

# **OBJECT ORIENTED PROGRAMMING SECOND MIDTERM PROJECT**

Tugay COŞKUN	1030510263
Emirhan DAĞDELEN	1030510319
Tunahan AKÇAY	1030510285
Ahmet Faruk ERDEM	1030510309



# HOSPITAL INFORMATION SYSTEM

## REQUIREMENTS WORKFLOW

### • APPLICATION DOMAIN

Hospital Information System is a software used for the follow-up of hospital works. It includes processes such as registering and logging into the hospital system, making an appointment, determining test and diagnosis, and writing prescriptions. The Hospital Information System saves time by helping patients, doctors and technicians maintain hospital operations in a more functional way. The Hospital Information System is useful for keeping track of a doctor's appointments as well as his patients.

### IMPORTANCE OF HOSPITAL INFORMATION SYSTEM

- The Hospital Information System is the most reliable system for the most efficient and easy management of all processes in a hospital.
- It will become easier to store data securely, and then the process of searching for any person's information will be faster.
- This system will reduce all manual work and the whole process can be managed with one click and editing.
- Anyone registered in the system can log in with their information and make an appointment and edit their appointment information.
- Technician and doctor also make their work easier with some extra powers and privileges. .

A single person can handle an entire system.

### Questions asked to the doctor about the library system:

1.Q- How do you perform the login process?

1.A- Login to the system with TR ID number. If the SMS sent to the phone is entered on the screen correctly, the login process is successfully completed.

2.Q - How are appointments listed?

2.A - Our patients appear in the system automatically when they are logged into the system.

3.Q - How do you access the information of the patients who will come to the appointment?

3.A - I click on the patients table and I am shown patients with appointments. When I click on that patient, I access their information.

4.Q - How do you perform the diagnosis of the patient?

4.A - I enter the diagnostic information given to the patient into the system and the system saves them.

5.Q - How do you create test information for the patient?

5.A - We enter the tests we have created for the patient in the tests table.

## 6.Q - How to prescribe medicine to the patient?

6.A - The laboratory staff sends us the final diagnoses of the patient, and in this way, we perform the prescribing process.

## SURVEY EXAMPLE

Variables	Intervention group (n=133)	Control group (n=139)	P value	Characteristics	Intervention group (n=154)	Control group (n=155)
Overall level of satisfaction with hospital stay:				Median (interquartile range) age (years)	85 (82-88)	84 (81-87)
Very satisfied	125 (93)	91 (65)	<0.001	Men	83 (54)	64 (41)
Satisfied	6 (5)	40 (29)		Main admission diagnosis:		
Not satisfied	2 (2)	8 (6)		Cardiac	53 (34)	50 (32)
Satisfaction with information provided in hospital:				Respiratory	50 (33)	47 (30)
Very satisfied	115 (86)	58 (42)	<0.001	Falls	13 (8)	22 (14)
Satisfied	14 (10)	56 (40)		Other	38 (25)	36 (24)
Not satisfied	4 (4)	25 (18)		Do not resuscitate form completed by medical team at admission	26 (17)	32 (21)
Satisfaction with being listened to in hospital:				Surrogate* appointed before admission:		
Very satisfied	127 (94)	72 (52)	<0.001	Yes	18 (12)	20 (13)
Satisfied	4 (4)	47 (34)		No	130 (84)	133 (86)
Not satisfied	2 (2)	20 (14)		Unknown	6 (4)	2 (1)
Satisfaction with level of involvement in decisions made in hospital:				Would want cardiopulmonary resuscitation†:		
Very satisfied	123 (92)	53 (38)	<0.001	Yes	72 (47)	80 (52)
Satisfied	8 (6)	64 (46)		No	58 (38)	59 (38)
Not satisfied	2 (2)	22 (16)		Don't know	24 (16)	16 (10)
Satisfaction with level of family involvement in decisions made in hospital:				Would want life prolonging treatment‡:		
Very satisfied	123 (92)	64 (46)	<0.001	Yes	114 (74)	122 (79)
Satisfied	7 (6)	58 (42)		No	37 (24)	31 (20)
Not satisfied	3 (2)	17 (12)		Don't know	3 (2)	2 (1)

\*Surrogate decision maker.

†Patient's wishes, as collected by medical researchers during eligibility and enrolment interview before randomisation.

Outcomes	Not appropriate*	Discussion, no wishes†	Discussion card only‡	Documented advance care plan§	P value
Patients	6 (5)	11 (9)	38 (30)	Total 70 (56); surrogate only, 26 (21); end of life wishes only, 14 (11); both, 30 (24)	—
Family present	2 (33)	6 (55)	17 (45)	65 (93)	<0.001
<b>End of life wishes</b>					
<b>Cardiopulmonary resuscitation:</b>					
Yes	NA	NA	3 (8)	0 (0)	—
Yes, depending on outcome¶	NA	NA	12 (32)	21 (30)	—
No	NA	NA	14 (36)	39 (56)	—
Decided by doctor or surrogate	NA	NA	9 (24)	10 (14)	—
<b>Life prolonging treatment:</b>					
Yes	NA	NA	3 (8)	0 (0)	—
Yes, depending on outcome¶	NA	NA	12 (32)	27 (39)	—
No	NA	NA	10 (26)	29 (41)	—
Decided by doctor or surrogate	NA	NA	11 (29)	14 (20)	—
Undecided	NA	NA	2 (5)	0 (0)	—

\*Facilitator decided to stop discussion because patient failed to grasp concept or refused further discussion.

†Advance care planning discussion held but patient elected not to appoint a surrogate or express any wishes.

‡Patient was clear about wishes but did not wish to complete an advance care plan. Wishes were documented by facilitator on an advance care planning discussion card filed in front of medical record.

§Legal appointment of surrogate or signed witnessed documentation of end of life wishes, or both, including cardiopulmonary resuscitation and life prolonging treatment.

¶Treatment to be provided only if doctor anticipates good outcome.

# DICTIONARY

- ***Patient***

A person who is receiving medical care, or who is cared for by a particular doctor or dentist when necessary.

- ***Doctor***

A person with a medical degree whose job is to treat people who are ill or hurt.

- ***Polyclinic***

A place that is not a hospital where doctors offer care and advice relating to many different medical conditions.

- ***Appointment***

A formal arrangement to visit hospital at a particular time and.

- ***Test***

The act of looking at or considering the patient carefully in order to discover the problem.

- ***Register***

To put information, especially your name, birth date etc. into an official list or record.

- ***Login***

Login with the information that already have been registered in the system.

- ***ID Card***

An official card with your name, date of birth, photograph, or other information on it that proves who you are.

- ***Date***

A numbered day in a month, often given with a combination of the name of the day, the month, and the year.

- ***Diagnosis***

A judgment about what a particular illness or problem is, made after examining it.

- ***Prescriptions***

A piece of paper on which a doctor writes the details of the medicine or drugs that someone needs.

- ***Laboratory***

A room or building where patient diagnoses are examined.

- ***Sequence Number***

When you took an appointment, you have automatically get an sequence number.

- ***Illness***

The state of being physically or mentally ill.

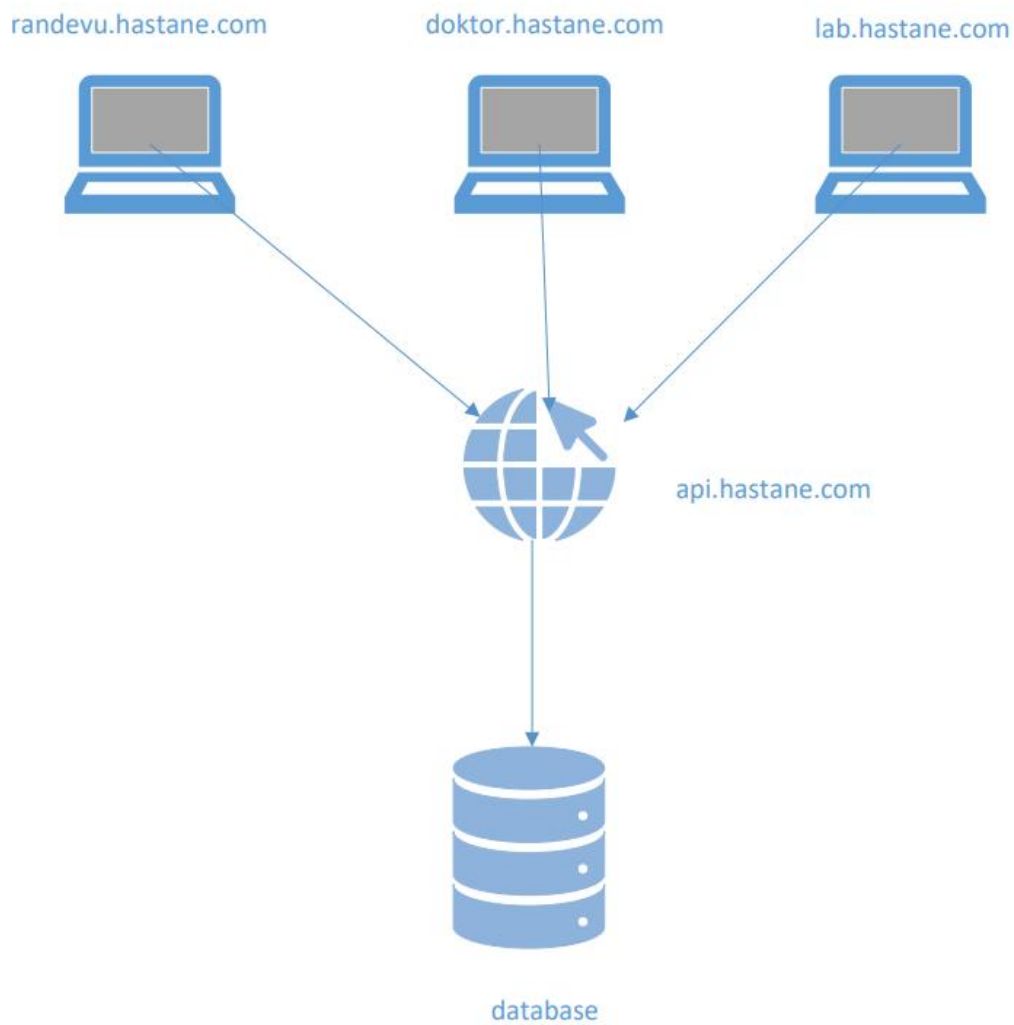
- ***Medicine***

It is a chemical substance used for the diagnosis, treatment, prevention of diseases, facilitating surgical interventions and changing some physiological events.

- ***Technician***

A person whose job is to deal with diagnostics.

## PROJECT OVERVIEW



## • BRIEF DESCRIPTIONS AND STEP-BY-STEP DESCRIPTIONS OF EACH DIAGRAM

### APPOINTMENT SYSTEM

#### REGISTER TO SYSTEM USE-CASE

This use case system takes user information and saves it to the system.

1. For each person who will register in the system
  - 1.1 TR no. Date of birth and phone number
  - 1.2 Confirmation code will be entered on the phone
  - 1.3 If the confirmation code is entered correctly, the registration is successful.

#### GET APPOINTMENT USE CASE

This use case allows the patient to make an appointment.

- 1 Initiates a series of procedures by which the patient can make an appointment.

#### CHOOSE POLİKLİNİK USE-CASE

It sends the user to the POLYCLINIC selection panel.

- 1 Saves the ID of the outpatient clinic selected by the patient to the local storage.

#### CHOOSE DOCTORS USE-CASE

Sends the user to the DOCTOR selection panel.

- 1 Saves the ID of the DOCTOR selected by the patient to the local storage.

#### CHOOSE DATE USE-CASE

Sends the user to the date selection panel.

- 1 saves the date selected by the user to the local data.

#### LOGİN TO SYSTEM USE-CASE

This use case takes the TC number and logs into the system.

- 1 For each person who will log into the system.
  - 1.1 An sms is sent to the TC registered in the system and the tel number registered to the TC.
  - 1.2 If the user has entered the incoming sms correctly, the login to the system is successful.
  - 1.3 If the user enters the incoming sms incorrectly, a notification is sent to repeat the sms sending.
2. If the user is successfully logged in, the system is allowed to login.

#### GET POLYCLİNİCS USE CASE

This use case enables the outpatient clinics in the hospital to be brought.

- 1 Returns the polyclinics pulled from the database in Json format.

#### GET DOCTORS USE-CASE

This use case ensures that the DOCTORS in the hospital are brought in.

- 1 Returns the DOCTORS pulled from the database in Json format.

#### GET DATE USE-CASE

This use case ensures that the available times of the doctor selected by the patient are brought in.

- 1 brings the appointments of the relevant doctor in the appointment table in the database.
- Returns 2 general appointments in Json format when the doctor has no appointments.

#### INSERT APPOINTMENT USE CASE

retrieves saved data and adds records to the appointments table

- 1 Retrieves saved data from local storage.
- 2 processes this data into appointment table in Json format.

## DOCTORS PANEL

### LOGIN TO SYSTEM USE-CASE

This use case takes the TC number and logs into the system.

- 1 For each doctor to log into the system.
  - 1.1 An sms is sent to the TC registered in the system and the tel number registered to the TC.
  - 1.2 If the doctor entered the incoming sms correctly, the login to the system is successful.
  - 1.3 If the doctor enters the wrong sms, a notification is sent to repeat the sms sending.
2. If the user is successfully logged in, the system is allowed to login.

### REQUEST PATIENT INFO USE CASE

The doctor requests the information of the patient in the examination.

1. requests patient information with the ID of the corresponding patient in the patients table in the database.

### REQUESTED TEST RESULT USE CASE

Returns the requested test result

1. Sends the requested examination ID.

### PRESCRIBE MEDICINE USE CASE

Sends the drug entered to the patient

- 1 Patient sends Id and drug prescriptions.

### SHOW APPOINTMENT USE CASE

Lists the doctor's appointments for that day.

### REQUEST TEST USE CASE

Requests tests for the patient in the examination

- 1.requests a patient's examination

### DIAGNOSIS USE CASE

Send the diagnosis made

- 1.Doctor enters diagnostic information
2. The entered information is sent to the API.

## LABORATORY

### LOGIN TO SYSTEM USE-CASE

This use case takes the TC number and logs into the system.

- 1 For each doctor to log into the system.
  - 1.1 An sms is sent to the TC registered in the system and the tel number registered to the TC.
  - 1.2 If the technician entered the incoming sms correctly, login to the system is successful.
  - 1.3 If the technician enters the incoming sms incorrectly, a notification is sent to repeat the sms sending.
- If the 2nd technician has successfully logged in, the system is allowed to log in.

### SHOW REQUESTED TEST USE CASE

Lists all requested tests.

For each test, it is displayed in the table.

### SEND TEST RESULT USE CASE

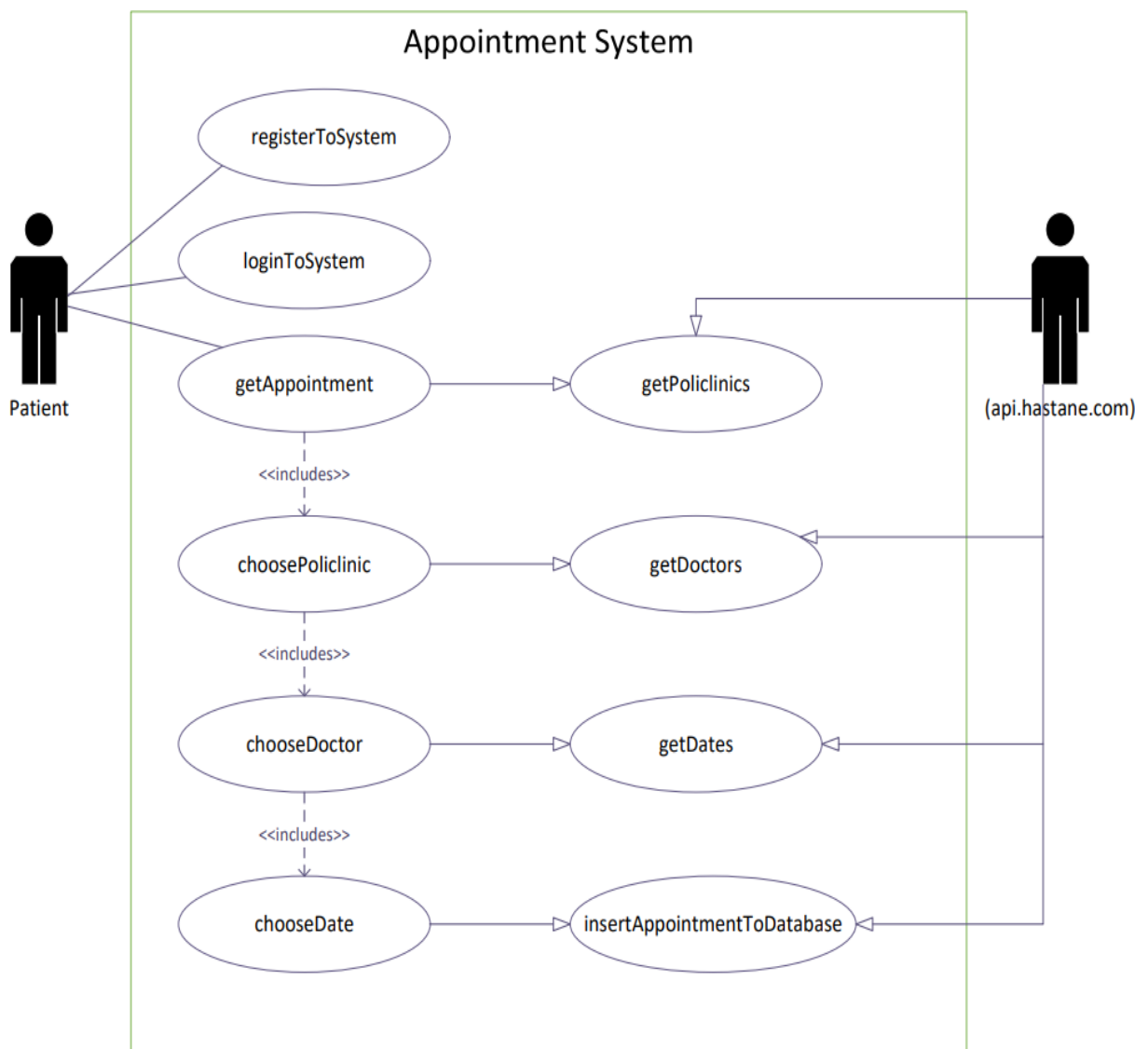
SUBMIT results of tests.

sends the test ID

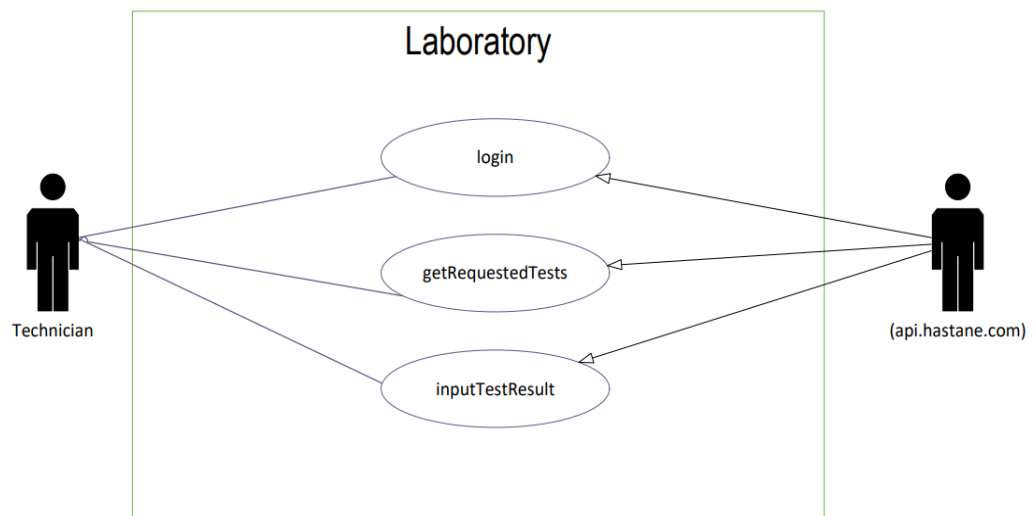
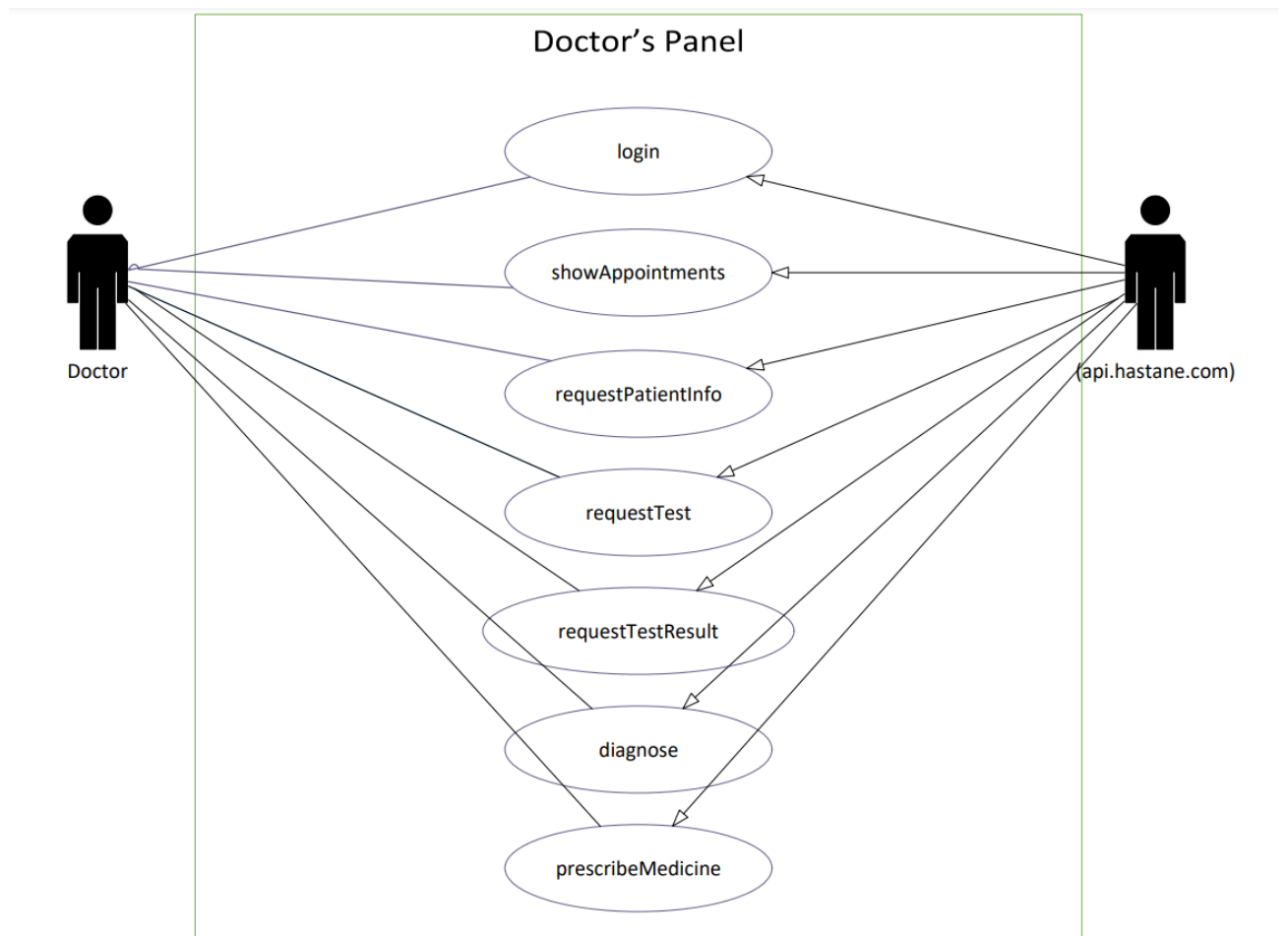
- **INITIAL BUSINESS MODEL USING USE-CASE MODELING**

## A BRIEF DESCRIPTION OF USE CASE

A use case diagram is a way to summarize details of a system and the users within that system. It is generally shown as a graphic depiction of interactions among different elements in a system. Use case diagrams will specify the events in a system and how those events flow, however, use case diagram does not describe how those events are implemented.







- **FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS**

## **FUNCTIONAL REQUIREMENTS;**

A Functional Requirement is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements in Software Engineering are also called Functional Specification.

FUNCTIONAL REQUIREMENT NO	FUNCTIONAL REQUIREMENTS'
DESCRIPTIONS	
1	THE PLATFORM SHOULD BE ABLE TO DO REGISTER A PATIENT TO DATABASE
2	PATIENTS, DOCTORS, AND LABORATORY TECHNICIAN SHOULD BE ABLE TO DO LOG IN TO THE PLATFORM
3	THE PATIENT SHOULD BE ABLE TO MAKE AN APPOINTMENT
4	THE PLATFORM SHOULD BE ABLE TO INSERT APPOINTMENTS TO THE DATABASE
5	THE DOCTOR SHOULD BE ABLE TO VIEW APPOINTMENTS, REQUEST TEST AND WRITE PRESCRIPTIONS.
6	THE TEST RESULTS SHOULD BE ABLE TO INSERT INTO THE SYSTEM BY THE LABORATORY TECHNICIAN

## NON-FUNCTIONAL REQUIREMENTS;

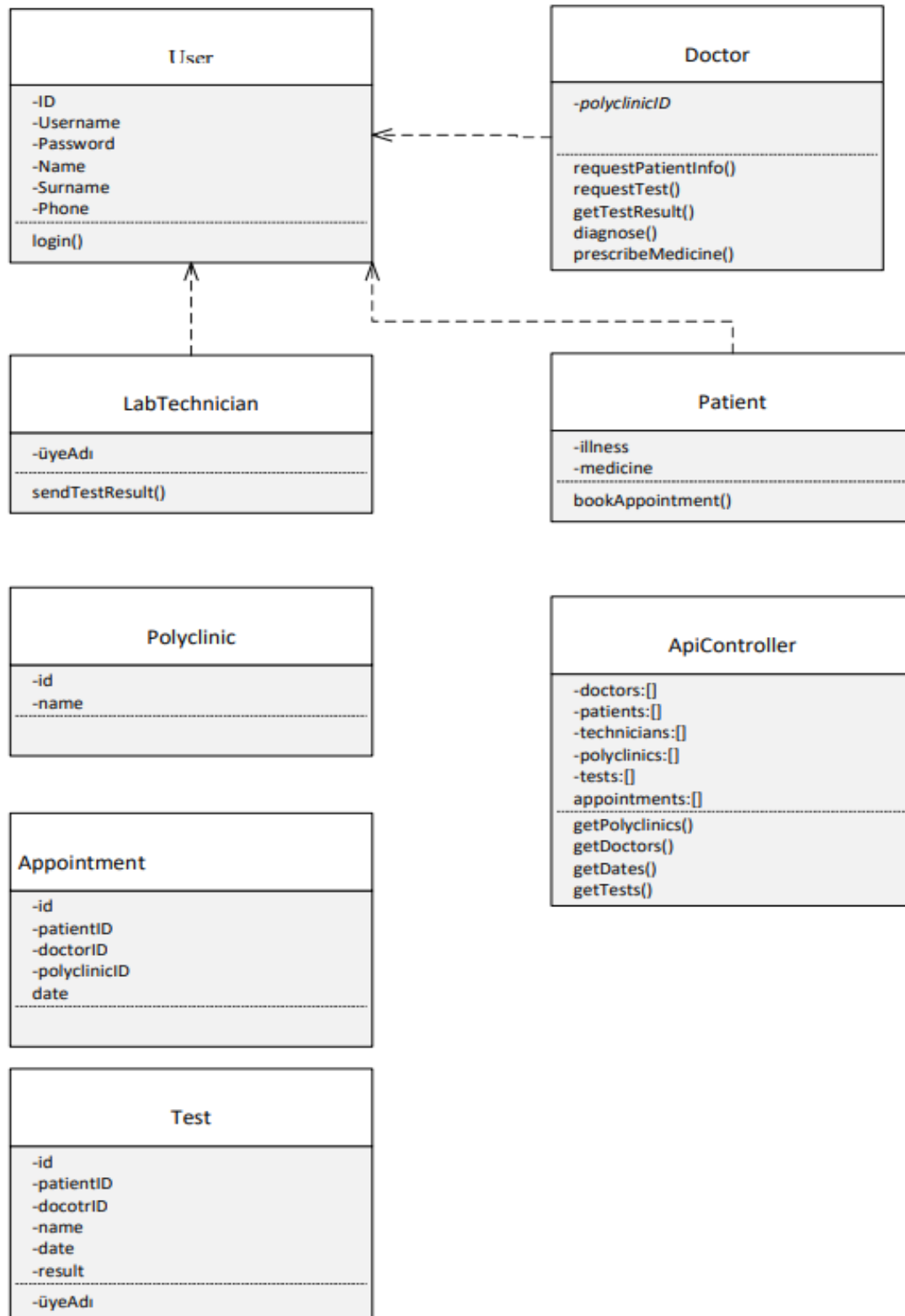
Nonfunctional requirement specifies properties of software product itself such as;

- Usability requirement
- Serviceability requirement
- Manageability requirement
- Recoverability requirement
- Security requirement
- Data Integrity requirement
- Capacity requirement
- Availability requirement
- Scalability requirement
- Interoperability requirement
- Reliability requirement
- Maintainability requirement
- Regulatory requirement
- Environmental requirement

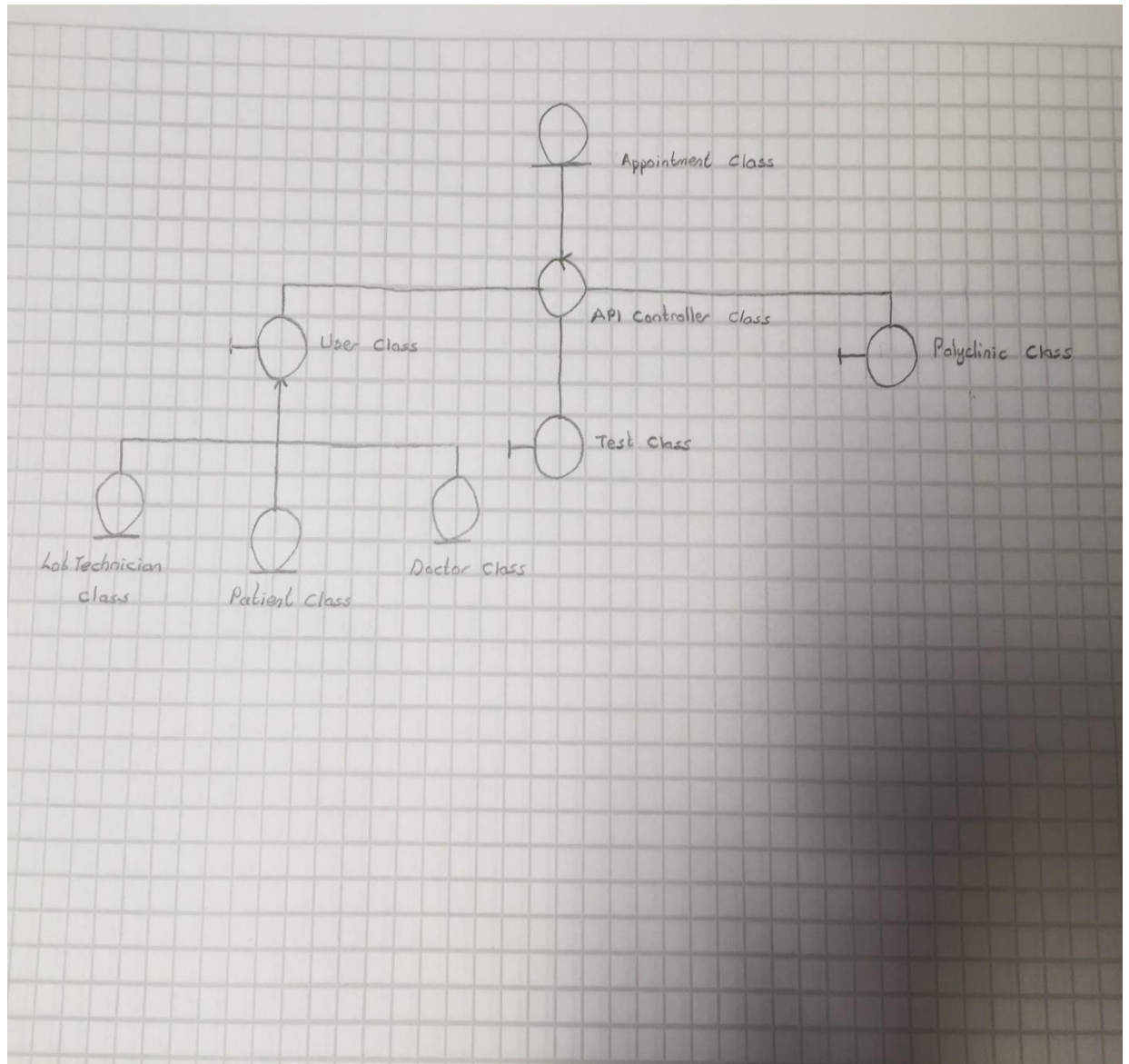
NON-FUNCTIONAL REQUIREMENT NO	NON-FUNCTIONAL REQUIREMENTS' DESCRIPTIONS
1	THE PLATFORM SHOULD BE ABLE TO PERFORM FAST
2	THE USER SHOULD BE ABLE TO NOT MISUSE BECAUSE OF DESIGN
3	THE HOSPITAL PROGRAM SHOULD BE SECURE FROM HACKING OR ANOTHER THREATS
4	PROGRAM DATABASE'S CAPACITY SHOULD BE ENOUGH FOR ALL APPOINTMENT AND USERS
5	THE PROGRAM SHOULD BE ABLE TO BE ACCESSED EVERYWHERE LIKE MOBILE
6	THE PROGRAM SHOULD NOT BE ABLE TO SHUT DOWN INCESSANTLY
7	MAINTENANCE SHOULD BE EASY
8	THE PROGRAM SHOULD BE EXECUTED IN HARMONY WITH ALL USERS

# TEST WORKFLOW

- UML CLASS DIAGRAM FOR THE PROJECT AND METHODS WITH ACCESS LEVELS



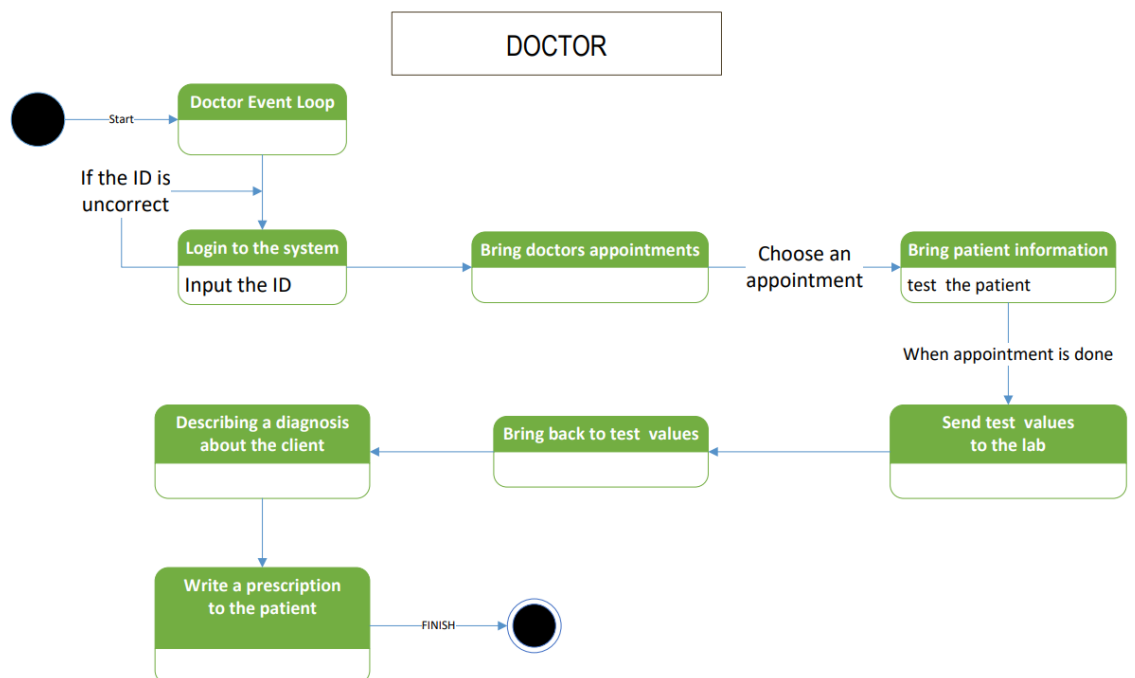
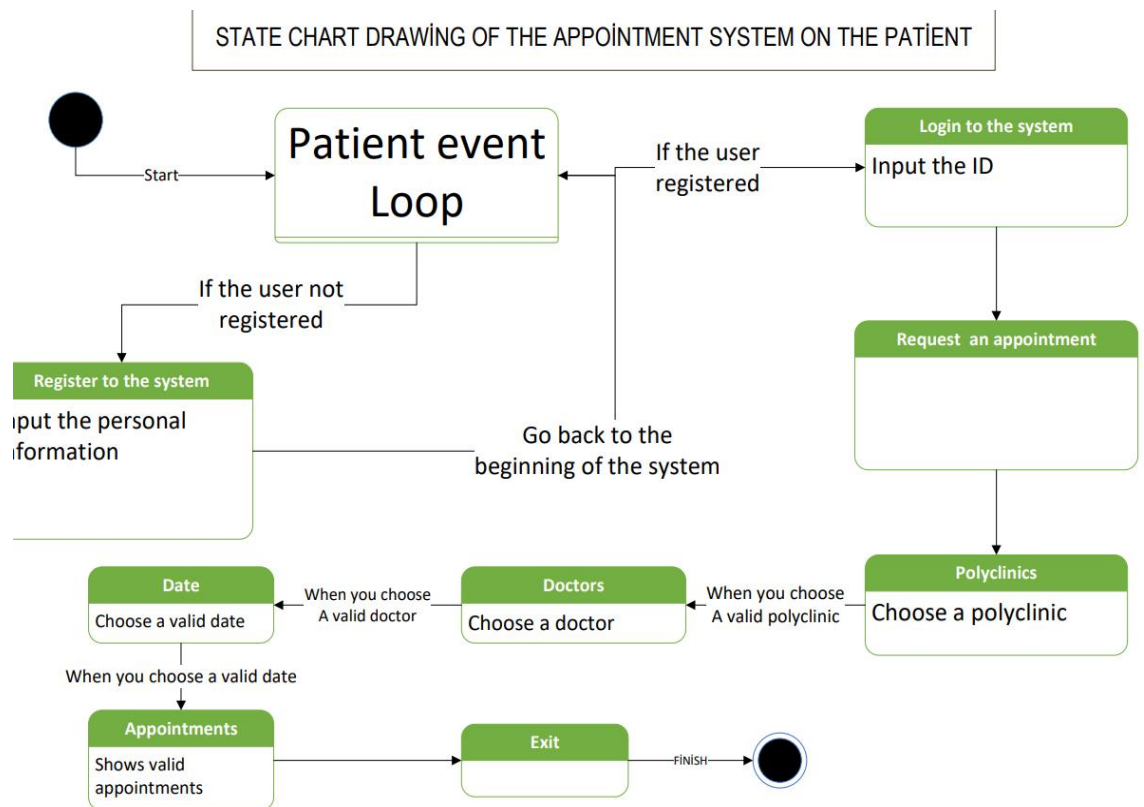
- **DETERMINING ENTITY, BOUNDARY AND CONTROL CLASS WITH SHEME**

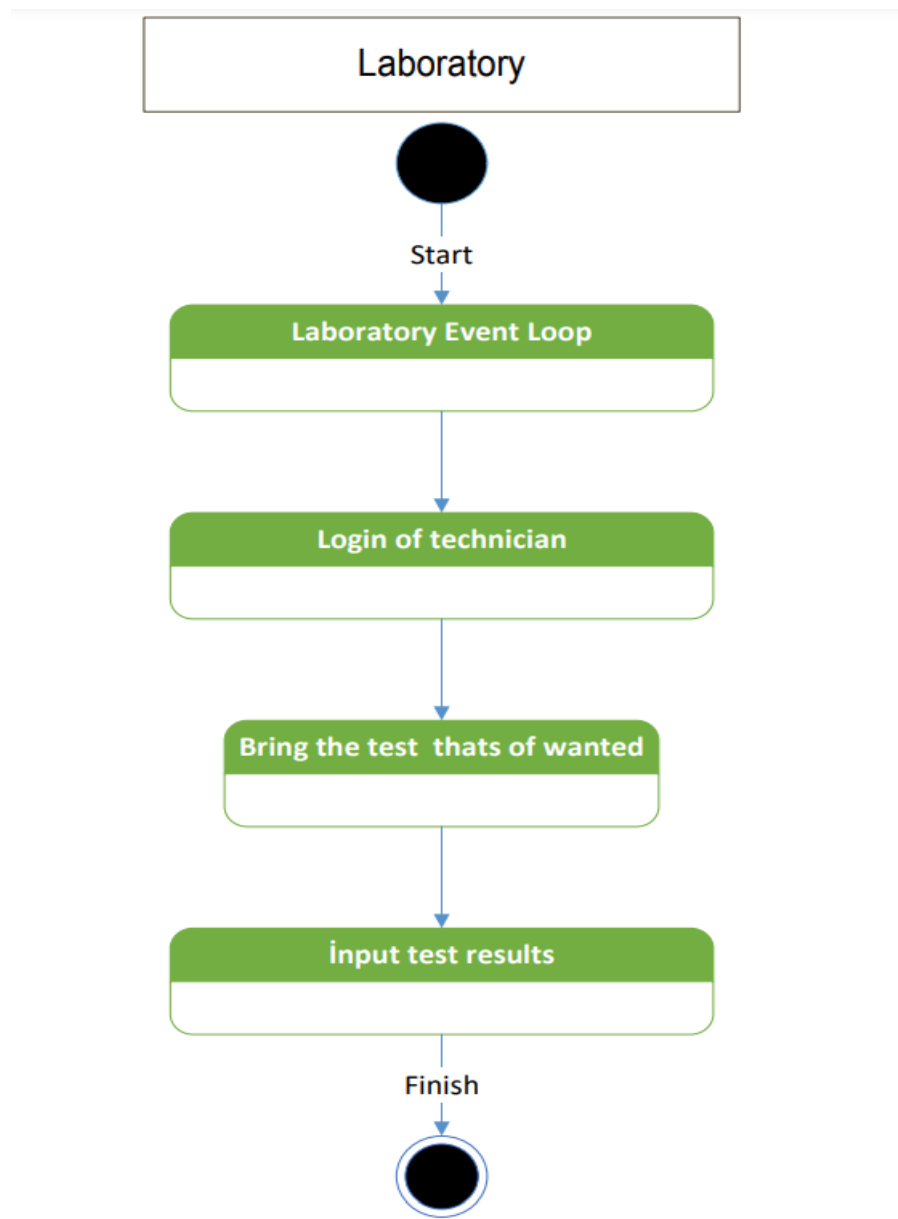


## CRC(CLASS RESPONSIBILITY COLLABORATION) CARDS OF THE PROJECT

<b>Patient Class</b> -Create an appointment. -Get test result.(Optional) -Appointment Class -Doctor Class -Polyclinic Class	<b>Doctor Class</b> -Review Appointments and Check-Up Patients in order. -Ask for a laboratory test. -Diagnose. -Prescribe medicine. -Appointment Class -Patients Class -Polyclinic Class
<b>Lab Technician Class</b> -Get requested tests(examinations). -Add results to database. -Doctor Class -Patient Class	<b>Appointment Class</b> -Keep in database the created appointments with the dates. -Doctor Class -Patient Class -Polyclinic Class
<b>User Class</b> -Sign up with your personal data -Login the Hospital System -Doctor Class -Patient Class -Lab Technician Class	<b>Polyclinic Class</b> -Doctor Class -Patient Class -Appointment Class
<b>Test Class</b> -Combines patient information and results with dates -Doctor Class -Patient Class -Lab Technician Class -Appointment Class	<b>Api Controller Class</b> -Retrieves the requested data from database -Doctor Class -Patient Class -User Class -Polyclinic Class -Lab Technician Class -Appointment Class

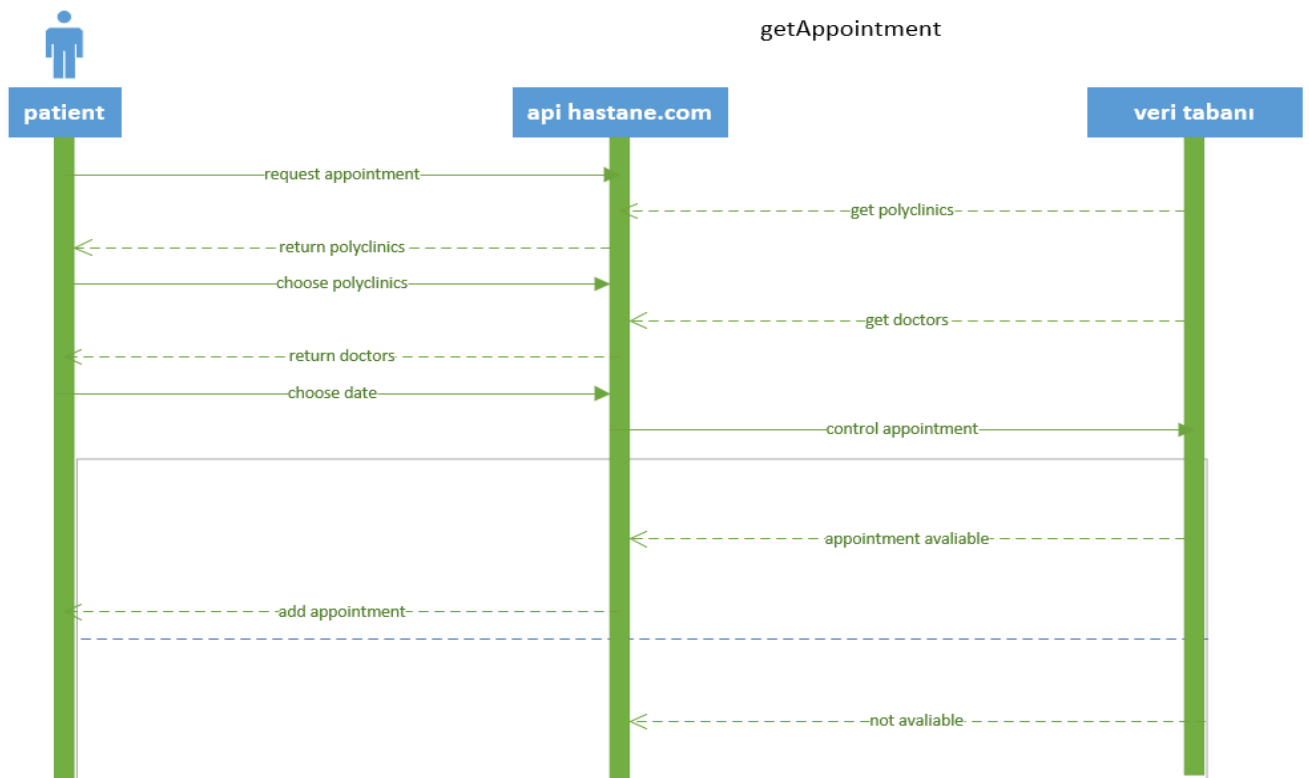
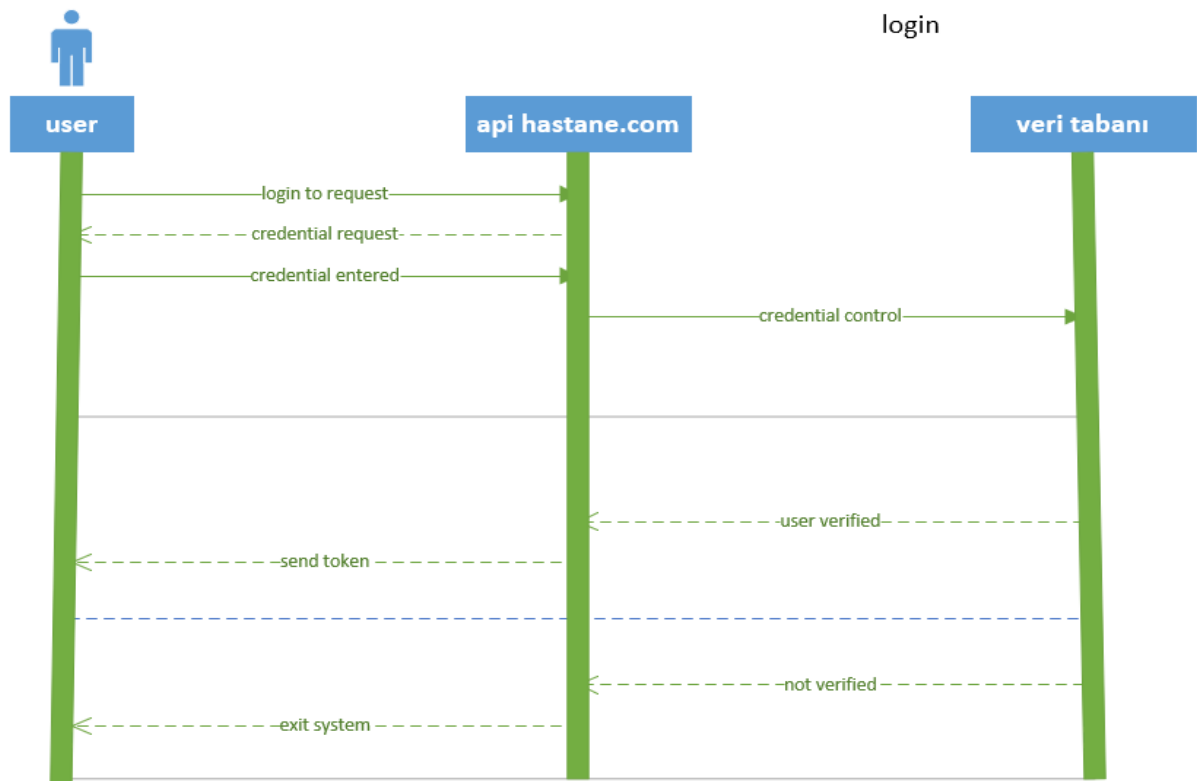
- STATECHART DRAWING OF EACH CLASS

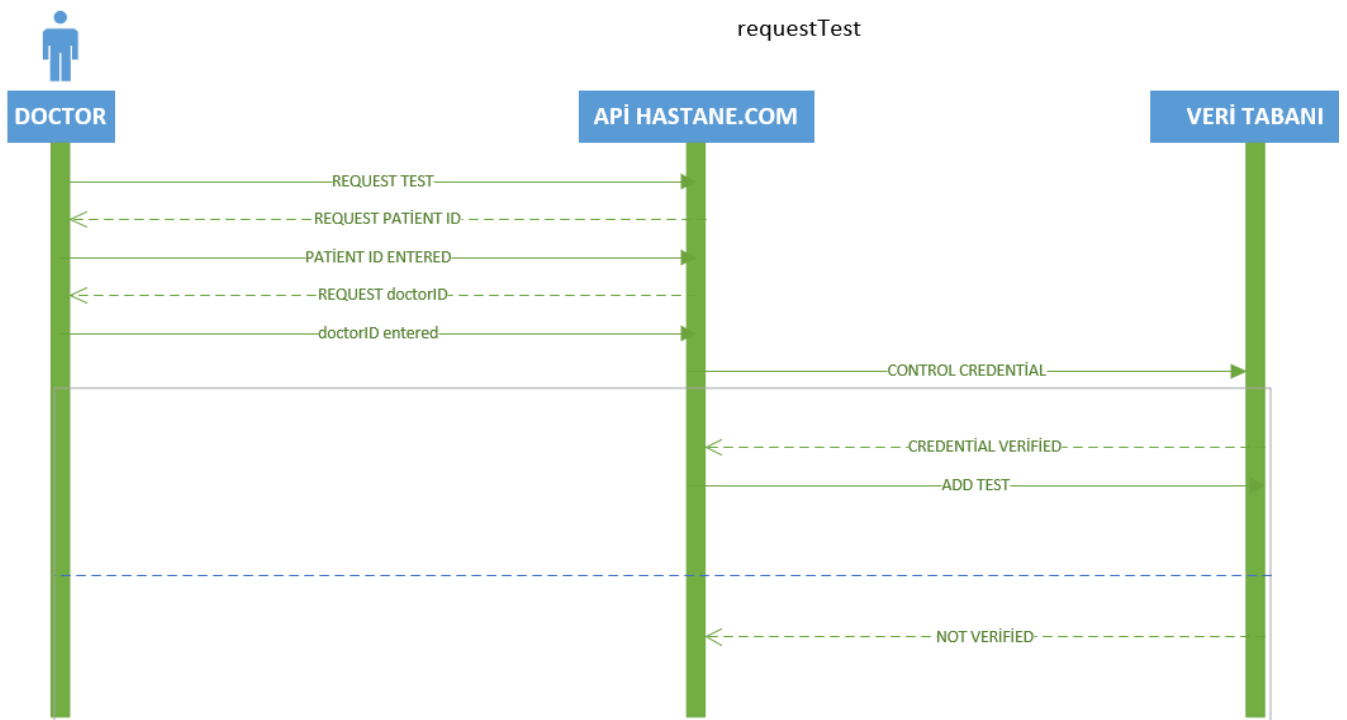




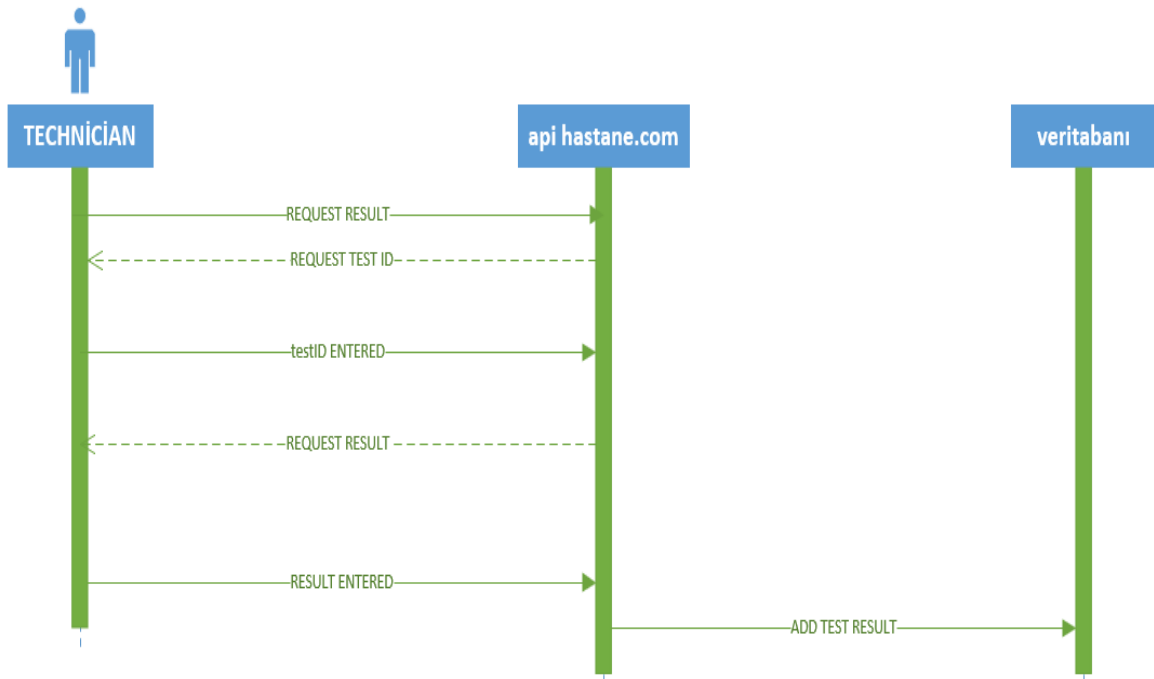


- **SEQUENCE DIAGRAMS FOR EACH SCENARIO**

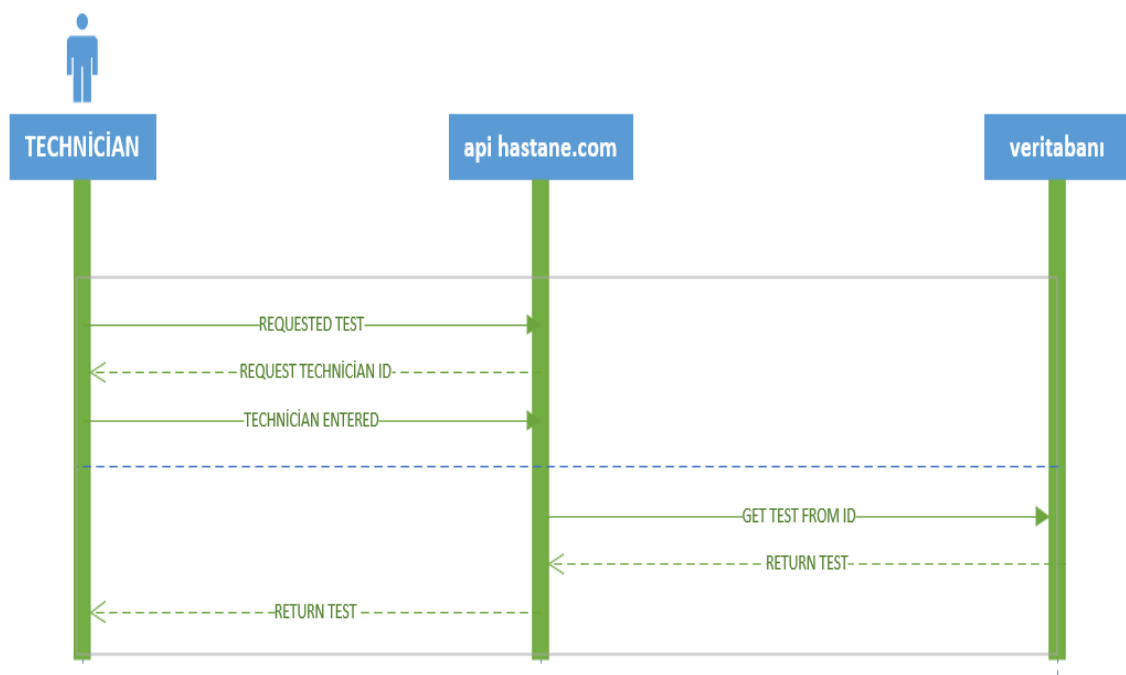




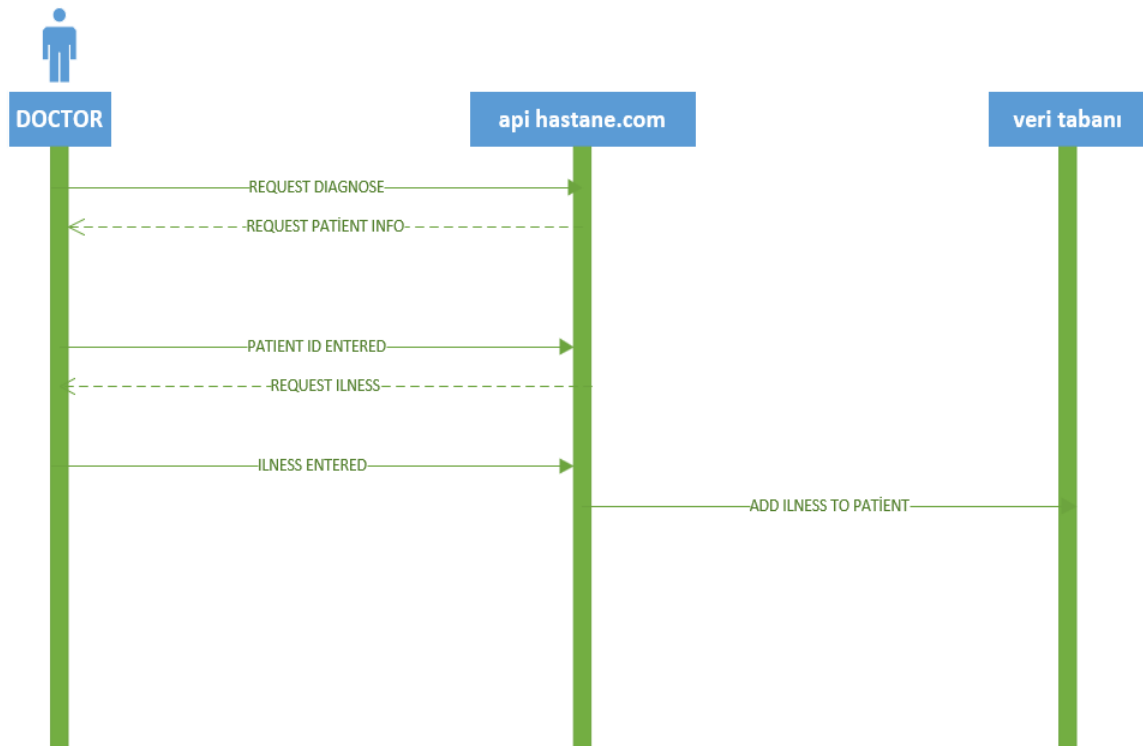
### sendTestResult



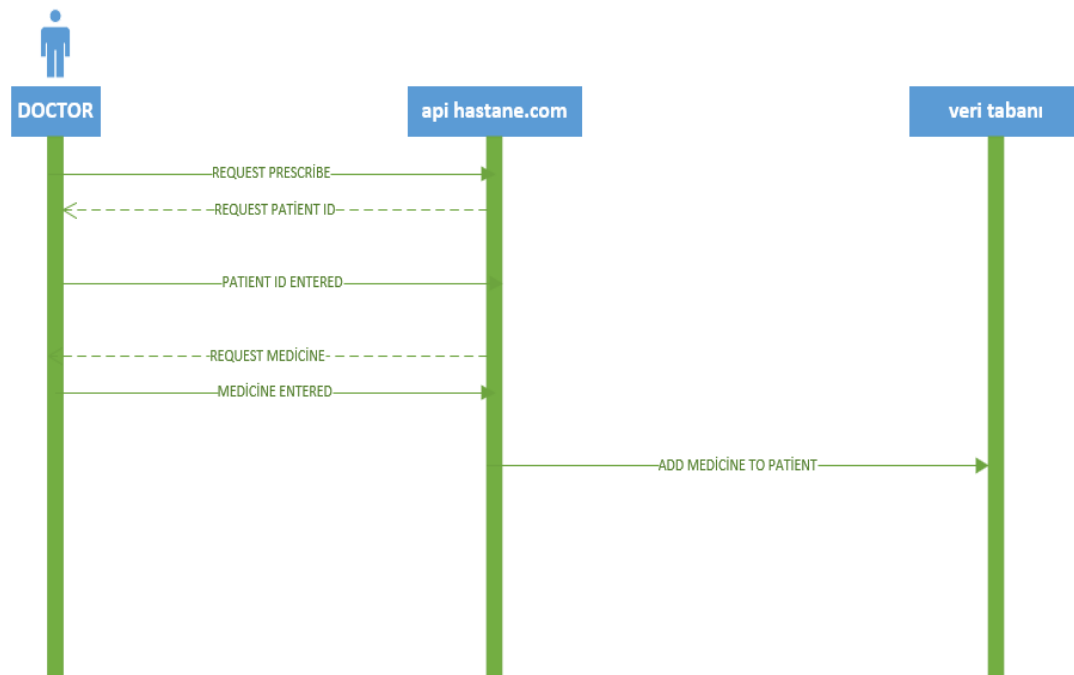
### showRequestedTest



## diagnose



## prescribeMedicine



# SOFTWARE PROJECT MANAGEMENT PLAN

## 1 Overview

### 1.1 Project Summary.

Hospital Information System; It aims to save time for hospital members, to make requests quickly and accurately, and to enable hospital staff to do their jobs more easily, regularly and accurately.

#### 1.1.1 Purpose, Scope, and Objectives.

The purpose of this project is to develop a software product that will help patients and patients, doctors and technicians speed up the processes in a coordinated manner during the patient's treatment process. The product will allow the patient to make an appointment, the doctor to request the patient's tests, the laboratory staff to add the results of the tests to the system, and the doctor to write prescriptions and medicine. The product will take necessary actions in these areas.

#### 1.1.2 Assumptions and Constraints.

Constraints contains the following.

- The hospital executives must be satiated by the product

- The deadline must be met.

- The budget constraint must be met.

- The product must be reliable.

- The architecture must be open so that additional functionality may be added later.

- The product must be user-friendly.

#### 1.1.3 Project Deliverables.

- The complete product, including user manual, will be delivered up to the deadline.

#### 1.1.4 Schedule and Budget Summary.

- Requirements workflow (one team members \$0)

- Test workflow (two team members \$0)

- Design workflow (1 team members \$0)

- Implementation workflow (0 team members \$0)

- Testing workflow (0 team members, \$0)

- The total development time is up to deadline, and the total internal cost is 0dollar.

## 1.2 Evolution of the Project Management Plan.

All changes in the project management plan must be agreed to library executives before they are implemented. All changes should be documented to keep the project management plan correct and up to date.

## 2 Reference Materials.

[https://www.researchgate.net/figure/Questionnaire-on-patient-satisfaction-administered-at-hospital-discharge-Values-are\\_tbl4\\_42440899](https://www.researchgate.net/figure/Questionnaire-on-patient-satisfaction-administered-at-hospital-discharge-Values-are_tbl4_42440899)

[https://www.youtube.com/@lucid\\_software](https://www.youtube.com/@lucid_software)

[https://www.tutorialspoint.com/uml/uml\\_statechart\\_diagram.htm](https://www.tutorialspoint.com/uml/uml_statechart_diagram.htm)

Visio

### 2.1 Process Model

The project uses an object-oriented design methodology based on the Objective lifecycle process and uses UML for software development. The development process is organized in various events. The members of the project are organized into teams. At the end of each event up to and including testing, each team submits documents describing the success of the event. Certified documents produced by teams are considered work products and are part of software documentation. The activities are described in the following sections.

#### 2.1.1 Project Planning

Project planning includes description of project tasks, activities and functions, dependencies, resource requirements.

#### 2.1.2 Requirements Test

The requirements test activity takes the problem statement and reviews it in terms of consistency, completeness and feasibility. During this activity, a set of models of the proposed system is determined by interacting with the hospital members resulting in the requirements model. The main part of the requirements model are four models: the use case model describing the complete functionality of the system, the object model, the functional model and the dynamic model.

#### 2.1.3 System Design

The purpose of the system design activity is to devise a system architecture that maps the test model to the chosen target environment. A major part of the system design phase is the design of subsystems, that is, the decomposition of the system with respect to the chosen target platform. The system design activity also refines the use cases from the test model and describes in terms of interaction diagrams how the objects interact in each specific use case.

#### 2.1.4 Test Review

Review of the software project management plan, requirements test and design.

#### 2.1.5 Client Project Review

Review of project plan, requirements test and design decisions. The client liaison will be present at the meeting

#### 2.1.6 Functional Prototype Demonstration

### **2.1.7 Object Design Phase**

The object design phase specifies the fully typed API for each subsystem. New classes are added to the test object model if necessitated by the system architecture. Attributes and methods for each object are fully typed.

### **2.1.8. System Integration Prototype Demonstration**

This activity involves the demonstration of a fully functional system prototype based on the subsystem decomposition.

### **2.1.9 Implementation**

The focus of this activity is on coding the individual objects described in the object design document.

### **2.1.10 Unit Testing**

During unit testing, test suites are designed and executed for objects or collections of objects in each subsystem. Unit testing enables the individual subsystems to be tested independent from the status of the other subsystems. The result of this activity is part of the test manual that describes how to operate the test suite and how to interpret the test results.

### **2.1.11 System Integration**

### **2.1.12 System Testing**

Structured Testing: This activity tests the main data paths throughout the system. Functional Testing: Tests the entire system for major functionality (use cases). The basis of the functional testing activity is the test guide, which is revised according to the results of the system testing phase. Alpha testing (Client Acceptance Test): The system has been tested to ensure that it passes the client acceptance criteria as defined in the project contract.

### **2.1.13 Manual Integration**

During this activity, project delivery outputs are revised. As a result, a complete document consisting of software project management plan, requirements test document, software design document, source code is presented to the customer.

## **3 Definitions and Acronyms.**

patient: a person who is receiving medical care, or who is cared for by a particular doctor or dentist when necessary.

doctor: a person with a medical degree whose job is to treat people who are ill or hurt.

polyclinic: a place that is not a hospital where doctors offer care and advice relating to many different medical conditions.

appointment: a formal arrangement to visit hospital at a particular time and.

test: the act of looking at or considering the patient carefully in order to discover the problem.

register: to put information, especially your name ,birth date etc. into an official list or record.

login: login with the information that already have beenregistered in the system.

TC card: an official card with your name, date of birth, photograph, or other information on it that proves who you are.

date: a numbered day in a month, often given with a combination of the name of the day, the month, and the year.

diagnosis: a judgment about what a particular illness or problem is, made after examining it.

prescriptions: a piece of paper on which a doctor writes the details of the medicine or drugs that someone needs.

laboratory: a room or building where patient diagnoses are examined.

sequence number: When you took an appointment you have automatically get an sequence number.

illness: the state of being physically or mentally ill.

medicine: It is a chemical substance used for the diagnosis, treatment, prevention of diseases, facilitating surgical interventions and changing some physiological events.

technician: a person whose job is to deal with diagnostics.

## 4 Project Organization.

### 4.1 External Interfaces.

All the work on this project will be performed by Tuna, Emirhan, Tugay and Ahmet Faruk.

### 4.2 Internal Structure.

The development team consists of Tuna, Emirhan, Tugay and Ahmet Faruk.

### 4.3 Roles and Responsibilities

Tunahan AKÇAY: Project Manager

Emirhan Dağdelen: Backend Developer

Tugay Coşkun: Backend Developer

Ahmet Faruk Erdem: Data Base Admin

## 5 Managerial Process Plans.

### 5.1 Start-up Plan.

#### 5.1.1 Estimation Plan.

The total developing time is foreseen up to deadline and the total internal cost dollar.

#### 5.1.2 Staffing Plan.

Tuna and Emirhan need to have at least 1 week to carry out Test Workflow, Tugay needs to have 4 days to design the project and Ahmet Faruk needs to have at least 4days to perform the respective allotted section.

#### 5.1.3 Resource Acquisition Plan.



#### 5.1.4 Project Staff Training Plan

No additional student training is needed for this project.

#### 5.2 Work Plan.

Days 1.2.3 Determined requirements workflow

DAYS 4.5. Postponed

DAYS 6.7. Produced test workflow,

Days 8.9. Produce design workflow,

##### 5.2.1 Resource Allocation.

The project had not resource allocation.

##### 5.2.2 Budget Allocation.

The project had not budget allocation.

#### 5.3 Control Plan.

#### 5.4 Risk Management Plan.

#### 5.5 Project Close-out Plan.

### 6 Technical Process Plans.

#### 6.1 Process Model.

#### 6.2 Methods, Tools, and Techniques.

The workflows have been performed with the Unified Process. The product stubs was extracted in Java.

#### 6.3 Infrastructure Plan.

The product developed using starUML and lucid chart running under windows on a personal computer.

#### 6.4 Product Acceptance Plan.

Acceptance of the product by our client has already achieved by following the steps of the Unified Process.

### 7 Supporting Process Plan

#### 7.1 Configuration Management Plan.

#### 7.2 Testing Plan.

There is no testing workflow in this project

#### 7.3 Documentation Plan.

Documentation was produced as specified in the Unified Process.

## 7.4–5 Quality Assurance Plan and Reviews and Audits Plan.

## 7.6 Problem Resolution Plan.

Any major problems faced by the team members will immediately be reported to Tuna.

## 7.7 Subcontractor Management Plan.

Did not applicated here.

## 7.8 Process Improvement Plan.

## 8. Additional Plans

The project did not need any addition plan.

## **TEST PLAN(IEEE 829-2008)**

- Introduction
- Test Items
- Approach
- Item Pass/Fail Criteria
- Suspension Criteria and Resumption Requirements
- Test Deliverables
- Testing Tasks
- Responsibilities
- Schedule

### 1. Introduction

- Summary of the items and features to be tested

### 2. Test Items

- Appointments
- Test
- Prescriptions

### 3. Approach

- General approach to testing
- specify the main activities, techniques and tools to be used to test the appointments, the test, the prescriptions
- Specify the required minimum sophistication
- Determine what techniques will be used to assess coverage
- Specify additional completion criteria
- Specify techniques to be used to monitor requirements

#### 4. Item Pass/Fail Criteria

- Specify criteria to use to determine whether each item passes the test

#### 5. Suspension Criteria and Resumption Requirements

- Specify the criteria to be used to make an appointment
- Specify test activities that need to be redone when the test is resumed

#### 6. Test Deliverables

- Identify deliverables: test plan, test design specifications, test case specifications, test procedure specifications, test item transfer reports, test logs, test case reports, test summary reports

#### 7. Testing Tasks

- Identify tasks necessary to prepare for and perform testing
- Identify all task interdependencies

#### 8. Responsibilities

- Identify groups responsible for directing, designing, preparing, executing, controlling, and analyzing

#### 9. Schedule


- Specify test stages
- Specify all item transfer events
- Estimate the time required to complete each test task
- Plan all test tasks and test milestones
- Specify usage times for each test source

## DESIGN WORKFLOW

### • DETERMINING THE INTERIOR DETAILS OF THE CLASSES

The interior design of the classes has already been set as you have seen in code-stubs and uml chart(i have set all the access levels as private because of the information hiding in addition, we can access the objects using set and get methods)

## • PSEUDO-CODES OF THE PROJECT



```
public class ApiController
{
    public List<Patient> Patients { get; set; }
    public List<Doctor> Doctors { get; set; }
    public List<Appointment> Appointments { get; set; }
    public List<LabTechnician> Technicians { get; set; }
    public List<Test> Test { get; set; }
    public List<Polyclinic> Polyclinics { get; set; }
}
```

codesnap.dev



```
public class Patient : User
{
    public string Illness { get; set; }
    public string Medicine { get; set; }

    public bool BookAppointment(Patient patient, Doctor doctor, DateTime dateTime)
    {
        // Check if the appointment is available
        bool isAvailable = CheckAvailability(doctor, dateTime);

        if (isAvailable)
        {
            // Book the appointment
            Appointment appointment = new Appointment(patient, doctor, dateTime);
            appointments.Add(appointment);
            Console.WriteLine("Appointment booked successfully!");
            return true;
        }
        else
        {
            Console.WriteLine("Sorry, that time slot is not available.");
            return false;
        }
    }

    private bool CheckAvailability(Doctor doctor, DateTime dateTime)
    {
        // Check if the doctor is available at the specified time.
        // This could involve checking a database or making a
        // network request to a scheduling service.

        // For the purposes of this example, we'll just pretend
        // that the doctor is always available.
        return true;
    }
}
```

codesnap.dev


```
public class User
{
    public int Id { get; set; }
    public string Username { get; set; }
    public string Password { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string PhoneNumber { get; set; }

    public bool Login(string username, string password)
    {
        // Check if the provided username and password are correct
        bool isValid = CheckCredentials(username, password);

        if (isValid)
        {
            Console.WriteLine("Login successful!");
            return true;
        }
        else
        {
            Console.WriteLine("Invalid username or password. Login failed.");
            return false;
        }
    }

    private bool CheckCredentials(string username, string password)
    {
        // Perform the necessary checks to determine if the provided
        // username and password are correct. This could involve
        // checking a database, making a network request, etc.

        // For the purposes of this example, we'll just pretend that
        // the username is "user" and the password is "pass".
        return username == "user" && password == "pass";
    }
}
```



```
public class Doctor : User
{
    public int PolyclinicID { get; set; }
    public void requestPatientInfo()
    {


    }
    public void requestTest()
    {

    }
    public void getTestResult()
    {

    }
    public void diagnose()
    {

    }
    public void prescribeMedicine()
    {

    }
}
```




```
public class Appointment
{
    public int AppointmentID { get; set; }
    public int PatientID { get; set; }
    public int DoctorID { get; set; }
    public int PolyclinicID { get; set; }
    public string DateTime {get; set; }

}

public class Polyclinic
{
    public int ID { get; set; }
    public string Name { get; set; }

}
```



```
public class LabTechnician:User
{
    public void sendTestResult()
    {

    }
}
public class Test
{
    public int ID { get; set; }
    public string Name { get; set; }
    public int PatientID { get; set;}
    public int DoctorID { get; set; }
    public string date { get; set; }
    public string result { get; set; }
}
```

codesnap.dev



# THE DESIGN OF FORM INTERFACES


## Hoşgeldiniz


Şifreni Unuttum?

Giriş Yap >


> e-Devlet ile giriş

> e-Nabız ile giriş






## Hastalarım

Gianluca Santiago 

### Hasta



Bernice Frederick

12/27/1995

12345678901

27

Tetkik İste

Tanı Koy

İlaç Yaz

### İstatistikler

12







Mutlu Hastalar

Bilinmiyor

37.5%

33.6%

### Sıradaki Hastalar

	Neve Gentry	10:30
	Tara Perry	11:30
	Carlos Yang	13:30
	Kezia Jennings	14:30
	Liam Rasmussen	15:30
	Dante Carey	16:30

## Randevu Al

Gianluca Santiago 

### Poliklinik Seç

#### Kardiyoloji

Doktor Sayısı: 6 Yatak Sayısı: 20 Kapasite: 250

#### Nöroloji

Doktor Sayısı: 6 Yatak Sayısı: 20 Kapasite: 250

#### KBB

Doktor Sayısı: 6 Yatak Sayısı: 20 Kapasite: 250

#### Hematoloji

Doktor Sayısı: 6 Yatak Sayısı: 20 Kapasite: 250

#### Cildiye

Doktor Sayısı: 6 Yatak Sayısı: 20 Kapasite: 250

### Doktor Seç



### Tarih Seç

09:30

10:30

11:30

13:30

14:30

15:30

16:30

## Tetkiklerim

Gianluca Santiago 

### Hasta



Bernice Frederick  
12345678901

12/27/1995  
27

Tetkik Gir

### İstatistikler



Biten Testler

37.5%

İlerleyen Testler

33.6%

### Sıradaki Tetkikler



Neve Gentry



Tara Perry



Carlos Yang



Kezia Jennings



Liam Rasmussen



Dante Carey