Case Study: Citizen and Passport Management System

Business Context:

A national government agency maintains records of citizens and the passports issued to them. The rule of the system is:

Each citizen can hold exactly one passport

Each passport must be assigned to only one citizen

This kind of relationship is a textbook example of a One-to-One association, where one record in the Citizen table corresponds to one record in the Passport table, and vice versa

pom.xml:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>

  <artifactId>CitizenPassportHibernate</artifactId>

  <version>0.0.1-SNAPSHOT</version>

    <properties>

    <maven.compiler.source>17</maven.compiler.source>

    <maven.compiler.target>17</maven.compiler.target>

  </properties>

  <dependencies>

    <!-- Hibernate Core -->

<dependency>

        <groupId>org.hibernate.orm</groupId>

        <artifactId>hibernate-core</artifactId>

        <version>6.2.7.Final</version>

    </dependency>

        <!-- Jakarta Persistence API -->
    <dependency>
        <groupId>jakarta.persistence</groupId>
        <artifactId>jakarta.persistence-api</artifactId>
        <version>3.1.0</version>
    </dependency>

    <!-- MySQL Connector -->
    <dependency>
        <groupId>com.mysql</groupId>
        <artifactId>mysql-connector-j</artifactId>
        <version>8.0.33</version>
    </dependency>
</dependencies>
</project>
```

Citizen class:

```
package com.example.entity;
import jakarta.persistence.*;
@Entity
public class Citizen {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    private String name;
    @OneToOne(cascade = CascadeType.ALL)
    @JoinColumn(name = "passport_id")
    private Passport passport;
    public int getId() { return id; }

    public void setId(int id) { this.id = id; }

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }

    public Passport getPassport() { return passport; }

    public void setPassport(Passport passport) { this.passport = passport; }

}
```

Passport class:

```
package com.example.entity;

import jakarta.persistence.*;

@Entity

public class Passport {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int id;

    private String passportNumber;

    @OneToOne(mappedBy = "passport")

    private Citizen citizen;

    public int getId() { return id; }

    public void setId(int id) { this.id = id; }

    public String getPassportNumber() { return passportNumber; }

    public void setPassportNumber(String passportNumber) { this.passportNumber = passportNumber; }

    public Citizen getCitizen() { return citizen; }

    public void setCitizen(Citizen citizen) { this.citizen = citizen; }

}
```

Hibernate.cfg.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<hibernate-configuration>
    <session-factory>
        <property
name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/passport_db</property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.connection.password">Sahithi@19</property>
        <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
        <property name="hibernate.hbm2ddl.auto">update</property>
        <property name="hibernate.show_sql">true</property>
        <mapping class="com.example.entity.Citizen"/>
        <mapping class="com.example.entity.Passport"/>
    </session-factory>
</hibernate-configuration>
```

HibernateUtil.java

```
package com.example.util;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
public class HibernateUtil {
    private static final SessionFactory sessionFactory;
    static {
        try {
            sessionFactory = new Configuration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            throw new ExceptionInInitializerError(ex);
        }
    }
    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}
```

App class:

```
package com.example.app;
import org.hibernate.Session;
import org.hibernate.Transaction;
import com.example.entity.Citizen;
import com.example.entity.Passport;
import com.example.util.HibernateUtil;
public class App {
    public static void main(String[] args) {
        Citizen citizen = new Citizen();
        citizen.setName("Adam");
        Passport passport = new Passport();
        passport.setPassportNumber("SA98709");
        citizen.setPassport(passport);
        passport.setCitizen(citizen);
        Session session = HibernateUtil.getSessionFactory().openSession();
        Transaction tx = session.beginTransaction();
        session.persist(citizen);
        tx.commit();

        session.close();

        System.out.println("Citizen and Passport saved successfully.");

    }

}
```