Student:Iuga Anca

**Group:30234**

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 4

3. System Architectural Design 4

4. UML Sequence Diagrams 6

5. System Testing 6

6. Bibliography

1. Requirements Analysis

# Assignment Specification

Use the 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.

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.

# Functional Requirements

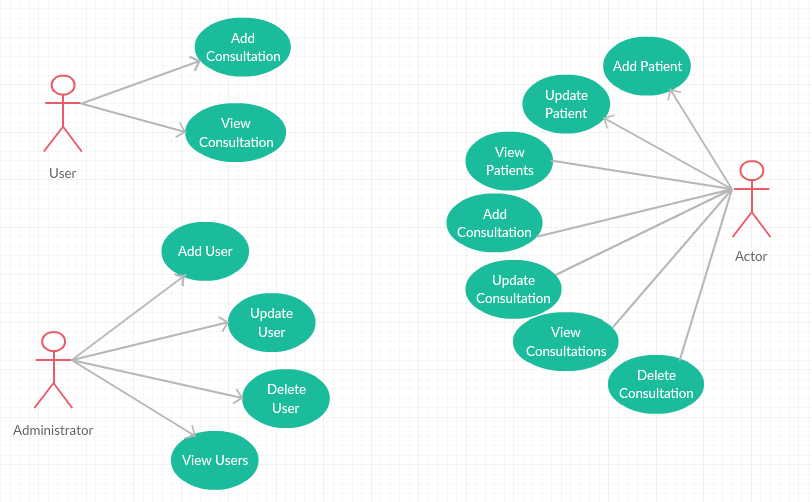
The operations of CRUD on patients and consultations.

# Non-functional Requirements

The security operation on user and the operations which store data in database .

2. Use-Case Model

# 2.1 UML Use-Case Diagrams



# 2.2 Use-Cases Identification

Use case: Add Patient

Level: user-goal level

Primary actor: Secretary

Main success scenario:

* The secretary introduces dates about the new user
* The server introduces dates in database

3. System Architectural Design

**3.1 Architectural Pattern Description**

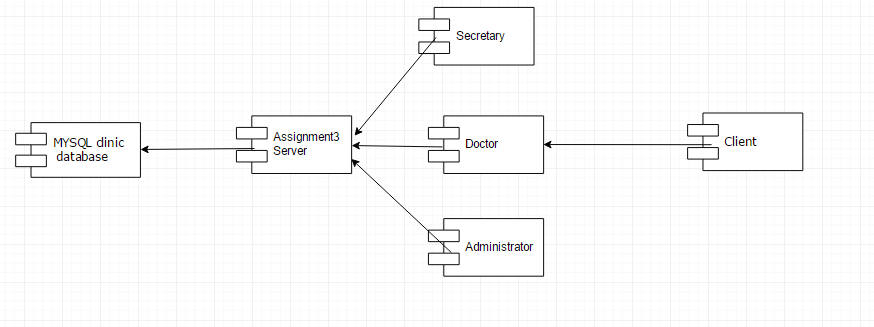
The **client–server model** is a [distributed application](https://en.wikipedia.org/wiki/Distributed_application) structure that partitions tasks or workloads between the providers of a resource or service, called [servers](https://en.wikipedia.org/wiki/Server_(computing)), and service requesters, called [clients](https://en.wikipedia.org/wiki/Client_(computing)). Often clients and servers communicate over a [computer network](https://en.wikipedia.org/wiki/Computer_network) on separate hardware, but both client and server may reside in the same system. A server [host](https://en.wikipedia.org/wiki/Host_(network)) runs one or more server programs which share their resources with clients.



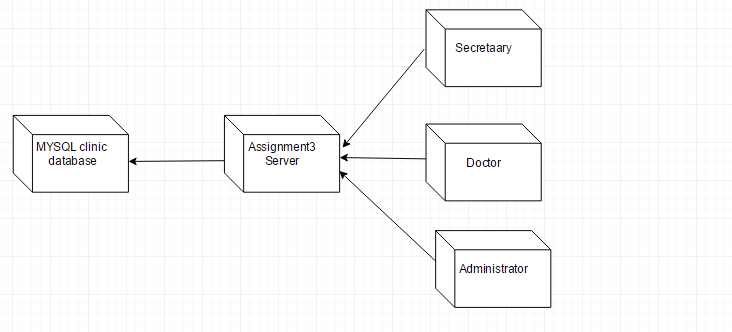
The picture above , illustrates how the client-server architectural pattern work .

**3.2 Diagrams**

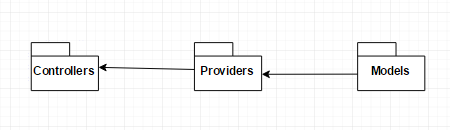
**3.2.1 Component Diagram**

****

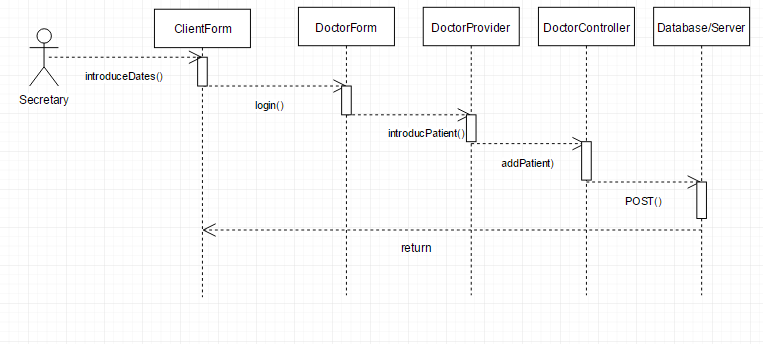
**3.2.2 Deployment Diagram**

****

**3.2.3 Package Diagram**

****

4. UML Sequence Diagrams

I created a sequence diagram for the add patient case, where the secretary request the server to add a new patient.

5. Data Model

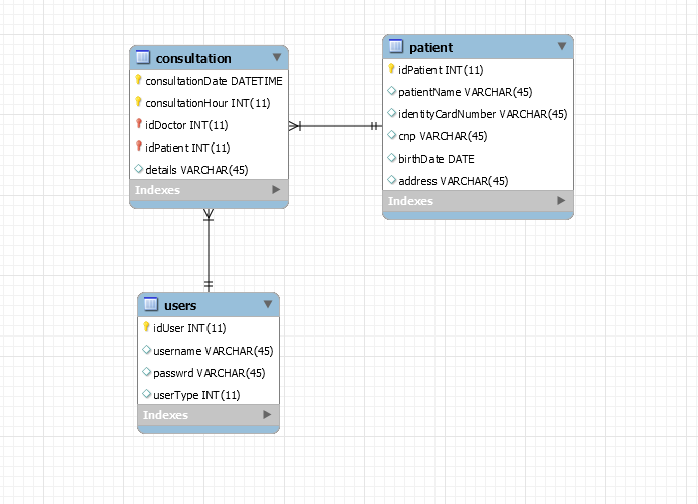
My data base contains the following table:

* User
* Consultation
* Patient

The patient have attributes like: name ,address, identification id, date of birth, cnp and identification card number.

The consultation have attributes like: the date and the hour, the id of the patient and the id of the doctor and details about the consultation. To be sure that we do not have a consultation at the same time , I set a multiple primary key on the following attributes :consultationDate, consultationHour, idPatient and idDoctor .

The users have basic attributes like username, password, identification number and type.



6. Bibliography

<https://en.wikipedia.org/wiki/Client%E2%80%93server_model>

<http://stackoverflow.com/>