

19.05.2023

Zespół projektowy nr 8

Amadeusz Reszke

Muhammad Zaindin

Patrycja Wierkin

Dokumentacja projektu aplikacji zarządzającej biblioteką.

1. Założenia stawiane przed aplikacją.

Na etapie projektowania przed aplikacją zostały postawione następujące wymagania:

- Zarządzanie listą wypożyczonych uprzednio i obecnie książek.
- Zarządzanie spisem znajdujących się na stanie książek.
- Zarządzanie listą klientów.
- Obsługa funkcjonalności kont pracowników i poziomów uprawnień.
- Zdolność prowadzenia statystyk.
- Zapewnienie bezpieczeństwa dostępu do informacji.
- Możliwość współdziałania wielu instancji aplikacji jednocześnie.

2. Analiza

Założenia, które zespół zdołał zrealizować:

- Zarządzanie listą wypożyczonych uprzednio i obecnie książek.

Aplikacja zawiera specjalną zakładkę, w której użytkownik może zarządzać nowymi i starymi wypożyczeniami. Korzysta przy tym z dynamicznych list dostępnych książek, autorów i klientów. Użytkownik ma również do dyspozycji filtry, wybierające wypożyczenia według zdefiniowanych kryteriów.

- Zarządzanie spisem znajdujących się na stanie książek.

W programie dostępna jest zakładka, poświęcona zarządzaniu listą znajdujących się na stanie pozycji. Zawiera ona mechanizmy do tego przeznaczone, m. in. dodawanie nowych książek czy modyfikację danych o ich ilości. Do tego dochodzi funkcjonalność zarządzania listą autorów książek czy wydawców.

- Zarządzanie listą klientów.

Aplikacja posiada dostęp do listy klientów oraz zawiera mechanizmy do zarządzania nią, takie jak dodawanie, usuwanie czy modyfikacja danych poszczególnych klientów.

- Obsługa funkcjonalności kont pracowników i poziomów uprawnień.

Każdy pracownik posiada swoje indywidualne konto, do którego przypisywane są wszystkie wykonywane przez niego akcje. Konto przypisany ma poziom uprawnień, określający możliwości danego użytkownika oraz zabezpieczone jest loginem i haszowanym hasłem.

- Zapewnienie bezpieczeństwa dostępu do informacji.

Wszelkie działania na danych i dostęp do nich są możliwe jedynie, gdy próbuje je wykonać uprawniona osoba. Zostało to osiągnięte za pomocą systemu kont i szyfrowanych haseł czy poziomów dostępu użytkowników. Aplikacja posiada także zabezpieczenie przed wstrzykiwaniem SQL.

- Możliwość współdziałania wielu instancji aplikacji jednocześnie.

Projekt przewiduje możliwość obsługi więcej niż jednej stacji roboczej w tym samym czasie bez konieczności przeprowadzania wyspecjalizowanej konfiguracji. Kolejne instancje aplikacji nie będą ze sobą kolidować.

Założenia, których zespół nie zrealizował:

- Zdolność prowadzenia statystyk.

Aplikacja nie jest w stanie prowadzić statystyk wypożyczeń, klientów ani użytkowników. Z powodu niedoboru czasu nie zostały przygotowane żadne rozwiązania w tej kwestii, jednakże pusta zakładka w aplikacji, na chwilę obecną ukryta, została przygotowana na poczet przyszłej realizacji.

Dodatkowo:

- Projekt nie zakłada osobnych aplikacji dla klienta i serwera.

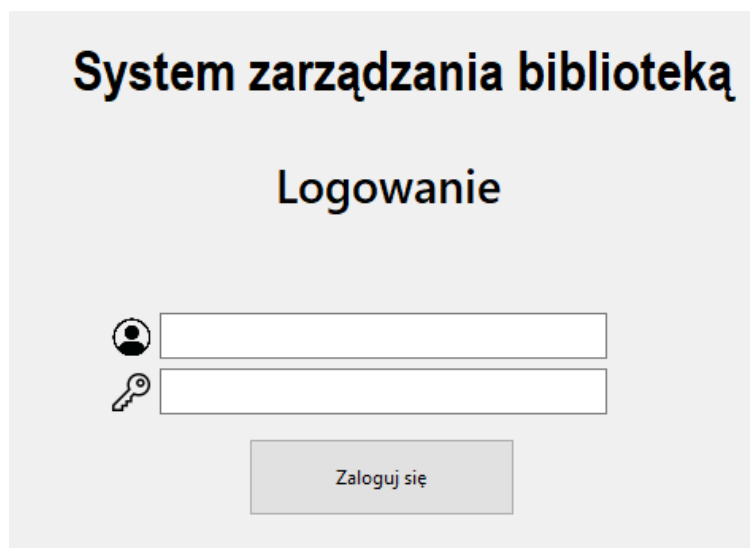
Oznacza to, że program klienta bezpośrednio komunikuje się z bazą danych, co jest bardziej niebezpieczne od modelu zawierającego dwie osobne aplikacje. Podział na część wewnętrzną i interfejs użytkownika zachodzi wewnątrz aplikacji.

- Projekt nie zapewnia pełnej walidacji wartości pobieranych z pól, które uzupełniają użytkownicy. Jako iż końcowymi użytkownikami aplikacji będą pracownicy biblioteki a nie jej klienci, istnienie pełnej walidacji nie było koniecznością. Na pracownikach spoczywa zatem obowiązek kontroli poprawności wprowadzanych danych (na przykład tytułów czy nazwisk).

- Projekt nie umożliwia przypisania jednej książki wielu autorów w sposób bezpośredni, można to zrobić, jedynie tworząc nowy rekord autora, reprezentujący wszystkich współautorów utworu.

3. Korzystanie z aplikacji

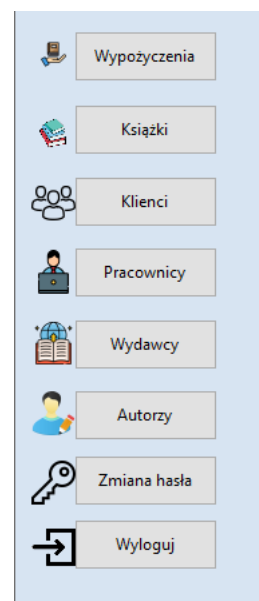
Do korzystania z aplikacji konieczne jest zalogowanie się za pomocą jednego z kont pracowników. Panel logowania włącza się domyślnie, po uruchomieniu aplikacji (fot 1). Przy logowaniu należy podać login i hasło konta. W przypadku podania błędnych danych aplikacja nie pozwoli na przejście dalej.



Fot. 1 – panel logowania

Po zalogowaniu, program przekieruje nas do głównego panelu aplikacji. Z lewej strony znajdują się przyciski kierujące nas do poszczególnych sekcji aplikacji (fot. 2). Oznaczone są one również wizualnymi ikonami.

W centralnej sekcji zakładki znajduje się tabela przedstawiająca listę elementów, których dana zakładka dotyczy (np. listę książek dla zakładki „Książki”, fot. 3). Dane w tabeli nie są edytowalne, ładowane są przy odświeżeniu panelu lub wyjściu z funkcjonalnego okna dialogowego. Możliwe jest zaznaczanie jednego wiersza lewym przyciskiem myszy (LPM), domyślnie zaznaczonym wierszem jest wiersz pierwszy (o indeksie 0). Zaznaczenie wiersza w tabeli wskazuje, jakiego rekordu mają dotyczyć operacje wywoływane przez przyciski funkcyjne, znajdujące się poniżej (fot. 4).

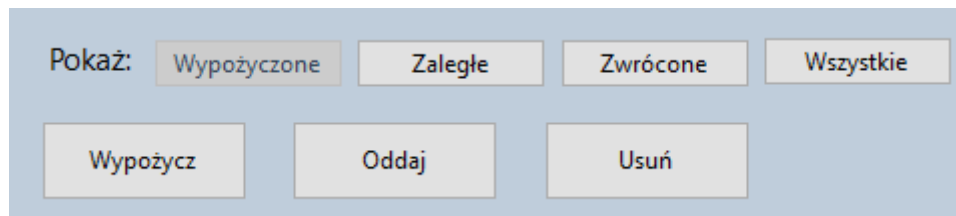


Fot. 2 – panel nawigacyjny

ID	Tytuł	Autor	Wydawca	Rok	Gatunek	Opis	Na stanie
2	Zielony Dom	Marcin Moskafa	Nowa Era	2003	kryminał		2 / 2
9	Magiczne drze...	Charles Dickens	Nowa Era	2003	fantasy		4 / 5
1	Hobbit	Marcin Karbo...	Skra	2023	nauka	Wspaniała przy...	2 / 2
10	Życie i przygod...	Aleksander Ma...	Skra	2021	horror	Epopeja	3 / 3
4	Puchatek	Aleksander Ma...	Dom	1945	kryminał	الجامعة البحرية ...	3 / 3

Fot. 3 – tabela rekordów

Przyciski funkcyjne reagują na zaznaczenie wiersza w tabeli, uaktywniając lub dezaktywując się w zależności od tego, czy operacja z nimi związana jest możliwa do wykonania na danym rekordzie. Przyciski są opisane w sposób umożliwiający łatwe określenie ich przeznaczenia, a ich naciśnięcie spowoduje wywołanie akcji którą reprezentują, lub otworenie nowego okna, służącego do wykonania tej operacji.



Fot. 4 – przyciski funkcyjne

Okna otwierane przy naciśnięciu niektórych z przycisków funkcyjnych, przeznaczone są do wprowadzenia dodatkowych danych, potrzebnych do zrealizowania żądanej operacji (fot. 5). Pola służące do wprowadzanie danych są opisane w odpowiedni sposób. Mogą w nich się znajdować tabele, działające podobnie do tych obecnych w podstawowych zakładkach aplikacji. Na każdym z takich okien znajdują się dwa przyciski: potwierdzający wykonanie danej operacji oraz przycisk służący do anulowania czynności i bez wprowadzania zmian w danych, powrotu do głównej części aplikacji. Są one opisane. Po potwierdzeniu operacji, w przypadku jej powodzenia użytkownik zostanie poinformowany poprzez wyskakujące okienko z komunikatem o sukcesie. W przypadku fiaska operacji, użytkownik zostanie poinformowany poprzez ujawnienie w widocznym miejscu ramki ze szczegółami błędu.

Przykładowe okno 'Nowe wypożyczenie' z formularzami i tabelami. Okno zawiera tytuł 'Nowe wypożyczenie' i przycisk zamykający. Formularz składa się z trzech głównych sekcji: 'Wybierz książkę:', 'Wybierz klienta:' i 'Wskaż dzień wypożyczenia:'. Każda sekcja zawiera tabelę z listą opcji i przyciskiem 'Wybrano:'. Na dole formularza znajdują się dwa przyciski: 'Wypożycz' i 'Anuluj'.

Wybierz książkę:

ID	Tytuł	Autor	Wydawca	Rok	Stan
2	Zielony Dom	Marcin Moskała	Nowa Era	2003	2
9	Magiczne drzewo	Charles Dickens	Nowa Era	2003	4
1	Hobbit	Marcin Karbowski	Skra	2023	2
10	Życie i przygody Re...	Aleksander Majkow...	Skra	2021	3
4	Puchatek	Aleksander Majkow...	Dom	1945	3

Wybrano: Marcin Moskała - Zielony Dom (Nowa Era, 2003)

Wybierz klienta:

ID	Imię i nazwisko
1	Jerzy Waszyngton
3	Maria Zielińska
4	Andrzej Kmicic
5	Jeremi Wiśniowiecki
6	Dawid Tylka
8	Adam Wąż
9	Karol Rathenow

Wybrano: Jerzy Waszyngton

Wskaż dzień wypożyczenia:

piątek , 26 maja 2023

☒ Dzisiaj

Wybrano 26.05.2023

Wskaż termin zwrotu:

piątek , 26 maja 2023

☐ Do zwrotu za dni: 0

Wybrano: 26.05.2023

Fot. 5 – przykładowe okno związane z funkcją

Uwaga! Zaleca się, by funkcje usuwania rekordów z bazy, opisane etykietą „Usuń”, wykonywane były jedynie w przypadku wprowadzenia do bazy przez pomyłkę błędnych rekordów. W przypadku wykorzystania wprowadzonego, błędnego rekordu w innych wierszach, jego usunięcie nie będzie już możliwe ze względu na jego powiązanie poprzez

relację w bazie danych z innym rekordem. W takim przypadku należy skontaktować się z administratorem bazy danych w celu poprawienia wprowadzonego błędu.

4. Dokumentacja aplikacji

Aplikację cechuje wirtualny podział na część zewnętrzną, frontend, zawierającą interfejs użytkownika, oraz część wewnętrzną, backend, która zawiera wszystkie procedury i metody służące do faktycznego wykonywania przez program operacji. Podział ten w praktyce umożliwia daleko idącą separację części wizualnej od części użytkowej do tego stopnia, że teoretycznie aplikacja jest w stanie pracować wywoływana programowo całkowicie bez interfejsu graficznego. Jest to założenie, którego należy przestrzegać przy dalszej rozbudowie projektu.

Projekt zawiera dwa główne zbiory klas. Pierwszy to klasy zapewniające obsługę interfejsu graficznego. Zaliczają się do nich:

- AddBookForm
- AddClientForm
- AddEmployeeForm
- ChangeBookQuantityForm
- ChangePasswordForm
- CustomerDeleteForm
- DeleteBookForm
- DeleteLendingForm
- EmployeeDeleteForm
- mainForm
- ModifyClientForm
- NewLendingForm
- ReturnLendingForm

Drugim zbiorem klas są klasy backendowe, zawierające procedury wykonywane przez program do zarządzania danymi w bazie. Zaliczają się do nich:

- ClientManagement
- LibraryManagement
- UserManagement
- UserSession

Zaleca się, by nowe klasy interfejsu graficznego tworzone przy rozbudowie projektu przestrzegały dotychczasowych zasad nazewnictwa oraz zasady separacji od faktycznych procedur aplikacji, a nowe procedury realizowane nie były przez klasy interfejsu, a specjalne klasy backendowe.

Ważną cechą aplikacji jest to, że wszystkie bezpośrednie polecenia do bazy danych, wysyłane są ze statycznej klasy DatabaseGateway. Zaleca się, by żadne inne klasy nie wysyłały do bazy żadnych zapytań, a jedynie korzystały z metod udostępnianych przez klasę do tego

przewidzianą. Poniżej jedna z metod służąca do wykonywania zapytań do bazy, które zwracają pojedynczą wartość.

```
public static string ExecuteScalarCommand(string sql, ref MySqlConnection
databaseConnection)
{
    //Funkcja pobierająca pojedynczą wartość z bazy, zwraca ją w formie stringa
    //W przypadku błędu zwraca pusty łańcuch
    ...
}
```

Dane o zalogowanym obecnie użytkowniku oraz aktualna połączenie z bazą danych przechowywane są w klasie `UserSession`. Klasa zawiera metody służące do zarządzania sesją użytkownika, np. `Login()` służąca do jego zalogowania.

```
public class UserSession
{
    //A class representing ongoing logged user session
    //loggedLogin being equal to "" means user is not logged in

    //id set to -1 means that user is not logged in
    private int loggedID;
    private string loggedLogin;
    private int privilege;
    private string firstName;
    private string lastName;

    private MySqlConnection databaseConnection;

    ...
}
```

Wskaźnik do instancji tej klasy jest przekazywany jako parametr do otwieranych przez przyciski funkcyjne okien, oraz do większości metod służących do wykonywania operacji na danych. Jako przykład fragment z konstruktorem okna zmiany hasła użytkownika `ChangePasswordForm`.

```
public partial class ChangePasswordForm : Form
{
    UserSession userSession;

    public ChangePasswordForm(ref UserSession userSession)
    {
        InitializeComponent();
        this.userSession = userSession;
    }

    ...
}
```

Aby wykonywać operacje na danych biblioteki, zbiorze klientów lub pracowników, należy wywoływać funkcje znajdujące w klasach: `ClientManagement`, `LibraryManagement`,

UserManagement. Funkcje są pogrupowane w klasach zgodnie ze zbiorem danych, którego dotyczą. Mają one nazwy, odzwierciedlające ich przeznaczenie. Dane potrzebne do ich wykonania należy przekazać w parametrze. Zwracają one wartości informujące o sukcesie lub niepowodzeniu operacji. Jako przykład fragment funkcji AddNewPublisher, dodającej nowego wydawcę do bazy, znajdującej się w klasie LibraryManagement.

```
public static int AddNewPublisher(ref UserSession userSession, string publisherName)
{
    /*
    * - FUNCTION RETURN VALUES -
    * 0 - successfull
    * 1 - not logged in
    * 3 - no permissions
    * 4 - sql/other error
    * 5 - invalid/empty parameters
    */

    ...
}
```

Konfiguracja bazy danych przechowywana jest w pliku *dbConfig.json*, znajdującym się w katalogu razem z głównym plikiem wykonywalnym programu. Nieobecność pliku lub nieprawidłowość jego zawartości uniemożliwi uruchomienie aplikacji. Prawidłowa zawartość pliku wygląda następująco:

```
{
  "server": "<serwer>",
  "user": "<login>",
  "password": "<hasło>",
  "database": "<baza danych>"
}
```

gdzie znaczniki *<serwer>*, *<login>*, *<hasło>* i *<baza danych>* zastąpione zostać muszą przez konfigurację bazy, kolejno: adres serwera na którym się ona znajduje, nazwę użytkownika bazy danych, hasło do tego konta oraz nazwę bazy danych (domyślnie „*library*”).

Aplikacja zapewnia również system zapisywania tekstowych informacji o błędach które wystąpiły do pliku tekstowego *log.txt*. Wpis z błędem zawiera znacznik czasowy jego wystąpienia oraz jego szczegóły. Przykładowa zawartość pliku z logami wygląda następująco:

```
18.05.2023 18:03:07 - Podano zły login przy próbie zalogowania użytkownika szymnek
18.05.2023 18:03:15 - Podano zły login przy próbie zalogowania użytkownika amares8
18.05.2023 18:06:26 - Próba wypożyczenia książki bez uprawnień, uzytkownik aolek
18.05.2023 18:06:41 - Błędne lub puste parametry przy próbie dodania nowego pracownika
```

Wpisywanie do pliku logów realizowane jest przez statyczną klasę Debugger i znajdującą się w niej funkcję CreateLogMessage. Znacznik czasowy dodawany jest automatycznie.

5. Dokumentacja bazy danych

Baza danych użyta w projekcie to relacyjna baza danych, korzystająca z języka SQL, pracująca na silniku MariaDB, dostępnym w pakiecie XAMPP. Dokładny opis bazy dostępny jest w dokumencie „Szczegółowy opis bazy danych”.

Baza nosi nazwę „library”, korzystać z niej można przy użyciu konta użytkownika o nazwie „librarian”, chronionego hasłem „Qwerty1@3”. Po fizycznej implementacji w obiekcie docelowym zaleca się zmianę hasła na inne niż domyślne.

Baza zawiera 9 tabel połączonych relacjami:

Tabela *bookauthors* służy do przechowywania autorów książek, które dostępne są w bibliotece.

Tabela *bookcategories* zawiera angielskie nazwy i identyfikatory kategorii książek. Kategorie te są przypisywane do każdej książki, i ułatwiają jej klasyfikację.

W tabeli *books* znajdują się wszystkie dostępne na stanie książki, wraz ze stanem magazynowym.

Tabela *customers* zawiera rekordy reprezentujące klientów zarejestrowanych w bibliotece. Są oni później przypisywani do każdego wypożyczenia.

Tabela *employees* przechowuje informacje o kontach pracowników biblioteki. Te konta są potem przypisywane do akcji wykonywanych przez użytkownika, np. do wypożyczenia klientowi książki. W tej tabeli znajdują się również dane logowania pracowników, kolumny: *login* i *password*. Hasło przechowywane w bazie jest zahaszowane.

Tabela *jobtitles* zawiera angielskie nazwy i identyfikatory stanowisk pracowników. Stanowiska te są przypisywane do każdego pracownika, i ułatwiają jego klasyfikację.

Tabela *lendings* to najważniejsza tabela, której rekordy reprezentują poszczególne wypożyczenia, zawierając informację o tym kto wypożyczył, co wypożyczył, czy i kiedy oddał itd. Jest to tabela w której spotykają się wszystkie relacje.

Tabela *lendingstatus* zawiera angielskie nazwy i identyfikatory statusów wypożyczenia (np. wypożyczony, oddany itp.). Statusy te są przypisywane do każdego wypożyczenia i określają stan, w jakim się ono znajduje.

Tabela *publishers* zawiera dane o wydawcach książek. Są one przypisywane do poszczególnych książek.

Nie należy usuwać poprawnie wprowadzonych rekordów z bazy. Aplikacja przewiduje odpowiednie oznaczanie zakończonych spraw, np. wypożyczeń, zamiast ich kasowania. Umożliwia to przeglądanie historii działania biblioteki oraz zbierania danych statystycznych. Aplikacja umożliwia usuwanie niektórych rekordów, jest to jednak zalecane tylko dla danych omyłkowo wprowadzonych.