

Uniwersytet Morski w Gdyni



Projekt – Programowanie interfejsów graficznych:

Temat: System zarządzania szklarnią

*Adam Bieszk, 47461
Paweł Frąckowiak, 47620
III rok Informatyka
AliM grupa I*

<i>Imię i nazwisko:</i> Adam Bieszk Paweł Frąckowiak	<i>Numer grupy:</i> 1	<i>Specjalizacja:</i> AliM
--	--------------------------	-------------------------------

1. Opis projektu:

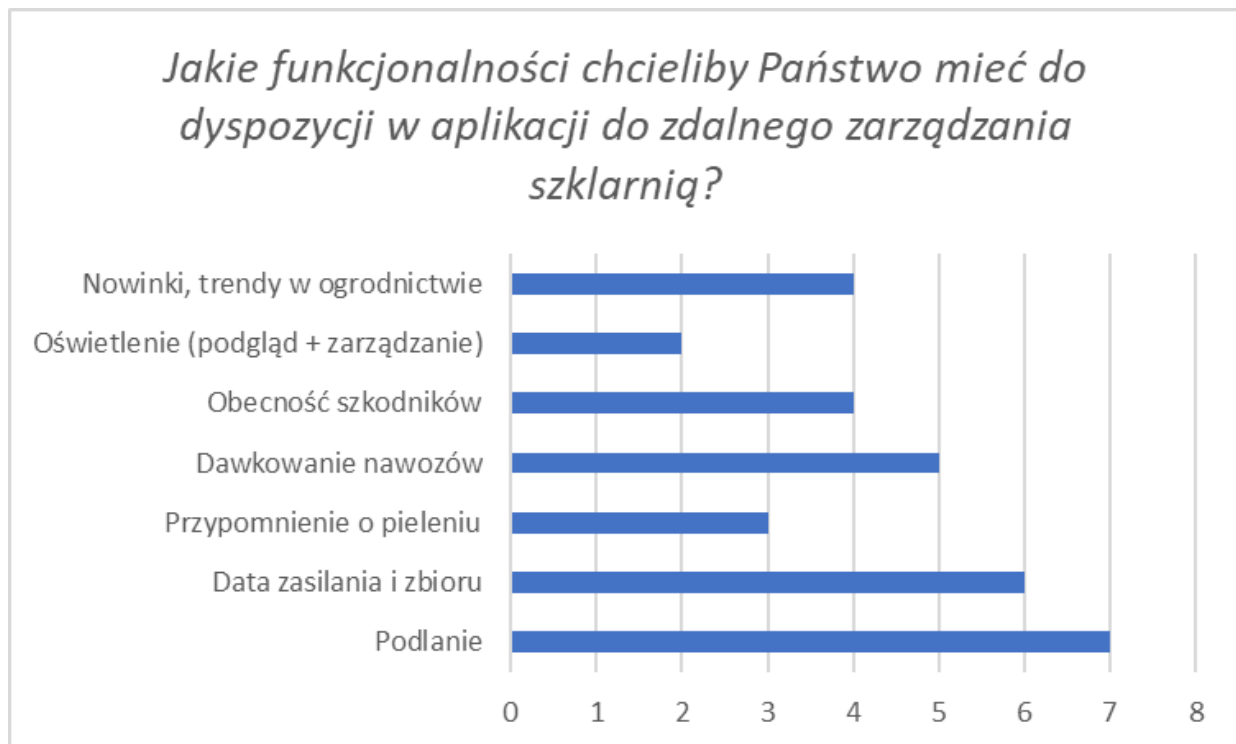
Celem projektu jest stworzenie aplikacji w programie MATLAB, która wspomaga i automatyzuje czynności związane z zarządzaniem szklarnią m.in. zdalne podlewanie i system ostrzeżeń o stanie nawodnienia gleby u poszczególnych roślin.

a. Analiza potrzeb użytkownika:

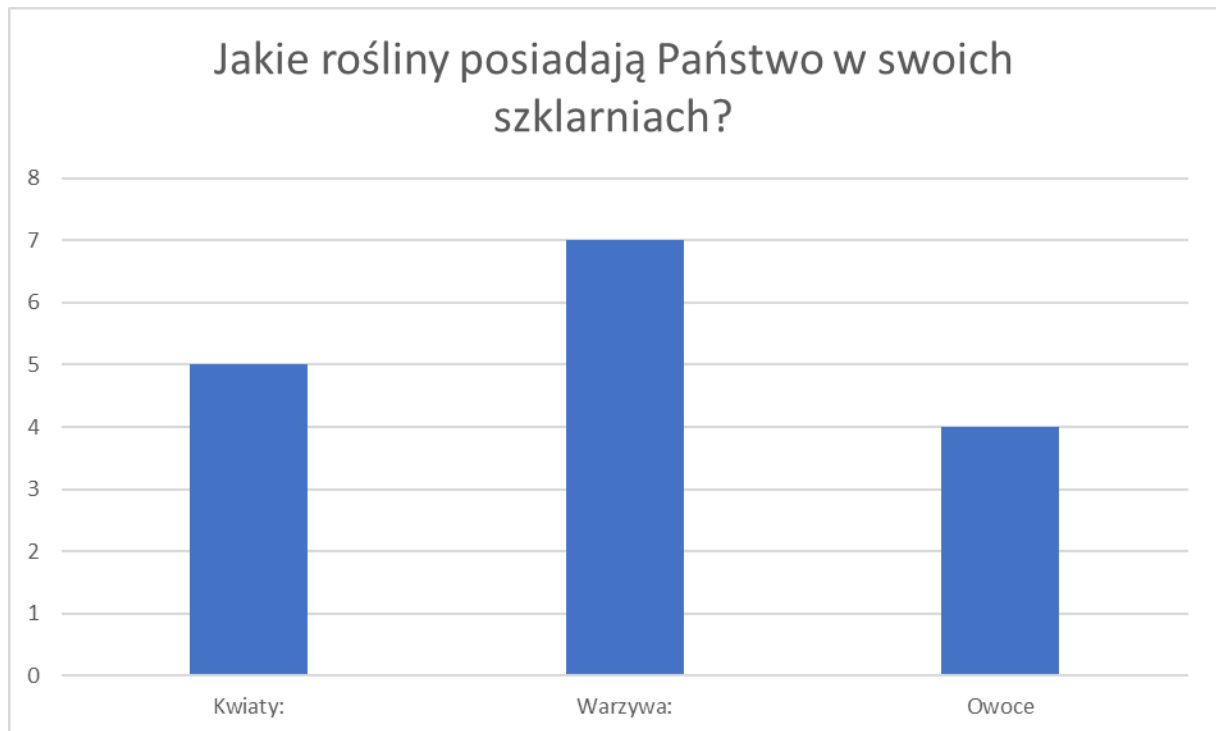
Naszym klientem docelowym są właściciele szklarni i pasjonaci ogrodnictwa, którzy chcą w prosty sposób zarządzać swoją szklarnią, kontrolować stan nawodnienia roślin oraz na bieżąco w czasie rzeczywistym otrzymywać w przejrzysty sposób informacje na ich temat. Aplikacja jest przeznaczona na komputery i laptopy, które będą w sposób zdalny komunikować się z systemem zarządzania szklarnią. Wcześniej przygotowane czujniki wsadzone w glebę będą przekazywały informacje do aplikacji i na podstawie parametrów przez nie przesłanych będzie można wyliczyć, czy jest to idealna pora, aby włączyć system nawadniania roślin. Dzięki temu mamy pełną kontrolę nad swoimi roślinami nawet do nich nie podchodząc. Również w interfejsie szklarni jest miejsce przeznaczone na kamerę, która w czasie rzeczywistym przesyła obraz. Zatem jeżeli z roślinami działałoby się coś nie tak, to na podglądzie możemy szybko sprawdzić co się dzieje i czy system działa poprawnie.

W ramach analizy potrzeb użytkownika została przeprowadzona ankieta, podczas której badający odpowiadali na następujące pytania:

1. *Jakie funkcjonalności chcieliby Państwo mieć do dyspozycji w aplikacji do zdalnego zarządzania szklarnią?*



2. *Jakie rośliny posiadają Państwo w swoich szklarniach?*



b. Opis funkcjonalności:

- Wyświetlanie podstawowych oraz szczegółowych informacji o roślinach
- Podgląd stanu nawodnienia, temperatury zewnętrznej i wewnętrznej
- Podaje datę następnego podlania i ostrzega przed wysuszeniem
- Możliwość dodawania, usuwania i edytowania roślin
- Możliwość podlania wszystkich roślin
- Zapisywanie danych do pliku tekstowego
- Odczytywanie danych z pliku tekstowego
- Zapamiętywanie wprowadzanych danych
- Możliwość zasięgnięcia pomocy w okienku "pomoc"
- Możliwość wyświetlania dodatkowych informacji po kliknięciu w dany rekord
- Możliwość zmiany nazwy i parametrów szklarni
- Możliwość odświeżenia widoku
- Wyświetlanie prawidłowej daty, godziny i pory roku na podstawie miesiąca
- Możliwość wstawienia zdjęcia w miejsce kamerki dla umilenia widoku :)

c. Projekt graficzny (pierwotny wzór aplikacji):

Eksportuj

Ustawienia

Pomoc

Data i godzina: 03-11-2022 09:30

Nazwa szklarni: Moja szklarnia

Pora roku: <Pora roku odczytana z daty>

☒ Kwiaty ☐ Warzywa ☐ Owoce

Stan	Lista roślin	Data następnego podlania
	Roślina1	03-11-2022
	Roślina2	05-11-2022
	Roślina3	05-11-2022
	Roślina4	04-11-2022

Podlej

Model szklarni

Temperatura wewnątrz: x°C

Temperatura na zewnątrz: x°C

Opis rośliny

Okno główne aplikacji

Parametry szklarni

Wymiar	Wartość
Nazwa	Moja szklarnia
Długość	4,2m
Szerokość	3,6m
Wysokość	2,4m

Nowa

Edytuj

Usuń

Lista wszystkich roślin

Wczytaj z pliku

Zapisz

Szczegółowe dane

Dodaj

Okienko szczegółów

Diody - legenda

Stan	Oznaczenie
	Usycha (należy podleć dzisiaj)
	Umiarkowany (należy podleć następnego dnia)
	Nawodniony (nie trzeba podlewać)

Eksportuj

Ustawienia

Pomoc

<Pomoc dotycząca polecenia Eksportuj>

<Pomoc dotycząca dostępnych ustawień>

<Otwiera to okienko z Pomocą>

Okienko Pomocy

1. Aplikacja:

a. Komponenty:

```
% Properties that correspond to app components
properties (Access = public)
    UIFigure                                matlab.ui.Figure
    OpisRoslinyLabel                        matlab.ui.control.Label
    OpisRoslinyTextArea                    matlab.ui.control.TextArea
    PodlejRoslinyKtoreTegoWymagajaButton  matlab.ui.control.Button
    PorarokuEditField                      matlab.ui.control.EditField
    PorarokuEditFieldLabel                 matlab.ui.control.Label
    NazwaszklarniEditField                 matlab.ui.control.EditField
    NazwaszklarniEditFieldLabel            matlab.ui.control.Label
    DataigodzinaEditField                  matlab.ui.control.EditField
    DataigodzinaEditField_2Label           matlab.ui.control.Label
    TabGroup                               matlab.ui.container.TabGroup
    RolinyTab                               matlab.ui.container.Tab
    StanRosliny14Lamp                      matlab.ui.control.Lamp
    StanRosliny13Lamp                      matlab.ui.control.Lamp
    StanRosliny12Lamp                      matlab.ui.control.Lamp
    StanRosliny11Lamp                      matlab.ui.control.Lamp
    StanRosliny10Lamp                      matlab.ui.control.Lamp
    StanRosliny9Lamp                       matlab.ui.control.Lamp
    StanRosliny8Lamp                       matlab.ui.control.Lamp
    StanRosliny7Lamp                       matlab.ui.control.Lamp
    StanRosliny6Lamp                       matlab.ui.control.Lamp
    StanRosliny5Lamp                       matlab.ui.control.Lamp
    StanRosliny4Lamp                       matlab.ui.control.Lamp
    StanRosliny3Lamp                       matlab.ui.control.Lamp
    StanRosliny2Lamp                       matlab.ui.control.Lamp
    RoslinyTable                           matlab.ui.control.Table
    StanRosliny1Lamp                       matlab.ui.control.Lamp
    PomocButton                            matlab.ui.control.Button
    SzczegolyButton                        matlab.ui.control.Button
    OdswiezButton                          matlab.ui.control.Button
    ModelszklarniPanel                     matlab.ui.container.Panel
    TemperaturaWewnatrzEditField            matlab.ui.control.NumericEditField
    TemperaturaWewnatrzEditFieldLabel       matlab.ui.control.Label
    TemperaturaZewnatrzJednostkaLabel       matlab.ui.control.Label
    TemperaturaWewnatrzJednostkaLabel       matlab.ui.control.Label
    TemperaturaZewnatrzEditField            matlab.ui.control.NumericEditField
    TemperaturanazewnatrzEditFieldLabel     matlab.ui.control.Label
    Image                                  matlab.ui.control.Image
end
```

Zmienne globalne (główne okno)

- *OpisRosliny* - pokazanie opisu pojedynczej rośliny
- *PoraRoku* - pokazanie obecnej pory roku
- *NazwaSzklarni* - pokazanie obecnej szklarni
- *DataiGodzina* - pokazanie obecnej daty i godziny
- *RoslinyTab* -> lista roślin
- *StanRoslinyNLamp* -> pokazanie stanu nawodnienia N-tej rośliny
- *TemperaturaWewnatrz* - wyświetlenie temperatury panującej wewnątrz szklarni
- *TemperaturaZewnatrz* - wyświetlenie temperatury panującej na zewnątrz szklarni

```

% Properties that correspond to app components
properties (Access = public)
    UIFigure                matlab.ui.Figure
    UITable                 matlab.ui.control.Table
    SzczegoloweDanePanel    matlab.ui.container.Panel
    ZawartoscPlikuLabel     matlab.ui.control.Label
    ZapiszDane              matlab.ui.control.Button
    EdytujDane              matlab.ui.control.Button
    ZawartoscPlikuTextArea  matlab.ui.control.TextArea
    ZainportujDaneButton    matlab.ui.control.Button
    NazwaPlikuEditField     matlab.ui.control.EditField
    NazwaplikuLabel         matlab.ui.control.Label
    ParametryszklarniPanel  matlab.ui.container.Panel
    WczytajSzklarnieButton  matlab.ui.control.Button
    ZapiszSzklarnieButton   matlab.ui.control.Button
    EdytujSzklarnieButton   matlab.ui.control.Button
    WysokoscJednostkaLabel  matlab.ui.control.Label
    SzerokoscJednostkaLabel matlab.ui.control.Label
    DlugoscJednostkaLabel   matlab.ui.control.Label
    WysokoscEditField       matlab.ui.control.NumericEditField
    WysokoEditFieldLabel    matlab.ui.control.Label
    SzerokoscEditField      matlab.ui.control.NumericEditField
    SzerokomLabel           matlab.ui.control.Label
    DlugoscEditField        matlab.ui.control.NumericEditField
    DugomLabel              matlab.ui.control.Label
    NazwaEditField          matlab.ui.control.EditField
    NazwaEditFieldLabel     matlab.ui.control.Label
end

```

Zmienne globalne (okienko szczegółów)

- Nazwa, Dlugosc, Szerokosc, Wysokosc - wyświetlenie parametrów szklarni (nazwa, długość, szerokość, wysokość)
- WczytajSzklarnieButton - wczytanie parametrów szklarni z pliku
- EdytujSzklarnieButton - włączenie możliwości edycji parametrów szklarni
- ZapiszSzklarnieButton - zapis parametrów szklarni do pliku

b. Fragmenty kodu z komentarzem:

- *Wybranie właściwej nazwy szklarni z pliku:*

```

% Wybranie właściwej nazwy szklarni z pliku
if (exist('szklarnia.txt','file') == 2)
    fileID = fopen('szklarnia.txt');
    dane = fscanf(fileID, '%s');
    params = strsplit(dane, ',');
    fclose(fileID);
    app.NazwaszklarniEditField.Value = string(params(1));
else
    app.NazwaszklarniEditField.Value = "Moja szklarnia";
end

```

- Wybranie aktualnej daty i godziny: - `datetime()`

```
% Wybranie aktualnej daty i godziny
date = datestr(datetime('now'), 'dd/mm/yy HH:MM')
app.DataigodzinaEditField.Value = date;
```

- Wybranie właściwej pory roku: - `ismember()`

```
% Wybranie właściwej pory roku
mo = month(datetime('now'));
if(ismember(mo, [12,1,2]))
    app.PorarokuEditField.Value = "Zima";
elseif(ismember(mo,[3,4,5]))
    app.PorarokuEditField.Value = "Wiosna";
elseif(ismember(mo,[6,7,8]))
    app.PorarokuEditField.Value = "Lato";
elseif(ismember(mo,[9,10,11]))
    app.PorarokuEditField.Value = "Jesień";
end
```

- FINALNY EFEKT powyższego kodu:

Nazwa szklarni	Szklania2023
Data i godzina	07/02/2023 20:22
Pora roku	Zima

- Funkcja do aktualizowania diody do stanu rośliny
(jeżeli "usycha" dioda zaświeci się na czerwono
jeżeli "umiarkowane" zaświeci się na żółto, a
jeżeli "nawodnione" zaświeci się na zielono)

```
% aktualizuje kolor jednej diody
function zmienKolorLampy(app, color)
    switch strjoin(color)
        case 'usycha'
            app.Color = [1 0 0];
        case 'nawodnione'
            app.Color = [0 1 0];
        case 'umiarkowane'
            app.Color = [1 1 0];
        otherwise
            app.Color = [0 0 0];
            error('Invalid color parameter');
    end
end
```


- Funkcja do podlewania wszystkich roślin i ustala nową datę następnego podlania

```
% Button pushed function: PodlejWszystkieRoslinyButton
function PodlejWszystkieRoslinyButtonPushed(app, event)
    table = app.RoslinyTable;
    data = get(table, 'Data');
    rowscount = height(data);
    colscount = 5;
    for row = 1:rowscount
        for column = 1:colscount
            if column == 4
                data(row, column) = {'nawodnione'};
            elseif column == 5
                data(row, column) = {datestr(datetime("today")+days(4), 'dd/mm/yyyy')};
            else
                data(row, column) = data(row, column);
            end
        end
    end
    table.Data = data;

    app.StanRosliny1Lamp.Color = [0 1 0];
    app.StanRosliny2Lamp.Color = [0 1 0];
    app.StanRosliny3Lamp.Color = [0 1 0];
    app.StanRosliny4Lamp.Color = [0 1 0];
    app.StanRosliny5Lamp.Color = [0 1 0];
end
```

- Aktualizacja koloru wszystkich diod

```
% aktualizuje kolor wszystkich diód
function aktualizujDiody()
    data = app.RoslinyTable.Data;
    iloscRoslin = height(data);
    %for r = 1:iloscRoslin
    %| if (any(cellfun(@ischar, data(r,4)))) zmienKolorLampy(app.StanRoslinyLamp, data(r,4)); end
    %end
    if (any(cellfun(@ischar, data(1,4)))) zmienKolorLampy(app.StanRosliny1Lamp, data(1,4)); end
    if (any(cellfun(@ischar, data(2,4)))) zmienKolorLampy(app.StanRosliny2Lamp, data(2,4)); end
    if (any(cellfun(@ischar, data(3,4)))) zmienKolorLampy(app.StanRosliny3Lamp, data(3,4)); end
    if (any(cellfun(@ischar, data(4,4)))) zmienKolorLampy(app.StanRosliny4Lamp, data(4,4)); end
    if (any(cellfun(@ischar, data(5,4)))) zmienKolorLampy(app.StanRosliny5Lamp, data(5,4)); end
    if (any(cellfun(@ischar, data(6,4)))) zmienKolorLampy(app.StanRosliny6Lamp, data(6,4)); end
    if (any(cellfun(@ischar, data(7,4)))) zmienKolorLampy(app.StanRosliny7Lamp, data(7,4)); end
    if (any(cellfun(@ischar, data(8,4)))) zmienKolorLampy(app.StanRosliny8Lamp, data(8,4)); end
    if (any(cellfun(@ischar, data(9,4)))) zmienKolorLampy(app.StanRosliny9Lamp, data(9,4)); end
    if (any(cellfun(@ischar, data(10,4)))) zmienKolorLampy(app.StanRosliny10Lamp, data(10,4)); end
    if (any(cellfun(@ischar, data(11,4)))) zmienKolorLampy(app.StanRosliny11Lamp, data(11,4)); end
    if (any(cellfun(@ischar, data(12,4)))) zmienKolorLampy(app.StanRosliny12Lamp, data(12,4)); end
    if (any(cellfun(@ischar, data(13,4)))) zmienKolorLampy(app.StanRosliny13Lamp, data(13,4)); end
end
```


- *Wczytywanie danych szklarni oraz roślin z pliku*

```
% Code that executes after component creation
function startupFcn(app)
    % Odczytanie i załadowanie danych z plików na starcie
    % Dane szklarni
    if (exist('szklarnia.txt','file') == 2)
        fileID = fopen('szklarnia.txt');
        dane = fscanf(fileID, '%s');
        params = strsplit(dane, ',');
        fclose(fileID);
        app.NazwaEditField.Value = string(params(1));
        app.DlugoscEditField.Value = str2double(params(2));
        app.SzerokoscEditField.Value = str2double(params(3));
        app.WysokoscEditField.Value = str2double(params(4));
    else
        app.NazwaEditField.Value = "Moja szklarnia";
    end

    % Dane roślin
    if (exist('rosliny.txt','file') == 2)
        fileID = fopen('rosliny.txt');
        dane = fscanf(fileID, '%s');
        params = strsplit(dane, ',');
        fclose(fileID);

        app.UITable.Data = {}; % wyczyszczenie tablicy przed załadowaniem danych
        app.UITable.ColumnName = {'Rodzaj', 'Nazwa', 'Pochłanianie', 'Szerokość', 'Długość', 'Wysokość', 'Data poprzedniego podlania', 'Stan'};
        app.UITable.Data = {'Pomidory', '4.3', '45', 'usycha', '1/17/2023'};
        app.UITable.Data = vertcat(app.UITable.Data, {'Tulipan', '0.3', '10', 'umiarkowane', '1/18/2023'});
        app.UITable.Data = vertcat(app.UITable.Data, {'Truskawki', '2.1', '15', 'nawodnione', '1/19/2023'});
        app.UITable.Data = vertcat(app.UITable.Data, {'Maliny', '1.8', '12', 'nawodnione', '1/19/2023'});
        app.UITable.Data = vertcat(app.UITable.Data, {'Róże', '0.5', '23', 'nawodnione', '1/20/2023'});
    end
end
```

- *Wczytywanie pliku podanego przez użytkownika*

```
% Button pushed function: WczytajSzklarnieButton
function WczytajSzklarnieButtonPushed(app, event)

    % Otworzenie okienka z wyborem pliku do odczytu danych
    [file, path] = uigetfile({'*.txt'}, 'Wybierz plik');
    if file ~= 0
        fullPath = fullfile(path, file);
        fileID = fopen(fullPath);
    else
        fileID = fopen('szklarnia.txt');
    end

    % Odczytanie danych z wybranego pliku
    dane = fscanf(fileID, '%s');
    params = strsplit(dane, ',');
    fclose(fileID);

    nazwa = string(params(1));
    dlugosc = str2double(params(2));
    szerokosc = str2double(params(3));
    wysokosc = str2double(params(4));

    % Wyświetlenie 4 parametrów w 4 EditField'ach
    app.NazwaEditField.Value = nazwa;
    app.DlugoscEditField.Value = dlugosc;
    app.SzerokoscEditField.Value = szerokosc;
    app.WysokoscEditField.Value = wysokosc;

end
```

- *Zapisanie parametrów szklarni do pliku*

```
% Button pushed function: ZapiszSzklarnieButton
function ZapiszSzklarnieButtonPushed(app, event)
    app.EdytujSzklarnieButton.Enable = "on";
    app.ZapiszSzklarnieButton.Enable = "off";
    app.NazwaEditField.Editable = "off";
    app.DlugoscEditField.Editable = "off";
    app.SzerokoscEditField.Editable = "off";
    app.WysokoscEditField.Editable = "off";

    NazwaSzklarni = num2str(app.NazwaEditField.Value);
    DlugoscSzklarni = num2str(app.DlugoscEditField.Value);
    SzerokoscSzklarni = num2str(app.SzerokoscEditField.Value);
    WysokoscSzklarni = num2str(app.WysokoscEditField.Value);

    % połączenie stringów i rozdzielenie przecinkiem
    resultString = strjoin({NazwaSzklarni, DlugoscSzklarni, SzerokoscSzklarni, WysokoscSzklarni}, ',');
    delete('szklarnia.txt'); % usuwamy plik, aby stworzyć nowy z danymi
    fid = fopen('szklarnia.txt', 'w'); % otwarcie pliku do zapisu
    fprintf(fid, '%s', resultString); % zapis do pliku
    fclose(fid); % zamknięcie pliku
end
```

2. Wnioski:

a. Raporty:

Zajęcia nr 1

Na pierwszych zajęciach odbyły się zajęcia wprowadzające, które pokazały podstawowe funkcje Matlaba przydatne do stworzenia aplikacji. Zostało dodane okienko główne wraz z przyciskami. Zrobiliśmy działające przejście do pozostałych okienek oraz panel z informacjami o szklarni. Dodano również okno szczegółów oraz okienko pomocy, które ma pomóc w zrozumieniu zasad działania aplikacji.

Zajęcia nr 2

Dodano klasę Roślina, reprezentująca rośliny występujące w danej szklarni. Wewnątrz niej utworzono niezbędne właściwości i atrybuty np. rodzaj, gatunek rośliny, obecny stan podlania itp. Utworzono również konstruktor oraz wymagane metody np. metoda obliczająca datę następnego podlania na podstawie stanu poprzedniego podlania oraz obecnej daty pobranej z systemu.

Zajęcia nr 3

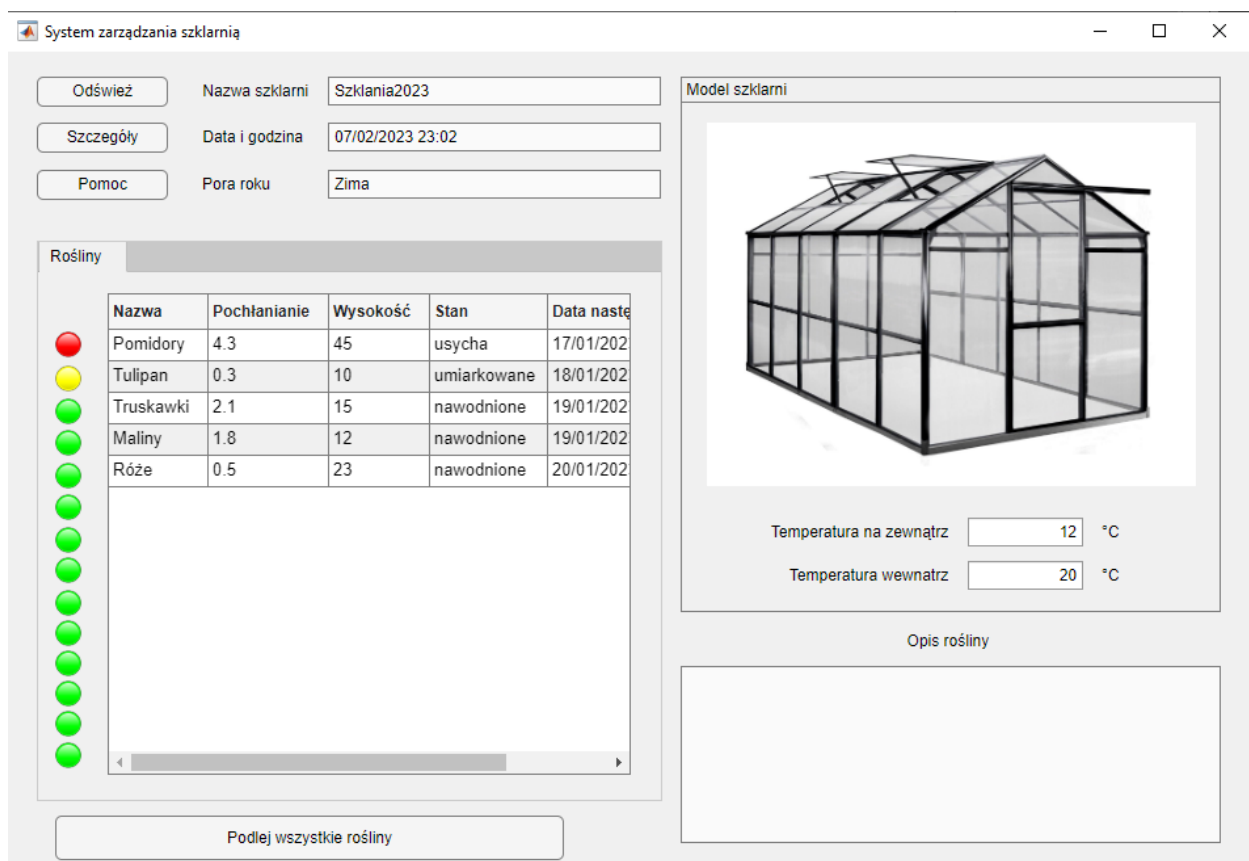
Dodano wyświetlanie obecnej daty i godziny oraz pory roku wyznaczonej na podstawie obecnego miesiąca. Dodatkowo w głównym oknie aplikacji dodano diody wskazujące na bieżąco stan podlania poszczególnej rośliny. Dodano również funkcję do aktualizowania koloru diod.

Zajęcia nr 4

Dodano możliwość zapisu i odczytu z pliku parametrów szklarni (nazwa, wymiary) oraz import oraz eksport danych wszystkich roślin. Dodatkowo w okienku szczegółów. Lekko zmieniono finalny wygląd okna szczegółów na bardziej uproszczony.

1. Dosyć często potykaliśmy się na błędach składni różnych funkcji, pętli, wywołań - tych podstawowych jak i bardziej skomplikowanych.
2. Duże trudności sprawiały nam dane wejściowe, które muszą być w odpowiednim formacie. Najczęstszy błąd przy czytaniu danych: "przykładowy_text" - nie jest stringiem. Czemu? Bo nie został przypisany do tego, aby był stringiem - choć można go wyświetlić w TextArea itp.
3. Funkcja do czytania plików z rozszerzeniem excela nie może się wywołać jeśli na komputerze nie ma zainstalowanego excela.
4. Matlab niewystarczająco wspiera obiektowość (brak mechanizmów typu polimorfizm).
5. Wprowadzane do Workspace'a pliki musiały być w pierwszej kolejności zweryfikowane, aby dane zostały odpowiednio wczytane.
6. W programie są ubogie możliwości w designie oraz UI, przez to wygląd aplikacji jest mało przyjazny dla nowych użytkowników.
7. Program nie posiada podstawowych funkcjonalności do modyfikowania komponentów, które występują w innych środowiskach i usprawniają pracę.

Okno główne - Zarządzanie szklarnią:



Okienko szczegółów - wprowadzanie, edytowanie, zapisywanie i wczytywanie danych:

Szczegóły

Parametry szklarni

Nazwa:

Długość: m

Szerokość: m

Wysokość: m

Szczegółowe dane

Nazwa pliku:




Zawartość pliku:

Rodzaj	Nazwa	Pochłanianie	Szerokość	Długość
Pomidory	4.3	45	usycha	1/17/2023
Tulipan	0.3	10	umiarkowane	1/18/2023
Truskawki	2.1	15	nawodnione	1/19/2023
Maliny	1.8	12	nawodnione	1/19/2023
Róże	0.5	23	nawodnione	1/20/2023

Okienko pomocy - wyjaśnia działanie aplikacji oraz kolory diod:

MATLAB App

Pomoc

Stan	Oznaczenie
	Nawodniona (nie ma potrzeby podlewania)
	Umiarkowanie (można podlać ale nie trzeba)
	Usycha (pilna potrzeba podlania)

Odświeżenie widoku aplikacji (stanu roślin, aktualnej daty i godziny itp.)

Umożliwia przejście do szczegółów aplikacji (

Otwiera to okienko