Article

Smartphone Application Using a Visual Programming Language to Compute Drying/Solar Drying Characteristics of Agricultural Products

Baher M. A. Amer 1,2,* and Houssam Chouikhi 3,4

1 Department of Agricultural Systems Engineering, College of Agricultural & Food Sciences, King Faisal University, P.O. Box 400, Al-Ahsa 31982, Saudi Arabia
2 Department of Agricultural Engineering, Faculty of Agriculture, Cairo University, Giza 12613, Egypt
3 Department of Mechanical Engineering, College of Engineering, King Faisal University, P.O. Box 380, Al-Ahsa 31982, Saudi Arabia; hchouikhi@kfuf.edu.sa
4 Laboratory of Electromechanical Systems (LASME), National School of Engineers of Sfax, University of Sfax, Sfax 3038, Tunisia

* Correspondence: bamer@kfuf.edu.sa; Tel.: +966–507-983-969

Received: 14 August 2020; Accepted: 28 September 2020; Published: 2 October 2020

Abstract: Drying is an important process for food security. A software application on a smartphone can be used to prevent mistakes while carrying out the calculations for drying investigation and research using several mathematical equations. The objective of this research was to develop a software program on cellphones depending on some of the important mathematical equations which are used to conduct some of the drying/solar drying characteristics to assist researchers and engineers who work in the food engineering field for sustainable development. This innovative software application can be installed on mobiles in Android and iOS systems to save effort and time. Mathematical models were developed to compute drying/solar drying calculations. These mathematical models were integrated into the flowchart to develop a software program using visual programming language in the operating system and create the user interface. The collected data from references were used to compare the results from the normal way of calculations with results from the developed software. The results were agreed for both normal calculations and the software. This application software can calculate the moisture content of product, drying ratio, quantity of removed moisture from the fresh products, required heat to evaporate moisture from fresh product, and efficiency of the solar collector.

Keywords: software application; visual programming language; smartphone; drying

1. Introduction

Drying is one of the important processes to preserve food and ensure food security. It is considered as a simultaneous application of heat and removal of moisture from foods [1]. A lot of researchers are working on drying food to improve it [2]. Solar drying is a method of food drying using a solar dryer that uses solar energy instead of the traditional types of energy and keeps the environment without pollution [3]. One of the main parts of a solar dryer is called a solar collector which collects solar energy to use it for evaporating the moisture from fresh food [4,5].

Many calculations are used to compute some characteristics of the drying process, such as the moisture content of the product, drying ratio, the quantity of removed moisture from the fresh products, required heat to evaporate moisture from the fresh product [6,7]. These calculations take a lot of time and effort from scientists and engineers to compute. Therefore, it is necessary to find a method to make the calculations in a short time. A software application operated by a smartphone is a good solution to
compute these calculations in a short time. In addition, the calculation of the efficiency for any type of solar collector is very important for the researchers [8,9].

A visual programming language (VPL) allows users to create programs by manipulating graphical elements instead of using textual programming. A VPL permits programming in spatial arrangements while using visual components such as blocks holding expressions, text, and graphical symbols [10]. These components can be easily resized, dragged, and dropped. Indeed, VPL allows non-expert developers to easily design the user interface by using the computer mouse. Moreover, VPL offers the ability to connect the code to every interface component. The connected code is then executed whenever the user interacts with the interface component in the installed application. In other words, VPL is easy to learn and to use and the users can easily test and use the developed mobile application. As pointed out by Kong and Abelson [11], VPL such as Scratch and MIT App Inventor are introduced to students in primary and secondary schools in order to learn programming and develop logical thinking skills. Furthermore, Scratch and MIT App Inventor are suitable for students to build scientific applications and games. For example, MIT App Inventor is a VPL mainly intended for non-computer science users, with little programming background and experience, to easily create applications for smart mobile devices [12]. However, MIT App Inventor is a tool invented to teach computational thinking concepts by the development of mobile applications for two operating systems: Android and iOS. It is a free and open-source software. It uses a graphical user interface (GUI): applications are created by dragging and dropping components into a design view and using a visual blocks language to program application behavior [13].

App-Inventor Companion is the program that allows the application to run and debug on Android devices. Its new version, which works on iOS devices, is still under development (i.e., in progress). MIT App Inventor is based on constructionist learning theories, which emphasize that programming can be a vehicle for engaging powerful ideas through active learning.

MIT App Inventor is recognized by the benefits of fast iterative design on learning [11]. Hence, App Inventor is used to build mobile applications for educational purposes such as mathematical applications [14], and multiple-choice question examination systems [15].

The objective of this research was to develop a software program on cellphones depending on some of the important mathematical equations used to calculate some of the drying/solar drying characteristics to assist researchers and postgraduate students working in this field. Usability analysis of this application is performed by a mobile application survey. This software application can be installed on mobiles in the Android system and the iOS system to save effort and time. Precisely, the software should calculate the quantity of removed moisture from the fresh products, the drying ratio, the required heat for evaporating moisture from the fresh product to dry, heat input to the solar collector, heat useful from the solar collector, and efficiency of the solar collector.

2. Materials and Methods

2.1. Data and Knowledge Collection

The mathematical equations were developed using the knowledge, data, and information on food drying available in the literature [1,16]. Furthermore, contact was made with experts in this field, such as postgraduate students, researches, and engineers in drying food factories in Saudi Arabia.

A software application was created to assist researchers and engineers in computing the characteristics of food drying using smartphones. This software application is designed to compute the amount of moisture removed from fresh food and its moisture content, the required heat to evaporate moisture from fresh product, as well as the energy gain from the solar collector.

However, the advantages of the combination of mathematical equations and a software application that uses data and inference methodology are saving effort and time besides solving problems that are difficult and needing significant expertise for their solution.
2.2. Application Description

The application designed in this paper allows to calculate some of the drying/solar drying characteristics to assist researchers and postgraduate students working in this field. The users of this application can calculate:

- the moisture content at any time dry basis (d.b.) and the moisture content at any time wet basis (w.b);
- the mass of moisture removed from the fresh product, the drying ratio, the latent heat, the sensible heat, and the overall heat required to evaporate moisture from food;
- the heat input to the solar collector, heat useful from solar collector, and efficiency of the solar collector.

The main steps of that application are given as follows:

- Step 1: open the application,
- Step 2: choose the language (Arabic or English) or exit (close the application),
- Step 3: choose the calculations to do (moisture content or drying characteristics or solar drying characteristics), go back to the home screen, or exit.
- Step 4: write the input parameters then click on buttons to calculate the output parameters, go back to the home screen, or exit.

The developed application is built in MIT App inventor (online version: http://ai2.appinventor.mit.edu). The application includes three screens: (1) Home screen, (2) English screen, and (3) Arabic screen. Only the design of the English screen is shown in this paper. The design of the Arabic screen is similar to the English one. The logic blocks of both screens 2 and 3 are similar and shown in this paper.

2.3. Moisture Content Calculations

Moisture content can be expressed in two ways [16].

The moisture content wet weight basis \( M_w \) is calculated as

\[
M_w = \frac{\text{mass of moisture}}{\text{mass of product}}
\]  

where the mass of product can be made up of moisture and dry matter. Thus,

\[
M_w = \frac{\text{mass of moisture}}{\text{mass of moisture} + \text{dry matter}} \times 100
\]  

The moisture content dry weight basis \( M_d \) is expressed as

\[
M_d = \frac{\text{mass of moisture}}{\text{mass of dry matter}}
\]

The mass \( W_d \) of the dried samples could be determined by drying the samples of mass food \( W_i \) in a vacuum oven at 70 °C until the mass of the samples became stable in its weight, referring to [17].

However, the moisture content of the product on a dry basis, \( M_d \) at any time was expressed as [2].

\[
M_d = \frac{W_i - W_d}{W_d}
\]

where:

\( W_i \) is the mass of the food at time \( t_i \).

It could determinate the mass of product by weighing the drying tray with its load of foods at any time during the drying process.
The moisture content of foods can sometimes be calculated on a wet basis. It can be shown by determination the mass of dry matter [16],

\[ M_w = \frac{100 M_d}{1 + M_d} \]  \hspace{1cm} (5)

In food composition tables and drying calculations, the moisture content is most often expressed on a wet basis, whereas it is more often expressed in sorption isotherms, drying curves, and studies by the moisture content on a dry basis. The amount of moisture in food is most easily eliminated by taking a representative sample and drying it in an oven to constant mass. Normally, a considerable moisture gradient is established within the food during dehydration processes.

2.4. Drying Characteristics

2.4.1. Quantity of Removed Moisture from the Fresh Products

The quantity of removed moisture from fresh products is given as follows:

\[ m_m = m_p \times \frac{M_i - M_f}{100 - M_f} \]  \hspace{1cm} (6)

where:
- \( m_m \) — the quantity of removed moisture from the total mass of fresh product (kg),
- \( m_p \) — mass of fresh product at the beginning of the drying process (kg),
- \( M_i \) — moisture content, wet basis, of the fresh product (%),
- \( M_f \) — final moisture content, wet basis, of the product (%).

2.4.2. Drying Ratio \( DR \)

The drying ratio can be defined as the ratio between the mass of fresh product to the mass of 1 kg dried material, and it can be expressed in the following equation:

\[ DR = \left( \frac{W_i}{W_d} \right) \left( \frac{W_d}{W_d} \right) \]  \hspace{1cm} (7)

2.4.3. Heat Required to Evaporate Moisture from Fresh Product \( Q_{overall} \)

It is the overall heat required to evaporate moisture from the fresh product and it can be calculated using the following equation:

\[ Q_{overall} = Q_{sen} + Q_L \]  \hspace{1cm} (8)

2.4.4. Latent Heat \( Q_L \)

The latent heat of vaporization is defined as heat that causes water to evaporate from the product. It may be expressed with the following equation:

\[ Q_L = m_m \lambda_m \]  \hspace{1cm} (9)

where:
- \( Q_L \) — latent heat of moisture vaporization per kg (J),
- \( \lambda_m \) — latent heat of moisture vaporization per kg taken as 2300 J/kg [18].
2.4.5. Sensible Heat $Q_{sen}$

It is defined as the heat required to raise the temperature of a product to the desired temperature for drying. It may be expressed with the following equation:

$$Q_{sen} = m_p c_{p,p} \Delta T_p = m_p c_{p,p} (T_{p,\text{out}} - T_{p,\text{in}})$$  \hspace{1cm} (10)

where:
- $Q_{sen}$ — sensible heat to raise foodstuff temperature to the desiring drying temperature (J),
- $c_{p,p}$ — specific heat of product, taken as 4180 J/kg K [19],
- $\Delta T_p$ — the difference between product outlet and inlet temperature of dryer (K),
- $T_{p,\text{in}}$ — product inlet temperature to the dryer (K),
- $T_{p,\text{out}}$ — product outlet temperature from the dryer (K).

2.5. Solar Drying Characteristics

2.5.1. Input Heat to Solar Collector $Q_{in,col}$

In the solar drying experiments, one may need to calculate more information about the input heat to the solar collector, which can be expressed by the area of collector in m$^2$ and inclined solar radiation in W/m$^2$, as the following:

$$Q_{in,col} = I_s \times A_{col}$$  \hspace{1cm} (11)

where:
- $A_{col}$ is the collector area (m$^2$),
- $I_s$ is the inclined solar radiation (W/m$^2$).

2.5.2. Useful Heat from Solar Collector $Q_{u,Col}$

The useful heat from the solar collector in Watt can be determined from the following equation [20]:

$$Q_{u,Col} = \dot{m}_a c_{p,a} (T_{a,\text{out}} - T_{a,\text{in}})$$  \hspace{1cm} (12)

The mass flow rate of air could be determined from the following equation [21]:

$$\dot{m}_a = \rho_a V_a = \rho_a u_a A_{sc}$$  \hspace{1cm} (13)

where:
- $\dot{m}_a$ is the mass flow rate of air (kg/s),
- $A_{sc}$ is the inlet cross-sectional area of the solar collector (m$^2$),
- $u_a$ is the air speed (m/s),
- $\rho_a$ is the air density (kg/m$^3$), taken as 1.106 kg/m$^3$ [19],
- $c_{p,a}$ is the specific heat of air (J/kg·K), taken as 1006 J/kg·K [19],
- $V_a$ is the air flow rate to the solar collector (m$^3$/s),
- $\Delta T_a$ — the difference between the air outlet and inlet temperature of solar collector (K),
- $T_{a,\text{in}}$ — air inlet temperature to the solar collector (K),
- $T_{a,\text{out}}$ — air outlet temperature from the solar collector (K).

2.5.3. Solar Collector Efficiency $\eta_{col}$

The solar collector efficiency ($\eta_{col}$) is obtained as in [22,23]:

$$\eta_{col} = \frac{Q_{u,Col}}{Q_{in,col}} \times 100 = \frac{\dot{m}_a c_{p,a} (T_{a,\text{out}} - T_{a,\text{in}})}{I_s \times A_{col}} \times 100$$  \hspace{1cm} (14)
2.6. Mobile Application Design and Programming

2.6.1. Application Design and Logic Blocks

In order to design the first screen in the mobile application as shown in Figure 1, some components from the lists of user interface and layout were added in the designer viewer, as a first step. These components are five labels, one horizontal arrangement, one button, and one list picker. The latter two were set together in a horizontal arrangement.

![Design of screen 1 (Home screen).](image)

The second step in designing the first screen was to program the different components in the blocks viewer as shown in Figure 2. A global name was initialized to make a list of languages (Arabic and English) as elements of the list picker. According to the user selection, another screen will be opened (screen 2 or 3). The exit button was programmed to close the application.
Figure 2. Logic blocks of screen 1 (Home screen).

In the logic block “When Language after Picking” shown in Figure 2, we started by a logic block “if”: if true then close the current screen. This latter logic block was added after testing the first version of this application. It seems that without the first logic block “if”, users would face a problem when they open screen 2 or 3 and then come back to the first screen. In this case, if they clicked on the exit button, they would not be able to close the application, they would have to click again. This discovery was also used in the programming of the other screens, i.e., before moving to any screen, the program should close the current screen then open the destination screen.

In the first part of the second screen, five buttons were set together in a vertical arrangement as shown in Figure 3. These buttons allow the user to come back to the home screen, go to moisture-content calculations, go to drying-characteristics calculations, go to solar drying calculations, or close the application.
Figure 3. Design of screen 2 (part 1: Calculations choice).

Figure 4 shows the logic blocks of the first part in screen 2. When the home button is clicked, the current screen (screen 2 or 3) will be closed and the first screen will be opened. When the Moisture content button is clicked, this latter will be invisible. Additionally, the third and fourth parts of the current screen (drying characteristics and solar drying table arrangements) will be invisible. However, drying characteristics and solar drying buttons, as well as the moisture table arrangement, will be visible. According to Figure 4, when the drying-characteristics button is clicked, this latter will be invisible. Moreover, the moisture content and solar drying table arrangements will be invisible. Nevertheless, the moisture content and solar drying buttons, together with the third part in the current screen (drying calculations), will be visible. In accordance with Figure 4, when the solar drying button is clicked, this latter will be invisible. Furthermore, the moisture-content and drying characteristics table arrangements will be invisible. But the moisture content and drying-characteristics buttons together with the fourth part in the current screen (solar drying calculations) will be visible.
Figure 4. Logic blocks of both screen 2 and screen 3 (part 1: Calculations choice).

The second part of the second screen is shown in Figure 5. A table arrangement is added as the main component in this part. Inside this table, five labels, two textboxes, and two buttons are added. According to Figure 5, the mass of product at any time and the mass of dry matter are considered as input parameters. On the other hand, the moisture content at any time d.b. and the moisture content at any time w.b. are considered as output parameters.
Figure 5. Design of screen 2 (part 2: Moisture content).

Figure 6 shows the logic blocks of calculations-choice buttons below parts 2, 3, and 4. These buttons are similar to those used in Figure 4 (calculations choice). The buttons are set in a horizontal arrangement below the tables of calculations in order to allow the user to move, freely and at any time, between the tables of calculations, to come back to the home screen, or to exit the application.
Figure 6. Logic blocks of both screen 2 and screen 3 (Calculations choice inside parts 2, 3, and 4).

Figure 7 shows the logic blocks used to calculate the output parameters in the moisture-content part. When the user clicks on the output buttons, these blocks calculate the output results based on Equations (4) and (5). According to Figure 7 and when $M_d$ button is clicked, the first step is to write “…” in the $RM_d$ label as an empty output result. In the first step, if the mass of product at any time ($m_i$) is empty then the application reminds the user to write it as input parameter. Else, if the mass of dry matter ($m_d$) is empty, then the application reminds the user to write it as an input parameter. Else, the moisture content at any time d.b. ($M_d$) will be calculated based on Equation (4) then stored in the $RM_d$ label as an output parameter with one decimal digit.
Figure 7. Logic blocks of both screen 2 and screen 3 (part 2: Moisture content).
A similar procedure is used to calculate the moisture content at any time w.b. \((M_{w})\) based on Equation (5) then stored in the \(Rm\) label as an output parameter with one decimal place. The procedure described here to find the output parameters \(M_{i}\) and \(M_{w}\) will be similarly used later in this paper to find the output parameters \((m_{m}, DR, Q_{L}, Q_{sen}, Q_{over}, Q_{in,col}, Q_{in,col}, \text{ and } \eta_{col})\) shown in figures.

The third part of the second screen is shown in Figure 8. A table arrangement is added to organize the different components of this part. Inside this table, six labels and six textboxes are added to represent the input parameters, while five labels and five buttons are added to represent the output parameters. According to Figure 8, the input parameters are the mass of fresh product, the initial moisture content, the final moisture content, the latent heat of moisture, the specific heat of product, and the product-temperature difference. However, the output parameters are the mass of moisture removed from the fresh product, the drying ratio, the latent heat, the sensible heat, and the overall heat.

![Figure 8. Design of screen 2 (part 3: Drying characteristics).](image)

Figure 9 shows the logic blocks used to calculate the mass of moisture removed from fresh product and the drying ratio (Equations (6) and (7)). According to Figure 9, when the \(m_{m}\) button is clicked, the first step is to write “…” in the \(Rm_{m}\) label as empty output result. In the first step, if the mass of fresh product \((m_{m})\) is empty then the application reminds the user to write it as an input parameter. Else, if the initial moisture content \((M_{i})\) is empty then the application recalls the user to write it as an input parameter. Else, if the final moisture content \((M_{f})\) is empty then the application reminds the user to write it as an input parameter. Else, the mass of moisture removed from the fresh product \((m_{m})\) will be calculated based on Equation (6) then stored in the \(Rm_{m}\) label as an output parameter with one decimal place. A similar procedure is used to calculate the drying ratio \((DR)\) based on Equation (7) then stored in the \(RDR\) label as an output parameter with one decimal digit.
Figure 9. Logic blocks of both screen 2 and screen 3 (part 3.1: Mass of removed moisture and drying ratio).
Figure 10 shows the logic blocks used to calculate the latent heat, the sensible heat, and the overall heat (Equations (8)–(10)). According to Figure 10, the same procedure used previously to find the output parameters ($m_f$ and $DR$) is used here to find the latent heat ($Q_L$), the sensible heat ($Q_{sen}$), and the overall heat ($Q_{overall}$). In addition, if the latent heat of moisture ($\lambda_m$) is empty then the application reminds the user to write it as an input parameter. Else, the latent heat ($Q_L$) will be calculated based on Equation (9) then stored in the $RQ_L$ label as output parameter with one decimal digit. Moreover, if the specific heat of product ($C_p$) is empty then the application reminds the user to write it as input parameter. Else, if the product-temperature difference ($\Delta T_p$) is empty then the application reminds the user to write it as an input parameter. Else, the sensible heat ($Q_{sen}$) will be calculated based on Equation (10) then stored in $RQ_{sen}$ label as an output parameter with one decimal place. Consequently, and by a simple addition, the overall heat ($Q_{overall}$) will be calculated based on Equation (8) then stored in $RQ_{overall}$ label as an output parameter with one decimal digit.

Figure 10. Logic blocks of both screen 2 and screen 3 (part 3.2: Latent heat, Sensible heat, and Overall heat).
The fourth part of the second screen is shown in Figure 11. A table arrangement includes the different components of this part. Inside this table, seven labels and seven textboxes are added to represent the input parameters, whereas three labels and three buttons are added to represent the output parameters. Accordingly, the input parameters are the area of solar collector, the solar intensity, the inlet cross-sectional area of solar collector, air velocity inside the collector, air density, air specific heat, and the air temperature difference inside the collector. On the other hand, the output parameters are the input heat to the solar collector, the useful heat from the solar collector, and the collector efficiency.

![Figure 11. Design of screen 2 (part 4: Solar drying calculations).](image)

Figures 12 and 13 show the logic blocks used to calculate the input heat to solar collector, the useful heat from solar collector, and the solar collector efficiency (Equations (11), (12), and (14)).
Figure 12. Logic blocks of both screen 2 and screen 3 (part 4: input heat and useful heat).
Figure 13. Logic blocks of both screen 2 and screen 3 (part 4: efficiency of solar collector).
According to Figure 12 and when the $Q_{\text{in,col}}$ button is clicked, the first step is to write “…” in the $RQ_{\text{in,col}}$ label as an empty output result. In the first step, if the area of solar collector ($A_{\text{col}}$) is empty then the application reminds the user to write it as an input parameter. Else, if the solar intensity ($I_s$) is empty then the application reminds the user to write it as an input parameter. Else, the input heat to the solar collector ($Q_{\text{in,col}}$) will be calculated based on Equation (11) then stored in $RQ_{\text{in,col}}$ label as an output parameter with one decimal digit.

Based on a similar procedure, the useful heat from the solar collector ($Q_{\text{u,col}}$) is then calculated according to Equation (12) and stored in $RQ_{\text{u,col}}$ label as an output parameter with one decimal place. Consequently, and by a simple division, as shown in Figure 13, the solar collector efficiency ($\eta_{\text{coll}}$) will be calculated based on Equation (14) then stored in the $R\eta_{\text{coll}}$ label as an output parameter with one decimal digit.

2.6.2. Application Programming

MIT App Inventor was employed in this work to develop a mobile application for both Android and the iOS devices. This application was made according to the operating steps shown in the mobile application flowchart (Figure 14), and by integrating the mathematical models presented in this paper, which relate the input and the output parameters of that application.

![Mobile application flowchart (operating steps)](image)

**Figure 14.** Mobile application flowchart (operating steps).

The programmed application contains three main screens. Many components are used to design the application, such as button, textbox, label, list picker, table arrangement, vertical arrangement, and horizontal arrangement. This application was made by default for phone size (505, 320). The screen
size could be easily changed in the software (i.e., MIT App Inventor) to tablet size (675, 480) or monitor size (1024, 768). The .apk and .aia files could be generated to be used later in specified devices.

The mobile application flowchart describing the operating steps is shown in Figure 14. According to this flowchart, when the user starts the application, the Home screen (Screen 1) will be opened. Then the user selects what to do: exit or select the language.

When the user clicks on exit, the application will be closed. Otherwise, when the user clicks on the language list picker, the list of languages appears, Arabic and English. The user selects either Arabic or English language. If the Arabic language is selected, then screen 3 will be opened. Else, if the English language is selected, then screen 2 will be opened. For both screens, the user will see five buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit. The user can select one of these buttons: (1) Home, (2) Moisture content, (3) Drying characteristics, (4) Solar drying characteristics, and (5) Exit.

In the case of moisture-content calculations, the user must write the values of input parameters: the mass of product at any time and the mass of dry matter. After that, if the user clicks on the output buttons then the results appear. In this case, the output parameters are the moisture content at any time d.b and the moisture content at any time w.b. In the case of drying characteristics, the user must write the values of input parameters: the mass of fresh product, the initial moisture content, the specific heat of product, and the product-temperature difference. After that, if the user clicks on the output buttons, then the results appear. In this case, the output parameters are the mass of moisture removed from the fresh product, the drying ratio, the latent heat, the sensible heat, and the overall heat. In the case of solar drying characteristics, the input parameters are the area of solar collector, the solar intensity, the inlet cross-sectional area of solar collector, air velocity inside the collector, air density, air specific heat, and the air temperature difference inside the collector. When the user fills the input labels then clicks on the output buttons, the output parameters immediately appear, more precisely, the input heat to the solar collector, the useful heat from the solar collector, and the collector efficiency. After the calculations are done, the user can redo the same calculations or move freely to another calculation screen. The user can also come back to the home screen or exit the application at any time.

2.6.3. Application Availability and Evaluation

The mobile application designed in this paper was initially tested by developers on different devices:

- Samsung Galaxy J4 Plus (2019),
- Huawei Y7 Prime (2018),
- LG K10 (2016),
- Alcatel 1T7 (2018),
- Samsung Galaxy A21s (2020).

The calculations done by this application and by the calculator are identical. The application is simple, clear, and fast in loading screens and calculations. The application screens are sufficiently detailed to make the user experience worthwhile (i.e., good image resolution). Let us recall that this application is designed for both the Android and the iOS systems. However, this application cannot run on Windows or Mac devices.

3. Usability Analysis

Usability analysis allows to check if the mobile application is easy to use. This analysis generally focuses on the interaction of users with the application. For example, we can measure the time that the user takes to perform a specific task, the number of clicks, the number of errors, etc., in order to
prove that the application is easy to use. It is important to ensure that the mobile application is high in usability standards. For instance, the application was tested to make sure it can be used by specific users to realize specific targets with effectiveness, efficiency, and overall satisfaction in a particular environment of use. Many factors must be considered to create the mobile application such as design, speed, responsiveness, quality of content, ease of learning, and efficiency of use. In other words, the interaction of users with the application was tested in order to improve the application interface.

Usability is highly relevant and required during the application development process. Usability testing is planned to identify usability problems in the mobile application. Several usability evaluation methods (UEM) have been developed in the literature. UEM serve to guarantee that the usability of the application is in concordance with the system requirements and the user expectations. System usability analysis serves to guarantee that the acceptability and effectiveness of the mobile application will be improved.

Developing successful mobile applications can be a challenging job which needs a lot of consideration and understanding of what the users need. It is important to understand how users experience the mobile application. There are different ways to collect mobile application feedback such as: Application Programming Interface (API), Software Development Kit (SDK), Webview, Mobile Application Survey (MAS), etc. API is a set of definitions, tools, and protocols used to build mobile applications. API allows logging continuing actions of the user interaction together with related window properties and timestamps. API is released and flexible on how to build and implement a mobile application. In this case, internet connection is not instantly required to get user feedback; i.e., the feedback response is hidden and then submitted when the user is connected to the internet. SDK is a downloadable software package that includes the tools needed to build mobile applications. SDK is an inclusive set of APIs that permits users to carry out any action required to build mobile applications [24]. Webview is defined as a browser packed into the mobile application to get a hybrid application; i.e., the application is built using web technologies, such as JavaScript or HTML, and then packaged as a native application available in the application store. Webview is easy to implement, but an internet connection is instantaneously required to get user feedback.

For more details about the usability analysis, the readers are referred to the work by Bastien [25], where the author reviewed many works conducted in the field of user testing. According to Bastien [25], the selected topics are pertinent for evaluating applications in the field of medical and health care informatics. These topics include the number of users, test procedure, usability testing tools, and evaluating mobile applications.

Due to recent advances in smartphone technology, the usability of mobile applications is indeed critical for their implementation because of the relatively small screens and awkward keyboards. Moreover, usability testing in traditional laboratory is frequently tedious and expensive, and does not reveal real usage cases. For these reasons, Ma et al. [24] proposed an effective toolkit allowing to automatically collect user interface events, related to the user interaction with the mobile application. This toolkit is implemented on Android apparatus and tested in a real installed application, and then compared with a traditional laboratory testing tool. Related to Ma et al. [24], the toolkit was built based on an upgraded SDK compatible with Android Version 2.3 (API level 10). More recently, Jeong et al. [26] combined some methods and tools to identify usability problems, effectively without any source code modifications, via usability testing in mobile applications. The authors proposed tracing, modeling, and comparison methods for measuring the similarity of user behaviors, then evaluated the similarity of user behavior and used it in detecting usability problems. The developed approach was experimentally tested in a laboratory to verify its validity, [26].

To explore what the users need, it is possible to ask them directly through online surveys. This allows to get their feedback quickly and easily. A mobile application survey (MAS) is a fast way to collect feedback from mobile application users, especially when the coding methods (SDK, API, Webview, etc.) are not easy to implement in a short period. For this reason, some users in the field of food process engineering were selected to test the application presented in this paper, and then
send their feedback through MAS. An evaluation survey was created to record the feedback about the mobile application from the four types of participants related to the field of food process engineering, which are undergraduate students, postgraduate students, researchers, and engineers. The items of this survey included the following: language, smartphone brand, role of participant, number of buttons in the application, usage in food engineering, interface of application, design, ease of use, user requirements, user satisfaction, efficiency and accuracy in the calculations, saving time and effort, and recommendation to colleagues.

4. Results

The developed software can compute the moisture content of product, the drying ratio, the quantity of removed moisture from the fresh products, the required heat to evaporate moisture from fresh products, and the solar collector efficiency. The developed application can be used by Arabic or English speakers (Figure 15).

Moreover, the user can choose the calculations to perform, such as calculating the moisture content of food, drying and solar drying characteristics (Figure 16). Furthermore, examples of input and output data are presented in Figures 17–19.

![Figure 15. Home screen: Language selection and Exit (screen 1).](image-url)
Figure 16. Calculations choice (screen 2).

Figure 17. Moisture content of food (screen 2).
The application was designed in a manner such that when the input parameters are added, the output parameters will appear just after clicking on the output buttons. The application can...
compute the moisture content of food at any time d.b., referring to Equation (5) and the moisture content at any time w.b., referring to Equation (6), as shown in Figure 17, where both the input and the output parameters are displayed. It can also compute the mass of moisture removed from the fresh product, the drying ratio, the latent heat, the sensible heat, and the overall heat required for drying food referring to Equations (7)–(10), as shown in Figure 18. Moreover, it computes the input heat to solar collector, the useful heat from solar collector, and the solar collector efficiency, referring to Equations (11)–(14), as shown in Figure 19.

Furthermore, the collected information and mathematical equations of moisture content and drying characteristics from the literature were used to compare the results by the normal way of calculations with results by the developed software. The results were agreed completely for both normal calculations and the software.

All previous screens are identical in the Arabic version on this software application. The Arabic version of this software is suitable for the Arabic countries which have some postgraduate students or engineers who are not dealing well with the English language as the Arabic language is their native language.

Figure 20 shows the results of the evaluation survey which contains 13 items and was delivered via Google Form. There are 40 participants who responded to the evaluation survey of the mobile application as 32.5% undergraduate students, 22.5% postgraduate students, 22.5% researchers and 22.5% engineers. Most of the participants who are engineers, researchers, and some of the postgraduate students prefer to use the mobile application in the English language. Some of them, especially, the undergraduate students and some postgraduate students, prefer the Arabic version. However, few of the users tested the application with both languages. Many of them have Samsung and Huawei mobiles, about 72.5%. The rest of them used Sony, LG, or some other brand. According to Figure 20a, almost all of them (97.5%) confirmed that the number of buttons is appropriate in the application to facilitate calculations. Most of them (92.5%) affirmed that the calculations in the application are useful in the field of food drying process. Many of them (85%) responded that the application interface is appropriate and contains information explaining the purpose of the application. A few of users did not agree (5%) and the rest were neutral. According to Figure 20b, the rating of participants for the application design was 82.5% as excellent and very good, 5% as good, 5% as fair, and 7.5% as poor. Moreover, their rate about the ease of using the application on mobile phone was 87.5% as excellent and very good, 5% as good, 7.5% as fair and poor. However, the users rated the application for meeting their requirements 87.5% as excellent and very good. With regard to the efficiency and accuracy of the application in results calculations, 87.5% rated it as excellent and very good, 5% as good, 7.5% as fair and poor. In addition, 90% rated the application in saving time and effort in calculating results as excellent and very good, 10% as good and poor. Furthermore, they rated the general satisfaction for using the application around 87.5% as excellent and very good, 5% as good, 7.5% as fair and poor. Finally, regrading recommending the application to their colleagues in food process engineering, the user ratings were 92.5% for excellent, very good, and good, but 7.5% for poor.

This developed software application can compute some items related to drying food in a fast way on a smartphone. However, it can be used successfully to calculate the amount of moisture removed from the fresh food and its moisture content, drying ratio, the required heat to evaporate moisture from fresh product, as well as the efficiency of the solar collector.
Figure 20. Cont.
Figure 20. Survey items (a) language, smartphone brand, role of participant, number of buttons in the application, usage in food engineering, and interface of application. (b) Design, ease of use, user requirements, user satisfaction, efficiency and accuracy in the calculations, saving time and effort, and recommend to colleagues.
5. Discussion

5.1. Concepts of the Software Application

The mobile application was developed using a visual environment which provides a graphical interface in order to design and build the application for Android and iOS devices. This environment, namely MIT App Inventor, contains two main parts: components designer and blocks editor. The components designer permits to drag and drop elements such as textboxes and buttons to create the application user interface. However, the blocks editor permits to program the components by arranging logic blocks that are designed similar to puzzle pieces in order to connect them easily. The mathematical equations needed to compute the output parameters in this paper are easily programmed in the blocks editor.

5.2. Advantages and Disadvantages of the Software Application

The main advantage of the MIT App Inventor is the possibility of creating an application without knowledge of any programming language. In addition, the advantages of the combination of mathematical equations and a software application that uses data and inference methodology are to save effort and time besides solving problems that are difficult and needing significant expertise for their solution.

The disadvantages of this application are that it could not run on Windows or Mac devices until now, but there is a further investigation to solve this issue.

5.3. Applicable Usage

This software application on a smartphone is provided to undergraduate students, postgraduate students, researchers, specialists in food engineering, and engineers in food factories. Therefore, an application survey was delivered to 40 participants from different categories, studying, researching, or working in the food engineering field, to get the information related to the design and usage of the mobile application. It is noted that most of the participants are accepting the items of the application and can use it, as well as recommend it to their colleagues.

5.4. Further Studies

Further integration for this software application could be done so that it can be used in Windows and Mac devices. Future works can include the usability analysis of the mobile application by using other methods such as Webview or SDK/API.

6. Conclusions

A software application on a smartphone can be used to prevent mistakes in carrying out the calculations for drying investigation and research using several mathematical equations. This software application can be installed on mobiles in the Android system and the iOS system to save effort and time.

A mathematical model was developed to compute the drying calculations. This mathematical model was integrated into the flowchart to develop a software program using the visual programming language in the Android system and the iOS system and create the user interface. The collected data were used to compare the results by the normal way of calculations with results by the developed software. The results were in agreement for both normal calculations and the software.

This application software on a smartphone can calculate:

- the moisture content of product,
- the quantity of removed moisture from the fresh products,
- the drying ratio,
- the required heat of evaporating moisture from the fresh product to dry,
• the efficiency of the solar collector.

The usability analysis of the mobile application is performed by using an application survey, where the results affirmed the user’s satisfaction, as well as the high effectiveness of the application.

Author Contributions: Conceptualization, B.M.A.A. and H.C.; methodology, B.M.A.A. and H.C.; software, H.C. and B.M.A.A.; investigation, B.M.A.A. and H.C.; writing—original draft preparation, B.M.A.A. and H.C.; revision and final editing, B.M.A.A. and H.C. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number IFT20055.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fellows, P. Food Processing Technology, Principles and Practice, 2nd ed.; Woodhead Publishing Limited and CRC Press LLC: Boca Raton, FL, USA, 2000.
2. Amer, B.M.A.; Gottschalk, K. Studying the affecting factors on drying rates of plum fruits under varying drying air conditions. In Proceedings of the 16th Triennial World Congress IFAC, Prague, Czech Republic, 4–8 July 2005; pp. 52–56.
3. Hossain, M.A.; Amer, B.M.A.; Gottschalk, K. Mathematical modelling of solar assisted hybrid dryer for banana. In Proceedings of the 5th International Technical Symposium on Food Processing, Monitoring Technol. in Bioprocesses and Food Quality Management, Potsdam, Germany, 31 August–2 September 2009; pp. 748–755.
4. Amer, B.M.A.; Hossain, M.A.; Gottschalk, K. Design and performance evaluation of a new hybrid dryer for banana. Energy Convers. Manag. 2010, 51, 813–820. [CrossRef]
5. Fterich, M.; Chouikhi, H.; Bentaher, H.; Maalej, A. Experimental parametric study of a mixed-mode forced convection solar dryer equipped with a PV/T air collector Sol. Energy 2018, 171, 751–760. [CrossRef]
6. Hossain, M.A.; Gottschalk, K.; Amer, B.M.A. Mathematical modelling for drying of tomato in hybrid dryer. Arab. J. Sci. Eng. 2010, 35, 239–262.
7. Natalia, V.; Mauricio, C.; Martínez-Correa Hugo, A.; Lobaton Hugo, F. Drying uniformity analysis in a tray dryer: An experimental and simulation approach. Adv. J. Food Sci. Tech. 2018, 15, 233–238. [CrossRef]
8. Amer, B.M.A.; Gottschalk, K.; Hossain, M.A. Integrated hybrid solar drying system and its drying kinetics of chamomile. Renew. Energy 2018, 121, 539–547. [CrossRef]
9. Simo-Tagne, M.; Ndukwu, M.C.; Zoulalian, A.; Bennamoun, L.; Kifani-Sahban, F.; Rogaume, Y. Numerical analysis and validation of a natural convection mix-mode solar dryer for drying red chilli under variable conditions Renew. Energy 2020, 151, 659–673. [CrossRef]
10. Tsai, C.Y. Improving students’ understanding of basic programming concepts through visual programming language: The role of self-efficacy. Comp. Hum. Behav. 2019, 95, 224–232. [CrossRef]
11. Kong, S.C.; Abelson, H. (Eds.) Computational Thinking Education; Springer: Singapore, 2019.
12. Papadakis, S. Evaluating the efficiency of two programming environments in shaping novices’ attitudes, perceptions, beliefs and knowledge in programming: A comparison between Scratch and App Inventor. Inter. J. Teach. Case Stud. 2018, 10, 31–52. [CrossRef]
13. Adiono, T; Anindya, S.F.; Fuada, S.; Afifah, K.; Purwanda, I.G. Efficient Android software development using MIT App Inventor 2 for bluetooth-based smart home. Wirel. Pers. Commun. 2019, 105, 233–256. [CrossRef]
14. Voštiná, P. Using App Inventor for creating educational applications. In Proceedings of the 9th International Conference on Education and New Learning Technologies (EDULEARN17), Barcelona, Spain, 3–5 July 2017; pp. 10128–10133.
15. Zubair, M.; Sana, I.; Nasir, K.; Iqbal, H.; Masud, F.; Ismail, S. Quizzes: Quiz application development using Android-Based MIT APP Inventor platform. IFACS 2016, 7. [CrossRef]
16. Sodha, M.S.; Bansal, N.K.; Kumar, A.; Bansal, P.K.; Malik, M.A.S. Solar Crop. Drying; CRC Press Inc.: Boca Raton, FL, USA, 1987.
17. AOAC. Official Methods of Analysis of the Association of Official Analytical Chemists, 14th ed.; Published by the Association of Official Analytical Chemists: Arlington, VA, USA, 1984.
18. Cengel, Y.A. Heat Transfer: A Practical Approach, 2nd ed.; McGraw-Hill: New York, NY, USA, 2003.
19. ASHRAE. *Handbook of Fundamentals “American Society of Heating, Refrigerating and Air Conditioning Engineers”*, Atlanta, GA: New York, NY, USA, 2006.

20. Arun, K.R.; Srinivas, M.; Saleel, C.A.; Jayaraj, S. Active drying of unripened bananas (Musa Nendra) in a multi-tray mixed mode solar cabinet dryer with backup energy storage. *Sol. Energy* 2019, 188, 1002–1012. [CrossRef]

21. Yang, T.; Athienitis, A.K. Experimental investigation of a two-inlet air-based building integrated photovoltaic/thermal (BIPV/T) system. *Appl. Energy* 2015, 159, 70–79. [CrossRef]

22. Shahsavari, A.; Eshapour, M.; Talebedehsardari, P. Experimental evaluation of novel photovoltaic/thermal systems using serpentine cooling tubes with different cross-sections of circular, triangular and rectangular. *Energy* 2020, 208, 118409. [CrossRef]

23. Azam, M.M.; Eltawil, M.A.; Amer, B.M.A. Thermal analysis of PV system and solar collector integrated with greenhouse dryer for drying tomatoes. *Energy* 2020, 212, 118764. [CrossRef]

24. Ma, X.; Yan, B.; Chen, G.; Zhang, C.; Huang, K.; Drury, J.; Wang, L. Design and implementation of a toolkit for usability testing of mobile apps. *Mob. Netw. Appl.* 2013, 18, 81–97. [CrossRef]

25. Bastien, J.M.C. Usability testing: A review of some methodological and technical aspects of the method. *Inter. J. Med. Inform. Hum. Factors Eng. Healthc. Appl.* 2010, 79, e18–e23. [CrossRef] [PubMed]

26. Jeong, J.; Kim, N.; In, H.P. Detecting usability problems in mobile applications on the basis of dissimilarity in user behavior. *Inter. J. Hum. -Comput. Stud.* 2020, 139, 102364. [CrossRef]