МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС

«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»

НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Лабораторна робота №2

з дисциплiни «Проектування інформаційних систем»

Виконав:

студент 4 курсу

групи КА-66

Дядюра О. Ю.

Прийняв:

Коновалюк М.М.

**Київ – 2020**

23. Система Больница. Врач определяет диагноз, делает назначение

Пациенту (процедуры, лекарства, операции). Назначение может выполнить

Медсестра (процедуры, лекарства) или Врач (любое назначение). Пациент

может быть выписан из Больницы, при этом фиксируется окончательный

диагноз.

Створення бізнес логіки.

***Лістинг програми:***

**Services**

**package** ua.kpi.myhospital.Service;  
  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.stereotype.Service;  
  
**import** ua.kpi.myhospital.DAO.DiagnosDAO;  
**import** ua.kpi.myhospital.Entities.Diagnos;  
**import** ua.kpi.myhospital.Entities.User;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
@Service  
**public class** DiagnosService {  
  
 @Autowired  
 **private** DiagnosDAO **diagnosDAO**;  
  
 **public** List<Diagnos> getAllDiagnoses(){  
 List<Diagnos> diagnoses = **new** ArrayList<>();  
 **diagnosDAO**.findAll().forEach(diagnoses::add);  
 **return** diagnoses;  
 }  
  
 **public** Diagnos getDiagnos(Integer idDiangos){  
 **return diagnosDAO**.findById(idDiangos);  
 }  
  
 **public void** saveDiagnos(Diagnos diagnos){  
 **diagnosDAO**.save(diagnos);  
 }  
  
 **public void** deleteDiagnos(Integer idDiagnos){  
 **diagnosDAO**.deleteById(idDiagnos);  
 }  
  
  
}

**package** ua.kpi.myhospital.Service;  
  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.stereotype.Service;  
**import** ua.kpi.myhospital.DAO.PrescriptionDAO;  
**import** ua.kpi.myhospital.Entities.Prescription;  
**import** ua.kpi.myhospital.Repo.PrescriptionRepository;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
  
@Service  
**public class** PrescriptionService {  
 @Autowired  
 **private** PrescriptionDAO **prescriptionDAO**;  
  
  
 **public** List<Prescription> getAllPrescriptions() {  
 List<Prescription> prescriptions = **new** ArrayList<>();  
 **prescriptionDAO**.findAll().forEach(prescriptions::add);  
 **return** prescriptions;  
 }  
  
 **public** Prescription getPrescription(Integer idPrescription){  
 **return prescriptionDAO**.findById(idPrescription);  
 }  
  
 **public void** savePrescription(Prescription prescription) {  
 **prescriptionDAO**.save(prescription);  
 }  
  
 **public void** deletePrescription(Integer idPrescription){  
 **prescriptionDAO**.deleteById(idPrescription);  
 }  
  
 **public void** updatePrescription(Prescription prescription, Integer idPrescription){  
 **prescriptionDAO**.save(prescription);  
 }  
}

**package** ua.kpi.myhospital.Service;  
  
  
  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.stereotype.Service;  
**import** ua.kpi.myhospital.DAO.UserDAO;  
**import** ua.kpi.myhospital.Data.UserData;  
**import** ua.kpi.myhospital.Entities.User;  
**import** ua.kpi.myhospital.Repo.UserRepository;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
@Service  
**public class** UserService {  
 @Autowired  
 **private** UserDAO **userDAO**;  
  
 **public** List<User> getAllUsers(){  
 List<User> users = **new** ArrayList<>();  
 **userDAO**.findAll().forEach(users::add);  
 **return** users;  
 }  
  
 **public** User getUser(Integer idUser){  
 **return userDAO**.findById(idUser);  
 }  
  
 **public void** saveUser(User user){  
 **userDAO**.save(user);  
 }  
  
 **public void** updateUser(User user, Integer id){  
 **userDAO**.save(user);  
 }  
  
 **public void** deleteUser(Integer userid){  
 **userDAO**.deleteById(userid);  
 }  
  
  
 **public** List<User> getByName(String name){  
 List<User> users = **new** ArrayList<>();  
 **userDAO**.findByName(name).forEach(users::add);  
 **return** users;  
 }  
  
  
  
}

**Controllers**

**package** ua.kpi.myhospital.Controllers;  
  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.web.bind.annotation.\*;  
**import** ua.kpi.myhospital.Entities.Diagnos;  
**import** ua.kpi.myhospital.Entities.User;  
**import** ua.kpi.myhospital.Service.DiagnosService;  
  
**import** java.util.List;  
  
@RestController  
**public class** DiagnosController {  
 @Autowired  
 **private** DiagnosService **diagnosService**;  
  
 @RequestMapping(**"/diagnoses"**)  
 **public** List<Diagnos> getAllDiagnoses(){  
 **return diagnosService**.getAllDiagnoses();  
 }  
  
 @RequestMapping(**"/diagnoses/{idDiagnos}"**)  
 **public** Diagnos getDiagnos(@PathVariable Integer idDiagnos){  
 **return diagnosService**.getDiagnos(idDiagnos);  
 }  
  
 @RequestMapping(method = RequestMethod.***POST***, value = **"/diagnoses"** )  
 **public void** addDiagnos(@RequestBody Diagnos diagnos){  
 **diagnosService**.saveDiagnos(diagnos);  
 }  
  
 @RequestMapping(method = RequestMethod.***PUT***, value = **"/diagnoses/{idDiagnos}"**)  
 **public void** updateDiagnos(@RequestBody Diagnos diagnos, @PathVariable Integer idDiagnos){  
 **diagnosService**.saveDiagnos(diagnos);  
 }  
  
 @RequestMapping(method = RequestMethod.***DELETE*** ,value = **"/diagnoses/{idDiagnos}"**)  
 **public void** deleteDiagnos(@PathVariable Integer idDiagnos){  
 **diagnosService**.deleteDiagnos(idDiagnos);  
 }  
}

**package** ua.kpi.myhospital.Controllers;  
  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.web.bind.annotation.RequestMapping;  
**import** org.springframework.web.bind.annotation.RestController;  
**import** ua.kpi.myhospital.Entities.Diagnos;  
**import** ua.kpi.myhospital.Entities.Prescription;  
**import** ua.kpi.myhospital.Entities.User;  
**import** ua.kpi.myhospital.Service.DiagnosService;  
**import** ua.kpi.myhospital.Service.PrescriptionService;  
**import** ua.kpi.myhospital.Service.UserService;  
  
**import** java.util.List;  
  
@RestController  
**public class** PopulateController {  
  
 @Autowired  
 **private** DiagnosService **diagnosService**;  
 @Autowired  
 **private** PrescriptionService **prescriptionService**;  
 @Autowired  
 **private** UserService **userService**;  
  
 @RequestMapping(**"/populate"**)  
 **public void** populate() {  
 List<User> users = List.*of*(**new** User(1, **"Bogdan"**, **"Surname"**, **"doc"**),  
 **new** User(2, **"Sasha"**, **"Surname"**, **"patient"**),  
 **new** User(3, **"Vika"**, **"Surname"**, **"nurse"**));  
 users.forEach(user -> **userService**.saveUser(user));  
  
  
 List<Prescription> prescriptions = List.*of*(**new** Prescription(1, **"operation "**, 1, 2, 1),  
 **new** Prescription(2, **"medcine"**, 1, 2, 1));  
 prescriptions.forEach((prescription -> **prescriptionService**.savePrescription(prescription)));  
  
 List<Diagnos> diagnosList = List.*of*(**new** Diagnos(1, 1,**"very sick"**, **"still alive"**),  
 **new** Diagnos(2, 2, **"quite healthy"**, **"alive"**));  
 diagnosList.forEach(diagnos -> **diagnosService**.saveDiagnos(diagnos));  
  
 }  
}

**package** ua.kpi.myhospital.Controllers;  
  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.web.bind.annotation.\*;  
**import** ua.kpi.myhospital.Entities.Prescription;  
**import** ua.kpi.myhospital.Service.PrescriptionService;  
  
**import** java.util.List;  
  
@RestController  
**public class** PrescriptionController {  
  
 @Autowired  
 **private** PrescriptionService **prescriptionService**;  
  
 @RequestMapping(**"/prescription"**)  
 **public** List<Prescription> getAllPrescriptions() {  
 **return prescriptionService**.getAllPrescriptions();  
 }  
  
 @RequestMapping(**"/prescription/{idPrescription}"**)  
 **public** Prescription getById(@PathVariable Integer idPrescription) {  
 **return prescriptionService**.getPrescription(idPrescription);  
 }  
  
 @RequestMapping(method = RequestMethod.***POST***, value = **"/prescription"**)  
 **public void** addPrescription(@RequestBody Prescription prescription) {  
 **prescriptionService**.savePrescription(prescription);  
 }  
  
 @RequestMapping(method = RequestMethod.***DELETE***, value = **"/prescription/{idPrescription}"**)  
 **public void** deletePrescription(@PathVariable Integer idPrescription) {  
 **prescriptionService**.deletePrescription(idPrescription);  
 }  
 @RequestMapping(method = RequestMethod.***PUT***, value = **"/prescription/{idPrescription}"**)  
 **public void** uppdatePrescription(@RequestBody Prescription prescription, @PathVariable Integer idPrescription){  
 **prescriptionService**.updatePrescription(prescription, idPrescription);  
 }  
  
  
  
}

**package** ua.kpi.myhospital.Controllers;  
  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.web.bind.annotation.\*;  
**import** ua.kpi.myhospital.Entities.User;  
**import** ua.kpi.myhospital.Service.UserService;  
  
**import** java.util.List;  
@RestController  
**public class** UserController {  
 @Autowired  
 **private** UserService **userService**;  
  
 @RequestMapping(**"/users"**)  
 **public** List<User> getAllUsers(){  
 **return userService**.getAllUsers();  
 }  
  
 @RequestMapping(**"/users/{idUser}"**)  
 **public** User getUser(@PathVariable Integer idUser){  
 **return userService**.getUser(idUser);  
 }  
  
 @RequestMapping(**"/user/{name}"**)  
 **public** List<User> getByName(@PathVariable String name){  
 **return userService**.getByName(name);  
 }  
  
 @RequestMapping(method = RequestMethod.***POST***, value = **"/users"** )  
 **public void** addUser(@RequestBody User user){  
 **userService**.saveUser(user);  
 }  
  
 @RequestMapping(method = RequestMethod.***PUT***, value = **"/users/{id}"**)  
 **public void** updateUser(@RequestBody User user, @PathVariable Integer id){  
 **userService**.updateUser(user, id);  
 }  
  
 @RequestMapping(method = RequestMethod.***DELETE***,value = **"/users/{idUser}"**)  
 **public void** deleteUser(@PathVariable Integer idUser){  
 **userService**.deleteUser(idUser);  
 }  
  
  
  
  
}

PopulatinController призначений для заповнення бази мінімальними даними