SYSTEM DESIGN of:

Hospital Management

software

Developed by: Lola Sanchez, Maria Rueda, Jaime Soler and Kacper Borysewicz

TABLE OF CONTENT

MAIN MENU

CASE 1

loginPatient()

When you press “login as a patient”, first you have to introduce an email and a password, which are strings. And then the method is going to use a user manager to check the password where we are going to use the method “digest” to encrypt the password and if that user has never been created you will have a successful login otherwise you will be asked for the data again.

CASE 2

createPatient()

This is a method, that its source it is going to be typing the data of the patient. We have name and email that are strings, severe that is going to be a boolean (yes/no), a phone that is going to be an integer, dob that stands for a date of birth using the class LocalDate and finally a password that is a string.

CASE 3

loginDoctor()

When you press “login as a doctor” first you have to introduce an email and a password, which are strings. And then the method is going to use a user manager to check the password where we are going to use the method “digest” to encrypt the password and if that user has never been created you will have a successful login otherwise you will be asked for the data again.

CASE 4

createDoctor()

This is a method, that its source it is going to be typing the data of the doctor. We have name and speciality that are strings and finally we add a doctor to the doctorManager that is a manager class of the doctor.

CASE 5

loginNurse()

When you press “login as a nurse” first you have to introduce an email and a password, which are strings. And then the method is going to use a user manager to check the password where we are going to use the method “digest” to encrypt the password and if that user has never been created you will have a successful login otherwise you will be asked for the data again.

CASE 6

createNurse()

This is a method, that its source it is going to be typing the data of the nurse. We have name that is a string and we are going to add the nurse to a nurseManage.

CASE 0

exit()

When you press “0” key, system shuts down. To do that, we call “disconnect” function to jdbcManager, and e”exit” to system.

PATIENT MENU

In this menu we are going to use all the options related to patients, with sources that comes from interfaces that are implemented in the JDBC of all the entities of the database.

CASE 1

updatePatient()

When we use this option, the main goal is to change the data of the patient. To do that we use a string that is stands for a query that is going to update a patient, changing the name, email, severity, phone and dob; and for this method we use a prepared statement, because we are deleting and adding a new patient’s information.

CASE 2

listMyDoctors()

Main goal of this option is to list all doctors for each patient. To do that we use a string that is stands for a query that is going to list all doctors for each patient. We are going to show at the screen all the doctors information for each patient.

CASE 3

listMySymptoms()

Main goal of this option is to list all symptoms which individual patient has. To do that we use a string that is stands for a query that is going to list symptoms. We are going to display at the screen all the patients's information.

CASE 4

listMyMedicines() & listMyDiseases()

Main goal of this option is to list medicines and diseases for individual patient. To do that we use a string that is stands for a query that is going to list the disease with correspponed medicines.

CASE 5

Marshal()

The main goal of this option is to export the patient to an XML file. To do that we use a string that is a name of file (with the extension .xml), to which we want to export, then we call marshalling process.

CASE 6

addPatient()

The main goal of this option is to import the patient from an XML file. To do that we use a string that is a name of file (with the extension .xml), from which we want to import, then we call unmarshalling process.

CASE 7

simpleTransform()

The main goal of this option is to save as HTML file. To do that we use 3 strings. First stands for a name of XML file, second for a name of XSLT name, last for a name of HTML file, to which we want to save.Then we call "Xml2HtmlPatient.java" and simpleTransform JAVAs function.

CASE 0

exit()

When you press “0” key, system shuts down. To do that, we call “disconnect” function to jdbcManager, and e”exit” to system.

DOCTOR MENU

In this menu we are going to use all the option related to doctors, with sources that comes from interfaces that are implemented in the JDBC of all the entities of the database.

CASE 1

updateDoctor()

When we use this option, the main goal is to change the data of the doctor. To do that we use a string that is stands for a query that is going to update a doctor, changing the name, email and speciality; and for this method we use a prepared statement, because we are deleting and adding a new doctors information.

CASE 2

listMyPatient()

Main goal of this option is to list all patient for each doctor. To do that we use a string that is stands for a query that is going to list all patients for each doctor. We are going to show at the screen all the patients’s information for each doctor.

CASE 3

listAllPatients()

Main goal of this option is to list all patient that are in the database. To do that we use a string that is stands for a query that is going to list all patients. We are going to display at the screen all the patients’s information.

CASE 4

getPatientByName() & getSymptomByName() & getMedicineByName() & getDiseaseByName()

Main goal of this option is to diagnose a patient. To do that we use some strings. The first one refers to the name of the requested patient, with which we can call "getPatientByName". After that we enter one of the 3 "do while" loops where we display the symptom list using "listAllSymptoms", then we use next string which stands for name of symptom and call “getSymptomByName”. In the second loop we display the disease list using "listAllDisease", then we use next string that stands for name of disease and call “getSymptomByDisease”. In the last loop we display the disease list using "listAllMedicines", then we use last string that stands for name of medicine and call “getSymptomByMedicine”. To exit every of that loops, string must be "exit".

CASE 5

Marshal()

The main goal of this option is to export the doctor to an XML file. To do that we use a string that is a name of file (with the extension .xml), to which we want to export, then we call marshalling process.

CASE 6

addDoctor()

The main goal of this option is to import the doctor from an XML file. To do that we use a string that is a name of file (with the extension .xml), from which we want to import, then we call unmarshalling process.

CASE 7

simpleTransform()

The main goal of this option is to save as HTML file. To do that we use 3 strings. First stands for a name of XML file, second for a name of XSLT name, last for a name of HTML file, to which we want to save.Then we call “Xml2HtmlPatient.java” and simpleTransform JAVAs function.

CASE 0

exit()

When you press “0” key, system shuts down. To do that, we call “disconnect” function to jdbcManager, and e”exit” to system.

NURSE MENU

In this menu we are going to use all the option related to nurses, with sources that comes from interfaces that are implemented in the JDBC of all the entities of the database.

CASE 1

updateNurse()

When we use this option, the main goal is to change the data of the nurse. To do that we use a string that is stands for a query that is going to update a nurse, changing the name and email; and for this method we use a prepared statement, because we are deleting and adding a new information.

CASE 2

listMyPatient()

Main goal of this option is to list all patient for each nurse. To do that we use a string that is stands for a query that is going to list all patients for each nurse. We are going to show at the screen all the patients’s information for each nurse.

CASE 3

listAllPatients()

Main goal of this option is to list all patient that are in the database. To do that we use a string that is stands for a query that is going to list all patients. We are going to display at the screen all the patients’s information.

CASE 4

updatePatientStatus()

The main goal of this option is to update patients status. To do that, we use a string that is a name of patient.

CASE 5

marshal()

The main goal of this option is to export the nurse to an XML file. To do that we use a string that is a name of file (with the extension .xml), to which we want to export, then we call marshalling process.

CASE 6

addNurse()

The main goal of this option is to import the nurse from an XML file. To do that we use a string that is a name of file (with the extension .xml), from which we want to import, then we call unmarshalling process.

CASE 7

simpleTransform()

The main goal of this option is to save as HTML file. To do that we use 3 strings. First stands for a name of XML file, second for a name of XSLT name, last for a name of HTML file, to which we want to save.Then we call "Xml2HtmlNurse.java" and simpleTransform JAVAs function.

CASE 0

exit()

When you press "0" key, system shuts down. To do that, we call "disconnect" function to jdbcManager, and e"exit" to system.