



**WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA**
z siedzibą w Rzeszowie

KOLEGIUM INFORMATYKI STOSOWANEJ

Kierunek: INFORMATYKA

Specjalność: Programowanie

Paweł Skrok
Nr albumu w67266

Aplikacja konsolowa w C# chłodni owoców

Promotor: mgr inż. Ewa Żesławska

Praca projektowa programowanie obiektowe C#

Rzeszów 2024

Spis treści

| | |
|---|-----------|
| Wstęp | 4 |
| 1 Opis założeń Projektu | 5 |
| 1.1 Cele Projektu | 5 |
| 1.2 Wymagania funkcjonalne i нефункционалне | 5 |
| 2 Opis struktury projektu | 6 |
| 2.1 Dane główne: | 6 |
| 2.2 Zarządzanie Danymi | 6 |
| 2.2.1 Struktura: | 6 |
| 2.2.2 Dane: | 6 |
| 2.2.3 Wymagania sprzętowe dla Aplikacji: | 6 |
| 2.3 Diagram Klas : | 6 |
| 2.3.1 Krótki opis klas: | 6 |
| 2.4 Baza Danych: | 7 |
| 3 Harmonogram prac nad Projektem | 8 |
| 3.1 Opis: | 8 |
| 3.1.1 Diagram Ganttа | 8 |
| 4 Prezentacja użycia aplikacji | 9 |
| 5 Podsumowanie | 13 |
| Bibliografia | 14 |
| Spis rysunków | 15 |

Wstęp

System zarządzania chłodnią to aplikacja konsolowa opracowana w celu uproszczenia procesu zarządzania owocami oraz pracownikami w chłodni. Zawiera funkcje dodawania, aktualizowania i usuwania owoców z zapasów, zarządzania danymi pracowników i wyświetlania kompleksowych informacji o przechowywanych owocach i zatrudnionych pracownikach, Aplikacja nie jest dopracowana pod kątem obsługi przez osoby niewykwalifikowane gdyż może to spowodować wytworzenie błędu.

Rozdział 1

Opis założeń Projektu

1.1 Cele Projektu

- Zabezpieczenie danych firmy, dostęp do danych oraz możliwość wprowadzania zmian.
- Stworzenie odpowiedniego systemu zarządzania który mógłby zapewnić wygodę oraz dostępność pracowników oraz owoców.
- Baza danych oparta na plikach .json w których dane są przechowywane jako słowniki
- Aplikacja musi być w miarę możliwości prosta aby każdy mógł podchodząc intuicyjnie opanować obsługę systemu

1.2 Wymagania funkcjonalne i нефункционалне

Wymagania funkcjonalne:

- użytkownik loguje się przez podanie loginu i hasła
- Możliwość dodawania Pracowników
- System umożliwia dodawania nowych oraz zmianę wartości dotychczasowych zapasów owoców

Wymagania нефункционалне:

- Aplikacja powinna być intuicyjna i łatwa do wyuczenia
- Czas reakcji systemu na działanie użytkownika powinien być minimalny
- Aplikacja powinna być zapisana zgodnie z zasadą czystego i przejrzystego kodu oraz zasadami programowania obiektowego w c#

Rozdział 2

Opis struktury projektu

2.1 Dane główne:

- Język Programowania: C#
- Aplikacja działa dla Platformy .NET
- Aplikacja korzysta z plików .json z Biblioteki:Newtonsoft.Json

2.2 Zarządzanie Danymi

2.2.1 Struktura:

W Programie jest 5 pomniejszych klas (Fruit, Employee,ToDoWithFruits,Add_Emplyeets,DisplayData) oraz jedna główna ToDoFruitColdStorage która spina pozostałe i jest kordynatorem wykonywania się programu

2.2.2 Dane:

Pobieranie, wczytywanie i zapis odbywa się na 2 plikach employeeData.json oraz Fruits_base.json w lokalizacji C:\Users\Acer\source\Project11\

2.2.3 Wymagania sprzętowe dla Aplikacji:

- System operacyjny zgodny z z Microsoft Visual studio oraz platformą .NET
- minimum 4 GB pamięci RAM,

2.3 Diagram Klas :

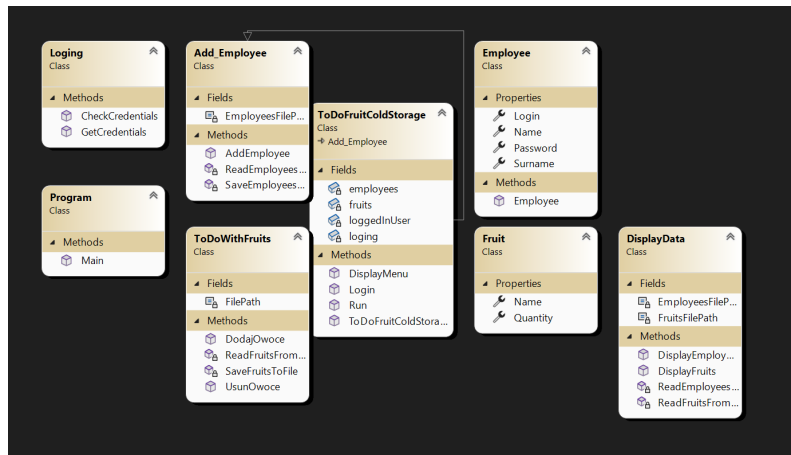
2.3.1 Krótki opis klas:

Klasa Fruit zawiera Pola które są pożyczone w klasie ToDoFruitColdStorage.

Klasa Employee zawiera Pola oraz konstruktor które wykorzystuje klasa Add_Employee do utworzenia obiektu.

Klasa ToDOWithFruit zawira 1 pole czyli ścieżkę do pliku zapisu , i 4 najważniejsze metody Read-FruitsFromFile() odczytująca plik .json ,dodaj owoce , usuń owoce i zapisz owoce w liście jako słownik

.



Rysunek 2.1: Diagram klas

Klasa DisplayData zawiera ścieżki do plików i ma za zadanie wczytanie danych z plików przechowujących dane o owocach jak i pracownikach i wypisanie ich.

Klasa Add_Employee zawiera ścieżkę dostępu do Pliku zapisu pracowników w formacie .json , oraz metody tworzenia obiektu pracownika oraz zapisu do pliku.

Klasa Logging jest w trakcie pracy zawiera ona Dane Logowania osoby która jest nad pracownikami która miała by dostęp do dodawania pracowników jak i usuwania, w przyszłości. W tym momencie sprawdza ona warunek czy istnieje pracownik o podanych danych logowania.

Klasa ToDoFruitColdStorage pola są typu: `private List<Object> objects;` najważniejszą metodą jest `DisplayMenu` które wyświetla menu wyboru, `Run` która jest główną pętlą programu , `Login` który jest podłożem przygotowanym pod logowanie dla np kierownika oraz konstruktor z wszystkimi polami.

Klasa Program przyjmuje jako pierwszy stworzony obiekt `Logging` który sprawdził by wprowadzone dane. oraz drugi odpowiedzialny za działanie całego programu

2.4 Baza Danych:

Baza danych opiera sie na plikach .json format przechowywanych danych jest listą w której obiekty takie jak pracownik, owoc są zapisywane jako słownik gdzie zmieniają wartości dla kluczy którymi są nazwy pól obiektów

plik zapisu pracowników -> `employeeData.json`

plik zapisu Owoców -> `Fruits_base.json`

Rozdział 3

Harmonogram prac nad Projektem

3.1 Opis:

Prace nad projektem trwały od 08.02.2024-26.02.2024. Praca została podzielona na następujące etapy:

- Przygotowanie środowiska VS Code - w tym etapie zainstalowanie odpowiednich bibliotek np. Newtonsoft.Json oraz utworzono repozytorium, które znajduje się pod adresem <https://github.com/zdrapek-jpg/c-wsiz-Application-FruitColdStorage>
- Planowanie działania aplikacji - przygotowanie planu, wymyślenie logiki programu, Hierarchii klas oraz głównych celów aplikacji
- Utworzenie klas programu.
- Zaimplementowanie plików .json do zapisu danych oraz zapis logiki z obsługą wyjątków dla danych
- Testy oraz obsługa wyjątków
- Utworzenie logiki inkrementacji wagi owoców jeśli nazwa jest taka sama
- Testowanie - poprawienie działania programu oraz eliminacja błędów i optymalizacja kodu
- Napisanie dokumentacji oraz wypchnięcie kodu do repozytorium na githubie

3.1.1 Diagram Gantt

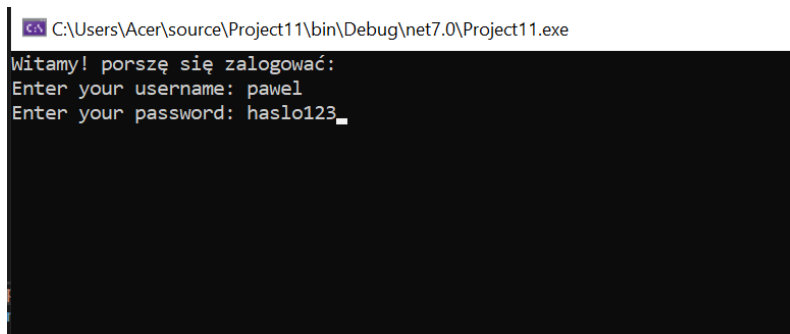
| Task | Start Date | End Date | Duration |
|---|------------|------------|----------|
| Przygotowanie środowiska VS Code | 08.02.2024 | 09.02.2024 | 2 days |
| Planowanie działania aplikacji | 10.02.2024 | 11.02.2024 | 2 days |
| Utworzenie klas programu | 12.02.2024 | 14.02.2024 | 3 days |
| Zaimplementowanie plików .json | 15.02.2024 | 16.02.2024 | 2 days |
| Testy oraz obsługa wyjątków | 17.02.2024 | 18.02.2024 | 2 days |
| Utworzenie logiki inkrementacji wagi owoców | 19.02.2024 | 20.02.2024 | 2 days |
| Testowanie | 21.02.2024 | 23.02.2024 | 3 days |
| Napisanie dokumentacji | 24.02.2024 | 25.02.2024 | 2 days |
| Wypchnięcie kodu do repozytorium | 26.02.2024 | 26.02.2024 | 1 day |

Rysunek 3.1: Diagram Gantt

Rozdział 4

Prezentacja użycia aplikacji

Przy otwarciu aplikacji mamy powitanie oraz funkcję która kiedyś będzie rozwijana czyli pierwsze logowanie do aplikacji, Następnie pokazuje się Menu Wyboru które jest zapętlone aż użytkownik wciśnie na klawiaturze 0.



Rysunek 4.1: Ekran główny

```
C:\Users\Acer\source\Project11\bin\Debug\net7.0\Project11.exe
Witamy! porszę się zalogować:
Enter your username: pawel
Enter your password: haslo123

===== Fruit Cold Storage Management System =====
1. Add Fruit
2. Remove Fruit
3. Add Employee
4. Login
5. Display all Data
0. Exit
Enter your choice: █
```

Rysunek 4.2: Menu Wyboru

Poniżej mamy przykład użycia funkcji dodawania owoców lub inkrementacji jeśli owoc jest już w bazie

```
===== Fruit Cold Storage Management System =====
1. Add Fruit
2. Remove Fruit
3. Add Employee
4. Login
5. Display all Data
0. Exit
Enter your choice: 1
Enter the fruit name: Jablko
Enter the fruit weight: 20
Fruit added/updated successfully!

===== Fruit Cold Storage Management System =====
1. Add Fruit
2. Remove Fruit
3. Add Employee
4. Login
5. Display all Data
0. Exit
Enter your choice: █
```

Rysunek 4.3: Opcja dodawania owocu

Usuwanie owcu z bazy danych jeśli takowy się znajduje, jeśli ilość do usunięcia jest większa od ilości dostępnej owoc jest usuwany, ;

```
===== Fruit Cold Storage Management System =====
1. Add Fruit
2. Remove Fruit
3. Add Employee
4. Login
5. Display all Data
0. Exit
Enter your choice: 2
Enter the fruit name: Malina
Enter the quantity of 'Malina' to be removed: 40
Fruit removed/updated successfully!

===== Fruit Cold Storage Management System =====
1. Add Fruit
2. Remove Fruit
3. Add Employee
4. Login
5. Display all Data
0. Exit
Enter your choice: 5
===== Fruits Data =====
Name: Malina, Quantity: 82
Name: Jabłko, Quantity: 20
===== Employees Data =====
Name: Paweł, Surname: Skrro, Login: login12, Password: haslo123
Name: Jan, Surname: Brzechwa, Login: brzechwa123, Password: 12345
```

Rysunek 4.4: Usuwanie owocu

Poniższy wybór opcji 5 pobiera dane z plików i je wypisuje w razie braku nie zwraca nic ;

```
===== Fruit Cold Storage Management System =====
1. Add Fruit
2. Remove Fruit
3. Add Employee
4. Login
5. Display all Data
0. Exit
Enter your choice: 5
===== Fruits Data =====
Name: Malina, Quantity: 122
Name: Jablko, Quantity: 12
Name: Jabłko, Quantity: 20
===== Employees Data =====
Name: Paweł, Surname: Skrro, Login: login12, Password: haslo123

===== Fruit Cold Storage Management System =====
1. Add Fruit
2. Remove Fruit
3. Add Employee
4. Login
5. Display all Data
0. Exit
Enter your choice: _
```

Rysunek 4.5: wypisanie danych Pracowników oraz owoców

Jeśli chcemy dodać pracownika wypełniamy jego dane które zostaną zapisane do pliku

```
===== Fruit Cold Storage Management System =====
1. Add Fruit
2. Remove Fruit
3. Add Employee
4. Login
5. Display all Data
0. Exit
Enter your choice: 3
Enter employee name: Jan
Enter employee surname: Brzechwa
Enter employee login: brzechwa123
Enter employee password: 12345
Employee 'brzechwa123' added successfully.
```

Rysunek 4.6: dodanie Pracownika

```
Enter your choice: 5
===== Fruits Data =====
Name: Malina, Quantity: 122
Name: Jabłko, Quantity: 20
===== Employees Data =====
Name: Pawel, Surname: Skrro, Login: login12, Password: haslo123
Name: Jan, Surname: Brzechwa, Login: brzechwa123, Password: 12345
```

Rysunek 4.7: Aktualizacja po dodaniu pracownika

4 to zapytanie o dane pracownika. w ten sposób możemy sprawdzić czy pracownik jest w bazie danych

```
===== Fruit Cold Storage Management System =====
1. Add Fruit
2. Remove Fruit
3. Add Employee
4. Login
5. Display all Data
0. Exit
Enter your choice: 4
Enter your username: login12
Enter your password: haslo123
Login successful!

===== Fruit Cold Storage Management System =====
1. Add Fruit
2. Remove Fruit
3. Add Employee
4. Login
5. Display all Data
0. Exit
Enter your choice:
```

Rysunek 4.8: sprawdzenie danych pracownika

zamknięcie programu za pomocą opcji 0 ;

```
===== Fruit Cold Storage Management System =====
1. Add Fruit
2. Remove Fruit
3. Add Employee
4. Login
5. Display all Data
0. Exit
Enter your choice: 0
Exiting the program. Goodbye!

C:\Users\Acer\source\Project11\bin\Debug\net7.0\Project11.exe (process 15512) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Rysunek 4.9: Exit

Rozdział 5

Podsumowanie

Projekt "Bazy Chłodni Owoców" to aplikacja konsolowa służąca do obsługi bazy, która przechowuje informacje o pracownikach oraz samych owocach. Podczas prac nad funkcjonalnościami zwróciłem uwagę głównie na wygodę użytkownika. Skoncentrowałem się na spełnianiu zarówno wymagań funkcjonalnych, jak i niefunkcjonalnych, co zaowocowało stworzeniem efektywnej, łatwej w obsłudze i responsywnej aplikacji.

Harmonogram realizacji projektu został przestrzegany, z wyjątkiem poprawek, które trwały do 28 lutego 2024 roku. To umożliwiło dalsze rozszerzenia i rozwijanie projektu. Ze względu na ograniczone dostępne narzędzia oraz brak wystarczającej wiedzy, projekt będzie kontynuowany w kolejnych krokach. Planowane jest dodanie nowych funkcji, takich jak:

- Utworzenie interfejsu graficznego
- Dodanie Logowania użytkownika do programu
- dodanie pliku który zapisuje historię działań użytkownika
- dodatkowe informacje o owocach jak i pracownikach
- dodanie nr pesel do pracownika i możliwości usuwania pracowników

Bibliografia

- [1] <https://learn.microsoft.com/en-us/dotnet/standard/serialization/system-text-json/deserialization>
- [2] <https://code-maze.com/csharp-read-and-process-json-file/>
- [3] <https://www.youtube.com/watch?v=GhQdlIFylQ8t=4801s>
- [4] <https://learn.microsoft.com/pl-pl/dotnet/csharp/fundamentals/exceptions/exception-handling>

Spis rysunków

| | | |
|-----|--|----|
| 2.1 | Diagram klas | 7 |
| 3.1 | Diagram Gantt | 8 |
| 4.1 | Ekran główny | 9 |
| 4.2 | Menu Wyboru | 10 |
| 4.3 | Opcja dodawania owocu | 10 |
| 4.4 | Usuwanie owocu | 11 |
| 4.5 | wypisanie danych Pracowników oraz owoców | 11 |
| 4.6 | dodanie Pracownika | 12 |
| 4.7 | Aktualizacja po dodaniu pracownika | 12 |
| 4.8 | sprawdzenie danych pracownika | 12 |
| 4.9 | Exit | 12 |