Consultation Manager

Analysis and Design Document

Student: Matei Ionela Stefania

**Group:30238**

Table of Contents

1. Requirements Analysis 3

1.1 Assignment Specification 3

1.2 Functional Requirements 3

1.3 Non-functional Requirements 3

2. Use-Case Model 3

3. System Architectural Design 3

4. UML Sequence Diagrams 3

5. Class Design 3

6. Data Model 3

7. System Testing 3

8. Bibliography 3

1. Requirements Analysis

# Assignment Specification

Use Java/C# API to design and implement a client-server application for managing the consultations of doctors in a clinic. The application has three types of users: the clinic secretary, the doctors and an administrator.

# Functional Requirements

The clinic secretary can perform the following operations:

* Add/update patients (patient information: name, identity card number, personal numerical code, date of birth, address).
* CRUD on patients’ consultations (e.g. scheduling a consultation, assigning a doctor to a patient based on the doctor’s availability).

The doctors can perform the following operations:

* Add/view the details of a patient’s (past) consultation.

The administrator can perform the following operations:

* CRUD on user accounts.

In addition, when a patient having a consultation has arrived at the clinic and checked in at the secretary desk, the application should inform the associated doctor by displaying a message.

containing the activities performed by an employee.

# Non-functional Requirements

Testability – the system will be tested by using test classes.

*Security*: all data should be protected by a password

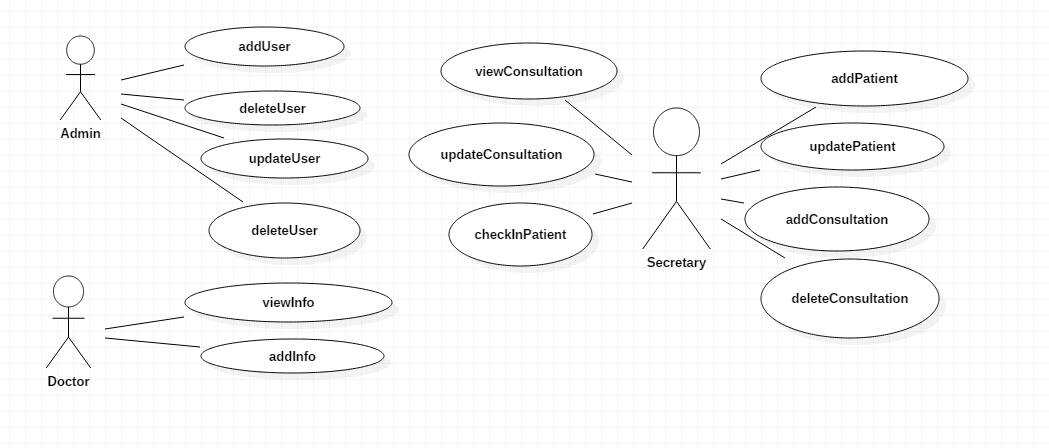
*Availability*: system must be available all the time, for all types of users. The only condition for availability, is that the server must run and must be accessible by the clients.

*Reusability:*some of the classes that have been used in this project, may be used in other projects because of their construction

*Extensibility*: because the application architecture is client-server, the application can be easily extended by adding new features and new message types to server.

Usability – the system will be easy to use by anyone

2. Use-Case Model



Use case: Add user

Level: user-goal

Primary actor: Administrator

Main success scenario: User is successfully created in model.

Use case: View user

Level: user-goal

Primary actor: Administrator

Main success scenario: User is successfully retrieved from model.

Use case: Add consultation

Level: user-goal

Primary actor: Secretary, Doctor

Main success scenario: Consultation is successfully created in the database.

Use case: View consultation

Level: user-goal

Primary actor: Secretary, Doctor

Main success scenario: Consultation is successfully retrieved from the database.

3. System Architectural Design

**3.1 Architectural Pattern Description**

The client–server model of computing is a distributed computing structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.[1] Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server host runs one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.

Examples of computer applications that use the client–server model are Email, network printing, and the World Wide Web.

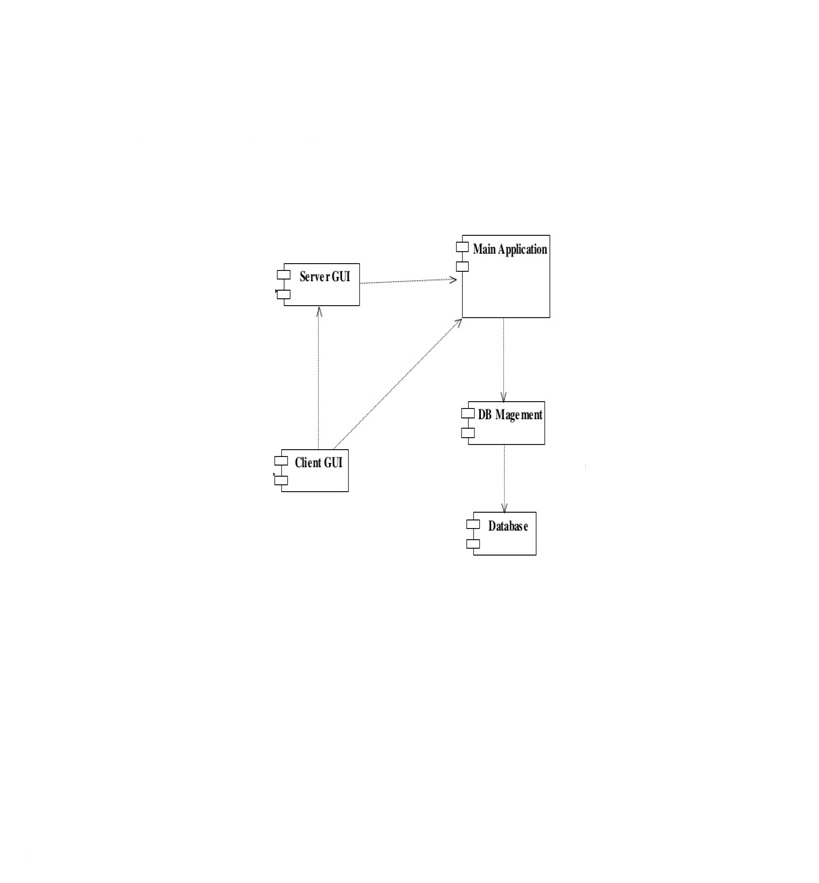
The client–server characteristic describes the relationship of cooperating programs in an application. The server component provides a function or service to one or many clients, which initiate requests for such services.

Servers are classified by the services they provide. For instance, a web server serves web pages and a file server serves computer files. A shared resource may be any of the server computer's software and electronic components, from programs and data to processors and storage devices. The sharing of resources of a server constitute a service.

Whether a computer is a client, a server, or both, is determined by the nature of the application that requires the service functions. For example, a single computer can run web server and file server software at the same time to serve different data to clients making different kinds of requests. Client software can also communicate with server software within the same computer.[2] Communication between servers, such as to synchronize data, is sometimes called inter-server or server-to-server communication.

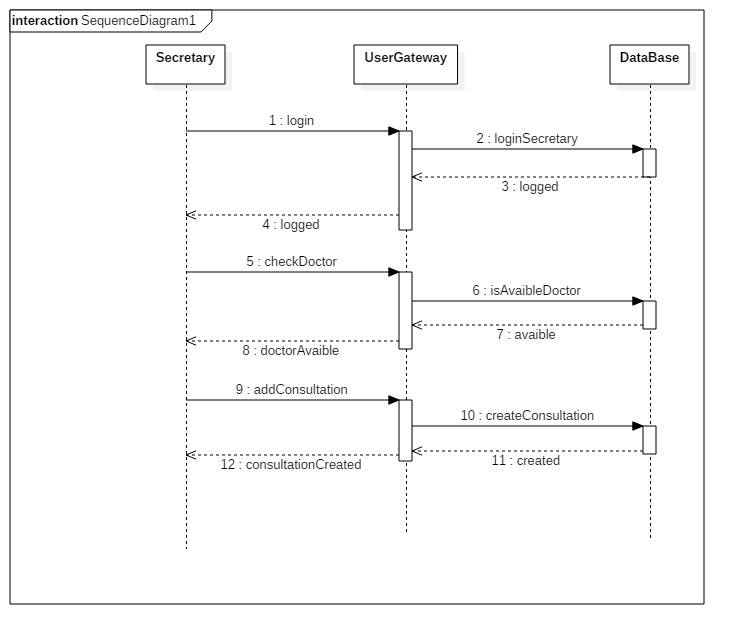
**3.2 Diagrams**

**3.2.1 Component Diagram**



4. UML Sequence Diagrams

Add Consultation

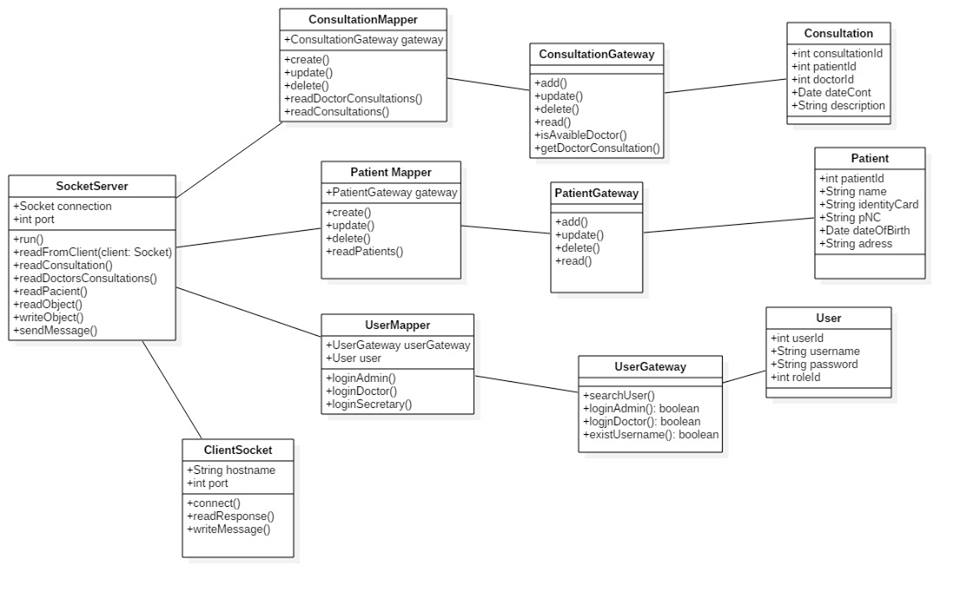


5. Class Design

**5.1 Design Patterns Description**

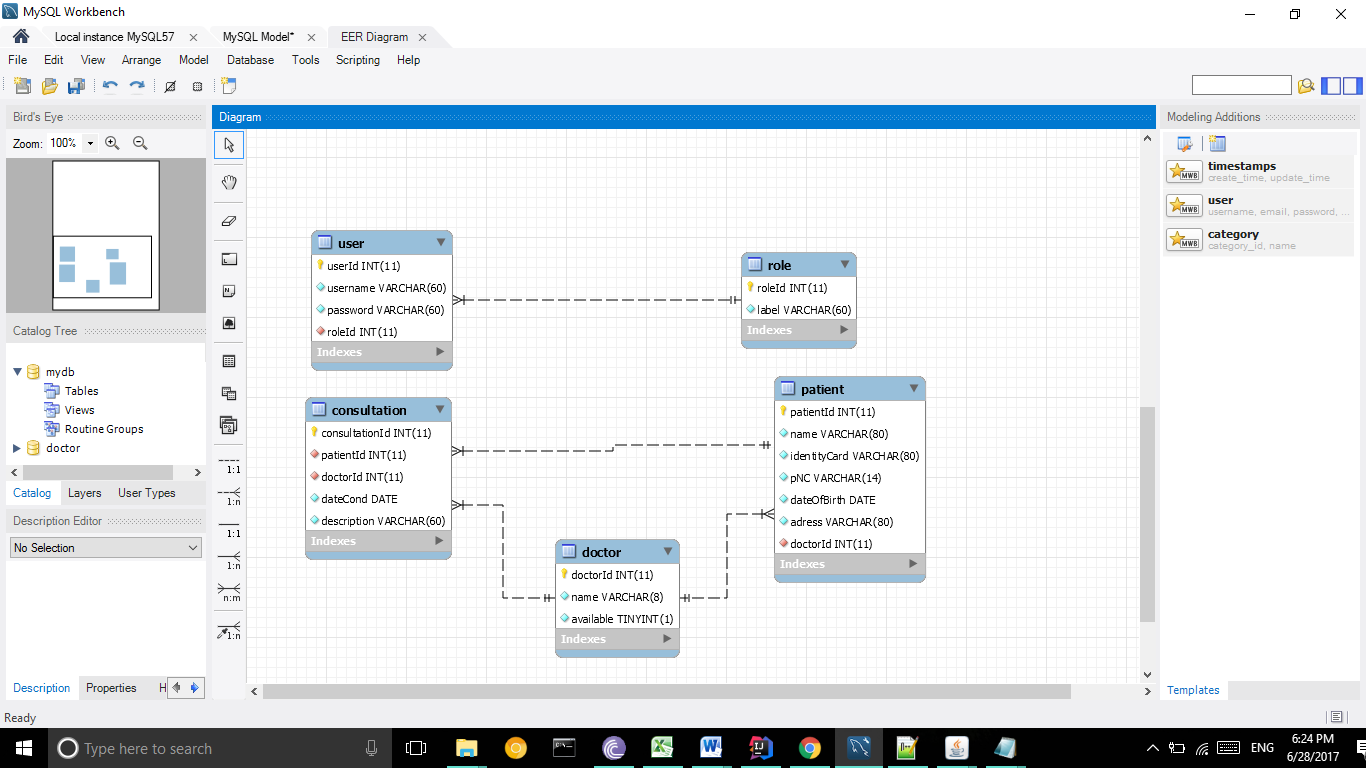
A Table Data Gateway holds all the SQL for accessing a single table or view: selects, inserts, updates, and deletes. Other code calls its methods for all interaction with the database.

**5.2 UML Class Diagram**



6. Data Model

The Data Model is represented by the connections between the tables from the database.



7. System Testing

I tested the exam by using manual testing.

8. Bibliography