Opis projektu:

1. Klient:

- Klient jest podstawowym użytkownikiem systemu.

- Po zalogowaniu ma dostęp do funkcji przeglądania produktów w sklepie.

- Może dodawać produkty do koszyka zakupowego.

- Klient ma możliwość składania zamówień, potwierdzania ich i dokonywania transakcji.

- Dodatkowo, klient może skonsultować się z pomocnikiem, zadając mu pytanie.

2. Pomocnik:

- Pomocnik to użytkownik posiadający specjalne uprawnienia do udzielania odpowiedzi na pytania klientów.

- Po zalogowaniu ma dostęp do listy pytań od klientów.

- Może udzielać odpowiedzi na pytania klientów i zapewniać im wsparcie.

- Nie ma dostępu do funkcji składania zamówień czy dokonywania transakcji.

3. Admin:

- Admin jest użytkownikiem o najwyższych uprawnieniach.

- Po zalogowaniu ma dostęp do funkcji zarządzania pomocnikami.

- Może rejestrować nowych pomocników oraz usuwać ich konta.

- Nie ma bezpośredniego dostępu do funkcji związanych z zakupami.

4. Proces zakupów:

- Klient może dodawać produkty do koszyka.

- Po zakończeniu wyboru produktów, klient składa zamówienie.

- Zamówienie musi zostać potwierdzone przez klienta.

- Po potwierdzeniu, klient może dokonać transakcji i sfinalizować zakup.

5. Komunikacja:

- Klient może zadawać pytania pomocnikowi w formie wiadomości.

- Pomocnik ma dostęp do systemu zgłoszeń od klientów i odpowiada na pytania.

- Administrator zarządza pomocnikami, rejestrując ich i usuwając konta w razie potrzeby.

6. Bezpieczeństwo:

- Wszyscy użytkownicy (klienci, pomocnicy, admin) muszą się zalogować, aby uzyskać dostęp do swoich funkcji.

- Bezpieczeństwo transakcji klienta musi być gwarantowane przez system.

Projekt ten opisuje interakcje pomiędzy trzema głównymi rolami: Klientem, Pomocnikiem a Administratorem, koncentrując się na funkcjonalności sklepu internetowego. Wprowadza również element obsługi pytań od klientów, co umożliwia lepsze zrozumienie i obsługę potrzeb klienta w trakcie zakupów online.

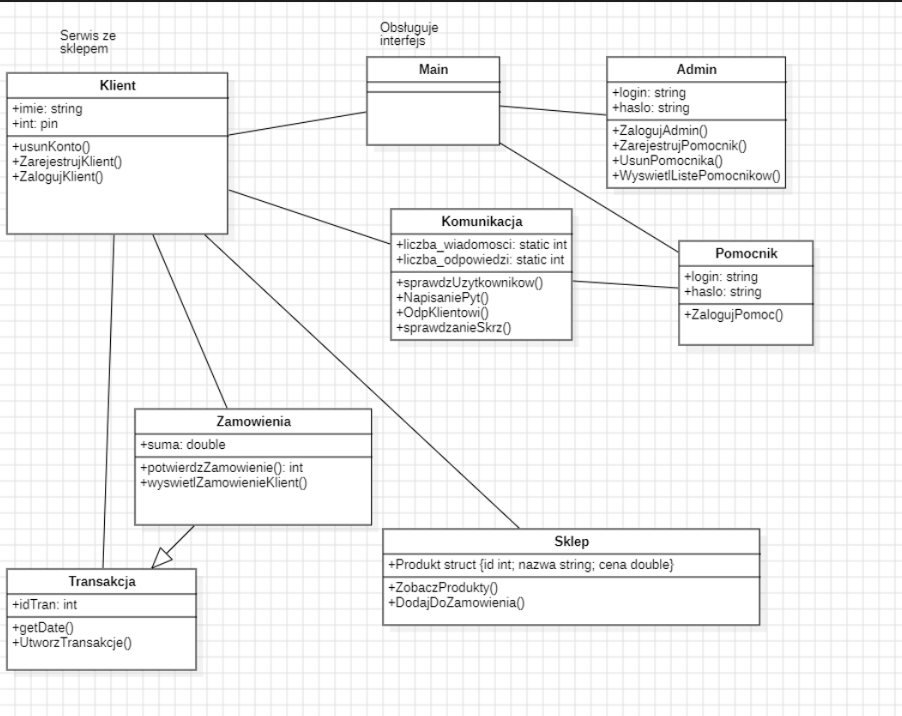
|  |  |
| --- | --- |
| Nazwa | Sklep |
| Twórca | Karol Goździewski |
| Poziom ważności | 5 (w skali 0-5) |
| Typ przypadku użycia | Niezbędny |
| Aktorzy | Klient |
| Krótki opis | Użytkownik ma możliwość korzystania ze sklepu |
| Warunki wstępne | Użytkownik wybrał opcje sklep w menu |
| Warunki końcowe | Użytkownik wybrał produkty |
| Główny przepływ zdarzeń | 1.Klient chce zlozyc zamowienie  2.Klient wybiera opcje sklep  3.Klient wybiera z listy produkty |
| Alternatywne przepływy zdarzeń | 1.Klient przerwał proces  2.Klient nacisnął 0 co powoduje powrot do menu klienta  3.Klient podał złe id produktu które chce zamowic. |
| Specjalne wymagania | Brak |
| Notatki i kwestie | Brak |

|  |  |
| --- | --- |
| Nazwa | Zamowienie |
| Twórca | Karol Goździewski |
| Poziom ważności | 5 (w skali 0-5) |
| Typ przypadku użycia | Niezbędny |
| Aktorzy | Klient |
| Krótki opis | Użytkownik sprawdza swoje zamówienie ze sklepu |
| Warunki wstępne | Użytkownik wybrał produkty ze sklepu |
| Warunki końcowe | Użytkownik potwierdzil zamowienie |
| Główny przepływ zdarzeń | 1.Klient po zlozeniu zamówienia w sklep, wraca do menu i wybiera opcje 2- zamowienia  2.Klient musi wybrać czy potwierdza zamowienie  3.Klient po zatwierdzeniu zachodzi transakcja |
| Alternatywne przepływy zdarzeń | 1.Klient przerwał proces  2.Klient nacisnął 2 co powoduje powrot do menu klienta |
| Specjalne wymagania | Brak |
| Notatki i kwestie | Brak |

|  |  |
| --- | --- |
| Nazwa | Transakcja |
| Twórca | Karol Goździewski |
| Poziom ważności | 5 (w skali 0-5) |
| Typ przypadku użycia | Niezbędny |
| Aktorzy | Klient |
| Krótki opis | System tworzy transakcje |
| Warunki wstępne | Użytkownik potwierdził swoje zamowienie |
| Warunki końcowe | System utworzył plik z transakcja i datą |
| Główny przepływ zdarzeń | 1.System dostaje wiadomość ze klient potwierdzil zamowienie  2.System odczytuje id produktów i podlicza je  3.System aplikacji tworzy plik transakcji w określonym folderze i zapisuje w nim date transakcji, produkty jakie wybrał klient i sume |
| Alternatywne przepływy zdarzeń | 1.Klient może podac rozne id i transakcja może się nie powieść |
| Specjalne wymagania | Brak |
| Notatki i kwestie | Brak |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Nazwa | Zaloguj jako klient | | Numer | au1 | | Twórca | Karol Gozdziewski | | Poziom ważności | 5 (w skali 0-5) | | Typ przypadku użycia | Niezbędny | | Aktorzy | Klient | | Krótki opis | Użytkownik loguje się jako klient | | Warunki wstępne | Użytkownik przeszedł pomyślnie proces autoryzacji | | Warunki końcowe | Użytkownik wylogował się | | Główny przepływ zdarzeń | 1.Klient chce się zalogować do serwisu  2.Klient pomyślnie zalogował się na swoje konto | | Alternatywne przepływy zdarzeń | 1.Klient przerwał proces logowania się 2.Wystąpił błąd połączenia z serwerem, przez co klient nie może się zalogować | | Specjalne wymagania | Brak | | Notatki i kwestie | Brak | |
| |  |  | | --- | --- | | Nazwa | Zarejestruj jako użytkownik(klient) | | Numer | au2 | | Twórca | Karol Goździewski | | Poziom ważności | 5 (w skali 0-5) | | Typ przypadku użycia | Niezbędny | | Aktorzy | Klient | | Krótki opis | Użytkownik rejestruje się jako klient | | Warunki wstępne | Użytkownik przeszedł pomyślnie proces autoryzacji | | Warunki końcowe | Użytkownik zakończył rejestracje | | Główny przepływ zdarzeń | 1.Klient chce się zarejestrować  2.Klient podaje swój mail i tworzy hasło  3.Klient pomyślnie utworzył konto | | Alternatywne przepływy zdarzeń | 1.Klient przerwał proces rejestracji 2.Klient podał zły format hasła lub e-mail | | Specjalne wymagania | Brak | | Notatki i kwestie | Brak | |
| |  |  | | --- | --- | | Nazwa | Napisanie zapytania o pomoc z urządzeniem do pracownika | | Twórca | Karol Goździewski | | Poziom ważności | 5 (w skali 0-5) | | Typ przypadku użycia | Niezbędny | | Aktorzy | Klient | | Krótki opis | Klient może napisać do Pomocnika online z prośbą o pomoc | | Warunki wstępne | Użytkownik przeszedł pomyślnie proces autoryzacji | | Warunki końcowe | Użytkowik wysłał zapytanie pomyślnie | | Główny przepływ zdarzeń | 1.Użytkownik wybiera opcje kontaktu z pomocnikiem  2. Użytkownik Pisze zapytanie w formie małego formularza  3. Użytkownik Wysyła zapytanie | | Alternatywne przepływy zdarzeń | 1.Użytkownik anuluje stworzenie zapytania | | Specjalne wymagania | Brak | | Notatki i kwestie | Brak | |
| |  |  | | --- | --- | | Nazwa | Zarejestruj Pomocnika online z uprawnieniamii uzupełnij profil pracownika | | Twórca | Karol Goździewski | | Poziom ważności | 5 (w skali 0-5) | | Typ przypadku użycia | Niezbędny | | Aktorzy | Administrator forum | | Krótki opis | Administrator forum tworzy profil dla użytkownika | | Warunki wstępne | Administrator przeszedł pomyślnie proces autoryzacji | | Warunki końcowe | Administrator utworzył nowy profil dla pracownika | | Główny przepływ zdarzeń | 1.Administrator wybiera opcje utworzenia nowego profilu pracownika z uprawnieniami  2.Wypełnia informacje podane przez pracownika  3.Tworzy zaufane konto dla pracownika | | Alternatywne przepływy zdarzeń | 1.Administrator może przerwać tworzenie konta | | Specjalne wymagania | Brak | | Notatki i kwestie | Brak | |
| |  |  | | --- | --- | | Nazwa | Zaloguj jako Pomocnik online | | Twórca | Karol Goździewski | | Poziom ważności | 5 (w skali 0-5) | | Typ przypadku użycia | Niezbędny | | Aktorzy | Pomocnik online | | Krótki opis | Pomocnik online loguje się na konto podane przez Administratora | | Warunki wstępne | Pomocnik online przeszedł pomyślnie proces autoryzacji | | Warunki końcowe | Pomocnik online przeszedł pomyślnie proces logowania | | Główny przepływ zdarzeń | 1.Po wejsciu na forum pomocnik loguje się na uprawnione konto | | Alternatywne przepływy zdarzeń | 1.Pomocnik może anulować logowanie się  2.Wystąpił błąd połączenia z serwerem, przez co pomocnik nie może się zalogować | | Specjalne wymagania | Brak | | Notatki i kwestie | Brak | |
|  |
| |  |  | | --- | --- | | Nazwa | Obsługa klienta w postaci prywatnych wiadomosci | | Twórca | Karol Goździewski | | Poziom ważności | 5 (w skali 0-5) | | Typ przypadku użycia | Niezbędny | | Aktorzy | Pomocnik online | | Krótki opis | Pomocnik online pomaga użytkownikom w prywatnych rozmowach, jeżeli ci wyślą do niego zapytanie | | Warunki wstępne | Pomocnik online przeszedł pomyślnie proces autoryzacji | | Warunki końcowe | Pomocnik online udzielił pomocy zwykłemu użytkownikowi | | Główny przepływ zdarzeń | 1.Po zalogowaniu się pomocnik online zagląda do skrzynki prywatnych wiadomości  2.Jeżeli w skrzynce znajdują się prośby o pomoc pracownik musi na nie odpowiedzieć | | Alternatywne przepływy zdarzeń | 1.W skrzynce nie ma zgłoszeń więc pracownik może zająć się obsługą forum  2. W przypadkach, gdzie nie można pomóc klientowi online, pomocnik powinien wyszukać serwis, który może pomóc klientowi w profesjonalny sposób i polecić ten serwis klientowi | | Specjalne wymagania | Brak | | Notatki i kwestie | Brak | |

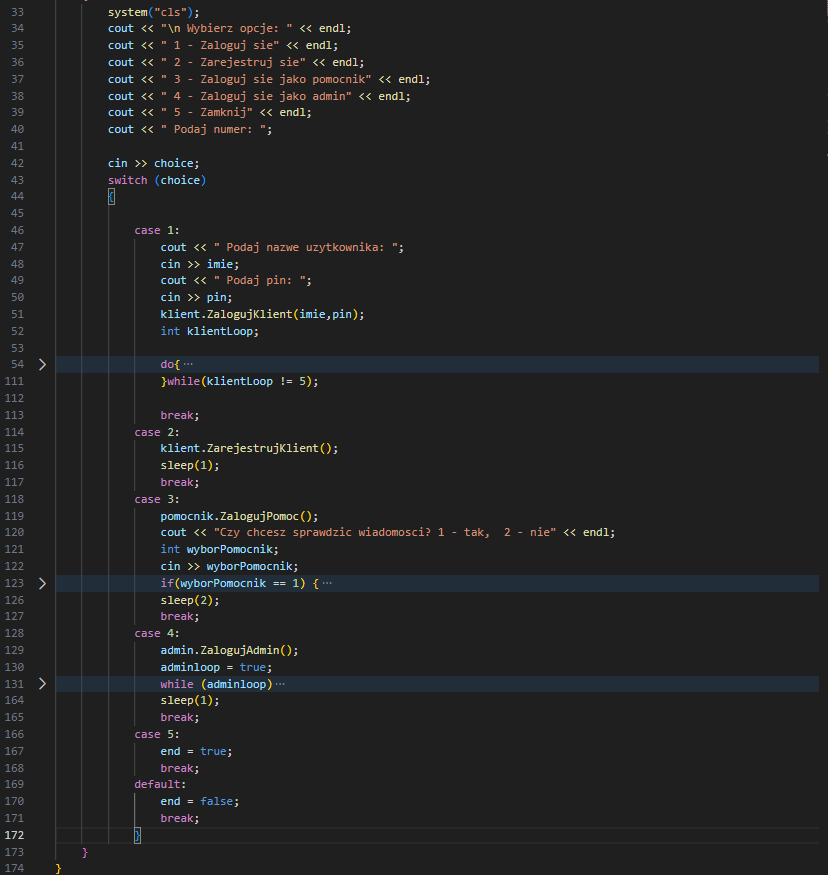
Czesc druga



System uml pokazuje zaleznosci klas i funkcji które sa zawarte w projekcie. Klasa klient moze korzystać z klas Sklep zamówienia i transakcje oraz komunikacja. Pomocnik używa klasy komunikacja, aby moc odpowiadać klientowi na zapytania. Klasa admin słuzy do tworzenia i usuwania nowych pomocnikow.

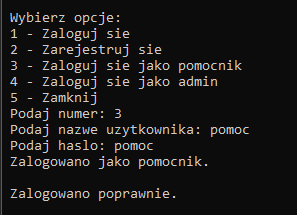
Czesc trzecia testy

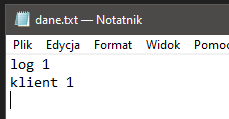
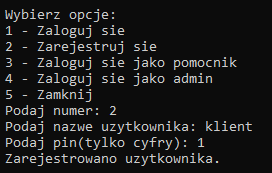
Test funkcjonalny dla Serwis.exe, program stworzony przez Karol Goździewski.

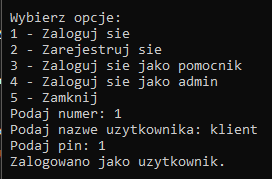
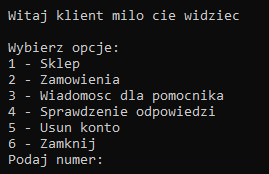


|  |  |
| --- | --- |
| **Nazwa testu** | **Test funkcjonalny menu** |
| Cel testu | Sprawdzenie poprawnosci działania menu i jego funkcji |
| Przygotowanie testu | Wpisywanie określonych cyfr aby sprawdzić poprawnosc |
| Kroki | 1.Sprawdzenie czy Logowanie jako administrator działa  2.Sprawdzenie czy logowanie jako pomocnik działa 3.Sprawdzenie czy rejestrowanie samodzielne przez klienta działa 4.Po utworzeniu konta przez rejestracje, zalogowanie się na konto klienta |
| Oczekiwane wyniki | 1.Dla admina powinno wyświetlić się „Zalogowano jako admin” I otworzyć menu admina  2.Dla pomocnika, po weryfikacji powinno wyświetlić się komuikat „Zalogowano jako pomocnik” i wyświetlic opcje pomocnika  3.Po zarejestrowaniu, imie i pin uzytkownika powinien być zapisany do pliku dane.txt, a nastepnie podczas logowania, program zczytuje z pliku imie i pin i loguje nas jako uzytkownik. |
| Wyniki | Całe menu funkcjonuje poprawnie, wynik testu funkcjonalnego – pomyślny |
| Wnioski | Menu programu Serwis działa poprawnie |

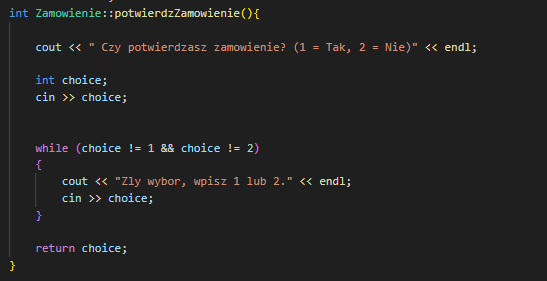
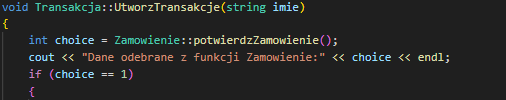
|  |  |
| --- | --- |
| 1 | 2 |
| 3 |  |

pomocnik:  


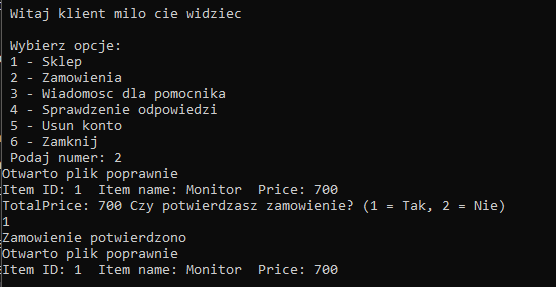
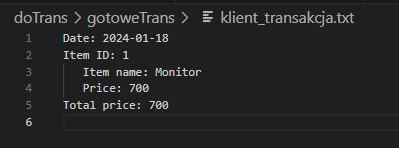
klient:  


Test integracyjny  
Test integracyjny dla Serwis.exe, program stworzony przez Karol Goździewski.

  
Funkcja ta jest uzyta w innej klasie w innej funkcji Transakcja  


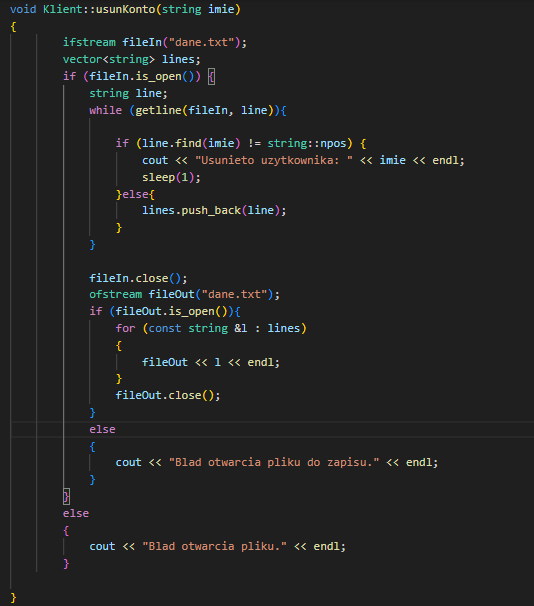
|  |  |
| --- | --- |
| **Nazwa testu** | **Test funkcji** |
| Cel testu | Sprawdzenie czy funkcja UtworzTransakcje wspolpracuje z funkcja potwierdzZamowienie |
| Przygotowanie testu | Zalogować się jako klient |
| Kroki | 1 Wejsc w sklep.  2 Wybrać produkty  3 Wrocic do menu klienta 4 W menu wybrac zamowienia i potwierdzic |
| Oczekiwane wyniki | „Dane odebrane z funkcji zamowienie: 1” |
| Wyniki | Program odczytuje ze dostal zgode i tworzy transakcje |
| Wnioski | Funkcje działają ze sobą poprawnie |

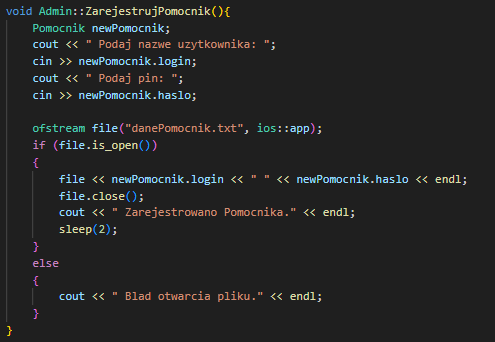
Testy Jednostkowe

|  |  |
| --- | --- |
| **Nazwa testu** | **Test jednostkowy funkcji usunKonto()** |
| Cel testu | Sprawdzenie czy funkcja usun konto działa |
| Przygotowanie testu | Zalogowanie się jako klient |
| Kroki | Po zalogowaniu się wybrać opcje 5  Wybrac odpowiedz [y-tak/n-nie] |
| Oczekiwane wyniki | Zalogowane Konto zostanie usunięte |
| Wyniki | Funkcja usuwa wszystkich uzytkownikow a nie jednego |
| Wnioski | Trzeba poprawic funkcje tak aby usuwała tylko zalogowanego uzytkownika |

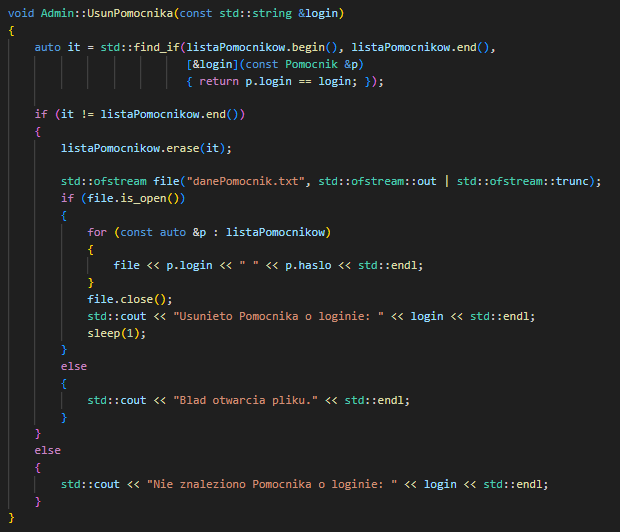
1



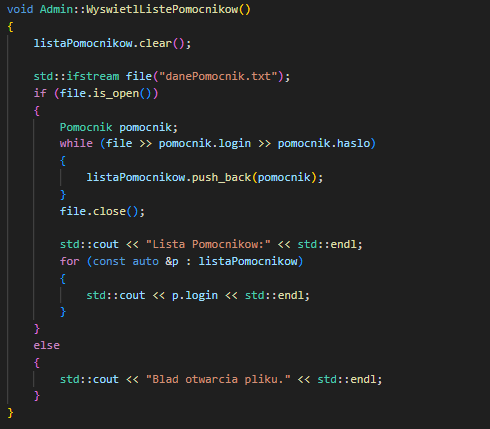
|  |  |
| --- | --- |
| **Nazwa testu** | **Test jednostkowy Admin::ZarejestrujPomocnika** |
| Cel testu | Sprawdzenie czy Admin utworzy nowy profil pomocnika |
| Przygotowanie testu | Zalogowanie jako admin |
| Kroki | Po zalogowaniu wybrać opcje nr 1 a następnie podac dane |
| Oczekiwane wyniki | Nowo utworzony pomocnik pojawi się w danepomocnika.txt |
| Wyniki | Po wykonaniu funkcji tworzy się nowy pomocnik |
| Wnioski | Funkcja działa prawidłowo |

2  


|  |  |
| --- | --- |
| **Nazwa testu** | **Test jednostkowy funkcji usunPomocnika** |
| Cel testu | Sprawdzenie czy funkcja usun Pomocnika działa |
| Przygotowanie testu | Zalogowanie się jako admin |
| Kroki | Po zalogowaniu się wybrać opcje 2  Wyswietlą się zarejestrowani pomocnicy wpisac login wybranego aby usunac |
| Oczekiwane wyniki | Wybrane Konto pomocnika zostanie usunięte |
| Wyniki |  |
| Wnioski | Funkcja działa poprawnie |

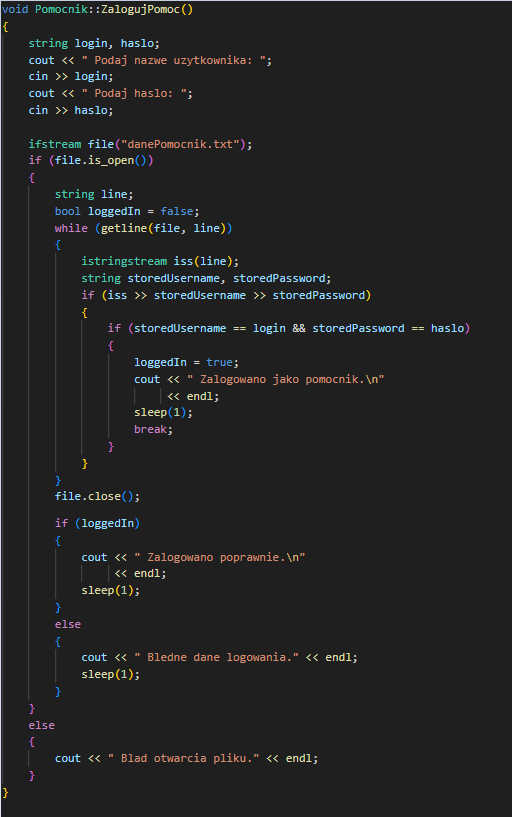
3  


|  |  |
| --- | --- |
| **Nazwa testu** | **Test jednostkowy funkcji Admin::WyswietlListePomocnikow** |
| Cel testu | Sprawdzenie czy funkcja Wyswietl liste pomocnikow działa |
| Przygotowanie testu | Zalogowanie się jako admin |
| Kroki | Po zalogowaniu się wybrać opcje 3  Wyswietlą się zarejestrowani pomocnicy |
| Oczekiwane wyniki | Zarejestrowane konta pomocnikow wyswietlą sie |
| Wyniki |  |
| Wnioski | Funkcja działa poprawnie |

4  


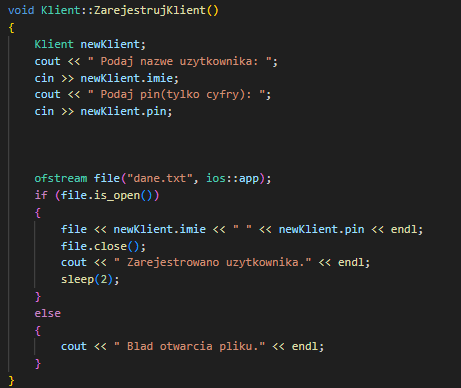
|  |  |
| --- | --- |
| **Nazwa testu** | **Test jednostkowy funkcji Pomocnik::ZalogujPomoc** |
| Cel testu | Sprawdzenie czy funkcja Zaloguj działa |
| Przygotowanie testu | Uruchomienie programu |
| Kroki | Po zalogowaniu się wybrać opcje 3  Wpisac dane pomocnika |
| Oczekiwane wyniki | Pomocnik zaloguje sie |
| Wyniki |  |
| Wnioski | Funkcja działa poprawnie |

5



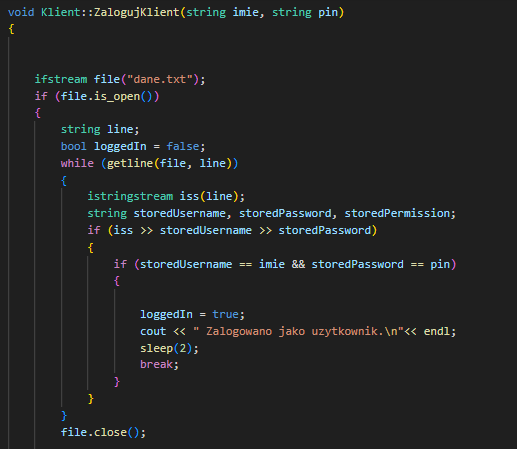
|  |  |
| --- | --- |
| **Nazwa testu** | **Test jednostkowy funkcji Klient::ZarejestrujKlient** |
| Cel testu | Sprawdzenie czy funkcja Zarejestruj działa |
| Przygotowanie testu | Uruchomienie programu |
| Kroki | Po zalogowaniu się wybrać opcje 2  Wpisac wybrane przez siebie dane |
| Oczekiwane wyniki | Klient rejestruje sie |
| Wyniki |  |
| Wnioski | Funkcja działa poprawnie |

6



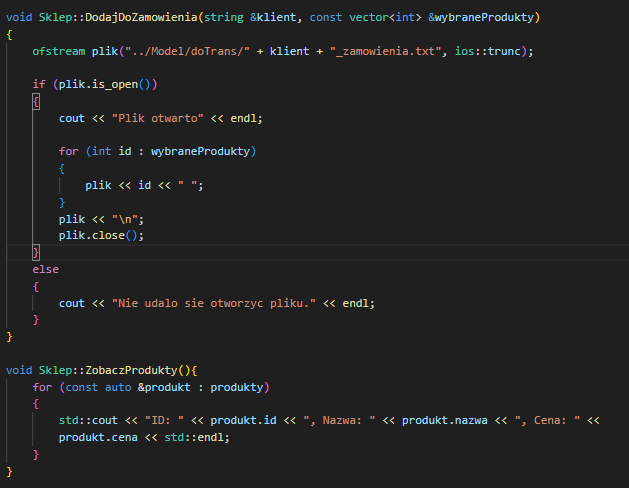
|  |  |
| --- | --- |
| **Nazwa testu** | **Test jednostkowy funkcji Klient::ZarejestrujKlient** |
| Cel testu | Sprawdzenie czy funkcja Zaloguj działa |
| Przygotowanie testu | Uruchomienie programu |
| Kroki | Po zalogowaniu się wybrać opcje 1  Wpisac utworzone przez siebie dane |
| Oczekiwane wyniki | Klient loguje sie |
| Wyniki |  |
| Wnioski | Funkcja działa poprawnie |

7

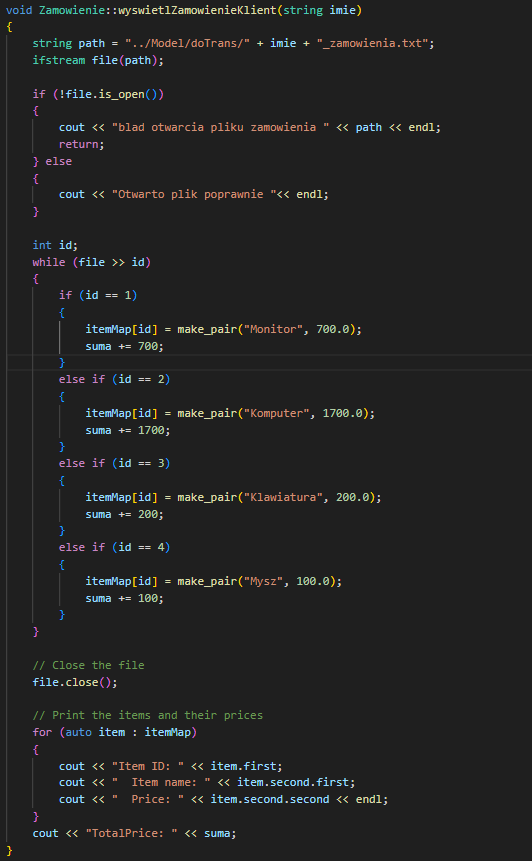


|  |  |
| --- | --- |
| **Nazwa testu** | **Test jednostkowy funkcji Sklep::DodajDoZamowienia** |
| Cel testu | Sprawdzenie czy funkcja Zapisuje do pliku id przedmioty |
| Przygotowanie testu | Uruchomienie programu |
| Kroki | Zalogowac jako klient Po zalogowaniu się wybrać opcje 1  Po wyswietleniu się przedmiotow wpisac id przedmiotu(np. 1) |
| Oczekiwane wyniki | Klient utworzy plik z nazwa klienta i zapisanym id |
| Wyniki |  |
| Wnioski | Funkcja działa poprawnie |

8



|  |  |
| --- | --- |
| **Nazwa testu** | **Test jednostkowy funkcji wyswietlZamowienieKlient** |
| Cel testu | Sprawdzenie czy funkcja dodaj do zamowienia zapisuje poprawnie i odczytuje id zamowionego przedmiotu |
| Przygotowanie testu | Uruchomienie programu |
| Kroki | Zalogowac jako klient Po zalogowaniu się wybrać opcje 2 |
| Oczekiwane wyniki | Klient utworzy plik z nazwa klienta i zapisanym id |
| Wyniki |  |
| Wnioski | Funkcja działa poprawnie |

9  


|  |  |
| --- | --- |
| **Nazwa testu** | **Test jednostkowy funkcji Transakcja::Utworz transakcje** |
| Cel testu | Sprawdzenie czy funkcja poprawnie zapisuje transakcje do pliku |
| Przygotowanie testu | Uruchomienie programu |
| Kroki | Zalogowac jako klient Po zalogowaniu się wybrać opcje 2  Potwierdzic wyswietlone przedmioty |
| Oczekiwane wyniki | Funkcja tworzy plik w ../model/doTrans/gotoweTrans który zapisuje razem z datą zamowione przedmioty |
| Wyniki |  |
| Wnioski | Funkcja działa poprawnie |