

Aplikacja konsolowa sklepu elektronicznego

Autor: Jakub Pakosiewicz

Opis aplikacji:

Aplikacja konsolowa sklepu elektronicznego umożliwia zarządzanie sklepem oraz obsługę klientów. Sklep oferuje szeroki wybór produktów elektronicznych, takich jak komputery, laptopy i telefony komórkowe. Użytkownicy mogą przeglądać asortyment, dodawać produkty do koszyka, zarządzać swoim kontem oraz dokonywać transakcji zakupowych.

Funkcje aplikacji:

1) Interfejs Klienta:

- a. **Logowanie i Rejestracja:** Użytkownicy mogą zalogować się na istniejące konto lub utworzyć nowe konto w sklepie.
- b. **Przeglądanie Produktów:** Klienci mogą przeglądać listę dostępnych produktów, w tym komputery, laptopy i telefony komórkowe.
- c. **Dodawanie do Koszyka:** Użytkownicy mogą dodawać produkty do koszyka zakupowego.
- d. **Zarządzanie Koszykiem:** Klienci mogą przeglądać zawartość koszyka, usuwać produkty oraz dokonywać zakupów.
- e. **Zarządzanie Saldem:** Użytkownicy mogą wpłacać lub wypłacać środki z konta użytkownika.

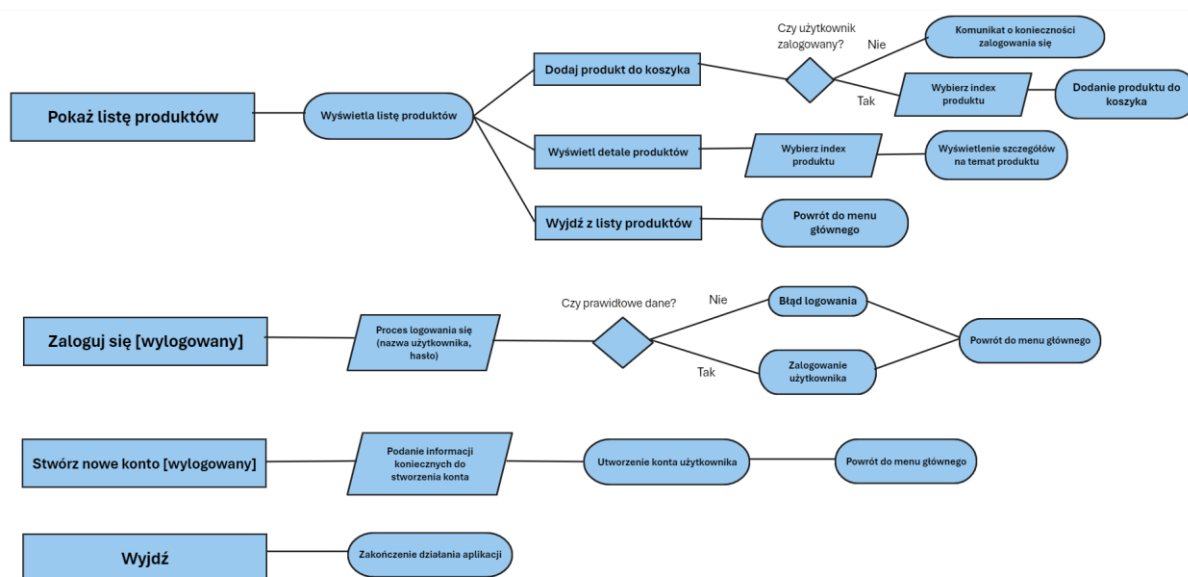
2) Interfejs Managera:

- a. **Przeglądanie Listy Pracowników:** Manager może przeglądać listę pracowników sklepu oraz zarządzać ich wynagrodzeniem oraz dniem wolnym.
- b. **Zarządzanie Tablicą Zadań:** Manager może przeglądać, dodawać i usuwać zadania z tablicy zadań dla pracowników.
- c. **Przeglądanie Listy Produktów:** Manager może przeglądać listę produktów w sklepie oraz zarządzać ich ilością oraz specyfikacją.

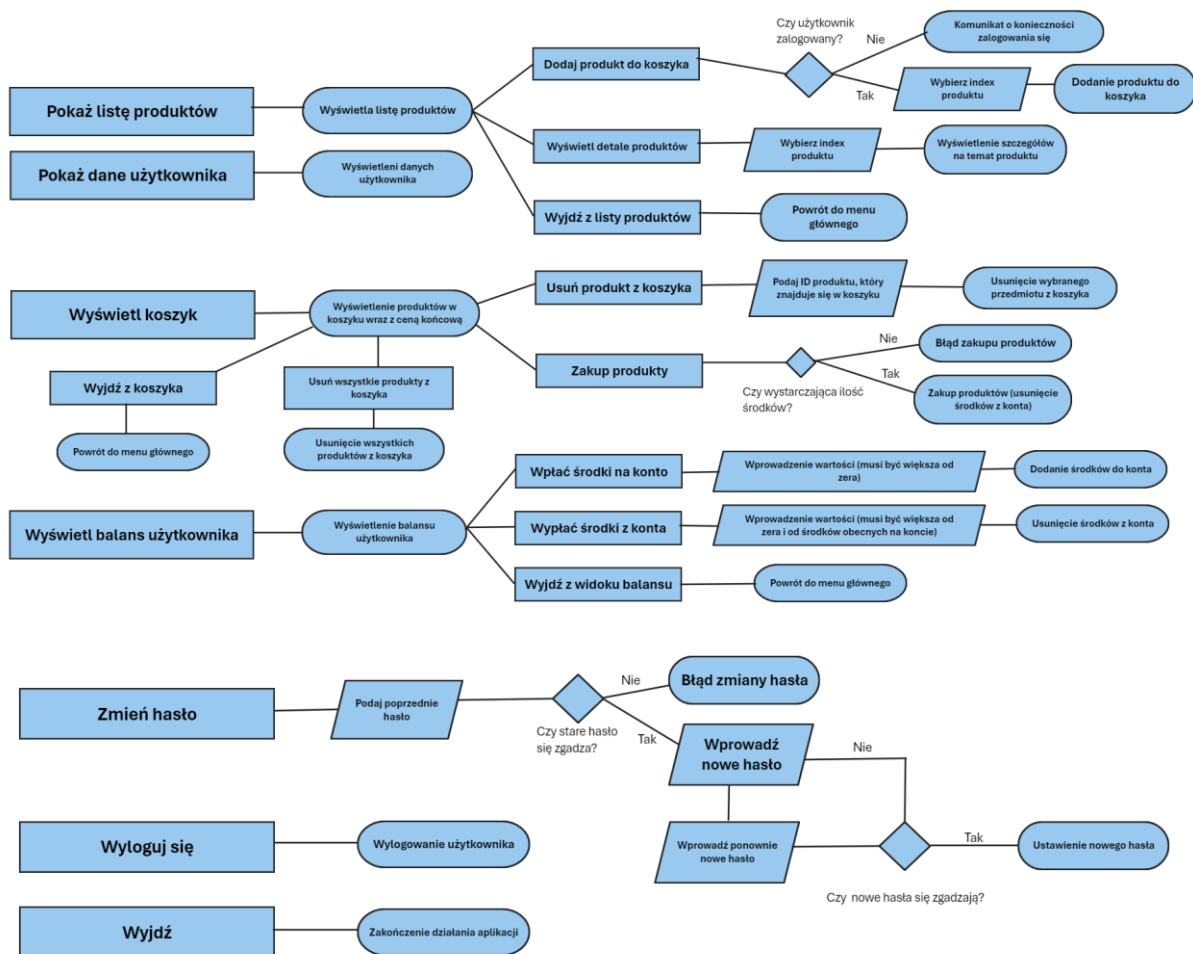
Kluczowe koncepcje:

1. **Klasy Produktów:** Każdy rodzaj produktu (komputer, laptop, telefon) jest reprezentowany przez odpowiadającą mu klasę. Klasy te przechowują informacje o produkcie, takie jak nazwa, cena, ilość dostępna oraz specyfikacja techniczna.
2. **Klasa Pracownika:** Przechowuje informacje o pracownikach sklepu oraz umożliwia modyfikowanie wypłaty wraz z dniami wolnymi danego pracownika.
3. **Tablica Zadań:** Tablica zadań dla pracowników jest reprezentowana przez odpowiednią klasę, która umożliwia dodawanie, usuwanie i przeglądanie zadań.
4. **Koszyk Zakupowy:** Koszyk zakupowy jest reprezentowany przez klasę, która umożliwia dodawanie produktów, usuwanie produktów oraz dokonywanie transakcji zakupowych.
5. **Saldo Użytkownika:** Umożliwia zarządzanie środkami użytkownika poprzez wypłatę oraz wpłatę środków. Klasa ta odpowiada także za zakup przedmiotów zawartych w koszyku (sprawdza, czy środki są wystarczające).
6. **Klasa Użytkownika:** Klasa użytkownika zarządza danymi użytkowników, takimi jak nazwa, hasło, adres e-mail i saldo konta. Łączy ona także funkcjonalności klasy koszyka zakupowego oraz salda klienta.

Schemat wyboru dla niezalogowanego klienta w aplikacji konsolowej:



Schemat wyboru dla zalogowanego klienta w aplikacji konsolowej:



Schemat wyboru dla interfejsu managera w aplikacji konsolowej:



Przykładowe zastosowanie:

- A) **Logowanie Użytkownika:** Użytkownik uruchamia aplikację, loguje się na swoje konto lub rejestruje nowe konto, aby uzyskać dostęp do funkcji sklepu.
- B) **Przeglądanie Produktów:** Klient przegląda dostępne produkty, wybiera interesujące go przedmioty i dodaje je do koszyka zakupowego.
- C) **Zarządzanie Koszykiem:** Użytkownik przegląda zawartość koszyka, usuwa niechciane produkty lub dokonuje zakupu.
- D) **Dodawanie Zadań dla Pracowników:** Manager dodaje zadania do tablicy zadań dla pracowników, określając ich priorytet i termin wykonania.
- E) **Zarządzanie Listą Produktów:** Manager zarządza listą produktów w sklepie, aktualizując ilość produktów lub specyfikację techniczną.

Testy jednostkowe przeprowadzone w aplikacji:

Testy zostały napisane dla każdej klasy z osobna. Testują one poprawność działania metod zawartych w klasach. Jeżeli w testach klas pojawi się jakiś błąd, to aplikacja się nie uruchomi, a jedynie pokaże, w którym miejscu test nie przeszedł pomyślnie.

Wymagania systemowe:

- System operacyjny obsługujący konsolę tekstową (np. Windows, Linux)
- Zainstalowany kompilator języka C++ (np. g++, Visual C++)
- Dostęp do standardowej biblioteki języka C++ (STL)

Dodatkowe informacje o projekcie:

- Projekt napisany został w C++, w środowisku Visual Studio (2022)