Progressive Web App for Crop Field Data Collection

Phunon Koysawat\textsuperscript{1}, Chayanon Boonprakob\textsuperscript{1}, Khwantri Saengprachatanarug\textsuperscript{2}, Arnut Chaosakul\textsuperscript{1}, Panupong Wanjantuk\textsuperscript{1}, Mahisorn Wongphati\textsuperscript{3}, Santawat Santiteerakul\textsuperscript{3}, Adulwit Chinapas\textsuperscript{1}, Kanda Runapongs\textsuperscript{a} Saikaew\textsuperscript{1}

\textsuperscript{1}Department of Computer Engineering, Faculty of Engineering, Khon Kaen University, 123 Mittraparp Rd., A. Muang, Khon Kaen, 40002
\textsuperscript{2}Department of Agricultural Engineering, Faculty of Engineering, Khon Kaen University, 123 Mittraparp Rd., A. Muang, Khon Kaen, 40002
\textsuperscript{3}HG Robotics Co., Ltd., 1407-1410, 14th Floor Chulalongkorn Research Building, 254 Chulalongkorn University, Phayathai Rd. Wang Mai, Pathum Wan District, Bangkok, 10330

Email: krunapon@kku.ac.th (Kanda Runapongsa Saikaew)

Abstract. General farming needs to have a matter of data collection. For efficient cultivation planning and harvesting, information must be collected for evaluation and analysis to make each harvest decision. For example, we need to assess plants' ages, changes in size, color, and other senses (sweetness, smell, etc.) to plan for different crop field growth phases. Also, data collection problems occur due to the need to collect data in multiple locations simultaneously. Collecting information manually in the form of documents makes the process challenging to analyze. There is a need to collect field data more efficiently. Thus, this article proposes developing a web application to replace manual note-taking to reduce errors in recording and gathering data and missing data. Users can locate and modify crop field locations using latitude and longitude and work seamlessly with the Google Earth application. Furthermore, to accommodate practical use in real-life, the web application developed using Progressive Web Application (PWA) can perform without the Internet. Besides, PWA web applications are as fast as using an application on a smartphone without downloading and saving space. Based on the experimental result, compared with the process without using the application, it was found that the proposed application could reduce time in collecting sugarcane sampling data by about 45.28%

1. Introduction
Nowadays, various technologies are involved in people's daily lives to support and solve work or activities problems more efficiently and conveniently. Smart farming is a new form of agriculture by bringing technology in farming more productively in several phases, such as plant selection and harvesting. To perform such tasks, we need a method to collect agricultural data effectively. The traditional approach in collecting data tends to cause errors and time-consuming. Thus, this article proposes and develops a progressive web application to assist farmers or merchants in collecting crop field data for making a more appropriate decision in crop growing or crop sale.
2. Related work
This section describes the basic knowledge needed to develop an application and related applications.

2.1. Progressive web application
Web applications generally require the Internet all the time. However, sometimes web application users may be in an environment that the Internet is unavailable or unstable. Several mobile applications can function without the Internet, but that comes with the extra space usage on mobile devices and continuous application updates. Nevertheless, there is a type of web application that can work without the Internet and does not need users to update the application continuously. That type of web application is called a progressive web application. A progressive web application can have customized notifications and support an offline mode. It also helps to increase the efficiency of the use of web applications even faster than before. The development will not look much different from general web application development. The development language would be HTML, CSS, JavaScript, which are typical web application development languages.

2.1.1. Operation of the service worker
To develop a progressive web application, we need to have a system called the service worker, which runs in the background of web applications. JavaScript controls the behavior of applications and caches (a type of memory) to increase web applications' performance. Working with a transmission network as the active API when running the API system is a network request Web server directly. Then it will store the data in the cache. And when the API is rerun instead of the network request, it is changed to the previously held cache instead. In addition to making web applications faster, caching can also define parts of the web application to be available offline. It also can set a notification. All of these functions can be controlled through the system. Service worker keeps web applications on the home screen of your device, just like regular applications. The app manifest customizes colors, themes, and icons.

2.2. Local storage and indexed DB
In general, browsers have a variety of methods and formats, local storage, and indexedDB. The two will be different from each other, which can be used according to the web application's operation's advantages and disadvantages. Local storage is suitable for the type of data storing that uses cookies. It is one of the HTML5 web storage APIs. On the other hand, IndexDB collects data that can be stored as a small database on the device that stores data as a key like web storage but will use an index to find information faster. It is asynchronous to avoid conflicts in the application system.

2.3. Related applications
Examples of applications that are close to use

2.3.1. Chaona Thai. It is an application developed by the Internal Trade Department for Thailand’s rice farmers. The system will estimate cost profit to make decisions before planting and creating an agricultural calendar that guides farmers on what to do and when to apply weed fertilizers with cost-saving and ready to be sold on the market. This application's advantage is that its recommendation system helps users make decisions, but it does not have functions that support crop field data collection.

2.3.2. Insect Shot. It is an application that automatically counts the number of Brown Planthoppers in the rice field. Images will be sent to the server and return the result, the estimated number of Brown Planthoppers. Unlike Insect Shot, the proposed application was developed to collect general data in the crop field and not focus on the automatic counting of images in the fields.

2.3.3. Chao Kaset Agriculture. It is an application developed by the National Electronics and Computer Technology Center (NECTEC) to collect plant types and areas from users to assist them in planning to
plant a crop field. However, unlike Chao Kaset Agriculture, the proposed application focus on collecting the crop field data products, such as their location and quality.

3. System Implementation

This section will discuss functional specification, system architecture, and system implementation of the proposed system. The proposed app can collect time information, GPS location, support offline mode, and support various types of users, including general user, guest, user admin, and system admin. It also supports scanning and recognizing text by using Google Vision API. Figure 1 depicts the system architecture of the proposed web application.

![Figure 1. Progressive web application](image)

Figure 1 shows the system is operated through a tool that controls the server's communication, called the service worker. There are two types of work through the cache and the work through the network. Working through caching is a function that will enable faster work on areas of previously used data such as images, text, and web applications when there is no Internet connection (offline). Working over a network is controlled by the service worker. The system will set the usage priority on network usage first. When there is an internet connection, it will be able to access the database's information.
There are four types of user access classification. The first category is a guest user who has not yet registered. Such users cannot access the main functions of the system. Because accessing the system requires an organization to exist, this user type can only select the organization to request the organization login to the system to collect information. The second category is a general user who can use the system login section, fill in the field crops, and view the history of filling in the database. The last two types are administrators. A user administrator can have access to all functions; on the other hand, a system administrator can create an organization and assign every user role.

The system has three main functions into three parts: 1) plant selection for data recording, 2) hold data for offline usage, and 3) show recordings in the database. The system authentication uses Firebase Authentication. Furthermore, the system has an OCR part to allow users to take photos and use Google Cloud Vision API to automatically scan and save data to fields according to each form data topic. The system used MongoDB as a NoSQL database to define the limits of the data (the Constraint) or relationship information (Entity Relationship) and saved it as JSON (JavaScript Object Notation).

4. Implementation Result
This section describes the screenshots of the proposed and developed web application.
Figure 3 shows the login page to which users can log in via Google mail. Figure 4 illustrates the working section of this page has tabs at the top to redirect the pages and display the affiliated organizations' information. Figure 5 shows the current user's GPS location with some errors within the threshold, adjustable by the administrator.

Figure 6 shows the page that holds data for working offline and managing data in the card form. When the Internet is available, users can update data to the system. Figure 7 illustrates the page where the user can record the data, including GPS location and a button to scan the paper with some text. Figure 8 shows the logging reports tab. This page is divided into sections that display user information with a logout button at the bottom, and the next section shows the details of the crop field data.
Figure 9 illustrates all organization members whose administrator can manage all users and can accept/reject access requests under the organization. Figure 10 shows the creation form of recording farm crops within the organization. Creating a form requires an internet connection.

5. Experimental Result Analysis

We also performed experiments to compare the data collection processes' execution time without using the app and using the app, as shown in Figure 11 and Figure 12.

| Step | Process name                                      | Units  | Distance | Time    | Symbol |
|------|---------------------------------------------------|--------|----------|---------|--------|
| 1    | Set the cone against the cane to take a picture  | 1      | 15       | 15      |        |
| 2    | Taking a picture of cane                         | 1      | 10       |         |        |
| 3    | Check the GPS location from the phone and take notes | 1 | 60       |       |        |
| 4    | Count the number of cane trunks and take notes   | 2      | 20       | 150     |        |
| 5    | Cut two cane samples                             | 1      | 60       |         |        |
| 6    | Take the sugarcane sample to the working point   | 1      | 20       | 20      |        |
| 7    | Brake cane leaves from the trunk                 | 1      | 60       |         |        |
| 8    | Weigh trunks and cane leaves and take notes      | 1      | 100      |         |        |
| 9    | Measure cane sweetness and take notes            | 1      | 60       |         |        |
| 10   | Write a cane sample bag                          | 1      | 40       |         |        |
| 11   | Two soil samples were collected                   | 1      | 90       |         |        |
| 12   | Chop the cane into pieces to fill the bag        | 1      | 40       |         |        |
| 13   | Check the cane sample                            | 1      | 30       |         |        |
| 14   | Collect the cane sample into the bag             | 1      | 30       |         |        |
| 15   | Check and validate information                    | 1      | 60       |         |        |
| 16   | Fill data and pictures into an Excel file        | 1      | 200      |         |        |
| 17   | Import Excel data and pin GPS to Google Earth app| 1      | 300      |         |        |

Total 265 1,325 13 0 2 1

Figure 11. Flow process without using the app
Figures 11 and 12 notice that steps 1-15 of the flow processes both without and using the app are the same. However, in several steps, it took less time when using the app than when not using the app. Moreover, it took about 500 seconds to fill in data in excel and export the data to Google Earth app for the process without using the app in the last few steps. On the other hand, for the process of using the app, it took about 40 seconds to export the data from the developed app to Google Earth app.

Using the app dramatically reduces the time when it comes to storing data in Excel and Google Earth because people do not need to sit and fill in the information before, thus reducing the time up to 460 seconds. The process with using the app had a total time less than that without using the app, about 45.28%.

6. Conclusions

We proposed and developed a web application that can store field crop data in a database. Such data includes photos, locations, and details of crop fields in different organizations. The objective of the developed app is to shorten the record-keeping time and minimizes human-incurred errors. The developed progressive web application can run as native mobile applications and use cache to speed up data access and work even without an internet signal. Our experimental result showed that the proposed application could save the processing time in collecting sugarcane sample data by about 45.28%. In the future, we can use the web application data for further analysis of agronomic crops such as cassava, which is another critical economic crop.

References

[1] Progressive Web Apps | Web. (n.d.). Google Developers. Retrieved September 15, 2019, from https://developers.google.com/web/progressive-web-apps/

[2] Build a PWA With Vue & Ionic 4. (n.d.). Retrieved September 15, 2019, from https://www.youtube.com/watch?v=6H1wftPS0oo
[3] Hartington, M. (2019, April 3). *Announcing the Ionic Vue Beta*. The Ionic Blog. https://ionicframework.com/blog/announcing-the-ionic-vue-beta/

[4] *Introduction—Vue.js*. (n.d.). Retrieved September 15, 2019, from https://vuejs.org/v2/guide/

[5] Ionic. (n.d.). *Ionic Framework—Ionic Documentation*. Ionic Docs. Retrieved September 15, 2019, from https://ionicframework.com/docs/