

UNIWERSYTET RZESZOWSKI
WYDZIAŁ NAUK ŚCISŁYCH I TECHNICZNYCH
INSTYTUT INFORMATYKI



Kamil Kopczyk
134927

Informatyka

*Projekt i implementacja systemu do rezerwacji usług łowiska
wędkarskiego z wykorzystaniem bazy danych MySQL i języka
JAVA*

Praca projektowa

Praca wykonana pod kierunkiem
mgr inż. Ewa Żesławska

Rzeszów 2025

Spis treści

1. Streszczenie	7
1.1. Streszczenie w języku polskim	7
1.2. Summary in English	7
2. Opis założeń projektu	8
2.1. Cele projektu	8
2.2. Wymagania funkcjonalne	8
2.3. Wymagania нефункционалне	8
3. Opis struktury	9
3.1. Opis struktury projektu	9
3.2. Wykorzystane technologie i narzędzia	9
3.3. Hierarchia i architektura klas	9
3.4. Diagram klas UML	11
3.5. Zarządzanie danymi i baza danych łowisko	12
3.6. Wymagania systemowe i narzędzia do uruchomienia projektu	13
4. Harmonogram realizacji projektu	14
4.1. Harmonogram realizacji projektu. - Diagram Gantta	14
4.2. System kontroli wersji i repozytorium	14
5. Prezentacja warstwy użytkowej projektu	16
5.1. Wprowadzenie	16
5.2. Ekran logowania i rejestracji	16
5.3. Ekran rejestracji	17
5.4. Ekran po zalogowaniu	18
5.5. Ekran bez logowania	19
5.6. Ekran rezerwacji łowisk	20
5.7. Ekran rezerwacji wędek	21
5.8. Ekran rezerwacji domków	22
5.9. Ekran historia rezerwacji	23
6. Testowanie okienek błędów i innych elementów GUI	24
6.1. Wprowadzenie	24
6.2. Okienka ostrzegawcze ekranu logowania	24
6.3. Okienka ostrzegawcze ekranu rejestracji	26
6.4. Okienka ostrzegawcze ekranu rezerwacji	29
6.5. Okienka ostrzegawcze ekranu rezerwacji i historii bez logowania	31
7. Podsumowanie	33
7.1. Wnioski	33
7.2. Przyszłe kierunki rozwoju	33

Spis rysunków	34
----------------------------	-----------

1. Streszczenie

1.1. Streszczenie w języku polskim

Celem projektu było stworzenie systemu do rezerwacji usług łowiska wędkarskiego. Aplikacja składa się z formularza logowania i formularza rejestracji który pozwala utworzyć konto do którego są przypisywane rezerwacje później logować się na nie rezerwować łowiska, wynajmować wędki, domki i przeglądać historię rezerwacji. System został napisany w języku **Java**, jego interfejs graficzny powstał przy użyciu biblioteki **Swing**, a za przechowywanie danych odpowiada baza **MySQL**.

1.2. Summary in English

The aim of the project was to create a system for booking fishing grounds services. The application consists of a login form and a registration form that allows you to create an account to which reservations are assigned, then log in to them, book fishing grounds, rent fishing rods, cottages and view the history of reservations. The system was written in **Java**, its graphical interface was created using the **Swing** library, and the **MySQL** database is responsible for storing data.

2. Opis założeń projektu

2.1. Cele projektu

Celem projektu było stworzenie systemu do rezerwacji usług łowiska wędkarskiego, plan zakładał budowę aplikacji która zapewni prostą obsługę rezerwacji łowisk, wędek i domków po przez proste schludne GUI jak i przeglądanie historii rezerwacji. Całość ma ułatwić i w pewnym sensie zautomatyzować obsługę kompleksu łowisk na tyle że klient tylko przyjeżdża odbiera swoją rezerwację i idzie się cieszyć wędkowaniem bez żadnych zbędnych dokumentów do wypełnienia.

2.2. Wymagania funkcjonalne

Aplikacja będzie oferować logowanie się na istniejące już konto w bazie danych MySQL lub rejestrację nowego konta, pozwala również na przeglądanie bez logowania jednak tylko użytkownicy zalogowani mogą korzystać w pełni z aplikacji. Każdy zalogowany użytkownik będzie miał swoją historię rezerwacji która przekazuje mu informacje od kiedy do kiedy trwała dana rezerwacja ile kosztowała i czego dotyczyła. Aplikacja będzie zrobiona bardzo intuicyjnie co pozwoli nawet tym mniej obeznanym z technologią ludziom przejść przez nią bez żadnego problemu z uśmiechem na twarzy.

2.3. Wymagania niefunkcjonalne

Poza samą funkcjonalnością założeniem jest żeby z aplikacji korzystało się efektywnie, przyjemnie i bezproblemowo. Dlatego zarówno osoba nie zalogowana jak i zalogowana ma czuć że program działa intuicyjnie i płynnie bez żadnego zbędnego czekania na odpowiedź aplikacji. Ważne jest też żeby aplikacja była przygotowana na nieprzewidziane awarie, takie jak na przykład problem z dostępem do bazy danych. Zapewnione będą również scenariusze awaryjne na takie sytuację jak na przykład nie pobranie jakiegoś id z bazy danych jak i scenariusze na wypadek błędu ze strony użytkownika aplikacji jak na przykład nie prawidłowy format daty lub wybranie w polu startu rezerwacji datę późniejszą niż data rozpoczęcia. Dlatego też system powinien w zrozumiały dla użytkownika sposób po-informować go o zaistniałym problemie. W dodatku dzięki zastosowaniu technologii Java, aplikacja będzie uniwersalna.

3. Opis struktury

3.1. Opis struktury projektu

W tym rozdziale zostanie przedstawiona struktura projektu, wraz z opisem poszczególnych elementów. Projekt jest podzielony na kilka głównych katalogów, z których każdy pełni określoną funkcję. Rozdział ten zamknie omówienie klas i ich funkcji w projekcie, co pozwoli na lepsze zrozumienie struktury i organizacji kodu. Skupi się też na wymaganiach do uruchomienia projektu, które są niezbędne do poprawnego działania aplikacji.

3.2. Wykorzystane technologie i narzędzia

Rdzeń systemu został zbudowany w oparciu o język Java, który zapewnił wysoką wydajność i niezawodność działania aplikacji. Interfejs użytkownika został zaimplementowany przy użyciu biblioteki Java Swing, oferującej bogaty zestaw komponentów graficznych.

Warstwa danych wykorzystuje następujące technologie:

MySQL jako system zarządzania relacyjną bazą danych

JDBC (Java Database Connectivity) jako interfejs łączący aplikację z bazą danych

Zaawansowane zapytania SQL z optymalizacją pod kątem wydajności

Środowisko developerskie obejmowało:

IntelliJ IDEA jako główne środowisko programistyczne

Git jako system kontroli wersji z repozytorium hostowanym na GitHubie

Zintegrowane narzędzia do debugowania i profilowania kodu

Cała architektura została zaprojektowana z naciskiem na:

Wydajność połączenia z bazą danych

Przejrzystość interfejsu użytkownika

Łatwość rozszerzania funkcjonalności

Bezpieczeństwo przechowywanych danych

3.3. Hierarchia i architektura klas

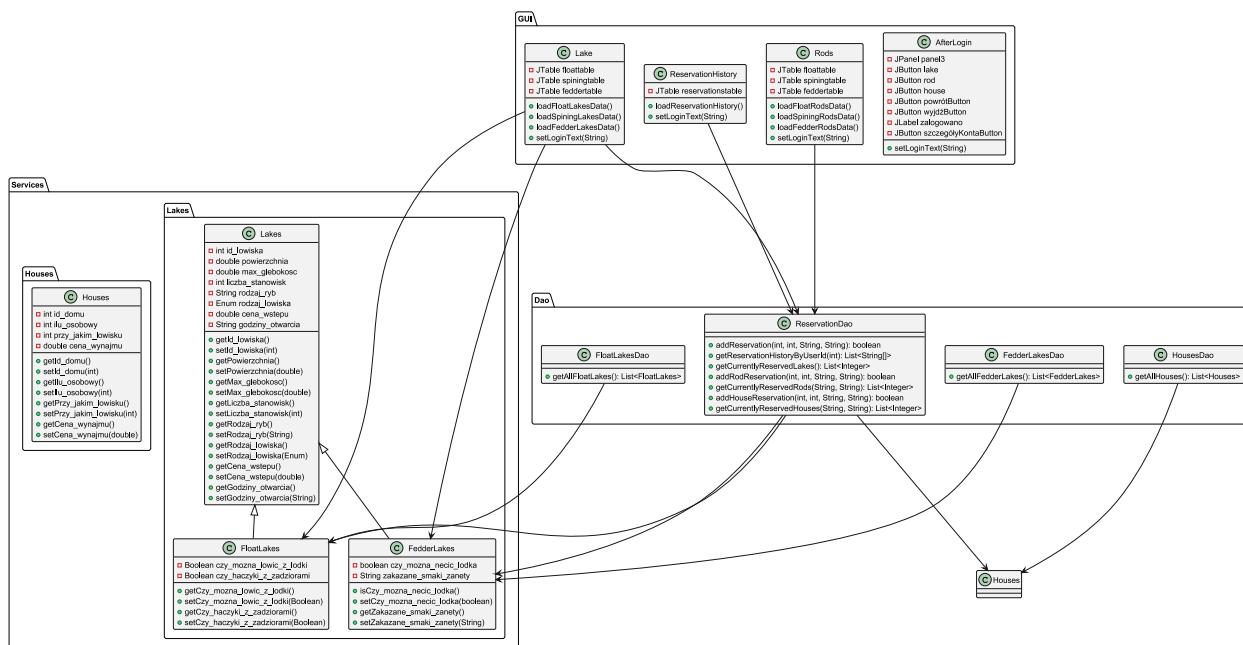
Projekt jest zorganizowany w sposób hierarchiczny, gdzie główną klasą jest 'Main', która uruchamia aplikację. Poniżej przedstawiono strukturę klas wraz z ich funkcjami:

- **Warstwa Dostępu do baz danych (Dao):** Ta warstwa odpowiada za komunikację z bazą danych MySQL. Zawiera klasy takie jak 'AddUsers', 'ReservationDao', 'HousesDao' itd., które implementują metody do wykonywania operacji CRUD (Create, Read, Update, Delete) na odpowiednich tabelach w bazie danych.

- **Warstwa Prezentacji (GUI):** Ta warstwa jest odpowiedzialna za interakcję z użytkownikiem. Zawiera klasy takie jak 'AfterLogin', 'Register', 'MainMenu' itd., które implementują graficzny interfejs użytkownika (GUI) przy użyciu biblioteki Swing. Użytkownik może logować się, rejestrować nowe konto oraz przeglądać dostępne usługi.
- **Warstwa Logiki Biznesowej (Services):** Ta warstwa zawiera klasy, które implementują logikę biznesową aplikacji. Przykładowe klasy to 'Lakes', 'Houses', 'FishingRods' itd. Odpowiadają one za przetwarzanie danych i wykonywanie operacji na obiektach reprezentujących rezerwacje, domki i wędki.

3.4. Diagram klas UML

Diagram klas (Rys. 3.1) to wizualna „mapa” projektu. Przedstawia on relacje między klasami, ich atrybuty i metody. Diagram ten jest kluczowym narzędziem do zrozumienia struktury projektu i jego komponentów. Umożliwia on szybkie zorientowanie się w hierarchii klas oraz ich wzajemnych powiązaniach, co jest szczególnie przydatne podczas rozwoju i utrzymaniu projektu.

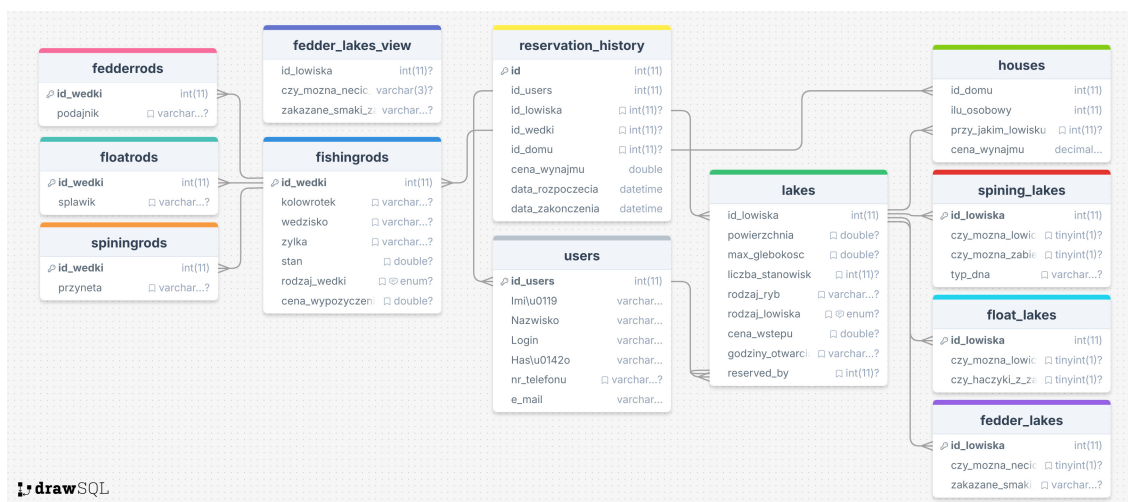


Rys. 3.1. Diagram klas UML projektu.

Źródło: Opracowane przy użyciu PlantUML

3.5. Zarządzanie danymi i baza danych łowisko

Baza danych MySQL jest kluczowym elementem projektu, przechowującym wszystkie istotne informacje o użytkownikach, rezerwacjach, łowiskach i innych zasobach. Struktura bazy danych została zaprojektowana w sposób umożliwiający łatwe zarządzanie danymi oraz ich efektywne przetwarzanie. Baza danych składa się z kilku tabel, które są ze sobą powiązane relacjami. Schemat bazy danych (Rys. 3.2) przedstawia strukturę tabel oraz ich relacje.



Rys. 3.2. Schemat bazy danych (ERD) łowiska wędkarskiego.

Źródło: Wygenerowano za pomocą <https://drawsql.app>

3.6. Wymagania systemowe i narzędzia do uruchomienia projektu

Aby uruchomić projekt, wymagane są następujące narzędzia i oprogramowanie:

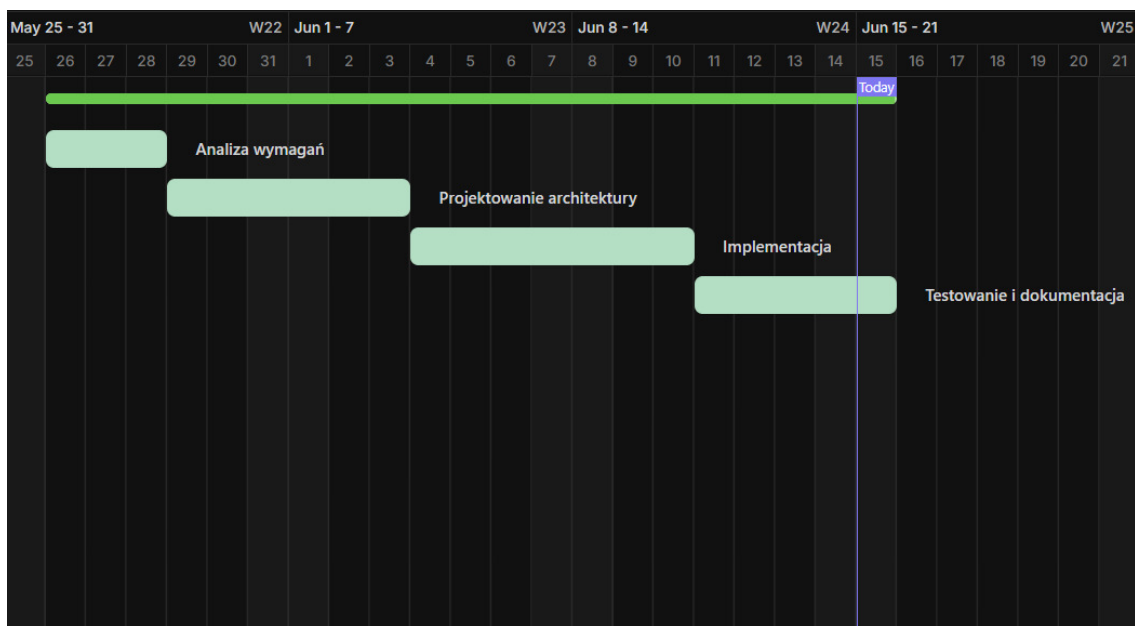
- **Pakiet XAMPP:** Jest to pakiet oprogramowania, który zawiera serwer Apache, MySQL oraz PHP. Jest niezbędny do uruchomienia bazy danych MySQL, która jest kluczowym elementem projektu.
- **Środowisko IntelliJ IDEA:** Jest to zintegrowane środowisko programistyczne (IDE) dla języka Java, w którym projekt został stworzony. IntelliJ IDEA oferuje zaawansowane funkcje, takie jak automatyczne uzupełnianie kodu, refaktoryzacja i debugowanie, co znacznie ułatwia pracę programisty.
- **Java Development Kit (JDK):** Wersja JDK 17 lub nowsza jest wymagana do kompilacji i uruchomienia aplikacji. JDK zawiera wszystkie niezbędne biblioteki i narzędzia do pracy z językiem Java.
- **MySQL Connector/J:** Jest to sterownik JDBC dla MySQL, który umożliwia aplikacji Java komunikację z bazą danych MySQL. Należy go dodać do project structure w modułach w projekcie IntelliJ IDEA.

4. Harmonogram realizacji projektu.

4.1. Harmonogram realizacji projektu. - Diagram Gantta

Harmonogram realizacji projektu jest kluczowym elementem planowania, który pozwala na efektywne zarządzanie czasem i zasobami. W projekcie dotyczącym systemu rezerwacji usług łowiska wędkarskiego, harmonogram został opracowany z uwzględnieniem wszystkich istotnych etapów, od analizy wymagań po testowanie i wdrożenie.

- **Analiza wymagań:** Zbieranie i analiza wymagań funkcjonalnych i нефункциональных systemu.
- **Projektowanie architektury:** Opracowanie struktury projektu, w tym diagramów UML i schematu bazy danych.
- **Implementacja:** Programowanie poszczególnych komponentów systemu, takich jak warstwa dostępu do danych, logika biznesowa i interfejs użytkownika.
- **Testowanie i dokumentacja:** Przeprowadzenie testów jednostkowych i integracyjnych, a także przygotowanie dokumentacji użytkownika i technicznej.



Rys. 4.1. Harmonogram realizacji projektu (Diagram Gantta).

Źródło: Wygenerowano za pomocą <https://clickup.com>

4.2. System kontroli wersji i repozytorium

W projekcie zastosowano system kontroli wersji Git, który umożliwia śledzenie zmian w kodzie źródłowym, zarządzanie wersjami oraz współpracę zespołową. Git pozwala na tworzenie gałęzi (branch), co umożliwia równoległe prace nad różnymi funkcjonalnościami bez wpływu na główną linię rozwoju projektu.

Jako centralne, zdalne repozytorium kodu wykorzystano platformę GitHub. Cały projekt jest publicznie dostępny pod adresem:

https://github.com/KamilKopczyk/Projekt_PO_JAVA_Kamil_Kopczyk_134927

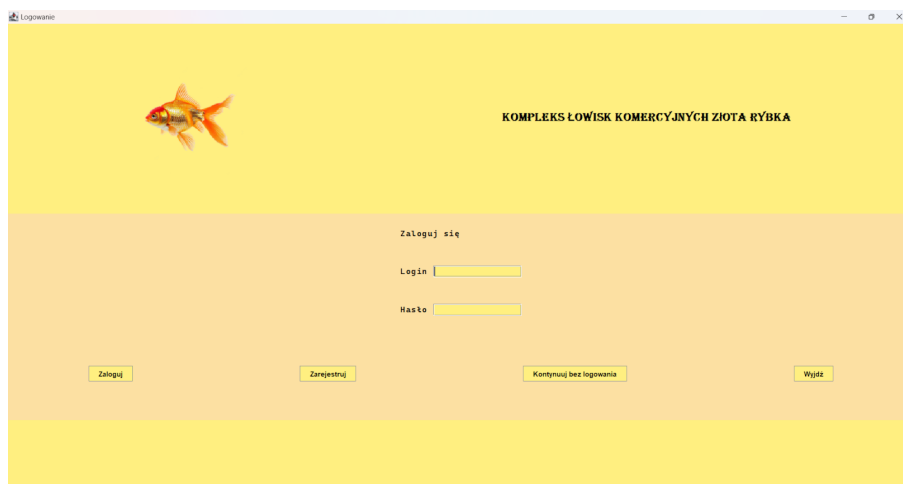
5. Prezentacja warstwy użytkowej projektu

5.1. Wprowadzenie

W tym rozdziale zostanie przedstawiona warstwa użytkowa projektu, która obejmuje interfejs graficzny aplikacji oraz sposób interakcji użytkownika z systemem. Warstwa ta została zaprojektowana z myślą o intuicyjności i łatwości obsługi, co jest kluczowe dla zapewnienia pozytywnego doświadczenia użytkownika.

5.2. Ekran logowania i rejestracji

Ekran logowania jest pierwszym krokiem, z jakim spotyka się użytkownik po uruchomieniu aplikacji. Umożliwia on zalogowanie się na istniejące konto lub rejestrację nowego użytkownika. Interfejs jest prosty i przejrzysty, co pozwala na szybkie wprowadzenie danych.




Rys. 5.1. Ekran logowania.

Źródło: Opracowane przy użyciu Java Swing

5.3. Ekran rejestracji

Ekran rejestracji pozwala na tworzenie nowych kont użytkowników. Użytkownik musi wprowadzić swoje dane, takie jak nazwa użytkownika, hasło i adres e-mail. Po poprawnym wypełnieniu formularza, użytkownik może zalogować się do aplikacji.



Strona główna

KOMPLEKS ŁOWISK KOMERCYJNYCH ZIOTA RYBKA

Zarejestruj

Imię

Nazwisko

Login

Hasło

Powtórz hasło

nr_telefonu

e_mail

Rys. 5.2. Ekran rejestracji.

Źródło: Opracowane przy użyciu Java Swing

5.4. Ekran po zalogowaniu

Po zalogowaniu użytkownik zostaje przekierowany do głównego menu aplikacji, gdzie może przeglądać dostępne usługi, takie jak rezerwacja łowisk, wędek i domków. Interfejs jest zaprojektowany w sposób umożliwiający łatwe nawigowanie między różnymi funkcjami.



Rys. 5.3. Ekran po zalogowaniu.

Źródło: Opracowane przy użyciu Java Swing

5.5. Ekran bez logowania

Ekran bez logowania umożliwia przeglądanie dostępnych usług bez konieczności logowania się. Różni się też tym że z racji że się nie logujemy to zamiast loginu w Zalogowano jako: wyświetla się Gość. Użytkownik może zobaczyć listę łowisk, wędek i domków, ale nie ma możliwości dokonywania rezerwacji ani przeglądania historii rezerwacji.



Rys. 5.4. Ekran bez logowania.

Źródło: Opracowane przy użyciu Java Swing

5.6. Ekran rezerwacji łowisk

Ekran rezerwacji łowiska umożliwia użytkownikowi dokonanie rezerwacji na wybrane łowisko. Użytkownik może wybrać datę rozpoczęcia i zakończenia rezerwacji, a także sprawdzić dostępność łowisk. Interfejs jest intuicyjny i pozwala na łatwe przeglądanie dostępnych opcji.

KOMPLEKS ŁOWISK KOMERCYJNYCH ZIOTA RYBKA

Zalogowano jako: kk134927

Szczegóły konta

Spławikowe		Spiningowe		Gruntowe						
ID Łowiska	Powierzchnia	Max Głębokość	Liczba Stawisk	Rodzaj Ryb	Rodzaj Łowiska	Cena Wstępu	Godziny Otwarcia	Czy Można Łowić z Łodzi	Czy Wycieczki z Zasiadaniem	
1	120.0	8.0	14	Leszcz, Wzdrę...	SPLAWIKOWE	20.0	7-18	Tak	Tak	
2	100.0	6.0	12	Wzdręga, Płoc...	SPLAWIKOWE	30.0	6-19	Nie	Nie	
3	80.0	4.0	10	Karp, Lin, Amur	SPLAWIKOWE	35.0	6-20	Nie	Tak	

Początek rezerwacji: 2025-06-15 16:12:22

Koniec rezerwacji: 2025-06-15 16:12:22

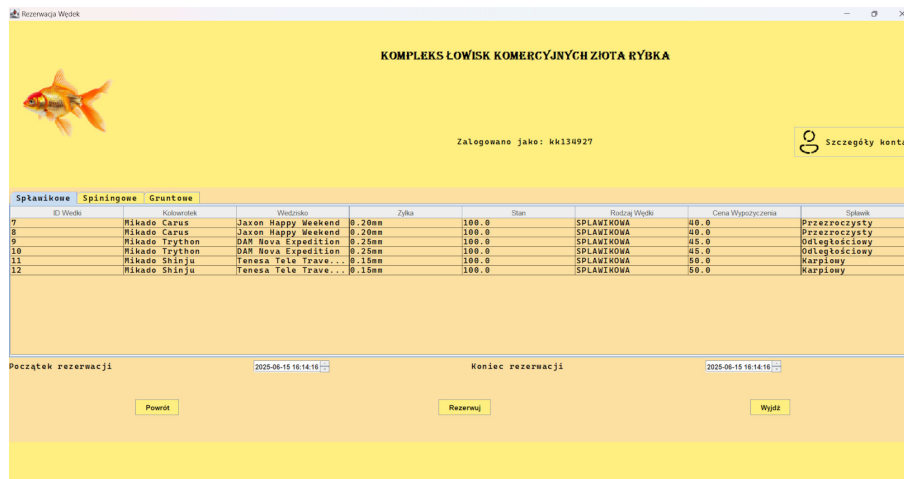
Powrót Rezerwuj Wydaj

Rys. 5.5. Ekran rezerwacji łowisk.

Źródło: Opracowane przy użyciu Java Swing

5.7. Ekran rezerwacji wędek

Ekran rezerwacji wędek pozwala użytkownikowi na wybór dostępnych wędek do rezerwacji. Użytkownik może przeglądać dostępne modele i dokonać rezerwacji na wybraną wędkę. Interfejs jest zaprojektowany w sposób umożliwiający łatwe poruszanie się po dostępnych opcjach.



Rys. 5.6. Ekran rezerwacji wędek.

Źródło: Opracowane przy użyciu Java Swing

5.8. Ekran rezerwacji domków

Ekran rezerwacji domków umożliwia użytkownikowi dokonanie rezerwacji na wybrany domek. Użytkownik może przeglądać dostępne domki, sprawdzać ich dostępność i dokonywać rezerwacji. Interfejs jest przyjazny dla użytkownika i pozwala na łatwą rezerwację.

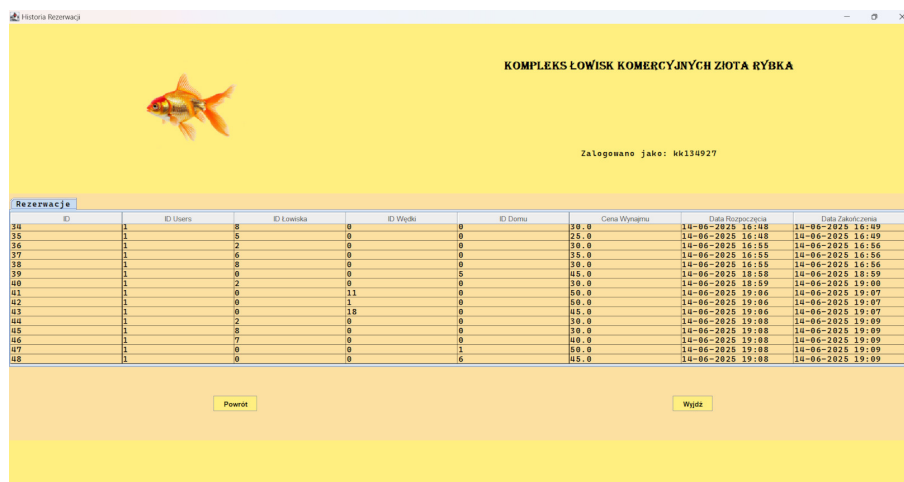
ID Domku	Nazwa Domku	Cena Wynajmu	Status Rezerwacji
1	Domki	800.0	
2	Domki	600.0	
3	Domki	600.0	
4	Domki	600.0	
5	Domki	600.0	
6	Domki	600.0	

Rys. 5.7. Ekran rezerwacji domków.

Źródło: Opracowane przy użyciu Java Swing

5.9. Ekran historia rezerwacji

Ekran historii rezerwacji pozwala użytkownikowi na przeglądanie wszystkich dokonanych rezerwacji. Użytkownik może zobaczyć szczegóły każdej rezerwacji, takie jak daty, koszty i rodzaje usług.



KOMPLEKS ŁOWISK KOMERCYJNYCH ZIOTA RYBKA

Zalogowano jako: kk134927

ID	ID Users	ID Łowiska	ID Wędk	ID Domu	Cena Wynajmu	Data Rozpoczęcia	Data Zakończenia
35	1	5	0	0	20.0	14-06-2025 16:48	14-06-2025 16:49
35	1	5	0	0	25.0	14-06-2025 16:48	14-06-2025 16:49
36	1	2	0	0	30.0	14-06-2025 16:55	14-06-2025 16:56
37	1	6	0	0	35.0	14-06-2025 16:55	14-06-2025 16:56
38	1	8	0	0	30.0	14-06-2025 16:55	14-06-2025 16:56
39	1	0	0	5	45.0	14-06-2025 18:59	14-06-2025 18:59
40	1	2	0	0	30.0	14-06-2025 18:59	14-06-2025 18:59
41	1	0	11	0	50.0	14-06-2025 19:06	14-06-2025 19:07
42	1	0	1	0	50.0	14-06-2025 19:06	14-06-2025 19:07
43	1	0	18	0	45.0	14-06-2025 19:06	14-06-2025 19:07
44	1	2	0	0	30.0	14-06-2025 19:08	14-06-2025 19:09
45	1	8	0	0	30.0	14-06-2025 19:08	14-06-2025 19:09
46	1	7	0	0	40.0	14-06-2025 19:08	14-06-2025 19:09
47	1	0	0	1	50.0	14-06-2025 19:08	14-06-2025 19:09
48	1	0	0	6	45.0	14-06-2025 19:08	14-06-2025 19:09

Powrót Wydruk

Rys. 5.8. Ekran historii rezerwacji.

Źródło: Opracowane przy użyciu Java Swing

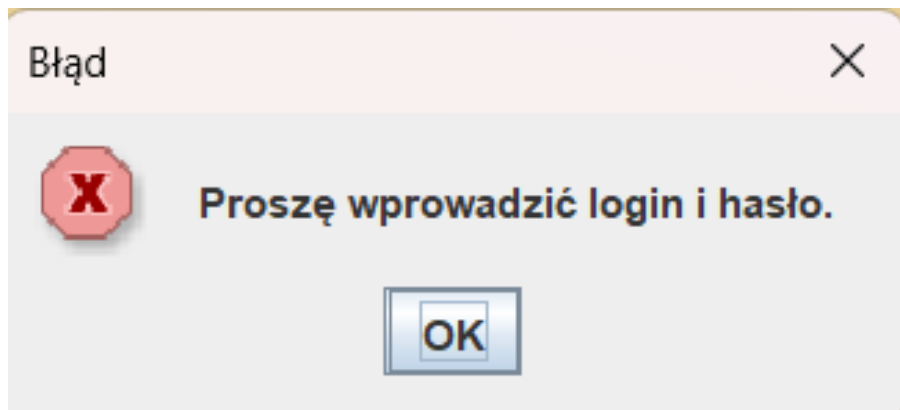
6. Testowanie okienek błędów i innych elementów GUI

6.1. Wprowadzenie

W tym rozdziale zostaną przedstawione testy przeprowadzone na okienkach błędów oraz innych elementach graficznego interfejsu użytkownika (GUI) aplikacji. Testowanie GUI jest kluczowym etapem w procesie tworzenia oprogramowania, ponieważ pozwala na identyfikację i naprawę błędów, które mogą wpływać na doświadczenie użytkownika.

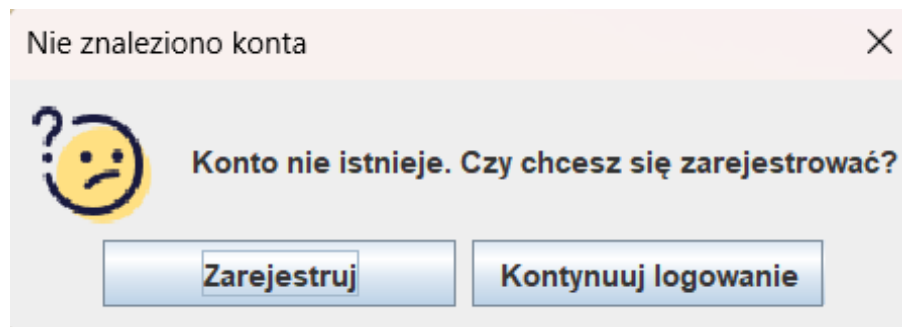
6.2. Okienka ostrzegawcze ekranu logowania

Okienka ostrzegawcze na ekranie logowania są istotnym elementem interfejsu użytkownika, który informuje o błędach wprowadzonych danych. Testy te mają na celu sprawdzenie, czy okienka poprawnie reagują na nieprawidłowe dane wejściowe.



Rys. 6.1. Brak danych w polach logowania.
Źródło: Opracowane przy użyciu Java Swing

Podczas testów sprawdzano czy jak ktoś wpisze błędne dane logowania to czy pyta go czy chce się zarejestrować, czy kontynuować logowanie.

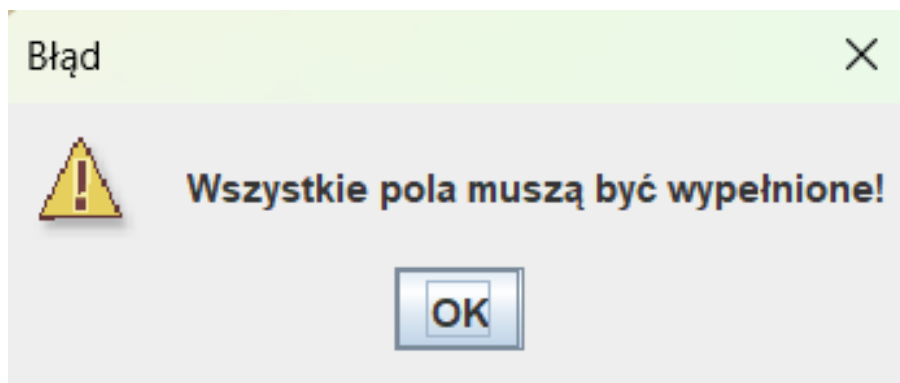


Rys. 6.2. Błędne dane logowania.

Źródło: Opracowane przy użyciu Java Swing

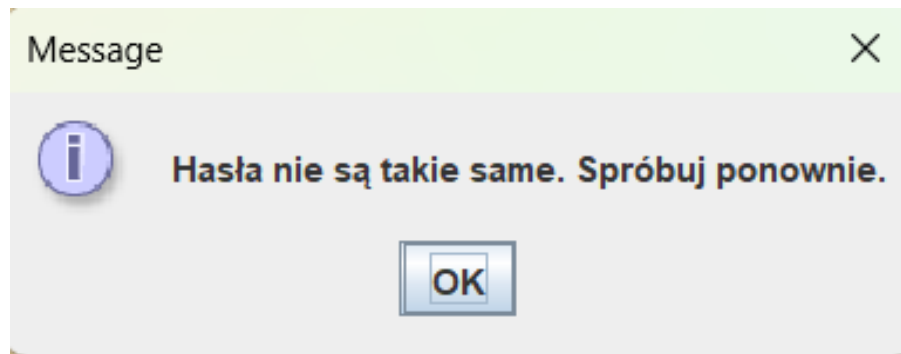
6.3. Okienka ostrzegawcze ekranu rejestracji

Okienka ostrzegawcze na ekranie rejestracji informują użytkownika o błędach wprowadzonych danych podczas tworzenia nowego konta. Testy te mają na celu sprawdzenie, czy aplikacja poprawnie reaguje na nieprawidłowe dane wejściowe, takie jak brak wymaganych pól lub nieprawidłowy format danych.



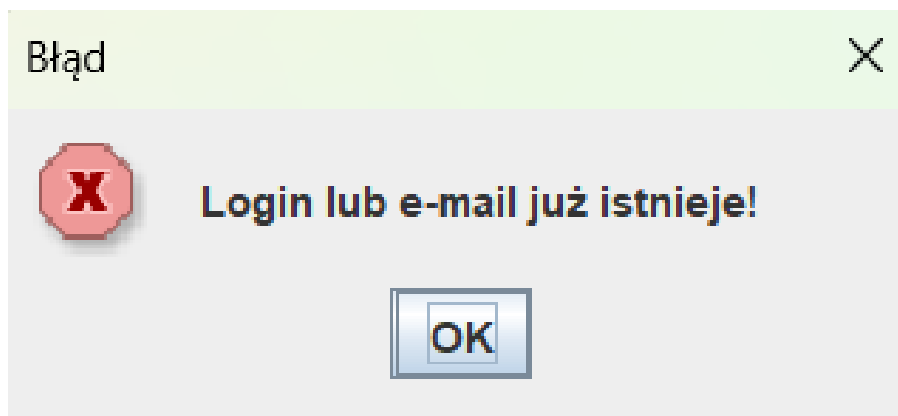
Rys. 6.3. Brak danych w polach rejestracji.

Źródło: Opracowane przy użyciu Java Swing



Rys. 6.4. Sprawdzanie czy hasła są identyczne.

Źródło: Opracowane przy użyciu Java Swing

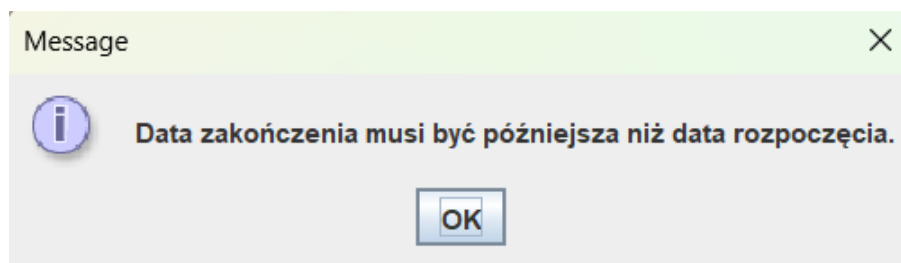


Rys. 6.5. Sprawdzanie czy login lub e-mail nie są już przez kogoś używane.

Źródło: Opracowane przy użyciu Java Swing

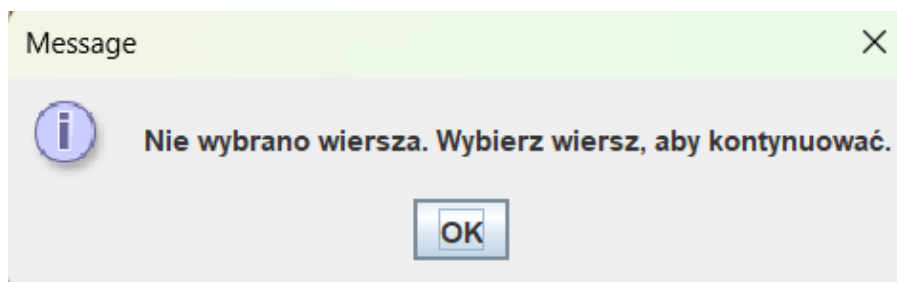
6.4. Okienka ostrzegawcze ekranu rezerwacji

Okienka ostrzegawcze na ekranie rezerwacji informują użytkownika o błędach związanych z rezerwacją, takich jak nieprawidłowy format daty lub próba dokonania rezerwacji w przeszłości. Testy te mają na celu sprawdzenie, czy aplikacja poprawnie reaguje na nieprawidłowe dane wejściowe i czy informuje użytkownika o konieczności wprowadzenia poprawnych danych.



Rys. 6.6. Data startu rezerwacji jest późniejsza niż końca rezerwacji.

Źródło: Opracowane przy użyciu Java Swing

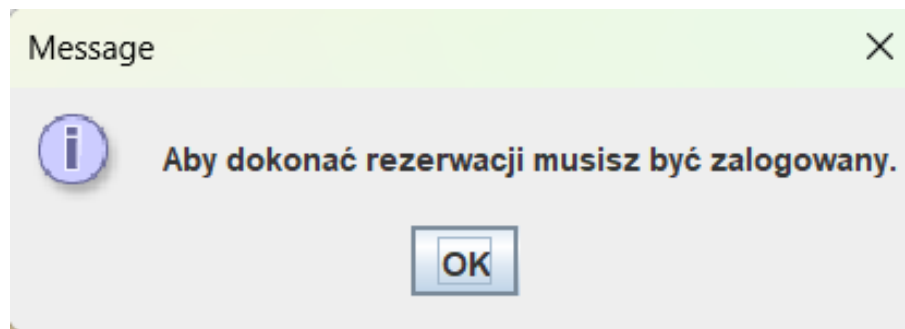


Rys. 6.7. Nie wybranie wiersza do rezerwacji.

Źródło: Opracowane przy użyciu Java Swing

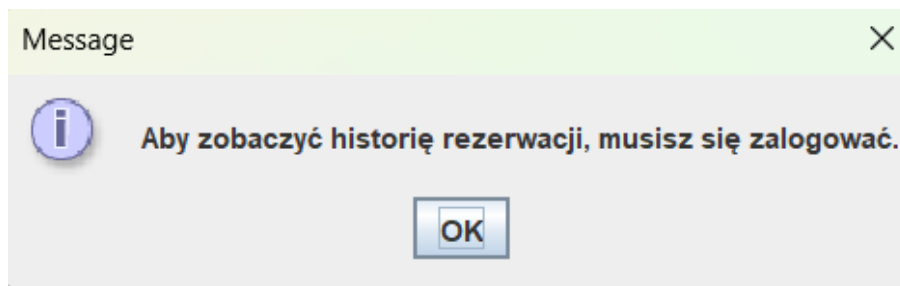
6.5. Okienka ostrzegawcze ekranu rezerwacji i historii bez logowania

Okienka ostrzegawcze na ekranie rezerwacji bez logowania informują użytkownika o konieczności zalogowania się, aby móc dokonać rezerwacji. Testy te mają na celu sprawdzenie, czy aplikacja poprawnie reaguje na próby dokonania rezerwacji przez niezalogowanych użytkowników i czy informuje ich o konieczności zalogowania się.



Rys. 6.8. Próba rezerwacji bez zalogowania.

Źródło: Opracowane przy użyciu Java Swing



Rys. 6.9. Próba sprawdzenia historii rezerwacji bez zalogowania.

Źródło: Opracowane przy użyciu Java Swing

7. Podsumowanie

7.1. Wnioski

Projekt systemu rezerwacji usług łowiska wędkarskiego został zrealizowany zgodnie z założeniami i wymaganiami określonymi na początku prac. Aplikacja umożliwia użytkownikom łatwe zarządzanie rezerwacjami łowisk, wędek i domków, a także zapewnia intuicyjny interfejs graficzny. Zastosowanie technologii Java oraz biblioteki Swing pozwoliło na stworzenie aplikacji, która jest zarówno funkcjonalna, jak i estetyczna. System został zaprojektowany z myślą o wydajności, bezpieczeństwie i łatwości obsługi, co zostało osiągnięte poprzez odpowiednią strukturę kodu.

7.2. Przyszłe kierunki rozwoju

W przyszłości planowane jest rozszerzenie funkcjonalności aplikacji o dodatkowe usługi, takie jak estetyczniejsze wybory dat, możliwość dodawania opinii o łowiskach i domkach. Dodatkowo rozważane jest wprowadzenie możliwości resetowania hasła przez użytkowników, co zwiększy bezpieczeństwo i wygodę korzystania z aplikacji.

Spis rysunków

3.1	Diagram klas UML projektu.	11
3.2	Schemat bazy danych (ERD) łowiska wędkarskiego.	12
4.1	Harmonogram realizacji projektu (Diagram Gantta).	14
5.1	Ekran logowania.	16
5.2	Ekran rejestracji.	17
5.3	Ekran po zalogowaniu.	18
5.4	Ekran bez logowania.	19
5.5	Ekran rezerwacji łowisk.	20
5.6	Ekran rezerwacji wędek.	21
5.7	Ekran rezerwacji domków.	22
5.8	Ekran historii rezerwacji.	23
6.1	Brak danych w polach logowania.	24
6.2	Błędne dane logowania.	25
6.3	Brak danych w polach rejestracji.	26
6.4	Sprawdzanie czy hasła są identyczne.	27
6.5	Sprawdzanie czy login lub e-mail nie są już przez kogoś używane.	28
6.6	Data startu rezerwacji jest późniejsza niż końca rezerwacji.	29
6.7	Nie wybranie wiersza do rezerwacji.	30
6.8	Próba rezerwacji bez zalogowania.	31
6.9	Próba sprawdzenia historii rezerwacji bez zalogowania.	32
7.1	Oświadczenie o samodzielnym wykonaniu projektu.	35

Załącznik nr 2 do Zarządzenia nr 228/2021 Rektora Uniwersytetu Rzeszowskiego z dnia 1 grudnia 2021 roku w sprawie ustalenia procedury antyplagiatowej w Uniwersytecie Rzeszowskim

OŚWIADCZENIE STUDENTA O SAMODZIELNOŚCI PRACY

..... Kamil Kopczyk

Imię (imiiona) i nazwisko studenta

Wydział Nauk Ścisłych i Technicznych

..... Informatyka

Nazwa kierunku

..... 134927

Numer albumu

1. Oświadczam, że moja praca projektowa pt.: Projekt i implementacja systemu do rezerwacji usług łowiska wędkarskiego z wykorzystaniem bazy danych MySQL i języka JAVA

- 1) została przygotowana przeze mnie samodzielnie*,
- 2) nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2021 r., poz. 1062) oraz dóbr osobistych chronionych prawem cywilnym,
- 3) nie zawiera danych i informacji, które uzyskałem/am w sposób niedozwolony,
- 4) nie była podstawą otrzymania oceny z innego przedmiotu na uczelni wyższej ani mnie, ani innej osobie.

2. Jednocześnie wyrażam zgodę/~~nie wyrażam zgody~~** na udostępnienie mojej pracy projektowej do celów naukowo-badawczych z poszanowaniem przepisów ustawy o prawie autorskim i prawach pokrewnych.

15.06.2026

(miejscowość, data)

Kamil Kopczyk

(czytelny podpis studenta)

* Uwzględniając merytoryczny wkład prowadzącego przedmiot

** – niepotrzebne skreślić

Rys. 7.1. Oświadczenie o samodzielny wykonaniu projektu.