**CMSC214**

**Final Project**

**Derya O. Kurin**

**Content:**

**1-Final Project Proposal**

**2- Installation Guide: Instructions to set the Database: ‘Medical\_Data’**

**3- Design: UML Diagram & Algorithm**

**4- Test cases**

**5- Users guide & Screen shots**

**1-Final Project Proposal**

**Project proposal:**

My final project will involve in creating a client-server network with a GUI accepting input data from the user to be used in SQL queries in server application to fetch data from a database for displaying medical information about a list of medications.

The user will be able to gather data about a specific medication through the client application to the server, and the server in return will make a query to the database and send the response back to client.

The database to be created in the programmer’s local machine via MySQL Database Service will have multiple tables which will store information about medications, such as medication name, dose, indications, contraindications, pregnancy categories etc.

Depending on the search, the server will make an appropriate SQL enquiry to the database using primary and foreign keys in different tables.

With this program, users will be able to get some information about certain medications; the recommended dose per day (dose for adults only), the biocompatibility with other medications, and possibly contraindicate health conditions etc.

The design and the content of the tables will be clearer once I gather some data for the medications (between 10 and 15) to be used in this project.

**Requirements:**

1. Create a client application with a GUI
2. Create a server application with a GUI
3. Create a network between client and server using Socket Programming
4. Create a database `MedicalData` with 3-4 tables using MySQL Database Service
5. For tables make use of primary and foreign keys so that the SQL queries should use them and also some conditions ideally.
6. In Server application, establish a connection to the database created in the previous step
7. Write methods that make SQL enquiries to the database when the user clicks buttons on Client GUI.

**2- Installation Guide**

**Instructions to set the Database: ‘Medical\_Data’**

Creating database for the final project:

~ $ cd /usr/local/mysql-8.0.22-macos10.15-x86\_64/

/u/l/mysql-8.0.22-macos10.15-x86\_64 $ cd bin

/u/l/m/bin $ ./mysql -u root -p

Enter password:----

mysql> create database medical\_data;

mysql> use mysql;

//Creating a user ‘dozdemir’ with password ‘final’

mysql> create user 'dozdemir'@'localhost' identified by 'final';

//Giving permissions to user dozdemir to manipulate mysql db

mysql> grant select, insert, update, delete, create, create view, drop, execute, references on \*.\* to 'dozdemir'@'localhost';

mysql> exit

/u/l/m/bin $ ./mysql -u dozdemir -p

Enter password: final

**Create tables:**

+------------------------+

| Tables\_in\_medical\_data |

+------------------------+

| Contraindications |

| Indications |

| Medical\_Data |

| PregnancyCategory |

+------------------------+

**PregnancyCategory**

create table PregnancyCategory (

id int not null,

category char(1),

primary key (id)

);

insert into PregnancyCategory values('1', 'A');

insert into PregnancyCategory values('2', 'B');

insert into PregnancyCategory values('3', 'C');

insert into PregnancyCategory values('4', 'D');

**Indications**

create table Indications (

id int not null,

health\_condition varchar(40),

primary key (id)

);

insert into Indications values('1', 'Headache');

insert into Indications values('2', 'Rheumatoid Arthritis');

insert into Indications values('3', 'Anticoagulant');

insert into Indications values('4', 'Fever');

insert into Indications values('5', 'Dental pain');

insert into Indications values('6', 'Arthritis');

insert into Indications values('7', 'Muscle pain');

insert into Indications values('8', 'Hypertension – High blood pressure');

insert into Indications values('9', 'Heart disease');

insert into Indications values('10', 'Diabetes');

insert into Indications values('11', 'Diarrhea');

insert into Indications values('12', 'Inflammatory Bowel Disease');

insert into Indications values('13', 'Kidney disease');

insert into Indications values('14', 'Bronchitis');

insert into Indications values('15', 'Cold/Flu');

**Contraindications:**

create table Contraindications (

id int not null,

health\_condition varchar(40),

primary key (id)

);

insert into Contraindications values('1', 'Allergy to Salicylate');

insert into Contraindications values('2', 'Allergy to NSAID’s');

insert into Contraindications values('3', 'Vitamin K deficiency');

insert into Contraindications values('4', 'Hemophilia');

insert into Contraindications values('5', 'Stomach problems');

insert into Contraindications values('6', 'Liver disease');

insert into Contraindications values('7', 'Asthma');

insert into Contraindications values('8', 'Kidney disease');

insert into Contraindications values('9', 'Heart disease');

insert into Contraindications values('10', 'High Blood Pressure');

insert into Contraindications values('11', 'Low Blood Pressure');

insert into Contraindications values('12', 'Angioedema');

**Medical\_Data**

create table Medical\_Data (

id int not null,

name varchar(40),

dose char(10),

categoryId int,

indicationId int,

contraindicationId int,

primary key (id),

foreign key(categoryId) references PregnancyCategory(id),

foreign key(indicationId) references Indications(id),  
 foreign key(contraindicationId) references Contraindications(id)

);

PregnancyCategory

+----+----------+

| id | category |

+----+----------+

| 1 | A |

| 2 | B |

| 3 | C |

| 4 | D |

+----+----------+

Indications:

+----+--------------------------------------+

| id | health\_condition |

+----+--------------------------------------+

| 1 | Headache |

| 2 | Rheumatoid Arthritis |

| 3 | Anticoagulant |

| 4 | Fever |

| 5 | Dental pain |

| 6 | Arthritis |

| 7 | Muscle pain |

| 8 | Hypertension – High blood pressure |

| 9 | Heart disease |

| 10 | Diabetes |

| 11 | Diarrhea |

| 12 | Inflammatory Bowel Disease |

| 13 | Kidney disease |

| 14 | Bronchitis |

| 15 | Cold/Flu |

+----+--------------------------------------+

Contraindications:

+----+-----------------------+

| id | health\_condition |

+----+-----------------------+

| 1 | Allergy to Salicylate |

| 2 | Allergy to NSAID’s |

| 3 | Vitamin K deficiency |

| 4 | Hemophilia |

| 5 | Stomach problems |

| 6 | Liver disease |

| 7 | Asthma |

| 8 | Kidney disease |

| 9 | Heart disease |

| 10 | High Blood Pressure |

| 11 | Low Blood Pressure |

| 12 | Angioedema |

+----+-----------------------+

**Medical\_Data data insertion:**

INSERT INTO Medical\_Data (id, name, dose, categoryId, indicationId, contraindicationId) VALUES ('1','Aspirin','300', '4', '1', '10');

INSERT INTO Medical\_Data (id, name, dose, categoryId, indicationId, contraindicationId) VALUES ('2','Acetaminophen','2000', '1', '1', '6');

INSERT INTO Medical\_Data (id, name, dose, categoryId, indicationId, contraindicationId) VALUES ('3','Ibuprofen','3200', '2', '4', '7');

INSERT INTO Medical\_Data (id, name, dose, categoryId, indicationId, contraindicationId) VALUES ('4','Valsartan','80', '4', '8', '11');

INSERT INTO Medical\_Data (id, name, dose, categoryId, indicationId, contraindicationId) VALUES ('5','Insulin glargine','160', '3', '10', '6') ;

INSERT INTO Medical\_Data (id, name, dose, categoryId, indicationId, contraindicationId) VALUES ('6','Rifaximin','1100', '3', '11', '6');

INSERT INTO Medical\_Data (id, name, dose, categoryId, indicationId, contraindicationId) VALUES ('7','Lisinopril','40', '4', '8', '6');

INSERT INTO Medical\_Data (id, name, dose, categoryId, indicationId, contraindicationId) VALUES ('8','Amlodipine','10', '3', '8', '9');

INSERT INTO Medical\_Data (id, name, dose, categoryId, indicationId, contraindicationId) VALUES ('9','Mucinex','600', '3', '14', '7');

INSERT INTO Medical\_Data (id, name, dose, categoryId, indicationId, contraindicationId) VALUES ('10','Edoxaban','60', '3', '3', '8');

+----+------------------+------+------------+--------------+--------------------+

| id | name | dose | categoryId | indicationId | contraindicationId |

+----+------------------+------+------------+--------------+--------------------+

| 1 | Aspirin | 300 | 4 | 1 | 10 |

| 2 | Acetaminophen | 2000 | 1 | 1 | 6 |

| 3 | Ibuprofen | 3200 | 2 | 4 | 7 |

| 4 | Valsartan | 80 | 4 | 8 | 11 |

| 5 | Insulin glargine | 160 | 3 | 10 | 6 |

| 6 | Rifaximin | 1100 | 3 | 11 | 6 |

| 7 | Lisinopril | 40 | 4 | 8 | 6 |

| 8 | Amlodipine | 10 | 3 | 8 | 9 |

| 9 | Guaifenesin | 600 | 3 | 14 | 7 |

| 10 | Edoxaban | 60 | 3 | 3 | 8 |

+----+------------------+------+------------+--------------+--------------------+

**3- Design: UML Diagram & Algorithm**

**Client 🡨---🡪 Server 🡨--🡪 Database**

**Diagram

Description automatically generated**

**Algorithm:**

Class: Client extends Application

Declare data field:

fromServer: DataInputStream

toServer: DataOutputStream

PROCESS start(primaryStage:Stage)

Create a elements for a Panel:

Declare BorderPane, TextArea,GridPane

ComboBox with all possible medicine names

A Search button

Place items in gridpane and gridpane into Stage

TRY

Create a socket with localhost and portal 8000.

Attach socket object to Data input and output stream to get the stream.

CATCH any IO exception and append it to textarea to display

Set button on action

TRY

to send the selected ComboBox value to server application

Then Flush toServer instance

Read the UTF response body from Server and append it to textarea

CATCH any IO exception and print it as an error

ENDPROCESS

(Application starting point)

PROCESS main(args:String[]):void

Launch args

ENDPROCESS

ENDCLASS: Client

Class: Server extends Application

Declare data field:

An empty medicine String

Stmt:Statement

Ta:TextArea

inputFromClient: DataInputStream

outputToClient:DataOutputStream

PROCESS:start(primaryStage:Stage):void

Invoke method:InitializeDB with no parameter

Assign ta to a new TextArea

Set stage with scene and a title

Start a new Thread

TRY

Creating a server socket with a portal 8000

When the Platform is ready, append text to show the Date when the server started

Append text to ta to show when the server is connected with a client application

Set the socket to Data input and output stream to receive and send data

Create an infinite while loop

To read the medicine name from client application

When the Platform is available, display it in text area

Invoke the method: showInfo with no parameter

ENDWHILE

CATCH IOException and print it to console

ENDThread and call start method.

END PROCESS

PROCESS: initializeDB(no arg):void

TRY

Load Java Database Connectivity by calling forName method with "com.mysql.jdbc.Driver"

Create a connection instance and Call getCconnection on DriveManager

Provide the database name with username and password for an access

Print Database connected to console

Call createStatement on connection object and assign it to stmt Statement object

CATCH any Exception and print it to console

ENDPROCESS

PROCESS showInfo(no args):void

TRY

Creating the sql query statement with the medicine name as its condition

Select all the columns from Medical\_Data table with a JOIN with the foreignid columns from PregnancyCategory, Indications and Contraindications tables where medicine name is medicine name received from the client

Save the resultset in an ResultSet instance object:rset

Use an if/else statement to save results in corresponding variables

If there is next item in the rset

Save the name, dose, pregnancy category id, indication id and contraindication id in separate Strings.

Set the text area with the received response

TRY

Writing the sql query response in a formatted way to client application.

Flush the output data stream to clean the remaining data after the data is sent to client.

CATCH IOException and print it to console.

ELSE

Set the text of text area as “Message: Medicine information not found”.

ENDIF

CATCH SQLException

Print exception message to the console

ENDPROCESS

(Application starting point)

PROCESS main(args:String[]):void

Launch args

ENDPROCESS

ENDCLASS: Server

**4-Test cases**

**Case 1:**

On the Client panel:

If “Aspirin” is chosen and Submit button is clicked:

The text on the Server and Client GUI should be:\

Medicine name: Aspirin

Dose: 300mg (max for adults per day)

Pregnancy Category: D

Indications: Headache

Contraindications: High Blood Pressure

**Case 2:**

If “Valsartan” is chosen and Submit button is clicked:

The text on the Server and Client GUI should be:

Medicine name: Valsartan

Dose: 80mg (max for adults per day)

Pregnancy Category: D

Indications: Hypertension – High blood pressure

Contraindications: Low Blood Pressure

**Case 3:**

If “Lisinopril” is chosen and Submit button is clicked:

The text on the Server and Client GUI should be:

Medicine name: Lisinopril

Dose: 40mg (max for adults per day)

Pregnancy Category: D

Indications: Hypertension – High blood pressure

Contraindications: Liver disease

**Case 4:**

If “Insulin glargine” is chosen and Submit button is clicked:

The text on the Server and Client GUI should be:

Medicine name: Insulin glargine

Dose: 160mg (max for adults per day)

Pregnancy Category: C

Indications: Diabetes

Contraindications: Liver disease

**5- Users guide & Screen shots**

The user will interact with the program with the Client GUI application.

In the dropdown menu there are a number of medicine names.

The user can select a medicine name that they want to loop up to and select Search button

To receive information about that drug from the database through a Server application.

Server also has a GUI panel to show the info received by the server and the data fetched from the mySQL database that it is connected to.

Server GUI is created for the purpose of this project, normally clients only interact with the Client GUI Panel.

**Screen Shots:**

**Case 1:**

Graphical user interface, text, application

Description automatically generated

**Case 2:**

Graphical user interface, text, application

Description automatically generated

**Case 3:**

Graphical user interface, text, application

Description automatically generated

**Case 4:**

Graphical user interface, text, application

Description automatically generated

**The area of improvements of the project:**

I wanted to add more than one indication and contraindication for each medicine but could not find a way to add more than one foreign id value into indicationId and contraindicationId columns for each medicine row. For this project specifically, I wanted to use SQL queries with foreign id columns, that’s why I did not change the current database structure.

**Example:**

Current:

INSERT INTO Medical\_Data (id, name, dose, categoryId, indicationId, contraindicationId) VALUES ('1','Aspirin','300', '4', '1', '10');

Ideally:

INSERT INTO Medical\_Data (id, name, dose, categoryId, indicationId, contraindicationId) VALUES ('1','Aspirin','300', '4', '1, 2,5,6..etc', '10, 4,7..etc'); or any other way to show more data for these 2 columns.

In real life one medicine has more than one indication and contraindication, so it would be better to reflect it on this project, but as I mentioned, I could not find a way to apply it.

P.S: Professor, I appreciate it if you could let me know if there is a way to achieve this display.

Thank you for your time!