OOP JAVA PROJECT

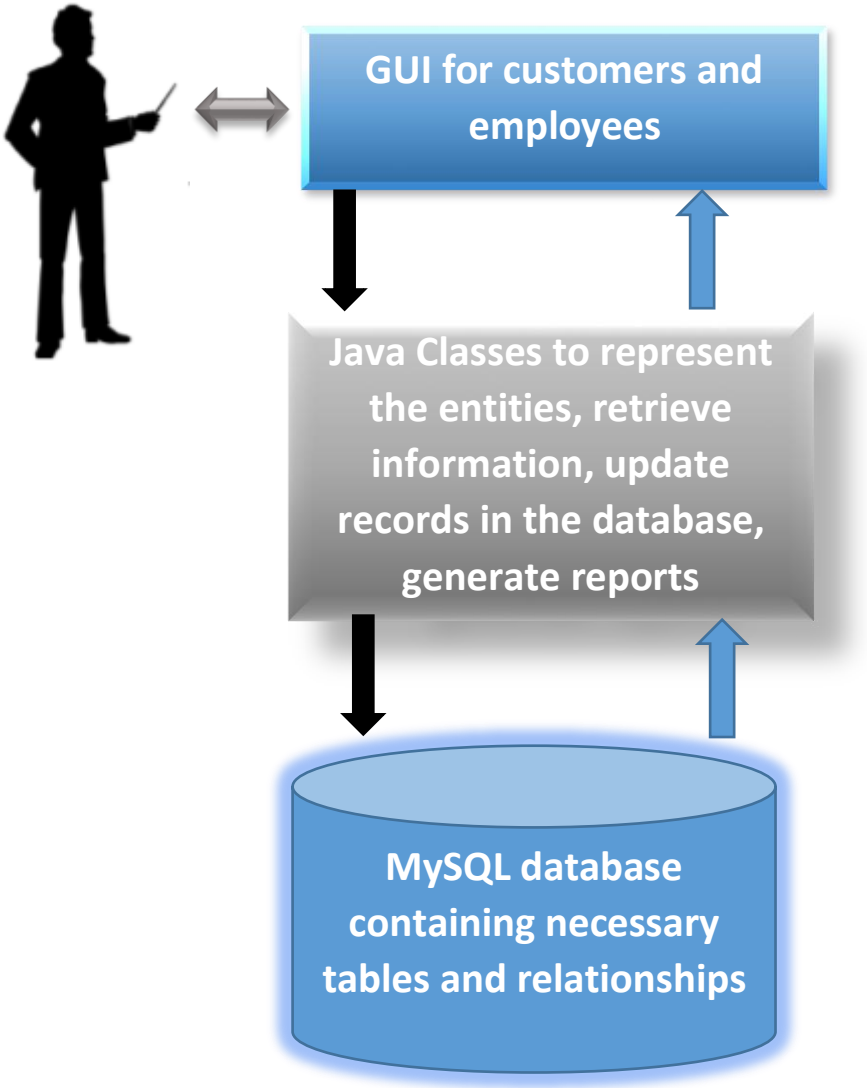
Contents

Group 4

Healthcare appointment booking project

Goal:

The goal of the project is to write an application to manage the appointments associated with healthcare services. The application the customer to book the appointments with the specialist doctors. The application will also support the healthcare management to maintain its records for bookings and customers.



Program Description:

In this program you will write a set of supporting classes for healthcare appointment booking application. Here is an example of GUI.

Pick a clinic & day:

*** Particular interest in Heart Conditions	Today 6th Nov	Tomorrow 7th Nov	Sunday 8th Nov	Monday 9th Nov	Tuesday 10th Nov	Wednesday 11th Nov	Thursday 12th Nov	Friday 13th Nov	Saturday 14th Nov	Sunday 15th Nov
Clapham Junction	12:45 - 14:30 K Chandan ***	N/A	N/A	09:30 - 14:15 A Chan	10:00 - 16:15 E Chege ***	12:45 - 14:30 K Chandan ***	10:00 - 16:15 E Chege ***	09:30 - 16:15 K Chandan ***	N/A	N/A
Victoria	12:30 - 16:15 E Chege ***	N/A	N/A	11:00 - 16:15 T Selvadurai	09:00 - 14:45 A Patel	N/A	13:00 - 16:45 A Chan	10:30 - 16:15 E Chege ***	N/A	N/A
Hammersmith	13:45 - 15:45 A Chan	N/A	N/A	N/A	09:30 - 15:45 M Moiz	N/A	N/A	10:15 - 15:00 A Chan	N/A	N/A
Liverpool Street	12:45 - 16:30 H Davies	N/A	N/A	09:00 - 16:45 J Stevenson	10:45 - 16:45 G Ndlovu	09:00 - 16:45 J Stevenson	10:30 - 16:45 G Ndlovu	10:30 - 16:45 G Ndlovu	N/A	N/A
Bond Street	13:30 - 16:30 J Stevenson	13:00 - 17:30 M Moiz	12:30 - 17:00 M Moiz	10:00 - 17:30 C Jogia ***	11:00 - 17:15 T Selvadurai	10:00 - 17:15 C Jogia ***	N/A	10:30 - 17:30 H Davies	10:00 - 17:30 M Moiz	12:15 - 17:00 M Moiz

The application should allow the customer/s to browse the available appointments with the specific doctors. The user should be able to choose the doctor as well as the time from the available slots and book the appointment. It is possible that one doctor runs the clinic at multiple locations. The patients can be of two types. New patients or returning patients. New patients will require to register and create a login while the returning patients will have an existing login to use. The returning patients will be able to browse through their previous appointments and notes about the appointments.

The application primarily involves details of the available appointments for the specific doctors, doctor specializations and qualifications, feels involved, customer details as well as booking records generated and maintained.

The application should be developed for two types of users:

1. Patients – Book an appointment, Review the previous appointments, browsing the availability etc.
2. Employees – Update the currently available appointments, introduce new doctors and appointments, maintain the customer records, review patient's history etc.

You are expected to design and develop the database for this application along with the Java classes necessary to implement the application.

Implementation requirements

- Necessary classes, methods and attributes should be designed using UML diagram notation. All the classes, methods and attributes should be explained in your documentation. Please discuss the design with me before you start implementation.
- You should be able to identify and introduce inheritance and aggregation relationship wherever applicable
- Necessary GUI screens should be added for successful execution.
- Records should be maintained in the database. Your Java code is expected to read and write to multiple tables as required.
- Every table must be populated with at least 6 records.

General System Architecture (MVC Pattern)

In this section, the general architecture for managing this schedule has been described. This system counts mainly 5 modules:

- The Information Search module: all possible requirements in the database, according to several search criteria
- The Data Update module : any modification, addition or deletion operation in the database
- The Reporting module: statistics in the form of graphs (pie charts, histograms etc.)
- The Data Access (DAO) module queries or updates the data in the database and communicates with the 3 modules previous
- The graphical interface communicates with the first 3 modules to visualize the schedule

According to the MVC pattern, your graphical interface constitutes the View (only the display) dependent on the actions of the user (event management) at Controller level (Research, Update and reporting modules).
This will ask the Model to retrieve or update- via the data access module (DAO) - the information of the database, organize and assemble them (for example, by storing them in collections). Then the Controller will ask the Model for the data, analyse it, make decisions and send the text back to the View.
You are advised to adopt the model for the development of a coherent project.
You can find more about the MVC Pattern at

https://openclassrooms.com/fr/courses/4670706-adoptez-une-architecture-mvc-en-php/4678736-comment-fonctionne-une-architecture-mvc

https://accu.org/journals/overload/16/88/grenyer_1524/

https://www.oracle.com/technical-resources/articles/javase/mvc.html

Guidance for structured development of project

Step 1: Relationship model

Review all possible requirements of the database and the search criteria. Identify the possible entities, attributes in the database. It is important to carefully recognize the role of each attribute and then decide the datatype of the attribute. It is will also play a key contribution towards determining the primary and foreign key attributes. Document the relationships between the entities.

Step 2: Creating database

Based on the relationship model above, create tables and relationships using MySQL. Insert the records into the tables.

Step 3: Information Finder

Review the user requirements to identify the possible range of information you need to retrieve from the database. Specifically in the case of business organizations, it is important for the employees to analyse the sales. It is equally essential for the customers to analyse their past purchases. For example:

- The doctor which was high in demand
- The number of per area/per age group

Develop the necessary classes to represent entities which will enable the user to query the database.

Step 4: Data access

This module represents the data access layer (DAO) in the DB. Via a JDBC access to the database, this module executes the queries responsible for retrieving or updating the data in the database. This is a type of object that loads to retrieve the data in the database and that another type of object be used to handle this data (business layer).

Step 5: GUI & Reporting

A welcome window will allow the user to connect to the database by entering their EMAIL and PASSWORD. These information, if stored in the USER table, will give him access or / and update rights to certain data of the schedule.

Your graphical interface will display in an ergonomic, clear and fluid way all the relevant information. It will allow you to navigate intuitively from one page to another. For example, a page of your interface graph can contain menus with menu items, or tabs if you prefer.

This module is used to generate statistics (pie charts, histograms, etc.) using JFreeChart. You can find the details in the resources section.

Deliverables

The deliverable should be a zipped file per team containing the following:

1. A PowerPoint presentation including
    a. Title
    b. Name of the team members
    c. Summary
    d. Class diagram(draw.io) or any other similar tool
    e. Database design
    f. GUI screen samples
    g. Your own evaluation of the project
    h. Bibliography
2. Java code: All the folders and files of the project developed on Eclipse or Netbeans with the sources, the executable .jar in mode graphics and Javadoc documentation with comments on classes and methods.

Resources

JDBC: https://www.jmdoudoux.fr/java/dej/chap-jdbc.htm, (Author: Jean-Michel Doudoux)

My SQL: http://dev.mysql.com/doc/refman/5.7/en/

JFreeChart:

The JFreeChart Class Library (Author: David Gilbert)

http://www.jfree.org/jfreechart/api/javadoc/index.html

http://www.java2s.com/Code/Java/Chart/CatalogChart.htm

http://www.jfree.org/forum/

Wireframe: https://webdesign.tutsplus.com/articles/a-beginners-guide-to-wireframing--webdesign-7399