Student: Sergiu Redeca

**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 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 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

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 the secretary desk, the application should inform the associated doctor by displaying a message.

# Functional Requirements

The users can perform the previously mentioned operations.

# Non-functional Requirements

*The software is easy-to-use.*

2. Use-Case Model

*Use-Case description format:*

*Use case: Admin logs in and does some operations*

*Level: summary level*

*Primary actor: administrator*

*Use case: <use case goal>*

*Level: <one of: summary level, user-goal level, sub-function>*

*Primary actor: <a role name for the actor who initiates the use case>*

*Main success scenario: <the steps of the main success scenario from trigger to goal delivery>*

*Extensions: <alternate scenarios of success or failure>*

*]*

3. System Architectural Design

**3.1 Architectural Pattern Description**

*The client server architectural pattern separates the data requesters (clients) from the data providers (servers).*

*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.*

**3.2 Diagrams**



4. UML Sequence Diagrams



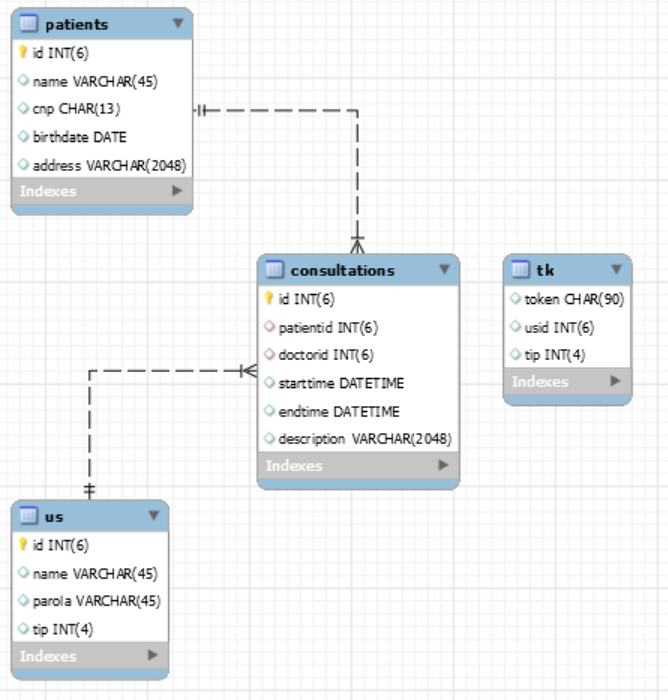
5. Paradigms

**5.1 Multithreading**

*Multithreading is the ability of a central processing unit (CPU) or a single core in a multi-core processor to execute multiple processes or threads concurrently, appropriately supported by the operating system. This approach differs from multiprocessing, as with multithreading the processes and threads share the resources of a single or multiple cores: the computing units, the CPU caches, and the translation lookaside buffer.*

*Where multiprocessing systems include multiple complete processing units, multithreading aims to increase utilization of a single core by using thread-level as well as instruction-level parallelism. As the two techniques are complementary, they are sometimes combined in systems with multiple multithreading CPUs and in CPUs with multiple multithreading cores..]*

6. Data Model

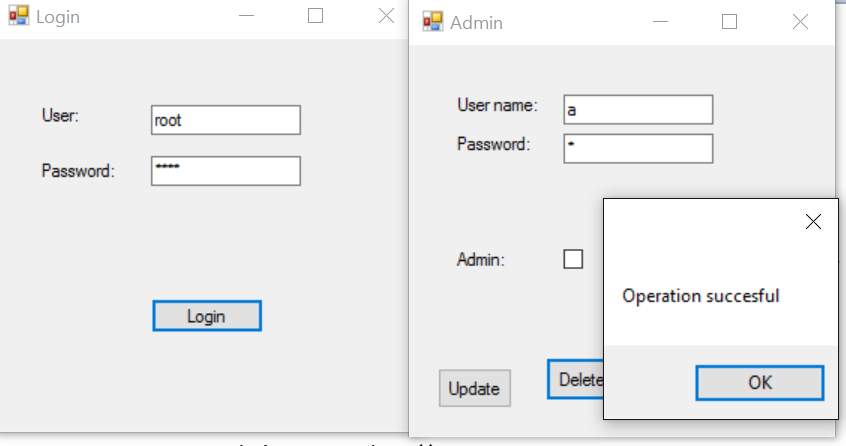
**

7. System Testing

*The system testing have been made using the .NET program and also using MySQL Workbench.*

*The main strategy used for testing was manual testing.*

*In the program there are displayed several error messages if the user makes an error and also there are some messages displayed in case of success.*

**

8. Bibliography

<https://docs.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api>

<https://github.com/zflaviu/StudentsAPI>

<https://msdn.microsoft.com/en-us/library/hh833994(v=vs.108).aspx>

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

<http://stackoverflow.com/questions/1195896/threadstart-with-parameters>

<https://en.wikipedia.org/wiki/Multithreading_(computer_architecture)>