# PRAKTYCZNE ZASTOSOWANIE TECHNOLOGII MOBILNEJ W UPRAWIE PAPRYKI POD OSŁONAMI



Kamil Porada<sup>1</sup>

<sup>1</sup>Uniwersytet Technologiczno-Humanistyczny, Wydział Transportu, Elektrotechniki i Informatyki, Malczewskiego 29, 26-600 Radom, Polska, kamil.porada@onet.pl

**Streszczenie** – Niniejsza publikacja naukowa ma na celu przedstawienie aplikacji mobilnej, która ma wspomagać pracę w gospodarstwie rolnym specjalizującym się w uprawie papryki pod osłonami. Oprogramowanie zostało stworzone jako natywna aplikacja mobilna dla systemu Android. Jest to swoisty asystent dla plantatora, który przechowuje notatki dotyczące różnych czynności wykonywanych w gospodarstwie. Aby umożliwić tę funkcjonalność, zastosowano model relacyjnej bazy danych SQLite. Pierwszy rozdział pracy skupia się na teoretycznym zbadaniu systemu operacyjnego Android. Autor analizuje jego strukturę, przedstawia języki programowania oraz narzędzia deweloperskie dedykowane do tworzenia tego typu aplikacji. W kolejnym rozdziale przedstawiony jest interfejs użytkownika aplikacji, a także opisane są podstawowe zadania, jakie ma spełniać stworzona aplikacja E-Paprykarz. W ostatnim rozdziale autor opisuje proces implementacji na podstawie dwóch przykładów kodu źródłowego. Aplikacja została również poddana testom w realnych warunkach, które przyniosły pozytywne efekty w procesie uprawy papryki w gospodarstwie w sezonie 2022.

Słowa kluczowe – Android, aplikacja mobilna, papryka, rolnictwo, telefon

#### WSTĘP

Już od zarania dziejów człowiek starał się usprawniać różne dziedziny życia, a matematyka odegrała w tym procesie istotną rolę. Nawet prehistoryczni Homo sapiens, mimo prymitywnego stylu życia, odczuwali potrzebę liczenia. Świadczy o tym fakt, że w prehistorii wykorzystywano prymitywne metody liczenia, jak zapisywanie kresek na ścianach jaskiń lub dodawanie kolejnych kamieni do stosu. Te pierwsze liczydła przyczyniły się do rozwoju matematyki jako nauki [1].

Wraz z dynamicznymi zmianami w XXI wieku, unowocześnienie i automatyzacja dotknęły niemal każdą dziedzinę życia, w tym także rolnictwo. Branża rolnicza doświadczyła znacznego postępu dzięki wykorzystaniu informatyzacji. Obecnie na rynku pojawiają się specjalne oprogramowania wykorzystujące infrastrukturę informatyczną, które mają na celu usprawnienie hodowli zwierząt, uprawy warzyw i owoców.

Jednym z takich oprogramowań jest E-Paprykarz, aplikacja mobilna dedykowana gospodarstwom rolnym specjalizującym się w uprawie papryki. Jest to natywna aplikacja mobilna stworzona dla systemu Android, która obejmuje 6 modułów: Kalkulatory ogrodnicze, Notatki gospodarstwa, Zabiegi pielęgnacyjne, Kontrola nawadniania, Dziennik dochodów oraz Zapisane lokalizacje. Każdy z tych modułów przyczynia się do usprawnienia różnych czynności związanych z uprawą papryki.

Autor aplikacji, mając ponad 10-letnie doświadczenie w uprawie papryki, zainspirował się tymi doświadczeniami

przy tworzeniu oprogramowania. Proces implementacji aplikacji odbywał się od maja do października, idealnie wpasowując się w okres uprawy papryki w południowym Mazowszu, głównym regionie uprawy tego warzywa. Każda część opracowanego oprogramowania została przetestowana w rzeczywistych warunkach gospodarstwa rolnego specjalizującego się w uprawie papryki w sezonie 2022, co przyczyniło się do optymalizacji działań związanych z pielęgnacją tego warzywa.

Nazwa aplikacji, E-Paprykarz, składa się z przedrostka "e-", symbolizującego powiązanie z Internetem lub elektronicznymi urządzeniami, oraz terminu "paprykarz", który jest lokalnym slangiem dla osoby zajmującej się uprawą papryki. Tworzenie tej aplikacji wynika z potrzeby wypełnienia luki na rynku aplikacji mobilnych, które specjalizują się w konkretnych dziedzinach rolnictwa. Większość dostępnych rozwiązań skupia się na hodowli zwierząt lub ogólnych grupach warzyw i owoców, dlatego E-Paprykarz wyróżnia się swoim skoncentrowaniem na uprawie papryki.

## I. TEORETYCZNE ASPEKTY TWORZENIA APLIKACJI MOBILNYCH DLA SYSTEMU ANDROID

Systemy operacyjne są nieodłączną częścią codziennego życia, choć często działają w ukryciu, pełniąc kluczową rolę w obsłudze urządzeń. Współczesne systemy operacyjne nie ograniczają się już tylko do komputerów i smartfonów. Możemy je znaleźć w różnorodnych urządzeniach, które wpływają na życie codzienne. Ich głównym celem jest umożliwienie korzystania z tych urządzeń w prosty i wydajny sposób. Dlatego warto

zainteresować się tym, jak te systemy operacyjne są zbudowane i jak działają. Poznanie ich podstaw pozwoli lepiej zrozumieć, jak wykorzystać pełny potencjał urządzeń.

System operacyjny Android opiera się na jądrze Linuksa, które stanowi fundament całej architektury [2]. Jądro to pełni wiele istotnych zadań, takich jak zapewnianie bezpieczeństwa, obsługa urządzeń (takich jak klawiatura, kamera, mikrofon), zarządzanie pamięcią, zarządzanie energią oraz zarządzanie siecią [3]. Stanowi ono podstawę, na której opierają się wszystkie inne warstwy systemu Android (Rys. 1). Niestety, jądro Linuksa ma również swoje wady, takie jak podatność na ataki hakerskie, które mogą umożliwić kontrolę nad systemem [4].



Rys. 1. Architektura system Android

Kolejną istotną warstwą jest Hardware Abstraction Layer (HAL), czyli warstwa abstrakcji sprzętowej. Jej głównym zadaniem jest komunikacja między jądrem Linuksa a urządzeniami. Dzięki HAL, Android jest niezależny od konkretnych sterowników sprzętowych, które znajdują się na niższych warstwach. Dodatkowo, HAL zawiera moduły bibliotek, które implementują interfejsy dla różnych komponentów sprzętowych [2].

Powyżej warstwy abstrakcji sprzętowej znajdują się natywne biblioteki systemowe Android, napisane w języku C lub C++ (Rys. 1). Należy do nich m.in. biblioteka SQLite, która umożliwia przechowywanie danych w relacyjnej bazie danych. Inną ważną biblioteką jest WebKit, służący do wyświetlania stron internetowych w formacie HTML. Istnieje także Surface Manager, który zarządza wyświetlaczem urządzenia, oraz Media Framework, odpowiedzialny za odtwarzanie multimediów [3]. Istnieje wiele innych bibliotek, takich jak OpenMAX AL., OpenGL ES, czy Libc, które wspierają różne funkcje systemu Android [2].

Obok bibliotek natywnych C/C++, znajduje się Android Runtime (ART), środowisko uruchomieniowe systemu Android (Rys. 1). ART umożliwia uruchamianie większej liczby wirtualnych maszyn w urządzeniach mobilnych, co jest istotne przy ograniczonej pamięci operacyjnej. Pliki DEX, zoptymalizowane pod kątem oszczędności pamięci, są kompilowane i wykonywane w czasie rzeczywistym. Wcześniej, przed wersją Lollipop, system Android korzystał z maszyny wirtualnej Dalvik [5]. W tej warstwie również działa garbage collector, który zarządza automatycznym zwalnianiem pamięci [6].

Kolejną warstwą w architekturze Androida jest Java API Framework (Rys. 1). Programiści mogą korzystać z różnych usług oferowanych przez tę warstwę za pomocą interfejsów API napisanych w języku Java. Dzięki nim możliwe jest korzystanie z funkcji takich jak obsługa połączeń telefonicznych, zarządzanie lokalizacją GPS, zarządzanie cyklem życia aplikacji, obsługa zasobów urządzenia czy powiadomienia [3].

Na samej górze architektury znajduje się warstwa aplikacji systemowych, która zawiera preinstalowane aplikacje, takie jak poczta, kalendarz, dyktafon, mapy czy zarządca kontaktów (Rys. 1). Aplikacje te są dostępne dla użytkownika od razu po zakupie urządzenia.

W dziedzinie tworzenia aplikacji mobilnych, system operacyjny Android odgrywa istotną rolę, umożliwiając użytkownikom korzystanie z różnorodnych funkcji i usług na swoich urządzeniach. W celu tworzenia aplikacji dla systemu Android istnieją różne języki programowania, które umożliwiają programistom pisanie aplikacji w sposób zrozumiały dla urządzeń mobilnych.

Język Java odgrywał kluczową rolę w rozwoju systemu operacyjnego Android od samego początku. Java jest językiem programowania ogólnego przeznaczenia, który podejście obiektowe. wykorzystuje Dzieki temu programiści mogą tworzyć aplikacje, które składają się z obiektów i korzystają z gotowych bibliotek oraz funkcji systemowych oferowanych przez platformę Android. Java jest językiem imperatywnym, co oznacza, że programista pisze instrukcje krok po kroku, określając, jakie operacje mają być wykonane. Język ten umożliwia również programowanie współbieżne i funkcyjne, co daje programistom szerokie możliwości w tworzeniu rozbudowanych aplikacji Android [2].

Wraz z pojawieniem się języka Kotlin, stał się on coraz popularniejszym wyborem dla programistów tworzących aplikacje na platformę Android. Kotlin jest językiem programowania statycznie typowanym, który oferuje wiele nowych funkcji i ułatwień w porównaniu do Javy. Jedną z głównych zalet Kotlina jest lepsza obsługa wartości null, co pomaga programistom uniknąć wielu błędów związanych z niezainicjalizowanymi zmiennymi. Kotlin również posiada bardziej zwięzłą i czytelną składnię niż Java, co przekłada się na krótszy i bardziej przejrzysty kod. Język ten wprowadza również wyrażenia lambda oraz funkcje jednoliniowe, które ułatwiają programowanie funkcyjne i zwiększają elastyczność tworzenia aplikacji na Androida. Dzięki temu Kotlin zdobył dużą popularność wśród programistów, którzy doceniają jego nowoczesne podejście i możliwości w tworzeniu aplikacji na platformę Android.

Zarówno Java, jak i Kotlin są używane do tworzenia aplikacji na system Android i mają swoje zalety i zastosowania. Programiści mają możliwość wyboru języka, który najlepiej odpowiada ich preferencjom i doświadczeniu. Java, jako język bardziej ustalony i dobrze zintegrowany z platformą Android, nadal jest szeroko wykorzystywana w istniejących projektach. Z drugiej Kotlin oferuje nowoczesne rozwiązania stronv. i ułatwienia programistyczne, co sprawia, że jest atrakcyjnym wyborem dla nowych projektów i programistów poszukujących innowacyjnych narzędzi i możliwości tworzenia aplikacji Android [7].

Tworzenie oprogramowania to skomplikowany proces, który wymaga nie tylko podstawowej wiedzy z zakresu programowania. Wiele czynników wpływa na stworzenie nawet najprostszej aplikacji. Jednym z kluczowych elementów jest posiadanie odpowiedniego sprzętu. Następnie konieczne jest zdecydowanie, w jakiej technologii chcemy tworzyć nasz program. Możliwości są liczne - można tworzyć aplikacje internetowe, desktopowe czy też mobilne. Wybór odpowiedniego języka programowania oraz narzędzi programistycznych jest również istotny dla pomyślnego wdrożenia aplikacji.

Narzędzia programistyczne to programy komputerowe, które służą do tworzenia, modyfikacji, testowania i konserwacji oprogramowania. Wśród programistów bardzo popularne są edytory kodu, kompilatory, zintegrowane środowiska interpretery oraz programistyczne (IDE) [8]. Wiele osób korzysta głównie z tych ostatnich, ponieważ oferują one wiele funkcjonalności w jednym miejscu. Zadania, które można wykonywać za pomocą tych narzędzi, obejmują kompilację kodu źródłowego, generowanie baz danych, edycję kodu oraz tworzenie zasobów aplikacji, takich jak okna dialogowe, raporty, obrazy czy ikony. Przykładami popularnych IDE są Microsoft Visual Studio, IntelliJ IDEA, Android Studio, Code::Blocks oraz Eclipse [9].

Studio środowisko Android to oficjalne dedykowane systemowi Android. programistyczne Zostało oparte na oprogramowaniu IntelliJ IDEA i jest szeroko uznawane za jedno z najbardziej profesjonalnych narzędzi do tworzenia aplikacji mobilnych na platformę Android. Środowisko to oferuje wiele zaawansowanych funkcji, w tym integrację z systemem budowania Gradle, wsparcie dla języków programowania takich jak Java, C++ i Kotlin, oraz narzędzia do debugowania, profilowania i testowania aplikacji. Android Studio umożliwia tworzenie aplikacji dla różnych urządzeń z systemem Android, jak smartfony, tablety, telewizory i inne. Jest również zintegrowane z platformą Google Cloud, co umożliwia korzystanie z usług chmurowych w aplikacjach. Studio dostarcza wygodne Android środowisko programistyczne, które ułatwia proces tworzenia aplikacji mobilnych na platforme Android [5].

## II. INTERFEJS UŻYTKOWNIKA APLIKACJI E-PAPRYKARZ

Strona, na którą użytkownik spogląda tuż po

uruchomieniu aplikacji mobilnej, stanowi kluczowy element każdego oprogramowania. W przypadku aplikacji E-Paprykarz, menu główne składa się z różnych komponentów, które współtworzą tę stronę. Wśród tych elementów znajduje się górny pasek narzędzi, logo wraz z tytułem aplikacji, menu składające się z sześciu kafelków, z których każdy zawiera nazwę modułu oraz ikonę, a także dolne menu (Rys. 2).

Głównym elementem nawigacyjnym aplikacji E-Paprykarz jest menu główne, które składa się z sześciu bloków o następujących tytułach: Kalkulatory ogrodnicze, Notatki gospodarstwa, Zabiegi pielęgnacyjne, Kontrola nawadniania, Dziennik dochodów oraz Zapisane lokalizacje. Każdy blok zawiera nazwę modułu oraz ikonę reprezentującą ten moduł. Ważnym aspektem jest fakt, że kafelki mają kształt liści, co ma symbolizować związki aplikacji z motywem roślinnym, przyrodą i wiejskim stylem życia. Po kliknięciu na wybrany kafelek użytkownik zostaje przeniesiony do odpowiadającej mu sekcji. Każda sekcja podzielona jest na trzy mniejsze podmoduły. Aplikacja E-Paprykarz oferuje szeroką gamę funkcji, a każdy moduł skupia się na innej dziedzinie związaną z ogrodnictwem i prowadzeniem gospodarstwa.



Rys. 2. Menu główne aplikacji E-Paprykarz

Pierwszy moduł aplikacji E-Paprykarz to Kalkulatory ogrodnicze, które zawierają trzy różne kalkulatory. Pierwszym z nich jest Kalkulator pola powierzchni, który umożliwia obliczenie całkowitej powierzchni działki. Należy wpisać długość i szerokość działki oraz wybrać jednostkę, w której chcemy otrzymać wynik. Kalkulator zwraca odpowiedź w dolnej części interfejsu.

Kolejnym kalkulatorem jest Kalkulator sadzonek, który pozwala obliczyć liczbę sadzonek papryki potrzebnych do obsadzenia 1 tunelu foliowego. W tym celu należy wybrać rodzaj tunelu oraz odległość między sadzonkami, a algorytm dostosowuje odpowiedź do standardowych rozmiarów tuneli. Po naciśnięciu przycisku "OBLICZ" zostaną zwrócone zalecenia dotyczące liczby rzędów i sadzonek (Rys. 3).



Rys. 3. Kalkulator sadzonek w aplikacji E-Paprykarz

Ostatnim kalkulatorem w tym module jest Kalkulator cieczy roboczej. Pomaga on obliczyć precyzyjne ilości wody i pestycydu potrzebne do zabiegu ochrony roślin. Na początku należy wpisać wiek papryki oraz liczbę tuneli, w których przeprowadzimy zabieg. Następnie jednostkę, w jakiej mają być przedstawione zalecane dawki. Algorytm prezentuje wyniki obliczeń, takie jak ilość wody i pestycydu.

Kolejny moduł w aplikacji E-Paprykarz to Notatki gospodarstwa, który pełni rolę prostego notatnika do gromadzenia informacji związanych z uprawą papryki i życiem gospodarstwa rolnego. Moduł ten składa się z trzech podmodułów: Podstawowe daty, Nowa notatka oraz Moje notatki.

W zakładce Podstawowe daty użytkownik może zapisywać najważniejsze daty związane z uprawą papryki. Przechowywane są 5 kluczowych dat, takich jak data zasiewu, pikowania, zasadzenia, pierwszych oraz ostatnich zbiorów (Rys. 4).



Rys. 4. Sekcja Podstawowe daty w aplikacji E-Paprykarz

Moduł Notatki gospodarstwa umożliwia tworzenie prostych notatek, które zawierają ważne wydarzenia związane z życiem gospodarstwa rolnego. Aby dodać nową notatkę, użytkownik przechodzi do sekcji Nowa notatka, gdzie wypełnia formularz z tytułem, datą i zawartością notatki. Można również wybrać ikonę wektorową, która będzie reprezentować notatkę. Po wypełnieniu formularza, notatka zostaje utworzona. Przez sekcję Moje notatki użytkownik może przeglądać wszystkie zapisane notatki. Każda notatka jest wyświetlana w formie listy, a na każdym elemencie można dokonać modyfikacji lub usunięcia notatki (Rys. 5).



Rys. 5. Sekcja Moje Notatki w aplikacji E-Paprykarz

Moduł Zabiegi pielęgnacyjne w aplikacji E-Paprykarz umożliwia skuteczne zarządzanie zabiegami ochrony roślin w gospodarstwie. Ten moduł składa się z trzech sekcji: Zaplanuj zabieg, Dziennik zabiegów oraz Katalog pestycydów.

W zakładce Katalog pestycydów aplikacja zapewnia przegląd najważniejszych pestycydów stosowanych w ochronie papryki. Katalog ten jest podzielony na trzy sekcje: insektycydy, fungicydy i herbicydy. Każda sekcja zawiera listę dostępnych pestycydów, które pomagają w zwalczaniu odpowiednio owadów, grzybów i chwastów na plantacji. Po kliknięciu na przycisk SZCZEGÓŁY obok nazwy konkretnego środka, użytkownik może zobaczyć szczegółowy opis pestycydu, w tym nazwę, ilustrację opakowania, rodzaj zwalczanych szkodników, zalecaną dawkę, czas karencji i zastosowanie (Rys. 6).

W sekcji Zaplanuj zabieg użytkownik może zaplanować zabieg pielęgnacyjny w oparciu o dostępne pestycydy. Wypełniając prosty formularz, użytkownik podaje rodzaj stosowanego pestycydu, datę, godzinę i liczbę tuneli, w których będzie przeprowadzony zabieg (Rys. 7). Wiek papryki jest automatycznie obliczany na podstawie daty zasadzenia, zapisanej w module Notatki gospodarstwa. Po wybraniu środka ochrony roślin, planowany zabieg zostanie zapisany.

#### **European Student Magazine**

1923 ● 📲 🔹 🛠 🖘 4 🖘 170% 🕯 = Sezon 2022 😗	19:23 ⊕ 🖬 🔹 🗞 🗞 at 70% 🖬 🗮 Sezon 2022 🔮	
← Katalog Pestycydów	FUNGICYD Zwalczany szkodnik Szara pleśń	
	Zalecana dawka	
Insektycydy Fungicydy Herbicydy	AMISTAR 250 SC S 3 dni	
AMISTAR 250 SC SZCZEGÓŁY	Zastosowanie	
AMYLO X-WG SZCZEGÓŁY	Kontrolować rośliny systematycznie, zwłaszcza na tych stanowiskach, na których choroba	
ARMICARB SP SZCZEGÓŁY	występowała w poprzednim sezonie. Opryskiwać rośliny po stwierdzeniu objawów. Zabieg wykonywać co 7-10 dni.	
AZARIUS 250 SC SZCZEGÓLY	OSTRZEŻENIE!	
BACTERPLANT SZCZEGÓŁY	szkodzić ludziom i środowisku, przestrzegoj	
DAGONIS SZCZEGÓŁY	pestycydu z płynami ustrojowymi niezwłocznie udo się na pogotowie!	
DISCUS 500 WG SZCZEGÓŁY	KATALOG PESTYCYDÓW	
Gospodarstwo Strona ołówna O anilkani	Gospodarstvo Strona płówna O anilkacii	

Rys. 6. Katalog pestycydów w aplikacji E-Paprykarz

Aplikacja E-Paprykarz wysyła powiadomienie właścicielowi gospodarstwa godzinę przed planowanym zabiegiem, przypominając o jego wykonaniu. Powiadomienie jest wyświetlane na telefonie oraz smartwatchu (Rys. 7).



Rys. 7. Planowanie zabiegu w aplikacji E-Paprykarz

W sekcji Dziennik zabiegów użytkownik może przeglądać wszystkie zaplanowane zabiegi pielęgnacyjne. Każdy wpis zawiera informacje dotyczące rodzaju zabiegu, daty, godziny wykonania, długości karencji, stosowanego pestycydu i aktualnego statusu. Jeśli zabieg został wykonany, a czas karencji nie minął, użytkownik otrzyma ostrzeżenie o aktualnym stanie zabiegu. Przez przycisk SZCZEGÓŁY użytkownik może uzyskać więcej informacji na temat konkretnego zabiegu, w tym instrukcję przygotowania cieczy roboczej. Na tej stronie znajdują się również przyciski, które umożliwiają zmianę statusu i wykonanie zabiegu (Rys. 8).

W sekcji Dziennik zabiegów użytkownik może

przeglądać wszystkie zaplanowane zabiegi pielęgnacyjne. Każdy wpis zawiera informacje dotyczące rodzaju zabiegu, daty, godziny wykonania, długości karencji, stosowanego pestycydu i aktualnego statusu. Jeśli zabieg został wykonany, a czas karencji nie minął, użytkownik otrzyma ostrzeżenie o aktualnym stanie zabiegu. Przez przycisk SZCZEGÓŁY użytkownik może uzyskać więcej informacji na temat konkretnego zabiegu, w tym instrukcję przygotowania cieczy roboczej. Na tej stronie znajdują się również przyciski, które umożliwiają zmianę statusu i wykonanie zabiegu (Rys. 8).



Rys. 8. Dziennik zabiegów w aplikacji E-Paprykarz

Moduł Kontrola nawadniania w aplikacji E-Paprykarz skupia się na usprawnieniu procesu nawadniania plantacji papryki i składa się z trzech sekcji: Planowanie nawadniania plantacji, Dziennik nawadniania oraz Zużycie wody.

W sekcji Planowanie nawadniania plantacji można zaplanować proces nawadniania. Formularz umożliwia wprowadzenie danych takich jak wydajność pompy, która będzie używana do podlewania roślin, oraz daty planowanego nawadniania. Najważniejszym krokiem jest jednak tworzenie kolejnych tur nawadniania. Po kliknięciu przycisku oznaczonego symbolem plusa można dodać turę nawadniania, podając liczbę tuneli i czas trwania nawadniania.

W sekcji Dziennik nawadniania znajdują się wszystkie zaplanowane nawadniania plantacji. Można usunąć poszczególne pozycje lub rozpocząć nawadnianie, klikając na przycisk z ikoną kranu. Czasomierz nawadniania przedstawia aktualnie realizowaną turę, liczby tuneli, czas trwania oraz wydajność pompy. Kolor kropli wody symbolizuje status nawadniania - czarny oznacza, że tura nie została zakończona, niebieski informuje o trwającym nawadnianiu, a zielony sygnalizuje zakończenie procesu nawadniania. Aby rozpocząć nawadnianie, należy kliknąć przycisk ROZPOCZNIJ. Po zakończeniu każdej tury zostanie wysłane powiadomienie na telefon i smartwatch. Po otrzymaniu powiadomienia i wykonaniu odpowiednich czynności związanych z przełączaniem zaworów wodnych, należy kliknąć na obraz symbolizujący tę czynność. Po zakończeniu ostatniej tury zostanie wyświetlona informacja nakazująca wyłączenie pompy, a następnie należy kliknąć przycisk ZAKOŃCZ (Rys. 9).

W sekcji Zużycie wody znajduje się wykres przedstawiający miesięczne zużycie wody do nawadniania plantacji. Dane są zaokrąglone do tysięcy litrów.



Rys. 9. Realizacja procesu nawadniania przy użyciu aplikacji E-Paprykarz

Moduł Dziennik dochodów w aplikacji E-Paprykarz zajmuje się prowadzeniem finansów gospodarstwa i składa się z trzech sekcji: Sprzedaż plonów, Wydatki oraz Bilans zysków i strat.

W sekcji Sprzedaż plonów gromadzone są informacje dotyczące każdej przeprowadzonej transakcji sprzedaży papryki. Aby dodać nową sprzedaż, należy kliknać przycisk oznaczony symbolem plusa, co spowoduje pojawienie się dodawania nowej sprzedaży. formularza Należv wprowadzić dane takie jak kolor, klasa papryki, cena, tonaż, stawka VAT oraz miejsce sprzedaży. Suma zostanie automatycznie obliczona i wyświetlana przy każdej pozycji (Rys. 10). W sekcji Wydatki można tworzyć kolejne wpisy w podobny sposób. Należy wybrać kategorię, datę oraz kwotę wydatków. Każdy wpis, bez względu na to, czy dotyczy dochodu czy wydatku, można edytować lub usunąć.



Rys. 10. Dodawania nowej transakcji sprzedaży plonów w aplikacji E-Paprykarz

W sekcji Bilans zysków i strat znajduje się statystyka budżetu gospodarstwa. Na pierwszej stronie tej sekcji znajduje się wykres kołowy, który przedstawia dochody i wydatki w tysiącach złotych w danym sezonie. Poniżej wykresu znajdują się dwa przyciski, które przenoszą użytkownika do stron szczegółowo analizujących dochody i wydatki.

W sekcji Szczegóły zysków znajduje się tabela analizująca sprzedaż papryki pod względem różnych parametrów, takich jak kolor i klasa. Tabela zawiera także informacje na temat średniej ceny papryki, średniego zysku, średniego zbioru z tunelu oraz sumy zarobków i zbiorów. Pola średni zysk i średni zbiór są obliczane na podstawie ustalonej liczby tuneli w sekcji Gospodarstwo. Kolejne strony modułu przedstawiają tę samą analizę dochodów w postaci wykresów. Aplikacja zawiera dwa wykresy kołowe, które prezentują procentowy udział poszczególnych kolorów i klas papryki uprawianych w gospodarstwie. Jest także wykres słupkowy, który przedstawia wielkość zbiorów w kolejnych miesiącach sezonu, oraz wykres liniowy, który pokazuje kształtowanie się cen papryki w zależności od koloru i klasy w kolejnych miesiącach sezonu. Na wykresie liniowym jest również zaznaczona średnia cena papryki w danym sezonie. Parametry wykresu można zmieniać, klikając na przycisk znajdujący się pod wykresem i dokonując odpowiednich zmian koloru i klasy papryki (Rys. 11).

Zestawienie wydatków również jest przedstawione w formie tabelarycznej. Każdy wiersz odpowiada określonej kategorii wydatków, a na końcu tabeli znajduje się suma wszystkich wydatków. Dane te są również przedstawione na wykresie słupkowym.



Rys. 11. Zestawienie sprzedaży papryki w aplikacji E-Paprykarz

Moduł Zapisane lokalizacje w aplikacji E-Paprykarz umożliwia kontrolowanie lokalizacji placówek, z którymi gospodarstwo współpracuje lub korzysta z ich usług. Składa się on z trzech sekcji: Nowa lokalizacja, Moje lokalizacje oraz Rynki hurtowe na mapie.

Aby dodać nową lokalizację, należy przejść do sekcji Nowa lokalizacja i wpisać nazwę lokalizacji w odpowiednim polu. Następnie należy kliknąć na ikonę globusa. Po chwili oczekiwania, Location Manager wczyta współrzedne geograficzne. Poprawność działania zostanie potwierdzona przez animację obracającego się globusa oraz wyświetlenie uzyskanych współrzędnych. Wszystkie dodane lokalizacje będą przechowywane w sekcji Moje lokalizacje. Aby zobaczyć dane miejsce na mapie, wystarczy kliknąć przycisk w formie mapki. Wyświetli się mapa z pinezką reprezentującą daną lokalizację. Dodatkowo, każdą dodaną lokalizację można usunąć (Rys. 12).





Rys. 12. Dodawanie nowej lokalizacji w aplikacji E-Paprykarz

Ostatnia sekcja, Rynki hurtowe na mapie, to mapa zawierająca szereg pinezek, z których każda odpowiada konkretnemu rynkowi hurtowemu w Polsce. Aby zapoznać się ze szczegółami danego miejsca, należy kliknąć na odpowiednią pinezkę. Wyświetli się okno dialogowe zawierające nazwę giełdy rolniczej, dokładny adres, e-mail oraz numer telefonu (Rys. 13).



Rys. 13. Sekcja rynki hurtowe na mapie w aplikacji E-Paprykarz

Archiwum w aplikacji E-Paprykarz służy do przechowywania danych z poprzednich sezonów (Rys. 14). Dane wyświetlane w poszczególnych modułach są filtrowane w zależności od aktualnego sezonu, zgodnie z zasadą, że rok dodania konkretnej pozycji musi być zgodny z wartością bieżącego sezonu.



Rys. 14. Archiwum aplikacji E-Paprykarz

#### **III. WYKONANIE APLIKACJI**

Aplikacja E-Paprykarz została stworzona jako natywna aplikacja mobilna dla systemu Android. Do implementacji kodu obsługującego różne klasy i bazę danych użyto języka Java. Tworzenie układów interfejsu użytkownika odbywało się poprzez język znaczników XML. Działania związane z realizacją projektu doprowadziły do powstania kompletnego oprogramowania. Cały proces implementacji odbył się w środowisku programistycznym Android Studio w wersji 2021.1.1.

Android Studio to zaawansowane zintegrowane środowisko programistyczne, które oferuje zalety takie jak kolorowanie składni kodu, wykrywanie błędów i ułatwienia w pracy. Jedną z kluczowych funkcji jest możliwość testowania tworzonego oprogramowania na wbudowanym emulatorze. Dzięki temu programiści mogą tworzyć i uruchamiać aplikacje na swoich komputerach. Emulator dostarcza predefiniowane konfiguracje różnych urządzeń, takich jak telefony, tablety, smartwatche i telewizory, które działają na systemie Android [10].

Podczas tworzenia aplikacji E-Paprykarz zdecydowano się na testowanie na fizycznych urządzeniach. W tym celu włączono w telefonie Samsung Galaxy A51 tryb programisty, który umożliwia włączenie bezprzewodowego debugowania. Po wykonaniu odpowiednich instrukcji, Android Studio może połączyć się z telefonem za pomocą sieci Wi-Fi, umożliwiając instalację aplikacji na rzeczywistym urządzeniu.

Android Studio jest również zintegrowane z serwisem GitHub, który służy do przechowywania kodu źródłowego w repozytoriach Git. To narzędzie umożliwia zarządzanie historią kodu i współpracę wielu programistów nad jednym projektem. Dzięki GitHubowi ryzyko utraty kodu źródłowego z powodu skasowania plików lub kradzieży sprzętu znacznie maleje, ponieważ kod jest przechowywany na serwerze. Właściciel może pobrać swój kod na dowolne urządzenie.

Tworzenie funkcjonalności od podstaw przy użyciu języka programowania jest długotrwałym i męczącym procesem. W przypadku aplikacji E-Paprykarz, zastosowano 3 biblioteki, które ułatwiają programowanie niektórych funkcji, takich jak system map dla aplikacji mobilnych.

Biblioteka Maps SDK Android umożliwia dodanie gotowych map dedykowanych dla systemu Android do aplikacji. Aby zaimplementować mapę, konieczne jest posiadanie konta rozliczeniowego Google, z którego pobierany jest unikalny klucz API umożliwiający dostęp do map Google [11]. Dzięki tej bibliotece, w module Zapisane lokalizacje, aplikacja wyświetla rynki hurtowe w Polsce oraz lokalizacje dodane przez użytkownika na mapie (Rys. 13).

MPAndroidChart to kolejna użyta biblioteka, umożliwiająca tworzenie różnego rodzaju animowanych wykresów, takich jak wykresy liniowe, słupkowe, kołowe, bąbelkowe [12]. Aby użyć tej biblioteki, należy dodać odpowiednią zależność w pliku Gradle. Wykresy te są wykorzystane w module Dziennik dochodów oraz Kontrola nawadniania (Rys 11).

SuperToasts to biblioteka, która rozszerza funkcjonalność o toasty, czyli krótkie wiadomości wyświetlanych jako dymki w aplikacji. Ta biblioteka umożliwia tworzenie bardziej rozbudowanych toastów niż standardowe toasty oferowane przez Androida [13]. W aplikacji E-Paprykarz, stylizowane toastv są wykorzystywane w różnych częściach aplikacji do informowania użytkownika o błędach w formularzu, niepowodzeniach lub sukcesach operacji. Pojawiają się w dolnej części aplikacji.

Aby aplikacja E-Paprykarz była wiarygodną bazą wiedzy, podczas jej tworzenia korzystano z rzetelnych źródeł. Na przykład, moduł "Zabiegi pielęgnacyjne" korzysta z katalogu pestycydów, który zawiera listę legalnych środków ochrony roślin w Polsce. Dla zachowania precyzji i uniknięcia błędów, dane z tego katalogu są oparte na papierowej wersji "Programu Ochrony Roślin Warzywnych na rok 2021" [14].

Odpowiedni wygląd graficzny aplikacji jest również ważnym elementem. Aplikacja E-Paprykarz została zaprojektowana z dbałością o szczegóły graficzne, aby przyciągać użytkowników poprzez dopasowaną oprawę wizualną. Wykorzystano szereg ikon wektorowych, w tym standardowe ikony udostępniane przez Android Studio oraz ikony pobrane ze strony Freepik [15].

Implementacja aplikacji mobilnej to skomplikowany proces, który wymaga współpracy wielu elementów. Tworzenie i rozwijanie aplikacji E-Paprykarz zajęło wiele godzin projektowania, programowania i testowania. Z tego względu zostaną zaprezentowane tylko dwa przykłady implementacji.

Pierwszy z nich to implementacja powiadomienia z wbudowanym minutnikiem. Musi on zostać zaimplementowany jako usługa działająca w tle, aby kontynuować swoje działanie nawet po zamknięciu aplikacji. W tym celu została utworzona klasa WateringService dziedzicząca po klasie Service, która musi być zarejestrowana w pliku AndroidManifest.xml, aby program wiedział o jej istnieniu.

Aby zrealizować minutnik, tworzony jest obiekt klasy CountDownTimer, który umożliwia odliczanie czasu w milisekundach. Konstruktor tego obiektu przyjmuje czas trwania odliczania i interwał, określający czas między kolejnymi tikami.



Rys. 15. Implementacja metody onTick

W metodzie onTick odliczania, po każdym interwale czasomierza, zwracana jest liczba sekund pozostałych do zakończenia odliczania. Ta informacja jest przekazywana jako broadcastowa wiadomość do głównej aktywności w celu aktualizacji interfejsu użytkownika.

Po tym etapie tworzone jest powiadomienie, które zawiera dużą ikonę oraz tytuł, przedstawiający aktualną turę podlewania. Dane te są pobierane za pomocą obiektu klasy SharedPreferences. Zawartość powiadomienia reprezentuje wartość odliczania minutnika. Pozostały czas jest przekazywany do metody, która zwraca wynik w formacie "minuta : sekunda". Powiadomienie ma również ustawiony wysoki priorytet ważności i jest publikowane za pomocą metody notify (Rys. 15).

Kolejny przykład implementacji to zaprezentowanie tworzenia mapy z lokalizacjami pobranymi z bazy danych. W tym celu potrzebny jest klucz API, który umożliwia dostęp do usługi. Wygenerowany klucz należy wpisać w pliku AndroidManifest.xml w sekcji meta-data. Ważnym krokiem jest również przyznanie uprawnień aplikacji do dostępu do lokalizacji przybliżonej i dokładnej za pomocą usługi GPS i Wi-Fi.

Do przedstawienia rynków hurtowych na mapie została stworzona klasa MarketsOnMapFragment. Aby zaimplementować mapy Google, klasa ta musi implementować metody z interfejsu OnMapReadyCallback.

W metodzie onMapReady otwierane jest połączenie z bazą danych. Następnie w pętli odczytywane są informacje dotyczące pozycji z tabeli. Na podstawie współrzędnych geograficznych tworzony jest obiekt klasy LatLng, który przyjmuje szerokość i długość geograficzną. Kolejne pinezki są dodawane do mapy za pomocą metody addMarker z klasy GoogleMap. Informacje, które muszą zostać podane to współrzędne geograficzne (LatLng), tytuł lokalizacji odczytany z bazy danych i indywidualne ustawienia dotyczące koloru pinezki (Rys. 16).

20	arte Anto ounabkeank(andreuab) (
	locationMap = googleMap;
	DataBaseHelper db = new DataBaseHelper(context);
	Cursor c = db.getHarkets();
	while (c.moveToNext())
	1
	<pre>String name = c.getString(c.getColumnIndexOrThrow(DataBaseNames.MarketItem.COLUMN_NAME));</pre>
	<pre>double latitude = c.getDouble(c.getColumnIndexOrThrow(DataBaseNames.MarketItem.COLUMN_LATITUDE));</pre>
	<pre>double longitude = c.getDouble(c.getColumnIndexOrThrow(DataBaseNames.MarketItem.COLUMN_LONGITUDE));</pre>
	Lating location = new Lating(latitude, longitude);
	Marker marker = locationMap.addMarker(nem MarkerOptions().position(location).title(name)
	.icon(BitmapDescriptorFactory.defaultHarker(BitmapDescriptorFactory.HUE_BLUE)));
	marker.hideInfoWindow();
	locationMap.setOnMarkerClickListener(marker1 -> {
	<pre>String title = marker1.getTitle();</pre>
	openMarketDialog(title);
	<pre>marker1.setTitle("");</pre>
	return false;
	));
	3
	if (ActivityCompat.checkSelfPermission(context, Manifest.permission.ACCESS_FINE_LOCATION) !=
	PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(context,
	Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
	ShowToast toast = new ShowToast();
	toast.showErrorToast(context, attention: "Nie uzyskano odpowiednich zezwoleń!", R.drawable.imoge_attention)
	3
	locationMap.setMyLocationEnabled(true);
	locationMap.getUiSettings().setMyLocationButtonEnabled(true);
	locationMap.animateCamera(CameraUpdateFactory.newLotingZoom(new Lating( = 52.06, vt 19.2), = 5.8f));

Rys. 16. Implementacja metody onMapReady

#### IV. WNIOSKI

Celem przedsięwzięcia było stworzenie i wdrożenie aplikacji mobilnej, która wspomaga uprawę papryki pod osłonami. Cel ten został osiągnięty poprzez wykonanie dedykowanej aplikacji dla systemu Android, która zawiera wszystkie zaplanowane funkcje i została przetestowana w rzeczywistych warunkach w gospodarstwie rolnym w sezonie 2022.

Aplikacja E-Paprykarz wyróżnia się spośród innych rozwiązań rolniczych, które zazwyczaj obejmują pielęgnację wielu warzyw jednocześnie. Jest to jedyna mobilna aplikacja skoncentrowana na szczegółowej uprawie papryki. Autor pracy czerpał inspirację z własnego doświadczenia jako producent papryki. W regionie południowego Mazowsza, gdzie produkcja papryki dynamicznie rośnie, natomiast brakuje narzędzi wspomagających uprawę tego warzywa. Testy aplikacji w praktyce potwierdziły, że przyczynia się ona do usprawnienia wielu czynności w gospodarstwie rolnym.

Podczas implementacji zapoznano się z różnymi możliwościami tworzenia aplikacji mobilnych. Android Studio, zintegrowane środowisko programistyczne, okazało się nieocenionym narzędziem, oferującym wsparcie w edycji kodu, wykrywaniu błędów i emulacji aplikacji na wirtualnym telefonie. Poznanie Android Studio jest niezbędnym krokiem dla wszystkich, którzy chcą rozpocząć przygodę z programowaniem mobilnym.

Stworzona aplikacja to natywna aplikacja mobilna dla systemu Android. W przyszłości warto rozważyć wykorzystanie narzędzi wspomagających programowanie wieloplatformowe, takich jak iOS, aby poszerzyć grupę użytkowników docelowych.

## THE PRACTICAL APPLICATION OF MOBILE TECHNOLOGY IN THE CULTIVATION OF PEPPERS UNDER COVER

Abstract: This research publication aims to present a mobile application to support the work of a farm specialising in the cultivation of peppers under cover. The software has been developed as a native mobile application for Android. It is a kind of assistant for the grower, which keeps notes on the various activities carried out on the farm. A SQLite relational database model was used to enable this functionality. The first chapter of the thesis focuses on a theoretical examination of the Android operating system. The author analyses its structure, presents the programming languages and developer tools dedicated to the creation of this type of application. In the next chapter, the user interface of the application is presented, and the basic tasks to be fulfilled by the created E-Paprikarz application are described. In the last chapter, the author describes the implementation process based on two source code examples. The application has also been tested under real-world conditions with positive results in the process of growing peppers on a farm in the 2022 season.

Key words: Android, mobile app, pepper, agriculture, phone

## BIBLIOGRAFIA

[1] Wołoszyn J., Porada K., (2023) "Analiza porównawcza algorytmów całkowania numerycznego zaimplementowanych w języku Java dla systemu Android". Informatyka XXI wieku Wyzwania i dylematy. Wydanie 1. Str. 93-110.

- [2] Honcharenko K., Smołka J., (2019) "Analiza rozwoju środowiska uruchomieniowego systemu Android". Journal Computer Sciences Institute. Str. 246.
- [3] Singh R., (2014) "An Overview of Android Operating System and Its Security Features". Journal of Engineering Research and Applications. Str. 519-521.
- [4] Górski G., Nowacki M., (2018) "Anazliza zagrożeń bezpieczeństwa dla współczesnych platform mobilnych". Zeszyty naukowe Wydziału Elektroniki i Informatyki Politechniki Koszalińskiej. Str. 49.
- [5] Android Developers. Documentation | Android Developers; [2 lipca 2023]. <u>https://developer.android.</u> <u>com/docs</u>.
- [6] Hagos T., (2018) "Learn Android Studio 3: Efficient Android App Development". Str. 5.
- [7] Sulowski D., Kozieł G., (2019) "Analiza porównawcza języków Kotlin i Java używanych do tworzenia aplikacji na system Android. Journal Computer Sciences. Str. 354.
- [8] Wikipedia, wolna encyklopedia. Narzędzie programistyczne 6 czerwca 2010 [2 lipca 2023]. https://pl.wikipedia.org/wiki/Narzędzie\_programistyc zne.
- [9] Wikipedia, wolna encyklopedia. Zintegrowane środowisko programistyczne; 23 lipca 2002 [2 lipca 2023]. <u>https://pl.wikipedia.org/wiki/Zintegrowane\_śr</u>odowisko programistyczne.
- [10] Android Developers. Run apps on the Android Emulator | Android Studio | Android Developers; [2 lipca 2023]. <u>https://developer.android.com/studio/ru</u> <u>n/emulator</u>.
- [11] Google for Developers. Maps SDK na Androida omówienie | Maps SDK for Android | Google for Developers; [2 lipca 2023].https://developers.google. com/maps/documentation/androidsdk/overview?hl=pl.
- [12] GitHub. GitHub PhilJay/MPAndroidChart: A powerful Android chart view / graph view library, supporting line- bar- pie- radar- bubble- and candlestick charts as well as scaling, panning and animations.; [2 lipca 2023]. <u>https://github.com/PhilJa</u> y/MPAndroidChart#documentation.
- [13] GitHub JohnPersano/SuperToasts: A library that extends the Android toast framework.; [2 lipca 2023. <u>https://github.com/JohnPersano/SuperToasts</u>.
- [14] Zespół redakcyjny PlantPress, (2021) "Program Ochrony Roślin Warzywnych na rok 2021". Str. 176-182.
- [15] Freepik. Freepik: darmowe klipy wideo, wektory, zdjęcia i pliki PSD; [2 lipca 2023]. https://pl.freepik.co m.