedicate time to developing and editing algorithms. App Inventor platform simplifies programming by eliminating many inexperienced beginners who commonly commit errors. Using traditional programming languages like Java or C++, low-experienced beginners frequently make typing errors that lead to misspelled keywords, losing punctuation characters, and other such mistakes. These sorts of errors are known as syntax errors. Suppose there is at least one syntax error. In that case, it cannot be translated into an executable program, or it would be an uncomplete-unuseful program. Therefore, beginners need to spend time tracking the error. However, a syntax error will not happen considerably with the App Inventor platform because users do not need to type codes [10].

Instead of typing codes, users can drag and drop code blocks into the editor window schematic construction blocks. The blocks, which symbolize actions and data, can be “snapped together, like puzzle parts, to build fully functional programming statements”. The idea is more than typing programs that perform calculations or analyze data. However, the users construct mobile Apps that they can perform on tablets or smartphones to support Android devices. If they have an excellent idea for an App, they can build and install it on devices. If they want to share applications with people, they can upload them to the Google Play Store or App Inventor Gallery [10]. Researchers used an image processed on MATLAB R2014 software, using an image processing toolbox to identify and count aphids on soybean. The algorithm for counting with SONY camera images has correlated (r^2=0.96) significantly with manual counting [11].

1.1 *The limitations of Using App Inventor*

Computational ability is frequently considered in conventional textual representations; for instance, the AP Computer Science A exam is evaluated in Java. It can be difficult for students who learn block-based expressions to transition to textual representations.
Consequently, it is essential to assist students in transitioning to textual languages while confirming that knowledge acquired in the visual language is not failed. A researcher is actively managing this by molding the App Inventor Java Bridge. The Java bridge permits users to translate an App Inventor application into a Java application consistent with Android Studio, the official text-based development environment employed to create native Android applications. Encouraging learners to transition from the AP Computer Science 0 curriculum to AP Computer Science A. Another existing restriction of App Inventor is that its structure impedes code reuse. Code reuse is one of the essential computational consideration concepts. Numerous text-based languages are vital for code libraries and dependency administration, allowing app creators to build on each other's work more efficiently. While App Inventor provides a gallery for posting completed app source code, the society has yet to develop the granularity of libraries standard in other programming languages. Suggesting a possibility of resuming the growth of the platform and user society and is a deserving subject for additional investigation [12].

This study focuses on the benefits of using a smartphone App to diagnose and control insect-attacking potatoes.

2. Materials and Methods

2.1 Data Source

The database (images and data) was collected from the University of Maine, cooperative extension, Potato IPM Program, Presque Isle, Maine State, USA. The data has specific information about the most common insects that attack potato crops in Maine State, such as Aphids, Colorado Potato Beetle, Cut Worm, European Corn Borer, Potato Leafhoppers, Potato Flea Beetles, White Grub, and Wireworms. The database contains a detailed description of each type of insect. It briefly describes insect life and where and when it can be found in soil or plant bodies, in addition to a list of effective insecticides that can treat insects. For each type of pesticide, there is a brief description regarding the trade name, rate (dosage) per acre of that chemical, other instructions regarding the active treatment time, and any conflict with other chemicals [13].

2.2 The Steps for Creating The App

App Inventor platform was used to build the App [14]. As a requirement, a google account was used to sign in and start working on a project. Starting from the "Create button" in the window's upper right. Then from the upper left of the window, the "New Project button" is used to write the project's name, "**Potato-Insect pests in Maine**.
"

The name must start with an alphabetical letter without spacing between letters, so the name of our project will be written in the form "**PotatoInsectpestsinMain**." After the first letter, the rest of the name characters can be alphabetical letters, numbers, or underscore symbols.

In the "Designer window," the user can set up the App's components and layout. Before discussing program setup steps, a scheme or a map explains how each button allows the user to navigate within the App from the App icon to the last step, "the exit" (Figure 1). Following the scheme shows that the App has been built primarily from screens and buttons. So, creating any new screen starts counting the buttons from number 01; that is why we have two buttons (01

and 02) on screen 01. Screen number 02 has eight buttons starting from 01 to 08. Screen 01 has two buttons: the icon that lets the user access the App and the contact information button, which provides contact information for the authors regarding any questions or help.

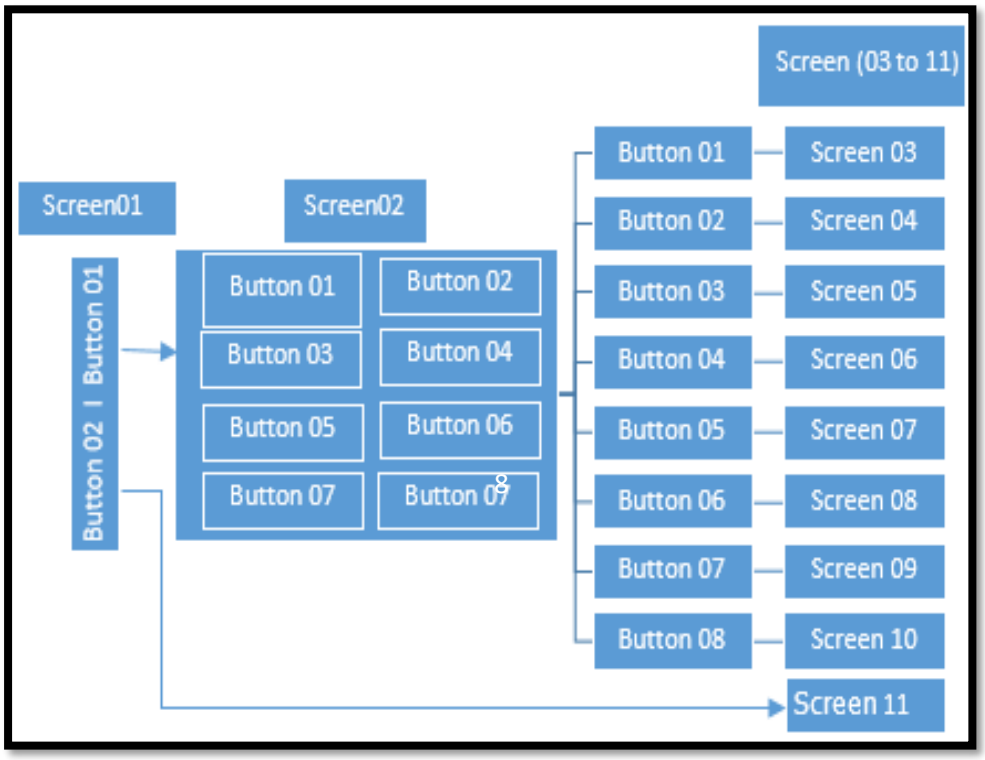


Figure 1: The scheme of buttons and screens distribution in the App

Button 01 on screen 01 takes the user to screen 02 with a list or table of images of common insects that attack potato crops in Maine State annually. Each image is considered a button for another screen; if the user clicks on that button (image), it takes him to a screen that briefly describes that insect. Button 03 takes the user to screen 03, and button 04 takes the user to screen 04 until button 08. Button 02 in screen 01 takes the user to screen 11, which has the author's contact information regarding any questions or help. Two buttons were created on screen 01: the first homepage and the contact information.

Back to the working steps, from the Layout list, which includes a "Horizontal Arrangement" and User Interface, a "Button" and a "Clock" were dragged and dropped in the viewer window. Screen 01 was used to manage regarding App's title, icon shape, background color, name of each screen, and alignment. "Horizontal Arrangement" was managed regarding the appearance of the length and width of the icons; 80 percent was chosen for each to have an appropriate appearance for the icon, the first homepage. Button 01 was managed to be 10% and 11% for length and width, respectively, located at the bottom of screen 01. As a reminder, all these steps have been carried out in the "Design window."

Regarding building codes, the window has switched from "Designer" to "Blocks," where block code is built by clicking on "Horizontal Arrangement," and from the drop list, the code that says (When Clock1. Timer. do) was dragged and dropped in the viewer window. For the Clock, the code that says (set Clock1. TimeEnabled to) was dragged and dropped in the viewer window. The "False" text code was dragged and dropped in the viewer window from the logic box. From "Control," the code that says (open another screen-screenName) was dragged and

dropped in the viewer window. The text box was pulled from "Text" and dropped in the viewer window, and screen02 was written inside this text box code. It looks like the "lock and key theory," where all these code blocks are connected (Figure 2). When the user clicks button 01, it permits him to access screen02. Regarding the contact information button, the code says that when the user clicks on button 02, the user can navigate to screen 11 (Figure 3).

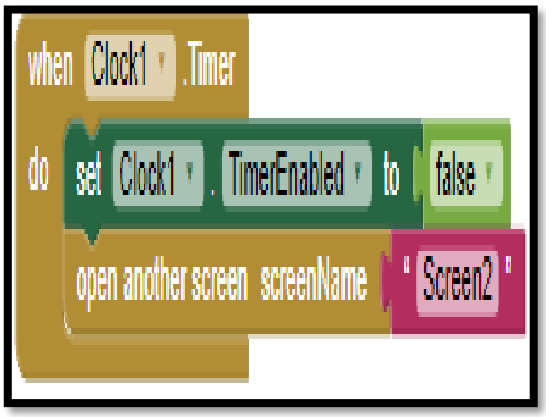


Figure 2: Blocks programming language and function; when the user clicks on the button, it permits the user and navigates him to screen 02.

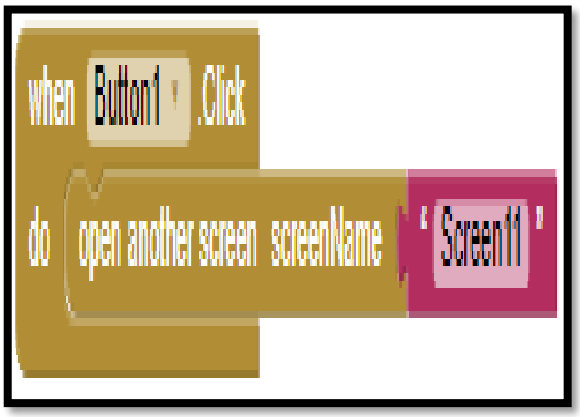


Figure 3: Blocks programming language and function; when the user clicks on the button 01, it navigates to screen 11.

Eight buttons in two rows, each with a label representing its name, were required for the homepage. The working area was switched to the "Blocks" window regarding building codes. For each button, codes are dragged and dropped in the viewer window according to the sought function, such as from Buttons (3 to 10), the code that says (when Button. Click, do) was dragged and dropped in the viewer window. Also, the code that says (open another screen screenName) was dragged and dropped in the viewer window. The text box was pulled and dropped from the text button in the list in the viewer window. The code parts were together in one block code (Figure 4). To explain the combined codes built, each code says, "when the user clicks on the button X, take him to another screen with a specific number. Screen 02 has a close (exit) button, and we will explain that code later, see (Figure 5). The screens (03 to 10) have the same two buttons in the upper part: the home button (2nd homepage) and the close button. For the home button, the combined parts of the codes (when home. Click do) and (open another screen screenName) with "Textbox" were dragged and dropped in the viewer window. In the text box, screen02 (2nd homepage) is the home button's direction when clicking on the home button (Figure 5). For the close button, the two parts of the code (when Close. Click do), the code says (close application) was dragged and dropped in the viewer window to explain the combined codes. It is "when the user clicks on this button, close the application completely" (Figure 6). Screen11 has just one button that lets the user return to the 1st homepage. The two parts of the code are (when Button01. Click do) and (open another screen screenName) with a text box that has the name of the screen,01, that takes the user to the 1st homepage (Figure 7).

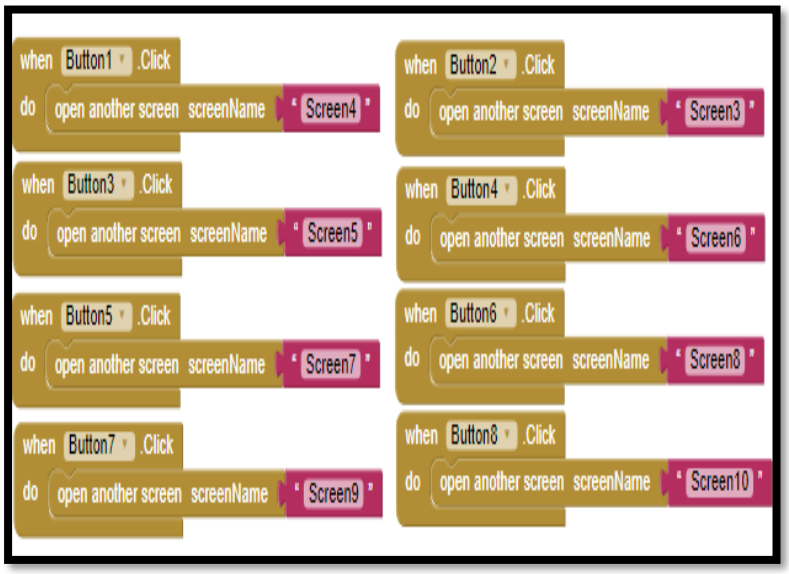


Figure 4: The blocks' programming language in screen02. When the user clicks on any specific button, it navigates him to a specific screen. For example, Button 01 navigates the user to screen 04, and Button 02 navigates the user to screen 03, and so on for the rest of the buttons.

Figure 5: Blocks' programming language and function "when the user clicks on home button, navigate to the homepage."

Figure 6: Blocks' programming language and function "when the user clicks on the close button, close the App completely."

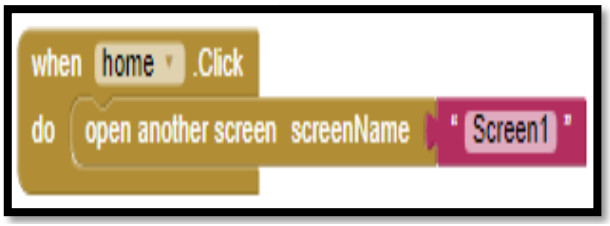


Figure 7: Blocks' programming language and function "when the user clicks on the home button, navigate him to the 1st homepage."

3. Results and Discussion

3.1 Steps of Using The App

After completing the smartphone application, it is ready for publication to be used by potato growers or anyone interested in potato insects. The App is not complicated, and users or beginners can follow it (Figure 8) to understand how to find a specific target. The first homepage has two buttons, the upper one, "the big icon," and the lower one, which navigates the user to the contact information (Figure 11). When the user clicks on the first homepage of the App (Figure 9), it navigates him to the second homepage (Figure 10); it has an alphabetical list of common insects that attack potato crops in Maine State. The list consists of images of insects and their names; by clicking on these insect images, the user can get a brief description

of that insect, such as the insect's life cycle, and a list of active insecticides that can be used for treatment. To make it accessible for potato growers regarding selecting insecticides, three essential points support each type of insecticide: the trade name, how to use it (treating soil or plant's body directly), and some guidelines. Regarding insect resistance management, essential instructions were attached regarding each insecticide; for example, see (Figures 12-14) that present the Colorado potato beetle.

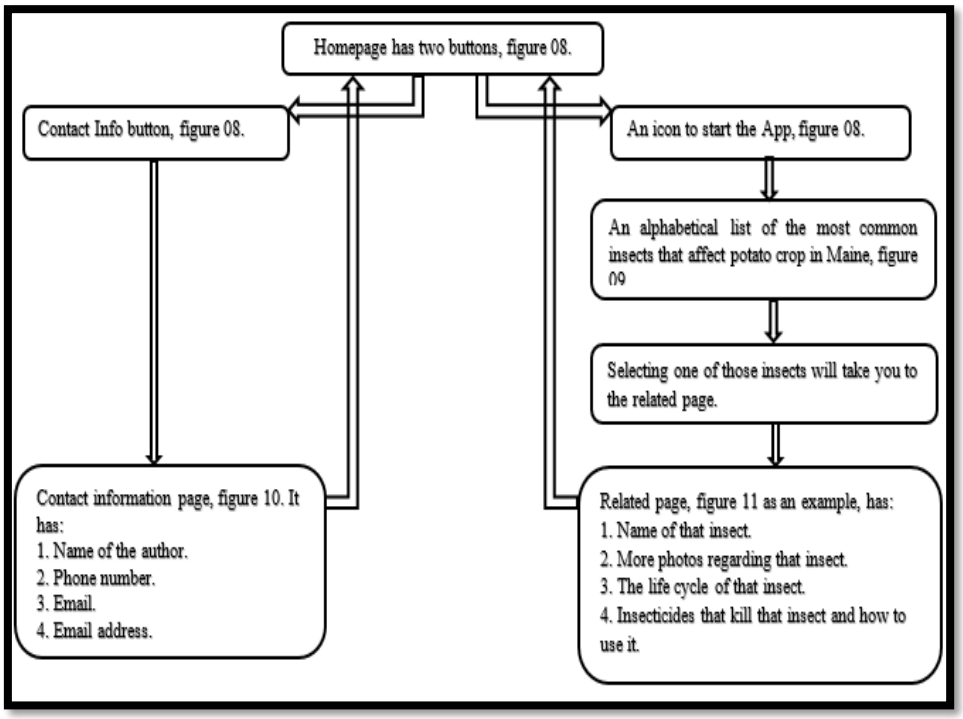


Figure 8: Scheme of using the smartphone application.

Figure 9: The 1st homepage of the App.

Figure10: The 2nd homepage of the App that has a list of the common insects that attacks potato crop in Maine State.

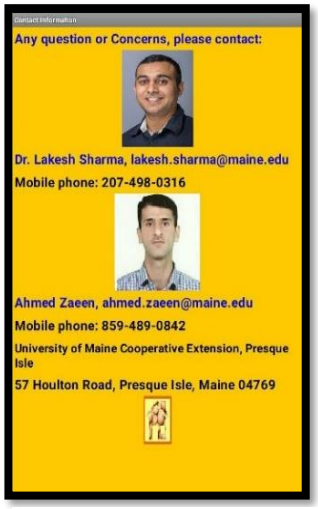


Figure11: The contact information for the authors regarding any question.

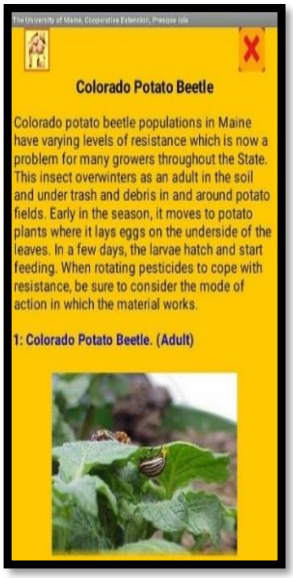
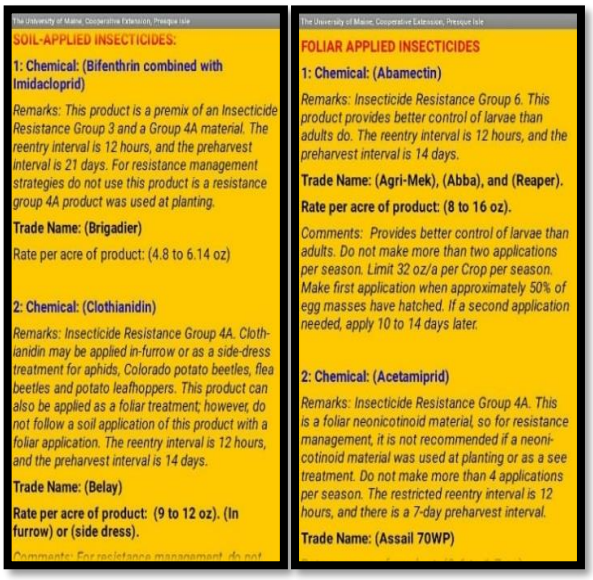


Figure 12: A brief life description about an insect.



Figures 13 and 14: An example of insecticide with information regarding use. Also, the two buttons on the bottom of each page, which are the homepage button and the exit button.

3.2 The advantages of this Procedure

For students operating on developing Apps in this field, using the App Inventor Platform is better than others for many reasons. Instead of writing codes, a user only needs to drag and drop code blocks and schematic construction blocks into an editor window. The code blocks, which symbolize actions and data, can be "clasped together, like puzzle parts, to build fully functional programming statements". If programmers have an excellent idea for an App, they can build and install it on the device. In addition, programmers can upload applications to the Google Play Store or App Inventor Gallery.

3.3 Internet is not Required to Run The App

Some smartphone applications need to be connected to the internet (Wi-Fi) to work, which is considered a lousy characteristic because some farms are far from the city, and there is no internet service. However, some smartphone applications can work without connecting to the internet, which is valuable. This App follows the Apps that can be used offline (without an Internet connection) because it downloads all content to the physical storage after installation. However, sometimes users face a problem using Android Apps offline (without an Internet connection), where they must have a fast internet connection to run the App. If users experience this issue, they can fix it instantly by following these steps, closing the App and going to the settings icon from Apps or applications, looking for Google play services, and looking for storage or managing space. Choose a clear cache and rerun the App without an active internet connection.

3.4 The Economic Efficiency of Using the App

A comparison between the App developed to diagnose and control insects and the conventional system that potato growers use when insects attack crops. In a conventional procedure, after recognizing the insect infection on the potato crop, farmers should collect samples from crop leaves and the attacking insect for diagnosis. Sometimes, farmers may collect the wrong sample incorrectly, leading to incorrect interpretation and using the wrong insecticide. Another disadvantage of using the conventional system is driving for taking samples to the lab and waiting for the diagnostic results may need a day or more depending on

staff availability. Compared to using the App, farmers do not need to collect plant or insect samples, saving time and reducing the cost of driving from the field to the lab and returning to the field. Farmers can diagnose insects by themselves, so they will be experts in insect diagnosis. In conclusion, potato growers could get a quick diagnosis, save money and time, and gain experience.

3.5 Future Work

In order to support potato growers completely, crops need Apps that help growers diagnose weeds and nutrient deficiency.

References

[1] S. Markova, "What is smart agriculture and why smart agriculture is the future?,*"* 2020. Available: https:/ / ondo.io/what_is_smart_agriculture/

[2] N. Serrano, J. Hernantes, and G. Gallardo, "Mobile web apps," *IEEE Software*, vol. 30, no. 5, pp. 22-27, 2013. https://ieeexplore.ieee.org/abstract/document/6588524

[3] K. Olmstead and M. Atkinson, "Apps permissions in the Google Play store," *Pew Research Center. November*, vol. 10, 2015. https://apo.org.au/node/58954

[4] R. Duncombe, "Digital technologies for agricultural and rural development in the Global South," CABI, Centre for Development Informatics (CDI), University of Manchester, Manchester, UK, doi: 10.1079/97817863933 64.0000, 2018.

[5] S. Pongnumkul, P. Chaovalit, and N. Surasvadi, "Applications of smartphone-based sensors in agriculture: a systematic review of research," *Journal of Sensors*, vol. 2015. https://www.hindawi. com/journals/ js/2015/ 195308/

[6] R. Lal, "Soil quality and sustainability in Methods for assessment of soil degradation": CRC Press, pp. 17-30, 2020.

[7] W. De Vries and L. Schulte-Uebbing, "Impacts of nitrogen deposition on forest ecosystem services and biodiversity," Atlas of ecosystem services: drivers, risks, and societal responses. *Springer International Publishing, Cham*, pp. 183-189, 2019.

[8] D. Tokel, B. N. Genc, and I. I. Ozyigit, "Economic impacts of Bt (Bacillus thuringiensis) cotton," *Journal of Natural Fibers*, vol. 19, no. 12, pp. 4622-4639, 2022.

[9] B. B. Machado, J. P. Orue, M. S. Arruda, C. V. Santos, D. S. Sarath, W. N. Goncalves, G. G. Silva, H. Pistori, A. R. Roel, and J. F. Rodrigues-Jr, "BioLeaf: A professional mobile application to measure foliar damage caused by insect herbivory," *Computers and Electronics in Agriculture*, vol. 129, pp. 44-55, 2016. https://arxiv.org/pdf/1609.08004.pdf

[10] M. A. Inventor, "With MIT App Inventor, anyone can build apps with global impact," ed, 2021.

[11] M. Maharlooei, S. Sivarajan, S. G. Bajwa, J. P. Harmon, and J. Nowatzki, "Detection of soybean aphids in a greenhouse using an image processing technique," *Computers and Electronics in Agriculture*, vol. 132, pp. 63-70, 2017. https://www.sciencedirect.com/science/article/abs/pii/S0168169916310791

[12] S. C. Kong and H. Abelson, "*Computational thinking education,"* Springer Nature, 2019.

[13] Cooperative-Extension-University-of-Maine, "*Potato Pest Control Guide-Cooperative Extension: Potatoes,"* 2011. Available: https://extension.umaine.edu/potatoes/pest-control-guide/

[14] X. Deng, "Group collaboration with app inventor," Massachusetts Institute of Technology, 2017.