

Empik Helper - Dokumentacja

Filip Zagórny, Weronika Janus

Czerwiec 2019

Contents

1	Opis projektu	3
2	Diagram klas, przypadków użycia, karty CRC	4
3	Dokumentacja kodu	7
3.1	Klasa Customer	7
3.2	Klasa Book	8
3.3	Klasa Item	8
3.4	Klasa MusicAlbum	9
4	Instrukcja obsługi	9
4.1	Panel startowy	9
4.2	Przyciski w lewej kolumnie	10
4.3	Przyciski "Add book" i "Add music album"	10
4.4	Przycisk "Statistics"	11
4.5	Przycisk "Customer Service"	11

1 Opis projektu

Aplikacja okienkowa ma na celu usprawnić prace w sklepach a'la Empik, wypożyczalniach czy bibliotekach. Aplikacja jest z widoku właściciela/administratora sklepu. Jest prymitywna baza danych, przechowująca informacje o klientach i przedmiotach znajdujących się w sklepie. Oprawiona w interfejs graficzny, zawiera opcje takie, jak:

- **Customer Service** - Obsługa klienta. Pierwszym krokiem jest sprawdzenie czy klient o danym imieniu i nazwisku znajduje się w bazie. Jeśli nie - automatycznie tworzy się nowa pozycja w bazie, wraz z ustawieniem parametrów początkowych dla nowego klienta. Następnie, wybieramy z listy przedmiot, który klient chce kupić. Wciskając 'OK', "w ukryciu" wywoływane są metody odpowiedzialne za aktualizację wszystkich danych w całej bazie.
- **Show all clients/music albums/books** - Opcja ta pozwala nam wyświetlić wszystkich klientów/książki/albumy muzyczne, wraz z pozostałymi danymi przypisanymi do danego np. klienta (w przypadku klienta - imię, nazwisko, ID, ilość wizyt, ilość wydanych pieniędzy)
- **Add book/music album** - Dodanie poszczególnych przedmiotów do baz i nadanie im parametrów.
- **Statistics** - Opcja nadająca sens całej aplikacji - wyłuskiwanie kluczowych informacji - który klient wydał najwięcej pieniędzy, który najczęściej odwiedza sklep, która książka/album muzyczny sprzedaje się najczęściej i ile razy została/został sprzedana/sprzedany.

2 Diagram klas, przypadków użycia, karty CRC

Diagram klas

Filip Zagórny, Weronika Janus | 20.05.2019

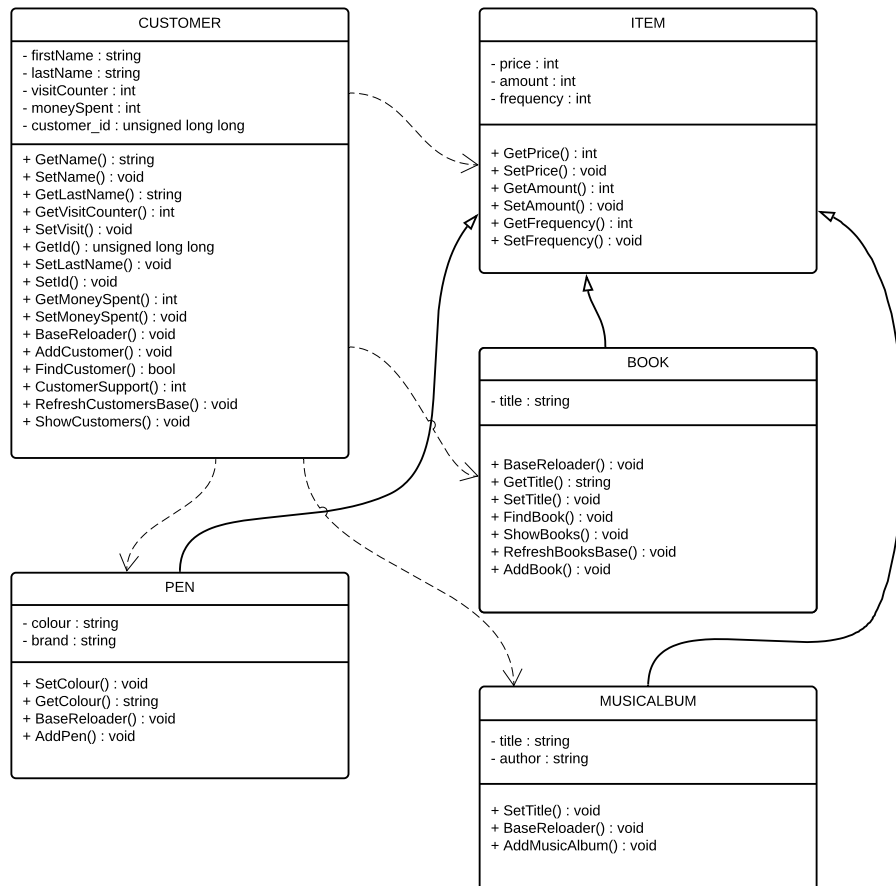


Diagram przypadków użycia

Filip Zagórny, Weronika Janus | 20.05.2019



Karty CRC

Filip Zagórny, Weronika Janus | 20.05.2019

CUSTOMER	
Obowiązki	Współpracownicy
<ul style="list-style-type: none"> • Zarządzanie danymi dotyczącymi klienta • Wygenerowanie unikalnego ID • Sprawdzanie czy jest w bazie • Robienie zakupów w sklepie 	<ul style="list-style-type: none"> • Item • Book • Pen • MusicAlbum

ITEM	
Obowiązki	Współpracownicy
<ul style="list-style-type: none"> • Zarządzanie danymi dotyczącymi przedmiotów • Dodanie przedmiotu do bazy • Usunięcie przedmiotu z bazy jeśli jego ilość wynosi 0 	<ul style="list-style-type: none"> • Customer

BOOK	
Obowiązki	Współpracownicy
<ul style="list-style-type: none"> • Zarządzanie danymi dotyczącymi książek 	<ul style="list-style-type: none"> • Customer

PEN	
Obowiązki	Współpracownicy
<ul style="list-style-type: none"> • Zarządzanie danymi dotyczącymi długopisów 	<ul style="list-style-type: none"> • Customer

MUSICALBUM	
Obowiązki	Współpracownicy
<ul style="list-style-type: none"> • Zarządzanie danymi dotyczącymi albumów muzycznych 	<ul style="list-style-type: none"> • Customer

3 Dokumentacja kodu

3.1 Klasa Customer

class Customer

Customer(); - **Konstrutkor**

Customer(std::string first, std::string last); **Konstrutkor**

Customer(); **Destrutkor**

std::string getName(); **Getter**

void setName(std::string const firstName); **Setter**

std::string GetlastName(); **Getter**

int GetVisitCounter(); **Getter**

void SetVisit(int visit); **Setter**

unsigned long long GetId(); **Getter**

void SetLastName(std::string const lastName); **Setter**

void SetId(unsigned long long *customer_id*); **Setter**

int GetMoneySpent(); **Getter**

void SetMoneySpent(int money); **Setter**

void BaseReloader(Customer customer, std::vector<Customer> list, std::fstream file); **Przeladowanie danych - wyczyszczenie 'starych' danych i uzupełnienie nowymi**

void AddCustomer(Customer customer, std::vector<Customer> list, std::fstream file, CustomerPopUpWindow customerpopupwindow); **Dodanie klienta do bazy - metoda otwiera plik tekstowy, który jest baza klientów, następnie wykorzystując funkcjonalności zawarte w bibliotekach Qt wyciąga dane wpisane przez użytkownika w rubryki.**

bool FindCustomer(std::vector<Customer> list, CustomerPopUpWindow customerpopupwindow); **Przeszukuje po imieniu i nazwisku, czy dany klient jest w bazie.**

int CustomerSupport(std::vector<Book> BooksBase, std::vector<Customer> CustomersBase, Cus-

tomServiceWindow customerservice); **Najważniejsza metoda w programie odpowiadająca za całą obsługę klienta - Wybranie np z listy książek tą, którą klient chce kupić.**

void ShowCustomers(std::vector<Customer> CustomersBase); **Wyświetlenie wszystkich klientów**

;

3.2 Klasa Book

Book(); - **Konstruktor**

Book(); **Destruktor**

void BaseReloader(Book book, std::vector<Book> list, std::fstream file); **Metoda odpowiadająca za aktualizacje danych w bazie z książkami**

std::string GetTitle(); **Getter**

void SetTitle(std::string title); **Setter**

void FindBook(std::string titleInput, std::vector<Book> BooksBase); **Wyszukiwanie książki w bazie**

void ShowBooks(std::vector<Book> BookBase); **Wyświetlenie wszystkich książek**

void AddBook(std::fstream file); **Dodanie książki do bazy**

3.3 Klasa Item

Item(); **Konstruktor**

virtual Item(); **Destruktor**

virtual std::string GetTitle(); **Getter**

int GetPrice(); **Getter**

void SetPrice(int price); **Setter**

int GetAmount(); **Getter**

void SetAmount(int amount); **Setter**

int GetFrequency(); **Getter**

void SetFrequency(int frequency); **Setter**

3.4 Klasa MusicAlbum

MusicAlbum(); **Konstruktor**

MusicAlbum(); **Destruktor**

void SetTitle(std::string title); **Setter**

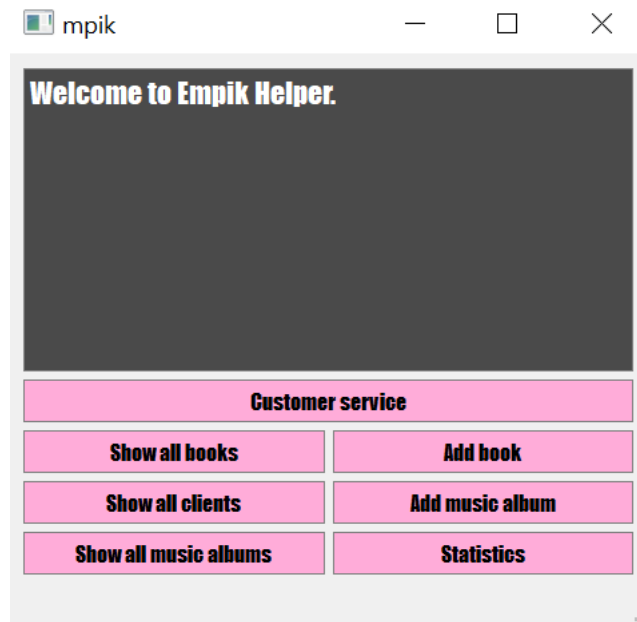
std::string GetTitle(); **Getter**

void BaseReloader(MusicAlbum musicalbum, std::vector<MusicAlbum> list, std::fstream file); **Zaktualizowanie danych**

void AddMusicAlbum(std::fstream file); **Dodanie albumu muzycznego do bazy**

4 Instrukcja obsługi

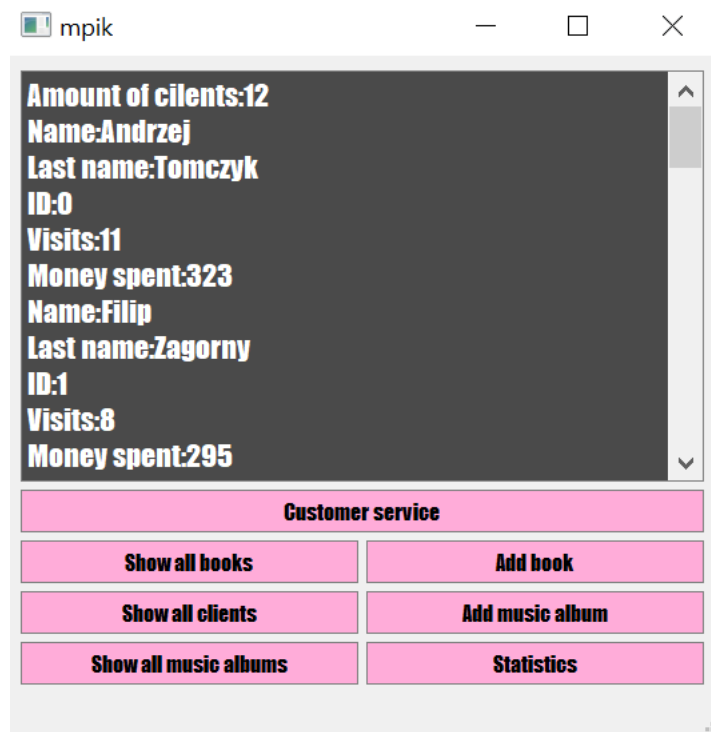
4.1 Panel startowy



Uruchamiając program, wyświetla się komunikat "Welcome to Empik Helper". Szary prostokąt to ekran, a pod nim znajdują się przyciski.

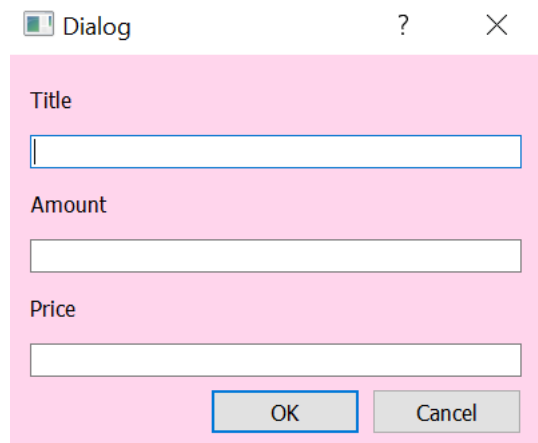
4.2 Przyciski w lewej kolumnie

Wszystkie spełniają tę samą funkcję - wyświetlenie wszystkich danych z poszczególnej bazy. Przykładowe zastosowanie dla "Show all clients":



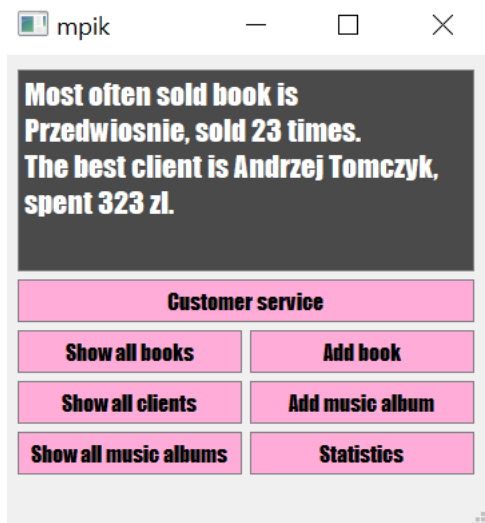
4.3 Przyciski "Add book" i "Add music album"

Przyciski pozwalają dodać przedmioty do bazy. Po wciśnięciu, wyskakuje okno dialogowe, takie jak poniżej:



Po wypełnieniu rubryk, wciskając "OK" dodamy przedmiot do bazy.

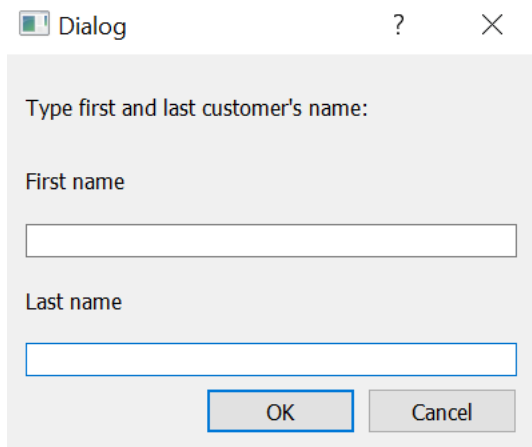
4.4 Przycisk "Statistics"



Po naciśnięciu na ekranie wyświetlane kluczowe dane: Co najczęściej się sprzedaje, ile razy, kto jest najlepszym klientem i ile pieniędzy wydał, itp.

4.5 Przycisk "Customer Service"

Po przyciśnięciu "Customer Service", pierwsze okno dialogowe wygląda następująco:



W nim wpisujemy imię i nazwisko klienta, następnie po kliknięciu przycisku "OK" pojawi się kolejne okno:

W nim wybieramy pozycję, która klient jest zainteresowany i wpisujemy jego ID.

Dialog ? X

Book title:

Dziady

ID of customer:

0

OK Cancel