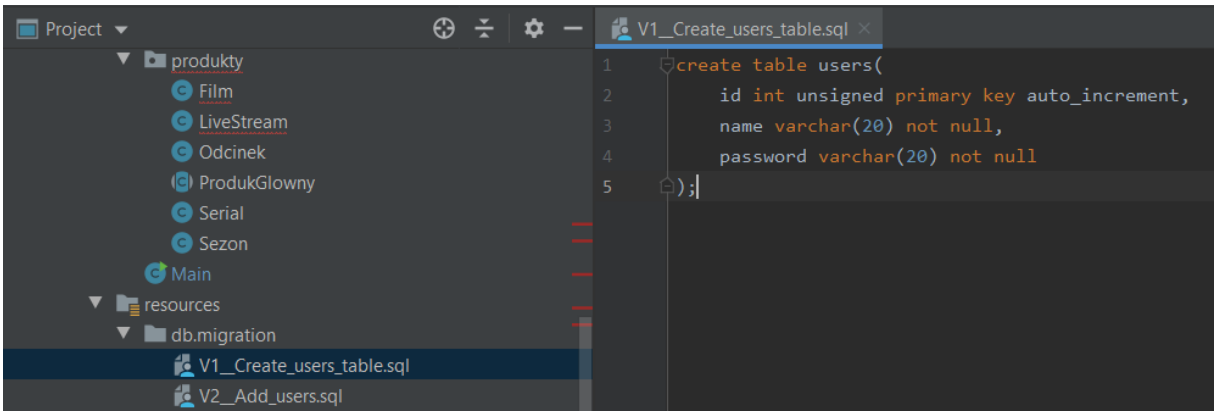
Oprócz podstawowej wersji jaką była wysłana na początku:

- ***Stos technologiczny:***
  - Java 13.0.1.9
  - Maven
  - GIT
  - Junit
  - Swing

Dodałem:
-Hibernate(5.4.7.Final)
-jetty (9.4.21.v20190926)
-flywayDB (5.2.4)
-h2 (1.4.197)

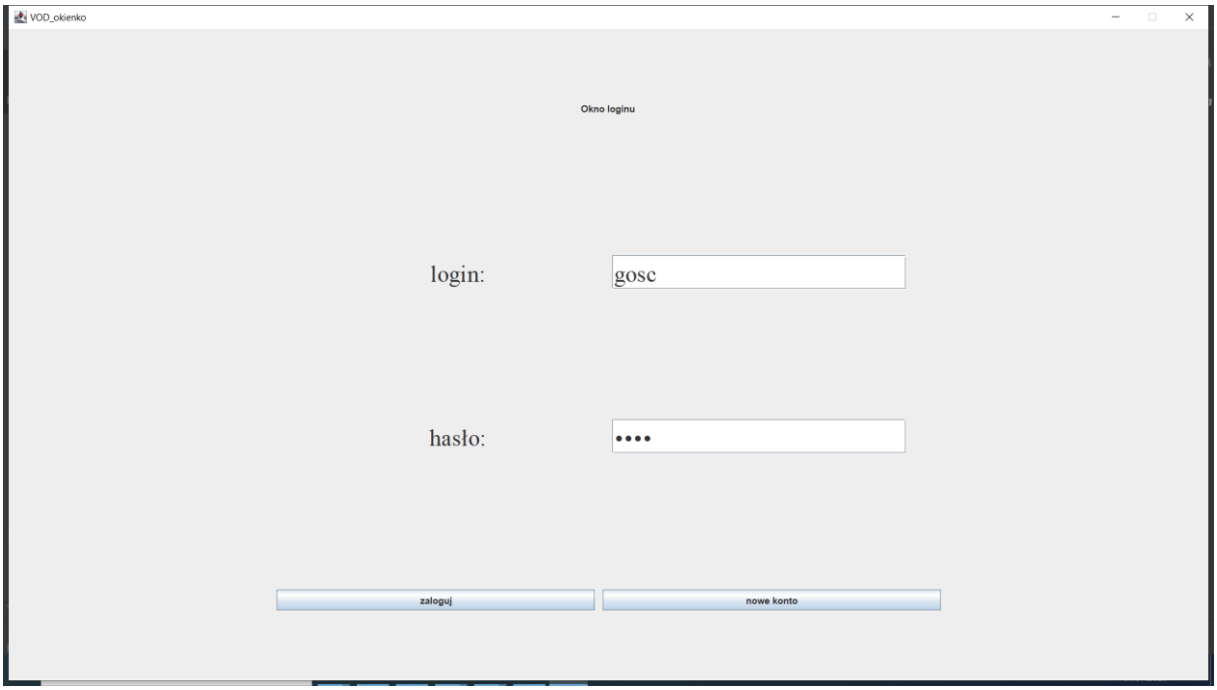
Są to rzeczy potrzebne aby mogła zachodzić wymiana informacji pomiędzy użytkownikiem a bazą danych. Silnikiem bazy danych jest H2. Wykorzystałem również lekki skryptowy framework FlyWay dzięki czemu uruchamiając aplikacje na początku stworzy się nasza baza danych i ewentualnie dodatkowe wartości.



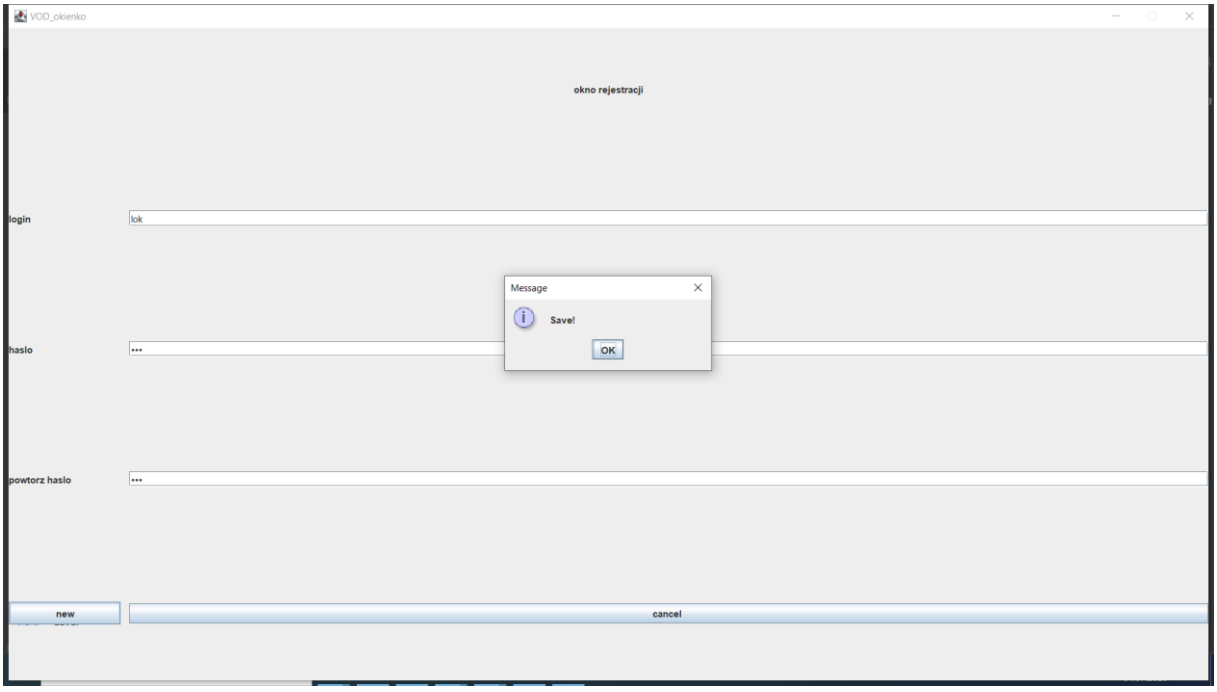
Jetty jest serverem na którym zostaje dokonane łączenie pomiędzy hibernatem a bazą danych.

Hibernate standard mapowania obiektowo-relacyjnego pozwala na wykonywanie zapytań do bazy danych z poziomu aplikacji.

Poprzednia wersja nie mogła zapisywać nowych użytkowników (wpisane było na sztywno „gosc”) teraz już okno rejestracji jest odblokowane.



Poprawne zapisanie ukazuje sie za pomocą okienka informacyjnego. Równiez przy zalogowaniu taką forme otrzymamy



Hibernate połączenie z bazą

```java
public class UserRepository {
    public boolean isUser(String name, String password) {
        boolean flagUser = false;
        var session = HibernateUtil.getSessionFactory().openSession();
        var transaction = session.beginTransaction();
        String hql = "FROM users";
        List<User> result = session.createQuery( s: "from User", User.class).getResultList();
        try {
            transaction.commit();
        } catch (Exception e) {
            transaction.rollback();
        }
        session.close();
        for (User temp : result) {
            if (temp.getName().equals(name) && temp.getPassword().equals(password)) {
                flagUser = true;
            }
        }
        return flagUser;
    }
    public boolean setUserIntoData(String name, String password) {
        boolean flagUser = false;
        var session = HibernateUtil.getSessionFactory().openSession();
        var transaction = session.beginTransaction();
        User user = new User();
        user.setUser(name, password);
        session.save(user);
        try {
            transaction.commit();
            flagUser = true;
        } catch (Exception e) {
            transaction.rollback();
        }
```

budowanie sesji

```java
public class HibernateUtil {
    private static final SessionFactory sessionFactory = buildSessionFactory();
    private HibernateUtil() {
    }
    public static void close()  {
        if ( sessionFactory != null ) {
            sessionFactory.close();
        }
    }

    public static SessionFactory getSessionFactory() { return sessionFactory; }
        private static SessionFactory buildSessionFactory() {
        // A SessionFactory is set up once for an application!
        final StandardServiceRegistry registry = new StandardServiceRegistryBuilder()
                .configure() // configures settings from hibernate.cfg.xml
                .build();
        try {
            return new MetadataSources( registry ).buildMetadata().buildSessionFactory();
        }
        catch (Exception e) {
            StandardServiceRegistryBuilder.destroy( registry );
            throw e;
        }
    }


}
```