

```

package com.service;

import java.util.List;
import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
//import java.time.LocalDateTime;

import com.entity.ApproveCitizens;
import com.entity.Citizens;
import com.entity.Login;
import com.repository.CitizensRepository;
import com.repository.LoginRepository;

@Service
public class CitizensService {

    @Autowired
    CitizensRepository citizenRepository;

    @Autowired
    LoginRepository loginRepository;

    public String storeCitizensDetails(Citizens citizens) {

        Optional<Login> op = loginRepository.findById(citizens.getEmailid());

        long count = citizenRepository.count();
        citizens.setAadharnumber(citizens.getAadharnumber()+count);
        citizens.setStatus("Alive");
        citizens.setIssue_date(java.time.LocalDateTime.now());
        citizenRepository.save(citizens);

        return "Account Created";

    }

    public Citizens findCitizen(long aadharNumber) {
        Optional<Citizens> op = citizenRepository.findById(aadharNumber);
        if(op.isPresent()) {
            return op.get();
        }else {
            return null;
        }
    }
}

```

```

    }

    public List<Citizens> findAllCitizens() {
        return citizenRepository.findAll();
    }

    public Citizens findCitizenByEmailId(String emailid) {
        return citizenRepository.findCitizenByEmailId(emailid);
    }

    public List<Citizens> findCitizenByStatus() {
        return citizenRepository.findCitizenByStatus("death");
    }

    public String deleteCitizens(long aadharid) {
        Optional<Citizens> result = citizenRepository.findById(aadharid);
        if(result.isPresent()) {
            citizenRepository.deleteById(aadharid);
            return "Citizen details deleted successfully";
        }else {

            return "Citizen details not deleted";
        }
    }

    public String updateCitizenDetails(Citizens citizen) {
        // Validate input if necessary

        Optional<Citizens> existingCitizen =
citizenRepository.findById(citizen.getAadharnumber());

        if (existingCitizen.isPresent()) {
            Citizens existing = existingCitizen.get();

            existing.setAddress(citizen.getAddress());
            existing.setDob(citizen.getDob());
            existing.setPhonenumber(citizen.getPhonenumber());

            citizenRepository.save(existing);
            return "Account Updated";
        } else {
            return "Account Not Updated";
        }
    }

}

```