### **Opis wersji Aplikacji**

### **1. Wstęp**

## Opis wersji aplikacji Librus – wersja 1.00.

### **2. Zakres dokumentu**

## Dokument zawiera informacje opisujące wersję aplikacji Librus – wersja 1.00.

### **3. Informacje o wersji/edycji aplikacji**

## Jest to wersja 1.00 aplikacji Librus, której front-end został zbudowany przy użyciu Java, a backend z wykorzystaniem Javy w środowisku IntelliJ IDEA. Aplikacja korzysta z bazy danych MySQL.

### **4. Charakterystyka komponentów/pakietów**

## Komponenty zostały zrealizowane zgodnie z najnowszymi wskazaniami i standardami, z naciskiem na efektywność oraz łatwość utrzymania.

### **5. Lista pakietów aplikacji**

## Lista użytych pakietów znajduje się w plikach pom.xml oraz w folderze zawierającym zewnętrzne zależności używane przez projekt.

### **6. Diagram instalacji UML aplikacji**

## Diagram instalacji znajduje się w pliku DiagramInstalacji.drawio.

### **7. Zmiany od poprzedniej wersji**

## Ponieważ jest to pierwsza wersja aplikacji, brak jest zmian w stosunku do poprzednich wersji.

### **8. Konfiguracja środowiska przedinstalacją**

## Wymagane jest środowisko programistyczne IntelliJ IDEA oraz dostęp do bazy danych MySQL.

### **9. Instrukcja instalacji**

## Aby zainstalować aplikację Librus, należy:

## Sklonować repozytorium z GitHub.

## W IntelliJ IDEA, otworzyć projekt.

## Skonfigurować połączenie z bazą danych MySQL.

## W terminalu IntelliJ IDEA wykonać następujące komendy: bash Skopiuj kod cd server

## mvn spring-boot:run

## cd ../client

## npm install

## npm run dev

## Uruchomić aplikację.

### **10. Problemy i nieusunięte błędy**

## Brak znanych problemów i nieusuniętych błędów.

### **11. Moduły**

#### **11.1.1 Moduł JCalendar w wersji 1.4**

#### Moduł JCalendar w Javie to biblioteka graficzna, która dostarcza komponenty do obsługi kalendarzy i dat w aplikacjach Java. Umożliwia użytkownikowi łatwe wybieranie daty z graficznego interfejsu, co jest przydatne np. przy tworzeniu formularzy, gdzie wymagane jest podanie daty.

#### **11.1.2 Moduł MySQL Connector**

#### Moduł MySQL Connector to sterownik JDBC, który pozwala aplikacjom Java łączyć się z bazą danych MySQL. Dzięki niemu można wykonywać operacje na bazach danych MySQL, takie jak pobieranie, dodawanie, usuwanie czy aktualizowanie danych.

#### **11.1.3 Moduł JUnit**

Moduł JUnit to popularny framework do testowania jednostkowego w Javie. Umożliwia pisanie i wykonywanie testów dla kodu, co pozwala na sprawdzenie, czy poszczególne metody lub klasy działają zgodnie z oczekiwaniami.

#### **11.1.4 Moduł JGoodies-Common**

Moduł JGoodies Common to część zestawu narzędzi JGoodies, który oferuje zestaw bibliotek wspierających tworzenie aplikacji Java z użyciem bibliotek graficznych takich jak Swing. Biblioteka JGoodies Common zawiera podstawowe klasy i funkcje pomocnicze, które ułatwiają zarządzanie różnymi aspektami aplikacji.

#### **11.1.5 Moduł JGoodies-Looks**

Moduł JGoodies Looks to biblioteka dla Javy, która dostarcza zestaw skórek (tzw. "Look and Feel") dla aplikacji napisanych z wykorzystaniem Swinga. Umożliwia on tworzenie atrakcyjnych, nowoczesnych interfejsów użytkownika, które mogą wyglądać lepiej niż domyślne skórki Swinga.

**11.2 Lista plików**

* Lista wszystkich plików wchodzących w skład pakietu:
* JCalendar-1.4.jar
* mysql-connector-java-8.4.0.jar
* junit-4.6.jar
* jgoodies-looks-2-4-1.jar
* jgoodies-common-1.2.0.jar
* Pliki źródłowe Java dla aplikacji (np. Main.java, DatabaseConnection.java, CalculatorTest.java).

### **11.3 Instrukcja kompilacji**

### Szczegółowe kroki dotyczące kompilacji pakietu w środowisku IntelliJ IDEA:

### Utwórz nowy projekt:

### Otwórz IntelliJ IDEA.

### Wybierz File > New > Project.

### Wybierz Java, ustaw wersję SDK i zatwierdź.

### Dodaj moduły (JARs):

### Kliknij prawym przyciskiem na projekt i wybierz Open Module Settings.

### W sekcji Libraries, dodaj pliki JAR: JCalendar-1.4.jar, mysql-connector-java-8.x.x.jar, junit-5.x.x.jar.

### Skonfiguruj klasę główną (Main Class):

### Upewnij się, że Main.java znajduje się w odpowiednim katalogu (np. src).

### Zdefiniuj klasę główną, która będzie uruchamiana (np. Main).

### Kompilacja projektu:

### Wybierz Build > Build Project.

### IntelliJ IDEA skompiluje wszystkie pliki źródłowe wraz z zaimportowanymi bibliotekami.

### Uruchomienie testów JUnit:

### Kliknij prawym przyciskiem na klasie testowej (np. CalculatorTest.java) i wybierz Run.

### IntelliJ uruchomi testy JUnit i wyświetli wyniki w konsoli.

### Uruchomienie aplikacji:

### Kliknij prawym przyciskiem na pliku Main.java i wybierz Run.

### Po wykonaniu tych kroków aplikacja zostanie skompilowana i uruchomiona, a testy zostaną przeprowadzone.

### **12. Specyfikacja unit testów**

#### **12.1 Lista wykonanych test case’ów**

## Przeprowadzono następujące test case’y:

## Weryfikacja czy funkcjonalność rejestrowania i następnego logowania przez użytkownika jest prosta, jasna i klarowna.

## Weryfikacja, czy użytkownik po zalogowaniu może poruszać się po panelu użytkownika bez negatywnych konsekwencji.

#### **12.2 Raport o błędach**

## Brak zarejestrowanych błędów w trakcie ciągłego testowania aplikacji.

## 