Student: Luminita Marghescu

Group: 30234

Assignment 3

Software Design

Table of Contents

1. Requirements Analysis

1.1 Assignment Specification

The C# API implements 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.

1.2 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.

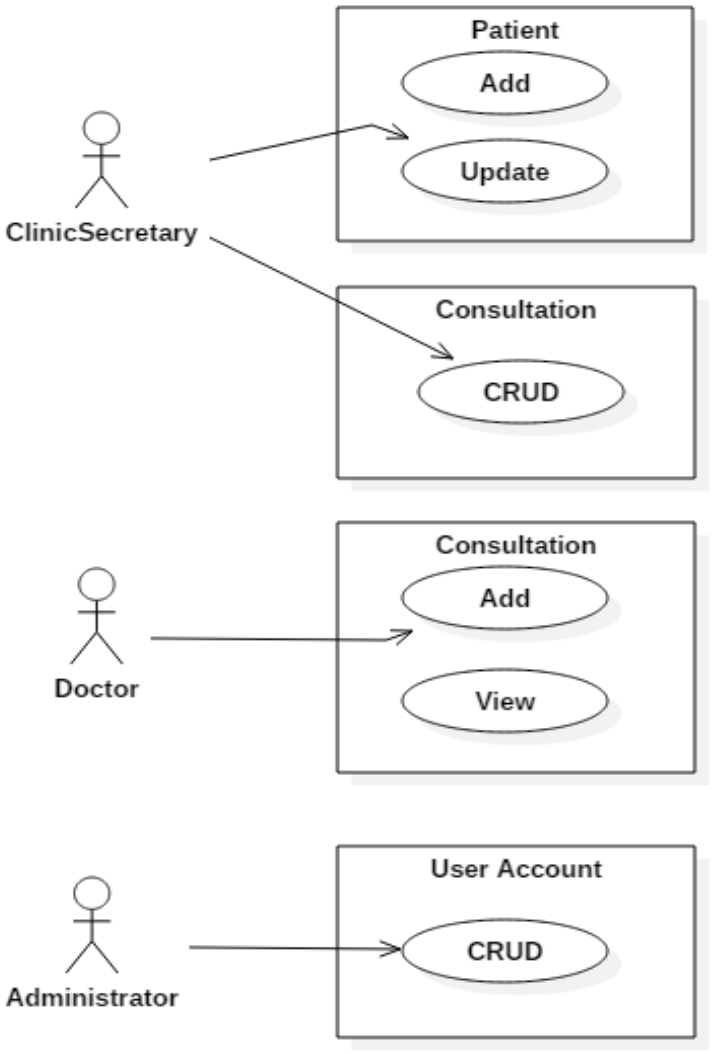
In addition, when a patient having a consultation has arrived at the clinic and checked in at the secretary desk, the application should inform the associated doctor by displaying a message.

1.3 Non-functional Requirements

- Accessibility
- Availability
- Data back-up
- Efficiency
- Price
- Privacy
- Portability
- Response Time
- Safety
- Security
- Testability

2. Use-Case Model

Use case: <Add Patient>
Level: <sub-function>
Primary actor: <Clinic Secretary>
Main success scenario: <Login as Clinic Secretary, Add Patient>
Extensions: <Scenario of failure: Login as Administrator>

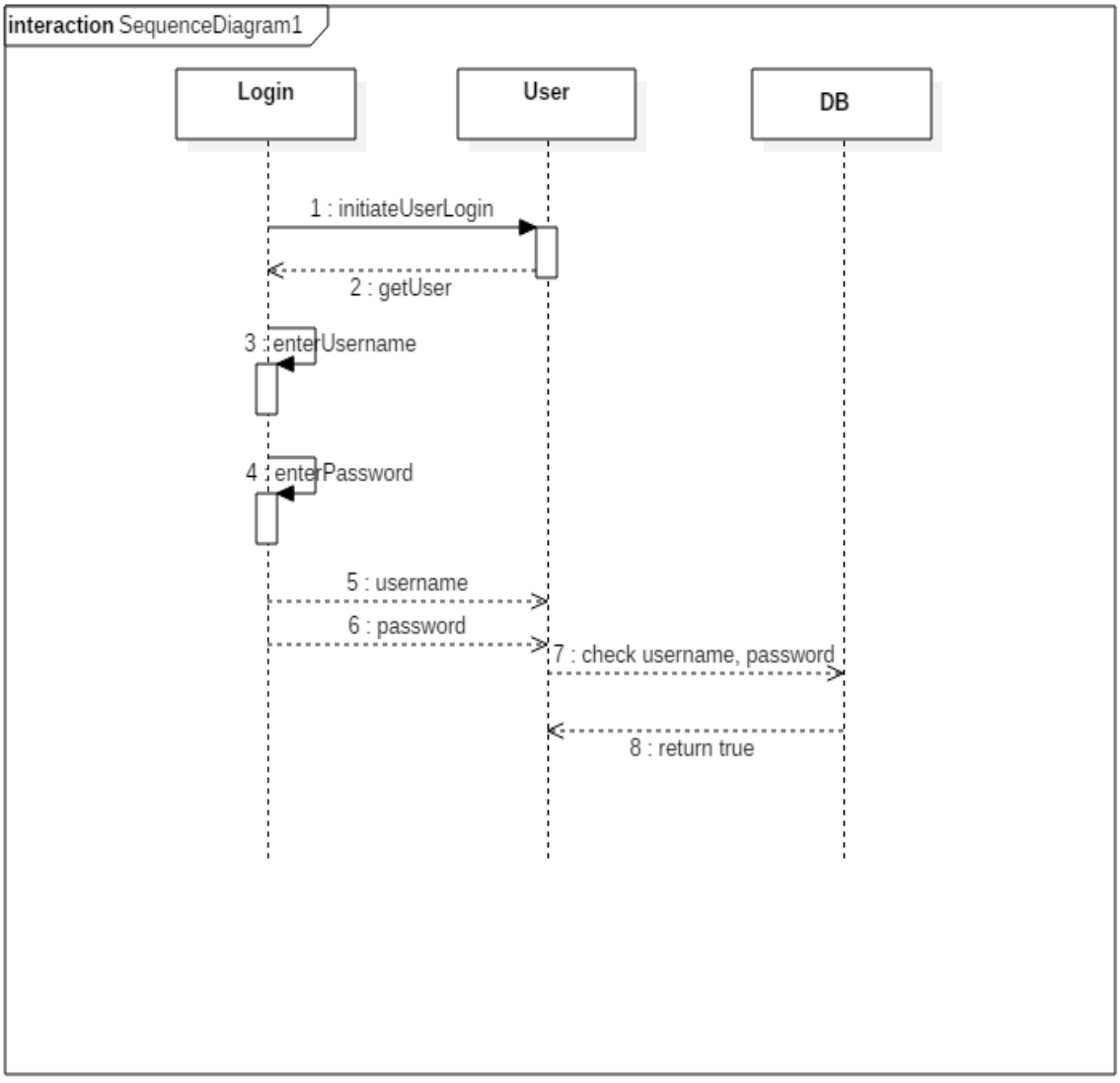


3. System Architectural Design

I divided the project into:

- AdministratorClient
- ClinicSecretaryClient
- DoctorClient
- ClinicServer

4. UML Sequence Diagrams



6. Data Model

I created this DB for using the application:

Database: clinic, Table: consultations, Purpose: Dumping data

idConsultation	idDoctor	idPatient	date	treatment
0	1	1	2017-05-19	analgezice
0	0	1	2017-05-18	a

Database: clinic, Table: doctors, Purpose: Dumping data

idDoctor	idUser	name	specialization
1	3	bogdan	chirurg
3	0	mihai	chirurg
6	0	luminita	oftalmolog

Database: clinic, Table: patients, Purpose: Dumping data

idPatient	Name	IdentityCardNumber	CNP	BirthDate	Address
1	Diana	111111	1111111111	2017-05-16	Observator

Database: clinic, Table: secretaries, Purpose: Dumping data

idSecretary	idUser	name	address
1	2	diana	Observator
2	0	flaviu	Cluj

Database: clinic, Table: users, Purpose: Dumping data

idUser	username	password	isSecretary	isDoctor	isAdmin
1	luminita	diana	0	0	1
2	diana	luminita	1	0	0
3	bogdan	nicoleta	0	1	0
4	mihai	luminita	0	1	0
5	luminita	83592796BC17705662DC9A750C8B6D0A4FD93396	0	1	0
6	luminita	83592796BC17705662DC9A750C8B6D0A4FD93396	0	1	0
7	flaviu	83592796BC17705662DC9A750C8B6D0A4FD93396	1	0	0

7. System Testing

I used an unit test for the method called "AddPatient" (I tested only one individual unit of source code) to see whether it fits for use. The test returned true, so I implemented it correctly. This is a Dataflow test because the user follows the value of variables and the points at which these values are used.

8. Bibliography

1. http://www.asp.net/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api