Source code:

User:

```
package com.ecommerce.entity;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;

@Entity
@Table(name = "users")
public class User {

@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private Long id;

@Column(nullable = false, unique = true)
private String username;

@Column(nullable = false, unique = true)
private String email;

@Column(nullable = false)
private String password;

public User() {
}

public User(String username, String email, String password) {
this.username = username;
this.password = password;
this.email = email;
}

public Long getId() {
return id;
}

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

public String getUsername() {
return username;
}

public void setUsername(String username) {
this.username = username;
}

public String getEmail() {
return email;
}
```

```
public void setEmail(String email) {
this.email = email;
}

public String getPassword() {
return password;
}

public void setPassword(String password) {
this.password = password;
}

}
```

Citizen:

```
package com.ecommerce.entity;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.FetchType;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.Table;

@Entity
@Table(name = "citizens")
public class Citizen {
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private Long id;

@Column(name = "name")
private String name;

@Column(name = "city")
private String city;

@Column(name = "no_of_doses")
private int noOfDoses;

@Column(name = "vaccination_status")
private String vaccinationStatus;

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "vaccination_center")
private VaccineCenter vaccinationCenter;

// Constructors, getters, setters, and other properties

public Long getId() {
return id;
}

public void setId(Long id) {
```

```
this.id = id;
}

public String getName() {
return name;
}

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

public String getCity() {
return city;
}

public void setCity(String city) {
this.city = city;
}

public int getNoOfDoses() {
return noOfDoses;
}

public void setNoOfDoses(int noOfDoses) {
this.noOfDoses = noOfDoses;
}

public String getVaccinationStatus() {
return vaccinationStatus;
}

public void setVaccinationStatus(String vaccinationStatus) {
this.vaccinationStatus = vaccinationStatus;
}

public VaccineCenter getVaccinationCenter() {
return vaccinationCenter;
}

public void setVaccinationCenter(VaccineCenter vaccinationCenter) {
this.vaccinationCenter = vaccinationCenter;
}

}
```

Vaccine center:

```
package com.ecommerce.entity;



import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;
```

```
import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.Table;


@Entity

@Table(name = "vaccine_centers")

public class VaccineCenter {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;


@Column(name = "center")

private String center;


@Column(name = "city")

private String city;


public VaccineCenter() {

// Default constructor required by JPA

}


public VaccineCenter(String center, String city) {

this.center = center;

this.city = city;

}


public Long getId() {
```

```
return id;

}

public void setId(Long id) {

this.id = id;

}

public String getCenter() {

return center;

}

public void setCenter(String center) {

this.center = center;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

}
```

Vaccination application:

```
package com.ecommerce.entity;

import org.springframework.boot.SpringApplication;
```

```
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class VaccinationApplication {

	public static void main(String[] args) {
		SpringApplication.run(VaccinationApplication.class, args);
	}

}
```

Citizen repo:

```
package com.ecommerce.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import com.ecommerce.entity.Citizen;
import com.ecommerce.entity.VaccineCenter;

public interface CitizenRepository extends JpaRepository<Citizen, Long> {

List<Citizen> findByVaccinationCenter(VaccineCenter vaccinationCenter);
}
```

User repo:

```
package com.ecommerce.repository;

import org.springframework.data.jpa.repository.JpaRepository;


import com.ecommerce.entity.User;


public interface UserRepository extends JpaRepository<User, Long> {


User findByUsername(String username);


}
```

Vaccine repo:

```
package com.ecommerce.repository;


import org.springframework.data.jpa.repository.JpaRepository;


import com.ecommerce.entity.VaccineCenter;


public interface VaccineCenterRepository extends JpaRepository<VaccineCenter, Long> {

// Add custom query methods if needed

}
```

Citizen service:

```
package com.ecommerce.service;


import java.util.List;


import com.ecommerce.entity.Citizen;
```

```
import com.ecommerce.entity.VaccineCenter;

public interface CitizenService {
void addCitizen(Citizen citizen);

void updateCitizen(Citizen citizen);

void deleteCitizen(Long id);

Citizen getCitizenById(Long id);

List<Citizen> getAllCitizens();

List<Citizen> getCitizensByVaccinationCenter(VaccineCenter vaccineCenter);
}
```

Citizen service impl:

```
package com.ecommerce.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.ecommerce.entity.Citizen;
import com.ecommerce.entity.VaccineCenter;
import com.ecommerce.repository.CitizenRepository;
```

```
@Service
public class CitizenServiceImpl implements CitizenService {

	@Autowired
	private CitizenRepository citizenRepository;

	@Override
	public void addCitizen(Citizen citizen) {
		citizenRepository.save(citizen);
	}

	@Override
	public void updateCitizen(Citizen citizen) {
		citizenRepository.save(citizen);
	}

	@Override
	public void deleteCitizen(Long id) {
		citizenRepository.deleteById(id);
	}

	@Override
	public Citizen getCitizenById(Long id) {
		return citizenRepository.findById(id).orElseThrow(() -> new
NotFoundException("Citizen not found"));
	}

	@Override
```

```
	public List<Citizen> getAllCitizens() {
		return citizenRepository.findAll();
	}

	@Override
	public List<Citizen> getCitizensByVaccinationCenter(VaccineCenter vaccineCenter) {
		return citizenRepository.findByVaccinationCenter(vaccineCenter);
	}

	public class NotFoundException extends RuntimeException {

		public NotFoundException(String message) {
			super(message);
		}
	}
}
```

User service:

```
package com.ecommerce.service;

import org.springframework.stereotype.Service;

import com.ecommerce.entity.User;
import com.ecommerce.repository.UserRepository;

@Service
public class UserService {
private final UserRepository userRepository;
```

```
public UserService(UserRepository userRepository) {
this.userRepository = userRepository;
}

public void registerUser(User user) {
userRepository.save(user);
}
}
```

Vaccine service:

```
package com.ecommerce.service;

import java.util.List;

import com.ecommerce.entity.VaccineCenter;

public interface VaccineCenterService {

VaccineCenter addVaccineCenter(VaccineCenter vaccineCenter);

void updateVaccineCenter(VaccineCenter updatedVaccineCenter);

void deleteVaccineCenter(Long id);

VaccineCenter getVaccineCenterById(Long id);

List<VaccineCenter> getAllVaccineCenters();
```

}

Vaccine service impl:

```
package com.ecommerce.service;


import java.util.List;


import org.springframework.stereotype.Service;


import com.ecommerce.entity.VaccineCenter;
import com.ecommerce.repository.VaccineCenterRepository;


@Service
public class VaccineCenterServiceImpl implements VaccineCenterService {
private final VaccineCenterRepository vaccineCenterRepository;


public VaccineCenterServiceImpl(VaccineCenterRepository vaccineCenterRepository) {
this.vaccineCenterRepository = vaccineCenterRepository;
}


@Override
public VaccineCenter addVaccineCenter(VaccineCenter vaccineCenter) {
return vaccineCenterRepository.save(vaccineCenter);
}


@Override
public void updateVaccineCenter(VaccineCenter updatedVaccineCenter) {
VaccineCenter existingCenter =
```

```
vaccineCenterRepository.findById(updatedVaccineCenter.getId())
.orElseThrow(() -> new NotFoundException("Vaccine center not found"));

existingCenter.setCenter(updatedVaccineCenter.getCenter());
existingCenter.setCity(updatedVaccineCenter.getCity());

vaccineCenterRepository.save(existingCenter);
}

@Override
public void deleteVaccineCenter(Long id) {
VaccineCenter existingCenter = vaccineCenterRepository.findById(id)
.orElseThrow(() -> new NotFoundException("Vaccine center not found"));

vaccineCenterRepository.delete(existingCenter);
}

@Override
public VaccineCenter getVaccineCenterById(Long id) {
return vaccineCenterRepository.findById(id)
.orElseThrow(() -> new NotFoundException("Vaccine center not found"));
}

@Override
public List<VaccineCenter> getAllVaccineCenters() {
return vaccineCenterRepository.findAll();
}
```

```
public class NotFoundException extends RuntimeException {

public NotFoundException(String message) {

super(message);

}

}

}
```

Application properties:

```
spring.datasource.url=jdbc:mysql://localhost:3306/vaccination?useSSL=false&serverT
imezone=UTC
spring.datasource.username=root
spring.datasource.password=root
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=update
server.port=3008
```